See Through Vehicles: Fully Occluded Vehicle Detection with Millimeter Wave Radar

Chenming He[†], Chengzhen Meng[†], Chunwang He[†], Xiaoran Fan[‡], Beibei Wang[◊],
Yubo Yan[†] [*], Yanyong Zhang[†] [◊] [*]
[†]University of Science and Technology of China, [‡]Google,
[◊]Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
{hechenming, czmeng, hechunwang}@mail.ustc.edu.cn, vanxf@google.com, wbb@iai.ustc.edu.cn, {yuboyan, yanyongz}@ustc.edu.cn

ABSTRACT

A crucial task in autonomous driving is to continuously detect nearby vehicles. Problems thus arise when a vehicle is occluded and becomes "unseeable", which may lead to accidents. In this study, we develop mmOVD, a system that can detect fully occluded vehicles by involving millimeter-wave radars to capture the ground-reflected signals passing beneath the blocking vehicle's chassis. The foremost challenge here is coping with ghost points caused by frequent multipath reflections, which highly resemble the true points. We devise a set of features that can efficiently distinguish the ghost points by exploiting the neighbor points' spatial and velocity distributions. We also design a cumulative clustering algorithm to effectively aggregate the unstable ground-reflected radar points over consecutive frames to derive the bounding boxes of the vehicles.

We have evaluated mmOVD in both controlled environments and real-world environments. In an underground garage and two campus roads, we conducted controlled experiments in 56 scenes with 8 vehicles, including a minibus and a motorcycle. Our system accurately detects occluded vehicles for the first time, with a 91.1% F1 score for occluded vehicle detection and a 100% success rate for occlusion event detection. More importantly, we drove 324km on crowded roads at a speed up to 70km per hour and show we could achieve an occlusion detection success rate of 92% and a low false alarm rate of 4% with only 10% of the training data in complex real-world environments.

[*]Corresponding authors

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM MobiCom '24, November 18–22, 2024, Washington D.C., DC, USA

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0489-5/24/09
https://doi.org/10.1145/3636534.3690658

CCS CONCEPTS

• **Hardware → Sensor applications and deployments**; **Wireless devices**; • **Computing methodologies → Object detection**.

KEYWORDS

Millimeter-wave Radar Sensing, Autonomous Driving

ACM Reference Format:
Chenming He, Chengzhen Meng, Chunwang He, Xiaoran Fan, Beibei Wang, Yubo Yan, Yanyong Zhang. 2024. See Through Vehicles: Fully Occluded Vehicle Detection with Millimeter Wave Radar. In *International Conference On Mobile Computing And Networking (ACM MobiCom '24), November 18–22, 2024, Washington D.C., DC, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3636534.3690658

1 INTRODUCTION

In autonomous/assisted driving, it is necessary to continuously detect and track neighboring vehicles. However, frequent occlusions between vehicles can cause the target vehicle to disappear, disrupting the tracking process [16, 53]. Specifically, when multiple vehicles are driving in parallel, a blind zone appears in the visual field of typical vehicle-mounted sensors (e.g., cameras or LiDARs), as illustrated in Figure 1. Such a blind zone can result in the disappearance of a target, impeding target tracking and path planning, and potentially causing hazards during maneuvers such as overtakes, lane changes, or turns. According to the latest data in the Traffic Safety Facts of the National Highway Traffic Safety Administration (NHTSA) of the United States, about 1,584 fatal accidents were attributed to blind zones in the US in 2021 [23]. Therefore, timely and accurate vehicle detection in occluded scenarios holds tremendous significance in enhancing the safety of autonomous/assisted driving systems.

Currently, there have been several attempts to address this challenge, all of which assume popular car-installed sensors such as cameras or LiDAR sensors. Some studies use camera-based or LiDAR-based detection algorithms to segment potentially occluded regions of interest and then use a combination of prior knowledge, deep learning, and other methods to identify the presence of occlusion [53]. Some

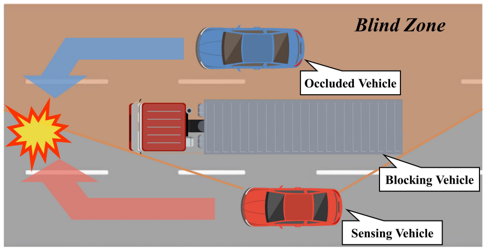


Figure 1: Illustration of the detection blind zone caused by the occlusion problem

studies take advantage of a vehicle's distinctive appearance, such as the windscreen, to detect vehicles in occlusion [39]. However, these methods can only help in *partially*-occluded cases with a large portion of the occluded car in sight. To the best of our knowledge, there has been no system that can detect fully or mostly occluded targets.

To address this void, we propose to utilize a less explored sensor, millimeter-wave radar [3, 32] (mmWave radar, in short) for occlusion detection. The mmWave radar, due to its high ranging accuracy, compact size, and low power consumption, has recently received an increasing amount of attention in the research community for detection, localization and communication [6, 7, 13, 21, 33, 50, 51]. Many public datasets for autonomous driving (e.g., nuScences[5]) have also adopted mmWave radars as one of the supplementary sensors. Millimeter waves, operating at GHz levels, reflect off surfaces as they propagate, resulting in changes to the propagation path. This is known as the multi-path effect [18, 26, 40, 41], which has been well-studied for other frequency bands(e.g., [11, 30, 34] for WiFi, and [42–44] for RFID) by the community.

In our setting, the multi-path effect can be viewed as a unique advantage of millimeter waves, enabling us to achieve non-line-of-sight detection. For example, radar signals that reflect off the ground [29] under the blocking vehicle generate a *mirror target*, which can be used to detect the occluded vehicle (as shown in Figure 2). This feature of millimeter waves allows us to detect objects in the occlusion zone that are not visible with traditional line-of-sight sensors. However, the multi-path effect also causes problems. Besides the ground, millimeter waves reflect off other moving vehicles as well. As such, when vehicles are driving close to each other, the waves can reflect multiple times between them, resulting in severe *ghost* signals [25, 27] which may lead to false alarms. The ghost signals generate ghost points which exhibit similar signal intensity and location to the true points from occluded vehicles, making them hard to discriminate. Additionally, since the non-line-of-sight detection relies on the signals after an indirect reflection path, the resulting signals are weak and unstable, making it difficult to detect the occluded vehicles accurately.

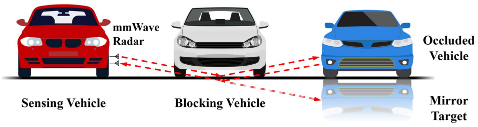


Figure 2: Occluded vehicle detection through ground-reflected mmWave signals.

This paper presents mmOVD, an occluded vehicle detection system based on mmWave radar. Initially, we generate radar point clouds using mmWave signals. The raw point clouds contain a large amount of noise, including the challenging ghost points. To eliminate the ghost points, the key lies in a well-designed feature set that can effectively discriminate them from the true points. Existing works [15, 25] focus on an individual point's properties, such as its signal strength and Doppler velocity, but we find their discriminative power is limited in our case. Instead, we examine a point's neighborhood and find that these two types of points have drastically different spatial and velocity distributions in the neighborhood. For example, a true point's neighbor point density is considerably higher than that of a ghost point; halfway between the radar and a ghost point, we often observe a cluster of points while this does not apply to true points; a true point's neighbor velocities (not the Doppler velocity) often exhibit similar trends while a ghost point's neighbor velocities are more chaotic. Based on these observations, we have carefully devised a set of *neighborhood behavior*-based point features. In addition, we can further enhance these features' robustness by exploring the temporal and spatial dimensions – e.g., a true point also has more neighbors in the previous frame or the neighbor radar's frame.

Further, to combat the challenge caused by the relatively unstable radar points from occluded vehicles, we propose a *conservative clustering, cumulative aggregation* approach. Specifically, we first perform conservative clustering to obtain point clusters that are small and possibly incomplete. We then accumulate these raw clusters from multiple frames and aggregate them into more complete clusters. In this way, we can accurately extract the bounding box of each vehicle and detect whether there is occlusion based on the spatial relationship of these boxes.

In summary, our work makes the following contributions:

- We have designed and prototyped an occluded vehicle detection system, mmOVD, which leverages mmWave radars to detect vehicles even in fully occlusion scenarios. To the best of our knowledge, this is the first system that can accurately detect fully occluded vehicles (including motorcycles).
- We devised a set of neighborhood spatial- and velocity distribution-based point features, which can be used to identify the ghost points in any vehicle detection/tracking system with mmWave radars. We have also designed a new detection system that can effectively cope with unstable radar

signals reflected from the occluded vehicle by adopting a conservative clustering, cumulative aggregation approach.

- We have evaluated mmOVD in great depth in controlled settings and through actual long-distance driving. In controlled settings, we conducted extensive experiments with 8 different vehicles, including a minibus and a motorcycle, in an underground garage and on two campus roads. The results show that mmOVD achieves an average accuracy of 94.2% for point classification, an F1 score of 91.1% for occluded vehicle detection, and a success rate of 100% for occlusion event detection with a false alarm rate lower than 5% in our dataset. Additionally, we drove 324km on crowded roads (most roads with a speed limit of 60km/h or 80km/h) at a speed up to 70km/h. The results demonstrate that mmOVD can achieve an occlusion detection success rate of 92% and a low false alarm rate of 4% with only 10% of the training data in complex real-world environments.

2 BACKGROUND

2.1 mmOVD: Using mmWave to Detect Occluded Targets

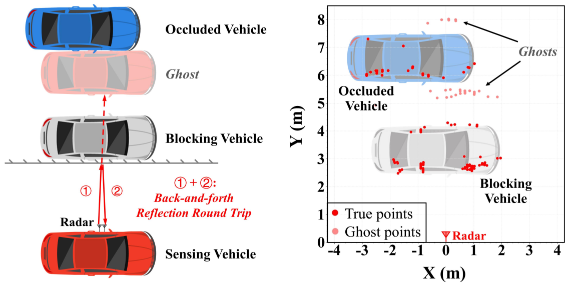
The mmWave radars have been increasingly used for detecting nearby vehicles in recent years. Typical automotive mmWave radars operate with a sequence of linear frequency-modulated continuous-wave (FMCW) signals to detect objects. The antennas on radar transmit and receive FMCW signals continuously. Then a frequency mixer and a low-pass filter are used to combine the signals and produce intermediate frequency (IF) signals. With the IF signals of different antennas, we can obtain the distance, Doppler velocity, and angle of each object through fast Fourier transform (FFT) [46].

Benefiting from the nature of electromagnetic waves, the reflection on smooth surfaces enables mmWave radar to achieve non-line-of-sight detection. This concept has been explored in other settings, including the detection of targets around corners in [1, 2, 9, 24, 52]. As signals can bounce off walls, the radar can detect objects hidden behind a corner. By taking into account the geometric structure of the walls, the locations of these hidden objects can be determined.

In this work, we utilize the reflection off the ground, which is the most common and stable reflecting surface in automotive environments, to detect occluded vehicles. This is the underlying motivation for the design of mmOVD. Nonetheless, the complexity of automotive environments has brought serious challenges to our system design.

2.2 Challenges in Designing mmOVD

Our mmOVD system has two main objectives: (1) to avoid false alarms when no occluded vehicle is present and (2) to accurately detect occluded vehicles. However, achieving these goals is challenging. By conducting motivation studies in both static and driving scenarios, we identify two main challenges as follows. Firstly, although we can detect hidden vehicles through ground reflections, interference from other reflectors in the environment complicates distinguishing between real vehicles and ghosts. Secondly, the dynamic movements of vehicles and constantly changing signal propagation paths further increase the complexity of the problem. Below we explain these challenges in more detail.



(a) Illustration of ghost phenomenon (b) A frame of radar point cloud

Figure 3: (a) Signals bounce between the vehicles, creating "ghost" vehicles. (b) Ghost points appear around true points from the occluded vehicle.

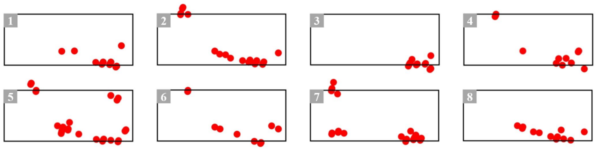


Figure 4: Radar points from an occluded vehicle in 8 consecutive frames (30 FPS), in which the black boxes represent the vehicle's bounding boxes.

False Alarm Due to Ghost Points: Vehicles are effective reflectors of mmWave signals as they are primarily made of metal and reinforced plastics. When vehicles drive closely together, millimeter waves bounce back and forth between them (Figure 3(a)). The signals after one back-and-forth reflection generate the true points, while those after two or more reflections create "ghost points", which might trigger false alarms. For instance, several ghost clusters appear (Figure 3(b)) due to multiple reflections. Ghost points typically exhibit similar signal intensity to true points and appear in locations where occluded vehicles may be present, making the elimination of ghost points an essential challenge.

Inaccurate Detection due to Unstable Radar Points: In occlusion scenarios, mmWave radar indirectly detects occluded vehicles through ground reflections, causing unstable points due to changing signal propagation paths as vehicles move. Figure 4 illustrates the radar points from an occluded vehicle over eight consecutive frames (at 30 FPS), where the black boxes represent the vehicle's bounding boxes. The radar points are rather unstable, appearing at different positions on the vehicle's body, and sometimes even forming

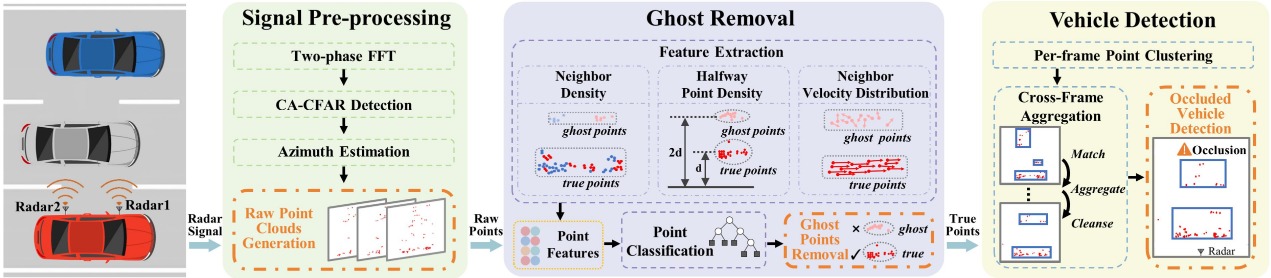


Figure 5: The mmOVD system involves the three main steps: (1) signal pre-processing, (2) ghost removal, and (3) vehicle detection.

several separate point clusters. Consequently, applying typical radar point clustering algorithms (e.g., DBSCAN [14]) can lead to significant positioning errors and even erroneously divide a single vehicle into multiple targets.

In this study, we have carefully devised algorithms to address the above challenges. By utilizing the neighborhood spatial/velocity features of radar points, we can effectively classify the points and eliminate ghosts. We also propose a cross-frame cumulative aggregation-based vehicle detection algorithm to accurately detect the occluded vehicles.

3 MMOVD DESIGN

3.1 Overview

Figure 5 depicts the overview of our mmOVD system, which consists of the following components:

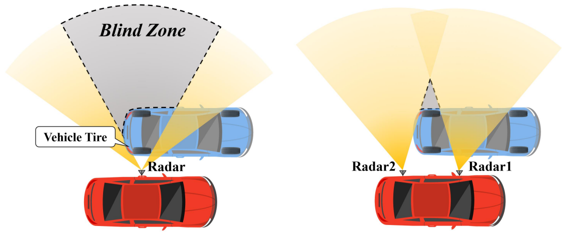
(1) **Radar Signal Pre-processing:** The received mmWave signals first go through the range measurement and velocity measurement processes. We then use a constant false alarm rate (CFAR) algorithm to generate point clouds. These points consist of true points (i.e., those from reflections from actual targets) and noise points (i.e., those include ghost points and background noises).

(2) **Neighborhood Spatial/Velocity-Based Ghost Removal:** In the raw point clouds, ghost points are the dominant noises and the most challenging to eliminate. By thoroughly examining the characteristics of radar points, we propose neighborhood spatial features and velocity distribution features to eliminate ghost points.

(3) **Cross-Frame Cumulative Aggregation-Based Vehicle Detection:** After removing ghost points, we need to detect vehicles from the point clouds. We first perform conservative clustering to get raw clusters of vehicles. We then propose a cumulative aggregation approach, which accumulate the incomplete clusters from multiple frames to achieve accurate detection for occluded vehicles.

3.2 Radar Placement and Radar Signal Pre-processing

The first consideration is the placement of radar sensors on the sensing vehicle. When relying on a single radar for detection, the propagation of mmWave signals may be significantly obstructed by the tires of the blocking vehicle, resulting in a substantial blind zone. Figure 6(a) shows such an example. By installing two (or more) radars on each side of the sensing vehicle, their fields of view can complement each other (as shown in Figure 6(b)). Our experiments indicate that installing two radars on each side of the vehicle and merging their data can significantly increase detection accuracy by over 20% compared to using only one radar. However, using three radars does not produce a significant gain, with detection accuracy only improving by 1.2% over using two radars. Hence, in designing mmOVD, we assume there are two radars on each side of the vehicle, and we can readily extend the system design to accommodate cases with more radars. Notably, to ensure the radars' fields of view are not simultaneously obstructed by the blocking vehicle's two tires, the distance between the neighboring radars should be less than the typical spacing between a vehicle's front and rear tires. Besides, since all radars have the same height and a fixed distance from each other, their data can be unified into the vehicle's center coordinate system by performing a simple translation transformation. The detailed implementation will be presented in Section 4.1 (see Figure 13).



(a) Single radar (b) Two radars

Figure 6: (a) A single radar may lead to a large blind zone since the mmWave signals can be significantly obstructed by the tires of the blocking vehicle. (b) Having two or more radars can greatly alleviate the blind zone problem due to their complementary coverage.

Next, when the radar signals are received, we perform distance measurements and velocity measurements using a two-phase Fast Fourier Transform (FFT). This process results

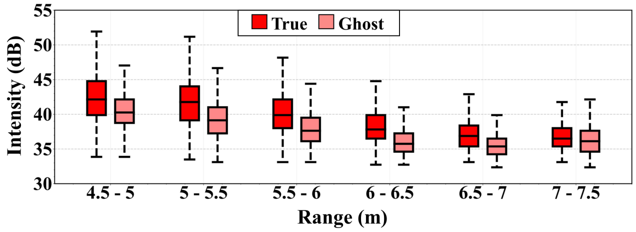


Figure 7: Intensity levels of true and ghost points are hard to be differentiated at all range intervals.

in a Range-Doppler Matrix (RDM), with each element containing the intensity of a specific range and Doppler velocity. We then apply a cell averaging constant false alarm rate (CA-CFAR) detector to detect peaks in the RDM. Finally, we estimate the azimuth angle of these peaks through the angle FFT. The output is a point cloud $\mathcal{P}$ comprising measurements of 5 parameters:

$$\mathcal{P} = \{p\},\ p = (r, v_d, \theta, SI, SNR), \tag{1}$$

where r represents the point's range, v_d represents its Doppler velocity, θ represents its azimuth angle, SI represents the signal intensity and SNR represents the signal-to-noise ratio.

3.3 Ghost Removal through Neighborhood Spatial/Velocity Feature Extraction

After the above pre-processing steps, we obtain the raw point clouds which consist of true points, ghost points, and background noise points. The background noise points, sparse and random, can be easily removed through the density-based spatial clustering of applications with noise algorithm (DBSCAN) [14], which is adopted in our system (detailed in Section 3.4). However, distinguishing between true and ghost points poses a dire challenge to the underlying design, as ghost points often exhibit similar signal intensity as the true points. Our experiments indicate a significant overlap in signal intensity between ghost points and true points at similar locations, as shown in Figure 7. Moreover, ghost points often appear in locations where occluded vehicles are located (illustrated in Figure 3), further complicating this task. In this section, we provide a detailed explanation of our ghost removal approach, which aims to accurately discriminate ghost points from true points. To clearly explain our design, we only focus on the sensing vehicle (with two radars on each side) and the blocking vehicle below.

To distinguish true points from ghost points, two distinct approaches are usually available. One approach is to use deep-learning networks to perform point classification [15]. The other approach is feature engineering, which involves extracting handcrafted features for each radar point and using these features to classify the points [25]. Considering the amount of data and computational resources required by the learning-based approaches, we adopt the second approach by thoroughly investigating the characteristics of radar points and deriving a suitable feature set for classification purposes.

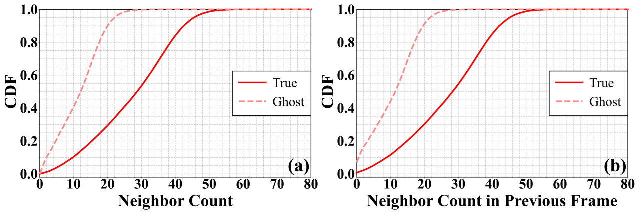


Figure 8: (a) CDF of a point's neighbor counts within a radius of 0.8m; (b) CDF of a point's neighbor counts within a radius of 0.8m in the previous frame.

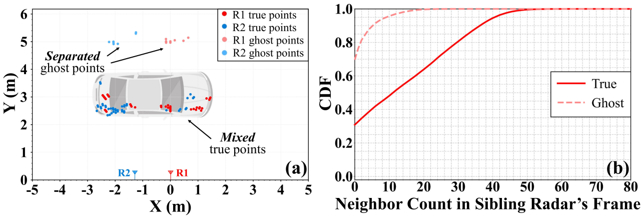


Figure 9: (a) Aggregated point clouds from the two sibling radars, in which the true point clusters are commonly mixed, while the ghost point clusters are separated. (b) CDF of a point's neighbor count in the sibling radar's corresponding frames.

3.3.1 Neighbor Density Features. We first examine the point density within a true/ghost point's local neighborhood. Our first observation is that true points tend to have more neighbor points than ghost points, as shown by the CDF results in Figure 8(a). This is because the mmWave signal energy fades quickly with a longer reflection path, resulting in many fewer ghost points which are also much more scattered.

We define our test point as $p_{test}^{r,f}$ with r denoting the radar ID that generates the point and f the frame ID. Then we define the *neighbor density* feature F_{nbr} for $p_{test}^{r,f}$ as the number of neighbor points within the distance threshold:

$$F_{nbr}^{r,f} = \left|\{p \in \mathcal{P} \mid dist(p, p_{test}^{r,f}) < T_{nbr}\}\right|, \tag{2}$$

where $\mathcal{P}$ is the radar point cloud, and T_{nbr} is the neighbor distance threshold. Next, we enhance this feature by incorporating additional temporal and spatial dimensions, generating a suite of neighbor density features. Its complete neighbor density features include:

- The neighbor density feature $F_{nbr}^{r,f}$. The first feature is the test point's neighbor density feature value in the current frame f from the same radar r;
- The history neighbor density feature $F_{nbr}^{r,f-1}$. The second feature calculates the neighbor density around the test point's location in the previous frame (with frame ID $f-1$). The rationale here is that a real vehicle appears in similar positions in two consecutive frames which generate dense points in both frames. Thus, if a test point has dense neighbors in consecutive

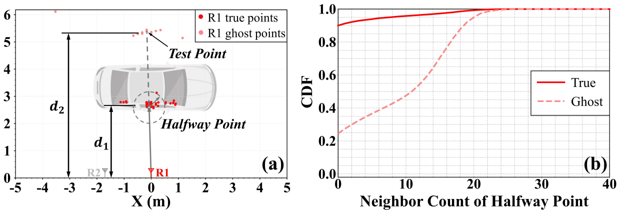


Figure 10: (a) A cluster of points commonly appears halfway between the radar and ghost points ($d_2 \approx 2d_1$). (b) CDF of halfway point's neighbor counts within a radius of 0.8m.

frames, its likelihood of being a true point increases. Figure 8(b) shows our experimental evidence.

- The sibling sensor's neighbor density feature $F_{nbr}^{\widetilde{r},f}$. The third feature calculates the neighbor density around the test point's location in frame f from the sibling radar (we assume there are two radars on the same side of the sensing car) with radar ID $\widetilde{r}$. The rationale here is that if a test point has dense neighbors in both radars' frames, its likelihood of being a true point increases. As illustrated in Figure 9(a), if we merge the two frames from both radars (with proper spatial calibration), we find that the true points from the two radars tend to cluster together, while their ghost points are further apart. As a result, the difference between the neighbor densities in the sibling radar's frames is even more pronounced, as shown in Figure 9(b).

3.3.2 Halfway Point Density Feature. We next examine the point distributions in areas beyond the ghost point's local neighborhood and find that there is usually a cluster of points halfway between the radar and the ghost point. That is, a ghost point's halfway point density is higher than a true point's halfway point density. We illustrate this pattern in Figure 10(a). In our case, the halfway location between the radar and the ghost point coincides with the blocking car's near-radar side. Besides, the CDF results depicted in Figure 10(b) also support this observation.

Below we explain this observation. In our case, the ghost points result from signal reflections between the sensing car and the blocking car (as shown in Figure 3). When the radar receives the reflected signal after a single round trip between the radar and the blocking car's near radar side (the reflecting surface), it generates the true points. When the radar receives reflected signals after two or more round trips, it generates the ghost points. When we focus on those ghost points from two round-trip reflections, we find their signal propagation distance is twice of the aforementioned true points. We can find a cluster of true points around halfway between the radar and the ghost points, hence the halfway point density feature. Here, we ignore those ghost points generated after more than two round trips because their propagation paths are too complex to analyze.

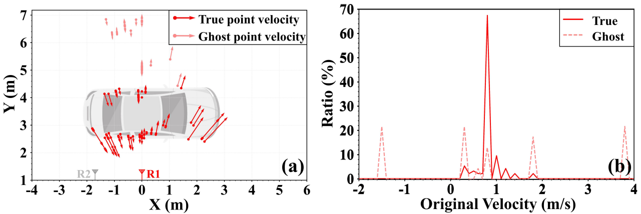


Figure 11: (a) Doppler velocities of radar points, with arrows representing direction and magnitude. (b) Original velocity distribution of the points from a real vehicle (red solid line) and a ghost (pink dashed line).

We define the halfway point density feature F_{half} as:

$$F_{half} = \left|\{p \in \mathcal{P} \mid dist(p, p_{half} < T_{nbr}\}\right|, \tag{3}$$

where the p_{half} represents the halfway point between the test point and the radar.

3.3.3 Neighbor Velocity Distribution Feature. After exploring the point density distribution as the discriminating factor, we next explore the dynamic aspect of points: velocity. To be more precise, we leverage the Doppler velocity of radar points [18] and differentiate between true points and ghost points by exploring differences in their velocity variation patterns. The mmWave radar detects an object's Doppler velocity, which represents the radial velocity (as shown in Figure 11(a)), by calculating the phase differences $\Delta\Phi$ of adjacent chirps in the reflected signals [20]. Given the wavelength λ of mmWave signals and the time interval T_c between chirps, the Doppler velocity v_d can be calculated as: $v_d = \frac{4\pi T_c}{\lambda \Delta\Phi}$. Notably, Doppler velocities of radar points can't be directly used for point discrimination. Instead, in the case of vehicles driving side by side, we can accurately reconstruct the original velocity based on the Doppler velocity v_d and azimuth angle θ of the radar points: $v = v_d / sin\theta$.

Among the points generated by the radar, the true points reflected by the same car exhibit the same velocity, i.e., the reflecting car's velocity. Conversely, ghost points, affected by accumulated phase errors from multiple reflections [49], are anticipated to exhibit different velocities. To illustrate, Figure 11(b) shows an example of the original velocity distributions of true points and ghost points, in which the x axis represents the velocity, and the y axis represents the ratio of radar points moving at different velocities. We observe that true points predominantly exhibit the same velocity, forming a main peak, while the ghost velocities are more chaotic and scattered, even with different directions.

We thus propose a feature that characterizes the test point's neighbors' velocity distribution, denoted as F_{vel}. We consider a velocity range of $(-10, 10)$m/s, with an interval of 0.1m/s. The neighbor velocity distribution vector $\boldsymbol{R_{vel}} = (r_1, ..., r_M)$ thus records the ratio of neighbor points whose velocity falls within the M ($M = 200$ in our implementation) velocity intervals. Notably, this velocity range refers to the range of velocity differences between the sensing car and neighboring cars. We believe such a velocity difference range is reasonable for real-world driving conditions, and our experiments in Section 5 have confirmed this.

In this raw form, such velocity distribution vectors are usually sparse – on average, 43.8% of the vector elements are zeros. We then employ principal component analysis (PCA) to record the velocity ratios in a reduced 3-dimensional vector. The resulting feature $\boldsymbol{F_{vel}}$ is calculated as:

$$\boldsymbol{F_{vel}} = \boldsymbol{R_{vel}}\boldsymbol{W}, \quad \boldsymbol{W} = (\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3), \tag{4}$$

where $\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3$ represent the top three principal components, each being a column vector of length N.

3.3.4 mmOVD Point Classification. We summarize our proposed radar point classification feature set in Table 1. Within the feature set, the test point is denoted as $p_{test}^{r,f}$. We set the neighbor distance threshold, denoted as T_{nbr}, to 0.8m. This value is chosen as vehicles typically maintain a lateral gap of over 1m for safety. Our experiments demonstrate that over 95% of points whose distances from each other are smaller than this threshold belong to the same vehicle or ghost.

A classifier is then utilized to differentiate between real and ghost points, using the features outlined in Table 1 as input. We have tested a variety of lightweight machine-learning-based classifiers, including decision tree [22], random forest [10], and support vector machine (SVM) [4]. We calculate the classification accuracy and computation latency of these classifiers, which are presented in Section 4.2. The points that are classified as ghosts will be eliminated, while the remaining points will be utilized for vehicle detection.

3.4 Occluded Vehicle Detection through Aggregation across Frames

After identifying the ghost points and removing them, we next cluster the clean points and estimate the bounding boxes of the detected vehicles. Then from the spatial relationship between the bounding boxes, we infer their relative positions and detect the occluded cars.

Table 1: Our mmOVD point feature set

Feature	*Description*
r, v_d, θ, SI, SNR	Basic features: range, azimuth, doppler velocity, signal intensity, and signal-to-noise ratio of $p_{test}^{r,f}$
$F_{nbr}^{r,f}$	Number of points within the distance threshold T_{nbr} to $p_{test}^{r,f}$
$F_{nbr}^{r,f-1}$	Number of points within the distance threshold T_{nbr} to $p_{test}^{r,f}$ in the previous frame
$F_{nbr}^{\tilde{r},f}$	Number of points within the distance threshold T_{nbr} to $p_{test}^{r,f}$ in the frame of the sibling radar
F_{half}	Number of points within the distance threshold T_{nbr} to the halfway point
F_{vel}	Reduced 3-dimensional velocity distribution of the points within the distance threshold T_{nbr} to $p_{test}^{r,f}$

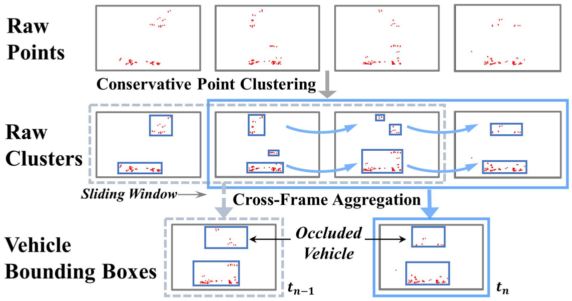


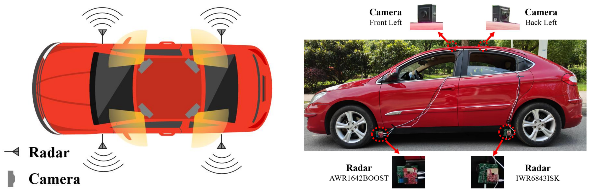
Figure 12: Illustration of the cross-frame cumulative aggregation approach

A typical approach is to first apply clustering algorithms (e.g., DBSCAN [14]) on each frame to obtain the cars’ bounding boxes, and then perform object tracking to associate the bounding boxes to refine their locations/sizes. However, we cannot directly apply this approach in our case because the radar points reflected from occluded vehicles are unstable (illustrated in Figure 4), rendering the per-frame clusters less reliable than desired. To make matters worse, these clusters also contain ghost points that are hard to remove.

To combat the challenge caused by the unstable nature of the reflected signals, we propose a *conservative clustering, cumulative aggregation* approach, which consists of two stages. First, we perform conservative clustering by employing a relatively small distance threshold, generating small and possibly incomplete clusters while avoiding aggressively grouping points from different objects into a cluster. Second, we accumulate the incomplete clusters for the same object from consecutive frames and then carefully aggregate them into a more complete cluster of each car. In this way, we can effectively cope with the unstable points from reflected signals and accurately extract the bounding box of an occluded car. Figure 12 shows the pipeline of our algorithm.

3.4.1 Per-Frame Point Clustering. We employ the DBSCAN algorithm [14], which is often used for radar point clustering under the assumption that a real target consists of points that are near each other. DBSCAN determines the physical distance between each pair of points, and points within a preset threshold are assigned to the same cluster. This threshold, denoted by $T_{cluster}$, is a critical parameter for the algorithm. We adopt a relatively small $T_{cluster}$ value, i.e., 0.5m in our current implementation. This decision considers the fact that vehicles typically maintain a lateral gap of over 1m for safety. Consequently, we use half of this shortest lateral distance as the distance threshold to ensure that point clouds from two vehicles are not clustered into one target and allows for some positioning error in the radar point cloud.

Based on the resulting clusters, we extract the center position and size of each cluster, which together constitute the bounding box of a detected object.



(a) Hardware setup (b) Prototype mmOVD

Figure 13: (a) Illustration of hardware setup of mmOVD. (b) Photo of the left side of our sensing vehicle.

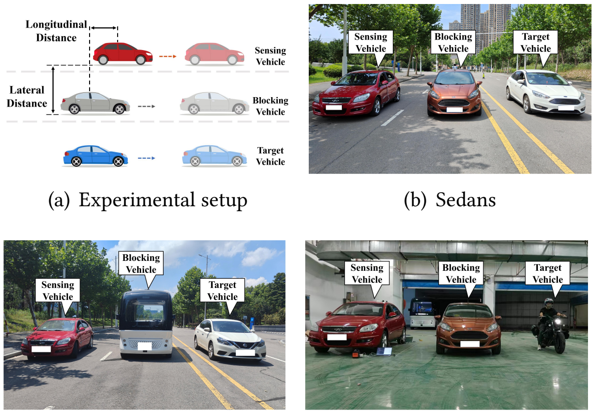
3.4.2 Cross-Frame Cumulative Cluster Aggregation. We establish a sliding window of length W, which continuously holds the clustering results of the latest W frames. We then perform cumulative cluster aggregation as follows:

- *Cluster Aggregation.* We sequentially fetch frames in the sliding window and match clusters across frames based on the center distances. The algorithm works by selecting the pair of clusters with the closest center distance and terminates when the closest center distance exceeds the threshold of $2T_{cluster}$ (i.e., 1m). This choice considers the fact that vehicles typically have a center distance of over 2m from each other. Each matched pair of clusters is considered to belong to the same object and is merged into one. This matching and merging operation is iterative by nature as we may have more than one cluster in a frame due to our conservative clustering approach. We continue to perform this action until all the frames in the aggregation window are processed.
- *Cluster Cleansing.* We then eliminate those clusters that have had no match in the previous step, assuming they are false targets. Finally, we check through the bounding boxes of all the remaining clusters. If two boxes intersect, we merge them into a single cluster.

3.4.3 Occlusion Detection. The last issue we consider is how to use the detection results to alert the driver to pay attention to occluded vehicles. We first examine the detected bounding boxes and determine whether a bounding box occludes another. Figure 12 showcases two output frames with occlusions. To deal with occasional false positive detections, we propose a simple yet effective strategy: when mmOVD detects an occluded vehicle in N consecutive frames, it triggers an alarm. We will show how to empirically select the appropriate threshold value N in Section 4.4.

4 EVALUATION THROUGH CONTROLLED EXPERIMENTS

4.1 Experimental Setup



(a) Experimental setup (b) Sedans

(c) Minibus (d) Motorcycle

Figure 14: (a) Illustration of the experimental setup. (b) - (d) Pictures of three controlled experiment setups.

Prototype mmOVD. We have implemented the prototype mmOVD system using two different types of mmWave radar: the TI AWR1642BOOST [36], operating in the frequency band of 77-81 GHz, and the TI IWR6843ISK [37], operating in 60-64 GHz. This choice ensures that they do not interfere with each other. The radars are configured to achieve a theoretical range resolution of 9.75 cm, a velocity resolution of 0.41 m/s, and an azimuth resolution of 15°. Each radar is connected to a TI DCA1000EVM data capture card [38] for achieving high-speed data transmission. Additionally, to emulate a typical vehicle detection system, we install 4 cameras on the experiment vehicle's roof, inspired by nuScenes [5]. Please note that these cameras are not part of our system but are solely used for performance comparison and verification of our system. The hardware setup is depicted in Figure 13(a).

Radar Placement. When implementing mmOVD, the placement of radars significantly influences the system's performance. Our experiments with radars in different positions revealed that the radars should be placed as low as possible. Lower placement allows more waves to propagate under the blocking vehicle, enabling clearer detection of occluded vehicles. Additionally, positioning the radars next to the tires rather than in the middle of the doors reduces the intensity of ghost points, as the tires are less reflective than flat doors, resulting in fewer and weaker ghost points. Therefore, we placed the radars near the four tires on the side of the vehicle, as shown in Figure 13(b). The height of the radars is 0.25 meters, corresponding to the chassis height of the vehicle, and the distance between neighboring radars is 1.7 meters.

Experiment Settings. To address potential occlusion scenarios in real driving situations, we carefully examine several important parameters of an occlusion scenario: (1) The longitudinal distance between vehicles, (2) the lateral distance between vehicles, (3) the types of blocking/blocked vehicles, and (4) the driving patterns of vehicles. Varying these parameters, our evaluation involves experiments conducted across 56 diverse scenes, as illustrated in Figure 14. These experiments can be categorized into two types of settings:

- **Setting I: Underground Garage**: In this setting, we construct various occlusion situations in an underground parking lot, under different parameter settings. We keep the sensing and blocking vehicles stationary while moving the target vehicle. Initially, we position the sensing vehicle and the blocking vehicle side by side in two adjacent lanes. The longitudinal distance between them varies from -2.5 meters to 2.5 meters, while the lateral distance ranges from 1 meter to 2.5 meters. Subsequently, we drive the target vehicle along the third lane, capturing the entire process as it enters and exits the blind zone. Our aim is to precisely detect the target vehicle within the blind zone. In this set of experiments, we recorded a total of 105 minutes of data across 35 distinct scenes. The vehicles we used include 4 sedans, 2 SUVs, 1 minibus, and 1 motorcycle.
- **Setting II: Campus Road**: In this setting, we drive the sensing vehicle, blocking vehicle, and target vehicle on campus roads in the following four scenarios: (1) Three vehicles drive side by side at similar speeds; (2) the sensing vehicle and blocking vehicle drive at similar speeds, while the target vehicle overtakes from behind; (3) the sensing vehicle and target vehicle drive at similar speeds, while the blocking vehicle overtakes from behind; and (4) only the sensing vehicle and the blocking vehicle drive at similar speeds, with no occluded vehicles within the blind zone. Our aim is to assess the system's performance in different driving patterns and to verify the effectiveness of our ghost removal scheme. We have recorded 105 minutes of data across 21 different scenes, encompassing various vehicle types, road environments, and driving patterns. Out of the 105 minutes of data, 60 minutes involve occlusion conditions. The vehicles we used include 3 sedans and 2 SUVs, with speeds ranging from 20 to 40 km/h.

Experiment Data Preparation. To summarize, we conducted our controlled experiments in an underground garage and on two campus roads, generating a total of over 200 minutes of data. We deployed a 32-line LiDAR on the roof of the blocking vehicle to obtain bounding boxes of target vehicles. The results obtained from the LiDAR are considered ground truth, which will be used for labeling the radar points.

The labeled radar data is then split into two distinct sets: a training set comprising 60 minutes of data, and a testing set comprising 150 minutes of data. Please note that the scenarios of the training set and the testing set do *not* overlap. The training set is used to train the point classifier and the test set is used for evaluation.

Performance Metrics. We adopt the following metrics to evaluate the proposed mmOVD system:

- *Accuracy of Point Classification*: The points generated by real vehicles are labeled as *positive*, while the ghost points and noises are labeled as *negative*. Therefore, the correctly classified points are counted as true positive (TP) and true negative (TN), while the incorrect ones are false positive (FP) and false negative (FN). The accuracy is defined as $\frac{TP+TN}{TP+TN+FP+FN}$.
- *Accuracy of Occluded Vehicle Detection*: As shown in Section 3.4, we extract the bounding box of the target vehicle from point clouds. If the center of detected bounding box is inside the ground-truth bounding box (which is obtained by the LiDAR), it is counted as a true positive (TP). Otherwise, the detections and ground-truth objects are counted as false positives (FP) and false negatives (FN), respectively. We mainly use the F1 score to assess the accuracy, which is the harmonic mean of precision (P) and recall (R), i.e., $F1 = \frac{2*P*R}{P+R}$.
- *Success Rate of Occlusion Event Detection*: We first define an *occlusion event* as the entire process of the target vehicle entering and exiting the blind zone, as shown in Figure 15. By monitoring when the target vehicles disappear from cameras' fields of view and subsequently reappear, we can determine the ground-truth start and end times of each occlusion event. mmOVD triggers an alarm when it detects an occluded vehicle in N consecutive frames (where N is a preset alarm threshold). If mmOVD raises an alarm for a true event, we consider the event as successfully detected. The success rate is calculated as the ratio between the number of detected occlusion events and the total number of occlusion events. To find the appropriate threshold N for achieving a high success rate while mitigating false alarms, we perform statistical analysis on false alarms in no occlusion scenarios.

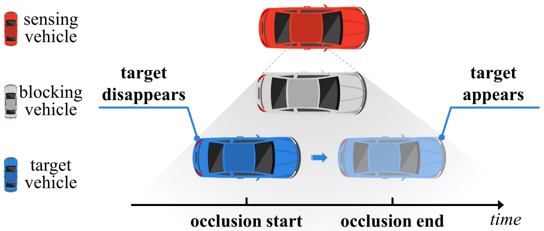


Figure 15: Illustration of an occlusion event. We get the start and end of an event by monitoring when the target vehicle disappears from cameras and when it reappears.

4.2 Radar Point Classification

Firstly, we evaluate the accuracy of point classification with our feature set under different classifiers and compare the results with the feature set proposed by Prophet et al. [25]. The comparison results are shown in Figure 16, which shows

Table 2: Classification accuracy and average computing time for each frame of radar point clouds under different classifiers on AMD Ryzen 7 5800H processor

Classifier	Random Forest	Decision Tree	Gaussian SVM	KNN	Linear SVM
Accuracy	94.2%	92.3%	90.5%	89.5%	88.1%
Latency	4.37 ms	3.55 ms	171 ms	31.9 ms	3.65 ms

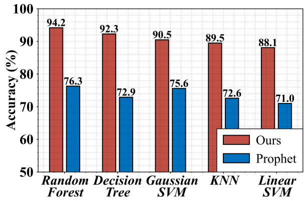


Figure 16: Comparison of our feature set with the baseline feature set in terms of point classification accuracy

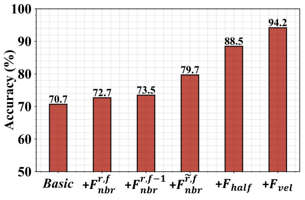


Figure 17: Point classification accuracy using Random Forest with the addition of each feature in our feature set

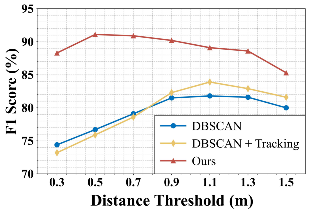


Figure 18: Comparison of mmOVD with the baseline schemes in terms of vehicle detection accuracy

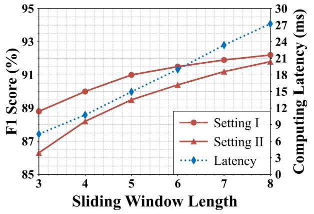


Figure 19: F1 score and computing latency of detection under different sliding window duration

that our feature set outperforms the baseline by a large margin. For example, our feature set, when combined with the decision tree model, can improve the accuracy from 72.9% to 92.3%, accounting for a 19.4% improvement.

Besides, we observe considerable variations in the computing times of different classifiers, as shown in Table 2. The processing is done by the AMD Ryzen 7 5800H processor. Among all the classifiers, random forest model incurs the best classification accuracy (94.2%) with an acceptable latency (4.37 ms per frame). Taking into consideration both accuracy and efficiency, we opt for random forest model as the preferred classifier for subsequent experimental evaluations.

Furthermore, to demonstrate the effectiveness of mmOVD feature set, we conduct ablation experiments on the five features, each carefully designed by us. We begin with solely utilizing the five basic features (i.e., range, azimuth, doppler velocity, intensity, and SNR) as our baseline. Subsequently, we systematically incorporated each of our custom-designed features, evaluating the accuracy of point cloud classification along the way. Figure 17 shows the results: after integrating our five features, the accuracy of point cloud classification improved by 23.5% compared to the baseline. Notably, $F_{nbr}^{\widetilde{r},f}$, F_{half}, and F_{vel} exhibited the most substantial enhancements, with improvements of 6.2%, 8.8%, and 5.7%, respectively. We have also examined the Gini importance [19] of each feature in the random forest model. Specifically, our five proposed features—$F_{nbr}^{r,f}$, $F_{nbr}^{r,f-1}$, $F_{nbr}^{\widetilde{r},f}$, F_{half}, and F_{vel}—have Gini importances of 8.7%, 6.5%, 13.5%, 15.6%, and 12.8%, respectively. These results confirm the significance of the features we proposed for point cloud classification.

4.3 Occluded Vehicle Detection

We then evaluate the overall performance of our cross-frame aggregation detection algorithm and two baseline schemes, illustrated in Figure 18. The blue line represents the performance of solely utilizing DBSCAN for vehicle detection under different distance thresholds, showing the optimal F1 score achieved is 81.8%. We also apply a typical tracking algorithm, AB3DMOT [45], on top of DBSCAN. The results are represented by the yellow line, with an optimal F1 score of 83.9%. Meanwhile, our method outperforms the baselines significantly, with an optimal F1 score of 91.1%.

In addition, we also evaluate the performance of our cross-frame aggregation-based vehicle detection algorithm under different sliding window sizes. Here, we adopted the AMD Ryzen 7 5800H processor for the task. We report the F1 score and average computing latency in Figure 19. The results demonstrate that the performance of the mmOVD becomes better as the length of the sliding window increases. Specifically, when the length of the sliding window reaches 6, the F1 score of both Setting I and Setting II exceeds 90%. Notably, the F1 score of Setting II is always slightly lower than Setting I, while the gap decreases as the window length increases. This difference arises because Setting II involves a rather challenging occlusion scenario, where three vehicles are driving side by side at similar speeds for 20 minutes. However, we note that vehicles driving side by side for an extended period of time are not common in the real world as it is considered a bad driving habit [48].

Furthermore, we observe a nearly linear increase in the computing latency with the increase in sliding window length. The reason is that the amount of computation in our detection algorithm is directly proportional to the window length. Considering that our radar transmits data at 30 frames per second, it is crucial to keep the computing latency of each frame below 30ms. Since our random forest model costs 4.37 ms per frame (Table 2), we have chosen the length of the sliding window to be 6, with an average computing latency of 18.99 ms per frame. All subsequent evaluations will be based on this window length.

In Setting I, we construct various occlusion scenarios under three important parameters: longitudinal distance, lateral distance, and types of vehicles. In Setting II, we conduct experiments in different driving patterns. We then quantify the influences of these parameters on occluded vehicle detection.

Impact of Longitudinal Distance Between Vehicles. The longitudinal distance between the sensing and blocking vehicles plays a crucial role in determining the extent of the blind zone. Specifically, we define the longitudinal distance as negative when the blocking vehicle is positioned behind

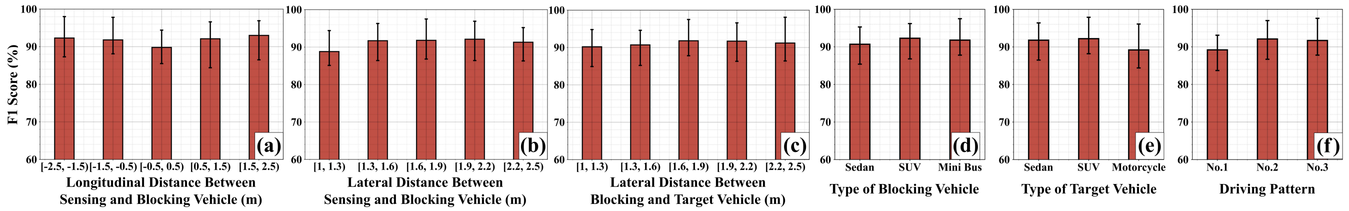


Figure 20: F1 score of detection under different (a) longitudinal distances between sensing and blocking vehicles, (b) lateral distances between sensing and blocking vehicles, (c) lateral distances between blocking and target vehicles, (d) types of blocking vehicles, (e) types of target vehicles, and (f) driving patterns.

sensing vehicle, and positive when it is ahead. Figure 20(a) shows the system's performance at different longitudinal distances. As the longitudinal distance approaches 0 (i.e., two vehicles are side by side), the F1 score of vehicle detection decreases by 2%, 2.3%, 2.5%, and 3.2%, respectively, compared to other four distance settings. The reason is that when the two vehicles are completely parallel, the blind zone becomes the largest, leading to the occlusion of the target vehicle for a longer duration. However, we note that this situation is not common in the real world [48].

Impact of Lateral Distance Between Vehicles. The lateral distance between the sensing and blocking vehicles is another important parameter to consider. As we focus on the occlusion scenarios, we assume that all three vehicles stay within their respective lanes and maintain a safe lateral distance of at least 1 meter between them. Figure 20(b) shows the system's performance under different lateral distances between the sensing vehicle and the blocking vehicle. It can be seen that when the distance is smaller than 1.3 meters, the F1 score is 2.5% to 3.3% lower compared to other situations. This decline is caused by the stronger multi-path effect and more severe occlusion when the blocking vehicle gets too close. However, vehicles driving in adjacent lanes usually do not maintain such a short distance in the real world. Additionally, Figure 20(c) presents the system's performance under different lateral distances between the blocking vehicle and the target vehicle. The F1 scores consistently exceed 90%, and there are no notable fluctuations across different situations. Collectively, the results demonstrate that mmOVD is rather robust against varying lateral distances.

Impact of Vehicle Type. The type of vehicles can indeed influence the way of forming ghost points, and hence the detection of occluded vehicles, due to their varying shapes and sizes. Therefore, we select several types of vehicles for experimentation. In addition to the common SUV and sedan types, our experiments also involve a minibus as the blocking vehicle and a motorcycle as the target vehicle. The results presented in Figure 20(d) indicate that when the blocking vehicle is a sedan, the F1 score is 1.1% to 1.6% lower compared to other types. This is attributed to sedans typically having lower chassis, making it more challenging to detect the occluded vehicle. Figure 20(e) shows that when the motorcycle is occluded, the F1 score of detection is 2.6% to 3% lower compared to vehicles. Due to its smaller size, the reflection of mmWave signals is much weaker in this case.

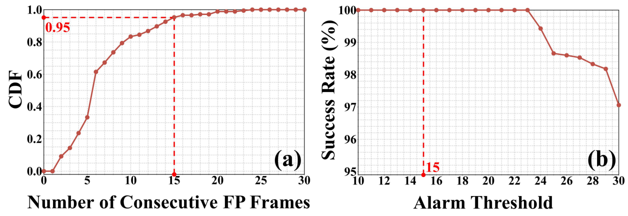


Figure 21: A 100% success rate and a 5% false alarm rate are achieved with the alarm threshold set to 15. (a) CDF of consecutive FP detection number when there is no occlusion. (b) Success rate of occlusion event detection under different alarm threshold values.

Impact of Driving Patterns. In Setting II, we emulate real driving scenarios on campus roads using six distinct experimental setups. In the first three setups, we employ an SUV as the blocking vehicle and a sedan as the target vehicle. Each setup features different driving patterns: (1) Three vehicles drive side by side at similar speeds; (2) the sensing vehicle and blocking vehicle drive at similar speeds, while the target vehicle overtakes from behind; and (3) the sensing vehicle and target vehicle drive at similar speeds, while the blocking vehicle overtakes from behind. Subsequently, we exchange the two vehicles and repeat the experiments. The performance of different driving patterns is shown in Figure 20(f). Notably, in driving pattern 1, in which three vehicles are driving side by side, the target vehicle remains continuously occluded. Consequently, the F1 score of driving pattern 1 is 2.5% to 2.9% lower than in the other patterns but still exceeds 89%, showcasing mmOVD's ability to handle long-term occlusion scenarios effectively.

4.4 Occlusion Event Detection

As mmOVD triggers an alarm when it detects an occluded vehicle in N consecutive frames, the choice of the alarm threshold N is crucial. On one hand, if N is too small, it may result in a high false alarm rate. On the other hand, if N is too

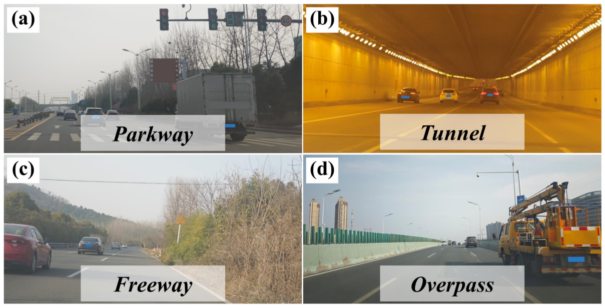


Figure 22: Our 324-kilometer route includes various types of crowded roads.

large, it may lead to missed alarms and delayed responses. In this work, we attempt to find the smallest N that gives a low enough (say, 5%) false alarm rate. This decision is inspired by the state-of-the-art LiDAR-based vehicle detection algorithm (VirConv-S [47]), which has a false alarm rate exceeding 5%.

In order to find the threshold value that gives a certain false alarm rate, we collect additional data from the campus road setting when there is no occlusion. Here, we drive the sensing vehicle and the blocking vehicle side by side on the campus road, ensuring that there are no occluded vehicles within the blind zone.In this set of experiments, we recorded 45 minutes of data across 15 distinct scenes.

Specifically, we calculate the likelihood of having N consecutive false positive (FP) detections when there is no occlusion. Then we obtain the CDF and present the results in Figure 21(a). We observe that by choosing $N = 15$, the corresponding CDF is 95%. That is, in an unknown situation, when we detect the presence of a hidden vehicle from 15 or more frames consecutively, the probability of having no occlusion is lower than 5%, thus the false alarm rate of 5%. As a result, we can choose the alarm threshold $N = 15$ in our system – mmOVD triggers an alarm when it detects an occluded vehicle in 15 consecutive frames. Notably, although increasing the value of N could further reduce the false alarm rate while keeping a 100% of success rate, it would also delay the response time. We chose N to be 15 to ensure that mmOVD maintains a short response latency.

In addition, we also plot the system's success rate of occlusion event detection with different alarm threshold values, shown in Figure 21(b). With the alarm threshold set to 15, mmOVD achieves a 100% success rate of occlusion event detection. As a result, we conclude $N = 15$ is a suitable threshold for issuing occlusion alarms for it achieves a 100% success rate while mitigating the false alarm rate to be lower than 5% in our dataset.

4.5 Limitations and Discussion

While mmOVD has achieved high performance in many scenarios, it is important to acknowledge its limitations. First, our study has primarily focused on three-lane environments. Although we anticipate that this approach could be extended to four-lane environments and detect fully occluded vehicles behind two blocking vehicles, this remains an intriguing topic for future research. Second, mmOVD's ability to detect smaller targets, such as bicycles and pedestrians, is limited because these targets are less reflective than cars and motorcycles. These limitations are partly due to the restricted power and resolution of the commercial off-the-shelf radars used in our system. Therefore, utilizing more advanced radars with higher power or developing methods to detect weak targets could potentially address these limitations.

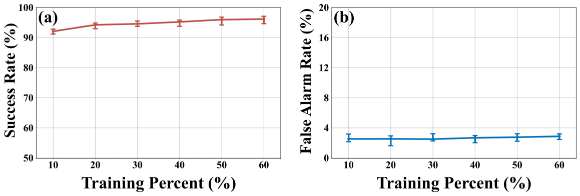


Figure 23: Success rate and false alarm rate of real-world trials under different training percent of data.

5 OCCLUSION DETECTION WHILE DRIVING IN THE REAL WORLD

After conducting extensive controlled experiments, we next assess the performance of mmOVD while we drive the sensing car in the real world.

Driving Scenarios and Data Acquisition. We drove multiple trips from our campus to a nearby town with a sensing car equipped with the mmOVD prototype. The route included parkways, tunnels, freeways, overpasses, and bridges, most with three or four lanes (as shown in Figure 22). We drove a total distance of **324 kilometers** over **580 minutes**, reaching a maximum speed of approximately **70 km/h**. We acquired 153,000 frames of valid data. We manually annotated the occlusion events by examining the camera images frame by frame. Our dataset comprises 226 occlusion events, each lasting between 0.7 and 4 seconds.

Occlusion Event Detection while Driving. To evaluate our system's overall performance with different training data amounts, we vary the training data volume from 10% to 60% of the total data and report the detection success rate and false alarm rate in Figure 23. For each data point, we conducted 10 experiments, each experiment using randomly sampled data. We discover that *even with just 10% of the total data (i.e., 15,300 frames) used for training, our system achieved a detection success rate of 92.3%.* Furthermore, the false alarm rate of the system remained consistently low (less than 4%), showcasing the robustness of our system when operating in real-world environments. Additionally, each occlusion commonly lasts 2-4 seconds in real-world

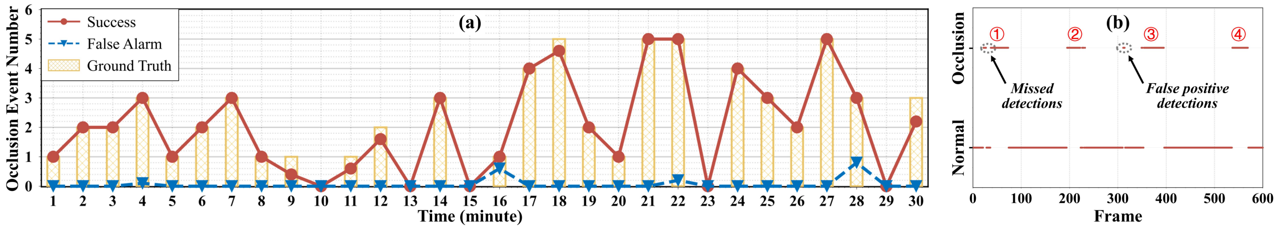


Figure 24: (a) The occlusion event detection results over a 30-minute period. (b) The occluded vehicle detection results over a 600-frame period.

scenarios, and mmOVD triggers a correct alarm on average within 0.7 seconds after the occlusion starts.

In Figure 24(a), we plot the number of occlusion events detected within a minute over a 30-minute period, collected while the vehicle was driving on a busy overpass. The yellow bars represent the ground-truth number of occlusion events during each time interval, while the brown and blue lines represent the average number of successfully detected occlusion events and falsely alarmed occlusion events during that interval, respectively. This figure demonstrates that our system exhibits excellent performance even in crowded scenarios. As for the occasional errors, we found that they were mainly caused by situations not considered in the controlled experiments. For example, we observed several trucks being occluded that our system failed to detect. The reason is that the chassis of these trucks are much higher than common cars, and are beyond the radars' detection range. Fortunately, these trucks are typically tall enough for cameras to detect even when occluded by other cars.

In Figure 24(b), we show the fine-grained per-frame occluded vehicle detection result over a 600-frame (20 seconds) period, containing 4 occlusion events. Due to the complexity of crowded road environments, occasional per-frame missed detection and false positives may occur, which is expected. However, our event-level detection mechanism can effectively accommodate these occasional per-frame errors.

6 RELATED WORK

Partial Occlusion Detection. In the area of autonomous driving, many works have been proposed to detect vehicles under occlusion [16, 53]. The researchers primarily employ cameras and LiDARs for perception and propose various occlusion handling strategies to enhance vehicle detection models. Some researchers utilize a vehicle's distinctive appearance, such as the windscreen [39], or rely solely on deep learning methods [16] to enhance detection performance. However, these methods have limitations and are only effective in partially-occluded scenarios.

Radar-based Detection. In the area of wireless sensing, many researchers have focused on achieving non-line-of-sight (NLOS) detection based on mmWave radar, e.g., detecting objects behind corners [8, 9, 12, 31, 35, 52]. For example, Guo et al. [8] proposed a NLOS location algorithm based on ray-tracing model using a mmWave radar. Yue et al. [52] achieved accurate indoor localization around corners, and addressed the limitations of the ray-tracing model used in past work. Additionally, Scheiner et al. [28] applied Doppler radars to detect and track hidden objects in outdoor environments. Although the principle of detecting targets around corners is similar to our work, the complexity of automotive environments has brought challenges to our system.

Ghost Classification. Identifying ghost detection remains a critical problem for mmWave radar sensing [17]. Some researchers utilize deep learning algorithms to detect ghost points [15]. Specifically, they utilize the PointNet++ architecture to learn the point features and detect ghosts. This approach can achieve good performance on the dataset while being time-consuming and resource-consuming. Besides, Prophet et al. [25] proposed a feature engineering approach to distinguish the ghost points. The performance comparison of our method and their method has been presented in Section 4.2, indicating our method is more effective in eliminating ghost points in occlusion scenarios.

7 CONCLUSION

We have presented mmOVD, a practical system for detecting fully occluded vehicles using millimeter-wave radars. mmOVD takes advantage of the multipath in the environment to see through vehicles, while addressing the challenges due to ghost points and unstable reflections. For the first time, we can accurately detect and localize an "unseeable" vehicle. We believe that the design, implementation, and evaluation of our results present important practical contributions toward safer autonomous driving.

8 ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (No. 62332016). It is also partially supported by National Key R&D Program of China No. 2021ZD0110400, NSFC with No. 62072424, U20A20181, Key Research Program of Frontier Sciences, CAS. No. ZDBS-LY-JSC001, Hefei Municipal Natural Science Foundation with No. 2022016, the Fundamental Research Funds for the Central Universities with No. WK3500000008.

REFERENCES

[1] Fadel Adib, Zachary Kabelac, and Dina Katabi. 2015. Multi-Person Localization via RF Body Reflections. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*. 279–292.

[2] Roshan Ayyalasomayajula, Aditya Arun, Chenfeng Wu, Sanatan Sharma, Abhishek Rajkumar Sethi, Deepak Vasisht, and Dinesh Bharadia. 2020. Deep learning based wireless localization for indoor navigation. In *Proceedings of ACM MobiCom*. 1–14.

[3] Carlos Baquero Barneto, Elizaveta Rastorgueva-Foi, Musa Furkan Keskin, Taneli Riihonen, Matias Turunen, Jukka Talvitie, Henk Wymeersch, and Mikko Valkama. 2022. Millimeter-wave mobile sensing and environment mapping: Models, algorithms and validation. *IEEE Transactions on Vehicular Technology* 71, 4 (2022), 3900–3916.

[4] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. A Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory* (Pittsburgh, Pennsylvania, USA) *(COLT '92)*. New York, NY, USA, 144–152.

[5] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. 2020. nuScenes: A Multimodal Dataset for Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[6] Junil Choi, Vutha Va, Nuria Gonzalez-Prelcic, Robert Daniels, Chandra R Bhat, and Robert W Heath. 2016. Millimeter-wave vehicular communication to support massive automotive sensing. *IEEE Communications Magazine* 54, 12 (2016), 160–167.

[7] Yimin Dai, Xian Shuai, Rui Tan, and Guoliang Xing. 2023. Interpersonal Distance Tracking with MmWave Radar and IMUs. In *Proceedings of the 22nd International Conference on Information Processing in Sensor Networks (IPSN)* (San Antonio, TX, USA) *(IPSN '23)*. New York, NY, USA, 123–135.

[8] Shisheng Guo, Qingsong Zhao, Guolong Cui, Songlin Li, Lingjiang Kong, and Xiaobo Yang. 2020. Behind corner targets location using small aperture millimeter wave radar in NLOS urban environment. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 13 (2020), 460–470.

[9] Magnus Gustafsson, Åsa Andersson, Tommy Johansson, Stefan Nilsson, Ain Sume, and Anders Örbom. 2015. Extraction of human micro-doppler signature in an urban environment using a sensing-behind-the-corner radar. *IEEE Geoscience and Remote Sensing Letters* 13, 2 (2015), 187–191.

[10] Tin Kam Ho. 1998. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20, 8 (1998), 832–844.

[11] Kevin Jiokeng, Gentian Jakllari, Alain Tchana, and André-Luc Beylot. 2020. When FTM Discovered MUSIC: Accurate WiFi-based Ranging in the Presence of Multipath. In *Proceedings of IEEE INFOCOM*. 1857–1866.

[12] Tommy Johansson, Åsa Andersson, Magnus Gustafsson, and Stefan Nilsson. 2016. Positioning of moving non-line-of-sight targets behind a corner. In *2016 European Radar Conference*. IEEE, 181–184.

[13] Ahmad Kamari, Yoon Chae, and Parth Pathak. 2023. mmSV: mmWave Vehicular Networking using Street View Imagery in Urban Environments. In *Proceedings of ACM MobiCom*. 1–16.

[14] Kamran Khan, Saif Ur Rehman, Kamran Aziz, Simon Fong, and Sababady Sarasvady. 2014. DBSCAN: Past, present and future. In *Proceedings of ICADIWT*. IEEE, 232–238.

[15] Florian Kraus, Nicolas Scheiner, Werner Ritter, and Klaus Dietmayer. 2020. Using Machine Learning to Detect Ghost Images in Automotive Radar. In *Proceedings of IEEE ITSC*. 1–7.

[16] Junliang Li, Hon-Cheng Wong, Sio-Long Lo, and Yuchen Xin. 2018. Multiple object detection by a deformable part-based model and an R-CNN. *IEEE Signal Processing Letters* 25, 2 (2018), 288–292.

[17] Yunda Li and Xiaolei Shang. 2022. Multipath Ghost Target Identification for Automotive MIMO Radar. In *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*. IEEE, 1–5.

[18] Haipeng Liu, Yuheng Wang, Anfu Zhou, Hanyue He, Wei Wang, Kunpeng Wang, Peilin Pan, Yixuan Lu, Liang Liu, and Huadong Ma. 2020. Real-Time Arm Gesture Recognition in Smart Home Scenarios via Millimeter Wave Sensing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 4, Article 140 (dec 2020), 28 pages.

[19] Gilles Louppe, Louis Wehenkel, Antonio Sutera, and Pierre Geurts. 2013. Understanding variable importances in forests of randomized trees. *Advances in neural information processing systems* 26 (2013).

[20] C Lovescu and S Rao. 2015. The Fundamentals of Millimeter Wave Radar Sensors. *Avaliable: https://www.ti.com/lit/pdf/spyy005* (2015).

[21] Chengzhen Meng, Yifan Duan, Chenming He, Dequan Wang, Xiaoran Fan, and Yanyong Zhang. 2024. mmPlace: Robust Place Recognition With Intermediate Frequency Signal of Low-Cost Single-Chip Millimeter Wave Radar. *IEEE Robotics and Automation Letters* (2024).

[22] Anthony J Myles, Robert N Feudale, Yang Liu, Nathaniel A Woody, and Steven D Brown. 2004. An introduction to decision tree modeling. *Journal of Chemometrics: A Journal of the Chemometrics Society* 18, 6 (2004), 275–285.

[23] NHTSA. 2023. TRAFFIC SAFETY FACTS 2021. https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/813527.

[24] Alain Olivier, Guillermo Bielsa, Irene Tejado, Michele Zorzi, Joerg Widmer, and Paolo Casari. 2016. Lightweight indoor localization for 60-GHz millimeter wave systems. In *Proceedings of IEEE SECON*. IEEE, 1–9.

[25] Robert Prophet, Javier Martinez, Juan-Carlos Fuentes Michel, Randolf Ebelt, Ingo Weber, and Martin Vossiek. 2019. Instantaneous Ghost Detection Identification in Automotive Scenarios. In *2019 IEEE Radar Conference*. 1–6.

[26] Kun Qian, Zhaoyuan He, and Xinyu Zhang. 2020. 3D Point Cloud Generation with Millimeter-Wave Radar. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 4, Article 148 (dec 2020), 23 pages.

[27] Fabian Roos, Mohammadreza Sadeghi, Jonathan Bechter, Nils Appenrodt, Jürgen Dickmann, and Christian Waldschmidt. 2017. Ghost target identification by analysis of the Doppler distribution in automotive scenarios. In *2017 18th International Radar Symposium (IRS)*. IEEE, 1–9.

[28] Nicolas Scheiner, Florian Kraus, Fangyin Wei, Buu Phan, Fahim Mannan, Nils Appenrodt, Werner Ritter, Jurgen Dickmann, Klaus Dietmayer, Bernhard Sick, et al. 2020. Seeing around street corners: Non-line-of-sight detection and tracking in-the-wild using doppler radar. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2068–2077.

[29] Robert Schneider, Dirk Didascalou, and Werner Wiesbeck. 2000. Impact of road surfaces on millimeter-wave propagation. *IEEE Transactions on Vehicular Technology* 49, 4 (2000), 1314–1320.

[30] Souvik Sen, Jeongkeun Lee, Kyu-Han Kim, and Paul Congdon. 2013. Avoiding multipath to revive inbuilding WiFi localization. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '13)*. 249–262.

[31] Pawan Setlur, Tadahiro Negishi, Natasha Devroye, and Danilo Erricolo. 2014. Multipath Exploitation in Non-LOS Urban Synthetic Aperture Radar. *IEEE Journal of Selected Topics in Signal Processing* 8, 1 (2014), 137–152.

[32] Anish Shastri, Neharika Valecha, Enver Bashirov, Harsh Tataria, Michael Lentmaier, Fredrik Tufvesson, Michele Rossi, and Paolo Casari. 2022. A review of millimeter wave device-based localization and device-free sensing technologies and applications. *IEEE Communications*

Surveys & Tutorials 24, 3 (2022), 1708–1749.

[33] Emerson Sie, Zikun Liu, and Deepak Vasisht. 2023. BatMobility: Towards Flying Without Seeing for Autonomous Drones. In *Proceedings of ACM MobiCom*. 1–16.

[34] Elahe Soltanaghaei, Avinash Kalyanaraman, and Kamin Whitehouse. 2018. Multipath Triangulation: Decimeter-level WiFi Localization and Orientation with a Single Unaided Receiver. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '18)*. 376–388.

[35] Khac-Phuc-Hung Thai, Olivier Rabaste, Jonathan Bosse, Dominique Poullin, Israel Hinostroza, Thierry Letertre, and Thierry Chonavel. 2017. Around-the-corner radar: Detection and localization of a target in non-line of sight. In *2017 IEEE Radar Conference*. 0842–0847.

[36] TI. 2020. AWR1642 single-chip 76-GHz to 81-GHz automotive radar sensor evaluation module. [Online]. https://www.ti.com/tool/AWR1642BOOST.

[37] TI. 2020. IWR6843 intelligent mmWave sensor standard antenna plug-in module. [Online]. https://www.ti.com/tool/IWR6843ISK.

[38] TI. 2020. Real-time data-capture adapter for radar sensing evaluation module. [Online]. https://www.ti.com/tool/DCA1000EVM.

[39] Huy Van Pham and Byung-Ryong Lee. 2015. Front-view car detection and counting with occlusion in dense traffic flow. *International Journal of Control, Automation and Systems* 13, 5 (2015), 1150–1160.

[40] Tristan Visentin, Jürgen Hasch, and Thomas Zwick. 2018. Analysis of multipath and DOA detection using a fully polarimetric automotive radar. *International Journal of Microwave and Wireless Technologies* 10, 5-6 (2018), 570–577.

[41] Dequan Wang, Xinran Zhang, Kai Wang, Lingyu Wang, Xiaoran Fan, and Yanyong Zhang. 2024. RDGait: A mmWave Based Gait Recognition System for Complex Indoor Environments Using Single-chip Radar. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* (2024).

[42] Ge Wang, Chen Qian, Kaiyan Cui, Xiaofeng Shi, Han Ding, Wei Xi, Jizhong Zhao, and Jinsong Han. 2020. A Universal Method to Combat Multipaths for RFID Sensing. In *Proceedings of IEEE INFOCOM*. 277–286.

[43] Jue Wang and Dina Katabi. 2013. Dude, where's my card? RFID positioning that works with multipath and non-line of sight. In *Proceedings of the ACM SIGCOMM* (Hong Kong, China). New York, NY, USA, 51–62.

[44] Zhongqin Wang, Min Xu, Ning Ye, Ruchuan Wang, and Haiping Huang. 2019. RF-Focus: Computer Vision-assisted Region-of-interest RFID Tag Recognition and Localization in Multipath-prevalent Environments. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 3, 1, Article 29 (mar 2019), 30 pages.

[45] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 2020. 3D Multi-Object Tracking: A Baseline and New Evaluation Metrics. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 10359–10366.

[46] Volker Winkler. 2007. Range Doppler detection for automotive FMCW radars. In *2007 European Radar Conference*. IEEE, 166–169.

[47] Hai Wu, Chenglu Wen, Shaoshuai Shi, Xin Li, and Cheng Wang. 2023. Virtual sparse convolution for multimodal 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 21653–21662.

[48] HH Xiong and JX Hu. 2019. Study on safe driving spacing control of co-directional vehicles on adjacent lanes in urban roads. *Advances in transportation studies* (2019).

[49] Zhou Xu, Chongyi Fan, and Xiaotao Huang. 2021. MIMO radar waveform design for multipath exploitation. *IEEE Transactions on Signal Processing* 69 (2021), 5359–5371.

[50] Hongfei Xue, Qiming Cao, Yan Ju, Haochen Hu, Haoyu Wang, Aidong Zhang, and Lu Su. 2022. M4esh: mmwave-based 3d human mesh construction for multiple subjects. In *Proceedings of ACM SenSys*. 391–406.

[51] Hongfei Xue, Qiming Cao, Chenglin Miao, Yan Ju, Haochen Hu, Aidong Zhang, and Lu Su. 2023. Towards generalized mmwave-based human pose estimation through signal augmentation. In *Proceedings of ACM MobiCom*. 1–15.

[52] Shichao Yue, Hao He, Peng Cao, Kaiwen Zha, Masayuki Koizumi, and Dina Katabi. 2022. Cornerradar: Rf-based indoor localization around corners. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 6, 1 (2022), 1–24.

[53] Wei Zhang, QM Jonathan Wu, Xiaokang Yang, and Xiangzhong Fang. 2008. Multilevel framework to detect and handle vehicle occlusion. *IEEE Transactions on Intelligent Transportation Systems* 9, 1 (2008), 161–174.