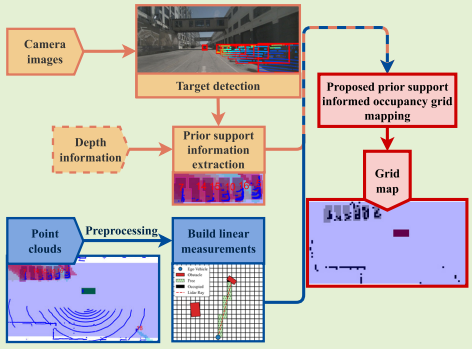
Camera-Aided Binary Prior Support-Informed Occupancy Grid Mapping

Peiyuan Zhai, *Graduate Student Member, IEEE*, Geethu Joseph, *Senior Member, IEEE*, Nitin Jonathan Myers, *Member, IEEE*, and Ashish Pandharipande, *Senior Member, IEEE*

***Abstract*—Occupancy grid mapping is a common approach to support automotive driving perception capabilities. We present an occupancy grid estimation algorithm using sensor point-cloud measurements aided by side information from other sensing modalities like cameras. This prior side information is in the form of an erroneous occupancy map estimate, referred to as prior support information. Specifically, we extract a prior map using you only look once (YOLO) object detection on camera images. A sparse Bayesian learning-based mapping algorithm is designed with a modified hierarchical model to incorporate this prior. Experiments done on public real-world driving datasets, nuScenes and RADIATE, demonstrate that our approach achieves better target detection and scatter noise reduction than the state-of-the-art methods. Furthermore, our method seamlessly works on the two datasets although we train YOLO only using camera images from nuScenes.**



***Index Terms*— Autonomous driving, light detection and ranging (LiDAR), radar, sensor fusion, sensor point cloud, sparse Bayesian learning.**

I. Introduction

AN AUTOMOTIVE perception system enables a vehicle to sense and understand its environment, making it indispensable for higher levels of driving autonomy [1], [2]. An important component of the perception system is the construction of accurate maps indicating the presence of obstacles in drivable areas and surroundings. Such maps employ occupancy grids with binary variables that indicate the occupancy status (occupied or free) of their corresponding location in the environment, and can be applied in tasks, such as collision avoidance [3] or target tracking [4]. In automotive driving, 2-D occupancy grid mapping (OGM) algorithms rely on measurements from single or multiple sensors, often using time-of-flight sensors such as light detection and ranging (LiDAR) or radar [5], [6]. The goal of this article is to devise a new OGM algorithm that uses LiDAR or radar measurements to construct an occupancy map by exploiting occupancy side information from other sensors and surrounding devices.

A popular OGM algorithm is the probabilistic approach called inverse sensor model (ISM) [7], [8]. ISM is based on a recursive implementation and has a low computational complexity. However, ISM fails to exploit the typical OGM structure wherein occupied or free cells often cluster together. Efforts to enhance ISM by including the geometrical features and the alternative approach of the forward sensor model implicitly incorporate the spatial structure, albeit at the cost of high computational complexity [9], [10]. Gaussian process-based approaches were used to construct correlated occupancy grid maps [11], [12]. While these approaches capitalize on smooth spatial correlations, they do not exploit the sparsity of occupancy. The sparse structure arises because, even in the presence of obstacles, only the borders or edges of obstacles are reflected in the map. The OGM algorithm in [13] exploits the sparse and clustered structure of the map, employing a pattern-coupled sparse Bayesian learning (PCSBL) framework [14]. This algorithm enhances the detection ability to separate small targets and suppresses ground-scatter effects. Existing OGM algorithms in [9], [10],

Received 24 October 2025; accepted 29 November 2025. Date of publication 17 December 2025; date of current version 2 February 2026. This work was supported by the Rijksdienst Voor Ondernemend Nederland under Grant PPS2207. The associate editor coordinating the review of this article and approving it for publication was Dr. Vanita Arora. *(Corresponding author: Ashish Pandharipande.)*

Peiyuan Zhai and Geethu Joseph are with the Signal Processing Systems Group, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: p.zhai@tudelft.nl; g.joseph@tudelft.nl).

Nitin Jonathan Myers is with Delft Center for Systems and Control, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: N.J.Myers@tudelft.nl).

Ashish Pandharipande is with NXP Semiconductors, 5600 KA Eindhoven, The Netherlands (e-mail: ashish.pandharipande@nxp.com).

Digital Object Identifier 10.1109/JSEN.2025.3642255

[11], [12], and [13], however, rely solely on measurements from one sensor, either LiDAR or radar, and seldom fuse both [15] to construct the map. These algorithms do not account for additional information from other sensors, like cameras or perception-related data from surrounding vehicles.

There are several works that discuss the object detection or mapping task with both visual and range sensors. In [16], the results of visual target detection are transformed into traversability and then fused with LiDAR data in an ISM-based OGM algorithm. This algorithm is simple and fast, but does not exploit spatial relationships across the map. Furthermore, Mazher et al. [17] discussed the possibility of using camera target detection results to enhance radar detection by formulating the regularized optimization problem. However, Mazher et al. [17] do not discuss the explicit construction of the regularization term based on given camera images.

To address the limitations of single-sensor or existing multimodal OGM algorithms, we investigate the problem of enhancing the detection accuracy of OGM algorithms by leveraging side information in the form of a partially erroneous 2-D occupancy grid map estimate called prior support information (PSI).

Our specific contributions are twofold.

1) We devise a new sparsity-aware 2-D OGM algorithm for mapping ego vehicle's immediate environment via a two-layer Bayesian hierarchical model. While the first layer is similar to the one in the SBL framework that promotes sparsity, the new second layer controls the hyperprior based on the PSI.
2) As an illustration, we extract a binary occupancy map estimate using the you only look once (YOLO) algorithm applied to the camera images of the ego vehicle's surroundings. We train YOLO using the nuScenes dataset and test our algorithm on nuScenes and RADIATE datasets [18], [19]. The results show that our algorithm effectively utilizes PSI to enhance the detection ability, suppress false alarms at cells tagged as unoccupied in PSI, and implicitly incorporate the clustered structure of the map.

Overall, we contribute to the study of occupancy mapping by providing a new mechanism to utilize prior knowledge about the environment by extracting and using an erroneous estimate of the occupancy grid map.

II. PSI-Aided OGM Problem

This section provides an overview of the LiDAR and radar sensor signal model, the notion of PSI, and the mapping problem formulated as a sparse recovery problem with PSI.

A. Sensor Model

To formulate the problem, it is necessary to transform the sensor readings into linear measurements of the occupancy probabilities. This section summarizes the procedure for building the linear sensor model.

1) Preprocessing: We start with the definition of the coordinate systems. In this work, the coordinate system of the map, LiDAR, radar, camera, and image plane are all assumed to be both extrinsically and intrinsically calibrated [20]. With parameters obtained from calibration or installation process, data points in the map, LiDAR, radar, or camera coordinate systems can be freely projected between one another or can be projected onto the image plane [21]. This work focuses on mapping the ego vehicle's immediate environment; thus, the map coordinate system is defined as a known, fixed frame attached to the ego vehicle.

For raw sensor data preprocessing, we recall that the LiDAR point cloud represents reflected laser rays from objects in the ego vehicle's environment. However, the point cloud contains ground scatters, which are irrelevant for obstacle detection and lead to noisy sensor observations. We alleviate this problem by removing the point cloud below a specified height threshold. We ignore very small obstacles (e.g., large potholes or road edges) that fall below the threshold. We consider such obstacles either negligible for determining the drivable area or handled by a separate task [22]. Therefore, removing point clouds below the threshold is acceptable for the 2-D OGM problem. The radar data are in the form of range-azimuth maps containing clutter or target-related point clouds. The radar point cloud is obtained by processing reflected signals at the radar. To reduce clutter, the constant false alarm rate detector is applied to obtain a radar point cloud [23]. In addition, the point cloud beyond the driving street lanes is excluded to focus on relevant areas for obstacle detection. These preprocessing steps also lower the computational complexity of OGM estimation by reducing the data that needs to be further analyzed.

After preprocessing, the remaining M points in the point cloud are likely to correspond to environmental obstacles. At this stage, the point cloud does not naturally correspond to occupancy status. Thus, sensor models are required to map sensor readings to occupancy status.

2) LiDAR and Radar Models: Next, we construct the LiDAR and radar models by discretizing the map into N grid cells. We map each point in the point cloud to a set of occupied and free grid cells on the occupancy map.

For a LiDAR point, we use Bresenham's line algorithm [24]. Specifically, each LiDAR point indicates that a corresponding reflection point, discretized to the nearest grid cell, is occupied. Meanwhile, the cells along the line connecting the ego vehicle and the reflection point remain unoccupied. Fig. 1(a) shows an example of free and occupied areas in the LiDAR model.

For the radar model, we account for its lower angle resolution compared to LiDAR. A radar measurement is modeled by an arc with its vertex at the ego vehicle location and touching the reflection point [25]. The grids around the reflection points are labeled as occupied detections, while the other grids covered by the arc are detected as free spaces. An example is shown in Fig. 1(b).

3) Linear Sensor Measurement Model: Using the free and occupied cells corresponding to each point in the point cloud, we present the linear measurement model from [13]. Each point can be used to generate two linear measurements of the underlying occupancy probabilities for free and occupied cells. We define $\boldsymbol{x} \in [0, 1]^N$ as the vectorized version of the occupancy probabilities of the N cells. The linear measurements

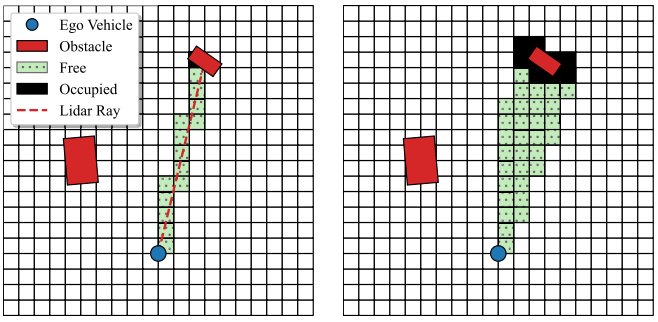


Fig. 1. LiDAR and radar sensor models. In these models, each point is used to construct two linear measurements of the occupancy grid map (a) LiDAR model. (b) Radar model.

corresponding to the mth point are given by

$$\begin{bmatrix} y_{\text{free}} \\ y_{\text{occ}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_{2m-1} \\ \boldsymbol{A}_{2m} \end{bmatrix} \boldsymbol{x} \tag{1}$$

where $\boldsymbol{A}_{2m-1} \in \{0,1\}^N$ is a row selection vector whose values are 1 at the free cell indices and 0 otherwise. Similarly, $\boldsymbol{A}_{2m} \in \{0,1\}^N$ is also a row selection vector with 1 at the occupied indices and 0 otherwise. The values for y_{occ} and y_{free} are prefixed such that y_{free} is close to 0, and y_{occ} is close to 1. Using all the M points from the point cloud, our linear measurement model is given by [13]

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{n} \tag{2}$$

where $\boldsymbol{y} \in [0,1]^{2M}$ is the occupancy status vector, $\boldsymbol{A} \in \{0,1\}^{2M \times N}$ is the measurement matrix, and the measurement noise is modeled as Gaussian $\boldsymbol{n} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_{2M})$, with unknown noise variance σ^2. The occupancy grid map estimation problem is to estimate $\boldsymbol{x}$ from $\boldsymbol{y}$ and $\boldsymbol{A}$ in (2).

B. Mapping Problem With PSI

With the linear sensor measurement model defined, we are ready to formulate the PSI-aided occupancy grid map estimation problem. We assume that PSI is a partially erroneous binary occupancy grid map estimate. We model the PSI using an index set $\mathcal{T} \subset \{1, 2, \ldots, N\}$, where $\mathcal{T}$ contains the indices of occupied grid cells in the estimate and its complement set $\mathcal{T}^{\complement}$ contains the indices of the remaining grid cells. The PSI can be extracted from any source, like camera images or static or dynamic digital maps, that provides information about the potential existence of obstacles.

As an example, we present one method for PSI extraction from mono camera images, which comprises three steps.

1) The first step is to process the camera images using object detection algorithms to identify potential obstacles in the environment. Some existing algorithms are edge detection or advanced deep learning methods such as YOLO networks, RetinaNet, or single-shot detectors [26], [27], [28]. The resultant detection outcome comprises bounding boxes of the identified obstacles in the image coordinate system.
2) The second step is to map these detection results onto the occupancy map coordinate system. To this end, LiDAR points are projected onto the image plane and matched with the identified obstacles. Subsequently, the LiDAR points matched with the obstacles are projected back into the map coordinate system.
3) The last step is identifying the occupied grid points on the map. For this, we construct a convex hull of the point cloud corresponding to each detected obstacle. Then, the indices of cells inside the convex hulls are defined as occupied in the PSI. In other words, the set $\mathcal{T}$ is the union of the indices of cells inside the convex hulls over all the obstacles identified by the object detection algorithm.

If the ego vehicle has stereo cameras, we can estimate PSI by recovering the range information from the stereo camera without relying on LiDAR data.

Given the linear measurement model in (2) and the PSI in the form of $\mathcal{T}$, our goal is to estimate the unknown sparse map $\boldsymbol{x}$. It should be noted that PSI is not considered a fully reliable source of information, but can contain false alarms or missdetections. Thus, we aim to integrate the PSI into the mapping algorithm using soft fusion via a Bayesian framework. In Section III, we present our Bayesian algorithm to estimate $\boldsymbol{x}$, which uses a hyperprior model designed to incorporate PSI into the sparse recovery algorithm.

III. PSI-Aided Bayesian Mapping Algorithm

We use the SBL framework to develop our OGM algorithm, which imposes a fictitious hierarchical prior on the unknown sparse map $\boldsymbol{x}$ and recovers the map by solving a type-II maximum a posteriori (MAP) problem.

The hierarchical prior on $\boldsymbol{x}$ consists of two layers. The first layer of the prior assumes a zero-mean Gaussian prior

$$\boldsymbol{x}|\boldsymbol{\alpha} \sim \prod_{n=1}^{N} \mathcal{N}\left(x_n|0, \alpha_n^{-1}\right) \tag{3}$$

where $\boldsymbol{\alpha}$ represents the precision vector containing the unknown hyperparameters. The hyperparameter $\boldsymbol{\alpha}$ influences the sparsity of $\boldsymbol{x}$. For a large α_n, the corresponding x_n has a very low variance and is close to 0. In contrast, when α_n is small, x_n has a large variance and is likely to have a nonzero value. The second layer controls the distribution of $\boldsymbol{\alpha}$ given by the Gamma-distributed hyperprior

$$\boldsymbol{\alpha} \sim \prod_{n=1}^{N} \text{Gamma}\left(\alpha_n|a_n, b_n\right) \tag{4}$$

where the probability density function is

$$\text{Gamma}\left(\alpha|a, b\right) = \Gamma\left(a\right)^{-1} b^a \alpha^{a-1} e^{-b\alpha} \tag{5}$$

with $\Gamma(a) = \int_0^\infty t^{a-1} e^{-t} dt$.

The standard SBL framework assumes a common hyperprior for all entries of $\boldsymbol{\alpha}$, i.e., $a_n = a$ and $b_n = b$ [29]. Here, $a = 0$ and $b = 0$ represent a noninformative hyperprior. Efforts to introduce informative third-layer priors include choosing b_n values from a binary set [30], and controlling both a_n and b_n with another prior layer based on side information [31]. Inspired by these works, we use an informative hyperprior

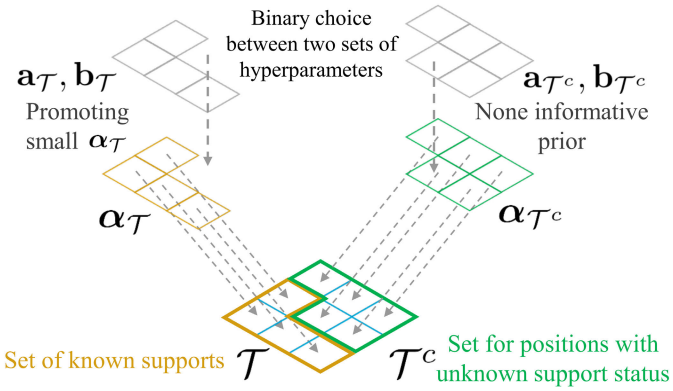


Fig. 2. Prior structure of our prior-aided algorithm. The measurement noise is omitted for simplicity.

to incorporate PSI. Here, we note that for $n \in \mathcal{T}$, x_n is more likely to be nonzero, whereas for $n \in \mathcal{T}^{\complement}$, it is less likely. We capture this information by choosing two sets of parameters for the hyperprior

$$(a_n, b_n) = \begin{cases} (a_{\mathcal{T}}, b_{\mathcal{T}}), & \text{if } n \in \mathcal{T} \\ (a_{\mathcal{T}^{\complement}}, b_{\mathcal{T}^{\complement}}), & \text{if } n \in \mathcal{T}^{\complement}. \end{cases} \tag{6}$$

Here, $0 < a_{\mathcal{T}} < a_{\mathcal{T}^{\complement}}$ and $0 < b_{\mathcal{T}^{\complement}} \ll b_{\mathcal{T}}$ to promote nonzero x_n when $n \in \mathcal{T}$. Also, $a_{\mathcal{T}^{\complement}}$ and $b_{\mathcal{T}^{\complement}}$ are chosen to represent noninformative priors that promote sparsity of the map. The prior structure of the two-layer PSI-OGM is summarized in Fig. 2. Finally, we complete the SBL prior model by using a Gamma distribution on the unknown measurement noise variance such that

$$\sigma^{-2} \sim \text{Gamma}\left(\sigma^{-2}|c, d\right) \tag{7}$$

where c and d are small positive values.

Given the prior model, we compute type-II MAP estimate of the probabilistic occupancy map $\boldsymbol{x}$. To this end, we first solve for the MAP estimate of $\boldsymbol{\alpha}$ and σ^2 from (2) by maximizing the log-likelihood function, i.e.,

$$\left(\hat{\boldsymbol{\alpha}}, \hat{\sigma}^2\right) = \underset{\boldsymbol{\alpha}, \sigma^2}{\arg\max}\ \log p\left(\boldsymbol{y}|\boldsymbol{\alpha}, \sigma^2\right). \tag{8}$$

These estimates, $\hat{\boldsymbol{\alpha}}$ and $\hat{\sigma}^2$, are then used to compute the MAP estimate of the occupancy grid map $\boldsymbol{x}$

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\arg\max}\ p\left(\boldsymbol{x}|\boldsymbol{y}, \hat{\boldsymbol{\alpha}}, \hat{\sigma}^2\right). \tag{9}$$

Given that $\boldsymbol{y}|\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{A}\boldsymbol{x}, \sigma^2 \boldsymbol{I}_{2M})$ from (2) and the Gaussian prior assumption in (3), we can deduce that the posterior distribution $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}, \sigma^2)$ is also Gaussian. Consequently, $\hat{\boldsymbol{x}}$ is determined by the posterior mean, which requires us to first compute the MAP estimates $\hat{\boldsymbol{\alpha}}$ and $\hat{\sigma}^2$.

Computing the MAP estimate from (8) in closed-form is intractable, and we use the iterative expectation–maximization (EM) algorithm. The kth EM iteration is given by

$$\textbf{E-step: } Q\left(\boldsymbol{\alpha}, \sigma^2\right) = \mathbb{E}_{\boldsymbol{x}|\boldsymbol{y};\boldsymbol{\alpha}^{(k)},(\sigma^2)^{(k)}}\left\{\log\left[p\left(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}, \sigma^2\right)\right]\right\} \tag{10}$$

$$\textbf{M-step: } \left(\boldsymbol{\alpha}^{(k+1)}, \left(\sigma^2\right)^{(k+1)}\right) = \underset{\boldsymbol{\alpha}, \sigma^2}{\arg\max}\ Q\left(\boldsymbol{\alpha}, \sigma^2\right) \tag{11}$$

where $\boldsymbol{\alpha}^{(k+1)}$ and $(\sigma^2)^{(k+1)}$ are the algorithm iterates in the kth iteration. Simplifying the above step [29], the kth iteration of the EM algorithm can be summarized as follows:

$$\alpha_n^{(k+1)} = \frac{1 + 2a_n}{\left(\mu_n^{(k)}\right)^2 + \Sigma_x^{(k)}[n, n] + 2b_n} \tag{12}$$

$$\left(\sigma^2\right)^{(k+1)} = \frac{2d + \left\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{\mu}_{\boldsymbol{x}}^{(k)}\right\|_2^2 + \text{tr}\left(\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\boldsymbol{\Sigma}_{\boldsymbol{x}}^{(k)}\right)}{2M + 2c}. \tag{13}$$

Here, $\boldsymbol{\mu}_{\boldsymbol{x}}^{(k)}$ and $\boldsymbol{\Sigma}_{\boldsymbol{x}}^{(k)}$ are the posterior mean and variance of the Gaussian posterior distribution $p(\boldsymbol{x}|\boldsymbol{y}, \boldsymbol{\alpha}^{(k)}, (\sigma^2)^{(k)})$, given by

$$\boldsymbol{\mu}_{\boldsymbol{x}}^{(k)} = \frac{1}{(\sigma^2)^{(k)}}\boldsymbol{\Sigma}_{\boldsymbol{x}}^{(k)}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{y} \tag{14}$$

$$\boldsymbol{\Sigma}_{\boldsymbol{x}}^{(k)} = \left[\frac{1}{(\sigma^2)^{(k)}}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A} + \text{diag}\left(\boldsymbol{\alpha}^{(k)}\right)\right]^{-1}. \tag{15}$$

Once the iterations converge, the probabilistic occupancy map estimate $\hat{\boldsymbol{x}}$ is the posterior mean given by (14). Then, we threshold $\hat{\boldsymbol{x}}$ to derive the binary occupancy map $\hat{\boldsymbol{x}}_{\text{b}}$. The resulting algorithm, referred to as PSI-aided Bayesian OGM (PSI-OGM), is summarized in Algorithm 1.

We note that our algorithm reduces to the standard SBL algorithm-based occupancy mapping when PSI is unavailable, i.e., the set $\mathcal{T}$ is empty.

Algorithm 1 PSI-OGM Algorithm

Input: point cloud with M points, N grid, PSI $\mathcal{T}$
Output: Occupancy map $\hat{\boldsymbol{x}}_{\text{b}}$
Parameters: $a_{\mathcal{T}}$, $b_{\mathcal{T}}$, $a_{\mathcal{T}^{\complement}}$, $b_{\mathcal{T}^{\complement}}$
Generate $\boldsymbol{y}$ and $\boldsymbol{A}$ based on point cloud using (2)
Set noise hyperparameters $c, d = 1 \times 10^{-4}$
Choose (a_n, b_n) using (6)
Initialize $\boldsymbol{\alpha}^{(0)} = \boldsymbol{0}$, $(\sigma^2)^{(0)} = 0.5$
repeat
    Update $\boldsymbol{\mu}^{(k)}$ and $\boldsymbol{\Sigma}_{\boldsymbol{x}}^{(k)}$ as per (14) and (15)
    Update $\boldsymbol{\alpha}^{(k+1)}$ according to (12)
    Update noise variance $(\sigma^2)^{(k+1)}$ according to (13)
until convergence
Set $\boldsymbol{\mu}^{(k-1)}$ as the estimated probabilistic map $\hat{\boldsymbol{x}}$
Threshold $\hat{\boldsymbol{x}}$ to obtain the binary map $\hat{\boldsymbol{x}}_{\text{b}}$

IV. Performance Evaluation

Experiments were conducted on both LiDAR and radar point clouds using two public datasets, nuScenes [18] and RADIATE [19]. The PSI set $\mathcal{T}$ is obtained from the camera images using our three-step PSI extraction technique described in Section III. For object detection, we use the YOLO networks. The YOLO network is trained using images from the nuScenes dataset and is employed to detect objects in the scenes from both the nuScenes and RADIATE datasets.

TABLE I
IoBB AND AS-NMSE VALUES FOR SCENE 125

Method	Target indices and IoBBs							Detected	AS-NMSE
	5	7	10	14	15	16	18		
PCSBL-OGM	0.138	**0.143**	0.056	0.000	0.091	0.182	0.500	6 / 7	0.022
SBL-OGM	0.138	0.095	0.083	0.000	0.061	0.182	0.500	6 / 7	0.018
PSI-OGM	**0.345**	**0.143**	**0.361**	**0.094**	**0.152**	**0.364**	**1.000**	**7** / 7	**0.014**

Three algorithms are compared in the simulations, namely, PCSBL-OGM [13], sparse Bayesian learning without prior information (SBL-OGM), and our PSI-OGM. Here, PCSBL-OGM is the state-of-the-art sparse-based OGM algorithm, which uses a pattern-coupled prior promoting block sparsity. We include SBL-OGM to demonstrate the effect of prior information because when no PSI is available, PSI-OGM reduces to the SBL-OGM algorithm.

For all methods, the noise hyperparameters are set to $c = d = 1 \times 10^{-4}$. For PCSBL-OGM, hyperparameters $a = 0.5$ and $b = 1 \times 10^{-4}$. For PSI-OGM, the following hyperparameter values are used: $a_{\mathcal{T}^\complement} = a = 0.5$, $b_{\mathcal{T}^\complement} = b = 1 \times 10^{-4}$, $a_{\mathcal{T}} = 0.25$, and $b_{\mathcal{T}} = 1$. The SBL-OGM is the same as PSI-OGM with $\mathcal{T} = \emptyset$, i.e., $a_n = a$ and $b_n = b$, for all values of n. The map resolution is 0.5×0.5 m. The threshold to generate the binary occupancy map is 0.3. The hyperparameter values for the evaluated algorithms are identical across all the considered datasets and scenes.

To assess the performance of OGM algorithms, we generate ground truth from the datasets by extracting the locations and dimensions of the boundary boxes of the obstacles. Two error metrics are used for performance analysis: the intersection over bounding box (IoBB) and angular scan normalized mean square error (AS-NMSE) [13]. The IoBB value is the fraction of the overlap between detected and true occupied cells of a given obstacle. A nonzero IoBB value indicates successful detection of the corresponding obstacle, quantifying the algorithm's detection quality. IoBB, however, does not capture the false alarms. On the other hand, AS-NMSE reflects a mapping algorithm's efficiency in minimizing false alarms and accurately detecting obstacle boundaries. To calculate the AS-NMSE, we project lines from the ego vehicle at predefined gridded angles until they intersect with obstacles or boundaries. The scan error along an angle is the difference between the distance from the ego vehicle to the first obstacle in the estimated and true maps. The AS-NMSE is the squared sum of these errors, where the sum is taken across all angles.

A. Results With LiDAR Data From nuScenes Scene 125

We consider frame 0 of scene 125 in nuScenes, which covers a region of 50×50 m. This scene has seven targets numbered 5, 7, 10, 14, 15, 16, and 18. Since the digital map of the drivable areas is provided by the nuScenes dataset, we consider only those cells within the driving areas of the map, eventually leading to $N = 4836$ grid cells. The visual target detection results and the extracted PSI are shown in Fig. 3(a) and (b), and results from the three algorithms are shown in Fig. 3(c)–(e). First, PSI extracted in Fig. 3(b) covers the region associated with all the targets, but it is erroneous as this region extends beyond the true boundaries of the obstacles. Fig. 3(c)–(e) shows that PSI-OGM exploits PSI and successfully detects target 14, which remains undetected by PCSBL-OGM and SBL-OGM. Although the other targets are detected by all the algorithms, PSI-OGM can detect them with clearer edges and more nonzero points. This observation is further supported by the high IoBB values in Table I. The higher IoBB values also indicate that PSI-OGM accounts for the clustered structure due to clustered PSI. Moreover, PSI-OGM handles the error in obstacle boundaries in the PSI extracted from the camera image, as evident from the low false detections in Fig. 3(e). This observation is corroborated by the better AS-NMSE offered by PSI-OGM compared with the other algorithms.

B. Statistical Results With LiDAR Data From 200 Scenes From nuScenes

The algorithms are evaluated using LiDAR data from 200 scenes sampled from the nuScenes dataset to compare their statistical performance. Two error metrics are used: the target detection rate and AS-NMSE. The detection rate for each target category records the proportion of detected targets (targets with nonzero IoBB) in the map relative to the total number of labeled targets in the scenes. It demonstrates the detection ability of the algorithms. The results are visualized as box plots as shown in Fig. 4. Based on the results, it can be observed that PSI-OGM outperforms the PCSBL-OGM and SBL-OGM algorithms in terms of detection rates. The PSI-OGM achieves a higher median value of 0.785 compared with 0.667 and 0.626, and a higher mean value of 0.784 compared to 0.697 and 0.634 for PCSBL-OGM and SBL-OGM, respectively. The variances of the detection rates are similar for all the algorithms, which are 0.055, 0.060, and 0.039 for PCSBL-OGM, SBL-OGM, and PSI-OGM, respectively. The results demonstrate that the PSI-OGM algorithm can effectively utilize PSI to improve the detection ability. The box plot of AS-NMSE results shows that PSI-OGM achieves both a lower median value of 0.007 compared to 0.014 and 0.011, and a lower average value of 0.012 compared with 0.020 and 0.017 for PCSBL-OGM and SBL-OGM, respectively. The statistical results of the AS-NMSE confirm a better performance of PSI-OGM in terms of drivable area detection compared to the PCSBL-OGM and SBL-OGM algorithms.

C. Results With LiDAR Data From RADIATE Scene City-3-7 Frame 132

We next present results on frame 132 of scene City-3-7 from the RADIATE dataset, as it contains large obstacles like

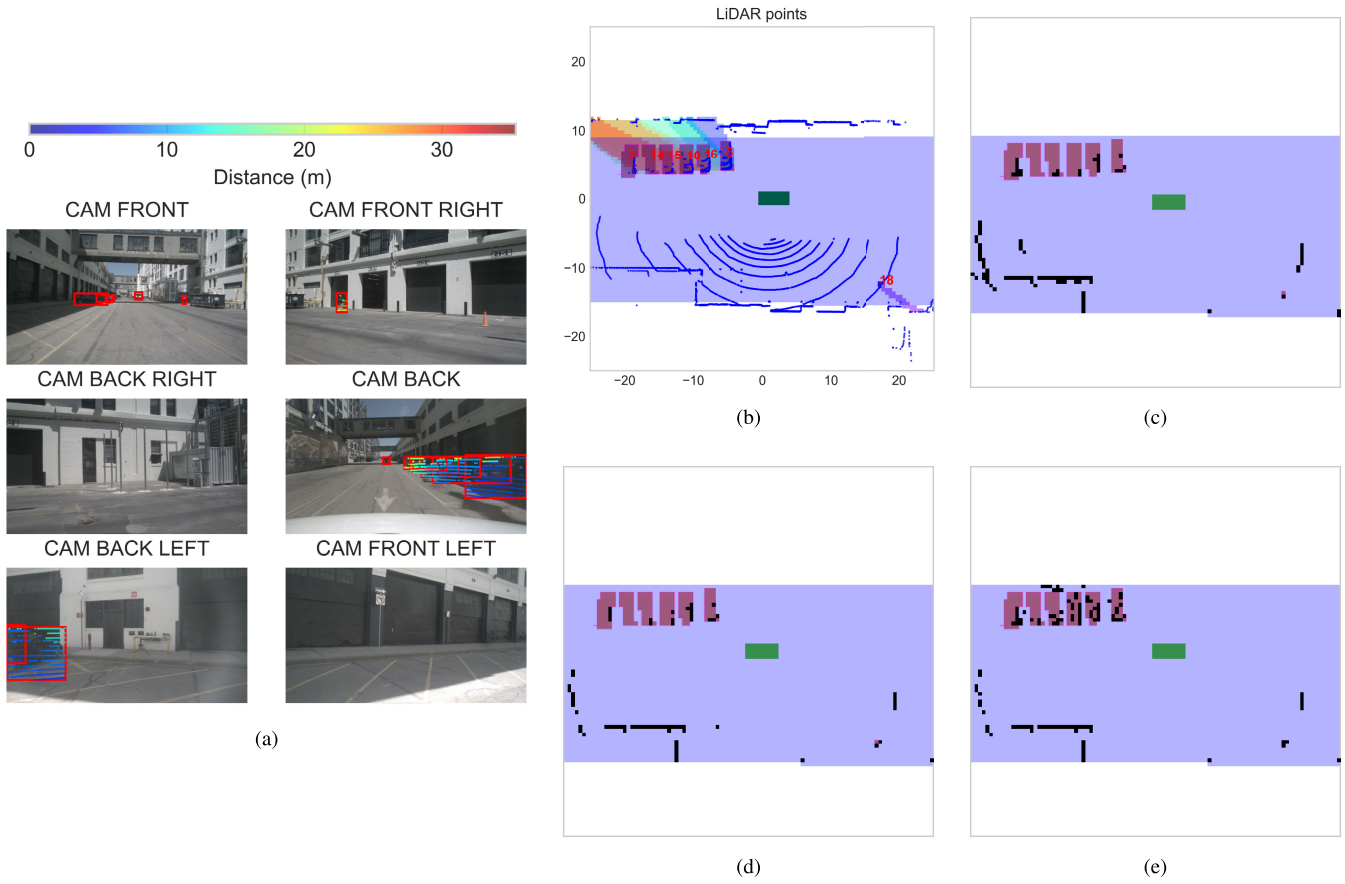


Fig. 3. Results based on LiDAR data from nuScenes scene 125 frame 0. (a) Target detection and LiDAR matching from a camera image. The colored points are the LiDAR points matched with the detected targets, where the distances are visualized by the color of the points. (b) PSI represented by different colored cells corresponding to various targets within the mapped area. The bounded red boxes are the ground-truth targets, where the red numbers are the target IDs. The green cells represent the ego vehicle. The blue background highlights the drivable area from the digital map. (c) Occupancy map from PCSBL-OGM. (d) Occupancy map from SBL-OGM. (e) Occupancy map from our PSI-OGM algorithm.

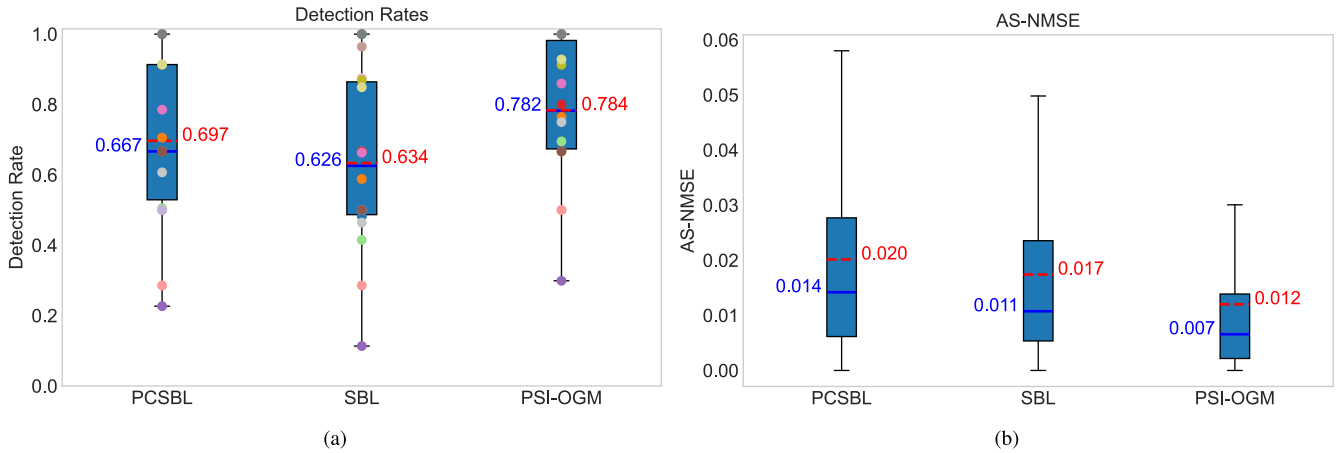


Fig. 4. Box plots using LiDAR data from 200 scenes from nuScenes in terms of (a) detection rates and (b) AS-NMSE. Solid blue lines show median values; dashed red lines show mean values. In (a), colored points show the detection rate of different categories of obstacles.

vehicles (targets 1, 2, 68, 69, 70, 71, and 72) and several small targets like pedestrians (targets 80, 81, 82, and 83). The map covers an area of 20 × 40 m, leading to $N = 3200$. The results are shown in Fig. 5, and the error values are given in Table II.

We note that the PSI extracted in Fig. 5(b) covers the region associated with all the targets visible in the camera image, although the YOLO network was not trained on this dataset. Target 68 cannot be identified from PSI, as it is not within the camera's field of view. Using this PSI, our PSI-OGM algorithm detects the same number of targets compared with PCSBL-OGM and one more target than SBL-OGM, as shown in Table II. For the detected targets inside the camera's field of view (i.e., targets 1, 69, 70, 71, 72, 81, 82, and 83),

TABLE II
IOBB AND AS-NMSE FOR SCENE CITY-3-7 FRAME 132

Method	Target indices and IoBBs											Detected	AS-NMSE
	1	2	68	69	70	71	72	80	81	82	83		
PCSBL-OGM	0.178	0.000	**0.194**	0.450	0.107	0.107	0.042	0.000	**0.167**	0.200	0.167	**9**/11	0.039
SBL-OGM	0.178	0.000	**0.194**	0.333	0.071	0.107	0.042	0.000	**0.167**	0.133	0.000	8/11	0.045
PSI-OGM	**0.222**	0.000	0.111	**0.467**	**0.143**	**0.250**	**0.167**	0.000	**0.167**	**0.267**	**0.417**	**9**/11	**0.036**

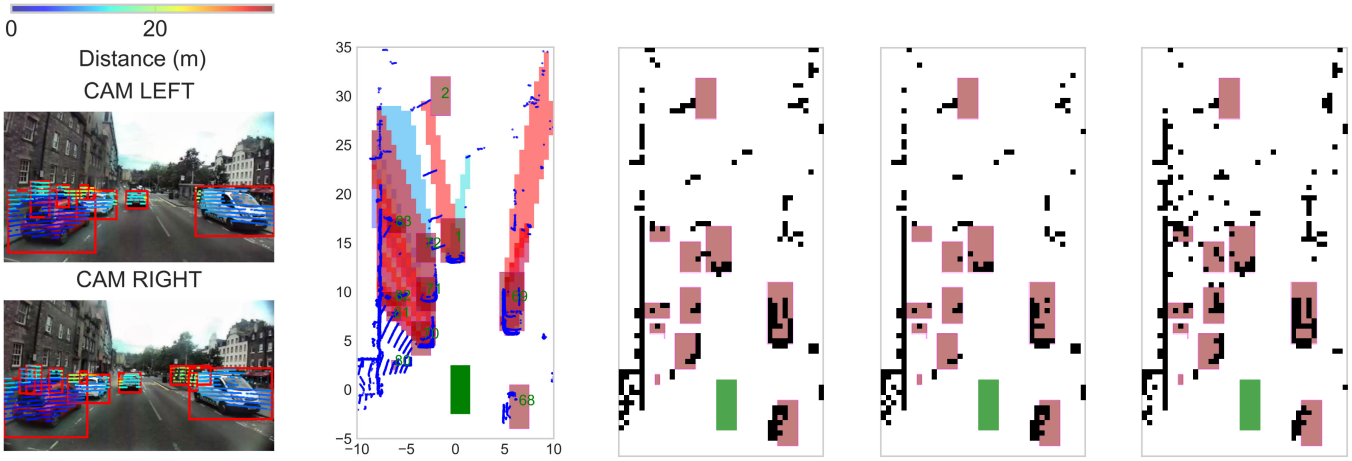


Fig. 5. Results based on LiDAR data from RADIATE City-3-7 frame 132. (a) Target detection and LiDAR matching from a camera image. The colored points are the LiDAR points matched with the detected targets, where the distances are visualized by the color of the points. (b) PSI represented by different colored cells corresponding to various targets within the mapped area. The bounded red boxes are the ground truth targets, where the green numbers are the target IDs. The green cells represent the ego vehicle. (c) Occupancy map from the PCSBL-OGM. (d) Occupancy map from the SBL-OGM. (e) Occupancy map from our PSI-OGM algorithm.

TABLE III
IOBB AND AS-NMSE FOR SCENE CITY-1-1 FRAME 240

Method	Target indices and IoBBs								Detected	AS-NMSE
	1	2	6	72	85	86	88	96		
PCSBL-OGM	0.111	0.000	0.000	0.222	0.261	0.357	**0.060**	**0.377**	6 / 8	0.144
SBL-OGM	0.037	0.000	0.000	0.130	0.065	0.089	0.000	0.077	5 / 8	0.407
PSI-OGM	**0.370**	**0.175**	0.000	**0.315**	**0.413**	**0.393**	**0.060**	0.185	**7** / 8	**0.124**

PSI-OGM achieves higher IoBB values compared to PCSBL-OGM and SBL-OGM. Specifically, PSI captures targets 82 and 83 as seen from Fig. 5(b), enabling their significantly better detection by PSI-OGM. Interestingly, YOLO does not directly detect target 72, but the erroneous PSI from target 71 enables its improved detection with our method. Target 2 is not detected by any method due to the lack of measurements. Similarly, target 80 is missed by all the methods as their measurements are corrupted by the ground scatter. The AS-NMSE values indicate that PSI-OGM has better general detection and lower false alarm rates than PCSBL-OGM and SBL-OGM in this driving scene. One exception of PSI-OGM having a lower IoBB than the other algorithms is target 68, located outside the PSI area. This problem may arise because PSI-OGM encourages the unoccupied grids in PSI to be sparser, leading to low IoBB values when PSI misses an obstacle. This issue needs to be addressed in the future.

D. Results With Radar Data From RADIATE Scene City-1-1 Frame 240

The front-view camera image and the PSI extracted from the scene are shown in Fig. 6(a) and (b). The mapping between the visual targets and the PSI map is assisted by LiDAR points instead of radar points because of the lack of elevation information from radar points. The mapping results are shown in Fig. 6(c)–(e), and the IOBB and AS-NMSE are shown in Table III. The results show that our PSI-OGM algorithm has the best detection ability among the three algorithms, successfully detecting seven out of eight targets. Notably, Target 2, which is missed by PCSBL-OGM and SBL-OGM due to insufficient point cloud reflections, is detected by PSI-OGM. Among the detected targets, six of them have higher IoBB values than the other two algorithms. For Target 96, PSI-OGM detects it with a lower IoBB value than PCSBL-OGM because the target is only partially covered by the PSI map. Furthermore, in this scene, the PCSBL-OGM demonstrates better detection ability than the SBL-OGM by exploiting the block-sparse structure of the map. Although PCSBL-OGM detects more false positives than SBL-OGM, it achieves a lower AS-NMSE because of its superior detection ability. Nonetheless, our PSI-OGM algorithm surpasses the detection ability of the PCSBL-OGM while concurrently reducing the false positives, leading to the best AS-NMSE value.

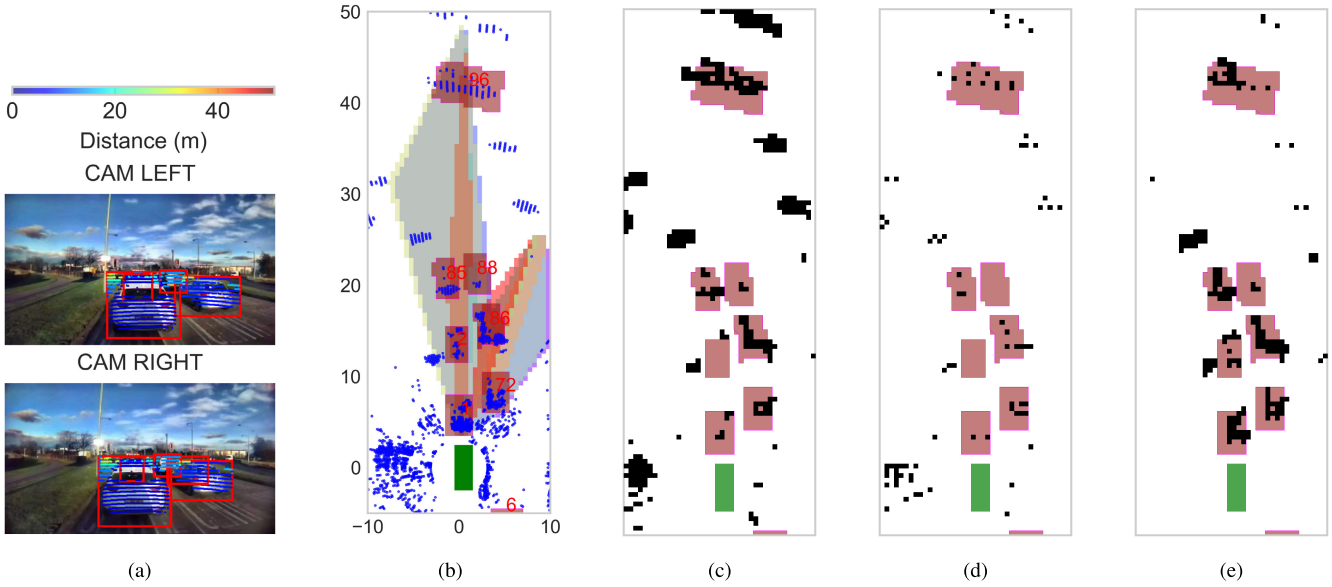


Fig. 6. Results based on radar data from RADIATE City-1-1 frame 240. (a) Target detection and LiDAR matching from a camera image. The colored points are the LiDAR points matched with the detected targets, where the distances are visualized by the color of the points. (b) PSI represented by different colored cells corresponding to various targets within the mapped area. The bounded red boxes are the ground truth targets, where the red numbers are the target IDs. The green cells represent the ego vehicle. (c) Occupancy map from the PCSBL-OGM. (d) Occupancy map from the SBL-OGM. (e) Occupancy map from our PSI-OGM algorithm.

To summarize, the results and analysis confirm that our PSI-OGM algorithm effectively uses the prior information of the camera to enhance the detection capabilities of OGM through adaptive hyperparameter adjustments. Prior information also enables the algorithm to implicitly integrate obstacle block structures. At the same time, our algorithm can suppress false positives, providing better area detection.

V. Conclusion

In this work, we present a camera-aided support-informed OGM algorithm that utilizes PSI extracted from camera-based target detection. Based on the PSI, our PSI-OGM algorithm assigns hyperparameters that promote the existence of obstacles for cells labeled as potential obstacles. We conducted simulations on multiple scenes from the real-world datasets nuScenes and RADIATE. With possibly erroneous PSI from the camera, our algorithm resulted in higher IoBB values, indicating better detection ability, and lower AS-NMSE, indicating better drivable area detection. This observation is consistent across multiple scenes from both datasets using either LiDAR or radar point clouds for mapping. Statistical results based on LiDAR data from 200 scenes in nuScenes demonstrated that our PSI-OGM algorithm outperforms the state-of-the-art PCSBL-OGM and SBL-OGM algorithms in terms of detection rates and AS-NMSE. The resulting maps also show that our algorithm can implicitly incorporate the spatial relationship of the existence of obstacles through PSI.

Our work opens several interesting avenues for future work. Our algorithm currently faces limitations in detecting targets that are partially or not covered by the PSI, which could be addressed by integrating the PCSBL structure into our approach. Also, in this work, the camera information is used to inform either LiDAR or radar-based OGM through the hyperprior, which can also be extended to combine camera information with LiDAR-radar fusion. Extending from 2-D to 3-D OGM is also valuable for addressing more complex environments, including those with overhanging or subterranean obstacles. Furthermore, our algorithm focuses on single-frame static OGM with camera knowledge, and extensions to dynamic OGM that consider interframe structure and dynamic knowledge would be a potential direction for future work.

References

[1] H. Zhu, K.-V. Yuen, L. Mihaylova, and H. Leung, "Overview of environment perception for intelligent vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 10, pp. 2584–2601, Oct. 2017.

[2] A. Pandharipande et al., "Sensing and machine learning for automotive perception: A review," *IEEE Sensors J.*, vol. 23, no. 11, pp. 11097–11115, Jun. 2023.

[3] K. Lee and D. Kum, "Collision avoidance/mitigation system: Motion planning of autonomous vehicle via predictive occupancy map," *IEEE Access*, vol. 7, pp. 52846–52857, 2019.

[4] S. Steyer, G. Tanzmeister, and D. Wollherr, "Object tracking based on evidential dynamic occupancy grids in urban environments," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 1064–1070.

[5] Y. Li and J. Ibanez-Guzman, "LiDAR for autonomous driving: The principles, challenges, and trends for automotive LiDAR and perception systems," *IEEE Signal Process. Mag.*, vol. 37, no. 4, pp. 50–61, Jul. 2020.

[6] F. Engels, P. Heidenreich, M. Wintermantel, L. Stäcker, M. Al Kadi, and A. M. Zoubir, "Automotive radar signal processing: Research directions and practical challenges," *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 4, pp. 865–878, Jun. 2021.

[7] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *Computer*, vol. 22, no. 6, pp. 46–57, Jun. 1989.

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

[9] M. A. Paskin and S. Thrun, "Robotic mapping with polygonal random fields," in *Proc. Conf. Uncertainty Artif. Intell.*, Jul. 2012, pp. 450–458.

[10] S. Thrun, "Learning occupancy grid maps with forward sensor models," *Auto. Robots*, vol. 15, no. 2, pp. 111–127, Sep. 2003.

[11] S. T. O'Callaghan and F. T. Ramos, "Gaussian process occupancy maps," *Int. J. Robot. Res.*, vol. 31, no. 1, pp. 42–62, Jan. 2012.

[12] K. Doherty, J. Wang, and B. Englot, "Bayesian generalized kernel inference for occupancy map prediction," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 3118–3124.

[13] Ç. Önen, A. Pandharipande, G. Joseph, and N. J. Myers, "Occupancy grid mapping for automotive driving exploiting clustered sparsity," *IEEE Sensors J.*, vol. 24, no. 7, pp. 9240–9250, Apr. 2024.

[14] J. Fang, Y. Shen, H. Li, and P. Wang, "Pattern-coupled sparse Bayesian learning for recovery of block-sparse signals," *IEEE Trans. Signal Process.*, vol. 63, no. 2, pp. 360–372, Jan. 2015.

[15] P. Zhai, G. Joseph, N. J. Myers, Ç. Önen, and A. Pandharipande, "Spatial sparsity-aware radar-LiDAR fusion for occupancy grid mapping in automotive driving," *IEEE Sensors J.*, vol. 25, no. 17, pp. 33328–33338, Sep. 2025.

[16] J. Sock, J. Kim, J. Min, and K. Kwak, "Probabilistic traversability map generation using 3D-LiDAR and camera," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 5631–5637.

[17] K. U. Mazher, S. Annaluru, A. Mezghani, and R. W. Heath, "Exploiting structural information in camera aided radar parameter estimation," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Nov. 2019, pp. 1–5.

[18] H. Caesar et al., "NuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11618–11628.

[19] M. Sheeny, E. De Pellegrin, S. Mukherjee, A. Ahrabian, S. Wang, and A. Wallace, "RADIATE: A radar dataset for automotive perception in bad weather," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 1–7.

[20] J. Domhof, J. F. P. Kooij, and D. M. Gavrila, "An extrinsic calibration tool for radar, camera and LiDAR," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 8107–8113.

[21] X. Meng, D. Duan, and T. Feng, "Multi-vehicle multi-sensor occupancy grid map fusion in vehicular networks," *IET Commun.*, vol. 16, no. 1, pp. 67–74, Jan. 2022.

[22] Y.-M. Kim, Y.-G. Kim, S.-Y. Son, S.-Y. Lim, B.-Y. Choi, and D.-H. Choi, "Review of recent automated pothole-detection methods," *Appl. Sci.*, vol. 12, no. 11, p. 5320, May 2022.

[23] S. M. Patole, M. Torlak, D. Wang, and M. Ali, "Automotive radars: A review of signal processing techniques," *IEEE Signal Process. Mag.*, vol. 34, no. 2, pp. 22–35, Mar. 2017.

[24] J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Syst. J.*, vol. 4, no. 1, pp. 25–30, Jul. 1965.

[25] E. Kaufman, T. Lee, Z. Ai, and I. S. Moskowitz, "Bayesian occupancy grid mapping via an exact inverse sensor model," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2016, pp. 5709–5715.

[26] G. Jocher, A. Chaurasia, and J. Qiu. (Jan. 2023). *Ultralytics YOLO*. [Online]. Available: https://github.com/ultralytics/ultralytics

[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[28] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: Analysis, applications, and prospects," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022.

[29] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, no. 3, pp. 211–244, Sep. 2001.

[30] J. Fang, Y. Shen, F. Li, H. Li, and Z. Chen, "Support knowledge-aided sparse Bayesian learning for compressed sensing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 3786–3790.

[31] M. Shekaramiz, T. K. Moon, and J. H. Gunther, "Sparse Bayesian learning boosted by partial erroneous support knowledge," in *Proc. 50th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2016, pp. 389–393.

Peiyuan Zhai (Graduate Student Member, IEEE) received the B.Eng. degree in information engineering from Xi'an Jiaotong University, Xi'an, China, in 2020, and the M.S. degree in electrical engineering from Delft University of Technology, Delft, The Netherlands, in 2022, where he is currently pursuing the Ph.D. degree with the Signal Processing Systems Group.

His research interests include automotive perception, multimodal sensing, and sparse Bayesian learning.

Geethu Joseph (Senior Member, IEEE) received the Ph.D. degree in electrical communication engineering from Indian Institute of Science, Bengaluru, India, in 2019.

She is currently a Tenured Assistant Professor with the Signal Processing Systems Group, Delft University of Technology, Delft, The Netherlands. Her research interests include statistical signal processing, network control, and machine learning.

Nitin Jonathan Myers (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from The University of Texas at Austin, Austin, TX, USA, in 2020.

He is currently a Tenured Assistant Professor with Delft Center for Systems and Control, Delft University of Technology, Delft, The Netherlands. His research interests include optimization and multidimensional signal processing, with applications to communications and sensing.

Ashish Pandharipande (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Iowa, Iowa City, IA, USA, in 2002.

He is currently a Technical Director with NXP Semiconductors, Eindhoven, The Netherlands, and a part-time Full Professor with Eindhoven University of Technology, Eindhoven. His research interests include sensing, machine learning, data analytics, and their applications in domains such as autonomous mobility, smart lighting systems, and cognitive wireless systems.