Detection of Parked Vehicles from a Radar Based Occupancy Grid

Renaud Dubé¹, Markus Hahn², Markus Schütz², Jürgen Dickmann² and Denis Gingras¹

Abstract—For autonomous parking applications to become possible, knowledge about the parking environment is required. Therefore, a real-time algorithm for detecting parked vehicles from radar data is presented. These data are first accumulated in an occupancy grid from which objects are detected by applying techniques borrowed from the computer vision field. Two random forest classifiers are trained to recognize two categories of objects: parallel-parked vehicles and cross-parked vehicles. Performances of the classifiers are evaluated as well as the capacity of the complete system to detect parked vehicles in real world scenarios.

I. INTRODUCTION

Object detection and scene understanding are key components in Advanced Driver Assistance Systems. In order to enable autonomous driving in semi-structured environments such as parking lots, low level mapping must be further processed to achieve a high level of awareness [1] and [2]. According to Thrun [3] object maps offer several advantages over their metric and topological counterparts. Among others, they better represent situations where static objects can become dynamic and they are closer related to the human perception of the environment. Object maps particularly adapt well to situations where many instances of objects of the same type are present in the environment [4], which supports the application for detecting parked vehicles in a parking lot.

A parked vehicle detection algorithm could be particularly useful for applications such as autonomous valet parking. As an example, a landmark-based localization algorithm could reduce its dependence on a particular landmark, knowing that it represents a vehicle which may not always be static. Similarly, this information could be useful for collision mitigation knowing that a vehicle, compared to a tree, can potentially move. The recognition of parked vehicles can also lead to the detection of free parking spots as demonstrated by [2].

Previous work has been done in order to detect parked vehicles from laser range data. Keat et al. [5] proposed an algorithm to extract vehicles from a SICK laser scan assuming that vehicles will be represented by an “L” shape. This assumption does not hold in dense cross-parking area where only vehicles front and back-ends bumpers are visible from laser data. Also, this strategy could not distinguish between a parked vehicle and the corner of a wall. Zhou et al. [2] demonstrated that it is possible to extract vehicle bumper from a laser scan. This solution solves the problem of dense cross parking but it is not applicable in parallel parking situations.


In this paper we present a real-time algorithm capable of detecting both parallel and cross-parked vehicles from radar data. A radar-based occupancy grid is built according to Elfes [8]. Parked vehicle candidates are then extracted from this grid. These candidates are described and classified in order to assert the presence of vehicles. The proposed method addresses the challenge of detecting both parallel and cross-parked vehicles. A distinction is made between cross-parked vehicles, which stand perpendicular to the lane direction, and parallel-parked vehicle, which are obviously parallel to the lane direction. Finally, radar sensors are considered in our work because this technology is already present in the high-end automotive market. Radar technology is also popular due to the reliable performances offered in different weather conditions and its relatively low price compared to laser scanners.

The paper is structured as follows: Section II introduces the proposed algorithm for detecting parked vehicles from radar data. The experiment described in section III measures the system performances. Finally, section IV concludes the discussion and makes a brief outlook for future work.

II. ALGORITHM DESCRIPTION

The proposed parked vehicle detection algorithm consists of four main steps: occupancy grid generation, candidates selection, features extraction and classification.

A. Occupancy grid generation

The goal of this first step is to compute an occupancy grid, as introduced by Elfes [8], in order to accumulate the radar data over time. Our occupancy grid map \(M_k = \{m_1, m_2, \ldots, m_N\}\) consists of \(N\) grid cells \(m_i\), which represent the environment as a 2D-space with equally sized cells. For the current parked vehicle detection application, the grid cell size is fixed to 0.1m \(\times\) 0.1m. Each cell is a probabilistic variable describing the occupancy of this cell. Assuming that \(M_k\) is the grid map at time \(k\) and that the grid cells are independent one to another, the occupancy grid map can be modeled as a posterior probability:

\[
P(M_k | Z_{1:k}, X_{1:k}) \propto \prod_i P(m_i | Z_{1:k}, X_{1:k})
\]

(1)

where \(P(m_i | Z_{1:k}, X_{1:k})\) is the inverse sensor model, which describes the probability of occupancy of the \(i\)th cell, given...
the measurements \(Z_{1..k}\) and the dynamic object state \(X_{1..k}\). Each measurement consists of \(n\) radar detections \(Z_j = \{z_{1,j}, z_{2,j}, \ldots, z_{n,j}\}\).

The occupancy value of each cell is calculated by a binary bayesian filter. In practice, the log posterior is used to integrate new measurements efficiently. Instead of performing multiplications, the usage of the log odds ratio simplifies the calculation to additions and avoids instabilities of calculating probabilities near zero or one. The log odds occupancy grid map is formalized as:

\[
L_k(m_i) = \log \frac{P(m_i|Z_{1:k}, X_{1:k})}{1 - P(m_i|Z_{1:k}, X_{1:k})} \quad (2)
\]

The recursive formulation of map update in log odds ratio form is given by [9] :

\[
L_k(m_i) = L_{k-1}(m_i) + \log \frac{P(m_i|Z_{1:k}, X_{1:k})}{1 - P(m_i|Z_{1:k}, X_{1:k})} - L_0(m_i) \quad (3)
\]

where \(L_{k-1}(m_i)\) and \(L_0(m_i)\) are the previous and prior log odds values of grid cell \(i\). Assuming that no prior knowledge is available, the prior probability of unknown cells is set to \(P_0(m_i) = 0.5\), the above equation produces the prior log odds ratio \(L_0 = 0\). The log odds formulation of equation (2) can be inverted to obtain the corresponding probability of map \(M_k\). A radar based occupancy map is displayed in Fig 1. The map is projected into the image of our documentation camera so that one can appreciate how the radars perceive the environment.

B. Object detection and candidate selection

This second step selects parked vehicles candidates which will be described and classified in later steps. Two lists of candidates are built: one for cross-parked vehicles and one for parallel-parked vehicles. This step can be further divided into three substeps: the detection of interest objects, the estimation of these objects directions and the final choice of candidates.

1) Object detection: In order to detect objects of interest, the map \(M\) is converted to a binary grid \(B\) using threshold \(\alpha\) as described in equations (4) and (5).

\[
b_i = 1 \quad \text{if} \quad m_i \geq \alpha \quad (4)
\]

\[
b_i = 0 \quad \text{if} \quad m_i < \alpha \quad (5)
\]

This grid \(b\) is clustered by a 8-n connected component analysis providing an initial list of objects of interest. Objects containing too few cells are removed from this list since they can hardly represent a parked vehicle leading to the list \(O = \{O_1, O_2, \ldots, O_N\}\).

2) Direction estimation: From the analysis of different parking scenarios, it is clear that the direction of parked vehicles is often correlated to the direction of the trajectory or to the main orientation of the lane structure. The problem of determining the vehicle direction is thus simplified by making the hypothesis that they are either perpendicular or parallel to the trajectory direction. Therefore, a direction \(\psi_i\) which is perpendicular to the trajectory and pointing toward the object \(O_i\) will be associated to every object.

3) Candidate selection: The candidate selection procedure differs for cross-parked and parallel-parked vehicles. The description is divided accordingly.

Cross-parked vehicles: In radar based occupancy grids representing dense parking lots, most often only one part of a cross-parked vehicle is observable. Knowing the direction \(\psi_i\) of an object \(O_i\), a rotation is made toward a normalized direction, yielding a cutout map \(c_i\) as displayed in Fig 2.
Note that the cutout map of each candidate is such oriented in order to provide the classifiers with consistent inputs. In order to determine if an object is a candidate for cross-parked vehicle, binary rules are applied to the dimensions of the cutout map \( c_i \). This decision yields a list \( C \) of \( Q \) cross-parked vehicles candidates:

\[
C = \{C_1, C_2, \ldots, C_Q\} \text{ where } C_u = \{O_i, c_i\}, i \in 1 \ldots N
\]  

(6)

**Parallel-parked vehicles**: Extracting candidates for parallel-parked vehicles is more challenging since two different situations need to be considered. In its simplest form, the shape of a parallel vehicle can be completely covered by a single object in the occupancy grid. Applying similar dimension decision on the cutout map of each object gives an initial list of parallel-parked vehicle candidates.

\[
P = \{P_1, P_2, \ldots, P_R\} \text{ where } P_u = \{O_i, c_i\}, i \in 1 \ldots N
\]  

(7)

A parallel-parked vehicle can also be represented by two L-shaped distinct objects in the grid: one for the front of the vehicle and one for the back. This situation occurs at higher driving speeds or when driving very close to the vehicle, therefore reducing the number of radar detection of the target. In this case, every pair of close objects is considered. The same set of dimension rules determine if the cutout maps resulting from these object pairs represent parallel-vehicle candidates.

\[
P_{u, u} = R + 1, R + 2, \ldots, S
\]  

(8)

where \( P_{u, u} = \{O_i, O_j, c_u\}, i \neq j, i \in 1 \ldots N, j \in 1 \ldots N \).  

(9)

At this point, it is relevant to illustrate in Fig 3 some cutout map samples extracted from the radar data. Note in Fig 3 (d) that two cross-parked vehicles make a parallel-parked vehicle candidate. It is important to understand that the classifier for parallel-parked vehicles will also see instances of cross-parked vehicles and will need to classify those as "false parallel-parked vehicles". Similarly, the top left sample of Fig 3 (b) represents a part of a parallel-parked vehicle and the cross-parked vehicle classifier will have to assign the "false cross-parked vehicle" label to it.

**C. Feature extraction**

In order to determine if a candidate is effectively a parked vehicle, its cutout map first needs to be processed. The purpose of feature extraction is to compress the raw data and build candidates signatures suitable for classification. Given the input cutout map \( c \) associated to each candidate, four descriptors are computed resulting in feature vector \( f = [f_1 \ f_2 \ f_3 \ f_4] \). Both types of candidates are described using the same set of features.

\( f_1 \) **Candidate size**: The height \( h \) and the width \( w \) of the cutout map (in meters).

\[
f_1 = [h \ w]
\]  

(10)

\( f_2 \) **Candidate response strength**: The average value of every cell of the \( n \times m \) cutout map \( c \).

\[
f_2 = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}
\]  

(11)

\( f_3 \) **Candidate shape**: This feature, along with \( f_4 \), is used to represent the frequent "U" or "L" shape of parked vehicles in a radar based occupancy grid. The sum of every cell of the cutout map along its horizontal dimension is computed:

\[
b = (b_1 \ b_2 \ \ldots \ b_n) \text{ where } b_u = \sum_{i=1}^{m} c_{iu}
\]  

(12)

The resulting vector \( b \) is sub-sampled to a vector \( f_3 \) of definite length (1x10) and normalized. The sub-sampling to
An example of this feature is shown in Fig. 5. Note the nonlinearity between 0 and \( \pi \) observable at bins 5 and 6 of Fig 5. The classifier presented in the following section will deal with this nonlinearity.

### D. Classification

The candidates are classified using two trained random forest classifiers as introduced by Breiman [12]. Two classifiers are used: one for categorizing the candidates from list \( C \) and one for categorizing the candidates from list \( P \). The classifiers are first presented followed by the supervised training process and the mean of combining the classification results.

1) Random forest classifier: The idea behind this classifier is to construct a multitude of different decision trees and to have them vote for the winning class. Randomness should be introduced in the training of each tree in order to lessen the correlation between them. Aside from this correlation level, classification performances of the forest are also defined by the individual strength of each tree.

According to [12], random forest offers classification performances similar to the AdaBoost algorithm. Also, it is less sensitive to noise in the output label (such as a misslabeled candidate) since it does not concentrate its efforts on misclassified candidates. Finally, random forest classifiers can be used in order to evaluate the importance of particular features. In this manner, a longer list of considered features was reduced to the four features described in the previous section.

2) Training: In order to balance the correlation level and the individual strength of each tree, Breiman [12] proposes two solutions for inserting randomness during the supervised training process. First, each tree should be trained using a bootstrapped subset of the training data set. Second, every tree node should be trained using a random subset of features.

The actual implementation of the random forest classifier is based on work from [13]. Some parameters are adjusted including the number of trees in the forest (50) and the number of random features for training each node (2). Also during bootstrapping of the training data, a weighted sampling is performed in order to balance between the two classes. That is, candidates from a class with lower a priori probability have higher chances to be selected during training.

Given a set of features \( f \) the random forest classifiers assign a classification score \( w \) of being a parked vehicle to each candidate of lists \( C \) and \( P \).

3) Classification results combination: In order to combine the results between the two classifiers, it is verified if candidates with a classification score \( w \) above 50% overlap with each other. In the case two positively classified candidates overlap, the candidate with the highest classification score is confirmed to be the parked vehicle at this position. In future work, we are interested to explore the application of Dempster-Shafer theory to resolve this kind of ambiguity in the decision process.
III. EXPERIMENT

A. Vehicle presentation

The Mercedes-Benz S 500 INTELLIGENT DRIVE research vehicle is considered for this experiment. It is equipped with four short range radar sensors at the vehicle corners, providing a 360° environment perception. The sensors operate at 76 GHz and have a coverage up to 40 m with an accuracy below 0.15 m. The sensors single field of view is 140° with an accuracy below 1°. The research vehicle and its sensors configuration are illustrated in Fig 6.

B. Classification performances

Radar data have been recorded during six different sequences ranging from a drive in a parking crowded with cross-parked vehicles to a drive on an urban street with dispersed parallel-parked vehicles. From these data, lists of candidate have been pre-processed and carefully labeled. As a result, two data sets were created. The cross-parked vehicles set contains 1119 samples (268 true vehicles and 851 false vehicles) while the parallel-parked vehicles set contains 1342 samples (300 true vehicles and 1042 false vehicles). Several samples are illustrated in Fig 3.

In order to predict the performance of the classifiers, a repeated random sub-sampling validation was performed. Each data set was randomly sampled to a training set (2/3) and a validation set (1/3). Classifiers performances were evaluated using this division and the process is repeated for a total of 100 rounds. This type of cross-validation offers the possibility to select the training / validation ratio. However, some samples may never be considered in the validation phase.

Over these 100 rounds, the classification results were averaged and summarized in Table I and Table II. Also, the respective averaged ROC curves were computed as shown in Fig. 7. Finally, the average accuracy of the classifiers is 97.9% ± 0.7% for cross-parked vehicles and 96.9% ± 0.7% for parallel-parked vehicles.

C. Experiment scenarios

Fig 8 depicts a parking scenario where both cross-parked vehicles and parallel-parked vehicles surround the trajectory. The green line represent the moving vehicle trajectory while the red and green boxes identify parked vehicles which are detected by the system. It is interesting to note that the classifiers correctly distinguished the two cross-parked SMART vehicles from the larger parallel-parked vehicle at the top right.

### Table I

<table>
<thead>
<tr>
<th></th>
<th>Classified as vehicle</th>
<th>Classified as non-vehicle</th>
</tr>
</thead>
<tbody>
<tr>
<td>True vehicle</td>
<td>95.9% ± 2.0%</td>
<td>4.1% ± 2.0%</td>
</tr>
<tr>
<td>True non-vehicle</td>
<td>1.5% ± 0.7%</td>
<td>98.5% ± 0.7%</td>
</tr>
</tbody>
</table>

### Table II

<table>
<thead>
<tr>
<th></th>
<th>Classified as vehicle</th>
<th>Classified as non-vehicle</th>
</tr>
</thead>
<tbody>
<tr>
<td>True vehicle</td>
<td>95.1% ± 2.0%</td>
<td>4.9% ± 2.0%</td>
</tr>
<tr>
<td>True non-vehicle</td>
<td>2.5% ± 0.9%</td>
<td>97.5% ± 0.9%</td>
</tr>
</tbody>
</table>

Fig. 7. Average ROC curves for cross-parked and parallel-parked vehicles classification
Installed at the back of the vehicle. The parked vehicles were detected in order to make use of the radars for mobile robots an object based approach,” Robotics and Autonomous Systems, vol. 55, no. 5, pp. 359–371, 2007.

This situation will not cover every case but important a priori knowledge can be obtained this way. Future work will mine the vehicle orientation according to the lane orientation. This paper presented an algorithm for detecting both parallel and cross-parked vehicles from radar data. Out of these data is built an occupancy grid where candidates are extracted. Two random forest classifiers are used to assert the presence of parked vehicles in two different modes: cross-parked and parallel-parked.

During the algorithm development, two hypotheses were made. First, the parked vehicles orientation was assumed to be correlated to the direction of the trajectory. Instead of using the trajectory as an input, one could determine the structure of the parking such as presented by [1] and determine the vehicle orientation according to the lane orientation. This situation will not cover every case but important a priori knowledge can be obtained this way. Future work will explore L-shape fitting and orientation estimation by classification in order to improve vehicle detection performance.

Second, only the parked vehicles behind the current moving vehicle were detected in order to make use of the radars installed at the back of the vehicle. The parked vehicles were labeled after a passage of the moving vehicle so that the classifiers were trained with prolonged integrated data. This issue could be solved in a next step by including time dependant features as well as information concerning the objects of interest positions relative to the moving vehicle position.

Finally, this vehicle detection approach could be adapted in order to detect other common objects found in parking lots such as curbstones, posts and charging stations for HEVs. This could lead to the implementation of a cognitive map such as presented by [14]. A cognitive map could enable the vehicle to better interpret and understand the parking lot environment. As introduced, knowledge about the presence of parked vehicle could be particularly useful for several tasks of autonomous parking such as collision mitigation, localization and parking spot detection.

ACKNOWLEDGMENT

We are thankful to Daimler AG who supported this work.

REFERENCES


