

# Detection of Parking Spots Using 2D Range Data

Jifu Zhou, Luis E. Navarro-Serment and Martial Hebert

**Abstract**—This paper addresses the problem of reliably detecting parking spots in semi-filled parking lots using on-board laser line scanners. In order to identify parking spots, one needs to detect parked vehicles and interpret the parking environment. Our approach uses a supervised learning technique to achieve vehicle detection by identifying vehicle bumpers from laser range scans. In particular, we use AdaBoost to train a classifier based on relevant geometric features of data segments that correspond to car bumpers. Using the detected bumpers as landmarks of vehicle hypotheses, our algorithm constructs a topological graph representing the structure of the parking space. Spatial analysis is then performed on the topological graph to identify potential parking spots. Algorithm performance is evaluated through a series of experimental tests.

## I. INTRODUCTION

There has been an increased need and interests in intelligent driving systems within the past decade, as exemplified by projects such as the DARPA Urban Challenge [1]. For intelligent vehicles, the ability to park autonomously is one of the most fundamental requirements. In particular, we are developing a *Virtual Valet* system [2] for vehicles to park themselves that mimics regular valet service. The major requirement for this system is to operate autonomously without any infrastructure support.

In this paper, we focus on detecting available spots in semi-filled parking spaces using an on-board laser line scanner. We consider a typical semi-filled parking space as one in which all vehicles are parked side-by-side as shown in Fig. 1, and where several spots are available among these parked vehicles. In order to identify these spots, one needs to detect parked vehicles, analyze the parking patterns, and interpret their spatial relationship.

Parking lots are typical semi-structured environments. Perception in such environments involves not only low-level detection of targets, but also high-level inference of the underlying structures. Usually, in a semi-filled parking lot, due to occlusions and limited space, sensors can only obtain limited measurements of vehicles, typically from only one side of them. If the low-level detection of vehicles is restricted or infeasible, the high-level spacial inference may fail to reveal the underlying structure of the parking environment. The identification of available parking spots,

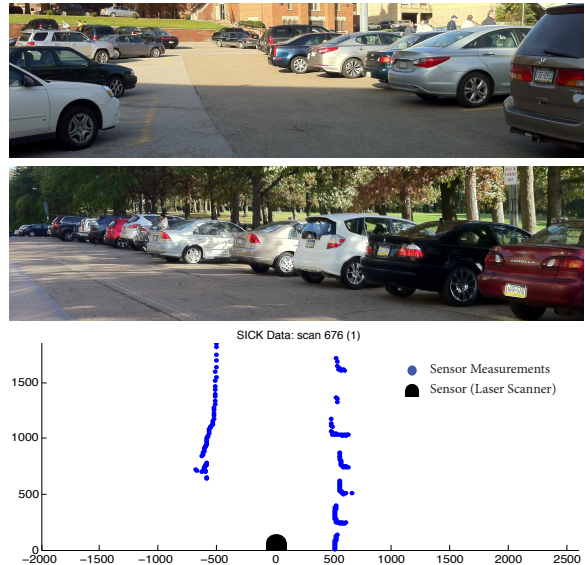


Fig. 1. Examples of Semi-Filled Parking Lots and View from Sensor; the parking space of the second figure corresponds to the sensor data plot.

which relies on the precise interpretation of the structure, can become a significant challenge.

For the *Virtual Valet* system, we developed an algorithm based on the assumption that the ability to detect vehicles from a partial contour enables the interpretation of the parking structure in relatively packed environments. Since the bumper is the least occluded side of vehicles in typical parking lots where cars are parked side-by-side (Fig. 1), we attempt to identify parked vehicles via detecting bumpers. Upon reliable detection of bumpers, our algorithm identifies available spots by interpreting the spatial structure of the given parking lot. The structure is represented by a spacial distribution of vehicle hypotheses constructed from detected bumpers.

The paper is organized as follows: Section II reviews the related work. Section III describes our algorithm that addresses the detection of vehicle bumpers and available spots in semi-filled parking space. Section IV presents the evaluation of our algorithm performance from a series of experiments. Finally, Section V summarizes the results and concludes the paper.

## II. RELATED WORK

A number of research efforts have been made for autonomous driving in structured, unstructured, and semi-structured environments. Structured environments [1] such

Jifu Zhou is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, USA jifuz@andrew.cmu.edu

Luis E. Navarro-Serment is with the Robotics Institute, Carnegie Mellon University, Pittsburgh PA 15213, USA lenscmu@ri.cmu.edu

Martial Hebert is with the Robotics Institute, Carnegie Mellon University, Pittsburgh PA 15213, USA hebert@ri.cmu.edu

as highways are usually imposed with explicit topological graphs, and perception usually does not require uncovering any underlying structures. Unstructured environments [3] which are typically encountered off paved roads have no explicit topological graphs imposed, and the vehicles are not as constrained when driving, thus perception in such environment usually addresses traversability analysis. Approaches used in these two cases are inadequate in semi-structured environments [1], [4], [5], which require both low-level detection of targets and high-level inference of the underlying structures.

In semi-filled parking spaces, the targets are parked vehicles and parking spots. While technologies such as loop detectors can be used to sense vehicles, they are not always available as a part of the infrastructure in every potential parking space. Many driving assistance systems use ultrasonic sensors or radar to detect targets [6], [7], but they are not applicable in the autonomous driving domain due to low accuracy. Several approaches have been attempted addressing detection of vehicles and parking spots, including detection with optical sensors (i.e. cameras), with sensors fusing of both laser and camera, and with only laser range scanners [1], [5], [8].

Optical sensor-based and sensor fusion-based approaches rely on cameras for perception. Cameras can capture a tremendous wealth of visual information, which can be used to identify more subtle changes and distinctions between objects. However, they are sensitive to lighting. Furthermore, in semi-filled parking lots (Fig. 1), in addition to the unstable lighting conditions, the targets occlude each other exposing only partial contours, which affects the camera's ability to accurately perceive the targets. On the other hand, though lasers can only obtain range data, they are largely unaffected by lighting. In our application, we chose to take advantage of the LADAR's ability to provide accurate range measurements at high rates, which makes it easier to determine the exact position of objects surrounding the vehicle.

In the DARPA Urban Challenge [1], many laser based techniques were used for autonomous driving in both structured and semi-structured environments including parking spaces. However, since the locations of parking lots and spots are known beforehand, provided by the high resolution Road Network Description File (RNDF) map, the techniques demonstrated focus mainly on navigation or motion planning of parking maneuvers. In real life, autonomous vehicles not always have access to such information. In fact, in the Challenge, the RNDF assumed that the parking spots were empty, which indeed affected the performance of the RRT path planner in one of the vehicles at a parking check point. This incidence indicates that even when precise prior information is given, due to the dynamical nature of the environment and the potential misinterpretation of the prior information, autonomous vehicles can not operate reliably without comprehensive perception systems for tasks such as spot detection.

Keat et al [5] attempted vehicle detection and car park mapping using only 2D laser range scans. In the task of

vehicle detection, they generate vehicle hypotheses from the 2D line segments by verifying the detection of a vehicle's two adjacent sides. Unfortunately, the environments may contain many objects such as corners of buildings which contain two sides suggestively similar to those of vehicles. These invalid cases can pass the validation of vehicle hypotheses test and mislead subsequent tests. Additionally, the experiments described were conducted in sparse parking spaces in which vehicles are not parked in a tightly packed pattern. Thus, the performance of vehicle detection was not evaluated for relatively packed environments, such as semi-filled parking spaces in busy downtown garages or shopping malls. Usually, due to occlusion, laser scanners could not obtain measurements of complete contours of vehicles containing two sides. Thus, the vehicle detection algorithm that validates two adjacent sides would fail. To address this issue, we believe that further classification criteria, such as geometric features, is necessary to achieve reliable detection.

In [8], Jung et al reviewed different parking assistance techniques and introduced a laser-based system. As their system is a driving assistance application, they assume that sufficient prior knowledge is available. For example, they assume that when the system is operating, the target spot and adjacent vehicles are in front of the sensor already, which is impractical in autonomous driving. Also, they discard cases of wide parking space between two adjacent vehicles, which eliminates potential situations where there are two vacant spots between the two adjacent vehicles. These priors are not applicable in autonomous driving scenarios, where certain intelligent behaviors are required and where the autonomous system needs to reliably and robustly interpret the environment.

Considering these issues, we propose an algorithm that can reliably detect vehicles from laser range scans of mostly single side of vehicles, in our case, bumpers. Specifically, we use AdaBoost [9] to detect vehicle bumpers. Arras et al [10] achieved impressive performance using pre-defined geometric features and an AdaBoost classifier to detect pedestrians from 2D laser range scans. In our application, we found that vehicle bumpers can be characterized through a set of geometric features that can be used for their detection.

Most directly related to our work in terms of inferring underlying structures is the research of Dolgov and Thrun [4], who described an algorithm addressing the detection of topological structures for autonomous driving in semi-structured environments such as parking lots. They focus on robust estimation of lane networks in parking lots, and address the navigation problem of path planning. In our work, we focus on the detection of available parking spots. Specifically, from a set of bumpers, our algorithm constructs the topological graph represented by a spatial distribution of the vehicle hypotheses, which reflects the spatial structure of the given parking lot. By interpreting this structure, our algorithm can identify available spots and maintain them in the graph representation using probabilistic methods.

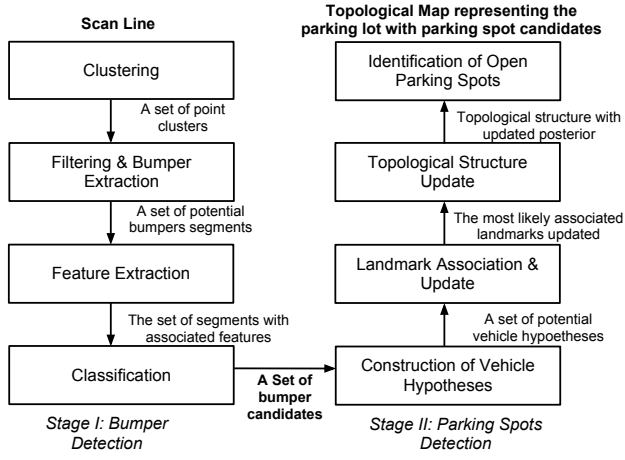


Fig. 2. Algorithm Overview and Process Flow

### III. ALGORITHM DESCRIPTION

There are two stages in the algorithm. Stage I performs detection of car bumpers. Stage II performs spatial analysis, probability update and parking spot identification. Fig. 2 shows an overview of the algorithm.

#### A. Stage I

In Stage I, the algorithm takes as inputs observations from laser range scans. Formally, each observation consists of consecutive bearings  $z = \{r_1, r_2, \dots, r_m\}$ , on which our algorithm performs bumper detection via clustering, filtering, bumper extraction, features extraction, and classification. The output is a set of segments and classification results. Each step is described in detail in the following sections.

1) *Clustering*: This step first converts ranges into points in Cartesian coordinates, and groups points that are possibly associated with the same objects using Dietmayer's clustering approach [11], [12], where the algorithm iterates through all the points of a scan in sequence, and separates the points whose distances between consecutive points are greater than a threshold. The output is an ordered sequence of clusters  $C = \{C_1, C_2, \dots, C_N\}$ , where  $\bigcup C_i = z$ , for  $i = 1, 2, \dots, N$ .

The distance threshold  $D_{thd}$  is:

$$D_{thd} = C_0 + C_1 \min(r_i, r_{i+1}),$$

where  $C_1 = \sqrt{2(1 - \cos \Delta \alpha)} = D(r_i, r_{i+1})/r_i$ , and  $C_0$  is a constant, in our case,  $C_0 = 30[cm]$ ;  $\Delta \alpha$  is the angle difference between the two beams;  $D(r_i, r_{i+1})$  is the Euclidean distance between two consecutive points that are returned by  $i$ th beam  $r_i$  and  $i + 1$ th beam  $r_{i+1}$ ,

$$D(r_i, r_{i+1}) = \sqrt{r_i^2 + r_{i+1}^2 - 2r_i r_{i+1} \cos \Delta \alpha}.$$

2) *Filtering & Bumper Extraction*: This step first performs occlusion reasoning and filtering of invalid measurements, after which a subset of original clusters  $C' \subseteq C$  is selected, where  $C' = \{C'_1, C'_2, \dots, C'_M\}$ . Among these clusters, there are instances of only vehicle bumpers, which do not need to be further segmented; there are also instances of L-shaped contours containing two sides of vehicles shown in Fig. 3, which need to be further segmented to extract bumpers. Formally, after occlusion reasoning and filtering, the algorithm takes in the set of clusters  $C'$ , segments all L-shaped clusters, and outputs the corresponding segments  $S = \{S_1, S_2, \dots, S_m\}$ , where for each L-shaped cluster  $C'_i$ ,  $C'_i = \{S_j, S_{j+1}\}$ . Specifically, there are 3 processes in this step:

a. *Occlusion reasoning*: In packed environments such as busy parking lots, occlusion is common and can separate a target's measurements into multiple clusters. Our algorithm identifies potential occlusion using the conditions described below, and joins those clusters that are believed to belong to the same targets. Specifically, the algorithm takes three consecutive clusters in each step, denoting  $C_1, C_2, C_3$ , and concludes  $C_2$  is occluding  $C_1$  and  $C_3$  if the three conditions are met: 1)  $C_2$ 's width is less than a threshold estimated from standard bumper width; 2)  $C_2$  is in front of  $C_1, C_3$  with jump distances greater than a threshold determined by clustering criteria; 3) by joining  $C_1$  and  $C_3$ , the result cluster displays continuity in shape.

b. *Filtering invalid and insufficient points*: Due to incompatibility between a target's albedo and the sensor's wavelength, a LADAR occasionally returns invalid measurements on car bumpers. The filtering module discards invalid range data, which are abnormally large values, and joins the adjacent clusters that are originally separated by these points and that are close enough to satisfy the clustering criterion.

c. *Bumper extraction*: The L-shaped segmentation technique is performed following the split and merge techniques [13], [14], [15]. In our algorithm, we only do split and merge once, obtaining two best-fit lines of the two resulting segments of the cluster. The cluster is determined to be L-shaped if the angle between the two lines is in the range  $[60^\circ, 120^\circ]$ . Fig. 3 shows an example in which bumpers are extracted from the segmentation.

One interesting observation is that front bumpers are generally rounder and lower than rear ones, especially for small cars. In L-shaped segmentation some front bumpers are occasionally divided into two segments. But the influence on overall performance is trivial because of evidence accumulation and Bayesian update.

3) *Feature Extraction*: For each segment  $S_i$ , this step computes features  $F_i = \{f_1, f_2, \dots, f_{11}\}$ . These eleven geometric features, with which car bumpers are well discriminated, will be used for classification by an AdaBoost classifier. Each feature is a function  $f : S_i \rightarrow \mathbb{R}$  that takes an input segment  $S_i \in S$  and outputs a real value. These features are:

*f1. Distance to Centroid*: The distance from the sensor to

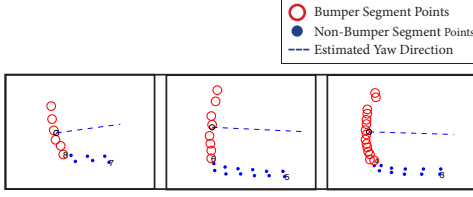


Fig. 3. Examples of bumper extraction; the three L-shaped clusters are segmented and potential bumpers (in red "o") are extracted.

the centroid of the segment. It is important as a normalizing term, since laser scans become sparser as distance increases.

*f2. Width:* The Euclidean distance between the first and the last point of the segment.

*f3. Convexity or non-Convexity:* This feature determines whether the segment is concave or convex. Most bumpers only appear convex from the sensor's perspective. It is determined by  $d_{circle} - d_{centroid}$ , where  $d_{centroid}$  is *f1*, and  $d_{circle}$  is the distance from sensor to the center of the segment's best fitting circle computed in *f9*. A convex segment corresponds to a positive value while a concave one corresponds to a negative value.

*f4. Mean Angular Difference:* This feature measures the degree of convexity/concavity of the segment. Different from *f3*, which decides whether a segment is convex or concave, this feature computes the degree of convexity/concavity: It is determined by traversing the points of the segment  $S_i = \{x_1, x_2, \dots, x_n\}$ , taking three consecutive points  $(x_{i-1}, x_i, x_{i+1})$  each time, and computes the average of the angles  $\beta_i$  between the vectors  $x_{i-1} \vec{x}_i$  and  $x_i \vec{x}_{i+1}$ :

$$\beta_i = \angle(x_{i-1} \vec{x}_i, x_i \vec{x}_{i+1}).$$

*f5. Density:* The density of a segment is measured by the standard deviation of points in the segment to its centroid.

*f6. Linearity*<sup>1</sup>: This feature, similar to the linear regression technique, measures the squared residual sum of the segment to its best fitting line in the least squares sense. It outputs the squared residual sum of all points in the segment, which indicates its degree of straightness.

*f7. Boundary Length:* As an alternative to width, this feature measures the boundary length of the segment's poly-line representation:

$$l = \sum_{j=1}^{n-1} |x_{j+1} - x_j|$$

where  $n$  is the number of points *f8*, and  $x_j$  is a point of Cartesian representation in the current segment  $S_i = \{x_1, x_2, \dots, x_n\}$ .

*f8. Number of Points:* The number of points in the segment  $S_i$ ,  $n = |S_i|$ .

*f9. Circularity*<sup>1</sup>: This feature measures the circularity of the segment. Similar to linearity, the circularity is the squared residual sum of the segment to its best fitting circle in the least squares sense.

*f10. Boundary Regularity*<sup>1</sup>: This feature is the standard deviation of all poly-line lengths of the segment, each poly-line length is  $l_{i,i+1} = |x_{j+1} - x_j|$  as computed in *f7*.

*f11. Mean Curvature*<sup>1</sup>: As an alternative reference to *f4*, this feature measures the mean curvature of the segment.

The order in which these features were presented above reflects their relevancy for classification in decreasing order. We notice that distance to the segment's centroid, width, convexity or non-convexity, mean angular difference (degree of convexity), and density are the most relevant for classification.

*4) Classification:* This step classifies each segment using the AdaBoost technique. With the eleven features extracted for each segment, the algorithm then feeds every segment's feature vector into the trained AdaBoost classifier. The classifier makes decisions on behalf of each segment  $S_i$  either belonging to bumper class or non bumper class. According to the weighted votes of the weak classifiers, the AdaBoost outputs the more likely class label  $l_i$ . In our case, each weak classifier is a binary threshold classifier.

Formally, the output is the set  $E$  of all segments, each with its classification score  $w_i \in \mathbb{R}$  and labels  $l_i = \text{sign}(w_i)$ :

$$E = \{(S_i, w_i, l_i)\}, i = 1, 2, \dots, |S|,$$

where  $l_i = +1$  indicates segment  $S_i$  is a bumper, and  $l_i = -1$  indicates  $S_i$  is not a bumper.

Note that for two segments  $S_j$  and  $S_{j+1}$  that were originally extracted from an L-shaped cluster  $C'_i = \{S_j, S_{j+1}\}$ , only one of them can be a bumper while the other being the side of the vehicle. At this stage, the AdaBoost classifier is used to classify each segment independently based on geometric features. In the next stage, additional filtering will be performed considering spatial relations without the assumption of independence.

## B. Stage II

In Stage II, the output of Stage I,  $E$ , is taken as input to initialize or update the posterior probability of bumper landmarks. These landmarks are used to construct a topological graph defined in section B3. Spatial analysis and parking pattern grouping are performed on the graph, upon which potential parking spots can be identified. These spots are then turned to landmarks whose posterior probabilities are being updated with subsequent observations. There are four steps in this process, which are described below.

*1) Construction of Vehicle Hypothesis:* For each segment  $S_i$  classified as bumper, we construct a vehicle hypothesis as the configuration

$$h_i = (l_i, \psi_i, p_i),$$

where  $l_i \in \mathbb{R}^2$  is the parked vehicle's location in Cartesian representation of 2D world frame,  $\psi_i \in \mathbb{R}^2$  is the yaw direction vector of the hypothesized vehicle, and  $p_i$  is the posterior probability of the hypothesis being a valid one, which will be updated as described in subsequent section.

<sup>1</sup>The implementation details can be found in [10].

The yaw direction  $\psi_i$  is determined by taking the normal vector of the segment's best fitting line. Given location and yaw, the algorithm generates a bounding box for each vehicle hypothesis. The width of the bounding box is determined by the bumper width (feature  $f_2$ ), and the length is approximated using the length-width ratio of standard vehicles. Fig. 4, Fig. 5 show examples of vehicle hypotheses overlaid with bounding boxes.

Since the previous classification is performed on each segment based on geometric features, there exist false-positives that result from other objects which are geometrically similar to a bumper from laser's perspective. These false positives can be filtered if additional information is taken into consideration:

*a. Filter by width:* The width (feature  $f_2$ ) of a potential bumper segment  $S_i$  is greater than a normal bumper could be, or  $S_i$ 's adjacent segment, which is potentially the side of a vehicle, is wider than a normal car side could be.

*b. Filter by obstacles in bounding boxes:* Some segment  $S'$  is believed to be not a part of a vehicle, but lies inside the bounding box of some vehicle hypothesis, the segment  $S'$  is potentially an obstacle. In this case it is unlikely that a vehicle can fit in. In either case, the probability of the associated hypothesis  $p_i$  is decreased.

Note that the filtering needs to be performed after the classification in Stage I because it is used to verify or disprove the classification results and the validity of bumper candidates generated from Stage I. In Stage I the classifier's decision is made based solely on eleven geometric features, and it assumes that each segment is independent from each other. By filtering with additional criteria and eliminating the assumption of segments' independence, the algorithm forms a stronger and more accurate belief than classification based on geometric features alone, which facilitates subsequent spots identification.

*2) Landmark Association & Update:* After filtering, data association from these bumpers to landmarks is performed. Specifically, for each bumper, the algorithm takes its current observation  $z^t$  at time  $t$ , which is the new scan ranges  $z^t = \{r_1^t, r_2^t, \dots, r_m^t\}$ . Then the algorithm performs association by finding the landmark  $h_i$  that maximizes the probability of it being an observation of the landmark  $z_i^t$ :

$$h_{i^*}^{t-1} = \arg \max_{h_{i^*}^{t-1}} p(z_i^t | h_{i^*}^{t-1}, z^t, s^t),$$

where  $s^t$  is the current state (location and orientation) of our vehicle. The association is performed using Nearest Neighbor (NN) approach.

Once the corresponding landmarks are found, the algorithm updates the posterior probability of these landmarks being valid bumpers based on the Bayesian framework for evidence accumulation [16]:

$$P(h_{i^*}^t | h_{i^*}^{t-1}, z^t, s^t).$$

In implementation, the classification output in Stage I for each segment,  $E$ , is taken to update the posterior of the

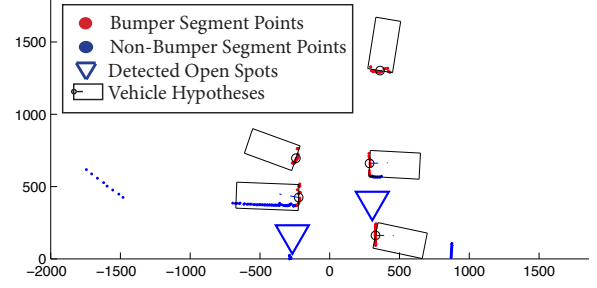


Fig. 4. An example of detected parking spots and parked vehicles in sensor frame. The red segments are determined by our algorithm to be bumpers; vehicle hypotheses are overlaid with bounding boxes; blue triangles are detected open parking spots.

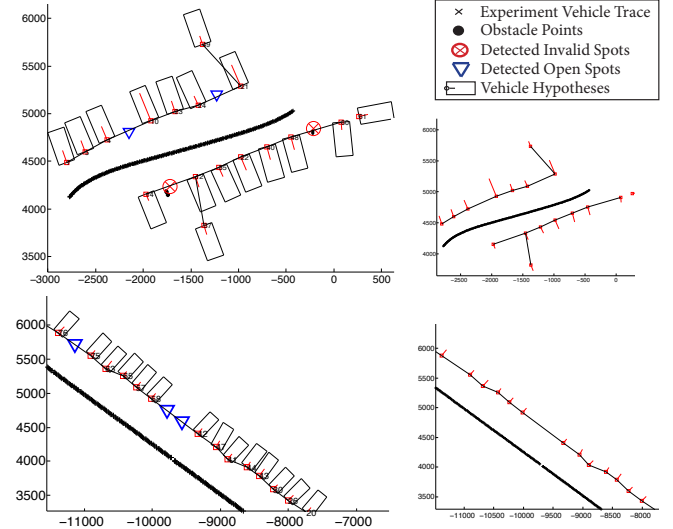


Fig. 5. An example of the topological structure. **Top-Left:** A semi-filled parking lot; parked vehicles in each parking group are connected by black lines (edges of Minimum Spanning Tree, formally defined in section B.3). Note that two potential spots in the bottom parking group are rejected because obstacles are detected in the potential spots' areas. **Top-Right:** Corresponding topological graph; red squares are nodes. **Bottom-Left:** A semi-filled uniformed-layout parking lot; note that two adjacent parking spots are identified because the space detected is wide enough to fit two vehicles according to the parking pattern of the parking group. **Bottom-Right:** Corresponding topological graph; red squares are nodes.

landmark in log-odds form. If no close enough landmarks are found matching some bumper candidate, a new landmark is initialized for it.

*3) Update of Topological Structure:* Topological graphs are regarded as efficient representations of the world that omit the irrelevant details in the environment and concern only the most relevant factors [14], [17]. For the purpose of identifying open spots in a parking space, our topological representation of the parking lots concerns the poses of parked vehicles and their spatial relationships. Thus, in our graph, we represent parked vehicles as nodes. Two nodes are connected with an edge if the two parked vehicles (nodes) are close to each other, and display similar parking patterns. Each node is encoded with the corresponding vehicle's unique landmark ID and its configuration  $h_i$ . Fig. 5 shows



examples of resulting topological graphs. The following three procedures are performed to construct the topological graph:

*a. Initializing Nodes:* A landmark  $k$  is initialized as node in the topological structure if  $p_k$  is greater than a threshold determined by the frequency of probability update, indicating the algorithm is confident about the landmark as a valid vehicle bumper.

*b. Grouping:* In real parking lots, vehicles are usually parked according to the layout of parking spots, which display uniform parking patterns in terms of aligned positions and directions. Different sets of parking spots display different parking patterns. Thus, to identify valid parking spots among these parking sets, a grouping procedure is necessary to encapsulate nodes that have similar parking patterns. The grouping is performed such that 1) nodes of each group are within a distance threshold and 2) the differences of landmarks yaw directions in each group are less than a threshold. The thresholds are estimated from standard layout in parking spaces. This ensures that nodes of each group present similar parking patterns.

*c. Connecting Nodes of Same Group:* In order to detect parking spots in each parking set, we need to extract the spatial relationship of the nodes in the parking set. The spatial relationship is simply represented as edges in the topological graphs. For each group, the algorithm constructs the Minimum Spanning Tree (MST) that connects the pairs of landmark nodes with least distance. The weight of each MST edge is simply the distance between the two end nodes. The MST representation ensures that two nodes connected by an edge are most likely the adjacent vehicles in their parking set (Fig. 5). This allows us to identify potential open spots in each parking set as described in the following section.

*4) Identification of Parking Spots:* A potentially open parking spot is identified in between two adjacent vehicle landmarks of the same parking set, provided that the distance between the two landmarks is greater than a threshold (3 meters in our case), estimated from standard parking spot spaces. Fig. 5 shows examples of spot landmarks between vehicle landmarks in the topological graph.

Note that given 2D range scans, we only initialize a parking spot hypothesis if it is located *between* two bumper landmarks of the same parking set. This imposes the requirement that a valid node of spot landmark must depend on two adjacent vehicles as references.

One may notice that this rules out many potential spots that are adjacent to only one vehicle landmark and spots that are standalone, commonly found in sparse parking lots. The consideration is that only one vehicle landmark can hardly provide sufficient and reliable reference to validate the spots, given only 2D range data. One can easily come up with a counter example where a spot is identified adjacent to one vehicle, while the vehicle is next to road and the detected spot stands in the middle of the road. In this case, a car that parks at this "spot" becomes an obstacle on the road. On the other hand, since we are interested in relatively



Fig. 6. Experimental Vehicle - NavLab 11

packed environments such as semi-filled parking spaces in city center or malls, one can expect a number of parked vehicles to provide sufficient references for spot detection. Thus, given the practical concerns, we require that one or a few consecutive spots must have two adjacent vehicles as references.

Once a potential spot is identified, it will be initialized with location, computed as the mean of locations of two referred vehicle nodes between which the spot locates. Similar to bumper landmarks, data association using Nearest Neighbor approach on the spot hypothesis will be performed to associate it with an existing spot landmark, or the spot hypothesis will be initialize as a new spot landmark if no close existing landmarks are found.

## IV. EXPERIMENTS

Our experimental vehicle is a Jeep Wrangler named Navlab 11 as shown in Fig. 6. The LADAR we used to collect data is a SICK LMS-291 laser range finder, mounted in front of the vehicle 60 cm above the ground. Data rate was 37.5Hz, 180° field of view, with 0.5° resolution.

### A. Training the Bumper Classifier

We trained our classifier for bumper detection with data collected in a semi-filled parking lot, where most of the spots are occupied with cars tightly close to each other, and several spots are open. Specifically, the points from a total of 731 scans were clustered into 5234 segments, among which 2825 segments are manually labeled as positive samples (bumpers), and 2409 segments negative samples (non bumpers). The training process uses multiple iterations to generate a single strong classifier that is an ensemble of weak classifiers. During each iteration of training, a new weak classifier is added with a weight adjusted to focus on examples that were misclassified in previous iteration. The complete set of labeled segments was divided into a training set and a test set, and 8-fold cross-validation was used.

### B. Experiments

To evaluate bumper detection, we used a data set collected in the semi-filled parking space as shown in Fig. 1(above). The data set is randomly divided into 8 folds, and we applied 8-fold cross validation. In each validation, 7 folds are used

TABLE I  
CONFUSION MATRIX FOR BUMPER & SPOT DETECTION

	Detected bumper	Detected non-bumper	Total
True bumper	2472 (87.50%)	353 (12.50%)	2825
True non-bumper	507 (21.05%)	1902 (78.95%)	2409

	Detected spot	Detected invalid spot	
True spot	34 (94.44%)	2 (5.54%)	36
True invalid spot	3 (12.50%)	21 (87.50%)	24

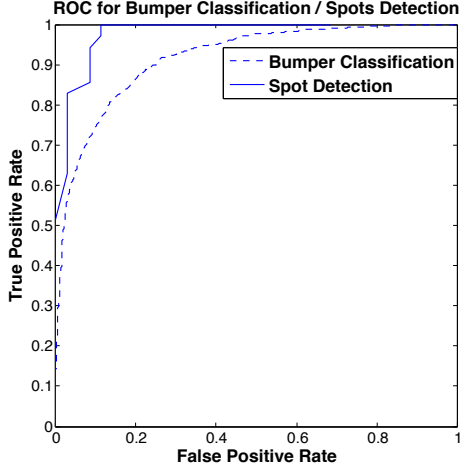


Fig. 7. ROC plot for bumper classification / spot detection

to train the classifier while the remaining fold is used for testing, and the average performance of the eight validations is used to evaluate the classifier. The confusion matrix and ROC plots for bumper detection are shown in Table I and Fig. 7. From the table we can see our algorithm obtained 87.50% bumper detection rate at the optimal operation point, where the highest iso-accuracy line under the class distribution intersects with the convex hull, resulting in the good separation of two classes with balanced tradeoff between detection rate and false positive rate. There were failures during our experiment, in which some vehicle bumpers were far from our laser, and as our experimental vehicle drove away, the laser did not obtain adequate measurements for evidence accumulation, resulting in low confidence and detection failure.

To evaluate spot detection, we used four data sets collected in different settings. Set I and II are collected in typical semi-filled parking lots in which most spots are occupied and several spots are available, as shown in Fig. 1; Set III is collected in a relatively sparse parking space where several spots are occupied while most are available; Set IV is collected in a less structured parking space, in which vehicles are parked in a less uniform pattern than in a standard parking lot in terms of pose alignment. In the test, we dialed the bumper detector to operate at the optimal operation point of 87.5% detection rate. Based on the bumper classification results gained in Stage I, the algorithm constructed the topological structure of the parking lot, and maintained landmarks of both vehicle and potential parking spots.

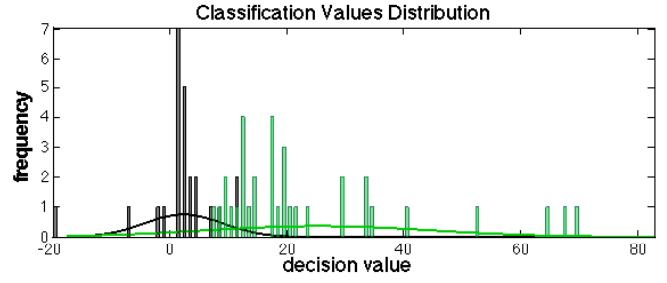


Fig. 8. Histogram for Parking Spots Detection Value Distribution; green line and black line are the Gaussian distributions of the positive samples (valid spots) and negative samples (invalid spots) respectively; green bars and black bars are the frequencies of the scores given by the algorithm on valid spots and invalid spots respectively.

The spot detection results are shown in Table I and Fig. 7. Of the 60 potential spots manually labeled as ground truth, 36 are valid and 24 are invalid. Note that when evaluating the spot detector, given the constraints of 2D range data, we count all the potential spots that our system should be able to detect. Specifically, a valid parking spot is an open space in between two reference vehicles that matches the parking pattern and is wider than 3 meters; an invalid spot is a wide enough space in between two reference vehicles that follows the parking pattern, but contains obstacles such as pedestrians, plants, pillars, electric poles, street lights, etc., shown in Fig. 5.

Our algorithm can operate at a point resulting in the performance shown in Table I. The distribution of the confidence scores for these 60 potential spots is displayed in Fig. 8 as a histogram. The ROC plot is shown in Fig. 7. As suggested by the decision value distribution, the detection performance is encouraging. We found that most failures were due to inadequate measurements for evidence accumulation as our experimental vehicle drove by. In particular, the false positive spots were invalid spots defined in previous paragraph, containing occluded obstacles, mostly pedestrians; these obstacles were hardly detected as our experimental vehicle drove by, resulting in inadequate measurements. Note that when operating in reality, false positive detections could lead to accidents. Thus, an applicable operation point for our autonomous system should be imposed at the false positive rate of 0%, where the corresponding true positive detection rate of 52% is obtained. This means that the system ensures that one out of every two empty spots may not be detected, but it also guarantees that the test vehicle will not collide with another car because of a false detection.

By comparison, Jung et al [8] proposed a system with spot detection rate of 98%. However, as mentioned in Section II, since their system is for the purpose of driving assistance, during their experiments the sensor was placed in front of parking spots prior to detection. While our system is for the purpose of fully autonomous parking, our autonomous vehicle detects the spots when driving around parking space without any prior knowledge of the environments, it instead relies on inference of the environments and interpretation of the underlying structures. Keat et al [5] presented the

car park mapping system, but they did not show evaluation results quantitatively.

## V. CONCLUSION

The paper proposes an approach to detect vehicles and parking spots in relatively packed environment with only one on-board 2D laser range finder. It combines both detection of targets and inference of the environments. With the use of an AdaBoost classifier and topological graph representation, our algorithm is capable of detecting vehicle bumpers and parking spots in semi-filled parking lots under occlusion and sensor constraints.

Because of concern for robustness, in our experiments, we tested settings such as sparse and less structured parking spaces, in which our algorithm is still able to robustly detect parking spots as long as enough reference is available to infer the spacial layout. In our approach, we rule out potential parking spots that have few reference vehicles, because perception with only 2D range data requires us to impose dependence on these adjacent vehicles as references. Also, since our algorithm detects vehicles through the detection of bumpers, it works in both row and diagonal parking structures described in [18] where cars are parked side-by-side and bumpers are visible; while its performance is limited in lane parking structures where cars are parked head-to-tail alongside the road and bumpers are only partially visible.

## VI. ACKNOWLEDGEMENTS

This work is based on research supported by the National Science Foundation under Grant No. EEE-0540865.

## REFERENCES

- [1] *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic* (Springer Tracts in Advanced Robotics), 1st ed. Springer, Nov. 2009. [Online]. Available: <http://www.worldcat.org/isbn/3642039901>
- [2] A. Suppé, L. E. Navarro-Serment, and A. Steinfeld, "Semi-autonomous virtual valet parking," in *Proceedings of the 2nd International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, ser. AutomotiveUI '10. New York, NY, USA: ACM, 2010, pp. 139–145. [Online]. Available: <http://doi.acm.org/10.1145/1969773.1969798>
- [3] M. Buehler, K. Iagnemma, and S. Singh, *The 2005 DARPA Grand Challenge: The Great Robot Race*, 1st ed. Springer Publishing Company, Incorporated, 2007.
- [4] D. Dolgov and S. Thrun, "Autonomous driving in semi-structured environments: Mapping and planning," in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, may 2009, pp. 3407–3414.
- [5] C. Keat, C. Pradalier, and C. Laugier, "Vehicle detection and car park mapping using laser scanner," in *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, aug. 2005, pp. 2054–2060.
- [6] M. Schmid, S. Ates, J. Dickmann, F. von Hundelshausen, and H.-J. Wuensche, "Parking space detection with hierarchical dynamic occupancy grids," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*, june 2011, pp. 254–259.
- [7] W.-J. Park, B.-S. Kim, D.-E. Seo, D.-S. Kim, and K.-H. Lee, "Parking space detection using ultrasonic sensor in parking assistance system," in *Intelligent Vehicles Symposium, 2008 IEEE*, june 2008, pp. 1039–1044.
- [8] H. G. Jung, Y. H. Cho, P. J. Yoon, and J. Kim, "Scanning laser radar-based target position designation for parking aid system," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 9, no. 3, pp. 406–424, sept. 2008.
- [9] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," 1999.
- [10] K. Arras, O. Mozos, and W. Burgard, "Using boosted features for the detection of people in 2d range data," in *Robotics and Automation, 2007 IEEE International Conference on*, april 2007, pp. 3402–3407.
- [11] K. C. J. Dietmayer, J. Sparbert, and D. Streller, *Model based object classification and object tracking in traffic scenes from range images*, 2000, pp. 1–6.
- [12] C. Premebida and U. Nunes, "Segmentation and geometric primitives extraction from 2d laser range data for mobile robot applications," *Proceedings of Robotic*, pp. 17–25, 2005.
- [13] K. J. Lee, "Reactive navigation for an outdoor autonomous vehicle," 2001.
- [14] R. Chatila, "Representation issues," Stockholm, 2002.
- [15] J. D. Tardos, "Data association in slam," KTH, Stockholm, Tech. Rep., 2002.
- [16] K. Murphy, "Bayesian map learning in dynamic environments," in *In Neural Info. Proc. Systems (NIPS)*. MIT Press, pp. 1015–1021.
- [17] A. Cohn, "Qualitative spatial representation and reasoning techniques," in *KI-97: Advances in Artificial Intelligence*, ser. Lecture Notes in Computer Science, G. Brewka, C. Habel, and B. Nebel, Eds. Springer Berlin / Heidelberg, 1997, vol. 1303, pp. 1–30, 10.1007/3540634932.
- [18] I. E. Paromtchik and C. Laugier, "Motion generation and control for parking an autonomous vehicle," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 1996, pp. 3117–3122.