Vehicle detection and car park mapping using laser scanner — **Source link** ↗

Christopher Tay Meng Keat, Cédric Pradalier, Christian Laugier

Institutions: French Institute for Research in Computer Science and Automation

Published on: 05 Dec 2005 - Intelligent Robots and Systems

Related papers:

- On-road vehicle detection: a review

- Real Time Suited Road Border Recognition Utilizing a Neural Network Technique

- Fast vehicle detection with probabilistic feature grouping and its application to vehicle tracking

- Vehicle pose estimation for vehicle detection and tracking based on road direction

- Monocular vision-based range estimation of on-road vehicles

Vehicle Detection And Car Park Mapping Using Laser Scanner

Christopher Tay, Cédric Pradalier, Christian Laugier

Vehicle Detection And Car Park Mapping Using Laser Scanner*

Christopher Tay Meng Keat, Cédric Pradalier and Christian Laugier

INRIA-GRAVIR-INP Grenoble, France

{*tay, cedric.pradalier, christian.laugier*}*@inrialpes.fr*

Abstract— **In this project, we took on the task of localizing an automatic vehicle and building a map of the car park in real time. This takes place within the car park of INRIA Rhône-Alpes on the CyCab vehicle with a Sick laser range scanner. Our method uses only laser scanners to retrieve the position and orientations of vehicles in the car park. With the detected vehicles as landmarks, CyCab performs a localization of itself and builds a map of the car park at the same time. Classical clustering and segmentation techniques to extract line segments from the laser scan data is applied. The key contribution of the paper is the extraction of vehicle poses from the line segments using bayesian programming. The method of FastSLAM is used in localizing CyCab and estimating the pose of vehicles in the car park. A set of hypotheses is obtained as a result. The second contribution is a method of combining the set of hypotheses together to form a final map of the car park.**

Index Terms— **SLAM, Vehicle Detection, Bayesian Programming**

I. INTRODUCTION

We describe in this paper our first step towards mapping of the car park with the autonomous vehicle, CyCab. The map of the car park will contain the positions and orientations of the vehicles that exists in the car park. Such a map can serve as a reference which indicates obstacle pose configurations. Furthermore, it can indicate the state of the parking lots in the car park, and will be used in more high level applications like automatic parking in the future.

There were different methods of interpreting the laser scan data. The two main interpretations of scan data are to perform SLAM by looking for certain specific features or to perform alignment of the scan data and our work falls into the category of the former. The main advantage of an object map is its compact representation, and it is an important step towards a higher level environmental scene understanding for more advanced robotic applications.

Several object based representation of the environment were proposed. Chatila et al. [2] represented the map with a set of lines. More advanced methods in mapping consists of approximating the environment using small polygons [5] [6]. Such methods used a variant of the Expectation-Maximization to generate increasingly accurate 3D maps as more observations are made. In our project, we used a high level representation (vehicles in this case) of the environment instead of fundamental geometrical entities. The main difference in this

paper is the inference of object configurations with a bayesian approach.

The general idea is using only the laser data without any artificial or predefined landmarks, CyCab will navigate the car park autonomously while generating a map of its environment (fig. 1). At each iteration of the mapping process, CyCab estimates its own position and orientation of the form (x, y, θ) and creates a map of the car park in the world frame of reference. The origin of the world frame of reference is taken to be the initial position of CyCab. The map is represented by a set of tuples, each containing the position and orientation of the vehicles detected. CyCab hypothesizes the configuration of the vehicles in the surrounding based on the laser scan inputs only.
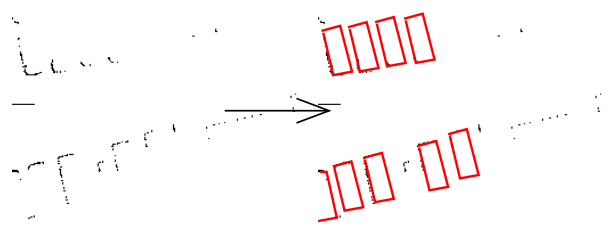

Fig. 1. Obtaining the configuration of vehicles based on data from scan data

The advantage of our approach is the ability to map any car park without installing any external aids. With the set of vehicle poses representing the map, we are also able to obtain a compact and semantic representation of the map.

The current state of the project works under the assumption that mapping is performed only in static car parks where vehicles are not moving. It is also assumed that vehicles in the car park belongs to the same vehicle class.

II. OVERVIEW

The mechanism of the entire system can be broken down into three fundamental portions, *vehicle detection*, the *simultaneous localization and mapping (slam)* and the *map construction* . CyCab is provided with two kinds of raw data, the laser scans and CyCab's odometric data. Figure 2 shows the block diagram of the different stages and its interactions:

1) **Vehicle Detection**: With only raw laser scan data, vehicle detection constructs hypotheses about the positions and orientation of vehicles in the car park.
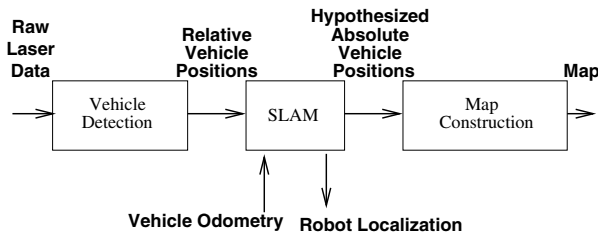
Fig. 2. Overview showing the mapping process

2) **SLAM**: Coupled with information about odometry of CyCab, SLAM makes use of the constructed vehicle hypotheses to estimate its own configuration. With CyCab's configuration, the configurations of vehicle hypotheses with respect to real world coordinates can be obtained.

3) **Map Construction**: Because the method of FastSLAM produces a set of hypotheses, a map construction module is required to merge the information from the different hypotheses to obtain the final map.

III. VEHICLE DETECTION

The qualification of a vehicle hypothesis is by verifying the detection of the two adjacent sides of a vehicle. This assumption is to increase the reliability of hypotheses as there are many possible objects in the environment and we seek geometry that fits the geometric description of a vehicle in a rather conservative manner. To obtain line segments indicating possible sides of the vehicles, a two stage filtering process is introduced. In the first stage, we have used classical methods in the extraction of vehicle hypotheses.

1) **Clustering** is performed to group the points close together, which probably belongs to a single object.

2) **Segmentation** is performed on the groups of points to obtain the contour of the object represented by line segments.

3) **Vehicle hypotheses** are obtained using bayesian programming, which gives the possibility of the existence of the vehicles at various configurations. The construction of hypotheses by bayesian programming results in a mechanism similar to that of hough transform. Peak values in the histogram indicates the most probable vehicle poses. Although there exists methods of recognition based on geometric constraints in the vision literature, we decided to explore the methods of bayesian programming in this context to take into account uncertaintites in the measurements to obtain a set of plausible vehicle hypotheses.

However, with real data, the previous stage produces too many vehicle hypotheses which are false positives. A second stage of filtering is applied to each vehicle hypothesis obtained after the first stage as a form of validation gate in order to reduce the number of false positives. It is broken down into two sub stages:

1) **Edge Filtering** is applied to extract the set of line segments that is only relevant to the vehicle hypothesis in question.

2) **Vehicle Support Filtering** is based on a metric which we call vehicle support. It measures how much of the two adjacent sides of a vehicle are detected. The metric is calculated using the set of edges from Edge Filtering. With the metric, we try to remove as many false positives as possible by ensuring that we are able to verify a substantial portion of the two adjacent sides of a vehicle.

A. Clustering and Segmentation

The aim of clustering is to group the impact points such that the clusters represents the same object. Clustering is performed by going through in sequence the impact points and separate the points into clusters using a simple distance criterion.

Segmentation is performed on each cluster. The segmentation proposed is commonly found in the literature [4] [1] [11]. Such techniques takes a group of points and try to fit a contiguous segment that will approimately fit the set of data points. Split and merge is used to obtain segments from the set of impact points by recursively splitting the group of points into subgroups and performing line fitting before merging the sub solutions.

B. Construction of vehicle hypotheses

A 3D histogram is constructed to represent the distribution of vehicle poses (x, y, θ). The histogram is constructed using bayesian programming [3]. The bayesian program results in a mechanism similar to that of the hough transform.
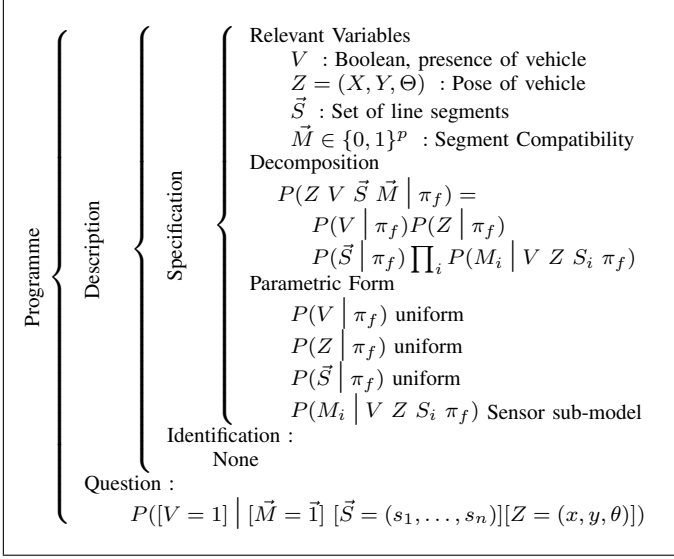
Briefly, a bayesian program is composed of:

- defining the list of *relevant variables*;
- a *decomposition* of the joint distribution by assuming conditional independence;
- the *parametrical form* of each factor of the decomposition;
- the *identification* of the parameters of these parametrical forms;
- a *question* in the form of probability distribution inferred from the joint distribution.

In the construction of the histogram, we would like to treat each line segment independently. In doing so, it will be sufficient to simply go through the list of segments and add necessary information into the histogram for each line segment. This is achieved by performing data fusion with diagnostic [10]. Finding the vehicle pose hypotheses is represented by the bayesian program in fig. 3 and fig. 4. In this paper, the following variables are adopted:

- V: A boolean value indicating the presence of a vehicle
- $Z = (x, y, \theta)$: The pose of a vehicle
- $\vec{S}$: Ensemble of extracted line segments
- $\vec{M} \in \{0, 1\}^p$: Compatibility of segments with vehicle pose

- $\vec{C} \in \{1,2\}^p$: A value of 1 or 2 if segment corresponds to the width and length of a vehicle respectively
- π: A priori knowledge

Programme { Description { Specification {

Relevant Variables
V : Boolean, presence of vehicle
$Z = (X, Y, \Theta)$: Pose of vehicle
$\vec{S}$: Set of line segments
$\vec{M} \in \{0,1\}^p$: Segment Compatibility
Decomposition
$$P(Z\ V\ \vec{S}\ \vec{M} \mid \pi_f) =$$
$$P(V \mid \pi_f)P(Z \mid \pi_f)$$
$$P(\vec{S} \mid \pi_f)\prod_i P(M_i \mid V\ Z\ S_i\ \pi_f)$$
Parametric Form
$P(V \mid \pi_f)$ uniform
$P(Z \mid \pi_f)$ uniform
$P(\vec{S} \mid \pi_f)$ uniform
$P(M_i \mid V\ Z\ S_i\ \pi_f)$ Sensor sub-model
Identification :
None
Question :
$$P([V = 1] \mid [\vec{M} = \vec{1}]\ [\vec{S} = (s_1,\ldots,s_n)][Z = (x,y,\theta)])$$

Fig. 3: Detection of vehicles

The question in fig. 3 represents the probability of a vehicle given the vehicle pose, the set of extracted line segments and that the line segments are compatible. The histogram contains values calculated by this question. The question from the bayesian program (fig. 3) can be simplified further using baye's rule:
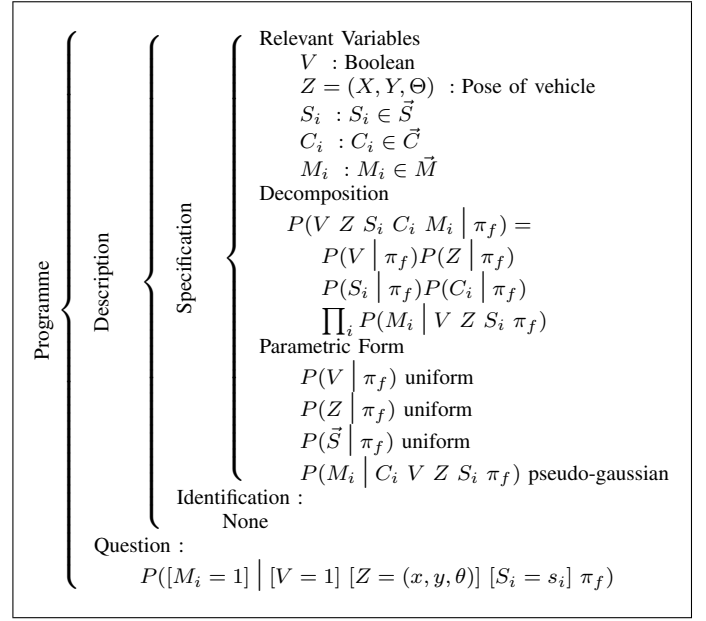
$$P([V = 1] \mid [\vec{M} = \vec{1}]\ [\vec{S}]\ Z\ \pi_f)$$
$$= \frac{P([V = 1]\ [\vec{M} = \vec{1}]\ [\vec{S}]\ Z \mid \pi_f)}{P([\vec{M} = \vec{1}]\ [\vec{S}]\ Z \mid \pi_f)}$$
$$= K \prod_i P([M_i = 1] \mid [V = 1]\ Z\ S_i\ \pi_f)$$
with K constant

The question can be decomposed as the product of the probability of the compatibility of line segments given its vehicle pose. Each factor of this product is calculated by a sub model as shown in fig. 4.

The sub-model for the calculation of the question $P(M_i \mid V\ Z\ S_i\ \pi_f)$ is described in fig. 4. This sub-model question can be further resolved:

$$P([M_i = 1] \mid [V = 1]ZS_i\pi_f)$$
$$= \frac{P([V = 1]ZS_i[M_i = 1] \mid \pi_f)}{P([V = 1]ZS_i\pi_f)}$$
$$= \frac{\sum_{C_i} P([V = 1]ZS_iC_i[M_i = 1] \mid \pi_f)}{P([V = 1]ZS_i\pi_f)}$$
$$= \frac{\sum_{C_i} \begin{array}{c} P([V = 1]\mid\pi_f)P(Z\mid\pi_f)P(S_i\mid\pi_f) \\ P(C_i\mid\pi_f)P([M_i = 1]\mid C_i[V = 1]ZS_i\pi_f) \end{array}}{P([V = 1]ZS_i\pi_f)}$$
$$= \sum_{C_i} P(C_i\mid\pi_f)P([M_i = 1]\mid C_i[V = 1]ZS_i\pi_f)$$

Programme { Description { Specification {

Relevant Variables
V : Boolean
$Z = (X, Y, \Theta)$: Pose of vehicle
S_i : $S_i \in \vec{S}$
C_i : $C_i \in \vec{C}$
M_i : $M_i \in \vec{M}$
Decomposition
$$P(V\ Z\ S_i\ C_i\ M_i \mid \pi_f) =$$
$$P(V \mid \pi_f)P(Z \mid \pi_f)$$
$$P(S_i \mid \pi_f)P(C_i \mid \pi_f)$$
$$\prod_i P(M_i \mid V\ Z\ S_i\ \pi_f)$$
Parametric Form
$P(V \mid \pi_f)$ uniform
$P(Z \mid \pi_f)$ uniform
$P(\vec{S} \mid \pi_f)$ uniform
$P(M_i \mid C_i\ V\ Z\ S_i\ \pi_f)$ pseudo-gaussian
Identification :
None
Question :
$$P([M_i = 1] \mid [V = 1]\ [Z = (x,y,\theta)]\ [S_i = s_i]\ \pi_f)$$

Fig. 4: Sensor Sub-Model

From the decomposition of the sub-model question, the remaining problems lies in expressing $P(C_i \mid \pi_f)$, the probability that a segment corresponds to the length L or width l of a vehicle. The expression $P([M_i = 1] \mid C_i\ [V = 1]\ [Z = (x,y,\theta)]\ [S_i = s_i]\ \pi_f)$, the probability that a segment and its given association to the sides of the vehicles, corresponds to a vehicle at pose (x,y,θ).

For $P(C_i \mid \pi_f)$, a simple model is given by:
- $P([C_i = 1]|\pi_f) = \frac{L}{l+L}$
- $P([C_i = 2]|\pi_f) = \frac{l}{l+L}$

$P([M_i = 1] \mid C_i\ [V = 1]\ [Z = (x,y,\theta)]\ [S_i = s_i]\ \pi_f)$ is expressed by a pseudo-gaussian function $G(S_c, s_i)$. Where S_c represents the segment of a side of the hypothesized vehicle position (x,y,θ). The more s_i is further or oriented differently with respect to S_c, the smaller the value of $G(S_c, s_i)$. In practice, as most of the values in the histogram are negligible the histogram is filtered by a treshold, it is sufficient to go through the list of segments and filling in the histogram values only for the possible vehicle poses (x,y,θ) that are compatible with the line segments. The vehicle poses can be easily calculated with simple geometry and this speeds up the construction of the histogram.

C. Edge Filtering

To calculate how well the length or width of a vehicle conform to a vehicle hypothesis configuration, for each hypothesized vehicle, the set of segments relevant to the hypothesized vehicle is required. The objective of edge filtering is to provide data for the calculation of vehicle support (section III-D). In this paper, the term edges and segments are used interchangably.

Edges of interests are the edges which lies around the contours of the vehicle. The edges bounded by an inner

and outer bounding box are to be selected. The bounding boxes are oriented and positioned in the same manner as the hypothesized vehicle. (Fig. 5).
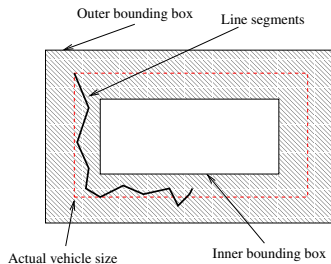

Fig. 5: Edge filtering with 2 bounding boxes. Shaded area indicates valid area

The algorithm begins with the segment that contains the endpoint nearest to the origin (where Sick is). Starting from this segment, the algorithm starts to grow outward by searching for any segments where any of its endpoints lies sufficiently close to one of the endpoints in the original segment (example in figure 6). This continues until a sufficiently close segment cannot be found.
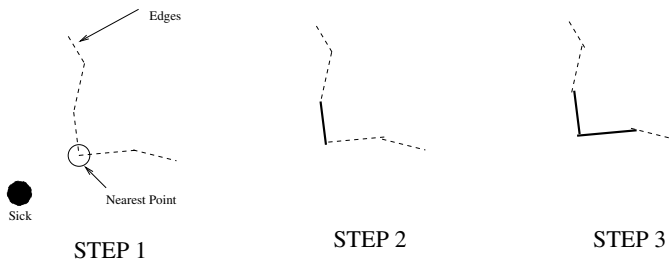

Fig. 6: An example showing the spreading of the edge method. Note here that the ends of the edges need not join together

D. Vehicle Support Calculation

The possibility of a vehicle hypothesis is evaluated by how well the filtered edges conform to the hypothesis in question. In adopting a conservative approach, both length and width of the vehicle must be adequately detected. Cases where only either the length or the width of the vehicle is detected potentially introduces ambiguous, potential false positives and such cases are eliminated.

The calculation of vehicle support is given by eqn. 1. Under the ideal case where there are only two segments perfectly aligned to the edges of the vehicle, the result is a multiplication of the length and width of the vehicle. Hence the equation for calculating the support:

$$support = \forall i, j \sum_{i \neq j} | S_i \times S_j | \qquad (1)$$

For example, if a large portion of the length but a small portion of the width is detected, the support will give a low value. The enforcement of these conservative rules are all conveniently embodied in a single equation (eqn 1).

IV. SLAM

Adopting the same naming convention in [7], the aim of SLAM from a probabilistic point of view is to estimate the SLAM posterior:

$$p(\Theta, s^t \mid z^t, u^t, n^t) \qquad (2)$$

Observations, z_t, in this case are the vehicle configuration hypotheses. Odometry information from CyCab provides sufficient information in predicting the motion of CyCab in the next time step with small uncertainty, assuming that the curvature is locally constant.

The SLAM method used is FastSLAM [8] [7]. FastSLAM is based on the observation that given the true path of the robot, the estimates of the various landmark positions will be conditionally independent. This statement is best expressed by the decomposition of the FastSLAM posterior as follows:

$$p(\Theta, s^t \mid z^t, u^t, n^t) = p(s^t \mid z^t, u^t, n^t) \prod_1^N p(\theta_n \mid s^t, z^t, u^t, n^t) \qquad (3)$$

Where the first term of the decomposition of eqn 3,

$$p(s^t \mid z^t, u^t, n^t) \qquad (4)$$

represents the path posterior of a mobile robot. The second term of the decomposition of eqn 3 ,

$$p(\theta_n \mid s^t, z^t, u^t, n^t) \qquad (5)$$

refers to the posterior distribution of landmark θ_n. Each individual probabilistic distribution in the decomposed SLAM posterior can be updated using an estimator recursively. Such a factorization was first introduced by Murphy and Russel in 1999 [9].

As a result of using FastSLAM, a set of hypothesis on the maps are obtained. Each hypothesis contains the pose of CyCab, and the set of landmark positions represented by the following:

$$S_t^{[m]} = s^{t,[m]}, \mu_{1,t}^{[m]}, \Sigma_{1,t}^{[m]}, \ldots, \mu_{N,t}^{[m]}, \Sigma_{N,t}^{[m]} \qquad (6)$$

V. MAP CONSTRUCTION

The different hypotheses from FastSLAM provides no single fix on robot path and landmark positions such that it is useful. Map construction is thus to produce the final map by combining the different hypotheses together.

A. Landmark Tagging

Landmark tagging is to assign an identification number (ID) to each physical landmark observed. Each of the different hypotheses contains its own belief on landmark locations.

Each time an observation is encountered, each hypothesis will make data associations of observations to landmarks. If the hypothesis associates the observation to a new landmark, a newly generated ID will be assigned. Otherwise, it will be an observation of a previously seen landmark. In this case, the old ID will be retained. This is to establish the association of landmarks across the different hypotheses with respect to a single obsevation.

B. Process Flow

The construction of the final map is performed over two stages sequentially:

1) The first stage calculates the posterior of each landmark for every hypothesis.
2) In the second stage, the landmark ID is used to group the set of landmark hypothesis together. However, different hypotheses makes the association between landmarks and observations differently and there is the possibility of having several IDs associated with a single observation. The second stage resolves this conflict and produces the final landmark to be included or updated on the map.

1) Landmark Posterior: Using the naming conventions in [7], the landmark posterior is of the form:

$$p(\theta \mid z) \qquad (7)$$

In this case, the posterior for every landmark that was believed to be observed for every hypothesis is evaluated. In FastSLAM, each and every landmark posterior within the context of a single hypothesis is given by the expression

$$p(\theta_n \mid s^t, z^t, u^t, n^t) \qquad (8)$$

This expression is calculated by the EKF during execution of FastSLAM. The distribution obtained from the EKF is a gaussian of the following form:

$$p(\theta_n \mid s^t, z^t, u^t, n^t) \sim N(\mu_{\theta_n}, \Sigma_{\theta_n}) \qquad (9)$$

Within the framework of FastSLAM, the reduction to obtain the posterior $p(\theta \mid z)$ is performed as follows:

$$\int_{s^{t,[m]}} p(\theta_n \mid s^t, z^t, u^t, n^t) p(s^t \mid z^t, u^t, n^t) ds^{t,[m]}$$
$$= \int_{s^{t,[m]}} p(\theta_n s^t \mid z^t, u^t n^t) ds^{t,[m]} \qquad (10)$$
$$= p(\theta_n \mid z^t n^t) \qquad (11)$$

Since, the particle filter in FastSLAM represents the distribution $p(s^t \mid z^t, u^t, n^t)$, equation 11 in the context of particle filters can be calculated by:

$$p(\theta_n \mid z^t n^t)$$
$$= \int_{s^{t,[m]}} p(\theta_n \mid s^t, z^t, u^t, n^t) \, p(s^t \mid z^t, u^t, n^t) ds^{t,[m]}$$
$$= \sum_{s_t^{[m]}} N(g^{-1}(s_t^{[m]}, z_t); \mu_{\theta_n}, \Sigma_{\theta_n}) \qquad (12)$$

The inverse measurement prediction model $g^{-1}(s_t, z_t)$ gives the expected landmark location given the state of the robot and the landmark observation by the robot.

2) Conflict Resolving: As explained previously (section V-B), there is the possibility of having several IDs associated to a single observation z_t. This set of M IDs is represented as a set I consisting of elements of unique IDs i_m:

$$I = \{i_1 \dots i_M\} \qquad (13)$$

For each i_m of the set I, there is a corresponding set, L_m, of hypothesized landmarks, observed by the different hypotheses, with the landmark ID i_m. The conflict is resolved by choosing one of the IDs to represent the landmark for the observation z_t made. The chosen ID is the most likely ID $i_{\widehat{m}}$ which gives the maximum value of the sum of all landmark posteriors in set L_m:

$$\widehat{m} = \arg\max_m \sum_{\theta_n \in L_m} p(\theta_n \mid s^t, z^t, u^t, n^t) \qquad (14)$$

Hence the landmarks with ID $i_{\widehat{m}}$ is to be used to compute the final landmark position in the final map and the rest of the landmarks, with IDs $I - i_{\widehat{m}}$, will be removed from the final map if it already exist in the final map. The landmark position is computed by the average of the landmarks in the set $L_{\widehat{m}}$.

However, a case to take note of is when the bulk of the hypotheses is relatively varied and there is no significant mass of consistency of hypotheses. This happens when there is no observations for some time and due to odometry uncertainty, the hypotheses in FastSLAM will start to spread out and report a larger number of IDs after an observation. This case can be easily detected as the maximum of the sum of landmark posteriors in equation 14 will have a smaller value in this case. The small value also implies high uncertainty and inconsistencies among the hypotheses.

VI. RESULTS

A. Vehicle Detection

Our method of stage 1 and stage 2 filtering in vehicle detection managed to produce correct vehicle hypotheses most of the time based on tests so far. After stage 1 only, the vehicle hypothesis to the left in figure 7 is accepted even when only a single side of the 'vehicle' is detected. But in fact, that side is in reality a wall. When stage 2 of vehicle detection is additionally applied (fig. 8), the same vehicle hypotheses as in figure 7 were rejected. The rejected hypotheses are represented by its additional inner and outer bounding boxes in this paper.

Due to the strong validation of stage 2, potentially correct hypotheses are inevitably eliminated in the process (vehicle hypothesis to the right in fig. 8). Currently, the validation of vehicles after stage 2 is processed independently of the previous validations.

Due to the numerous possibilities that could occur, our method of vehicle detection is not entirely foolproof. For example there might be cases where segments are cut off in a very suggestive manner that validates the vehicle hypothesis when in fact, it is a corner of a building (example in fig. 9). The laser scan data is limited as it provides only a planar geometrical view of the environment.

B. Map Construction

Map construction is able to form a consistent map on tests performed so far. This is mainly due to the fact that the tests were conducted within the context of the project. Vehicles as
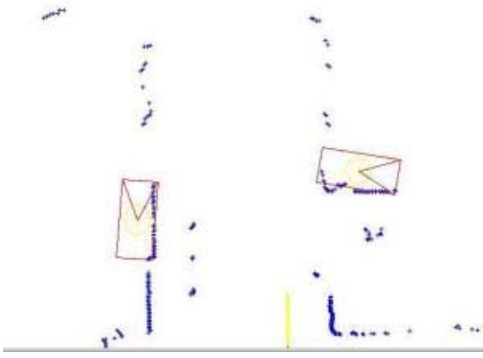
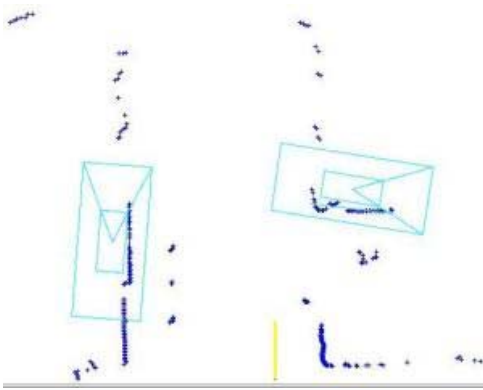Fig. 7: Accepted vehicle hypothesis after stage 1 of vehicle detection

Fig. 8: The same configuration as in figure 7 but with rejected vehicle hypothesis in their inner and outer bounding boxes after stage 2 of vehicle detection
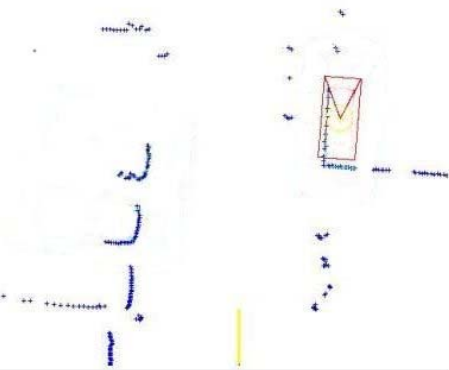

Fig. 9: A wrongly made hypothesis despite edge and vehicle support filtering

landmarks do not make very cluttered landmarks and hence, the ambiguity is low. Therefore, the performance of the map construction method in more cluttered environments remains to be seen

In figure 10, the different CyCab hypotheses are followed by a curvy line indicating its mean path taken and the various landmark hypotheses.
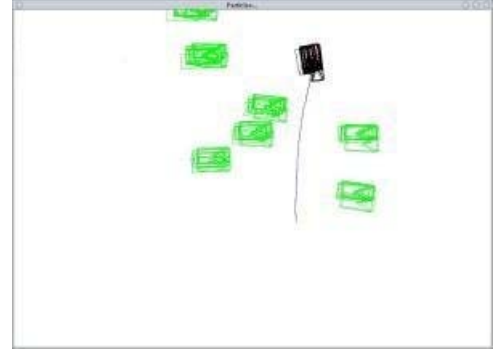

Fig. 10: The various CyCab and landmark hypotheses

Figure 11 illustrates what the map construction module perceives. It is basically a map of all the various hypotheses from FastSLAM combined together, overlaid with impact points from a single laser scan. Its consistency can be verified manually in this example.
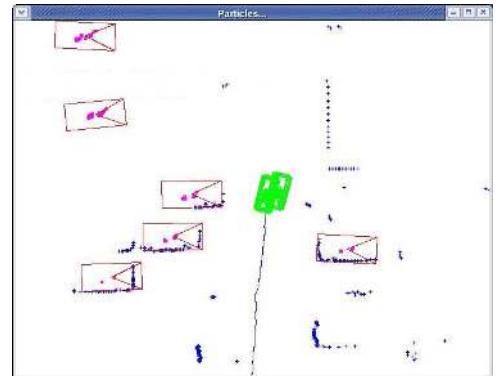

Fig. 11: Final map obtained by fusion of different hypotheses. With a single laser scan overlaid.

VII. Conclusion and Further Work

In this paper, the extraction of hypothesized vehicle poses from laser scan data only is presented. However the conservative vehicle detection approach is at the cost of eliminating potential vehicle hypotheses. Using the hypothesized vehicle poses as landmarks, FastSLAM is used to localize CyCab and build a final map by merging the different hypotheses. The main advantage is that the vehicle is able to perform mapping in any car park without installing any artifical external aids or references.

From our experiments laser scans alone are insufficient for creating reliable vehicle hypotheses. A camera to fuse information with the laser scans is envisaged. This can be

extended to include other objects that might potentially be found in the car park environment. Extensions will also be made to take the temporal aspects of observation and vehicle dynamics into account.

We hope that a map of this sort will be able to serve higher purposes within the framework of the automated car park.

REFERENCES

[1] R. Chatila. Representation issues. Technical report, Stockholm, KTH, 2002.

[2] R. Chatila and J.-P Laumond. Position referencing and consistent world modeling for mobile robots. In *IEEE International Conference on Robotics and Automation*, pages 138–145, 1985.

[3] O. Lebeltel. *Programmation Bayienne des Robots*. PhD thesis, Institut National Polytechnique de Grenoble, France, October 1999.

[4] Kenneth Jay Lee. Reactive navigation for an outdoor autonomous vehicle. Tech. report, University of Sydney, 2001.

[5] Y. Liu, R. Emery, D. Chakrabarti, W. Burgard, and S. Thrun. Using EM to learn 3D models with mobile robots. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2001.

[6] C. Martin and S. Thrun. Online acquisition of compact volumetric maps with mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, Washington, DC, 2002. ICRA.

[7] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada, 2002. AAAI.

[8] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexico, 2003. IJCAI.

[9] K. Murphy. Bayesian map learning in dynamic environments. In *Advances in Neural Information Processing Systems 11*. MIT Press, 1999.

[10] C. Pradalier and F. Colas. Expressing bayesian fusion as a product of distributions: Applications in robotics. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, October 2003.

[11] J. D. Tardos. Data association in slam. Technical report, Stockholm, KTH, 2002.