Semantic Feature Detection Statistics in Set Based Simultaneous Localization and Mapping

Felipe Inostroza[†], Keith Y. K. Leung[*], Martin Adams[‡]

Advanced Mining Technology Center, Universidad de Chile, Santiago, Chile
finostro@ug.uchile.cl[†], keith.leung@amtc.uchile.cl[*], martin@ing.uchile.cl[‡]

***Abstract*—The use of random finite sets (RFSs) in simultaneous localization and mapping (SLAM) has many advantages over the traditional random vector based approaches. These include the consideration of detection and clutter statistics and the circumvention of data association and map management heuristics in the estimation stage. To take full advantage of RFS based estimators in feature based SLAM, the feature detector's detection and false alarm statistics should be modelled and used in each SLAM estimation update stage. This paper presents principled techniques to obtain these statistics for semantic features extracted from laser range data, and focusses on the example of the extraction of circular cross-sectioned features, such as trees, pillars and lamp-posts, in outdoor environments. Comparisons of an RFS based SLAM algorithm - Rao-Blackwellized, Probability Hypothesis Density (RB-PHD)-SLAM, which utilizes the derived, variable feature probabilities of detection, and the same SLAM algorithm based on the typically assumed constant feature detection probabilities, within the sensor field of view, are provided. The results demonstrate the advantages of explicitly modelling feature detection statistics.**

I. Introduction

SLAM is a problem in robotics in which a robot uses its available sensor measurements to estimate a map of the operating environment, while concurrently determining its pose relative to the map. The general probabilistic approach currently adopted by the mobile robotics community uses random vectors to represent the robot and map state, and solves SLAM through stochastic filtering, or batch estimation [4]. Recently, a different representation has been introduced for feature-based maps using RFSs [16, 17], in which random vectors, typically representing the spatial location of individual landmarks, are placed in a set in which the cardinality (feature number) is also a random variable.

There are several benefits in using a RFS-based filtering approach to estimate the map in SLAM compared to a vector-based approach. Typically in vector-based approaches, the correspondence between measurements and landmarks is performed separately from the actual filter, and is determined using heuristics (e.g., by comparing the measurement to landmark likelihood with a preset threshold). These correspondences are required to determine which landmark estimate is updated by which measurement. In contrast, under an RFS SLAM framework, data association becomes a part of the landmark estimate update process for which Bayes theorem is applied. Essentially, the RFS approach updates landmark estimates by simultaneously associating them with every measurement, eliminating the need for heuristics. Another benefit of RFS-based filtering is that it accounts for detection statistics (feature probabilities of detection and false alarm). Finally, the RFS approach not only estimates the spatial position of landmarks, but also the number of landmarks that have entered the field of view of the robot's sensors. This is because the cardinality of a RFS is also a random variable that is estimated. Therefore, to fully utilize the capabilities of RFS based filters, modelling of a feature detector's detection statistics is necessary. Such statistics can be easily estimated in simulations, and approximated by various detection theoretic methods developed for radar [1, 16]. However, the most popular sensors used in robotics (vision and laser range finders) are typically used with complex, higher level feature detectors, often unaccompanied by detection statistics.

This paper proposes a method of obtaining the detection statistics of a laser data feature extractor, and its use in a RFS-SLAM implementation, known as *Probability Hypothesis Density* (PHD)-SLAM, one of the simplest mathematical finite set statistics (FISST) tools for estimation with RFSs, developed by Mahler [12]. The contribution of this paper is to demonstrate the implementation of PHD-SLAM with commonly used, 2D scanning laser range finders, and the importance of modelling sensor detection statistics in a principled manner. A simple feature detection strategy will be presented, in which the expected and variable probabilities of detection associated with laser range data are derived. Results of applying the laser based feature detector under a RB-PHD-SLAM framework will be presented and compared with results obtained from the same algorithm, with the usually assumed constant feature probabilities of detection within the sensor's field of view.

II. Random Finite Set SLAM and its requirements

In this section RFS-SLAM is introduced and its inclusion of the detection statistics of a feature detector is demonstrated.

A. System Model

SLAM is a state estimation problem in which the best estimate of the robot trajectory and map feature positions is sought over time, using all sensor measurements. In general, we can represent the underlying stochastic system using the

non-linear discrete-time equations:

$$\boldsymbol{x}_k = \boldsymbol{g}(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}, \boldsymbol{\delta}_{k-1}) \quad (1)$$

$$\boldsymbol{z}_k^i = \boldsymbol{h}(\boldsymbol{x}_k, \boldsymbol{m}^j, \boldsymbol{\epsilon}_k) \quad (2)$$

where
$\boldsymbol{x}_k$ represents the robot pose at time-step k,
$\boldsymbol{g}$ is the robot motion model,
$\boldsymbol{u}_k$ is the the odometry measurement at time-step k,
$\boldsymbol{\delta}_k$ is the process noise at time-step k,
$\boldsymbol{z}_k^i$is the i-th measurement vector at time-step k,
$\boldsymbol{h}$ is the sensor-specific measurement model,
$\boldsymbol{m}^j$ is a random vector for the position of landmark j,
$\boldsymbol{\epsilon}_k$ is the spatial measurement noise

Traditional vector-based approaches to SLAM concatenate random vectors for the robot and landmarks into a single vector for the estimation process. Furthermore, the generally complex data association problem needs to be solved so that i and j correspond to the same landmark. Within the RFS approach, the observed landmarks up to and including time-step k, are defined as

$$\mathcal{M}_k \equiv \{\boldsymbol{m}^1, \boldsymbol{m}^2, ..., \boldsymbol{m}^m\} \quad (3)$$

where the number of landmarks, $|\mathcal{M}_k| = m$, is also a random variable. In general, the landmark from which a measurement is generated is unknown. Furthermore, there is a probability of detection, P_D , associated with every landmark, implying that it may be missed with probability $1 - P_D$. Measurements may also be generated from sensor noise or objects of non-interest (clutter), with known distributions. The set of all n measurements at time-step k is defined as:

$$\mathcal{Z}_k \equiv \{\boldsymbol{z}_k^1, \boldsymbol{z}_k^2, ..., \boldsymbol{z}_k^n\} \quad (4)$$

Using a probabilistic framework and a filtering approach, the probability density function (PDF)

$$p\left(\boldsymbol{x}_{0:k}, \mathcal{M}_k | \mathcal{Z}_k, \boldsymbol{u}_{0:k}\right) \quad (5)$$

is sought, relative to the initial robot's pose, at each time step.

B. Rao-Blackwellized (RB)-Probability Hypothesis Density (PHD) SLAM

The posterior PDF (5) can be factored into the form [13, 16]

$$p\left(\boldsymbol{x}_{0:k} | \mathcal{Z}_k, \boldsymbol{u}_{0:k}\right) p\left(\mathcal{M}_k | \boldsymbol{x}_{0:k}, \mathcal{Z}_k, \boldsymbol{u}_{0:k}\right) \quad (6)$$

such that the first term in (6) is a conditional PDF on the robot trajectory and sampled using particles. The second term in (6) is the density of the map conditioned on the robot trajectory, which can be represented using a Gaussian mixture (GM). In the RFS-based approach, the map RFS is also assumed to follow a multi-object Poisson distribution[1]. This allows the PDF of the map RFS to be approximated as a time varying intensity function, v_k, represented as a GM:

$$v_k = \sum_i \omega_k^{[i]} \mathcal{N}\left(\boldsymbol{\mu}_k^{[i]}, \boldsymbol{\Sigma}_k^{[i]}\right). \quad (7)$$

In contrast to vector-based Rao-Blackwellized (RB)-particle filter (PF) approaches, which use the Extended Kalman filter (EKF) to update the Gaussians for individual landmarks, a probability hypothesis density (PHD) filter is used to update the map intensity function [16]. A brief overview of the main steps in the RB-PHD filter now follows, highlighting the importance of detection statistics.

1) Particle Propagation: At time-step k, the particles representing the prior distribution,

$$\boldsymbol{x}_{k-1}^{[i]} \sim p\left(\boldsymbol{x}_{0:k-1} | \mathcal{Z}_{1:k-1}, \boldsymbol{u}_{0:k-1}\right) \quad (8)$$

are propagated forward in time by sampling the motion noise, $\boldsymbol{\delta}_k^{[i]}$, and using the motion model (1):

$$\boldsymbol{x}_k^{[i]} = \boldsymbol{g}(\boldsymbol{x}_{k-1}^{[i]}, \boldsymbol{u}_{k-1}, \boldsymbol{\delta}_{k-1}^{[i]}) \sim p\left(\boldsymbol{x}_{0:k} | \mathcal{Z}_{1:k-1}, \boldsymbol{u}_{0:k-1}\right) \quad (9)$$

2) Generate Birth Gaussians: For each particle, its map intensity from the previous update, v_{k-1}, is augmented with $|\mathcal{Z}_{k-1}|$ new Gaussians with (an arbitrarily small) weight, ω_B, according to the PHD filter predictor equation:

$$v_k^- = v_{k-1}^+ + \sum_i^{|\mathcal{Z}_{k-1}|} \omega_B \mathcal{N}\left(\boldsymbol{\mu}_k^{[i]}, \boldsymbol{\Sigma}_k^{[i]}\right) \quad (10)$$

These new Gaussians, created at time-step k, represent potential new landmarks in the map, with means and covariances, $\left(\boldsymbol{\mu}_k^{[i]}, \boldsymbol{\Sigma}_k^{[i]}\right)$. These are determined by using the inverse measurement model from equation (2), i.e. $\boldsymbol{m}^j = \boldsymbol{h}^{-1}(\boldsymbol{x}_k, \boldsymbol{z}_k)$, with the previously updated pose $\boldsymbol{x}_{k-1}^{[i]}$, and every measurement in the previous measurement set, $\mathcal{Z}_{k-1}$. If the measurement model is not invertible, another strategy has to be devised to generate potential new features. These can be random Gaussians with large covariances in the areas were new features could be observed.

3) Map Update: The map intensity for each particle is updated with the latest measurements according to the PHD filter corrector equation:

$$v_k^+ = (1 - P_D) v_k^- + \sum_i^{|\mathcal{Z}_k|} \sum_j^{N_k^-} \omega_k^{i,j} \mathcal{N}\left(\boldsymbol{\mu}_k^{[i,j]}, \boldsymbol{\Sigma}_k^{[i,j]}\right) \quad (11)$$

where N_k^- is the number of Gaussians that comprise v_k^-. Here the first term is a copy of v_k^- with lowered weights to account for the possibility of missed detections. The second term adds a new Gaussian for each pair comprising a new measurement and an existing Gaussian in the intensity map. Note that instead of determining data association based on heuristics, the PHD filter determines how much a measurement should influence each landmark estimate. This is carried out by the weighting

[1]This implies that features are independently and identically distributed, while the number of features is Poisson distributed

factor calculation:

$$\omega_k^{i,j} = \frac{P_D \omega_k^j q\left(\boldsymbol{z}_k^i, \boldsymbol{\mu}_k^{[j]}, \boldsymbol{\Sigma}_k^{[j]}\right)}{\kappa + \sum_{l=1}^{N_k^-} P_D \omega_k^l q\left(\boldsymbol{z}_k^i, \boldsymbol{\mu}_k^{[l]}, \boldsymbol{\Sigma}_k^{[l]}\right)} \tag{12}$$

where $q()$ is the measurement likelihood given a feature estimate, and κ is the clutter density. The mean and covariance parameters for each new Gaussian, created from measurement i and landmark j, $\left(\boldsymbol{\mu}_k^{[i,j]}, \boldsymbol{\Sigma}_k^{[i,j]}\right)$, are determined using the EKF update step (Note that other variants of the Kalman filter (KF) would also be possible).

4) Importance Weighting and Re-sampling: The weighting and re-sampling of particles is the method used to update the robot trajectory PDF after propagation (also known as the proposal distribution). This is given by

$$p\left(\boldsymbol{x}_{0:k} | \mathcal{Z}_{1:k-1}, \boldsymbol{u}_{0:k-1}\right). \tag{13}$$

This has to to be updated to become a new PDF representing the robot trajectory after measurement updates (or the target distribution),

$$p\left(\boldsymbol{x}_{0:k} | \mathcal{Z}_{1:k}, \boldsymbol{u}_{0:k-1}\right). \tag{14}$$

Bayes rule allows the weighting distribution in terms of (13) and (14) to be expressed as

$$\frac{p\left(\boldsymbol{x}_{0:k} | \mathcal{Z}_{1:k-1}, \boldsymbol{u}_{0:k-1}\right)}{p\left(\boldsymbol{x}_{0:k} | \mathcal{Z}_{1:k}, \boldsymbol{u}_{0:k-1}\right)} = \eta p\left(\mathcal{Z}_k | \boldsymbol{x}_{0:k}, \mathcal{Z}_{1:k-1}\right), \tag{15}$$

in which η is a normalizing constant. Since (13) and (14) are sampled using particles, the weighting distribution, defined as w_k, is also sampled such that a weight is calculated for each particle. To solve (15), we use Bayes Theorem to express it as:

$$\begin{aligned} w_k &\equiv p\left(\mathcal{Z}_k | \boldsymbol{x}_{0:k}, \mathcal{Z}_{1:k-1}\right) \\ &= p\left(\mathcal{Z}_k | \mathcal{M}_k, \boldsymbol{x}_{0:k}\right) \frac{p\left(\mathcal{M}_k | \mathcal{Z}_{1:k-1}, \boldsymbol{x}_{0:k}\right)}{p\left(\mathcal{M}_k | \mathcal{Z}_{1:k}, \boldsymbol{x}_{0:k}\right)} \end{aligned} \tag{16}$$

Equation (16) can be solved because the map RFS is assumed to be multi-object, Poisson distributed. Note from (16) that the choice of the map, $\mathcal{M}_k$, for which the expression is evaluated in its general form is theoretically arbitrary since the left-hand side of (16) is independent of the map. This has led to multiple solutions that adopt the empty-set strategy, the single-feature strategy and multi-feature strategy in determining the particle weight in (16). It was previously shown that the choice of the map can have a significant effect on the performance of the filter and that the performance multi-feature strategy is superior to the others [9]. This is achieved at the cost of an increased computational cost. The multi-feature strategy is adopted in this work.

5) Merging and Pruning of the Map: Gaussians with small weights can be eliminated from the intensity function, while Gaussians that are close to each other can be merged [15, 17] This approximation is critical in limiting the computational requirement of the RB-PHD filter.

Importantly, within the above five steps, the map update and particle weighting steps require the knowledge of both the probability of detection of the feature detector and the distribution (PHD) for its false alarms. These important requirements are the subject of the next section.

III. Inferring Detection Statistics for Semantic Features

A. Estimating a Feature's Probability of Detection

Within the autonomous robotic navigation literature, feature detection statistics are largely ignored, and the uncertainty is considered to lie solely in the spatial domain, and typically modelled as range and bearing uncertainties [4, 20]. This implies that the probabilities of detection of features are assumed to be unity, and the probabilities of false alarm are assumed to be zero. In turn, it is then considered the task of the external map management and association heuristics to "deal" with false alarms and missed detections, before map estimation takes place.

Within the tracking community, detection statistics are considered to be of prime importance, however object detection probabilities are usually naively considered to be constant during trials, despite the fact that the relative positions of objects and the sensor, and any occlusions typically has a large effect on that object's detection probability [11]. Little attention is given to the shape of a sensor's field of view and the possibility of partial or total object occlusion, and their quantified effects on the expected detection statistics.

In [6], the requirement for a feature detector is removed by modelling laser range data, in which multiple measurements can be produced, by single "extended" features. The number of such extended feature measurements is modelled as a Poisson RFS. They model the probability of detection using the estimated features.

The aim of this section is to therefore provide a quantified model of the probabilities of detection and false alarm, based on laser range based features. This model does not use any information on the feature detector itself and can therefore be used with any detector that estimates both the position and shape of the object. This excludes line detectors that don't provide beginning and end points. In the context of [6], the method to calculate probabilities of detection proposed in this section can be used not only to calculate these for Granstrom's feature detector, but also to estimate the mean of his proposed extended feature RFS.

As shown in Figure 1, given an estimate of the robot's location and the location and other attributes, such as the shape, of features (i.e. a SLAM estimate), the number of laser range points that the feature is expected to return can be estimated. This point estimation process is a sensor modelling technique referred to as *ray tracing* in the robotics literature [20]. Comparing estimated and measured distances, allows expected feature estimates to be labelled as either occluded, partially occluded or unoccluded. The number of estimated, unoccluded points per feature determines the proportion of the landmark that is in the field of view of the sensor. The analysis in this section demonstrates that feature probabilities

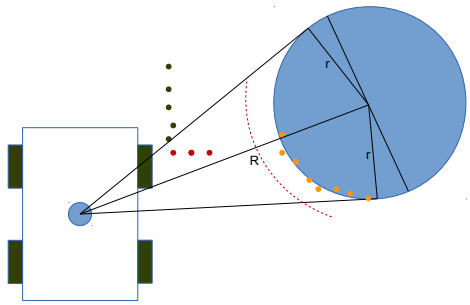


Fig. 1: Analysis of range data from a circular shaped feature. Based on the SLAM state estimate, the laser range beams that would hit the feature, if not occluded, can be determined (red and yellow circles). Beams with range values several times the range standard deviation shorter than expected (red points) are discarded from the detection probability analysis. The remaining (yellow) points are used to estimate the feature's probability of detection.

of detection can be experimentally quantified based on this number, via statistical analyses on laser range data sets. Initially, a dataset is required from an environment where the ground truth positions of features are known, via independent means. A simple way to achieve this is through the use of features identifiable by humans - i.e. semantic features.

B. Estimating Probabilities of False Alarm

In the case of the probability of false alarm it is infeasible to theoretically model every possible laser range scan that does not contain a semantic feature of choice. Importantly, the statistical representation of false alarms in RB-PHD- SLAM is a Poisson random set, which only requires an estimate of their expected number. In a manner similar to the probabilities of detection outlined above, the statistical analysis of laser range based data, known to not contain the chosen semantic features, can yield an informative estimate of the probability of false alarm.

These concepts for estimating feature detection and false alarm probabilities will be applied to a laser range finder based circle detector in Section V. The next section introduces the circle detector itself, in the context of well known feature detectors within the robotics field.

IV. Laser Based Features and Their Detection Statistics

This section provides a brief overview of the main feature detection algorithms applied to laser range data and, at the same time, highlights the publicized problems in their application to data sets in which chosen feature types yield few data points per scan. It will then present a simple circular feature detector, which can be applied in outdoor scenarios in which approximately circular cross-sectioned features such as trees, pillars and lamp posts are abundant. This circular feature detector is an extension of that proposed by Durrant-Whyte *et al* [3].

A. Why Semantic Features?

Since laser range finders yield range decisions, it should be possible to incorporate all of these into a SLAM estimation algorithm. Various mapping algorithms achieve this, via scan matching techniques [10], although these techniques typically require good initial estimated robot pose to scan alignment estimates for them to function correctly.

The reasons why most SLAM algorithms do not attempt to process every laser range value are as follows. Firstly, contrary to many radar and sonar devices, commercially available laser range finders usually internally process the received power values to provide range decisions at distinct bearing angles, instead of outputting the entire received power array (a-scope) at predefined range increments. This means that the device makes its own hypothesis test on a per a-scope basis, and provides only the final decision of this test, yielding a single range decision. Under favorable operating conditions, these range decisions typically correspond well with the true distances to objects, however they are still prone to the problems of false alarms and missed detections under sub-optimal target/environmental conditions. Secondly, because of the high angular resolution of laser range finders, they usually provide the user with a multitude of range decisions, far in excess of which most SLAM algorithms can process.

These two facts have advocated the compression of laser range data into so called high level, and typically semantic, features. This is to minimize the negative impact of individual false alarms and missed detections and simultaneously keep SLAM input data levels manageable.

B. Current Laser Range Based Feature Detectors

Global detectors, such as RANSAC and the Hough transform, have been applied to laser range data, mostly to extract lines [18]. These methods have several advantages, such as tolerance to partial occlusions, however they rely on many feature inliers being available within the laser data sets.

In [19] Nuñez et al demonstrated a detector capable of extracting both line segments and circular curves by estimating the curvature of the scan. Zero curvature segments are detected as lines and constant curvature segments are detected as circular arcs. The algorithm was designed to work with indoor scans where circles are usually observed at close range and return many range values.

Other laser point based feature detectors include the recursive split and merge algorithm [7] and Gauss-Newton extraction algorithm [21].

Despite the varying degrees of mathematical rigor in state of the art feature detection algorithms, global detectors have been shown to not improve the results enough to compensate for their increased computational complexity [18]. Durrant-Whyte *et al* presented a simple detector which seeks clusters of points and assigns a circle to represent these, with diameter equal to the distance between the first and last points of the cluster. To demonstrate the importance of estimating detection statistics, and their integration into PHD-SLAM, the simple circle detector of [3] will be extended in the next section,

for the robust detection of circular cross sectioned objects including pillars, trees and lamp posts.

C. Detection of Circular Objects

The circle based detector of [3] is extended here by, replacing it's heuristic estimation of the circle parameters by a non-linear optimization approach. Also an additional step has been added to remove some of the false alarms produced by the algorithm.

The detector works in 3 steps: Clustering; circle fitting and false alarm reduction.

1) Clustering: The first step of the algorithm is to segment the laser scan in a set of simple clusters of closely spaced points. To obtain these clusters the whole scan is iterated in its natural order. If the Euclidean distance between two consecutive points is greater than a threshold, a break between two different clusters is declared.

2) Circle Fitting: A circle is fitted to each cluster by minimizing the squared error of the fit as shown in equation (17).

$$\underset{x_c,y_c,r_c}{\text{minimize}} \quad \sum_i (\sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} - r_c)^2 \qquad (17)$$

Where (x_c, y_c, r_c) is the center and radius of the circle and (x_i, y_i) are the coordinates of each laser range point in the cluster. This optimization problem is the same as the one presented in [21] but is solved using the Levenberg-Marquardt algorithm, which has been shown to be more robust than the Gauss-Newton method [14]. To initialize the algorithm the mean point position is used as the circle's center and its radius is set as half the distance between the first and last point.

3) False Alarm Reduction: At this stage every cluster has an associated, fitted circle. Cluster pruning is then necessary, in which clusters and their corresponding circles are removed based on a detection theoretic, statistical analysis of their parameters.

Each cluster is characterized by three parameters:

- Fitting Error for the circle fit (FE)

$$\text{FE} = \sum_i (\sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} - r_c)^2 \qquad (18)$$

- Radius of the detected circle (r_c)
- Convexity (C) of the circle. This is a measure of the difference between the distance from the robot to the center of the fitted circle and the mean of the points (See Figure 2) and is given by

$$C = \sqrt{(x_r - x_c)^2 + (y_r - y_c)^2} - \sqrt{(x_r - \frac{1}{n}\sum_i^n x_i)^2 + (y_r - \frac{1}{n}\sum_i^n y_i)^2} \qquad (19)$$

To achieve false alarm reduction, based on the above parameters, concepts from detection theory can be applied [8]. Histograms representing correctly and falsely detected circular features were generated with respect to each of the above parameters. This required the generation of ground truth information within a test area, containing the true centers and radii of circular sectioned objects, such as trees. This test area comprised a ground truth map of a park area near the Universidad de Chile.

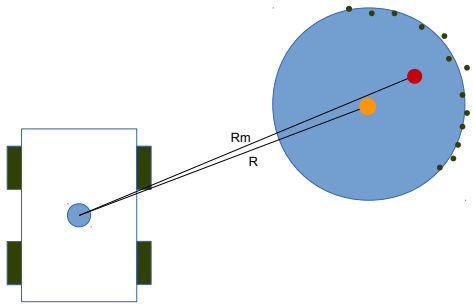


Fig. 2: If the mean of the laser returns (red point) is further away from the robot than the center of the estimated circle (orange point) then the object does not form a convex circular cross section with respect to the robot's location.

Naturally the generality of such an environment is questionable, in terms of the circular features contained within it. However, since the SLAM experiments were to take place in an environment containing a significant number of trees, this environment was deemed sufficiently general. In general, if the sought features are based on *any type* of semantic information, detectors for those semantics can be tuned in a similar manner, using ground truth data sets from environments known to contain a significant number of the type of feature sought.

Within the park environment, multiple 2D laser scans from different positions were recorded and approximately manually aligned to form an initialization for the Iterative Closest Point [2] algorithm, which in turn generated a more exact alignment of the data [10]. This resulted in a registered 2D point cloud of the Park. Point clusters were then manually extracted and compared to the actual environment to determine if they corresponded to actual circular sectioned features. Based on positive matches, the cluster centers and actual features' radii (measured by hand) were noted. This resulted in a list of circular feature (typically tree trunk) center coordinates and their respective radii. ICP [2] also determined the position at which each scan was recorded.

After the ground truth list was attained, the laser range finder and circular feature detector were used to automatically detect multiple circular sectioned features, at multiple locations, based on the procedure outlined in Sections IV-C1 and IV-C2 within the test area. From the multiple detections, the histograms in Figures 3, 4 and 5 were generated. These histograms could be used directly to achieve false alarm reduction. From each histogram it is evident that the application of appropriate, independent detection thresholds on the FE, radius and convexity measure could be applied so as to reject the false alarms which correspond to feature parameters outside of the bounds which contain the detections. However, care is necessary before the application of such a simplistic treatment, since any correlation between these parameters must first be determined.

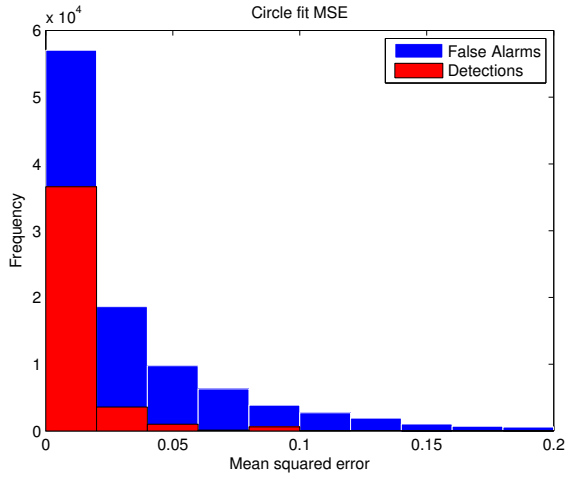


Fig. 3: The fitting error can be used to remove some of the false alarms.

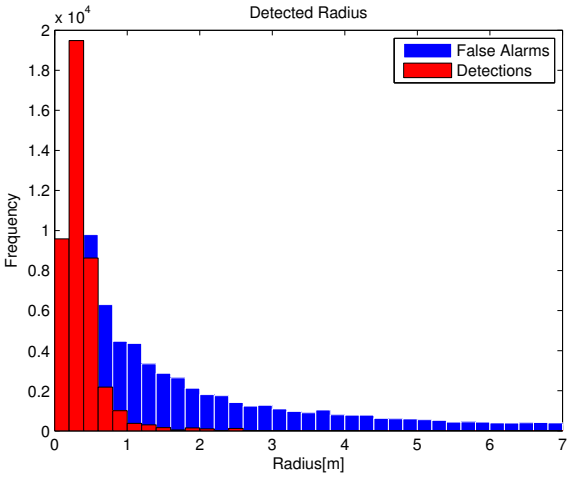


Fig. 4: Histogram for the detected radius. Note: The tail for the false alarm distribution is very long and is not shown in this figure.

To determine such correlations, and appropriate methods to discard some of the false alarms, standard techniques such as Fisher's linear discriminant [5] or Hotelling ellipsoids [8] can be used. After comparing these two methods, higher detection rates, for given false alarm rates (i.e. superior Receiver Operating Characteristics (ROCs)) were noted for the Hotelling ellipsoidal method - which is therefore adopted here.

Hotelling Ellipsoid Method

This method fits a multivariate Normal distribution to the detected circle's parameters and uses the distribution parameters to create a confidence ellipsoid, as shown in Figure 6. This figure shows the distribution of the three variables for the case of true detections and a corresponding 99.9% confidence ellipsoid representing a multivariate Normal distribution, based on the parametric data points. The size of the ellipse can be determined using the estimated covariance and the multivariate normal distribution. The Hotelling Ellipsoidal method will reject any measurement that falls outside of the ellipse.

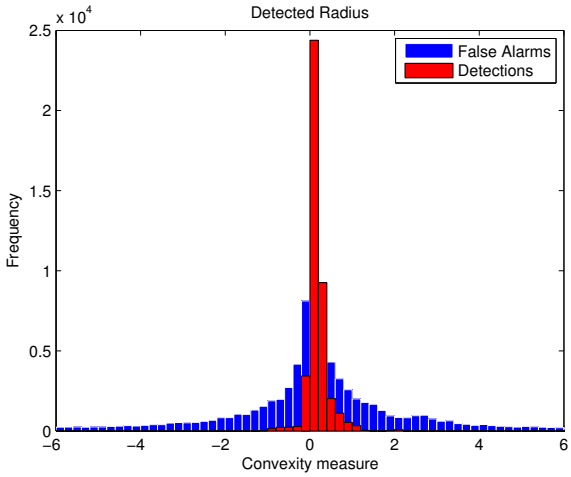


Fig. 5: Histogram for the Convexity measure. Note: The tails for the false alarm distribution are very long and are not shown in this figure.

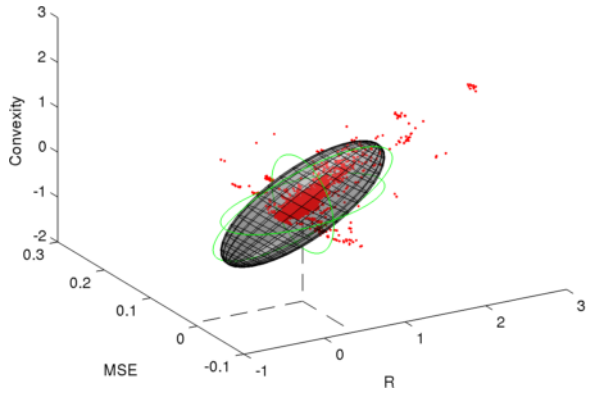


Fig. 6: A Gaussian approximation of the detections can be used to generate an ellipse to remove much of false alarm and keep most of the detections.

V. On-Line Determination of Detection Statistics

In this section the results of applying the method described in section III to the detector presented in section IV using the ground truth dataset will be shown. The circular feature detector was applied to the scans collected from the park data set. By using the known pose of the robot and landmark, every measurement was manually associated with it's closest feature. If the distance between these measurements and their closest associated features was larger than one meter, the measurements were deemed to be false alarms.

The associated measurements were used to determine the probability of detection of the algorithm conditioned on the number of unoccluded points. First the expected number of unoccluded points for each feature was calculated based on the estimated robot pose and feature positions. By determining the ratio of actual detections to the total number of times that a specific number of unoccluded points was calculated, the probability of detection for each detected feature was determined (Equation 21).

$$P_D(\boldsymbol{m}_i|\mathcal{M}_k, \boldsymbol{x}_k) \sim P_D(\boldsymbol{m}_i|n_p = N_p(\mathcal{M}_k, \boldsymbol{x}_k)) \qquad (20)$$

Number of unoccluded points	P_D
0	0.0014 (383/273025)
1	0.0244 (1514/62025)
2	0.0696 (1990/28582)
3	0.3595 (6983/19423)
4	0.7562 (7845/10374)
5	0.8854 (6118/6909)
6	0.8987 (3761/4185)
7	0.7824 (2078/2656)
8	0.7607 (1068/1404)
9	0.7679 (870/1133)

TABLE I: The number of unoccluded points greatly influences the probability of detection. The amount of detections and instances were specific number of points was expected is shown in brackets.

$$P_D(\boldsymbol{m}_i|n_p) \sim \frac{\sum_{j=0}^{N_s} \sum_{\boldsymbol{m}\in\mathcal{M}_j} N_p(\boldsymbol{m},\boldsymbol{x}_j) = n_p \wedge C(\boldsymbol{m},j)}{\sum_{j=0}^{N_s} \sum_{\boldsymbol{m}\in\mathcal{M}_j} N_p(\boldsymbol{m},\boldsymbol{x}_j) = n_p} \tag{21}$$

Where
n_p is the number of unoccluded points,
$N_p()$ is the function that estimates n_p by doing ray tracing,
N_s is the total number of scans in the dataset,
$C(m,j)$ is an indicator function that shows whether feature m was detected at time j.

As can be seen in Table I the probability of detecting a circular object (in this case a tree) is highly dependant on the number of unoccluded points. It should be noted that there are less instances where the number of unoccluded points was high, so for higher number of unoccluded points the variance of the estimated P_D will be higher. In the implementation of the RFS-SLAM if more than six points were not occluded the probabilities of detection were replaced with the probability of six unoccluded points. Also the probabilities that were lower than 10% were therefore replaced with a zero.

Determining the distribution of false alarms is much easier since the RFS-PHD framework does not model spatial variations on its distribution. Only the distribution for the number of false alarms per laser scan is necessary, which does not require any SLAM map estimates. Accordingly, the false alarm histogram, obtained by plotting the number of times a particular false alarm number occurs over N_s scans, is shown in figure 7 where a continuous Poisson distribution is also fitted to the data. Although the PHD Filter assumes this Poisson type distribution, it is noted that the discrete histogram is similar to it. The average number of false alarms N_{fa} per laser scan was estimated to be:

$$E(N_{fa}) = 9.27 \tag{22}$$

VI. RB-PHD-SLAM with Circular Feature Detection Statistics

In this section a comparison of RB-PHD-SLAM in a park environment, based on the use of the derived detection statistics and a naive implementation with assumed constant feature detection probabilities, within the 40m radius sensor field of view, is shown in Figure 8.

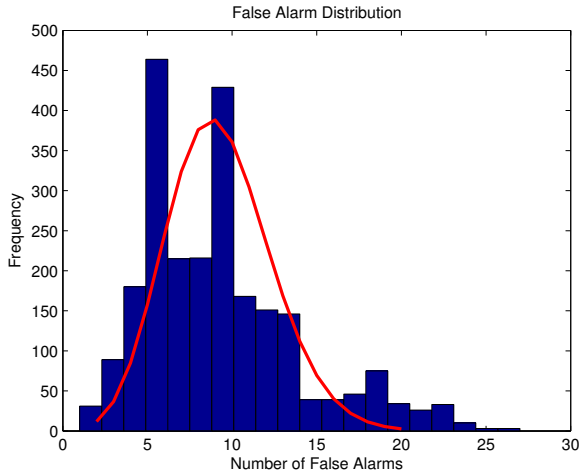


Fig. 7: Histogram for the false alarms. A Poisson distribution is fitted to the data.

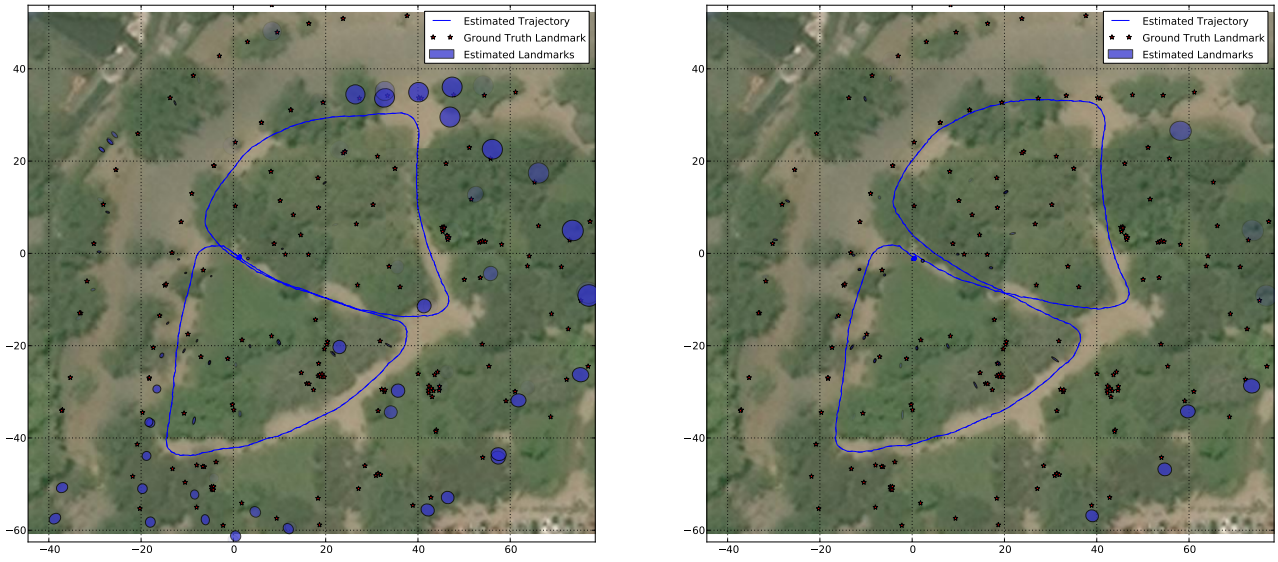
The robotic platform used was a Clearpath Husky A-200 robot equipped with a Sick LD-LRS-1000 laser range finder. The Husky's wheel encoders provided odometry measurements $\boldsymbol{u}_{0:k-1}$ for the motion model in Equation 8. The robot was run in the same park used to determine the detection statistics.

Figures 8a and 8b show the performance of RB-PHD-SLAM. The map estimates are represented by plotting an ellipse for each gaussian in the gaussian mixture (v_k^+ (Equation 11) and estimated vehicle trajectories are built using the current position for the particle $\boldsymbol{x}_k^{[i]}$ (Equation 9) with the highest weight w_k (Equation 16) at every time step. A superior SLAM performance is indicated in Figure 8a, in which the vehicle trajectory and quality of the associated map estimates, in terms of circular feature location and number, out-performs the assumed constant detection probability based RB-PHD-SLAM results shown in Figure 8b. In the latter case, the mismatch between the real and estimated probabilities of detection causes features to be removed from the map soon after they leave the field of view of the robot's laser scanner.

VII. Conclusions

The importance of modelling detection statistics within real SLAM experiments has been high-lighted in this paper. A simple semantic feature based detector was presented, together with detection theoretic based methods for the evaluation of each features' probabilities of detection and false alarm. The methods presented can be used with any feature detector that estimates the shape of an object. The techniques were applied to a simple circle detector, for use in environments in which multiple circular cross sectioned features are expected.

The derived detection statistics where used in a RB-PHD-SLAM framework in a park environment, in which the primary circular sectioned features were trees. The results demonstrated superior SLAM estimates, in terms of vehicle trajectory, map feature number and location estimates when



(a) (b)

Fig. 8: RB-PHD-SLAM results in a park environment. The vehicle traversed an approximate "figure eight" shape along the dirt track shown in the superimposed satellite images. The blue ellipsis represent the spatial feature estimates, stars represent the ground truth map and the blue line is the estimated robot trajectory. Figure (a) shows the results from running the filter with the probability of detection proposed in this paper. Figure (b) shows the filter running with a constant probability of detection within a circular field of view.

the derived probability of detection model was compared with a typical implementation which assumed constant detection statistics within the sensor field of view.

ACKNOWLEDGEMENTS

The authors would like to acknowledge Conicyt, Fondecyt Project 1110579, the Advanced Mining Technology Centre (AMTC), Universidad de Chile and Clearpath Robotics Inc.

REFERENCES

[1] M. Adams, J. Mullane, E. Jose, and B.N. Vo. *Robotic Navigation and Mapping with Radar*. Artech House, Boston, USA, 2012.

[2] P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(2):239–256, Feb 1992.

[3] M. W M G Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (slam) problem. *Robotics and Automation, IEEE Transactions on*, 17(3):229–241, Jun 2001.

[4] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part I. *Robotics & Automation Magazine, IEEE*, 13(2):99–110, 2006.

[5] Ronald A Fisher. The statistical utilization of multiple measurements. *Annals of eugenics*, 8(4):376–386, 1938.

[6] Karl Granstrom and Umut Orguner. A phd filter for tracking multiple extended targets using random matrices. *Signal Processing, IEEE Transactions on*, 60(11):5657–5671, 2012.

[7] Steven L Horowitz and Theodosios Pavlidis. Picture segmentation by a tree traversal algorithm. *Journal of the ACM (JACM)*, 23(2):368–388, 1976.

[8] S. Kay. *Fundamentals of Statistical Signal Processing, Vol II - Detection Theory*. Prentice Hall, 1998.

[9] K. Y. K. Leung, F. Inostroza, and M Adams. An improved weighting strategy for rao-blackwellized probability hypothesis density simultaneous localization and mapping. In *Proceedings of International Conference on Control, Automation, and Information Sciences*, 2013.

[10] F. Lu and E. Milios. Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans. *Journal of Intelligent and Robotic Systems*, 18(3):249–275, 1997.

[11] R. Mahler. A survey of PHD filter and CPHD filter implementations. In *Proc. SPIE Defense & Security Symposium of Signal Processing, Sensor Fusion and Target Recognition XII*, April 2007.

[12] R. P. S. Mahler. *Statistical multisource-multitarget information fusion*, volume 685. Artech House Boston, 2007.

[13] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, et al. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the National conference on Artificial Intelligence*, pages 593–598, 2002.

[14] Jorge J Moré. The levenberg-marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105–116. Springer, 1978.

[15] J Mullane, B-N Vo, and M. D. Adams. Rao-blackwellised phd slam. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 5410–5416, 2010.

[16] J. Mullane, B.-N. Vo, M. D. Adams, and B.-T. Vo. A random-finite-set approach to bayesian slam. *Robotics, IEEE Transactions on*, 27(2):268–282, 2011.

[17] J. Mullane, B.-N. Vo, M. D. Adams, and B.-T. Vo. Random finite sets for robot mapping and slam. *Springer Tracts in Advanced Robotics*, 72, 2011.

[18] V. Nguyen, A. Martinelli, N. Tomatis, and R. Siegwart. A comparison of line extraction algorithms using 2d laser rangefinder for indoor mobile robotics. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 1929–1934, Aug 2005.

[19] P. Núñez, R. Vázquez-Martín, J.C. del Toro, A. Bandera, and F. Sandoval. Natural landmark extraction for mobile robot navigation based on an adaptive curvature estimation. *Robotics and Autonomous Systems*, 56(3):247 – 264, 2008.

[20] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*, volume 1. MIT press Cambridge, 2005.

[21] Sen Zhang, M Adams, Fan Tang, and Lihua Xie. Geometrical feature extraction using 2d range scanner. In *Control and Automation, 2003. ICCA'03. Proceedings. 4th International Conference on*, pages 901–905. IEEE, 2003.