Evaluating Set Measurement Likelihoods in Random-Finite-Set SLAM

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

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

***Abstract*—The use of random finite set (RFS) 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 estimator. However, the equations involved in the RFS-SLAM formulation are computationally more complex compared to the vector-based formulation. The evaluation of the set measurement likelihood is one of the computationally complex steps, as it is necessary to consider the likelihood of all possible landmark to measurement correspondences. In general, a brute-force approach in calculating a set measurement likelihood is computationally intractable, and such an approach prevents a RFS-SLAM algorithm to perform in real time. This paper presents a collection of methods for efficiently computing and approximating the set measurement likelihood. The proposed methods are validated in both simulations and using real experimental data.**

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 the SLAM solution through stochastic filtering, or batch estimation [1]. Recently, a different representation has been introduced for feature-based maps using RFSs [2, 3], in which, random vectors typically representing the spatial location of individual landmarks are placed in a set, in which the cardinality (or size) 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 many (but not all) vector-based approaches, data association (or the correspondence between measurements and landmarks) is performed separately from the actual filter, and is determined using heuristics (e.g., by comparing the measurement likelihood to a landmark with a preset threshold). These correspondences are required to determine which landmark estimate is updated by a 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, and does not rely on any heuristics in the process. Another benefit of RFS-based filtering is that it can account for detection statistics (i.e., the probability of detection of landmarks, and the amount of clutter or outliers expected from a scene). 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.

Fig. 1: The robotic platform and scanning laser range finder used in generating the experimental dataset.

Similar to vector-based filtering methods such as the Kalman filter (KF), RFS-based filtering methods also stem from the the recursive Bayesian filtering paradigm. A set of mathematical tools called finite set statistics (FISST) was developed by Mahler [4] for handling multi-target estimation problems in which RFSs are used, and allows the application of Bayesian estimation techniques for use with RFSs. However, the use RFSs causes the equations of the estimator to become more complex, especially in the evaluation of the measurement likelihood that appears in the update step of the Bayes filter. In [5], a method for evaluating set measurement likelihoods was proposed whereby the probability of detection of landmarks is assumed to be one, while the clutter intensity is assumed to be zero. This simplified the the set measurement likelihood calculation, but it is still considered a brute-force approach as it still has factorial complexity.

This paper proposes a collection of methods for the efficient evaluation of the set-based measurement likelihoods in RFS

SLAM. The contribution of this paper is to show how the set measurement likelihood can be evaluated and approximated for practical implementation that is more appropriate for real time applications. In Section II, A brief formulation of RFS SLAM, and its realization as the Rao-Blackwellized (RB)-probability hypothesis density (PHD) SLAM algorithm will be reviewed to show where the set measurement likelihood appears. Section III will show the brute-force approach in evaluating the set measurement likelihood and will present several methods in which the process can be made to be more computationally tractable. Sections IV and V will show the simulation and experimental results of performing SLAM using the set measurement evaluation techniques proposed in this paper. The experimental dataset we use was collected using a scanning laser range finder on the robotic platform shown in Fig. 1.

II. Problem Formulation

This section will show where the set measurement likelihood appears in the RFS formulation of SLAM, to motivate the need for its efficient computation. SLAM is a state estimation problem in which the best estimate of the robot trajectory and map feature positions over time are sought by using all available sensor measurements. In general, the underlying stochastic system can be represented using the non-linear discrete-time equations:

$$\mathbf{x}_k = \mathbf{g}\left(\mathbf{x}_{k-1}, \mathbf{u}_k, \boldsymbol{\delta}_k\right) \tag{1}$$

$$\mathbf{z}_k^j = \mathbf{h}\left(\mathbf{x}_k, \mathbf{m}^i, \boldsymbol{\epsilon}_k\right) \tag{2}$$

where

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

The set of all n measurements at time-step k is defined as:

$$\mathcal{Z}_k \equiv \left\{\mathbf{z}_k^1, \mathbf{z}_k^2, \ldots, \mathbf{z}_k^n\right\} \tag{3}$$

The measurements in this set may originate from a real landmark, or may be clutter (or false alarms). The likelihood of a measurement set being clutter is defined as $p_\kappa\left(\mathcal{Z}\right)$.

Similarly, by placing the independent random vectors for the map into a RFS, the observed landmarks up to time-step k are defined as:

$$\mathcal{M}_k \equiv \left\{\mathbf{m}^1, \mathbf{m}^2, \ldots, \mathbf{m}^m\right\}, \tag{4}$$

where the number of landmarks, $|\mathcal{M}_k| = m$, is also a random variable. Using the above definitions in (3) and (4), the required estimate is:

$$p\left(\mathbf{x}_{0:k}, \mathcal{M}_k | \mathcal{Z}_{1:k}, \mathbf{u}_{0:k}\right) \tag{5}$$

An advantage of RFS estimation is the possibility of incorporating detection statistics. The *probability of detection*, $P_D\left(\mathbf{x}, \mathbf{m}\right)$, is the probability of obtaining a measurement given a robot pose, and a landmark position.

Similar to a vector-based formulations, (5) can theoretically be solved by Bayesian estimation [4] using the following recursive equations:

$$\begin{aligned} &p\left(\mathbf{x}_{0:k}, \mathcal{M}_k | \mathcal{Z}_{1:k-1}, \mathbf{u}_{0:k}\right) \\ =&p\left(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k\right) p\left(\mathbf{x}_{0:k-1}, \mathcal{M}_{k-1} | \mathcal{Z}_{1:k-1}, \mathbf{u}_{0:k-1}\right) \end{aligned} \tag{6}$$

$$\begin{aligned} &p\left(\mathbf{x}_{0:k}, \mathcal{M}_k | \mathcal{Z}_{1:k}, \mathbf{u}_{0:k}\right) \\ =&\frac{p\left(\mathcal{Z}_k | \mathbf{x}_{0:k}, \mathcal{M}_k\right) p\left(\mathbf{x}_{0:k}, \mathcal{M}_k | \mathcal{Z}_{1:k-1}, \mathbf{u}_{0:k}\right)}{\int p\left(\mathcal{Z}_k | \mathbf{x}_{0:k}, \mathcal{M}_k\right) p\left(\mathcal{M}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k}\right) \mathrm{d}\mathcal{M}_k} \end{aligned} \tag{7}$$

In (7), the set measurement likelihood factor

$$p\left(\mathcal{Z}_k | \mathbf{x}_{0:k}, \mathcal{M}_k\right) \tag{8}$$

appears both in the numerator, and the normalizing factor in the denominator. The Bayes filter is generally computationally intractable in both the vector form and the RFS form without making further assumptions. Before exploring how (8) can be evaluated, it is also useful to see where it appears in a realization of the RFS-based Bayes filter.

In the vector form of the Bayes filter, Gaussian assumptions allow the KF to be derived from the Bayes filter [6]. With RFSs, the map can be assumed to follow a *multi-object Poisson distribution*[1]. Clutter measurements are also assumed to be multi-object Poisson distributed. By approximating these distributions using their first statistical moments (their PHDs or *intensities*, v), the Bayes filter can be implemented as the PHD filter. Furthermore, by representing elements within $\mathcal{M}_k$ with Gaussian random vectors, the PHD of the map can be expressed as a Gaussian mixture (GM):

$$v_k = \sum_{i=1}^{m} w_k^i \mathcal{N}\left(\boldsymbol{\mu}_k^i, \boldsymbol{\Sigma}_k^i\right) \tag{9}$$

where w_k^i is the weight of the i-th Gaussian. From this, the Bayes filter is approximated as the GM-PHD filter [7]. The update of each Gaussian is performed using the KF or one of its derivative such as the Extended Kalman filter (EKF) for non-linear systems, where each Gaussian is updated using every measurement. The details of this procedure can be found in [5].

To the best knowledge of the authors, the works by Mullane et al. [3] and Lee et al. [8] are currently the only computationally tractable approaches to RFS-based SLAM. Both approaches use a similar technique to a factored solution to SLAM (FastSLAM) [9] and factors (5) into:

$$p\left(\mathbf{x}_{0:k} | \mathcal{Z}_{1:k}, \mathbf{u}_{1:k}\right) p\left(\mathcal{M}_k | \mathbf{x}_{0:k}, \mathcal{Z}_{1:k}, \mathbf{u}_{1:k}\right) \tag{10}$$

The first term in (10) is a conditional probability density function (PDF) on the robot trajectory and is sampled using particles. The second factor in (10) is the density of the map conditioned on the robot trajectory. Mullane et al. [3] used the

[1] This implies that features are independently and identically distributed, while the number of features follow a Poisson distribution [4].

GM-PHD filter [7] to solve for the second factor, and called this method RB-PHD-SLAM. The approach by Lee et al. [8] is based on single-cluster processes.

The detailed formulation of the RB-PHD-SLAM algorithm can be found in [3, 10]. The focus here will be on the importance weighting step of RB-PHD-SLAM since it involves the use of the set measurement likelihood (8). The importance weighting step is essential in updating the trajectory estimate in RB-PHD-SLAM after a measurement update. The importance weight of a particle can be expressed as:

$$\omega_k^{[\iota]} \equiv \omega_{k-1}^{[\iota]} \frac{p\left(\mathbf{x}_{0:k}|\mathcal{Z}_{1:k}, \mathbf{u}_{1:k}\right)}{p\left(\mathbf{x}_{0:k}|\mathcal{Z}_{1:k-1}, \mathbf{u}_{1:k}\right)} \tag{11}$$

$$= \omega_{k-1}^{[\iota]} \eta p\left(\mathcal{Z}_k|\mathbf{x}_{0:k}^{[\iota]}, \mathcal{Z}_{1:k-1}\right) \tag{12}$$

The normalizing constant η can be ignored as all particle weights will be multiplied by this same constant. To solve (12), it can be expressed as:

$$\omega_k^{[\iota]} = \omega_{k-1}^{[\iota]} \int p\left(\mathcal{Z}_k|\mathcal{M}_k, \mathbf{x}_{0:k}^{[\iota]}\right) p\left(\mathcal{M}_k|\mathbf{x}_{0:k}^{[\iota]}, \mathcal{Z}_{1:k-1}\right) \mathrm{d}\mathcal{M}_k \tag{13}$$

The form of (13) is similar to the importance weighting equation in FastSLAM. However, (13) is computationally intractable due to the set integral involved with the RFS formulation. Instead, an alternate expression, derived using Bayes theorem, must be used:

$$\omega_k^{[\iota]} = \omega_{k-1}^{[\iota]} p\left(\mathcal{Z}_k|\mathcal{M}_k, \mathbf{x}_{0:k}^{[\iota]}\right) \frac{p\left(\mathcal{M}_k|\mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k}^{[\iota]}\right)}{p\left(\mathcal{M}_k|\mathcal{Z}_{1:k}, \mathbf{x}_{0:k}^{[\iota]}\right)} \tag{14}$$

Here, it can be seen that the set measurement likelihood appears on the right hand side of (14). Also, the RFS $\mathcal{M}_k$ is a free variable that can be chosen arbitrarily since it appears only on the right hand side. This has led to the use of different strategies for solving (14). These include: a) The empty-set strategy when $\mathcal{M}_k = \emptyset$, b) the single-feature strategy when an arbitrary map location is used, $\mathcal{M}_k = \{\mathbf{m}\}$, and c) the multi-feature strategy when multiple map locations are used, $\mathcal{M}_k = \left\{\mathbf{m}^1, \mathbf{m}^2 \dots \mathbf{m}^m\right\}$ [5]. In general, it has been found that choosing $\mathcal{M}_k$ to include the location of all the features estimated to exist in $p\left(\mathcal{M}_k|\mathcal{Z}_{1:k}, \mathbf{x}_{0:k}^{[\iota]}\right)$ produces the best results in terms of estimation error. However, using a large $\mathcal{M}_k$ also increases the computational complexity in calculating the set measurement likelihood. Therefore, it is essential to find a computationally tractable way of calculating the set measurement likelihood.

III. Calculating the Set Measurement Likelihood

Given a set of estimated landmarks and a set of measurements, the first step in calculating the set measurement likelihood is to determine the measurement likelihood of individual landmark to measurement pairings. This is necessary because the set measurement likelihood considers all possible pairs of landmark to measurement correspondences, as well as the possibilities of landmarks being mis-detected, and measurements being clutter. To aid with the mathematical presentation, the following definitions are made:

$$\mathcal{Z}_k^0 \equiv \mathcal{Z}_k \tag{15}$$

$$\mathcal{Z}_k^1 \equiv \left\{\mathcal{Z}_k^0 - \left\{\mathbf{y}^1\right\}\right\}, \quad \mathbf{y}^1 \in \mathcal{Z}_k^0 \tag{16}$$

$$\mathcal{Z}_k^r \equiv \left\{\mathcal{Z}_k^{r-1} - \left\{\mathbf{y}^r\right\}\right\}, \quad \mathbf{y}^r \in \mathcal{Z}_k^{r-1} \tag{17}$$

In the above, $\mathcal{Z}_k^0$ is the set of measurements at time-step k. $\mathcal{Z}_k^1$ is the set of measurement with one measurement taken away from set $\mathcal{Z}_k^0$. Continuing with this pattern, $\mathcal{Z}_k^2$ is the set of measurements with two measurements subtracted from the original set. Also let $p_\kappa\left(\right)$ represent the likelihood of measurements being clutter. Using these definitions, the set measurement likelihood (8) can be expressed as in (18).

In the last line of (18), the variable r represents the number of measurement to landmark pairings. The upper limit of r cannot exceed the number of measurements, n, nor the number of landmark estimates, m. For a given number of pairings (i.e., a given value of r), all permutations of measurement to landmark estimate pairings are considered. Unpaired measurements give the clutter factor, $p_\kappa\left(\mathcal{Z}_k^r\right)$. Paired couples (from $i = \{1 \dots r\}$)) provide the probability of detection and the single-landmark measurement likelihood factors, $P_D\left(\mathbf{m}^i\right) p\left(\mathbf{y}^i|\mathbf{m}^i, \mathbf{x}_{0:k}\right)$. Lastly, unpaired landmark estimates give the mis-detection factors, $\left(1 - P_D\left(\mathbf{m}^i\right)\right)$.

A. The Brute-Force Approach

The most computationally expensive method of calculating the set measurement likelihood is to iterate through every possible combination of landmark to measurement correspondences (i.e., calculate every term in (18). Computationally, this is problematic when the number of landmarks and measurements is large, as the complexity of this brute-force method is $\mathcal{O}((|\mathcal{M}| + |\mathcal{Z}|)!)$. Therefore, the brute-force approach quickly becomes impractical when $|\mathcal{M}| + |\mathcal{Z}|$ exceeds 8 to 10. With many robotic applications and sensors, the number of landmarks and measurements considered exceeds this limitation. Hence, it is essential to find other computationally efficient methods for calculating and approximating the set measurement likelihood.

In [5], the evaluation of the set measurement likelihood was performed by assuming that the probability of detection of landmarks is close to 1 while the clutter intensity is 0. However, this method becomes increasingly inaccurate when the detection statistics deviate from these assumptions. Furthermore, the technique in [5] still retains a factorial complexity. The techniques presented next makes no such assumptions on the detection statistics, and yet are able to calculate the set measurement likelihood in a computationally efficient manner.

B. Landmark and Measurement Grouping

In general, the measurement likelihood between any landmark and measurement is non-zero. For practical purposes, small likelihood values (from landmarks and measurements that are spatially well-separated) below a certain threshold can

$$
\begin{aligned}
&p\left(\mathcal{Z}_k|\mathcal{M}_k,\mathbf{x}_{0:k}\right)=p\left(\mathcal{Z}_k|\left\{\mathbf{m}^1,\mathbf{m}^2,\ldots,\mathbf{m}^m\right\},\mathbf{x}_{0:k}\right)\\
=&p_\kappa\left(\mathcal{Z}_k\right)\prod_{i=1}^{m}\left(1-P_D\left(\mathbf{m}^i\right)\right)+\sum_{\mathbf{y}^1\in\mathcal{Z}_k^0}\left(p_\kappa\left(\mathcal{Z}_k^1\right)\prod_{i=1}^{1}\left(P_D\left(\mathbf{m}^i\right)p\left(\mathbf{y}^1|\mathbf{m}^i,\mathbf{x}_{0:k}\right)\right)\prod_{i=2}^{m}\left(1-P_D\left(\mathbf{m}^i\right)\right)\right)+\ldots+\\
&\sum_{\mathbf{y}^1\in\mathcal{Z}_k^0}\sum_{\mathbf{y}^2\in\mathcal{Z}_k^1}\cdots\sum_{\mathbf{y}^r\in\mathcal{Z}_k^{r-1}}\left(p_\kappa\left(\mathcal{Z}_k^r\right)\prod_{i=1}^{r}\left(P_D\left(\mathbf{m}^i\right)p\left(\mathbf{y}^i|\mathbf{m}^i,\mathbf{x}_{0:k}\right)\right)\prod_{i=r+1}^{m}\left(1-P_D\left(\mathbf{m}^i\right)\right)\right)+\ldots\\
=&\sum_{r=0}^{\min(m,n)}\left\{\sum_{\mathbf{y}^1\in\mathcal{Z}_k^0}\sum_{\mathbf{y}^2\in\mathcal{Z}_k^1}\cdots\sum_{\mathbf{y}^r\in\mathcal{Z}_k^{r-1}}\left(p_\kappa\left(\mathcal{Z}_k^r\right)\prod_{i=1}^{r}\left(P_D\left(\mathbf{m}^i\right)p\left(\mathbf{y}^i|\mathbf{m}^i,\mathbf{x}_{0:k}\right)\right)\prod_{i=r+1}^{m}\left(1-P_D\left(\mathbf{m}^i\right)\right)\right)\right\}
\end{aligned}
\tag{18}
$$

be assumed to be 0. This allows landmarks and measurements to be grouped in a way such that landmarks from one group have zero likelihoods with measurements from any other group (or conversely, the measurements from one group have zero likelihoods with the landmarks from all other groups). This grouping method is similar to techniques used in multiple-target tracking for determining likely data association hypotheses [11]. However, the evaluation of the set measurement likelihood requires not only the best hypotheses, but rather the likelihood of all hypotheses.

Numerous methods can be used for the grouping process. One possible method is to represent landmarks and measurements as nodes in a graph, where an edge exists between nodes with non-zero measurement likelihoods (after thresholding). A connected-component analysis [12] can be performed to identify the landmark-measurement groups. The complexity of this method is linear in the number of nodes and edges. Therefore, in the worse case scenario where all nodes are connected, the complexity is at most $\mathcal{O}((|\mathcal{M}|+|\mathcal{Z}|)^2)$.

Once the original likelihood table has been broken down into smaller groups, it is possible to then find the overall measurement likelihood by finding the set measurement likelihood of each group, and then finding their product. Next, two methods will be presented on how the set measurement likelihood can be found for individual groups.

C. Lexicographical Iteration

If the size of a group is small (e.g., less than 8 landmarks and measurements), the brute-force approach can be used to determine the set measurement likelihood of the group, while constraining the computational complexity to be constant. Lexicographical ordering [13] can be used to iterate through all permutations of landmark to measurement correspondences. The set measurement likelihood of the group can be calculated by summing the joint likelihood of all the permutations.

As an example, consider the case of 2 landmarks and 3 measurements. The notation $[a_1, a_2|a_3, a_4, a_5]$ will be used to represent a permutation, where the first two entries represent the measurement associated to the landmarks. The remaining three entries are for non-associated measurements. The value of a_i can range from 1 to 4, with 1 to 3 corresponding to an actual measurement, while 4 represents no measurement. The first three permutations in the lexicographical sequence are: $[1, 2|3, 4, 4]$, $[1, 2|4, 3, 4]$, $[1, 2|4, 4, 3]$.

In the first sequence, the first landmark corresponds with measurement 1 and the second landmark corresponds with measurement 2, while measurement 3 is considered clutter. The next two permutations, although being different than the first, actually represents the same correspondences as the first permutation, where measurement 3 is considered clutter. For calculating the set measurement likelihood, it is important to avoid these double-counting instances in the lexicographical sequence. This can be accomplished by always reversing the last n elements of each permutation. In the above example, the first permutation should become $[1, 2|4, 4, 3]$. From this, the next permutation in the sequence should be $[1, 3|2, 4, 4]$, but by reversing the last 3 elements, this should become $[1, 3|4, 4, 2]$. This indicates that measurements 1 and 3 are associated with the landmarks, while measurement 2 is considered clutter. Following this, the third permutation should then be $[1, 4|4, 3, 2]$, where landmark 2 is mis-detected, while measurements 2 and 3 are considered clutter. Since lexicographical iteration is only practical for groups with a small number of landmarks and measurements, another method is required to handle larger groups.

D. Modified Murty's Algorithm

Murty's algorithm [14] can be used to sequentially determine permutations of correspondences from the highest to lowest joint measurement likelihoods. The algorithm works by iteratively partitioning the permutation with the highest likelihood, and then finding the highest likelihood from the new partitions. Assuming that the current highest-likelihood permutation is $[a_1, a_2, a_3, a_4, a_5]$, the partitioning generates 4 new permutations with certain constraints, from which the next permutation with the highest weight can be found: $[b_1 \neq a_1, b_2, b_3, b_4, b_5]$, $[a_1, b_2 \neq a_2, b_3, b_4, b_5]$, $[a_1, a_2, b_3 \neq a_3, b_4, b_5]$, $[a_1, a_2, a_3, b_4 \neq a_4, b_5]$. Here, the first partition does not contain the a_1 correspondence, while the second partition contains the a_1 correspondence, but not the a_2 correspondence. The correspondences, b_1, b_2, b_3, b_4, b_5, are determined by looking for the highest likelihood that satisfies the constraints for the respective partition. This can be achieved by using the linear assignment algorithms such as the Hungarian method [15] or the JV algorithm [16].

Similarly to the lexicographical iteration, double counting may occur when different partitions generate permutations that have the same physical interpretation. For example, assume that there are 2 landmarks and 3 measurements, with the highest-likelihood permutation sequence $[1, 2|3, 4, 4]$. The third partition of this permutation sequence is: $[b_1 = 1, b_2 = 2|b_3 \neq 3, b_4, b_5]$ Using a linear assignment algorithm to find b_4 and b_5, the resulting sequence will end up being $[1, 2|4, 3, 4]$, or $[1, 2|4, 4, 3]$, where measurement 3 is still considered clutter in all cases. To avoid this, Murty's algorithm can be modified to not create partitions for cases where a measurement can only be considered as clutter.

Using this modified Murty's algorithm with complexity $\mathcal{O}((|\mathcal{M}| + |\mathcal{Z}|)^4)$, the set measurement likelihood can be approximated by summing the joint likelihood of successive permutations with decreasing likelihoods, until the change in the sum becomes insignificant, or after the i-best permutations have been summed. For example, the set measurement likelihood for a group can be approximated by stopping the modified Murty's algorithm after the change in the joint likelihood sum is less than 0.1%, or when the number of permutations returned by Murty's algorithm exceeds 200.

In summary, the set measurement likelihood can be calculated by dividing the set of landmarks and measurements into spatially well-separated groups, and by taking the product of the set measurement likelihood of all the groups. The set measurement likelihood for individual groups can be determined using lexicographical iteration if the group has a small number of landmarks and measurements. Otherwise, the modified Murty's algorithm can be used to approximate the set measurement likelihood.

IV. Simulations

2D simulations using the RB-PHD-SLAM algorithm were used as a first validation of the proposed method for evaluating set measurement likelihoods. Many simulation trials have been performed, but the specific results from one particular simulation run will be shown. The results presented here are representative of the results observed from all the other simulation trials.

Recall from (14) that the set measurement likelihood appears in the importance weighting step of the algorithm, where an arbitrary map is selected. For the proposed approach, the map set is selected from the GM PHD as the means of Gaussians with weights higher than 0.75.

The use of a simulation allows the detection and clutter statistics to be controlled. For the particular simulation trail that will be examined, the robot trajectory starts at $(0, 0)$, and ends near $(35, 80)$. The robot has a simulated range-bearing sensor and landmarks between the sensing distances of 5m to 25m in any direction may be detected with a probability of detection of 0.5. Independent of the real detections, false measurements are added with the count being Poisson distributed, and uniformly distributed over the measurement space, with an intensity of $0.005\mathrm{m}^{-2}$. With a sensing area of $1885\mathrm{m}^2$, the expected number of false measurements per time-step is 9.45 within the sensing area.

Before showing the results of the proposed method of set measurement likelihood evaluation, it is of interest to show the simulation results of using FastSLAM [9], which can be viewed as the vector-based counterpart of the RB-PHD-SLAM algorithm. Under the conditions specified above, the estimate results of using FastSLAM are shown in Fig. 2, where the ground-truth robot trajectory is shown as a dashed line, and landmark positions are shown as points. A binary Bayes filter was used alongside the FastSLAM algorithm to estimate the probability of existence of landmarks, which are represented by the blue ellipses. Darker-filled ellipses have a high probability of existence, and lighter-filled ones correspond with lower probabilities of existence. It can be observed that there are large errors in both the robot trajectory and landmark estimates due to the presence of measurement clutter. This motivates the use of RFS-based algorithms such as RB-PHD-SLAM, which takes the clutter statistics into account when calculating state estimates.

The estimate produced by RB-PHD-SLAM using the proposed set measurement likelihood evaluation method is shown in Fig. 3. The shading on the landmark estimate ellipses correspond with the weight of a estimate in the Gaussian mixture map. This is analogous to the probability of existence, and darker-filled ellipses again correspond with landmarks with higher probabilities of existence. Visually, the robot trajectory and landmark position estimates appear to be consistent. To further validate the proposed method, the results of the proposed method are compared against the results from the brute-force method introduced in [5]. Due to computational limitations and the factorial complexity of the brute force approach, a map set with the means from the 15 Gaussians with the highest weights in the GM PHD are chosen for evaluating (14). The estimate produced from the brute-force approach is shown in Fig. 4, which is similar to the results in Fig. 3.

Fig. 5 compares the robot trajectory error of the proposed and brute-force approaches. In both cases, the errors have the same order of magnitude, with the proposed method performing slightly better. This is likely due to the ability of the proposed method to select a larger map set for (14), and also not approximating probabilities of detection as 1, and clutter intensity as 0.

Both methods perform similarly for the map estimate. Fig. 6 compares the proposed and brute-force approaches using the optimal sub-pattern assignment (OSPA) metric [17], which accounts for both spatial and cardinality errors for the map. For both approaches, the order of magnitude of the error is approximately the same. Examining Fig. 7, both the proposed and brute-force methods also perform approximately the same in estimating the number of landmarks.

Based on these results, it is evident that the proposed evaluation of the set measurement likelihood indeed yields consistent RB-PHD-SLAM estimates. Although the brute-

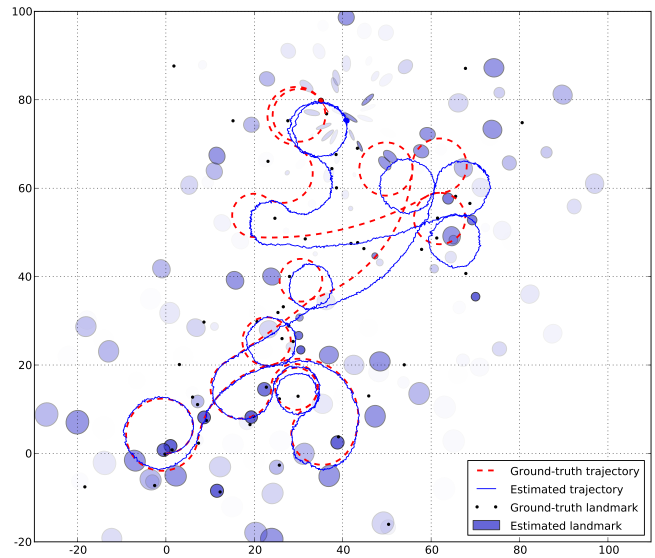


Fig. 2: The estimate produced by FastSLAM.

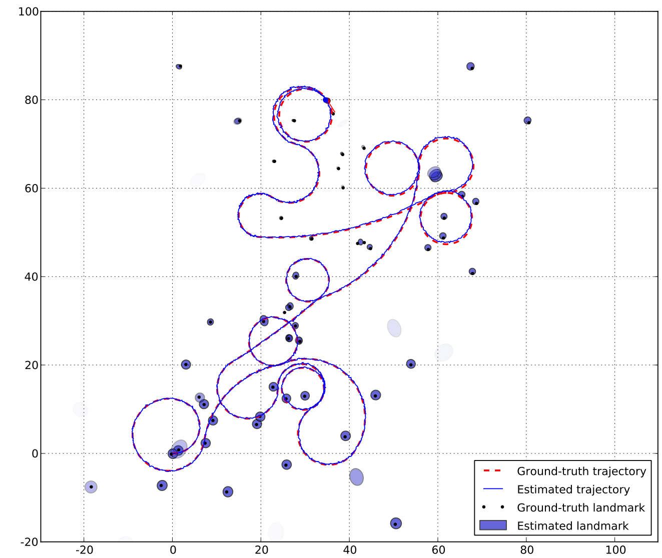


Fig. 4: The estimate produced by RB-PHD-SLAM using the brute-force method of set measurement likelihood evaluation.

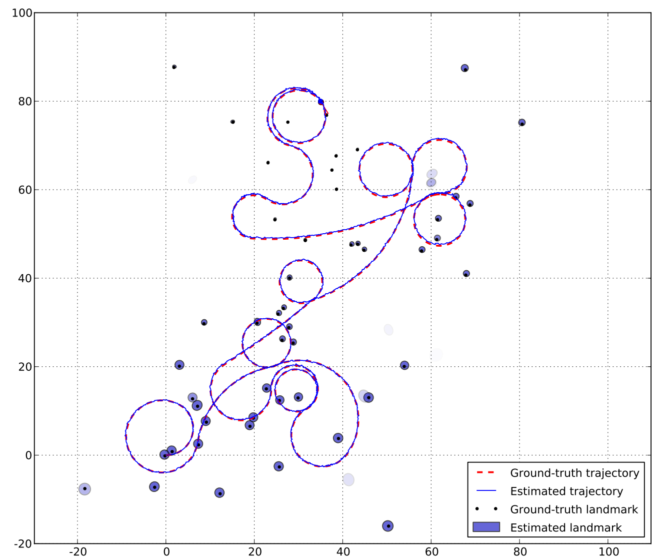


Fig. 3: The estimate produced by RB-PHD-SLAM using the proposed method of set measurement likelihood evaluation.

force approach is also able to produce similar results, it is important to make note of the computational time. From the simulation trials that have been performed using the same groundtruth trajectory and landmark positions as in Fig. 3, the average total computation time of the particle weighting step is 28.27 seconds. In comparison, the brute-force approach averaged at 5562.91 seconds. For reference, the simulation spans 300 seconds. These timing results were obtained by running a C++ implementation of the RB-PHD-SLAM algorithm on a single computer core of an Intel i7 2.4 GHz CPU. From simulation trials with other trajectory and landmark configurations, the speed-up gained by the proposed set measurement likelihood approach is determined to be at least 150 times. This improvement in computational speed makes it more viable for the RB-PHD-SLAM to run in real-time. Next, validation using hardware experimental data will be presented.

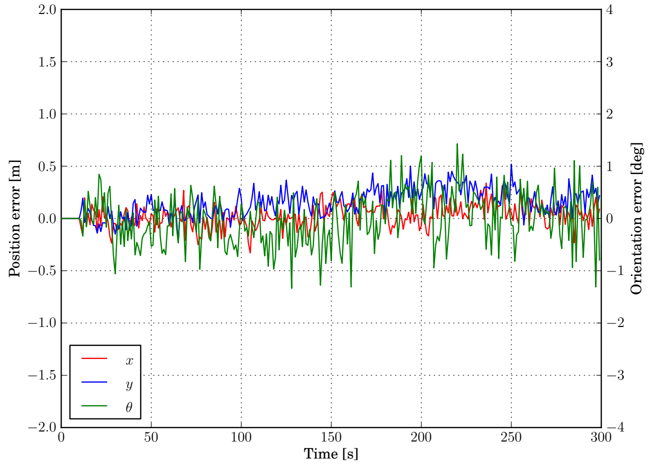
V. Experiments

The robotic platform shown in Fig. 1 was used to collect a dataset in an outdoor park environment to provide further validation of the proposed set measurement likelihood evaluation method. The area in which the dataset was collected is approximately 120m $\times$ 120m, as shown in Fig. 8. There are two curved dirt paths in the area. For the experiment, the robot traversed an approximate figure-eight path once and returned to its starting position $(0, 0)$.

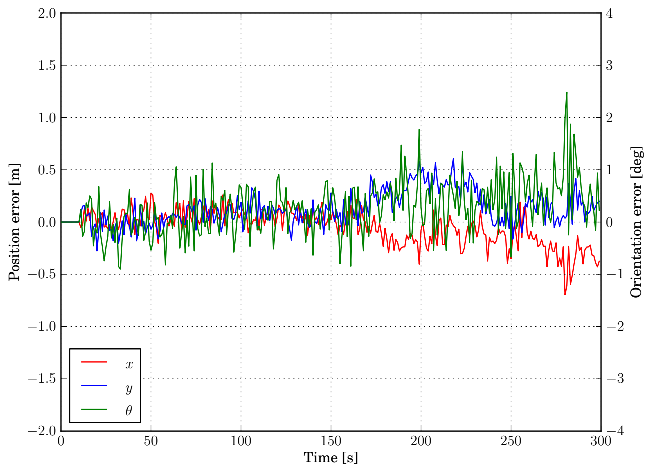
A SICK LD-LRS-1000 scanning lidar mounted on the robot provided planar 2-D scans of the environment at 7.5Hz, with a resolution of 0.5 degrees and a maximum range of 250m. In processing the data, a feature extractor searched for close-to-circular objects (such as tree trunks) from each scan. These features were used as measurements in the RB-PHD-SLAM algorithm (i.e., the raw scans were not used directly). Outliers were included due to moving objects (such as people and cars) that were captured in the scans. They were also generated from foliage. Aside from the lidar, the robot has wheel encoders that measured displacements at 10Hz.

To validate the proposed approach, the experimental dataset was processed by running RB-PHD-SLAM. For particle weighting, where it is necessary to evaluate the set measurement likelihood, the algorithm used a maximum of 20 features with weights over 0.75. Note that using this number of features is computationally infeasible for the brute-force approach, which motivates the use of the proposed methods.

The result of running RB-PHD-SLAM on the experimental dataset is shown in Fig. 8. It can be seen that the estimated



(a) Proposed set measurement likelihood evaluation method



(b) The brute-force approach

Fig. 5: Robot trajectory (position and orientation) errors for (a) the proposed set measurement likelihood evaluation method, and (b) the brute-force approach, in RB-PHD-SLAM.

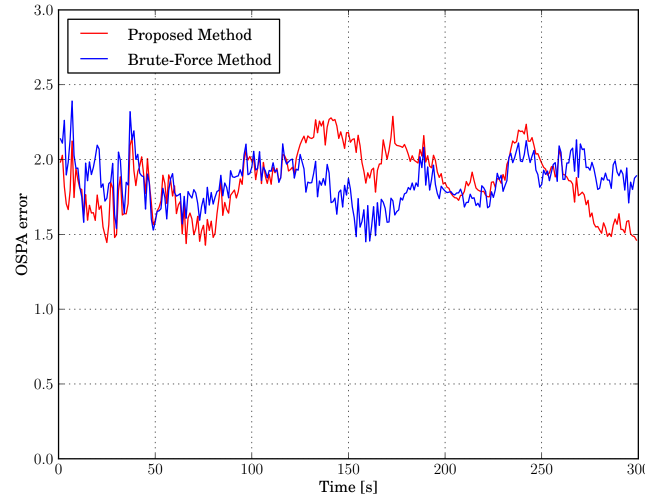


Fig. 6: The OSPA errors of the proposed and brute-force approaches.

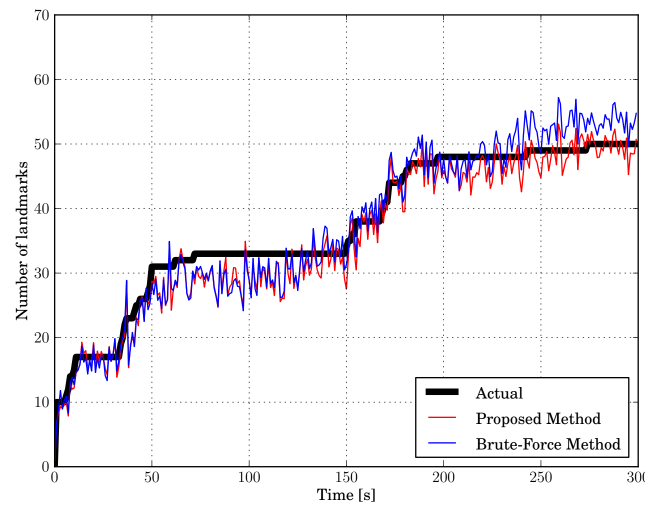


Fig. 7: The estimates on the number of landmarks for the proposed and brute-force approaches.

trajectory of the robot stays on the dirt path and completes the figure-eight traversal. The map generated by RB-PHD-SLAM appears to lack some landmark estimates. This is a known phenomenon caused by uncertainty in the detection statistics. As the robot moves away from a landmark, its probability of detection tends to be overestimated due to the uncertainty in the robot pose and landmark position, as well as the presence of occlusions that may not be modeled. This causes the lowering of the Gaussian weight during landmark estimate updates and leads to the eventual pruning of the Gaussian. Nevertheless, the algorithm is able to produce a consistent robot trajectory estimate by using closer landmarks.

To further motivate the use of the RFS formulation, Fig. 9 shows the results of running the vector-based FastSLAM algorithm [9], the vector-based counterpart of RB-PHD-SLAM. The trajectory produced by FastSLAM appears to be consistent up to the completion of the upper loop of the figure-eight traversal. The presence of outliers then cause the estimate (consisting of particles) to turn left more than it is supposed to. This then causes the estimate for the second half of the traverse and the map to be inconsistent. This inconsistency is due the inability for FastSLAM to handle the level of clutter intensity in the experiment, as well as low probabilities of detection. It is therefore necessary to use the RB-PHD-SLAM algorithm in this case, which requires an efficient method of evaluating set measurement likelihoods.

VI. Conclusions

This paper examined a collection of methods for efficient evaluation of the set measurement likelihood, which appears in RFS formulations of SLAM. Although the RFS formulation provides many advantages over the traditional vector-based formulation, the equations involved in RFS-SLAM are more complex, such as in the calculation of

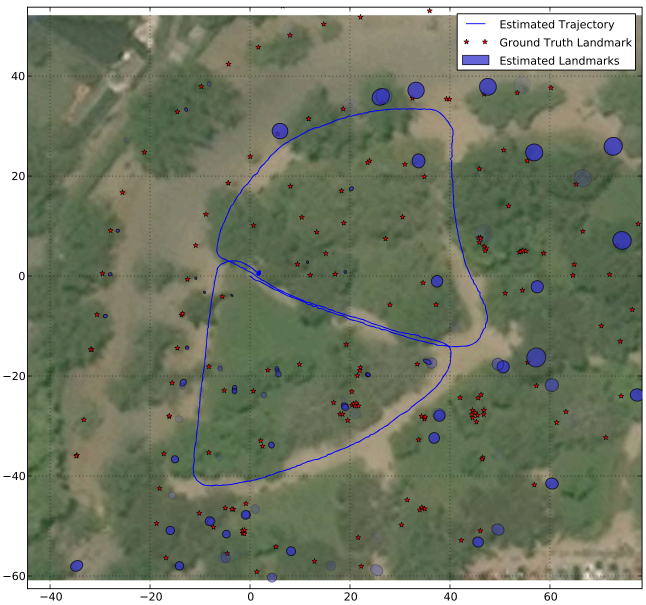


Fig. 8: Result of the RB-PHD-SLAM algorithm on the dataset.

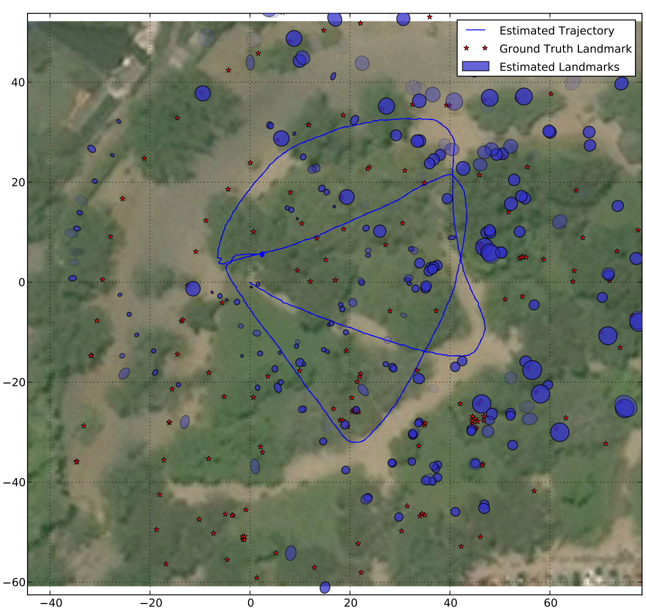


Fig. 9: Result of the FastSLAM algorithm on the dataset.

the set measurement likelihood. The brute-force approach in calculating the set measurement likelihood is computationally intractable, in general. The method proposed in this paper uses a divide-and-conquer method of separating landmarks and measurements into small groups, and then using either lexicographical iteration or a modified version of Murty's algorithm for determining the set measurement likelihood for each group. The overall likelihood can then be calculated by taking the likelihood product of all the groups. The proposed method was tested with the RB-PHD-SLAM algorithm, and was shown to work in a SLAM simulation. It was also validated using real experimental data from a robotic platform. An implementation of the proposed approach and RB-PHD-SLAM can be found in the open-source C++ RFS-SLAM library at https://github.com/kykleung/RFS-SLAM.git.

Acknowledgment

This work was supported by Conicyt DAAD project PCCI2012-009 and Conicyt Fondecyt Project 1110579. The robot used in the experiments is from Clearpath Robotics.

References

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

[2] 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.

[3] 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.

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

[5] 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.

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

[7] Ba-Ngu Vo and Wing-Kin Ma. The Gaussian mixture probability hypothesis density filter. *Signal Processing, IEEE Transactions on*, 54 (11):4091–4104, 2006.

[8] C. S. Lee, D. Clark, and J. Salvi. SLAM with dynamic targets via single-cluster PHD filtering. *Selected Topics in Signal Processing, IEEE Journal of*, 7(3):543–552, 2013.

[9] 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.

[10] 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.

[11] D. Reid. An algorithm for tracking multiple targets. *Automatic Control, IEEE Transactions on*, 24(6):843–854, 1979.

[12] J. Hopcroft and R. Tarjan. Algorithm 447: efficient algorithms for graph manipulation. *Communications of the ACM*, 16(6):372–378, 1973.

[13] S. V. Pemmaraju and S. S. Skiena. *Computational discrete mathematics: combinatorics and graph theory with mathematica*. Cambridge University Press, 2003.

[14] K. G. Murty. An algorithm for ranking all the assignments in order of increasing cost. *Operations Research*, 16(3):682–687, 1968.

[15] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[16] R. Jonker and A. Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340, 1987.

[17] Dominic Schuhmacher, B-T Vo, and B-N Vo. A consistent metric for performance evaluation of multi-object filters. *Signal Processing, IEEE Transactions on*, 56(8):3447–3457, 2008.