

An Improved Weighting Strategy for Rao-Blackwellized Probability Hypothesis Density Simultaneous Localization and Mapping

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

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

Abstract—The use of random finite sets (RFSs) in simultaneous localization and mapping (SLAM) for mobile robots is a new concept that provides several advantages over traditional vector-based approaches. These include: 1) the incorporation of detection statistics, as well as the usual spatial uncertainty, in an estimation algorithm, 2) the ability to estimate the number of landmarks in a map, and 3) the circumvention of the need for data association heuristics. Solutions to SLAM can be obtained through the Rao-Blackwellized Probability Hypothesis Density (RB-PHD) filter, which is an approximation of the Bayes filter for RFSs using both particles to represent the robot trajectories, and Gaussian mixtures to represent their associated maps. This paper proposes an improved multi-feature particle weighting strategy for the RB-PHD filter and shows through simulations that it outperforms existing weighting strategies. The proposed strategy makes the RB-PHD filter a generalization of multi-hypothesis (MH) FastSLAM, a vector-based SLAM solution that uses the RB-particle filter.

I. INTRODUCTION

Simultaneous localization and mapping (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. Stochastic filtering and batch estimation are two general approaches for solving SLAM problems. In both cases, the general practice adopted by the robotics community uses random vectors to represent the robot and map state [1].

Recently, a different representation has been introduced for feature-based maps using random finite sets (RFSs) [2], in which, random vectors for individual features (or landmarks) are placed in a set instead of being concatenated into a single vector (as in the vector-based approach). The cardinality of the set is also a random variable. This approach has been used in multi-target tracking [3], and has since been adapted in both single vehicle [4] and multi-robot mapping problems [5].

The benefits of using a RFS-based filtering approach to SLAM are that: a) Data associations (or the correspondence between measurements and state) do not have to be explicitly determined using heuristics before the application of Bayes theorem; b) RFS-based filtering naturally accounts for detection statistics (i.e., the probability of detection of features and clutter); c) The RFS approach estimates both the spatial position of landmarks, as well as the number of landmarks that have been observed by on-board sensors.

Finite set statistics (FISST) is a set of tools developed by Mahler [6] that use RFSs for handling multi-target estimation problems. It includes the generalization of the Bayes filter for use with RFSs. A computationally tractable approximation of the filter can be achieved by using a first statistical moment approximation of a RFS, which is its probability hypothesis density (PHD) or *intensity*. This approximation of the Bayes filter is called the PHD filter.

In adapting the Gaussian mixture (GM)-PHD filter [7] for SLAM, Mullane et al. [8] introduced the Rao-Blackwellized (RB)-PHD filter, which is based on the RB-particle filter (PF). In this filter, the robot trajectory estimate is represented by particles, while the per-particle map is estimated using the PHD filter. This is similar to the vector-based RB-PF approach known as FastSLAM [9], which uses the Extended Kalman filter (EKF) for map updates.

In any PF [10], it is necessary to assign importance weightings to particles so that they can be re-sampled. However, the evaluation of the weighting term in the RB-PHD can be calculated with an arbitrary set of map features. To the best knowledge of the authors, all implementations of RB-PHD SLAM have only used the *empty-set* or the *single-feature* strategies for importance weighting. The purpose of this paper is to demonstrate the concepts, advantages, and computational consequences of implementing a *multi-feature* strategy, which is more robust to sensor noise and clutter by taking into account the measurement likelihoods to multiple features. This in turn allows the RB-PHD filter to provide better trajectory and map estimates in comparison to the empty-set and single-feature strategies. Furthermore, we show that using the RB-PHD filter with the multi-feature strategy is a generalization of vector-based RB-PF solutions that use single-hypothesis (e.g., FastSLAM [9]) or MH data association (e.g., MH-FastSLAM [11]).

We will begin in section II by providing the formulation and review of RB-PHD SLAM. In section III, we will examine the existing weighting strategies and derive our proposed multi-feature strategy. Furthermore, we will show that it is a generalization of vector-based approaches that use the RB-PF. In section IV, we verify the performance of our proposed approach with simulations and make comparisons with multi-hypotheses FastSLAM.

II. RFS-BASED SLAM FORMULATION USING THE RB-PHD FILTER

A. System Model

SLAM is a state estimation problem in which we seek the best estimate of the robot trajectory and map feature positions over time by using all sensor measurements. In general, we can represent the underlying stochastic system using the non-linear discrete-time equations:

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

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

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^i is the i -th measurement vector at time-step k
- \mathbf{h} is the sensor-specific measurement model,
- \mathbf{m}^j is a random vector for the position of landmark j ,
- $\boldsymbol{\epsilon}_k$ is the measurement noise

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

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

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 measurement. Measurements may also be clutter that are generated from sensor noise appearing with a probability of false alarm, P_F . We define the set of all n measurements at time-step k as:

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

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

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

relative to the initial pose of the robot at every time-step.

B. The RB-PHD Filter

The posterior PDF (5) can be factored into the form:

$$p(\mathbf{x}_{0:k} | \mathcal{Z}_{1:k}, \mathbf{u}_{1:k}) p(\mathcal{M}_k | \mathbf{x}_{0:k}, \mathcal{Z}_{1:k}, \mathbf{u}_{1:k}), \quad (6)$$

This is the same approach taken in [9] 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 we represent using a GM. In the RFS-based approach, we also assume that the map RFS has a *multi-object Poisson distribution*¹. This

¹This implies that features are independently and identically distributed, while the number of features follow a Poisson distribution [6].

allows us to approximate the PDF of the map RFS as an intensity function, v , represented as a GM [7]:

$$v_k(\mathcal{M}_k) = \sum_i w_k^{[i]} \mathcal{N}(\mu_k^{[i]}, \Sigma_k^{[i]}) \quad (7)$$

In contrast to the vector-based RB-PF approach of using EKFs to update the Gaussians for individual landmarks, a PHD filter is used instead to update the map intensity function.

We will provide a brief overview of the main steps in the RB-PHD filter [2, 8]. Our main focus is on the importance weighting step, which we will cover in detail in Section III.

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

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

are propagated forward in time by sampling the motion noise, $\boldsymbol{\delta}_{k-1}^{[i]}$, and using the motion model (1) [12]:

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

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

$$v_k^- (\mathcal{M}_k) = v_{k-1}^+ + \sum_i^{|\mathcal{Z}_{k-1}|} w_B \mathcal{N}(\mu_k^{[i]}, \Sigma_k^{[i]}). \quad (10)$$

These new Gaussians created at time-step k represent potential new landmarks in the map, with mean and covariance, $(\mu_k^{[i]}, \Sigma_k^{[i]})$. These are determined by using the inverse measurement model from equation (2) with the previous updated pose $\mathbf{x}_{k-1}^{[i]}$, and previous measurements, \mathcal{Z}_{k-1} .

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

$$v_k^+ (\mathcal{M}_k) = (1 - P_D) v_k^- + \sum_{i=1}^{|\mathcal{Z}_k|} \sum_{j=1}^{N_k^-} w_k^{i,j} \mathcal{N}(\mu_k^{[i,j]}, \Sigma_k^{[i,j]}) \quad (11)$$

where N_k^- is the number of Gaussians that compose v_k^- . Here the first term is a copy of v_k^- with lowered weights to account for the possibility of missed detections. In this paper, we will assume that the probability of detection, P_D , is constant to simplify the presentation of our equations, but it can be generalized such that P_D is a function of \mathbf{x}_k and a landmark from \mathcal{M}_k . The second term adds a new Gaussian for each pair comprising a new measurement and an existing Gaussian in the intensity map. In other words, instead of determining data association based on heuristics, we let the PHD filter determine how much a measurement should influence a landmark estimate. This is carried out by the weighting factor calculation:

$$w_k^{i,j} = \frac{P_D w_k^j q(\mathbf{z}_k^i, \mathcal{N}(\mu_k^{[j]}, \Sigma_k^{[j]}))}{\kappa(P_F) + \sum_{l=1}^{N_k^-} P_D w_k^l q(\mathbf{z}_k^i, \mathcal{N}(\mu_k^{[l]}, \Sigma_k^{[l]}))} \quad (12)$$

where $q()$ is the measurement likelihood given a feature estimate, and $\kappa(P_F)$ is the clutter density, which is a function of the probability of false alarm. The mean and covariance for each new Gaussian created from measurement i and landmark j , $(\mu_k^{[i,j]}, \Sigma_k^{[i,j]})$, 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*: Particle weighting and re-sampling are necessary to update the trajectory estimates [12]. The method in which this is performed is the focus of this paper, and we will explore this in greater details in section III.

5) *Merging and Pruning of the Map*: Gaussians with small weights are eliminated from the intensity function, while Gaussians that are close to each other are merged together [2, 13]. This approximation is critical in limiting the computational requirement of the RB-PHD filter.

III. IMPORTANCE WEIGHTING

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:

$$p(\mathbf{x}_{0:k} | \mathcal{Z}_{1:k-1}, \mathbf{u}_{1:k}), \quad (13)$$

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

$$p(\mathbf{x}_{0:k} | \mathcal{Z}_{1:k}, \mathbf{u}_{1:k}). \quad (14)$$

Bayes rule allows us to express the weighting distribution in terms of (13) and (14):

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

in which η is a normalizing constant. Since (13) and (14) are sampled using particles, the weighting distribution, which we will define as ω_k , is also sampled such that we calculate a weight for each particle. To solve (15), we can express it as:

$$\begin{aligned} \omega_k &\equiv p(\mathcal{Z}_k | \mathbf{x}_{0:k}, \mathcal{Z}_{1:k-1}) \\ &= \int p(\mathcal{Z}_k | \mathcal{M}_k, \mathbf{x}_{0:k}) p(\mathcal{M}_k | \mathbf{x}_{0:k}, \mathcal{Z}_{1:k-1}, \mathbf{u}_{1:k}) d\mathcal{M}_k \end{aligned} \quad (16)$$

Since the weighting distribution is sampled, we solve (16) by using the trajectory-dependent map for each individual particle. This is relatively easy if we were dealing with Gaussian random vectors. However, with RFSs, the computation of a set integral is computationally infeasible. Therefore, we use an alternate expression for (15) obtained from Bayes theorem:

$$\begin{aligned} \omega_k &\equiv p(\mathcal{Z}_k | \mathbf{x}_{0:k}, \mathcal{Z}_{1:k-1}) \\ &= p(\mathcal{Z}_k | \mathcal{M}_k, \mathbf{x}_{0:k}) \frac{p(\mathcal{M}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k})}{p(\mathcal{M}_k | \mathcal{Z}_{1:k}, \mathbf{x}_{0:k})} \end{aligned} \quad (17)$$

In contrast, (17) is hard to solve with a vector-based approach, but possible with RFSs because we assume that the map RFS is multi-object Poisson distributed. Let m^- and m^+ represent the

sums of all the Gaussian weights in v_k^- and v_k^+ , respectively. The map density terms in (17) can be expressed as:

$$p(\mathcal{M}_k | \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k}) \approx \frac{\prod_{\mathbf{m}_k \in \mathcal{M}_k} v_k^-(\mathbf{m}_k)}{\exp m^-} \quad (18)$$

$$p(\mathcal{M}_k | \mathcal{Z}_{1:k}, \mathbf{x}_{0:k}) \approx \frac{\prod_{\mathbf{m}_k \in \mathcal{M}_k} v_k^+(\mathbf{m}_k)}{\exp m^+} \quad (19)$$

We note from (17) that the choice of the map, \mathcal{M}_k , for which we evaluate the expression in its general form is theoretically arbitrary since the left-hand side of (17) is independent of the map. This has led to the simplest solutions that adopt the *empty-set strategy* and the *single-feature strategy* in determining the particle weight in (17). However, due to the multi-object Poisson approximations shown in (18) and (19), which we must use to calculate particle weights in a feasible manner, different choices of the map set can have significant performance effects of the filter. We will show in Section IV that the *multiple-feature strategy*, to be proposed in section III-C, yields significantly better estimation results. The main difference between the three strategies is the way in which the measurement likelihood is evaluated. Before presenting the multi-feature strategy and understanding how it is derived, it is useful to review the empty-set and single-feature strategies.

A. The Empty-Set Strategy

For the empty-set strategy, $\mathcal{M}_k = \emptyset$. All measurements are considered to be multi-object Poisson distributed clutter, and the likelihood term in (17) becomes:

$$p(\mathcal{Z}_k | \mathcal{M}_k, \mathbf{x}_{0:k}) \approx \frac{\prod_{\mathbf{z}_k \in \mathcal{Z}_k} c(\mathbf{z}_k | \mathbf{x}_k)}{\exp \int v(\mathbf{z} | \mathbf{x}_k) d\mathbf{z}} \quad (20)$$

Note that in many applications, clutter is assumed to be uniformly distributed. For presentation purposes in the rest of this paper, and without loss of generality, let

$$c = c(\mathbf{z}_k | \mathbf{x}_k), \quad N_c = \int v(\mathbf{z} | \mathbf{x}_k) d\mathbf{z}. \quad (21)$$

In this strategy, the ratio of map densities in (17) simplifies to

$$\frac{\prod_{\mathbf{m}_k \in \mathcal{M}_k} v_k^-(\mathbf{m}_k)}{\exp m^-} \frac{\exp m^+}{\prod_{\mathbf{m}_k \in \mathcal{M}_k} v_k^-(\mathbf{m}_k)} = \exp(m^+ - m^-). \quad (22)$$

Although easy to implement, this strategy performs the worst out of the three methods, and often causes filter divergence.

B. The Single-Feature Strategy

For the single-feature strategy, $\mathcal{M}_k = \mathbf{m}$. All but one measurement are considered clutter, and the likelihood term in equation (17) (with a constant P_D) becomes:

$$\begin{aligned} p(\mathcal{Z}_k | \mathcal{M}_k, \mathbf{x}_{0:k}) &\approx (1 - P_D) \frac{c^{|\mathcal{Z}_k|}}{\exp N_c} \\ &+ P_D \sum_{\mathbf{z}_k \in \mathcal{Z}_k} \left(\frac{c^{|\mathcal{Z}_k - \mathbf{z}_k|}}{\exp N_c} p(\mathbf{z}_k | \mathbf{m}, \mathbf{x}_{0:k}) \right) \end{aligned} \quad (23)$$

The above expression accounts for both the cases where the single feature is miss-detected or detected. This strategy yields better filter performance in comparison to the empty-set strategy, but still lacks robustness in comparison with the multi-feature strategy which is proposed next.

C. The Multi-Feature Strategy

Our proposed weighting strategy is to select \mathcal{M}_k such that it includes all the estimated landmarks that currently exists in the robot sensor's field of view (FOV). Let us first consider the simplest case by assuming that there are two features, $\mathcal{M}_k = \{\mathbf{m}_1, \mathbf{m}_2\}$. The measurement likelihood term in (17) becomes:

$$\begin{aligned}
p(\mathcal{Z}_k | \mathcal{M}_k, \mathbf{x}_{0:k}) &\approx (1 - P_D)^2 \frac{c^{|\mathcal{Z}_k|}}{\exp N_c} \\
&+ P_D (1 - P_D) \sum_{\mathbf{z}_k \in \mathcal{Z}_k} \left(\frac{c^{|\mathcal{Z}_k - \mathbf{z}_k|}}{\exp N_c} p(\mathbf{z}_k | \mathbf{m}_1, \mathbf{x}_{0:k}) \right) \\
&+ P_D (1 - P_D) \sum_{\mathbf{z}_k \in \mathcal{Z}_k} \left(\frac{c^{|\mathcal{Z}_k - \mathbf{z}_k|}}{\exp N_c} p(\mathbf{z}_k | \mathbf{m}_2, \mathbf{x}_{0:k}) \right) \\
&+ P_D^2 \sum_{\mathbf{z}_1 \in \mathcal{Z}_k} \sum_{\mathbf{z}_2 \in \mathcal{Z}_k - \mathbf{z}_1} \left(\frac{c^{|\mathcal{Z}_k - \{\mathbf{z}_1, \mathbf{z}_2\}|}}{\exp N_c} p(\mathbf{z}_1 | \mathbf{m}_1, \mathbf{x}_{0:k}) \right. \\
&\quad \left. p(\mathbf{z}_2 | \mathbf{m}_2, \mathbf{x}_{0:k}) \right) \quad (24)
\end{aligned}$$

The complexity of (24) increases greatly as we select \mathcal{M}_k to include more landmarks. Approximations need to be made to ensure that the multi-feature strategy is computationally tractable. Our first approximation is valid for sensors in which the amount of clutter is low and the probability of detection is high. This allows us to approximate (24) using its last term (highest order in terms of the probability of detection). The validity of this approximation is shown in Fig. 1 from multiple simulated 2-D environments with 5 landmarks. With this approximation, the measurement likelihood term for an arbitrary number of n features can be expressed as

$$p(\mathcal{Z}_k | \mathcal{M}_k, \mathbf{x}_{0:k}) \approx \sum_{\mathcal{Z}_k, \mathcal{M}_k} s(\mathcal{Z}_k, \mathcal{M}_k, \mathbf{x}_{0:k}), \quad (25)$$

where s represents the likelihood of a possible set of pairings between all the measurements in \mathcal{Z}_k with the landmarks in \mathcal{M}_k . The summation over s implies that we are considering all possible pairings. From the last term in (24), if $|\mathcal{Z}_k| \geq |\mathcal{M}_k|$, some measurements are considered clutter, and s can be evaluated as:

$$\begin{aligned}
s(\mathcal{Z}_k, \mathcal{M}_k, \mathbf{x}_{0:k}) &= \frac{c^{|\mathcal{Z}_k| - |\mathcal{M}_k|}}{\exp N_c} p(\mathbf{z}_1 | \mathbf{m}_1, \mathbf{x}_{0:k}) \times \\
&p(\mathbf{z}_2 | \mathbf{m}_2, \mathbf{x}_{0:k}) \dots p(\mathbf{z}_{|\mathcal{M}_k|} | \mathbf{m}_{|\mathcal{M}_k|}, \mathbf{x}_{0:k}) \quad (26)
\end{aligned}$$

Where:

$$\begin{aligned}
\mathbf{z}_1 &\in \mathcal{Z}_k, & \mathbf{z}_2 &\in \mathcal{Z}_k - \mathbf{z}_1, & \mathbf{z}_i &\in \mathcal{Z}_k - \{\mathbf{z}_j\} \forall j < i \\
\mathbf{m}_1 &\in \mathcal{M}_k, & \mathbf{m}_2 &\in \mathcal{M}_k - \mathbf{m}_1, & \mathbf{m}_i &\in \mathcal{M}_k - \{\mathbf{m}_j\} \forall j < i
\end{aligned}$$

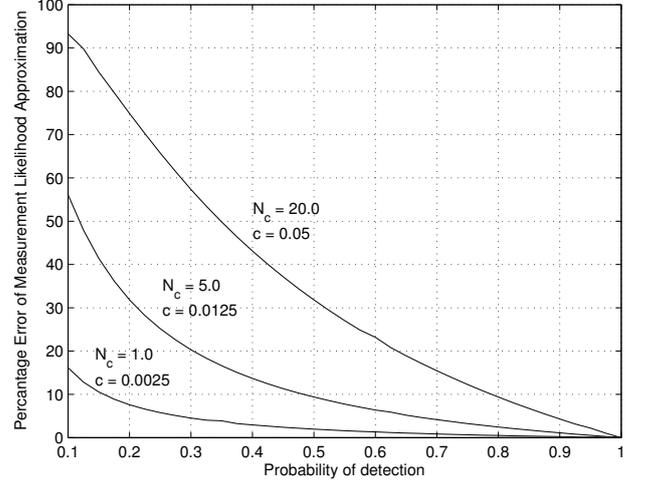


Fig. 1: The percentage error in using the highest-order term for approximating the RFS measurement likelihood.

For the case where $|\mathcal{Z}_k| < |\mathcal{M}_k|$:

$$\begin{aligned}
s(\mathcal{Z}_k, \mathcal{M}_k, \mathbf{x}_{0:k}) &= p(\mathbf{z}_1 | \mathbf{m}_1, \mathbf{x}_{0:k}) \times \\
&p(\mathbf{z}_2 | \mathbf{m}_2, \mathbf{x}_{0:k}) \dots p(\mathbf{z}_{|\mathcal{Z}_k|} | \mathbf{m}_{|\mathcal{Z}_k|}, \mathbf{x}_{0:k}) \quad (27)
\end{aligned}$$

At this point, the computation of (25) is still computationally intensive (of complexity $\mathcal{O}(|\mathcal{M}_k|!)$). Hence, we make a second approximation by setting small values of measurement likelihoods to zero. This effectively reduces the number of likelihood products (i.e., (26) or (27)) that we need to sum up. We can extend this further by selecting spatially well-separated landmarks in \mathcal{M}_k . This reduces that complexity of calculating (25) to $\mathcal{O}(|\mathcal{M}_k| |\mathcal{Z}_k|)$.

D. Relationship to vector-based RB-PF SLAM

It is of interest how our proposed strategy relates to vector-based RB-PF approaches. In the traditional vector-based approaches, we do not use negative information (of not observing a landmark) in map updates. Furthermore, it is assumed that the probability of detection is always equal to one. In performing data association, measurements that do not correspond to any landmarks are not used for updating the map, and a measurement can only update the estimate of the landmark to which it is associated (i.e., the likelihood of the same measurement to any other landmark is zero). If we let $j = c(i)$ represent the data association of measurement i to landmark j , the expression for the measurement likelihood for the vector-based approach is:

$$\begin{aligned}
p(\mathbf{z}_k | \mathbf{m}_k, \mathbf{x}_{0:k}) &= p(\mathbf{z}_k^1 | \mathbf{m}^{c(1)}, \mathbf{x}_{0:k}) \times \\
&p(\mathbf{z}_k^2 | \mathbf{m}^{c(2)}, \mathbf{x}_{0:k}) \dots p(\mathbf{z}_k^n | \mathbf{m}^{c(n)}, \mathbf{x}_{0:k}) \quad (28)
\end{aligned}$$

This is a particular case of (25). To see this, we can extend (24) to n features, remove all the clutter terms, and set the probability of detection to one. From this we will arrive at (25) without needing approximations. Applying data association implies that only a single sequence of measurement to

landmark pairings (i.e., a single term of (27)) will be non-zero, which makes (25) equivalent to (28).

We can see in (16) and (17) that the measurement likelihood terms that appear in (27) and (28) are not the only terms involved when we calculate particle weights. However, the expressions of (16) and (17) are equivalent, and both are functions of the same measurement likelihood.

In the map update step of the RB-PHD filter in (11), having perfect data association, perfect detection, and no clutter, will force the weighting term (12) to equal one for measurement and feature pairs that correspond to each other, and equal zero for all other pairings. From this and the observations above, we can conclude that the RB-PHD filter is a generalization of the vector-based RB-PF approach used in FastSLAM.

The generalization that we have shown also covers MH-FastSLAM [11], in which particles are replicated for each possible and likely data association. This is similar to the second approximation that we make on (25), by setting low measurement likelihoods to zero. However, instead of replicating particles in the RB-PHD filter, we sum the likelihoods of all possible and likely correspondences. This result is proportional to the sum of all the replicated particles in the multi-hypothesis RB-PF filter.

IV. SIMULATIONS

A. Setup

In these initial 1-D experiments (which allow us to use linear system models and not be influenced by linearization errors), we performed SLAM simulations and compared the estimates produced by the RB-PHD filter using the different importance weighting strategies. We also implemented multi-hypothesis (MH)-FastSLAM [11] for comparison. The robot's motion prediction is corrupted by zero-mean Gaussian noise on its odometry measurements. The robot also has a limited sensing range of $5m$. There are 10 landmarks that enter the FOV of the robot sensor during its motion. All measurements are also corrupted by zero-mean Gaussian spatial noise.

In the RB-PHD filter, 25 particles were used in each simulation trial. Re-sampling of particles was set to occur when the effective number of particles [12] falls below 6.25. In the multi-feature particle weighting strategy, the map set is chosen using the peaks of all the Gaussians in the map that are within the sensor's range. Measurement likelihoods that fall below 0.01 are considered zero. For the single-feature strategy, the one-element map set is selected using the Gaussian with the highest weight that is within the sensor's range.

For implementing MH-FastSLAM for comparison, we allowed each particle to spawn at most 4 copies of itself using data associations with highest likelihoods. Furthermore, we re-sampled whenever the number of particles exceeded 125, to prevent the number of particles from growing without bound. To make a fair comparison, we implemented a binary Bayes filter for tracking the probability of existence of landmarks for map management purposes. The probability of detection and false alarm are kept the same as the RB-PHD filter. We also assumed that the prior probability of existence for a landmark,

$P_E(\mathbf{m}^i)$, is 0.5 when it is first observed. Let the posterior log-odds for landmark existence be defined as:

$$l_k \equiv \log \frac{P_E(\mathbf{m}^i | \mathcal{Z}_{1:k})}{1 - P_E(\mathbf{m}^i | \mathcal{Z}_{1:k})} \quad (29)$$

This is recursively updated at time-step k for landmarks that are expected to be in the sensor's FOV:

$$l_k = \log \frac{P_E(\mathbf{m}^i | \mathcal{Z}_k)}{1 - P_E(\mathbf{m}^i | \mathcal{Z}_k)} + l_{k-1} \quad (30)$$

where if \mathbf{m}^i is associated with a measurement in \mathcal{Z}_k ,

$$P_E(\mathbf{m}^i | \mathcal{Z}_k) = \frac{(1 - P_D)P_F P_E(\mathbf{m}^i) + P_D P_E(\mathbf{m}^i)}{P_F + (1 - P_F)P_D P_E(\mathbf{m}^i)}, \text{ else} \quad (31)$$

$$P_E(\mathbf{m}^i | \mathcal{Z}_k) = \frac{(1 - P_D)P_E(\mathbf{m}^i)}{(1 - P_E(\mathbf{m}^i)) + (1 - P_D)P_E(\mathbf{m}^i)}. \quad (32)$$

B. Results

We ran many simulation trials but will show the results from trials performed at $P_D = 0.95$ for two levels of uniform clutter intensities, $N_c = 1.0$ and $N_c = 2.75$ (see equation (21)). Note that due to the limited sensing range, the robot only senses at most 4 features per time-step, hence both clutter levels are significant. For both clutter settings, Fig. 2 shows the ground-truth trajectories, along with the trajectory estimates obtained by the highest-weight particle for each of the tested methods. The empty-set strategy performs worse than dead-reckoning, and we believe that the poor performance is attributed by the inability of the empty-set strategy to significantly distinguish the weights between particles that fit the measurement information well, and those that do not. In Fig. 3, the multi-feature strategy obtains the lowest estimation errors for both clutter settings. MH-FastSLAM also performs well under the low clutter setting, but it is unable to cope with high clutter. As we shall see, this is due to its inability to produce a useful map in high clutter.

In Fig. 4, the maps produced by the empty-set strategy remotely resemble the real map. The maps for the single-feature strategy are shifted for both clutter settings. We believe that this is a consequence of basing particle weights on one single landmark. Hence a single clutter measurement can give high weights to particles that are far from the ground-truth position. The multi-feature strategy produced maps that closely resemble the true map, and is more robust towards clutter, as only a small map offset is experienced for the high-clutter case. Lastly, the map produced by MH-FastSLAM under low clutter resembles the true map, but there are spurious landmarks produced by clutter measurements. In the high-clutter case, MH-FastSLAM breaks down and is unable to produce a useful map. Note that the maps shown are the expected a posteriori (EAP) maps produced by the weighted average of the map from all particles. For MH-FastSLAM we used the probability of existence of a Gaussian as its weight.

The feature counts estimated from each approach are shown in Fig. 5. The multi-feature and single-feature strategies perform better than the empty-set strategy in estimating the number of features. Finally, we examine the map estimation error

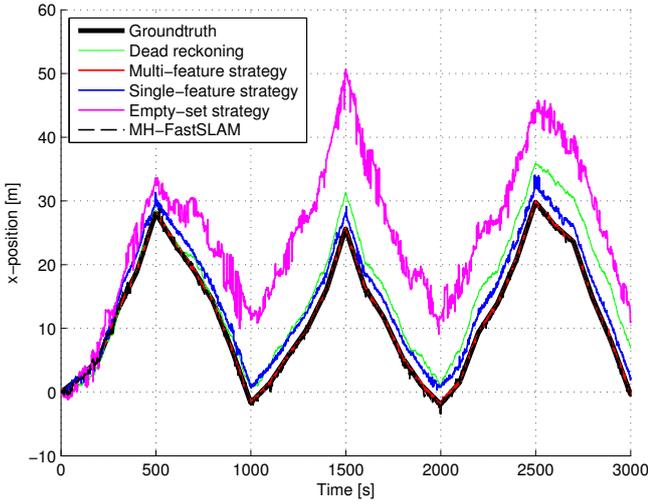
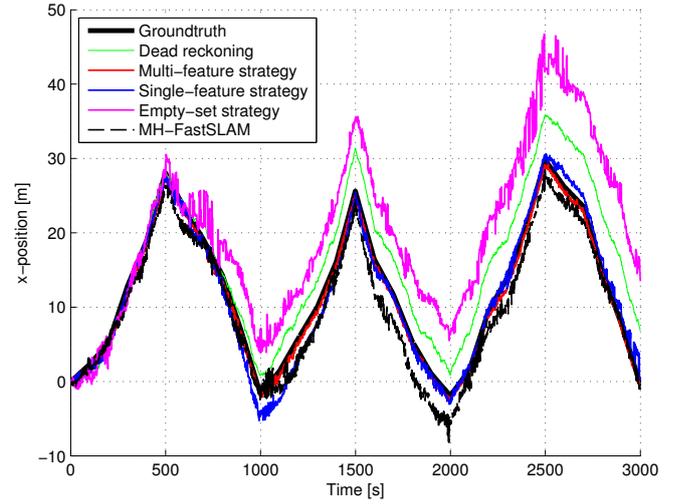
(a) $N_c = 1.00$ (b) $N_c = 2.75$

Fig. 2: Robot trajectory estimates

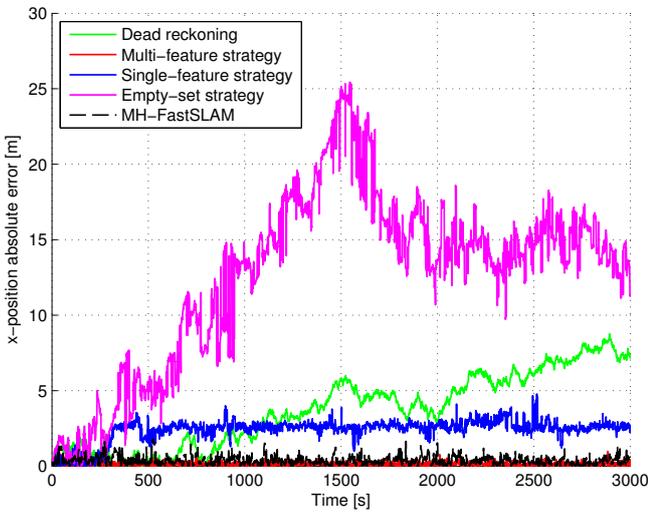
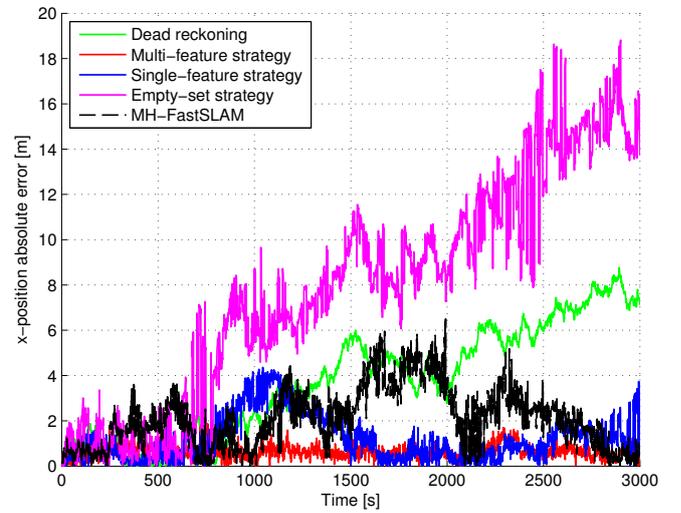
(a) $N_c = 1.00$ (b) $N_c = 2.75$

Fig. 3: Robot trajectory estimate errors

using a metric based on the optimal sub-pattern assignment (OSPA) distance that measures the distance between sets [14]:

$$e = \sqrt{\frac{1}{|\mathcal{M}_g|} \left(\min_{j \in \{1, \dots, |\mathcal{M}_g|\}} \sum_{i=1}^{|\mathcal{M}_k|} d_{i,j}^2 + c^2 \left| |\mathcal{M}_g| - |\mathcal{M}_k| \right| \right)} \quad (33)$$

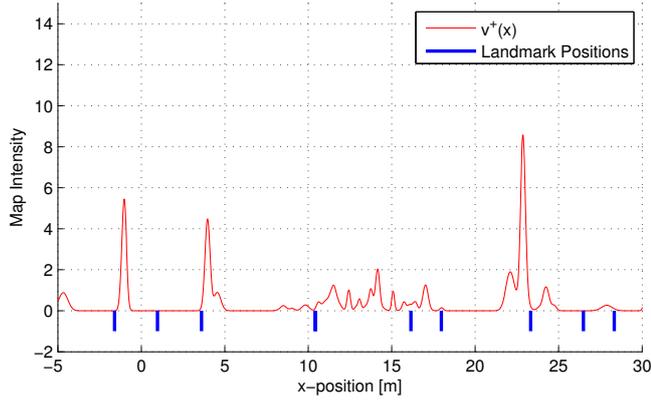
where, \mathcal{M}_k is the estimated map set, \mathcal{M}_g is the ground-truth map set, $d_{i,j}$ is the Euclidean distance between landmark \mathbf{m}^i and \mathbf{m}^j , and c is the cut-off distance parameter² set at a value of 3. To obtain \mathcal{M}_k from a GM, we selected the means of all the Gaussians that have a weight higher than 0.5 and rounded their weights to the nearest integer to determine the number of features presented by each Gaussian. The error, e , increases as the spatial estimate of landmarks increase, and as the landmark count error increases. The results in Fig. 6 show that overall,

² c weighs feature count error against spatial error in the map estimate.

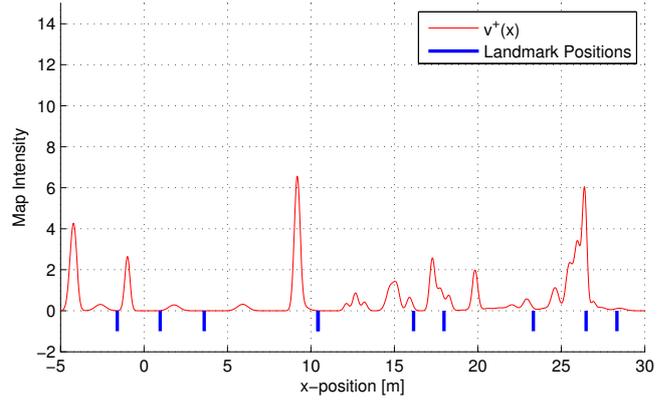
the multi-feature strategy achieves a lower map error compared to the other methods. Although not shown, we have observed that the multi-feature strategy performs better than the other methods when P_D is decreased, and only begins to experience significant map errors when P_D is less than 0.7 for the low clutter case, and 0.85 for the high clutter case. In comparison, the other methods start failing at a higher P_D .

V. CONCLUSIONS

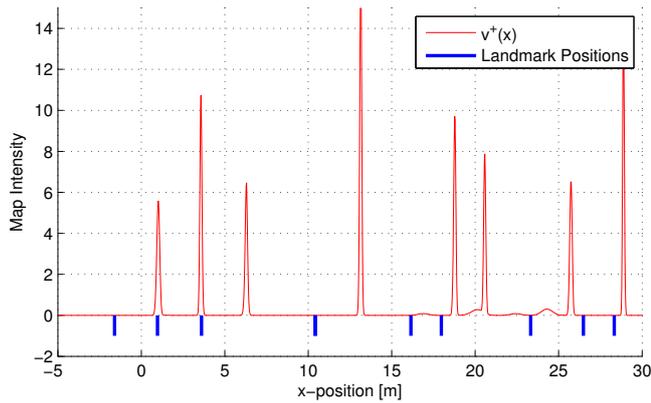
In this paper, we examined an approach to RFS SLAM that approximates the Bayes filter with the RB-PHD filter. We proposed an improved weighting strategy for this filter, which in simulation, proves to be more robust and performs better in providing more accurate estimates overall. We also examined the mathematics behind the proposed strategy, and discovered that it is a generalization of FastSLAM and MH-FastSLAM. As a continuation of this work, we are currently examining ways of improving the computational speed for the proposed weighting method, and working on evaluating our approach



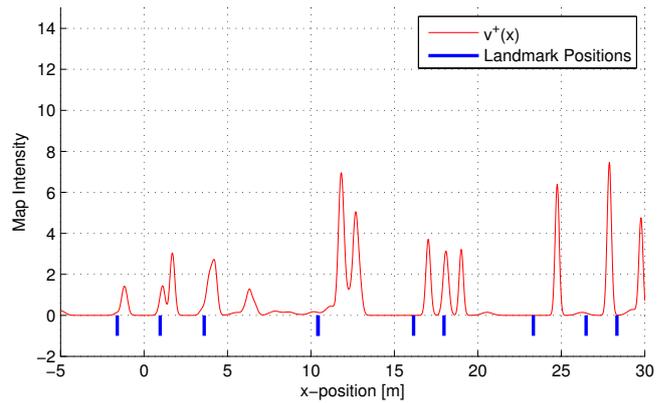
(a) Empty-set strategy map, $N_c = 1.00$



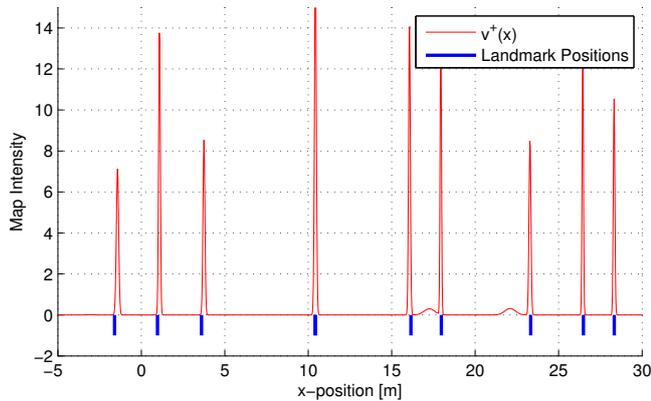
(b) Empty-set strategy map, $N_c = 2.75$



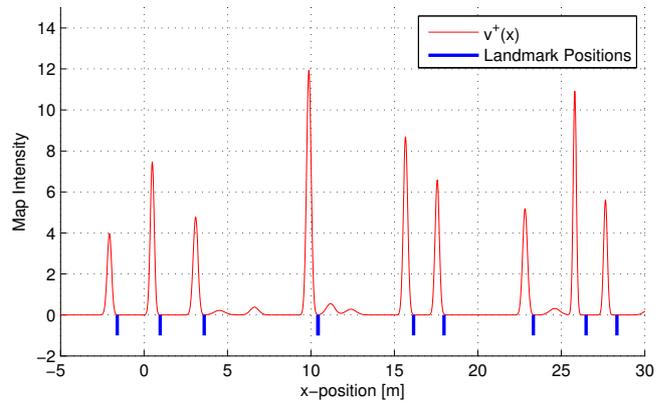
(c) Single-feature strategy map, $N_c = 1.00$



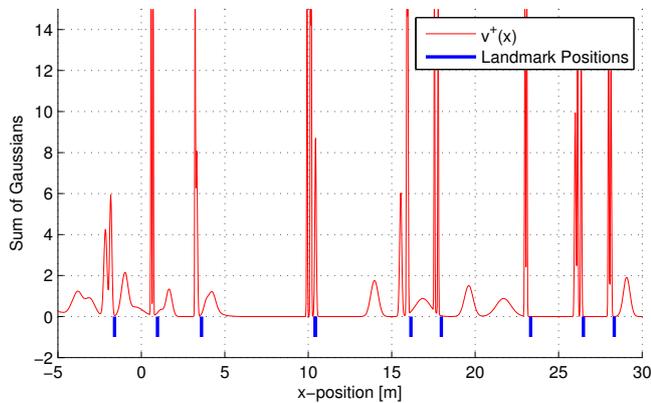
(d) Single-feature strategy map, $N_c = 2.75$



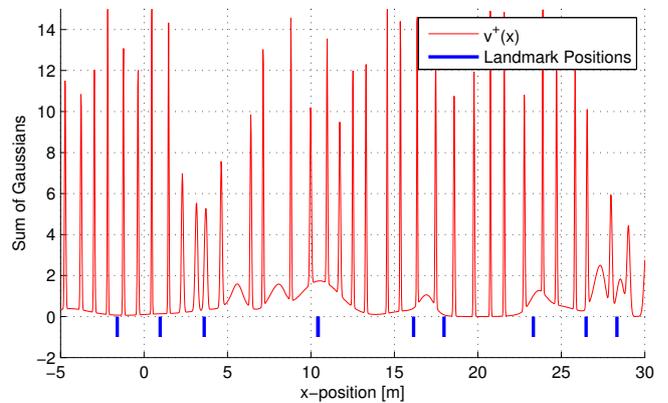
(e) Multi-feature strategy map, $N_c = 1.00$



(f) Multi-feature strategy map, $N_c = 2.75$



(g) MH-FastSLAM map, $N_c = 1.00$



(h) MH-FastSLAM map, $N_c = 2.75$

Fig. 4: Map intensities produced by each weighting strategy and by MH-FastSLAM. There are two landmarks near $x = 10$.

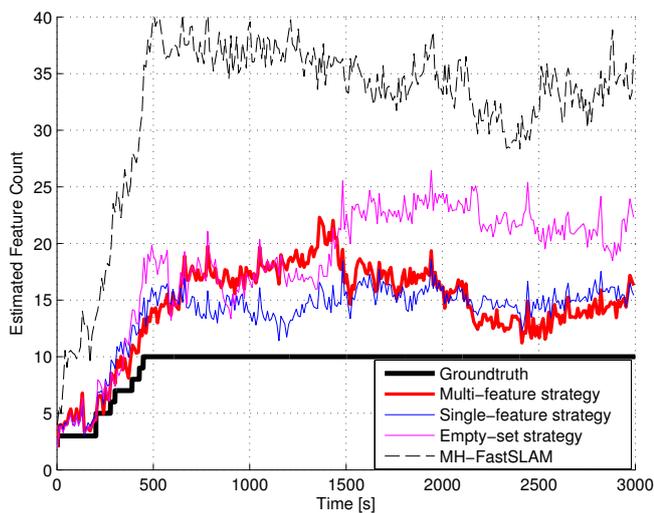
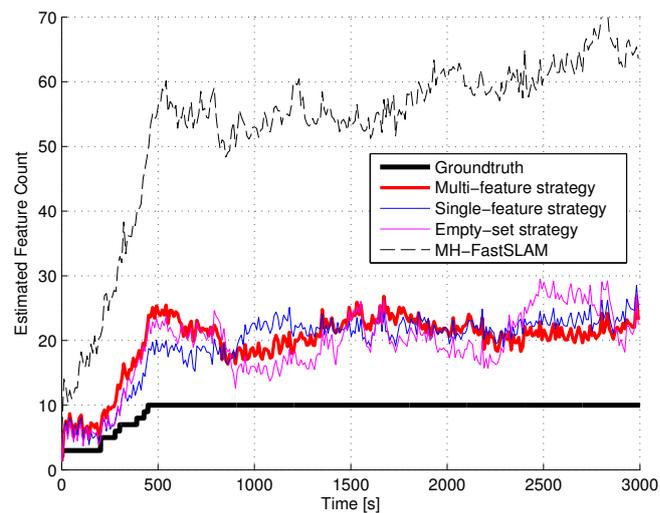
(a) $N_c = 1.00$ (b) $N_c = 2.75$

Fig. 5: Feature count estimates

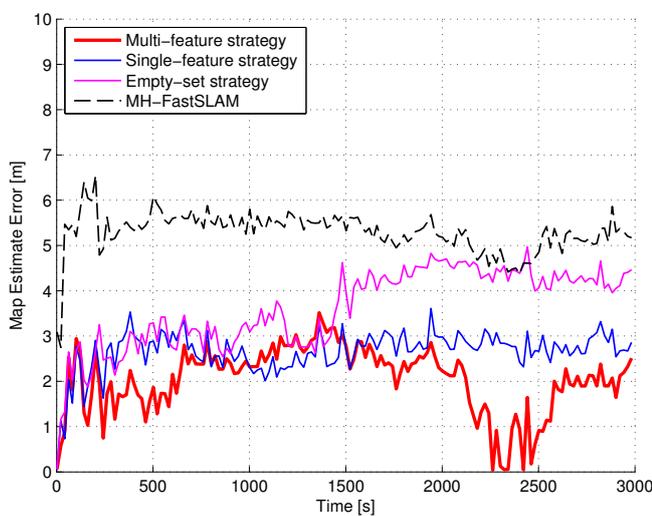
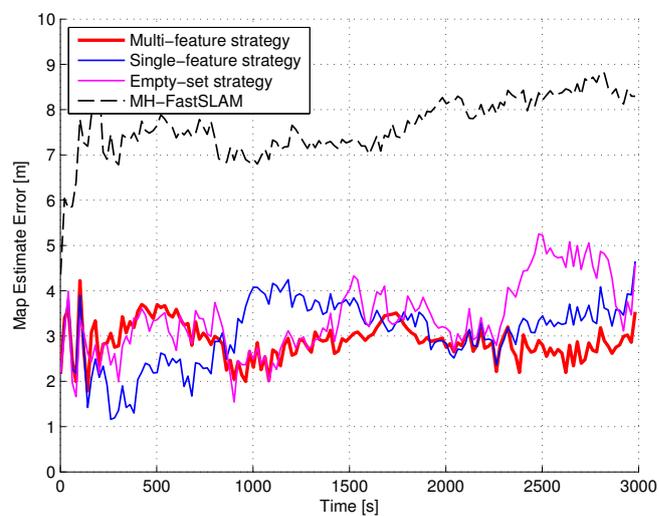
(a) $N_c = 1.00$ (b) $N_c = 2.75$

Fig. 6: Map errors based on the OSPA distance metric

with experimental data from a real robot.

REFERENCES

- [1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 2, pp. 99–110, 2006.
- [2] 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*, vol. 72, 2011.
- [3] D. Clark and S. Godsill, "Group target tracking with the gaussian mixture probability hypothesis density filter," in *Intelligent Sensors, Sensor Networks and Information, International Conference on*. IEEE, 2007, pp. 149–154.
- [4] C. Lundquist, L. Hammarstrand, and F. Gustafsson, "Road intensity based mapping using radar measurements with a probability hypothesis density filter," *Signal Processing, IEEE Transactions on*, vol. 59, no. 4, pp. 1397–1408, 2011.
- [5] D. Moratuwage, B.-N. Vo, and D. Wang, "A hierarchical approach to the multi-vehicle slam problem," in *International Conference on Information Fusion*. IEEE, 2012, pp. 1119–1125.
- [6] R. P. S. Mahler, *Statistical multisource-multitarget information fusion*. Artech House Boston, 2007, vol. 685.
- [7] B.-N. Vo and W.-K. Ma, "The gaussian mixture probability hypothesis density filter," *Signal Processing, IEEE Transactions on*, vol. 54, no. 11, pp. 4091–4104, 2006.
- [8] J. Mullane, B.-N. Vo, M. D. Adams, and B.-T. Vo, "A random-finite-set approach to bayesian slam," *Robotics, IEEE Transactions on*, vol. 27, no. 2, pp. 268–282, 2011.
- [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*, 2002, pp. 593–598.
- [10] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," in *IEE Proceedings F (Radar and Signal Processing)*, vol. 140, no. 2, 1993, pp. 107–113.
- [11] J. Nieto, J. Guivant, E. Nebot, and S. Thrun, "Real time data association for fastslam," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1. IEEE, 2003, pp. 412–418.
- [12] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press Cambridge, 2005, vol. 1.
- [13] J. Mullane, B.-N. Vo, and M. D. Adams, "Rao-blackwellised phd slam," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 5410–5416.
- [14] D. Schuhmacher, B.-T. Vo, and B.-N. Vo, "A consistent metric for performance evaluation of multi-object filters," *Signal Processing, IEEE Transactions on*, vol. 56, no. 8, pp. 3447–3457, 2008.