Multifeature-Based Importance Weighting for the PHD SLAM Filter

KEITH Y. K. LEUNG, Member, IEEE
Applanix Corporation
Richmond Hill, Canada

FELIPE INOSTROZA, Student Member, IEEE
MARTIN ADAMS, Senior Member, IEEE
Universidad de Chile
Santiago, Chile

The probability-hypothesis-density simultaneous localization and mapping filter is a random-finite-set estimation method that incorporates the probability-hypothesis-density filter within a Rao–Blackwellized particle filter, and was developed for navigation and mapping problems. However, the filter tends to diverge due to the existing importance-weighting methods used in the Rao–Blackwellized particle filter. This article introduces a new importance-weighting method that drastically improves the robustness of the probability-hypothesis-density simultaneous localization and mapping filter. Performance evaluations are conducted using both simulations and real experimental data sets.

Manuscript received August 10, 2015; revised January 28, 2016; released for publication April 21, 2016.

DOI. No. 10.1109/TAES.2016.150566.

Refereeing of this contribution was handled by D. Clark.

This work was funded by the Advanced Mining Technology Center of the Universidad de Chile, and by Conicyt Fondecyt Projects 3150066 and 1150930, as well as CONICYT-PCHA Doctorado Nacional 2014. The robot used in the experiments is from Clearpath Robotics.

Authors' addresses: K. Y. K. Leung, Applanix Corporation, 85 Leek Crescent, Richmond Hill, Ontario L4B 3B3, Canada; F. Inostroza, M. Adams, Universidad de Chile, Department of Electrical Engineering, Av. Tupper 2007, 837-0451 Santiago, Chile. Corresponding author is K. Y. K. Leung, E-mail: (kleung@applanix.com).

I. INTRODUCTION

Navigation is the process of estimating a vehicle's pose (i.e., position and orientation) through the use of sensor measurements that contain uncertainty. Satellite-based positioning systems provide absolute exteroceptive measurements that allow this task to be performed in a fixed reference frame. Alternatively, a vehicle can utilize its onboard sensors to obtain relative exteroceptive measurements for navigation. Upon initiation, these measurements will be referenced to an arbitrary coordinate system. The task of navigation may include the tracking and localization of objects in the local environment, which can be labeled as targets, map features, or landmarks depending on the application. This type of navigation and mapping problem is known as simultaneous localization and mapping (SLAM). It is a joint state-estimation problem of the dynamic vehicle (or sensor) state and the position of usually static, but possibly dynamic, targets or landmarks. The problem is of great relevance in the aerospace domain, with various applications using different sensing modalities. Some examples include radar-based navigation for flying vehicles and lidar or vision-based navigation for planetary exploration scenarios. This article will examine a random-finite-set (RFS)-based method for solving the SLAM navigation problem.

The common probabilistic approach to navigation uses random vectors to represent the state of the system and estimates the solution through stochastic filtering, or smoothing (i.e., batch estimation via least-squares optimization) [1]. RFSs were introduced to the field of target tracking as an alternative to formulating tracking filters with random vectors [2]. This newer approach allows for multitarget tracking, in which both the position and number of targets are estimated concurrently.[1] The RFS formulation has since been adapted for vehicle navigation, tracking, and mapping problems [3–10].

Numerous RFS filters, such as those presented in [11–14], have been developed since the initial contribution by Mahler [2]. Mullane et al. [3] were among the first to adapt an RFS filter for navigation problems. Specifically, a Rao–Blackwellized particle filter (RB-PF) was designed for estimating the vehicle pose, while a Gaussian-mixture probability-hypothesis-density (GM-PHD) filter [11] was used for target and landmark estimation. One of the problems with this PHD SLAM filter is that it is not robust, and will often lead to estimate divergence due to the strategies used for importance weighting in the PF. A similar SLAM method based on the single-cluster PHD filter developed by Lee et al. [5] showed much improvement over the currently existing approaches. The purpose of this article is to present an improved method of importance weighting that makes the PHD SLAM filter more robust and allows it to consistently provide more

[1] This is compared to single-target tracking, where only the state vector is estimated.

accurate state estimates. Furthermore, this article will examine practical aspects of the proposed method, including implementation and computational requirements. The improved PHD SLAM filter is validated through thousands of simulation trials, as well as two experimental data sets collected using real hardware.

The remainder of this article is organized as follows. In the next section, additional relevant background information will be presented. In Section III, the PHD SLAM formulation will be reviewed along with existing particle-weighting strategies, and the proposed weighting strategy will be presented. Section IV will detail the implementation of the SLAM filter using the various particle-weighting strategies explained in the previous section. The validation of the proposed approach with 2-D simulations is presented in Section V, which also includes the results from real experimental data sets.

II. BACKGROUND INFORMATION

Similar to vector-based filtering methods for single-target tracking, such as the Kalman filter (KF), RFS filtering methods stem from the recursive Bayesian filtering paradigm. Mathematical tools called finite set statistics were developed by Mahler [2] for RFS multitarget estimation.

One of the key differences between a random vector and an RFS filter is the representation of the system state. Mathematically, the size of a random vector is fixed, and the ordering of its elements is important. In contrast, there is no inherent ordering of an RFS's elements, and its cardinality is a random variable. This makes RFS filtering more appropriate for navigational and mapping tasks in environments where the number of targets or landmarks is unknown a priori. Traditionally, single-target tracking filters (such as the extended Kalman filter [EKF]) have been used for solving such problems, but they usually require the use of separate heuristic-based management algorithms for changing the state vector size as new features are assumed to be detected. Note that there are also vector-based non-Bayesian estimators based on Gaussian nonlinear optimization, such as [15–19], which are also suitable for navigation and mapping tasks.

Another distinction of RFS-based filtering is its ability to account for detection statistics (i.e., the probability of detection and the expected amount of clutter or outliers). This provides RFS filters with increased robustness and eliminates the need for filter-independent outlier-rejection methods that are typically used with single-target estimation filters. From their remaining inlier measurements, single-target estimation filters typically rely on data-association routines to determine the correspondences of measurements to targets for their position updates. More advanced methods such as the joint probabilistic data-association filter [20, 21] perform KF updates to landmarks using combined innovations, calculated from the sum of measurement-likelihood-weighted innovations. With RFS-based filters, every target or landmark is updated using all measurements. The updating effect of a measurement on a target or landmark is determined by the the ratio of the measurement likelihood based on that landmark and the sum of the measurement likelihoods to all landmarks.

As with random vectors, the RFS Bayes filter remains computationally intractable in general. One feasible approximation can be achieved by using the first statistical moment of an RFS, which is also known as its PHD or *intensity*, assuming that the RFSs follow a multitarget Poisson distribution.[2] This approximation results in the PHD filter. When a GM is used to represent the PHD, it is known as the GM-PHD filter [11]. A generalization of the PHD filter, known as the cardinalized PHD filter [12], relaxes the Poisson assumption by treating RFSs as independently and identically distributed clusters. This provides better filtering performance at a higher computational cost. Alternatively, the RFS Bayes filter can be assumed to follow a multi-Bernoulli (MB) distribution, where elements within the RFS are independent and Bernoulli-distributed. This leads to the MB filter, which produces a biased cardinality estimate. The cardinality-balanced MB filter was introduced as a remedy [13]. Track labels, or identification indices for objects being estimated, were added to the outputs of the MB filter through postprocessing. To facilitate the inclusion of track labels into the filtering process, the generalized labeled MB (GLMB) filter was introduced [22]. In the same work, the δ-GLMB filter was introduced as a subclass of GLMB filters offering better computational properties for target-tracking applications. The labeled MB filter detailed in [23] can also be derived from the GLMB with specific assumptions that allow uncertainty propagation of a single set of track labels for a reduction in computational cost. This filter was applied to SLAM in [24].

The work presented in this article is focused on the use of the PHD filter. In adapting the filter for navigation and mapping, Mullane et al. [3] introduced the the PHD SLAM filter. This approach utilizes a RB-PF, wherein the vehicle trajectory estimate is represented by particles and per-particle landmark estimates are updated by GM-PHD filters. This is similar to the vector-based factored solution to SLAM (FastSLAM) algorithm [25], which instead uses the EKF for map updates.

Using a PF, it is necessary to assign importance-weighting factors to particles so that the posterior they represent can be updated through resampling [26]. From the RFS formulation and the Poisson assumption made by the PHD filter, there exist, in theory, different methods for evaluating the importance factors for the PHD SLAM filter. In [3, 4], the empty-set and single-feature strategies are introduced. Unfortunately, these methods are not robust and almost always lead to estimate divergence (as

[2] This implies that features are independently and identically distributed, while the number of features follows a Poisson distribution [2].

will be shown in Section V). Another weighting method based on the cluster process, which provides an exact solution to the measurement likelihood set integral, is introduced in the single-cluster (SC) PHD filter [5], which allows the PHD filter to perform more robustly in navigation and mapping tasks.

The contribution of this article is the presentation of an improved method for evaluating particle importance factors in the PHD SLAM filter, known as the multifeature strategy. Through a comprehensive set of simulations and hardware experiments, the performance of the proposed method over the existing methods will be shown. The work presented in this article is a continuation of the work in [27], which shows only the derivation of a double-feature strategy and uses 1-D simulations for validation. The method for approximating the multifeature strategy is a continuation of the work in [28], but this article presents a more in-depth reasoning for the approximations and validates the multifeature strategy with real experimental data. Furthermore, comparisons are made with various existing random-vector and RFS filters.

III. RFS SLAM PROBLEM FORMULATION

For completeness, this section will briefly review the formulation of the navigation and mapping problem using RFSs.

A. RFS SLAM

In general, the underlying stochastic system for the navigation and mapping problem can be represented using the nonlinear 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 $\mathbf{x}_k$ is a random vector representing the vehicle pose at time step k, $\mathbf{g}$ is the vehicle motion model, $\mathbf{u}_k$ is the control input at time step k, $\boldsymbol{\delta}_k$ is the process noise at time step k, $\mathbf{z}_k^j$ is the jth 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 target or landmark i, and $\boldsymbol{\epsilon}_k$ is the measurement noise.

Placing the independent random vectors of landmarks into an RFS, the observed targets or landmarks up to time step k are defined as

$$\mathcal{M}_k \equiv \left\{\mathbf{m}^1, \mathbf{m}^2, \ldots, \mathbf{m}^m\right\}, \tag{3}$$

where the cardinality $|\mathcal{M}_k| = m$ is also a random variable.

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

$$\mathcal{Z}_k \equiv \left\{\mathbf{z}_k^1, \mathbf{z}_k^2, \ldots, \mathbf{z}_k^n\right\} \tag{4}$$

and may contain clutter (false alarms). The clutter intensity κ is the expected number of clutter measurements over a space. The probability of detection $P_{\mathrm{D}}(\mathbf{x}, \mathbf{m})$ is the probability of obtaining a measurement from a landmark at position $\mathbf{m}$ and vehicle pose $\mathbf{x}$.

In the navigation problem being considered, the best estimate of the vehicle trajectory and target positions are sought, using the information contained in the available control inputs and sensor measurements. Using a probabilistic approach, the required estimate at each time step is

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

Similar to vector-based formulations, (5) can theoretically be solved by recursive Bayesian estimation. Following the FastSLAM approach, (5) can be factored as

$$p\left(\mathbf{x}_{0:k} \middle| \mathcal{Z}_{1:k}, \mathbf{u}_{1:k}\right) p\left(\mathcal{M}_k \middle| \mathbf{x}_{0:k}, \mathcal{Z}_{1:k}, \mathbf{u}_{1:k}\right). \tag{6}$$

The first term in (6) is a conditional probability density function on the vehicle trajectory and can be estimated using a PF; the second term is the density of landmarks conditioned on the trajectory. In this article, it is assumed that landmarks are static and not dependent on any control inputs. In general, their dynamics can follow any process model. The map density can be updated using the latest measurement according to the RFS Bayes filter:

$$p(\mathcal{M}_k|\mathbf{x}_{0:k}, \mathcal{Z}_{1:k}) = \frac{p(\mathcal{Z}_k|\mathbf{x}_{0:k}, \mathcal{M}_k)p(\mathcal{M}_k|\mathbf{x}_{0:k}, \mathcal{Z}_{1:k-1})}{p(\mathcal{Z}_k|\mathbf{x}_{0:k}, \mathcal{Z}_{1:k-1})}. \tag{7}$$

Since the Bayes filter is computationally intractable in general, approximations are made to obtain closed-form solutions for the update process. Furthermore, instead of updating the map density directly, its first moment, also known as the intensity or PHD, can be used as an approximation for the update. The PHD of the map can be expressed using $\delta_{\mathbf{w}}$, the Dirac delta density (concentrated at $\mathbf{w}$), as [2]

$$v\left(\mathbf{m}\right) = \int \sum_{\mathbf{w}\in\mathcal{M}} \delta_{\mathbf{w}}\left(\mathbf{m}\right) p\left(\mathcal{M}\right) \delta\mathcal{M}. \tag{8}$$

Let $v^-(\mathbf{m})$ represent the map intensity corresponding to the prior map before update—that is, $p(\mathcal{M}_k|\mathbf{x}_{0:k}, \mathcal{Z}_{1:k-1})$ and $v^+(\mathbf{m})$ the map intensity corresponding to the posterior map after update—that is, $p(\mathcal{M}_k|\mathbf{x}_{0:k}, \mathcal{Z}_{1:k})$.

For the vehicle trajectory, the predicted estimate using the latest process input is calculated as

$$\begin{aligned} &p\left(\mathbf{x}_{0:k} \middle| \mathcal{Z}_{1:k}, \mathbf{u}_{1:k}\right) \\ &\quad = p\left(\mathbf{x}_k \middle| \mathbf{x}_{k-1}, \mathbf{u}_{1:k}\right) p\left(\mathbf{x}_{0:k-1} \middle| \mathcal{Z}_{1:k-1}, \mathbf{u}_{1:k-1}\right), \end{aligned} \tag{9}$$

where the first term is the transition density from the process model. The trajectory estimate is updated using the latest measurements according to

$$\begin{aligned} &p\left(\mathbf{x}_{0:k} \middle| \mathcal{Z}_{1:k}, \mathbf{u}_{1:k}\right) \\ &\quad = \frac{p\left(\mathcal{Z}_k \middle| \mathbf{x}_{0:k}, \mathcal{Z}_{1:k-1}\right) p\left(\mathbf{x}_{0:k} \middle| \mathcal{Z}_{1:k-1}, \mathbf{u}_{1:k}\right)}{p\left(\mathcal{Z}_k \middle| \mathcal{Z}_{1:k-1}\right)} \end{aligned} \tag{10}$$

$$= \left(\eta p\left(\mathcal{Z}_k \middle| \mathbf{x}_{0:k}, \mathcal{Z}_{1:k-1}\right)\right) p\left(\mathbf{x}_{0:k} \middle| \mathcal{Z}_{1:k-1}, \mathbf{u}_{1:k}\right). \tag{11}$$

In using a PF to estimate the vehicle trajectory, the parenthesized terms are used to calculate importance-weighting factors for each particle. The calculation of these factors is the focus of this article.

B. Importance-Weighting Strategies

Existing particle importance-weighting strategies will first be reviewed. The proposed strategy will then be presented. Implementation details of the various strategies will be covered in Section IV.

One possible way of determining the importance-weighting factor is to apply Bayes's theorem [4]:

$$\eta p\left(\mathcal{Z}_k|\,\mathbf{x}_{0:k},\,\mathcal{Z}_{1:k-1}\right) = \eta p\left(\mathcal{Z}_k|\,\mathcal{M}_k,\,\mathbf{x}_{0:k}\right)\frac{p\left(\mathcal{M}_k|\,\mathcal{Z}_{1:k-1},\,\mathbf{x}_{0:k}\right)}{p\left(\mathcal{M}_k|\,\mathcal{Z}_{1:k},\,\mathbf{x}_{0:k}\right)}. \quad (12)$$

Note from (12) that the RFS $\mathcal{M}_k$ is a free variable that can be chosen arbitrarily, since it appears only on the right-hand side. Theoretically, this choice should not make a difference to the resulting particle weight. For computational tractability however, some assumptions need to be made on the map densities in the right-hand-side fraction of (12). They can be approximated as multiobject Poisson distributions such that

$$p\left(\mathcal{M}_k|\,\mathcal{Z}_{1:k-1},\,\mathbf{x}_{0:k}\right) = p\left(\left\{\mathbf{m}_k^1,\,\mathbf{m}_k^2,\,\ldots,\,\mathbf{m}_k^n\right\}\middle|\,\mathcal{Z}_{1:k-1},\,\mathbf{x}_{0:k}\right) = \frac{m^{-n}}{\exp m^-}\prod_{i=1}^{n} p\left(\mathbf{m}^i\middle|\,\mathcal{Z}_{1:k-1},\,\mathbf{x}_{0:k}\right) \quad (13)$$

$$p\left(\mathcal{M}_k|\,\mathcal{Z}_{1:k},\,\mathbf{x}_{0:k}\right) = p\left(\left\{\mathbf{m}_k^1,\,\mathbf{m}_k^2,\,\ldots,\,\mathbf{m}_k^n\right\}\middle|\,\mathcal{Z}_{1:k},\,\mathbf{x}_{0:k}\right) = \frac{m^{+n}}{\exp m^+}\prod_{i=1}^{n} p\left(\mathbf{m}^i\middle|\,\mathcal{Z}_{1:k},\,\mathbf{x}_{0:k}\right), \quad (14)$$

where m^- and m^+ are the respective means of the cardinality distributions. Due to the multiobject Poisson assumption, the choice of $\mathcal{M}_k$ does make a difference when the particle weight is calculated.

1) *The Empty-Set Strategy*: From (12), the simplest approach is to assume that $\mathcal{M}_k = \emptyset$. All measurements are considered to be multiobject Poisson-distributed clutter, and the measurement likelihood term in (12) becomes

$$p\left(\mathcal{Z}_k|\,\mathcal{M}_k,\,\mathbf{x}_{0:k}\right) = p_\kappa\left(\mathcal{Z}_k\right) = \frac{\prod_{\mathbf{z}_k\in\mathcal{Z}_k} c\left(\mathbf{z}_k|\,\mathbf{x}_k\right)}{\exp\int c\left(\mathbf{z}\,|\,\mathbf{x}_k\right)\mathrm{d}\mathbf{z}}. \quad (15)$$

Note that in many applications, clutter is assumed to be uniformly distributed in space. From here on, let

$$c = c\left(\mathbf{z}_k|\,\mathbf{x}_k\right) \quad (16)$$

and let the expected number of clutter measurements be

$$N_c \equiv \int c\left(\mathbf{z}|\,\mathbf{x}_k\right)\mathrm{d}\mathbf{z}. \quad (17)$$

Then (12) simplifies to:

$$\eta p\left(\mathcal{Z}_k|\,\mathbf{x}_{0:k},\,\mathcal{Z}_{1:k-1}\right) = \eta\frac{c^{|\mathcal{Z}_k|}}{N_c}\exp\left(m^+ - m^-\right). \quad (18)$$

In Section V, it will be shown that this choice for $\mathcal{M}_k$ is a poor approximation of the map density, and it causes filter divergence in most situations.

2) *The Single-Feature Strategy*: Another simplistic approximation is to assume a map that is based on a single feature, $\mathcal{M}_k = \{\mathbf{m}\}$. All measurements but one are considered clutter, and the measurement likelihood term in (12) becomes

$$p\left(\mathcal{Z}_k|\,\mathcal{M}_k,\,\mathbf{x}_{0:k}\right) \approx (1-P_\mathrm{D})\frac{c^{|\mathcal{Z}_k|}}{\exp N_c} + P_\mathrm{D}\sum_{\mathbf{z}_k\in\mathcal{Z}_k}\left(\frac{c^{|\mathcal{Z}_k-\{\mathbf{z}_k\}|}}{\exp N_c}p\left(\mathbf{z}_k|\,\mathbf{m},\,\mathbf{x}_{0:k}\right)\right). \quad (19)$$

The first term on the right-hand side of (19) represents the case where the single feature is misdetected and all the measurements are considered clutter. The second term represents the cases where one of the $|\mathcal{Z}_k|$ measurements originates from the single feature, while the remaining measurements are clutter. With this approach, the fractional term of map densities in (12) becomes

$$\frac{p\left(\mathcal{M}_k|\,\mathcal{Z}_{1:k-1},\,\mathbf{x}_{0:k}\right)}{p\left(\mathcal{M}_k|\,\mathcal{Z}_{1:k},\,\mathbf{x}_{0:k}\right)} = \frac{m^- p\left(\mathbf{m}^i\middle|\,\mathcal{Z}_{1:k-1},\,\mathbf{x}_{0:k}\right)}{m^+ p\left(\mathbf{m}^i\middle|\,\mathcal{Z}_{1:k},\,\mathbf{x}_{0:k}\right)}\exp\left(m^+ - m^-\right). \quad (20)$$

It will be shown in Section V that this strategy does not provide much improvement over the empty-set strategy.

3) *The SC Filter*: Another way of determining the importance-weighting factor in (12) is through the evaluation of its equivalent set integral:

$$p\left(\mathcal{Z}_k|\,\mathbf{x}_{0:k},\,\mathcal{Z}_{1:k-1}\right) = \int p\left(\mathcal{Z}_k|\,\mathcal{M}_k,\,\mathbf{x}_{0:k}\right)p\left(\mathcal{M}_k|\,\mathbf{x}_{0:k},\,\mathcal{Z}_{1:k-1}\right)\delta\mathcal{M}_k. \quad (21)$$

In [29] an exact solution to (21) is shown for independent cluster processes, and a more specific solution for the SC-PHD filter based on the Poisson cluster process is given in [30]. This is implemented for SLAM in [5]. From [30], the weighting factor in (21) is evaluated as

$$\eta p\left(\mathcal{Z}_k|\,\mathbf{x}_{0:k},\,\mathcal{Z}_{1:k-1}\right) = \eta\exp\left(-\int v^-\left(\mathbf{m}\right)P_\mathrm{D}\mathrm{d}\mathbf{m}\right) \times \prod_{\mathbf{z}_k\in\mathcal{Z}_k}\left(\kappa\left(\mathbf{z}\right) + \int v^-\left(\mathbf{m}\right)P_\mathrm{D}\,p\left(\mathbf{z}_k|\,\mathbf{m},\,\mathbf{x}_{0:k}\right)\mathrm{d}\mathbf{m}\right). \quad (22)$$

This weighting method has been shown to produce consistent estimates in various scenarios and will be used in Section V as a benchmark for comparison.

4) *The Multifeature Strategy*: The proposed particle-weighting strategy in this article is based on (12), and its development was motivated by the poor performance of the empty-set and single-feature strategies, which are also derived from (12). The proposed strategy assumes that $\mathcal{M}_k$ includes all the estimated landmarks in the vehicle sensor's field of view (FOV).[3] That is, $\mathcal{M}_k = \{\mathbf{m}^1, \mathbf{m}^2, \ldots, \mathbf{m}^m\}$. The intuition behind this is that, since the set measurement likelihood is a contributing factor in (12), it would be more beneficial to choose $\mathcal{M}_k$ based on all current landmark estimates. This allows all

[3] This is equivalent to including all the estimated landmarks in the map, since the probability of detection of features outside the FOV is zero.

measurements to be considered as both real or clutter. Consider first the simplest multifeature case, where there are in the FOV two targets or landmarks, $\mathcal{M}_k = \{\mathbf{m}^1, \mathbf{m}^2\}$, with respective probabilities of detection P_{D}^1 and P_{D}^2. The measurement likelihood term in (12) becomes

$$\begin{aligned} &p\left(\mathcal{Z}_k \middle| \mathcal{M}_k, \mathbf{x}_{0:k}\right) \\ &\quad = \left(1 - P_{\mathrm{D}}^1\right)\left(1 - P_{\mathrm{D}}^2\right) \frac{c^{|\mathcal{Z}_k|}}{\exp N_c} \\ &\qquad + P_{\mathrm{D}}^1\left(1 - P_{\mathrm{D}}^2\right) \sum_{\mathbf{z}_k \in \mathcal{Z}_k} \left(\frac{c^{|\mathcal{Z}_k - \{\mathbf{z}_k\}|}}{\exp N_c} p\left(\mathbf{z}_k \middle| \mathbf{m}^1, \mathbf{x}_{0:k}\right)\right) \\ &\qquad + P_{\mathrm{D}}^2\left(1 - P_{\mathrm{D}}^1\right) \sum_{\mathbf{z}_k \in \mathcal{Z}_k} \left(\frac{c^{|\mathcal{Z}_k - \{\mathbf{z}_k\}|}}{\exp N_c} p\left(\mathbf{z}_k \middle| \mathbf{m}^2, \mathbf{x}_{0:k}\right)\right) \\ &\qquad + P_{\mathrm{D}}^1 P_{\mathrm{D}}^2 \sum_{\mathbf{z}_1 \in \mathcal{Z}_k} \sum_{\mathbf{z}_2 \in \mathcal{Z}_k - \mathbf{z}_1} \Big(\frac{c^{|\mathcal{Z}_k - \{\mathbf{z}_1 \mathbf{z}_2\}|}}{\exp N_c} p\left(\mathbf{z}_1 | \mathbf{m}^1, \mathbf{x}_{0:k}\right) \\ &\qquad \times p\left(\mathbf{z}_2 | \mathbf{m}^2, \mathbf{x}_{0:k}\right)\Big). \end{aligned} \tag{23}$$

The first term represents the possibility of both features not being detected by any measurements, which are all considered as clutter. In the second term, only $\mathbf{m}^1$ is detected by one of the measurements, while the rest are considered clutter. Similarly, in the third term, only $\mathbf{m}^2$ is detected. The last term represents the case where both landmarks are detected and $|\mathcal{Z}_k| - 2$ measurements are considered clutter.

Consider now the case where there are m landmarks within the vehicle sensor's FOV and $\mathcal{M}_k = \{\mathbf{m}^1, \mathbf{m}^2, \ldots, \mathbf{m}^m\}$. To aid with the mathematical presentation, let $\boldsymbol{\theta}$ represent an assignment variable, where $\boldsymbol{\theta}^j = i$ indicates that measurement j is assigned to landmark i and $\boldsymbol{\theta}^j = 0$ if measurement j is unassigned. Furthermore, given $\boldsymbol{\theta}$, we have

$$\mathcal{Z}_k^{\boldsymbol{\theta}} \equiv \left\{ \mathbf{z}^j \in \mathcal{Z}_k \middle| \boldsymbol{\theta}^j \neq 0 \right\} \tag{24}$$

$$\overline{\mathcal{Z}_k^{\boldsymbol{\theta}}} \equiv \mathcal{Z}_k \setminus \mathcal{Z}_k^{\boldsymbol{\theta}} \tag{25}$$

$$\mathcal{M}_k^{\boldsymbol{\theta}} \equiv \left\{ \mathbf{m}^m \in \mathcal{M}_k \middle| \exists j, \boldsymbol{\theta}^j = m \right\} \tag{26}$$

$$\overline{\mathcal{M}_k^{\boldsymbol{\theta}}} \equiv \mathcal{M}_k \setminus \mathcal{M}_k^{\boldsymbol{\theta}}. \tag{27}$$

These define the sets of assigned and unassigned measurements and landmarks. The full set measurement likelihood on the right-hand side of (12) can then be written as

$$\begin{aligned} p\left(\mathcal{Z}_k \middle| \mathcal{M}_k, \mathbf{x}_{0:k}\right) &= p\left(\mathcal{Z}_k \middle| \left\{\mathbf{m}^1, \mathbf{m}^2, \ldots, \mathbf{m}^m\right\}, \mathbf{x}_{0:k}\right) \\ &= p_\kappa\left(\mathcal{Z}_k\right) \prod_i^{|\mathcal{M}_k|} \left(1 - P_{\mathrm{D}}\left(\mathbf{x}_k, \mathbf{m}^i\right)\right) \sum_{\boldsymbol{\theta}} \prod_{\substack{j=1 \\ \boldsymbol{\theta}^j \neq 0}}^{|\mathcal{Z}_k|} \frac{P_{\mathrm{D}}\left(\mathbf{x}_k, \mathbf{m}^{\boldsymbol{\theta}^j}\right) p\left(\mathbf{z}^j \middle| \mathbf{m}^{\boldsymbol{\theta}^j}, \mathbf{x}_{0:k}\right)}{\left(1 - P_{\mathrm{D}}\left(\mathbf{x}_k, \mathbf{m}^{\boldsymbol{\theta}^j}\right)\right) p_\kappa\left(\mathbf{z}^j\right)} \\ &= p_\kappa\left(\mathcal{Z}_k\right) \prod_i^{|\mathcal{M}_k|} \left(1 - P_{\mathrm{D}}\left(\mathbf{x}_k, \mathbf{m}^i\right)\right) \sum_{\boldsymbol{\theta}} \left(p\left(\mathcal{Z}_k^{\boldsymbol{\theta}} \middle| \mathcal{M}_k^{\boldsymbol{\theta}}, \mathbf{x}_{0,k}, \boldsymbol{\theta}\right) \prod_{\mathbf{m} \in \mathcal{M}_k^{\boldsymbol{\theta}}} \frac{P_{\mathrm{D}}\left(\mathbf{x}_k, \mathbf{m}\right)}{\left(1 - P_{\mathrm{D}}\left(\mathbf{x}_k, \mathbf{m}\right)\right)} \prod_{\mathbf{z} \in \mathcal{Z}_k^{\boldsymbol{\theta}}} \frac{1}{p_\kappa\left(\mathbf{z}\right)} \right) \\ &= \sum_{\boldsymbol{\theta}} \left(p\left(\mathcal{Z}_k^{\boldsymbol{\theta}} \middle| \mathcal{M}_k^{\boldsymbol{\theta}}, \mathbf{x}_{0,k}, \boldsymbol{\theta}\right) \prod_{\mathbf{m} \in \mathcal{M}_k^{\boldsymbol{\theta}}} P_{\mathrm{D}}\left(\mathbf{x}_k, \mathbf{m}\right) \prod_{\mathbf{m} \in \overline{\mathcal{M}_k^{\boldsymbol{\theta}}}} \left(1 - P_{\mathrm{D}}\left(\mathbf{x}_k, \mathbf{m}\right)\right) p_\kappa\left(\overline{\mathcal{Z}_k^{\boldsymbol{\theta}}}\right) \right). \end{aligned} \tag{28}$$

The summation over $\boldsymbol{\theta}$ considers all possible permutations of assignments. Unpaired measurements give the clutter factor $p_\kappa(\overline{\mathcal{Z}_k^{\boldsymbol{\theta}}})$, while unpaired landmarks give the misdetection factors $(1 - P_{\mathrm{D}}(\cdot))$. Paired couples (where $\boldsymbol{\theta}^j = i \neq 0$) provide the probabilities of detection $P_{\mathrm{D}}(\cdot)$ and the measurement likelihood factor $p(\mathcal{Z}_k^{\boldsymbol{\theta}} | \mathcal{M}_k^{\boldsymbol{\theta}}, \mathbf{x}_{0,k}, \boldsymbol{\theta})$. Furthermore, using this strategy, the fraction of map densities in (12) becomes

$$\begin{aligned} &\frac{p\left(\mathcal{M}_k \middle| \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k}\right)}{p\left(\mathcal{M}_k \middle| \mathcal{Z}_{1:k}, \mathbf{x}_{0:k}\right)} \\ &= \frac{m^{-n} \prod_{i=1}^n p\left(\mathbf{m}^i \middle| \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k}\right)}{m^{+n} \prod_{i=1}^n p\left(\mathbf{m}^i \middle| \mathcal{Z}_{1:k}, \mathbf{x}_{0:k}\right)} \exp\left(m^+ - m^-\right). \end{aligned} \tag{29}$$

In Section V it will be shown that using the multifeature strategy drastically improves the quality of estimates over the empty-set and single-feature strategies and produces a slight improvement over the SC-PHD filter. The practical implementation of the proposed method in a PHD SLAM framework will be detailed next.

IV. FILTER IMPLEMENTATION

In the random-vector form of the Bayes filter, Gaussian assumptions allow the KF to be derived [31]. With RFSs, the map cardinality can be assumed to follow a multiobject Poisson distribution. Clutter measurements are also assumed to be multiobject Poisson distributed. By approximating these distributions using their first statistical moments, the Bayes filter can be approximated as the PHD filter. Furthermore, by assuming the elements within $\mathcal{M}_k$ to be Gaussian random vectors, the PHD of landmarks can be expressed as a GM of an arbitrary number N of Gaussians $\mathcal{N}_k^r \equiv \mathcal{N}(\boldsymbol{\mu}_k^r, \boldsymbol{\Sigma}_k^r)$:

$$v_k = \sum_{r=1}^{N} w_k^r \mathcal{N}_k^r, \tag{30}$$

where w_k^r is the time-varying weight of the rth Gaussian, with time-varying mean $\boldsymbol{\mu}_k^r$ and covariance $\boldsymbol{\Sigma}_k^r$. From this, the Bayes filter is approximated as the GM-PHD filter [11], within which the (spatial) update of each Gaussian can be performed using the KF or a derivative such as the EKF for nonlinear systems.

Returning to the desired factored posterior (6), recall that the first term on the right-hand side is a conditional probability density function on the robot trajectory and can be sampled using particles. The second factor is the density of landmarks conditioned on the trajectory. Mullane et al. [4] used the GM-PHD filter to approximate and calculate the second factor. The main steps in an iteration of this filter will now be reviewed, with a focus on the implementation of the particle importance weighting.

A. Particle Propagation

At time step k, each particle $[\iota]$ with state $\mathbf{x}_{k-1}^{[\iota]}$, sampled from the prior trajectory distribution

$$p\left(\mathbf{x}_{0:k-1} | \, \mathcal{Z}_{1:k-1}, \mathbf{u}_{1:k-1}\right), \tag{31}$$

is propagated forward in time by sampling the process noise $\boldsymbol{\delta}_{k-1}^{[\iota]}$ and using the motion model (1) [32]:

$$\mathbf{x}_k^{[\iota]} = \mathbf{g}\left(\mathbf{x}_{k-1}^{[\iota]}, \mathbf{u}_{k-1}, \boldsymbol{\delta}_{k-1}^{[\iota]}\right). \tag{32}$$

B. Generation of Birth Gaussians

For each particle, its PHD from the previous update, $v_{k-1}^{+[\iota]}$, is added to $|\mathcal{Z}_{k-1}|$ new Gaussians with (arbitrarily small) weights w_{B}, representing potential new targets or landmarks:

$$v_k^{-[\iota]}(\mathbf{m}) = v_{k-1}^{+[\iota]} + \sum_b^{|\mathcal{Z}_{k-1}|} w_{\mathrm{B}} \mathcal{N}_k^{b,[\iota]}\left(\mu_k^{b,[\iota]}, \Sigma_k^{b,[\iota]}\right), \tag{33}$$

where the intensity $v_0^{+[\iota]}$ initially contains no Gaussians. The parameters of the birth Gaussians can be determined by using the inverse measurement model, derived from (2).

C. Map Update

The PHD for each particle is updated with measurements according to the PHD filter corrector equation

$$\begin{aligned} v_k^{+[\iota]}(\mathbf{m}) = &\sum_{r=1}^{N_k^{-[\iota]}} \left(1 - P_{\mathrm{D}}^{r,[\iota]}\right) w_k^{r,[\iota]} \mathcal{N}_k^{r,[\iota]} \\ &+ \sum_{j=1}^{|\mathcal{Z}_k|} \sum_{r=1}^{N_k^{-[\iota]}} w_k^{(j,r),[\iota]} \mathcal{N}_k^{(j,r),[\iota]}, \end{aligned} \tag{34}$$

where $N_k^{-[\iota]}$ is the number of Gaussians that compose $v_k^{-[\iota]}$. The first term in (34) reflects the possibility of misdetections. The second term adds a new Gaussian for each measurement-to-Gaussian pairing, such that every measurement is used to update every Gaussian. The weighting factor is determined as

$$w_k^{(j,r)[\iota]} = \frac{P_{\mathrm{D}}^{r,[\iota]} w_k^{r,[\iota]} q\left(\mathbf{z}_k^j, \mathcal{N}_k^{r,[\iota]}\right)}{\kappa + \sum_{l=1}^{N_k^{-[\iota]}} P_{\mathrm{D}}^{l,[\iota]} w_k^{l,[\iota]} q\left(\mathbf{z}_k^i, \mathcal{N}_k^{l,[\iota]}\right)}, \tag{35}$$

where $q(\cdot)$ is the measurement likelihood given a Gaussian within the current map intensity. The mean and covariance for a new Gaussian created from measurement j and Gaussian r in (34) is determined using the EKF update step.

D. Merging and Pruning of the Map

Gaussians with small weights are eliminated from the intensity function, while those that are close to each other are merged together. This approximation is critical in limiting the computational requirement of the GM-PHD filter, which would otherwise grow exponentially [11].

E. Importance Weighting and Resampling

From (12)–(14), the importance weighting factor $\gamma_k^{[\iota]}$ of particle ι can be evaluated as

$$\begin{aligned} \gamma_k^{[\iota]} &= \eta p\left(\mathcal{Z}_k | \, \mathbf{x}_{0:k}^{[\iota]}, \mathcal{Z}_{1:k-1}\right) \gamma_{k-1}^{[\iota]} \\ &= \eta p\left(\mathcal{Z}_k | \, \mathcal{M}_k, \mathbf{x}_{0:k}^{[\iota]}\right) \frac{m^{-^n} \prod_{\mathbf{m} \in \mathcal{M}_k} p\left(\mathbf{m} | \, \mathcal{Z}_{1:k-1}, \mathbf{x}_{0:k}^{[\iota]}\right)}{m^{+^n} \prod_{\mathbf{m} \in \mathcal{M}_k} p\left(\mathbf{m} | \, \mathcal{Z}_{1:k}, \mathbf{x}_{0:k}^{[\iota]}\right)} \\ &\quad \times \exp\left(m^{+[\iota]} - m^{-[\iota]}\right) \gamma_{k-1}^{[\iota]} \end{aligned} \tag{36}$$

$$\begin{aligned} &= \eta p\left(\mathcal{Z}_k | \, \mathcal{M}_k, \mathbf{x}_{0:k}^{[\iota]}\right) \frac{\prod_{\mathbf{m} \in \mathcal{M}_k} v_k^{-[\iota]}(\mathbf{m})}{\prod_{\mathbf{m} \in \mathcal{M}_k} v_k^{+[\iota]}(\mathbf{m})} \\ &\quad \times \exp\left(m^{+[\iota]} - m^{-[\iota]}\right) \gamma_{k-1}^{[\iota]}, \end{aligned} \tag{37}$$

where m^- and m^+ can be calculated by summing all the Gaussian weights in v_k^- and v_k^+, respectively:

$$m^{-[\iota]} = \sum_{r=1}^{N_k^{-[\iota]}} w_k^{r,[\iota]}, \quad m^{+[\iota]} = \sum_{r=1}^{N_k^{+[\iota]}} w_k^{r,[\iota]}. \tag{38}$$

When the weighting for all particles has been calculated, resampling occurs, with higher-weighted particles having a higher probability of being sampled. The normalizing constant η will be ignored from here on, since all particle weights will be multiplied by this same constant, thus having no effect on the particle resampling procedure. The implementation of each weighting strategy now follows.

1) *The Empty-Set Strategy*: Following the results from (18), for the empty-set strategy (37) simplifies to

$$\gamma_k^{[\iota]} = \frac{c^{|\mathcal{Z}_k|}}{N_c} \exp\left(m^{+[\iota]} - m^{-[\iota]}\right) \gamma_{k-1}^{[\iota]}. \tag{39}$$

2) *The Single-Feature Strategy*: For the single-feature strategy, (37) simplifies to

$$\begin{aligned} &\gamma_k^{[\iota]} \\ &= \left[(1 - P_{\mathrm{D}}) \frac{c^{|\mathcal{Z}_k|}}{\exp N_c} + P_{\mathrm{D}} \sum_{\mathbf{z}_k \in \mathcal{Z}_k} \left(\frac{c^{|\mathcal{Z}_k - \{\mathbf{z}_k\}|}}{\exp N_c} p\left(\mathbf{z}_k | \, \mathbf{m}, \mathbf{x}_{0:k}^{[\iota]}\right)\right)\right] \\ &\quad \times \frac{v_k^{-[\iota]}(\mathbf{m}) \exp\left(m^{+[\iota]}\right)}{v_k^{+[\iota]}(\mathbf{m}) \exp\left(m^{-[\iota]}\right)} \gamma_{k-1}^{[\iota]}, \end{aligned} \tag{40}$$

where, for example, $\mathbf{m}$ could be selected to equal the mean of the highest-weighted Gaussian in the GM posterior map intensity.

3) *The SC Filter*: From (21), it was shown in [5] that the particle weighting from the SC-PHD filter can be calculated using the prior map PHD $v_k^{-[\iota]}$ as

$$\gamma_k^{[\iota]} = \exp\left(m^{-[\iota]}\right) \times \prod_{\mathbf{z}\in\mathcal{Z}_k}\left(\kappa(\mathbf{z}) + P_{\mathrm{D}}\sum_{r=1}^{N_k^{-[\iota]}} p\left(\mathbf{z}|\,\mathcal{N}_k^{r,[\iota]}, \mathbf{x}_{0:k}^{[\iota]}\right) w_k^{r,[\iota]}\right)\gamma_{k-1}^{[\iota]}. \quad (41)$$

It is interesting to note that in [33], in an attempt to reduce the computational complexity of evaluating the set measurement likelihood, a similar equation to (41) was presented that did not include the exponential term. A physical interpretation of the equation in [33] (product of sums of individual likelihoods) is akin to summing all possible permutations of likelihoods—as in (28)—but allowing measurements to be paired with more than one landmark.

4) *The Multifeature Strategy*: From (28) and (29), a particle's importance-weighting factor based on the proposed multi-feature strategy can be calculated as

$$\gamma_k^{[\iota]} = \sum_{\boldsymbol{\theta}}\left(p\left(\mathcal{Z}_k^{\boldsymbol{\theta}}|\,\mathcal{M}_k^{\boldsymbol{\theta}}, \mathbf{x}_{0,k}^{[\iota]}, \boldsymbol{\theta}\right)\prod_{\mathbf{m}\in\mathcal{M}_k^{\boldsymbol{\theta}}} P_{\mathrm{D}}\left(\mathbf{x}_k^{[\iota]}, \mathbf{m}\right) \times \prod_{\mathbf{m}\in\overline{\mathcal{M}_k^{\boldsymbol{\theta}}}}\left(1 - P_{\mathrm{D}}\left(\mathbf{x}_k^{[\iota]}, \mathbf{m}\right)\right) p_\kappa\left(\overline{\mathcal{Z}_k^{\boldsymbol{\theta}}}\right)\right) \times \frac{\prod_{\mathbf{m}\in\mathcal{M}_k} v_k^{-[\iota]}(\mathbf{m})}{\prod_{\mathbf{m}_k\in\mathcal{M}_k} v_k^{+[\iota]}(\mathbf{m})}\frac{\exp m^{+[\iota]}}{\exp m^{-[\iota]}}\gamma_{k-1}^{[\iota]}. \quad (42)$$

In $\mathcal{M}_k$, all the landmark positions from the posterior map within a particle's FOV should be included. These can be obtained from the posterior map intensity by selecting the means of Gaussians that have weights above a threshold value.

Solving the multifeature set measurement likelihood—that is, the summation term in (42)—is computationally complex if a brute-force approach is used. The most expensive method is to iterate through every possible set of correspondences (i.e., through methods such as lexicographical ordering). The factorial complexity of this approach makes calculating the measurement likelihood infeasible when the number of landmarks and measurements exceeds eight to ten [28]. In many navigation scenarios, the number of targets or landmarks and measurements greatly exceeds this limit. Hence, it is essential to find computationally efficient methods for approximating the set measurement likelihood. The following two methods were implemented.

a) *Landmark and measurement grouping*: In general, the measurement likelihood between any landmark and measurement is nonzero. For practical purposes, small likelihood values (from landmarks and measurements that are spatially well separated) below a certain threshold can be assumed to be 0. This allows landmarks and measurements to be grouped in a way such that a landmark from one group has (almost) zero likelihood based on measurements from any other group. This divide-and-conquer method is similar to techniques used in multiple-target tracking for determining likely data-association hypotheses [34]. However, the evaluation of the set measurement likelihood requires not merely the best hypotheses, but the likelihood of all hypotheses.

One possible method to execute the grouping process—the one used in this article—is to represent landmarks and measurements as nodes in a graph, where an edge exists between nodes with nonzero measurement likelihood (after thresholding). A connected-component analysis [35] can be performed to identify the landmark–measurement groups. The complexity of this method is linear in the number of nodes and edges. After division into smaller groups, it is possible to find the overall measurement likelihood by taking the product of each group's set measurement likelihood.

b) *Matrix permanent*: The permanent of an $n \times n$ matrix $\mathbf{M}$ with elements $m_{i,j}$ is defined as

$$\mathrm{perm}(\mathbf{M}) = \sum_{\pi\in\Pi}\prod_{i=1}^{n} m_{i,\pi_i}, \quad (43)$$

where Π is a symmetric group that includes all possible permutations of one-to-one matchings between the row and column indices of $\mathbf{M}$ and π is a specific permutation. This is closely related to the evaluation of the set measurement likelihood, where the permutations are the different assignments. Matrix $\mathbf{M}$ can be constructed as a square matrix of size $|\mathcal{M}_k| + |\mathcal{Z}_k|$. An element in the first $|\mathcal{Z}_k|$ rows and $|\mathcal{M}_k|$ columns should take the form $p(\mathbf{z}_k^j|\mathbf{m}^j, \mathbf{x}_{0,k})P_{\mathrm{D}}(\mathbf{x}_k, \mathbf{m}_k^j)$, $\mathbf{z}_k^i \in \mathcal{Z}_k$, $\mathbf{m}_k^j \in \mathcal{M}_k$. The submatrix from the last $|\mathcal{M}_k|$ rows and first $|\mathcal{M}_k|$ columns should be diagonal with entries of the form $(1 - P_{\mathrm{D}}(\mathbf{x}_k, \mathbf{m}_k^j))$. The submatrix from the first $|\mathcal{Z}_k|$ rows and last $|\mathcal{M}_k|$ columns should also be diagonal, with entries of the form $c(\mathbf{z}_k^i)/\exp N_c$. The remaining submatrix from the last $|\mathcal{M}_k|$ rows and $|\mathcal{Z}_k|$ columns should be filled with 1s.

As mentioned previously, computing (43) is expensive, but this is performed on small subgroups of measurements and landmarks. Furthermore, the Ryser formula [36] can be used to calculate (43) exactly, and has a lower, exponential complexity in comparison to the brute-force factorial complexity. The results presented in this article are generated with the implementation of the Ryser formula. It is possible to further reduce computational costs by approximating the matrix permanent [37]. Jerrum et al. [38] developed an approximation method that has polynomial complexity (of power 7) and is based on fully polynomial randomized approximate schemes. However, Huang and Jebara [39] have pointed out that in practice, the method in [38] does not provide computational

advantages unless the size of the matrix is large. Alternatively, the method developed in [39] provides an approximation based on graphical models and belief propagation that has cubic complexity. Compared with other RFS filtering methods, the GLMB class of filters also achieves cubic complexity through the use of Murty's algorithm [40] to curb the exponential growth of components in the multiobject densities. In the predecessor to this article [28], Murty's algorithm was also proposed. For the purpose of evaluating the set measurement likelihood, however, it was not possible to quantify the error that would be introduced. Therefore, methods for approximating the matrix permanent with a known error bound are preferred. It is also worth noting that a physical interpretation of the matrix-permanent approach is the summation of likelihoods from all permutations of one-to-one measurement-to-landmark correspondences, which differs from the approximation in [33] that allows for many-to-one correspondences.

V. EXPERIMENTS

Simulations and hardware experimental data sets are used to evaluate the performance of the proposed multifeature strategy for the PHD SLAM filter in various navigation and mapping scenarios. Comparisons will be made with the empty-set strategy, the single-feature strategy, the SC-PHD filter, and FastSLAM from the traditional vector formulation.

A. Simulations

Simulations allow the detection and clutter statistics to be controlled and the ground truth (vehicle trajectory and landmark positions) to be known exactly. The vehicle has a simulated range-bearing sensor, and landmarks between the sensing distances of 5–25 m in any direction may be detected with a predefined probability of detection. Independent of the real detections, false measurements are added uniformly in the measurement space at each time step, with their number being Poisson distributed with a predefined intensity value. The results from a specific simulation trial will be examined first. In all cases, 200 particles were used for each filter, and the start of the simulated trajectory was at the origin.[4] With the probability of detection set to 0.5 and clutter intensity set at 0.005 m^{-2} (9.45 expected false measurements per time step), the results from various filters are shown in Fig. 1.

The FastSLAM algorithm [25] is based on the traditional random-vector formulation. As a well-known navigation and mapping filter, it will serve as a benchmark for comparisons.[5] Data association was performed by choosing the set of correspondences that maximizes the joint likelihood of all the measurements at a time step. This is accomplished by using optimal assignment algorithms such as [43, 44]. A binary Bayes filter was implemented to track the probabilities of existence for landmarks. This takes into account measurement statistics, and is needed for map-management purposes. The method is not expected to perform well in the presence of clutter, as is evident in Fig. 1a, where darker shading on a landmark-estimate ellipse corresponds to a higher probability of existence. The high-clutter setting caused mistakes in data association, which led to estimate divergence.

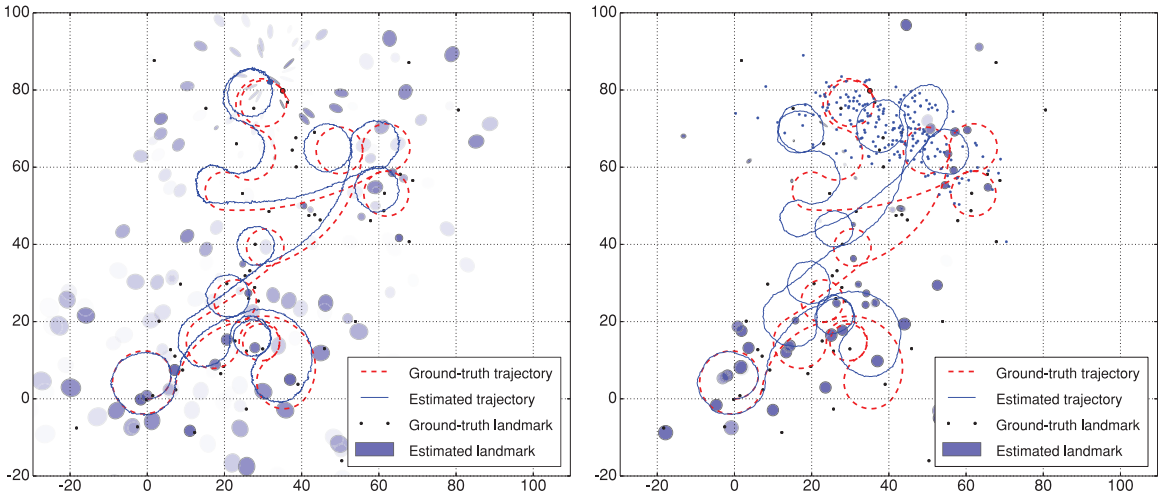
Estimation results from the PHD SLAM filter using the previously established empty-set and single-feature strategies are shown, respectively, in Figs. 1b, 1c. The shading on the landmark-estimate ellipses corresponds to the weight of a Gaussian in the GM, with darker shading representing a higher weight. The solution from the empty-set strategy can be seen to diverge shortly after the start of the trajectory. This poor performance can be attributed to the strategy's reliance on only the change in the number of landmark estimates, which ignores measurement likelihoods by assuming all measurements are clutter—see (18) and (39). The single-feature strategy yields slightly better estimation results, although both trajectory and landmark estimates still have large errors. For the implementation of this strategy, the single feature is selected as the mean of the highest-weighted Gaussian in the map intensity function. Although the likelihood of all measurements to this single landmark is considered, all measurements but one are still assumed to be clutter. This is considered as the contributing factor to the poor estimation results. The estimation results produced from the SC-PHD filter are shown in Fig. 1d, and both trajectory and landmark estimates appear to contain small errors. The results using the proposed multifeature strategy are shown in Fig. 1e. For implementation, the means of all Gaussians in the map intensity with weights higher than 0.75 and within the FOV of a particle were included in the multifeature set for particle weighting. Good estimates were obtained for the vehicle trajectory and landmark positions.

To examine the estimation errors in a more quantified manner, Fig. 2a compares the robot trajectory displacement errors for the various filters. The error from dead reckoning is also shown, and can be seen to be better than the estimate from the empty-set strategy during most of the simulation. The errors from FastSLAM and the single-feature strategy are similar, but the SC-PHD filter performed better. The PHD SLAM filter with the proposed multifeature strategy has the lowest trajectory estimation error.

The cardinalized optimal linear assignment (COLA) metric [45] in Fig. 2b is used to quantify the performance

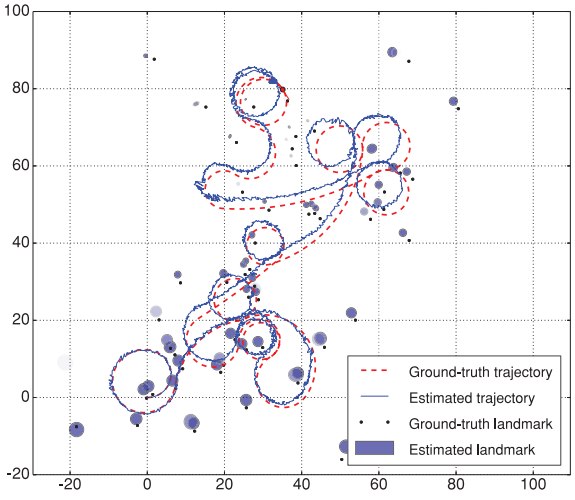
[4] All other filter parameters, such as spatial uncertainty, can be found in the open-source C++ library at https://github.com/kykleung/RFS-SLAM.git, which has an implementation of all the filters used in this article.

[5] FastSLAM 1.0 is used because it is algorithmically more similar to PHD SLAM. FastSLAM 2.0 [41] is expected to run faster with fewer particles but show similar results, as it still relies on a data-association routine that is separate from the filter. Multihypothesis implementations for FastSLAM such as [42] may improve robustness slightly, but they are computationally more expensive.

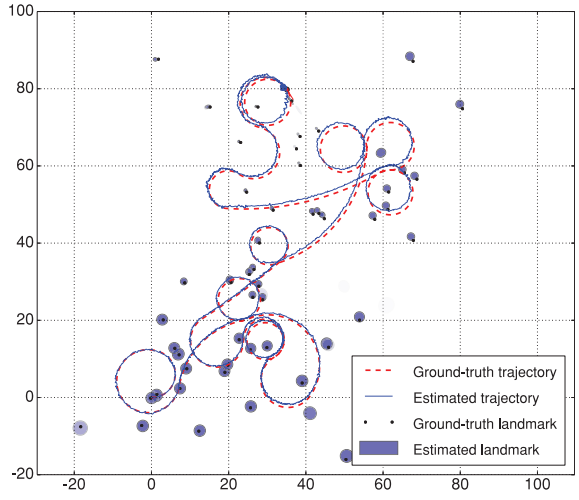


(a) FastSLAM.

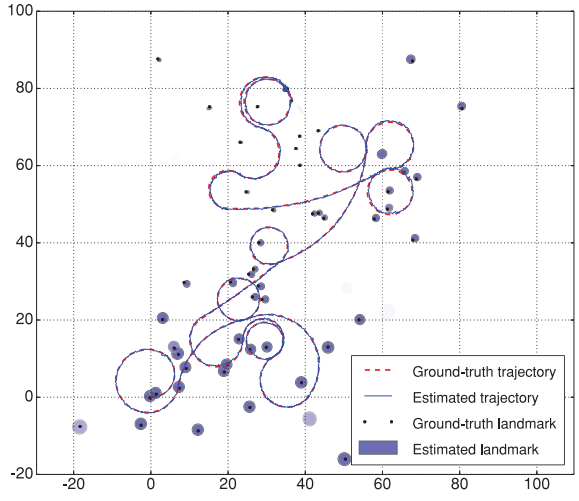
(b) PHD SLAM – empty-set strategy.



(c) PHD SLAM – single-feature strategy.



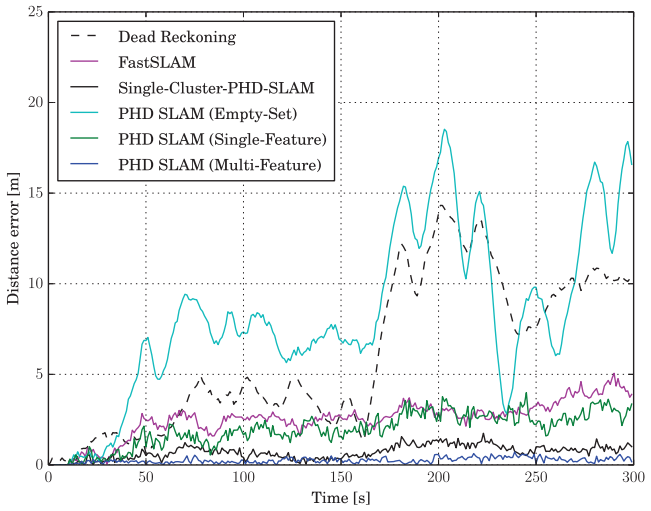
(d) Single-cluster-PHD filter.



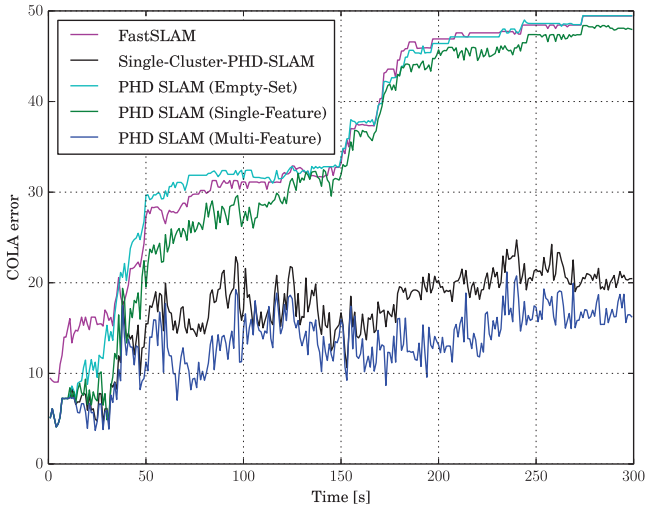
(e) PHD SLAM – multi-feature strategy.

Fig. 1. 2-D navigation simulation results from various RFS and vector-based filters, with axis units in meters.

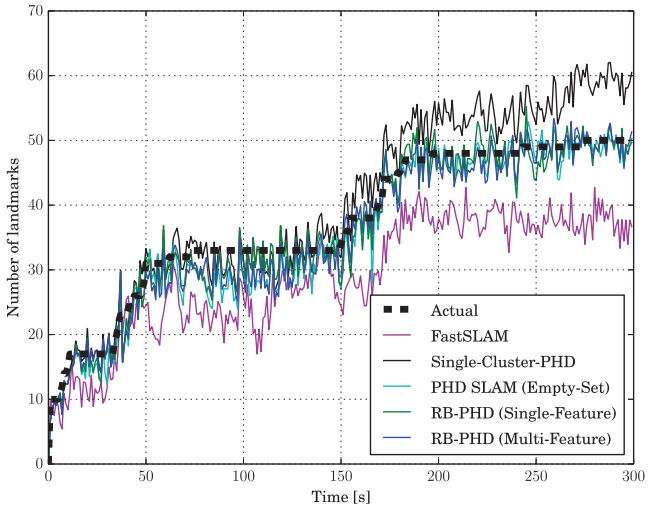
of the landmark estimates. This metric is similar to the optimal subpattern assignment metric [46] that has been used in other work in RFS filtering to quantify performance. While the optimal subpattern assignment metric is more appropriate for target-tracking problems where the performances from test scenarios involving different numbers of ground-truth targets need to be compared, the COLA metric is suitable for making



(a) Trajectory estimate displacement error.



(b) Landmark estimate COLA error.



(c) Landmark estimate cardinality error.

Fig. 2. 2-D simulation errors from the various RFS and vector-based approaches.

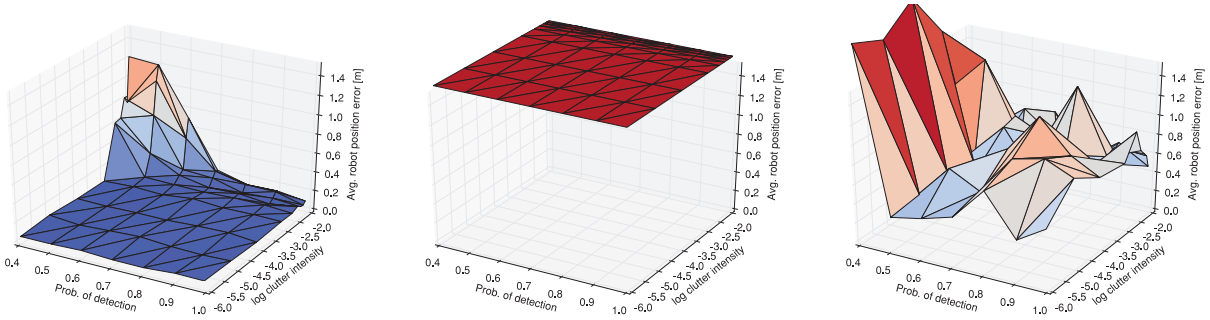
comparisons when the ground truth (map) is fixed [45]. Gaussians in the intensity map that have a weight higher than 0.75 were used in the evaluation of the metric. The landmark estimates of FastSLAM and the empty-set and single-feature strategies have high errors. Estimates generated by the SC-PHD filter have a lower error, but still higher than those of the PHD SLAM filter with the proposed multifeature strategy. Fig. 2c shows the cardinality estimate errors for all the tested methods. FastSLAM underestimated the number of landmarks, while the SC-PHD filter slightly overestimated the number of landmarks. The PHD SLAM filter performed best. The dashed black line shows the theoretically correct number of features which have passed through the simulated sensor's FOV.

To gain a better understanding of how each filter performs over a spectrum of detection statistics, over 2000 simulation trials were conducted to obtain the averaged trajectory and landmark estimate COLA errors under different probabilities of detection and clutter intensities. These are shown in Figs. 3 and 4, respectively. FastSLAM showed very low errors except in nonideal conditions, where the probabilities of detection are low and clutter intensity is high. These are the conditions that justify the use of RFS filters. The empty-set strategy performed the worst, with high errors regardless of the detection-statistic settings. The single-feature strategy performed only moderately better. Both the multifeature strategy and the SC-PHD filter performed similarly and were able to produce estimates with relatively low errors when the probability of detection is low and the clutter intensity is high.

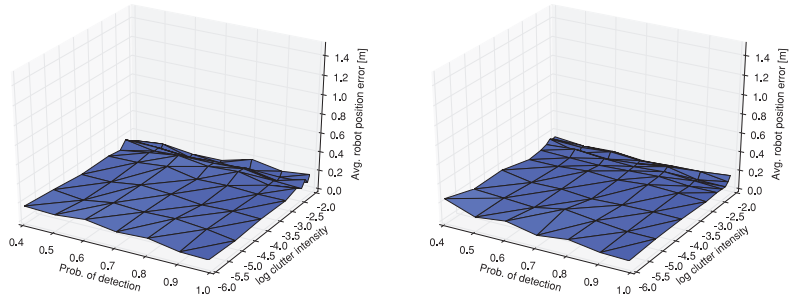
B. Victoria Park Data Set

To conduct a more realistic evaluation of the proposed multifeature weighting strategy, the publicly available Victoria Park data set [47] was used.[6] The data set contains a low amount of clutter, insufficient to cause estimate divergence except when the empty-set and single-feature strategies are used with the PHD SLAM filter. This is shown in Figs. 5a, 5b. The results obtained from FastSLAM are shown in Fig. 5c, and are consistent with the results shown in [25, 41]. To make the detection conditions less ideal and provide a more challenging scenario for all the algorithms, artificial clutter was added to the data set. The number of false measurements at each measurement time followed a Poisson distribution, and the false measurements were uniformly distributed within the sensor FOV. Hence, on top of the 52 974 measurements in the data set, about 21 500 artificial clutter measurements were added. Fig. 6 shows an image of the entire robot trajectory with all the gathered measurements superimposed. With the artificial clutter, the FastSLAM estimate diverged as shown in Fig. 5d. In the same

[6] In this data set, GPS-positioning ground-truth robot trajectory data are available only intermittently.

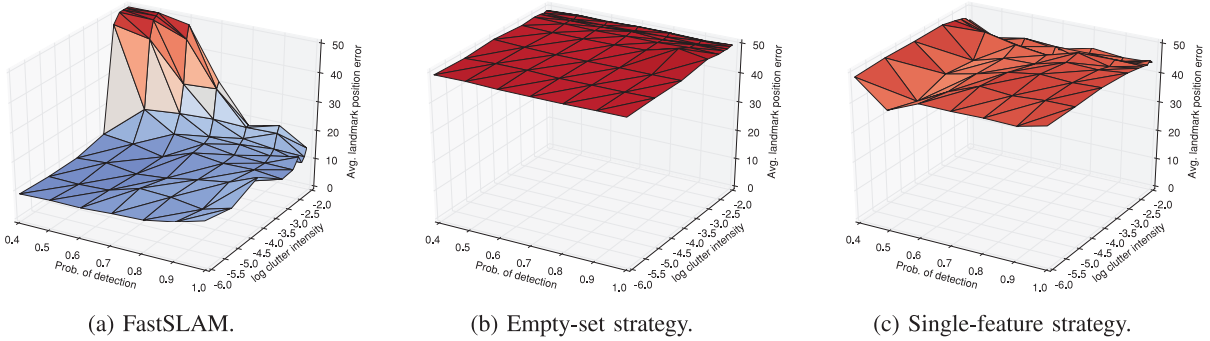


(a) FastSLAM. (b) Empty-set strategy. (c) Single-feature strategy.

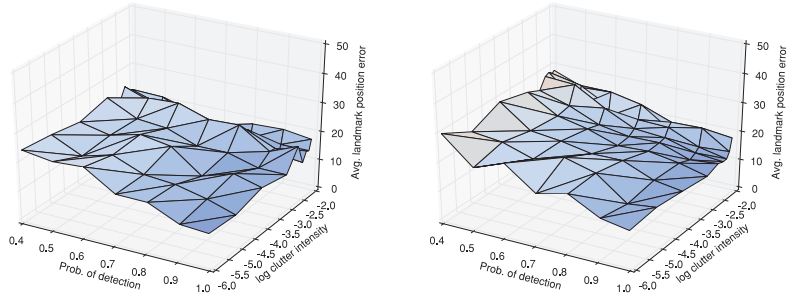


(d) Single-cluster-PHD filter. (e) Multi-feature strategy.

Fig. 3. Averaged trajectory-estimate errors under various probabilities of detection and clutter intensities.



(a) FastSLAM. (b) Empty-set strategy. (c) Single-feature strategy.



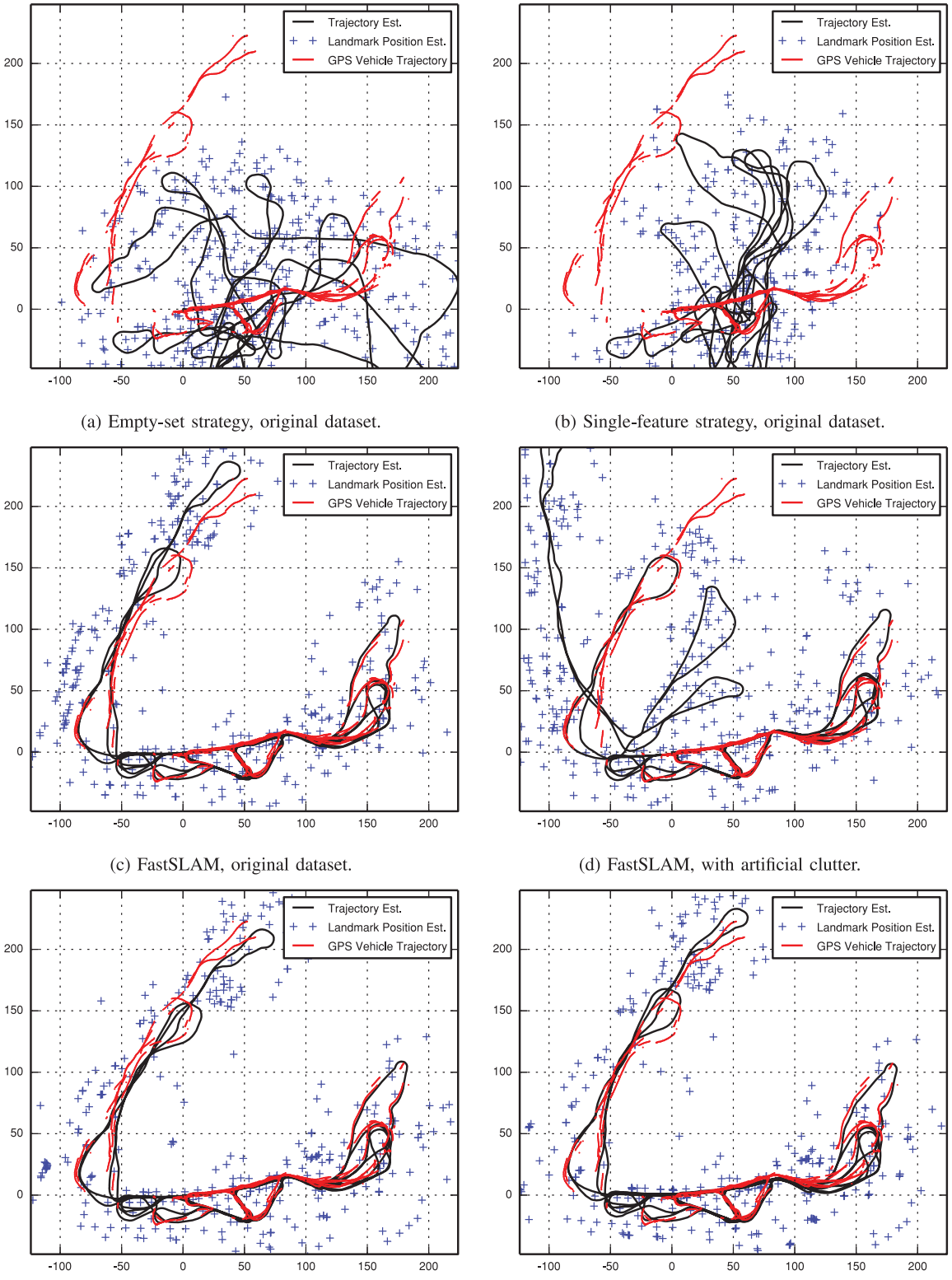
(d) Single-cluster-PHD filter. (e) Multi-feature strategy.

Fig. 4. Averaged landmark-estimate (COLA) errors under various probabilities of detection and clutter intensities.

conditions, the SC-PHD filter and the proposed multifeature strategy generated reasonable estimates, with the latter performing slightly better in terms of the trajectory estimate. Overall, the results are consistent with the simulations.

C. Parque O'Higgins Data Set

Since clutter had to be artificially added to the Victoria Park data set to create less ideal detection conditions, another data set was gathered in a more realistic high-clutter park environment. The vehicle platform,



(a) Empty-set strategy, original dataset.

(b) Single-feature strategy, original dataset.

(c) FastSLAM, original dataset.

(d) FastSLAM, with artificial clutter.

(e) Single-cluster process, with artificial clutter.

(f) Multi-feature strategy, with artificial clutter.

Fig. 5. Filtering results from Victoria Park data set, with axis units in meters.

shown in Fig. 7, was deployed in Parque O'Higgins in Santiago, Chile, where the vehicle traversed an approximate figure-eight path once and returned to its starting position. This dirt path can be seen from the satellite-image underlays in Fig. 8. GPS coverage was poor and thus the GPS track is not shown. A SICK LD-LRS1000 scanning lidar mounted on the vehicle provided planar 2-D scans of the environment, and a

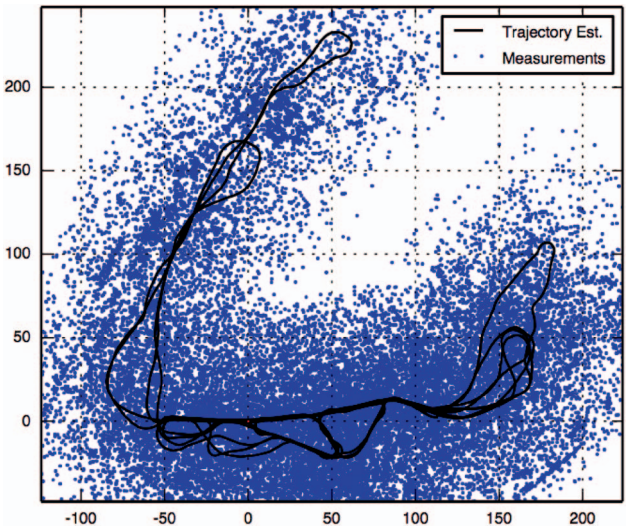


Fig. 6. All measurements (with artificial clutter) superimposed, with axis units in meters

Fig. 7. Vehicle platform and scanning laser range finder used in generating Parque O'Higgins data set.

feature extractor was used to extract nearly circular objects (such as tree trunks) from each scan [48]. These features were used as measurements and include a great deal of clutter generated by pedestrians, foliage, cars, and other moving objects. Wheel displacements were also measured via encoders. Landmark ground-truth positions were obtained from a separate data set, where point clouds from 17 static scan positions were aligned. From this, tree-trunk positions were manually identified. The results generated from the various filtering methods are shown in Fig. 8, with Fig. 8f showing all the measurements acquired over the entire trajectory superimposed onto one image.

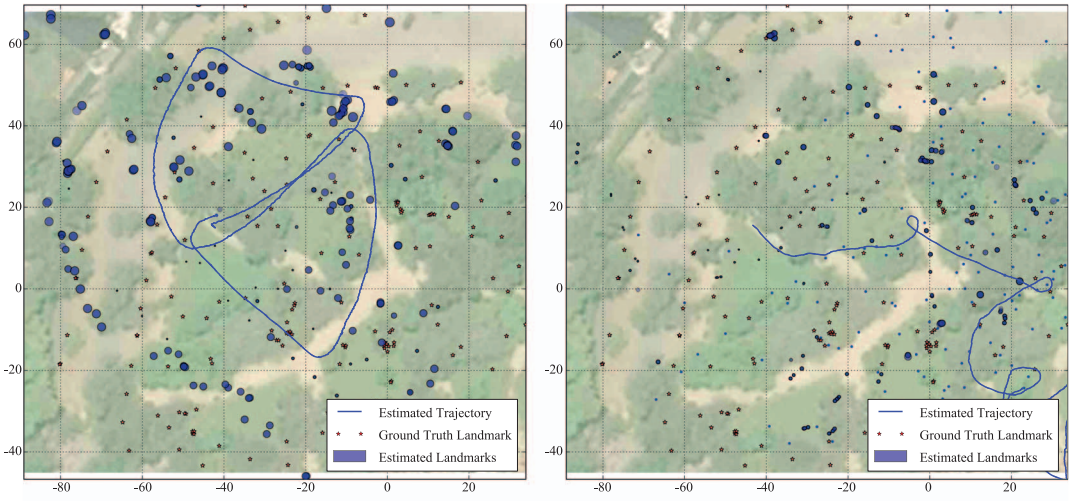
Similar to previous results for high-clutter scenarios, FastSLAM and the empty-set and single-feature strategies did not perform well. The presence of clutter caused the FastSLAM trajectory estimate to diverge shortly after commencing the traverse. The empty-set and single-feature strategies failed to assign proper weightings to particles to generate good trajectory estimates, and their trajectory estimates are off the figure-eight dirt path in Figs. 8b, 8c. The SC-PHD filter performed better in terms of its trajectory estimate, with the endpoint close to the starting position. The multifeature strategy appears to have performed equally well, with a slight improvement in the end-position estimate.

Overall, since the SC-PHD filter has been shown to perform well in many scenarios, it is postulated that the multitarget Poisson assumption made on the map is one that is reasonable. Hence, it is believed that the same assumptions made on the map densities in the proposed multifeature weighting strategy are also valid. Therefore, it is the assumption made on the map set for particle weighting that has a more pronounced effect on the filter performance, and the results in this article have shown that the empty-set and single-feature map assumptions often cause filter divergence. The multifeature strategy makes use of all the Gaussians in the map intensity, in a manner similar to the SC-PHD filter. One important point that must be remembered with the RFS formulation is that the set measurement likelihood accounts for all possible association of measurements to landmarks according to the detection statistics (which also need to be estimated). In reality, a measurement is generated from one single landmark, and there is only one true set of correspondences for all measurements. Both the proposed method in this article and the SC-PHD filter account for this set of correct correspondences, and it makes up a large portion of the calculated weight of a particle. Both methods also account for all the incorrect correspondences in their own ways, but those make up only a small portion of the particle-weight calculation, since individual measurement likelihoods for incorrect correspondences are small and close to zero in most cases. This is hypothesized as the reason that both SLAM filters (based on different derivations and assumptions) generate similar results.

D. Computation

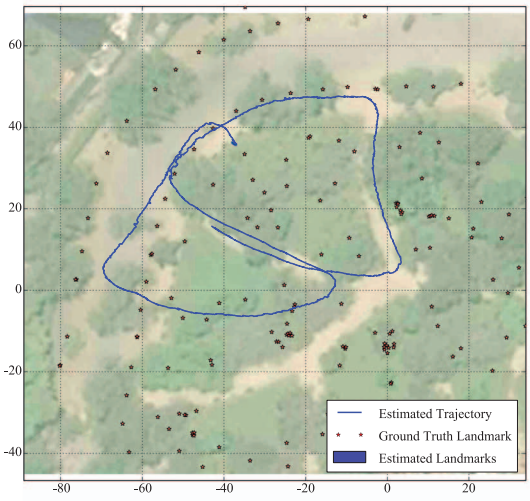
The increase in robustness and the quality of the estimates from the PHD SLAM filter with the proposed multifeature strategy come with increased computational costs. Fig. 9 shows the CPU timing analysis from processing simulation trials with different detection conditions and the Victoria Park data set. The times were measured on a single computer core of an Intel i7 2.4 GHz CPU. The times in each figure are further subdivided according to the different activities during the filtering process.

Fig. 9a shows that under low clutter and high probabilities of detection, the proposed multifeature strategy is the slowest, while FastSLAM is the fastest. Note, however, that the use of RFS filters is unnecessary under such almost-ideal conditions. Under nonideal conditions, Fig. 9b shows that the multifeature strategy is only slower than the SC-PHD filter. The single-feature strategy, empty-set strategy, and FastSLAM are slower

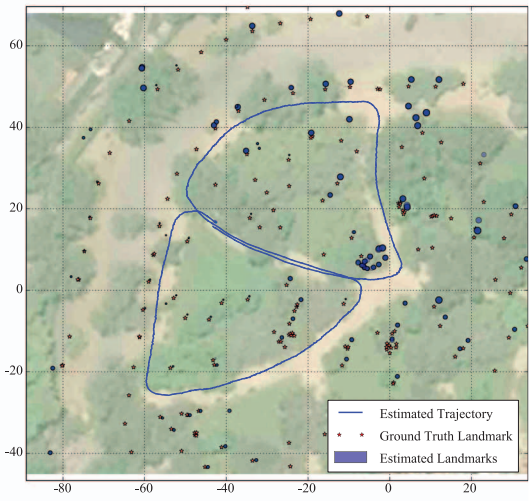


(a) FastSLAM.

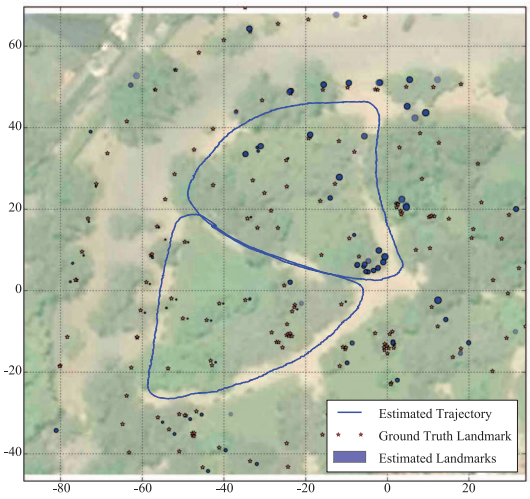
(b) Empty-set strategy.



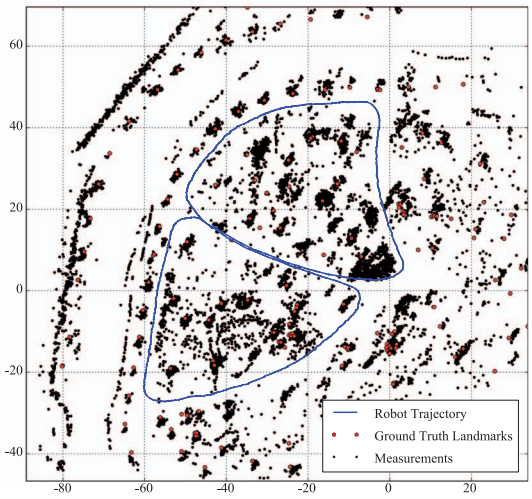
(c) Single-feature strategy.



(d) Single-cluster process



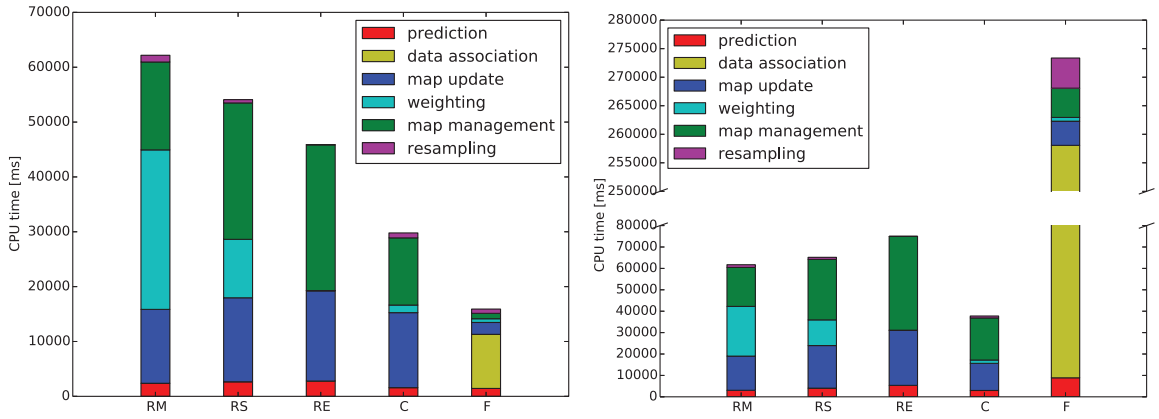
(e) Multi-feature strategy.



(f) All measurements superimposed.

Fig. 8. Filtering results from Parque O'Higgins data set, with axis units in meters.

due to estimate divergence, causing the generation of many spurious landmarks. This has an especially profound effect on computational time for FastSLAM, in which a large computational time for data association is necessary. Figs. 9c, 9d are timing data from the Victoria Park data set, and both show similar results. Although FastSLAM has the fastest time, its estimate diverged when artificial clutter was added. Note that the amount of artificial clutter



(a) Simulation ($P_D = 0.99, c = 0.0000001m^{-2}$)

(b) Simulation ($P_D = 0.5, c = 0.005m^{-2}$)

(c) Victoria Park dataset.

(d) Victoria park dataset with artificial clutter.

Fig. 9. CPU timing performance for various filters; RM = multifeature strategy, RS = single-feature strategy, RE = empty-set strategy, C = SC-PHD filter, F = FastSLAM.

added here is much less than in the simulation. Therefore, the data-association time for FastSLAM did not increase substantially as in the case of the simulations. In the Victoria Park data set, the SC-PHD filter was also faster than the proposed multifeature strategy, but it also produced less accurate estimates.

VI. CONCLUSIONS

The PHD SLAM filter is an RFS-based filter applicable to navigation and mapping problems. Previous implementations of this RB-PF-based method are not robust in regularly producing consistent estimates. This is mainly attributed to the importance-weighting calculation in the RB-PF, which uses either the empty-set or single-feature strategy. This article proposed and derived a more sophisticated multifeature strategy for importance weighting, which allows the PHD SLAM filter to be more robust against divergence and to generate higher-accuracy estimates. Methods for the efficient implementation of the proposed strategy were presented. The multifeature strategy was first validated using 2-D simulations, and comparisons were made to existing random-vector-based and RFS-based filters. It was shown that the presented particle-weighting method outperforms all of the previously established approaches in the simulations tested. The proposed strategy was also validated using real experimental data. Similar to the simulation results, PHD SLAM filtering with the multifeature strategy performed better than existing methods. Computational-timing analysis revealed that the increase in estimation performance incurs greater computational cost, but only if the other filters also converge. Overall, the proposed method is still practical and computationally feasible. An implementation of the proposed approach and all the filters tested in this article, as well as their configurations and settings, can be found in an open-source C++ library at https://github.com/kykleung/RFS-SLAM.git.

REFERENCES

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

[2] Mahler, R. P. S.
Statistical Multisource-Multitarget Information Fusion.
Norwood, MA: Artech House, 2007.

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

[4] Mullane, J., Vo, B.-N., Adams, M., and Vo, B.-T.
Random Finite Sets for Robot Mapping and SLAM: New Concepts in Autonomous Robotic Map Representations.
Berlin: Springer-Verlag, 2011.

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

[6] Lundquist, C., Hammarstrand, L., and Gustafsson, F.
Road intensity based mapping using radar measurements with a probability hypothesis density filter.
IEEE Transactions on Signal Processing, **59**, 4 (Apr. 2011), 1397–1408.

[7] Moratuwage, D., Vo, B.-N., and Wang, D.
A hierarchical approach to the multi-vehicle SLAM problem.
In *15th IEEE International Conference on Information Fusion*, Singapore, July 2012.

[8] Clark, D., and Godsill, S.
Group target tracking with the Gaussian mixture probability hypothesis density filter.
In *3rd International Conference on Intelligent Sensors, Sensor Networks and Information*, Melbourne, Australia, Dec. 2007, 149–154.

[9] Hoseinnezhad, R., Vo, B.-N., Vo, B.-T., and Suter, D.
Bayesian integration of audio and visual information for multi-target tracking using a CB-MeMBer filter.
In *IEEE International Conference on Acoustics, Speech and Signal Processing*, Prague, Czech Republic, May 2011, 2300–2303.

[10] Hoseinnezhad, R., Vo, B.-N., and Vo, B.-T.
Visual tracking in background subtracted image sequences via multi-Bernoulli filtering.
IEEE Transactions on Signal Processing, **61**, 2 (Jan. 2013), 392–397.

[11] Vo, B.-N., and Ma, W.-K.
The Gaussian mixture probability hypothesis density filter.
IEEE Transactions on Signal Processing, **54**, 11 (Nov. 2006), 4091–4104.

[12] Vo, B.-T., Vo, B.-N., and Cantoni, A.
Analytic implementations of the cardinalized probability hypothesis density filter.
IEEE Transactions on Signal Processing, **55**, 7 (July 2007), 3553–3567.

[13] Vo, B.-T., Vo, B.-N., and Cantoni, A.
The cardinality balanced multi-target multi-Bernoulli filter and its implementations.
IEEE Transactions on Signal Processing, **57**, 2 (Feb. 2009), 409–423.

[14] Vo, B.-T., Vo, B.-N., Hoseinnezhad, R., and Mahler, R. P. S.
Robust multi-Bernoulli filtering.
IEEE Journal of Selected Topics in Signal Processing, **7**, 3 (June 2013), 399–409.

[15] Kaess, M., Ranganathan, A., and Dellaert, F.
iSAM: Incremental smoothing and mapping.
IEEE Transactions on Robotics, **24**, 6 (Dec. 2008), 1365–1378.

[16] Strasdat, H., Montiel, J. M. M., and Davison, A. J.
Visual SLAM: Why filter?
Image and Vision Computing, **30**, 2 (2012), 65–77.

[17] Agarwal, P., Tipaldi, G. D., Spinello, L., Stachniss, C., and Burgard, W.
Robust map optimization using dynamic covariance scaling.
In *IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, May 2013, 62–69.

[18] Latif, Y., Cadena, C., and Neira, J.
Robust loop closing over time for pose graph SLAM.
The International Journal of Robotics Research, **32**, 14 (2013), 1611–1626.

[19] Folkesson, J., and Christensen, H.
Graphical SLAM—A self-correcting map.
In *IEEE International Conference on Robotics and Automation*, New Orleans, LA, May 2004, **1**, 383–390.

[20] Dezert, J., and Bar-Shalom, Y.
Joint probabilistic data association for autonomous navigation.
IEEE Transactions on Aerospace and Electronic Systems, **29**, 4 (Oct. 1993), 1275–1286.

[21] Fortmann, T., Bar-Shalom, Y., and Scheffe, M.
Sonar tracking of multiple targets using joint probabilistic data association.
IEEE Journal of Oceanic Engineering, **8**, 3 (July. 1983), 173–184.

[22] Vo, B.-T., and Vo, B.-N.
Labeled random finite sets and multi-object conjugate priors.
IEEE Transactions on Signal Processing, **61**, 13 (July 2013), 3460–3475.

[23] Reuter, S., Vo, B.-T., Vo, B.-N., and Dietmayer, K.
The labeled multi-Bernoulli filter.
IEEE Transactions on Signal Processing, **62**, 12 (June 2014), 3246–3260.

[24] Deusch, H., Reuter, S., and Dietmayer, K.
The labeled multi-Bernoulli SLAM filter.
IEEE Signal Processing Letters, **22**, 10 (Oct. 2015), 1561–1565.

[25] Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B.
FastSLAM:
A factored solution to the simultaneous localization and mapping problem.
In *National Conference on Artificial Intelligence*, Edmonton, Canada, 2002, 593–598.

[26] Gordon, N. J., Salmond, D. J., and Smith, A. F. M.
Novel approach to nonlinear/non-Gaussian Bayesian state estimation.
IEE Proceedings F—Radar and Signal Processing, **140**, 2 (Apr. 1993), 107–113.

[27] Leung, K. Y. K., Inostroza, F., and Adams, M.
An improved weighting strategy for Rao–Blackwellized probability hypothesis density simultaneous localization and mapping.
In *International Conference on Control, Automation and Information Sciences*, Nha Trang, Vietnam, Nov. 2013.

[28] Leung, K. Y. K., Inostroza, F., and Adams, M.
Evaluating set measurement likelihoods in random-finite-set slam.
In *17th International Conference on Information Fusion*, Salamanca, Spain, July 2014.

[29] Swain, A., and Clark, D. E.
First-moment filters for spatial independent cluster processes.
Proceedings of SPIE, **7697** (2010), 79670I.

[30] Swain, A., and Clark, D.
The single-group PHD filter: An analytic solution.
In *Proceedings of the 14th International Conference on Information Fusion*, Chicago, IL, July 2011.

[31] Jazwinski, A. H.
Stochastic Processes and Filtering Theory. New York: Academic Press Inc., 1970.

[32] Thrun, S., Burgard, W., and Fox, D.
Probabilistic Robotics, Cambridge, MA: MIT Press, 2005.

[33] Reuter, S.
Multi-object tracking using random finite sets.
Ph.D. dissertation, Universität Ulm, Germany, 2014.

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

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

[36] Ryser, H. J.

Combinatorial Mathematics. Washington, DC: Mathematical Association of America, 1963.

[37] Atanasov, N., Zhu, M., Daniilidis, K., and Pappas, G. J.
Semantic localization via the matrix permanent.
In *Robotics: Science and Systems*, Berkeley, CA, July 2014.

[38] Jerrum, M., Sinclair, A., and Vigoda, E.
A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries.
Journal of the ACM, **51**, 4 (2004), 671–697.

[39] Huang, B., and Jebara, T.
Approximating the permanent with belief propagation. (2009). [Online]. https://arxiv.org/abs/0908.1769.

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

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

[42] Nieto, J., Guivant, J., Nebot, E., and Thrun, S.
Real time data association for FastSLAM.
In *IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, Sept. 2003, **1**, 412–418.

[43] Kuhn, H. W.
The Hungarian method for the assignment problem.
Naval Research Logistics Quarterly, **2**, 1–2 (1955), 83–97.

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

[45] Barrios, P., Naqvi, G., Adams, M., Leung, K., and Inostroza, F.
The cardinalized optimal linear assignment (COLA) metric for multi-object error evaluation.
In *18th International Conference on Information Fusion*, Washington, DC, July 2015.

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

[47] Guivant, J., Nebot, E., and Baiker, S.
Autonomous navigation and map building using laser range sensors in outdoor applications.
Journal of Robotic Systems, **17**, 10 (2000), 565–583.

[48] Inostroza, F., Leung, K. Y. K., and Adams, M.
Semantic feature detection statistics in set based simultaneous localization and mapping.
In *17th International Conference on Information Fusion*, Salamanca, Spain, July 2014.

Keith Y. K. Leung is an IEEE member. He holds a B.A.Sc. degree (2005) and an M.A.Sc. degree (2007) from the University of Waterloo in mechanical engineering. He received his Ph.D. degree (2012) from the University of Toronto, where he was a member of the Autonomous Space Robotics Lab and the Flight Systems Control Lab at the Institute for Aerospace Studies. From 2012 to 2016, he was a postdoctoral researcher at the Advanced Mining Technology Center and Department of Electrical Engineering at the Universidad de Chile. His areas of expertise are state estimation, robot navigation, and simultaneous localization and mapping. He is currently an expert in vision processing, perception, and robotics with Applanix Corporation (Trimble).

Felipe Inostroza is an IEEE student member and a doctoral student in the Department of Electrical Engineering, Universidad de Chile. He obtained his first degree in engineering science at the Universidad de Chile in 2013 and a master's degree in electrical engineering in 2015 at the same university. His research interests focus on robotics in general, and simultaneous localization and mapping (SLAM) in particular. He is currently studying the effects of including detection statistics into the SLAM problem.

Martin Adams is a professor of electrical engineering at the Department of Electrical Engineering, Universidad de Chile and is currently chair of the IEEE Chilean Robotics and Automation Chapter. He is also a principal investigator in the Advanced Mining Technology Center. He obtained his first degree in engineering science at the University of Oxford, United Kingdom, in 1988, and he continued to study at the Robotics Research Group, University of Oxford, for a D.Phil. degree, which he received in 1992. He continued his research in autonomous robot navigation as a project leader and part-time lecturer at the Institute of Robotics, Swiss Federal Institute of Technology, Zurich, Switzerland. He was employed as a guest professor and taught control theory in St. Gallen (Switzerland) from 1994 to 1995. From 1996 to 2000, he served as a senior research scientist in robotics and control, in the field of semiconductor assembly automation, at the European Semiconductor Equipment Center, Switzerland. From 2000 to 2010, he was an associate professor at the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research work focuses on autonomous robot navigation, space situational awareness, sensing, data sensor-interpretation, and control, and he has published many technical papers in these fields. He has been the principal investigator and leader of many robotics and industrially sponsored projects, coordinating researchers from local industries and local and overseas universities, and has served as associate editor on various journal and conference editorial boards.