New Concepts in Map Estimation

SLAM Gets a PHD

© ARTVILLE

By Martin Adams, Ba-Ngu Vo, Ronald Mahler, and John Mullane

Having been referred to as the Holy Grail of autonomous robotics research, simultaneous localization and mapping (SLAM) lies at the core of most the autonomous robotic applications [1]. This article explains the recent advances in the representations of robotic sensor measurements and the map itself as well as their consequences on the robustness of SLAM. Fundamentally, the concept of a set-based measurement and map state representation allows all of the measurement information, spatial and detection, to be incorporated into joint Bayesian SLAM frameworks. Modeling measurements and the map state as random finite sets (RFSs) rather than the traditionally adopted random vectors is not merely a triviality of notation. It will be demonstrated that a set-based framework circumvents the necessity for any fragile data association and map management heuristics, which are necessary in vector-based solutions.

The article focuses on an implementation of the simplest Bayesian set-based estimator—the probability hypothesis density (PHD) filter—and its application to SLAM. Using particles to represent hypothesized vehicle trajectories and a Rao–Blackwellized (RB) PHD filter for each particle's map, estimates of the location and, equally importantly, the number of features that have passed through the field of view (FoV) of a vehicle's sensor(s) are estimated.

Vector-based SLAM solutions do not estimate this number of features and instead rely on map management and data association heuristics, which are implemented outside of the Bayesian (e.g., Kalman, unscented Kalman, particle filter) estimator to determine which features to update. In the parallel but closely related field of target tracking, recent research has shown that

Digital Object Identifier 10.1109/MRA.2014.2304111
Date of publication: 9 May 2014

1070-9932/14/$31.00©2014IEEE

estimating the number of targets, as well as their spatial locations, is central to any tracking problem [2]. A philosophy often adopted in the SLAM community is that it is not important to estimate all of the features that have passed through the FoV of the robot's sensor(s), provided enough can be estimated to achieve good robot localization results. In response, the reader is briefly diverted to the results section "Comparisons of PHD and Vector-Based Slam in a Marine Environment," where many false alarms (false positives) are apparent. Due to these false alarms, a vector-based multihypothesis (MH) factorized solution to SLAM (FastSLAM) implementation estimates the SLAM map to contain between 30 and 35 point features when, in fact, only 22 had entered the FoV of the sensor. How can such a SLAM estimate be trusted? Over (or under) estimating the feature number through map management and (false) data association decisions can only impair the vehicle location and map estimates in their entirety. This problem was noted in [3], where a useful metric was defined that could quantify feature-based (FB) SLAM map quality when represented as an RFS in terms of number and spatial estimation errors.

The necessity for an RFS-based map state has been noted in the recent literature, with other researchers addressing SLAM with dynamic targets [4], [5] and multirobot solutions [6]. To estimate feature number in a principled manner, the RFS Bayes SLAM filter incorporates estimates of the probabilities of detection and false alarm of feature detectors. This could be interpreted in a negative sense in that set-based SLAM methods appear to need two extra parameters (probabilities of detection and false alarm) that are not considered necessary in vector-based SLAM. It should be noted, however, that in vector-based SLAM, these values are (possibly inadvertently) used, since in the Bayesian update, the unity probability of detection and zero probability of false alarm are assumed as a result of the complete reliance on the external feature management and association heuristics. Moreover, recent work has shown that the probability of detection and false alarm rates can also be estimated along with the map.

To provide motivation for set-based SLAM methods, this article begins by demonstrating the true uncertainty, which results when well-known vision, laser range finder (LRF), and short-range radar-based feature detectors are applied to real data. The randomness in the number as well as the location of detections is highlighted. This leads to the remodeling of the SLAM concept with the RFSs. Set-based estimation requires the use of finite set statistics (FISSTs), the concepts of which are also introduced, together with the simplest FISST, the PHD filter. A full diagrammatic implementation of PHD-based SLAM is given with comparative results with a state-of-the-art vector-based SLAM method in a challenging coastal environment. (The coastal SLAM results were previously published in [3], [7], and [9].)

Sensing and Detection Errors

In FB-SLAM, detection algorithms are used in conjunction with various sensors. Using examples showing state-of-the-art feature detectors with a millimeter wave (MMW) radar, LRF, and a camera, common sensing and detection errors will be highlighted. This will motivate the need for a reevaluation of stochastic FB-SLAM concepts, which jointly consider sensing and detection errors. Consider the sensor suite used in the acquisition of the data together with the robotic platform, shown in Figure 1. The robotic platform is the Clearpath Husky A200 skid steer vehicle, used for sensing and mapping experiments in mining environments. The radar is a high-speed scanning, pencil-beam (narrow beam width of approximately 1.8°), MMW radar from Acumine Pty. Ltd., providing one scan per second with a range resolution of 50 cm up to a maximum range of 200 m. The LRF is the Sick LD-LRS1000, capable of continuous 360° scanning and range measurements between 2.5 and 80 m. Images are provided by the monocular camera from an Xbox Kinect camera system. Figure 2(a)–(f) shows two sets of detected features based on radar point features, extracted with a constant false alarm rate (CFAR) processor laser-based, random sampling consensus (RANSAC) line features and the visual speeded-up robust features (SURF) algorithm. The purpose of the two columns is to show the random nature, not only in the spatial locations of the detected features but in the number and nature of the detections themselves.

Feature Detection with Radar

Radars can give received power plan position indicator images, which display multiple received power values at discrete points in space. These power values are corrupted with random receiver noise, and the challenge is to detect those power values emanating from true objects within an environment. In Figure 2(a), radar detections are based on the application of a commonly used ordered statistics (OS)-CFAR

Figure 1. The Husky A200 with a camera, an LRF, and an MMW radar.

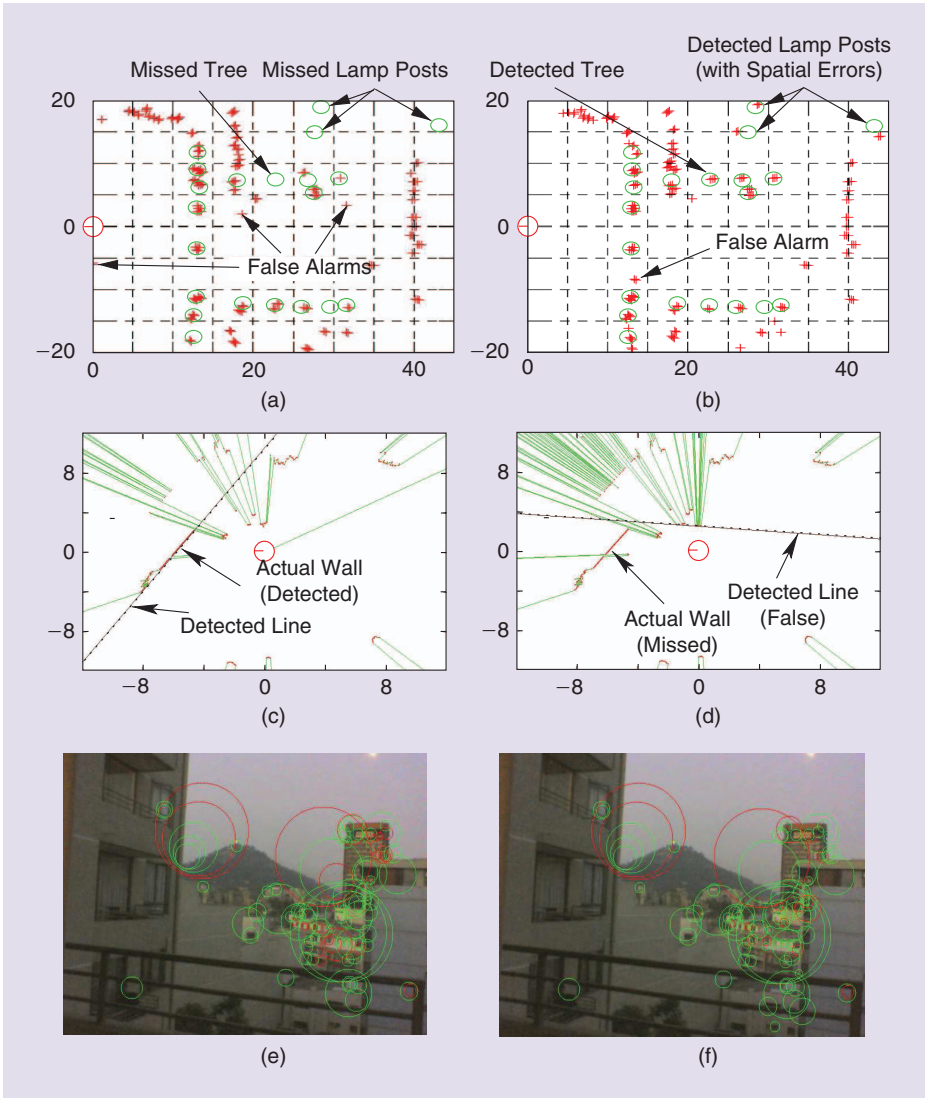

Figure 2. The random nature of detections, based on a radar CFAR detector, line-based RANSAC, and the SURF detection algorithm: (a) and (b) radar point detections based on an OS–CFAR detector, (c) and (d) laser range data with a single RANSAC line detector, and (e) and (f) visual SURF features.

detector applied to the received power values [8], [9]. All power values exceeding the OS-CFAR threshold are shown as crosses, together with the superimposed ground truth location of objects (trees, lamp posts, and so forth) shown as ellipses. The two experiments were carried out in the same environment under the same environmental conditions. It is evident that, due to the random power noise in each radar scan, some of the features detected in one image are missed in the other, as is the random presence of false alarms. Under an OS-CFAR detector, the probabilities of false alarms P_{fa} and detection P_D can be quantified under certain noise distribution assumptions.

RANSAC Line Detection with LRFs

Figure 2(c) and (d) shows data samples from a 360° laser range scan recorded in an environment in which a single wall was present (labeled). The RANSAC algorithm was applied to extract line segments from the data [10]. Due to RANSAC's random nature in selecting its initial model inlier points, Figure 2(c) shows the line correctly detected, whereas Figure 2(d) shows that the true wall is missed and a false line is detected. Similar to the radar detection concepts, a possible detection threshold for RANSAC can be set, again yielding an approximate P_{fa} and, depending on the number of data inliers within the actual line, P_D.

Visual Feature Detection with SURF

In the case of the camera, a well-known blob detector called the *SURF detector* is applied in Figure 2(e) and (f). This detector searches for local maxima in the determinant of Hessian (DoH) applied to a two-dimensional (2-D) array of image pixels at various scales [11]. Such a maxima is an indication of a distinctive region in the image space corresponding to a possible feature likely to be detected from other locations. The SURF descriptors are shown as circles, the area of which correspond to the scale at which SURF determined the maximum DoH, at the pixel center of that circle. In Figure 2(f) the ambient light level has reduced slightly. In low-light environments, the camera increases its gain, which also amplifies the noise. In any blob detector (e.g., SURF), this variation in pixel intensities introduces randomness into the detection process. The green circles in Figure 2 correspond to SURF features, which were detected in over 90% of the 300 recorded images of the same scene, and are therefore taken to approximate ground truth SURF features. In this case, varying the light conditions in a scene so that the DoH varies can give approximate but meaningful P_{fa} and P_D values for the SURF features.

General Sensing Errors

Although spatial errors occur in terms of the estimated location of the features, of far greater importance is the fact that the correctness of the detections themselves is questionable. Consequently, the realistic uncertainty associated with measurement data includes detection (or feature

extraction) uncertainty, spurious measurements, and spatial uncertainty. The ability to account for all of these errors in a joint and principled manner has a huge impact on FB-SLAM and provides the motivation for remodeling FB-SLAM under a set-based framework.

Remodeling SLAM to Account for Detection and Spatial Uncertainty

Feature Measurements Modeled as RFSs

In the visual, laser-, and radar-based measurement examples of Figure 2, it is clear that, in general, a measurement consists of a random number of detections, each with spatial uncertainty. This implies that the result of any realistic feature detection process is a measurement that is not a vector. For example, for the point feature radar detections in Figure 2(a), resulting in measured range and bearing values r and θ, respectively, let the collected detections be $z^1 = [r^1\ \theta^1]^T, \ldots, z^\jmath = [r^\jmath\ \theta^\jmath]^T$, which could be false or otherwise. A vector model for this measurement $[z^1 \ldots z^\jmath]^T$ has a fixed dimension $\jmath$, but the number of detections can vary from zero to some arbitrarily large number due to the possibility of missed detections and multiple false alarms. Another problem is that the components of the vector $z^1, \ldots, z^\jmath$ have a fixed order, but the actual detections $z^1, \ldots, z^\jmath$ have no inherent physical order. Therefore, a more precise model of the measurement is a finite observation set, which, by definition, has no fixed order and has elements comprising the individual detections

$$\mathcal{Z} = \{z^1, \ldots, z^\jmath\} = \{[r^1\ \theta^1]^T, \ldots, [r^\jmath\ \theta^\jmath]^T\}. \tag{1}$$

Therefore, at any instant, a sensor can be considered to collect a finite set $\mathcal{Z} = \{z^1, \cdots, z^\jmath\}$ of measurements $z^1, \ldots, z^\jmath$ from a measurement space $\mathcal{Z}_0$ as follows:

$$\begin{aligned}
\mathcal{Z} &= \emptyset && \text{(no features detected)}\\
\mathcal{Z} &= \{z^1\} && \text{(one feature } z^1 \text{ detected)}\\
\mathcal{Z} &= \{z^1, z^2\} && \text{(two features } z^1, z^2 \text{ detected)}\\
&\vdots && \vdots\\
\mathcal{Z} &= \{z^1, \ldots, z^\jmath\} && (\jmath \text{ features } z^1, \ldots, z^\jmath \text{ detected}).
\end{aligned} \tag{2}$$

Since the number of feature detections in $\mathcal{Z}$ as well as the values of the individual detections z^i are random in nature, $\mathcal{Z}$ is referred to as an RFS.

Vehicle State Modeled as a Random Vector

In SLAM formulations, the vehicle's current pose state is typically modeled as a time-varying vector X_k, containing its 2-D x_k, y_k position and its orientation ϕ_k at time k. The three-dimensional (3-D) vehicle states are also possible, containing the six-degrees-of-freedom state variables, x_k, y_k, z_k, as well as the vehicle's roll pitch and yaw angles and possibly more states containing velocity, acceleration, and higher-order variables. Irrespective of the complexity of the chosen vehicle state, its dimensions are fixed as time progresses, and the order of the variables in the vector remain the same, i.e., unlike the map feature estimates and measurements, in single-robot SLAM, the state related to the robot's position has no dimensional uncertainty. Therefore, the robot vehicle state is adequately modeled as a random vector, which, in its general form, is propagated forward in discrete time according to the state transition equation

$$X_k = f^{\text{veh}}(X_{k-1}, U_{k-1}, \nu_{k-1}), \tag{3}$$

where $f^{\text{veh}}(\cdot)$ is the (generally nonlinear) state transition function, a specific example of which will be demonstrated in the marine application in (11), U_{k-1} are any deterministic inputs applied to the vehicle at time $k-1$, and ν_{k-1} models the assumed zero mean random noise modeling the uncertainty of the function $f^{\text{veh}}(\cdot)$.

Map State Space Modeled as an RFS

The number of features in the map state can vary from zero to some arbitrarily large number. Ideally, it should grow monotonically as features enter the FoV of the sensor(s). This further justifies the need for a set-based map representation containing individual feature states as follows:

$$\begin{aligned}
\mathcal{M} &= \emptyset && \text{(no features present)}\\
\mathcal{M} &= \{m^1\} && \text{(one feature with state } m^1 \text{ present)}\\
\mathcal{M} &= \{m^1, m^2\} && \text{(two features } m^1 \neq m^2 \text{ present)}\\
&\vdots && \vdots\\
\mathcal{M} &= \{m^1, \ldots, m^m\} && (m \text{ features } m^1 \neq \cdots \neq m^m \text{ present}).
\end{aligned}$$

Relating RFS Measurements to the SLAM State

To encapsulate detection uncertainty as well as spatial measurement noise, the detected features from a vehicle with pose X_k at time k can be mathematically modeled by an RFS $\mathcal{Z}_k$. This is formed by the union of a set of features expected to be generated under the current map estimate and a set of false detections. Importantly, each set encapsulates the aforementioned detection and spatial uncertainties, and hence

$$\underbrace{\mathcal{Z}_k}_{\substack{\text{All}\\ \text{Features}}} = \bigcup_{m \in \mathcal{M}_k} \underbrace{\mathcal{D}_k(m, X_k)}_{\substack{\text{Expected}\\ \text{Features}}} \cup \underbrace{\mathcal{C}_k(X_k)}_{\substack{\text{False}\\ \text{Features}}}, \tag{4}$$

where $\mathcal{D}_k(m, X_k)$ is the RFS of measurements generated by a feature at location m and $\mathcal{C}_k(X_k)$ is the RFS of the spurious measurements at time k, which may depend on the vehicle pose X_k. $\mathcal{Z}_k = \{z_k^1, z_k^2, \ldots, z_k^{\jmath_k}\}$ consists of a random number, $\jmath_k$ of spatial measurements z_k^i, whose order of appearance has no physical significance with respect to the estimated map of features. For each feature, $m \in \mathcal{M}_k$ and $z_k^i \in \mathcal{Z}_k$,

$$\mathcal{D}_k(m,X_k)=\{z_k^i\} \tag{5}$$

with probability density $P_D(m,X_k)g(z_k^i\,|\,m,X_k)$ and $\mathcal{D}_k(X_k,m)=\emptyset$ with probability $1-P_D(m,X_k)$. $P_D(m,X_k)$ is the probability of detection of feature m. This could depend also on the vehicle pose X_k and the feature's coordinates m as well as other characteristics of the feature in question. Here, $g(z_k^i\,|\,m,X_k)$ represents the sensor's likelihood of detecting z_k^i, given the spatial information m and X_k. In the robotics literature, this is commonly introduced as

$$z_k^i=h^{\text{spatial}}(X_k,m)+\text{spatial noise terms}, \tag{6}$$

where $h^{\text{spatial}}(\cdot)$ is the spatial function relating the vehicle's coordinates and the feature's parameters. (In the case of point features, these would be its coordinates [1].)

Note the principled association of both detection ($P_D(m,X_k)$) and spatial ($g(z_k^i\,|\,m,X_k)$) uncertainties with the RFS $\mathcal{D}_k(m,X_k)$. The spatial uncertainty term $g(z_k^i\,|\,m,X_k)$ would reflect the statistics of the appropriate spatial noise terms in (6). Similarly, the spurious measurement rate statistics corresponding to $C_k(X_k)$ are typically a priori assigned based on an expected P_{fa}, available from the chosen detection method.

RFS Map Prediction Model

Let $\mathcal{M}$ be the RFS representing the entire unknown and unexplored static map. The explored map $\mathcal{M}_{k-1}$ then evolves in time according to

$$\underbrace{\mathcal{M}_k}_{\substack{\text{Current}\\ \text{Map Set}}}=\underbrace{\mathcal{M}_{k-1}}_{\substack{\text{Previous}\\ \text{Map Set}}}\cup\Big(\underbrace{\text{FoV}(X_k)}_{\substack{\text{Sensor(s)}\\ \text{FoV}}}\cap\bar{\mathcal{M}}_{k-1}\Big), \tag{7}$$

where $\bar{\mathcal{M}}_{k-1}$ is the set of features not in $\mathcal{M}_{k-1}$. This map state transition equation describes the fact that the set of map features grows monotonically as a vehicle's sensor(s) FoV covers more of the unexplored environment.

Bayesian SLAM with a Finite Set Feature Map

An RFS can be represented by a discrete distribution that characterizes the time-varying number of features, $\mathfrak{m}_k$, in the set and a family of joint distributions, which characterize their spatial distributions. Therefore, an RFS (FB)-SLAM state can be described by its probability density function (PDF) $p_{k|k}(X^k,\mathcal{M}_k=\{m^1,m^2,\cdots,m^{\mathfrak{m}_k}\}\,|\,\mathcal{Z}^k,U^{k-1},X_0)$, where $\mathcal{Z}^k$ represents the set of all measurements from time 0 to k, U^{k-1} represents all inputs from time 0 to $k-1$, and X_0 is the initial pose of the vehicle.

Similarly to the FastSLAM concept [12], the PHD-SLAM joint posterior can be factorized using an RB implementation, in which the map is represented as a conditional PDF, conditioned on an entire vehicle trajectory X^k [13]. Under an RB implementation, the vehicle trajectory X^k is represented as vector particles $(X^k)^{(i)}$, each of which maintain their own set-based map estimate $p_{k|k}^{(i)}(\mathcal{M}_k\,|\,\mathcal{Z}^k,(X^k)^{(i)})$. The Bayesian recursion of the map, per trajectory particle, is then

$$\underbrace{p_{k|k}(\mathcal{M}_k\,|\,\mathcal{Z}^k,(X^k)^{(i)})}_{\substack{\text{Map posterior conditioned}\\ \text{on trajectory particle } i}}\propto\underbrace{p_{k|k-1}(\mathcal{M}_k\,|\,\mathcal{Z}^{k-1},(X^k)^{(i)})}_{\substack{\text{predicted map conditioned}\\ \text{on trajectory particle } i}}\times\underbrace{g(\mathcal{Z}_k\,|\,\mathcal{M}_k,(X_k)^{(i)})}_{\substack{\text{Measurement}\\ \text{likelihood}}}. \tag{8}$$

It is important to note that the measurement likelihood $g(\mathcal{Z}_k\,|\,\mathcal{M}_k,(X_k)^{(i)})$, and the predicted SLAM state $p_{k|k-1}(X^k,\mathcal{M}_k\,|\,\mathcal{Z}^{k-1},U^{k-1},X_0)$, are FISST densities representing the RFSs, which, contrary to the vector-based implementation of Bayes theorem, do not have to be of compatible dimensions. This means that the feature number in the predicted SLAM map does not have to equal that corresponding to the measurement likelihood, eliminating the need for fragile map management routines. By adopting an RFS map model, integrating over the map becomes a set integral, i.e., an integral, that sums over not only all spatial possibilities of the map state (as is the case of a vector integral) but over its cardinality possibilities also. This feature map recursion therefore encapsulates the inherent feature number uncertainty of the map, introduced by detection uncertainty, spurious measurements and vehicle maneuvers, as well as the feature location uncertainty introduced by spatial measurement noise. Features are not rigidly placed in a map vector nor are measurements simply a direct function of the map state, due to the explicit modeling of false measurements.

Finite Set Statistics

FISSTs are necessary to implement the map component of the factored SLAM solution in (8). As in the set-based measurements and map representations, the ith trajectory map posterior $p_{k|k}(\mathcal{M}_k\,|\,\mathcal{Z}^k,(X^k)^{(i)})$ is a multifeature PDF, which encompasses all the possibilities:

$$\begin{aligned}&p_{k|k}(\mathcal{M}_k\,|\,\mathcal{Z}^k,(X^k)^{(i)})\longrightarrow p_{k|k}(\emptyset\,|\,\mathcal{Z}^k,(X^k)^{(i)});\\ &\quad p_{k|k}(\{m^1\}\,|\,\mathcal{Z}^k,(X^k)^{(i)});\ p_{k|k}(\{m^1,m^2\}\,|\,\mathcal{Z}^k,(X^k)^{(i)});\\ &\quad\cdots\ p_{k|k}(\{m^1,\ldots,m^{\mathfrak{m}_k}\}\,|\,\mathcal{Z}^k,(X^k)^{(i)}).\end{aligned}$$

As in vector-based SLAM, as a consequence of the above multifeature PDF, the general RFS Bayes recursion of (8) is mathematically intractable since multiple integrals on the space of features would be required.

The PHD Filter

An approximation to set-based estimation, which is tractable, is to exploit the physical intuition of the first moment of an RFS, known as its PHD, v_k. Also known as the *intensity function*, the PHD at a point gives the density of the expected number of features occurring at that point. The strategy of

this filter is opposite to that of conventional vector-based approaches, which require external methods to fix the number of map features and then attempt to optimize their location estimates. The PHD filter tracks only the overall feature map behavior and then attempts to detect and track individual features as new measurements are made.

A PHD function v_k must have the following two properties:

- The mass (integral of the density over the volume) of the PHD within a given spatial region S gives the expected number of features in S.
- As a consequence, the peaks of the PHD indicate locations with high probability of feature existence.

RFS SLAM with the PHD Filter

The aforementioned RB SLAM implementation, which uses a PHD approximation for the set-based map, is adopted, conditioned on the vehicle trajectory. A weighted sum-of-Gaussians is used as the PHD function, and the mapping recursion is approximated by a GM-PHD filter. The trajectory recursion adopts a particle filter [3]. This is referred to as *RB-PHD-SLAM*.

The map is predicted with a GM form of the PHD predictor, the implementation of which will be explained in the next section.

$$\underbrace{v_{k|k-1}(m \mid X^k)}_{\text{Predicted PHD}} = \underbrace{v_{k-1|k-1}(m \mid X^{k-1})}_{\text{Prior PHD}} + \underbrace{b(m \mid X_k)}_{\text{Birth PHD}}, \tag{9}$$

where $v_{k-1|k-1}(m \mid X^{k-1})$ is the previous GM estimate of the PHD, $v_{k|k-1}(m \mid X^k)$ is its prediction at time k, and $b(m \mid X_k)$ is the GM-PHD of the birth RFS, used to model the new features predicted to enter the FoV of the vehicle's sensor(s), i.e., the bracketed term on the right hand side (RHS) of (7). Here, $b(m \mid X_k)$ is similar to the proposal function used in particle filters and is used to give some a priori information to the filter about where features are likely to appear in the map. In SLAM, with no a priori information, $b(m \mid X_k)$ may be uniformly distributed in a noninformative manner about the space of features. However, in this article, the feature birth proposal at time k is chosen to be a GM containing $J_{b,k}$ Gaussian components, representing the set of measurements at time $k-1$, $\mathcal{Z}_{k-1}$ [7].

The PHD corrector equation is [2]

$$\underbrace{v_{k|k}(m \mid X^k)}_{\text{Posterior PHD}} = \underbrace{v_{k|k-1}(m \mid X^k)(1 - P_D(m \mid X_k))}_{\text{All predicted features weighted by their probabilities of missed detection}} + \underbrace{v_{k|k-1}(m \mid X^k) \sum_{z \in \mathcal{Z}_k} \frac{\Lambda(m \mid X_k)}{c_k(z) + \int_{\mathcal{M}_k} \Lambda(\xi \mid X_k) v_{k|k-1}(\xi \mid X^k)\, d\xi}}_{\text{All predicted features, updated by the spatial locations of all the new measurements, and their probabilities of detection}}, \tag{10}$$

where $v_{k|k}(m \mid X^k)$ is the new GM estimate of the PHD at time k, $\Lambda(m \mid X_k) = P_D(m \mid X_k) g_k(z \mid m, X_k)$, and

$P_D(m \mid X_k)$ = the probability of detecting a land mark at m, from vehicle pose X_k.

$c_k(z)$ = PHD of the clutter RFS C_k in (4) at time k.

Implementing the RB-PHD-SLAM Filter

The PHD-SLAM density at time $k-1$ can be represented by a set of N particles, each accompanied by their own GM-PHDs representing their belief of the map. The RB-PHD-SLAM filter then proceeds to approximate the posterior density by a new set of weighted particles according to the block diagrams in Figures 3–5.

Per Particle PHD Mapping—Implementation

Prediction—Implementing (9)

Figure 3 implements the predictor (9) on a per-trajectory particle basis. First, the $J_{b,k}$ birth Gaussians replicate the spatial locations of the $|\mathcal{Z}_{k-1}|$ prior measurements (i.e., $J_{b,k} = |\mathcal{Z}_{k-1}|$), using the inverse spatial measurement model $(h^{\text{spatial}})^{-1}(\cdot)$, and are each assigned equal weight. Second, each prior map Gaussian is predicted forward in time yielding $J_{k-1|k-1}$ propagated Gaussians. For a static map (assumed here), these propagated Gaussians simply equal the prior, in terms of their means, covariances, and weights. Any knowledge of dynamic map behavior would be incorporated at this point. Finally, the $J_{b,k}$ birth and $J_{k-1|k-1}$ propagated Gaussians are added to form the $J_{k|k-1} = J_{b,k} + J_{k-1|k-1}$ predicted Gaussians on the RHS of Figure 3, thus implementing (9).

Correction—Implementing (10)

Based on the predicted GM-PHD, if the measurement likelihood $g_k(z \mid m, X_k)$ is also of Gaussian form, it follows from (10) that the posterior map PHD, $v^{(i)}_{k|k}(m \mid X^{(i)}_k)$ is also a GM. Figure 4 shows the per trajectory particle update implementation procedure of (10).

Note that in the filtering actions block update GM-PHD missed detection components, the means and covariance of all the $J_{k|k-1}$ predicted Gaussians are simply copied into the posterior GM-PHD map estimate but with their weights reduced by the probability of missed detection $1 - P_D\left(m \mid X^{(i)}_{k|k-1}\right)$. This takes into account the possibility that they may not be observed in the new measurement set $\mathcal{Z}_k$. This represents the first term on the RHS of (10).

To implement the second term on the RHS of (10), each of the $J_{k|k-1}$ predicted Gaussian component's spatial means and covariances are corrected by each of the $\mathfrak{z}_k$ measurements. This can be achieved by the standard extended Kalman filter equations, as described in Figure 4. The weights of each of these $J_{k|k-1} \times \mathfrak{z}_k$ new Gaussian components are updated based on the probability of detection of each predicted Gaussian, the Mahalanobis distance between that component's predicted spatial measurement and each actual

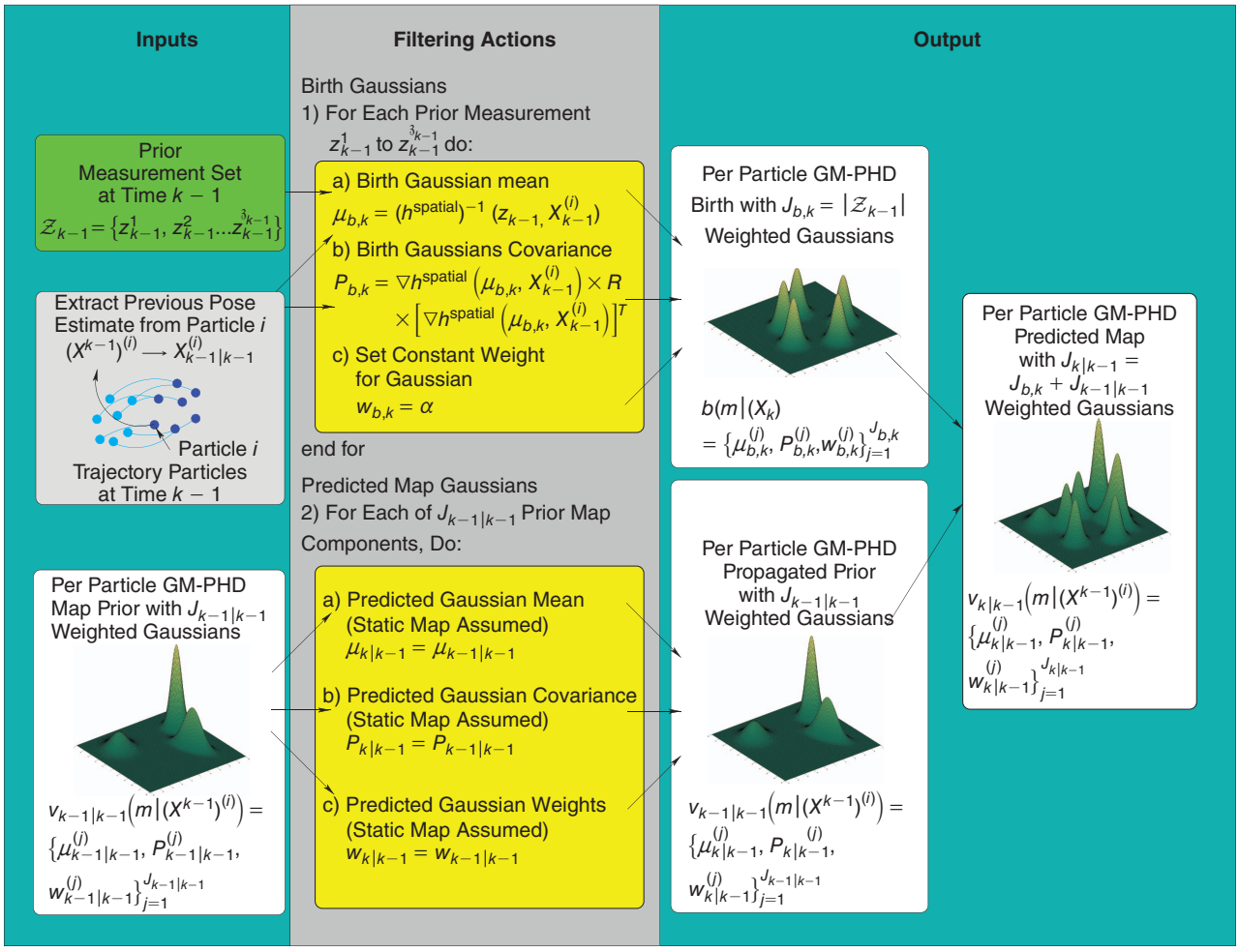

Figure 3. The implementation of the PHD predictor (9).

measurement and the false alarm variable $c_k(z)$. The false alarm PHD $c_k(z)$ represents the prior knowledge of the probability of false alarm P_{fa}. In a scan in which N_d feature detection hypotheses are made (whether determined to be detections or not), an average of $n_c = P_{\text{fa}} N_d$ false measurements will result. Therefore, $c_k(z) = n_c V U(z)$, where n_c is the false alarm rate per scan, V is the volume (or area in the 2-D experiments presented here) corresponding to the FoV of the sensor(s), and $U(z)$ is a uniform distribution over the measurement space. The missed detection and weighted prediction Gaussian components are then merged to form the $J_{k|k-1} + (J_{k|k-1} \times \jmath_k)$ Gaussians forming the updated GM-PHD $v_{k|k}(m\,|\,(X^{k-1})^{(i)})$.

The PHD filter tracks only the overall feature map behavior.

It is important to note that, unlike in vector-based SLAM implementations, feature initialization, termination, and association routines are unnecessary in the PHD filter implementation. A reduced weight copy of the feature predictions are incorporated into the final map estimate, allowing for the possibility that the sensor(s) may have missed them. In addition, all measurements are fused with all predictions, so that no assumptions on the nature of the measurements is necessary. Finally, to curb the explosive growth in the number of Gaussians formed between the prediction and update stages of the RB-PHD-SLAM filter, Gaussian merging and pruning can be adopted as shown in Figure 4. Note that although a computational limit has to be set, in terms of the number of Gaussians realizable, no feature pruning or map management heuristics were necessary in this implementation, and the final weights of each Gaussian maintain an estimate of the number of features they represent. The equivalent computational limit in a vector-based, MH-FastSLAM implementation would require heuristic-based curbing of measurements and map predictions, forcing each to be of the same, computationally manageable dimensions, with no such Bayesian estimate on the true number of features, which have intersected the FoV of the vehicle's sensor(s).

Updating the Vehicle Trajectory Particles—Implementation

The PHD-SLAM filter adopts a particle approximation of the posterior vehicle trajectory, $p_{k|k}(X^k)$. Figure 5 shows how

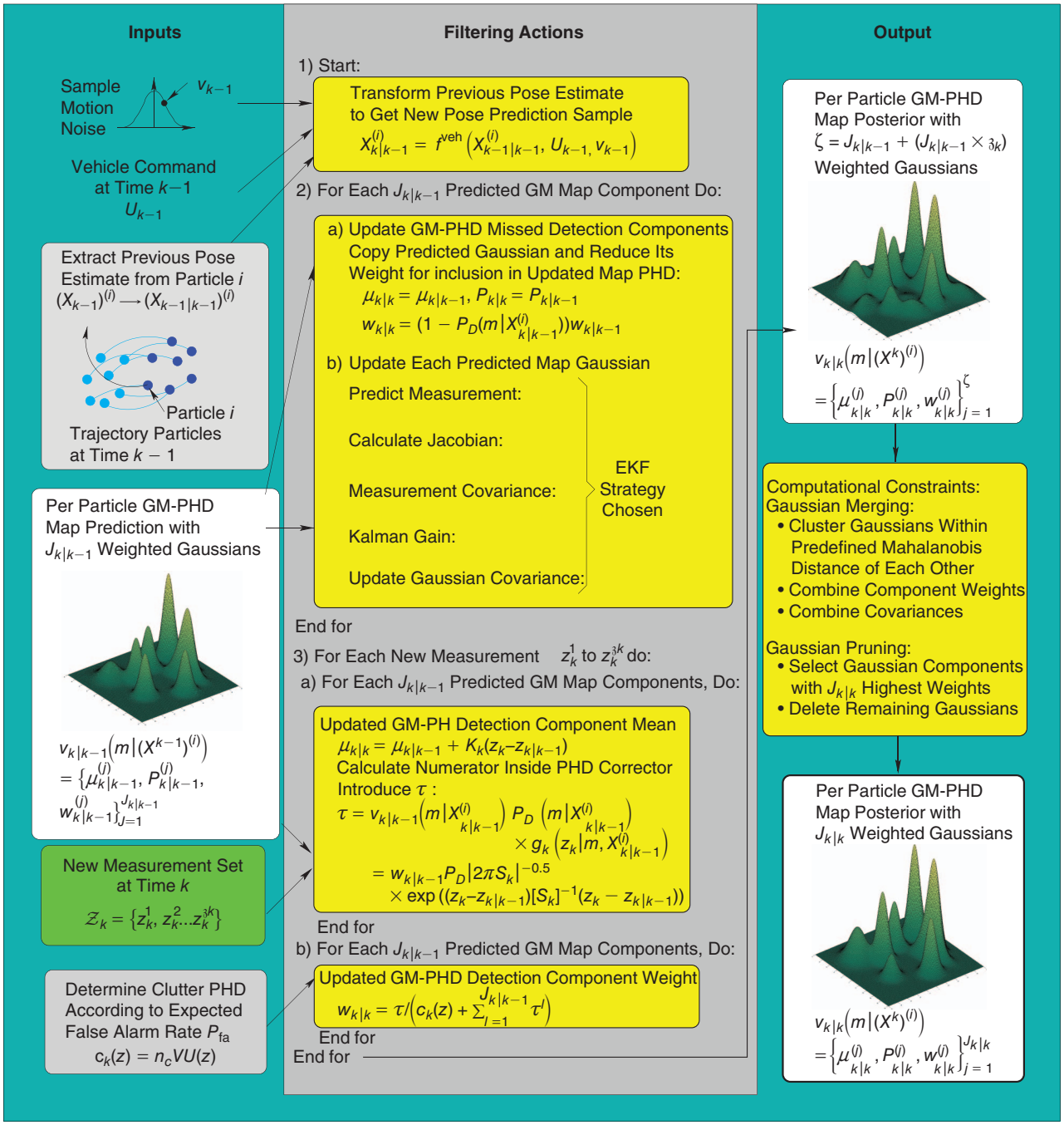

Figure 4. The implementation of the PHD corrector (10).

the trajectory particles are propagated forward in time. A fundamental difference between RB-PHD-SLAM and FastSLAM should be noted. In FastSLAM, each pose particle is used to generate a predicted measurement vector. The actual measurement vector, recorded from the unknown, true vehicle location, is then superimposed onto each particle. The likelihood of that measurement vector corresponding to that particle's predicted measurement vector is calculated to form a particle weight through the measurement likelihood $g(Z_k \mid Z^{k-1}, (X^k)^{(i)})$. This requires the usual, fragile predicted and observed feature management and association routines, for which there is no concept within the RFS framework.

In RB-PHD-SLAM, the measurement likelihood is $g(\mathcal{Z}_k \mid \mathcal{Z}^{k-1}, (X^k)^{(i)})$, which is defined on the space of finite sets, unlike its FastSLAM counterpart, which is defined on a Euclidean space. Therefore, alternative methods are necessary to evaluate $g(\mathcal{Z}_k \mid \mathcal{Z}^{k-1}, (X^k)^{(i)})$, and hence the trajectory particle's new weight. Mullane et al. showed that this set-based measurement likelihood can be evaluated in closed form, based on an arbitrary choice of map set $\mathcal{M}_k$, the simplest of which is the empty map strategy $\mathcal{M}_k = \emptyset$ [3].

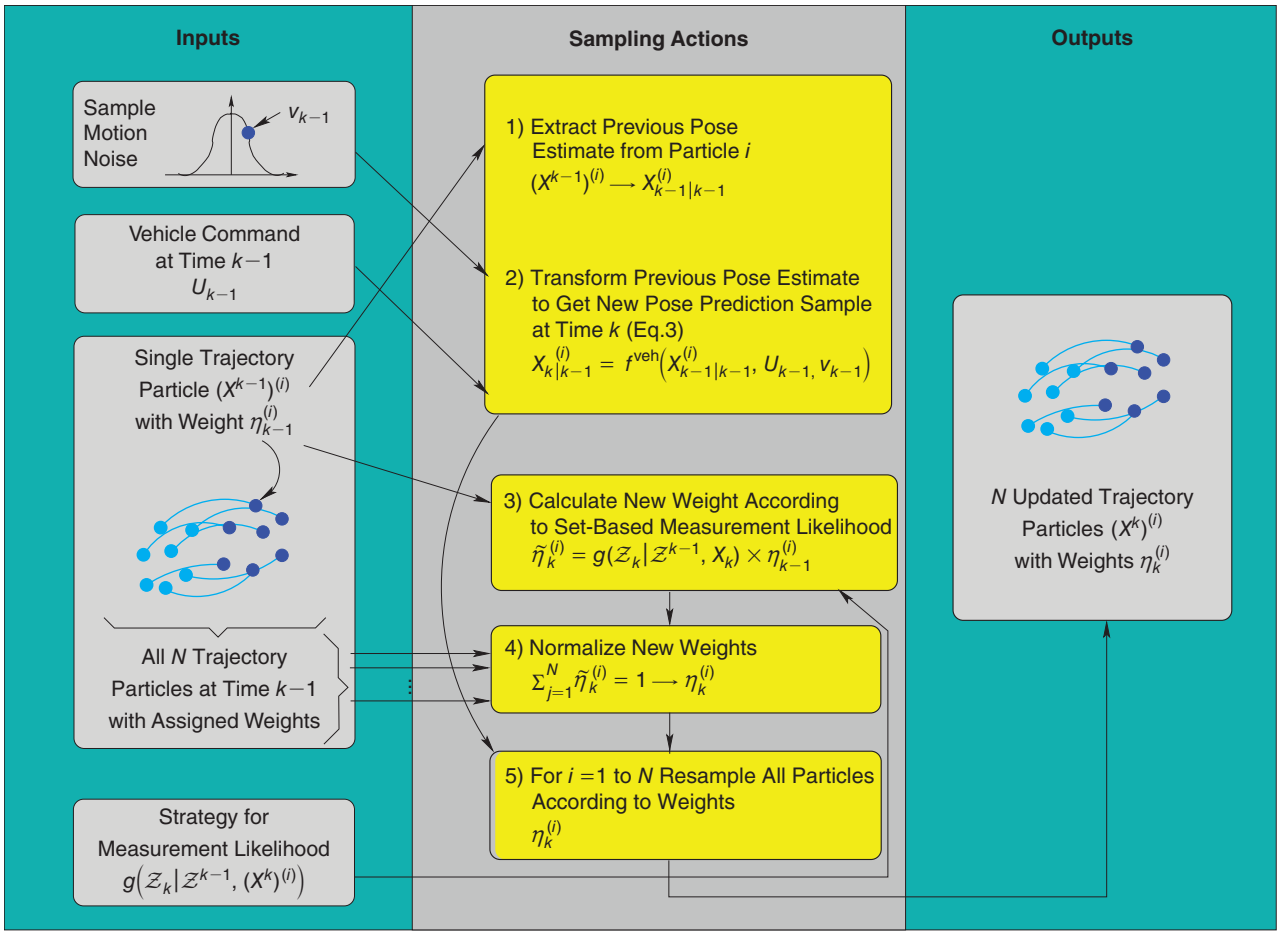

Figure 5. The implementation of the vehicle trajectory particle filter-based estimator.

Estimating the Posterior Trajectory and Map—Implementation

In MH-FastSLAM, the final trajectory is selected as the particle $(X^k)^{(i)}$ with the highest weight, and the final map estimate is its corresponding map. This is known as the *maximum a posteriori* (MAP) map estimate. (It should be noted that the particle with the maximum weight is, in general, only an approximation of the MAP estimate [14], [15].) This strategy could also be used for RB-PHD-SLAM, however, in contrast to vector-based SLAM algorithms, the PHD map representation yields a natural ability to average feature maps to give an expected a posteriori map. Map estimates from the N updated trajectory particles $(X^k)^{(i)}$ output in Figure 5 can be averaged into an expected map, even with map estimates of different sizes and without having to resolve the intramap feature associations [9]. The estimated number of features in the posterior map PHD is simply the sum of the Gaussian weights in the output $v_{k|k}(m \mid X^k)$. The expected FB map estimate can then be extracted by choosing the $\widehat{m}_{k|k}$ highest local maxima. Pseudocode implementation details of this algorithm are given in [3], [7], and [9].

Peak points in the radon transform will correspond to straight line segments in the image

RB-PHD-SLAM Computational Complexity

The computational complexity of RB-PHD-SLAM is $O(m_k \mathfrak{z}_k N)$, i.e., linear in the number of features (in the FoV), linear in the number of measurements, and linear in the number of trajectory particles. RB-PHD-SLAM simulations in [7] have shown that the computational time is comparable with that of the MH-FastSLAM algorithm. Note that due to the RB structure of RB-PHD-SLAM, binary tree-based enhancements, such as those applied to traditional FastSLAM [12], can be readily developed to further reduce the complexity to $O((\mathfrak{z}_k N \log(m_k))$.

Comparisons of PHD and Vector-Based SLAM in a Marine Environment

To demonstrate the robustness of PHD-SLAM in the presence of many false alarms and missed feature detections, SLAM algorithms, based on a commercially available X-band marine radar, are implemented on a sea kayak referred to as an *autonomous surface craft* (ASC). The ASC was remote controlled in an offshore test site in Singapore's

southern Selat Puah marine environment. Data from the radar, global positioning system (GPS), and an inexpensive single-axis gyro were logged using an on-board processing unit as the ASC traversed the enviromnent, which comprised geographical and sea-surface vessel landmarks. The standard, automated OS-CFAR feature detector, introduced in the "Feature Detection with Radar" section, was applied to the radar data to provide the features to be input to the SLAM algorithms. With restrictive feature modeling and a lack of vehicle control input information, it is demonstrated that by adopting the RFS concepts and the PHD filter, useful localization and mapping results can be obtained, despite an actively rolling and pitching ASC on the sea surface. The vector-based SLAM algorithm MH-FastSLAM is also implemented and compared.

The ASC and the Coastal Environment

The ASC was originally developed at the Department of Mechanical and Ocean Engineering, Massachusetts Institute of Technology (MIT), for experiments in autonomous navigation in rivers and coastal environments [16]. For stabilization, lateral buoyancy aids were added to the platform, as depicted in Figure 6. The figure shows the ASC at sea. with the X-Band radar mounted on a 1.5-m-length pole above the sea surface. The X-Band radar used was the M-1832 BlackBox Radar from Furuno and was primarily used to detect buoys and ships at large distances (several kilometers), which were approximated to be point features. The mechanically scanned beam has a width of 3.9° in azimuth and 20° in elevation. The large elevation beam width makes the sensor robust to the sometimes severe pitch and roll of the ASC. A GPS receiver (Crescent Hemisphere 110) as well as a KVH Industries, Inc., DSP5000 single-axis gyroscope for 3-D pose (x_k, y_k, ϕ_k) measurements were also used in the experiments. An on-board processing unit logged the GPS and gyro data at a rate of 1 Hz, with the radar data being sampled and logged at a scan rate of 0.5 Hz, i.e., one full 360° sweep of the environment required 2 s. The radar-range bin resolution, $\delta r(q)$, was set to 7.5 m, with a maximum range of 7.68 km. All power values that exceeded the OS-CFAR threshold were considered as valid point features in the RB-PHD-SLAM and MH-FastSLAM experiments.

Together with the known GPS locations of the surrounding buoys, an automatic identification system (AIS) receiver was used for ground truth verification of the map features in the experiments. An AIS is a short-range coastal tracking system used for identifying and locating sea vessels by electronically exchanging data. This enables the system to receive position and speed estimates from a large number of vessels present in the area. Since those vessels were used as features in the SLAM algorithms, this source of information was used as ground truth to verify and compare the features extracted from the radar data, with the position delivered by the AIS.

The ASC Process Model

A sea-based ASC is subject to numerous uncertain disturbances such as currents and wind, moving the ASC in any arbitrary direction. To account for this, the following nonlinear process model $f^{\text{veh}}(\cdot)$ in (3) is adopted:

$$\begin{aligned} x_k &= x_{k-1} + V_{k-1} \triangle T_k \cos(\phi_{k-1} + \delta\phi_{k-1}) + v^x_{k-1} \\ y_k &= y_{k-1} + V_{k-1} \triangle T_k \sin(\phi_{k-1} + \delta\phi_{k-1}) + v^y_{k-1} \\ \phi_k &= \phi_{k-1} + \delta\phi_{k-1} + v^{\phi}_{k-1}, \\ \text{i.e., } X_k &= f^{\text{veh}}(X_{k-1}, U_{k-1}, v_{k-1}), \end{aligned} \tag{11}$$

where x_k, y_k, and ϕ_k represent the easting, northing, and ASC heading angle with respect to north at time k, $X_k = [x_k \ y_k \ \phi_k]^T$ and $f^{\text{veh}}(\cdot)$ is the vehicle motion vector function encapsulating (11). U_{k-1} represents a vector comprising the input velocity signal and the measured angular change, i.e., $U_{k-1} = [V_{k-1} \ \delta\phi_{k-1}]^T$, recorded by an on board single axis gyroscope. Here, v^x_{k-1}, v^y_{k-1} and v^{ϕ}_{k-1} represent random perturbations in the ASC motion due to external sea forces and are modeled by white Gaussian signals, encapsulated in the noise vector $v_{k-1} = [v^x_{k-1} \ v^y_{k-1} \ v^{\phi}_{k-1}]^T$. Here, $\Delta T_k = t_k - t_{k-1}$ is determined from the measurement rate of the gyro. In this experiment, for simplicity, $V_k = V_{k-1}$ and is chosen a priori due to the lack of suitable Doppler velocity log sensors. A constant-velocity model could also be assumed, accompanied by the recursive estimation of V_k integrated into the SLAM algorithm. This vehicle process model will be used in both the SLAM algorithms, developed for comparison purposes, in this article.

Figure 6. The ASC-adapted kayak.

The RB-PHD-SLAM approach can be seen to generate more accurate localization and feature number estimates.

Vector-Based Multihypothesis FastSLAM Comparison

FastSLAM estimates the map on a per-particle basis, meaning that different particles can be associated with different features [12]. This means that the FastSLAM filter has the possibility to

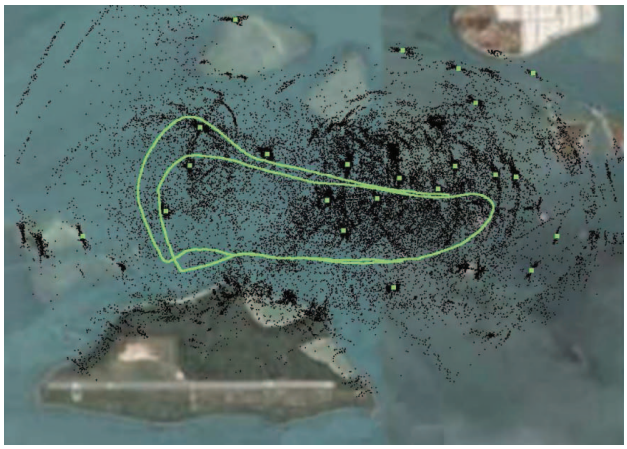

Figure 7. An overview of the test site showing the GPS trajectory (green line) and AIS coordinates (green dots) of the point feature map. The point feature measurement history is also provided (black dots). (Photo used with permission from [3].)

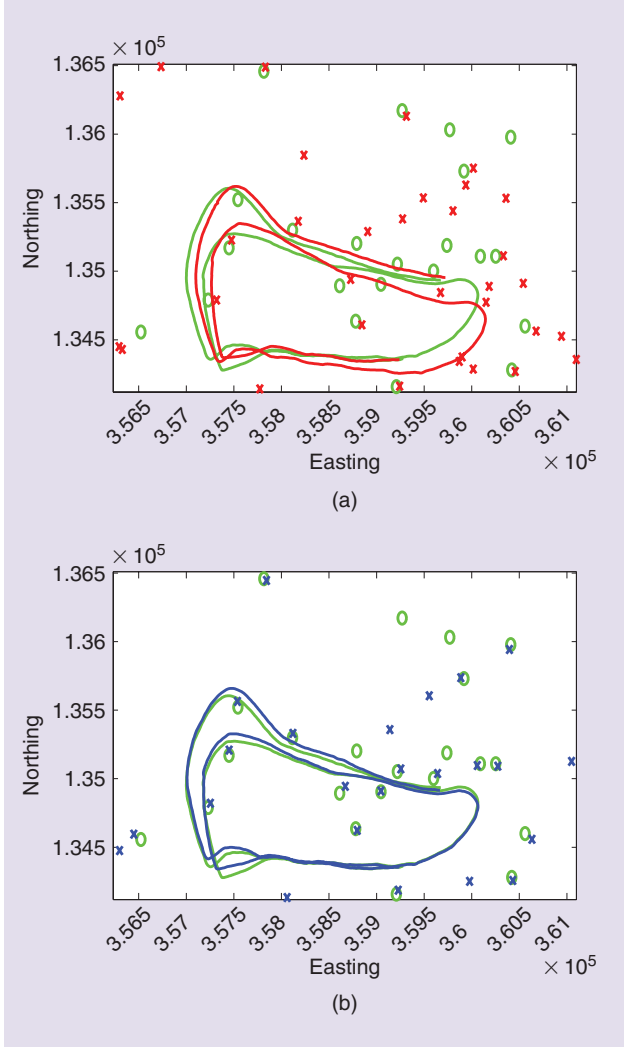

Figure 8. (a) The posterior SLAM estimate (red) from MH-FastSLAM. (b) The posterior SLAM estimate (blue) from RB-PHD-SLAM, in comparison to the ground truth (green). Crosses represent the estimated landmark locations in each case and the circles represent their ground truth (AIS) locations.

maintain different tracks for each possible hypothesis of each detected feature, known as *multiple hypothesis tracking*. A new particle is created for each new hypothesis of each measurement, meaning that each particle is split into $n + 2$ new particles, one for each of the n possible associations, one particle for the nonassociation hypothesis, and the other particle for a new feature hypothesis. Particles with incorrect data association are more likely to be eliminated than those that were based on correct associations. This step reduces the number of particles back to its original number. This vector-based, MH-FastSLAM method was implemented for comparison based exactly on the methods given in [12], [17].

A Comparison of RFS and Vector-Based SLAM at Sea

For the RB-PHD-SLAM filter, a Monte Carlo analysis is presented based on 50 sample runs using 100 trajectory particles in each trial. In this comparative experiment, the ASC was remote controlled to execute a curved trajectory of ~13 km, and the maximum range of the radar was limited to just over 1 km. Multiple loops were traversed. The AIS provided a quantifiable ground truth map in terms of feature number and location as time progressed. Figure 7 shows a GPS trajectory with the entire history of all point detections superimposed. These are shown as black points.

This is all superimposed onto a satellite image of the area. Many of the measured features appear close to the sea vessels, detected by the independent AIS (shown as green dots), which were taken as ground truth. It is also evident that many false alarms are present due to the sea clutter and noise. Figure 8(a) compares the posterior SLAM estimates from MH-FastSLAM and (b) from RB-PHD-SLAM. For fair comparison with MH-FastSLAM, the RB-PHD-SLAM final trajectory particle and final GM-PHD map estimates are the MAP values explained in the "Estimating the Posterior Trajectory and Map—Implementation" section. The estimated map features (crosses) are the maxima of the Final

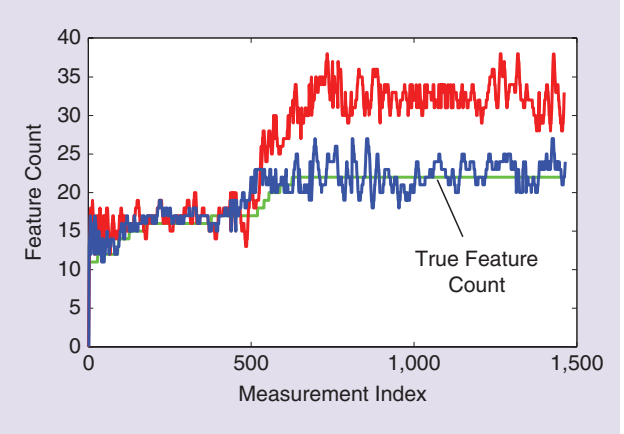

Figure 9. The comparison of the number of estimated features for each approach. The number estimated by the MH-FastSLAM algorithm is shown in the upper curve, and that estimated by the RB-PHD-SLAM filter is shown in the lower curve. The true feature count (based on the AIS) is labeled and settles at a constant value of 21.

GM-PHD map estimate. It can be seen that MH-FastSLAM overestimates the number of features due to multiple data association failures as a result of many false measurements, and it exhibits larger spatial errors for those features that truly correspond to actual landmarks. Figure 9 compares the estimated map sizes. The labeled true feature count finally settles at 21, when all of the ships detected by the AIS should have entered the FoV of the ASC's radar. Note that the RB-PHD-SLAM feature number estimate closely resembles and, on average, tends to equal the true feature number. Since the MH-FastSLAM vector-based feature management routines are typically dependant on the data association decisions; this dramatically increases its map estimation error. The RB-PHD-SLAM approach can be seen to generate more accurate localization and feature number estimates, but it can also be seen that some feature estimates are misplaced in comparison to the ground truth feature map. However, as a first approximation to RFS-based estimation, the PHD filter is still demonstrated to be useful for high false alarm FB-SLAM applications.

Conclusions

Bayes optimality of the SLAM problem, which utilizes all measurements and estimated map features and yields estimates of the number of features that have passed through the FoV of the vehicle's sensor(s) as well as their location estimates, is so far only possible under the RFS framework. Such a framework requires recently developed FISST tools, and the simplest of these, the PHD filter, was introduced. The implementation of a PHD filter SLAM was the focus of the article. With computational complexity comparable with that of a state-of-the-art MH-FastSLAM, RB-PHD-SLAM's ability to jointly estimate both detected target number as well as location was shown, as it provided superior estimates of these quantities as time progressed. Further enhancements, which adopt higher-order FISSTs, such as the cardinalized PHD filter and multi-Bernoulli RFS techniques, are avenues for future work.

Acknowledgments

The authors would like to thank the AMTC, Universidad de Chile, Conicyt Fondecyt Project 1110579, Conicyt Anillo Project ACT 1120; Clearpath Robotics, Canada; the Australian Research Council Future Fellowship FT0991854; and the Singapore-MIT Alliance for Research and Technology. They would also like to thank B.-T. Vo, D. Liihr, F. Inostroza, and K. Leung for their contributions.

References

[1] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Trans. Robot. Autom.*, vol. 17, no. 3, pp. 229–241, June 2001.

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

[3] J. Mullane, B.-N. Vo, M. D. Adams, and B.-T. Vo, "A random-finite set approach to Bayesian SLAM," *IEEE Trans. Robot.*, vol. 27, no. 2, pp. 268–282, Apr. 2011.

[4] C. S. Lee, D. E. Clark, and J. Salvi, "SLAM with dynamic target via single cluster PHD filtering," *IEEE Select. Topics Signal Processing*, vol. 7, no. 3, pp. 543–552, June 2013.

[5] C. S. Lee, D. E. Clark, and J. Salvi, "SLAM with single cluster PHD filters," in *Proc. IEEE Int. Conf. Robotics Automation*, St. Paul, Minnesota, May 2012, pp. 2096–2101.

[6] D. Moratuwage, B.-N. Vo, and D. Wang, "A hierarchical approach to the multi-vehicle slam problem," in *Proc. 15th Int. Conf. Information Fusion*, Singapore, July 2012, pp. 1119–1125.

[7] J. Mullane, B.-N. Vo, M. Adams, and B.-T. Vo, "Random finite sets for robot mapping and SLAM," in *Springer Tracts in Advanced Robotics*. vol. 72, Heidelberg Berlin, Germany: Springer, 2011.

[8] H. Rohling, "Radar CFAR thresholding in clutter and multiple target situations," *IEEE Trans. Aerospace Electron. Syst.*, vol. 19, no. 4, pp. 608–621, 1983.

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

[10] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

[11] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vision Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[12] M. Montemerlo, S. Thrun, and B. Siciliano, *FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics*. Berlin, Germany: Springer, 2007.

[13] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "Fastslam: A factored solution to the simultaneous localization and mapping problem," in *Proc. AAAI National Conf. Artificial Intelligence*, 2004, pp. 593–598.

[14] S. Godsill, A. Doucet, and M. West, "Maximum a posteriori sequence estimation using Monte Carlo particle filters," *Ann. Inst. Stat. Math.*, vol. 53, no. 1, pp. 82–96, 2001.

[15] S. Saha, Y. Boers, J. N. Driessen, P. K. Mandal, and A. Bagchi, "Particle filter based map state estimation: A comparison," in *Proc. 12th Int. Conf. Information Fusion*, Seattle, July 2009, pp. 278–283.

[16] J. E. Manley, "Unmanned surface vehicles, 15 years of development," in *Proc. OCEANS 2008*, pp. 1–4.

[17] J. Nieto, J. Guivant, E. Nebot, and S. Thrun, "Real time data association for fastSLAM," in *Proc. IEEE Int. Conf. Robotics Automation*, pp. 412–418, vol. 1, Sept. 2003.

Martin Adams, Department of Electrical Engineering and Advanced Mining Technology Center (AMTC), Universidad de Chile, Santiago, Chile. E-mail: martin@ing.uchile.cl.

Ba-Ngu Vo, Curtin University, 6009 Perth, Australia. E-mail: bangu. vo@curtin.edu.au, bangu. vo@uwa.edu.au.

Ronald Mahler, Lockheed Martin Advanced Technology Laboratories, Eagan, Minnesota, United States. E-mail: ronald.p.mahler@lmco.com.

John Mullane, Projective Space Pte. Ltd., Science Park 1, Singapore. E-mail: jmullane@projectivespace.com.