Chapter 6
Rao-Blackwellised RFS Bayesian SLAM

6.1 Introduction

This chapter proposes an alternative Bayesian framework for feature-based SLAM, again in the general case of uncertain feature number and data association. As in Chapter 5, a first order solution, coined the probability hypothesis density (PHD) SLAM filter, is used, which jointly propagates the posterior PHD of the map and the posterior distribution of the vehicle trajectory. In this chapter however, a Rao-Blackwellised (RB) implementation of the PHD-SLAM filter is proposed based on the GM PHD filter for the map and a particle filter for the vehicle trajectory, with initial results presented in [56] and further refinements in [57].

A tractable PHD approximation to the SLAM problem is derived, which propagates the posterior PHDs of *multiple trajectory-conditioned maps* and the posterior distribution of the trajectory of the vehicle. Furthermore, this approach to SLAM admits the concept of an 'expected' map via the PHD construct, which is not available in previous SLAM approaches.

The chapter is organised as follows. Section 6.2 discusses the factorised RFS SLAM recursion, in which the posterior density of the map, conditioned on the trajectory, and the trajectory itself can be propagated jointly. The RFS framework is then applied to this factorised solution, where it is demonstrated that subtle differences, regarding the use of sets, make a direct, naive implementation of FastSLAM to the RFS problem inappropriate. In particular, the likelihood of the measurement, conditioned on the trajectory, which is necessary for the calculation of the particle weights, cannot be approximated under an EKF framework, as in FastSLAM [58]. Solutions, which give a closed form solution to this problem, are presented in this section. Section 6.3 outlines a Rao-Blackwellised implementation of the PHD-SLAM filter. The necessary steps to implement the PHD filter for the estimation of the map and the vehicle trajectory are given, along with pseudo code. Section 6.4 presents and discusses the Rao-Blackwellised RFS SLAM performance.

J. Mullane et al.: Random Finite Sets for Robot Mapping & SLAM, STAR 72, pp. 97–126.
springerlink.com

Demonstrations of simulated examples are given, due to the simplicity of generating both trajectory and map ground truth values, necessary for true performance evaluation assessment. This is followed by an implementation with real, short range, millimetre wave (MMW) radar data, and a mobile robot platform, in a car park environment. The advantages of these sensors over other devices such as laser range finders is discussed in [16]. Data sets from the radar are recorded, along with odometry and single axis gyro rate data, from a moving vehicle. Comparisons are made with classical, vector based, EKF SLAM which utilises the Joint Compatibility Branch and Bound (JCBB) [39] data association method and FastSLAM with Multiple Hypothesis (MH) data association. Further comparative results, in a much larger scenario, where accurate SLAM performance in the presence of high clutter is essential, are demonstrated at sea, in a coastal environment, using an "Autonomous kayak" [59] as the vehicle and a commercially available X-Band radar. The performance improvement, in the presence of clutter is clearly demonstrated. Comparisons and discussions of the computational complexity of the algorithms is also given.

6.2 The Rao-Blackwellised (RB) PHD-SLAM Filter

Since the full RFS-SLAM Bayes filter of equations 2.14 and 2.15 is numerically intractable, it is again necessary to look for tractable but principled approximations. This section derives a recursion that jointly propagates the posterior PHD of the map and the posterior density of the vehicle trajectory. Analogous to FastSLAM, the RFS-SLAM recursion can be factorised as shown in Section 6.2.1. Section 6.2.2 discusses the PHD estimator in the context of this factorised recursion, Section 6.2.3 addresses the PHD representation of the map component only while Section 6.2.4 extends this algorithm to perform SLAM.

6.2.1 The Factorised RFS-SLAM Recursion

Using standard conditional probability, the joint posterior RFS-SLAM density of equation 2.15 can be decomposed as,

$$p_k(\mathcal{M}_k, X_{1:k}|\mathcal{Z}_{0:k}, U_{0:k-1}, X_0) = \\ p_k(X_{1:k}|\mathcal{Z}_{0:k}, U_{0:k-1}, X_0) p_k(\mathcal{M}_k|\mathcal{Z}_{0:k}, X_{0:k}). \quad (6.1)$$

Thus, the recursion for the joint RFS map-trajectory posterior density according to equation 2.15 is equivalent to jointly propagating the posterior density of the map conditioned on the trajectory and the posterior density of the trajectory. In this section, as before, for compactness,

$$p_{k|k-1}(\mathcal{M}_k|X_{0:k}) = p_{k|k-1}(\mathcal{M}_k|\mathcal{Z}_{0:k-1}, X_{0:k}) \tag{6.2}$$

$$p_k(\mathcal{M}_k|X_{0:k}) = p_k(\mathcal{M}_k|\mathcal{Z}_{0:k}, X_{0:k}) \tag{6.3}$$

$$p_k(X_{1:k}) = p_k(X_{1:k}|\mathcal{Z}_{0:k}, U_{0:k-1}, X_0) \tag{6.4}$$

and it follows that,

$$p_{k|k-1}(\mathcal{M}_k|X_{0:k}) = \int f_{\mathcal{M}}(\mathcal{M}_k|\mathcal{M}_{k-1}, X_k) p_{k-1}(\mathcal{M}_{k-1}|X_{0:k-1}) \delta\mathcal{M}_{k-1} \tag{6.5}$$

$$p_k(\mathcal{M}_k|X_{0:k}) = \frac{g_k(\mathcal{Z}_k|\mathcal{M}_k, X_k) p_{k|k-1}(\mathcal{M}_k|X_{0:k})}{g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1}, X_{0:k})} \tag{6.6}$$

$$p_k(X_{1:k}) = g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1}, X_{0:k}) \frac{f_X(X_k|X_{k-1}, U_{k-1}) p_{k-1}(X_{1:k-1})}{g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1})}. \tag{6.7}$$

Apart from adopting RFS likelihoods for the measurement and map, the recursion defined by equations 6.5, 6.6 and 6.7 is similar to that in FastSLAM [58], [60]. However, the use of RFS likelihoods has important consequences in the evaluation of equation 6.7, to be seen later in Section 6.2.4. In FastSLAM, it should be noted that the map recursion of equation 6.6 is further decomposed into the product of K independent densities for each of the K features assumed to exist in the map. Furthermore, FastSLAM is conditioned on the inherently unknown data association assignments. In contrast, RFS-SLAM is not conditioned on any data association hypotheses to determine the number of features to estimate and the recursion of equation 6.6 is that of a RFS feature map. Consequently, the propagation of the map involves probability densities of random finite sets and marginalisation over the map involves set integrals. Similar to FastSLAM, the effect of the trajectory conditioning on RFS-SLAM is to render each feature estimate conditionally independent and thus the map correlations, critical to EKF-SLAM [1], are not required.

6.2.2 The PHD in RFS-SLAM

Recall from Section 3.3.1, that an optimal estimator for a random vector is the conditional expectation. Many state-of-the-art SLAM algorithms adopt Sequential Monte Carlo (SMC) methods. It is well known that SMC techniques are more amenable to expectation operations than maximisation operations, since if p is approximated by a set of weighted samples $\{\eta^{(i)}, X^{(i)}\}_{i=1}^{N}$, then [61], [62],

$$\sum_{i=1}^{N} \eta^{(i)} X^{(i)} \rightarrow \mathbb{E}[X] \tag{6.8}$$

as $N \rightarrow \infty$. However, in FastSLAM [58], it is common to choose the trajectory sample with the highest weight as the estimate of the vehicle path, and its corresponding map, as the estimate of the map. It is easier to establish convergence in SMC implementations if we use the expected path and expected map. However, it is not clear how the expected map is interpreted.

The PHD construct allows an alternative notion of expectation for maps, thereby fully exploiting the advantage of an SMC approximation. The PHD, v, is a function defined on the feature space satisfying equation 3.18. Recall from Section 3.3.4.1, that the value of the PHD at a point gives the expected number of features at that point while the integral over any given region gives the expected number of features in that region. A salient property of the PHD construct in map estimation is that the posterior PHD of the map is indeed the expectation of the trajectory-conditioned PHDs. More concisely,

$$v_k(m) = \mathbb{E}\left[v_k(m|X_{1:k})\right], \tag{6.9}$$

where the expectation is taken over the vehicle trajectory $X_{1:k}$. This result follows from the standard properties of the PHD (intensity function) of an RFS, see for example classical texts such as [31], [32]. Thus the PHD construct provides a natural framework to average feature map estimates, while incorporating both unknown associations and different feature numbers. This differs dramatically from vector based map averaging methods which require feature identification tracking and rule-based combinations [63]. In contrast, map averaging for grid-based maps is straight forward due to both known grid alignments and number of cells. While the practical merits of an expected feature map estimate for SLAM using a single sensor may be unclear at this time, related operations such as 'feature map addition' may be of use in sensor fusion and multi-robot SLAM applications. Furthermore, the PHD construct admits a Bayes optimal estimator for the map, as discussed previously in Section 3.3.1.

6.2.3 PHD Mapping

This section details the trajectory-conditioned PHD mapping recursion of equation 6.6, as was first proposed in [64]. The predicted and posterior RFS maps are approximated by Poisson RFSs with PHDs $v_{k|k-1}(m|X_{0:k})$ and $v_k(m|X_{0:k})$ respectively,

$$p_{k|k-1}(\mathcal{M}_k|X_{0:k}) \approx \frac{\prod\limits_{m} v_{k|k-1}(m|X_{0:k})}{\exp\left(\int v_{k|k-1}(m|X_{0:k})dm\right)} \tag{6.10}$$

$$p_k(\mathcal{M}_k|X_{0:k}) \approx \frac{\prod\limits_{m\in\mathcal{M}_k} v_k(m|X_{0:k})}{\exp\left(\int v_k(m|X_{0:k})dm\right)}. \tag{6.11}$$

In essence, this approximation assumes that features are IID and the number of features is Poisson distributed. This PHD approximation has been proven to be powerful and effective in multi-target tracking [3]. Poisson approximations for the number of new features have also been adopted in certain SLAM solutions [14]. In light of the above advantages of representing an RFS with sequential Monte Carlo methods, the PHD filter for the SLAM problem can be implemented in Rao-Blackwellised form. Again, referring to the PHD predictor – corrector of equations 3.23 and 3.24, substituting

$$\Gamma_k \longrightarrow m|X_{0:k} \tag{6.12}$$

the PHD predictor equation then becomes

$$v_{k|k-1}(m|X_{0:k}) = v_{k-1}(m|X_{0:k-1}) + b(m|X_k) \tag{6.13}$$

where $b(m|X_k)$ is the PHD of the new feature RFS, $\mathcal{B}(X_k)$, initially introduced in Section 3.4.

The corresponding Rao-Blackwellised, PHD corrector equation is then

$$\begin{aligned} v_k(m|X_{0:k}) = v_{k|k-1}(m|X_{0:k})\Bigg[1 - P_D(m|X_k)+ \\ \sum_{z\in\mathcal{Z}_k} \frac{P_D(m|X_k)g_k(z|m,X_k)}{c_k(z|X_k) + \int P_D(\xi|X_k)g_k(z|\xi,X_k)v_{k|k-1}(\xi|X_{0:k})d\xi}\Bigg] \end{aligned} \tag{6.14}$$

where

$$\begin{aligned} P_D(m|X_k) &= \text{the probability of detecting a feature at} \\ &\quad\; m\text{, from vehicle pose } X_k. \\ c_k(z|X_k) &= \text{PHD of the clutter RFS } \mathcal{C}_k(X_k) \text{ (in equation 2.10)} \\ &\quad\; \text{at time } k \text{ and,} \\ g_k(z|m,X_k) &= \text{the measurement model of the sensor at time } k. \end{aligned} \tag{6.15}$$

In contrast to the “Brute force” SLAM approach of chapter 5, the RB PHD SLAM filter estimates multiple, conditionally independent PHDs (intensity functions). Each independent map PHD, is conditioned on each of the N hypothesised vehicle trajectories (particles). Referring again to the GM example representations of PHDs in figures 3.2 and 3.3, in any particular map PHD, each Gaussian representing a/some possible feature(s) is conditioned on one of the N hypothesised vehicle trajectories. N such conditionally independent PHDs, one per hypothesised trajectory, are then propagated.

6.2.4 PHD-SLAM

This section extends the trajectory-conditioned PHD mapping recursion to the SLAM problem. With the hindsight of FastSLAM [58], the most obvious extension of PHD mapping [64] to SLAM is to exploit the factorisation equations 6.5, 6.6 and 6.7, e.g. PHD for mapping and particle filtering for localisation. This technique requires the computation of the posterior density of the vehicle trajectory in equation 6.7, in particular the term $g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1}, X_{0:k})$, which requires set integration,

$$g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1}, X_{0:k}) = \int p(\mathcal{Z}_k, \mathcal{M}_k|\mathcal{Z}_{0:k-1}, X_{0:k})\delta\mathcal{M}_k. \tag{6.16}$$

This set integral is numerically intractable and a naive approach is to directly apply the EKF approximation proposed for FastSLAM [65]. However, an EKF approximation cannot be used since the likelihood equation 6.16, defined on the space of finite-sets, and its FastSLAM counterpart, defined on a Euclidean space, are two fundamentally different quantities and it is not known how they are even related. Therefore, in this case, it is fundamentally incorrect to use the EKF approximation in [58] as it will not result in a valid density, and thus its product with equation 6.6 cannot give the joint posterior of the map and pose. An EKF approximation requires a hypothesised data association assignment. Since there is no concept of data association in the RFS-SLAM framework (there is no fixed ordering of features or measurements in the state), alternative methods of evaluation of equation 6.16 are required.

Fortunately, by rearranging equation 6.6, it can be seen that $g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1}, X_{0:k})$ is merely the normalising constant,

$$g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1}, X_{0:k}) = \frac{g_k(\mathcal{Z}_k|\mathcal{M}_k, X_k)p_{k|k-1}(\mathcal{M}_k|X_{0:k})}{p_k(\mathcal{M}_k|X_{0:k})}. \tag{6.17}$$

Note in the above, that the LHS does not contain the variable $\mathcal{M}_k$, while the RHS has $\mathcal{M}_k$ in both the denominator and numerator. In essence, $\mathcal{M}_k$ in equation 6.17 is a dummy variable, and thus equation 6.17 holds for any arbitrary choice of $\mathcal{M}_k$. This allows the substitution of any choice of $\mathcal{M}_k$ to evaluate $g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1}, X_{0:k})$. This is an important result, which allows for the likelihood of the measurement conditioned on the trajectory (but not the map), to be calculated in closed-form, as opposed to using the EKF approximations in [58]. The following considers two simple choices: (derivations can be seen in Appendix B.)

6.2.4.1 The Empty Strategy

It was mentioned in Section 3.3.4.4, that if the RFS $\mathcal{M}_k$ is Poisson distributed in its number, and the points within $\mathcal{M}_k$ are IID distributed, then the probability density of $\mathcal{M}_k$ can be recovered exactly from the PHD intensity function according to equation 3.19. Similarly the predicted and posterior RFS maps can be approximated by Poisson RFSs with PHDs $v_{k|k-1}(m|X_{0:k})$ and $v_k(m|X_{0:k})$ respectively,

$$p_{k|k-1}(\mathcal{M}_k|X_{0:k}) \approx \frac{\prod\limits_{m\in\mathcal{M}_k} v_{k|k-1}(m|X_{0:k})}{\exp\left(\int v_{k|k-1}(m|X_{0:k})dm\right)} \tag{6.18}$$

$$p_k(\mathcal{M}_k|X_{0:k}) \approx \frac{\prod\limits_{m\in\mathcal{M}_k} v_k(m|X_{0:k})}{\exp\left(\int v_k(m|X_{0:k})dm\right)}. \tag{6.19}$$

Setting $\mathcal{M}_k = \emptyset$, and using the Poisson RFS approximations, equation 6.18 and equation 6.19, as well as the RFS measurement likelihood, equation 5.4 shown in Section 5.2, it follows that (see Appendix B)

$$g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1}, X_{0:k}) \approx \kappa_k^{\mathcal{Z}_k} \exp\left(\hat{\mathfrak{m}}_k - \hat{\mathfrak{m}}_{k|k-1} - \int c_k(z|X_k)dz\right), \tag{6.20}$$

where, $\kappa_k^{\mathcal{Z}_k} = \prod\limits_{z\in\mathcal{Z}_k} c_k(z|X_k)$ with, $c_k(z|X_k)$ being the PHD of the measurement clutter RFS $\mathcal{C}_k(X_k)$. In addition, $\hat{\mathfrak{m}}_k = \int v_k(m|X_{0:k})dm$ and $\hat{\mathfrak{m}}_{k|k-1} = \int v_{k|k-1}(m|X_{0:k})dm$. Equation 6.20 gives the closed form likelihood of the measurement, conditioned on the trajectory, and not on the map.

6.2.4.2 The Single Feature Strategy

In a similar manner, to evaluate the normalising constant for the case of $\mathcal{M}_k = \{\bar{m}\}$, again using equations 6.18, 6.19, and 5.4,

$$\begin{aligned} g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1}, X_{0:k}) \approx \frac{1}{\Gamma}\Bigg[\Bigg((1 - P_D(\bar{m}|X_k))\kappa_k^{\mathcal{Z}_k} + \\ P_D(\bar{m}|X_k)\sum_{z\in\mathcal{Z}_k} \kappa_k^{\mathcal{Z}_k-\{z\}} g_k(z|\bar{m}, X_k)\Bigg) v_{k|k-1}(\bar{m}|X_{0:k})\Bigg] \end{aligned} \tag{6.21}$$

with,

$$\Gamma = \exp\left(\hat{\mathfrak{m}}_{k|k-1} - \hat{\mathfrak{m}}_k + \int c_k(z)dz\right) v_k(\bar{m}|X_{0:k}). \tag{6.22}$$

For this choice of $\mathcal{M}_k$, $\bar{m}$ can be, for instance, the feature with the least uncertainty or the maximum measurement likelihood. It is possible to choose $\mathcal{M}_k$ with multiple features, however this will increase the computational burden. Due to the presence of the measurement likelihood term $g_k(z|\bar{m}, X_k)$, it is expected that, in general, the single feature map update will outperform that of the empty map update. Note that in equation 6.17, every choice of $\mathcal{M}_k$ would give the same result, however equations 6.20 and 6.21 use different *approximations* of $p_k(\mathcal{M}_k|X_{0:k})$, yielding slightly different results. In principle, any map strategy can be used including more features, however the computation required to evaluate the trajectory conditioned measurement likelihood would also increase. The following section presents a Rao-Blackwellised implementation of the proposed PHD-SLAM filter.

6.3 Rao-Blackwellised Implementation of the PHD-SLAM Filter

Following the description of the PHD-SLAM filter in the previous section, a Rao-Blackwellised (RB) implementation is detailed in this section. In essence, a particle filter is used to propagate the vehicle trajectory (equation 6.7), and a Gaussian mixture (GM) PHD filter is used to propagate each trajectory-conditioned map PHD (equation 6.6). As such, let the PHD-SLAM density at time $k-1$ be represented by a set of N particles,

$$\left\{\eta_{k-1}^{(i)}, X_{0:k-1}^{(i)}, v_{k-1}^{(i)}(m|X_{0:k-1}^{(i)})\right\}_{i=1}^{N},$$

where $X_{0:k-1}^{(i)} = [X_0, X_1^{(i)}, X_2^{(i)}, \ldots, X_{k-1}^{(i)}]$ is the i^{th} hypothesised vehicle trajectory and $v_{k-1}^{(i)}(m|X_{0:k-1}^{(i)})$ is its map PHD. The filter then proceeds to approximate the posterior density by a new set of weighted particles,

$$\left\{\eta_{k}^{(i)}, X_{0:k}^{(i)}, v_{k}^{(i)}(m|X_{0:k}^{(i)})\right\}_{i=1}^{N},$$

as follows:

6.3.1 PHD Mapping – Implementation

Let the prior map PHD for the i^{th} particle, $v_{k-1}^{(i)}(m|X_{k-1}^{(i)})$, be a Gaussian mixture of the form,

$$v_{k-1}^{(i)}(m|X_{k-1}^{(i)}) = \sum_{j=1}^{J_{k-1}^{(i)}} \omega_{k-1}^{(i,j)} \mathcal{N}\big(m; \mu_{k-1}^{(i,j)}, P_{k-1}^{(i,j)}\big) \tag{6.23}$$

which is a mixture of $J_{k-1}^{(i)}$ Gaussians, with $\omega_{k-1}^{(i,j)}$, $\mu_{k-1}^{(i,j)}$ and $P_{k-1}^{(i,j)}$ being the corresponding predicted weights, means and covariances respectively for the j^{th} Gaussian component of the map PHD of the i^{th} trajectory. Let the new feature intensity for the particle, $b(m|\mathcal{Z}_{k-1}, X_k^{(i)})$, from the sampled pose, $X_k^{(i)}$ at time k also be a Gaussian mixture of the form

$$b(m|\mathcal{Z}_{k-1}, X_k^{(i)}) = \sum_{j=1}^{J_{b,k}^{(i)}} \omega_{b,k}^{(i,j)} \mathcal{N}\big(m; \mu_{b,k}^{(i,j)}, P_{b,k}^{(i,j)}\big) \tag{6.24}$$

where, $J_{b,k}^{(i)}$ defines the number of Gaussians in the new feature intensity at time k and $\omega_{b,k}^{(i,j)}$, $\mu_{b,k}^{(i,j)}$ and $P_{b,k}^{(i,j)}$ are the corresponding components. This is analogous to the proposal distribution in the particle filter and provides an initial estimate of the new features entering the map.

The predicted intensity is therefore also a Gaussian mixture,

$$v_{k|k-1}^{(i)}(m|X_k^{(i)}) = \sum_{j=1}^{J_{k|k-1}^{(i)}} \omega_{k|k-1}^{(i,j)} \mathcal{N}\big(m; \mu_{k|k-1}^{(i,j)}, P_{k|k-1}^{(i,j)}\big) \tag{6.25}$$

which consists of $J_{k|k-1}^{(i)} = J_{k-1}^{(i)} + J_{b,k}^{(i)}$ Gaussians representing the union of the prior map intensity, $v_{k-1}(m|X_{k-1}^{(i)})$, and the proposed new feature intensity according to equation 6.13. Since the measurement likelihood is also of Gaussian form, it follows from equation 6.14 that the posterior map PHD, $v_k^{(i)}(m|X_k^{(i)})$ is then also a Gaussian mixture given by

$$v_k^{(i)}(m|X_k^{(i)}) = v_{k|k-1}^{(i)}(m|X_k^{(i)}) \Bigg[1 - P_D(m|X_k^{(i)}) + \sum_{z \in \mathcal{Z}_k} \sum_{j=1}^{J_{k|k-1}^{(i)}} v_{G,k}^{(i,j)}(z, m|X_k^{(i)})\Bigg]. \tag{6.26}$$

The components of the above equation are given by,

$$v_{G,k}^{(i,j)}(z, m|X_k^{(i)}) = \psi_k^{(i,j)}(z|X_k^{(i)}) \mathcal{N}(m; \mu_{k|k}^{(i,j)}, P_{k|k}^{(i,j)}) \tag{6.27}$$

$$\psi_k^{(j)}(z|X_k^{(i)}) = \frac{P_D(m|X_k^{(i)}) \omega_{k|k-1}^{(i,j)} \mathcal{N}\big(z; H_k \mu_{k|k-1}^{(i,j)}, S_k^{(i,j)}\big)}{c(z) + \sum_{\ell=1}^{J_{k|k-1}^{(i)}} P_D(m|X_k^{(i)}) \omega_{k|k-1}^{(i,\ell)} \mathcal{N}\big(z; H_k \mu_{k|k-1}^{(i,\ell)}, S_k^{(i,\ell)}\big)} \tag{6.28}$$

The terms $\mu_{k|k}$, $P_{k|k}$ and S_k can be obtained using any standard filtering technique such as EKF or UKF [66]. In this chapter, the EKF updates are adopted.

The clutter RFS, C_k, is assumed Poisson distributed [14] in number and uniformly spaced over the mapping region. Therefore the clutter intensity is given by, $c(z) = \lambda_c\, \mathcal{U}(z)$, where λ_c is the average number of clutter measurements and $\mathcal{U}(z)$ denotes a uniform distribution on the measurement space. As with other feature-based GM implementations [38], pruning and merging operations are required to curb the explosive growth in the number of Gaussian components of the posterior map PHD. These operations are carried out as in [33].

6.3.2 The Vehicle Trajectory – Implementation

The proposed filter adopts a particle approximation of the posterior vehicle trajectory, $p_k(X_{1:k})$, which is sampled/re-sampled as follows:

Step 1: <u>*Sampling Step*</u>

- For $i = 1, ..., N$, sample $\widetilde{X}_k^{(i)} \sim f_X(\widetilde{X}_k^{(i)} | X_{k-1}^{(i)}, U_{k-1})$ and set

$$\widetilde{\eta}_k^{(i)} = \frac{g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1}, \widetilde{X}_{0:k}^{(i)}) f_X(\widetilde{X}_k^{(i)}|X_{k-1}^{(i)}, U_{k-1})}{f_X(\widetilde{X}_k^{(i)}|X_{k-1}^{(i)}, U_{k-1})} \eta_{k-1}^{(i)}. \tag{6.29}$$

- Normalise weights: $\sum_{i=1}^N \widetilde{\eta}_k^{(i)} = 1$.

Step 2: <u>*Resampling Step*</u>

- Resample $\left\{\widetilde{\eta}_k^{(i)}, \widetilde{X}_{0:k}^{(i)}\right\}_{i=1}^N$ to get $\left\{\eta_k^{(i)}, X_{0:k}^{(i)}\right\}_{i=1}^N$.

Since the vehicle transition density is chosen as the proposal density as with FastSLAM 1.0 [58],

$$\widetilde{\eta}_k^{(i)} = g_k(\mathcal{Z}_k|\mathcal{Z}_{0:k-1}, \widetilde{X}_{0:k}^{(i)}) \eta_{k-1}^{(i)} \tag{6.30}$$

which can be evaluated in closed form according to $\mathcal{M}_k$ being the empty map (equation 6.20) or $\mathcal{M}_k$ being a single feature map (equation 6.21), where

$$\widehat{\mathfrak{m}}_{k|k-1}^{(i)} = \sum_{j=1}^{J_{k|k-1}^{(i)}} \omega_{k|k-1}^{(i,j)} \text{ and } \widehat{\mathfrak{m}}_k^{(i)} = \sum_{j=1}^{J_k^{(i)}} \omega_k^{(i,j)}. \tag{6.31}$$

Note that the incorporation of the measurement conditioned proposal of FastSLAM 2.0 can also be accommodated in this framework. This direction of research focuses on more efficient SMC approximations and is an avenue for further studies.

6.3.3 SLAM State Estimation and Pseudo-code

As mentioned in the introduction, in contrast to previous SLAM algorithms, the PHD map representation allows for a natural ability to average feature maps. That is, independent map estimates from N independent trajectory particles can be readily averaged into an expected estimate, even with map estimates of different size and without having to resolve the intra-map feature associations. Consequently, in the case of the RB-PHD-SLAM filter, both the expected vehicle trajectory and feature map can be determined as follows:

Given the posterior set of weighted particles and corresponding map PHDs,

$$\left\{\eta_k^{(i)}, X_{0:k}^{(i)}, v_k^{(i)}(m|X_{0:k}^{(i)})\right\}_{i=1}^{N},$$

and $\bar{\eta} = \sum_{i=1}^{N} \eta_k^{(i)}$ then,

$$\widehat{X}_{0:k} = \frac{1}{\bar{\eta}} \sum_{i=1}^{N} \eta_k^{(i)} X_{0:k}^{(i)}. \tag{6.32}$$

As demonstrated previously in Section 6.2.4, the posterior PHD of the map is the expectation of the trajectory-conditioned PHDs and thus

$$v_k(m|X_{0:k}) = \frac{1}{\bar{\eta}} \sum_{i=1}^{N} \eta_k^{(i)} v_k^{(i)}(m|X_{0:k}^{(i)}). \tag{6.33}$$

If $\hat{\mathfrak{m}}_k = \int v_k(m|X_{0:k})dm$, is the mass of the posterior map PHD, the expected map estimate can then be extracted by choosing the $\hat{\mathfrak{m}}_k$ highest local maxima. The pseudo-code for the RB-PHD-SLAM filter are provided in tables 6.1, 6.2, while that of appropriate estimators is provided in Tables 6.3 and 6.4, which continues as Table 6.4. The following section presents the results and analysis of the proposed filter, with comparisons to standard SLAM algorithms.

6.4 Results and Analysis

This section presents the results and analysis of the proposed approach using both simulated and real experimental datasets. Initial comparisons are made with the FastSLAM [58] algorithm with maximum likelihood data association, using a mutual exclusion constraint and a 95% χ^2 confidence gate. These comparisons are demonstrated with a simulated single loop vehicle trajectory carrying a simulated range – bearing measuring sensor and a real, land based vehicle using a millimetre wave (MMW) radar for feature extraction. To further demonstrate the abilities of the RB-RFS-SLAM approach,

Table 6.1 RB-PHD-SLAM-Predict

```
Algorithm RB-PHD-SLAM-Predict
$\left(\{\eta_{k-1}^{(i)}, X_{0:k-1}^{(i)}, v_{k-1}^{(i)}(m|\mathcal{Z}_{k-1}, X_{k-1}^{(i)})\}_{i=1}^{N}, \mathcal{Z}_{k-1}, U_{k-1}\right)$
// Construct (6.25)
for i = 1 to $N$ do
// Sample pose
    $\tilde{X}_k^{(i)} \sim f_X(\tilde{X}_k^{(i)}|X_{k-1}^{(i)}, U_{k-1})$
// Predict map
    GMM-PHD-FBRM-Predict $\big(\mathcal{Z}_{k-1}, X_{k-1}^{(i)}, v_{k-1}^{(i)}(m|\mathcal{Z}_{k-1}, X_{k-1}^{(i)})$
end for
return $\left(\{\eta_{k-1}^{(i)}, \tilde{X}_k^{(i)}, v_{k|k-1}^{(i)}(m|\mathcal{Z}_{k-1}, X_{k-1}^{(i)})\}_{i=1}^{N}\right)$
```

Table 6.2 RB-PHD-SLAM-Update

```
Algorithm RB-PHD-SLAM-Update
$\left(\{\eta_{k-1}^{(i)}, \tilde{X}_k^{(i)}, v_{k|k-1}^{(i)}(m|\mathcal{Z}_{k-1}, X_{k-1}^{(i)})\}_{i=1}^{N}, \mathcal{Z}_k\right)$
for i = 1 to $N$ do
// Update map PHD
    GMM-PHD-FBRM-Update $\left(\mathcal{Z}_k, \tilde{X}_k^{(i)}, v_{k|k-1}^{(i)}(m|\mathcal{Z}_{k-1}, X_{k-1}^{(i)})\right)$
// Predicted PHD mass
    $\hat{\mathfrak{m}}_{k|k-1} = \sum_{j=1}^{J_{k|k-1}^{(i)}} \omega_{k|k-1}^{(i,j)}$
// Updated PHD mass
    $\hat{\mathfrak{m}}_k = \sum_{j=1}^{J_k^{(i)}} \omega_k^{(i,j)}$
    if( Empty Strategy TRUE ) do
// Updated trajectory weight of (6.20)
        $\tilde{\eta}_k^{(i)} = \left(c(z)^{|\mathcal{Z}_k|} \exp^{(\hat{\mathfrak{m}}_k - \hat{\mathfrak{m}}_{k|k-1} - \lambda_c)}\right)\eta_{k-1}^{(i)}$
    end if
    if( Single Feature Strategy TRUE ) do
// Select a given map state
        j = $\{i = 1, \ldots, J_k^{(i)} | m^{(i,j)} = \bar{m}\}$
        $a = (1 - P_D)c(z)^{|\mathcal{Z}_k|} + P_D\omega_{k|k-1}^{(i,j)} \times \sum_{z \in \mathcal{Z}_k} (c(z)^{|\mathcal{Z}_k|-1})\mathcal{N}(z; z_{k|k-1}^{(i,j)}, S_k^{(i,j)})$
        $b = \exp^{(\hat{\mathfrak{m}}_{k|k-1} - \hat{\mathfrak{m}}_k + \lambda_c)} \omega_k^{(i,j)}$
// Update trajectory weight of (6.21)
        $\tilde{\eta}_k^{(i)} = \frac{a}{b}\tilde{\eta}_{k|k-1}^{(i)}$
    end if
end for
return $\left(\{\tilde{\eta}_k^{(i)}, \tilde{X}_k^{(i)}, v_k^{(i)}(m|\mathcal{Z}_k, X_k)\}_{i=1}^{N}\right)$
```

Table 6.3 RB-PHD-SLAM-MAPestimate

```
Algorithm RB-PHD-SLAM-MAPestimate
    $\left(\{\eta_{k-1}^{(i)}, X_k^{(i)}, v_{k|k-1}^{(i)}(m|\mathcal{Z}_{k-1}, X_{k-1})\}_{i=1}^{N}, \mathcal{Z}_k, T_{\text{feature}}\right)$
// Initialise the map estimate
$\hat{\mathcal{M}}_k = \emptyset$
I = {1,...,N}
// Get index of max weight component
j = $\arg\max_{i \in I} \eta_k^{(i)}$
// Estimated Trajectory
$\hat{X}_{0:k} = X_{0:k}^{(j)}$
// Estimate Map from corresponding map PHD
for i = 1 to $J_k^{(j)}$ do
    if $\omega_k^{(j,i)} > T_{\text{feature}}$
// concatenate estimate
        $\hat{\mathcal{M}}_k = [\hat{\mathcal{M}}_k \ \mu_k^{(j,i)}]$
    end if
end for
// RB-PHD-SLAM MAP Estimate
return $\left(\hat{X}_{0:k}, \hat{\mathcal{M}}_k\right)$
```

further, somewhat more complicated, experiments are carried out in which the benchmark algorithms used are the classical FastSLAM [58] but with with Multiple Hypothesis Data association [67] and the Joint Compatibility Branch and Bound (JCBB) EKF [39]. In this second set of experiments, in the simulation, multiple vehicle loop trajectories are executed and for the real experiment, a much larger scenario, where accurate SLAM performance in the presence of high clutter is essential, is demonstrated at sea, in a coastal environment, using an “Autonomous kayak” [59] as the vehicle and a commercially available X-Band radar.

In all experiments, the ‘single feature map’ trajectory weighting of equation 6.21 is adopted for the proposed RB-PHD-SLAM filter. An implementation using the ‘empty map update’ of equation 6.20 was presented in [56]. While any feature can theoretically be selected to generate the trajectory weighting, in this implementation, that which generates the maximum likelihood amongst all measurements is chosen. A comprehensive study as to the best suited feature selection strategies is left to future work.

Current SLAM filters deal with clutter through ‘feature management’ routines, such as the landmark’s quality [1] or a binary Bayes filter [58]. These operations are typically independent of the joint SLAM filter update, whereas the proposed approach unifies feature management, data association

Table 6.4 RB-PHD-SLAM-EAPestimate

```
Algorithm RB-PHD-SLAM-EAPestimate
        $\left(\{\eta_{k-1}^{(i)}, X_k^{(i)}, v_{k|k-1}^{(i)}(m|\mathcal{Z}_{k-1}, X_{k-1})\}_{i=1}^{N}, \mathcal{Z}_k, T_{\text{feature}}, D_{\min}\right)$
1. $\Omega_k = 0$
2. for i = 1 to $N$ do
3.     $\Omega_k = \Omega_k + \eta_k^{(i)}$
4. end for
// expected trajectory
5. $\hat{X}_{0:k} = \frac{1}{\Omega_k} \sum_{i=1}^{N} \eta_k^{(i)} X_{0:k}^{(i)}$
// Initialise number of Gaussian components
6. l = 0
7. for i = 1 to $N$
8.     for j = 1 to $J_k^{(i)}$
9.         l = l+1
10.        $\bar{\omega}_k^{(l)} = \eta_k^{(i)} \omega_k^{(i,j)}$
11.        $\bar{\mu}_k^{(l)} = \mu_k^{(i,j)}$
12.        $\bar{P}_k^{(l)} = P_k^{(i,j)}$
13.    end for
14. end for
15. $\mathcal{R} = \{1, \ldots, l\}$
// Initialise number of merged Gaussian components
16. L = 0
// Gaussian merging
17. do while $\mathcal{R} \neq \emptyset$
// Increment component counter
18.    L = L+1
// Get index of max weight component
19.    j = $\arg\max_{r \in \mathcal{R}} \bar{\omega}_k^{(r)}$
// Cluster those within distance $D_{\min}$
20.    K = $\{r \in \mathcal{R} | (\bar{\mu}_k^{(r)} - \bar{\mu}_k^{(j)})^T [\bar{P}_k^{(r)}]^{-1} (\bar{\mu}_k^{(r)} - \bar{\mu}_k^{(j)}) \leq D_{\min}\}$
// Combine component weights
21.    $\tilde{\omega}_k^{(L)} = \sum_{i \in K} \bar{\omega}_k^{(i)}$
// Weighted average mean
22.    $\tilde{\mu}_k^{(L)} = \frac{1}{\tilde{\omega}_k^{(L)}} \sum_{i \in K} \bar{\omega}_k^{(i)} \bar{\mu}_k^{(i)}$
```

Table 6.4 (*Continued*)

```
// Combined covariance
// Remove K from $\mathcal{R}$ and repeat
    $\tilde{P}_k^{(L)} = \frac{1}{\tilde{\omega}_k^{(L)}} \sum_{i \in K} \bar{\omega}_k^{(i)} \big(\bar{P}_k^{(i)} + (\tilde{\mu}_k^{(L)} - \bar{\mu}_k^{(i)})(\tilde{\mu}_k^{(L)} - \bar{\mu}_k^{(i)})^T\big)$
    $\mathcal{R} = \mathcal{R} - K$
end while
for i = 1 to $L$ do
    if $\tilde{\omega}_k^{(i)} > T_{\text{feature}}$
// concatenate estimate
        $\hat{\mathcal{M}}_k = [\hat{\mathcal{M}}_k \ \tilde{\mu}_k^{(i)}]$
    end if
end for
// RB-PHD-SLAM EAP Estimate
return $\big(\hat{X}_{0:k}, \hat{\mathcal{M}}_k\big)$
```

and state filtering into a single Bayesian update. While these methods have been used successfully, they generally discard the sensor's detection (P_D) and false alarm (P_{FA}) probabilities and thus can be erroneous when subject to large clutter rates and/or measurement noise. Since the proposed approach assumes knowledge of these probabilities, as seen in equation 6.14, a modified feature management routine coined the 'feature existence filter' (see Appendix C), which incorporates both P_D and P_{FA}, is used with the benchmark algorithms in an attempt to be 'fairer' to them in the comparisons.

To quantify the map estimation error, a metric must be adopted which jointly evaluates the error in the feature location *and* number estimates. Current methods typically examine the location estimates of a selected number of features and obtain their Mean Squared Error (MSE) using ground truth [1]. As such, vector-based error metrics are applied to feature maps and the error in the estimated *number* of features is neglected. While there are several metrics for finite-set-valued estimation error, that of [23] has been demonstrated to be most suitable [64], [56]. Therefore, the set map error metric described in Chapter 4 (equation 4.6) is therefore once again used to gauge the mapping performance in terms of estimated and actual feature number, as well spatial error. In the following sections, this metric along with the root mean squared error (RMSE) and graphical comparisons are used to demonstrate the merits of the RB-PHD-SLAM filter.

6.4.1 Simulated Datasets

Comparisons of RB-RFS-SLAM with standard vector based SLAM algorithms are firstly presented in the form of simulated trials due to the ease of generating known ground truth (trajectories and maps) for estimation error evaluation.

6.4.1.1 Simulation: Single Loop Trajectory

The filter parameters used for the single loop trajectory simulated trial are shown in Table 6.5. The measurement noise was inflated to hinder data association in the vector-based filter. For both filters, both the maximum a posterior (MAP) and expected a posterior (EAP) trajectories are reported. For FastSLAM, the map of the highest weighted trajectory estimate is used as the map estimate, while for the proposed filter, both the map of the highest weighted trajectory and the EAP map estimate are determined for comparison. 50 Monte Carlo (MC) trials were carried out.

Table 6.5 Filter parameters used for the single loop trajectory trial.

Filter Parameter	Value
Velocity input standard deviation (std)	1.5 m/s
Steering input std.	9.5^o
Range measurement std.	1.75m
Bearing measurement std.	3.5^o
Probability of Detection P_D	0.95
Clutter rate λ_c	5m^{-2}
Sensor maximum range	10m
Sensor Field-of-View (FoV)	360^o
Feature existence threshold	0.5

Figure 6.1 shows a sample of the raw data used in the trials, with the green circles depicting the true feature locations. A quantitative evaluation of the estimation results is provided through the RMSE, along with standard deviations, of the trajectory estimate as shown in figure 6.2. Without knowledge of P_D and P_{FA}, the benchmark approach can be made to appear highly erroneous due to poor feature management. Incorporating this information can improve the result, however the feature management is still effectively a post-filter update processing method. The RB-PHD-SLAM algorithm is significantly more robust due to the RFS feature map representation and Bayesian recursion which jointly performs feature management and state estimation.

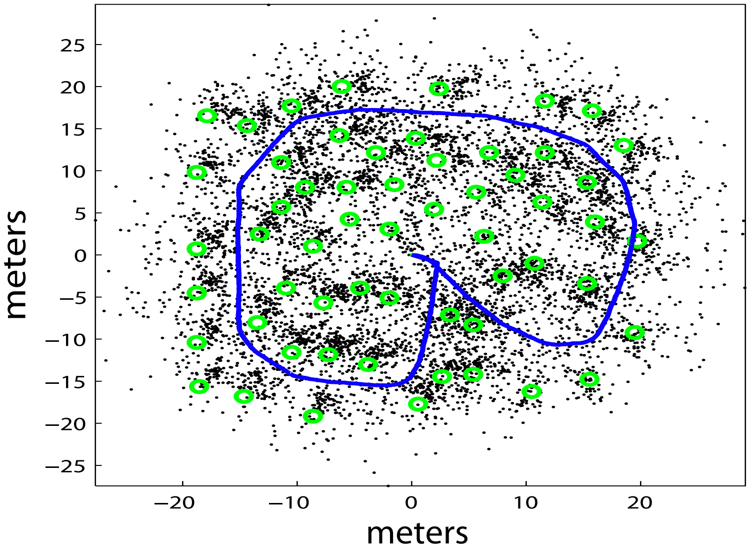

Fig. 6.1 The simulated environment showing point features (green circles). A sample measurement history (black points) plotted from a sample noisy trajectory (blue line) is also shown.

In terms of the map estimation, figure 6.3 depicts both the true and estimated number of features as the robot explores the map, with the proposed approach closely tracking the true number. Note that since this trial is a simulation, the true number of features which have entered the vehicle's FoV during its entire trajectory, can be calculated exactly. Since this result does not examine the locations of the estimated features, the set metric of equation 4.6 is used to compare map estimates, as shown in figure 6.4. The figure shows the 'ideal' mapping error (i.e. every feature is instantly estimated by its true coordinates when it enters the sensor FoV), which converges to zero once all features in the map have been scanned. The mean and std of the map estimates for both the benchmark and proposed approach are plotted, with that of the RB-PHD-SLAM filter reporting less map estimation error. A qualitative depiction of the posterior estimates from both approaches is provided in figures 6.5 and 6.6, demonstrating the usefulness of the RFS approach and the associated RB-PHD-SLAM filter. In both figures, the green line and circles represent the ground truth vehicle trajectory and feature locations respectively. The black crosses represent the estimated map. In the case of FastSLAM, this is derived with respect to the MAP FastSLAM trajectory estimate (the particle (trajectory) with the final maximum weight). In each figure, the blue line indicates the MAP trajectory estimate, which corresponds to the particle with the maximum weight, at each time and the red line corresponds to the expected trajectory estimate, which is the weighted average of all particles at each time (see Tables 6.3 and 6.4). Given that the filter jointly incorporates data association and feature number uncertainty into its Bayesian recursion, it is more robust to large sensing error, as it does not rely on hard measurement-feature assignment decisions. Furthermore, it jointly estimates the number of features and their locations, alleviating the need for popular feature management methods [1], [58].

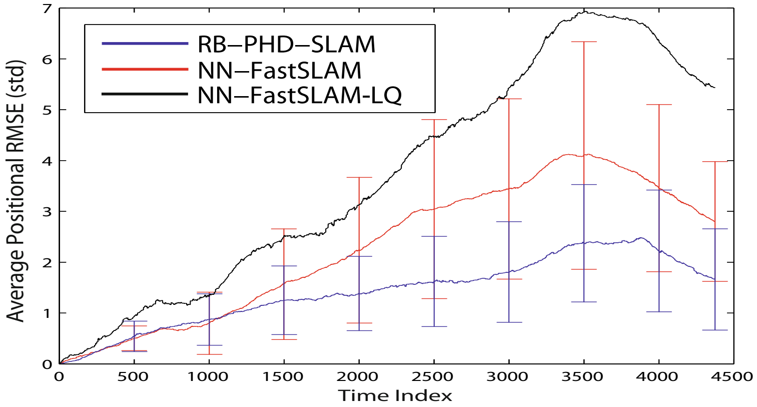

Fig. 6.2 The mean and standard deviation of the expected trajectory estimates of RB-PHD-SLAM vs. that of FastSLAM over 50 MC runs. LQ refers to an implementation with the 'landmark quality' method of [1].

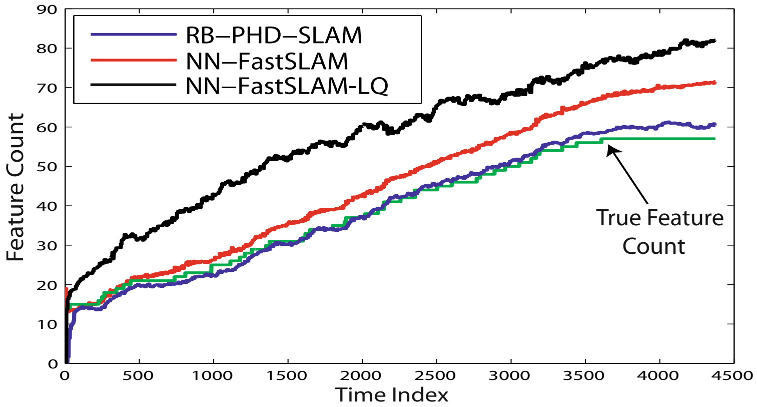

Fig. 6.3 The average estimated number of features in the map vs. ground truth for each approach. The feature number estimate of RB-PHD-SLAM can be seen to closely track that of the ground truth. Clearly there is no distinction between correctly estimated feature and false feature in this result.

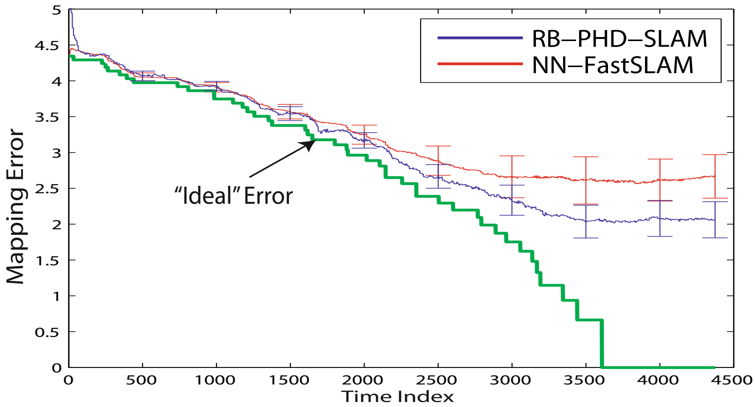

Fig. 6.4 A comparative plot of the mean and standard deviation of the map estimation error vs. time. The error incorporates that of the number of features, shown in figure 6.3 as well as their positional estimates. Note that the 'ideal' error converges to zero, an important property for SLAM filters and map estimation comparisons.

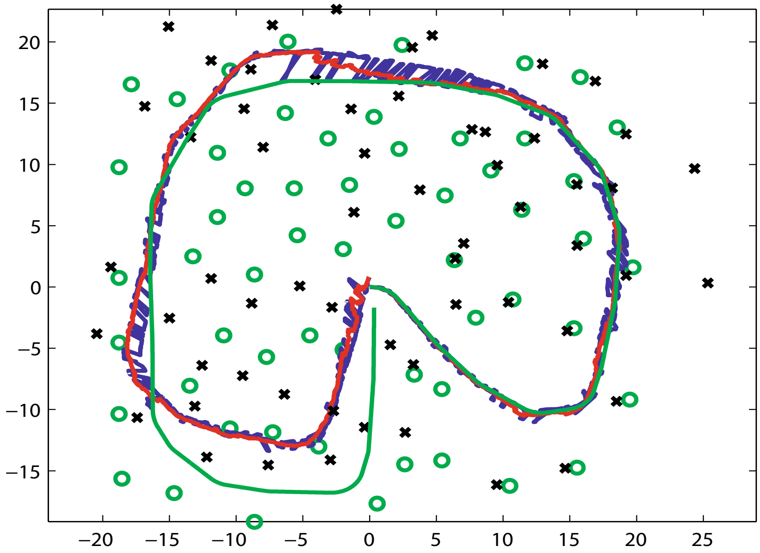

Fig. 6.5 A sample posterior estimate from FastSLAM showing error in the estimated trajectory and feature map. The green circles and line show the ground truth feature locations and path respectively. The black crosses show the FastSLAM estimated map (feature locations). The blue line shows the MAP trajectory estimate and the red line shows the expected trajectory estimate.

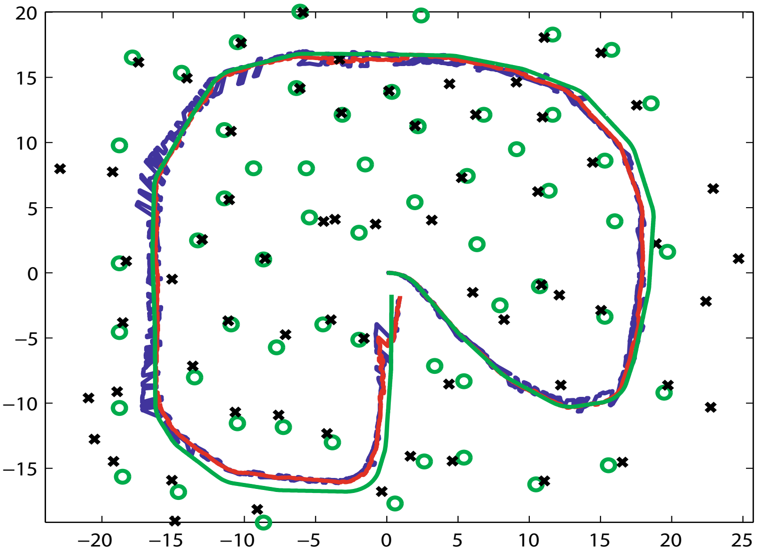

Fig. 6.6 The posterior estimate given the same inputs / measurements as those used in figure 6.5. Again, the green circles and line show the ground truth feature locations and path respectively. The black crosses show the RB-PHD-SLAM estimated map (feature locations). The RB-PHD-SLAM filter demonstrates its robustness and accuracy given high clutter and data association ambiguity.

6.4.1.2 Simulation: Multiple Loop Trajectories

The parameters for the more complex, multiple loop trajectory, simulated trials are shown in table 6.6. A 95% validation gate is used throughout. For each

Table 6.6 Filter parameters used for the single loop trajectory trial.

Filter Parameter	Value
Velocity input standard deviation (std)	2.0 m/s
Steering input std.	5.0^o
Range measurement std.	1.00m
Bearing measurement std.	2.0^o
Probability of Detection P_D	0.95
Clutter rate λ_c	20m^{-2}
Sensor maximum range	10m
Sensor Field-of-View (FoV)	360^o
Feature existence threshold	0.5

SLAM filter, 50 Monte Carlo (MC) trials were carried out in which all methods received identical sequences of control inputs and measurements. The RB based filters used 50 trajectory particles each, while for MH-FastSLAM a maximum limit of 2000 particles (number of hypotheses considered prior to resampling) was used.

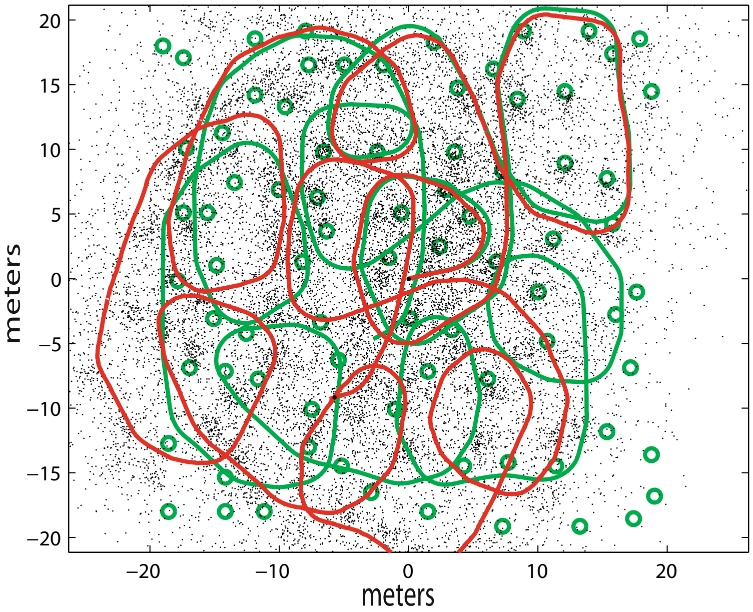

Fig. 6.7 The simulated environment showing point features (green circles) and true vehicle trajectory (green line). A sample measurement history plotted from a sample noisy trajectory (red line) is also shown (black points).

Figure 6.7 shows a sample of the raw input data used in the trials, superimposed onto the ground truth feature map and path. A comparison of the average trajectory estimation errors for all three filters is then presented in Figure 6.8. In terms of the estimated trajectory, the first order RB-PHD-SLAM algorithm can be seen to outperform the vector based FastSLAM with robust data association, but does not quite achieve the estimation accuracy of JCBB-EKFSLAM. This is primarily due to the fact that JCBB is very conservative in its choice of measurement-feature associations (jointly compatible constraint) resulting in very few false association pairs influencing the trajectory estimation. However, later results in Figures 6.9, 6.10 and 6.12 highlight the drawbacks of achieving such impressive localisation results.

In terms of the map estimation component of each SLAM algorithm, Figure 6.9 depicts both the true and estimated number of features as the vehicle explores the map, with the proposed RB-PHD-SLAM approach seen to closely track the true number of features in the explored map. Erroneous associations for the MH-FastSLAM approach result in excessive feature declarations, while the conservative (only including those which are jointly compatible) association decisions of JCBB-EKF SLAM reduces the number of correct associations. Since vector based feature management routines are typically dependant on the data association decisions, this dramatically influences the map estimation error.

Incorporating the estimation error in both the number and location of features in the map, Figure 6.10 plots the map error distance of equation 4.6 for each approach. Note that in order to generate this result, the vector based maps of FastSLAM and JCBB-EKFSLAM are temporarily 'assumed' to be sets. The proposed method can be seen to report the least mapping error due it is robust ability to jointly incorporate uncertainty in feature locations and numbers, while erroneous feature estimates contribute to increased mapping error for the vector based approaches. A qualitative depiction of the posterior estimates is provided in Figure 6.11, demonstrating the usefulness of the RFS-SLAM framework and the proposed RB-PHD-SLAM filter.

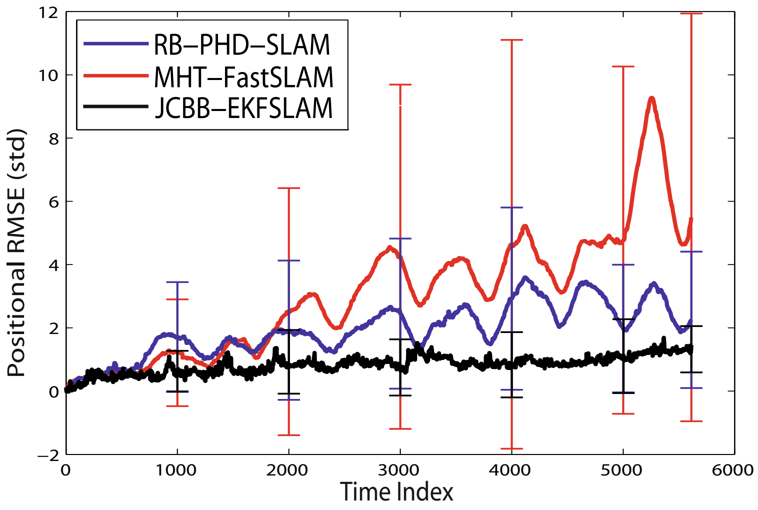

Fig. 6.8 The mean and standard deviation of the trajectory estimates from each filter over 50 MC runs versus time.

6.4.2 A Note on Computational Complexity

As can be observed from the implementation of Section 6.3, the computational complexity of RB-PHD-SLAM is, $\mathcal{O}(\mathfrak{m}_k \mathfrak{z}_k N)$ i.e. linear in the number of features (in the FoV), linear in the number of measurements and linear in

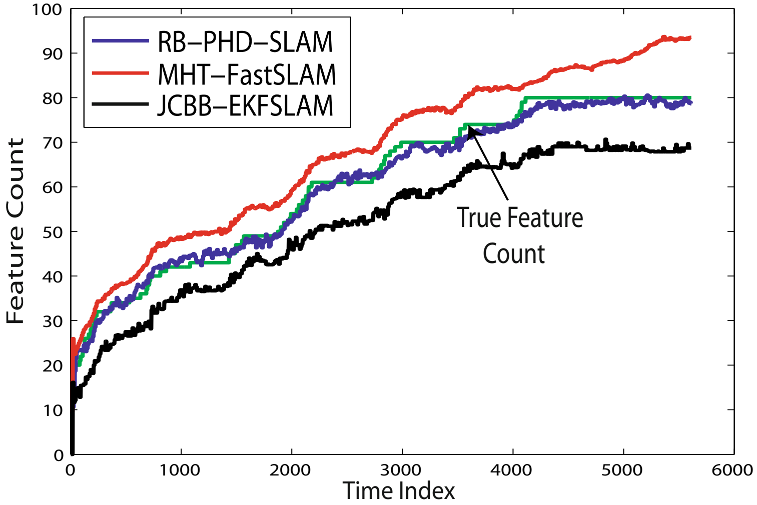

Fig. 6.9 The average estimated number of features in the map for each filter versus time, compared to the ground truth number of features in the explored map $\mathcal{M}_k$. The feature number estimate of RB-PHD-SLAM can be seen to closely track that of the ground truth.

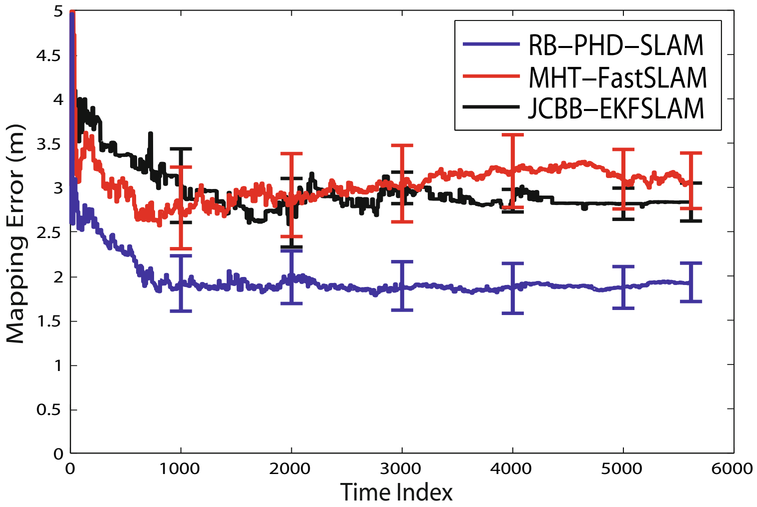

Fig. 6.10 A comparative plot of the mean and standard deviation of the map estimation error for each filter vs. time. At any given time, for the ideal case, the mapping error from equation 4.6 wrt. the explored map is zero.

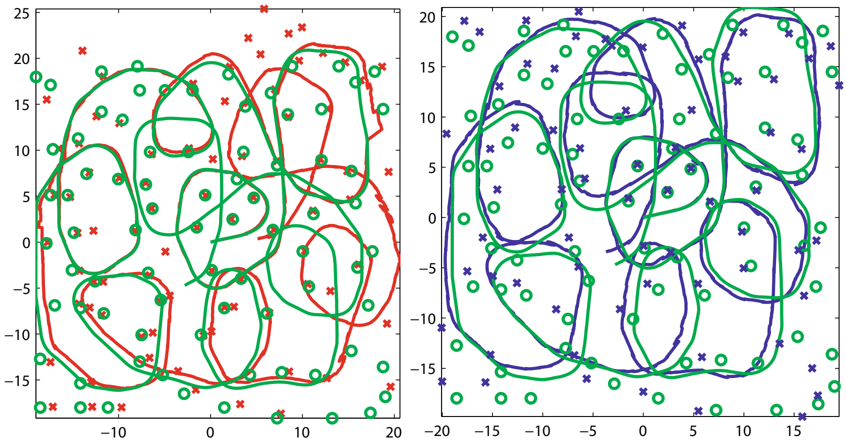

Fig. 6.11 Comparisons of the posterior SLAM estimates from MH-FastSLAM (left, red) and the proposed RB-PHD-SLAM (right, blue). The ground truth trajectory and map are represented by the green line and circles respectively. The RB-PHD-SLAM filter demonstrates its robustness and accuracy given high clutter and data association ambiguity.

the number of trajectory particles. The linear complexity of each particle in the mapping filter was verified previously in Figure 4.8.

For a single thread implementation, Figure 6.12 shows that the computational time is comparable with that of the MH-FastSLAM algorithm, both of which are less expensive than JCBB-EKF SLAM as its hypothesis tree grows in the presence of high clutter. Note that due to the Rao-Blackwellised structure of RB-PHD-SLAM, binary tree based enhancements, such as those applied to traditional FastSLAM [58], can be readily developed to further reduce the complexity to $\mathcal{O}\big(\mathfrak{z}_k N \log(\mathfrak{m}_k)\big)$. Furthermore, in contrast to data association based methods, the proposed approach admits numerous other computational enhancements, since the map PHD update can be segmented, executed in parallel and subsequently fused for state estimation. This is in contrast to DA based approaches which are scalable.

6.4.3 Outdoor Experiments

6.4.3.1 Land Based SLAM with MMW Radar

This section discusses the performance of the proposed framework, using a millimetre wave radar SLAM dataset in a university car park environment. Millimetre wave radar offers numerous advantages over standard laser-based systems, returning a power vs. range spectrum. This allows for customised detection algorithms to be developed, however it can be prone to high clutter

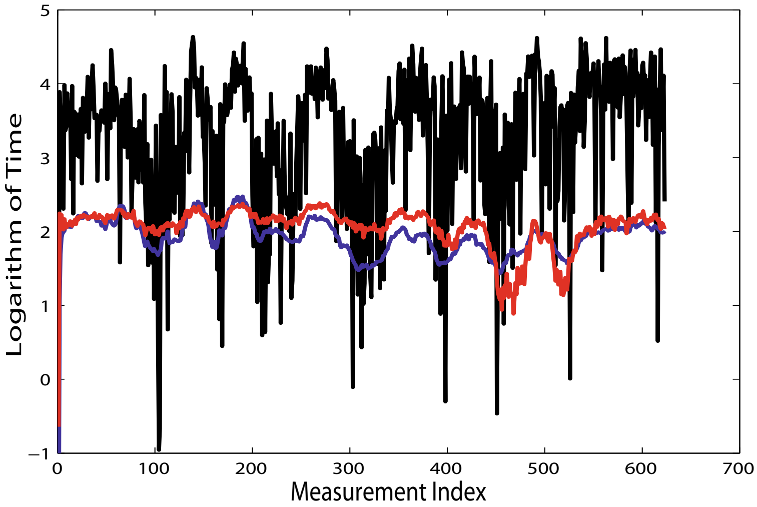

Fig. 6.12 A comparison of the computation time per measurement update for RB-PHD-SLAM (blue), MH-FastSLAM (red) and JCBB-EKFSLAM (black).

rates [16]. The extracted radar point clusters, plotted relative to the odometry only pose estimates of the vehicle, as well as the odometry pose estimates themselves are depicted in figure 6.13 (left). The information displayed in this figure can be thought of as the information input to the SLAM algorithms, which must be processed to yield the best estimated trajectory and map. Given the tree coverage and surrounding buildings in the area, GPS is generally not available. Ground truth was thus obtained by manually matching successive scans from a laser range finder which was also mounted on the vehicle, with graphical verification also provided in figure 6.13 (right). The vehicle was driven at approximately 1.5m/s around 3 loops, with a control input frequency of 10Hz and a radar measurement frequency of 2.5Hz. The car park environment is comprised of buildings, bushes, trees, fire hydrants, curbs, medians, a car etc.

Given the small-sized loop, the maximum range of the radar was set at 15m and both FastSLAM, with maximum likelihood data association, and RB-PHD-SLAM were executed on the dataset. Figure 6.14 depicts the posterior estimated trajectory and map using the FastSLAM algorithm (left) and that from RB-PHD-SLAM (right), given the same control input samples. Given the noisy measurements from the radar sensor, the merits of the proposed approach are demonstrated. It should be noted that, as is the case with any experimental dataset, the ground truth feature map is extremely difficult to obtain, making it challenging to evaluate the feature map estimation error.

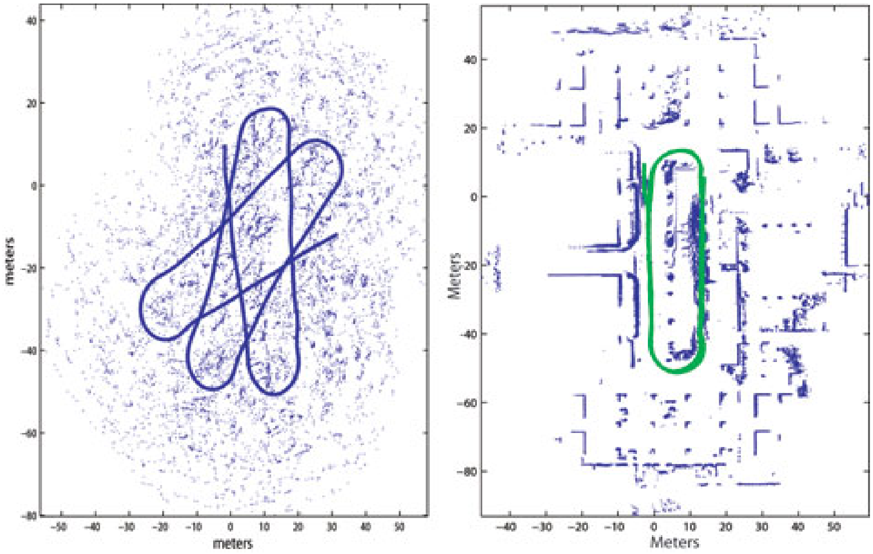

Fig. 6.13 Left: Odometery and extracted clusters from the radar data, representing the raw inputs to the SLAM algorithms. Right: The ground truth trajectory (green) obtained by matching laser data due to a lack of GPS data.

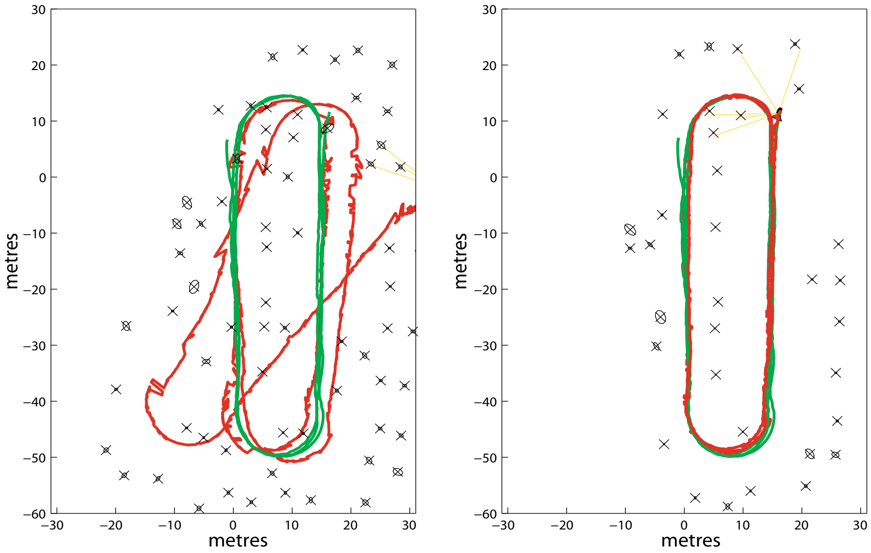

Fig. 6.14 Left: The posterior estimate from FastSLAM using the radar-based car park dataset. Right: The posterior estimate from RB-PHD-SLAM using the same dataset. The proposed integrated Bayesian framework for SLAM, incorporating DA and feature management enhances the robustness of the SLAM algorithm given noisy measurements.

6.4.3.2 Sea Based SLAM with X-Band Radar

This section discusses the filter's performance in a surface-based marine environment, using an X-band radar mounted on a powerboat. In order to maximise the detection of all sea surface point features (comprising boats, buoys, etc.), a low detection threshold is required, which subsequently increases the clutter rate. GPS data is available for measuring the ground truth trajectory, while sea charts and data from surrounding vessels' Automatic Identification Systems provide the feature map ground truth. The test site is off the Southern coast of Singapore, as shown in Figure 6.15, where the boat was driven in a loop trajectory of 13Km. Adaptive thresholding methods were applied to extract relative point measurements from the radar data [59]. The maximum range of the radar, logging at 0.5Hz, was limited to 1Km. While heading measurements were available via a low grade on-board single axis gyroscope, due to the lack of Doppler velocity logs, the speed was estimated at 8 knots (4.1 m/s).

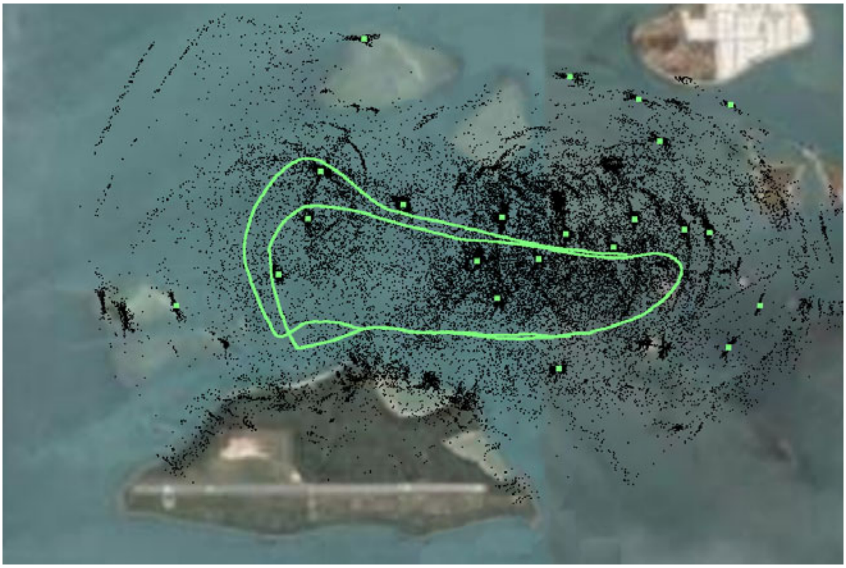

Fig. 6.15 Overview of the test site ($1^o 13'$ N,$103^o 43'$ E), showing the GPS trajectory (green line) and GPS coordinates-ordinates (green dots) of the point feature map. The point feature measurement history is also provided (black dots).

Figure 6.16 compares the posterior SLAM estimates from MH-FastSLAM and RB-PHD-SLAM, with Figure 6.17 comparing the estimated map sizes. The proposed approach can be seen to generate more accurate localisation and feature number estimates, however it can also be seen that some feature estimates are misplaced in comparison to the ground truth feature map. The framework is still demonstrated to be useful for high clutter feature-based SLAM applications.

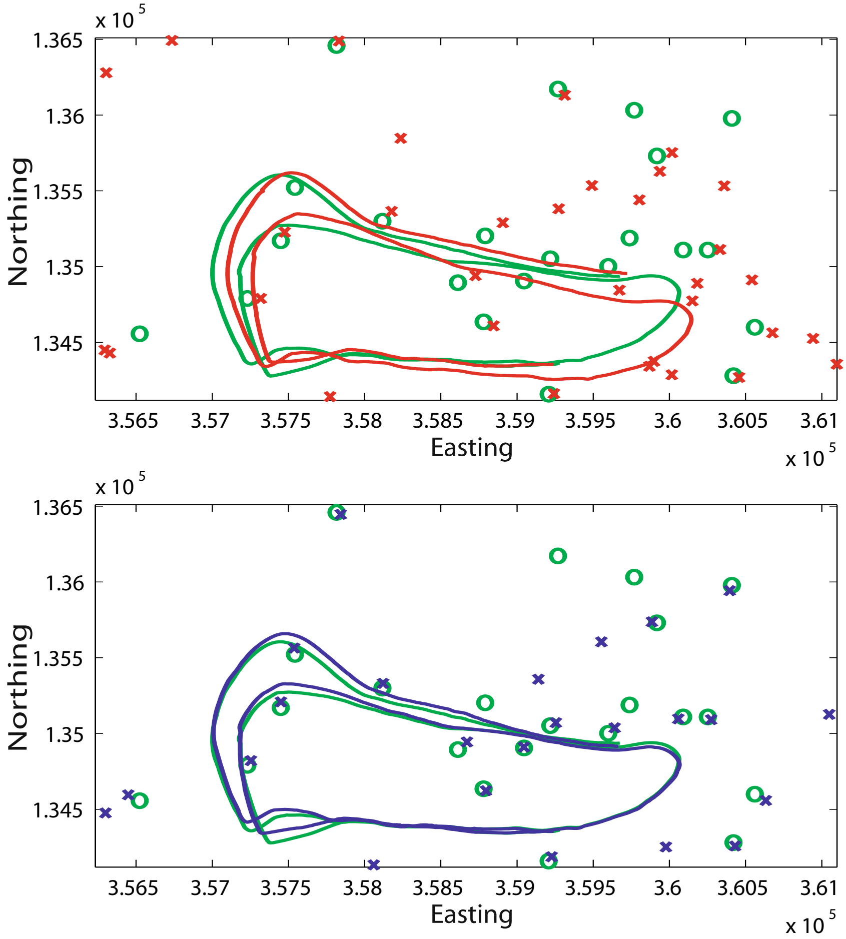

Fig. 6.16 Top: The posterior SLAM estimate (red) from MH-FastSLAM and Bottom: The posterior SLAM estimate (blue) from RB-PHD-SLAM, in comparison to the ground truth (green).

6.5 Summary

This chapter presented a tractable solution for the feature-based SLAM problem. The finite set representation of the map admits the notion of an expected

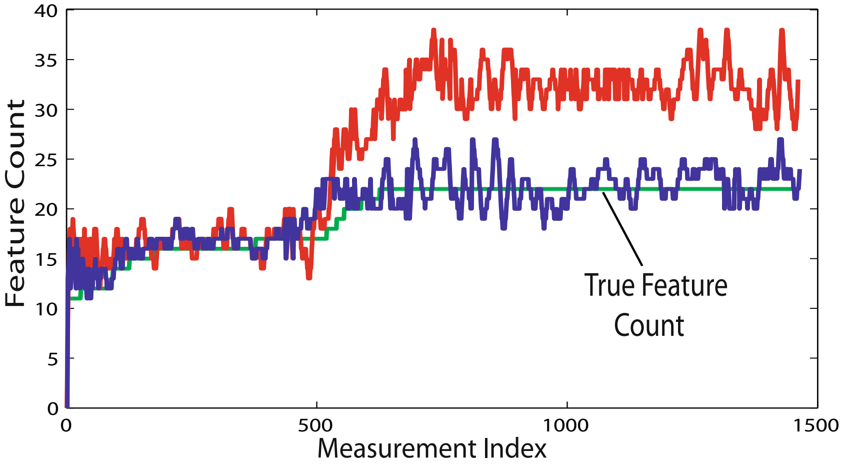

Fig. 6.17 Comparison of the number of estimated features for each approach. The noisy estimates are likely due to deviations from the Poisson clutter assumption in places.

map in the form of a PHD or intensity function. A Rao-Blackwellised implementation of the filter was proposed, in which the PHD of the map was propagated using a Gaussian mixture PHD filter, and a particle filter propagated the vehicle trajectory density. A closed form solution for the trajectory weighting was also presented, alleviating the need for approximation, which is commonly used.

Analysis was carried out, both in a simulated environment through Monte Carlo trials and an outdoor SLAM experimental dataset based on a millimetre wave radar sensor. Results demonstrated the robustness of the proposed filter, particularly in the presence of large data association uncertainty and clutter, illustrating the merits of adopting an RFS approach to SLAM.

In terms of its computational complexity, the Rao-Blackwellised SLAM filter was shown to be linear in the number of estimated features, measurements and trajectory particles. It should be noted that computational enhancements are possible, in terms of parallelisable operations, which are not possible with vector based approaches requiring data association.

6.6 Bibliographical Remarks

The RFS approach to SLAM was first suggested in [54] with preliminary studies using 'brute force' implementations shown in Chapter 5. The approach modelled the joint vehicle trajectory and map as a single RFS, and recursively propagated its first order moment.

Initial results of a Rao-Blackwellised (RB) implementation of the PHD-SLAM filter, were presented in [56]. This chapter extends [56], to present a more rigorous analysis of the RFS approach to SLAM, an improved version of the PHD-SLAM filter as well as real and simulated experimental results, and is an extended version of [57]. The merits of the RFS approach are demonstrated, particularly in situations of high clutter and data association ambiguity.

A factorised approach to SLAM was established in the, now well known, FastSLAM concept [58]. However, this chapter has shown that the same factorisation method applied to vectors in FastSLAM, cannot be applied to sets, since it results in invalid densities in the feature space. Therefore one of the main contributions of this chapter is a technique which allows such a factorisation to be applied to sets in a principled manner.