Chapter 5
An RFS 'Brute Force' Formulation for Bayesian SLAM

5.1 Introduction

The feature-based (FB) SLAM scenario is a vehicle moving through an environment represented by an unknown number of features. The classical problem definition is one of "*a state estimation problem involving a variable number of dimensions*" [28]. The SLAM problem requires a robot to navigate in an unknown environment and use its suite of on board sensors to both construct a map and localise itself within that map without the use of any *a priori* information. Often, in the planar navigation context, a vehicle is assumed to acquire measurements of its surrounding environment using on board range-bearing measuring sensors. This requires joint estimates of the three dimensional robot pose (Cartesian x and y coordinates, as well as the heading angle θ), the number of features in the map as well as their two dimensional Euclidean coordinates. For a real world application, this should be performed incrementally as the robot manoeuvres about the environment. As the robot motion introduces error, coupled with a feature sensing error, both localisation and mapping must be performed simultaneously [8]. As mentioned in Chapter 2, for any given sensor, an FB decision is subject to detection and data association uncertainty, spurious measurements and measurement noise, as well as bias.

The majority of proposed algorithms, stemming from the seminal work of [8], adopt an augmented state containing both the vehicle pose estimate and the estimate of the map. It is important to note however, that the example discussed in [8] consisted of a map containing features of unity detection probability, assumed the measurement-feature association was known, and that the sensor reported no spurious measurements. With these strict assumptions, the Kalman based SLAM estimate is indeed Bayes optimal. This work was incorporated into multiple Kalman-based solutions to the SLAM problem [52].

J. Mullane et al.: Random Finite Sets for Robot Mapping & SLAM, STAR 72, pp. 79–96.
springerlink.com

The presence of detection uncertainty and spurious measurements have also been long acknowledged in the SLAM community, and subsequently feature initialisation and termination algorithms are frequently incorporated into the vector-based SLAM algorithm [52] (also shown in figure 1.1). This chapter again emphasises that these are required due to the inability of a vector representation to incorporate uncertainty in the *number* of dimensions, and highlights that such methods (which are independent of the filter recursion) compromise filter performance. As shown in this chapter, this can result in filter divergence and large mapping error, especially in scenarios of high clutter and large data association ambiguity. Through the re-formulation of the classical SLAM problem, and by explicitly incorporating the problem of a variable number of dimensions into the filter recursion, increased robustness in noisy scenarios will be demonstrated.

A finite set-valued measurement allows for the inclusion of spurious measurements directly into the measurement equation, as introduced in equation 2.10, which is then the union of the set-state dependant measurements (as is the case in the classical Bayesian SLAM formulations) and the set of spurious measurements. A finite set-valued map state can be constructed from the set union of the existing features and the new features which may appear in the map due to the motion of the robot, as introduced in equation 3.21.

This chapter subsequently casts the SLAM problem into a random set theoretic filtering problem that incorporates the joint estimation of the vehicle pose, feature number and corresponding feature locations. The term "Brute force" is used to describe the concept presented here, since each estimated feature is augmented with a hypothesised vehicle trajectory. While this is theoretically sound, and simple to implement, its computational burden becomes obvious when one considers that a single PHD is propagated, which requires many Gaussian functions to represent a single feature, since that feature can be augmented with many hypothesised trajectories. Although computationally intractable in any realistic environment with significant numbers of features, its implementation is included here to demonstrate a viable, and theoretically simple, RFS based SLAM solution. A more elegant, and computationally tractable solution, based on Rao-Blackwellisation, will be the subject of Chapter 6.

This chapter is organised as follows. A recap of the RFS Bayesian, SLAM formulation is given in Section 5.2, based on the RFS map transition function introduced in Chapter 3 and the RFS measurement model, introduced in Chapter 2. Section 5.3 introduces an augmented joint vehicle-map RFS to incorporate vehicle location uncertainty. The PHD of the augmented state recursion is then presented, and the PHD-SLAM filter is introduced. Using Gaussian noise assumptions, an extended-Kalman Gaussian Mixture (GM) implementation is developed in Section 5.4. This implementation accounts for the non-linearity in the measurement equation and jointly estimates the feature number in the map, their corresponding states and the vehicle pose. Importantly, this can be achieved without the need for explicit data

association decisions and/or feature management algorithms. Simulated mapping and pose estimation results are shown in Section 5.6 where the proposed GM-PHD SLAM filter is tested on simulated data which contains a large number of spurious measurements. Results show the efficacy of the proposed framework for solving the Bayesian SLAM problem. Real RFS based SLAM results in both outdoor and coastal environments (at sea) will be given for the formulation presented in chapter 6.

5.2 RFS Formulation of the Bayesian SLAM Problem

The starting point for the RFS, SLAM formulation is a recap of the map transition equation for RFSs, introduced in Chapter 2 (equation 2.10) and the RFS measurement model introduced in Chapter 3 (equation 3.21).

To incorporate the fact that new features enter the map, $\mathcal{M}_k$, with time, let the map state $\mathcal{M}_k$ be an RFS which evolves in time according to,

$$\mathcal{M}_k = \mathcal{M}_{k-1} \cup \mathcal{B}_k \tag{5.1}$$

comprising the set union of the previous RFS multi-feature map, $\mathcal{M}_{k-1}$ and the RFS of the new features at time k, $\mathcal{B}_k$. These sets are assumed mutually independent.

As in equation 2.10 (repeated below for convenience), to contend with the realistic situation of missed detections and clutter, the measurement is also modelled as an RFS. Given the vehicle state, X_k, and the map $\mathcal{M}_k$, the measurement consists of a set union,

$$\mathcal{Z}_k = \bigcup_{m \in \mathcal{M}_k} \mathcal{D}_k(m, X_k) \cup \mathcal{C}_k(X_k) \tag{5.2}$$

where $\mathcal{D}_k(m, X_k)$ is the RFS of the measurement generated by a feature at m and $\mathcal{C}_k(X_k)$ is the RFS of the spurious measurements at time k.

For each feature, $m \in \mathcal{M}_k$, and $z \in \mathcal{Z}_k$,

$$\mathcal{D}_k(m, X_k) = \{z\} \tag{5.3}$$

with probability density $P_D(m, X_k)g(z|m, X_k)$ and $\mathcal{D}_k(X_k, m) = \emptyset$ with probability $1 - P_D(m, X_k)$, where $P_D(m^{\mathfrak{m}_k}, X_k)$ is the probability of the sensor detecting the $\mathfrak{m}_k^{th}$ feature from pose X_k. Using Finite Set Statistics [3], the probability density that the sensor produces the measurement set $\mathcal{Z}_k$ given the vehicle state X_k and map $\mathcal{M}_k$ at time k is then given by [2]:

$$g_k(\mathcal{Z}_k|\mathcal{M}_k, X_k) = \sum_{\mathcal{W} \subseteq \mathcal{Z}_k} g_{\mathcal{D}}(\mathcal{W}|\mathcal{M}_k, X_k) g_{\mathcal{C}}(\mathcal{Z}_k - \mathcal{W}) \tag{5.4}$$

with $g_{\mathcal{D}}(\mathcal{W}|\mathcal{M}_k, X_k)$ denoting the density of the RFS of observations, $\mathcal{D}_k(m, X_k)$, generated from the features in the observed map $\mathcal{M}_k$ given the state of the vehicle, and $g_{\mathcal{C}}(\mathcal{Z}_k - \mathcal{W})$ denoting the density of the clutter RFS, $\mathcal{C}_k$. $g_{\mathcal{D}}(\mathcal{W}|\mathcal{M}_k, X_k)$ describes the likelihood of receiving a measurement from the elements of the set-valued map which incorporates detection uncertainty and measurement noises. $g_{\mathcal{C}}(\mathcal{Z}_k - \mathcal{W})$ models the spurious measurement rate of the sensor and is typically *a priori* assigned [18] [14]. Expanding the multi-target RFS Bayes recursion of [3] to include the vehicle state, the optimal Bayesian SLAM filter then jointly propagates the set of features and the vehicle location according to,

$$p_k(X_{0:k}, \mathcal{M}_k|\mathcal{Z}_{0:k}, U_{0:k-1}, X_0) = \frac{g_k(\mathcal{Z}_k|\mathcal{M}_k, X_k)p_{k|k-1}(X_{0:k}, \mathcal{M}_k|\mathcal{Z}_{0:k-1}, U_{0:k-1}, X_0)}{\int\int g_k(\mathcal{Z}_k|\mathcal{M}_k, X_k)p_{k|k-1}(X_{0:k}, \mathcal{M}_k|\mathcal{Z}_{0:k-1}, U_{0:k-1}, X_0)dX_k\mu(d\mathcal{M}_k)} \quad (5.5)$$

where,

$$p_{k|k-1}(X_{0:k}, \mathcal{M}_k|\mathcal{Z}_{0:k-1}, U_{0:k-1}, X_0) = \int f_X(X_k|X_{k-1}, U_{k-1})p_{k-1}(X_{0:k-1}, \mathcal{M}_{k-1}|\mathcal{Z}_{0:k-1}, U_{0:k-2}, X_0)dX_{k-1} \quad (5.6)$$

and μ is a reference measure on the space of features. As noted in Chapter 2, in a direct implementation of the vector-based Bayesian SLAM recursion of equation 2.15, computational complexities and multiple integrals generally lead to intractable solutions. Therefore the PHD approximation introduced in Chapter 3 is used.

5.3 The 'Brute Force' PHD SLAM Filter

The RFS formulation of the general Bayesian SLAM problem was described in the previous section. This section modifies the formulation to admit a compact PHD solution. The key to the approach is the introduction of a new RFS state, $\mathcal{Y}$, which comprises the unordered set of $\mathfrak{n}$ elements, ζ, such that at time k,

$$\mathcal{Y}_k = \{\zeta_k^1, \zeta_k^2, \cdots, \zeta_k^{\mathfrak{n}_k}\} \quad (5.7)$$

where each ζ_k comprises a map state, m_k, conditioned on a vehicle trajectory, $X_{0:k}$. $\mathfrak{n}_k$ will be defined below. Conditioning each feature state, m, on the history of vehicle poses $X_{0:k}$, introduces a conditional independence between feature measurements allowing the joint states, ζ_k to be independently propagated through the PHD SLAM framework [17]. Following the introduction of the PHD SLAM filter in Section 3.4, a "Brute force" PHD SLAM filter can then be derived if, in equations 3.23 and 3.24,

$$\Gamma_k \longrightarrow \zeta_k. \quad (5.8)$$

Each element of $\mathcal{Y}_k$, evolves in time according to the transition $f(\zeta_k|\zeta_{k-1}, U_{k-1})$ and, if the feature in ζ_k is detected by the sensor at the conditioning pose X_k, a measurement z is generated with likelihood $P_D(\zeta_k)g_k(z|\zeta_k)$. The precise reason why the conditioning of the feature state m, with each of the N hypothesised vehicle trajectories to form the state ζ_k, is permitted under the RFS framework is due to Campbell's theorem, as explained in Appendix A. Therefore, let the vehicle state be sampled by N particles, to produce $N \times \mathfrak{m}_k = \mathfrak{n}_k$ augmented states, ζ_k. Given a set of augmented features, ζ_k, joint estimates of the number of features, their locations, as well as the vehicle state, can then be obtained. Hence this SLAM implementation estimates a potentially extremely large, single PHD (intensity function) containing representations of each feature, conditioned on each vehicle trajectory. Referring to the GM PHDs of figures 3.2 and 3.3, in the "Brute force" SLAM case, multiple Gaussians are necessary to represent each single feature, since each must be conditioned on every one of the N hypothesised trajectories. In contrast to the FBRM case of Chapter 4, computationally, a copy of each hypothesised vehicle trajectory (rather than the single known vehicle pose) is necessary to condition each feature. The PHD-SLAM recursion can then be formulated in terms of the state elements ζ_k. The prediction of the state intensity $v_{k|k-1}(\zeta_k)$ is then given by

$$\begin{aligned} v_{k|k-1}(\zeta_k) &= \int f(\zeta_k|\zeta_{k-1}, U_{k-1}) v_{k-1}(\zeta_{k-1}) d\zeta_{k-1} + b_k(\zeta_k|X_k) \\ &= \int f(\zeta_k|X_{k-1}, m_k, U_{k-1}) v_{k-1}(X_{k-1}, m_k) dX_{k-1} + b_k(\zeta_k|X_k) \end{aligned} \tag{5.9}$$

where $f(\zeta_k|\zeta_{k-1}, U_{k-1})$ incorporates both the assumed vehicle and feature predicted transition functions. As before, $b_k(\zeta_k|X_k)$ models the new features entering the vehicles FoV. The corrector equation for the combined state, SLAM intensity function is

$$\begin{aligned} v_k(\zeta_k) = v_{k|k-1}(\zeta_k) \Bigg[& 1 - P_D(\zeta_k) \\ & + \sum_{z \in \mathcal{Z}_k} \frac{P_D(\zeta_k) g_k(z|\zeta_k)}{c_k(z|X_k) + \int P_D(\xi) g_k(z|\xi) v_{k|k-1}(\xi) d\xi} \Bigg] \end{aligned} \tag{5.10}$$

where again at time k,

$$\begin{aligned} b_k(\zeta_k|X_k) &= \text{intensity of the new feature RFS } \mathcal{B}_k, \\ g_k(z|\zeta_k) &= \text{likelihood of } z, \text{ given the joint state } \zeta_k, \\ P_D(\zeta_k) &= \text{probability of detection of the feature in} \\ &\quad \zeta_k, \text{ given the pose in } \zeta_k, \\ c_k(z|X_k) &= \text{intensity of the clutter RFS } \mathcal{C}_k(X_k). \end{aligned}$$

In [33], Gaussian noise assumptions were used to obtain closed form solutions for the target tracking PHD filter. Similarly for the PHD-SLAM filter, GM techniques can be applied to solve the PHD-SLAM joint intensity recursion of equation 5.11. It is also possible to use a particle-based approach [4], however, for mildly non-linear problems the Gaussian mixture approach is much more efficient. The following section thus presents a GM implementation of the PHD-SLAM filter, while a particle based version is left until Chapter 6.

5.4 Gaussian Mixture (GM) PHD-SLAM

Let the joint intensity, $v_{k-1}(\zeta_{k-1})$, at time $k-1$ be a Gaussian mixture of the form,

$$v_{k-1}(\zeta_{k-1}) = \sum_{i=1}^{N \times J_{k-1}} w_{k-1}^{(i)} \mathcal{N}\big(\zeta; \mu_{k-1}^{(i)}, P_{k-1}^{(i)}\big) \tag{5.11}$$

composed of $N \times J_{k-1}$ Gaussians, with $w_{k-1}^{(i)}$, $\mu_{k-1}^{(i)}$ and $P_{k-1}^{(i)}$ being their corresponding weights, means and covariances respectively. Note that the weight, $w_{k-1}^{(i)}$ is a weight on *both* a particular feature state, m, and a particular vehicle pose $X_{k-1}^{(n)}$, i.e. on the joint state $\zeta_{k-1}^{(i)}$.

Since the map is assumed static, the joint state transition density is

$$f_X(X_k^{(n)} | X_{k-1}^{(n)}, U_{k-1}) \delta(m_k - m_{k-1}) \tag{5.12}$$

where $X_{k-1}^{(n)}$ is one of N vehicle pose particles at time $k-1$ and $\delta(m_k - m_{k-1})$ is a Dirac delta function to mathematically incorporate the fact that the map must remain static in this case. Let the new feature intensity at time k also be a Gaussian mixture,

$$b_k = \sum_{i=1}^{N \times J_{b,k}} w_{b,k}^{(i)} \mathcal{N}\big(\zeta; \mu_{b,k}^{(i)}, P_{b,k}^{(i)}\big) \tag{5.13}$$

where $w_{b,k}^{(i)}$, $\mu_{b,k}^{(i)}$ and $P_{b,k}^{(i)}$ determine the shape of the new feature GM proposal density according to a chosen strategy. This is analogous to the proposal distribution in the particle filter [17] and provides an initial estimate of the new features entering the map (see Section 5.4.1). Again, each new feature density component is conditioned on each predicted vehicle pose particle, $X_k^{(n)}$ to form the $N \times J_{b,k}$ components of the GM new feature intensity. That is, for each hypothesised vehicle trajectory, a set of $J_{b,k}$ Gaussian components are initialised. Recalling that each Gaussian models the feature and vehicle trajectory, $J_{b,k}$ copies of a given trajectory are distributed to $J_{b,k}$ features. This is in contrast to classical Rao-Blackwellised approaches, as presented in the following chapter, where only a single copy of each trajectory is required.

However, in the PHD-SLAM filter, this conditioning allows for each joint state (feature and trajectory) to be independently estimated via the PHD framework. Therefore, the predicted intensity, $v_{k|k-1}(\zeta_k)$ is also a Gaussian mixture

$$v_{k|k-1}(\zeta_k) = \sum_{i=1}^{J_{k|k-1}} w_{k|k-1}^{(i)} \mathcal{N}\big(\zeta; \mu_{k|k-1}^{(i)}, P_{k|k-1}^{(i)}\big) \tag{5.14}$$

where, $J_{k|k-1} = N(J_{b,k} + J_{k-1})$ and,

$$\left.\begin{array}{l} w_{k|k-1}^{(i)} = w_{k-1}^{(i)} \\ \mu_{k|k-1}^{(i)} = \mu_{k|k-1}^{(i)} \\ P_{k|k-1}^{(i)} = P_{k-1}^{(i)} \end{array}\right\} \text{for i} \in \{1, \ldots, N{\times}J_{k-1}\} \text{ (previously observed features)}$$

$$\left.\begin{array}{l} w_{k|k-1}^{(i)} = w_{b,k}^{(i)} \\ \mu_{k|k-1}^{(i)} = \mu_{b,k}^{(i)} \\ P_{k|k-1}^{(i)} = P_{b,k}^{(i)} \end{array}\right\} \text{for i} \in \{N{\times}J_{k-1}+1, \ldots, N{\times}J_{b,k}\} \text{ (newly observed features).}$$

Assuming a Gaussian measurement likelihood, $g(z|\zeta_k)$, analysis of equation 5.11 shows that the joint posterior intensity, $v_k(\zeta_k)$, is consequently also a Gaussian mixture,

$$v_k(\zeta_k) = v_{k|k-1}(\zeta_k)\left[1 - P_D(\zeta_k) + \sum_{z \in \mathcal{Z}_k} \sum_{i=1}^{J_{k|k-1}} v_{G,k}^{(i)}(z|\zeta_k)\right] \tag{5.15}$$

where,

$$v_{G,k}^{(i)}(z|\zeta_k) = w_k^{(i)} \mathcal{N}(\zeta; \mu_{k|k}^{(i)}, P_{k|k}^{(i)}) \tag{5.16}$$

$$w_k^{(i)} = \frac{P_D(\zeta_k) w_{k|k-1}^{(i)} q^{(i)}(z, \zeta_k)}{c_k(z) + \sum_{j=1}^{J_{k|k-1}} P_D(\zeta_k) w_{k|k-1}^{(j)} q^{(j)}(z, \zeta_k)} \tag{5.17}$$

with, $q^{(i)}(z, \zeta_k) = \mathcal{N}\big(z;\ H_k \mu_{k|k-1}^{(i)},\ S_k^{(i)}\big)$. The component distributions are represented by their first and second moments obtained from the standard EKF update equations,

$$S_k^{(i)} = R_k + \nabla H_k P_{k|k-1}^{(i)} \nabla H_k^T \tag{5.18}$$

$$K_k^{(i)} = P_{k|k-1}^{(i)} \nabla H_k^T [S_k^{(i)}]^{-1} \tag{5.19}$$

$$\mu_{k|k}^{(i)} = \mu_{k|k-1}^{(i)} + K_k^{(i)}(z - H_k \mu_{k|k-1}^{(i)}) \tag{5.20}$$

$$P_{k|k}^{(i)} = [I - K_k^{(i)} \nabla H_k] P_{k|k-1}^{(i)} \tag{5.21}$$

with ∇H_k being the Jacobian of the measurement equation with respect to the features estimated location. As stated previously, the clutter RFS, $\mathcal{C}_k$, is assumed Poisson distributed [18], [14] in number and uniformly spaced over the sensor surveillance region. Therefore the clutter intensity is,

$$c_k(z) = \lambda_c V \mathcal{U}(z) \tag{5.22}$$

where λ_c is the average number of clutter returns, V is the volume of the sensor's surveillance region and $\mathcal{U}(z)$ denotes a uniform distribution over range and bearing. As stated in Section 5.1, the Gaussian number growth of this formulation becomes very large and hence Gaussian pruning and merging methods are used as in [33].

5.4.1 The SLAM New Feature Proposal Strategy

The new feature proposal density, equation 5.13, 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_k, may be uniformly distributed in a non-informative manner about the space of features (analogous to the prior map used in occupancy grid algorithms). However, in this work the feature birth proposal at time k is chosen to be the set of measurements at time k–1, $\mathcal{Z}_{k-1}$. The sum $\sum_{i=1}^{N \times J_{b,k}} w_{b,k}^{(i)}$ then gives an estimate of the expected number of new features to appear at time k. The components of the Gaussian mixture used to form b_k are determined in exactly the same way as for the FBRM case, described in Section 4.5.2.1.

5.5 Brute Force SLAM Pseudo-code

This section details the pseudo-code of the proposed PHD-SLAM algorithm. Table 5.1 presents the birth proposal algorithm which accommodates new features entering the map as well as aiding particle diversity since each new potential feature is seeded with a corresponding potential vehicle pose. Table 5.2 outlines in the prediction module while Tables 5.3 and 5.3 describe the update module. Vehicle pose and map estimation are achieved via the process detailed in Table 5.4.

Table 5.1 PHD-SLAM-Birth

Algorithm PHD-SLAM-Birth$\big(\mathcal{Z}_{k-1}, v_{k-1}(\zeta|\mathcal{Z}_{k-1})\big)$
// Generate eqn.(5.13). Note: Any arbitrary strategy is valid.
// For each measurement
1. for i = 1 to $\mathfrak{z}_{k\text{-}1}$ do
// For each particle
2. for j = 1 to J_{k-1} do
// initialise the feature mean
3. $\nu_{b,k} = h^{-1}(z_{k-1}^{(i)}, X_{k-1}^{(j)})$
// initialise the concatenated state mean
4. $\mu_{b,k}^{((i-1)\times J_{k-1}+j)} = [X_{k-1}^{(j)}\ \nu_{b,k}]$
// initialise the covariance
5. $P_{b,k}^{((i-1)\times J_{k-1}+j)} = h^{'}(\nu_{b,k}, X_{k-1}^{(j)})$
// initialise the weight
6. $\omega_{b,k}^{((i-1)\times J_{k-1}+j)} = \alpha$
7. end for
8. end for
// Set the number of birth components
9. $J_{b,k} = \mathfrak{z}_{k\text{-}1} \times J_{k-1}$
// Construct birth PHD
10. $b_k(\zeta|\mathcal{Z}_{k-1}) = \{\mu_{b,k}^{(i)}, P_{b,k}^{(i)}, \omega_{b,k}^{(i)}\}_{i=1}^{J_{b,k}}$
11. return $\big(\ b_k(\zeta|\mathcal{Z}_{k-1})\ \big)$

5.6 Algorithm Performance

This section analyses the performance of the proposed GM-PHD SLAM filter in a simulated environment, and compares it to a FastSLAM implementation using maximum likelihood data association decisions and Log-Odds feature management [17]. The vehicle is assumed to be travelling at $3ms^{-1}$ while subject to velocity and steering input noises of $1ms^{-1}$ and 5^o respectively. Only 10 particle samples are used for both filters and both filters receive the same noisy input samples and sensor measurements. Two simulated comparisons are performed in an 'Easy' and 'Difficult scenario'. For the 'easy' scenario, the clutter parameter, $\lambda_c = 0$ m^{-2}, feature detection probability is 0.95, and the measurement noises are $0.25m$ in range and 0.5^o in bearing. For the 'difficult' scenario, $\lambda_c = 10$ m^{-2} (i.e. 10 false alarms occur for every square metre of area within the sensors FoV), feature detection probability is again 0.95 and the measurements noises are set at $12.5m$ in range and 25^o in bearing. The effect of the artificially large measurement noises are to give the appearance of closely spaced features, hampering data association decisions and feature map building.

Table 5.2 PHD-SLAM-Predict

Algorithm PHD-SLAM-Predict$\left(v_{k-1}(\zeta|\mathcal{Z}_{k-1}), U_{k-1}\right)$
// Generate eqn.(5.14)
// Generate map birth components
1. GMM-PHD-FBRM-Birth$(\mathcal{Z}_{k-1}, v_{k-1}(\zeta|\mathcal{Z}_{k-1}))$
// append PHD with birth components
2. for i = 1 to $J_{b,k}$ do
3. $\quad \omega_{k-1}^{(J_{k-1}+i)} = \omega_{b,k}^{(i)}$
4. $\quad \mu_{k-1}^{(J_{k-1}+i)} = \mu_{b,k}^{(i)}$
5. $\quad P_{k-1}^{(J_{k-1}+i)} = P_{b,k}^{(i)}$
6. end for
// increment component counter
7. $J_{k|k-1} = J_{k-1} + J_{b,k}$
8. for i = 1 to $J_{k|k-1}$
// sample a pose from the vehicle model
9. $\quad X_{k|k-1}^{(i)} \sim f_X(X_k^{(i)}|X_{k-1}^{(i)}, U_{k-1})$
// static map assumption
10. $\quad \nu_{k|k-1}^{(i)} = \nu_{k-1}^{(i)}$
11. $\quad \mu_{k|k-1}^{(i)} = [X_{k|k-1}^{(i)} \ \nu_{k|k-1}^{(i)}]$
12. $\quad P_{k|k-1}^{(i)} = P_{k-1}^{(i)}$
13. $\quad \omega_{k|k-1}^{(i)} = \omega_{k-1}^{(i)}$
14. end for
// The predicted SLAM PHD
15. $v_{k|k-1}(\zeta|\mathcal{Z}_{k-1}) = \{\mu_{k|k-1}^{(i)}, P_{k|k-1}^{(i)}, \omega_{k|k-1}^{(i)}\}_{i=1}^{J_{k|k-1}}$
16. return$\left(v_{k|k-1}(\zeta|\mathcal{Z}_{k-1})\right)$

Figure 5.1 shows the estimated vehicle trajectory and corresponding feature map from both filters, in the case of the 'Easy scenario'.

Both results compare well with ground truth (green). This result verifies the accuracy of the proposed PHD-SLAM filter, in its ability to jointly estimate the vehicle trajectory, the number of features, and their corresponding location, without the need for external data association and feature map management methods, as are required by FastSLAM (and other vector-based solutions).

The missed feature declaration highlights an issue of the proposed method with respect to $P_D(\zeta_k)$. In the presented implementation, this is simply a binary function which has an assumed value of 0.95 if the feature is predicted to be within the sensor FoV, and 0 if it is not. Vehicle and feature estimation uncertainty may result in a feature erroneously being hypothesised of being within the FoV, or vice-versa. If the proposed filter then receives a

Table 5.3 PHD-SLAM-Update

Algorithm PHD-SLAM-Update$\left(\mathcal{Z}_k, v_{k|k-1}(\zeta|\mathcal{Z}_{k-1})\right)$

// Initialise number of Gaussian components

1. L = 0

// Missed detections and Update Terms

2. for i = 1 to $J_{k|k-1}$ do

// Increment component counter

3. L = L+1

// Updated feature equals predicted feature

4. $\nu_k = \nu_{k|k-1}^{(i)}, P_k^{(L)} = P_{k|k-1}^{(i)}$

// Updated joint mean

5. $\mu_k^{(L)} = [\nu_k \; X_{k|k-1}^{(i)}]$

// weight decreased

6. $\omega_k^{(L)} = (1 - P_D)\omega_{k|k-1}^{(i)}$

// measurement prediction

7. $z_{k|k-1}^{(i)} = h(\nu_{k|k-1}^{(i)}, X_{k|k-1}^{(i)})$

3// Calculate Jacobian

8. $H = h^{'}(\nu_{k|k-1}^{(i)}, X_{k|k-1}^{(i)})$

// Innovation Covariance of (5.18)

9. $S_k^{(i)} = HP_{k|k-1}^{(i)}[H]^T + R$

// Kalman Gain of (5.19)

10. $K_k^{(i)} = P_{k|k-1}^{(i)}[H]^T[S_k^{(i)}]^{-1}$

// Updated Covariance of (5.21)

11. $P_{U,k}^{(i)} = [I - K_k^{(i)}H]P_{k|k-1}^{(i)}$

12. end for

// For each measurement

13. for i = 1 to $\mathfrak{z}_k$ do

// For each component

14. for j = 1 to $J_{k|k-1}$ do

// Updated map component mean

15. $\nu_k = \nu_{k|k-1}^{(j)} + K_k^{(j)}(z_k^{(i)} - z_{k|k-1}^{(j)})$

// Updated joint mean

16. $\mu_k^{(L+j)} = [\nu_k \; X_{k|k-1}^{(i)}]$

// Updated GMM component covariance

17. $P_k^{(L+j)} = P_{U,k}^{(j)}$

// Numerator of (5.16)

18. $\tau^{(j)} = P_D\omega_{k|k-1}^{(j)}|2\pi S_k^{(j)}|^{-0.5}$
$\times \exp\left((z_k^{(i)} - z_{k|k-1}^{(j)})[S_k^{(j)}]^{-1}(z_k^{(i)} - z_{k|k-1}^{(j)})\right)$

19. end for

Table 5.3 (*Continued*)

// For each map PHD component
20. for j = 1 to $J_{k|k-1}$ do
// denominator of (5.16)
21. $\omega_k^{(L+j)} = \tau^{(j)}/(c(z) + \sum_{l=1}^{J_{k|k-1}} \tau^{(l)})$
22. end for
23. L = L+$J_{k|k-1}$
24. end for
// Number of components in updated GMM
25. J_k = L
// The updated map PHD
26. $v_k(\zeta|\mathcal{Z}_k) = \{\mu_k^{(i)}, P_k^{(i)}, \omega_k^{(i)}\}_{i=1}^{J_k}$
27. return$\big(v_k(\zeta|\mathcal{Z}_k)\big)$

Table 5.4 PHD-SLAM-Estimate

Algorithm PHD-SLAM-Estimate$\big(v_k(\zeta|\mathcal{Z}_k), T_{\text{feature}}\big)$
// Initialise the map estimate
1. $\hat{\mathcal{M}}_k = \emptyset$
2. $\Omega_k = 0$
3. for i = 1 to J_k do
4. $\Omega_k = \Omega_k + \omega_k^{(i)}$
5. if $\omega_k^{(i)} > T_{\text{feature}}$
// concatenate estimate
6. $\hat{\mathcal{M}}_k = [\hat{\mathcal{M}}_k\ \nu_k^{(i)}]$
7. end if
8. end for
// expected pose
9. $\hat{X}_k = \dfrac{1}{\Omega_k} \sum_{i=1}^{J_k} \omega_k^{(i)} X_k^{(i)}$
10. return $\big(\hat{X}_k, \hat{\mathcal{M}}_k\big)$

measurement contrary to the prediction, the resulting feature weight may be detrimentally reduced, and a missed feature declaration may occur. The uncertainty in the estimated sensor FoV is not considered in this implementation.

The raw measurements as well as the final posterior joint estimate of both filters for the 'hard' scenario are presented respectively in figures 5.2 and 5.3. As can be seen, the FastSLAM filter shows divergence of the estimated vehicle trajectory from its true value, as well as many falsely estimated map

features. The proposed filter however, displays dramatically reduced feature-based mapping error in the face of large data association uncertainty and large quantities of spurious measurements, reporting only a single false feature and a single missed feature over the entire run. This is expected, as the clutter rate is integrated directly into the filter recursion in an optimal manner and feature management is performed jointly with feature and vehicle location estimation. The key to this vast difference in performance in cluttered environments, is evident in figure 5.4. The figure shows the estimated number of features in the map over time, for both the discussed filters, as well as the number of false measurements at each time instant. The proposed filter accurately tracks the true number of features over time, whereas the FastSLAM filter deviates drastically in the face of the challenging spurious measurements and data association ambiguities. Estimating the true number of state dimensions influences the accuracy of the overall SLAM filter. The estimated vehicle trajectory also displays less error than that of the FastSLAM approach. Similarly to FastSLAM, increased trajectory estimation accuracy may be possible by increasing the number of pose samples. Figure 5.5 compares the estimated vehicle heading over the course of the test, highlighting the increased accuracy of the proposed filter.

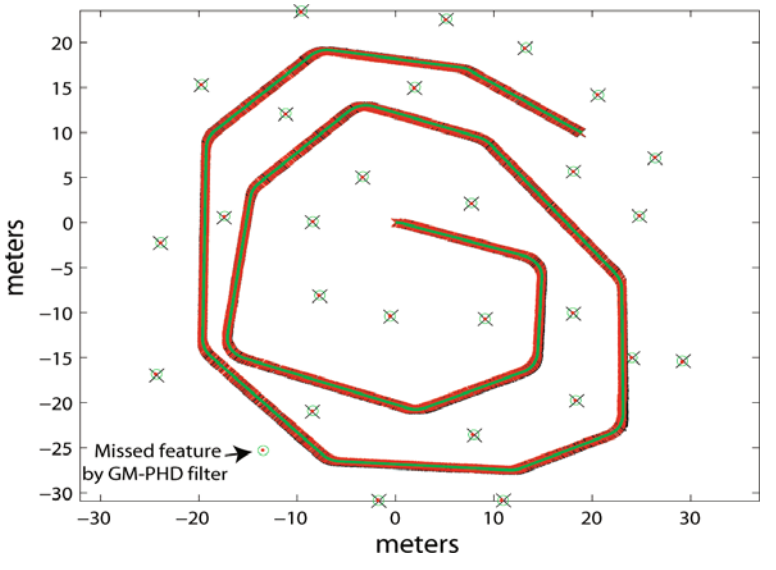

Fig. 5.1 Comparative results for the proposed GM-PHD SLAM filter (black) and that of FastSLAM (red), compared to ground truth (green).

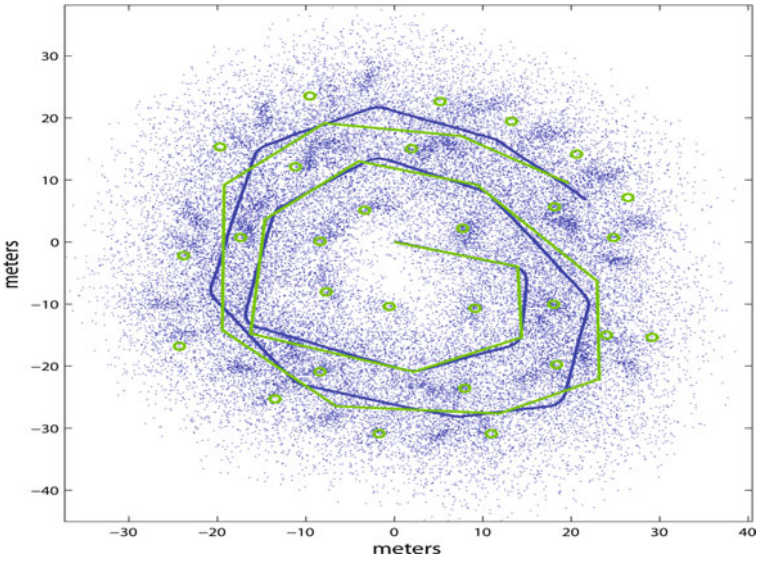

Fig. 5.2 The predicted vehicle trajectory (blue) along with the raw sensor measurements for the 'hard' scenario, at a clutter density of $0.03m^{-2}$. Also superimposed are the ground truth trajectory and feature map (black crosses).

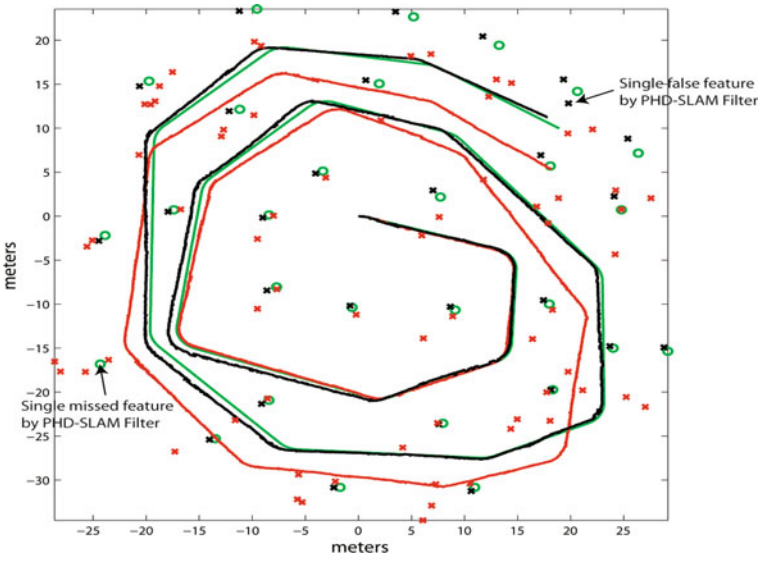

Fig. 5.3 The estimated trajectories of the GM-PHD SLAM filter (black) and that of FastSLAM (red). Estimated feature locations (crosses) are also shown with the true features (green circles)

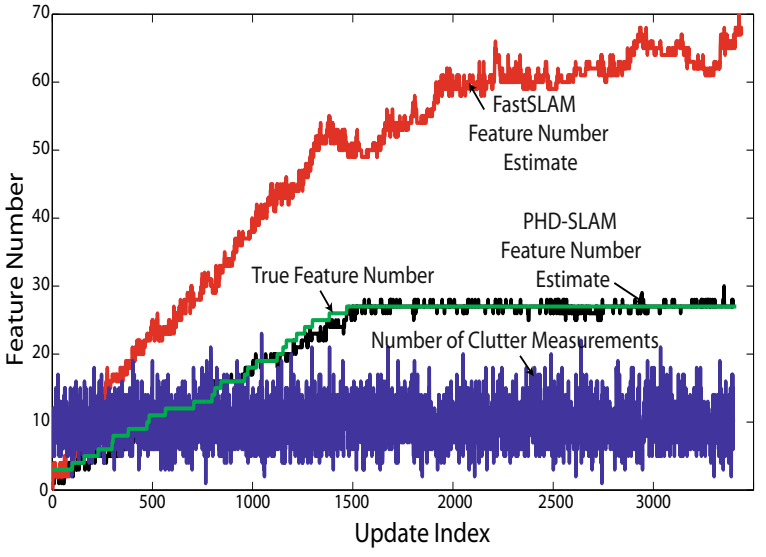

Fig. 5.4 The estimated feature number for the proposed (black) and FastSLAM (red) filters. The green curve shows the ideal estimate, which grows as more features enter the FoV of the sensor, during vehicle motion. The blue curve shows the randomly generated number of clutter measurements.

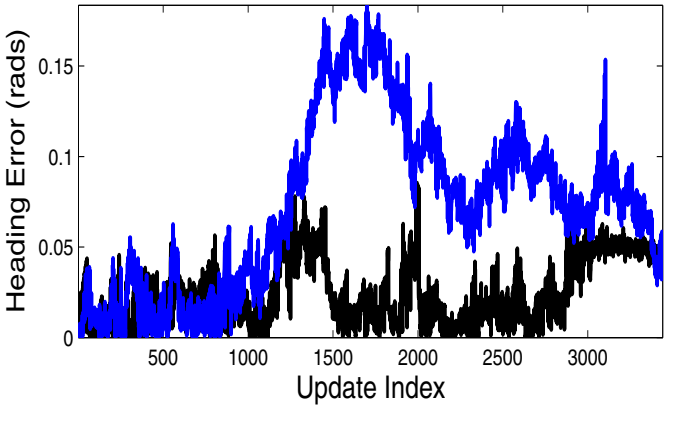

Fig. 5.5 The error in vehicle heading estimate for the proposed (black) and FastSLAM (blue) filters.

5.6.1 A Note on Computational Complexity

As is evident from the update of equation 5.15, the proposed algorithm scales linearly with $\mathcal{O}(NJ_{k|k-1}\mathfrak{Z}k)$, which equals that of a naive FastSLAM implementation. Future work will address reducing this to a Log order complexity of the number predicted map states $J_{k|k-1}$. The presented results illustrate the effectiveness of the new finite-set based SLAM framework, and the proposed GM-PHD implementation, when compared to vector-based solutions which fail to jointly consider the entire system uncertainty.

5.7 Summary

This chapter outlined a 'Brute force' formulation of the Bayesian SLAM problem, using random set theory. The set theoretic approach allows for detection uncertainty, spurious measurements as well as data association uncertainty to be incorporated directly into the filter recursion. This is in contrast to vector-based SLAM which requires additional algorithms and pre/post processing to solve the data association problem prior to filter update, and to extract estimates of the number of features present in the map. These are necessary as such sources of uncertainty are not considered in the classical vector-based measurement model and subsequent filter recursion. Previous Bayesian SLAM solutions also lack a concept of Bayesian optimality as the variable dimensionality problem is not jointly considered.

Propagating the first order statistic of the random set (the probability hypothesis density) is a common method of reducing the computational requirements of implementing the set-valued Bayesian recursion. By augmenting the feature state with a history of vehicle poses, conditional independencies between the features and the vehicle state are introduced. The joint vehicle feature RFS was shown to maintain the necessary Poisson assumptions for application of the tracking based PHD recursion for the PHD-SLAM problem. A Gaussian mixture implementation of the PHD-SLAM filter was outlined assuming a Gaussian system with non-linear measurement and process models. The proposed finite-set filter was compared to a FastSLAM implementation with explicit (per particle) data association decisions and feature management methods. Results show the proposed filter performing similarly to FastSLAM in an 'easy' scenario, and considerably outperforming it in a 'hard' scenario.

5.8 Bibliographical Remarks

As explained in Chapter 2, it is critical for data association algorithms not to use falsely declared targets in their hypothesis decision making process. In [17], This is a well known fact in the SLAM community, which has been addressed by various researchers. For example, M. Montemerlo *et. al.* use methods from occupancy-grid based robotic mapping [12] to estimate the number of landmarks in the map state. A probabilistic evidence for landmark existence is derived from the measurement and propagated through a Log-Odds discrete Bayes recursion. This discrete recursion is independent of the main RM state estimation filter and is effectively a post-processing of the map state estimate at each time step. The output of the maximum likelihood data association decision is incorporated into the existence update. If the landmark is not associated (and therefore assumed undetected) the existence posterior is reduced, thus inherently assuming that the probability of detection of all the landmarks is unity. Based on a predefined-defined log-odds threshold, landmarks are then either added or removed from the map state. The number of landmarks in the map state at any given time then gives an estimate of the number of landmarks in the map, as *every* component of the map state is assumed to be a valid landmark. However, since in reality, landmarks have non-unity detection probabilities, a missed-detection does not always indicate the non-presence of the landmark.

Another method of landmark management was introduced by D. Makarsov in [14] and used in [1] which outlined the so-called 'Geometric feature track quality' measure of landmark existence. This measure is inversely proportional to the innovation between a predicted landmark and the measurements and is therefore only updated when measurements are made (and associated). It does not consider the frequent sensor errors in terms of detection uncertainty and spurious measurements. Other techniques [18] simply use the number of successive associations over a fixed set of measurement frames which requires both low clutter rates and correct association hypotheses. Again, such methods are effectively post/pre-processing techniques which are independent of the state estimating filter recursion.

Sequential Monte Carlo (SMC) solutions to Bayesian SLAM also gained popularity [28] through the use of Rao-Blackwellised particle filters. FastSLAM [17] displayed impressive results by sampling over the vehicle trajectory and applying independent Kalman filters to estimate the location of the hypothesised map features. By conditioning the map estimates on the history of vehicle poses, a conditional measurement independence is invoked which allows the correlations introduced in [8] to be discarded.

A Gaussian mixture solution to the Bayesian SLAM problem was also described in [53] which approximated both the transition and measurement densities as Gaussian mixtures and propagated the joint state through a Bayes recursion.

Extensions to of the PHD filter to SLAM were first given in [54] [55], upon which this chapter is based.