Chapter 2
Why Random Finite Sets?

2.1 Introduction

We begin the justification for the use of RFSs by re-evaluating the basic issues of feature representation, and considering the fundamental mathematical relationship between environmental feature representations, and robot motion. We further the justification for the use of RFSs in FBRM and SLAM by considering an issue of fundamental mathematical importance in any estimation problem - estimation error.

2.2 Environmental Representation: Fundamentals

2.2.1 FBRM and SLAM New Concepts

Consider a simplistic, hypothetical scenario in which a mobile robot traverses three different trajectories, amongst static objects, as shown in Figure 2.1. If the trajectory taken by the robot were X^1 (red), then it would seem logical that an on board sensor, with a limited range capability, may sense feature m_1 followed by m_2 followed by m_3 etc. Hence after completing trajectory X^1, if a vector M is used to represent the map, then the estimated map could be

$$\widehat{M} = [m_1 \;\; m_2 \;\; m_3 \;\; m_4 \;\; m_5 \;\; m_6 \;\; m_7]^T \tag{2.1}$$

Alternatively, had the robot pursued trajectory X^2 (blue) instead, the order in which the features would be sensed would likely be very different, and the resulting estimated map could be

$$\widehat{M} = [m_4 \;\; m_2 \;\; m_3 \;\; m_1 \;\; m_5 \;\; m_7 \;\; m_6]^T \tag{2.2}$$

J. Mullane et al.: Random Finite Sets for Robot Mapping & SLAM, STAR 72, pp. 11–25.
springerlink.com

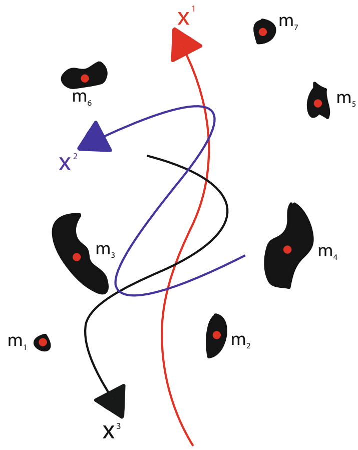

Fig. 2.1 A hypothetical scenario in which a mobile robot executes three different trajectories X^1, X^2, X^3, amidst static objects (features) m_1 to m_7.

and had the robot pursued trajectory X^3 (black), the following estimated map vector could result

$$\widehat{M} = [m_6 \;\; m_7 \;\; m_5 \;\; m_4 \;\; m_3 \;\; m_2 \;\; m_1]^T. \tag{2.3}$$

Since the order of the elements within a vector is of importance (a change in the order yields a different vector), three different map vectors result. However, since the map features themselves were assumed static, it seems odd that this estimated vector is actually dependent on the vehicle's trajectory. In a strict mathematical sense, the order of the features within the map estimate should not be significant, as any permutation of the vectors results in a valid representation of the map. By definition, the representation which captures all permutations of the elements within the vector, and therefore the features in the map, is a *finite set* $\mathcal{M}$, whose estimate $\widehat{\mathcal{M}}$ would be as shown in representation 2.4.

$$\underbrace{\begin{array}{c} \widehat{M} = [m_1 \; m_2 \; m_3 \; m_4 \; m_5 \; m_6 \; m_7]^T \\ \widehat{M} = [m_4 \; m_2 \; m_3 \; m_1 \; m_5 \; m_7 \; m_6]^T \\ \vdots \\ \widehat{M} = [m_6 \; m_7 \; m_5 \; m_4 \; m_3 \; m_2 \; m_1]^T \end{array}}$$

$$\widehat{\mathcal{M}} = \{m_1 \; m_2 \; m_3 \; m_4 \; m_5 \; m_6 \; m_7\} \tag{2.4}$$

Note that for notational purposes, we denote vector representations in italics (e.g. for the map M) and set representations in mathcal format (e.g. for the map $\mathcal{M}$).

2.2.2 Eliminating the Data Association Problem

For most sensors/sensor models considered in SLAM, the order in which sensor readings are recorded at each sampling instance, simply depends on the direction in which the vehicle/sensor is oriented, and has no significance on the state of the map, which typically evolves in a globally defined coordinate system, independent of the vehicle's pose. This is illustrated in Figure 2.2 in which a measurement to state assignment problem is evident. It can be seen

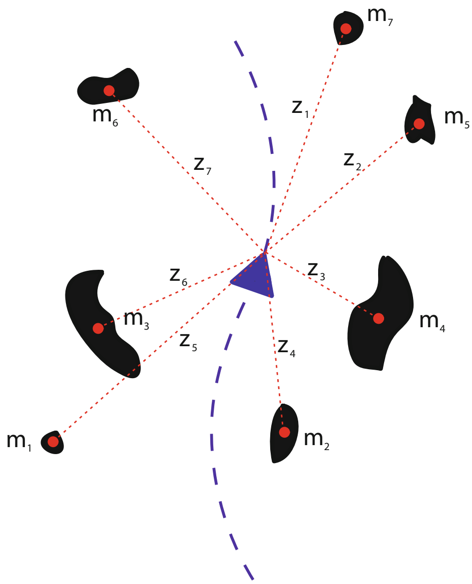

Fig. 2.2 The order in which observations (features) z_1 to z_7 are detected/extracted from the sensor data is usually different from the order of the currently estimated features m_1 to m_7 in the state vector.

in the figure, that even for an ideal sensor, which is always able to detect all of the features, all of the time, under the vector based representation, a re-ordering of the observed feature vector Z is necessary. This is because, in general, observed feature z_1 will not correspond to the current estimate m_1 etc. (Figure 2.2) and the correct feature associations must be determined - i.e.:

$$\begin{aligned} Z &= [z_1\ \ z_2\ \ z_3\ \ z_4\ \ z_5\ \ z_6\ \ z_7]^T \\ &\qquad\quad \swarrow\ \downarrow\ \searrow\ ??? \qquad\qquad \text{Feature Association} \\ \widehat{M} &= [m_1\ \ m_2\ \ m_3\ \ m_4\ \ m_5\ \ m_6\ \ m_7]^T \end{aligned} \tag{2.5}$$

It can be seen in Figure 1.1, that this data association step is necessary, before any vector based, Bayesian update can take place. Hence, current vector based formulations of the FBRM and SLAM problems require this feature association problem to be solved *prior* to the Bayesian (EKF, UKF etc) update. This is because, feature estimates and measurements are rigidly ordered in their respective finite vector valued, map states.

The proposed RFS approach on the other hand, represents both features and measurements as finite valued sets $\mathcal{M}$ and $\mathcal{Z}$ respectively, which assume no distinct ordering of the features, as shown in representations 2.4 and 2.6.

$$\underbrace{\begin{array}{c} Z = [z_1\ z_2\ z_3\ z_4\ z_5\ z_6\ z_7]^T \\ Z = [z_4\ z_2\ z_3\ z_1\ z_5\ z_7\ z_6]^T \\ \vdots \\ Z = [z_6\ z_7\ z_5\ z_4\ z_3\ z_2\ z_1]^T \end{array}} \\ \mathcal{Z} = \{z_1\ z_2\ z_3\ z_4\ z_5\ z_6\ z_7\} \tag{2.6}$$

Since the finite set representations 2.4 and 2.6 naturally encapsulate all possible permutations of the feature map and measurement, feature association assignment does not have to be addressed. This will be demonstrated throughout the book.

2.2.3 Eliminating the Map Management Problem

For the more realistic case of non-ideal sensors/feature extraction algorithms, the number of measurements, $\mathfrak{z}_k$, at any given time is not fixed due to detection uncertainty, spurious measurements and unknown number of true features. As the robot moves through its environment, more and more features are detected, as they enter the field of view (FoV) of its sensors. Hence the map size grows monotonically as shown in Figure 2.3. In the figure it can be seen that 5 features have been detected, although there are seven features in the environment shown. Objects m_5, m_6 and m_7 lie out of range of the sensor, in the robot's current position. Due to sensor and/or feature detection

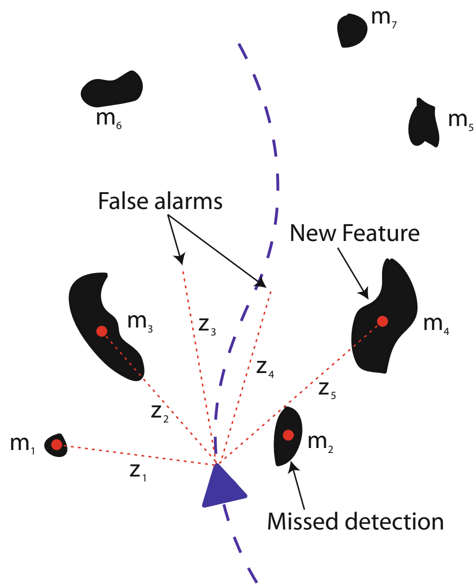

Fig. 2.3 Feature detection with a more realistic sensor. As the robot moves, new features will enter the FoV of the sensor(s). In general, some features may be undetected (missed detections), and some falsely detected features (false alarms) may be declared, due to less than ideal sensor/feature detection algorithm performance.

algorithm imperfections, two false alarms z_3 and z_4 have occurred. These can originate from clutter, sensor noise or incorrect feature detection algorithm performance. Notice also, that although object m_2 lies within the FoV of the sensor, it has not been detected, and constitutes a *missed detection*.

Suppose that features m_1, m_2 and m_3 already exist at time $k-1$ in a vector based map representation, and that feature m_4 now falls into the robot's sensor(s) FoV. Feature m_4 is to now be initialised and included in the state estimate at time k. From a strict mathematical point of view, it is unclear, using a vector based framework, how this should be carried out, as shown in equation 2.7.

$$\begin{aligned}\widehat{M}_{k-1} &= [m_1 \;\; m_2 \;\; m_3]^T \\ \widehat{M}_k &\stackrel{?}{=} [m_1 \;\; m_2 \;\; m_3]^T \text{``}+\text{''} [m_4]\end{aligned} \tag{2.7}$$

where $\stackrel{?}{=}$ is used here to mean "how do we assign $\widehat{M}_k$?" M_{k-1} represents the vector based state at time $k-1$ and M_k the corresponding state at time k. A clear mathematical operation for combining vectors of different dimensions is not defined. To date, many FBRM and SLAM techniques use vector augmentation methods. However if the map is defined as a set, then a set based map transition function can be mathematically defined as

$$\begin{aligned} \widehat{\mathcal{M}}_{k-1} &= \{m_1 \;\; m_2 \;\; m_3\} \\ \widehat{\mathcal{M}}_k &= \{m_1 \;\; m_2 \;\; m_3\} \cup \{m_4\} \end{aligned} \tag{2.8}$$

Another fundamental component of any FBRM or SLAM framework is a necessity to relate observations to the estimated state. As can be seen in equations 2.9, the relationship between observations and the estimated state is not clearly defined under a vector based framework.

$$\begin{aligned} Z_k &= h([m_1 \;\; m_2 \;\; m_3 \;\; m_4], X_k) + \text{noise} \\ \text{i.}e.: \;\; [z_1 \;\; z_2 \;\; z_3 \;\; z_4 \;\; z_5]^T &\stackrel{?}{=} h([m_1 \;\; m_2 \;\; m_3 \;\; m_4], X_k) + \text{noise} \end{aligned} \tag{2.9}$$

where Z_k represents the observation vector at time k, and here the observation example of Figure 2.3 is used. X_k represents the vehicles pose at time k and $h()$ is the (typically non-linear) function relating map feature locations and the vehicle pose, to the observations. Equation 2.9 highlights the problem of relating, for example, five observations to just four feature locations, and the robot's pose. The extra observed features are clearly the result of clutter in this case, and one feature has been missed (undetected). How such "clutter" observations, and missed detections can be incorporated into the vector based measurement equation is undefined. Clearly, assuming that single features give rise to at most single observations, there is an inconsistency, due to the mismatch in the map state and observation vectors' dimensions. In the case of vectors, map management heuristics are typically used to first remove one of the observed features so that the equation can be "forced to work".

If set based measurements and state map estimates are used, a strict mathematical relationship is possible as shown in equation 2.10

$$\mathcal{Z}_k = \bigcup_{m \in \mathcal{M}_k} \mathcal{D}_k(m, X_k) \cup \mathcal{C}_k(X_k) \tag{2.10}$$

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

As a result of the data association methods and map management rules which are necessary when the vector based representation is used for FBRM and SLAM, it is clear that the uncertainty in the *number of features* is not modelled. Typically, post-processing (outside of the Bayesian estimation component) filters are required to estimate the feature number [14]. If an RFS approach is used however, the uncertainty in both the feature state values (typically locations) *and* number can be modelled in a consistent, joint mathematical manner.

2.3 FBRM and SLAM Error Quantification

Fundamental to any state estimation problem is the concept of estimation error. While the concept of error quantification is well established in the occupancy grid literature [16, 19], in the feature-based literature the topic is less rigorously addressed. Current error evaluations of feature-based frameworks typically analyse the consistency of a subset of the feature location estimates [1], [15]. While this may illustrate the consistent spatial state estimates of the selected features, it gives no indication as to the quality of the estimate of the joint multi-feature map state. Qualitative analysis, in which estimated map features and robot location are superimposed on satellite images [17], is also not mathematically consistent and overlooks the underlying estimation problems of the feature map, namely that of the error in the estimated *number and location* of features in the map.

Whilst the majority of autonomous navigation work focuses on the localisation accuracy that can be achieved, the accuracy of the resulting map estimate is of equal importance. A precise measurement of the robots surroundings is essential to any task or behaviour the robot may be required to perform. A broad range of exteroceptive sensors are generally deployed on autonomous vehicles to acquire information about the surrounding area. Many sensors, such as laser range finders, sonars and some types of radar, measure the relative range and bearing from the vehicle to environmental landmarks and are used to update the time predicted map state. Such measurements are however subject to uncertainty such as range and bearing measurement noise, detection uncertainty, spurious measurements and data association uncertainty [20], [16].

This section demonstrates that, in the context of jointly evaluating the error in the estimated number of features and their locations, and their true values, the collection of features, should be represented by a finite set. The rationale behind this representation traces back to a fundamental consideration in estimation theory - estimation error. Without a meaningful notion of estimation error, estimation has very little meaning. Despite the fact that mapping error is equally as important as localisation error, its formal treatment has been largely neglected.

To illustrate this point, recall that in existing SLAM formulations the map is constructed by stacking features into a vector, and consider the simplistic scenarios depicted in figure 2.4. Figure 2.4a depicts a scenario in which there are two true features at coordinates $(0, 0)$ and $(1, 1)$. The true map, M, is

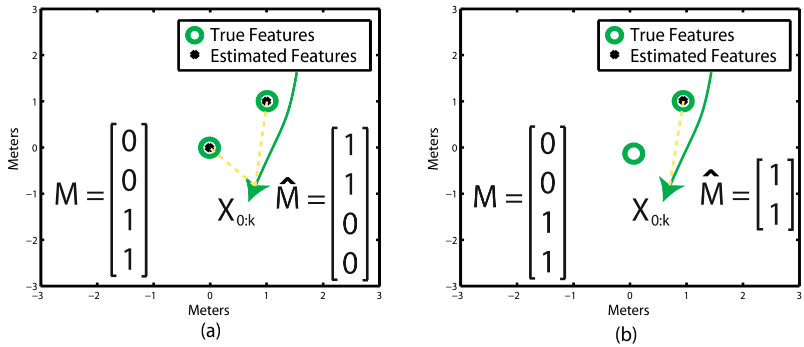

Fig. 2.4 A hypothetical scenario showing a fundamental inconsistency with vector representations of feature maps. If M is the true map, how should the error be assigned when the number of features in the map estimate, $\widehat{\mathrm{M}}$, is incorrect?

then represented by the vector $M = [0\ 0\ 1\ 1]^T$. If features are stacked into a vector in order of appearance then, given a vehicle trajectory $X_{0:k}$ (e.g. as shown in the figure) and perfect measurements, the estimated map may be given by the vector $\widehat{M} = [1\ 1\ 0\ 0]^T$. Despite a seemingly perfect estimate of the map, the Euclidean error of the estimated map, $||M - \widehat{M}||$, is 2. This inconsistency arises because the ordering of the features in the representation of the map should not be relevant. A vector representation however, imposes a mathematically strict arrangement of features in the estimated map based on the order in which they were detected [21], [1]. Intuitively, the elements of $\widehat{M}$ could be permuted to obtain a zero error, however, the representation of all possible permutations of the elements of a vector is, by definition, a set. Hence, such a permuting operation implies that the resulting error distance is no longer a distance for vectors but a distance for sets, and thus this book derives a set based approach to SLAM. Another problem is depicted in figure 2.4b, in which there are again two features at (0,0) and (1,1), but due to a missed detection (for instance), the estimated map comprises only one feature at (1,1). In such a situation, it is difficult to define a mathematically consistent error metric (Euclidean error, Mean Squared Error) between the vectors M and $\widehat{M}$ since they contain a different number of elements. It is evident from these examples that stacking individual features into a single vector does not lead to a natural notion of mapping error, in general.

A *finite set* representation of the map, $\mathcal{M}_k = \{m_k^1, \ldots, m_k^{\mathfrak{m}_k}\}$, where $m^1, \ldots, m^{\mathfrak{m}_k}$ are the $\mathfrak{m}_k$ features present at time k, admits a mathematically consistent notion of estimation error since the 'distance', or error between sets, is a well understood concept. Examples of such 'distance' metrics include the Hausdorff, Optimal Mass Transfer (OMAT) [22] and Optimal Sub-pattern Assignment (OSPA) [23] distances.

To the authors' knowledge, despite widespread quantification of localisation estimation error, the absolute difference between all estimated and actual features in the map is rarely jointly considered[1]. As an example, Figure 2.5 shows a hypothetical posterior map estimate returned by two separate feature mapping filters. If the true feature map, $M = \{m^1, \ldots, m^{\mathfrak{m}_k}\}$ (shown as green circles) and the estimated map $\widehat{M} = \{\widehat{m}^1, \ldots, \widehat{m}^{\hat{\mathfrak{m}}_k}\}$ (shown as black crosses), where $\mathfrak{m}_k$ is the total number of features in the map and $\hat{\mathfrak{m}}_k$ is the estimated number of features in the map, which map estimate is closer to M?. While visual perception may indicate that the left-hand map estimate is superior, an accepted metric to answer this fundamental question is lacking in the mobile robotics community. Suitable error metrics to address this problem, will be the subject of Chapter 4.

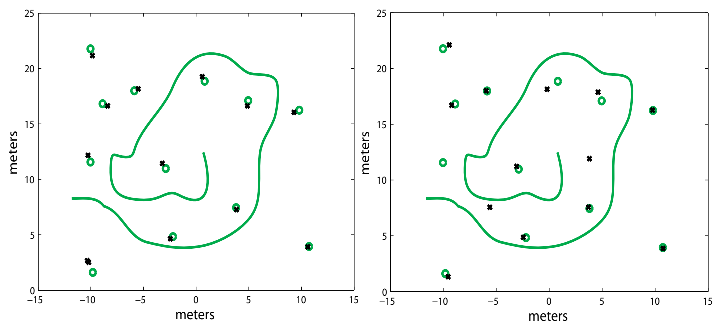

Fig. 2.5 Hypothetical posterior estimates from a feature mapping filter, $\widehat{M}_{left}$ and $\widehat{M}_{right}$, with true feature locations (green circles) and estimated feature locations (black crosses) shown.

2.4 Bayesian FBRM and SLAM with Vectors and Sets

This section outlines the Bayesian recursion which is central to the majority of FBRM and SLAM stochastic mapping algorithms. Let M denote a generic mathematical representation of the environment to be mapped, $Z_k = \{z_k^1, ..., z_k^{\mathfrak{z}_k}\}$ denote the collection of $\mathfrak{z}_k$ sensor measurements at time k and X_k be the vehicle pose, at time k. In the case of FBRM, the aim is to then propagate the posterior density of the map from the measurement and pose history, $Z_{0:k} = [Z_1, ..., Z_k]$ and $X_{0:k} = [X_1, ..., X_k]$ respectively. Maximum a posteriori (MAP) or expected a posteriori (EAP) estimates may then be extracted from the posterior density at each time k.

[1] Approaches examining the consistency of a subset of feature estimates are common however [1, 15].

Assuming that such a density exists, from an optimal Bayesian perspective the posterior Probability Density Function (PDF)

$$p_{k|k}(M|Z_{0:k}, X_{0:k})$$

captures all the relevant statistical information about the map, up to and including time k. The posterior PDF of the map can in principle be propagated in time via the well-known Bayes recursion,

$$p_{k|k}(M|Z_{0:k}, X_{0:k}) \propto g_k(Z_k|M, X_k) p_{k|k-1}(M|Z_{0:k-1}, X_{k-1}). \tag{2.11}$$

For clarity of exposition, this static mapping only problem is adhered to in the first part of this book. This formulation can however be easily extended to the SLAM problem in which the full posterior $p_{k|k}(X_{0:k}, M_k|Z_{0:k})$ can be propagated in time. The formulation can be further extended to incorporate dynamic maps and multiple vehicle SLAM, which will be the subject of the final chapter of this book.

A mathematical representation of the map, M, is required before the likelihood, $g_k(Z|M, X_k)$, and prior, $p_{k|k-1}(M|Z_{0:k-1}, X_{k-1})$, can be well defined. Bayesian based estimation of both occupancy grid (OG), vector FB and RFS FB map representations are now addressed. The following sections highlight the advantages of RFS over vector based formulations, in terms of Bayes optimality.

2.4.1 Bayesian Estimation with Occupancy Grids

Since its inception by Moravec and Elfes [7], the occupancy grid map, denoted $M = [m^1, m^2, \cdots, m^{\mathfrak{m}}]$, has been widely accepted as a viable mathematical representation of a given environment. In the context of an Occupancy Grid, $\mathfrak{m}$, represents a fixed number of spatial cells, usually distributed in the form of a lattice, which are obtained via an *a priori* tessellation of the spatial state space. Each grid cell is then denoted 'Occupied', if a landmark[2] exists in the cell, and 'Empty', if the cell is empty of landmarks. The recursion of equation 2.11, then propagates the posterior density on the occupancy grid, typically by invoking a grid cell independence assumption,

$$p_{k|k}(M|Z_{0:k}, X_{0:k}) = \prod_{i=1}^{i=\mathfrak{m}} p_{k|k}(m^i|Z_{0:k}, X_{0:k})$$

with $p_{k|k}(m^i|Z_{0:k}, X_{0:k})$ denoting the probability, α, of a landmark existing in cell m^i. The occupancy grid environment representation is attractive due

[2] Note in this work, a 'landmark' refers to any physical object in the environment. A 'feature' then refers to a simplified representation of a landmark.

to its ability to model arbitrary landmarks as the cell number tends to infinity. An important, and rarely examined aspect of the grid approach however, is that the *number* of grid cells, $\mathfrak{m}$, is inherently known *a priori*. This has a fundamental impact on the optimality of the Bayesian recursion since it means that only the occupancy of each cell needs to be estimated and not the *number* of grid cells. Thus a vector-valued map state can be used to represent the grid cells since, in this case, it is not necessary to encapsulate uncertainty in the number of states. Given the existence estimation state space of the representation, stochastic detector dependent measurement likelihoods are also required [16]. Much of the grid based mapping literature distributes occupancy uncertainty in the spatial space to model the uncertainty of the sensing and map estimation process [24], [20]. However, while this environmental representation deals with detection and spurious measurements to propagate the landmark existence estimate, such a representation in its mathematical structure does not inherently encapsulate and propagate the spatial uncertainty of sensor measurements [16]. This will be explained further in Section 2.5. A true spatial state space is explicitly considered in the feature map representation described next.

2.4.2 Bayesian Estimation with a Vector Feature Map

While defining a vector-valued feature map representation may appear to be a trivial case of terminology, in fact it has already been demonstrated that it has numerous mathematical consequences [3], namely an inherent rigid ordering of variables and a fixed length. The feature map approach has long been recognised as a "*a state estimation problem involving a variable number of dimensions (features)*" [25], however a vector representation for a feature map can only represent a *fixed* number of features. That is, the posterior vector feature map density,

$$p_{k|k}(M = [m^1, m^2, \cdots, m^{\hat{\mathfrak{m}}_k}]|Z_{0:k}, X_{0:k})$$

represents the spatial density of $\hat{\mathfrak{m}}_k$ features only, and does not encapsulate uncertainty in feature number. This limitation of vector representations is not new to robotics researchers and the sub-optimal map management methods mentioned in Section 2.2 and shown in Figure 1.1 are subsequently adopted to adjust the estimate of $\mathfrak{m}_k$ through 'augmenting' and 'pruning' filtering/heuristic based operations [17], [1]. More advanced methods, which allow reversible data association across a finite window of time frames have also been considered [18], [26]. Furthermore, the order of the features $1, \ldots, \hat{\mathfrak{m}}_k$ in the vector is fixed, coupled with a vector-valued measurement equation (also of rigid order), which leads to the need for costly data association algorithms to decide the measurement-feature assignment. This can be seen in the case of SLAM, as applying Bayes theorem (equation 2.11) to a vector valued map involves the following steps:

- Predicted time update, based on the previous vehicle pose and previous inputs to the robot (typically speed, steering commands):

$$\begin{aligned} p_{k|k-1}(X_{0:k}, M_k|Z_{0:k-1}, U_{0:k-1}, X_0) = \\ \int f_X(X_{0:k}, M_k|X_{0:k-1}, M_{k-1}, U_{k-1}) \times \\ p_{k-1|k-1}(X_{0:k-1}, M_{k-1}|Z_{0:k-1}, U_{0:k-2}, X_0)dX_{k-1} \end{aligned} \tag{2.12}$$

- Acquire the measurement vector Z_k.
- Carry out feature associations *before* the application of Bayes theorem.
- Perform the measurement update:

$$p_{k|k}(X_{0:k}, M_k|Z_{0:k}, U_{0:k-1}, X_0) = \frac{g_k(Z_k|M_k, X_k)p_{k|k-1}(X_{0:k}, M_k|Z_{0:k-1}, U_{0:k-1}, X_0)}{\int\int g_k(Z_k|M_k, X_k)p_{k|k-1}(X_{0:k}, M_k|Z_{0:k-1}, U_{0:k-1}, X_0)dX_k dM_k} \tag{2.13}$$

- Perform independent map management.

It is important to note that when both the measurement likelihood[3] $g_k(Z_k|M_k, X_k)$ and the predicted SLAM state $p_{k|k-1}(X_{0:k}, M_k|Z_{0:k-1}, U_{0:k-1}, X_0)$, in the numerator of equation 2.13, are represented by random vectors, they must be of compatible dimensions *before* the Bayes update can be carried out. This is why the independent data association step is necessary. It is also of importance to note that the SLAM state and feature number are not jointly propagated or estimated.

The next section introduces the finite set representation for a feature map, which yields the joint encapsulation of the feature number and spatial uncertainty as well as their optimal joint estimation.

2.4.3 Bayesian Estimation with a Finite Set Feature Map

Inconsistencies in the classical vector feature map representation can be demonstrated through a simple question: How is a map with no features represented by a vector? A set can represent such a case through the null set. Furthermore, due to the unknown number of features in a map and the

[3] Note the notational change for the measurement likelihood. Throughout this book, the p_k notation is used only on the densities from which state estimates are to be extracted via a suitable Bayes optimal estimator. While not commonly used, we believe that denoting the measurement likelihood by g_k, adds clarity and improves readability.

physical insignificance of their order, the feature map can be naturally represented as a finite set, $\mathcal{M} = \{m^1, m^2, \cdots, m^{\mathfrak{m}}\}$. A random finite set (RFS) then encapsulates the uncertainty in the finite set, i.e. uncertainty in feature number and their spatial states. Thus, an RFS feature map can be completely specified by a discrete distribution that models the uncertainty in the number of features, and continuous joint densities that model their spatial uncertainty, conditioned on a given number estimate. In a similar vein to the previous vector feature map (for FBRM), an RFS can be described by its PDF

$$p_{k|k}(\mathcal{M} = \{m^1, m^2, \cdots, m^{\hat{\mathfrak{m}}_k}\} | \mathcal{Z}_{0:k}, X_{0:k})$$

and propagated through a Bayesian recursion as follows:

- Predicted time update, based on the previous vehicle pose and previous inputs to the robot:

$$\begin{aligned} p_{k|k-1}(X_{0:k}, \mathcal{M}_k | \mathcal{Z}_{0:k-1}, U_{0:k-1}, X_0) = \\ \int f_X(X_{0:k}, \mathcal{M}_k | X_{0:k-1}, \mathcal{M}_{k-1}, U_{k-1}) \times \\ p_{k-1|k-1}(X_{0:k-1}, \mathcal{M}_{k-1} | \mathcal{Z}_{0:k-1}, U_{0:k-2}, X_0) dX_{k-1} \end{aligned} \quad (2.14)$$

- Acquire the measurement set $\mathcal{Z}_k$.
- Perform the measurement update:

$$\begin{aligned} p_{k|k}(X_{0:k}, \mathcal{M}_k | \mathcal{Z}_{0:k}, U_{0:k-1}, X_0) = \\ \frac{g_k(\mathcal{Z}_k | \mathcal{M}_k, X_k) p_{k|k-1}(X_{0:k}, \mathcal{M}_k | \mathcal{Z}_{0:k-1}, U_{0:k-1}, X_0)}{\int \int g_k(\mathcal{Z}_k | \mathcal{M}_k, X_k) p_{k|k-1}(X_{0:k}, \mathcal{M}_k | \mathcal{Z}_{0:k-1}, U_{0:k-1}, X_0) dX_k \delta \mathcal{M}_k} \end{aligned} \quad (2.15)$$

  where δ implies set integration.

Contrary to the vector based implementation of Bayes theorem in equation 2.13, it is important to note that the measurement likelihood $g_k(\mathcal{Z}_k | \mathcal{M}_k, X_k)$ and predicted SLAM state $p_{k|k-1}(X_{0:k}, \mathcal{M}_k | \mathcal{Z}_{0:k-1}, U_{0:k-1}, X_0)$ in the numerator of equation 2.15, are finite set statistics (FISST) representing the RFS, which do not have to be of compatible dimensions.

Integration over the map in the denominator of equation 2.15 requires integration over all possible feature maps (all possible locations *and* numbers of features). By adopting an RFS map representation, integrating over the map becomes a set integral. This feature map recursion therefore encapsulates the inherent feature number uncertainty of the map, introduced by detection uncertainty, spurious measurements and vehicle manoeuvres, as well as the feature location uncertainty introduced by measurement noise. Features are not rigidly placed in a map vector, nor are measurements simply a direct function of the map state, due to the explicit modelling of clutter. Therefore, contrary to previous vector represented approaches, no explicit measurement-feature assignment (the data association problem) is required.

Hence, by adopting an RFS representation of the mapped and observed features, Bayes theorem can be applied to jointly estimate the feature state, number and vehicle pose for SLAM.

2.5 Further Attributes of the RFS Representation

To date, a map representation which unifies the existence filtering state-space of the occupancy map representation and the spatial state-space of the feature map representation remains elusive. While previous researchers generally adopt independent filters to propagate the spatial and existence posteriors of a vector feature map, such an approach leads to some theoretical inconsistencies. For instance, consider the posterior density for a single feature map, $p_{k|k}(M{=}[m]|Z_{0:k})$. In order for the Bayesian recursion of equation 2.11 to be valid, the density must be a PDF, i.e. $\int p_{k|k}(M|Z_{0:k})dM = 1$. This however implicitly implies that the feature definitely exists somewhere in the map. By using a separate existence filter to obtain an existence probability of α, the implication is that $\int p_{k|k}(M|Z_{0:k})dM = \alpha$, which subsequently violates a fundamental property of a PDF $\forall\ \alpha \neq 1$. For such a case, it is evident that a vector-valued feature map representation cannot jointly incorporate feature existence and location uncertainty.

An RFS framework can readily overcome these issues. For instance, an analogous joint recursion can be obtained by adopting a Poisson RFS to represent the feature map. This approach does not maintain an existence estimate on each feature, but instead propagates a density which represents the mean number of features in the map as well as their spatial densities. An alternative RFS map model is a multi-Bernoulli RFS, as will be shown in Chapter 3 (equation 3.1), which can jointly encapsulate the positional and existence uncertainty of each individual feature under a valid PDF, which can be subsequently propagated and estimated via the so called MeMBer Filter.

2.6 Summary

This chapter has provided several motivations for the theoretical representation of feature based maps to take the form of RFSs as opposed to the classically used random vectors. Indeed it has been demonstrated that a vector representation of the map introduces many algorithmic/mathematical consequences, in the forms of the ordering of features within the estimated map and observation vectors; the data association problem; the map management problem; feature map error quantification and the problems of vector dimensionality differences within a vector based, Bayes recursion. It was demonstrated that these mathematical consequences result in algorithmic routines which typically augment or truncate vectors outside of the Bayesian FBRM/SLAM recursions, resulting in Bayes optimality only being possible on a predetermined subset of the detected features.

The RFS representation has been conceptually introduced as a means in which the Bayes optimal estimation of both feature number and spatial state, is achievable without the need for such pre-Bayesian augmenting/truncating methods. Indeed, it was highlighted that no data association is necessary at all, under the RFS framework. This naturally leads us to the scope of Chapter 3, in which mathematically tractable, RFS based approximations are derived, for Bayes optimal FBRM and SLAM.