

37. Simultaneous Localization and Mapping

Simultaneous Localization and Mapping

Sebastian Thrun, John J. Leonard

This chapter provides a comprehensive introduction in to the *simultaneous localization and mapping problem*, better known in its abbreviated form as **SLAM**. **SLAM** addresses the problem of a robot navigating an unknown environment. While navigating the environment, the robot seeks to acquire a map thereof, and at the same time it wishes to localize itself using its map. The use of **SLAM** problems can be motivated in two different ways: one might be interested in detailed environment models, or one might seek to maintain an accurate sense of a mobile robot's location. **SLAM** serves both of these purposes.

We review three major paradigms of algorithms from which a huge number of recently published methods are derived. First comes the traditional approach, which relies on the extended Kalman filter (**EKF**) for representing the robot's best estimate. The second paradigm draws its intuition from the fact that the **SLAM** problem can be viewed as a sparse graph of constraints, and it applies nonlinear optimization for recovering the map and the robot's locations. Finally, we survey the particle filter paradigm, which applies nonparametric

| | |
|---|-----|
| 37.1 Overview | 871 |
| 37.2 SLAM: Problem Definition | 872 |
| 37.2.1 Mathematical Basis | 872 |
| 37.2.2 Example: SLAM in Landmark Worlds | 873 |
| 37.2.3 Taxonomy of the SLAM Problem..... | 873 |
| 37.3 The Three Main SLAM Paradigms | 875 |
| 37.3.1 Extended Kalman Filters | 875 |
| 37.3.2 Graph-Based Optimization Techniques..... | 877 |
| 37.3.3 Particle Methods | 881 |
| 37.3.4 Relation of Paradigms | 883 |
| 37.4 Conclusion and Future Challenges | 885 |
| 37.5 Suggestions for Further Reading | 886 |
| References | 886 |

density estimation and efficient factorization methods to the **SLAM** problem. This chapter discusses extensions of these basic methods. It elucidates variants of the **SLAM** problem and proposes a taxonomy for the field. Relevant research is referenced extensively, and open research problems are discussed.

37.1 Overview

This chapter provides a comprehensive introduction into one of the key enabling technologies of mobile robot navigation: *simultaneous localization and mapping* (**SLAM**). **SLAM** addresses the problem of acquiring a spatial map of a mobile robot environment while simultaneously localizing the robot relative to this model. The **SLAM** problem is generally regarded as one of the most important problems in the pursuit of building truly autonomous mobile robots. Despite significant progress in this area, it still poses great challenges. At present, we have robust methods for

mapping environments that are static, structured, and of limited size. Mapping unstructured, dynamic, or large-scale environments remains largely an open research problem.

The historical roots of **SLAM** can be traced back to *Gauss* [37.1], who is largely credited with inventing the least-squares method, for calculating planetary orbits. In the 20th century, a number of fields outside robotics have studied the making of environment models from a moving sensor platform, most notably in *photogrammetry* [37.2] and *computer vision* [37.3,4]. **SLAM** builds

on this work, often extending the basic paradigms into more scalable algorithms.

This chapter begins with a definition of the **SLAM** problem, which shall include a brief taxonomy of different versions of the problem. The centerpiece of this chapter is a layman's introduction into the three major paradigms in this field, and the various extensions that exist. As the reader will quickly recognize, there is no single best solution to the **SLAM** problem. The method

chosen by the practitioner will depend on a number of factors, such as the desired map resolution, the update time, the nature of the features in the map, and so on. Nevertheless, the three methods discussed in this chapter cover the major paradigms in this field. For an in-depth study of **SLAM** algorithms, we refer the reader to a recent textbook on probabilistic robotics, which dedicates a number of chapters to the topic of **SLAM** [37.5]. Also see [37.6, 7] for a recent in-depth tutorial for **SLAM**.

37.2 SLAM: Problem Definition

37.2.1 Mathematical Basis

The **SLAM** problem is defined as follows. A mobile robot roams an unknown environment, starting at a location with known coordinates. Its motion is uncertain, making it gradually more difficult to determine its global coordinates. As it roams, the robot can sense its environment. The **SLAM** problem is the problem of building a map the environment while simultaneously determining the robot's position relative to this map.

Formally, **SLAM** is best described in probabilistic terminology. Let us denote time by t , and the robot location by x_t . For mobile robots on a flat ground, x_t is usually a three-dimensional vector, consisting of its two-dimensional coordinate in the plane plus a single rotational value for its orientation. The sequence of locations, or *path*, is then given as

$$X_T = \{x_0, x_1, x_2, \dots, x_T\}. \quad (37.1)$$

Here T is some terminal time (T might be ∞). The initial location x_0 is known; other positions cannot be sensed.

Odometry provides relative information between two consecutive locations. Let u_t denote the odometry that characterized the motion between time $t - 1$ and time t ; such data might be obtained from the robot's wheel encoders or from the controls given to those motors. Then the sequence

$$U_T = \{u_1, u_2, u_3, \dots, u_T\} \quad (37.2)$$

characterizes the relative motion of the robot. For noise-free motion, U_T would be sufficient to recover the past X_T from the initial location x_0 . However, odometry measurements are noisy, and path-integration techniques inevitably diverge from the truth.

Finally, the robot senses objects in the environment. Let m denote the *true* map of the environment. The

environment may be comprised of landmarks, objects, surfaces, etc., and m describes their locations. The environment map m is typically assumed to be time invariant (i.e., static).

The robot measurements establish information between features in m and the robot location x_t . If we, without loss of generality, assume that the robot takes exactly one measurement at each point in time, the sequence of measurements is given as

$$Z_T = \{z_1, z_2, z_3, \dots, z_T\}. \quad (37.3)$$

Figure 37.1 illustrates the variables involved in the **SLAM** problem. It shows the sequence of locations and sensor measurements, and the causal relationships between these variables. Such a diagram is known as a *graphical model*. It is useful in understanding the dependencies in the **SLAM** problem.

The **SLAM** problem is now the problem of recovering a model of the world m and the sequence of robot locations X_T from the odometry and measure-

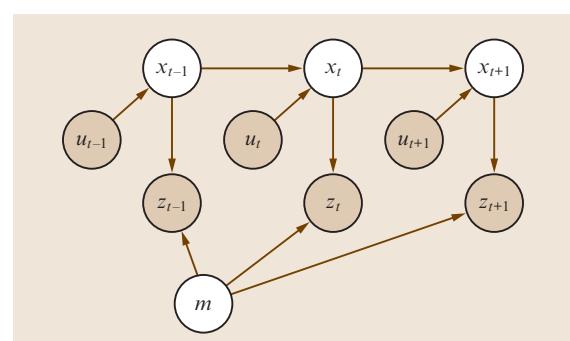


Fig. 37.1 Graphical model of the **SLAM** problem. Arcs indicate causal relationships, and shaded nodes are directly observable to the robot. In **SLAM**, the robot seeks to recover the unobservable variables



ment data. The literature distinguishes two main forms of the **SLAM** problem, which are both of equal practical importance. One is known as the *full SLAM problem*: it involves estimating the posterior over the entire robot path together with the map:

$$p(X_T, m | Z_T, U_T). \quad (37.4)$$

Written in this way, the full **SLAM** problem is the problem of calculating the joint posterior probability over X_T and m from the available data. Notice that the variables right of the conditioning bar are all directly observable to the robot, whereas those on the left are the ones that we want. As we shall see, algorithms for the offline **SLAM** problem are often batch, that is, they process all data at the same time.

The second, equally important **SLAM** problem is the *online SLAM problem*. This problem is defined via

$$p(x_t, m | Z_T, U_T). \quad (37.5)$$

Online **SLAM** seeks to recover the present robot location, instead of the entire path. Algorithms that address the online problem are usually incremental and can process one data item at a time. In the literature such algorithms are typically called *filters*.

To solve either **SLAM** problem, the robot needs to be endowed with two more models: a mathematical model that relates odometry measurements u_t to robot locations x_{t-1} and x_t and a model that relates measurements z_t to the environment m and the robot location x_t . These models correspond to the arcs in Fig. 37.1.

In **SLAM**, it is common to think of those mathematical models as probability distributions: $p(x_t | x_{t-1}, u_t)$ characterizes the probability distribution of the location x_t assuming that a robot started at a known location x_{t-1} and measured odometry data u_t , and likewise $p(z_t | x_t, m)$ is the probability for measuring z_t if this measurement is taken at a known location x_t in a known environment m . Of course, in the **SLAM** problem we do *not* know the robot location, and neither do we know the environment. As we shall see, Bayes rule takes care of this, by transforming these mathematical relationships into a form where we can *recover* probability distributions over those latent variables from the measured data.

37.2.2 Example: SLAM in Landmark Worlds

One common setting of **SLAM** involves the assumption that the environment is populated by point landmarks. When building two-dimensional (2-D) maps, point landmarks may correspond to door posts and corners of

rooms, which, when projected into a 2-D map are characterized by a point coordinate. In a 2-D world, each point landmark is characterized by two coordinate values. Hence the world is a vector of size $2N$, where N is the number of point landmarks in the world.

In a commonly studied setting, the robot can sense three things: the relative range to nearby landmarks, their relative bearing, and the identity of these landmarks. The range and bearing may be noisy, but in the most simple case the identity of the sensed landmarks is known perfectly.

To model such a setup, one begins with defining the *exact*, noise-free measurement function. The measurement function h describes the workings of the sensors: it accepts as input a description of the environment m and a robot location x_t , and it computes the measurement:

$$h(x_t, m). \quad (37.6)$$

Computing h is straightforward in our simplified landmark setting; it is a simple exercise in trigonometry.

The probabilistic measurement model is derived from this measurement function by adding a noise term; it is a probability distribution that peaks at the noise-free value $h(x_t, m)$ but allows for measurement noise:

$$p(z_t | x_t, m) \sim \mathcal{N}(h(x_t, m), Q_t). \quad (37.7)$$

Here \mathcal{N} denotes the two-dimensional normal distribution, which is centered at $h(x_t, m)$. The 2-by-2 matrix Q_t is the noise covariance, indexed by time.

The motion model is derived from a kinematic model of robot motion. Given the location vector x_{t-1} and the motion u_t , textbook kinematics tells us how to calculate x_t . Let this function be denoted by g :

$$g(x_{t-1}, u_t). \quad (37.8)$$

The motion model is then defined by a normal distribution centered at $g(x_{t-1}, u_t)$ but subject to Gaussian noise:

$$p(x_t | x_{t-1}, u_t) = \mathcal{N}(g(x_{t-1}, u_t), R_t). \quad (37.9)$$

Here $Q R_t$ is a covariance; it is of size 3-by-3, since the location is a three-dimensional vector.

With these definitions, we have all we need to develop a **SLAM** algorithm. While in the literature, point landmark problems with range-bearing sensing are by far the most widely studied, **SLAM** algorithms are not confined to landmark worlds. However, no matter what the map representation and the sensor modality, any **SLAM** algorithm needs a similarly crisp definition of

the features in m , the measurement model $p(z_t | x_t, m)$, and the motion model $p(x_t | x_{t-1}, u_t)$.

37.2.3 Taxonomy of the SLAM Problem

SLAM problems are distinguished along a number of different dimensions. Most important research papers identify the type of problems addressed by making the underlying assumptions explicit. We already encountered one such distinction: full versus online. Other common distinctions are as follows.

Volumetric Versus Feature-Based

In volumetric SLAM, the map is sampled at a resolution high enough to allow for photorealistic reconstruction of the environment. The map m in volumetric SLAM is usually quite high dimensional, with the result that the computation can be quite involved. Feature-based SLAM extracts sparse features from the sensor stream. The map is then only comprised of features. Our point-landmark example is an instance of feature-based SLAM. Feature-based SLAM techniques tend to be more efficient, but their results may be inferior to volumetric SLAM due to the fact that the extraction of features discards information in the sensor measurements.

Topological Versus Metric

Some mapping techniques recover only a qualitative description of the environment, which characterizes the relation of basic locations. Such methods are known as topological. A topological map might be defined over a set of distinct places and a set of qualitative relations between these places (e.g., place A is adjacent to place B). Metric SLAM methods provide metric information between the relation of such places. In recent years, topological methods have fallen out of fashion, despite ample evidence that humans often use topological information for navigation.

Known Versus Unknown Correspondence

The correspondence problem is the problem of relating the identity of sensed things to other sensed things. In the landmark example above, we assumed that the identity of landmarks is known. Some SLAM algorithms make such an assumption, while others do not. The algorithms that do not make this assumption provide special mechanisms for estimating the correspondence of measured features to previously observed landmarks in the map. The problem of estimating the correspondence is known as the *data association problem*, and is one of the most difficult problems in SLAM.

Static Versus Dynamic

Static SLAM algorithms assume that the environment does not change over time. Dynamic methods allow for changes in the environment. The vast majority of the literature on SLAM assumes static environments; dynamic effects are often treated just as measurement outliers. Methods that reason about motion in the environment are more involved, but they tend to be more robust in most applications.

Small Versus Large Uncertainty

SLAM problems are distinguished by the degree of location uncertainty that they can handle. The most simple SLAM algorithms allow only for small errors in the location estimate. They are good for situations in which a robot goes down a path that does not intersect itself, and then returns along the same path. In many environments it is possible to reach the same location from multiple directions. Here the robot may accrue a large amount of uncertainty. This problem is known as the *loop-closing problem*. When closing a loop, the uncertainty may be large. The ability to close loops is a key characteristic of modern-day SLAM algorithms. The uncertainty can be reduced if the robot can sense information about its position in some absolute coordinate frame, e.g., through the use of a satellite-based global positioning system (GPS) receiver.

Active Versus Passive

In passive SLAM algorithms, some other entity controls the robot, and the SLAM algorithm is purely observing. The vast majority of algorithms are of this type; they give the robot designer the freedom to implement arbitrary motion controllers, and pursue arbitrary motion objectives. In active SLAM, the robot actively explores its environment in the pursuit of an accurate map. Active SLAM methods tend to yield more accurate maps in less time, but they constrain the robot motion. There exist hybrid techniques in which the SLAM algorithm controls only the pointing direction of the robot's sensors, but not the motion direction.

Single-Robot Versus Multirobot SLAM

Most SLAM problems are defined for a single-robot platform, although recently the problem of multirobot exploration has gained in popularity. Multirobot SLAM problems come in many flavors. In some, robots are able to observe each other, while in others robots are told their relative initial locations. Multirobot SLAM problems are also distinguished by the type of communication allowed between the different robots. In some, the robots can



communicate with no latency and infinite bandwidth. More realistic are setups in which only nearby robots can communicate, and the communication is subject to latency and bandwidth limitations.

As this taxonomy suggests, there exists a flurry of **SLAM** algorithms. Most modern-day conferences

dedicate multiple sessions to **SLAM**. This chapter focuses on the very basic **SLAM** setup. In particular it assumes a static environment with a single robot. Extensions are discussed towards the end of this chapter, in which the relevant literature is also discussed.

37.3 The Three Main SLAM Paradigms

This section reviews three basic **SLAM** paradigms, from which most others are derived. The first, known as extended Kalman filter (**EKF**) **SLAM**, is historically the earliest but has recently become slightly unpopular due to its limiting computational properties. The second, which is based on graphical representations, successfully applies sparse nonlinear optimization methods to the **SLAM** problem, and has become the main paradigm for solving the full **SLAM** problem. The third and final method uses nonparametric statistical filtering techniques known as particle filters. It is a popular method for online **SLAM**, and provides a fresh new solution to the data association problem in **SLAM**.

37.3.1 Extended Kalman Filters

Historically, the **EKF** formulation of **SLAM** is the earliest, and perhaps the most influential. **EKF SLAM** was introduced in [37.8–10] and [37.11, 12], which were the first papers to propose the use of a single state vector to estimate the locations of the robot and a set of features in the environment, with an associated error covariance matrix representing the uncertainty in these estimates, including the correlations between the vehicle and feature state estimates. As the robot moves through its environment taking measurements, the system state vector and covariance matrix are updated using the the extended Kalman filter [37.13–15]. As new features are observed, new states are added to the system state vector; the size of the system covariance matrix grows quadratically.

This approach assumes a metrical, feature-based environmental representation, in which objects can be effectively represented as points in an appropriate parameter space. The position of the robot and the locations of features form a network of uncertain spatial relationships. The development of appropriate representations is a critical issue in **SLAM**, and intimately related to the topics of sensing and world modeling discussed in Chap. 36 and in Part C in this handbook.

The **EKF** algorithm represents the robot estimate by a multivariate Gaussian:

$$p(x_t, m \mid Z_T, U_T) \sim \mathcal{N}(\mu_t, \Sigma_t). \quad (37.10)$$

The high-dimensional vector μ_t contains the robot's best estimate of its own location and the location of the features in the environment. In our point-landmark example, the dimension of μ_t would be $3 + 2N$, since we need three variables to represent the robot location and $2N$ variables for the N landmarks in the map.

The matrix Σ_t is the covariance of the robot's assessment of its expected error in the guess μ_t . As a quadratic matrix, Σ_t is of size $(3 + 2N) \times (3 + 2N)$. In **SLAM**, this matrix is usually distinctly non-sparse. The off-diagonal elements capture the correlations in the estimates of different variables. Nonzero correlations come along because the robot's location is uncertain, and as a result the locations of the landmarks in the maps are uncertain. The importance of maintaining those off-diagonal elements is one of the key properties of **EKF SLAM** [37.16].

The **EKF SLAM** algorithm is easily derived for our point-landmark example. Suppose, for a moment that the motion function g and the measurement function h were *linear* in their arguments. Then the vanilla Kalman filter, as described in any textbook on Kalman filtering, would be applicable. **EKF SLAM** linearizes the functions g and h using Taylor-series expansion – again, this is standard textbook material. Thus, in its most basic form (and in the absence of any data association problems), **EKF SLAM** is nothing but the application of the vanilla **EKF** to the online **SLAM** problem.

Figure 37.2 illustrates the **EKF SLAM** algorithm for an artificial example. The robot navigates from a start pose that serves as the origin of its coordinate system. As it moves, its own pose uncertainty increases, as indicated by uncertainty ellipses of growing diameter. It also senses nearby landmarks and maps them with an uncertainty that combines the fixed measurement uncertainty with the increasing pose uncertainty. As a result, the uncertainty in the landmark locations grows over time. The

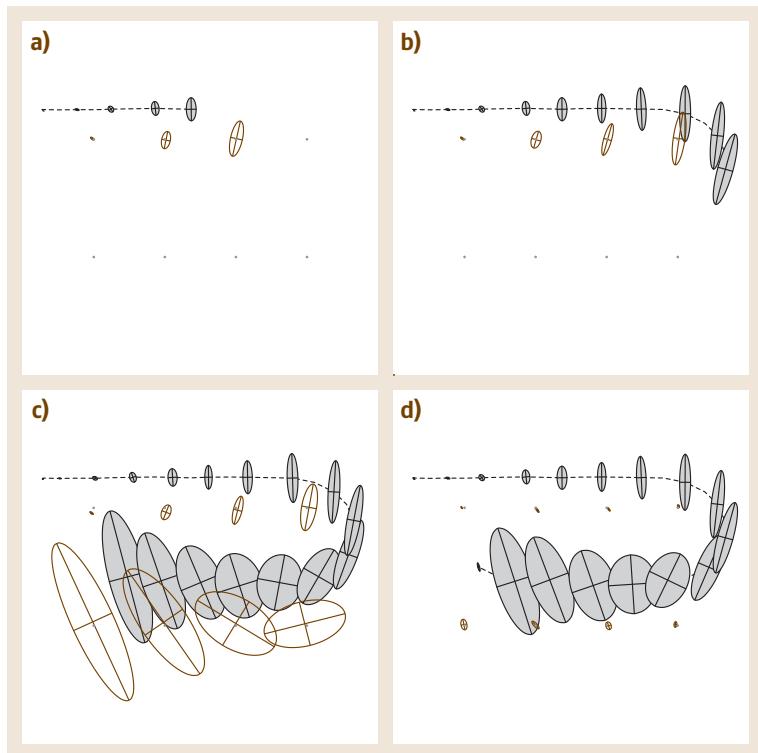


Fig. 37.2a–d EKF applied to the online SLAM problem. The robot’s path is a *dotted line*, and its estimates of its own position are *shaded ellipses*. Eight distinguishable landmarks of unknown location are shown as *small dots*, and their location estimates are shown as *white ellipses*. In (a–c) the robot’s positional uncertainty is increasing, as is its uncertainty about the landmarks it encounters. In (d) the robot senses the first landmark again, and the uncertainty of *all* landmarks decreases, as does the uncertainty of its current pose. (Image courtesy of Michael Montemerlo, Stanford University)

interesting transition is illustrated in Fig. 37.2d: Here the robot observes the landmark it saw in the very beginning of mapping, and whose location is relatively well known. Through this observation, the robot’s pose error is reduced, as indicated in Fig. 37.2d – notice the very small error ellipse for the final robot pose. This observation also reduces the uncertainty for other landmarks in the map. This phenomenon arises from a correlation that is expressed in the covariance matrix of the Gaussian posterior. Since most of the uncertainty in earlier landmark estimates is caused by the robot pose, and since this very uncertainty persists over time, the location estimates of those landmarks are correlated. When gaining information on the robot’s pose, this information spreads to previously observed landmarks. This effect is probably the most important characteristic of the SLAM posterior [37.16]. Information that helps localize the robot is propagated through the map, and as a result improves the localization of other landmarks in the map.

EKF SLAM also addresses the *data association problem*. If the identity of observed features is unknown, the basic EKF idea becomes inapplicable. The solution here is to reason about the most likely data association

when a landmark is observed. This is usually done based on proximity: *which of the landmarks in the map most likely corresponds to the landmark just observed?* The proximity calculation considers the measurement noise and the actual uncertainty in the poster estimate, and the metric used in this calculation is known as a Mahalanobis distance, which is a weighted quadratic distance. To minimize the chances of false data associations, many implementations use visible features to distinguish individual landmarks and associate groups of landmarks observed simultaneously [37.17, 18]. Modern-day implementations also maintain a *provisional landmark list* and only add landmarks to the internal map when they have been observed sufficiently frequently [37.19–22]. With an appropriate landmark definition and careful implementation of the data association step, EKF SLAM becomes a powerful method for feature-based SLAM.

EKF SLAM has been applied successfully to a large range of navigation problems, involving airborne, underwater, indoor, and various other vehicles. Figure 37.3a shows an example result obtained with a state-of-the-art implementation of EKF SLAM. Shown there is an underwater map obtained with the underwater robot Oberon, developed at the University of Sydney, Aus-

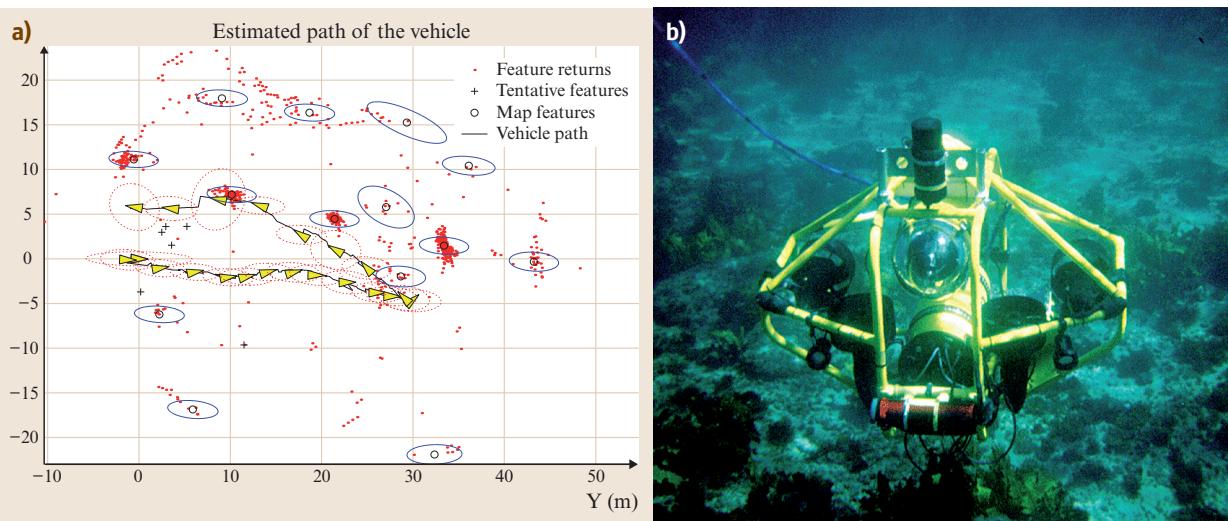


Fig. 37.3 (a) Example of Kalman filter estimation of the map and the vehicle pose. (b) The underwater vehicle Oberon, developed at the University of Sydney. (Images courtesy of Stefan Williams and Hugh Durrant-Whyte, Australian Centre for Field Robotics)

tralia, and shown in Fig. 37.3b. This vehicle is equipped with a pencil-beam, mechanically scanned sonar that can produce high-resolution range and bearing measurements of objects up to 50 m away. To facilitate the mapping problem, researchers have deposited long, small vertical objects in the water, which can be extracted from the sonar scans with relative ease. In this specific experiment, there is a row of such objects, spaced approximately 10 m apart. In addition, a more distant cliff offers additional point features that can be detected using the scanning sonar.

The map shown in Fig. 37.3a depicts the robot path, marked by the triangles connected by a line. Around each triangle one can see an ellipse, which corresponds to the covariance matrix of the Kalman filter estimate projected into the robot location. The ellipse shows the variance; the larger it is, the less certain the robot is about its current pose. Various small dots in Fig. 37.3a show landmark sightings, obtained by searching the sonar scan for small and highly reflective objects. The majority of these sightings is rejected using statistical outlier rejection techniques [37.20]. However, some are believed to correspond to a landmark and are added to the map. At the end of the run, the robot has classified 14 such objects as landmarks, each of which is plotted with the projected uncertainty ellipse in Fig. 37.3a. These landmarks include the artificial landmarks put out by the researchers, but they also include various other terrain features in the vicinity of the robot. The residual pose uncertainty is small.

The basic formulation of **EKF SLAM** assumes that the location of features in the map is fully observable from a single position of the robot. The method has been extended to situations with partial observability, with range-only [37.23] or angle-only [37.24, 25] measurements. The technique has also been utilized using a featureless representation, in which the state consists of current and past robot poses, and measurements take the form of constraints between the poses (derived, for example, from laser scan matching or from camera measurements) [37.26, 27].

A key concern of the **EKF** approach to **SLAM** lies in the quadratic nature of the covariance matrix. A number of researchers have proposed extensions to the **EKF SLAM** algorithms that gain remarkable scalability by decomposing the map into submaps, for which covariances are maintained separately. Relevant literature can be found in [37.19, 28–32]. Other researchers have developed hybrid **SLAM** techniques, which combine **EKF**-style **SLAM** techniques with volumetric map representation; see [37.33–36]. Finally, researchers have developed data association techniques for **SLAM** [37.37–39] based on advanced statistical techniques such as Dempster’s **EM** algorithm [37.40].

37.3.2 Graph-Based Optimization Techniques

A second family of algorithms solves the **SLAM** problem through nonlinear sparse optimization. They draw

their intuition from a graphical representation of the **SLAM** problem. Graph-based techniques were first mentioned in [37.8, 41], but a seminal paper [37.42] provided a first working solution. The representation in this section is closely related to a series of recent papers [37.43–52]. We note that most contemporary techniques are offline and address the full **SLAM** problem, although some online versions exist and will be discussed below.

The basic intuition of graph-based **SLAM** is as follows. Landmarks and robot locations can be thought of as nodes in a graph. Every consecutive pair of locations x_{t-1}, x_t is tied together by an arc that represents the information conveyed by the odometry reading u_t . Other arcs exist between locations x_t and landmarks m_i , assuming that at time t the robot sensed landmark i . Arcs in this graph are soft constraints. Relaxing these constraints yields the robot's best estimate for the map and the full path.

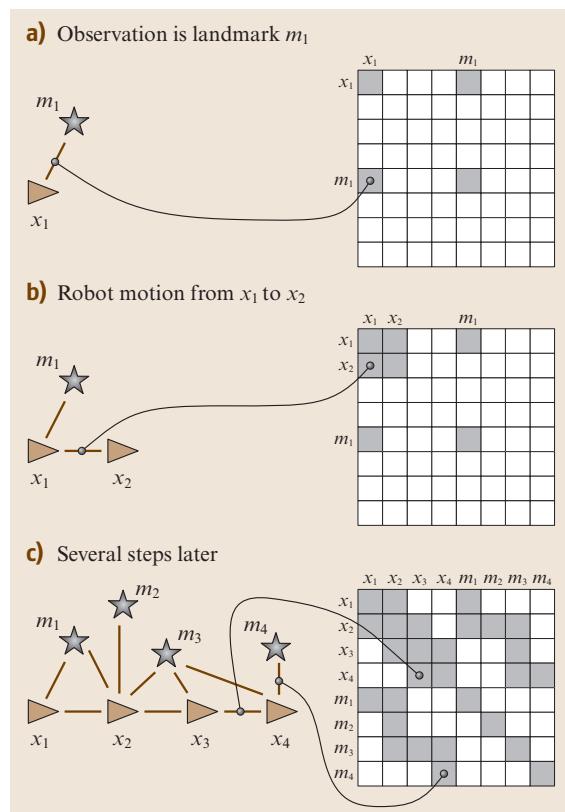


Fig. 37.4a–c Illustration of the graph construction. The *left* diagram shows the graph, the *right* the constraints in matrix form

The construction of the graph is illustrated in Fig. 37.4. Suppose that at time $t = 1$ the robot senses landmark m_1 . This adds an arc in the (yet highly incomplete) graph between x_1 and m_1 . When caching the edges in a matrix format (which happens to correspond to a quadratic equation defining the resulting constraints), a value is added to the elements between x_1 and m_1 , as shown on the right-hand side of Fig. 37.4a.

Now suppose the robot moves. The odometry reading u_2 leads to an arc between nodes x_1 and x_2 , as shown in Fig. 37.4b. Consecutive application of these two basic steps leads to an graph of increasing size, as illustrated in Fig. 37.4c. Nevertheless this graph is *sparse*, in that each node is only connected to a small number of other nodes. The number of constraints in the graph is (at worst) *linear* in the time elapsed and in the number of nodes in the graph.

If we think of the graph as a spring–mass model [37.50], computing the **SLAM** solution is equivalent to computing the state of minimal energy of this model. To see this, note that the graph corresponds to the log-posterior of the full **SLAM** problem (cf. (37.4)):

$$\log p(X_T, m \mid Z_T, U_T). \quad (37.11)$$

Without derivation, we state that this logarithm is of the form

$$\begin{aligned} \log p(X_T, m \mid Z_T, U_T) &= \text{const} \\ &+ \sum_t \log p(x_t \mid x_{t-1}, u_t) \\ &+ \sum_t \log p(z_t \mid x_t, m). \end{aligned} \quad (37.12)$$

Each constraint of the form $\log p(x_t \mid x_{t-1}, u_t)$ is the result of exactly one robot motion event, and it corresponds to an arc in the graph. Likewise, each constraint of the form $\log p(z_t \mid x_t, m)$ is the result of one sensor measurement, to which we can also find a corresponding arc in the graph. The **SLAM** problem is then simply to find the mode of this equation.

$$X^*, m^* = \operatorname{argmax}_{X_T, m} \log p(X_T, m \mid Z_T, U_T). \quad (37.13)$$

Without derivation, we note that, under the Gaussian noise assumption that was made in the point-landmark example, this expression resolves to the following



quadratic form:

$$\begin{aligned} \log p(X_T, m \mid Z_T, U_T) \\ = \text{const} \\ + \sum_t [x_t - g(x_{t-1}, u_t)]^T R_t^{-1} [x_t - g(x_{t-1}, u_t)] \\ + \sum_t [z_t - h(x_t, m)]^T Q_t^{-1} [z_t - h(x_t, m)]. \end{aligned} \quad (37.14)$$

This is a sparse function. A number of efficient optimization techniques can be applied. Common choices include gradient descent, conjugate gradient, and others. Most **SLAM** implementations rely on some sort of iterative linearizing the functions g and h , in which case the objective in (37.14) becomes quadratic in all of its variables.

The graphical paradigm is easily extended to handle the data association problems. This is because (37.14) is easily extended to integrate additional knowledge on data association. Suppose some oracle informed us that landmarks m_i and m_j in the map corresponded to one and the same physical landmark in the world. Then we can either remove m_j from the graph and attach all adjacent arcs to m_i , or we can add a *soft correspondence constraint* [37.53] of the form

$$[m_j - m_i]^T \Gamma [m_j - m_i]. \quad (37.15)$$

Here Γ is a 2-by-2 diagonal matrix whose coefficients determine the penalty for *not* assigning identical locations to two landmarks (hence we want Γ to be large). Since graphical methods are usually used for the full **SLAM** problem, the optimization can be interleaved

with the search for the optimal data association. State-of-the-art implementations rely on **RANSAC** [37.54] or branch-and-bound methods [37.55, 56].

Graphical **SLAM** methods have the advantage that they scale to much higher-dimensional maps than **EKF SLAM**. The key limiting factor in **EKF SLAM** is the covariance matrix, which takes space (and update time) quadratic in the size of the map. No such constraint exists in graphical methods. The update time of the graph is constant, and the amount of memory required is linear (under some mild assumptions). Performing the optimization can be expensive, however. Technically, finding the optimal data association is suspected to be an *NP-hard* problem, although in practice the number of plausible assignments is usually small. The continuous optimization of the log-likelihood function in (37.14) depends among other things on the number and size of loops in the map.

Figure 37.5 shows the result of a state-of-the-art **SLAM** algorithm based on analyzing the constraint graph and a nested search of the best data association. The data for this map was acquired by CMU's Groundhog robot [37.57], built to explore and map abandoned underground mines. Groundhog is equipped with a laser range finder which measures the range to obstacles along a horizontal slice of the world. The specific map shown here covers an area of 250 m × 150 m. The form of the map is known as a *occupancy grid map*, which is due to *Elfes* and *Moravec* [37.58, 59]. Occupancy grid maps use Bayesian reasoning to estimate the posterior probability that a cell is free, thereby accommodating noise in range finders.

As a baseline for comparison, Fig. 37.5a shows a map constructed in a much simpler way: here scans

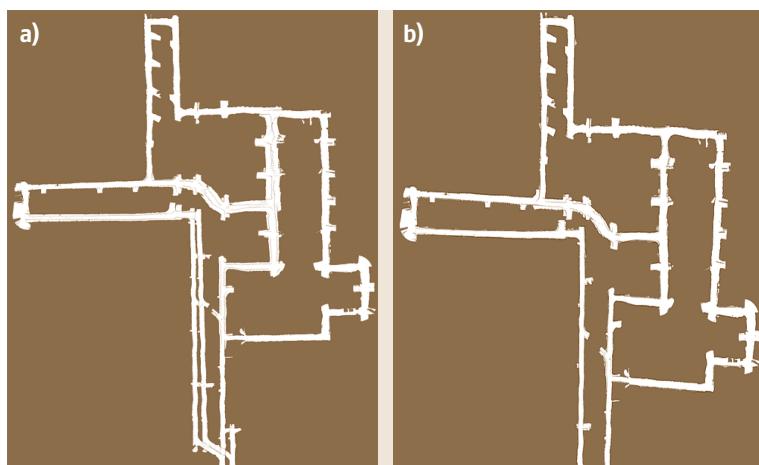


Fig. 37.5a,b An occupancy grid map of an abandoned mine: (a) estimates the data association incrementally, and only in reference to the most recent sensor measurement. The right map (b) is the result of a global data association search and a graphical optimization (Images courtesy of Dirk Hähnel, University of Freiburg)

are localized relative to slightly older scans and, once localized, are added to the map under the assumption that the estimated location is correct. Such a technique is called *scan matching* [37.42]. Scan matching is a **SLAM** method, but it can only accommodate very small amounts of location uncertainties. The failure to close loops is obvious from Fig. 37.5a. In fact, pairwise scan matching can be thought of as a version of a graphical **SLAM** algorithm, but correspondence is only established (and constraints inserted in the graph) between immediately consecutive scans.

To map this data into a graph of manageable size, the algorithm decomposes the map into small local submaps, one for each 5 m of robot travel. Within these 5 m, the maps are sufficiently accurate, as general drift is small and hence scan matching performs essentially flawlessly. Each submap coordinates become a pose node in the GraphSLAM. Adjacent submaps are linked through the relative motion constraints between them. The resulting structure is shown in Fig. 37.5b.

On this graph, we can now perform a branch-and-bound recursive search for correspondences. For finding

good submaps that might correspond, this algorithm uses a correlation analysis for two overlaying maps. Once two suitable maps are found, a soft constraint of the type stated in (37.15) is added to the graph, followed by an optimization step of the resulting set of constraints. Figure 37.6 illustrates the process of data association: each circle corresponds to a new constraint that would be found in the search. The figure illustrates the iterative nature of the search: certain correspondences are only discovered when others have been propagated, and others are dissolved in the process of the search. The final model is stable, in that additional search for new data association induces no further changes. The resulting grid map is shown in Fig. 37.5b.

Other graph-based techniques for **SLAM** have produced similar results. Figure 37.7 shows a map of the same data set generated by [37.26], using an algorithm called *Atlas*. This algorithm decomposes maps into submaps whose relation is maintained through information-theoretic relative link.

We note that the graph-based paradigm is very closely linked to information theory, in that the soft con-

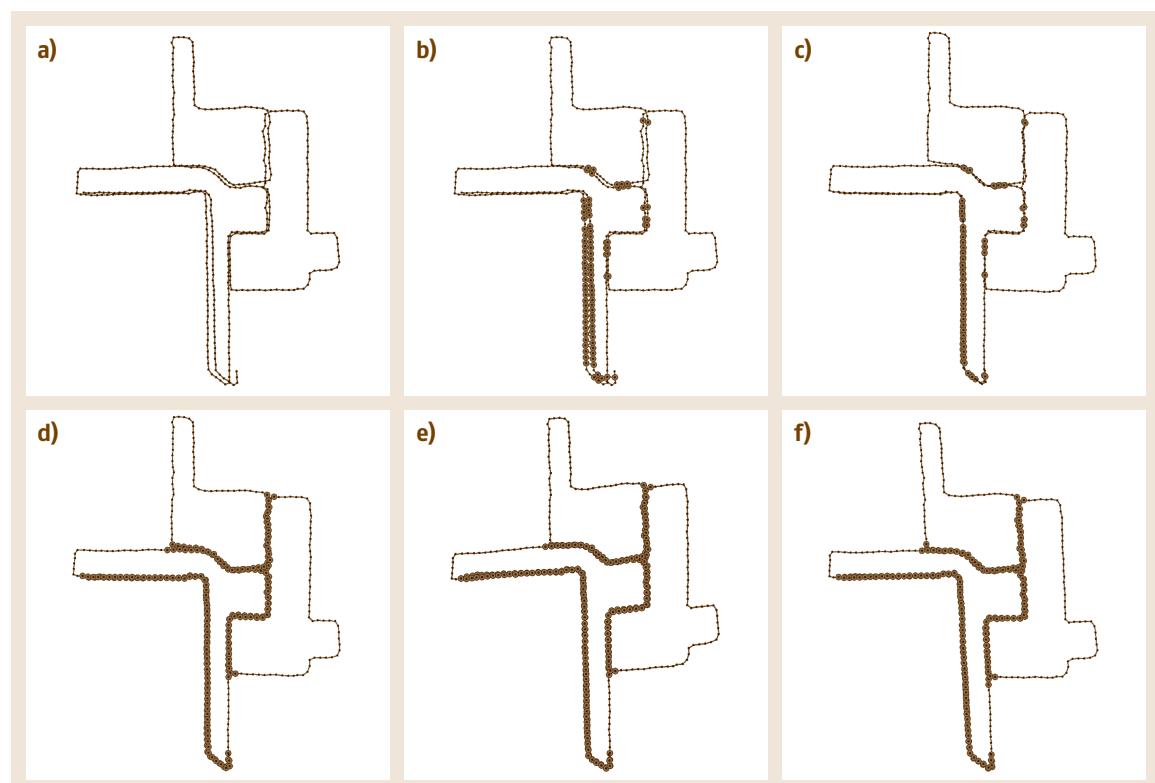


Fig. 37.6a–f Data association search. See text



straints constitute the information the robot has on the world (in an information-theoretic sense [37.60]). Most methods in the field are inherently offline, that is, they optimize for the entire robot path. If the robot path is long, the optimization may become cumbersome. This is one of the disadvantages of the graph-based paradigm. There exists a number of crossovers that manipulate the graph online so as to factor our past robot location variables. The resulting algorithms are filters [37.26, 61–63] and they tend to be intimately related to information filter methods [37.16, 63–67]. Many of the original attempts to decompose **EKF SLAM** representations into smaller submaps to scale up are based on motivations that are not dissimilar to the graphical approach; see [37.29, 30, 68].

As this chapter is being written, graphical and optimization-based **SLAM** algorithm are the subject of intense research. Recent results show that the paradigm scales to maps with 10^8 features [37.26, 43–49, 51, 52, 57]. Arguably, the graph-based paradigm has generated some the largest **SLAM** maps ever built, but usually in an offline fashion.

37.3.3 Particle Methods

The third principal **SLAM** paradigm is based on particle filters. Particle filters can be traced back to [37.69], but they have become popular only in recent years. Particle filter represent a posterior through a set of *particles*. For the novice in **SLAM**, each particle is best thought as a concrete guess as to what the true value of the state may be. By collecting many such guesses into a set of guesses, or set of particles, the particle filters captures a representative sample from the posterior distribution. The particle filter has been shown under mild conditions to approach the true posterior as the particle set size goes to infinity. It is also a nonparametric representation that represents multimodal distributions with ease. In recent years, the advent of extremely efficient microprocessors has made particle filters a popular algorithm [37.70–74].

The key problem with the particle filter in the context of **SLAM** is that the space of maps and robot paths is huge. Suppose we have a map with 1000 features. How many particles would it take to populate that space? In fact, particle filters scale exponentially with the dimension of the underlying state space. Three or four dimensions are thus acceptable, but 100 dimensions are generally not.

The trick to make particle filters amenable to the **SLAM** problem goes back to [37.75, 76]. The trick was introduced into the **SLAM** literature in [37.77], followed by [37.78], who coined the name FastSLAM. Let us

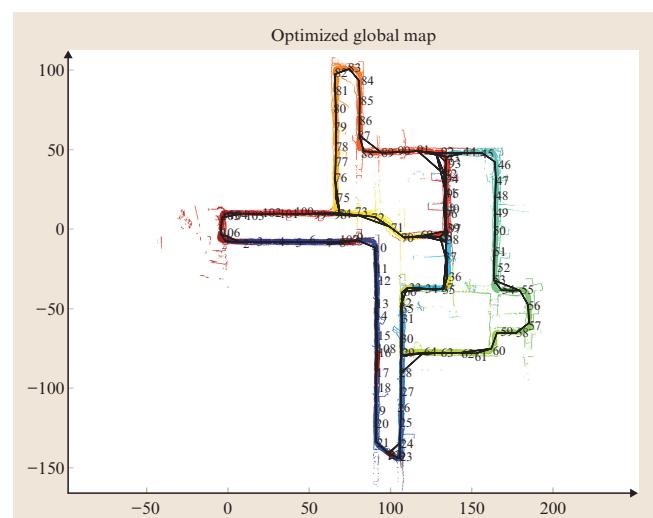


Fig. 37.7 Mine map generated by the *Atlas SLAM* algorithm by [37.26] (after Bosse et al. [37.26])

first explain the basic FastSLAM algorithm on the simplified point-landmark example, and then discuss the justification for this approach.

At any point in time, FastSLAM maintains K particles of the type:

$$X_t^{[k]}, \quad \mu_{t,1}^{[k]}, \dots, \mu_{t,N}^{[k]}, \quad \Sigma_{t,1}^{[k]}, \dots, \Sigma_{t,N}^{[k]}. \quad (37.16)$$

Here $[k]$ is the index of the sample. This expression states that a particle contains

- a sample path $X_t^{[k]}$
- a set of N two-dimensional Gaussians with means $\mu_{t,n}^{[k]}$ and variances $\Sigma_{t,n}^{[k]}$, one for each landmark in the environment

Here n ($1 \leq n \leq N$) is the index of the landmark. From this it follows that K particles possess K path samples. It also possesses KN Gaussians, each of which models exactly one landmark for one of the particles.

Initializing FastSLAM is simple: just set each particle's robot location to its known starting coordinates, and zero the map. The particle update then proceeds as follows.

- When an odometry reading is received, new location variables are generated stochastically, one for each of the particles. The distribution for generating those location particles is based on the motion model

$$x_t^{[k]} \sim p(x_t | x_{t-1}^{[k]}, u^t). \quad (37.17)$$

Here $x_{t-1}^{[k]}$ is the previous location, which is part of the particle. This probabilistic sampling step is easily implemented for any robot whose kinematics can be computed.

- When a measurement z_t is received, two things happen: first, FastSLAM computes for each particle the probability of the new measurement z_t . Let the index of the sensed landmark be n . Then the desired probability is defined as

$$w_t^{[k]} := \mathcal{N}(z_t | x_t^{[k]}, \mu_{t,n}^{[k]}, \Sigma_{t,n}^{[k]}). \quad (37.18)$$

The factor $w_t^{[k]}$ is called the *importance weight*, since it measures how *important* the particle is in the light of the new sensor measurement. As before, \mathcal{N} denotes the normal distribution, but this time it is calculated for a specific value, z_t . The importance weights of all particles are then normalized so that they sum to 1.

Next, FastSLAM draws with replacement from the set of existing particles a set of new particles. The probability of drawing a particle is its normalized importance weight. This step is called *resampling*. The intuition behind resampling is simple: particles for which the measurement is more plausible have a higher chance of surviving the resampling process. Finally, FastSLAM updates for the new particle set the mean $\mu_{t,n}^{[k]}$ and covariance $\Sigma_{t,n}^{[k]}$, based on the

measurement z_t . This update follows the standard **EKF** update rules.

This all sounds complex, but FastSLAM is quite easy to implement. Sampling from the motion model is usually straightforward, since it involves a simple kinematic calculation. Computing the importance of a measurement is, too, straightforward, especially for Gaussian measurement noise. And updating a low-dimensional particle filter is also straightforward. This makes FastSLAM one of the easiest-to-implement algorithms presently available.

FastSLAM has been shown to approximate the full **SLAM** posterior. The derivation of FastSLAM exploits three techniques: Rao–Blackwellization, conditional independence, and resampling. Rao–Blackwellization is the following concept. Suppose we would like to compute a probability distribution $p(a, b)$, where a and b are arbitrary random variables. The vanilla particle filter would draw particles from the joint distributions, that is, each particle would have a value for a and one for b . However, if the conditional $p(b | a)$ can be described in closed form, it is equally legitimate to just draw particles from $p(a)$, and attach to each particle a closed-form description of $p(b | a)$. This trick is known as Rao–Blackwellization, and it yields better results than sampling from the joint distribution. FastSLAM applies this technique, in that it samples from the path posterior $p(X_t^{[k]} | U_t, Z_t)$ and represents the map $p(m | X_t^{[k]}, U_t, Z_t)$ in Gaussian form.

FastSLAM also breaks down the posterior over maps (conditioned on paths) into sequences of low-dimensional Gaussians. The justification for this decomposition is subtle. It arises from a specific conditional independence assumption that is native to **SLAM**. Figure 37.8 illustrates the concept graphically. In **SLAM**, knowledge of the robot path renders all landmark estimates independent. This is easily shown for the graphical network in Fig. 37.8: we find that if we remove the path variables from Fig. 37.8 then the landmark variables are all disconnected [37.79]. Thus, in **SLAM** any dependence between multiple landmark estimates is mediated through the robot path. This subtle but important observation implies that, even if we used a large, monolithic Gaussian for the entire map (one per particle, of course), the off-diagonal element between different landmarks would simply remain zero. It is therefore legitimate to implement the map more efficiently, using N small Gaussians, one for each landmark. This explains the efficient map representation in FastSLAM.

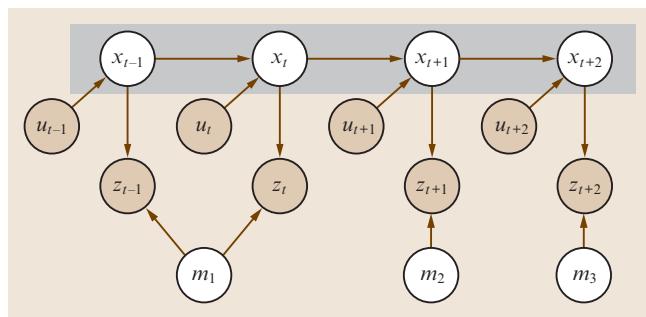


Fig. 37.8 The **SLAM** problem depicted as a Bayes network graph. The robot moves from location x_{t-1} to location x_{t+2} , driven by a sequence of controls. At each location x_t it observes a nearby feature in the map $m = \{m_1, m_2, m_3\}$. This graphical network illustrates that the location variables *separate* the individual features in the map from each other. If the locations are known, there remains no other path involving variables whose value is not known, between any two features in the map. This lack of a path renders the posterior of any two features in the map conditionally independent (given the locations)



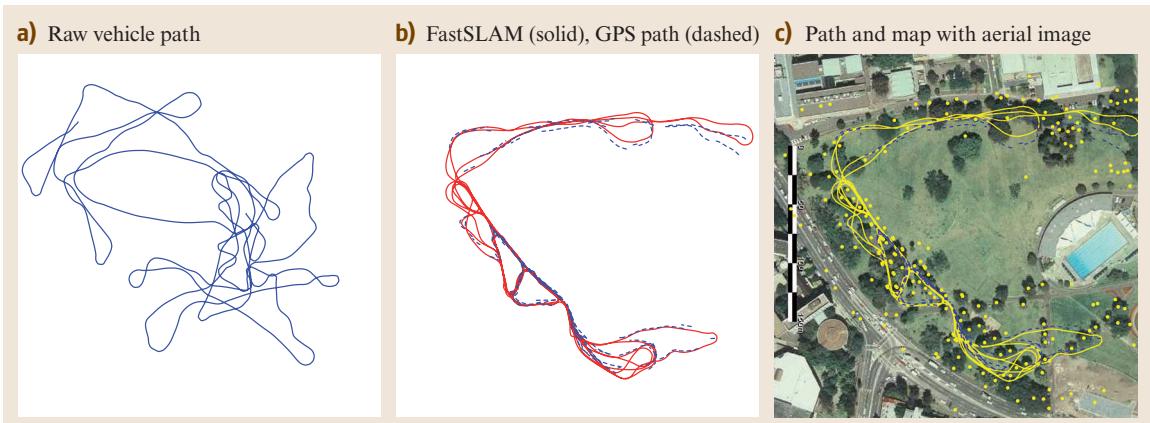


Fig. 37.9 (a) Vehicle path predicted by the odometry; (b) True path (dashed line) and FastSLAM 1.0 path (solid line); (c) Victoria Park results overlaid on aerial imagery with the GPS path in blue (dashed), average FastSLAM 1.0 path in yellow (solid), and estimated features as yellow dots (Data and aerial image courtesy of José Guivant and Eduardo Nebot, Australian Center for Field Robotics)

We also note that FastSLAM uses a particle filter. Derivations of the particle filter can be found in the literature referenced above. Here we note that both steps – the motion and the measurement steps – retain the property that (asymptotically) samples are drawn from the full **SLAM** posterior. This is quite easy to see for the motion update step. For the measurement step, the property is retained through resampling, which adjusts the particle population in response to the new information added by the measurement.

Figure 37.9 shows results for a point-feature problem; here the point features are the centers of tree trunks as observed by an outdoor robot. The dataset used here is known as the Victoria Park dataset [37.80]. Figure 37.9a shows that the path of the vehicle obtained by integrating the vehicle controls, without perception controls, are a poor predictor of location for this vehicle; after 30 min of driving, the estimated position of the vehicle is well over 100 m away from its **GPS** position.

The FastSLAM algorithm has a number of remarkable properties, which may not be intuitive to the untrained eye. First, it solves both the full and the online **SLAM** problems. Each particle has a sample of an entire path (and in fact, conditioning on the entire path is required for its mathematical derivation), but the actual update equation only uses the most recent pose. This makes FastSLAM a filter, similar to the **EKF**. Second, FastSLAM makes it easy to pursue multiple data association hypotheses. It is straightforward to make data association decisions on a per-particle basis, in-

stead of having to adopt the same hypothesis for the entire filter. While we will not give any mathematical justification, we note that the resulting FastSLAM algorithm samples the correct posteriors even for **SLAM** problems with unknown data association – something that neither of the previous two algorithms can claim. And third, FastSLAM can be implemented very efficiently: using advanced tree methods to represent the map estimates, the update can be performed in time logarithmic in the size of the map N , and linear in the number of particles M . These properties, along with the relative ease of implementation, has made FastSLAM a popular choice.

FastSLAM has been extended in great many ways. One important variant is a grid-based version of FastSLAM, in which the Gaussians are replaced by an occupancy grid map [37.81]. This variant is illustrated in Figs. 37.10 and 37.11. Figure 37.10 shows the situation just before closing a large loop. The three different particles each stand for different paths, and they also possess their own local maps. When the loop is closed, importance resampling selects those particles whose maps are most consistent with the measurement. A resulting large-scale map is shown in Fig. 37.11.

Significant extensions of the FastSLAM method can be found in [37.82, 83], whose methods **DP-SLAM** and *ancestry trees* provide efficient tree update methods for grid-based maps. The work in [37.84] provides a way to incorporate new observations into the location sampling process, based on prior work in [37.85].

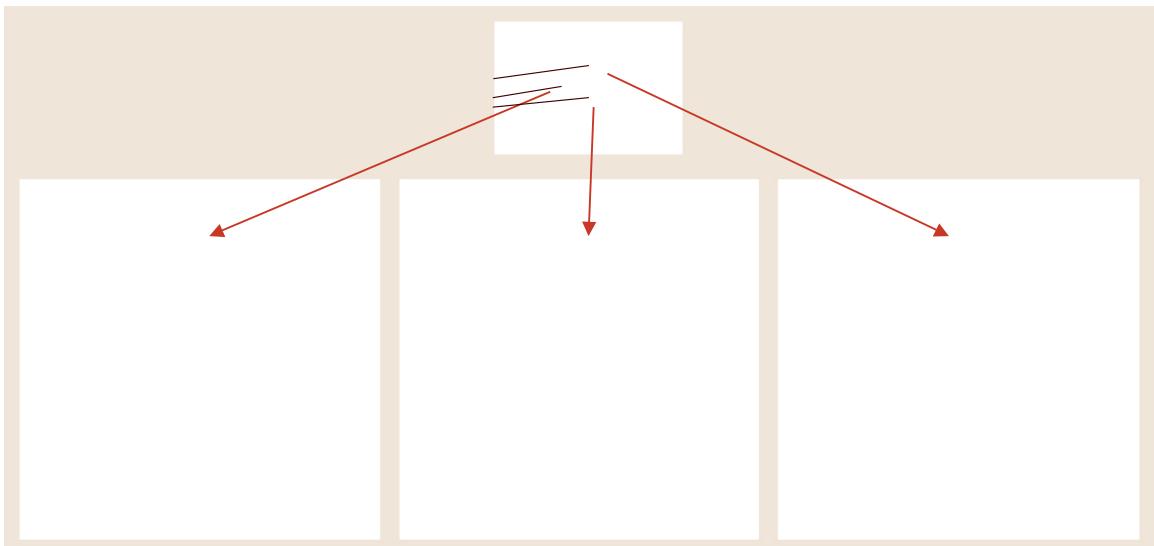


Fig. 37.10 Application of the grid-based variant of the FastSLAM algorithm. Each particle carries its own map, and the importance weights of the particles are computed based on the likelihood of the measurements given the particle's own map

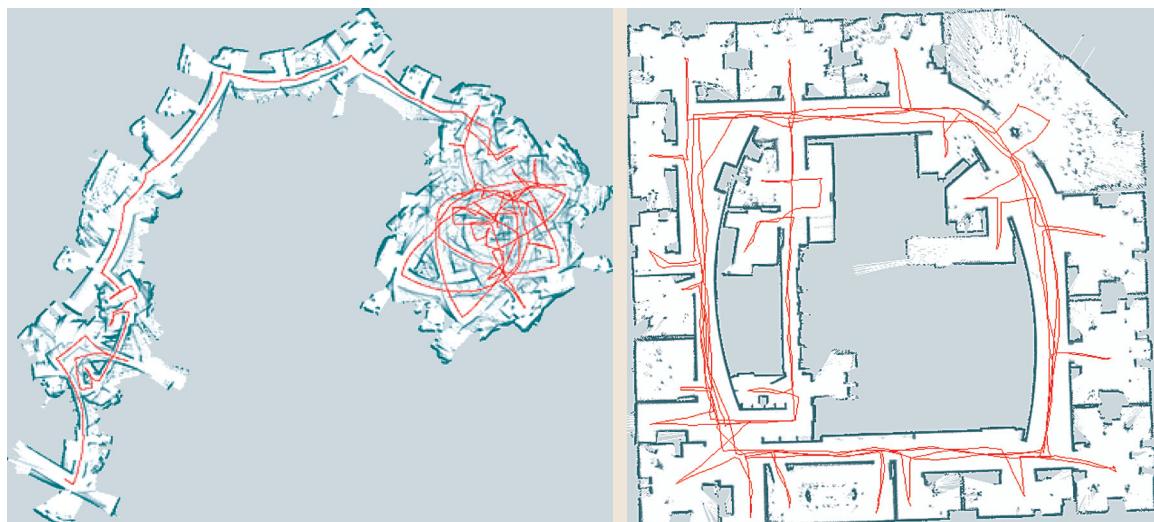


Fig. 37.11 Occupancy grid map generated from laser range data and based on pure odometry (All images courtesy of Dirk Hähnel, University of Freiburg)

37.3.4 Relation of Paradigms

The three paradigms just discussed cover the vast majority of work in the field of **SLAM**. As discussed, **EKF SLAM** comes with a computational hurdle that poses serious scaling limitations. The most promising extensions of **EKF SLAM** are based on

building local submaps; however, in many ways the resulting algorithms resemble the graph-based approach.

Graph-based methods address the full **SLAM** problem, and hence are by nature not online. They draw their intuition from the observation that **SLAM** can be modeled by a sparse graph of soft constraints,



where each constraint either corresponds to a motion or a measurement event. Due to the availability of highly efficient optimization methods for sparse nonlinear optimization problems, graph-based **SLAM** has become the method of choice for building large-scale maps offline. Data association search is quite easily incorporated into the basic mathematical framework, and a number of search techniques exist for finding suitable correspondences. There are also extensions of the graph-based **SLAM** for the online **SLAM** problem. Those tend to remove old robot locations from the graph.

Particle filter methods sidestep some of the issues arising from the natural interfeature correlations in the map, which plagued the **EKF**. By sampling from robot poses, the individual landmarks in the map become

independent, and hence are decorrelated. As a result, FastSLAM can represent the posterior by a sampled robot pose, and many local, independent Gaussians for its landmarks. The particle representation of FastSLAM has a number of advantages. Computationally, FastSLAM can be used as a filter, and its update requires linear-logarithmic time where **EKF** needed quadratic time. Further, FastSLAM can sample over data association, which makes it a prime method for **SLAM** with unknown data association. On the negative side, the number of necessary particles can grow very large, especially for robots seeking to map multiple nested loops. We discussed extensions of FastSLAM that use occupancy grid maps instead of Gaussian landmarks, and showed state-of-the-art examples in large map building.

37.4 Conclusion and Future Challenges

This chapter provided a comprehensive introduction to the **SLAM** problem and its primary solutions. The **SLAM** problem was defined as the problem faced by a mobile platform roaming an unknown environment, and seeking to localize and map its environment at the same time. The chapter discussed three main paradigms in **SLAM**, which are based on the extended Kalman filter, graph-based sparse optimization techniques, and particle filters. It pointed out some of the advantages and disadvantages of those methods. For a more in-depth discussion, the interested reader should refer to a recent textbook covering **SLAM** [37.5].

Interestingly, the field of **SLAM** is still relatively young, and it has made enormous progress within just the past decade. In fact, nearly every method described here has been developed within the past few years. Despite all this progress, there remains a great number of open research issues that warrant future research.

In particular, **SLAM** techniques mostly deal with static environments, yet nearly every actual robot environment is dynamic. Early applications of **SLAM** methods to dynamic environments can be found in [37.86–88]. More work is needed to understand the interaction of moving and nonmoving objects in **SLAM**.

Most **SLAM** work addresses single-robot mapping, yet sometimes one is given a team of robots. Early and highly restrictive work on multirobot **SLAM** can be found in [37.89, 90]. More recent methods

include those in [37.91–93]. Multirobot **SLAM** has benefited greatly from substantial recent interest; nevertheless, the existing methods have not yet matured to a level where they can be used by nonexperts in the field.

One of the primary challenges in **SLAM** is to pursue significant implementations. While the theory of **SLAM** is now quite well developed, **SLAM** has not yet been used extensively in industrial or commercial applications. There exist promising prototypes, including methods for building large-scale three-dimensional (3-D) volumetric maps [37.94–97], detailed underwater reconstruction [37.31, 98, 99], and mapping of abandoned underground mines [37.57]. Nevertheless, the authors feel that more work is needed to mature the technology into industrial-strength applications.

An ultimate goal is to realize the challenge of *persistent* navigation and mapping – the capability for a robot to perform **SLAM** robustly for days, weeks, or months at a time with minimal human supervision, in complex, dynamic environments. Taking the limit as $t \rightarrow \infty$ poses difficult challenges to most current algorithms; in fact, most robot mapping and navigation algorithms are doomed to fail with the passage of time, as errors inevitably accrue. Research is needed to develop techniques that can recover from mistakes and enable robots to deal with changes in the environment and to recover from mistakes, enabling a long-term autonomous existence.

37.5 Suggestions for Further Reading

The following references provide an in-depth tutorial on **SLAM** and much greater depth of coverage on the details of current **SLAM** algorithms:

- H. Durrant-Whyte, T. Bailey: Simultaneous localization and mapping: Part I, *IEEE Robot. Autom. Mag.*, 99–108 (2006)
- T. Bailey, H. Durrant-Whyte: Simultaneous localization and mapping: Part II, *IEEE Robot. Autom. Mag.*, 108–117 (2006)
- S. Thrun, W. Burgard, D. Fox: *Probabilistic Robotics* (MIT Press, Cambridge 2005)

References

- 37.1 K.F. Gauss: *Theoria Motus Corporum Coelestium (Theory of the Motion of the Heavenly Bodies Moving about the Sun in Conic Sections)* (Little, Brown, and Co., Republished in 1857, and by Dover in 1963, 1809)
- 37.2 G. Konecny: *Geoinformation: Remote Sensing, Photogrammetry and Geographical Information Systems* (Taylor Francis, New York 2002)
- 37.3 C. Tomasi, T. Kanade: Shape and motion from image streams under orthography: A factorization method, *Int. J. Comput. Vis.* **9**(2), 137–154 (1992)
- 37.4 S. Soatto, R. Brockett: Optimal structure from motion: Local ambiguities and global estimates, *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)* (Santa Barbara 1998) pp. 282–288
- 37.5 S. Thrun, W. Burgard, D. Fox: *Probabilistic Robotics* (MIT, Cambridge 2005)
- 37.6 H. Durrant-Whyte, T. Bailey: Simultaneous localization and mapping: Part I, *IEEE Robot. Autom. Mag.* (2006) pp. 99–108
- 37.7 T. Bailey, H. Durrant-Whyte: Simultaneous localization and mapping: Part II, *IEEE Robot. Autom. Mag.* (2006) pp. 108–117
- 37.8 P. Cheeseman, P. Smith: On the representation and estimation of spatial uncertainty, *Int. J. Robot.* **5**, 56–68 (1986)
- 37.9 R.C. Smith, P. Cheeseman: On the representation and estimation of spatial uncertainty, *Int. J. Robot. Res.* **5**(4), 56–68 (1986)
- 37.10 R. Smith, M. Self, P. Cheeseman: Estimating uncertain spatial relationships in robotics, *Autonomous Robot Vehicles*, ed. by I.J. Cox, G.T. Wilfong (Springer, Berlin, Heidelberg 1990) pp. 167–193
- 37.11 P. Moutarlier, R. Chatila: An experimental system for incremental environment modeling by an autonomous mobile robot, *1st International Symposium on Experimental Robotics* (Montreal 1989)
- 37.12 P. Moutarlier, R. Chatila: Stochastic multisensory data fusion for mobile robot location and environment modeling, *5th Int. Symposium on Robotics Research* (Tokyo 1989)
- 37.13 A.M. Jazwinsky: *Stochastic Processes and Filtering Theory* (Academic, New York 1970)
- 37.14 R.E. Kalman: A new approach to linear filtering and prediction problems, *Trans. ASME J. Basic Eng.* **82**, 35–45 (1960)
- 37.15 P.S. Maybeck: The Kalman filter: An introduction to concepts. In: *Autonomous Robot Vehicles*, ed. by I.J. Cox, G.T. Wilfong (Springer, Berlin, Heidelberg 1990)
- 37.16 M. Csorba: *Simultaneous Localisation and Map Building*. Ph.D. Thesis (University of Oxford, Oxford 1997)
- 37.17 J. Neira, J.D. Tardós: Data association in stochastic mapping using the joint compatibility test, *IEEE Trans. Robot. Autom.* **17**(6), 890–897 (2001)
- 37.18 J. Neira, J.D. Tardós, J.A. Castellanos: Linear time vehicle relocation in SLAM, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (Taiwan 2003)
- 37.19 T. Bailey: *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments*. Ph.D. Thesis (University of Sydney, Sydney 2002)
- 37.20 G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, M. Csorba: A solution to the simultaneous localisation and map building (SLAM) problem, *IEEE Trans. Robot. Autom.* **17**(3), 229–241 (2001)
- 37.21 G. Dissanayake, S.B. Williams, H. Durrant-Whyte, T. Bailey: Map management for efficient simultaneous localization and mapping (SLAM), *Autonom. Robot.* **12**, 267–286 (2002)
- 37.22 S. Williams, G. Dissanayake, H.F. Durrant-Whyte: Constrained initialization of the simultaneous localization and mapping algorithm, *Proceedings of the Symposium on Field and Service Robotics* (Helsinki 2001)
- 37.23 J.J. Leonard, R.J. Rikoski, P.M. Newman, M. Bosse: Mapping partially observable features from multiple uncertain vantage points, *Int. J. Robot. Res.* **21**(10), 943–975 (2002)
- 37.24 A.J. Davison: Real-Time Simultaneous Localisation and Mapping with a Single Camera, *International Conference on Computer Vision* (Nice 2003) pp. 1403–1410
- 37.25 J.M.M. Montiel, J. Civera, A.J. Davison: Unified inverse depth parametrization for monocular SLAM,



- Proc. of the Robotics Science and Systems Conference (RSS06), Vol.1 (Philadelphia 2006)
- 37.26 M. Bosse, P. Newman, J. Leonard, S. Teller: Simultaneous localization and map building in large-scale cyclic environments using the atlas framework, *Int. J. Robot. Res.* **23**(12), 1113–1139 (2004)
- 37.27 J. Nieto, T. Bailey, E. Nebot: Scan-SLAM: Combining ekf-slam and scan correlation, *Proc. IEEE Int. Conf. Robotics and Automation* (Barcelona 2005)
- 37.28 J. Folkesson, H.I. Christensen: Outdoor exploration and slam using a compressed filter, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (Taiwan 2003) pp. 419–427
- 37.29 J. Guivant, E. Nebot: Optimization of the simultaneous localization and map building algorithm for real time implementation, *IEEE Trans. Robot. Autom.* **17**(3), 242–257 (2001)
- 37.30 J.J. Leonard, H.J.S. Feder: A computationally efficient method for large-scale concurrent mapping and localization, *Proceedings of the Ninth International Symposium on Robotics Research*, ed. by J. Hollerbach, D. Koditschek (Salt Lake City 1999)
- 37.31 S.B. Williams, G. Dissanayake, H. Durrant-Whyte: An efficient approach to the simultaneous localisation and mapping problem, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (Washington 2002) pp. 406–411
- 37.32 J.D. Tardós, J. Neira, P.M. Newman, J.J. Leonard: Robust mapping and localization in indoor environments using sonar data, *Int. J. Robot. Res.* **21**(4), 311–330 (2002)
- 37.33 S. Betgé-Brezetz, R. Chatila, M. Devy: Object-based modelling and localization in natural environments, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (Osaka 1995)
- 37.34 S. Betgé-Brezetz, P. Hébert, R. Chatila, M. Devy: Uncertain map making in natural environments, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (Minneapolis 1996)
- 37.35 J.E. Guivant, E.M. Nebot, J. Nieto, F. Masson: Navigation and mapping in large unstructured environments, *Int. J. Robot. Res.* **23**(4), 449–472 (2004)
- 37.36 J. Nieto, J.E. Guivant, E.M. Nebot: The hybrid metric maps (HYMMs): A novel map representation for dense SLAM, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (New Orleans 2004)
- 37.37 W. Burgard, D. Fox, H. Jans, C. Matenar, S. Thrun: Sonar-based mapping of large-scale mobile robot environments using EM, *Proceedings of the International Conference on Machine Learning* (Bled 1999)
- 37.38 H. Shatkay, L. Kaelbling: Learning topological maps with weak local odometric information, *Proceedings of IJCAI-97* (Nagoya 1997)
- 37.39 S. Thrun, D. Fox, W. Burgard: A probabilistic approach to concurrent mapping and localization for mobile robots, *Machine Learn.* **31**, 29–53 (1998), Also appeared in *Autonomous Robots* 5, 253–271 (joint issue)
- 37.40 A.P. Dempster, A.N. Laird, D.B. Rubin: Maximum likelihood from incomplete data via the EM algorithm, *J. R. Statist. Soc. Ser. B* **39**(1), 1–38 (1977)
- 37.41 H.F. Durrant-Whyte: Uncertain geometry in robotics, *IEEE Trans. Robot. Autom.* **4**(1), 23–31 (1988)
- 37.42 F. Lu, E. Milios: Globally consistent range scan alignment for environment mapping, *Autonom. Robot.* **4**, 333–349 (1997)
- 37.43 F. Dellaert: Square root SAM, *Proceedings of the Robotics Science and Systems Conference*, ed. by S. Thrun, G. Sukhatme, S. Schaal, O. Brock (Cambridge 2005)
- 37.44 T. Duckett, S. Marsland, J. Shapiro: Learning globally consistent maps by relaxation, *Proceedings of the IEEE International Conference on Robotics and Automation* (San Francisco 2000) pp. 3841–3846
- 37.45 T. Duckett, S. Marsland, J. Shapiro: Fast, on-line learning of globally consistent maps, *Auton. Robot.* **12**(3), 287–300 (2002)
- 37.46 J. Folkesson, H.I. Christensen: Graphical SLAM: A self-correcting map, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (New Orleans 2004)
- 37.47 J. Folkesson, H.I. Christensen: Robust SLAM, *Proceedings of the International Symposium on Autonomous Vehicles* (Lisboa 2004)
- 37.48 U. Frese, G. Hirzinger: Simultaneous localization and mapping – a discussion, *Proceedings of the IJCAI Workshop on Reasoning with Uncertainty in Robotics* (Seattle 2001) pp. 17–26
- 37.49 U. Frese, P. Larsson, T. Duckett: A multigrid algorithm for simultaneous localization and mapping, *IEEE Trans. Robot.* **21**(2), 196–207 (2005)
- 37.50 M. Golfarelli, D. Maio, S. Rizzi: Elastic correction of dead-reckoning errors in map building, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Victoria 1998) pp. 905–911
- 37.51 K. Konolige: Large-scale map-making, *Proceedings of the AAAI National Conference on Artificial Intelligence* (San Jose 2004) pp. 457–463
- 37.52 M. Montemerlo, S. Thrun: Large-scale robotic 3-d mapping of urban structures, *Proceedings of the International Symposium on Experimental Robotics (ISER)* (Singapore 2004)
- 37.53 Y. Liu, S. Thrun: Results for outdoor-SLAM using sparse extended information filters, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (Taiwan 2003)

- 37.54 M.A. Fischler, R.C. Bolles: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Commun. ACM* **24**, 381–395 (1981)
- 37.55 D. Hähnel, W. Burgard, B. Wegbreit, S. Thrun: Towards lazy data association in SLAM, *Proceedings of the 11th International Symposium of Robotics Research (ISRR'03)* (Sienna 2003)
- 37.56 B. Kuipers, J. Modayil, P. Beeson, M. MacMahon, F. Savelli: Local metrical and global topological maps in the hybrid spatial semantic hierarchy, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (New Orleans 2004)
- 37.57 S. Thrun, S. Thayer, W. Whittaker, C. Baker, W. Burgard, D. Ferguson, D. Hähnel, M. Montemerlo, A. Morris, Z. Omohundro, C. Reverte, W. Whittaker: Autonomous exploration and mapping of abandoned mines, *IEEE Robot. Autom. Mag.* **11**(4), 79–91 (2004)
- 37.58 A. Elfes: Sonar-based real-world mapping and navigation, *IEEE J. Robot. Autom.* **RA-3**(3), 249–265 (1987)
- 37.59 H.P. Moravec: Sensor fusion in certainty grids for mobile robots, *AI Mag.* **9**(2), 61–74 (1988)
- 37.60 T.M. Cover, J.A. Thomas: *Elements of Information Theory* (Wiley, New York 1991)
- 37.61 P. Newman, J.L.R. Rikoski: Towards constant-time slam on an autonomous underwater vehicle using synthetic aperture sonar, *Proceedings of the International Symposium of Robotics Research* (Sienna 2003)
- 37.62 M.A. Paskin: Thin junction tree filters for simultaneous localization and mapping, *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)* (Acapulco 2003)
- 37.63 S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte, A.Y. Ng: Simultaneous mapping and localization with sparse extended information filters, *Proceedings of the Fifth International Workshop on Algorithmic Foundations of Robotics*, ed. by J.-D. Boissonnat, J. Burdick, K. Goldberg, S. Hutchinson (Nice 2002)
- 37.64 E. Nettleton, S. Thrun, H. Durrant-Whyte: Decentralised slam with low-bandwidth communication for teams of airborne vehicles, *Proceedings of the International Conference on Field and Service Robotics* (Lake Yamanaka 2003)
- 37.65 P. Newman: On the Structure and Solution of the Simultaneous Localisation and Map Building Problem. Ph.D. Thesis (Australian Centre for Field Robotics, University of Sydney, Sydney 2000)
- 37.66 P.M. Newman, H.F. Durrant-Whyte: Geometric projection filter: An efficient solution to the SLAM problem, *Proc. SPIE* **4571** (2001)
- 37.67 S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani, H. Durrant-Whyte: Simultaneous localization and mapping with sparse extended information filters, *Int. J. Robot. Res.* **23**(7–8), 693–716 (2004)
- 37.68 S.B. Williams: Efficient Solutions to Autonomous Mapping and Navigation Problems. Ph.D. Thesis (University of Sydney, Sydney 2001)
- 37.69 N. Metropolis, S. Ulam: The Monte Carlo method, *J. Am. Stat. Assoc.* **44**(247), 335–341 (1949)
- 37.70 D.B. Rubin: Using the SIR algorithm to simulate posterior distributions, *Bayesian Statistics 3*, ed. by M.H. Bernardo, K.M. an DeGroot, D.V. Lindley, A.F.M. Smith (Oxford Univ. Press, Oxford 1988)
- 37.71 A. Doucet: *On sequential simulation-based methods for Bayesian filtering*. Technical Report CUED/F-INFENG/TR 310 (Cambridge University, Cambridge 1998)
- 37.72 G. Kitagawa: Monte Carlo filter and smoother for non-Gaussian nonlinear state space models, *J. Comput. Graph. Statist.* **5**(1), 1–25 (1996)
- 37.73 J. Liu, R. Chen: Sequential monte carlo methods for dynamic systems, *J. Am. Stat. Assoc.* **93**, 1032–1044 (1998)
- 37.74 M. Pitt, N. Shephard: Filtering via simulation: auxiliary particle filter, *J. Am. Stat. Assoc.* **94**, 590–599 (1999)
- 37.75 D. Blackwell: Conditional expectation and unbiased sequential estimation, *Ann. Math. Statist.* **18**, 105–110 (1947)
- 37.76 C.R. Rao: Information and accuracy obtainable in estimation of statistical parameters, *Bull. Calcutta Math. Soc.* **37**, 81–91 (1945)
- 37.77 K. Murphy, S. Russell: Rao–Blackwellized particle filtering for dynamic Bayesian networks. In: *Sequential Monte Carlo Methods in Practice*, ed. by A. Doucet, N. de Freitas, N. Gordon (Springer, Berlin, Heidelberg 2001) pp. 499–516
- 37.78 M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit: FastSLAM: A factored solution to the simultaneous localization and mapping problem, *Proceedings of the AAAI National Conference on Artificial Intelligence* (Edmonton 2002)
- 37.79 J. Pearl: *Probabilistic reasoning in intelligent systems: networks of plausible inference* (Morgan Kaufmann, San Mateo 1988)
- 37.80 J. Guivant, E. Nebot, S. Baiker: Autonomous navigation and map building using laser range sensors in outdoor applications, *J. Robot. Syst.* **17**(10), 565–583 (2000)
- 37.81 D. Hähnel, D. Fox, W. Burgard, S. Thrun: A highly efficient FastSLAM algorithm for generating cyclic maps of large-scale environments from raw laser range measurements, *Proceedings of the Conference on Intelligent Robots and Systems (IROS)* (Las Vegas 2003)
- 37.82 A. Eliazar, R. Parr: DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks, *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)* (Acapulco 2003)



- 37.83 A. Eliazar, R. Parr: DP-SLAM 2.0, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (New Orleans 2004)
- 37.84 M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit: FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges, Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI) (Acapulco 2003)
- 37.85 R. van der Merwe, N. de Freitas, A. Doucet, E. Wan: The unscented particle filter. In: *Adv. in Neural Inform. Process. Syst.* **13** (2001)
- 37.86 D. Hähnel, D. Schulz, W. Burgard: Mobile robot mapping in populated environments, *Autonom. Robot.* **17**(7), 579–598 (2003)
- 37.87 C.-C. Wang, C. Thorpe, S. Thrun: Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (Taiwan 2003)
- 37.88 D.F. Wolf, G.S. Sukhatme: Mobile robot simultaneous localization and mapping in dynamic environments, *Autonom. Robot.* **19**(1), 53–65 (2005)
- 37.89 J.-S. Gutmann, K. Konolige: Incremental mapping of large cyclic environments, Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA) (2000)
- 37.90 E.W. Nettleton, P.W. Gibbens, H.F. Durrant-Whyte: Closed form solutions to the multiple platform simultaneous localisation and map building (slam) problem, *Sensor Fusion: Architectures, Algorithms, and Applications IV*, Vol. 4051, ed. by Bulur V. Dasarathy (Bellingham 2000) pp. 428–437
- 37.91 J. Fenwick, P. Newman, J. Leonard: Collaborative concurrent mapping and localization, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (Washington 2002)
- 37.92 I.M. Rekleitis, G. Dudek, E.E. Milios: Multi-robot collaboration for robust exploration, *Ann. Math. Artif. Intell.* **31**(1–4), 7–40 (2001)
- 37.93 S. Thrun, Y. Liu: Multi-robot SLAM with sparse extended information filters, Proceedings of the 11th International Symposium of Robotics Research (ISRR'03) (Sienna 2003)
- 37.94 C. Frueh, A. Zakhor: Constructing 3d city models by merging ground-based and airborne views, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) (Madison 2003)
- 37.95 M. Devy, C. Parra: 3-d scene modelling and curve-based localization in natural environments, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (Leuven 1998)
- 37.96 L. Iocchi, K. Konolige, M. Bajracharya: Visually realistic mapping of a planar environment with stereo, Proceedings of the 2000 International Symposium on Experimental Robotics (Waikiki 2000)
- 37.97 S. Teller, M. Antone, Z. Bodnar, M. Bosse, S. Coorg, M. Jethwa, N. Master: Calibrated, registered images of an extended urban area, Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR) (Kauai 2001)
- 37.98 R. Eustice, H. Singh, J. Leonard, M. Walter, R. Ballard: Visually navigating the RMS Titanic with SLAM information filters, Proceedings of the Robotics Science and Systems Conference, ed. by S. Thrun, G. Sukhatme, S. Schaal, O. Brock (Cambridge 2005)
- 37.99 R. Rikoski, J. Leonard, P. Newman, H. Schmidt: Trajectory sonar perception in the ligurian sea, Proceedings of the International Symposium on Experimental Robotics (ISER) (Singapore 2004)