Multimedia Contents

46. Simultaneous Localization and Mapping

Cyrill Stachniss, John J. Leonard, Sebastian Thrun

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 main perception 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 the three major paradigms from which many published methods for SLAM are derived: (1) the extended Kalman filter (EKF); (2) particle filtering; and (3) graph optimization. We also review recent work in three-dimensional (3-D) SLAM using visual and red green blue distance-sensors (RGB-D), and close with a discussion of open research problems in robotic mapping.

46.1 **SLAM: Problem Definition** 1154
46.1.1 Mathematical Basis 1154
46.1.2 Example: SLAM in Landmark Worlds 1155
46.1.3 Taxonomy of the SLAM Problem .. 1156
46.2 **The Three Main SLAM Paradigms** 1157
46.2.1 Extended Kalman Filters 1157
46.2.2 Particle Methods 1159
46.2.3 Graph-Based Optimization Techniques..... 1162
46.2.4 Relation of Paradigms..... 1166
46.3 **Visual and RGB-D SLAM** 1166
46.4 **Conclusion and Future Challenges** 1169
Video-References..... 1170
References..... 1171

This chapter provides a comprehensive introduction into one of the key enabling technologies of mobile robot navigation: *simultaneous localization and mapping*, or in short *SLAM*. SLAM addresses the problem of acquiring a spatial map of an 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. It is of great practical importance; if a robust, general-purpose solution to SLAM can be found, then many new applications of mobile robotics will become possible.

While the problem is deceptively easy to state, it presents many challenges, despite significant progress made in this area. At present, we have robust methods for mapping environments that are mainly static, structured, and of limited size. Robustly mapping unstructured, dynamic, and large-scale environments in an online fashion remains largely an open research problem.

The historical roots of methods that can be applied to address the SLAM problem can be traced back to Gauss [46.1], who is largely credited for inventing the least squares method. In the Twentieth Century, a number of fields outside robotics have studied the making of environment models from a moving sensor platform, most notably in *photogrammetry* [46.2–4] and *computer vision* [46.5]. Strongly related problems in these fields are bundle adjustment and structure from motion. SLAM builds on this work, often extending the basic paradigms into more scalable algorithms. Modern SLAM systems often view the estimation problem as solving a sparse graph of constraints and applying

nonlinear optimization to compute the map and the trajectory of the robot. As we strive to enable long-lived autonomous robots, an emerging challenge is to handle massive sensor data streams.

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 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 method. The method chosen by the practitioner will depend on a number of factors, such as the desired map resolution, the update time, and 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 more a detailed treatment of SLAM, we refer the reader to Durrant-Whyte and Bailey [46.6, 7], who provide an in-depth tutorial for SLAM, Grisetti et al. for a tutorial on graph-based SLAM [46.8], and Thrun et al., which dedicates a number of chapters to the topic of SLAM [46.9].

46.1 SLAM: Problem Definition

The SLAM problem is defined as follows: A mobile robot roams an unknown environment, starting at an initial location x_0. Its motion is uncertain, making it gradually more difficult to determine its current pose in global coordinates. As it roams, the robot can sense its environment with a noisy sensor. The SLAM problem is the problem of building a map of the environment while simultaneously determining the robot's position relative to this map given noisy data.

46.1.1 Mathematical Basis

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, comprising its two-dimensional (2-D) coordinate in the plane plus a single rotational value for its orientation. The sequence of locations, or *path*, is then given as

$$X_T = \{x_0, x_1, x_2, \ldots, x_T\} \,. \tag{46.1}$$

Here T is some terminal time (T might be ∞). The initial location x_0 often serves as a point of reference for the estimation algorithm; 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

$$U_T = \{u_1, u_2, u_3 \ldots, u_T\} \tag{46.2}$$

characterizes the relative motion of the robot. For noise-free motion, U_T would be sufficient to recover the poses 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 often 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

$$Z_T = \{z_1, z_2, z_3, \ldots, z_T\} \,. \tag{46.3}$$

Figure 46.1 illustrates the variables involved in the SLAM problem. It shows the sequence of locations and sensor measurements, and the causal relationships between these variables. This diagram represents a *graph-*

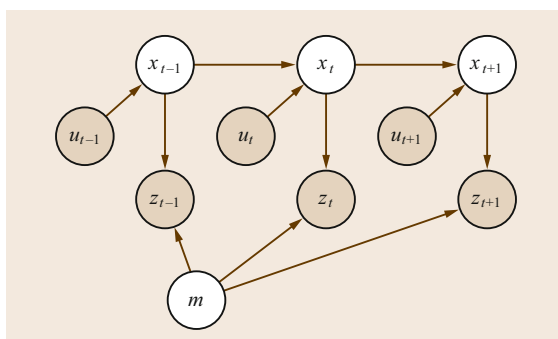

Fig. 46.1 Graphical model of the SLAM problem. Arcs indicate causal relationships, and shaded nodes are directly observable to the robot. In SLAM, the robot seeks to recover the unobservable variables

ical model for SLAM. It is useful in understanding the dependencies in the problem at hand.

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 measurement 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

$$p(X_T, m \mid Z_T, U_T) \; . \tag{46.4}$$

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 full 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

$$p(x_t, m \mid Z_t, U_t) \; . \tag{46.5}$$

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* the 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. 46.1.

In SLAM, it is common to think of those mathematical models as probability distributions: $p(x_t \mid x_{t-1}, u_t)$ characterizes the probability distribution of the location x_t assuming that a robot started at a known location x_{t-1} and measured the odometry data u_t. And likewise, $p(z_t \mid x_t, m)$ 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.

46.1.2 Example: SLAM in Landmark Worlds

One common setting of SLAM involves an assumption that the environment is populated by point-landmarks. When building 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. Determining the identity of the sensed landmarks is also known as the *data association problem*. In practice, it is one of the most difficult problems in SLAM.

To model the above described 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

$$h(x_t, m) \; . \tag{46.6}$$

Computing h is straightforward in our simplified landmark setting; it is a simple exercise in trigonometry. The probabilistic measurement model can be 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, for example,

$$p(z_t \mid x_t, m) = \mathcal{N}(h(x_t, m), \mathbf{Q}_t) \; . \tag{46.7}$$

Here $\mathcal{N}$ denotes the 2-D normal distribution, which is centered at $h(x_t, m)$. The 2-by-2 matrix $\mathbf{Q}_t$ 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

$$g(x_{t-1}, u_t) \; . \tag{46.8}$$

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

$$p(x_t \mid x_{t-1}, u_t) = \mathcal{N}(g(x_{t-1}, u_t), \mathbf{R}_t) \; . \tag{46.9}$$

Here $\mathbf{R}_t$ is a covariance. It is of size 3-by-3, since the location is a three-dimensional 3-D 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 studied, SLAM algorithms are not confined to landmark worlds. But 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 $p(z_t \mid x_t, m)$, and the motion model $p(x_t \mid x_{t-1}, u_t)$. Note that none of those distributions has to be restricted to Gaussian noise as done in the example above.

46.1.3 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 photo-realistic 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, others do not. The ones that do not 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 *data association problem*. It 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 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 Multi-Robot

Most SLAM problems are defined for a single robot platform, although recently the problem of multi-robot exploration has gained in popularity. Multi-robot SLAM problems come in many flavors. In some, robots get to observe each other, 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.

Any-Time and Any-Space

Robots that do all computations onboard have limited resources in memory and computation power. Anytime and any-space SLAM systems are an alternative to traditional methods. They enable the robot to compute a solution given the resource constraints of the system. The more resources available, the better the solution.

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 discussed.

46.2 The Three Main SLAM Paradigms

This section reviews three basic SLAM paradigms, from which most others are derived. The first, known as EKF SLAM, is in robotics historically the earliest but has become less popular due to its limiting computational properties and issues resulting from performing single linearizations only. The second approach uses nonparametric statistical filtering techniques known as particle filters. It is a popular method for online SLAM and provides a perspective on addressing the data association problem in SLAM. The third paradigm is based on graphical representations and successfully applies sparse nonlinear optimization methods to the SLAM problem. It is the main paradigm for solving the full SLAM problem and recently also incremental techniques are available.

46.2.1 Extended Kalman Filters

Historically, the EKF formulation of SLAM is the earliest, and perhaps the most influential, SLAM algorithm. EKF SLAM was introduced in [46.10, 11] and [46.12, 13], 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 extended Kalman filter [46.14, 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.

The EKF algorithm represents the robot estimate by a multivariate Gaussian

$$p(x_t, m \mid Z_t, U_t) = \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \,. \tag{46.10}$$

The high-dimensional vector $\boldsymbol{\mu}_t$ contains the robot's best estimate of its own current location x_t and the location of the features in the environment. In our point-landmark example, the dimension of $\boldsymbol{\mu}_t$ 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 $\boldsymbol{\Sigma}_t$ is the covariance of the robot's assessment of its expected error in the guess $\boldsymbol{\mu}_t$. The matrix $\boldsymbol{\Sigma}_t$ is of size $(3 + 2N) \times (3 + 2N)$ and it is positive semi-definite. In SLAM, this matrix is usually dense. 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 EKF SLAM algorithm is easily derived for our point-landmark example. Suppose, for a moment, 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. In its most basic form and in the absence of any data association problems, EKF SLAM is basically the application of the EKF to the online SLAM problem.

Figure 46.2 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

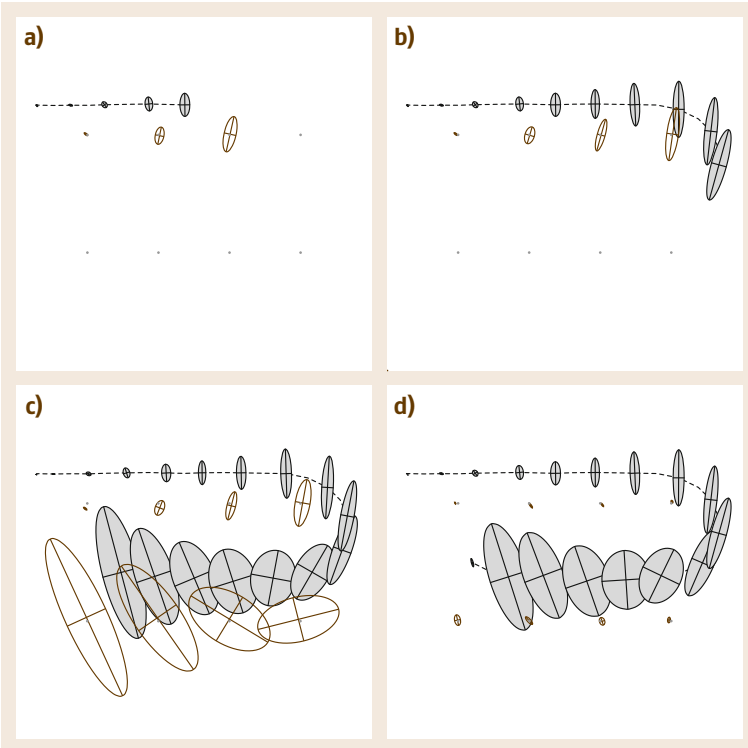

Fig.46.2a–d EKF applied to the online SLAM problem. The robot's path is a *dotted line*, and its estimates of its own position are *shaded ellipses*. Eight distinguishable landmarks of unknown location are shown as *small dots*, and their location estimates are shown as *white ellipses*. In **(a–c)** 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)

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 interesting transition happens in Fig. 46.2d: 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. 46.2d – 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 [46.16]. Information that helps localize the robot is propagated through the map, and as a result improves the localization of other landmarks in the map.

With a few adaptations, EKF SLAM can also be applied in the presence of *uncertain data association*. 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 corresponds most likely 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 [46.17, 18], although distinct features can also be computed from laser data [46.19, 20]. Typical implementations also maintain a *provisional landmark list* and only add landmarks to the internal map when they have been observed sufficiently frequently [46.16, 21]. With an appropriate landmark definition and careful implementation of the data association step, EKF SLAM has been applied successfully in a wide range of environments, using airborne, underwater, indoor, and various other platforms.

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 [46.22] or angle-only [46.23, 24] measurements. The technique has also been utilized using a feature-less 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) [46.25, 26].

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 achieve scalability, for example through submap decomposition [46.27–30]. A related family of approaches [46.31–34] employs the Extended Information Filter, which operates on the inverse of the covariance matrix. A key insight is that whereas the EKF covariance is densely populated, the information matrix is sparse when the full robot trajectory is maintained, leading to the development of efficient algorithms and providing a conceptual link to the pose graph optimization methods described in Sect. 46.2.3.

The issues of consistency and convergence in EKF SLAM have been investigated in [46.35, 36]. Observability-based rules for designing consistent EKF SLAM estimators are presented in [46.37].

46.2.2 Particle Methods

The second principal SLAM paradigm is based on particle filters. Particle filters can be traced back to [46.38], but they have become popular only in the last two decades. Particle filters 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 filter approximates the posterior distribution. Under mild conditions, the particle filter has been shown 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.

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 100 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 [46.39, 40] and is known as Rao–Blackwellization. It has been introduced into the SLAM literature in [46.41], followed by [46.42], who coined the name fastSLAM (fast simultaneous localization and mapping). Let us 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

$$X_t^{[k]}, \boldsymbol{\mu}_{t,1}^{[k]}, \ldots, \boldsymbol{\mu}_{t,N}^{[k]}, \boldsymbol{\Sigma}_{t,1}^{[k]}, \ldots, \boldsymbol{\Sigma}_{t,N}^{[k]} . \tag{46.11}$$

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

- A sample path $X_t^{[k]}$, and
- A set of N 2-D Gaussians with means $\boldsymbol{\mu}_{t,n}^{[k]}$ and variances $\boldsymbol{\Sigma}_{t,n}^{[k]}$), one for each landmark in the environment.

Here n is the index of the landmark (with $1 \leq n \leq N$). From that 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 the starting coordinates, typically $(0, 0, 0)^{\mathrm{T}}$, 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

  $$x_t^{[k]} \approx p(x_t \mid x_{t-1}^{[k]}, u_t) . \tag{46.12}$$

  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 follows

  $$w_t^{[k]} = \mathcal{N}(z_t \mid x_t^{[k]}, \boldsymbol{\mu}_{t,n}^{[k]}, \boldsymbol{\Sigma}_{t,n}^{[k]}) . \tag{46.13}$$

  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 that 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 $\boldsymbol{\mu}_{t,n}^{[k]}$ and covariance $\boldsymbol{\Sigma}_{t,n}^{[k]}$, based on the measurement z_t. This update follows the standard EKF update rules – note that the extended Kalman filters maintained in FastSLAM are, in contrast to EKF SLAM, all low-dimensional (typically 2-D).

This all may sound complex, but FastSLAM is quite easy to implement. Sampling from the motion model usually involves simple kinematic calculations. Computing the importance of a measurement is often straightforward too, especially for Gaussian measurement noise. And updating a low-dimensional particle filter is also not complicated.

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 $p(b \mid a)$ 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 $p(b \mid a)$. This trick is known as Rao–Blackwellization, and it yields better results than sampling from the joint. FastSLAM applies this technique, in that it samples from the path posterior $p(X_t^{[k]} \mid U_t, Z_t)$ and represents the map $p(m \mid X_t^{[k]}, U_t, Z_t)$ in Gaussian form.

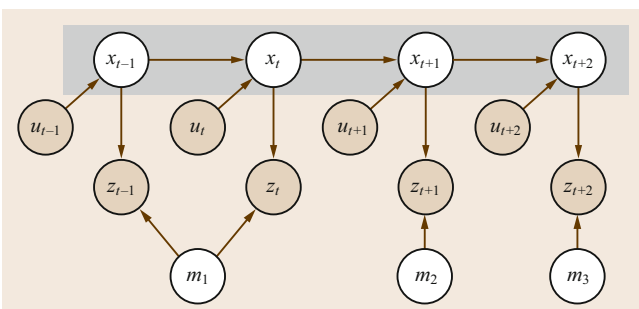

Fig. 46.3 The SLAM problem depicted as Bayes network graph. The robot moves from location x_{t-1} to location x_{t+2}, driven by a sequence of controls. At each location x_t it observes a nearby feature in the map $m = \{m_1, m_2, m_3\}$. This graphical network illustrates that the location variables *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)

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. Fig. 46.3 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. 46.3: we find that if we remove the path variables from Fig. 46.3, then the landmark variables are all disconnected [46.43]. 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.

Figure 46.4 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 [46.44]. Fig. 46.4a shows the path of the vehicle obtained by integrating the vehicle controls, without perception. As can be seen, 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 interesting properties. First, it solves both full and online SLAM problems. Each particle has a sample of an entire path but the actual update equation only uses the most recent pose. This makes FastSLAM a filter. Second, FastSLAM can maintain multiple data association hypotheses. It is straightforward to make data association decisions on a per-particle basis, instead 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 can even deal with unknown data association – something that the extended Kalman filter cannot 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.

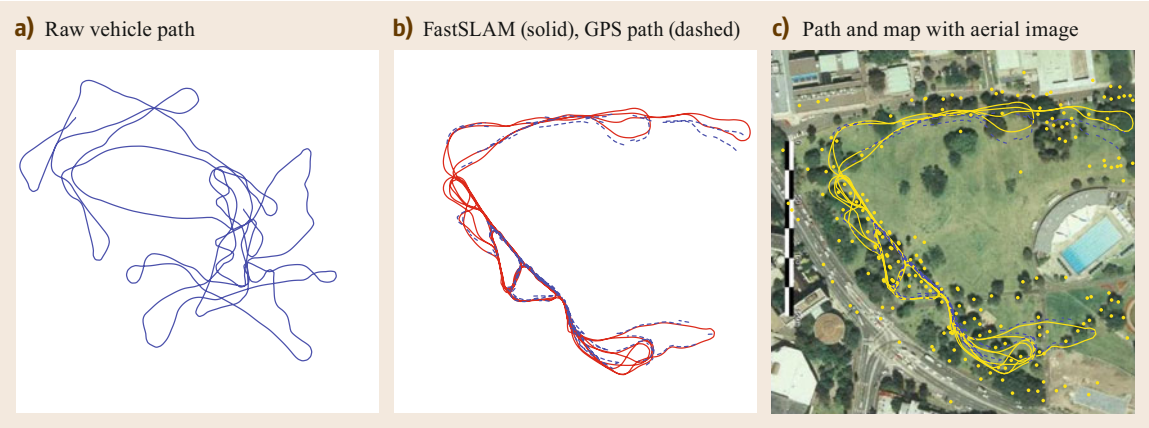

Fig. 46.4 **(a)** Vehicle path predicted by the odometry; **(b)** True path (*dashed line*) and FastSLAM 1.0 path (*solid line*); **(c)** Victoria Park results overlaid on aerial imagery with the GPS path in *blue* (*dashed*), average FastSLAM 1.0 path in *yellow* (*solid*), and estimated features as *yellow dots* (data and aerial image courtesy of José Guivant and Eduardo Nebot, Australian Centre for Field Robotics)

Fig. 46.5 Occupancy grid map generated from laser range data and based on pure odometry (image courtesy of Dirk Hähnel, University of Freiburg)

FastSLAM has been extended in several ways. One set of variants are grid-based versions of FastSLAM, in which the Gaussians used to model point landmarks are replaced by an occupancy grid map [46.45–47]. The variant of [46.46] is illustrated in Fig. 46.5.

Figure 46.6 illustrates a simplified situation with three particles just before closing a large loop. The three different particles each stand for different paths, and they also posses 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. 46.5. Further extensions can be found in [46.48, 49], whose methods are called DP-SLAM and operate on ancestry trees to provide efficient tree update methods for grid-based maps. Related to that, approximations to FastSLAM in which particles share their maps have been proposed [46.50].

The works in [46.45, 47, 51] provide ways to incorporate new observations into the location sampling

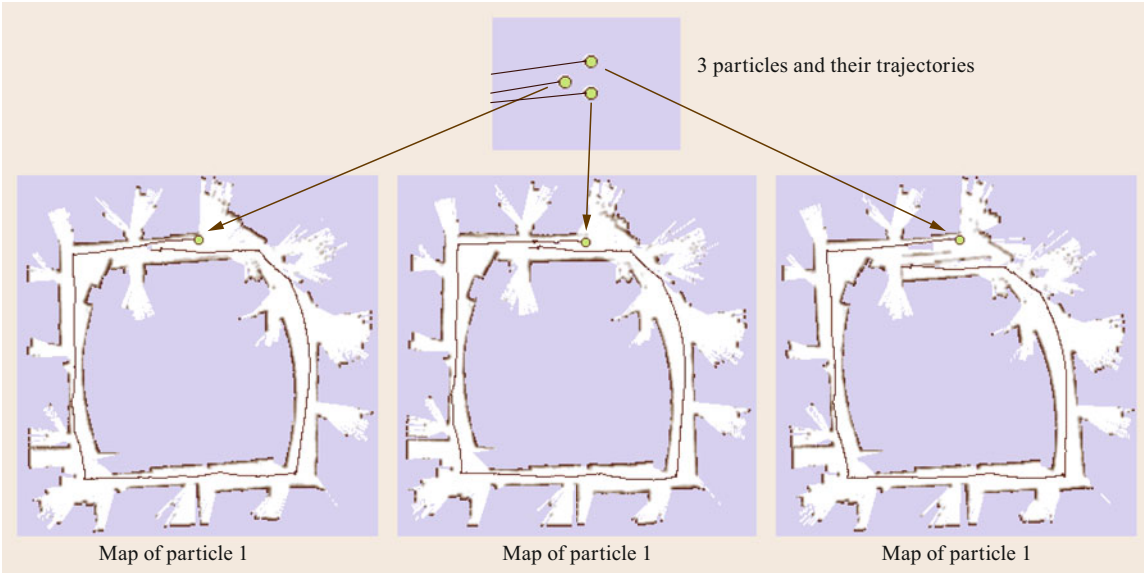

Fig. 46.6 Application of the grid-based variant of the FastSLAM algorithm. Each particle carries its own map and the importance weights of the particles are computed based on the likelihood of the measurements given the particle's own map

process for landmarks and grid maps, based on prior work in [46.52]. This leads to an improved sampling process

$$x_t^{[k]} \approx \frac{p(z_t \mid m_{t-1}^{[k]}, x_t)\, p(x_t \mid x_{t-1}^{[k]}, u_t)}{p(z_t \mid m_{t-1}^{[k]}, x_{t-1}^{[k]}, u_t)} , \qquad (46.14)$$

which incorporates the odometry and the observation at the same time. Using an improved proposal distribution leads to more accurately sampled locations. This in turn leads to more accurate maps and requires a smaller number of particles compared to approaches using the sampling process given in (46.12). This extension makes FastSLAM and especially its grid-based variants robust tools for addressing the SLAM problem.

Finally, there are approaches that aim to overcome the assumption that the observations show Gaussian characteristics. As shown in [46.47], there are several situations in which the model is nonGaussian and also multimodal. A sum of Gaussians model on a per-particle bases, however, can be efficiently considered in the particle filter and it eliminates this problem in practice without introducing additional computational demands.

The so-far developed particle filters-based SLAM systems suffer from two problems. First, the number of samples that are required to compute consistent maps is often set manually by making an *educated guess*. The larger the uncertainty that the filter needs to represent during mapping, the more critical becomes this parameter. Second, nested loops combined with extensive re-visits of previously mapped areas can lead to particle depletion, which in turn may prevent the system from estimating a consistent map. Adaptive resampling strategies [46.45], particles sharing maps [46.50], or filter backup approaches [46.53] improve the situation but cannot eliminate this problem in general.

46.2.3 Graph-Based Optimization Techniques

A third family of algorithms solves the SLAM problem through nonlinear sparse optimization. They draw their intuition from a graphical representation of the SLAM problem and the first working solution in robotics was proposed in [46.54]. The graph-based representation used here is closely related to a series of papers [46.55–64]. We note that most of the earlier techniques are offline and address the full SLAM problem. In more recent years, new incremental versions that effectively re-use the previously computed solution have been proposed such as [46.65–67].

The basic intuition of graph-based SLAM is a 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 edge that represents the information conveyed by the odometry reading u_t. Further edges exist between the nodes that correspond to locations x_t and landmarks m_i, assuming that at time t the robot sensed landmark i. Edges in this graph are

soft constraints. Relaxing these constraints yields the robot's best estimate for the map and the full path.

The construction of the graph is illustrated in Fig. 46.7. Suppose 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. 46.7a.

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. 46.7b. Consecutive application of these two basic steps leads to an graph of increasing size, as illustrated in Fig. 46.7c. Nevertheless this graph is *sparse*, in that each node is only connected to a small number of other nodes (assuming a sensor with limited sensing range). The number of constraints in the graph is (at worst) *linear* in the time elapsed and in the number of nodes in the graph.

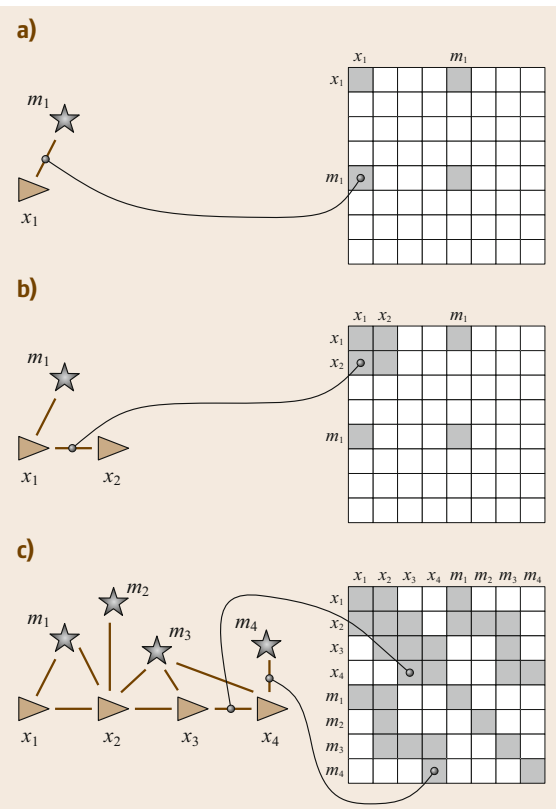

Fig.46.7a–c Illustration of the graph construction. The **(a)** diagram shows the graph, the **(b)** the constraints in matrix form. **(a)** Observation ls landmark m_1. **(b)** Robot motion from x_1 to x_2. **(c)** Several steps later

If we think of the graph as a spring-mass model [46.60], computing the SLAM solution is equivalent to computing the state of minimal energy of this model. To see, we note that the graph corresponds to the log-posterior of the full SLAM problem (46.4)

$$\log p(X_T, m \mid Z_T, U_T) \,. \tag{46.15}$$

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

$$\begin{aligned} &\log p(X_T, m \mid Z_T, U_T) \\ &= \text{const} + \sum_t \log p(x_t \mid x_{t-1}, u_t) \\ &\qquad + \sum_t \log p(z_t \mid x_t, m) \,, \end{aligned} \tag{46.16}$$

assuming independence between the individual observations and odometry readings. 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 edge 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 edge in the graph. The SLAM problem is then simply to find the mode of this equation, i. e.,

$$X_T^*, m^* = \underset{X_T, m}{\operatorname{argmax}} \log p(X_T, m \mid Z_T, U_T) \,. \tag{46.17}$$

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

$$\begin{aligned} &\log p(X_T, m \mid Z_T, U_T) = \text{const} \\ &\quad + \sum_t \underbrace{[x_t - g(x_{t-1}, u_t)]^{\mathrm{T}} R_t^{-1} [x_t - g(x_{t-1}, u_t)]}_{\text{odometry reading}} \\ &\quad + \sum_t \underbrace{[z_t - h(x_t, m)]^{\mathrm{T}} Q_t^{-1} [z_t - h(x_t, m)]}_{\text{feature observation}} \,. \end{aligned} \tag{46.18}$$

This quadratic form yields a sparse system of equations and a number of efficient optimization techniques can be applied. Common choices include direct methods such as sparse Cholesky and QR decomposition, or iterative ones such as gradient descent, conjugate gradient, and others. Most SLAM implementations rely on iteratively linearizing the functions g and h, in which case the objective in (46.18) becomes quadratic in all of its variables.

Extensions to support an effective correction of large-scale graphs are hierarchical methods. One of

the first is the ATLAS framework [46.25], which constructs a two-level hierarchy combining a Kalman filter that operates in the lower level and a global optimization at the higher level. Similar to that, Hierarchical SLAM [46.68] is a technique for using independent local maps, which are merged in case of re-visiting a place. A fully hierarchical approach has been presented in [46.65]. It builds a multilevel pose-graph and employs an incremental, lazy optimization scheme that allows for optimizing large graphs and at the same time can be executed at each step during mapping. An alternative hierarchical approach is [46.69], which recursively partitions the graph into multiple-level submaps using the nested dissection algorithm.

When it comes to computing highly accurate environment reconstructions, approaches that do not only optimize the poses of the robot but also each individual measurement of a dense sensor often provide better results. In the spirit of bundle adjustment [46.4], approaches for laser scanners [46.70] and Kinect cameras [46.71] have been proposed.

The graphical paradigm can be extended to handle the data association problems as we can integrate additional knowledge on data association into (46.18). 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 edges to m_i, or we can add a soft correspondence constraint [46.72] of the form

$$(m_j - m_i)^{\mathrm{T}} \, \mathbf{\Gamma} \, (m_j - m_i) \; . \tag{46.19}$$

Here $\mathbf{\Gamma}$ is 2-by-2 diagonal matrix whose coefficients determine the penalty for *not* assigning identical locations to two landmarks (hence we want $\mathbf{\Gamma}$ 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.

Data association errors typically have a strong impact in the resulting map estimate. Even a small number of wrong data associations is likely to result in inconsistent map estimates. Recently, novel approaches have been proposed that are robust under a certain number of false associations. For example, [46.73, 74] propose an iterative procedure that allows for disabling constraints, an action that is associated with a cost. A generalization of this method introduced in [46.75] formulates [46.74] as a robust cost function also reducing the computational requirements. Such approaches can deal with a significant number of false associations and still provide high-quality maps. Consistency checks for loop closure hypotheses can be found in other approaches as well, both in the front-end [46.76] and in the optimizer [46.77]. There has also been an extension that can deal with multimodal constraints [46.78], proposing a max-mixture representation for maintaining efficiency of the log likelihood optimizing in (46.16). As a result of that, the multimodal extension has only little impact on the runtime and can easily be incorporated in most optimizers. Also robust cost function are used for SLAM, for example pseudo Huber and several alternatives [46.75, 79–81].

Graphical SLAM methods have the advantage that they scale to much higher-dimensional maps than EKF SLAM, exploiting the sparsity of the graph. 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). A further advantage of graph-based methods over the EKF is their ability to constantly relinearize the error function which often leads to better results. 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 (46.18) depends among other things on the number and size of loops in the map. Also the initialization can have a strong impact on the result and a good initial guess can simplify the optimization substantially [46.8, 82, 83].

We note that the graph-based paradigm is very closely linked to information theory, in that the soft constraints constitute the information the robot has on the world (in an information-theoretic sense [46.92]). Most methods in the field are offline and they optimize for the entire robot path. If the robot path is long, the optimization may become cumbersome. Over the last five years, however, incremental optimization techniques have been proposed that aim at providing a sufficient but not necessarily perfect model of the environment at every point in time. This allows a robot to make decisions based on the current model, for example, to determine exploration goals. In this context, incremental variants [46.93, 94] of stochastic gradient descent techniques [46.8, 91] have been proposed that estimate which part of the graph requires re-optimization given new sensor data. Incremental methods [46.66, 79, 95] in the smoothing and mapping framework can be execute at each step of the mapping process and achieve the performance by variable ordering and selective relinearization. As also evaluated in [46.96] for the SLAM problem, variable ordering impacts the performance of the optimization. Others use hierarchical data structures [46.89] and pose-graphs [46.97] combined with

Table 46.1 Recent open-source graph-based SLAM implementations

Name	Comment
Dynamic covariance scaling (DCS) [46.75]	Optimization with a robust cost function for dealing with outliers Integrated into g^2o
g^2o [46.80]	Flexible and easily extendable optimization framework for SLAM Comes with different optimization approaches and error functions Supports external plugins
GTSAM2.1 [46.79]	Flexible optimization framework for SLAM and SFMstructure from motion Implements direct and iterative optimization techniques Implements smoothing and mapping (SAM), iSAM, and iSAM2 Implements bundle adjustment for Visual SLAM and SFM
HOG-Man [46.65]	Incremental optimization approach via hierarchical pose graphs and lazy optimization Requires pose-graphs with full rank constraints
iSAM2 [46.66]	General incremental nonlinear optimization with variable elimination Variable re-ordering to retain sparsity On-demand re-linearization of selected variables
KinFu (KinectFusion reimplemented)	Open source reimplementation of KinectFusion [46.84] within the point cloud library (PCL) Dense and highly accurate reconstruction using a Kinect camera Currently limited to medium sized rooms
MaxMixture [46.78]	Optimization for multimodal constraints and outliers Robust to outliers Plugin for g^2o
Parallel tracking and mapping (PTAM) [46.85]	System for tracking a hand-held monocular camera and observed features Operates on comparably small workspaces
RGBD-SLAM [46.86]	Kinect-frontend for HOG-Man and g^2o Fairly standard combination of SURF matching and RANSAC
ScaViSLAM [46.87]	SLAM system for stereo and Kinect-style cameras Combines local bundle adjustment with sparse global optimization for on-the-fly processing
SLAM6-D [46.88]	SLAM system that operates on point clouds from 3-D laser data Applies iterative closest point algorithm (ICP) and global relaxation
Sparse surface adjustment (SSA) [46.70, 71]	Optimizes robot poses and proximity sensor data jointly Provides smooth surface estimates Assumes a range sensor (e.g., laser scanner, Kinect, or similar)
TreeMap [46.89]	Incremental optimization approach Update in $O(\log N)$ time Provides only a mean estimate
TORO [46.90]	Optimization approach that extends stochastic gradient descent (SGD) [46.91] Robust under bad initial guesses Recovers quickly from large errors but slow convergence at minimum Assumes that constraints have roughly spherical covariance matrices Provides only a mean estimate
Vertigo [46.74]	Switchable constraints for robust optimization Plugin for g^2o

a lazy optimization for on-the-fly mapping [46.65]. As an alternative to global methods, relative optimization approaches [46.98] aim at computing locally consistent geometric maps but only topological maps on the global scale. Hybrid approaches [46.87] seek to combine the best of both worlds.

There also exists a number of cross-overs that manipulate the graph online so as to factor out past robot location variables. The resulting algorithms are filters [46.25, 33, 99, 100], and they tend to be intimately related to information filter methods. 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 [46.27, 28, 101].

Recently, researchers addressed the problem of long-term operation and frequent revisits of already mapped terrain. To avoid densely connected pose-graphs that lead to slow convergence behavior, the robot can switch between SLAM and localization, can merge nodes to avoid a growth of the graph [46.90, 102], or can discard nodes or edges [46.32, 103–105].

Graphical and optimization-based SLAM algorithm are still subject of intense research and the paradigm scales to maps large numbers of nodes [46.25, 55, 57, 59, 63–65, 89, 90, 106, 107]. Arguably, the graph-based

paradigm has generated some the largest SLAM maps ever built. Furthermore, the SLAM community started to release flexible optimization frameworks and SLAM implementations under open source licenses to support further developments and to allow for efficient comparisons, (Table 46.1). Especially the optimization frameworks [46.66, 79, 80] are flexible and powerful state of the art tools for developing graph-based SLAM systems. They can be either used as a *black box* or can be easily extended though plugins.

46.2.4 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 and the linearization may lead to inconsistent maps. 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.

Particle filter methods sidestep some of the issues arising from the natural inter-feature correlations in the map – which hindered 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 offers advantages for SLAM as it allows for computationally efficient updates and for sampling over data associations. On the negative side, the number of necessary particles can grow very large, especially for robots seeking to map multiple nested loops.

Graph-based methods address the full SLAM problem, hence are in the standard formulation not online. They draw their intuition form the fact 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. Recent developments have brought up several graph-based methods for incremental map building that can be executed at every time step during navigation. Data association search can be incorporated into the basic mathematical framework and different approaches that are even robust under wrong data associations are available today.

46.3 Visual and RGB-D SLAM

A popular and important topic in recent years has been Visual SLAM – the challenge of building maps and tracking the robot pose in full 6-DOF using data from cameras [46.108] or RGB-D (Kinect) sensors [46.86, 109]. Visual sensors offer a wealth of information that enables the construction of rich 3-D models of the world. They also enable difficult issues such as loop-closing to be addressed in novel ways using appearance information [46.110]. Visual SLAM is anticipated to be a critical area for future research in perception for robotics, as we seek to develop low-cost systems that are capably of intelligent physical interaction with the world.

Attempting SLAM with monocular, stereo, omnidirectional, or RGB-D cameras raises the level-of-difficulty of many of the SLAM components, such as data association and computational efficiency, described above. A key challenge is robustness. Many visual SLAM applications of interest, such as augmented reality [46.85], entail handheld camera motions, which present greater difficulties for state estimation, in comparison to the motion of a wheeled robot across a flat floor.

Visual navigation and mapping was a key early goal in the mobile robotics community [46.111, 112], but early approaches were hampered by the lack of sufficient computational resources to handle massive video data streams. Early approaches were typically based on extended Kalman filters [46.113–116], but did not compute the full covariance for the feature poses and camera trajectory, resulting in a loss of consistency. Visual SLAM is closely related to the structure from motion (SFM) problem in computer vision [46.4, 5]. Historically, SFM was primarily concerned with offline batch processing, whereas SLAM seeks to achieve a solution for online operation, suitable for closed-loop interaction of a robot or user with its environment. In comparison to laser scanners, cameras provide a fire hydrant of information, making online processing nearly impossible until recent increases in computation have become available.

Davison was an early pioneer in developing complete visual SLAM systems, initially using a real-time active stereo head [46.121] that tracked distinctive visual features with a full covariance EKF approach. Subsequent work developed the first real-time SLAM system that operated with a single freely moving camera as the only data source [46.23, 122]. This system could build sparse, room-size maps of indoor scenes at 30 Hz frame-rate in real-time, a notable historical achievement

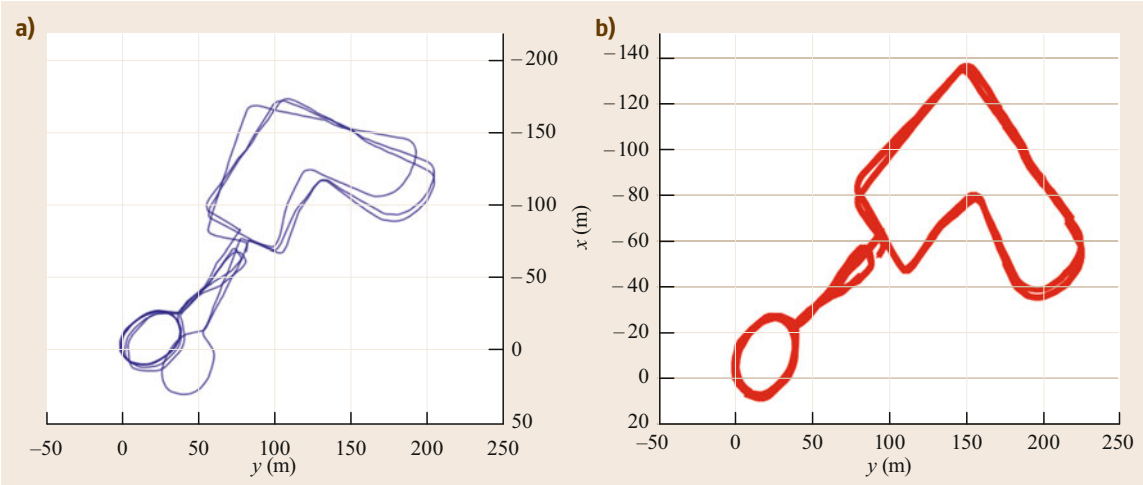

Fig. 46.8 **(a)** A 2-km path and 50 000 frames estimated for the New College Dataset (after [46.117]) using relative bundle adjustment (after [46.98]). **(b)** the relative bundle adjustment solution is easily improved by taking FAB-MAP (after [46.118]) loop-closures into account – this is achieved without global optimization (after [46.119]). Sibley et al. advocate that relative metric accuracy and topological consistency are the requirements for autonomous navigation, and these are better achieved using a relative manifold representation instead of using a conventional single Euclidean representation [46.120]

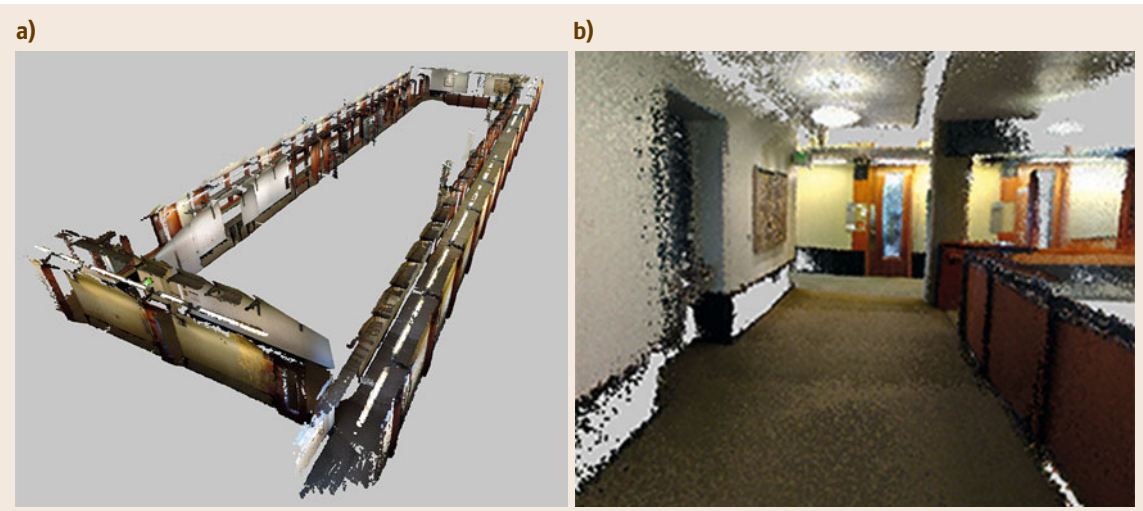

Fig. 46.9 **(a)** 3-D Model and **(b)** close-up view of a corridor environment in the Paul G. Allen building at University of Washington built from Kinect data (after [46.109]; image courtesy of Peter Henry, University of Washington)

in visual SLAM research. A difficulty encountered with initial monocular SLAM [46.23] was coping with the initialization of points that were far away from the camera, due to nonGaussian distributions of such feature locations resulting from poor depth information. This limitation was overcome in [46.24], introducing an inverse depth parameterization for monocular SLAM, a key development for enabling a unified treatment of initialization and tracking of visual features in real-time.

A milestone in creating robust visual SLAM systems was the introduction of keyframes in parallel tracking and mapping (PTAM) [46.85], which separated the tasks of keyframe mapping and localization into parallel threads, improving robustness and performance for online processing. Keyframes are now a mainstream concept for complexity reduction in visual SLAM systems. Related approaches using keyframes include [46.87, 123–126]. The work

in [46.108] analyzes the tradeoffs between filtering and keyframe-based bundle adjustment in visual SLAM, and concluded that keyframe bundle adjustment outperforms filtering, as it provides the most accuracy per unit of computing time.

As pointed out by Davison and other researchers, an appealing aspect of visual SLAM is that camera measurements can provide odometry information, and indeed visual odometry is a key component of modern SLAM systems [46.127]. Here, [46.128] and [46.129] provide an extensive tutorial of techniques for visual odometry, including feature detection, feature matching, outlier rejection, and constraint estimation, and trajectory optimization. Finally, a publicly available visual odometry library [46.130] that is optimized for efficient operation on small unmanned aerial vehicles is available today.

Visual information offers a tremendous source of information for loop closing, not present in the canonical 2-D laser SLAM systems developed in the early 2000s. The work in [46.110] was one of the first to employ techniques for visual object recognition [46.131] to location recognition. More recently FAB-MAP [46.118, 132] has demonstrated appearance-only place recognition at large scale, mapping trajectories with a length of 1000 km. Combining a bag-of-features approach with a probabilistic place model and Chow–Liu tree inference leads to place recognition that is robust against perceptual aliasing while remaining computationally efficient. Other work on place recognition includes [46.133], which combines bag-of-words loop closing with tests of geometrical consistency based on conditional random fields (Fig. 46.8).

The techniques described above have formed the basis for a number of notable large-scale SLAM systems developed in recent years. A 2008 special issue of the IEEE Transactions on Robotics provides a good snapshot of recent state-of-the-art SLAM techniques [46.135]. Other notable recent examples include [46.98, 119, 126, 136–138]. The idea of employing relative bundle adjustment [46.98] to compute a full maximum likelihood solution in an online fashion, even for loop closures, by employing a manifold representation that does not attempt to enforce Euclidean constraints results in maps that can be computed at high frame rate (see also Fig. 46.8). Finally, view-based mapping systems [46.126, 136, 137] aim at large-scale and/or life-long visual mapping based on the pose graph optimization techniques described above in Section 46.2.3.

Several compelling 3-D mapping and localization have been created in recent years with RGB-D (Kinect) sensors. The combination of direct range measurements with dense visual imagery can enable dramatic improvements in mapping and navigation systems for indoor environments. State-of-the-art RGB-D SLAM systems include [46.109] and [46.86]. Figure 46.9 shows examples of the output of these systems.

Other researchers aim at exploiting the surface properties of scanned environments to correct for sen-

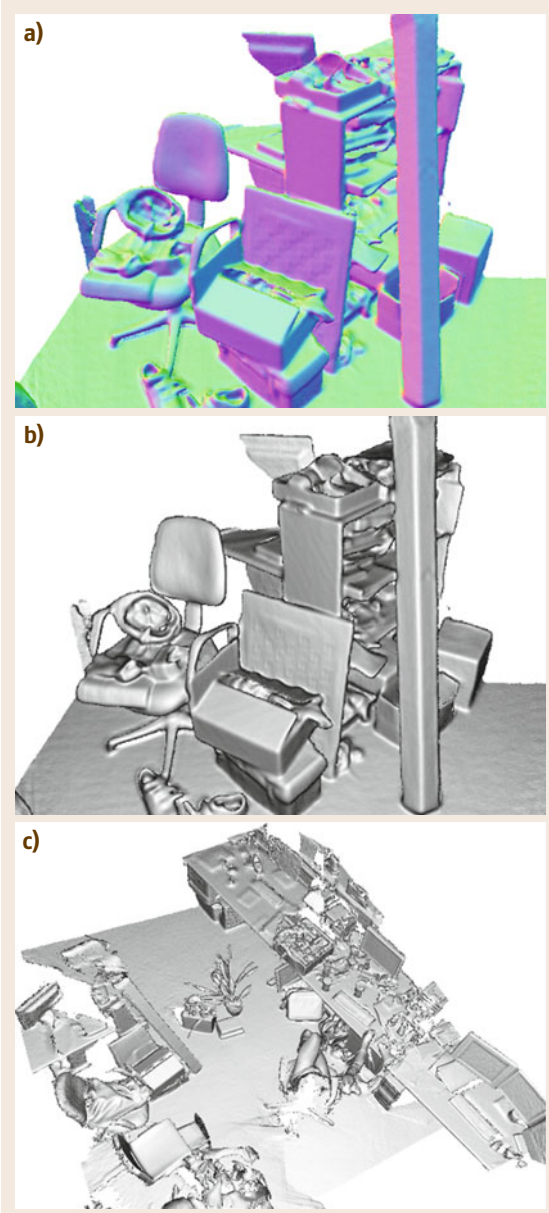

Fig.46.10a–c Results obtained with KinectFusion (after [46.134]). **(a)** A local scene as a normal map and **(b)** as a Phong-shaded rendering. The **(c)** image depicts a larger scene (image courtesy of Richard Newcombe, Imperial College London)

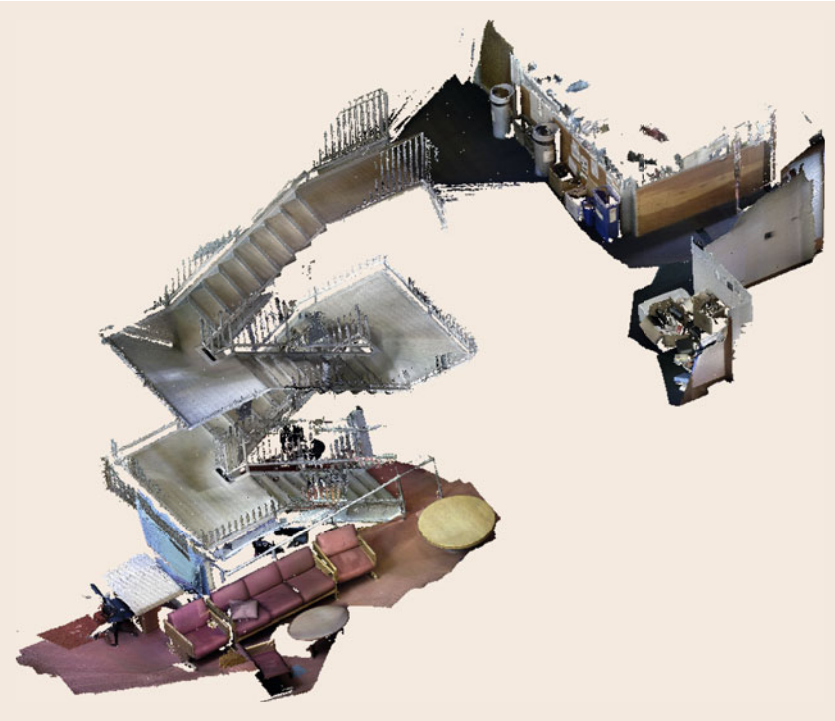

Fig. 46.11 Spatially extended KinectFusion output produced in real-time with Kintinuous (after [46.139])

sor noise of range sensors such as the Kinect [46.71]. They jointly optimize the poses of the sensor and the positions of the surface points measured and iteratively refine the structure of the error function by recomputing the data associations after each optimization, resulting in accurate smooth models of the environment.

An emerging area for future research is the development of fully dense processing methods that exploit recent advances in commodity graphical processing unit (GPU) technology. Kinect-based dense tracking and mapping, a fully-dense method for small-scale visual tracking and reconstruction is described in [46.140]. Dense modeling and tracking are achieved via highly parallelized operations on commodity GPU hardware to yield a system that outperforms previous methods such as PTAM for challenging camera trajectories. Dense methods offer an interesting perspective from which to address long-standing problems, such as visual odometry, from a fresh perspective, without requiring explicit feature detection and matching [46.141]. KinectFusion [46.84, 134] is a dense modeling system that tracks the 3-D pose of a handheld Kinect while concurrently reconstructing high-quality scene 3-D models in real-time. See Fig. 46.10 for an example. KinectFusion has been applied to spatially extended environments in [46.139, 142]. An example is shown in Fig. 46.11.

46.4 Conclusion and Future Challenges

This chapter has provided an introduction into SLAM, which is defined as the problem faced by a mobile platform roaming an unknown environment, and seeking to localize itself while concurrently building a map of the environment. The chapter discussed three main paradigms in SLAM, which are based on the extended Kalman filter, particle filters, and graph-based sparse optimization techniques, and then described recent progress in Visual/Kinect SLAM.

The following references provide an in-depth tutorial on SLAM and much greater depth of coverage on the details of popular SLAM algorithms. Furthermore, several implementations of popular SLAM systems, including most of the approaches listed in Table 46.1, can be found in online resources such as http://www.openslam.org or in the references [46.6, 9, 21, 62].

The considerable progress in SLAM in the past decade is beyond doubt. The core state estimation at

the heart of SLAM is now quite well understood, and a number of impressive implementations have been developed, including several widely used open source software implementations and some commercial projects. None-the-less, a number of open research challenges remain for the general problem of robotic mapping in complex and dynamic environments over extended periods of time, including robots sharing, extending, and revising previously built models, efficient failure recovery, zero user intervention, and operation on resource-constrained systems. Another exciting area for the future is the further development of fully dense visual mapping systems exploiting the latest advances in GPU hardware development.

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 and dynamic environments. Taking the limit as $t \to \infty$ 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. Despite recent encouraging solutions [46.102, 105], more research is needed for techniques that can recover from mistakes and enable robots to deal with changes in the environment and enabling a long-term autonomous existence.

Video-References

VIDEO 439 Deformation-based loop closure for Dense RGB-D SLAM
available from http://handbookofrobotics.org/view-chapter/46/videodetails/439
VIDEO 440 Large-scale SLAM using the Atlas framework
available from http://handbookofrobotics.org/view-chapter/46/videodetails/440
VIDEO 441 Graph-based SLAM
available from http://handbookofrobotics.org/view-chapter/46/videodetails/441
VIDEO 442 Graph-based SLAM
available from http://handbookofrobotics.org/view-chapter/46/videodetails/442
VIDEO 443 Graph-based SLAM
available from http://handbookofrobotics.org/view-chapter/46/videodetails/443
VIDEO 444 Graph-based SLAM
available from http://handbookofrobotics.org/view-chapter/46/videodetails/444
VIDEO 445 Graph-based SLAM
available from http://handbookofrobotics.org/view-chapter/46/videodetails/445
VIDEO 446 Graph-based SLAM using TORO
available from http://handbookofrobotics.org/view-chapter/46/videodetails/446
VIDEO 447 Sparse pose adjustment
available from http://handbookofrobotics.org/view-chapter/46/videodetails/447
VIDEO 449 Pose graph compression for laser-based SLAM
available from http://handbookofrobotics.org/view-chapter/46/videodetails/449
VIDEO 450 Pose graph compression for laser-based SLAM
available from http://handbookofrobotics.org/view-chapter/46/videodetails/450
VIDEO 451 Pose graph compression for laser-based SLAM
available from http://handbookofrobotics.org/view-chapter/46/videodetails/451
VIDEO 452 DTAM: Dense tracking and mapping in real-time
available from http://handbookofrobotics.org/view-chapter/46/videodetails/452
VIDEO 453 MonoSLAM: Real-time single camera SLAM
available from http://handbookofrobotics.org/view-chapter/46/videodetails/453
VIDEO 454 SLAM++: Simultaneous localisation and mapping at the level of objects
available from http://handbookofrobotics.org/view-chapter/46/videodetails/454
VIDEO 455 Extended Kalman filter SLAM
available from http://handbookofrobotics.org/view-chapter/46/videodetails/455

References

46.1 C.F. Gauss: *Theoria Motus Corporum Coelestium (Theory of the Motion of the Heavenly Bodies Moving about the Sun in Conic Sections)* (Perthes and Bessen, Hamburg 1809), Republished in 1857 and by Dover in 1963

46.2 D.C. Brown: The bundle adjustment – Progress and prospects, Int. Arch. Photogramm. **21**(3), 3:3–3:35 (1976)

46.3 G. Konecny: *Geoinformation: Remote Sensing, Photogrammetry and Geographical Information Systems* (Taylor Francis, London 2002)

46.4 B. Triggs, P. McLauchlan, R. Hartley, A. Fitzgibbon: Bundle adjustment – A modern synthesis, Lect. Notes Comput. Sci. **62**, 298–372 (2000)

46.5 R. Hartley, A. Zisserman: *Multiple View Geometry in Computer Vision* (Cambridge Univ. Press, Cambridge 2003)

46.6 T. Bailey, H.F. Durrant-Whyte: Simultaneous localisation and mapping (SLAM): Part II, Robotics Autom. Mag. **13**(3), 108–117 (2006)

46.7 H.F. Durrant-Whyte, T. Bailey: Simultaneous localisation and mapping (SLAM): Part I, Robotics Autom. Mag. **13**(2), 99–110 (2006)

46.8 G. Grisetti, C. Stachniss, W. Burgard: Nonlinear constraint network optimization for efficient map learning, IEEE Trans. Intell. Transp. Syst. **10**(3), 428–439 (2009)

46.9 S. Thrun, W. Burgard, D. Fox: *Probabilistic Robotics* (MIT Press, Cambridge, 2005)

46.10 R. Smith, M. Self, P. Cheeseman: A stochastic map for uncertain spatial relationships, Proc. Int. Symp. Robotics Res. (ISRR) (MIT Press, Cambridge 1988) pp. 467–474

46.11 R. Smith, M. Self, P. Cheeseman: Estimating uncertain spatial relationships in robotics. In: *Autonomous Robot Vehicles*, ed. by I.J. Cox, G.T. Wilfong (Springer Verlag, Berlin, Heidelberg 1990) pp. 167–193

46.12 P. Moutarlier, R. Chatila: Stochastic multisensory data fusion for mobile robot location and environment modeling, 5th Int. Symp. Robotics Res. (ISRR) (1989) pp. 207–216

46.13 P. Moutarlier, R. Chatila: An experimental system for incremental environment modeling by an autonomous mobile robot, 1st Int. Sym. Exp. Robotics (ISER) (1990)

46.14 R. Kalman: A new approach to linear filtering and prediction problems, J. Fluids **82**, 35–45 (1960)

46.15 A.M. Jazwinsky: *Stochastic Processes and Filtering Theory* (Academic, New York 1970)

46.16 M.G. Dissanayake, P.M. Newman, S. Clark, H.F. Durrant-Whyte, M. Csorba: A solution to the simultaneous localization and map building (SLAM) Problem, IEEE Trans. Robotics Autom. **17**(3), 229–241 (2001)

46.17 J. Neira, J. Tardos, J. Castellanos: Linear time vehicle relocation in SLAM, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2003) pp. 427–433

46.18 J. Neira, J.D. Tardos: Data association in stochastic mapping using the joint compatibility test, IEEE Trans. Robotics Autom. **17**(6), 890–897 (2001)

46.19 G.D. Tipaldi, M. Braun, K.O. Arras: Flirt: interest regions for 2D range data with applications to robot navigation, Proc. Int. Symp. Exp. Robotics (ISER) (2010)

46.20 G.D. Tipaldi, L. Spinello, W. Burgard: Geometrical flirt phrases for large scale place recognition in 2D range data, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2013)

46.21 T. Bailey: Mobile Robot Localisation and Mapping in Extensive Outdoor Environments, Ph.D. Thesis (Univ. of Sydney, Sydney 2002)

46.22 J.J. Leonard, R.R. Rikoski, P.M. Newman, M. Bosse: Mapping partially observable features from multiple uncertain vantage points, Int. J. Robotics Res. **21**(10), 943–975 (2002)

46.23 A.J. Davison: Real-time simultaneous localisation and mapping with a single camera, Proc. IEEE 9th Int. Conf. Comput. Vis. (2003) pp. 1403–1410

46.24 J.M.M. Montiel, J. Civera, A.J. Davison: Unified inverse depth parametrization for monocular SLAM, Robotics Sci. Syst., Vol. 1 (2006)

46.25 M. Bosse, P.M. Newman, J. Leonard, S. Teller: Simultaneous localization and map building in large-scale cyclic environments using the Atlas Framework, Int. J. Robotics Res. **23**(12), 1113–1139 (2004)

46.26 J. Nieto, T. Bailey, E. Nebot: Scan-SLAM: Combining EKF-SLAM and scan correlation, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2005)

46.27 J. Guivant, E. Nebot: Optimization of the simultaneous localization and map building algorithm for real time implementation, IEEE Trans. Robotics. Autom. **17**(3), 242–257 (2001)

46.28 J.J. Leonard, H. Feder: A computationally efficient method for large-scale concurrent mapping and localization, Proc. 9th Int. Symp. Robotics Res. (ISRR), ed. by J. Hollerbach, D. Koditschek (1999) pp. 169–176

46.29 J.D. Tardós, J. Neira, P.M. Newman, J.J. Leonard: Robust mapping and localization in indoor environments using sonar data, Int. J. Robotics Res. **21**(4), 311–330 (2002)

46.30 S.B. Williams, G. Dissanayake, H.F. Durrant-Whyte: Towards multi-vehicle simultaneous localisation and mapping, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2002) pp. 2743–2748

46.31 R.M. Eustice, H. Singh, J.J. Leonard: Exactly sparse delayed-state filters for view-based SLAM, IEEE Trans. Robotics **22**(6), 1100–1114 (2006)

46.32 V. Ila, J.M. Porta, J. Andrade-Cetto: Information-based compact pose SLAM, IEEE Trans. Robotics **26**(1), 78–93 (2010)

46.33 S. Thrun, D. Koller, Z. Ghahramani, H.F. Durrant-Whyte, A.Y. Ng: Simultaneous mapping and localization with sparse extended information filters, Proc.5th Int. Workshop Algorithmic Found. Robotics, ed. by J.-D. Boissonnat, J. Burdick, K. Goldberg, S. Hutchinson (2002)

46.34 M.R. Walter, R.M. Eustice, J.J. Leonard: Exactly sparse extended information filters for feature-based SLAM, Int. J. Robotics Res. **26**(4), 335–359 (2007)

46.35 T. Bailey, J. Nieto, J. Guivant, M. Stevens, E. Nebot: Consistency of the EKF-SLAM algorithm, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2006) pp. 3562–3568

46.36 S. Huang, G. Dissanayake: Convergence and consistency analysis for extended Kalman filter based SLAM, IEEE Trans. Robotics **23**(5), 1036–1049 (2007)

46.37 G.P. Huang, A.I. Mourikis, S.I. Roumeliotis: Observability-based Rules for Designing Consistent EKF SLAM Estimators, Int. J. Robotics Res. **29**, 502–528 (2010)

46.38 N. Metropolis, S. Ulam: The Monte Carlo method, J. Am. Stat. Assoc. **44**(247), 335–341 (1949)

46.39 D. Blackwell: Conditional expectation and unbiased sequential estimation, Ann. Math. Stat. **18**, 105–110 (1947)

46.40 C.R. Rao: Information and accuracy obtainable in estimation of statistical parameters, Bull. Calcutta Math. Soc. **37**(3), 81–91 (1945)

46.41 K. Murphy, S. Russel: Rao–Blackwellized particle filtering for dynamic Bayesian networks. In: *Sequential Monte Carlo Methods in Practice*, ed. by A. Docout, N. de Freitas, N. Gordon (Springer, Berlin 2001) pp. 499–516

46.42 M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit: FastSLAM: A factored solution to the simultaneous localization and mapping problem, Proc. AAAI Natl. Conf. Artif. Intell. (2002)

46.43 J. Pearl: *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference* (Morgan Kaufmann, San Mateo 1988)

46.44 J. Guivant, E. Nebot, S. Baiker: Autonomous navigation and map building using laser range sensors in outdoor applications, J. Robotics Syst. **17**(10), 565–583 (2000)

46.45 G. Grisetti, C. Stachniss, W. Burgard: Improved techniques for grid mapping with Rao-Blackwellized particle filters, IEEE Trans. Robotics **23**, 34–46 (2007)

46.46 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, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2003)

46.47 C. Stachniss, G. Grisetti, W. Burgard, N. Roy: Evaluation of gaussian proposal distributions for mapping with rao-blackwellized particle filters, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2007)

46.48 A. Eliazar, R. Parr: DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks, Proc. 16th Int. Jt. Conf. Artif. Intell. (IJCAI) (2003) pp. 1135–1142

46.49 A. Eliazar, R. Parr: DP-SLAM 2.0, Proc. IEEE Int. Conf. Robotics Autom. (ICRA), Vol. 2 (2004) pp. 1314–1320

46.50 G. Grisetti, G.D. Tipaldi, C. Stachniss, W. Burgard, D. Nardi: Fast and accurate SLAM with Rao-Blackwellized particle filters, J. Robotics Auton. Syst. **55**(1), 30–38 (2007)

46.51 D. Roller, M. Montemerlo, S. Thrun, B. Wegbreit: FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges, Int. Jt. Conf. Artif. Intell. (IICAI) (Morgan Kaufmann, San Francisco 2003) pp. 1151–1156

46.52 R. van der Merwe, N. de Freitas, A. Doucet, E. Wan: The unscented particle filter, Proc. Adv. Neural Inform. Process. Syst. Conf. (2000) pp. 584–590

46.53 C. Stachniss, G. Grisetti, W. Burgard: Recovering particle diversity in a Rao-Blackwellized particle filter for SLAM after actively closing loops, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2005) pp. 655–660

46.54 F. Lu, E. Milios: Globally consistent range scan alignment for environmental mapping, Auton. Robots **4**, 333–349 (1997)

46.55 F. Dellaert: Square root SAM, Robotics Sci. Syst., ed. by S. Thrun, G. Sukhatme, S. Schaal, O. Brock (MIT Press, Cambridge 2005)

46.56 T. Duckett, S. Marsland, J. Shapiro: Learning globally consistent maps by relaxation, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2000) pp. 3841–3846

46.57 T. Duckett, S. Marsland, J. Shapiro: Fast, online learning of globally consistent maps, Auton. Robots **12**(3), 287–300 (2002)

46.58 J. Folkesson, H.I. Christensen: Graphical SLAM: A self-correcting map, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2004) pp. 383–390

46.59 U. Frese, G. Hirzinger: Simultaneous localization and mapping – A discussion, Proc. IJCAI Workshop Reason. Uncertain. Robotics (2001) pp. 17–26

46.60 M. Golfarelli, D. Maio, S. Rizzi: Elastic correction of dead-reckoning errors in map building, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (1998) pp. 905–911

46.61 J. Gutmann, K. Konolige: Incremental mapping of large cyclic environments, Proc. IEEE Int. Symp. Comput. Intell. Robotics Autom. (CIRA) (2000) pp. 318–325

46.62 G. Grisetti, R. Kümmerle, C. Stachniss, W. Burgard: A Tutorial on Graph-based SLAM, IEEE Trans. Intell. Transp. Syst. Mag. **2**, 31–43 (2010)

46.63 K. Konolige: Large-scale map-making, Proc. AAAI Natl. Conf. Artif. Intell. (MIT Press, Cambridge 2004) pp. 457–463

46.64 M. Montemerlo, S. Thrun: Large-scale robotic 3-D mapping of urban structures, Springer Tract. Adv. Robotics **21**, 141–150 (2005)

46.65 G. Grisetti, R. Kümmerle, C. Stachniss, U. Frese, C. Hertzberg: Hierarchical optimization on manifolds for online 2D and 3D mapping, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2010)

46.66 M. Kaess, H. Johannsson, R. Roberts, V. Ila, J.J. Leonard, F. Dellaert: iSAM2: Incremental smoothing and mapping using the Bayes tree, Int. J. Robotics Res. **31**, 217–236 (2012)

46.67 M. Kaess, A. Ranganathan, F. Dellaert: iSAM: Incremental Smoothing and Mapping, IEEE Trans. Robotics **24**(6), 1365–1378 (2008)

46.68 C. Estrada, J. Neira, J.D. Tardós: Hierarchical SLAM: Real-time accurate mapping of large environments, IEEE Trans. Robotics **21**(4), 588–596 (2005)

46.69 K. Ni, F. Dellaert: Multi-level submap based SLAM using nested dissection, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2010)

46.70 M. Ruhnke, R. Kümmerle, G. Grisetti, W. Burgard: Highly accurate maximum likelihood laser mapping by jointly optimizing laser points and robot poses, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2011)

46.71 M. Ruhnke, R. Kümmerle, G. Grisetti, W. Burgard: Highly accurate 3D surface models by sparse surface adjustment, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2012)

46.72 Y. Liu, S. Thrun: Results for outdoor-SLAM using sparse extended information filters, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2003)

46.73 N. Sünderhauf, P. Protzel: BRIEF-Gist – Closing the loop by simple means, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2011) pp. 1234–1241

46.74 N. Sünderhauf, P. Protzel: Switchable constraints for robust pose graph SLAM, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2012)

46.75 P. Agarwal, G.D. Tipaldi, L. Spinello, C. Stachniss, W. Burgard: Robust map optimization using dynamic covariance scaling, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2013)

46.76 E. Olson: Recognizing places using spectrally clustered local matches, J. Robotics Auton. Syst. **57**(12), 1157–1172 (2009)

46.77 Y. Latif, C. Cadena Lerma, J. Neira: Robust loop closing over time, Robotics Sci. Syst. (2012)

46.78 E. Olson, P. Agarwal: Inference on networks of mixtures for robust robot mapping, Robotics Sci. Syst. (2012)

46.79 F. Dellaert: *Factor graphs and GTSAM: A hands-on introduction*, Tech. Rep. GT-RIM-CP & R-2012-002 (Georgia Tech, Atlanta 2012)

46.80 R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, W. Burgard: G^2o: A general framework for graph optimization, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2011)

46.81 D.M. Rosen, M. Kaess, J.J. Leonard: An incremental trust-region method for robust online sparse least-squares estimation, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2012) pp. 1262–1269

46.82 L. Carlone, R. Aragues, J. Castellanos, B. Bona: A linear approximation for graph-based simultaneous localization and mapping, Robotics Sci. Syst. (2011)

46.83 G. Grisetti, R. Kümmerle, K. Ni: Robust optimization of factor graphs by using condensed measurements, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2012)

46.84 S. Izadi, R.A. Newcombe, D. Kim, O. Hilliges, D. Molyneaux, S. Hodges, P. Kohli, J. Shotton, A.J. Davison, A. Fitzgibbon: Kinectfusion: Real-time dynamic 3D surface reconstruction and interaction, ACM SIGGRAPH Talks (2011) p. 23

46.85 G. Klein, D. Murray: Parallel tracking and mapping for small AR workspaces, IEEE ACM Int. Symp. Mixed Augment. Real. (ISMAR) (2007) pp. 225–234

46.86 F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, W. Burgard: An evaluation of the RGB-D SLAM system, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2012)

46.87 H. Strasdat, A.J. Davison, J.M.M. Montiel, K. Konolige: Double window optimisation for constant time visual SLAM, Int. Conf. Computer Vis. (ICCV) (2011)

46.88 A. Nüchter: 3D robot mapping, Springer Tract. Adv. Robotics **52** (2009)

46.89 U. Frese: Treemap: An $O(\log n)$ algorithm for indoor simultaneous localization and mapping, Auton. Robots **21**(2), 103–122 (2006)

46.90 G. Grisetti, C. Stachniss, S. Grzonka, W. Burgard: A tree parameterization for efficiently computing maximum likelihood maps using gradient descent, Robotics Sci. Syst. (2007)

46.91 E. Olson, J.J. Leonard, S. Teller: Fast iterative alignment of pose graphs with poor initial estimates, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2006) pp. 2262–2269

46.92 T.M. Cover, J.A. Thomas: *Elements of Information Theory* (Wiley, New York 1991)

46.93 E. Olson, J.J. Leonard, S. Teller: Spatially-adaptive learning rates for online incremental SLAM, Robotics Sci. Syst. (2007)

46.94 G. Grisetti, D. Lordi Rizzini, C. Stachniss, E. Olson, W. Burgard: Online constraint network optimization for efficient maximum likelihood map learning, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2008)

46.95 M. Kaess, A. Ranganathan, F. Dellaert: Fast incremental square root information smoothing, Int. Jt. Conf. Artif. Intell. (ISCAI) (2007)

46.96 P. Agarwal, E. Olson: Evaluating variable reordering strategies for SLAM, Proc. IEEE/RSJ Int. Conf. Intel Robots Syst. (IROS) (2012)

46.97 K. Ni, D. Steedly, F. Dellaert: Tectonic SAM: exact, out-of-core, submap-based SLAM, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2007) pp. 1678–1685

46.98 G. Sibley, C. Mei, I. Reid, P. Newman: Adaptive relative bundle adjustment, Robotics Sci. Syst. (2009)

46.99 P.M. Newman, J.J. Leonard, R. Rikoski: Towards constant-time SLAM on an autonomous underwater vehicle using synthetic aperture sonar, 11th Int. Symp. Robotics Res. (2003)

46.100 M.A. Paskin: Thin junction tree filters for simultaneous localization and mapping, Int. Jt. Conf. Artif. Intell. (IJCAI) (Morgan Kaufmann, New York 2003) pp. 1157–1164

46.101 S.B. Williams: Efficient Solutions to Autonomous Mapping and Navigation Problems, Ph.D. Thesis (ACFR Univ. Sydney, Sydney 2001)

46.102 H. Johannsson, M. Kaess, M.F. Fallon, J.J. Leonard: Temporally scalable visual SLAM using a reduced pose graph, RSS Workshop Long-term Oper. Auton. Robotic Syst. Chang. Environ. (2012)

46.103 N. Carlevaris-Bianco, R.M. Eustice: Generic factor-based node marginalization and edge sparsification for pose-graph SLAM, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2013)

46.104 M. Kaess, F. Dellaert: Covariance recovery from a square root information matrix for data association, J. Robotics Auton. Syst. **57**(12), 1198–1210 (2009)

46.105 H. Kretzschmar, C. Stachniss: Information-theoretic compression of pose graphs for laser-based SLAM, Int. J. Robotics Res. **31**(11), 1219–1230 (2012)

46.106 U. Frese, L. Schröder: Closing a million-landmarks loop, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2006)

46.107 J. McDonald, M. Kaess, C. Cadena, J. Neira, J.J. Leonard: Real-time 6-DOF multi-session visual SLAM over large scale environments, J. Robotics Auton. Syst. **61**(10), 1144–1158 (2012)

46.108 H. Strasdat, J.M.M. Montiel, A.J. Davison: Real-time monocular SLAM: Why filter?, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2010)

46.109 P. Henry, M. Krainin, E. Herbst, X. Ren, D. Fox: RGB-D mapping: Using depth cameras for dense 3D modeling of indoor environments, Int. J. Robotics Res. **31**(5), 647–663 (2012)

46.110 D. Nister, H. Stewenius: Scalable recognition with a vocabulary tree, Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (ICCVPR) (2006) pp. 2161–2168

46.111 R.A. Brooks: Aspects of mobile robot visual map making, Proc. Int. Symp. Robotics Res. (ISRR) (MIT Press, Cambridge 1984) pp. 287–293

46.112 H. Moravec: Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover, Ph.D. Thesis (Stanford Univ., Stanford 1980)

46.113 N. Ayache, O. Faugeras: Building, registrating, and fusing noisy visual maps, Int. J. Robotics Res. **7**(6), 45–65 (1988)

46.114 D. Kriegman, E. Triendl, T. Binford: Stereo vision and navigation in buildings for mobile robots, IEEE Trans. Robotics Autom. **5**(6), 792–803 (1989)

46.115 L. Matthies, S. Shafer: Error modeling in stereo navigation, IEEE J. Robotics Autom. **3**(3), 239–248 (1987)

46.116 S. Pollard, J. Porrill, J. Mayhew: Predictive feed-forward stereo processing, Alvey Vis. Conf. (1989) pp. 97–102

46.117 M. Smith, I. Baldwin, W. Churchill, R. Paul, P. Newman: The new college vision and laser data set, Int. J. Robotics Res. **28**(5), 595–599 (2009)

46.118 M. Cummins, P.M. Newman: Appearance-only SLAM at large scale with FAB-MAP 2.0, Int. J. Robotics Res. **30**(9), 1100–1123 (2010)

46.119 P.M. Newman, G. Sibley, M. Smith, M. Cummins, A. Harrison, C. Mei, I. Posner, R. Shade, D. Schroter, L. Murphy, W. Churchill, D. Cole, I. Reid: Navigating, recognising and describing urban spaces with vision and laser, Int, J. Robotics Res. **28**, 11–12 (2009)

46.120 G. Sibley, C. Mei, I. Reid, P. Newman: Vast-scale outdoor navigation using adaptive relative bundle adjustment, Int. J. Robotics Res. **29**(8), 958–980 (2010)

46.121 A. Davison, D. Murray: Mobile robot localisation using active vision, Eur. Conf. Comput. Vis. (ECCV) (1998) pp. 809–825

46.122 A.J. Davison, I. Reid, N. Molton, O. Stasse: MonoSLAM: Real-time single camera SLAM, IEEE Trans., Pattern Anal. Mach. Intell. **29**(6), 1052–1067 (2007)

46.123 R.O. Castle, G. Klein, D.W. Murray: Wide-area augmented reality using camera tracking and mapping in multiple regions, Comput. Vis. Image Underst. **115**(6), 854–867 (2011)

46.124 E. Eade, T. Drummond: Unified loop closing and recovery for real time monocular SLAM, Br. Mach. Vis. Conf. (2008)

46.125 E. Eade, P. Fong, M.E. Munich: Monocular graph SLAM with complexity reduction, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2010) pp. 3017–3024

46.126 K. Konolige, M. Agrawal: FrameSLAM: From bundle adjustment to real-time visual mapping, IEEE Trans. Robotics **24**(5), 1066–1077 (2008)

46.127 D. Nister, O. Naroditsky, J. Bergen: Visual odometry for ground vehicle applications, J. Field Robotics **23**(1), 3–20 (2006)

46.128 D. Scaramuzza, F. Fraundorfer: Visual odometry. Part I: The first 30 years and fundamentals, IEEE Robotics Autom. Mag. **18**(4), 80–92 (2011)

46.129 F. Fraundorfer, D. Scaramuzza: Visual odometry. Part II: Matching, robustness, optimization, and applications, IEEE Robotics Autom. Mag. **19**(2), 78–90 (2012)

46.130 A.S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, N. Roy: Visual odometry and mapping for autonomous flight using an RGB-D camera, Proc. Int. Symp. Robotics Res. (ISRR) (2011)

46.131 J. Sivic, A. Zisserman: Video Google: A text retrieval approach to object matching in videos, Int. Conf. Computer Vis. (ICCV) (2003) p. 1470

46.132 M. Cummins, P.M. Newman: Probabilistic appearance based navigation and loop closing, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2007) pp. 2042–2048

46.133 C. Cadena, D. Gálvez, F. Ramos, J.D. Tardós, J. Neira: Robust place recognition with stereo cameras, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2010)

46.134 R.A. Newcombe, A.J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, A. Fitzgibbon: Kinectfusion: Real-time dense surface mapping and tracking, IEEE/ACM Int. Sym. Mixed Augment. Real. (ISMAR) (2011) pp. 127–136

46.135 J. Neira, A.J. Davison, J.J. Leonard: Guest editorial special issue on visual SLAM, IEEE Trans. Robotics **24**(5), 929–931 (2008)

46.136 K. Konolige, J. Bowman: Towards lifelong visual maps, Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) (2009) pp. 1156–1163

46.137 K. Konolige, J. Bowman, J.D. Chen, P. Mihelich, M. Calonder, V. Lepetit, P. Fua: View-

based maps, Int. J. Robotics Res. **29**(8), 941–957 (2010)
46.138 G. Sibley, C. Mei, I. Reid, P. Newman: Planes, trains and automobiles – Autonomy for the modern robot, Proc. IEEE Int. Conf. Robotics Autom. (ICRA) (2010) pp. 285–292
46.139 T. Whelan, H. Johannsson, M. Kaess, J.J. Leonard, J.B. McDonald: Robust real-time visual odometry for dense RGB-D mapping, IEEE Int. Conf. Robotics Autom. (ICRA (2013)
46.140 R.A. Newcombe, S.J. Lovegrove, A.J. Davison: DTAM: Dense tracking and mapping in real-time, Int. Conf. Computer Vis. (ICCV) (2011) pp. 2320–2327
46.141 F. Steinbruecker, J. Sturm, D. Cremers: Real-time visual odometry from dense RGB-D images, Workshop Live Dense Reconstr. Mov. Cameras Int. Conf. Comput. Vis. (ICCV) (2011)
46.142 H. Roth, M. Vona: Moving volume kinectfusion, Br. Mach. Vis. Conf. (2012)