Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms

Hugh Durrant-Whyte, *Fellow, IEEE*, and Tim Bailey

Abstract— **This tutorial provides an introduction to Simultaneous Localisation and Mapping (SLAM) and the extensive research on SLAM that has been undertaken over the past decade. SLAM is the process by which a mobile robot can build a map of an environment and at the same time use this map to compute it's own location. The past decade has seen rapid and exciting progress in solving the SLAM problem together with many compelling implementations of SLAM methods. Part I of this tutorial (this paper), describes the probabilistic form of the SLAM problem, essential solution methods and significant implementations. Part II of this tutorial will be concerned with recent advances in computational methods and new formulations of the SLAM problem for large scale and complex environments.**

I. Introduction

The Simultaneous Localisation and Mapping (SLAM) problem asks if it is possible for a mobile robot to be placed at an unknown location in an unknown environment and for the robot to incrementally build a consistent map of this environment while simultaneously determining its location within this map. A solution to the SLAM problem has been seen as a 'holy grail' for the mobile robotics community as it would provide the means to make a robot truly autonomous.

The 'solution' of the SLAM problem has been one of the notable successes of the robotics community over the past decade. SLAM has been formulated and solved as a theoretical problem in a number of different forms. SLAM has also been implemented in a number of different domains from indoor robots, to outdoor, underwater and airborne systems. At a theoretical and conceptual level, SLAM can now be considered a solved problem. However, substantial issues remain in practically realizing more general SLAM solutions and notably in building and using perceptually rich maps as part of a SLAM algorithm.

This two-part tutorial and survey of SLAM aims to provide a broad introduction to this rapidly growing field. Part I (this paper) begins by providing a brief history of early developments in SLAM. Section III introduces the structure the SLAM problem in now standard Bayesian form, and explains the evolution of the SLAM process. Section IV describes the two key computational solutions to the SLAM problem; through the use of the extended Kalman filter (EKF-SLAM) and through the use of Rao-Blackwellised particle filters (FastSLAM). Other recent solutions to the SLAM problem are discussed in Part II of this tutorial. Section V describes a number of important real-world implementations of SLAM and also highlights implementations where the sensor data and software are freely down-loadable for other researchers to study. Part II of this tutorial describes major issues in computation, convergence and data association in SLAM. These are subjects that have been the main focus of the SLAM research community over the past five years.

Hugh Durrant-Whyte, and Tim Bailey are with the Australian Centre for Field Robotics (ACFR) J04, The University of Sydney, Sydney NSW 2006, Australia, hugh@acfr.usyd.edu.au,tbailey@acfr.usyd.edu.au

II. History of the SLAM Problem

The genesis of the probabilistic SLAM problem occurred at the 1986 IEEE Robotics and Automation Conference held in San Francisco. This was a time when probabilistic methods were only just beginning to be introduced into both robotics and AI. A number of researchers had been looking at applying estimation-theoretic methods to mapping and localisation problems; these included Peter Cheeseman, Jim Crowley, and Hugh Durrant-Whyte. Over the course of the conference many paper table cloths and napkins were filled with long discussions about consistent mapping. Along the way, Raja Chatila, Oliver Faugeras, Randal Smith and others also made useful contributions to the conversation.

The result of this conversation was a recognition that consistent probabilistic mapping was a fundamental problem in robotics with major conceptual and computational issues that needed to be addressed. Over the next few years a number of key papers were produced. Work by Smith and Cheesman [39] and Durrant-Whyte [17] established a statistical basis for describing relationships between landmarks and manipulating geometric uncertainty. A key element of this work was to show that there must be a high degree of correlation between estimates of the location of different landmarks in a map and that indeed these correlations would grow with successive observations.

At the same time Ayache and Faugeras [1] were undertaking early work in visual navigation, Crowley [9] and Chatila and Laumond [6] in sonar-based navigation of mobile robots using Kalman filter-type algorithms. These two strands of research had much in common and resulted soon after in the landmark paper by Smith, Self and Cheeseman [40]. This paper showed that as a mobile robot moves through an unknown environment taking relative observations of landmarks, the estimates of these landmarks are all necessarily correlated with each other because of the common error in estimated vehicle location [27]. The implication of this was profound: A consistent full solution to the combined localisation and mapping problem would require a joint state composed of the vehicle pose and every landmark position, to be updated following each landmark observation. In turn, this would require the estimator to employ a huge state vector (of order the number of landmarks maintained in the map) with computation scaling as the square of the number of landmarks.

Crucially, this work did not look at the convergence properties of the map or its steady-state behavior. Indeed, it was widely assumed at the time that the estimated map errors would not converge and would instead exhibit a random walk behavior with unbounded error growth. Thus, given the computational complexity of the mapping problem and without knowledge of the convergence behavior of the map, researchers instead focused on a series of approximations to the consistent mapping problem solution which assumed or even forced the correlations between landmarks to be minimized or eliminated so reducing the full filter to a series of decoupled landmark to vehicle filters ([28], [38] for example). Also for these reasons, theoretical work on the combined localisation and mapping problem came to a temporary halt, with work often focused on either mapping or localisation as separate problems.

The conceptual break-through came with the realisation that the combined mapping and localisation problem, once formulated as a single estimation problem, was actually convergent. Most importantly, it was recognised that the correlations between landmarks, that most researchers had tried to minimize, were actually the critical part of the problem and that, on the contrary, the more these correlations grew, the better the solution. The structure of the SLAM problem, the convergence result and the coining of the acronym 'SLAM' was first presented in a mobile robotics survey paper presented at the 1995 International Symposium on Robotics Research [16]. The essential theory on convergence and many of the initial results were developed by Csorba [11], [10]. Several groups already working on mapping and localisation, notably at MIT [29], Zaragoza [5], [4], the ACFR at Sydney [20], [45] and others [7], [13], began working in earnest on SLAM[1] applications in indoor, outdoor and sub-sea environments.

At this time, work focused on improving computational efficiency and addressing issues in data association or 'loop closure'. The 1999 International Symposium on Robotics Research (ISRR'99) [23] was an important meeting point where the first SLAM session was held and where a degree of convergence between the Kalman-filter based SLAM methods and the probabilistic localisation and mapping methods introduced by Thrun [42] was achieved. The 2000 IEEE ICRA Workshop on SLAM attracted fifteen researchers and focused on issues such as algorithmic complexity, data association and implementation challenges. The following SLAM workshop at the 2002 ICRA attracted 150 researchers with a broad range of interests and applications. The 2002 SLAM summer school hosted by Henrik Christiansen at KTH in Stockholm attracted all the key researchers together with some 50 PhD students from around the world and was a tremendous success in building the field. Interest in SLAM has grown exponentially in recent years, and workshops continue to be held at both ICRA and IROS. The SLAM summer school ran in 2004 in Tolouse and will run at Oxford in 2006.

[1]Also called Concurrent Mapping and Localisation (CML) at this time.

III. Formulation and Structure of the SLAM problem

SLAM is a process by which a mobile robot can build a map of an environment and at the same time use this map to deduce it's location. In SLAM both the trajectory of the platform and the location of all landmarks are estimated on-line without the need for any *a priori* knowledge of location.

A. Preliminaries

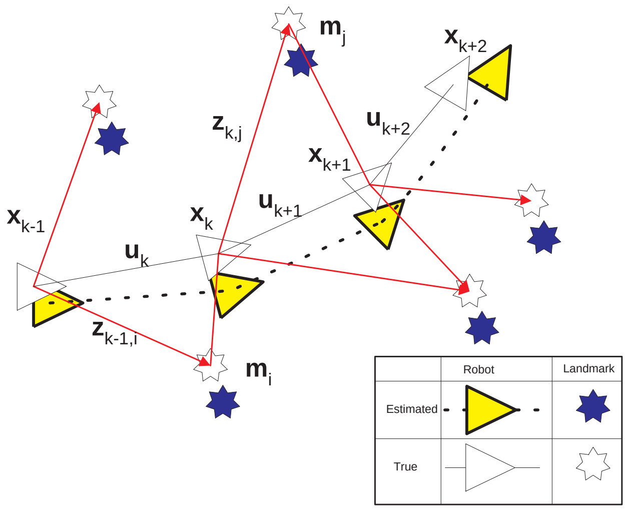

Fig. 1. *The essential SLAM problem. A simultaneous estimate of both robot and landmark locations is required. The true locations are never known or measured directly. Observations are made between true robot and landmark locations. See text for details.*

Consider a mobile robot moving through an environment taking relative observations of a number of unknown landmarks using a sensor located on the robot as shown in Figure 1. At a time instant k, the following quantities are defined:

- $\mathbf{x}_k$: The state vector describing the location and orientation of the vehicle.
- $\mathbf{u}_k$: The control vector, applied at time $k-1$ to drive the vehicle to a state $\mathbf{x}_k$ at time k.
- $\mathbf{m}_i$: A vector describing the location of the i^{th} landmark whose true location is assumed time invariant.
- $\mathbf{z}_{ik}$: An observation taken from the vehicle of the location of the i^{th} landmark at time k. When there are multiple landmark observations at any one time or when the specific landmark is not relevant to the discussion, the observation will be written simply as $\mathbf{z}_k$.

In addition, the following sets are also defined:

- $\mathbf{X}_{0:k} = \{\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_k\} = \{\mathbf{X}_{0:k-1}, \mathbf{x}_k\}$: The history of vehicle locations.
- $\mathbf{U}_{0:k} = \{\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_k\} = \{\mathbf{U}_{0:k-1}, \mathbf{u}_k\}$: The history of control inputs.
- $\mathbf{m} = \{\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_n\}$ The set of all landmarks.
- $\mathbf{Z}_{0:k} = \{\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_k\} = \{\mathbf{Z}_{0:k-1}, \mathbf{z}_k\}$: The set of all landmark observations.

B. Probabilistic SLAM

In probabilistic form, the Simultaneous Localisation and Map Building (SLAM) problem requires that the probability distribution

$$P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0) \tag{1}$$

be computed for all times k. This probability distribution describes the *joint* posterior density of the landmark locations and vehicle state (at time k) given the recorded observations and control inputs up to and including time k together with the initial state of the vehicle. In general, a recursive solution to the SLAM problem is desirable. Starting with an estimate for the distribution $P(\mathbf{x}_{k-1}, \mathbf{m} \mid \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k-1})$ at time $k-1$, the joint posterior, following a control $\mathbf{u}_k$ and observation $\mathbf{z}_k$, is computed using Bayes Theorem. This computation requires that a state transition model and an observation model are defined describing the effect of the control input and observation respectively.

The **observation model** describes the probability of making an observation $\mathbf{z}_k$ when the vehicle location and landmark locations are known, and is generally described in the form

$$P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m}). \tag{2}$$

It is reasonable to assume that once the vehicle location and map are defined, observations are conditionally independent given the map and the current vehicle state.

The **motion model** for the vehicle can be described in terms of a probability distribution on state transitions in the form

$$P(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k) \tag{3}$$

That is, the state transition is assumed to be a Markov process in which the next state $\mathbf{x}_k$ depends only on the immediately proceeding state $\mathbf{x}_{k-1}$ and the applied control $\mathbf{u}_k$, and is independent of both the observations and the map.

The SLAM algorithm is now implemented in a standard two-step recursive (sequential) prediction (time-update) correction (measurement-update) form:

Time-update

$$\begin{aligned}
&P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k}, \mathbf{x}_0) \\
&= \int P(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k) \\
&\times P(\mathbf{x}_{k-1}, \mathbf{m} \mid \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k-1}, \mathbf{x}_0) \mathrm{d}\mathbf{x}_{k-1}
\end{aligned} \tag{4}$$

Measurement Update

$$\begin{aligned}
&P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0) \\
&= \frac{P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m}) P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k}, \mathbf{x}_0)}{P(\mathbf{z}_k \mid \mathbf{Z}_{0:k-1}, \mathbf{U}_{0:k})}
\end{aligned} \tag{5}$$

Equations 4 and 5 provide a recursive procedure for calculating the joint posterior $P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0)$ for the robot state $\mathbf{x}_k$ and map $\mathbf{m}$ at a time k based on all observations $\mathbf{Z}_{0:k}$ and all control inputs $\mathbf{U}_{0:k}$ up to and including time k. The recursion is a function of a vehicle model $P(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k)$ and an observation model $P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m})$.

It is worth noting that the map building problem may be formulated as computing the conditional density $P(\mathbf{m} \mid \mathbf{X}_{0:k}, \mathbf{Z}_{0:k}, \mathbf{U}_{0:k})$. This assumes that the location of the vehicle $\mathbf{x}_k$ is known (or at least deterministic) at all times, subject to knowledge of initial location. A map $\mathbf{m}$ is then constructed by fusing observations from different locations. Conversely, the localisation problem may be formulated as computing the probability distribution $P(\mathbf{x}_k \mid \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{m})$. This assumes that the landmark locations are known with certainty and the objective is to compute an estimate of vehicle location with respect to these landmarks.

C. Structure of Probabilistic SLAM

To simplify the discussion in this section we will drop the conditioning on historical variables in Equation 1 and write the required joint posterior on map and vehicle location as $P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{z}_k)$ and even $P(\mathbf{x}_k, \mathbf{m})$ as the context permits.

The observation model $P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m})$ makes explicit the dependence of observations on both the vehicle and landmark locations. It follows that the joint posterior can not be partitioned in the obvious manner

$$P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{z}_k) \neq P(\mathbf{x}_k \mid \mathbf{z}_k) P(\mathbf{m} \mid \mathbf{z}_k),$$

and indeed it is well known from the early papers on consistent mapping [39], [17] that a partition such as this leads to inconsistent estimates. However, the SLAM problem has more structure than is immediately obvious from these Equations.

Referring again to Figure 1, it can be seen that much of the error between estimated and true landmark locations is common between landmarks and is in fact due to a single source; errors in knowledge of where the robot is when landmark observations are made. In turn, this implies that the errors in landmark location estimates are highly correlated. Practically, this means that the relative location between any two landmarks, $\mathbf{m}_i - \mathbf{m}_j$, may be known with high accuracy, even when the absolute location of a landmark $\mathbf{m}_i$ is quite uncertain. In probabilistic form this means that the joint probability density for the pair of landmarks $P(\mathbf{m}_i, \mathbf{m}_j)$ is highly peaked even when the marginal densities $P(\mathbf{m}_i)$ may be quite dispersed.

The most important insight in SLAM was to realize that the correlations between landmark estimates increase *monotonically* as more and more observations are made[2]. Practically, this means that knowledge of the relative location of landmarks always improves and never diverges, regardless of robot motion. In probabilistic terms, this means that the joint probability density on all landmarks $P(\mathbf{m})$ becomes monotonically more peaked as more observations are made.

This convergence occurs because the observations made by the robot can be considered as 'nearly independent' measurements of the *relative* location between landmarks. Referring again to Figure 1, consider the robot at location $\mathbf{x}_k$ observing the two landmarks $\mathbf{m}_i$ and $\mathbf{m}_j$. The relative location of observed landmarks is clearly independent of the coordinate frame of the vehicle and successive observations from this fixed location would yield further independent measurements of the relative relationship between landmarks. Now, as the robot moves to location $\mathbf{x}_{k+1}$, it again observes landmark $\mathbf{m}_j$ this allows the estimated location of the robot and landmark to be updated relative to the previous location $\mathbf{x}_k$. In turn this propagates back to update landmark $\mathbf{m}_i$ *even though this landmark is not seen from the new location*. This occurs because the two landmarks are highly correlated (their relative location is well known) from previous measurements. Further, the fact that the same measurement data is used to update these two landmarks makes them *more* correlated. The term 'nearly independent' measurement is appropriate because the observation errors will be correlated through successive vehicle motions. Also note that in Figure 1 at location $\mathbf{x}_{k+1}$ the robot observes two new landmarks *relative* to $\mathbf{m}_j$. These new land-marks are thus immediately linked or correlated to the rest of the map. Later update to these landmarks will also update landmark $\mathbf{m}_j$ and through this landmark $\mathbf{m}_i$ and so on. That is, all landmarks end up forming a network linked by relative location or correlations whose precision or value increases whenever an observation is made.

This process can be visualized (Figure 2) as a network of springs connecting all landmarks together, or as a rubber sheet in which all landmarks are embedded. An observation in a neighbourhood acts like a displacement to spring system or rubber sheet such that it's effect is great in the neighbourhood and, dependent on local stiffness (correlation) properties, diminishes with distance to other landmarks. As the robot moves through this environment and takes observations of the landmarks, the the springs become increasingly (and monotonically) stiffer. In the limit, a rigid map of landmarks or an accurate *relative* map of the environment is obtained. As the map is built, the location accuracy of the robot measured relative to the map is bounded only by the quality of the map and relative measurement sensor. In the theoretical limit, robot relative location accuracy becomes equal to the localisation accuracy achievable with a given map.

[2]These results have only been proved for the linear Gaussian case [14]. Formal proof for the more general probabilistic case remains an open problem.

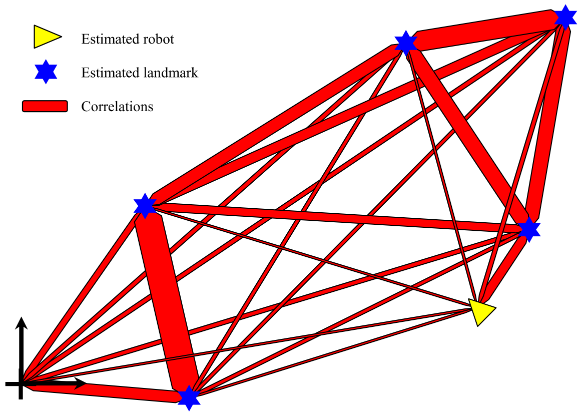

Fig. 2. *Spring network analogy. The landmarks are connected by springs describing correlations between landmarks. As the vehicle moves back and forth through the environment, spring stiffness or correlations increase (red links become thicker). As landmarks are observed and estimated locations are corrected, and these changes are propagated through the spring network. Note, the robot itself is correlated to the map.*

IV. Solutions to the SLAM Problem

Solutions to the probabilistic SLAM problem involve finding an appropriate representation for the observation model Equation 2 and motion model Equation 3 which allows efficient and consistent computation of the prior and posterior distributions in Equations 4 and 5. By far the most common representation is in the form of a state-space model with additive Gaussian noise, leading to the use of the extended Kalman filter (EKF) to solve the SLAM problem as described in Section IV-A. One important alternative representation is to describe the vehicle motion model in Equation 3 as a set of samples of a more general non-Gaussian probability distribution. This leads to the use of the Rao-Blackwellised particle filter, or FastSLAM algorithm, to solve the SLAM problem as described in Section IV-B. While EKF-SLAM and FastSLAM are the two most important solution methods, newer alternatives, which offer much potential, have been proposed including the use of the information-state form [43]. These are discussed further in Part II of this tutorial.

A. EKF-SLAM

The basis for the EKF-SLAM method is to describe the vehicle motion in the form

$$P(\mathbf{x}_k \mid \mathbf{x}_{k-1}, \mathbf{u}_k) \Longleftrightarrow \mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{u}_k) + \mathbf{w}_k, \quad (6)$$

where $\mathbf{f}(\cdot)$ models vehicle kinematics and where $\mathbf{w}_k$ are additive, zero mean uncorrelated Gaussian motion disturbances with covariance $\mathbf{Q}_k$. The observation model is described in the form

$$P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m}) \Longleftrightarrow \mathbf{z}(k) = \mathbf{h}(\mathbf{x}_k, \mathbf{m}) + \mathbf{v}_k, \quad (7)$$

where $\mathbf{h}(\cdot)$ describes the geometry of the observation and where $\mathbf{v}_k$ are additive, zero mean uncorrelated Gaussian observation errors with covariance $\mathbf{R}_k$.

With these definitions the standard EKF method [31], [14] can be applied to compute the mean

$$\begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{m}}_k \end{bmatrix} = \mathrm{E}\begin{bmatrix} \mathbf{x}_k \\ \mathbf{m} \end{bmatrix} \mid \mathbf{Z}_{0:k} \Bigg] ,$$

and covariance

$$\begin{aligned} \mathbf{P}_{k|k} &= \begin{bmatrix} \mathbf{P}_{xx} & \mathbf{P}_{xm} \\ \mathbf{P}_{xm}^T & \mathbf{P}_{mm} \end{bmatrix}_{k|k} \\ &= \mathrm{E}\left[\begin{pmatrix} \mathbf{x}_k - \hat{\mathbf{x}}_k \\ \mathbf{m} - \hat{\mathbf{m}}_k \end{pmatrix} \begin{pmatrix} \mathbf{x}_k - \hat{\mathbf{x}}_k \\ \mathbf{m} - \hat{\mathbf{m}}_k \end{pmatrix}^T \mid \mathbf{Z}_{0:k} \right] \end{aligned}$$

of the joint posterior distribution $P(\mathbf{x}_k, \mathbf{m} \mid \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0)$ from:

Time-update

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{f}(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k) \quad (8)$$

$$\mathbf{P}_{xx,k|k-1} = \nabla\mathbf{f}\ \mathbf{P}_{xx,k-1|k-1} \nabla\mathbf{f}^T + \mathbf{Q}_k \quad (9)$$

where $\nabla\mathbf{f}$ is the Jacobian of $\mathbf{f}$ evaluated at the estimate $\hat{\mathbf{x}}_{k-1|k-1}$. There is generally no need to perform a time-update for stationary landmarks[3].

Observation-update

$$\begin{bmatrix} \hat{\mathbf{x}}_{k|k} \\ \hat{\mathbf{m}}_k \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{x}}_{k|k-1} \\ \hat{\mathbf{m}}_{k-1} \end{bmatrix} + \mathbf{W}_k \left[\mathbf{z}(k) - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1}, \hat{\mathbf{m}}_{k-1}) \right] \quad (10)$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{W}_k \mathbf{S}_k \mathbf{W}_k^T \quad (11)$$

where

$$\mathbf{S}_k = \nabla\mathbf{h} \mathbf{P}_{k|k-1} \nabla\mathbf{h}^T + \mathbf{R}_k$$

$$\mathbf{W}_k = \mathbf{P}_{k|k-1} \nabla\mathbf{h}^T \mathbf{S}_k^{-1}$$

and where $\nabla\mathbf{h}$ is the Jacobian of $\mathbf{h}$ evaluated at $\hat{\mathbf{x}}_{k|k-1}$ and $\hat{\mathbf{m}}_{k-1}$.

This EKF-SLAM solution is very well known and inherits many of the same benefits and problems as the standard EKF solutions to navigation or tracking problems. Four of the key issues are briefly discussed here:

Convergence: In the EKF-SLAM problem, convergence of the map is manifest in the monotonic convergence of the determinant of the map covariance matrix $\mathbf{P}_{mm,k}$, and all land-mark pair sub-matrices, toward zero. The individual land-mark variances converge toward a lower bound determined by initial uncertainties in robot position and observations. The typical convergence behaviour of landmark location variances is shown in Figure 3 (from [14]).

Computational Effort: The observation update step requires that all landmarks and the joint covariance matrix be updated every time an observation is made. Naively, this means computation grows quadratically with the number of landmarks. There has been a great deal of work undertaken in developing efficient variants of the EKF-SLAM solution and real-time implementations with many thousands of landmarks have been demonstrated [21], [29]. Efficient variants of the EKF-SLAM algorithm are discussed in Part II of this tutorial.

[3]However, a time-update is necessary for landmarks that may move [44].

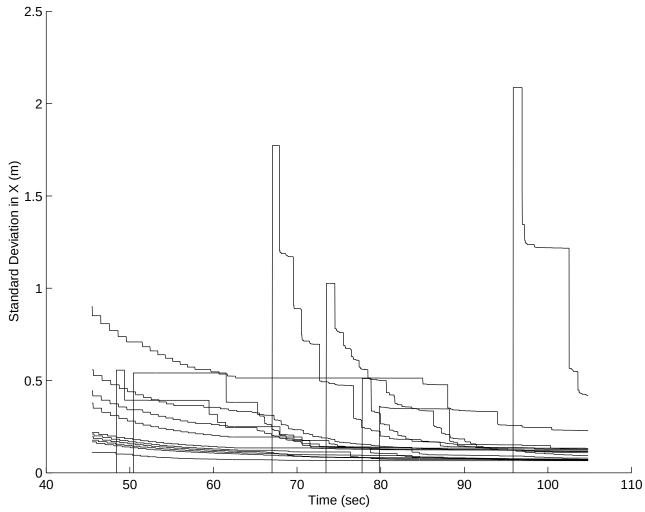

Fig. 3. *The convergence in landmark uncertainty. The plot shows a time history of standard deviations of a set of landmark locations. A landmark is initially observed with uncertainty inherited from the robot location and observation. Over time, the standard deviations reduce monotonically to a lower bound. New landmarks are acquired during motion (from [14]).*

Data Association: The standard formulation of the EKF-SLAM solution is especially fragile to incorrect association of observations to landmarks [35]. The 'loop-closure' problem, when a robot returns to re-observe landmarks after a large traverse, is especially difficult. The association problem is compounded in environments where landmarks are not simple points and indeed look different from different view-points. Current work in this area will be described in Part II of this tutorial.

Non-linearity: EKF-SLAM employs linearised models of non-linear motion and observation models and so inherits many caveats. Non-linearity can be a significant problem in EKF-SLAM and leads to inevitable, and sometimes dramatic, inconsistency in solutions [24]. Convergence and consistency can only be guaranteed in the linear case.

B. Rao-Blackwellised Filter

The FastSLAM algorithm, introduced by Montemerlo *et al.* [32], marked a fundamental conceptual shift in the design of recursive probabilistic SLAM. Previous efforts focused on improving the performance of EKF-SLAM, while retaining its essential linear Gaussian assumptions. FastSLAM with its basis on recursive Monte Carlo sampling, or particle filtering, was the first to directly represent the non-linear process model and non-Gaussian pose distribution.[4] This approach was influenced by earlier probabilistic mapping experiments of Murphy [34] and Thrun [41].

The high dimensional state-space of the SLAM problem makes direct application of particle filters computationally infeasible. However, it is possible to reduce the sample-space by applying Rao-Blackwellisation (R-B),

[4]Note, FastSLAM still linearises the observation model, but this is typically a reasonable approximation for range-bearing measurements when the vehicle pose is known.

whereby a joint state is partitioned according to the product rule $P(\mathbf{x}_1, \mathbf{x}_2) = P(\mathbf{x}_2 \mid \mathbf{x}_1)P(\mathbf{x}_1)$ and, if $P(\mathbf{x}_2 \mid \mathbf{x}_1)$ can be represented analytically, only $P(\mathbf{x}_1)$ need be sampled $\mathbf{x}_1^{(i)} \sim P(\mathbf{x}_1)$. The joint distribution, therefore, is represented by the set $\{\mathbf{x}_1^{(i)}, P(\mathbf{x}_2 \mid \mathbf{x}_1^{(i)})\}_i^N$ and statistics such as the marginal

$$P(\mathbf{x}_2) \approx \frac{1}{N}\sum_i^N P(\mathbf{x}_2 \mid \mathbf{x}_1^{(i)})$$

can be obtained with greater accuracy than is possible by sampling over the joint space.

The joint SLAM state may be factored into a vehicle component and a conditional map component.

$$\begin{aligned} &P(\mathbf{X}_{0:k}, \mathbf{m} \mid \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0) \\ &= P(\mathbf{m} \mid \mathbf{X}_{0:k}, \mathbf{Z}_{0:k})P(\mathbf{X}_{0:k} \mid \mathbf{Z}_{0:k}, \mathbf{U}_{0:k}, \mathbf{x}_0). \quad (12) \end{aligned}$$

Here the probability distribution is on the trajectory $\mathbf{X}_{0:k}$ rather than the single pose $\mathbf{x}_k$ because, when conditioned on the trajectory, the map landmarks become independent (see Figure 4). This is a key property of FastSLAM, and the reason for its speed; the map is represented as a set of independent Gaussians, with linear complexity, rather than a joint map covariance with quadratic complexity.

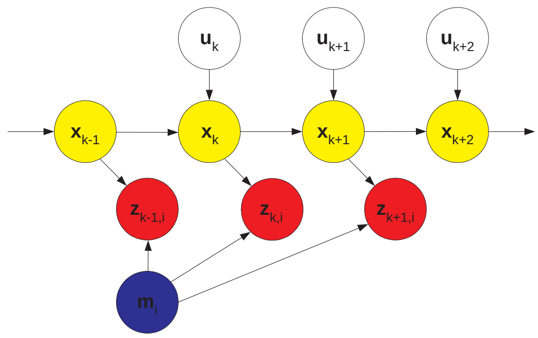

Fig. 4. *A graphical model of the SLAM algorithm. If the history of pose states are known exactly then, since the observations are conditionally independent, the map states are also independent. For FastSLAM, each particle defines a different vehicle trajectory hypothesis.*

The essential structure of FastSLAM, then, is a Rao-Blackwellised state, where the trajectory is represented by weighted samples and the map is computed analytically. Thus, the joint distribution, at time k, is represented by the set $\{w_k^{(i)}, \mathbf{X}_{0:k}^{(i)}, P(\mathbf{m} \mid X_{0:k}^{(i)}, \mathbf{Z}_{0:k})\}_i^N$, where the map accompanying each particle is composed of independent Gaussian distributions $P(\mathbf{m} \mid X_{0:k}^{(i)}, \mathbf{Z}_{0:k}) = \prod_j^M P(\mathbf{m}_j \mid X_{0:k}^{(i)}, \mathbf{Z}_{0:k})$. Recursive estimation is performed by particle filtering for the pose states, and the EKF for the map states.

Updating the map, for a given trajectory particle $X_{0:k}^{(i)}$, is trivial. Each observed landmark is processed individually as an EKF measurement update from a known pose (see Figure 5). Unobserved landmarks are unchanged. Propagating the pose particles, on the other hand, is more complex, as we discuss below.

We forego giving a background on particle filters, except to say that it is derived from a recursive form of sampling known as *sequential important sampling* (SIS) [15], which actually samples from a joint *state history*, but "telescopes" the joint into a recursion via the product rule.

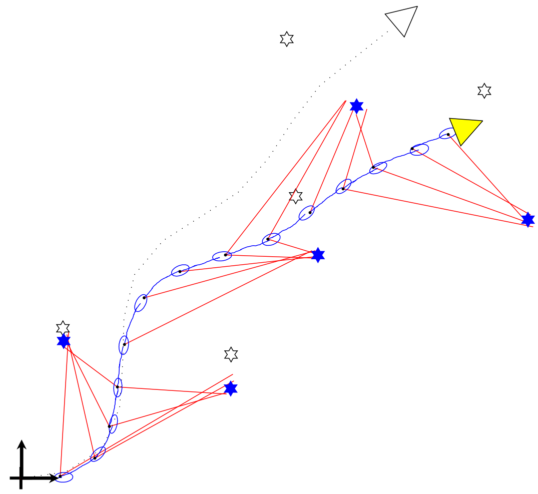

Fig. 5. *A single realisation of robot trajectory in the FastSLAM algorithm. The ellipsoids show the proposal distribution for each update stage, from which a robot pose is sampled and, assuming this pose is perfect, the observed landmarks are updated. Thus, the map for a single particle is governed by the accuracy of the trajectory. Many such trajectories provide a probabilistic model of robot location.*

$$\begin{aligned} &P(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_T \mid \mathbf{Z}_{0:T}) \\ &= P(\mathbf{x}_0 \mid \mathbf{Z}_{0:T})P(\mathbf{x}_1 \mid \mathbf{x}_0, \mathbf{Z}_{0:T}) \ldots P(\mathbf{x}_T \mid \mathbf{X}_{0:T-1}, \mathbf{Z}_{0:T}). \end{aligned}$$

At each time-step k, particles are drawn from a *proposal distribution* $\pi(\mathbf{x}_k \mid \mathbf{X}_{0:k-1}, \mathbf{Z}_{0:k})$, which approximates the true distribution $P(\mathbf{x}_k \mid \mathbf{X}_{0:k-1}, \mathbf{Z}_{0:T})$, and the samples are given importance weights to compensate for any discrepancy. The approximation error grows with time (and inherent joint state-space), increasing the variation in sample weights, degrading statistical accuracy. A *resampling* step reinstates uniform weighting, but causes loss of historical particle information. This leads to a crucial property: *SIS with resampling can produce reasonable statistics only for systems that "exponentially forget" their past* [8] (i.e., systems whose process noise cause the state at time k to become increasingly independent of preceding states).

The general form of a R-B particle filter for SLAM is as follows. We assume that, at time $k-1$, the joint state is represented by $\{w_{k-1}^{(i)}, \mathbf{X}_{0:k-1}^{(i)}, P(\mathbf{m} \mid X_{0:k-1}^{(i)}, \mathbf{Z}_{0:k-1})\}_i^N$.

1. For each particle, compute a proposal distribution, conditioned on the specific particle history, and draw a sample from it

$$\mathbf{x}_k^{(i)} \sim \pi(\mathbf{x}_k \mid \mathbf{X}_{0:k-1}^{(i)}, \mathbf{Z}_{0:k}, \mathbf{u}_k). \quad (13)$$

This new sample is (implicitly) joined to the particle history $\mathbf{X}_{0:k}^{(i)} \triangleq \left\{\mathbf{X}_{0:k-1}^{(i)}, \mathbf{x}_k^{(i)}\right\}$.

2. Weight samples according to the importance function

$$w_k^{(i)} = w_{k-1}^{(i)} \frac{P(\mathbf{z}_k \mid \mathbf{X}_{0:k}^{(i)}, \mathbf{Z}_{0:k-1})P(\mathbf{x}_k^{(i)} \mid \mathbf{x}_{k-1}^{(i)}, \mathbf{u}_k)}{\pi(\mathbf{x}_k^{(i)} \mid \mathbf{X}_{0:k-1}^{(i)}, \mathbf{Z}_{0:k}, \mathbf{u}_k)}. \quad (14)$$

The numerator terms of this equation are the observation model and the motion model, respectively. The former differs from Equation 2 because R-B requires dependency on the map be marginalised away.

$$P(\mathbf{z}_k \mid \mathbf{X}_{0:k}, \mathbf{Z}_{0:k-1}) = \int P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{m}) P(\mathbf{m} \mid \mathbf{X}_{0:k-1}, \mathbf{Z}_{0:k-1}) d\mathbf{m} \quad (15)$$

3. If necessary,[5] perform resampling. Resampling is accomplished by selecting particles, with replacement, from the set $\{\mathbf{X}_{0:k}^{(i)}\}_i^N$, including their associated maps, with probability of selection proportional to $w_k^{(i)}$. Selected particles are given uniform weight, $w_k^{(i)} = \frac{1}{N}$.
4. For each particle, perform an EKF update on the observed landmarks as a simple mapping operation with known vehicle pose.

The two versions of FastSLAM in the literature, FastSLAM 1.0 [32] and FastSLAM 2.0 [33], differ only in terms of the form of their proposal distribution (step 1) and, consequently in their importance weight (step 2). FastSLAM 2.0 is by far the more efficient solution.

For FastSLAM 1.0, the proposal distribution is the motion model

$$\mathbf{x}_k^{(i)} \sim P(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{(i)}, \mathbf{u}_k) \quad (16)$$

Therefore, from Equation 14, the samples are weighted according to the marginalised observation model.

$$w_k^{(i)} = w_{k-1}^{(i)} P(\mathbf{z}_k \mid \mathbf{X}_{0:k}^{(i)}, \mathbf{Z}_{0:k-1}) \quad (17)$$

For FastSLAM 2.0, the proposal distribution includes the current observation

$$\mathbf{x}_k^{(i)} \sim P(\mathbf{x}_k \mid \mathbf{X}_{0:k-1}^{(i)}, \mathbf{Z}_{0:k}, \mathbf{u}_k) \quad (18)$$

where

$$P(\mathbf{x}_k \mid \mathbf{X}_{0:k-1}^{(i)}, \mathbf{Z}_{0:k}, \mathbf{u}_k) = \frac{1}{C} P(\mathbf{z}_k \mid \mathbf{x}_k, \mathbf{X}_{0:k-1}^{(i)}, \mathbf{Z}_{0:k-1}) P(\mathbf{x}_k \mid \mathbf{x}_{k-1}^{(i)}, \mathbf{u}_k) \quad (19)$$

and C is a normalising constant. The importance weight according to Equation 14 is $w_k^{(i)} = w_{k-1}^{(i)} C$. The advantage of FastSLAM 2.0 is that its proposal distribution is locally optimal [15]. That is, for each particle, it gives the smallest possible variance in importance weight $w_k^{(i)}$ conditioned upon the available information, $\mathbf{X}_{0:k-1}^{(i)}$, $\mathbf{Z}_{0:k}$ and $\mathbf{U}_{0:k}$.

Statistically, FastSLAM (1.0 and 2.0) suffers degeneration due to its inability to forget the past. Marginalising the map in Equation 15 introduces dependence on the pose and measurement history, and so, when resampling depletes this history, statistical accuracy is lost [2]. Nevertheless, empirical results of FastSLAM 2.0 in real outdoor environments [33] show that the algorithm is capable of generating an accurate map in practice.

[5]When best to instigate resampling is an open problem. Some implementations resample every time-step, others after a fixed number of time-steps, and others once the weight variance exceeds a threshold.

V. Implementation of SLAM

Practical realisations of probabilistic SLAM have become increasingly impressive in recent years, covering larger areas in more challenging environments. Here we discuss two representative implementations and mention other notable applications.

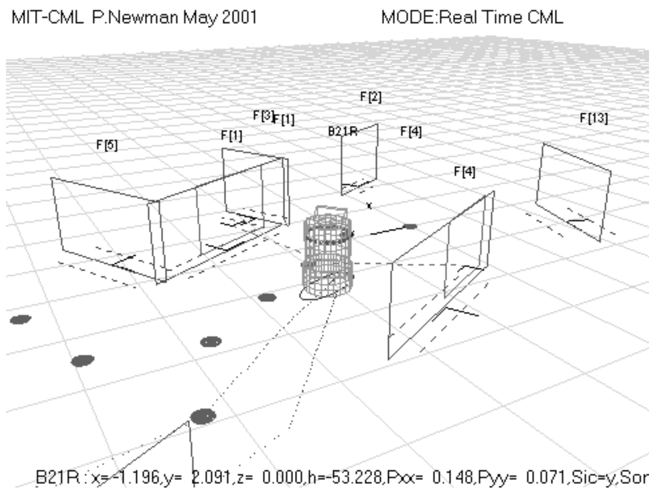

Fig. 6. *Real-time SLAM visualisation by Newman et al. [37].*

The "explore and return" experiment by Newman *et al.* [37] was a moderate-scale indoor implementation that validated the non-divergence properties of EKF-SLAM by returning to a precisely marked starting point. The experiment is remarkable because its return trip was fully autonomous. The robot was manually driven during the exploration phase, although without visual contact by the operator, who relied solely on a real-time rendering of the robot's map (see Figure 6). For the return trip, the robot plans a path and returns without human intervention.

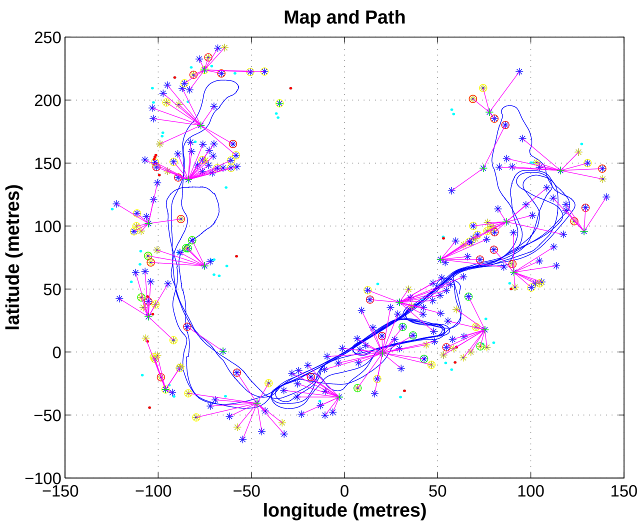

Fig. 7. *Large-scale outdoor SLAM by Guivant and Nebot [21].*

Guivant and Nebot [21] pioneered the application of SLAM in very large outdoor environments (see Figure 7).

Table 1: Open Source SLAM Software		
Author	**Description**	**Link**
Kai Arras	The *CAS Robot Navigation Toolbox*, a MATLAB simulation toolbox for robot localization and mapping.	`www.cas.kth.se/toolbox/index.html`
Tim Bailey	MATLAB simulators for EKF-SLAM, UKF-SLAM, and FastSLAM 1.0 and 2.0.	`www.acfr.usyd.edu.au/homepages/academic/tbailey/software/index.html`
Mark Paskin	Java library with several SLAM variants, including Kalman filter, information filter, and thin junction tree forms. Includes a MATLAB interface.	`www.stanford.edu/~paskin/slam/`
Andrew Davison	*Scene*, a C++ library for map-building and localisation. Facilitates real-time single camera SLAM.	`www.doc.ic.ac.uk/~ajd/Scene/index.html`
José Neira	MATLAB EKF-SLAM simulator that demonstrates *joint compatibility branch-and-bound* data association.	`http://webdiis.unizar.es/~neira/software/slam/slamsim.htm`
Dirk Hähnel	C language grid-based version of FastSLAM.	`www.informatik.uni-freiburg.de/~haehnel/old/download.html`
Various	MATLAB code from the 2002 SLAM summer school.	`www.cas.kth.se/slam/toc.html`

Table 2: Online Datasets		
Author	**Description**	**Link**
Jose Guivant, Juan Nieto and Eduardo Nebot	Numerous large-scale outdoor datasets, notably the popular Victoria Park data.	`www.acfr.usyd.edu.au/homepages/academic/enebot/dataset.htm`
Chieh-Chih Wang	Three large-scale outdoor datasets collected by the Navlab11 testbed.	`www.cs.cmu.edu/~bobwang/datasets.html`
Radish (The Robotics Data Set Repository)	Many and varied indoor datasets, including large-area data from the CSU Stanislaus library, the Intel Research Lab in Seattle, the Edmonton Convention Centre, and more.	`http://radish.sourceforge.net/`
IJRR (The International Journal of Robotics Research)	IJRR maintains a webpage for each article, often containing data and video of results. A good example is a paper by Bosse *et al.* [3], which has data from Killian Court at MIT.	`www.ijrr.org/contents/23_12/abstract/1113.html`

They addressed computational issues of real-time operation, while also dealing with high-speed vehicle motion, non-flat terrain, and dynamic clutter. Their results are particularly interesting because they are accompanied by accurate RTK-GPS ground truth, showing the practical veracity of the algorithm, which involved closing several large loops. The logged data from their Victoria Park trials is available online, and has become a popular benchmark within the SLAM research community.

SLAM applications now exist in a wide variety of domains. They include indoor [4], [7], [12], [3], outdoor [21], [19], aerial [25], and subsea [45], [36], [18]. There are different sensing modalities such as bearing only [13] and range only [30].

We also make honourable mention of *consistent pose estimation* (CPE) [22], [26], which is an entirely different SLAM paradigm based on topological mapping and data alignment, due to its exemplary results in large indoor environments.

Various researchers in the SLAM community have written software demonstrating SLAM, implemented in MATLAB, C++, and Java, and available online (see Table 1). Collections of logged data are listed in Table 2. These datasets are from real sensors in real environments, and are a valuable resource to assess and benchmark the various SLAM algorithms.

VI. Conclusions

This paper has described the SLAM problem, the essential methods for solving the SLAM problem and has summarised key implementations and demonstrations of the method. While there are still many practical issues to overcome, especially in more complex outdoor environments, the general SLAM method is now a well understood and established part of robotics. Part II of this tutorial will summarise more recent work in addressing some of the remaining issues in SLAM including; computation, feature representation and data association.

References

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

[2] T. Bailey, J. Nieto, and E. Nebot. Consistency of the FastSLAM algorithm. In *Proc. IEEE Int. Conf. Robotics and Automation*, 2006.

[3] M. Bosse, P. Newman, J. Leonard, and S. Teller. Simultaneous localization and map building in large-scale cyclic environments using the Atlas framework. *Int. J. Robotics Research*, 23(12):1113–1140, 2004.

[4] J.A. Castellanos, J.M. Martnez, J. Neira, and J.D. Tardós. Experiments in multisensor mobile robot localization and map building. In *Third IFAC Symposium on Intelligent Autonomous Vehicles*, pages 173–178, 1998.

[5] J.A. Castellanos, J.D. Tardós, and G. Schmidt. Building a global map of the environment of a mobile robot: The importance of correlations. In *IEEE International Conference on Robotics and Automation*, pages 1053–1059, 1997.

[6] R. Chatila and J.P. Laumond. Position referencing and consistent world modeling for mobile robots. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 138–143, 1985.

[7] K.S. Chong and L. Kleeman. Feature-based mapping in real, large scale environments using an ultrasonic array. *International Journal Robotics Research*, 18(1):3–19, 1999.

[8] D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on Signal Processing*, 50(3):736–746, 2002.

[9] J. Crowley. World modeling and position estimation for a mobile robot using ultra-sonic ranging. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 674–681, 1989.

[10] M. Csorba. *Simultaneous Localisation and Map Building*. PhD thesis, University of Oxford, 1997.

[11] M. Csorba and H.F. Durrant-Whyte. A new approach to simultaneous localisation and map building. In *Proceedings of SPIE Aerosense*, Orlando, 1996.

[12] A.J. Davison, Y.G. Cid, and N. Kita. Real-time 3D SLAM with wide-angle vision. In *IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, 2004.

[13] M. Deans and M. Hebert. Experimental comparison of techniques for localization and mapping using a bearing-only sensor. In *Int. Symp. Experimental Robotics*, 2000.

[14] G. Dissanayake, P. Newman, H.F. Durrant-Whyte, S. Clark, and M. Csobra. A solution to the simultaneous localisation and mapping (SLAM) problem. *IEEE Trans. Robotics and Automation*, 17(3):229–241, 2001.

[15] A. Doucet. On sequential simulation-based methods for Bayesian filtering. Technical report, Cambridge University, Department of Engineering, 1998.

[16] H. Durrant-Whyte, D. Rye, and E. Nebot. Localisation of automatic guided vehicles. In G. Giralt and G. Hirzinger, editors, *Robotics Research: The 7th International Symposium (ISRR'95)*, pages 613–625. Springer Verlag, 1996.

[17] H.F. Durrant-Whyte. Uncertain geometry in robotics. *IEEE Trans. Robotics and Automation*, 4(1):23–31, 1988.

[18] R. Eustice, H. Singh, J. Leonard, M. Walter, and R. Ballard. Visually navigating the RMS Titanic with SLAM information filters. In *Robotics: Science and Systems*, 2005.

[19] J. Folkesson and H. I. Christensen. Graphical SLAM - a self-correcting map. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 791–798, 2004.

[20] J. Guivant, E.M. Nebot, and S. Baiker. Localization and map building using laser range sensors in outdoor applications. *Journal of Robotic Systems*, 17(10):565–583, 2000.

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

[22] J.S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 318–325, 1999.

[23] J. Hollerbach and D. Koditscheck (Editors). *Robotics Research, The Ninth International Symposium (ISRR'99)*. Springer-Verlag, 2000.

[24] S.J. Julier and J.K. Uhlmann. A counter example to the theory of simultaneous localization and map building. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 4238–4243, 2001.

[25] J. Kim and S. Sukkarieh. Airborne simultaneous localisation and map building. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 406–411, 2003.

[26] K. Konolige. Large-scale map-making. In *National Conference on AI (AAAI)*, pages 457–463, 2004.

[27] J.J. Leonard and H.F. Durrant-Whyte. Simultaneous map building and localisation for an autonomous mobile robot. In *Proc. IEEE Int. Workshop on Intelligent Robots and Systems (IROS)*, pages 1442–1447, Osaka, Japan, 1991.

[28] J.J. Leonard and H.F. Durrant-Whyte. *Directed Sonar Navigation*. Kluwer Academic Press, 1992.

[29] J.J. Leonard and H.J.S. Feder. A computational efficient method for large-scale concurrent mapping and localisation. In J. Hollerbach and D. Koditscheck, editors, *Robotics Research, The Ninth International Symposium (ISRR'99)*, pages 169–176. Springer-Verlag, 2000.

[30] J.J. Leonard, R.J. Rikoski, P.M. Newman, and M.C. Bosse. Mapping partially observable features from multiple uncertain vantage points. *Int. J. Robotics Research*, 21(10–11):943–975, 2002.

[31] P.S. Maybeck. *Stochastic Models, Estimaton and Control, Vol. I*. Academic Press, 1979.

[32] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *AAAI National Conference on Artificial Intelligence*, pages 593–598, 2002.

[33] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *International Joint Conference on Artificial Intelligence*, pages 1151–1156, 2003.

[34] K. Murphy. Bayesian map learning in dynamic environments. In *Advances in Neural Information Processing Systems*, 1999.

[35] J. Neira and J.D. Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897, 2001.

[36] P.M. Newman and J.J. Leonard. Pure range-only subsea SLAM. In *Proc. IEEE Int. Conf. Robotics and Automation*, 2003.

[37] P.M. Newman, J.J. Leonard, J. Neira, and J. Tardos. Explore and return: Experimental validation of real time concurrent mapping and localization. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 1802–1809, 2002.

[38] W. D. Renken. Concurrent localization and map building for mobile robots using ultrasonic sensors. In *Proc. IEEE Int. Workshop on Intelligent Robots and Systems (IROS)*, 1993.

[39] R. Smith and P. Cheesman. On the representation of spatial uncertainty. *Int. J. Robotics Research*, 5(4):56–68, 1987.

[40] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I.J. Cox and G.T. Wilfon, editors, *Autonomous Robot Vehicles*, pages 167–193. Springer-Verlag, 1990.

[41] S. Thrun, W. Bugard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In *International Conference on Robotics and Automation*, pages 321–328, 2000.

[42] S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31(1):29–53, 1998.

[43] S. Thrun, Y.Liu, D. Koller, A. Ng, and H. Durrant-Whyte. Simultaneous localisation and mapping with sparse extended information filters. *Int. J. Robotics Research*, 23(7–8):693–716, 2004.

[44] C.C. Wang, C. Thorpe, and S. Thrun. On-line simultaneous localisation and mapping with detection and tracking of moving objects. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 2918–2924, 2003.

[45] S.B. Williams, P. Newman, G.Dissanayake, and H.F. Durrant-Whyte. Autonomous underwater simultaneous localisation and map building. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pages 1793–1798, San Francisco, USA, April 2000.