The International Journal of Robotics Research

http://ijr.sagepub.com/

Sebastian Thrun
Yufeng Liu

Carnegie Mellon University
Pittsburgh, PA, USA

Daphne Koller
Andrew Y. Ng

Stanford University
Stanford, CA, USA

Zoubin Ghahramani

Gatsby Computational Neuroscience Unit
University College London, UK

Hugh Durrant-Whyte

University of Sydney
Sydney, Australia

Simultaneous Localization and Mapping With Sparse Extended Information Filters

Abstract

In this paper we describe a scalable algorithm for the simultaneous mapping and localization (SLAM) problem. SLAM is the problem of acquiring a map of a static environment with a mobile robot. The vast majority of SLAM algorithms are based on the extended Kalman filter (EKF). In this paper we advocate an algorithm that relies on the dual of the EKF, the extended information filter (EIF). We show that when represented in the information form, map posteriors are dominated by a small number of links that tie together nearby features in the map. This insight is developed into a sparse variant of the EIF, called the sparse extended information filter (SEIF). SEIFs represent maps by graphical networks of features that are locally interconnected, where links represent relative information between pairs of nearby features, as well as information about the robot's pose relative to the map. We show that all essential update equations in SEIFs can be executed in constant time, irrespective of the size of the map. We also provide empirical results obtained for a benchmark data set collected in an outdoor environment, and using a multi-robot mapping simulation.

KEY WORDS—mobile robotics, mapping, SLAM, filters, Kalman filters, information filters, multi-robot systems, robotic perception, robot learning

1. Introduction

Simultaneous localization and mapping (SLAM) is the problem of acquiring a map of an unknown environment with a moving robot, while simultaneously localizing the robot relative to this map (Leonard and Durrant-Whyte 1992; Dissanayake et al. 2001). The SLAM problem addresses situations where the robot lacks a global positioning sensor. Instead, it has to rely on a sensor of incremental egomotion for robot position estimation (e.g., odometry, inertial navigation). Such sensors accumulate error over time, making the problem of acquiring an accurate map a challenging one (Thorpe and Durrant-Whyte 2001). In recent years, the SLAM problem has received considerable attention by the scientific community, and a flurry of new algorithms and techniques has emerged (Leonard et al. 2002).

Existing algorithms can be subdivided into batch and on-line techniques. The former offer sophisticated techniques to cope with perceptual ambiguities (Shatkay and Kaelbling 1997; Thrun, Fox, and Burgard 1998; Burgard et al. 1999), but they can only generate maps after extensive batch processing. On-line techniques are specifically suited to acquire maps as the robot navigates (Smith and Cheeseman 1985; Dissanayake et al. 2001). On-line SLAM is of great practical importance in many navigation and exploration problems (Burgard et al. 2000; Simmons et al. 2000). Today's most widely used on-line algorithms are based on the extended Kalman filter (EKF), whose application to SLAM problems was developed in a series of seminal papers (Smith and Cheeseman 1985; Moutarlier and Chatila 1989; Smith, Self, and

693

Cheeseman 1990). The EKF calculates a Gaussian posterior over the locations of environmental features and the robot itself. The estimation of such a joint posterior probability distribution solves one of the most difficult aspects of the SLAM problem, namely the fact that the errors in the estimates of features in the map are mutually dependent, by virtue of the fact that they are acquired through a moving platform with inaccurate positioning. Unfortunately, maintaining a Gaussian posterior imposes a significant burden on the memory and space requirements of the EKF. The covariance matrix of the Gaussian posterior requires space quadratic in the size of the map, and the basic update algorithm for EKFs requires quadratic time per measurement update. This quadratic space and time requirement imposes severe scaling limitations. In practice, EKFs can only handle maps that contain a few hundred features. In many application domains, it is desirable to acquire maps that are orders of magnitude larger (Julier and Uhlmann 2000).

This limitation has long been recognized, and a number of approaches exist that represent the posterior in a more structured way; some of those will be discussed in detail towards the end of the paper. Possibly the most popular idea is to decompose the map into collections of smaller, more manageable submaps (Leonard and Feder 1999; Guivant and Nebot 2001; Bosse, Leonard, and Teller 2002; Tardós et al. 2002; Williams and Dissanayake 2002), thereby gaining representational and computational efficiency. An alternative structured representation effectively estimates posteriors over entire paths (along with the map), not just the current robot pose. This makes it possible to exploit a conditional independence that is characteristic of the SLAM problem, which in turn leads to a factored representation (Murphy 2000; Montemerlo et al. 2002; 2003). Most of these structured techniques are approximate, and most of them require memory linear in the size of the map. Some can update the posterior in constant time, whereas others maintain quadratic complexity at a much reduced constant factor.

In this paper we describe a SLAM algorithm that represents map posterior by relative information between features in the map, and between the map and the robot's pose. This idea is not new; in fact, it is at the core of recent algorithms by Newman (2000), Csorba (1997) and Deans and Hebert (2000), and it is related to an algorithm by Lu and Milios (1997). Just as in recent work by Nettleton, Gibbens, and Durrant-Whyte (2000), our approach is based on the well-known information form of the EKF, also known as the extended information filter (EIF; Maybeck 1979). This filter maintains an information matrix, instead of the common covariance matrix. The main contribution of this paper is an EIF that maintains a sparse information matrix, called the sparse extended information filter (SEIF). This sparse matrix defines a Web-like network of local relative constraints between pairs of adjacent features in the map, reminiscent of a Gaussian Markov random field (Weiss and Freeman 2001). The sparsity has important ramifications on the computational properties of solving SLAM problems.

The use of sparse matrices, or local links, is motivated by a key insight: the posterior distribution in SLAM problems is dominated by a small number of relative links between adjacent features in the map. This is best illustrated through an example. Figure 1 shows the result of the vanilla EKF (Smith and Cheeseman 1985; Moutarlier and Chatila 1989; Smith, Self, and Cheeseman 1990) applied to the SLAM problem, in an environment containing 50 landmarks. The left panel shows a moving robot, along with its probabilistic estimate of the location of all 50 point features. The central information maintained by the EKF solution is a covariance matrix of these different estimates. The normalized covariance, i.e., the correlation, is visualized in the center panel of this figure. Each of the two axes lists the robot pose ($x-y$ location and orientation) followed by the $x-y$ locations of the 50 landmarks. Dark entries indicate strong correlations. It is known that in the limit of SLAM, all x-coordinates and all y-coordinates become fully correlated (Dissanayake et al. 2001). The checkerboard appearance of the correlation matrix illustrates this fact. Maintaining these cross-correlations—of which there are quadratically many in the number of features in the map—are essential to the SLAM problem. This observation has given rise to the (false) suspicion that on-line SLAM inherently requires update time quadratic in the number of features in the map.

The key insight that underlies SEIF is shown in the right panel of Figure 1. Shown there is the inverse covariance matrix (also known as the information matrix; Maybeck 1979; Nettleton, Gibbens, and Durrant-Whyte 2000), normalized just like the correlation matrix. Elements in this normalized information matrix can be thought of as constraints, or links, which constrain the relative locations of pairs of features in the map; the darker an entry in the display, the stronger the link. As this depiction suggests, the normalized information matrix appears to be naturally sparse: it is dominated by a small number of strong links; and it possesses a large number of links whose values, when normalized, are near zero. Furthermore, the strength of each link is related to the distance of the corresponding features: strong links are found only between metrically nearby features. The more distant two features, the weaker their link. As will become more obvious in the paper, this sparseness is not coincidental; rather, it directly relates to the way information is acquired in SLAM. This observation suggest that the EKF solution to SLAM can indeed be approximated using a sparse representation—despite the fact that the correlation matrix is densely populated. In particular, while any two features are fully correlated in the limit, the correlation arises mainly through a network of local links, which only connect nearby features. It is important to notice that this structure naturally emerges in SLAM; the results in Figure 1 are obtained using the vanilla EKF algorithm in Smith and Cheeseman (1986).
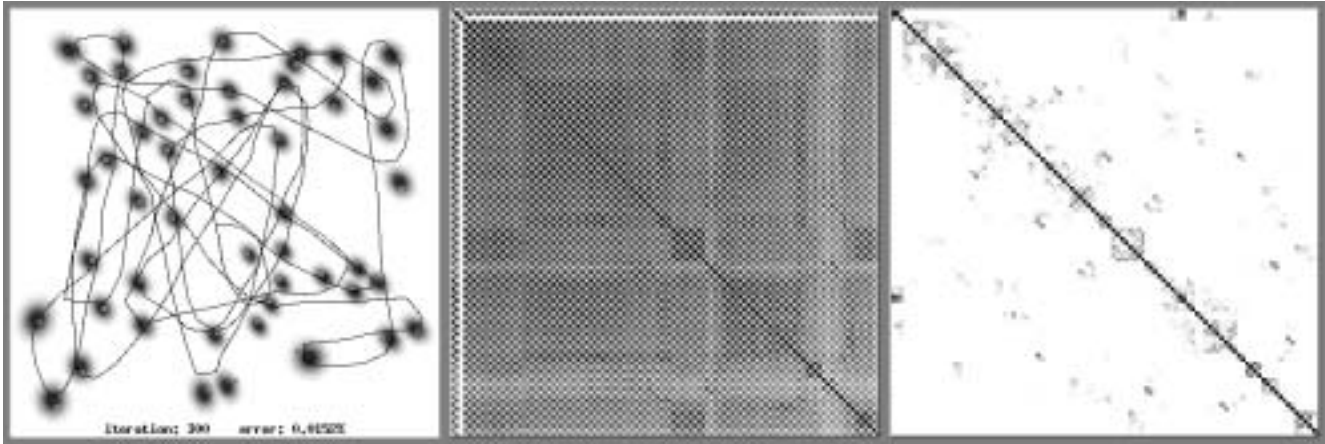
Fig. 1. Typical snapshots of EKFs applied to the SLAM problem. Shown here is a map (left panel), a correlation (center panel), and a normalized information matrix (right panel). Notice that the normalized information matrix is naturally almost sparse, motivating our approach of using sparse information matrices in SLAM.
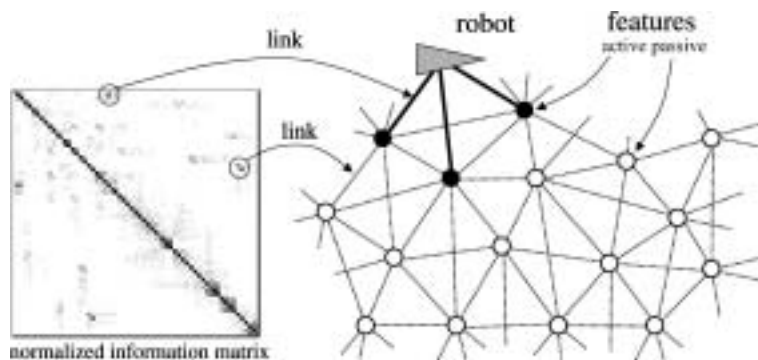

Fig. 2. Illustration of the network of features generated by our approach. Shown on the left is a sparse information matrix, and on the right a map in which entities are linked whose information matrix element is non-zero. As argued in the paper, the fact that not all features are connected is a key structural element of the SLAM problem, and at the heart of our constant time solution.

As noted above, our approach exploits this insight by maintaining a sparse information matrix, in which only nearby features are linked through a non-zero element. The resulting network structure is illustrated in the right panel of Figure 2, where disks correspond to point features and dashed arcs to links, as specified in the information matrix visualized on the left. This diagram also shows the robot, which is linked to a small subset of all features only. Those features are called active features and are drawn in black. Storing a sparse information matrix requires space linear in the number of features in the map. More importantly, all essential updates in SLAM can be performed in constant time, regardless of the number of features in the map. This result is somewhat surprising, as a naive implementation of motion updates in information filters

requires inversion of the entire information matrix, which is an $O(N^3)$ operation; plain EKFs, in comparison, require $O(N^2)$ time (for the perceptual update).

The remainder of this paper is organized as follows. In Section 2 we formally introduce the EIF, which forms the basis of our approach. SEIFs are described in Section 3, which states the major computational results of this paper. In this section we develop the constant time algorithm for maintaining sparse information matrices, and we also present an amortized constant time algorithm for recovering a global map from the relative information in the SEIF. The important issue of data association finds its treatment in Section 4, which describes a constant time technique for calculating local probabilities necessary to make data association decisions. Experimental

results are provided in Section 5. We specifically compare our new approach to the EKF solution, using a benchmark data set collected in an outdoor environment (Dissanayake et al. 2001; Guivant and Nebot 2001). These results suggest that the sparseness constraint introduces only very small errors in the resulting maps, when compared to the computationally more cumbersome EKF solution. The paper is concluded by a literature review in Section 6 and a discussion of open research issues in Section 7.

2. Extended Information Filters

In this section we review the EIF, which forms the basis of our work. EIFs are computationally equivalent to EKFs, but they represent information differently: instead of maintaining a covariance matrix, the EIF maintains an inverse covariance matrix, also known as information matrix. EIFs have previously been applied to the SLAM problem, most notably by Nettleton et al. (2000) and Nettleton, Gibbens, and Durrant-Whyte (2000), but they are much less common than the EKF approach.

Most of the material in this section applies equally to linear and nonlinear filters. We have chosen to present all material in the more general nonlinear form, since robots are inherently nonlinear. The linear form is easily obtained as a special case.

2.1. Information Form of the SLAM Problem

Let x_t denote the pose of the robot at time t. For rigid mobile robots operating in a planar environment, the pose is given by its two Cartesian coordinates and the robot's heading direction. Let N denote the number of features (e.g., landmarks) in the environment. The variable y_n with $1 \leq n \leq N$ denotes the pose of the nth feature. For example, for point landmarks in the plane, y_n may comprise the two-dimensional Cartesian coordinates of this landmark. In SLAM, it is usually assumed that features do not change their location over time; see Hähnel et al. (2003c) and Wang, Thorpe, and Thrun (2003) for a treatment of SLAM in dynamic environments.

The robot pose x_t and the set of all feature locations Y together constitute the state of the environment. It will be denoted by the vector $\xi_t = \begin{pmatrix} x_t & y_1 & \ldots & y_N \end{pmatrix}^{\mathrm{T}}$, where the superscript "T" refers to the transpose of a vector.

In the SLAM problem, it is impossible to sense the state ξ_t directly—otherwise there would be no mapping problem. Instead, the robot seeks to recover a probabilistic estimate of ξ_t. Written in a Bayesian form, our goal shall be to calculate a posterior distribution over the state ξ_t. This posterior $p(\xi_t \mid z^t, u^t)$ is conditioned on past sensor measurements $z^t = z_1, \ldots, z_t$ and past controls $u^t = u_1, \ldots, u_t$. Sensor measurements z_t might, for example, specify the approximate range and bearing to nearby features. Controls u_t specify the robot motion command asserted in the time interval $(t-1; t]$.

Following the rich EKF tradition in the SLAM literature, our approach represents the posterior $p(\xi_t \mid z^t, u^t)$ by a multivariate Gaussian distribution over the state ξ_t. The mean of this distribution will be denoted μ_t, and covariance matrix Σ_t:

$$p(\xi_t \mid z^t, u^t) \quad \propto \quad \exp\left\{-\tfrac{1}{2}(\xi_t - \mu_t)^{\mathrm{T}} \Sigma_t^{-1} (\xi_t - \mu_t)\right\}. \quad (1)$$

The proportionality sign replaces a constant normalizer that is easily recovered from the covariance Σ_t. The representation of the posterior via the mean μ_t and the covariance matrix Σ_t is the basis of the EKF solution to the SLAM problem (and to EKFs in general).

Information filters represent the same posterior through a so-called information matrix H_t and an information vector b_t—instead of μ_t and Σ_t. These are obtained by multiplying out the exponent of eq. (1):

$$\begin{aligned} p(\xi_t \mid z^t, u^t) &\propto \exp\left\{-\tfrac{1}{2}\left[\xi_t^{\mathrm{T}} \Sigma_t^{-1} \xi_t - 2\mu_t^{\mathrm{T}} \Sigma_t^{-1} \xi_t + \mu_t^{\mathrm{T}} \Sigma_t^{-1} \mu_t\right]\right\} \\ &= \exp\left\{-\tfrac{1}{2}\xi_t^{\mathrm{T}} \Sigma_t^{-1} \xi_t + \mu_t^{\mathrm{T}} \Sigma_t^{-1} \xi_t - \tfrac{1}{2}\mu_t^{\mathrm{T}} \Sigma_t^{-1} \mu_t\right\}. \end{aligned} \quad (2)$$

We now observe that the last term in the exponent, $-\tfrac{1}{2}\mu_t^{\mathrm{T}} \Sigma_t^{-1} \mu_t$ does not contain the free variable ξ_t and hence can be subsumed into the constant normalizer. This gives us the form:

$$\propto \quad \exp\{-\tfrac{1}{2}\xi_t^{\mathrm{T}} \underbrace{\Sigma_t^{-1}}_{=:H_t} \xi_t + \underbrace{\mu_t^{\mathrm{T}} \Sigma_t^{-1}}_{=:b_t} \xi_t\}. \quad (3)$$

The information matrix H_t and the information vector b_t are now defined as indicated:

$$H_t = \Sigma_t^{-1} \qquad \text{and} \qquad b_t = \mu_t^{\mathrm{T}} H_t. \quad (4)$$

Using these notations, the desired posterior can now be represented in what is commonly known as the information form of the Kalman filter:

$$p(\xi_t \mid z^t, u^t) \quad \propto \quad \exp\left\{-\tfrac{1}{2}\xi_t^{\mathrm{T}} H_t \xi_t + b_t \xi_t\right\}. \quad (5)$$

As the reader may easily notice, both representations of the multi-variate Gaussian posterior are functionally equivalent (with the exception of certain degenerate cases): the EKF representation of the mean μ_t and covariance Σ_t, and the EIF representation of the information vector b_t and the information matrix H_t. In particular, the EKF representation can be "recovered" from the information form via the following algebra:

$$\Sigma_t = H_t^{-1} \qquad \text{and} \qquad \mu_t = H_t^{-1} b_t^{\mathrm{T}} = \Sigma_t b_t^{\mathrm{T}}. \quad (6)$$

The advantage of the EIF over the EKF will become apparent further below, when the concept of sparse EIFs will be introduced.

Of particular interest will be the geometry of the information matrix. This matrix is symmetric and positive-definite:

$$H_t = \begin{pmatrix} H_{x_t,x_t} & H_{x_t,y_1} & \cdots & H_{x_t,y_N} \\ H_{y_1,x_t} & H_{y_1,y_1} & \cdots & H_{y_1,y_N} \\ \vdots & \vdots & \ddots & \vdots \\ H_{y_N,x_t} & H_{y_N,y_1} & \cdots & H_{y_N,y_N} \end{pmatrix}. \quad (7)$$

Each element in the information matrix constrains one (on the main diagonal) or two (off the main diagonal) elements in the state vector. We will refer to the off-diagonal elements as "links": the matrices H_{x_t,y_n} link together the robot pose estimate and the location estimate of a specific feature, and the matrices $H_{y_n,y_{n'}}$ for $n \neq n'$ link together two feature locations y_n and $y_{n'}$. Although rarely made explicit, the manipulation of these links is the very essence of Gaussian solutions to the SLAM problem. It will be an analysis of these links that ultimately leads to a constant-time solution to the SLAM problem.

2.2. Measurement Updates

In SLAM, measurements z_t carry spatial information on the relation of the robot's pose and the location of a feature. For example, z_t might be the approximate range and bearing to a nearby feature. Without loss of generality, we will assume that each measurement z_t corresponds to exactly one feature in the map. Sightings of multiple features at the same time may easily be processed one-after-another.

Figure 3 illustrates the effect of measurements on the information matrix H_t. Suppose the robot measures the approximate range and bearing to the feature y_1, as illustrated in Figure 3(a). This observation links the robot pose x_t to the location of y_1. The strength of the link is given by the level of noise in the measurement. Updating EIFs based on this measurement involves the manipulation of the off-diagonal elements $H_{x_t,y}$ and their symmetric counterparts H_{y,x_t} that link together x_t and y. Additionally, the on-diagonal elements H_{x_t,x_t} and H_{y_1,y_1} are also updated. These updates are additive. Each observation of a feature y increases the strength of the total link between the robot pose and this very feature, and with it the total information in the filter. Figure 3(b) shows the incorporation of a second measurement of a different feature, y_2. In response to this measurement, the EIF updates the links $H_{x_t,y_2} = H_{y_2,x_t}^{\mathrm{T}}$ (and H_{x_t,x_t} and H_{y_2,y_2}). As this example suggests, measurements introduce links only between the robot pose x_t and observed features. Measurements never generate links between pairs of features, or between the robot and unobserved features.

For a mathematical derivation of the update rule, we observe that Bayes rule enables us to factor the desired posterior into the following product:

$$\begin{aligned} p(\xi_t \mid z^t, u^t) &\propto p(z_t \mid \xi_t, z^{t-1}, u^t) \, p(\xi_t \mid z^{t-1}, u^t) \\ &= p(z_t \mid \xi_t) \, p(\xi_t \mid z^{t-1}, u^t). \end{aligned} \quad (8)$$

The second step of this derivation exploited common (and obvious) independences in SLAM problems (Thrun 2002). For the time being, we assume that $p(\xi_t \mid z^{t-1}, u^t)$ is represented by $\bar{H}_t$ and $\bar{b}_t$. These will be discussed in the next section, where robot motion will be addressed. The key question addressed in this section thus concerns the representation of the probability distribution $p(z_t \mid \xi_t)$ and the mechanics of carrying out the multiplication above. In the "extended" family of filters, a common model of robot perception is one in which measurements are governed via a deterministic non-linear measurement function h with added Gaussian noise:

$$z_t = h(\xi_t) + \varepsilon_t. \quad (9)$$

Here ε_t is an independent noise variable with zero mean, whose covariance will be denoted Z. Put into probabilistic terms, eq. (9) specifies a Gaussian distribution over the measurement space of the form

$$p(z_t \mid \xi_t) \propto \exp\left\{-\tfrac{1}{2}(z_t - h(\xi_t))^{\mathrm{T}} Z^{-1}(z_t - h(\xi_t))\right\}. \quad (10)$$

Following the rich literature of EKFs, EIFs approximate this Gaussian by linearizing the measurement function h. More specifically, a Taylor series expansion of h gives us

$$h(\xi_t) \approx h(\mu_t) + \nabla_\xi h(\mu_t)[\xi_t - \mu_t], \quad (11)$$

where $\nabla_\xi h(\mu_t)$ is the first derivative (Jacobian) of h with respect to the state variable ξ, taken $\xi = \mu_t$. For brevity, we will write $\hat{z}_t = h(\mu_t)$ to indicate that this is a prediction given our state estimate μ_t. The transpose of the Jacobian matrix $\nabla_\xi h(\mu_t)$ and will be denoted C_t. With these definitions, eq. (11) reads as follows:

$$h(\xi_t) \approx \hat{z}_t + C_t^{\mathrm{T}}(\xi_t - \mu_t). \quad (12)$$

This approximation leads to the following Gaussian approximation of the measurement density in eq. (10):

$$\begin{aligned} p(z_t \mid \xi_t) \propto \exp\Big\{-\tfrac{1}{2}(z_t - \hat{z}_t - C_t^{\mathrm{T}}\xi_t + C_t^{\mathrm{T}}\mu_t)^{\mathrm{T}} \\ Z^{-1}(z_t - \hat{z}_t - C_t^{\mathrm{T}}\xi_t + C_t^{\mathrm{T}}\mu_t)\Big\}. \end{aligned} \quad (13)$$

Multiplying out the exponent and regrouping the resulting terms gives us

$$= \exp\Big\{-\tfrac{1}{2}\xi_t^{\mathrm{T}} C_t Z^{-1} C_t^{\mathrm{T}}\xi_t + (z_t - \hat{z}_t + C_t^{\mathrm{T}}\mu_t)^{\mathrm{T}} Z^{-1} C_t^{\mathrm{T}}\xi_t \quad (14)$$

$$-\tfrac{1}{2}(z_t - \hat{z}_t + C_t^{\mathrm{T}}\mu_t)^{\mathrm{T}} Z^{-1}(z_t - \hat{z}_t + C_t^{\mathrm{T}}\mu_t)\Big\}.$$

As before, the final term in the exponent does not depend on the variable ξ_t and hence can be subsumed into the proportionality factor:

$$\propto \exp\Big\{-\tfrac{1}{2}\xi_t^{\mathrm{T}} C_t Z^{-1} C_t^{\mathrm{T}}\xi_t + (z_t - \hat{z}_t + C_t^{\mathrm{T}}\mu_t)^{\mathrm{T}} Z^{-1} C_t^{\mathrm{T}}\xi_t\Big\}. \quad (15)$$
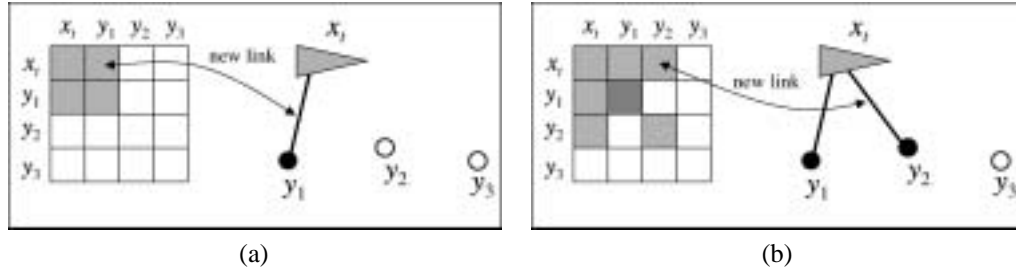
Fig. 3. The effect of measurements on the information matrix and the associated network of features. (a) Observing y_1 results in a modification of the information matrix elements H_{x_t,y_1}. (b) Similarly, observing y_2 affects H_{x_t,y_2}. Both updates can be carried out in constant time.

We are now in the position to state the measurement update equation, which implements the probabilistic law (8).

$$p(\xi_t \mid z^t, u^t) \; \propto \; \exp\left\{-\tfrac{1}{2}\xi_t^{\mathrm{T}} \bar{H}_t \xi_t + \bar{b}_t \xi_t\right\}$$

$$\cdot \exp\left\{-\tfrac{1}{2}\xi_t^{\mathrm{T}} C_t Z^{-1} C_t^{\mathrm{T}} \xi_t + (z_t - \hat{z}_t + C_t^{\mathrm{T}} \mu_t)^{\mathrm{T}} Z^{-1} C_t^{\mathrm{T}} \xi_t\right\}$$

$$= \; \exp\{-\tfrac{1}{2}\xi_t^{\mathrm{T}}(\underbrace{\bar{H}_t + C_t Z^{-1} C_t^{\mathrm{T}}}_{H_t})\xi_t$$

$$+ (\underbrace{\bar{b}_t + (z_t - \hat{z}_t + C_t^{\mathrm{T}} \mu_t)^{\mathrm{T}} Z^{-1} C_t^{\mathrm{T}}}_{b_t})\xi_t\}. \tag{16}$$

Thus, the measurement update of the EIF is given by the following additive rule:

$$H_t \;=\; \bar{H}_t + C_t Z^{-1} C_t^{\mathrm{T}} \tag{17}$$

$$b_t \;=\; \bar{b}_t + (z_t - \hat{z}_t + C_t^{\mathrm{T}} \mu_t)^{\mathrm{T}} Z^{-1} C_t^{\mathrm{T}}. \tag{18}$$

In the general case, these updates may modify the entire information matrix H_t and vector b_t, respectively. A key observation of all SLAM problems is that the Jacobian C_t is sparse. In particular, C_t is zero except for the elements that correspond to the robot pose x_t and the feature y_t observed at time t.

$$C_t \;=\; \left(\; \tfrac{\partial h}{\partial x_t} \quad 0 \cdots 0 \quad \tfrac{\partial h}{\partial y_t} \quad 0 \cdots 0 \;\right)^{\mathrm{T}}. \tag{19}$$

This well-known sparseness of C_t (Dissanayake et al. 2001) is due to the fact that measurements z_t are only a function of the relative distance and orientation of the robot to the observed feature. As a pleasing consequence, the update $C_t Z^{-1} C_t^{\mathrm{T}}$ to the information matrix in eq. (17) is only non-zero in four places: the off-diagonal elements that link the robot pose x_t with the observed feature y_t, and the main-diagonal elements that correspond to x_t and y_t. Thus, the update equations (17) and (18) are well in tune with our intuitive description given at the beginning of this section, where we argued that measurements only strengthen the links between the robot pose and observed features, in the information matrix.

To compare this to the EKF solution, we notice that even though the change of the information matrix is local, the resulting covariance usually changes in non-local ways. Put differently, the difference between the old covariance $\bar{\Sigma}_t = \bar{H}_t^{-1}$

and the new covariance matrix $\Sigma_t = H_t^{-1}$ is usually non-zero everywhere.

2.3. Motion Updates

The second important step of SLAM concerns the update of the filter in accordance to robot motion. In the standard SLAM problem, only the robot pose changes over time. The environment is static.

The effect of robot motion on the information matrix H_t is slightly more complicated than that of measurements. Figure 4(a) illustrates an information matrix and the associated network before the robot moves, in which the robot is linked to two (previously observed) features. If robot motion was free of noise, this link structure would not be affected by robot motion. However, the noise in robot actuation weakens the link between the robot and all active features. Hence H_{x_t,y_1} and H_{x_t,y_2} are decreased by a certain amount. This decrease reflects the fact that the noise in motion induces a loss of information of the relative location of the features to the robot. Not all of this information is lost, however. Some of it is shifted into between-feature links H_{y_1,y_2}, as illustrated in Figure 4(b). This reflects the fact that even though the motion induced a loss of information of the robot relative to the features, no information was lost between individual features. Robot motion thus has the effect that features that were indirectly linked through the robot pose become linked directly.

To derive the update rule, we begin with a Bayesian description of robot motion. Updating a filter based on robot motion motion involves the calculation of the following posterior:

$$p(\xi_t \mid z^{t-1}, u^t) = \int p(\xi_t \mid \xi_{t-1}, z^{t-1}, u^t) \; p(\xi_{t-1} \mid z^{t-1}, u^t)$$

$$\mathrm{d}\xi_{t-1}. \tag{20}$$

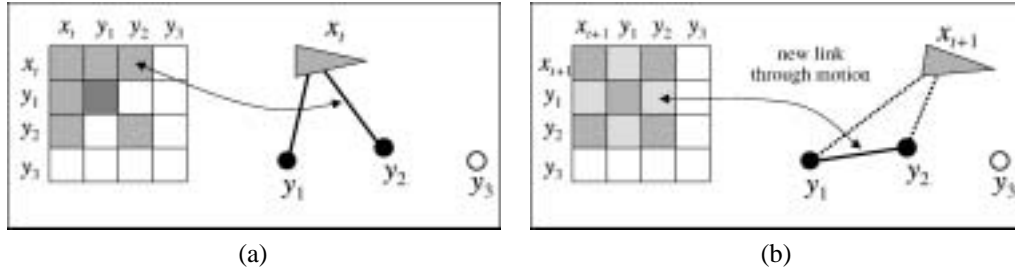Exploiting the common SLAM independences (Thrun 2002) leads to

Fig. 4. The effect of motion on the information matrix and the associated network of features: (a) before motion; (b) after motion. If motion is non-deterministic, motion updates introduce new links (or reinforce existing links) between any two active features, while weakening the links between the robot and those features. This step introduces links between pairs of features.

$$p(\xi_t \mid z^{t-1}, u^t) = \int p(\xi_t \mid \xi_{t-1}, u_t)\, p(\xi_{t-1} \mid z^{t-1}, u^{t-1})$$

$$\mathrm{d}\xi_{t-1}. \tag{21}$$

The term $p(\xi_{t-1} \mid z^{t-1}, u^{t-1})$ is the posterior at time $t - 1$, represented by H_{t-1} and b_{t-1}. Our concern will therefore be with the remaining term $p(\xi_t \mid \xi_{t-1}, u_t)$, which characterizes robot motion in probabilistic terms.

Similar to the measurement model above, it is common practice to model robot motion by a nonlinear function with added independent Gaussian noise:

$$\xi_t = \xi_{t-1} + \Delta_t \text{ with } \Delta_t = g(\xi_{t-1}, u_t) + S_x \delta_t. \tag{22}$$

Here g is the motion model, a vector-valued function which is non-zero only for the robot pose coordinates, as feature locations are static in SLAM. The term labeled Δ_t constitutes the state change at time t. The stochastic part of this change is modeled by δ_t, a Gaussian random variable with zero mean and covariance U_t. This Gaussian variable is a low-dimensional variable defined for the robot pose only. Here S_x is a projection matrix of the form $S_x = (\, I \;\; 0 \;\ldots\; 0\,)^{\mathrm{T}}$, where I is an identity matrix of the same dimension as the robot pose vector x_t and as of δ_t. Each 0 in this matrix refers to a null matrix, of which there are N in S_x. The product $S_x \delta_t$, hence, gives the following generalized noise variable, enlarged to the dimension of the full state vector ξ: $S_x \delta_t = (\, \delta_t \;\; 0 \;\ldots\; 0\,)^{\mathrm{T}}$. In EIFs, the function g in eq. (22) is approximated by its first degree Taylor series expansion:

$$\begin{aligned} g(\xi_{t-1}, u_t) &\approx g(\mu_{t-1}, u_t) + \nabla_\xi g(\mu_{t-1}, u_t)[\xi_{t-1} - \mu_{t-1}] \\ &= \hat{\Delta}_t + A_t \xi_{t-1} - A_t \mu_{t-1}. \end{aligned} \tag{23}$$

Here $A_t = \nabla_\xi g(\mu_{t-1}, u_t)$ is the derivative of g with respect to ξ at $\xi = \mu_{t-1}$ and u_t. The symbol $\hat{\Delta}_t$ is short for the predicted motion effect, $g(\mu_{t-1}, u_t)$. Plugging this approximation into

eq. (22) leads to an approximation of ξ_t, the state at time t:

$$\xi_t \approx (I + A_t)\xi_{t-1} + \hat{\Delta}_t - A_t \mu_{t-1} + S_x \delta_t. \tag{24}$$

Hence, under this approximation the random variable ξ_t is again Gaussian distributed. Its mean is obtained by replacing ξ_t and δ_t in eq. (24) by their respective means:

$$\begin{aligned} \bar{\mu}_t &= (I + A_t)\mu_{t-1} + \hat{\Delta}_t - A_t \mu_{t-1} + S_x 0 \\ &= \mu_{t-1} + \hat{\Delta}_t. \end{aligned} \tag{25}$$

The covariance of ξ_t is simply obtained by scaled and adding the covariance of the Gaussian variables on the right-hand side of eq. (24):

$$\begin{aligned} \bar{\Sigma}_t &= (I + A_t)\Sigma_{t-1}(I + A_t)^{\mathrm{T}} + 0 - 0 + S_x U_t S_x^{\mathrm{T}} \\ &= (I + A_t)\Sigma_{t-1}(I + A_t)^{\mathrm{T}} + S_x U_t S_x^{\mathrm{T}}. \end{aligned} \tag{26}$$

Update equations (25) and (26) are in the EKF form, i.e., they are defined over means and covariances. The information form is now easily recovered from the definition of the information form in eq. (4) and its inverse in eq. (6). In particular, we have

$$\begin{aligned} \bar{H}_t &= \bar{\Sigma}_t^{-1} = \left[(I + A_t)\Sigma_{t-1}(I + A_t)^{\mathrm{T}} + S_x U_t S_x^{\mathrm{T}} \right]^{-1} \\ &= \left[(I + A_t)H_{t-1}^{-1}(I + A_t)^{\mathrm{T}} + S_x U_t S_x^{\mathrm{T}} \right]^{-1} \end{aligned} \tag{27}$$

$$\begin{aligned} \bar{b}_t &= \bar{\mu}_t^{\mathrm{T}} \bar{H}_t = \left[\mu_{t-1} + \hat{\Delta}_t \right]^{\mathrm{T}} \bar{H}_t = \left[H_{t-1}^{-1} b_{t-1}^{\mathrm{T}} + \hat{\Delta}_t \right]^{\mathrm{T}} \bar{H}_t \\ &= \left[b_{t-1} H_{t-1}^{-1} + \hat{\Delta}_t^{\mathrm{T}} \right] \bar{H}_t. \end{aligned} \tag{28}$$

These equations appear computationally involved, in that they require the inversion of large matrices. In the general case, the complexity of the EIF is therefore cubic in the size of the state space. In the next section, we provide the surprising result that both $\bar{H}_t$ and $\bar{b}_t$ can be computed in constant time if H_{t-1} is sparse.

3. Sparse Extended Information Filters

The central, new algorithm presented in this paper is the SEIF. The SEIF differs from the EIF described in the previous section in that it maintains a sparse information matrix. An information matrix H_t is considered sparse if the number of links to the robot and to each feature in the map is bounded by a constant that is independent of the number of features in the map. The bound for the number of links between the robot pose and other features in the map will be denoted θ_x; the bound on the number of links for each feature (not counting the link to the robot) will be denoted θ_y. The motivation for maintaining a sparse information is mainly computational, as will become apparent below. Its justification has already been discussed above, when we demonstrated that in SLAM the normalized information matrix is already almost sparse. This suggests that by enforcing sparseness, the induced approximation error is small.

3.1. Constant Time Results

We begin by proving three important constant time results, which form the backbone of SEIFs. All proofs can be found in the Appendix.

LEMMA 1.   The measurement update in Section 2.2 requires constant time, irrespective of the number of features in the map.

This lemma ensures that measurements can be incorporated in constant time. Notice that this lemma does not require sparseness of the information matrix; rather, it is a well-known property of information filters in SLAM.

Less trivial is the following lemma.

LEMMA 2.   If the information matrix is sparse and $A_t = 0$, the motion update in Section 2.3 requires constant time. The constant-time update equations are given by

$$
\begin{aligned}
L_t &= S_x[U_t^{-1} + S_x^{\mathrm{T}} H_{t-1} S_x]^{-1} S_x^{\mathrm{T}} H_{t-1} \\
\bar{H}_t &= H_{t-1} - H_{t-1} L_t \qquad\qquad (29) \\
\bar{b}_t &= b_{t-1} + \hat{\Delta}_t^{\mathrm{T}} H_{t-1} - b_{t-1} L_t + \hat{\Delta}_t^{\mathrm{T}} H_{t-1} L_t.
\end{aligned}
$$

This result addresses the important special case $A_t = 0$, i.e., the Jacobian of pose change with respect to the absolute robot pose is zero. This is the case for robots with linear mechanics, and with nonlinear mechanics where there is no "cross-talk" between absolute coordinates and the additive change due to motion.

In general, $A_t \neq 0$, since the x–y update depends on the robot orientation. This case is addressed by the next lemma.

LEMMA 3.   If the information matrix is sparse, the motion update in Section 2.3 requires constant time if the mean μ_t is available for the robot pose and all active features. The constant-time update equations are given by

$$
\begin{aligned}
\Psi_t &= I - S_x(I + [S_x^{\mathrm{T}} A_t S_x]^{-1})^{-1} S_x^{\mathrm{T}} \\
H'_{t-1} &= \Psi_t^{\mathrm{T}} H_{t-1} \Psi_t \\
\Delta H_t &= H'_{t-1} S_x[U_t^{-1} + S_x^{\mathrm{T}} H'_{t-1} S_x]^{-1} S_x^{\mathrm{T}} H'_{t-1} \\
\bar{H}_t &= H'_{t-1} - \Delta H_t \\
\bar{b}_t &= b_{t-1} - \mu_{t-1}^{\mathrm{T}}(\Delta H_t - H_{t-1} + H'_{t-1}) + \hat{\Delta}_t^{\mathrm{T}} \bar{H}_t \quad (30)
\end{aligned}
$$

For $A_t \neq 0$, a constant-time update requires knowledge of the mean μ_{t-1} before the motion command, for the robot pose and all active features (but not the passive features). This information is not maintained by the standard information filter, and extracting it in the straightforward way (via eq. (6)) requires more than constant time. A constant-time solution to this problem will now be presented.

3.2. Sparsification

3.2.1. General Idea

The final step in SEIFs concerns the sparsification of the information matrix H_t. Because sparsification is so essential to SEIFs, let us first discuss it in general terms before we apply it to the information filter. Sparsification is an approximation whereby a posterior distribution is approximated by two of its marginals. Suppose a, b, and c are sets of random variables (b is not to be confused with the information vector b_t). Suppose we are given a joint distribution $p(a, b, c)$ over these variables. To sparsify this distribution, suppose we would like to remove any direct link between the variables a and b. In other words, we would like to approximate p by a distribution $\tilde{p}$ for which the following property holds: $\tilde{p}(a \mid b, c) = p(a \mid c)$ and $\tilde{p}(b \mid a, c) = p(b \mid c)$. This conditional independence is commonly known as "d-separation" (Pearl 1988). In multivariate Gaussians, it is easily shown that d-separation is equivalent to the absence of a direct link between a and b, i.e., the corresponding element in the information matrix is zero.

A good approximation $\tilde{p}$ is obtained by a term proportional to the product of the marginals, $p(a, c)$ and $p(b, c)$. Neither of these marginals retain dependence between the variables a and b, since they both contain only one of those variables. Thus, the product $p(a, c)\, p(b, c)$ does not contain any direct dependences between a and b; instead, a and b are d-separated by c. However, $p(a, c)\, p(b, c)$ is not yet a valid probability distribution over a, b, and c. This is because c occurs twice in this expression. However, proper normalization by $p(c)$ yields a probability distribution (assuming $p(c) > 0$):

$$
\tilde{p}(a, b, c) = \frac{p(a, c)\, p(b, c)}{p(c)} \qquad (31)
$$

To understand the effect of this approximation, we apply the following transformation:

$$\tilde{p}(a, b, c) = \frac{p(a, b, c)}{p(a, b, c)} \frac{p(a, c)\ p(b, c)}{p(c)}$$

$$= p(a, b, c) \frac{p(a, c)}{p(c)} \frac{p(b, c)}{p(a, b, c)}$$

$$= p(a, b, c) \frac{p(a \mid c)}{p(a \mid b, c)}. \tag{32}$$

In other words, removing the direct dependence between a and b is equivalent to approximating the conditional $p(a \mid b, c)$ by a conditional $p(a \mid c)$. We also note (without proof) that among all approximations q of p in which c d-separates a and b, the one described here is "closest" to p, where closeness is measured by the Kullback Liebler divergence, a common information-theoretic measure of the "nearness" of probability distributions (see Cover and Thomas 1991, for a definition and discussion of KL divergence):

$$\tilde{p} = \operatorname{argmin}_q D(q \mid\mid p). \tag{33}$$

An important observation pertains to the fact that the original $p(a \mid b, c)$ is at least as informative as $p(a \mid c)$; the conditional hat replaces $p(a \mid b, c)$ in $\tilde{p}$. This is because $p(a \mid b, c)$ is conditioned on a superset of variables of the conditioning variables in $p(a \mid c)$. For Gaussians, this implies that the variance of the approximation $p(a \mid c)$ is equal or larger than the variance of the original conditional, $p(a \mid b, c)$. Further, the variances of the marginals $\tilde{p}(a)$, $\tilde{p}(b)$, and $\tilde{p}(c)$ are also larger than or equal to the corresponding variances of $p(a)$, $p(b)$, and $p(c)$. In other words, it is impossible that the variance shrinks under this approximation. Such an operation is commonly referred to as consistent in the SLAM literature (Dissanayake et al. 2001). However, we note that the consistency of a single-step update does not imply that the posterior of a sparsified Bayes filter remains consistent—a phenomenon we will discuss in detail below.

3.2.2. Application to Extended Information Filters

SEIFs apply the idea of sparsification to the posterior $p(x_t, Y \mid z^t, u^t)$, thereby maintaining a matrix H_t that is sparse at all times. This sparseness is at the core of SEIF's efficiency. Sparsification is necessarily an approximative step, since information matrices in SLAM are naturally not sparse—even though normalized information matrices tend to be almost sparse. In the context of SLAM, it suffices to remove links (deactivate) between the robot pose and individual features in the map; if done correctly, this also limits the number of links between pairs of features.

To see, let us briefly consider the two circumstances under which a new link may be introduced. First, observing a passive feature activates this feature, that is, introduces a new link between the robot pose and the very feature. Thus, measurement updates potentially violate the bound θ_x. Secondly,

motion introduces links between any two active features, and hence leads to violations of the bound θ_y. This consideration suggests that controlling the number of active features can avoid violation of both sparseness bounds.

Our sparsification technique is illustrated in Figure 5. Shown there is the situation before and after sparsification. The removal of a link in the network corresponds to setting an element in the information matrix to zero; however, this requires the manipulation of other links between the robot and other active features. The resulting network is only an approximation to the original one, whose quality depends on the magnitude of the link before removal.

To define the sparsification step, it will prove useful to partition the set of all features into three disjoint subsets

$$Y = Y^+ + Y^0 + Y^-, \tag{34}$$

where Y^+ is the set of all active features that shall remain active. Y^0 are one or more active features that we seek to deactivate (remove the link to the robot). Y^- are all currently passive features. Since $Y^+ \cup Y^0$ contains all currently active features, the posterior can be factored as follows:

$$
\begin{aligned}
p(x_t, Y \mid z^t, u^t) &= p(x_t, Y^0, Y^+, Y^- \mid z^t, u^t) \\
&= p(x_t \mid Y^0, Y^+, Y^-, z^t, u^t) \\
&\quad p(Y^0, Y^+, Y^- \mid z^t, u^t) \\
&= p(x_t \mid Y^0, Y^+, Y^- = 0, z^t, u^t) \\
&\quad p(Y^0, Y^+, Y^- \mid z^t, u^t). \tag{35}
\end{aligned}
$$

In the last step we exploit the fact that if we know the active features Y^0 and Y^+, the variable x_t does not depend on the passive features Y^-. We can hence set Y^- to an arbitrary value without affecting the conditional posterior over x_t, $p(x_t \mid Y^0, Y^+, Y^-, z^t, u^t)$. Here we simply chose $Y^- = 0$.[1]

Following the sparsification idea discussed in general terms in the previous section, we now replace $p(x_t \mid Y^0, Y^+, Y^- = 0)$ by $p(x_t \mid Y^+, Y^- = 0)$, that is, we drop the dependence on Y^0:

$$
\begin{aligned}
\tilde{p}(x_t, Y \mid z^t, u^t) &= p(x_t \mid Y^+, Y^- = 0, z^t, u^t) \\
&\quad p(Y^0, Y^+, Y^- \mid z^t, u^t). \tag{36}
\end{aligned}
$$

This approximation is obviously equivalent to the following expression:

$$
\begin{aligned}
\tilde{p}(x_t, Y \mid z^t, u^t) &= \frac{p(x_t, Y^+ \mid Y^- = 0, z^t, u^t)}{p(Y^+ \mid Y^- = 0, z^t, u^t)} \\
&\quad p(Y^0, Y^+, Y^- \mid z^t, u^t). \tag{37}
\end{aligned}
$$

3.2.3. Constant-Time Calculation

The approximate posterior $\tilde{p}$ defined in eq. (37) is now easily calculated in constant time. In particular, we begin by calculating the information matrix for the distribution $p(x_t, Y^0, Y^+ \mid$

---

1. Another choice would have been to integrate out the variables Y^-; however, the resulting sparsification requires inversions of large matrices, and numerical truncation errors may yield non-sparse matrices.

$Y^- = 0$) of all variables but Y^-, and conditioned on $Y^- = 0$. This is obtained by extracting the submatrix of all state variables but Y^-:

$$H'_t \;\; = \;\; S_{x,Y+,Y^0} S^T_{x,Y+,Y^0} H_t S_{x,Y+,Y^0} S^T_{x,Y+,Y^0}. \qquad (38)$$

With that, the matrix inversion lemma[2] leads to the following information matrices for the terms $p(x_t, Y^+ \mid Y^- = 0, z^t, u^t)$ and $p(Y^+ \mid Y^- = 0, z^t, u^t)$, denoted H^1_t and H^2_t, respectively:

$$
\begin{aligned}
H^1_t \;\; &= \;\; H'_t - H'_t S_{Y_0} (S^T_{Y_0} H'_t S_{Y_0})^{-1} S^T_{Y_0} H'_t \\
H^2_t \;\; &= \;\; H'_t - H'_t S_{x,Y_0} (S^T_{x,Y_0} H'_t S_{x,Y_0})^{-1} S^T_{x,Y_0} H'_t. \quad (40)
\end{aligned}
$$

Here the various S-matrices are projection matrices, analogous to the matrix S_x defined above. The final term in our approximation (37), $p(Y^0, Y^+, Y^- \mid z^t, u^t)$, has the following information matrix:

$$H^3_t \;\; = \;\; H_t - H_t S_{x_t} (S^T_{x_t} H_t S_{x_t})^{-1} S^T_{x_t} H_t. \qquad (41)$$

Putting these expressions together according to eq. (37) yields the following information matrix, in which the feature Y^0 is now indeed deactivated:

$$
\begin{aligned}
\tilde{H}_t \;\; &= \;\; H^1_t - H^2_t + H^3_t = H_t - H'_t S_{Y_0} (S^T_{Y_0} H'_t S_{Y_0})^{-1} S^T_{Y_0} H'_t \\
&\quad + H'_t S_{x,Y_0} (S^T_{x,Y_0} H'_t S_{x,Y_0})^{-1} S^T_{x,Y_0} H'_t \\
&\quad - H_t S_{x_t} (S^T_{x_t} H_t S_{x_t})^{-1} S^T_{x_t} H_t. \qquad (42)
\end{aligned}
$$

The resulting information vector is now obtained by the following simple consideration:

$$
\begin{aligned}
\tilde{b}_t \;\; &= \;\; \mu^T_t \tilde{H}_t = \mu^T_t (H_t - H_t + \tilde{H}_t) \\
&= \;\; \mu^T_t H_t + \mu^T_t (\tilde{H}_t - H_t) = b_t + \mu^T_t (\tilde{H}_t - H_t). \quad (43)
\end{aligned}
$$

All equations can be computed in constant time, regardless of the size of H_t. The effect of this approximation is the deactivation of the features Y^0, while introducing only new links between active features. The sparsification rule requires knowledge of the mean vector μ_t for all active features, which is obtained via the approximation technique described in the previous section. From eq. (43), it is obvious that the sparsification does not affect the mean μ_t, that is, $H^{-1}_t b^T_t = \tilde{H}^{-1}_t \tilde{b}^T_t$. However, the mean can be affected by a number of other aspects of SEIF, such as the use of an approximate H_t matrix in subsequent filter updates.

The sparsification is executed whenever a measurement update or a motion update would violate a sparseness constraint. Active features are chosen for deactivation in reverse order of the magnitude of their link. This strategy tends to deactivate features whose last sighting is furthest away in time.

---

2. The matrix inversion lemma (Sherman–Morrison–Woodbury formula), as used throughout this paper, is stated as follows:

$$\left(H^{-1} + SBS^T\right)^{-1} \;\; = \;\; H - HS \left(B^{-1} + S^T H S\right)^{-1} S^T H. \quad (39)$$

Empirically, it induces approximation errors that are negligible for appropriately chosen sparseness constraints θ_x and θ_y. In practice, our implementation constrains only θ_x. This induces a bound on the number of between-landmark links, simply because only adjacent links tend to be active at the same time. All our experiments below, thus, constrain θ_x but use $\theta_y = N$.

3.3. Amortized Approximate Map Recovery

Before deriving an algorithm for recovering the state estimate μ_t from the information form, let us briefly consider what parts of μ_t are needed in SEIFs, and when. SEIFs need the state estimate μ_t of the robot pose and the active features in the map. These estimates are needed at three different occasions: (1) the linearization of the nonlinear measurement and motion model; (2) the motion update according to Lemma 3; (3) the sparsification technique described further below. For linear systems, the means are only needed for the sparsification (third point above). We also note that we only need constantly many of the values in μ_t, namely the estimate of the robot pose and of the locations of active features.

As stated in eq. (6), the mean vector μ_t is a function of H_t and b_t:

$$\mu_t \;\; = \;\; H^{-1}_t b^T_t \;\; = \;\; \Sigma_t b^T_t. \qquad (44)$$

Unfortunately, calculating eq. (44) directly involves inverting a large matrix, which would requires more than constant time.

The sparseness of the matrix H_t allows us to recover the state incrementally. In particular, we can do so on-line, as the data are being gathered and the estimates b and H are being constructed. To do so, it will prove convenient to pose eq. (44) as an optimization problem.

LEMMA 4.   The state μ_t is the mode $\hat{v}_t := \operatorname{argmax}_{v_t} p(v_t)$ of the Gaussian distribution, defined over the variable v_t:

$$p(v_t) \;\; = \;\; \text{const.} \cdot \exp\left\{-\tfrac{1}{2} v^T_t H_t v_t + b^T_t v_t\right\}. \quad (45)$$

Here v_t is a vector of the same form and dimensionality as μ_t. This lemma suggests that recovering μ_t is equivalent to finding the mode of eq. (45). Thus, it transforms a matrix inversion problem into an optimization problem. For this optimization problem, we will now describe an iterative hill climbing algorithm which, thanks to the sparseness of the information matrix, requires only constant time per optimization update.

Our approach is an instantiation of coordinate descent. For simplicity, we state it here for a single coordinate only; our implementation iterates a constant number K of such optimizations after each measurement update step. The mode $\hat{v}_t$ of eq. (45) is attained at

$$
\begin{aligned}
\hat{v}_t \;\; &= \;\; \operatorname{argmax}_{v_t} \; p(v_t) = \operatorname{argmax}_{v_t} \; \exp\left\{-\tfrac{1}{2} v^T_t H_t v_t + b^T_t v_t\right\} \\
&= \;\; \operatorname{argmin}_{v_t} \; \tfrac{1}{2} v^T_t H_t v_t - b^T_t v_t. \qquad (46)
\end{aligned}
$$

We note that the argument of the min-operator in eq. (46) can be written in a form that makes the individual coordinate variables $v_{i,t}$ (for the ith coordinate of v_t) explicit:

$$\frac{1}{2}v_t^{\mathrm{T}} H_t v_t - b_t^{\mathrm{T}} v_t = \frac{1}{2}\sum_i \sum_j v_{i,t}^{\mathrm{T}} H_{i,j,t} v_{j,t}$$
$$- \sum_i b_{i,t}^{\mathrm{T}} v_{i,t}, \qquad (47)$$

where $H_{i,j,t}$ is the element with coordinates (i, j) in H_t, and $b_{i,t}$ if the ith component of the vector b_t. Taking the derivative of this expression with respect to an arbitrary coordinate variable $v_{i,t}$ gives us

$$\frac{\partial}{\partial v_{i,t}} \left\{ \frac{1}{2} \sum_i \sum_j v_{i,t}^{\mathrm{T}} H_{i,j,t} v_{j,t} - \sum_i b_{i,t}^{\mathrm{T}} v_{i,t} \right\}$$
$$= \sum_j H_{i,j,t} v_{j,t} - b_{i,t}^{\mathrm{T}}. \qquad (48)$$

Setting this to zero leads to the optimum of the ith coordinate variable $v_{i,t}$ given all other estimates $v_{j,t}$:

$$v_{i,t}^{[k+1]} = H_{i,i,t}^{-1} \left[b_{i,t}^{\mathrm{T}} - \sum_{j \neq i} H_{i,j,t} v_{j,t}^{[k]} \right]. \qquad (49)$$

The same expression can conveniently be written in matrix notation, where S_i is a projection matrix for extracting the ith component from the matrix H_t:

$$v_{i,t}^{[k+1]} = (S_i^{\mathrm{T}} H_t S_i)^{-1} S_i^{\mathrm{T}} \left[b_t - H_t v_t^{[k]} + H_t S_i S_i^{\mathrm{T}} v_t^{[k]} \right]. \quad (50)$$

All other estimates $v_{i',t}$ with $i' \neq i$ remain unchanged in this update step, i.e., $v_{i',t}^{[k+1]} = v_{i',t}^{[k]}$.

As is easily seen, the number of elements in the summation in eq. (49), and hence the vector multiplication in eq. (50), is constant if H_t is sparse. Hence, each update requires constant time. To maintain the constant-time property of our SLAM algorithm, we can afford a constant number of updates K per time-step. This will generally not lead to convergence, but the relaxation process takes place over multiple time-steps, resulting in small errors in the overall estimate.

4. Data Association

Data association refers to the problem of determining the correspondence between multiple sightings of identical features. Features are generally not unique in appearance, and the robot has to make decisions with regards to the identity of individual features. Data association is generally acknowledged to be a key problem in SLAM, and a number of solutions have been proposed (Dissanayake et al. 2001; Montemerlo et al. 2003; Tardós et al. 2002). Here we follow the standard maximum likelihood approach described in Dissanayake et al. (2001).

This approach requires a mechanism for evaluating the likelihood of a measurement under an alleged data association, so as to identify the association that makes the measurement most probable. The key result here is that this likelihood can be approximated tightly in constant time.

4.1. Recovering Data Association Probabilities

To perform data association, we augment the notation to make the data association variable explicit. Let n_t be the index of the measurement z_t, and let n^t be the sequence of all correspondence variables leading up to time t. The domain of n_t is $1, \ldots, N_{t-1} + 1$ for some number of features N_{t-1} that is increased dynamically as new features are acquired. We distinguish two cases, namely that a feature corresponds to a previously observed one (hence $n_t \leq N_{t-1}$), or that z_t corresponds to a new, previously unobserved feature ($n_t = N_{t-1} + 1$). We will denote the robot's guess of n_t by $\hat{n}_t$.

To make the correspondence variables explicit in our notation, the posterior estimated by SEIF will henceforth be denoted

$$p(\xi_t \mid z^t, u^t, \hat{n}^t). \qquad (51)$$

Here $\hat{n}^t$ is the sequence of the estimated values of the correspondence variables n^t. Notice that we choose to place the correspondences on the right side of the conditioning bar. The maximum likelihood approach simply chooses the correspondence that maximizes the measurement likelihood at any point in time:

$$\hat{n}_t = \operatorname{argmax}_{n_t} p(z_t \mid z^{t-1}, u^t, \hat{n}^{t-1}, n_t)$$
$$= \operatorname{argmax}_{n_t} \int p(z_t \mid \xi_t, n_t) \underbrace{p(\xi_t \mid z^{t-1}, u^t, \hat{n}^{t-1})}_{\bar{H}_t, \bar{b}_t} \, d\xi_t$$
$$= \operatorname{argmax}_{n_t} \int \int p(z_t \mid x_t, y_{n_t}, n_t)$$
$$p(x_t, y_{n_t} \mid z^{t-1}, u^t, \hat{n}^{t-1}). \qquad (52)$$

Our notation $p(z_t \mid x_t, y_{n_t}, n_t)$ of the sensor model makes the correspondence variable n_t explicit. Calculating this probability exactly is not possible in constant time, since it involves marginalizing out almost all variables in the map (which requires the inversion of a large matrix). However, the same type of approximation that was essential for the efficient sparsification can also be applied here as well.

In particular, let us denote by $Y_{n_t}^+$ the combined Markov blanket of the robot pose x_t and the landmark y_{n_t}. This Markov blanket is the set of all features in the map that are linked to the robot of landmark y_{n_t}. Figure 6 illustrates this set. Notice that $Y_{n_t}^+$ includes by definition all active landmarks. The sparseness of $\bar{H}_t$ ensures that $Y_{n_t}^+$ contains only a fixed number of features, regardless of the size of the map N.

All other features will be collectively referred to as $Y_{n_t}^-$, i.e.,

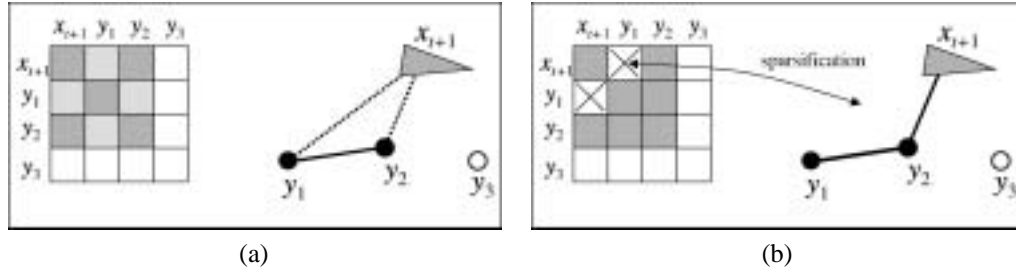$$Y_{n_t}^- = Y - Y_{n_t}^+ - \{y_{n_t}\}. \qquad (53)$$

Fig. 5. Sparsification: a feature is deactivated by eliminating its link to the robot. To compensate for this change in information state, links between active features and/or the robot are also updated. The entire operation can be performed in constant time.

The set $Y_{n_t}^-$ contains only features whose location asserts only an indirect influence on the two variables of interest, x_t and y_{n_t}. Our approach approximates the probability $p(x_t, y_{n_t} \mid z^{t-1}, u^t, \hat{n}^{t-1})$ in eq. (52) by essentially ignoring these indirect influences:

$$
\begin{aligned}
& p(x_t, y_{n_t} \mid z^{t-1}, u^t, \hat{n}^{t-1}) \\
&= \iint p(x_t, y_{n_t}, Y_{n_t}^+, Y_{n_t}^- \mid z^{t-1}, u^t, \hat{n}^{t-1}) \, dY_{n_t}^+ \, dY_{n_t}^- \\
&= \iint p(x_t, y_{n_t} \mid Y_{n_t}^+, Y_{n_t}^-, z^{t-1}, u^t, \hat{n}^{t-1}) \\
& \quad\quad p(Y_{n_t}^+ \mid Y_{n_t}^-, z^{t-1}, u^t, \hat{n}^{t-1}) \\
& \quad\quad p(Y_{n_t}^- \mid z^{t-1}, u^t, \hat{n}^{t-1}) \, dY_{n_t}^+ \, dY_{n_t}^- \quad\quad (54) \\
&\approx \int p(x_t, y_{n_t} \mid Y_{n_t}^+, Y_{n_t}^- = \mu_n^-, z^{t-1}, u^t, \hat{n}^{t-1}) \\
& \quad\quad p(Y_{n_t}^+ \mid Y_{n_t}^- = \mu_n^-, z^{t-1}, u^t, \hat{n}^{t-1}) \, dY_{n_t}^+.
\end{aligned}
$$

This probability can be computed in constant time. In complete analogy to various derivations above, we note that the approximation of the posterior is simply obtained by carving out the submatrix corresponding to the two target variables:

$$
\begin{aligned}
\Sigma_{t:n_t} &= S_{x_t, y_n}^{\mathrm{T}} (S_{x_t, y_n, Y_n^+}^{\mathrm{T}} H_t \, S_{x_t, y_n, Y_n^+})^{-1} S_{x_t, y_n} \\
\mu_{t:n_t} &= \mu_t S_{x_t, y_n}. \quad\quad (55)
\end{aligned}
$$

This calculation is constant time, since it involves a matrix whose size is independent of N. From this Gaussian, the desired measurement probability in eq. (52) is now easily recovered, as described in Section 2.2. In our experiment, we found this approximation to work surprisingly well. In the results reported further below using real-world data, the average relative error in estimating likelihoods is 3.4×10^{-4}. Association errors due to this approximation were practically non-existent.

New features are detected by comparing the likelihood $p(z_t \mid z^{t-1}, u^t, \hat{n}^{t-1}, n_t)$ to a threshold α. If the likelihood is smaller than α, we set $\hat{n}_t = N_{t-1} + 1$ and $N_t = N_{t-1} + 1$; otherwise the size of the map remains unchanged, that is, $N_t = N_{t-1}$. Such an approach approach is standard in the context of EKFs (Dissanayake et al. 2001).
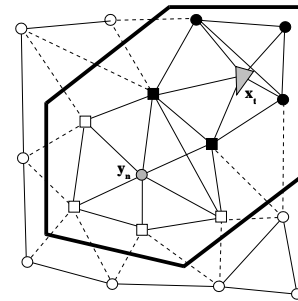

Fig. 6. The combined Markov blanket of feature y_n and robot x_t is sufficient for approximating the posterior probability of the feature locations, conditioning away all other features. This insight leads to a constant time method for recovering the approximate probability distribution $p(x_t, y_n \mid z^{t-1}, u^t)$.

4.2. Map Management

Our exact mechanism for building up the map is closely related to common procedures in the SLAM community (Dissanayake et al. 2001). Due to erroneous feature detections caused for example by moving objects or measurement noise, additional care has to be taken to filter out those interfering measurements. For any detected object that cannot be explained by existing features, a new feature candidate is generated but not put into SEIF directly. Instead it is added into a provisional list with a weight representing its probability of being a useful feature. In the next measurement step, the newly arrived candidates are checked against all candidates in the waiting list; reasonable matches increase the weight of corresponding candidates. Candidates that are not matched lose weight because they are more likely to be a moving object. When a candidate has its weight above a certain threshold, it joins the SEIF network of features.

We notice that data association violates the constant-time property of SEIFs. This is because when calculating data associations, multiple features have to be tested. If we can ensure
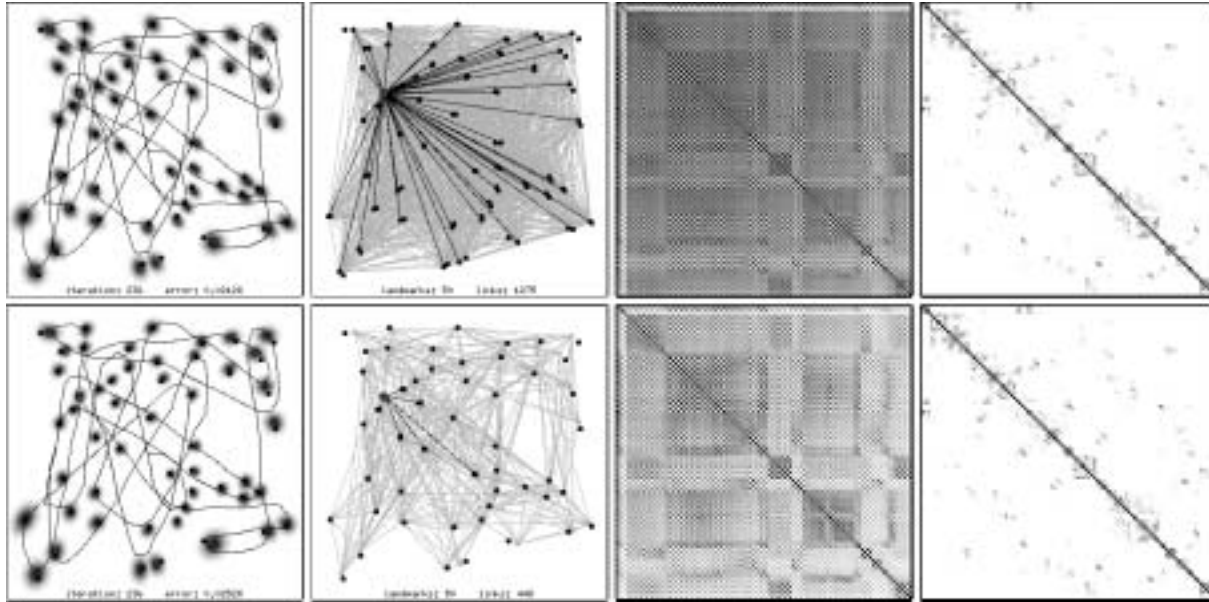
Fig. 7. Comparison of EKFs (top column) with SEIFs (bottom column) using a simulation with $N = 50$ landmarks. In both diagrams, the left panels show the final filter result, which indicates higher certainties for our approach due to the approximations involved in maintaining a sparse information matrix. The center panels show the links: black, between the robot and the active landmarks; gray, between landmarks. The right panels show the resulting covariance and normalized information matrices for both approaches. Notice the similarity. Even though the information matrix in SEIFs is sparse, the resulting correlation matrix is almost equivalent to that produced by the EKF.

that all plausible features are already connected in the SEIF by a short path to the set of active features, it would be feasible to perform data association in constant time. In this way, the SEIF structure naturally facilitates the search of the most likely feature given a measurement. However, this is not the case when closing a cycle for the first time, in which case the correct association might be far away in the SEIF adjacency graph. Using incremental versions of kd-trees (Lomet and Salzberg 1990; Procopiuc et al. 2002), it appears to be feasible to implement data association in logarithmic time by recursively partitioning the space of all feature locations using a tree. However, our present implementation does not rely on such trees, hence is overly inefficient.

As a final aside, we notice that another important operation can be done in constant time in SEIF: the merge of identical features previously mistreated as two or more unique ones. It is simply accomplished by adding corresponding values in the H_t matrix and b_t vector. This operation is necessary when collapsing multiple features into one upon the arrival of further sensor evidence, a topic that is presently not implemented.

5. Experimental Results

5.1. Real Vehicle Results

The primary purpose of our experimental comparison was to evaluate the performance of the SEIF against that of the popu-

Fig. 8. The vehicle used in our experiments is equipped with a two-dimensional laser range finder and a differential GPS system. The vehicle's ego-motion is measured by a linear variable differential transformer sensor for the steering, and a wheel-mounted velocity encoder. In the background, the Victoria Park test environment can be seen.

lar EKF algorithm, from which the SEIF is derived. We begin our exposition with experiments using a real-world benchmark data set, which has commonly used to evaluate SLAM algorithms (Guivant and Nebot 2001; Montemerlo et al. 2003; Neira, Tardós, and Castellanos 2003). This data set was
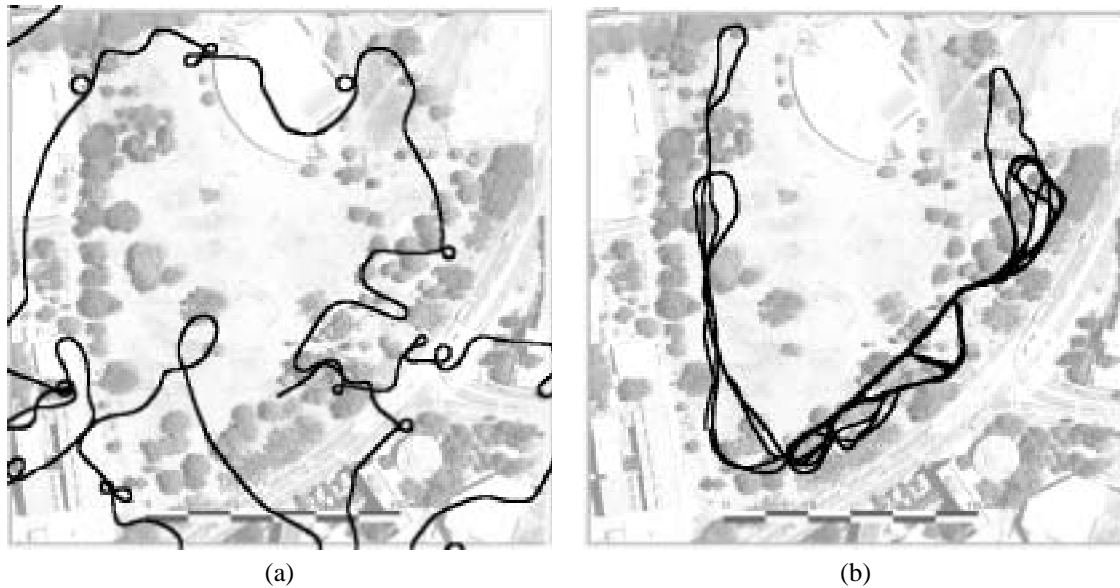
(a)   (b)

Fig. 9. The testing environment. A 350×350 m^2 patch in Victoria Park in Sydney. (a) shows integrated path from odometry readings and (b) shows the path as the result of SEIF.
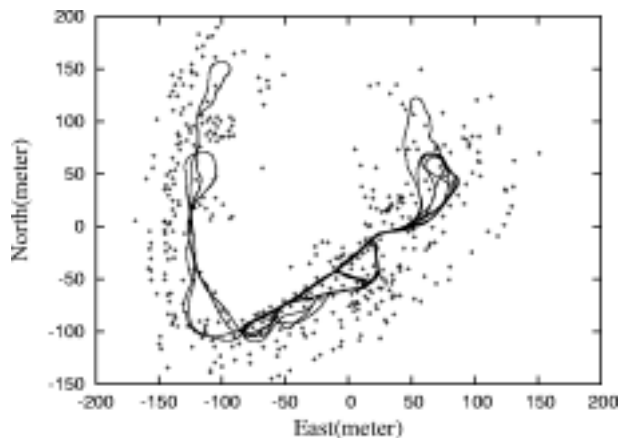

Fig. 10. Overlay of estimated landmark positions and robot path.

collected with an instrumented outdoor vehicle driven through a park in Sydney, Australia.

The vehicle and its environment are shown in Figures 8 and 9, respectively. The robot is equipped with a SICK laser range finder and a system for measuring steering angle and forward velocity. The laser is used to detect trees in the environment, but it also picks up hundreds of spurious features such as corners of moving cars on a nearby highway. The raw odometry, as used in our experiments, is poor, resulting in several hundred meters of error when used for path integration along the vehicle's 3.5 km path. This is illustrated in Fig-

ure 9(a), which shows the path of the vehicle. The poor quality of the odometry information along with the presence of many spurious features make this data set particularly amenable for testing SLAM algorithms.

The path recovered by the SEIF is shown in Figure 9(b). This path is quantitatively indistinguishable from the one produced by the EKF and related variants (Guivant and Nebot 2001; Montemerlo and Thrun 2003; Montemerlo et al. 2003; Neira, Tardós, and Castellanos 2003). The average position error, as measured through differential GPS, is smaller than 0.50 m, which is small compared to the overall path length of 3.5 km. Compared with EKF, SEIF runs approximately twice as fast and consumes less than a quarter of the memory EKF uses. Moreover, the residual error is approximately the same as that of other state-of-the-art techniques, such as those reported in Guivant and Nebot (2001) Montemerlo and Thrun (2003), and Montemerlo et al. (2003).

We conclude that the SEIF performs as well on a physical benchmark data set as far as its accuracy is concerned; however, even though the overall size of the map is small, using SEIFs results in noticeable savings both in memory and execution time.

5.2. Simulation Results

Unfortunately, the real-world data set prohibits systematic variation of key parameters, such as the size of the environment and the amount of measurement noise. The results reported in the remainder of this paper are based on simulation.

In our simulations, we focused particularly on the "loop closing" problem, which is generally acknowledged to be one of the hardest problems in SLAM (Lu and Milios 1997; Gutmann and Konolige 1999; Thrun 2000; Bosse et al. 2003; Hähnel et al. 2003a). When closing a loop, usually many landmark locations are affected. This puts to the test our amortized map recovery mechanism under difficult circumstances. As noted above, loop closures are the only condition under which SEIFs cannot be executed in constant time per update, since the most likely data association requires non-local search.

The robot simulator is set up to always generate maps with the same average density of landmarks; as the number of landmarks is increased, so is the size of the environment. Each unit interval possesses 50 landmarks (on average). Landmarks are uniformly drawn in a squared region of size $\sqrt{50\,N}$ by $\sqrt{50\,N}$; however, only landmarks are retained that meet a minimum distance requirement to previously drawn landmarks. By growing the size of the environment by the square root of N, the average density of landmarks remains constant, regardless of the number of landmarks involved. The noise of robot motion and measurements are all modeled by zero mean Gaussian noise. Specifically, the variance is 10^{-4} for forward velocity, 10^{-3} for rotational velocity, 0.002 for range detection and 0.003 for bearings measurements. In each iteration of the simulation, the robot takes one move and one measurement, at which it may sense a variable number of nearby landmarks. In each of our experiments, we performed a total of $20N$ iterations, which leads roughly to the same number of sightings of individual landmarks. The maximum sensor range is set to 0.2, which results in approximately six landmark detections on average for one measurement step. Unless otherwise noted, the number of active landmarks bounded by $\theta_x = 6$. The variable θ_y remains unconstrained, since the constraint on θ_x effectively restricts the number of between-landmark links.

Figures 11 and 12 show that SEIFs outperform EKFs in terms of computation and memory usage. In particular, Figure 11 illustrates that in SEIFs, the computation time virtually levels off at $N = 300$, regardless of the number of landmarks involved. In EKFs, in contrast, the time increases quadratically with the number of landmarks N. Clearly, this makes EKFs prohibitively slow for larger maps. EKFs, on the other hand, outperform SEIFs for very small number of landmarks ($N \leq 200$), due to the additional computational overhead involved in the sparsification and the map recovery. Figure 12 illustrates that the memory requirement of SEIFs is strictly superior of that of EKFs. The memory consumed by SEIFs increases linearly with the size of the map, whereas that of EKFs grows quadratically.

A key open question pertains to the degree at which maintaining sparse matrices affects the overall error of the map. Empirical simulation results are shown in Figure 13, which plots the empirical error as a function of the map size N. In absolute terms, the error in each of these maps is extremely
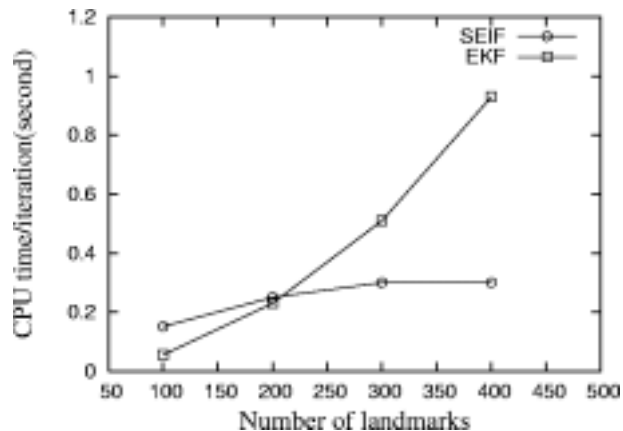

Fig. 11. The comparison of average CPU time between SEIF and EKF.
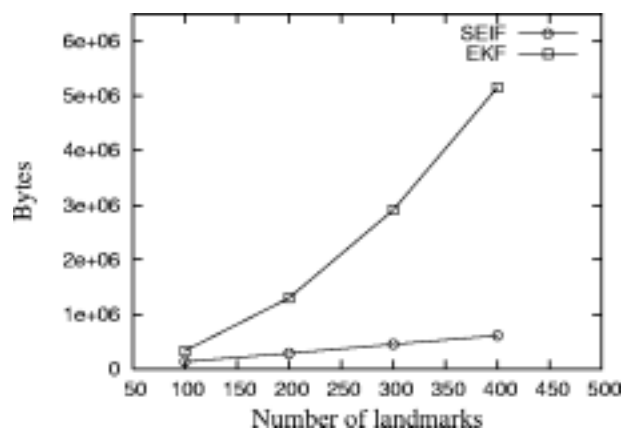

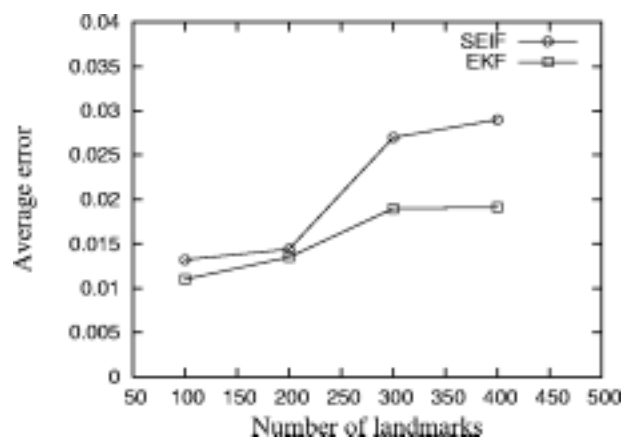Fig. 12. The comparison of average memory usage between SEIF and EKF.

Fig. 13. The comparison of root mean square distance error between SEIF and EKF.
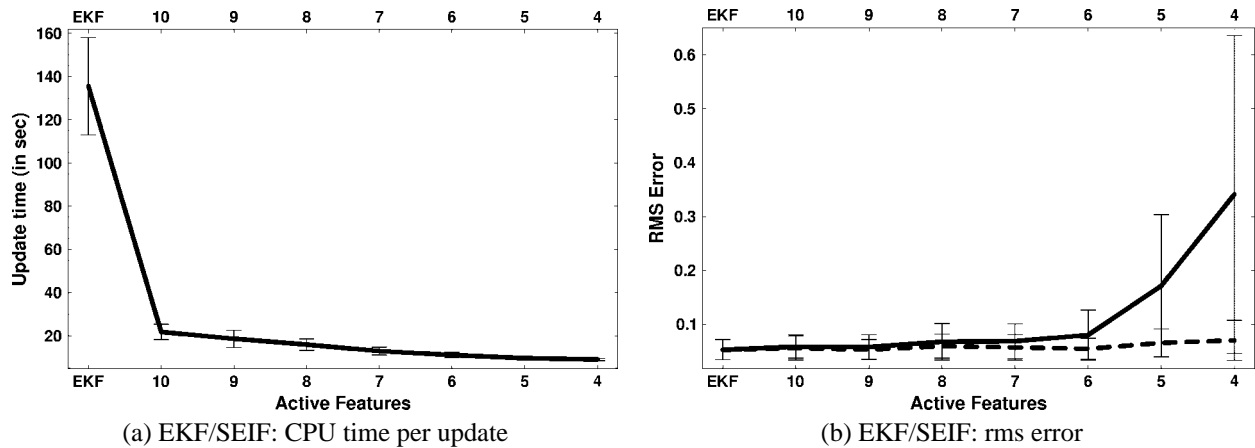
(a) EKF/SEIF: CPU time per update

(b) EKF/SEIF: rms error

Fig. 14. Comparison of EKFs and SEIFs for different degrees of sparseness, induced by different values of θ_x: (a) update time; (b) rms error in the residual estimate.

small. Recall that for $N = 200$, the landmarks are spread in a region of size 100 by 100, whereas both methods yield an approximate error of 0.015 per landmark. Both curves increase approximately linearly with N. This should not surprise, as the total area of the environment also increases linearly with N. However, SEIFs perform noticeably poorer than EKFs in this experiment. This increase in error is due to the various approximations underlying SEIFs.

In a final series of experiments, we evaluated the dependence of the computation time and the error on the sparsity of the filter. We systematically varied the threshold θ_x, which determines the maximum number of landmarks than can be active at a time. Because between-landmark arcs can only develop between landmarks that are active at the same time, limiting θ_x also limits the number of between-landmark arcs.

Figure 14 shows the basic result. The left diagram depicts the update time for EKFs and SEIFs with varying numbers of active landmarks, and the solid curve in the right diagram shows the corresponding map errors. For a map with $N = 50$ landmarks, EKFs require 135 ± 22.5 s per update on a low-end PC. SEIFs with $\theta_x = 10$ active landmarks require 21.8 ± 3.50 s. The increase in error is quite small. Whereas EKF's error is 0.0526 ± 0.0189, the SEIF error is 0.0584 ± 0.0215. As the number of active landmarks is reduced, the update becomes increasingly efficient, but at the expense of an increased error. For $\theta_x = 6$, we obtain an update time of 13.0 ± 1.81, with an error of 0.0800 ± 0.0463, which is a 51% increase in error for a tenfold speedup. Beyond this, the error grows more rapidly. For example for $\theta_x = 6$ the update time is 9.07 ± 0.513, but the error is now 0.341 ± 0.295, which is a 548% increase over EKFs. From these results, it appears that five active landmarks give good results; less than that induces a significant loss—although the final selection of θ_x inevitably will depend on the costs of mapping error relative to the costs of computation.

This result raises the question as to what causes this error. To dissect possible sources of error, we implemented SEIFs using the exact equations for recovering the mean and covariance, as defined in eq. (6). In this way, we can separate the error arising from the amortized recovery of the mean, from the error induced by the sparsification. The dashed curve in Figure 14(b) shows the resulting error. As this curve illustrates, even a highly sparse SEIF is capable of producing accurate results. The error for $\theta_x = 4$ active landmarks is 0.0701 ± 0.0372, which is only 33% larger than that of EKFs (instead of 548%).

5.3. Consistency

SEIFs can be overconfident, that is, the covariance of the posterior estimates can suggest a higher degree of confidence than actually warranted by the sensor measurements. Such overconfidence is commonly called "inconsistency" in the SLAM literature. It arises from a number of factors. First, the linearization frequently causes overconfidence, which affects both the EKF and the SEIF solution. Further, the truncation of direct long-range links in SEIFs—a result of the sparsification—can further induce overconfidence. Inconsistency does not necessarily induce error or jeopardize convergence. In fact, a recent result proves convergence for a filter that maintains no covariance estimate, hence is maximally overconfident (Montemerlo et al. 2003). However, overconfidence can adversely affect the ability to perform data association (as can underconfidence). For this reason, characterizing the degree of overconfidence is a common step in evaluating the viability of a new SLAM algorithm. Here we are interested in the additional confidence arising from the sparsification, and we compare it to the confidence levels of EKFs.

The confidence of SEIFs is depicted in Figure 15, which plots the determinant of the covariance $|\Sigma|$ as a function of
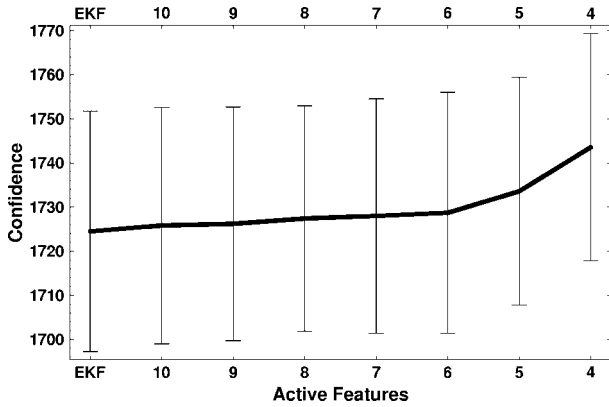
Fig. 15. Overconfidence in SEIFs. The determinant of the covariance matrix $\Sigma_t = H_T^{-1}$ plotted for EKFs and SEIFs with varying degrees of sparseness. This determinant characterizes the overall confidence in the posterior estimate.

the algorithm, for our simulation with $N = 50$ landmarks. The larger this value, the more confident the filter. While the determinant of EKFs is 1724 ± 27.4, SEIFs with $\theta_x = 10$ active landmarks yield a determinant of 1726 ± 26.8. This 0.1% increase is not statistically significant. With $\theta_x = 6$ active landmarks, we observe $|\Sigma_t^{-1}| = 1729 \pm 27.3$, a 0.3% increase that also lacks statistical significance.

To understand the effect of the overconfidence on the error, we modified the basic SEIF algorithm to yield less confident results. Our modification was straightforward. After each update, the links between the robot and the active landmarks were weakened, by a "soft" sparsification rule. More specifically, imagine after the tth update we are given an information matrix H_t and an information vector b_t. Our approach first sparsifies away all active landmark links using the math described above (applied to all active landmarks). Let the result of this operation be denoted H_t^0 and b_t^0. Our approach then mixes $\langle H_t, b_t \rangle$ and $\langle H_t^0, b_t^0 \rangle$ using a mixing ratio ρ:

$$H_t \quad \longleftarrow \quad (1 - \rho) H_t + \rho H_t^0 \quad \text{and}$$
$$b_t \quad \longleftarrow \quad (1 - \rho)\, b_t + \rho b_t^0. \tag{56}$$

The resulting estimate is less confident (by definition) than the original one, where ρ characterizes the loss of confidence. This is illustrated in Figure 16(a), which depicts the determinant of the covariance $|\Sigma_t|$ for different levels of ρ (here with $\theta_x = 6$).

The interesting finding is that this reduction in confidence—in fact, the resulting estimate is "consistent"—adversely affects the RMS map error. This is illustrated in Figure 16(b), which shows the error for different values of ρ. The more confident the filter, the smaller the resulting error. While this approach is just one way out of many to reduce confidence by taking information out of the system, this ex-

ample illustrates that a small amount of overconfidence (0.1% in our case) may be well tolerable, assuming that the goal of the filter is to maximize the accuracy in the map. In fact, given the result in Montemerlo et al. (2003), the relation between consistency and error remains unclear.

5.4. Multi-Vehicle SLAM

In a final series of experiments we applied SEIFs to a restricted version of the multi-robot SLAM problem, commonly studied in the literature (Nettleton, Gibbens, and Durrant-Whyte 2000). In our implementation, the robots are informed of their initial pose. This is a common assumption in multi-robot SLAM, necessary for the type linearization that is applied both in EKFs and SEIFs (Nettleton, Gibbens, and Durrant-Whyte 2000). Recent work that enables vehicles to build joint maps without initial knowledge of their relative pose can be found in Gutmann and Konolige (1999), Stewart et al. (2003), and Thrun and Liu (2003).

Our simulation involves a team of three air vehicles. The vehicles are not equipped with GPS; hence they accrue positioning error over time. Figure 17 shows the joint map at different stages of the simulation. As in Nettleton, Gibbens, and Durrant-Whyte (2000), we assume that the vehicles communicate updates of their information matrices and vectors, enabling them to generate a single, joint map. As argued there, the information form provides the important advantage over EKFs that communication can be delayed arbitrarily, which overcomes a need for tight synchronization inherent to the EKF. This characteristic arises directly from the fact that the information matrix H_t and the information vector b_t in SEIFs is additive, whereas covariance matrices are not. In particular, let $\langle H_t^i, b_t^i \rangle$ be the posterior of the ith vehicle. Assuming that all posteriors are expressed over the same coordinate system and that each map uses the same numbering for all landmarks, the joint posterior integrating all of these local maps is given by $\langle \sum_i H_t^i, \sum_i b_t^i \rangle$. This additive nature of the information form is well known, and has in the context of SLAM previously been exploited by Nettleton, Gibbens, and Durrant-Whyte (2000). SEIFs offer over the work in Nettleton, Gibbens, and Durrant-Whyte (2000) that the messages sent between vehicles are small, due to the sparse nature of the information form. A related approach for generating small messages in multi-vehicle SLAM has recently been described in Nettleton, Thrun, and Durrant-Whyte (2003).

Figure 17 shows a sequence of snapshots of the multi-vehicle system, using three different air vehicles. Initially, the vehicle starts our in different areas, and the combined map (illustrated by the uncertainty ellipses) consists of three disjoint regions. During steps 62–64, the top two vehicles discover identical landmarks; as a result, the overall uncertainty of their respective map region decreases; This illustrates that the SEIF indeed maintains the correlations in the individual landmark's uncertainties; albeit using a sparse information matrix instead
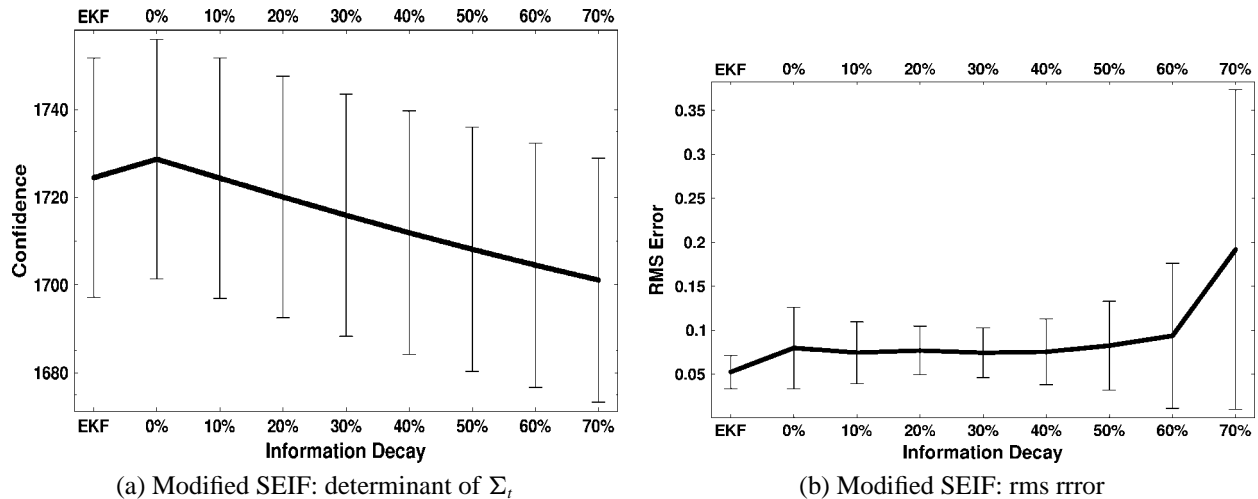
(a) Modified SEIF: determinant of Σ_t (b) Modified SEIF: rms rrror

Fig. 16. Making SEIFs underconfident (consistent). Shown on the left is the determinant of Σ_t for EKFs and the modified SEIF algorithm, with $\theta_x = 6$ active landmarks, but for different levels of information decay (see text). The right diagram depicts the corresponding error.

of the covariance matrix. Similarly, in steps 85–89, the third vehicle begins to discover identical landmarks also seen by another vehicle. Again, the resulting uncertainty of the entire map is reduced, as can be seen easily. The last panel in Figure 17 shows the final map, obtained after 500 iterations. This example shows that SEIFs are well suited for multi-robot SLAM, assuming that the initial poses of the vehicles are known.

6. Related Work

SEIFs are related to a rich body of literature on SLAM and high-dimensional filtering. Recently, several researchers have developed hierarchical techniques that decompose maps into collections of smaller, more manageable submaps (Leonard and Feder 1999; Guivant and Nebot 2001; Bailey 2002; Bosse et al. 2002; Tardós et al. 2002; Williams and Dissanayake 2002). While, in principle, hierarchical techniques can solve the SLAM problem in linear time, many of these techniques still require quadratic time per update. One recent technique updates the filter in constant time (Leonard and Feder 1999) by restricting all computation to the submap in which the robot presently operates. Using approximation techniques for transitioning between submaps, this work demonstrated that consistent error bounds can be maintained with a constant-time algorithm (which is not necessarily the case for SEIFs). However, the method does not propagate information to previously visited submaps unless the robot subsequently revisits these regions. Hence, this method suffers a slower rate of convergence in comparison to the $O(N^2)$ full covariance solution. Alternative methods based on decomposition into submaps,

such as the sequential map joining techniques described in Tardós et al. (2002) and Williams, Dissanayake, and Durrant-Whyte (2002) can achieve the same rate of convergence as the full EKF solution, but incur an $O(N^2)$ computational burden.

A different line of research has relied on particle filters for efficient mapping (Doucet, de Freitas, and Gordon 2001). The FastSLAM algorithm (Montemerlo et al. 2002, 2003; Hähnel et al. 2003b) and earlier related mapping algorithms (Murphy 2000; Thrun 2001) require time logarithmic in the number of features in the map, but they depend linearly on a particle-filter specific parameter (the number of particles). There exists now evidence that a single particle may suffice for convergence in idealized situations (Montemerlo et al. 2002), but the number of particles required for handling data association problems robustly is still not fully understood. More recently, thin junction trees have been applied to the SLAM problem by Paskin (2002). This work establishes a viable alternative to the approach proposed here, with somewhat different computational properties. However, at the present point this approach lacks an efficient technique for making data association decisions.

As noted in the introduction of this paper, the idea of representing maps by relative information has previously been explored by a number of authors, most notably in recent algorithms by Newman (2000) and Csorba (1997) and Deans and Hebert (2000); it is also related to an earlier algorithm by Lu and Milios (1997) and Gutmann and Nebel (1997). The Newman algorithm assumes sensors that provide relative information between multiple landmarks, which enables it to bypass the issue of sparsification of the information matrix. The work by Lu and Milios uses robot poses as the core representation, hence the size of the filter grows linearly over time (even for maps of finite size). As a result, the approach is not

Step $t = 3$ Step $t = 62$

Step $t = 65$ Step $t = 85$

Step $t = 89$ Step $t = 500$
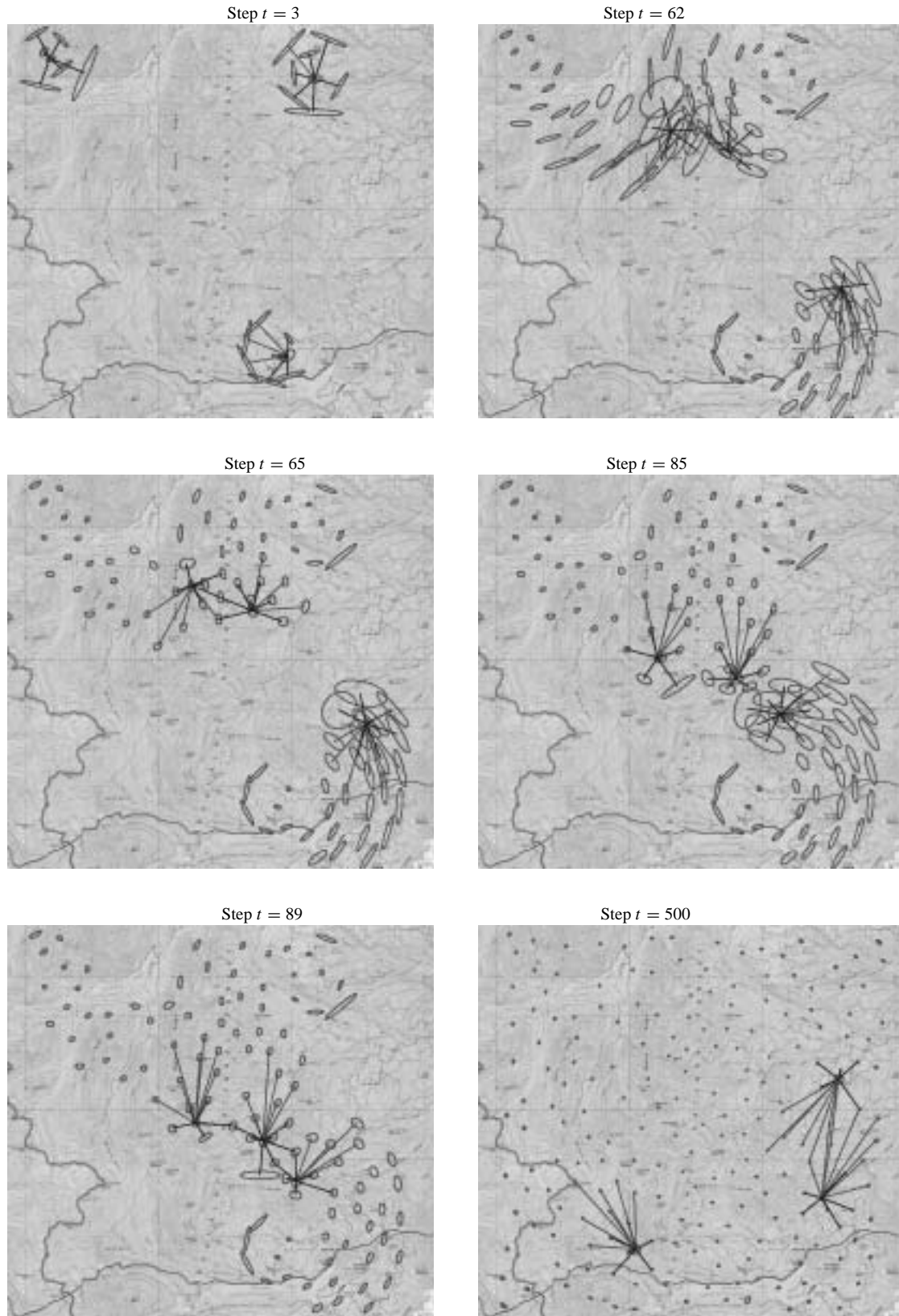

Fig. 17. Snapshots from our multi-robot SLAM simulation at different points in time. Initially, the poses of the vehicles are known. During steps 62–64, vehicles 1 and 2 traverse the same area for the first time; as a result, the uncertainty in their local maps shrinks. Later, in steps 85–89, vehicle 2 observes the same landmarks as vehicle 3, with a similar effect on the overall uncertainty. After 500 steps, all landmarks are accurately localized.

applicable on-line. However, the approach by Lu and Milios relies on local links between adjacent poses, similar to the local links maintained by SEIFs between nearby landmarks. It therefore shares many of the computational properties of SEIFs when applied to data sets of limited size.

Just as in recent work by Nettleton, Gibbens, and Durrant-Whyte (2000), our approach is based on the information form of the EKF (Maybeck 1979), as noted above. However, Nettleton and colleagues focus on the issue of communication between multiple robots; as a result, they have not addressed computational efficiency problems (their algorithm requires $O(N^3)$ time per update). Relative to this work, a central innovation in SEIFs is the sparsification step, which results in an increased computational efficiency. A second innovation is the amortized constant time recovery of the map.

As noted above, the information matrix and vector estimated by the SEIF defines a Gaussian Markov random field (GMRF; Weiss and Freeman 2001). As a direct consequence, a rich body of literature in inference in sparse GMRFs becomes directly applicable to a number of problems addressed here, such as the map recovery, the sparsification, and the marginalization necessary for data association (Pearl 1988; Murphy, Weiss, and Jordan 1999; Wainwright 2002). Also applicable is the rich literature on sparse matrix transformations (Gupta, Karypis, and Kumar 1997).

7. Discussion

In this paper we have proposed an efficient algorithm for the SLAM problem. Our approach is based on the well-known information form of the EKF. Based on the empirical observation that the information matrix is dominated by a small number of entries that are found only between nearby features in the map, we have developed a SEIF. This filter enforces a sparse information matrix, which can be updated in constant time. In the linear SLAM case with known data association, all updates can be performed in constant time; in the nonlinear case, additional state estimates are needed that are not part of the regular information form of the EKF. We have proposed an amortized constant-time coordinate descent algorithm for recovering these state estimates from the information form. We have also proposed an efficient algorithm for data association in SEIFs that requires logarithmic time, assuming that the search for nearby features is implemented by an efficient search tree. The approach has been implemented and compared to the EKF solution. Overall, we find that SEIFs produce results that differ only marginally from that of the EKFs, yet at a much improved computational speed. Given the computational advantages of SEIFs over EKFs, we believe that SEIFs should be a viable alternative to EKF solutions when building high-dimensional maps.

SEIFs, represented here, possess a number of critical limitations that warrant future research. First and foremost, SEIFs may easily become overconfident, a property often referred to as "inconsistent" (Leonard and Feder 1999; Julier and Uhlmann 2000). The overconfidence mainly arises from the approximation in the sparsification step. Such overconfidence is not necessarily an problem for the convergence of the approach (Montemerlo et al. 2003), but it may introduce errors in the data association process. In practice, we did not find the overconfidence to affect the result in any noticeable way; however, it is relatively easy to construct situations in which it leads to arbitrary errors in the data association process.

Another open question concerns the speed at which the amortized map recovery converges. Clearly, the map is needed for a number of steps; errors in the map may therefore affect the overall estimation result. Again, our real-world experiments show no sign of noticeable degradation, but a small error increase was noted in one of our simulated experiments.

Finally, SEIF inherits a number of limitations from the common literature on SLAM. Among those are the use of Taylor expansion for linearization, which can cause the map to diverge; the static world assumption which makes the approach inapplicable to modeling moving objects (Wang, Thorpe, and Thrun 2003); the inability to maintain multiple data association hypotheses, which makes the approach brittle in the presence of ambiguous features; the reliance on features, or landmarks; and the requirement that the initial pose be known in the multi-robot implementation. Virtually all of these limitations have been addressed in the recent literature. For example, a recent line of research has devised efficient particle filtering techniques (Murphy 2000; Hähnel et al. 2003b; Montemerlo et al. 2003) that address most of these shortcomings. The issues addressed in this paper are somewhat orthogonal to these limitations, and it appears feasible to combine efficient particle filter sampling with SEIFs. We also note that in a recent implementation, a new lazy data association methodology was developed that uses a SEIF-style information matrix to robustly generate maps with hundreds of meters in diameter (Thrun et al. 2003).

The use of sparse matrices in SLAM offers a number of important insights into the design of SLAM algorithms. Our approach puts a new perspective on the rich literature on hierarchical mapping discussed further above. As in SEIFs, these techniques focus updates on a subset of all features, to gain computational efficiency. SEIFs, however, compose submaps dynamically, whereas past work relied on the definition of static submaps. We conjecture that our sparse network structures capture the natural dependences in SLAM problems much better than static submap decompositions, and in turn lead to more accurate results. They also avoid problems that frequently occur at the boundary of submaps, where the estimation can become unstable. However, the verification of these claims will be subject to future research. A related paper discusses the application of constant-time techniques to information exchange problems in multi-robot SLAM (Nettleton, Thrun, and Durrant-Whyte 2002).

Finally, we note that our work sheds some fresh light on the ongoing discussion on the relation of topological and metric maps, a topic that has been widely investigated in the cognitive mapping community (Kuipers and Byun 1988; Chown, Kaplan, and Kortenkamp 1995). Links in SEIFs capture relative information, in that they relate the location of one landmark to another (see also Csorba 1997; Deans and Hebert 2000; Newmann 2000). This is a common characteristic of topological map representations (Matarić 1990; Kuipers and Byun 1991; Choset 1996; Shatkay and Kaelbling 1997). SEIFs also offer a sound method for recovering absolute locations and affiliated posteriors for arbitrary submaps based on these links, of the type commonly found in metric map representations (Smith and Cheeseman 1986; Moravec 1988). Thus, SEIFs bring together aspects of both paradigms, by defining simple computational operations for changing relative to absolute representations, and vice versa.

Appendix: Proofs

Proof of Lemma 1: Measurement updates are realized via eqs. (17) and (18), restated here for the reader's convenience:

$$H_t = \bar{H}_t + C_t Z^{-1} C_t^{\mathrm{T}} \tag{57}$$
$$b_t = \bar{b}_t + (z_t - \hat{z}_t + C_t^{\mathrm{T}} \mu_t)^{\mathrm{T}} Z^{-1} C_t^{\mathrm{T}}. \tag{58}$$

From the estimate of the robot pose and the location of the observed feature, the prediction $\hat{z}_t$ and all non-zero elements of the Jacobian C_t can be calculated in constant time, for any of the commonly used measurement models g. The constant-time property follows now directly from the sparseness of the matrix C_t, discussed already in Section 2.2. This sparseness implies that only finitely many values have to be changed when transitioning from $\bar{H}_t$ to H_t, and from $\bar{b}_t$ to b_t.   □

Proof of Lemma 2: For $A_t = 0$, eq. (28) gives us the following updating equation for the information matrix:

$$\bar{H}_t = [H_{t-1}^{-1} + S_x U_t S_x^{\mathrm{T}}]^{-1}. \tag{59}$$

Applying the matrix inversion lemma leads to the following form:

$$\begin{aligned}
\bar{H}_t &= H_{t-1} - H_{t-1} \underbrace{S_x [U_t^{-1} + S_x^{\mathrm{T}} H_{t-1} S_x]^{-1} S_x^{\mathrm{T}} H_{t-1}}_{=:L_t} \\
&= H_{t-1} - H_{t-1} L_t. \tag{60}
\end{aligned}$$

The update of the information matrix, $H_{t-1} L_t$, is a matrix that is non-zero only for elements that correspond to the robot pose and the active features. To see, we note that the term inside the inversion in L_t is a low-dimensional matrix which is of the same dimension as the motion noise U_t. The inflation via the matrices S_x and S_x^{T} leads to a matrix that is zero except for elements that correspond to the robot pose. The key insight now is that the sparseness of the matrix H_{t-1} implies that only

finitely many elements of $H_{t-1} L_t$ may be non-zero, namely those corresponding to the robot pose and active features. They are easily calculated in constant time.

For the information vector, we obtain from eqs. (28) and (60):

$$\begin{aligned}
\bar{b}_t &= [b_{t-1} H_{t-1}^{-1} + \hat{\Delta}_t^{\mathrm{T}}] \bar{H}_t \\
&= [b_{t-1} H_{t-1}^{-1} + \hat{\Delta}_t^{\mathrm{T}}](H_{t-1} - H_{t-1} L_t) \\
&= b_{t-1} + \hat{\Delta}_t^{\mathrm{T}} H_{t-1} - b_{t-1} L_t + \hat{\Delta}_t^{\mathrm{T}} H_{t-1} L_t. \tag{61}
\end{aligned}$$

As above, the sparseness of H_{t-1} and of the vector $\hat{\Delta}_t$ ensures that the update of the information vector is zero except for entries corresponding to the robot pose and the active features. Those can also be calculated in constant time.   □

Proof of Lemma 3: The update of $\bar{H}_t$ requires the definition of the auxiliary variable $\Psi_t := (I + A_t)^{-1}$. The non-trivial components of this matrix can essentially be calculated in constant time by virtue of

$$\begin{aligned}
\Psi_t &= (I + S_x S_x^{\mathrm{T}} A_t S_x S_x^{\mathrm{T}})^{-1} \\
&= I - I S_x (S_x I S_x^{\mathrm{T}} + [S_x^{\mathrm{T}} A_t S_x]^{-1})^{-1} S_x^{\mathrm{T}} I \\
&= I - S_x (I + [S_x^{\mathrm{T}} A_t S_x]^{-1})^{-1} S_x^{\mathrm{T}}. \tag{62}
\end{aligned}$$

Notice that Ψ_t differs from the identity matrix I only at elements that correspond to the robot pose, as is easily seen from the fact that the inversion in eq. (62) involves a low-dimensional matrix.

The definition of Ψ_t allows us to derive a constant-time expression for updating the information matrix H:

$$\begin{aligned}
\bar{H}_t &= [(I + A_t) H_{t-1}^{-1} (I + A_t)^{\mathrm{T}} + S_x U_t S_x^{\mathrm{T}}]^{-1} \\
&= [(\underbrace{\Psi_t^{\mathrm{T}} H_{t-1} \Psi_t}_{=:H_{t-1}'})^{-1} + S_x U_t S_x^{\mathrm{T}}]^{-1} \\
&= [(H_{t-1}')^{-1} + S_x U_t S_x^{\mathrm{T}}]^{-1} \\
&= H_{t-1}' - \underbrace{H_{t-1}' S_x [U_t^{-1} + S_x^{\mathrm{T}} H_{t-1}' S_x]^{-1} S_x^{\mathrm{T}} H_{t-1}'}_{=:\Delta H_t} \\
&= H_{t-1}' - \Delta H_t. \tag{63}
\end{aligned}$$

The matrix $H_{t-1}' = \Psi_t^{\mathrm{T}} H_{t-1} \Psi_t$ is easily obtained in constant time and, by the same reasoning as above, the entire update requires constant time. The information vector $\bar{b}_t$ is now obtained as follows:

$$
\begin{aligned}
\bar{b}_t &= [b_{t-1}H_{t-1}^{-1} + \hat{\Delta}_t^{\mathrm{T}}]\bar{H}_t \\
&= b_{t-1}H_{t-1}^{-1}\bar{H}_t + \hat{\Delta}_t^{\mathrm{T}}\bar{H}_t \\
&= b_{t-1}H_{t-1}^{-1}(\bar{H}_t + \underbrace{H_{t-1} - H_{t-1}}_{=0} + \underbrace{H'_{t-1} - H'_{t-1}}_{=0}) + \hat{\Delta}_t^{\mathrm{T}}\bar{H}_t \\
&= b_{t-1}H_{t-1}^{-1}(H_{t-1} + \underbrace{\bar{H}_t - H'_{t-1}}_{-\Delta H_t} - H_{t-1} + H'_{t-1}) + \hat{\Delta}_t^{\mathrm{T}}\bar{H}_t \\
&= b_{t-1}H_{t-1}^{-1}(H_{t-1} - \Delta H_t - H_{t-1} + H'_{t-1}) + \hat{\Delta}_t^{\mathrm{T}}\bar{H}_t \\
&= b_{t-1} - b_{t-1}H_{t-1}^{-1}(\Delta H_t - H_{t-1} + H'_{t-1}) + \hat{\Delta}_t^{\mathrm{T}}\bar{H}_t \\
&= b_{t-1} - \mu_{t-1}^{\mathrm{T}}H_{t-1}H_{t-1}^{-1}(\Delta H_t - H_{t-1} + H'_{t-1}) + \hat{\Delta}_t^{\mathrm{T}}\bar{H}_t \\
&= b_{t-1} - \mu_{t-1}^{\mathrm{T}}(\Delta H_t - H_{t-1} + H'_{t-1}) + \hat{\Delta}_t^{\mathrm{T}}\bar{H}_t. \quad (64)
\end{aligned}
$$

The update ΔH_t is non-zero only for elements that correspond to the robot pose or active features. Similarly, the difference $H'_{t-1} - H_{t-1}$ is non-zero only for constantly many elements. Therefore, only those mean estimates in μ_{t-1} are necessary to calculate the product $\mu_{t-1}^{\mathrm{T}}\Delta H_t$.   □

Proof of Lemma 4: The mode $\hat{v}_t$ of eq. (45) is given by

$$
\begin{aligned}
\hat{v}_t &= \mathrm{argmax}_{v_t} \ p(v_t) \\
&= \mathrm{argmax}_{v_t} \ \exp\left\{-\tfrac{1}{2}v_t^{\mathrm{T}}H_t v_t + b_t^{\mathrm{T}}v_t\right\} \\
&= \mathrm{argmin}_{v_t} \ \tfrac{1}{2}v_t^{\mathrm{T}}H_t v_t - b_t^{\mathrm{T}}v_t. \quad (65)
\end{aligned}
$$

The gradient of the expression inside the minimum in eq. (65) with respect to v_t is given by

$$
\frac{\partial}{\partial v_t}\left\{\tfrac{1}{2}v_t^{\mathrm{T}}H_t v_t - b_t^{\mathrm{T}}v_t\right\} = H_t v_t - b_t^{\mathrm{T}}, \quad (66)
$$

whose minimum $\hat{v}_t$ is attained when the derivative (66) is 0, i.e.,

$$
\hat{v}_t = H_t^{-1}b_t^{\mathrm{T}}. \quad (67)
$$

From this and eq. (44) it follows that $\hat{v}_t = \mu_t$.   □

Acknowledgments

References

Bailey, T. 2002. *Mobile Robot Localization and Mapping in Extensive Outdoor Environments*. PhD thesis, University of Sydney, Sydney, NSW, Australia.

Bosse, M., Leonard, J., and Teller, S. 2002. Large-scale CML using a network of multiple local maps. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Workshop Notes of the ICRA Workshop on Concurrent Mapping and Localization for Autonomous Mobile Robots (W4)*, Washington, DC, May 11–15.

Bosse, M., Newman, P., Soika, M., Feiten, W., Leonard, J., and Teller, S. 2003. An atlas framework for scalable mapping. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan.

Burgard, W., Fox, D., Jans, H., Matenar, C., and Thrun, S. 1999. Sonar-based mapping of large-scale mobile robot environments using EM. *Proceedings of the International Conference on Machine Learning*, Bled, Slovenia.

Burgard, W., Fox, D., Moors, M., Simmons, R., and Thrun, S. 2000. Collaborative multi-robot exploration. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, San Francisco, CA, April 24–28.

Choset, H. 1996. *Sensor Based Motion Planning: The Hierarchical Generalized Voronoi Graph*. PhD thesis, California Institute of Technology.

Chown, E., Kaplan, S., and Kortenkamp, D. 1995. Prototypes, location, and associative networks (plan): towards a unified theory of cognitive mapping. *Cognitive Science* 19:1–51.

Cover, T. M. and Thomas, J. A. 1991. *Elements of Information Theory*. Wiley, New York.

Csorba, M. 1997. *Simultaneous Localization and Map Building*. PhD thesis, Department of Engineering Science, University of Oxford, Oxford, UK.

Deans, M. and Hebert, M. 2000. Invariant filtering for simultaneous localization and mapping. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, San Francisco, CA, April 24–28, pp. 1042–1047.

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

Doucet, A., de Freitas, J. F. G., and Gordon, N. J. editors. 2001. *Sequential Monte Carlo Methods In Practice*. Springer Verlag, New York.

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

Gupta, A., Karypis, G., and Kumar, V. 1997. Highly scalable parallel algorithms for sparse matrix factorization. *IEEE Transactions on Parallel and Distributed Systems* 8(5):502–520.

Gutmann, J.-S., and Konolige, K. 1999. Incremental mapping of large cyclic environments. *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, Monterey, CA.

Gutmann, J.-S., and Nebel, B. 1997. Navigation mobiler roboter mit laserscans. *Autonome Mobile Systeme*. Springer Verlag, Berlin.

Hähnel, D., Burgard, W., Wegbreit, B., and Thrun, S. 2003a. Towards lazy data association in SLAM. *Proceedings of the 11th International Symposium of Robotics Research (ISRR'03)*, Sienna, Italy.

Hähnel, D., Fox, D., Burgard, W., and Thrun, S. 2003b. A highly efficient FastSLAM algorithm for generating cyclic maps of large-scale environments from raw laser range measurements. *Proceedings of the Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, October 27–31.

Hähnel, D., Triebel, R., Burgard, W., and Thrun, S. 2003c. Map building with mobile robots in dynamic environments. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan.

Julier, S. J., and Uhlmann, J. K. 2000. Building a million beacon map. *Sensor Fusion and Decentralized Control in Robotic Systems IV*, Bellingham, WA, *Proceedings of the SPIE* 4571.

Kuipers, B., and Byun, Y.-T. 1988. A robust qualitative method for spatial learning in unknown environments. *Proceeding of the 8th National Conference on Artificial Intelligence AAAI-88*, Menlo Park, Cambridge.

Kuipers, B., and Byun, Y.-T. 1991. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Journal of Robotics and Autonomous Systems* 8:47–63.

Leonard, J. J. and Durrant-Whyte, H. F. 1992. *Directed Sonar Sensing for Mobile Robot Navigation*. Kluwer Academic, Boston, MA.

Leonard, J. J. and Feder, H. J. S. 1999. A computationally efficient method for large-scale concurrent mapping and localization. *Proceedings of the 9th International Symposium on Robotics Research*, Salt Lake City, Utah.

Leonard, J., Tardós, J. D., Thrun, S., and Choset, H. editors. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Workshop Notes of the ICRA Workshop on Concurrent Mapping and Localization for Autonomous Mobile Robots (W4)*, Washington, DC, May 11–15.

Lomet, D. B. and Salzberg, B. 1990. The hb-tree: a multiattribute indexing method. *ACM Transactions on Database Systems* 15(4):625–658.

Lu, F., and Milios, E. 1997. Globally consistent range scan alignment for environment mapping. *Autonomous Robots* 4:333–349.

Matarić, M. J. 1990. *A Distributed Model for Mobile Robot Environment-Learning and Navigation*. Master's thesis, MIT, Cambridge, MA. Also available as MIT Artificial Intelligence Laboratory Tech Report AITR-1228.

Maybeck, P. 1979. *Stochastic Models, Estimation, and Control*, Vol. 1. Academic, New York.

Montemerlo, M., and Thrun, S. 2003. Simultaneous localization and mapping with unknown data association using FastSLAM. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan.

Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. 2002. FastSLAM: a factored solution to the simultaneous localization and mapping problem. *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada.

Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. 2003. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexico.

Moravec, H. P. 1988. Sensor fusion in certainty grids for mobile robots. *AI Magazine* 9(2):61–74.

Moutarlier, P., and Chatila, R. 1989. An experimental system for incremental environment modeling by an autonomous mobile robot. *Proceedings of the 1st International Symposium on Experimental Robotics*, Montreal, Canada, June.

Murphy, K. 2000. Bayesian map learning in dynamic environments. *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, Cambridge, MA.

Murphy, K. P., Weiss, Y., and Jordan, M. I. 1999. Loopy belief propagation for approximate inference: an empirical study. *Proceedings of the Conference on Uncertainty in AI (UAI)*, Stockholm, Sweden, pp. 467–475.

Neira, J., Tardós, J. D., and Castellanos, J. A. 2003. Linear time vehicle relocation in SLAM. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan.

Nettleton, E. W., Gibbens, P. W., and Durrant-Whyte, H. F. 2000. Closed form solutions to the multiple platform simultaneous localization and map building (slam) problem. *Sensor Fusion: Architectures, Algorithms, and Applications IV*, Bellingham, WA, Vol. 4051, Bulur V. Dasarathy, editor. pp. 428–437.

Nettleton, E., Durrant-Whyte, H., Gibbens, P., and Goktoğan, A. 2000. Multiple platform localization and map building. *Sensor Fusion and Decentralized Control in Robotic Systems III*, Bellingham, WA, Vol. 4196, G.T. McKee and P.S. Schenker, editors, pp. 337–347.

Nettleton, E., Thrun, S., and Durrant-Whyte, H. 2002. A constant time communications algorithm for decentralized SLAM. Submitted for publication.

Nettleton, E., Thrun, S., and Durrant-Whyte, H. 2003. Decentralized SLAM with low-bandwidth communication for teams of airborne vehicles. *Proceedings of the Interna-*

tional Conference on Field and Service Robotics, Lake Yamanaka, Japan.

Newman, P. 2000. *On the Structure and Solution of the Simultaneous Localization and Map Building Problem*. PhD thesis, Australian Centre for Field Robotics, University of Sydney, Sydney, Australia.

Paskin, M. A. 2002. Thin junction tree filters for simultaneous localization and mapping. Technical Report UCB/CSD-02-1198, University of California, Berkeley, CA.

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

Procopiuc, O., Agarwal, P. K., Arge, L., and Vitter, J. S. 2003. Bkd-tree: a dynamic scalable kd-tree. *Advances in spatial and temporal awareness*, T. Hadzilacos, Y. Manolopoulos, J. F. Roddick and Y. Theodoridis, editors. Springer-Verlag, Santorini Island, Greece.

Shatkay, H., and Kaelbling, L. 1997. Learning topological maps with weak local odometric information. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Simmons, R., Apfelbaum, D., Burgard, W., Fox, M., an Moors, D., Thrun, S., and Younes, H. 2000. Coordination for multi-robot exploration and mapping. *Proceedings of the AAAI National Conference on Artificial Intelligence*, Austin, TX.

Smith, R. C. and Cheeseman, P. 1985. On the representation and estimation of spatial uncertainty. Technical Report 4760 and 7239, SRI International, Menlo Park, CA.

Smith, R. C. and Cheeseman, P. 1986. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research* 5(4):56–68.

Smith, R., Self, M., and Cheeseman, P. 1990. Estimating uncertain spatial relationships in robotics. *Autonomous Robot Vehicles*, I.J. Cox and G.T. Wilfong, editors. Springer-Verlag, Berlin, pp. 167–193.

Stewart, B., Ko, J., Fox, D., and Konolige, K. 2003. A hierarchical bayesian approach to mobile robot map structure estimation. *Proceedings of the Conference on Uncertainty in AI (UAI)*, Acapulco, Mexico.

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

Thorpe, C. and Durrant-Whyte, H. 2001. Field robots. *Proceedings of the 10th International Symposium of Robotics Research (ISRR'01)*, Lorne, Australia.

Thrun, S. 2000. Towards programming tools for robots that integrate probabilistic computation and learning. *Proceed-*

ings of the IEEE International Conference on Robotics and Automation (ICRA), San Francisco, CA, April 24–28.

Thrun, S. 2001. A probabilistic on-line mapping algorithm for teams of mobile robots. *International Journal of Robotics Research* 20(5):335–363.

Thrun, S. 2002. Robotic mapping: a survey. *Exploring Artificial Intelligence in the New Millennium*, G. Lakemeyer and B. Nebel, editors. Morgan Kaufmann, San Mateo, CA.

Thrun, S., and Liu, Y. 2003. Multi-robot SLAM with sparse extended information filers. *Proceedings of the 11th International Symposium of Robotics Research (ISRR'03)*, Sienna, Italy.

Thrun, S., Fox, D., and Burgard, W. 1998. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning* 31:29–53. Also appeared in *Autonomous Robots* 5:253–271 (joint issue).

Thrun, S., Hähnel, D., Ferguson, D., Montemerlo, M., Triebel, R., Burgard, W., Baker, C., Omohundro, Z., Thayer, S., and Whittaker, W. 2003. A system for volumetric robotic mapping of abandoned mines. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan.

Wainwright, M. J. 2002. *Stochastic Processes on Graphs With Cycles: Geometric and Variational Approaches*. PhD thesis, Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA.

Wang, C.-C., Thorpe, C., and Thrun, S. 2003. On-line simultaneous localization and mapping with detection and tracking of moving objects: theory and results from a ground vehicle in crowded urban areas. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan.

Weiss, Y., and Freeman, W. T. 2001. Correctness of belief propagation in gaussian graphical models of arbitrary topology. *Neural Computation* 13(10):2173–2200.

Williams, S., and Dissanayake, G. 2002. Efficient simultaneous localisation and mapping using local submaps. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Workshop Notes of the ICRA Workshop on Concurrent Mapping and Localization for Autonomous Mobile Robots (W4)*, Washington, DC, May 11–15.

Williams, S. B., Dissanayake, G., and Durrant-Whyte, H. 2002. An efficient approach to the simultaneous localization and mapping problem. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Washington, DC, May 11–15, pp. 406–411.