

**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

### **Abstract**

This paper describes 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). This paper advocates 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 filters (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.

## **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 [9, 21]. 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 [47]. In recent years, the SLAM problem has received considerable attention by the scientific community, and a flurry of new algorithms and techniques has emerged [20].

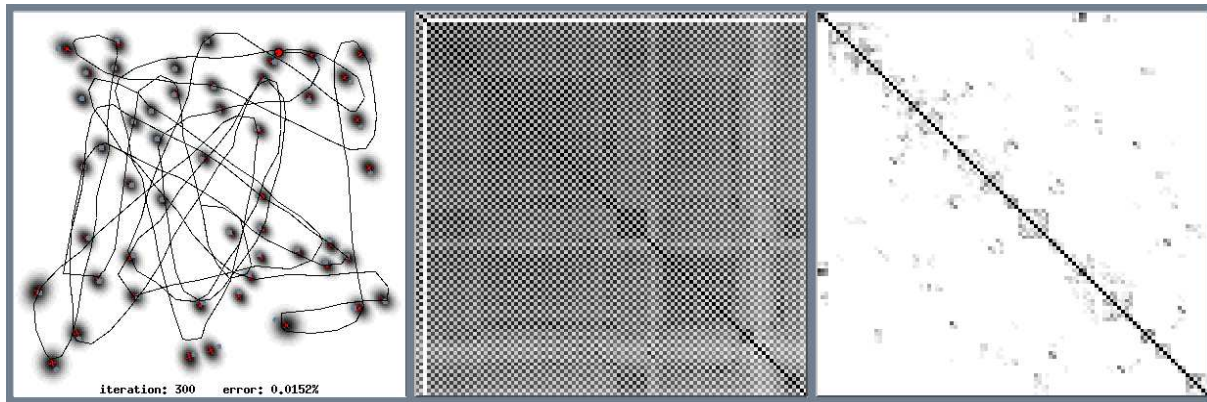
Existing algorithms can be subdivided into batch and online techniques. The former offer sophisticated techniques to cope with perceptual ambiguities [3, 41, 50], but they can only generate maps after extensive batch processing. Online techniques are specifically suited to acquire maps as the robot navigates [9, 44]. Online SLAM is of great practical importance

in many navigation and exploration problems [4, 42]. Today’s most widely used online algorithms are based on the extended Kalman filter (EKF), whose application to SLAM problems was developed in a series of seminal papers [30, 43, 44]. 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 [17].

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 [2, 11, 22, 46, 55], thereby gaining representational and computational efficiency. An alternative structured representations 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 [31, 27, 28]. 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.

This paper describes 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 [37] and Csorba [7, 8], and it is related to an algorithm by Lu and Milios [24]. Just as in recent work by Nettleton et al. [36], our approach is based on the well-known *information form* of the EKF, also known as the extended information filter (EIF) [26]. 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 [54]. The sparsity has important ramifications on the computational properties of solving SLAM problems.

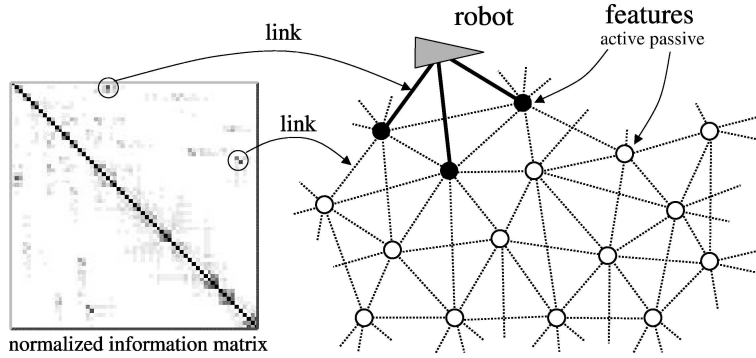
The use of sparse matrices, or local links, is motivated by a key insight: the posterior distribution in SLAM problems is dominated by a small number of relative links between adjacent features in the map. This is best illustrated through an example. Figure 1 shows the result of



**Figure 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.

the vanilla EKF [30, 44, 43] 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 [9]. 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 online 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 *information matrix* [26, 36]), 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 suggests 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



**Figure 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.

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 [45].

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 corresponds 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 require 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. Section 2 formally introduces the extended information filter (EIF), which forms the basis of our approach. SEIFs are described in Section 3, which states the major computational results of this paper. The section develops the constant time algorithm for maintaining sparse information matrices, and it also presents 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 [9, 11]. 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

This section reviews the extended information filter (EIF), which forms the basis of our work. EIFs are computationally equivalent to extended Kalman filters (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 and colleagues [33, 36], 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  $n$ -th 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 [16, 53] 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  $\xi_t = \begin{pmatrix} x_t & y_1 & \dots & y_N \end{pmatrix}^T$ , where the superscript  $T$  refers to the transpose of a vector.

In the SLAM problem, it is impossible to sense the state  $\xi_t$  directly—otherwise there would be no mapping problem. Instead, the robot seeks to recover a probabilistic estimate of  $\xi_t$ . Written in a Bayesian form, our goal shall be to calculate a posterior distribution over the state  $\xi_t$ . This posterior  $p(\xi_t \mid z^t, u^t)$  is conditioned on past sensor measurements  $z^t = z_1, \dots, z_t$  and past controls  $u^t = u_1, \dots, u_t$ . 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  $\xi_t$ . The mean of this distribution will be denoted  $\mu_t$ , and covariance matrix  $\Sigma_t$ :

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

The proportionality sign replaces a constant normalizer that is easily recovered from the covariance  $\Sigma_t$ . The representation of the posterior via the mean  $\mu_t$  and the covariance matrix  $\Sigma_t$  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  $\mu_t$  and  $\Sigma_t$ . These are obtained by multiplying out the exponent of (1):

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

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

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

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

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

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

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

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  $\mu_t$  and covariance  $\Sigma_t$ , 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:

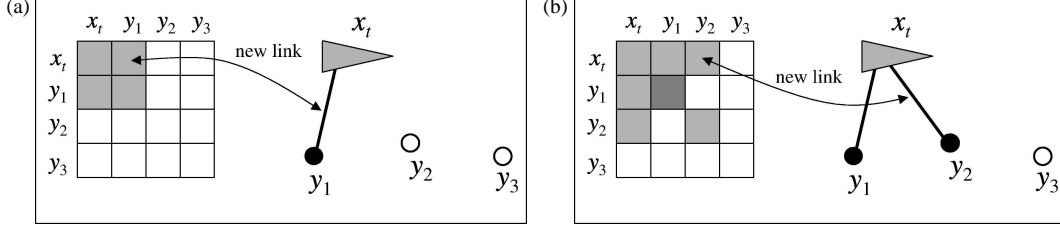
$$\Sigma_t = H_t^{-1} \quad \text{and} \quad \mu_t = H_t^{-1} b_t^T = \Sigma_t b_t^T \quad (6)$$

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 constraints 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.



**Figure 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.

## 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 3a. 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  $H_{x_t, y_1}$  and their symmetric counterparts  $H_{y_1, x_t}$  that link together  $x_t$  and  $y_1$ . 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 3b shows the incorporation of a second measurement of a different feature,  $y_2$ . In response to this measurement, the EIF updates the links  $H_{x_t, y_2} = H_{y_2, x_t}^T$  (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:

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

The second step of this derivation exploited common (and obvious) independences in SLAM problems [49]. For the time being, we assume that  $p(\xi_t \mid z^{t-1}, u^t)$  is represented by  $\bar{H}_t$  and  $\bar{b}_t$ . Those 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 nonlinear measurement function  $h$  with added Gaussian noise:

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

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

$$p(z_t | \xi_t) \propto \exp \left\{ -\frac{1}{2} (z_t - h(\xi_t))^T Z^{-1} (z_t - h(\xi_t)) \right\} \quad (10)$$

Following the rich literature of EKF, 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  $\xi$ , 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  $\mu_t$ . The transpose of the Jacobian matrix  $\nabla_{\xi} h(\mu_t)$  and will be denoted  $C_t$ . With these definitions, Equation (11) reads as follows:

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

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

$$p(z_t | \xi_t) \propto \exp \left\{ -\frac{1}{2} (z_t - \hat{z}_t - C_t^T \xi_t + C_t^T \mu_t)^T Z^{-1} (z_t - \hat{z}_t - C_t^T \xi_t + C_t^T \mu_t) \right\} \quad (13)$$

Multiplying out the exponent and regrouping the resulting terms gives us

$$\begin{aligned} &= \exp \left\{ -\frac{1}{2} \xi_t^T C_t Z^{-1} C_t^T \xi_t + (z_t - \hat{z}_t + C_t^T \mu_t)^T Z^{-1} C_t^T \xi_t \right. \\ &\quad \left. - \frac{1}{2} (z_t - \hat{z}_t + C_t^T \mu_t)^T Z^{-1} (z_t - \hat{z}_t + C_t^T \mu_t) \right\} \end{aligned} \quad (14)$$

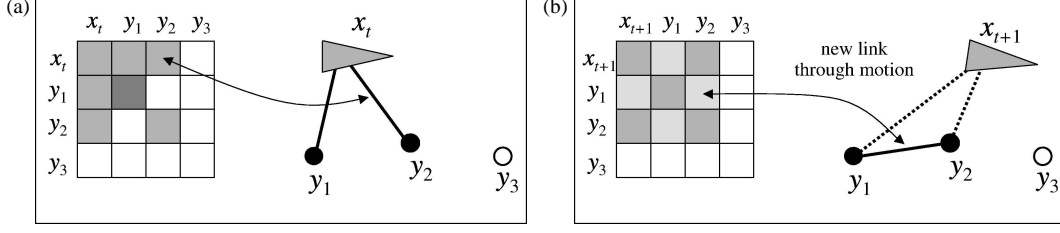
As before, the final term in the exponent does not depend on the variable  $\xi$  and hence can be subsumed into the proportionality factor:

$$\propto \exp \left\{ -\frac{1}{2} \xi_t^T C_t Z^{-1} C_t^T \xi_t + (z_t - \hat{z}_t + C_t^T \mu_t)^T Z^{-1} C_t^T \xi_t \right\} \quad (15)$$

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

$$\begin{aligned} p(\xi_t | z^t, u^t) &\propto \exp \left\{ -\frac{1}{2} \xi_t^T \bar{H}_t \xi_t + \bar{b}_t^T \xi_t \right\} \\ &\quad \cdot \exp \left\{ -\frac{1}{2} \xi_t^T C_t Z^{-1} C_t^T \xi_t + (z_t - \hat{z}_t + C_t^T \mu_t)^T Z^{-1} C_t^T \xi_t \right\} \\ &= \exp \left\{ -\frac{1}{2} \xi_t^T \underbrace{(\bar{H}_t + C_t Z^{-1} C_t^T)}_{H_t} \xi_t + \underbrace{(\bar{b}_t + (z_t - \hat{z}_t + C_t^T \mu_t)^T Z^{-1} C_t^T)}_{b_t} \xi_t \right\} \end{aligned} \quad (16)$$





**Figure 4:** The effect of motion on the information matrix and the associated network of features: (a) before motion, and (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.

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

$$H_t = \bar{H}_t + C_t Z^{-1} C_t^T \quad (17)$$

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

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  $t$ .

$$C_t = \left( \frac{\partial h}{\partial x_t} \quad 0 \dots 0 \quad \frac{\partial h}{\partial y_t} \quad 0 \dots 0 \right)^T \quad (19)$$

This well-known sparseness of  $C_t$  [9] 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  $C_t Z^{-1} C_t^T$  to the information matrix in (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 in 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  $H_t$  are slightly more complicated than that of measurements. Figure 4a 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 4b. 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 involves the calculation of the following posterior:

$$p(\xi_t | z^{t-1}, u^t) = \int p(\xi_t | \xi_{t-1}, z^{t-1}, u^t) p(\xi_{t-1} | z^{t-1}, u^t) d\xi_{t-1} \quad (20)$$

Exploiting the common SLAM independences [49] leads to

$$p(\xi_t | z^{t-1}, u^t) = \int p(\xi_t | \xi_{t-1}, u_t) p(\xi_{t-1} | z^{t-1}, u^{t-1}) d\xi_{t-1} \quad (21)$$

The term  $p(\xi_{t-1} | z^{t-1}, u^{t-1})$  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  $p(\xi_t | \xi_{t-1}, u_t)$ , 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:

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

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  $\Delta_t$  constitutes the state change at time  $t$ . The stochastic part of this change is modeled by  $\delta_t$ , 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  $S_x = (I \ 0 \ \dots \ 0)^T$ , where  $I$  is an identity matrix of the same dimension as the robot pose vector  $x_t$  and as of  $\delta_t$ . 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, give the following generalized noise variable, enlarged to the dimension of the full state vector  $\xi$ :  $S_x \delta_t = (\delta_t \ 0 \ \dots \ 0)^T$ . In EIFs, the function  $g$  in (22) is approximated by its first degree Taylor series expansion:

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

Here  $A_t = \nabla_{\xi} g(\mu_{t-1}, u_t)$  is the derivative of  $g$  with respect to  $\xi$  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 (22)

leads to an approximation of  $\xi_t$ , the state at time  $t$ :

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

Hence, under this approximation the random variable  $\xi_t$  is again Gaussian distributed. Its mean is obtained by replacing  $\xi_t$  and  $\delta_t$  in (24) by their respective means:

$$\bar{\mu}_t = (I + A_t)\mu_{t-1} + \hat{\Delta}_t - A_t\mu_{t-1} + S_x 0 = \mu_{t-1} + \hat{\Delta}_t \quad (25)$$

The covariance of  $\xi_t$  is simply obtained by scaled and adding the covariance of the Gaussian variables on the right-hand side of (24):

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

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

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

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

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 *sparse extended information filter*, or *SEIF*. The SEIF differs from the extended information filter 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  $\theta_x$ ; the bound on the number of links for each feature (not counting the link to the robot) will be denoted  $\theta_y$ . The motivation for maintaining a sparse information is mainly computational, as will become apparent below. Its justification was already 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:*

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

This result addresses the important special case  $A_t = 0$ , that is, 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  $\mu_t$  is available for the robot pose and all active features. The constant-time update equations are given by:*

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

For  $A_t \neq 0$ , a constant time update requires knowledge of the mean  $\mu_{t-1}$  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 Equation (6)) requires more than constant time. A constant-time solution to this problem will now be presented.

### 3.2 Sparsification

The final step in SEIFs concerns the sparsification of the information matrix  $H_t$ . 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  $\theta_x$ . Second, motion introduces links between any two active features, and hence lead to violations of the bound  $\theta_y$ . 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.

We will now present a constant-time sparsification technique. To do so, it will prove useful to partition the set of all features into three disjoint subsets:

$$Y = Y^+ + Y^0 + Y^- \quad (31)$$

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). Finally,  $Y^-$  are all currently passive features.

The sparsification is best derived from first principles. If  $Y^+ \uplus Y^0$  contains all currently active features, the posterior can be factored as follows:

$$\begin{aligned} p(x_t, Y \mid z^t, u^t) &= p(x_t, Y^0, Y^+, Y^- \mid z^t, u^t) \\ &= p(x_t \mid Y^0, Y^+, Y^-, z^t, u^t) p(Y^0, Y^+, Y^- \mid z^t, u^t) \\ &= p(x_t \mid Y^0, Y^+, Y^- = 0, z^t, u^t) p(Y^0, Y^+, Y^- \mid z^t, u^t) \end{aligned} \quad (32)$$

In the last step we exploited 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  $Y^- = 0$ .

To sparsify the information matrix, the posterior is approximated by the following distribution, in which we simply drop the dependence on  $Y^0$  in the first term. It is easily shown that this distribution minimizes the KL divergence to the exact, non-sparse distribution:

$$\begin{aligned} \tilde{p}(x_t, Y \mid z^t, u^t) &= p(x_t \mid Y^+, Y^- = 0, z^t, u^t) p(Y^0, Y^+, Y^- \mid z^t, u^t) \\ &= \frac{p(x_t, Y^+ \mid Y^- = 0, z^t, u^t)}{p(Y^+ \mid Y^- = 0, z^t, u^t)} p(Y^0, Y^+, Y^- \mid z^t, u^t) \end{aligned} \quad (33)$$

This posterior is calculated in constant time. In particular, we begin by calculating the information matrix for the distribution  $p(x_t, Y^0, Y^+ \mid Y^- = 0)$  of all variables but  $Y^-$ , and conditioned

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

$$H'_t = S_{x,Y^+,Y^0} S_{x,Y^+,Y^0}^T H_t S_{x,Y^+,Y^0} S_{x,Y^+,Y^0}^T \quad (34)$$

With that, the *matrix inversion lemma*<sup>1</sup> leads to the following information matrices for the terms  $p(x_t, Y^+ | Y^- = 0, z^t, u^t)$  and  $p(Y^+ | Y^- = 0, z^t, u^t)$ , denoted  $H_t^1$  and  $H_t^2$ , respectively:

$$\begin{aligned} H_t^1 &= H'_t - H'_t S_{Y_0} (S_{Y_0}^T H'_t S_{Y_0})^{-1} S_{Y_0}^T H'_t \\ H_t^2 &= H'_t - H'_t S_{x,Y_0} (S_{x,Y_0}^T H'_t S_{x,Y_0})^{-1} S_{x,Y_0}^T H'_t \end{aligned} \quad (36)$$

Here the various  $S$ -matrices are projection matrices, analogous to the matrix  $S_x$  defined above. The final term in our approximation (33),  $p(Y^0, Y^+, Y^- | z^t, u^t)$ , has the following information matrix:

$$H_t^3 = H_t - H_t S_{x_t} (S_{x_t}^T H_t S_{x_t})^{-1} S_{x_t}^T H_t \quad (37)$$

Putting these expressions together according to Equation (33) yields the following information matrix, in which the feature  $Y^0$  is now indeed deactivated:

$$\begin{aligned} \tilde{H}_t &= H_t^1 - H_t^2 + H_t^3 = H_t - H'_t S_{Y_0} (S_{Y_0}^T H'_t S_{Y_0})^{-1} S_{Y_0}^T H'_t \\ &\quad + H'_t S_{x,Y_0} (S_{x,Y_0}^T H'_t S_{x,Y_0})^{-1} S_{x,Y_0}^T H'_t - H_t S_{x_t} (S_{x_t}^T H_t S_{x_t})^{-1} S_{x_t}^T H_t \end{aligned} \quad (38)$$

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

$$\begin{aligned} \tilde{b}_t &= \mu_t^T \tilde{H}_t = \mu_t^T (H_t - H_t + \tilde{H}_t) \\ &= \mu_t^T H_t + \mu_t^T (\tilde{H}_t - H_t) = b_t + \mu_t^T (\tilde{H}_t - H_t) \end{aligned} \quad (39)$$

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  $\mu_t$  for all active features, which is obtained via the approximation technique described in the previous section. From (39), it is obvious that the sparsification does not affect the mean  $\mu_t$ , that is,  $H_t^{-1} b_t^T = [\tilde{H}_t]^{-1} [\tilde{b}_t]^T$ . Furthermore, our approximation minimizes the KL divergence to the correct posterior. This property is essential for the consistency of our approximation.

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. Empirically, it induces approximation errors that are negligible for appropriately chosen sparseness constraints  $\theta_x$  and  $\theta_y$ .

---

<sup>1</sup>The matrix inversion lemma (Sherman-Morrison-Woodbury formula), as used throughout this article, is stated as follows:

$$(H^{-1} + SBS^T)^{-1} = H - HS(B^{-1} + S^T HS)^{-1} S^T H \quad (35)$$

### 3.3 Amortized Approximate Map Recovery

Before deriving an algorithm for recovering the state estimate  $\mu_t$  from the information form, let us briefly consider what parts of  $\mu_t$  are needed in SEIFs, and when. SEIFs need the state estimate  $\mu_t$  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, and (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  $\mu_t$ , namely the estimate of the robot pose and of the locations of active features.

As stated in (6), the mean vector  $\mu_t$  is a function of  $H_t$  and  $b_t$ :

$$\mu_t = H_t^{-1}b_t^T = \Sigma_t b_t^T \quad (40)$$

Unfortunately, calculating Equation (40) directly involves inverting a large matrix, which would require 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 is being gathered and the estimates  $b$  and  $H$  are being constructed. To do so, it will prove convenient to pose (40) as an optimization problem:

**Lemma 4:** *The state  $\mu_t$  is the mode  $\hat{\nu}_t := \operatorname{argmax}_{\nu_t} p(\nu_t)$  of the Gaussian distribution, defined over the variable  $\nu_t$ :*

$$p(\nu_t) = \text{const.} \cdot \exp \left\{ -\frac{1}{2} \nu_t^T H_t \nu_t + b_t^T \nu_t \right\} \quad (41)$$

Here  $\nu_t$  is a vector of the same form and dimensionality as  $\mu_t$ . This lemma suggests that recovering  $\mu_t$  is equivalent to finding the mode of (41). 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.

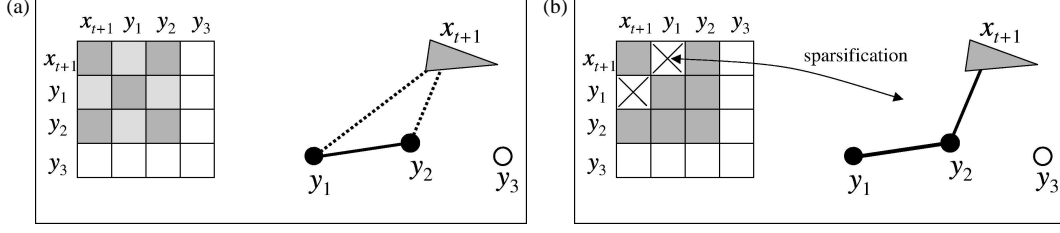
Our approach is an instantiation of coordinate descent. For simplicity, we state it here for a single coordinate only; our implementation iterates a constant number  $K$  of such optimizations after each measurement update step. The mode  $\hat{\nu}_t$  of (41) is attained at:

$$\begin{aligned} \hat{\nu}_t &= \operatorname{argmax}_{\nu_t} p(\nu_t) = \operatorname{argmax}_{\nu_t} \exp \left\{ -\frac{1}{2} \nu_t^T H_t \nu_t + b_t^T \nu_t \right\} \\ &= \operatorname{argmin}_{\nu_t} \frac{1}{2} \nu_t^T H_t \nu_t - b_t^T \nu_t \end{aligned} \quad (42)$$

We note that the argument of the min-operator in (42) can be written in a form that makes the individual coordinate variables  $\nu_{i,t}$  (for the  $i$ -th coordinate of  $\nu_t$ ) explicit:

$$\frac{1}{2} \nu_t^T H_t \nu_t - b_t^T \nu_t = \frac{1}{2} \sum_i \sum_j \nu_{i,t}^T H_{i,j,t} \nu_{j,t} - \sum_i b_{i,t}^T \nu_{i,t} \quad (43)$$

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



**Figure 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.

$\nu_{i,t}$  gives us

$$\frac{\partial}{\partial \nu_{i,t}} \left\{ \frac{1}{2} \sum_i \sum_j \nu_{i,t}^T H_{i,j,t} \nu_{j,t} - \sum_i b_{i,t}^T \nu_{i,t} \right\} = \sum_j H_{i,j,t} \nu_{j,t} - b_{i,t}^T \quad (44)$$

Setting this to zero leads to the optimum of the  $i$ -th coordinate variable  $\nu_{i,t}$  given all other estimates  $\nu_{j,t}$ :

$$\nu_{i,t}^{[k+1]} = H_{i,i,t}^{-1} \left[ b_{i,t}^T - \sum_{j \neq i} H_{i,j,t} \nu_{j,t}^{[k]} \right] \quad (45)$$

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

$$\nu_{i,t}^{[k+1]} = (S_i^T H_t S_i)^{-1} S_i^T \left[ b_t - H_t \nu_t^{[k]} + H_t S_i S_i^T \nu_t^{[k]} \right] \quad (46)$$

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

As is easily seen, the number of elements in the summation in (45), and hence the vector multiplication in (46), 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 has been proposed [9, 28, 46]. Here we follow the standard maximum likelihood approach described in [9]. 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  $1, \dots, N_{t-1} + 1$  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

$$p(\xi_t \mid z^t, u^t, \hat{n}^t) \quad (47)$$

Here  $\hat{n}^t$  is the sequence of the estimated values of the correspondence variables  $n^t$ . Notice that we chose 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:

$$\begin{aligned} \hat{n}_t &= \operatorname{argmax}_{n_t} p(z_t \mid z^{t-1}, u^t, \hat{n}^{t-1}, n_t) \\ &= \operatorname{argmax}_{n_t} \int p(z_t \mid \xi_t, n_t) \underbrace{p(\xi_t \mid z^{t-1}, u^t, \hat{n}^{t-1})}_{\bar{H}_t, \bar{b}_t} d\xi_t \\ &= \operatorname{argmax}_{n_t} \int \int p(z_t \mid x_t, y_{n_t}, n_t) p(x_t, y_{n_t} \mid z^{t-1}, u^t, \hat{n}^{t-1}) \end{aligned} \quad (48)$$

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.

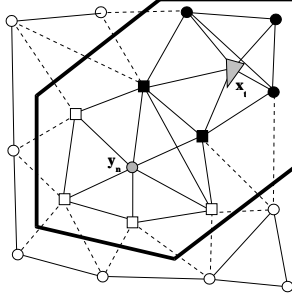
In particular, let us denote by  $Y_{n_t}^+$  the combined Markov blanket of the robot pose  $x_t$  and the landmark  $y_{n_t}$ . This Markov blanket is the set of all features in the map that are linked to the robot or landmark  $y_{n_t}$ . Figure 6 illustrates this set. Notice that  $Y_{n_t}^+$  includes 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  $Y_{n_t}^-$ , that is:

$$Y_{n_t}^- = Y - Y_{n_t}^+ - \{y_{n_t}\} \quad (49)$$

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 Equation (48) by essentially ignoring these indirect influences:

$$p(x_t, y_{n_t} \mid z^{t-1}, u^t, \hat{n}^{t-1})$$



**Figure 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  $p(x_t, y_n \mid z^{t-1}, u^t)$ .

$$\begin{aligned}
&= \int \int p(x_t, y_{n_t}, Y_{n_t}^+, Y_{n_t}^- \mid z^{t-1}, u^t, \hat{n}^{t-1}) dY_{n_t}^+ dY_{n_t}^- \\
&= \int \int p(x_t, y_{n_t} \mid Y_{n_t}^+, Y_{n_t}^-, z^{t-1}, u^t, \hat{n}^{t-1}) p(Y_{n_t}^+ \mid Y_{n_t}^-, z^{t-1}, u^t, \hat{n}^{t-1}) \\
&\quad p(Y_{n_t}^- \mid z^{t-1}, u^t, \hat{n}^{t-1}) dY_{n_t}^+ dY_{n_t}^- \\
&\approx \int p(x_t, y_{n_t} \mid Y_{n_t}^+, Y_{n_t}^- = \mu_n^-, z^{t-1}, u^t, \hat{n}^{t-1}) p(Y_{n_t}^+ \mid Y_{n_t}^- = \mu_n^-, z^{t-1}, u^t, \hat{n}^{t-1}) dY_{n_t}^+
\end{aligned} \tag{50}$$

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:

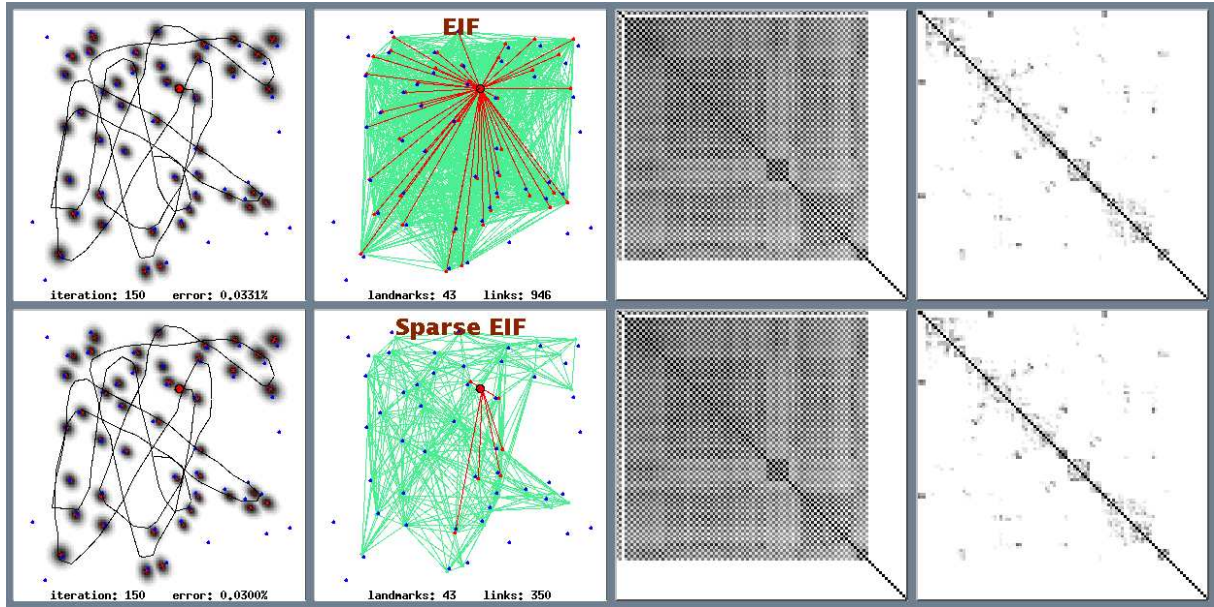
$$\begin{aligned}
\Sigma_{t:n_t} &= S_{x_t, y_n}^T (S_{x_t, y_n, Y_n^+}^T H_t S_{x_t, y_n, Y_n^+})^{-1} S_{x_t, y_n} \\
\mu_{t:n_t} &= \mu_t S_{x_t, y_n}
\end{aligned} \tag{51}$$

This calculation is constant time, since it involves a matrix whose size is independent of  $N$ . From this Gaussian, the desired measurement probability in Equation (48) 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  $3.4 \times 10^{-4}$ . 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  $\alpha$ . If the likelihood is smaller than  $\alpha$ , 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 is standard in the context of EKFs [9].

## 4.2 Map Management

Our exact mechanism for building up the map is closely related to common procedures in the SLAM community [9]. 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 can not 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



**Figure 7:** Comparison of EKFs with SEIFs 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 (red: between the robot and landmarks; green: between landmarks). The right panels show the resulting covariance and normalized information matrices for both approaches. Notice the similarity!

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 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 [23, 40], 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.



**Figure 8:** The vehicle used in our experiments is equipped with a 2D 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.

## 5 Experimental Results

The primary purpose of our experimental comparison was to evaluate the performance of the SEIF against that of the “gold standard” in SLAM, which is the EKF, from which the SEIF is derived. Some of our experiments rely on a benchmark dataset, commonly used to evaluate SLAM algorithms [9, 11]. Others rely on simulation, to facilitate more systematic evaluations.

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 unit for measuring steering angle and forward velocity. The laser is used to detect trees in the park, but it also picks up hundreds of spurious features such as corners of moving cars on a nearby highway. The raw odometry of the vehicle is extremely poor, resulting in several hundred meters of error when used for path integration along the vehicle’s 3.5km path. This is illustrated in Figure 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 dataset particularly amenable for testing SLAM algorithms.

The path recovered by the SEIF is shown in Figure 9(b). This path is visually indistinguishable from the one produced by the EKF and related variants [9, 11]. The average position error is smaller than 0.50 meters, which is vanishingly small compared to the overall path length of 3.5 km. Comparing with EKF, SEIF runs approximately twice as fast and consumes less than a quarter of the memory EKF uses. 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 result in noticeable savings both in memory and execution time.

In addition to the real-world data, we also used a robot simulator. The simulator has the advantage that we know the ground truth (which is unknown for the real-world data sets), and that it facilitates systematic experiments specifically with regards to scaling up SEIFs to large environments. In our simulations, we focused particularly on the *loop closing* problem, which is generally acknowledged to be one of the hardest problems in SLAM [13, 24]. When closing a loop, usually many landmark locations are affected, testing our amortized map recovery

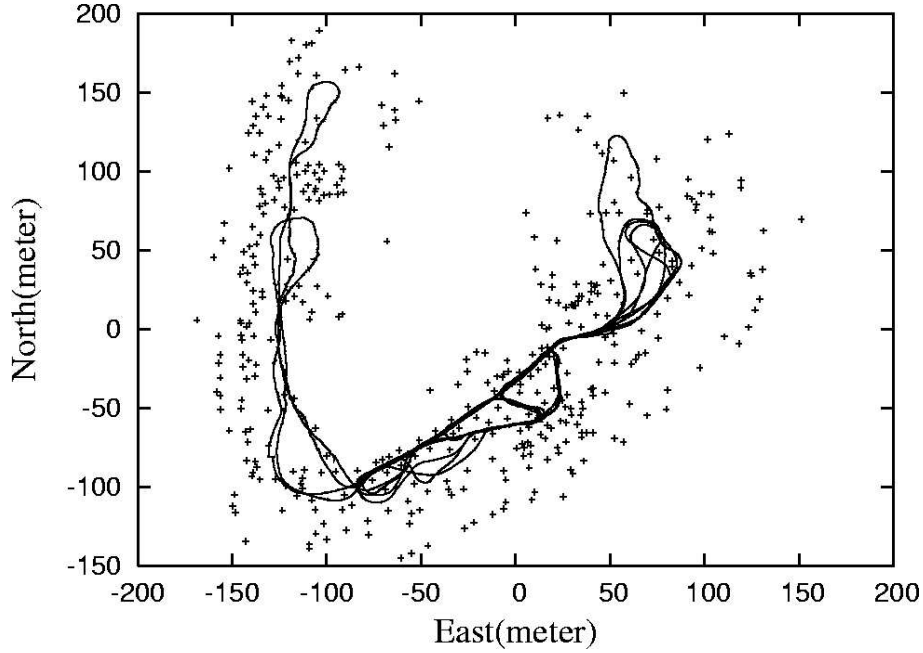


**Figure 9:** The testing environment: A 350 meters by 350 meters patch in Victoria Park in Sydney. (a) shows integrated path from odometry readings (b) shows the path as the result of SEIF.

mechanism under the hardest possible 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 density of landmarks (on average); as the number of landmarks is increased, so is the size of the environment. Each unit interval possesses 50 landmarks (on average). The landmarks are randomly distributed in a squared region with a minimum distance between any pair of landmarks. 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 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 6 landmark detections on average for one measurement step. The maximum number of active landmarks is always set to be 10, a parameter that worked well empirically.

Figure 11 and 12 show that SEIF beats EKF in terms of computation and memory usage. This does not surprise, given that SEIFs maintain sparse matrices. As predicted by the theory, EKF's usage of computation and memory both increase quadratically with the number of landmarks  $N$ . In SEIFs, the CPU time per iteration appears to stay roughly constant for maps with 300 or more landmark. The memory used to store the information matrix increases only linearly as expected. Due to the approximation of the information matrix and amortized map recovery, SEIF has bigger error than EKF. This is shown in Figure 13, which plots the empirical error as a



**Figure 10:** Overlay of estimated landmark positions and robot path.

function of the map size  $N$ . However, this increase in error is small, and can easily be justified by the reduced computation and memory costs when the map is large.

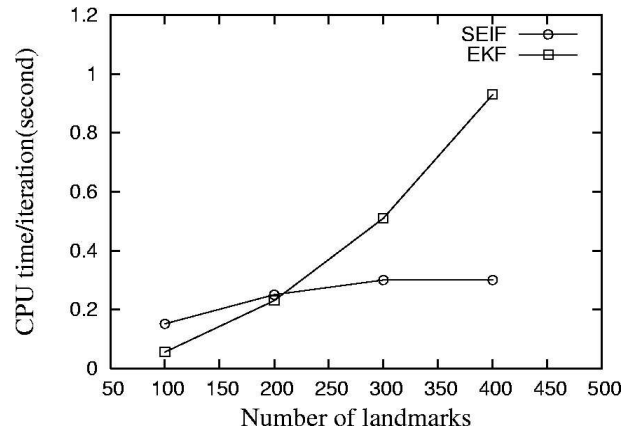
In another series of experiments, we were particularly interested in the source of possible approximation errors. In particular, we sought to elucidate what fraction of the residual error was due to the sparsification, and what fraction was due to the amortized recovery of the map. To this end we compared three algorithms: EKFs, SEIFs, and a variant of SEIFs in which the exact state estimate  $\mu_t$  is always available. The latter was implemented using matrix inversion, as described in Equation (6). Clearly, the recovery of the map does not run in constant time, but it always provides an exact result.

The following table depicts results for  $N = 50$  landmarks, after 500 update cycles, at which point all three approaches are near convergence.

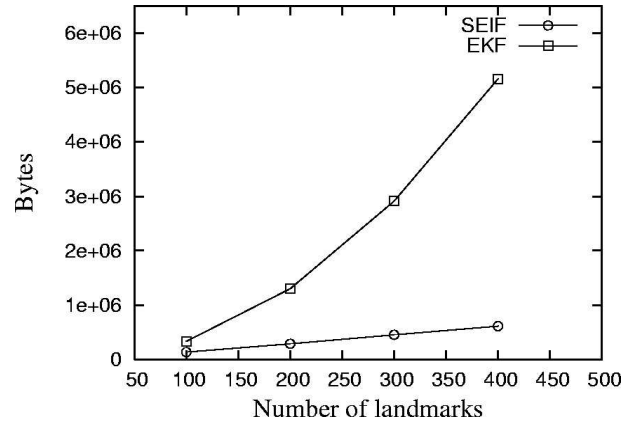
	# experiments	final error (with 95% conf. interval)	final # of links (with 95% conf. interval)	computation (per update)
EKF	1,000	$(5.54 \pm 0.67) \cdot 10^{-3}$	1,275	$O(N^2)$
SEIF with exact $\mu_t$	1,000	$(4.75 \pm 0.67) \cdot 10^{-3}$	$549 \pm 1.60$	$O(N^3)$
SEIF (constant time)	1,000	$(6.35 \pm 0.67) \cdot 10^{-3}$	$549 \pm 1.59$	$O(1)$

As these results suggest, our approach approximates EKF very tightly. The residual map error of our approach is with  $6.35 \cdot 10^{-3}$  approximately 14.6% higher than that of the extended Kalman filter. This error appears to be largely caused by the coordinate descent procedure, and is possibly inflated by the fact that  $K = 10$  is a small value given the size of the map. Enforcing the sparseness constraint seems not to have any negative effect on the overall error of the resulting map, as the results for our sparse filter implementation suggest.

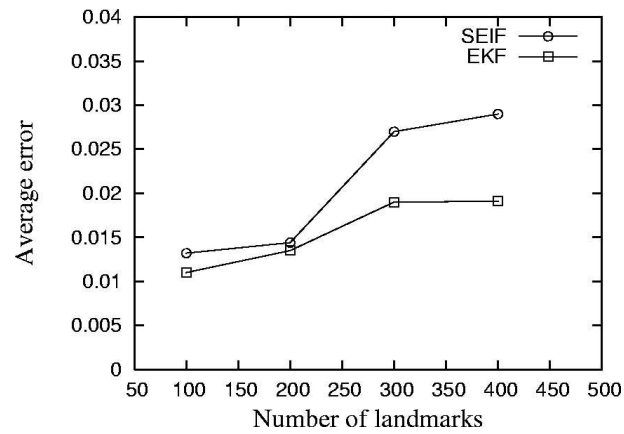
In a final series of experiments we applied SEIFs to a restricted version of the multi-robot SLAM problem, commonly studied in the literature [36]. In our implementation, the robots are



**Figure 11:** The comparison of average CPU time between SEIF and EKF.

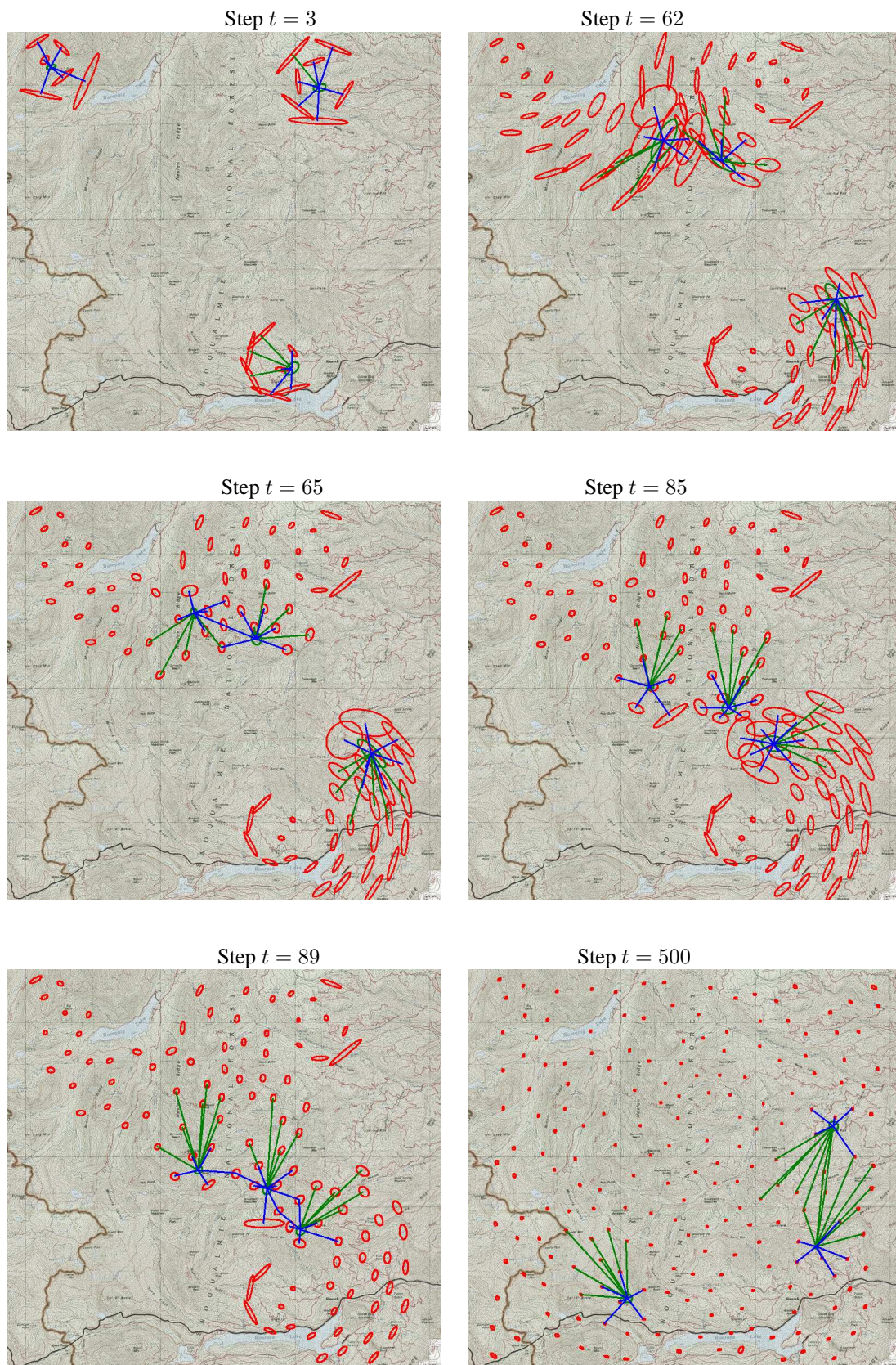


**Figure 12:** The comparison of average memory usage between SEIF and EKF.



**Figure 13:** The comparison of root mean square distance error between SEIF and EKF.





**Figure 14:** Snapshots from our multi-robot SLAM simulation at different points in time. Initially, the poses of the vehicles are known. During Steps 62 through 64, vehicle 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 through 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.



informed of their initial pose. This is a common assumption in multi-robot SLAM, necessary for the type linearization that is applied both in EKF and SEIFs [36].

Our simulation involves a team of three air vehicles. The vehicles are not equipped with GPS; hence they accrue positioning error over time. Figure 14 shows the joint map at different stages of the simulation. As in [36], 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 EKF 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 information in SEIFs is additive, whereas covariance matrices are not. SEIFs offer over the work in [36] that the messages sent between vehicles are small. A related approach for generating small messages in multi-vehicle SLAM has recently been described in [35].

Figure 14 shows a sequence of snapshots of the multi-vehicle system, using 3 different air vehicles. Initially, the vehicle start out in different areas, and the combined map (illustrated by the uncertainty ellipses) consists of three disjoint regions. During Steps 62 through 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 of the covariance matrix. Similarly, in steps 85 through 89, the third vehicle begins to 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 14 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 [1, 2, 11, 22, 46, 55]. 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 [22] 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 [46, 56] 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 [10]. The FastSLAM algorithm [15, 27, 28] and earlier related mapping algorithms [31, 48] 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 [27], 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 [38]. 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 article, the idea of representing maps by relative information has previously been explored by a number of authors, most notably in recent algorithms by Newman [37] and Csorba [7, 8]; it is also related to an earlier algorithm by Lu and Milios [24, 14]. Newman’s 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 applicable online. 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 et al. [36], our approach is based on the information form of the EKF [26], 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 [54] (GMRF). 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 [32, 39, 52]. Also applicable is the rich literature on sparse matrix transformations [12].

## 7 Discussion

This paper proposed an efficient algorithm for the SLAM problem. Our approach is based on the well-known information form of the extended Kalman filter. 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 *sparse* extended information filter, or 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 proposed an amortized constant-time coordinate descent algorithm for recovering these state estimates from the information form. We 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, as 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* [22, 17]. The overconfidence mainly arises from the approximation in the sparsification step. Such overconfidence is not necessarily an problem for the convergence of the approach [28], 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 [53]; 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 [15, 28, 31] 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 [51].

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, composes submaps dynamically, whereas past work relied on the definition of static submaps. We conjecture that

our sparse network structures capture the natural dependencies 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 [34].

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 [6, 18]. Links in SEIFs capture relative information, in that they relate the location of one landmark to another (see also [7, 8, 37]). This is a common characteristic of topological map representations [5, 41, 19, 25]. 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 [29, 45]. Thus, SEIFs bring together aspects of both paradigms, by defining simple computational operations for changing relative to absolute representations, and vice versa.

## Acknowledgment

The authors would like to acknowledge invaluable contributions by the following researchers: Wolfram Burgard, Geoffrey Gordon, Tom Mitchell, Kevin Murphy, Eric Nettleton, Michael Stevens, and Ben Wegbreit. This research has been sponsored by DARPA's MARS Program (contracts N66001-01-C-6018 and NBCH1020014), DARPA's CoABS Program (contract F30602-98-2-0137), and DARPA's MICA Program (contract F30602-01-C-0219), all of which is gratefully acknowledged. The authors furthermore acknowledge support provided by the National Science Foundation (CAREER grant number IIS-9876136 and regular grant number IIS-9877033).

## References

- [1] T. Bailey. *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments*. PhD thesis, University of Sydney, Sydney, NSW, Australia, 2002.
- [2] M. Bosse, J. Leonard, and S. Teller. Large-scale CML using a network of multiple local maps. In J. Leonard, J.D. Tardós, S. Thrun, and H. Choset, editors, *Workshop Notes of the ICRA Workshop on Concurrent Mapping and Localization for Autonomous Mobile Robots (W4)*, Washington, DC, 2002. ICRA Conference.
- [3] W. Burgard, D. Fox, H. Jans, C. Matenar, and S. Thrun. Sonar-based mapping of large-scale mobile robot environments using EM. In *Proceedings of the International Conference on Machine Learning*, Bled, Slovenia, 1999.

- [4] W. Burgard, D. Fox, M. Moors, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, San Francisco, CA, 2000. IEEE.
- [5] H. Choset. *Sensor Based Motion Planning: The Hierarchical Generalized Voronoi Graph*. PhD thesis, California Institute of Technology, 1996.
- [6] E. Chown, S. Kaplan, and D. Kortenkamp. Prototypes, location, and associative networks (plan): Towards a unified theory of cognitive mapping. *Cognitive Science*, 19:1–51, 1995.
- [7] M. Csorba. *Simultaneous Localization and Map Building*. PhD thesis, Department of Engineering Science, University of Oxford, Oxford, UK, 1997.
- [8] M. Deans and M. Hebert. Invariant filtering for simultaneous localization and mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1042–1047, San Francisco, CA, 2000. IEEE.
- [9] G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localisation and map building (SLAM) problem. *IEEE Transactions of Robotics and Automation*, 2001. In Press.
- [10] A. Doucet, J.F.G. de Freitas, and N.J. Gordon, editors. *Sequential Monte Carlo Methods In Practice*. Springer Verlag, New York, 2001.
- [11] J. Guivant and E. Nebot. Optimization of the simultaneous localization and map building algorithm for real time implementation. *IEEE Transactions of Robotics and Automation*, May 2001. In press.
- [12] Anshul Gupta, George Karypis, and Vipin Kumar. Highly scalable parallel algorithms for sparse matrix factorization. *IEEE Transactions on Parallel and Distributed Systems*, 8(5):502–520, 1997.
- [13] J.-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, 2000.
- [14] J.-S. Gutmann and B. Nebel. Navigation mobiler roboter mit laserscans. In *Autonome Mobile Systeme*. Springer Verlag, Berlin, 1997.
- [15] D. Hähnel, D. Fox, W. Burgard, and S. Thrun. A highly efficient FastSLAM algorithm for generating cyclic maps of large-scale environments from raw laser range measurements. Submitted for publication, March 2003.

- [16] D. Hähnel, R. Triebel, W. Burgard, and S. Thrun. Map building with mobile robots in dynamic environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2003.
- [17] S.J. Julier and J. K. Uhlmann. Building a million beacon map. In *Proceedings of the SPIE Sensor Fusion and Decentralized Control in Robotic Systems IV*, Vol. #4571, 2000.
- [18] B. Kuipers and Y.-T. Byun. A robust qualitative method for spatial learning in unknown environments. In *Proceeding of Eighth National Conference on Artificial Intelligence AAAI-88*, Menlo Park, Cambridge, 1988. AAAI Press / The MIT Press.
- [19] B. Kuipers and Y.-T. Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Journal of Robotics and Autonomous Systems*, 8:47–63, 1991.
- [20] J. Leonard, J.D. Tardós, S. Thrun, and H. Choset, editors. *Workshop Notes of the ICRA Workshop on Concurrent Mapping and Localization for Autonomous Mobile Robots (W4)*. ICRA Conference, Washington, DC, 2002.
- [21] J. J. Leonard and H. F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation*. Kluwer Academic Publishers, Boston, MA, 1992.
- [22] J.J. Leonard and H.J.S. Feder. A computationally efficient method for large-scale concurrent mapping and localization. In J. Hollerbach and D. Koditschek, editors, *Proceedings of the Ninth International Symposium on Robotics Research*, Salt Lake City, Utah, 1999.
- [23] D. B. Lomet and B. Salzberg. The hb-tree: A multiattribute indexing method. *ACM Transactions on Database Systems*, 15(4):625–658, 1990.
- [24] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.
- [25] M. J. Matarić. A distributed model for mobile robot environment-learning and navigation. Master’s thesis, MIT, Cambridge, MA, January 1990. also available as MIT Artificial Intelligence Laboratory Tech Report AITR-1228.
- [26] P. Maybeck. *Stochastic Models, Estimation, and Control, Volume I*. Academic Press, Inc, 1979.
- [27] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada, 2002. AAAI.

- [28] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexico, 2003. IJCAI.
- [29] H. P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, 9(2):61–74, 1988.
- [30] P. Moutarlier and R. Chatila. An experimental system for incremental environment modeling by an autonomous mobile robot. In *1st International Symposium on Experimental Robotics*, Montreal, June 1989.
- [31] K. Murphy. Bayesian map learning in dynamic environments. In *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 1999.
- [32] K.P. Murphy, Y. Weiss, and M.I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Conference on Uncertainty in AI (UAI)*, pages 467–475, 1999.
- [33] E. Nettleton, H. Durrant-Whyte, P. Gibbens, and A. Goktoğın. Multiple platform localisation and map building. In G.T. McKee and P.S. Schenker, editors, *Sensor Fusion and Decentralised Control in Robotic Systems III*, volume 4196, pages 337–347, Bellingham, 2000.
- [34] E. Nettleton, S. Thrun, and H. Durrant-Whyte. A constant time communications algorithm for decentralised SLAM. Submitted for publication, 2002.
- [35] E. Nettleton, S. Thrun, and H. Durrant-Whyte. Decentralised slam with low-bandwidth communication for teams of airborne vehicles. In *Proceedings of the International Conference on Field and Service Robotics*, Lake Yamanaka, Japan, 2003.
- [36] E.W. Nettleton, P.W. Gibbens, and H.F. Durrant-Whyte. Closed form solutions to the multiple platform simultaneous localisation and map building (slam) problem. In Bulur V. Dasarathy, editor, *Sensor Fusion: Architectures, Algorithms, and Applications IV*, volume 4051, pages 428–437, Bellingham, 2000.
- [37] P. Newman. *On the Structure and Solution of the Simultaneous Localisation and Map Building Problem*. PhD thesis, Australian Centre for Field Robotics, University of Sydney, Sydney, Australia, 2000.
- [38] M.A. Paskin. Thin junction tree filters for simultaneous localization and mapping. Technical Report UCB/CSD-02-1198, University of California, Berkeley, CA, 2002.
- [39] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers, San Mateo, CA, 1988.

- [40] O. Procopiuc, P.K. Agarwal, L. Arge, and J.S. Vitter. Bkd-tree: A dynamic scalable kd-tree. Submitted for publication, 2002.
- [41] H Shatkay and L. Kaelbling. Learning topological maps with weak local odometric information. In *Proceedings of IJCAI-97*. IJCAI, Inc., 1997.
- [42] R. Simmons, D. Apfelbaum, W. Burgard, M. Fox, D. an Moors, S. Thrun, and H. Younes. Coordination for multi-robot exploration and mapping. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Austin, TX, 2000. AAAI.
- [43] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I.J. Cox and G.T. Wilfong, editors, *Autonomous Robot Vehicles*, pages 167–193. Springer-Verlag, 1990.
- [44] R. C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. Technical Report 4760 & 7239, SRI International, Menlo Park, CA, 1985.
- [45] R.C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68, 1986.
- [46] J.D. Tardós, J. Neira, P.M. Newman, and J.J. Leonard. Robust mapping and localization in indoor environments using sonar data. *Int. J. Robotics Research*, 21(4):311–330, April 2002.
- [47] C. Thorpe and H. Durrant-Whyte. Field robots. In *Proceedings of the 10th International Symposium of Robotics Research (ISRR’01)*, Lorne, Australia, 2001.
- [48] S. Thrun. A probabilistic online mapping algorithm for teams of mobile robots. *International Journal of Robotics Research*, 20(5):335–363, 2001.
- [49] S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002. to appear.
- [50] S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31:29–53, 1998. also appeared in *Autonomous Robots* 5, 253–271 (joint issue).
- [51] S. Thrun, D. Hähnel, D. Ferguson, M. Montemerlo, R. Triebel, W. Burgard, C. Baker, Z. Omohundro, S. Thayer, and W. Whittaker. A system for volumetric robotic mapping of abandoned mines. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2003.
- [52] M. J. Wainwright. *Stochastic processes on graphs with cycles: geometric and variational approaches*. PhD thesis, Dept. of Electrical Engineering and Computer Science, MIT, Cambridge, MA, January 2002.



- [53] C.-C. Wang, C. Thorpe, and S. Thrun. Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2003.
- [54] Y. Weiss and W.T. Freeman. Correctness of belief propagation in gaussian graphical models of arbitrary topology. *Neural Computation*, 13(10):2173–2200, 2001.
- [55] S. Williams and G. Dissanayake. Efficient simultaneous localisation and mapping using local submaps. In J. Leonard, J.D. Tardós, S. Thrun, and H. Choset, editors, *Workshop Notes of the ICRA Workshop on Concurrent Mapping and Localization for Autonomous Mobile Robots (W4)*, Washington, DC, 2002. ICRA Conference.
- [56] S.B. Williams, G. Dissanayake, and H. Durrant-Whyte. An efficient approach to the simultaneous localisation and mapping problem. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 406–411, Washington, DC, 2002. ICRA.

## Appendix: Proofs

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

$$H_t = \bar{H}_t + C_t Z^{-1} C_t^T \quad (52)$$

$$b_t = \bar{b}_t + (z_t - \hat{z}_t + C_t^T \mu_t)^T Z^{-1} C_t^T \quad (53)$$

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  $b_t$ . Q.E.D.

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

$$\bar{H}_t = [H_{t-1}^{-1} + S_x U_t S_x^T]^{-1} \quad (54)$$

Applying the matrix inversion lemma leads to the following form:

$$\begin{aligned} \bar{H}_t &= H_{t-1} - H_{t-1} \underbrace{S_x [U_t^{-1} + S_x^T H_{t-1} S_x]^{-1} S_x^T H_{t-1}}_{=: L_t} \\ &= H_{t-1} - H_{t-1} L_t \end{aligned} \quad (55)$$

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 (28) and (55):

$$\begin{aligned}\bar{b}_t &= [b_{t-1}H_{t-1}^{-1} + \hat{\Delta}_t^T]\bar{H}_t \\ &= [b_{t-1}H_{t-1}^{-1} + \hat{\Delta}_t^T](H_{t-1} - H_{t-1}L_t) \\ &= b_{t-1} + \hat{\Delta}_t^T H_{t-1} - b_{t-1}L_t + \hat{\Delta}_t^T H_{t-1}L_t\end{aligned}\tag{56}$$

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. *Q.E.D.*

**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:

$$\begin{aligned}\Psi_t &= (I + S_x S_x^T A_t S_x S_x^T)^{-1} \\ &= I - I S_x (S_x I S_x^T + [S_x^T A_t S_x]^{-1})^{-1} S_x^T I \\ &= I - S_x (I + [S_x^T A_t S_x]^{-1})^{-1} S_x^T\end{aligned}\tag{57}$$

Notice that  $\Psi_t$  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 (57) involves a low-dimensional matrix.

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

$$\begin{aligned}\bar{H}_t &= [(I + A_t)H_{t-1}^{-1}(I + A_t)^T + S_x U_t S_x^T]^{-1} \\ &= [\underbrace{(\Psi_t^T H_{t-1} \Psi_t)^{-1}}_{=: H'_{t-1}} + S_x U_t S_x^T]^{-1} \\ &= [(H'_{t-1})^{-1} + S_x U_t S_x^T]^{-1} \\ &= H'_{t-1} - \underbrace{H'_{t-1} S_x [U_t^{-1} + S_x^T H'_{t-1} S_x]^{-1} S_x^T H'_{t-1}}_{=: \Delta H_t} \\ &= H'_{t-1} - \Delta H_t\end{aligned}\tag{58}$$

The matrix  $H'_{t-1} = \Psi_t^T H_{t-1} \Psi_t$  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:

$$\begin{aligned}\bar{b}_t &= [b_{t-1}H_{t-1}^{-1} + \hat{\Delta}_t^T]\bar{H}_t \\ &= b_{t-1}H_{t-1}^{-1}\bar{H}_t + \hat{\Delta}_t^T \bar{H}_t\end{aligned}$$

$$\begin{aligned}
&= b_{t-1} H_{t-1}^{-1} (\bar{H}_t + \underbrace{H_{t-1} - H_{t-1}}_{=0} + \underbrace{H'_{t-1} - H'_{t-1}}_{=0}) + \hat{\Delta}_t^T \bar{H}_t \\
&= b_{t-1} H_{t-1}^{-1} (H_{t-1} + \underbrace{\bar{H}_t - H'_{t-1}}_{-\Delta H_t} - H_{t-1} + H'_{t-1}) + \hat{\Delta}_t^T \bar{H}_t \\
&= b_{t-1} H_{t-1}^{-1} (H_{t-1} - \Delta H_t - H_{t-1} + H'_{t-1}) + \hat{\Delta}_t^T \bar{H}_t \\
&= b_{t-1} - b_{t-1} H_{t-1}^{-1} (\Delta H_t - H_{t-1} + H'_{t-1}) + \hat{\Delta}_t^T \bar{H}_t \\
&= b_{t-1} - \mu_{t-1}^T H_{t-1} H_{t-1}^{-1} (\Delta H_t - H_{t-1} + H'_{t-1}) + \hat{\Delta}_t^T \bar{H}_t \\
&= b_{t-1} - \mu_{t-1}^T (\Delta H_t - H_{t-1} + H'_{t-1}) + \hat{\Delta}_t^T \bar{H}_t \tag{59}
\end{aligned}$$

The update  $\Delta H_t$  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  $\mu_{t-1}$  are necessary to calculate the product  $\mu_{t-1}^T \Delta H_t$ . *Q.E.D.*

Proof of Lemma 4: The mode  $\hat{\nu}_t$  of (41) is given by

$$\begin{aligned}
\hat{\nu}_t &= \operatorname{argmax}_{\nu_t} p(\nu_t) \\
&= \operatorname{argmax}_{\nu_t} \exp \left\{ -\frac{1}{2} \nu_t^T H_t \nu_t + b_t^T \nu_t \right\} \\
&= \operatorname{argmin}_{\nu_t} \frac{1}{2} \nu_t^T H_t \nu_t - b_t^T \nu_t \tag{60}
\end{aligned}$$

The gradient of the expression inside the minimum in (60) with respect to  $\nu_t$  is given by

$$\frac{\partial}{\partial \nu_t} \left\{ \frac{1}{2} \nu_t^T H_t \nu_t - b_t^T \nu_t \right\} = H_t \nu_t - b_t^T \tag{61}$$

whose minimum  $\hat{\nu}_t$  is attained when the derivative (61) is 0, that is,

$$\hat{\nu}_t = H_t^{-1} b_t^T \tag{62}$$

From this and Equation (40) it follows that  $\hat{\nu}_t = \mu_t$ .

*Q.E.D.*