Simultaneous Mapping and Localization With Sparse Extended Information Filters: Theory and Initial Results

Sebastian Thrun[1], Daphne Koller[2], Zoubin Ghahramani[3], Hugh Durrant-Whyte[4], and Andrew Y. Ng[2]

[1] Carnegie Mellon University, Pittsburgh, PA, USA
[2] Stanford University, Stanford, CA, USA
[3] Gatsby Computational Neuroscience Unit, University College London, UK
[4] 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 determining the location of environmental features with a roving robot. Many of today's popular techniques are based on extended Kalman filters (EKFs), which require update time quadratic in the number of features in the map. This paper develops the notion of sparse extended information filters (SEIFs), as a new method for solving the SLAM problem. SEIFs exploit structure inherent in the SLAM problem, representing maps through local, Web-like networks of features. By doing so, updates can be performed in constant time, irrespective of the number of features in the map. This paper presents several original constant-time results of SEIFs, and provides simulation results that show the high accuracy of the resulting maps in comparison to the computationally more cumbersome EKF solution.

1 Introduction

The simultaneous localization and mapping (SLAM) problem is the problem of acquiring a map of an unknown environment with a moving robot, while simultaneously localizing the robot relative to this map [6,12]. The SLAM problem addresses situations where the robot lacks a global positioning sensor, and instead has to rely on a sensor of incremental ego-motion 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. Within mobile robotics, the SLAM problem is often referred to as one of the most challenging ones [30].

In recent years, the SLAM problem has received considerable attention by the scientific community, and a flurry of new algorithms and techniques has emerged, as attested, for example, by a recent workshop on this topic [11]. Existing algorithms can be subdivided into batch and online techniques. The former provide sophisticated techniques to cope with perceptual ambiguities [2,25,33], but they can only generate maps after extensive batch processing. Online techniques are specifically suited to acquire maps as the robot navigates [6,28], which is of great practical importance in many navigation and exploration problems [26]. Today's most widely used online algorithms are based on extended Kalman filters (EKFs), based on a seminal series of papers [18,28,27]. EKFs calculate Gaussian posteriors over the locations of environmental features and the robot itself.

A key bottleneck of EKFs—which has been subject to intense research—is their computational complexity. The standard EKF approach requires time quadratic in the number of features in the map, for each incremental update. This computational burden restricts EKFs to relatively sparse maps with no more than a few hundred features. Recently, several researchers have developed hierarchical techniques that decompose maps into collections of smaller, more manageable submaps [1,8,34]. While in principle, hierarchical techniques can solve this problem in linear time, many of these techniques still require quadratic time per update. One recent technique updates the estimate in constant time [13] 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. 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 [29,35] can achieve the same rate of convergence as the full EKF solution, but incur a $\mathcal{O}(n^2)$ computational burden. A different line of research has relied on particle filters for efficient mapping [7]. The FastSLAM algorithm [17] and related mapping algorithms [19] 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), whose scaling with environmental size is still poorly understood. None of these approaches, however, offer constant time updating while simultaneously maintaining global consistency of the map. More recently (and motivated by this paper), thin junction trees have been applied to the SLAM problem by Paskin [24]. This work establishes a viable alternative to the approach proposed here, with somewhat different computational properties.

This paper proposes a new SLAM algorithm whose updates require constant time, independent of the number of features in the map. Our approach is based on the well-known *information form* of the EKF, also known as the extended information filter (EIF) [22]. To achieve constant time updating, we develop an approximate EIF which maintains a *sparse* representation of environmental dependencies. Empirical simulation results provide evidence that the resulting maps are comparable in accuracy to the computationally much more cumbersome EKF solution, which is still at the core of most work in the field.

Our approach is best motivated by investigating the workings of the EKF. Figure 1 shows the result of EKF mapping in an environment with 50 landmarks. The left panel shows a moving robot, along with its Gaussian estimates 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 [6]. The checkerboard appearance of the correlation matrix illustrates this fact. Maintaining these cross-correlations—of which there are quadratically many in the number

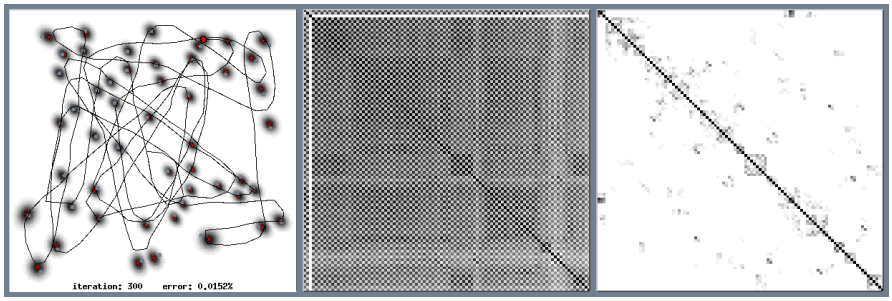

Figure 1. Typical snapshots of EKFs applied to the SLAM problem: Shown here is a map (left panel), a correlation (center panel), and a normalized information matrix (right panel). Notice that the normalized information matrix is naturally almost sparse, motivating our approach of using sparse information matrices in SLAM.

of features in the map—are essential to the SLAM problem. This observation has given rise to the (false) suspicion that online SLAM is inherently quadratic in the number of features in the map.

The key insight that motivates our approach is shown in the right panel of Figure 1. Shown there is the *inverse* covariance matrix (also known as *information matrix* [16,22]), normalized just like the correlation matrix. Elements in this normalized information matrix can be thought of as constraints, or links, between the locations of different features: 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 possesses a large number of links whose values, when normalized, are near zero. Furthermore, link strength is related to distance of features: Strong links are found only between geometrically nearby features. The more distant two landmarks, the weaker their link. This observation suggest that the EKF solution to SLAM possesses important structure that can be exploited for more efficient solutions. While any two features are fully correlated in the limit, the correlation arises mainly through a network of local links, which only connect nearby landmarks.

Our approach exploits this structure 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. Shown also is the robot, which is linked to a small subset of all features only, called active features and drawn in black. Storing a sparse information matrix requires linear space. More importantly, updates can be performed in constant time, regardless of the number of features in the map. The resulting filter is a *sparse extended information filter*, or SEIF. We show empirically that the SEIFs tightly approximate conventional extended information filters, which previously applied to SLAM problems in [20,22] and which are functionally equivalent to the popular EKF solution.

Our technique is probably most closely related to work on SLAM filters that represent relative distances, such as Newman's geometric projection filter [23] and extensions [5], and Csorba's relative filter [4]. It is also highly related to prior work

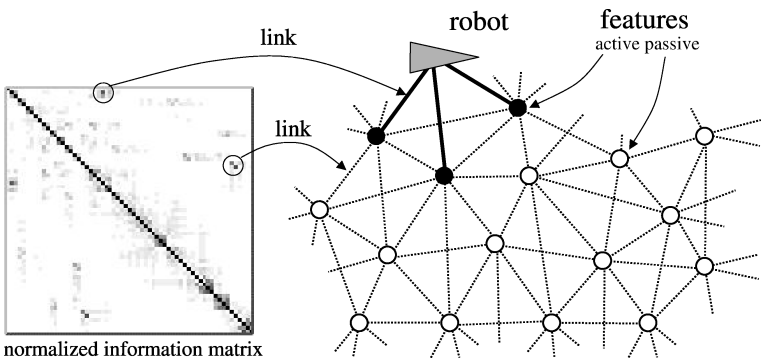

Figure 2. Illustration of the network of landmarks generated by our approach. Shown on the left is a sparse information matrix, and on the right a map in which entities are linked whose information matrix element is non-zero. As argued in the paper, the fact that not all landmarks are connected is a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

by Lu and Milion [**?**], who use poses as basic state variables in SLAM, between which they define local constraints obtained via scam matching. The locality of these constraints is similar to the local constraints in SEIFs, despite the fact that Lu and Milios do not formulate their filter in the information form. The problem of calculating posterior over paths is that both the computation and the memory grows with the path length, even in environments of limited size. It appears feasible to condense this information by subsuming multiple traversals of the same area into a single variable. We suspect that such a step would be aproximate, and that it would require similar approximations as proposed in this paper. At present, neither of these approaches permit constant time updating in SLAM, even though it appears that several of these techniques could be developed into constant time algorithms. Our work is also related to the rich body of literature on topological mapping [3,10,15,36], which typically does not explicitly represent dependencies and correlations in the representation of uncertainty. One can view SEIFs as a representation of local relative information between nearby landmarks; a feature shared by many topological approaches to mapping.

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 [20,22], but they are much less common than the EKF approach.

Most of the material in this section applies equally to linear and non-linear filters. We have chosen to present all material in the extended, non-linear form, since robots are inherently non-linear.

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 pose (or location) over time.

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

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

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

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

$$\begin{aligned} &= \exp\left\{-\tfrac{1}{2}\left[\xi_t^T \Sigma_t^{-1} \xi_t - 2\mu_t^T \Sigma_t^{-1} \xi_t + \mu_t^T \Sigma_t^{-1} \mu_t\right]\right\} \\ &= \exp\left\{-\tfrac{1}{2}\xi_t^T \Sigma_t^{-1} \xi_t + \mu_t^T \Sigma_t^{-1} \xi_t - \tfrac{1}{2}\mu_t^T \Sigma_t^{-1} \mu_t\right\} \end{aligned} \tag{2}$$

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 ξ_t and hence can be subsumed into the constant normalizer. This gives us the form:

$$\propto \exp\{-\tfrac{1}{2}\xi_t^T \underbrace{\Sigma_t^{-1}}_{=:H_t} \xi_t + \underbrace{\mu_t^T \Sigma_t^{-1}}_{=:b_t} \xi_t\} \tag{3}$$

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

$$H_t = \Sigma_t^{-1} \qquad \text{and} \qquad b_t \;=\; \mu_t^T H_t \tag{4}$$

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

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

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

$$\Sigma_t = H_t^{-1} \qquad \text{and} \qquad \mu_t \;=\; H_t^{-1} b_t^T \;=\; \Sigma_t b_t^T \tag{6}$$

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:

$$H_t = \begin{pmatrix} H_{x_t,x_t} & H_{x_t,y_1} & \cdots & H_{x_t,y_N} \\ H_{y_1,x_t} & H_{y_1,y_1} & \cdots & H_{y_1,y_N} \\ \vdots & \vdots & \ddots & \vdots \\ H_{y_N,x_t} & H_{y_N,y_1} & \cdots & H_{y_N,y_N} \end{pmatrix} \tag{7}$$

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.

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 landmark. 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 $H_{x_t,y}$ and their symmetric counterparts H_{y,x_t} that link together x_t and y. Additionally, the on-diagonal elements H_{x_t,x_t} and H_{y_1,y_1} are also updated. These updates are additive: Each observation of a feature y increases the strength of the

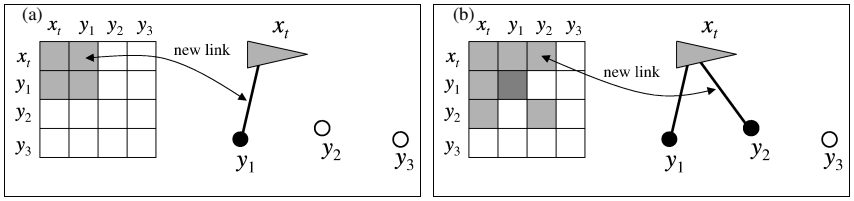

Figure 3. The effect of measurements on the information matrix and the associated network of features: (a) Observing y_1 results in a modification of the information matrix elements H_{x_t,y_1}. (b) Similarly, observing y_2 affects H_{x_t,y_2}. Both updates can be carried out in constant time.

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 $H_{x_t,y_2} = H^T_{y_2,x_t}$ (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 landmarks, or between the robot and unobserved landmarks.

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 [32]. 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 non-linear measurement function h with added Gaussian noise:

$$z_t = h(\xi_t) + \varepsilon_t \tag{9}$$

Here ε_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

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

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

$$h(\xi_t) \approx h(\mu_t) + \nabla_\xi h(\mu_t)[\xi_t - \mu_t] \tag{11}$$

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

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

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

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

Multiplying out the exponent and regrouping the resulting terms gives us

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

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

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

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

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

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

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

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

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

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

This sparseness 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

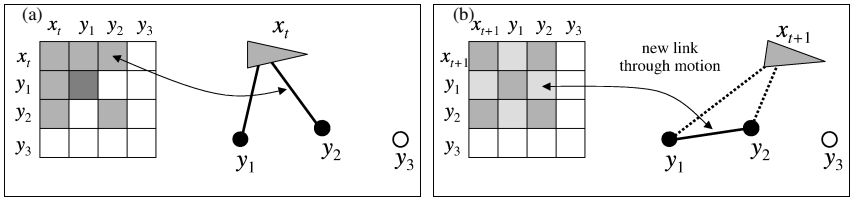

Figure 4. The effect of motion on the information matrix and the associated network of features: (a) before motion, and (b) after motion. If motion is non-deterministic, motion updates introduce new links (or reinforce existing links) between any two active features, while weakening the links between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

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) landmarks. 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-landmark 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 motion involves the calculation of the following posterior:

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

Exploiting the common SLAM independences [32] leads to

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

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

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

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

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

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

Here $A_t = \nabla_\xi g(\mu_{t-1}, u_t)$ is the derivative of g with respect to ξ at $\xi = \mu_{t-1}$ and u_t. The symbol $\hat{\Delta}_t$ is short for the predicted motion effect, $g(\mu_{t-1}, u_t)$. Plugging this approximation into (22) leads to an approximation of ξ_t, the state at time t:

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

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

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

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

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

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

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

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

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

3 Sparse Extended Information Filters

The central, new algorithm presented in this paper is the *Sparse Extended Information Filter*, or *SEIF*. SEIF differ from the extended information filter described in the previous section in that is maintains a *sparse* information matrix. An information matrix H_t is considered *sparse* if the number of links to the robot and to each feature in the map is bounded by a constant that is independent of the number of features in the map. The bound for the number of links between the robot pose and other features in the map will be denoted θ_x; the bound on the number of links for each feature (not counting the link to the robot) will be denoted θ_y. The motivation for maintaining a sparse information matrix was already given above: 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:*

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

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 non-linear mechanics where there is no 'cross-talk' between absolute coordinates and the additive change due to motion.

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

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

$$\Psi_t = I - S_x(I + [S_x^T A_t S_x]^{-1})^{-1} S_x^T$$

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

For $A_t \neq 0$, a constant time update requires knowledge of the mean μ_{t-1} before the motion command, for the robot pose and all active landmarks (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 Amortized Approximated Map Recovery

Before deriving an algorithm for recovering the state estimate μ_t from the information form, let us briefly consider what parts of μ_t are needed in SEIFs, and when. SEIFs need the state estimate μ_t of the robot pose and the active features in the map. These estimates are needed at three different occasions: (1) the linearization of the non-linear 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 μ_t, namely the estimate of the robot pose and of the locations of active features.

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

$$
\mu_t = H_t^{-1} b_t^T \;=\; \Sigma_t b_t^T \tag{31}
$$

Unfortunately, calculating (31) directly involves inverting a large matrix, which would requires more than constant time.

The sparseness of the matrix H_t allows us to recover the state incrementally. In particular, we can do so on-line, as the data is being gathered and the estimates b and H are being constructed. To do so, it will prove convenient to pose (31) as an optimization problem:

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

$$
p(\nu_t) = \text{const.} \cdot \exp\left\{-\tfrac{1}{2} \nu_t^T H_t \nu_t + b_t^T \nu_t\right\} \tag{32}
$$

Here ν_t is a vector of the same form and dimensionality as μ_t. This lemma suggests that recovering μ_t is equivalent to finding the mode of (32). 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 (32) is attained at:

$$
\hat{\nu}_t = \operatorname*{argmax}_{\nu_t} \; p(\nu_t) \;=\; \operatorname*{argmax}_{\nu_t} \; \exp\left\{-\tfrac{1}{2} \nu_t^T H_t \nu_t + b_t^T \nu_t\right\}
$$

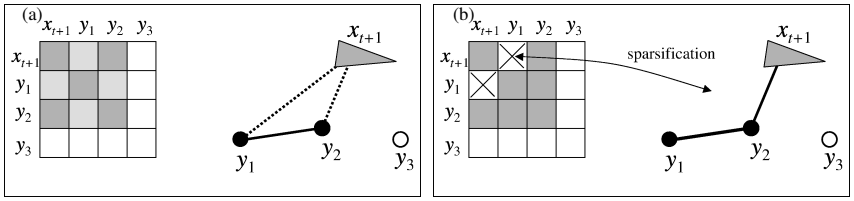

Figure 5. Sparsification: A feature is deactivated by eliminating its link to the robot. To compensate for this change in information state, links between active features and/or the robot are also updated. The entire operation can be performed in constant time.

$$= \underset{\nu_t}{\operatorname{argmin}} \ \tfrac{1}{2}\nu_t^T H_t \nu_t - b_t^T \nu_t \tag{33}$$

We note that the argument of the min-operator in (33) can be written in a form that makes the individual coordinate variables $\nu_{i,t}$ (for the i-th coordinate of ν_t) explicit:

$$\tfrac{1}{2}\nu_t^T H_t \nu_t - b_t^T \nu_t = \tfrac{1}{2} \sum_i \sum_j \nu_{i,t}^T H_{i,j,t} \nu_{j,t} - \sum_i b_{i,t}^T \nu_{i,t} \tag{34}$$

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 $\nu_{i,t}$ gives us

$$\frac{\partial}{\partial \nu_{i,t}} \left\{ \tfrac{1}{2} \sum_i \sum_j \nu_{i,t}^T H_{i,j,t} \nu_{j,t} - \sum_i b_{i,t}^T \nu_{i,t} \right\} = \sum_j H_{i,j,t} \nu_{j,t} - b_{i,t}^T \tag{35}$$

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

$$\nu_{i,t}^{[k+1]} = H_{i,i,t}^{-1} \left[b_{i,t}^T - \sum_{j \neq i} H_{i,j,t} \nu_{j,t}^{[k]} \right] \tag{36}$$

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

$$\nu_{i,t}^{[k+1]} = (S_i^T H_t S_i)^{-1} S_i^T \left[b_t - H_t \nu_t^{[k]} + H_t S_i S_i^T \nu_t^{[k]} \right] \tag{37}$$

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 (36), and hence the vector multiplication in (37), 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.

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

Our sparsification technique is illustrated in Figure 5. Shown there is the situation before and after sparsification. The removal of a link in the network corresponds to setting an element in the information matrix to zero; however, this requires the manipulation of other links between the robot and other active landmarks. 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 subsets:

$$Y = Y^+ \uplus Y^0 \uplus Y^- \tag{38}$$

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:

$$\begin{aligned} p(x_t, Y \mid z^t, u^t) &= p(x_t, Y^0, Y^+, Y^- \mid z^t, u^t) \\ &= p(x_t \mid Y^0, Y^+, Y^-, z^t, u^t)\; p(Y^0, Y^+, Y^- \mid z^t, u^t) \\ &= p(x_t \mid Y^0, Y^+, Y^- = 0, z^t, u^t)\; p(Y^0, Y^+, Y^- \mid z^t, u^t) \end{aligned} \tag{39}$$

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:

$$\begin{aligned} \tilde{p}(x_t, Y \mid z^t, u^t) &= p(x_t \mid Y^+, Y^- = 0, z^t, u^t)\; p(Y^0, Y^+, Y^- \mid z^t, u^t) \\ &= \frac{p(x_t, Y^+ \mid Y^- = 0, z^t, u^t)}{p(Y^+ \mid Y^- = 0, z^t, u^t)}\; p(Y^0, Y^+, Y^- \mid z^t, u^t) \end{aligned} \tag{40}$$

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^-:

$$H'_t = S_{x,Y^+,Y^0} S^T_{x,Y^+,Y^0} H_t S_{x,Y^+,Y^0} S^T_{x,Y^+,Y^0} \tag{41}$$

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

$$\begin{aligned} H^1_t &= H'_t - H'_t S_{Y_0} (S^T_{Y_0} H'_t S_{Y_0})^{-1} S^T_{Y_0} H'_t \\ H^2_t &= H'_t - H'_t S_{x,Y_0} (S^T_{x,Y_0} H'_t S_{x,Y_0})^{-1} S^T_{x,Y_0} H'_t \end{aligned} \tag{42}$$

Here the various S-matrices are projection matrices, analogous to the matrix S_x defined above. The final term in our approximation (40), $p(Y^0, Y^+, Y^- \mid z^t, u^t)$, has the following information matrix:

$$H^3_t = H_t - H_t S_{x_t} (S^T_{x_t} H_t S_{x_t})^{-1} S^T_{x_t} H_t \tag{43}$$

Putting these expressions together according to Equation (40) yields the following information matrix, in which the landmark Y^0 is now indeed deactivated:

$$\begin{aligned} \tilde{H}_t = H^1_t - H^2_t + H^3_t \;\; &= \;\; H_t - H'_t S_{Y_0} (S^T_{Y_0} H'_t S_{Y_0})^{-1} S^T_{Y_0} H'_t \\ &+ H'_t S_{x,Y_0} (S^T_{x,Y_0} H'_t S_{x,Y_0})^{-1} S^T_{x,Y_0} H'_t - H_t S_{x_t} (S^T_{x_t} H_t S_{x_t})^{-1} S^T_{x_t} H_t \end{aligned} \tag{44}$$

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

$$\begin{aligned} \tilde{b}_t = \mu^T_t \tilde{H}_t \;\; &= \;\; \mu^T_t (H_t - H_t + \tilde{H}_t) \\ = \mu^T_t H_t + \mu^T_t (\tilde{H}_t - H_t) \;\; &= \;\; b_t + \mu^T_t (\tilde{H}_t - H_t) \end{aligned} \tag{45}$$

All equations can be computed in constant time. The effect of this approximation is the deactivation of the features Y^0, while introducing only new links between active features. The sparsification rule requires knowledge of the mean vector μ_t for all active features, which is obtained via the approximation technique described in the previous section. From (45), it is obvious that the sparsification does not affect the mean μ_t, that is, $H^{-1}_t b^T_t = [\tilde{H}_t]^{-1} [\tilde{b}_t]^T$. Furthermore, our approximation minimizes the KL divergence to the correct posterior. These 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 θ_x and θ_y.

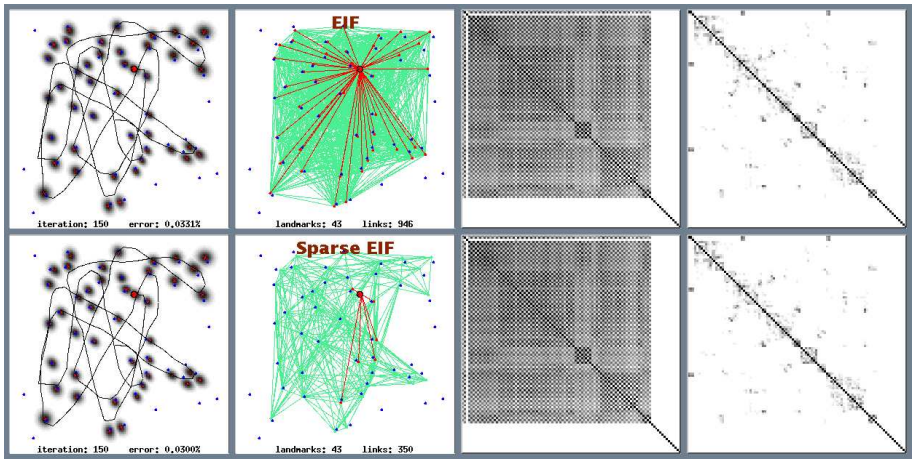

Figure 6. Comparison of EKFs with SEIFs using a simulation with $N = 50$ landmarks. In both diagrams, the left panels show the final filter result, which indicates higher certainties for our approach due to the approximations involved in maintaining a sparse information matrix. The center panels show 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!

4 Experimental Results

Our present experiments are preliminary: They only rely on simulated data, and they require known data associations. Our primary goal was to compare SEIFs to the computationally more cumbersome EKF solution that is currently in widespread use.

An example situation comparing EKFs with our new filter can be found in Figure 6. This result is typical and was obtained using a sparse information matrix with $\theta_x = 6$, $\theta_x = 10$, and a constant time implementation of coordinate descent that updates $K = 10$ random landmark estimates in addition to the landmark estimates connected to the robot at any given time. The key observation is the apparent similarity between the EKF and the SEIF result. Both estimates are almost indistinguishable, despite the fact that EKFs use quadratic update time whereas SEIF require only constant time.

We also performed systematic comparisons of three algorithms: EKFs, SEIFs, and a variant of SEIFs in which the exact state estimate μ_t is available. The latter was implemented using matrix inversion (hence does not run in constant time). It allowed us to tease apart the error introduced by the amortized mean recovery step, from the error induced through sparsification. The following table depicts results for $N = 50$ landmarks, after 500 update cycles, at which point all three approaches are near convergence.

	# experiments (so far)	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 μ_t	1,000	$(4.75 \pm 0.67) \cdot 10^{-3}$	549 ± 1.60	$O(N^3)$
SEIF (constant time)	1,00	$(6.35 \pm 0.67) \cdot 10^{-3}$	549 ± 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. Experimental results using a real-world data set can be found in [14].

5 Discussion

This paper proposed a constant time algorithm for the SLAM problem. Our approach adopted the information form of the EKF to represent all estimates. Based on the empirical observation that in the information form, most elements in the normalized information matrix are near-zero, we 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, all updates can be performed in constant time; in the *non-linear case*, additional state estimates are needed that are not part of the regular information form of the EKF. We proposed a amortized constant-time coordinate descent algorithm for recovering these state estimates from the information form.

The approach has been fully implemented and compared to the EKF solution. Overall, we found that SEIFs produce results that differ only marginally from that of the EKFs. 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.

Our approach puts a new perspective on the rich literature on hierarchical mapping, briefly outlined in the introduction to this paper. Like SEIF, 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 [21].

Acknowledgment

The authors would like to acknowledge invaluable contributions by the following researchers: Wolfram Burgard, Geoffrey Gordon, 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 fur-

thermore acknowledge support provided by the National Science Foundation (CAREER grant number IIS-9876136 and regular grant number IIS-9877033).

References

1. M. Bosse, J. Leonard, and S. Teller. Large-scale CML using a network of multiple local maps. In [11].
2. W. Burgard, D. Fox, H. Jans, C. Matenar, and S. Thrun. Sonar-based mapping of large-scale mobile robot environments using EM. *Proc. OCML-99*.
3. H. Choset. *Sensor Based Motion Planning: The Hierarchical Generalized Voronoi Graph*. PhD thesis, Caltech, 1996.
4. M. Csorba. *Simultaneous Localization and Map Building*. PhD thesis, Univ. of Oxford, 1997.
5. M. Deans and M. Hebert. Invariant filtering for simultaneous localization and mapping. *Proc. ICRA-00*.
6. G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localisation and map building (SLAM) problem. *Transactions of Robotics and Automation*, 2001.
7. A. Doucet, J.F.G. de Freitas, and N.J. Gordon, editors. *Sequential Monte Carlo Methods In Practice*. Springer, 2001.
8. J. Guivant and E. Nebot. Optimization of the simultaneous localization and map building algorithm for real time implementation. *Transactions of Robotics and Automation*, 2001.
9. J.-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. *Proc. ICRA-00*.
10. 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, 1991.
11. J. Leonard, J.D. Tardós, S. Thrun, and H. Choset, editors. *ICRA Workshop Notes (W4)*, 2002.
12. J. J. Leonard and H. F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation*. Kluwer, 1992.
13. J.J. Leonard and H.J.S. Feder. A computationally efficient method for large-scale concurrent mapping and localization.
14. Y. Liu and S. Thrun. Results for outdoor-SLAM using sparse extended information filters. Submitted to ICRA-03.
15. M. J. Matarić. A distributed model for mobile robot environment-learning and navigation. MIT AITR-1228.
16. P. Maybeck. *Stochastic Models, Estimation, and Control, Volume 1*. Academic Press, 1979.
17. M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. *Proc. AAAI-02*.
18. P. Moutarlier and R. Chatila. An experimental system for incremental environment modeling by an autonomous mobile robot. *Proc. ISER-89*.
19. K. Murphy. Bayesian map learning in dynamic environments. *Proc. NIPS-99*.
20. E. Nettleton, H. Durrant-Whyte, P. Gibbens, and A. Goktoğan. Multiple platform localisation and map building. *Sensor Fusion and Decentralised Control in Robotic Stystems III*: 4196, 2000.
21. E. Nettleton, S. Thrun, and H. Durrant-Whyte. A constant time communications algorithm for decentralised SLAM. Submitted.
22. 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. *Sensor Fusion: Architectures, Algorithms, and Applications IV*: 4051, 2000.

23. P. Newman. *On the Structure and Solution of the Simultaneous Localisation and Map Building Problem*. PhD thesis, University of Sydney, 2000.
24. M.A. Paskin. Thin junction tree filters for simultaneous localization and mapping. TR UCB/CSD-02-1198, University of California, Berkeley, 2002.
25. H Shatkay and L. Kaelbling. Learning topological maps with weak local odometric information. *Proc. IJCAI-97*.
26. R. Simmons, D. Apfelbaum, W. Burgard, M. Fox, D. an Moors, S. Thrun, and H. Younes. Coordination for multi-robot exploration and mapping. *Proc. AAAI-00*.
27. R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. *Autonomous Robot Vehicles*, Springer, 1990.
28. R. C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. TR 4760 & 7239, SRI, 1985.
29. 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, 2002.
30. C. Thorpe and H. Durrant-Whyte. Field robots. *Proc. ISRR-01*.
31. S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte, and A.Y. Ng. Simultaneous mapping and localization with sparse extended information filters:theory and initial results. TR CMU-CS-02-112, CMU, 2002.
32. S. Thrun. Robotic mapping: A survey. *Exploring Artificial Intelligence in the New Millenium*, Morgan Kaufmann, 2002.
33. S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31, 1998.
34. S. Williams and G. Dissanayake. Efficient simultaneous localisation and mapping using local submaps. In [11].
35. S.B. Williams, G. Dissanayake, and H. Durrant-Whyte. An efficient approach to the simultaneous localisation and mapping problem. *Proc. ICRA-02*.
36. U.R. Zimmer. Robust world-modeling and navigation in a real world. *Neurocomputing*, 13, 1996.

Appendix: Proofs

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

$$H_t = \bar{H}_t + C_t Z^{-1} C_t^T \tag{46}$$

$$b_t = \bar{b_t} + (z_t - \hat{z}_t + C_t^T \mu_t)^T Z^{-1} C_t^T \tag{47}$$

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:

$$\bar{H}_t = [H_{t-1}^{-1} + S_x U_t S_x^T]^{-1} \tag{48}$$

Applying the *matrix inversion lemma* [1] leads to the following form:

$$\begin{aligned}\bar{H}_t &= H_{t-1} - H_{t-1} \underbrace{S_x[U_t^{-1} + S_x^T H_{t-1} S_x]^{-1} S_x^T H_{t-1}}_{=:L_t} \\ &= H_{t-1} - H_{t-1} L_t \end{aligned} \tag{50}$$

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 (50):

$$\begin{aligned}\bar{b}_t &= [b_{t-1}H_{t-1}^{-1} + \hat{\Delta}_t^T]\bar{H}_t \\ &= [b_{t-1}H_{t-1}^{-1} + \hat{\Delta}_t^T](H_{t-1} - H_{t-1}L_t) \\ &= b_{t-1} + \hat{\Delta}_t^T H_{t-1} - b_{t-1}L_t + \hat{\Delta}_t^T H_{t-1} L_t \end{aligned} \tag{51}$$

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:

$$\begin{aligned}\Psi_t &= (I + S_x S_x^T A_t S_x S_x^T)^{-1} \\ &= I - I S_x (S_x I S_x^T + [S_x^T A_t S_x]^{-1})^{-1} S_x^T I \\ &= I - S_x (I + [S_x^T A_t S_x]^{-1})^{-1} S_x^T \end{aligned} \tag{52}$$

Notice that Ψ_t differs from the identity matrix I only at elements that correspond to the robot pose, as is easily seen from the fact that the inversion in (52) involves a low-dimensional matrix.

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

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

[1] The inversion lemma, as used throughout this paper, is stated as follows:

$$\left(H^{-1} + SBS^T\right)^{-1} = H - HS\left(B^{-1} + S^T H S\right)^{-1} S^T H \tag{49}$$

$$
\begin{aligned}
&= [(H'_{t-1})^{-1} + S_x U_t S_x^T]^{-1} \\
&= H'_{t-1} - \underbrace{H'_{t-1} S_x [U_t^{-1} + S_x^T H'_{t-1} S_x]^{-1} S_x^T H'_{t-1}}_{=:\Delta H_t} \\
&= H'_{t-1} - \Delta H_t
\end{aligned} \tag{53}
$$

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:

$$
\begin{aligned}
\bar{b}_t &= [b_{t-1} H_{t-1}^{-1} + \hat{\Delta}_t^T] \bar{H}_t \\
&= b_{t-1} H_{t-1}^{-1} \bar{H}_t + \hat{\Delta}_t^T \bar{H}_t \\
&= b_{t-1} H_{t-1}^{-1} (\bar{H}_t + \underbrace{H_{t-1} - H_{t-1}}_{=0} + \underbrace{H'_{t-1} - H'_{t-1}}_{=0}) + \hat{\Delta}_t^T \bar{H}_t \\
&= b_{t-1} H_{t-1}^{-1} (H_{t-1} + \underbrace{\bar{H}_t - H'_{t-1}}_{-\Delta H_t} - H_{t-1} + H'_{t-1}) + \hat{\Delta}_t^T \bar{H}_t \\
&= b_{t-1} H_{t-1}^{-1} (H_{t-1} - \Delta H_t - H_{t-1} + H'_{t-1}) + \hat{\Delta}_t^T \bar{H}_t \\
&= b_{t-1} - b_{t-1} H_{t-1}^{-1} (\Delta H_t - H_{t-1} + H'_{t-1}) + \hat{\Delta}_t^T \bar{H}_t \\
&= b_{t-1} - \mu_{t-1}^T H_{t-1} H_{t-1}^{-1} (\Delta H_t - H_{t-1} + H'_{t-1}) + \hat{\Delta}_t^T \bar{H}_t \\
&= b_{t-1} - \mu_{t-1}^T (\Delta H_t - H_{t-1} + H'_{t-1}) + \hat{\Delta}_t^T \bar{H}_t
\end{aligned} \tag{54}
$$

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

Proof of Lemma 4: The mode $\hat{\nu}_t$ of (32) is given by

$$
\begin{aligned}
\hat{\nu}_t &= \operatorname*{argmax}_{\nu_t} \; p(\nu_t) \\
&= \operatorname*{argmax}_{\nu_t} \; \exp\left\{-\tfrac{1}{2} \nu_t^T H_t \nu_t + b_t^T \nu_t\right\} \\
&= \operatorname*{argmin}_{\nu_t} \; \tfrac{1}{2} \nu_t^T H_t \nu_t - b_t^T \nu_t
\end{aligned} \tag{55}
$$

The gradient of the expression inside the minimum in (55) with respect to ν_t is given by

$$
\frac{\partial}{\partial \nu_t} \left\{\tfrac{1}{2} \nu_t^T H_t \nu_t - b_t^T \nu_t\right\} = H_t \nu_t - b_t^T \tag{56}
$$

whose minimum $\hat{\nu}_t$ is attained when the derivative (56) is 0, that is,

$$
\hat{\nu}_t = H_t^{-1} b_t^T \tag{57}
$$

From this and Equation (31) it follows that $\hat{\nu}_t = \mu_t$. $Q.E.D.$