Results for Outdoor-SLAM Using Sparse Extended Information Filters

Yufeng Liu and Sebastian Thrun

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
yufeng@cs.cmu.edu , thrun@cs.cmu.edu

Abstract

In [13], a new algorithm was proposed for efficiently solving the simultaneous localization and mapping (SLAM) problem. In this paper, we extend this algorithm to handle data association problems and report real-world results, obtained with an outdoor vehicle. We find that our approach performs favorably when compared to the extended Kalman filter solution from which it is derived.

1 Introduction

This paper investigates a scalable algorithm for the simultaneous mapping and localization (SLAM) problem, and evaluates it in the context of outdoor navigation. The 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 [2, 5]. 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 errors over time, making the problem of acquiring an accurate map challenging. The SLAM problem has attracted immense attention in the past a few years [4].

This paper addresses computational issues in performing real-world SLAM. The classical SLAM solution, based on the extended Kalman filter (EKF) [8, 9, 12, 11], scales quadratically with the number of landmarks in the map. As a result, practical implementations of this approach are limited to a few hundred landmarks [2]. This deficiency has long been recognized and has spurred a flurry of research on more efficient SLAM algorithms. One thrust of research involves the development of hierarchical techniques, which decompose large maps into smaller, computationally more manageable sub-maps [1, 3, 6, 15]. Such techniques are more efficient than the quadratic time EKF, but most of them still require quadratic time for maintaining global consistency between multiple sub-maps. Additionally, the consistency is difficult to maintain as the vehicle crosses boundaries between different sub-maps.

In this paper, we follow a different approach. In a recent workshop paper [13], we proposed a SLAM algorithm that requires constant time for updating, yet still maintains global consistency. This approach is based on the *information form* of the Kalman filter [7, 10], known as *extended information filter* (EIF). EIFs are mathematically identical to EKF, yet they represent map estimates by sets of pairwise constraints between landmarks. In practice, these constraints are usually sparse. This insight led us to define the *sparse extended information filter*, or SEIF. SEIFs can be updated in constant time, which is significantly faster than the quadratic update time of EKFs. Meanwhile it maintains a globally consistent estimate of the robot pose and the map. However, the original paper [13] only provided theoretical results and did not analyze the performance of SEIFs using real-world data. It also did not provide a method for handling data association problems that commonly occur in real-world settings.

This paper describes SEIFs, our extension to handle data association problems, and empirical results. Because SEIFs are approximations of EKFs, an important question is the accuracy of this approximation. This paper presents first experimental results that compare SEIFs with EKFs using both simulated and real-world data sets. We find that empirically, SEIFs are highly accurate approximations to EKFs. Our empirical comparison utilizes a benchmark data set recorded with an outdoor vehicle [2]. On the computational end, we find that SEIFs are significantly more efficient as is predicted, and their efficiency makes them scalable to much larger maps than EKFs can handle.

2 Sparse Extended Information Filters

2.1 Preliminaries

Let us begin the disposition by highlighting the intuition behind our approach. The standard approach for solving feature-based SLAM problems is based on the extended Kalman filter (EKF) [11, 2]. Figure 1 shows the result of EKF mapping in an environment with 50 landmarks. The normalized covariance of the EKF is the correlation matrix, which is visualized in Figure 1a. 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 [2]. The checkerboard appearance of the correlation matrix illustrates this fact.

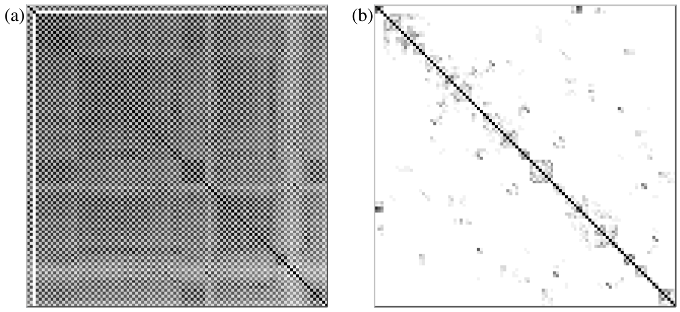


Figure 1: Typical snapshot of EKFs applied to the SLAM problem: (a) a correlation matrix (normalized covariance) and (b) the normalized inverse covariance, or information matrix. This plot illustrates the basic insight of SEIFs: Correlation matrices are dense, whereas their normalized inverses are naturally sparse.

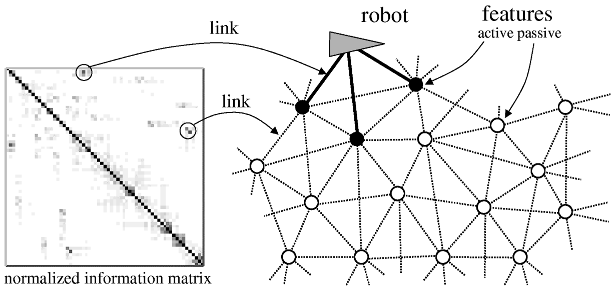


Figure 2: Illustration of the network of landmarks generated by our approach. Shown on the left is a sparse information matrix, and on the right a map in which entities are linked when the corresponding elements in the information matrix are non-zero. As argued in the paper, the fact that not all landmarks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

The key insight that motivates our approach is shown in Figure 1b. Shown there is the *inverse* covariance matrix (also known as *information matrix* [7, 10]), 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 landmarks: The darker an entry is in the display, the stronger the link is. As this depiction suggests, the normalized information matrix appears to be naturally sparse: It is dominated by a small number of strong links between geographically nearby landmarks, and possesses a large number of links whose values, when normalized, are near zero. Furthermore, link strengths are related to distances between landmarks: Strong links are found only between geometrically nearby landmarks. The more distant two landmarks are from each other, the weaker their link is.

SEIFs exploit this structure by maintaining a *sparse* information matrix, in which only nearby landmarks are linked through a non-zero element. The resulting network structure is illustrated in the right panel of Figure 2, where disks correspond to landmarks 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 landmarks only. This subset of landmarks are called active landmarks 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 landmarks in the map. The resulting filter is a *sparse extended information filter*, or SEIF.

2.2 Information State

Let x_t denote the pose of the robot at time t, and y_n with $1 \leq n \leq N$ the location of the n-th landmark, with N being the total number of landmarks in the environment (a quantity that is estimated during mapping). The robot pose x_t and the set of all landmark locations Y together constitute the *state* of the environment:

$$\xi_t = \begin{pmatrix} x_t & y_1 & \dots & y_N \end{pmatrix}^T \tag{1}$$

As is common in SLAM literature, SEIFs present the posterior by a multi-variate Gaussian over the state ξ_t. Such a Gaussian can be represented by a mean μ_t and a covariance Σ_t, or equally by the so-called *natural parameters* of the Kalman filter:

$$b_t = \mu_t^T \Sigma_t^{-1} \tag{2}$$
$$H_t = \Sigma_t^{-1} \tag{3}$$

The EKF representation using the information vector b_t and the information matrix H_t is known as the *information form* of the EKF, or *extended information filter* (EIF). The mean μ_t and covariance Σ_t are easily recovered from the information form:

$$\Sigma_t = H_t^{-1} \tag{4}$$
$$\mu_t = \Sigma_t b_t^T \tag{5}$$

The information matrix H_t was already discussed above, and an example was shown in Figure 1b. Sparse EIFs, or SEIFs, are EIFs whose information matrix H_t is sparse. Put differently, each row and each column in H_t contains only a limited number of non-zero elements, and the limit does not depend on the size of the matrix N. Sparseness is achieved by an update rule that occasionally removes links from the posterior so as to maintain sparseness, as described further below.

2.3 Measurement Updates

One of the key update steps in SLAM involves the incorporation of a measurement (a landmark sighting). The measurement at time t is denoted z_t. In [13], it is assumed the index of this landmark can be sensed without error—a classical assumption known in SLAM as "known data association," necessary for maintaining Gaussian estimates. For now, let us adopt this assumption; further below, we will discuss our approach for estimating the landmark identity during the estimation process.

Figure 3 illustrates the effect of measurements on the information matrix H_t. Suppose the robot senses landmark y_1,

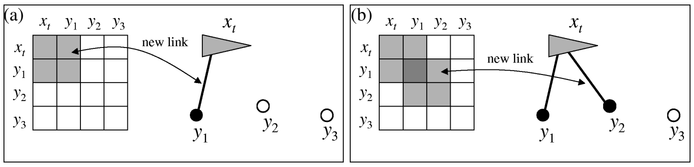


Figure 3: The effect of measurements on the information matrix and the associated network of landmarks: (a) Observing y_1 results in a modification of the information matrix element H_{x_t,y_1}. (b) Similarly, observing y_2 affects H_{x_t,y_2}. Both updates can be carried out in constant time.

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 element H_{x_t,y_1} and its symmetric counterpart H_{y_1,x_t} that link together x_t and y_1. Additionally, the diagonal elements H_{x_t,x_t} and H_{y_1,y_1} are also updated. These updates are additive: Each observation of a landmark y increases the strength of the total link between the robot pose and this very landmark, thus the total information in the filter. Figure 3b shows the incorporation of a second measurement of a different landmark, y_2. In response to this measurement, the EIF updates the links H_{x_t,y_2}, H_{y_2,x_t}, 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 landmarks. Measurements never generate links between pairs of landmarks, or between the robot and unobserved landmarks.

Incorporating measurements into the information filters naturally requires constant time. The canonical update rule is the following:

$$
\begin{aligned}
H_t &= \bar{H}_t + C_t Z^{-1} C_t^T && (6)\\
b_t &= \bar{b_t} + (z_t - h(\mu_t) + C_t^T \mu_t)^T Z^{-1} C_t^T && (7)
\end{aligned}
$$

Here h is the measurement function that maps state ξ_t into measurement z_t. The measurement noise is Gaussian with covariance Z. Finally, the matrix C_t is the gradient of the measurement function h with respect to the state vector ξ, taken at $\xi = \mu_t$:

$$
C_t = \nabla_\xi h(\mu_t) \qquad (8)
$$

In general filter applications, such an update may require more than constant time. In SLAM, however, each measurement involves only a single landmark (or a limited number of landmarks, regardless of N). For this reason, C_t is zero except for a limited number of elements. With such a sparse matrix C_t, constant time updates can be implemented.

2.4 Motion Updates

Figure 4a illustrates an information matrix and the associated network before the robot moves. 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

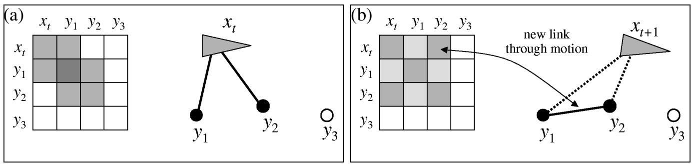


Figure 4: The effect of motion on the information matrix and the associated network of landmarks: (a) before motion, and (b) after motion. If motion is non-deterministic, motion updates introduce new links (or reinforce existing links) between any two active landmarks, while weakening the links between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

motion. However, the noise in robot actuation weakens the link between the robot and all active landmarks. Hence H_{x_t,y_1} and H_{x_t,y_2} are decreased by certain amounts. This decrease reflects the fact that the noise in robot motion causes a loss of information about the relative positions of the landmarks with respect to the robot. Not all of this information is lost, however. Some of it is shifted into between-landmark link H_{y_1,y_2}, as illustrated in Figure 4b. This reflects the fact that even though the motion induces a loss of information of the robot relative to the landmarks, no information is lost between individual landmarks. Robot motion, thus, has the effect that landmarks that were indirectly linked through the robot pose become linked directly.

Motion updates in the information form of the Kalman filter are usually *not* achievable in constant time. However, as proven in [13], the update can be performed in constant time if the information matrix H_t is sparse. The equations for the general case are as follows:

$$
\begin{aligned}
\Psi_t &= I - S_x (I + [S_x^T A_t S_x]^{-1})^{-1} S_x^T\\
H'_{t-1} &= \Psi_t^T H_{t-1} \Psi_t\\
\Delta H_t &= H'_{t-1} S_x [U_t^{-1} + S_x^T H'_{t-1} S_x]^{-1} S_x^T H'_{t-1}\\
\bar{H}_t &= H'_{t-1} - \Delta H_t\\
\bar{b}_t &= b_{t-1} - \mu_{t-1}^T (\Delta H_t + H'_{t-1} - H_{t-1}) + \hat{\Delta}_t^T \bar{H}_t
\end{aligned}
\qquad (9)
$$

Here S_x is the canonical projection matrix from the full state to the robot pose coordinates. These equations are mathematically exact (not just approximations). They may appear quite complicated, but they sure are constant time update rules. However, they may cause violations to the sparseness constraints by adding new links (non-zero elements) in the information matrix H_t. The removal of some links is a key approximation step in SEIFs, which enables them to maintain sparse information matrices.

2.5 Sparsification

SEIF's sparsification technique is illustrated in Figure 5. Shown there are the situations 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

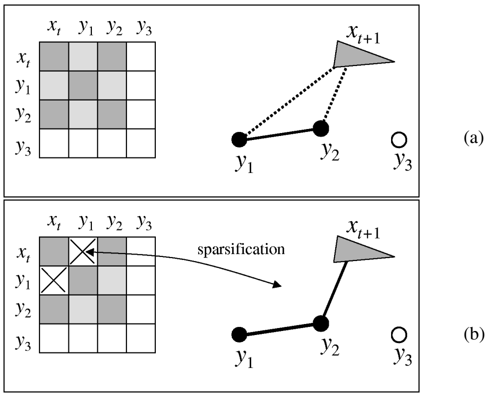


Figure 5: Sparsification: A landmark is deactivated by eliminating its link to the robot. To compensate for this change in information state, links between active landmarks and/or the robot are also updated. The entire operation can be performed in constant time.

requires the manipulation of some links between the robot and other active landmarks. The resulting network is only an approximation to the original one, and its quality depends on the strength of the link before its removal.

$$\begin{aligned}
H'_t &= S_{x,Y^+,Y^0} S^T_{x,Y^+,Y^0} H_t S_{x,Y^+,Y^0} S^T_{x,Y^+,Y^0} \\
b'_t &= b_t S_{x,Y^+,Y^0} S^T_{x,Y^+,Y^0} \\
L'_t &= [S_x (S_x^T H_t S_x)^{-1} S_x^T + S_{Y^0} (S^T_{Y^0} H_t S_{Y^0})^{-1} S^T_{Y^0} \\
&\quad - S_{x,Y^0} (S^T_{x,Y^0} H_t S_{x,Y^0})^{-1} S^T_{x,Y^0}] H'_t \\
\tilde{H}_t &= H_t - H'_t L'_t \\
\tilde{b}_t &= b_t - b'_t L'_t S_{Y^0} S^T_{Y^0} + (\mu_t^T \tilde{H}_t - b_t) S_{x,Y^+} S^T_{x,Y^+}
\end{aligned} \tag{10}$$

To implement these equations, it is necessary to subdivide the set of landmarks into three subsets: the set of active landmarks Y^+ that contain a non-zero link to the robot pose in the information matrix H_t; the set of passive landmarks Y^-; and finally the set of landmarks Y^0 that are active before the sparsification step, but passive afterwards. Our sparsification equations have the effect of removing links between Y^0 and the robot pose — a step necessary if the number of landmarks linked to the robot exceeds a given sparsity threshold. By doing so, the number of between-landmark links also remains limited. Consequently, the sparsification step ensures the sparseness of the information matrix H_t in SEIFs. We note that this step is approximate. The question as to whether this approximation affects SEIF's performance in practice has not been addressed previously.

2.6 Amortized Map Recovery

Finally, SEIFs offer an algorithm that also requires constant time for recovering the means μ_t from the information form. The recovery of the means might be interesting if one would like to visualize the map: The information form contains the map only implicitly, and the obvious recovery via Equations (4) and (5) would require cubic time. More importantly, the means of the robot pose and active landmarks' locations are required in several of the above update steps, (8), (9) and (10). SEIFs use an amortized iterative method, similar to the Jacobi method or the slightly different Gauss-Seidel method, to gradually recover μ_t. One sufficient condition for this kind of method to converge is the positive definite condition which the H_t matrix satisfies. The fact that H_t is diagonally dominant also makes these methods converge fast. To describe the algorithm in detail, let us write H_t in four blocks $H_t = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$ and accordingly have $\mu_t = (\mu_A\ \mu_C), b_t = (b_A\ b_C)$. Block A should include the components of H_t that correspond to robot pose, active landmarks' positions and possibly a constant number of other elements, such as the locations of those landmarks linked to active landmarks. To recover μ_t in one step, one needs to solve two equations

$$\begin{aligned}
A\mu_A + B\mu_C &= b_A \\
B^T\mu_A + C\mu_C &= b_C
\end{aligned} \tag{11}$$

Following the idea of iterative methods mentioned above, only part of μ_t, i.e. μ_A is updated by $\mu_A = A^{-1}(b_A - B\mu_C)$ in one step. Since matrices A and B have only a limited number of nonzero elements, an update shall be carried out in constant time. Iterations are performed whenever components of μ_t are inquired. If some components of μ_A are changed by significant amounts, for example, when a loop is being closed, extra steps may be taken to update those components together with landmarks linked directly with them. Further, such extra steps can be buffered to work out when the computer is idling, thus the computing power is better utilized. As the robot explores the environment, active landmarks change, so all the components of μ_t get chances to be updated. This amortized map recovery introduces additional error to the system. However experiments show that the error is insignificant when compared to the advantages of SEIF in other aspects.

3 Data Association

3.1 Recovering Data Association Probabilities

Finally, practical domains are characterized by *data association problems*. Data association problems arise when individual landmarks in the environment cannot be identified uniquely based on sensor measurements alone. The data association problem is pervasive in real-world SLAM problems. However, the original publication [13] did not address this important problem.

Our mechanism for handling the data association problem uses a maximum likelihood estimator, together with a thresholded χ^2 test. In particular, our approach selects the landmark that best explains a measurement. If we write n_t as the landmark index of the landmark seen at time t, the maximum likelihood estimator determines

$$n_t^* = \underset{n}{\operatorname{argmax}}\, p(n_t | z^t, u^t)$$

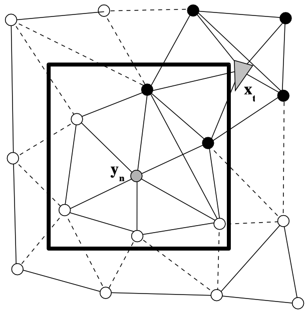


Figure 6: The combined Markov blanket of landmark y_n and robot x_t is sufficient for calculating the posterior probability of the landmark locations, conditioning away all other landmarks. This insight leads to a constant time method for recovering the approximate probability distribution $p(x_t, y_n|z^{t-1}, u^t)$.

$$= \quad \underset{n}{\text{argmax}}\, p(n_t|z_t)\, \underbrace{p(\xi|z^{t-1}, u^t)}_{\bar{H}_t, \bar{b}_t} \qquad (12)$$

where $z^t = z_1, ..., z_t$ and $u^t = u_1, ..., u_t$. If the expression inside the argmax is smaller than a threshold α, that is, none of the landmarks in the map explains the measurement with a minimum required probability, the landmark is considered new and the filter is grown accordingly. By using this test, the resulting SEIF gradually builds up a network of landmarks, while nearby landmarks are connected by links. This approach is commonly used in the context of EKFs [2]. In EKFs, calculating $p(n_t|z^t, u^t)$ is easy since it is straightforward to extract the mean and the covariance of a landmark position together with the robot pose from the full state estimate. The mean and the covariance define a probability density $p(x_t, y_n|z^{t-1}, u^t)$ which are then used to calculate the probability $p(n_t|z^t, u^t)$.

In SEIFs, the situation is more complicated: Recovering the covariance of a landmark location and the robot pose in the naive way would require inverting a large matrix, which is a $O(N^3)$ operation. However, we can once again exploit the sparseness of the information matrix to obtain a high fidelity approximation of the necessary covariances.

Suppose we would like to calculate the probability distribution of the n-th landmark y_n and the robot pose x_t. The idea is to do so by conditioning on all state variables outside the Markov blankets of these variables. The Markov blanket of the robot pose x_t is simply the set of all active landmarks. Likewise, the Markov blanket of landmark y_n is the set of all landmarks (and possibly the robot pose) directly connected to this landmark in the SEIF. Figure 6 illustrates the situation. Shown there is a landmark, a robot pose, the Markov blanket of the landmark (inside the square box) and the set of all active landmarks (in black). All other variables are not considered during this operation, since they do not assert a *direct* influence on the robot pose or landmark in question.

Mathematically, we do the following approximation, where Y_n^+ is the combined Markov blanket:

$$\begin{aligned} & p(x_t, y_n|z^{t-1}, u^t) \\ &\quad = \int p(x_t, y_n|Y_n^+, z^{t-1}, u^t)\; p(Y_n^+|z^{t-1}, u^t) dY_n^+ \\ &\quad \approx \int p(x_t, y_n|Y_n^+)\; p(Y_n^+|Y_n^-) dY_n^+ \qquad (13) \end{aligned}$$

Here

$$Y_n^- \quad = \quad Y \backslash \{x_t, y_n\} \backslash Y_n^+ \qquad (14)$$

is the set of all state variables not included in the Markov blanket Y_n^+, and also excluding y_n and x_t. This approximation ignores a residual uncertainty in remote state variables. However, we found that empirically it approximates the true posterior probability needed for data association with double-digit accuracy in most cases.

Apart from the mathematical reasoning, the operation in matrix form is simple. The distribution $p(x_t, y_n|z^{t-1}, u^t)$ is approximated by a Gaussian with covariance

$$\Sigma_{t:n} \quad = \quad (S^T_{x_t, y_n, Y_n^+} H_t S_{x_t, y_n, Y_n^+})^{-1} \qquad (15)$$

This calculation is constant time, since it involves a matrix whose size is independent of N.

In our experiment, we found this approximation to work surprisingly well. In the results reported further below using real-world data, the average relative error in estimating likelihoods is 3.4×10^{-4}. Association errors due to this approximation were practically non-existent.

3.2 Map Management

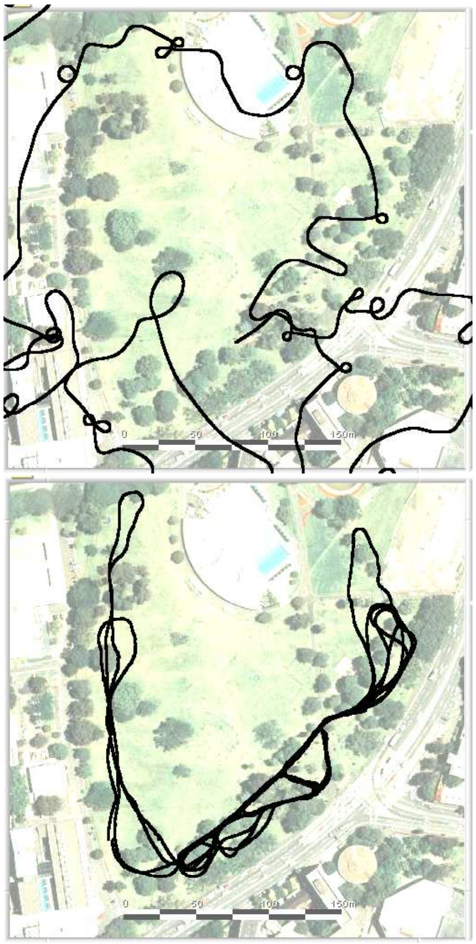
Our exact mechanism for building up the map is closely related to standard procedures in the SLAM community [2]. Due to fake landmark detections caused by moving objects, additional care has to be taken to filter out those interfering measurements. For any detected object that can not be explained by existing landmarks, a new landmark candidate is generated but not put into SEIF directly. Instead it is added into a waiting list with a weight representing its probability of being a useful landmark. 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 landmarks.

We notice that data association violates the constant time property of SEIFs. This is because when calculating data associations, multiple landmarks have to be tested. If we can ensure that all plausible landmarks are already connected in the SEIF by a short path to the set of active landmarks, it would be feasible to perform data association in constant time. In this way, the SEIF structure naturally facilitates the search of the most

Figure 7: 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.

likely landmark 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 kd-trees, it appears to be feasible to implement data association in logarithmic time by recursively partitioning the space of all landmark locations using a tree.

Finally, we notice that another important operation can be done in constant time in SEIF: the merge of identical landmarks 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 landmarks into one upon the arrival of further sensor evidence.



(a)

(b)

Figure 8: 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.

4 Experimental Results

The purpose of our comparison was to evaluate the performance of SEIFs against that of the "gold standard," which is EKF from which SEIFs are derived. The vehicle and its environment are shown in Figures 7 and 8. The vehicle 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, see Figure 8(a). The data used for our experiments was previously used as a benchmark in several publications (see [2]). In our experiment on this real-world data, SEIF correctly recovers the robot path shown in Figure 8(b). Comparing with EKF, SEIF runs twice as fast and consumes less than a quarter of the memory EKF uses. SEIF's landmark position estimations differ from those of EKF's by 0.5 meter in terms of root mean square distance.

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 experiments on scaling our approach to different environment sizes. In our simulations, we focused particularly on the *loop closing* problem, which is generally acknowledged to be one of the hardest problems in SLAM. When closing a loop, usually many landmark locations are affected, testing our amortized map recovery mechanism under the hardest possible circumstances.

The robot simulator is set up so that unit area has 50 landmarks on average. The landmarks are randomly distributed in a squared region with a minimum distance of 0.05 between landmarks. As the number of landmarks increases, so does the area. 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. For k number of landmarks, 20k iterations are performed. This roughly makes the average number of visits to each landmark the same for the simulations of different number of landmarks. Maximum sensor range is set to 0.2, which translates to 6 landmark detections

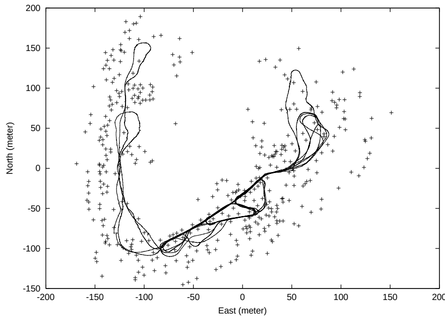


Figure 9: Overlay of estimated landmark positions and robot path.

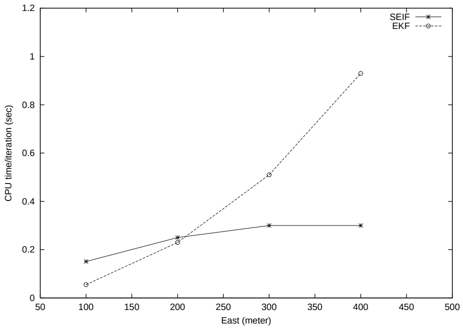


Figure 10: The comparison of average CPU time between SEIF and EKF.

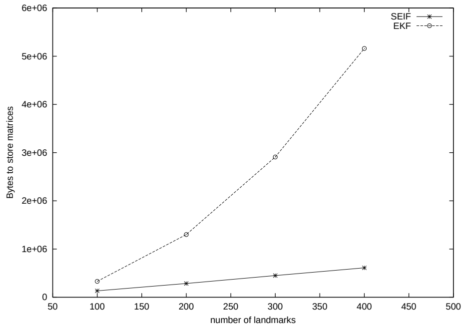


Figure 11: The comparison of average memory usage between SEIF and EKF.

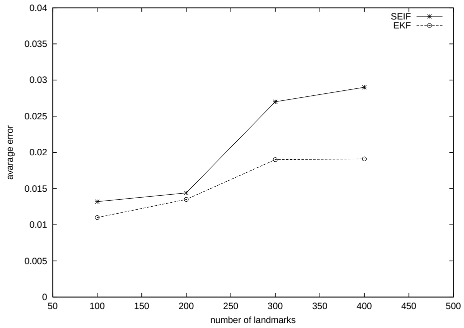


Figure 12: The comparison of root mean square distance error between SEIF and EKF.

on average for one measurement step. The maximum number of active landmarks is chosen to be 10. Figure 10 and 11 clearly show that SEIF beats EKF in terms of computation and memory usage. In the case of EKF, the usage of both computation and memory increases quadratically with respect to the number of landmarks, whereas for SEIF, CPU time per iteration comes to a constant when the number of landmarks goes beyond 300, and the memory used to store the information matrix increases only linearly. Due to the approximation of the information matrix and amortized map recovery, SEIF has bigger error than EKF as is shown in Figure 12. However the decrease in computation and memory costs can easily outweigh this small increase in errors.

5 Discussion

This paper summarized a new algorithm for the simultaneous localization and mapping (SLAM) problem, which can maintain globally consistent maps with constant update time. Our approach is based on the observation that in the information form of traditional Kalman-filter algorithms (EKF), most elements in the normalized information matrix are near zero. The sparse extended information filter, or SEIF, enforces a sparse information matrix, which can be updated in constant time. This paper also proposed a data association mechanism for SEIFs based on the maximum likelihood principle.

Since this algorithm is approximate, we presented empirical results, comparing SEIFs to the common EKF solution. Our results show that SEIFs produce results comparable to that of EKFs, but at a much improved computational time complexity (constant instead of quadratic time). The results were obtained for a well-known reference data set, recorded by researchers at the University of Sydney. Thus, with this paper, we fill an important gap: while previous results regarding SEIF were purely theoretical in nature, the results presented here shed light on the practical side of SEIFs. Based on our findings, we believe that SEIFs are scalable to much larger maps than the EKF or related hierarchical sub-mapping approaches.

Acknowledgment

The authors would like to acknowledge invaluable contributions by the following researchers: Hugh Durrant-Whyte, Zoubin Ghahmarani, Geoffrey Gordon, Daphne Koller, Kevin Murphy, Andrew Ng, Mark Paskin, and Michael Stevens. This research has been sponsored by DARPA's MARS Program (contract N66001-01-C-6018 and contract NBCH1020014), DARPA's CoABS Program (contract F30602-98-2-0137), and DARPA's MICA Program (contract F30602-01-C-0219), all of which are gratefully acknowledged.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

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

[6] 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.

[7] P. Maybeck. *Stochastic Models, Estimation, and Control, Volume 1*. Academic Press, Inc, 1979.

[8] 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.

[9] P. Moutarlier and R. Chatila. Stochastic multisensory data fusion for mobile robot location and environment modeling. In *5th Int. Symposium on Robotics Research*, Tokyo, 1989.

[10] 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.

[11] 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.

[12] R. C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. Technical Report TR 4760 & 7239, SRI, 1985.

[13] S. Thrun, D. Koller, Z. Ghahmarani, and H. Durrant-Whyte. SLAM updates require constant time. Submitted for publication, 2002.

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

[15] 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.