
Tradeoffs in SLAM with Sparse Information Filters

Zhan Wang, Shoudong Huang, and Gamini Dissanayake

ARC Centre of Excellence for Autonomous Systems (CAS),
Faculty of Engineering, University of Technology, Sydney, Australia
{zwang,sdhuang,gdissa}@eng.uts.edu.au

Summary. Designing filters exploiting the sparseness of the information matrix for efficiently solving the simultaneous localization and mapping (SLAM) problem has attracted significant attention during the recent past. The main contribution of this paper is a review of the various sparse information filters proposed in the literature to date, in particular, the compromises used to achieve sparseness. Two of the most recent algorithms that the authors have implemented, Exactly Sparse Extended Information Filter (ESEIF) by Walter et al. [5] and the D-SLAM by Wang et al. [6] are discussed and analyzed in detail. It is proposed that this analysis can stimulate developing a framework suitable for evaluating the relative merits of SLAM algorithms.

1 Introduction

Use of an Extended Information Filter (EIF) to solve SLAM to gain computational savings has been demonstrated by a number of researchers. A notable result has been that from Thrun et al. [2] where a sparsification process is used to reduce the number of non-zero elements in the information matrix. Another way to ensure that the information matrix is sparse is to include both features and a sequence of robot poses in the state vector [13] [3]. Alternatively SLAM can also be formulated with a state vector containing only robot poses, as shown in the Exactly Sparse Delayed State Filter (ESDSF) [4]. Very recently, two SLAM algorithms, which can achieve exactly sparse information matrix without including previous robot poses in the state vector, were developed independently. One is the Exactly Sparse Extended Information Filter (ESEIF) by Walter et al. [5] and the other is the D-SLAM by Wang et al. [6].

All above algorithms rely on the sparse structure of the information matrix to achieve computational benefits. Sparseness is achieved through various tradeoffs. Although the estimation process becomes efficient, probabilistic data association requires the recovery of the relevant elements of the state as well as the associated covariance matrix, and may have a significant impact on the computational cost of a practical implementation. Although each algorithm claims different advantages, no comprehensive examination is so far available to make clear the relative merits and the compromises used. This paper addresses this important issue by providing a classification and a qualitative analysis of SLAM algorithms

Table 1. A summary of sparse information filters used in SLAM – here N is the total number of 2D features and M is the number of selected 3D robot poses

SLAM Algorithm	Dimension of state vector	Consistency	Information loss	Covariance recovery for data association
SEIF [2]	$2N+3$	No[5]	–	approximate
SAM [3][7]	$3M+2N$	Yes	No	exact [7]
ESDSF [4]	$3M$	Yes	See Section 2.3	approximate
ESEIF [5]	$2N+3$	Yes	Yes	approximate
D-SLAM [6]	$2N$	Yes	Yes	exact

based on sparse information filters published to date. ESEIF and D-SLAM are analyzed in detail and a quantitative comparison is provided. It is hoped that this paper will stimulate further research towards the development of a comprehensive strategy and benchmark datasets for evaluating SLAM algorithms under a variety of different conditions.

The paper is organized as follows. Section 2 summarizes different sparse information filters for SLAM. Simulation and experiment comparisons of ESEIF and D-SLAM are provided in Section 3. Section 4 concludes the paper.

2 Summary of Sparse Information Filters in SLAM

This section classifies the sparse information filters published so far based on the strategy used to achieve sparseness. Only the algorithms based on a filtering process are discussed. Other valuable contributions that use sparse representations such as [10], [1], [11] and [9] are not examined.

2.1 Approximate the Information Matrix in EIF SLAM

An empirical finding that normalized information matrix obtained when the SLAM problem is formulated in the information form is approximately sparse, motivates the work by Thrun et al. [2], Sparse Extended Information Filter (SEIF). Theoretical explanation for this observation was later presented by Frese [13]. Sparsification essentially removes the weak links in the information matrix by setting elements that are smaller than a given threshold to zero, while strengthening other links to make up for the effect of this change.

State estimates for robot pose and a subset of features that are needed for computing Jacobians are recovered by solving a set of linear equations with a sparse coefficient matrix using relaxation. Data association is solved by

approximating the data association probability that an observation under consideration originates from a feature in the map and using the standard maximum likelihood method. However, Walter et al. [5] demonstrates that the sparsification process leads to inconsistent estimates.

2.2 Including Previous Robot Poses and Features in the State Vector

When all the previous robot poses and all the features are included in the state vector, the SLAM problem becomes a static estimation problem. This situation is discussed in Frese [13]. A considerable amount of off-diagonal elements in the information matrix are exactly zero and the information matrix is exactly sparse. This state vector is used in Square Root SAM [3].

In [3], SLAM is formulated as a linear least squares problem and is solved by factorizing the smoothing information matrix using Choleskey or QR factorization in a batch or incremental manner. Exact state and covariance recovery can be achieved by exploiting the special sparse structure of the factorization matrix resulting from the QR factorization [7]. The advantage of the SAM algorithm is that the quality of the estimate can be better than traditional EKF SLAM since in each step the estimates of all previous robot poses are updated together with the feature location estimates through the smoothing process. Thus the linearization error is reduced. Therefore, the SAM algorithm is less prone to estimator inconsistency that can arise due to linearization errors than all other EIF algorithms discussed in this paper.

However, the sparseness of the information matrix is achieved through increasing the state dimension, which keeps increasing even when robot is revisiting previously explored regions. Therefore, the computational cost increases over time and is not bounded by the number of features in the environment.

2.3 Including Only Robot Poses in the State Vector

By computing the relationship between two consecutive robot poses using the observation made at each pose, SLAM can be solved using a state vector containing only robot poses. The resulting information matrix is exactly sparse, as shown in the Exactly Sparse Delayed State Filter (ESDSF) [4].

A suboptimal partial state recovery is achieved by keeping the irrelevant states fixed at their current estimates and solving a set of sparse equations relating the state, the information matrix and the information vector. Data association is solved by estimating a bound for the covariance matrix. The key advantage of ESDSF is that it is suitable for the scenarios where features are difficult to extract or the number of features is too large as compared with robot poses, as demonstrated by the excellent maps shown in [4].

However, the resulting “map” is only an alignment of a sequence of observations (such as images or laser scans). There are no statistical map updates, thus improvements to the state estimates achieved through feature location updates

in traditional SLAM is not present. Extent of the information loss due to this has not yet been analyzed.

2.4 Including Only Features in the State Vector

D-SLAM, recently proposed by Wang et al. [6] uses a state vector that only contains the feature locations to generate maps of an environment. Robot location estimate is obtained through a concurrent yet separate process.

In D-SLAM mapping, the original measurements relating the robot and features are first transformed into relative distances and angles among features. Then these transformed measurements are fused into the map using EIF. It is shown that only the features that are observed at the same time instant have links in the information matrix making it exactly sparse. The extent of sparseness is governed by the sensor range and feature density. Localization is performed by combining two estimates: one is obtained by solving a “kidnapped robot problem”; the other is obtained by a local EKF SLAM where only the features currently observed are retained in the state vector. The two correlated estimates are fused by Covariance Intersection (CI) [6].

Exact state and covariance recovery is achieved by preconditioned Conjugated Gradient (PCG). A good preconditioner produced by an iterative Cholesky factorization method by exploiting the similarity between the information matrices of successive steps is used to make the PCG efficient. Data association is solved by a combination of the standard maximum likelihood approach and a chi-square test.

There is some information loss in D-SLAM, and this is further addressed in Section 3.

2.5 Periodic Marginalization of Robot Pose

The ESEIF algorithm by Walter et al. [5] achieves sparse information matrix by periodically marginalizing out and relocating robot.

Similar to SEIF, ESEIF exploits the fact that when the robot location is marginalized out from the state vector, new links will only be built up among the features that were previously linked with the robot in the information matrix. The set of features that are linked with the robot is called “active features”. Thus the information matrix will be sparse if the number of “active features” is bounded. In contrast to the sparsification process in SEIF, ESEIF controls the number of “active features” by “kidnapping” the robot when the number of “active features” is about to become larger than a predefined threshold Γ_a . This is followed by “relocating” the robot using a set of selected measurements. Thus, the EIF information matrix is kept sparse without any approximations that can lead to inconsistency. The extent of sparseness is controlled by the active feature bound Γ_a , the sensor range and feature density.

There is some information loss in ESEIF due to “kidnapping” and “relocating” the robot. This is further addressed in Section 3.

3 Comparison of Information Loss in ESEIF and D-SLAM

ESEIF [5] and D-SLAM [6] have been selected for a detailed comparison as both these algorithms have a state vector of the same character and do not use any approximations to achieve sparseness. As the authors do not have access to an efficient implementation of ESEIF, a quantitative analysis of the computational cost can not be presented. This section, therefore, focuses on the extent of information loss.

3.1 Comparison Using 1D Simulations

An analysis of the mapping performance of ESEIF and D-SLAM based on a linear one-dimensional form of the SLAM problem is presented below. The 1D simulation is used so that the effects due to linearization errors are avoided.

The scenario consists of a set of uniformly distributed features arranged on a straight line with a interval of $1m$. The robot, moving forward and backward along this line, can measure the distances to features within its sensor range ($9m$). At each time step, robot observes 9 features. The robot only makes one observation before moving.

Four parameters are considered: the active feature bound Γ_a , the process noise, the robot speed and the time steps. Extensive simulations were conducted by varying these parameters over a large range of values. The trace of the sub-matrix of the covariance matrix corresponding to all features from ESEIF, P_{mm} , and the trace of the covariance matrix from D-SLAM mapping, P , are used to indicate the accumulated information. The results obtained are summarized in Table 2, from which the following conclusions can be drawn.

(1) When the active feature bound Γ_a is smaller than the number of feature in each observation, ESEIF performs worse than D-SLAM, because in this case robot is marginalized out and relocated at each step in ESEIF. When the active feature bound Γ_a is gradually increased, number of instances where the robot needs to be relocated reduces and at some point ESEIF map becomes better than that obtained using D-SLAM.

(2) When all other parameters are fixed, the larger the robot speed, the larger the range of Γ_a in which D-SLAM map is better (compare rows 4 – 6 with rows 1 – 3 in Table 2). This is because when robot runs faster, robot relocation step becomes more frequent resulting in more information loss.

(3) When all other parameters are fixed and the process noise is small enough, ESEIF map is always better than D-SLAM as long as Γ_a is at least equal to the the number of features in one observation (see rows 10 – 12 in Table 2). When all other parameters are fixed, the larger the process noise, the larger the range of Γ_a in which D-SLAM map is better (compare rows 7 – 9 with rows 1 – 3 in Table 2). With larger process noise, process model contains less information, thus the information loss in D-SLAM is less.

(4) When other parameters are fixed, the smaller the time steps, the larger the range of Γ_a in which ESEIF map is better (compare rows 13 – 15 with rows

Table 2. Evaluation of mapping in ESEIF and D-SLAM using 1D simulations

Test No.	Active feature bound Γ_a	Process noise (m/s)	Speed (m/s)	Time steps	D-SLAM trace of P	ESEIF trace of P_{mm}	Trace ratio $\frac{D-SLAM}{ESEIF}$
1	8	0.5	1	280	0.087099	0.099094	0.87895
2	10	0.5	1	280	0.087099	0.087564	0.99468
3	11	0.5	1	280	0.087099	0.087074	1.00029
4	8	0.5	2	280	0.423266	0.518746	0.87895
5	23	0.5	2	280	0.423266	0.423358	0.99978
6	24	0.5	2	280	0.423266	0.423229	1.00009
7	8	2	1	280	0.087099	0.104539	0.83317
8	33	2	1	280	0.087099	0.087105	0.99994
9	34	2	1	280	0.087099	0.087097	1.00002
10	8	0.2	1	280	0.087099	0.089111	0.97743
11	9	0.2	1	280	0.087099	0.083668	1.04101
12	10	0.2	1	280	0.087099	0.08311	1.04801
13	8	0.5	1	600	0.236643	0.281696	0.84006
14	13	0.5	1	600	0.236643	0.23673	0.99963
15	14	0.5	1	600	0.236643	0.236378	1.00112

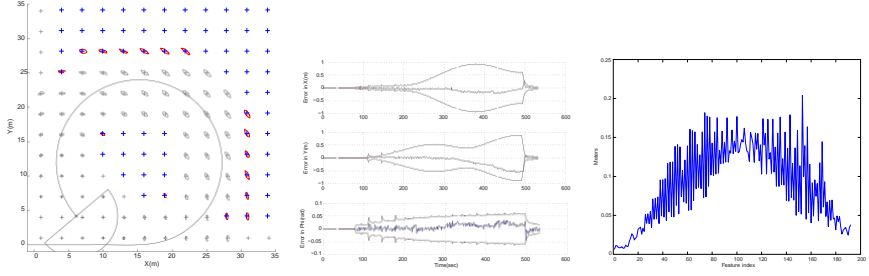
1 – 3 in Table 2). This is due to the different accumulation speed of information loss in the two algorithms.

3.2 Comparison Using 2D Simulations

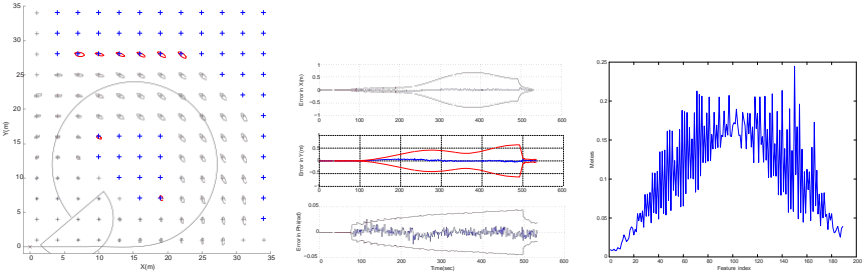
A 2D simulation using larger number of features was conducted to further compare the two algorithms. The environment used is a 35 meter square with 144 features arranged in uniformly spaced rows and columns. The robot trajectory are shown in Figures 1(a) and 1(d). A sensor with a field of view of 180 degrees and a range of 6 meters is simulated to generate relative range and bearing measurements between the robot and the features.

In this simulation, all the parameters are set to values for an indoor scenario. The standard deviations of the noises in robot speed and turnrate are 0.05 m/s and 4 degree/s respectively, and those of the noises in sensor range and bearing are 0.1 m and 1 degree respectively. Robot explores in the environment at the speed of 0.4 m/s for 533 time steps. At each time step, robot can observe 5 – 6 features. Γ_a is set to be 10.

As expected, with small process noise and robot speed, ESEIF provides more accurate map than D-SLAM as shown in Figures 1(a) and 1(d). This can be seen more clearly in Figures 1(c) and 1(f). Both algorithms provide consistent



(a) Map from ESEIF (b) Robot location estimation errors and 2σ error for all feature location estimates from ESEIF (c) Final 1σ error bounds for all feature location estimates from ESEIF



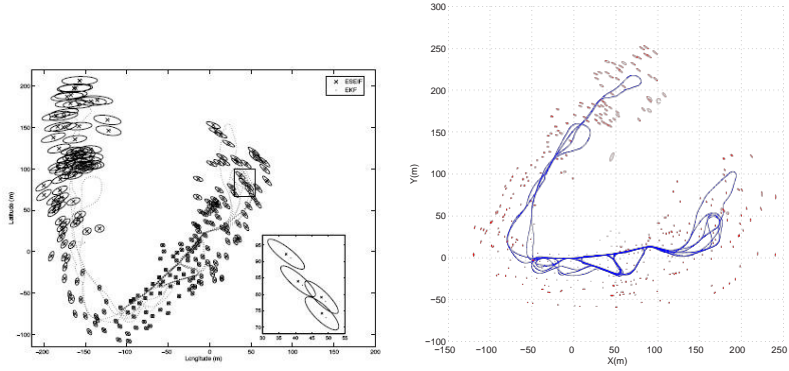
(d) Map from D-SLAM (e) Robot location estimation errors and 2σ error for all feature location estimates from D-SLAM (f) Final 1σ error bounds for all feature location estimates from D-SLAM

Fig. 1. Simulation results

robot location estimation. However, robot location estimation in D-SLAM is more accurate as shown in Figures 1(b) and 1(e).

3.3 Experimental Comparison

The large-scale outdoor Victoria Park data set [14] is also used for the comparison. As the ground truth is not available, it is impossible to use this data set for evaluating whether a given algorithm is consistent. Furthermore, some sparse information filter based SLAM papers use this data set with different pre-defined data associations, and different noise parameters. It is, therefore, unreasonable to compare the outcomes from this data set except as a proof that the algorithms concerned are able to deal with some of the practical issues such as non-stationery objects and spurious measurements. Interestingly, the gross character of the vehicle trajectories reported in some publications differs from the results presented by the original authors who collected this data. Figures extracted from the original publications of D-SLAM and ESEIF are presented to highlight these issues.



(a) Map and robot trajectory from ESEIF (ellipses indicate 3σ error bounds; the figure is from [5]) (b) Map and robot trajectory from D-SLAM (ellipses indicate 3σ error bounds)

Fig. 2. Outdoor, large-scale implementation using Victoria Park data set

3.4 Analysis of Comparison Results between ESEIF and D-SLAM

Since D-SLAM transforms the original measurements to relative distances and angles among features, the information from process model is lost in mapping, as compared with traditional EKF SLAM. In localization, the information loss comes from using CI, which is a conservative way to fuse two pieces of information with unknown correlations. The main factors that affect the extent of information loss include: process and sensor noises, feature density, and time steps before robot starts moving. This is analyzed in detail in [6].

In ESEIF, when marginalizing out and relocating robot, the information from process model is lost. This process results in poor robot location estimation for the following step and the loss of the correlations between robot and features, as compared with traditional EKF SLAM. These correlations together with the robot uncertainty govern the improvement to the feature location estimates. Information is also lost due to the fact that one part of measurements (z_a in [5]) is used in normal update and another part (z_b in [5]) is used in relocating robot. During this process, the implicit information that these two parts of measurements are made from the same robot location is not exploited. The main factor governing the extent of information loss is how often robot is kidnapped and re-located. This is mainly determined by the active feature bound Γ_a , sensor range and the robot speed.

4 Conclusions and Discussions

Following the pioneering work by Thrun et al. [2], many different sparse information filters have been developed for SLAM. This paper summarized the relative merits of different sparse information filters. Especially, the extent of information loss of D-SLAM and ESEIF is compared in detail.

When the number of robot poses (at which observations are made) is not significant as compared with the total number of features, including all the robot poses in the state vector will produce better estimation than traditional EKF SLAM due to the smoothing of previous robot poses [3]. In the case when features are difficult to extract or the number of features is too large as compared with robot poses. The view-based ESDSF [4] is a good option since only robot poses are included in the state vector. When the number of robot poses is large as compared with the total number of features, ESEIF [5] and D-SLAM [6] are two efficient exactly sparse information filters.

Overall, sparse information filters provide promising solutions for the SLAM problems. The computational saving is significant although there are some trade-offs. More work is necessary to further investigate the recovery of state and covariance matrix such that real time data association can be achieved. Furthermore, it is perhaps a critical requirement that SLAM researchers collaborate to generate a set of benchmark data sets, both simulated and real-life, as well as performance metrics so that plethora of emerging SLAM algorithms can be compared in a consistent setting. The recently proposed sparse information filters based on submaps, Tectonic SAM [17], Treemap [18], D-SLAM local map joining [15], Sparse local submap joining filter (SLSJF) [16], which are not discussed in this paper due to the space limitation, introduce significant flexibilities in achieving the sparseness and the comparison of these algorithms remain the future work.

Acknowledgment

This work is supported by the ARC Centre of Excellence programme, funded by the Australian Research Council (ARC) and the New South Wales State Government.

References

1. Frese, U., Larsson, P., Duckett, T.: A multilevel relaxation algorithm for simultaneous localization and mapping. *IEEE Transactions on Robotics* 21(2), 196–207 (2005)
2. Thrun, S., Liu, Y., Koller, D., Ng, A.Y., Ghahramani, Z., Durrant-Whyte, H.: Simultaneous localization and mapping with sparse extended information filters. *International Journal of Robotics Research* 23(7-8), 693–716 (2004)
3. Dellaert, F., Kaess, M.: Square Root SAM: simultaneous location and mapping via square root information smoothing. *International Journal of Robotics Research* 25(12), 1181–1203 (2006)
4. Eustice, R.M., Singh, H., Leonard, J.: Exactly sparse delayed-state filters for view-based SLAM. *IEEE Transactions on Robotics* 22(6), 1100–1114 (2006)
5. Walter, M., Eustice, R., Leonard, J.: Exactly sparse Extended Information filters for feature-based SLAM. *International Journal of Robotics Research* 26(4), 335–359 (2007)

6. Wang, Z., Huang, S., Dissanayake, G.: D-SLAM: a decoupled solution to simultaneous localization and mapping. *International Journal of Robotics Research* 26(2), 187-204 (2007)
7. Kaess, M., Ranganathan, A., Dellaert, F.: Fast incremental square root information smoothing. In: *Proc. International Joint Conferences on Artificial Intelligence (IJCAI)*, pp. 2129-2134 (2007)
8. Krauthausen, P., Kipp, A., Dellaert, F.: Exploiting locality in SLAM by Nested Dissection. In: *Proc. Robotics: Science and Systems* (2006)
9. Paskin, M.: Thin junction tree filters for simultaneous localization and mapping. In: *Proc. International Joint Conferences on Artificial Intelligence (IJCAI)*, pp. 1157-1164 (2003)
10. Folkesson, J., Christensen, H.I.: Graphical SLAM - a self-correcting map. In: *Proc. IEEE International Conference on Robotics and Automation*, pp. 383-390 (2004)
11. Thrun, S., Montemerlo, M.: The GraphSLAM algorithm with applications to large-scale mapping of urban structures. *International Journal of Robotics Research* 25(5-6), 403-429 (2004)
12. Eustice, R., Singh, H., Leonard, J., Walter, M., Ballard, R.: Visually navigating the RMS Titanic with SLAM information filters. In: *Proc. Robotics: Science and Systems* (2005)
13. Frese, U.: A proof for the approximate sparsity of SLAM information matrices. In: *Proc. IEEE International Conference on Robotics and Automation*, pp. 331-337 (2005)
14. Guivant, J.E., Nebot, E.M.: Optimization of the simultaneous localization and map building (SLAM) algorithm for real time implementation. *IEEE Transactions on Robotics and Automation* 17(3), 242-257 (2001)
15. Huang, S., Wang, Z., Dissanayake, G.: Mapping large scale environments using relative position information among landmarks. In: *Proc. International Conference on Robotics and Automation*, pp. 2297-2302 (2006)
16. Wang, Z.: Exactly sparse information filters for simultaneous localization and mapping. Ph.D. Thesis. Centre of Excellence for Autonomous Systems, University of Technology, Sydney (2007)
17. Ni, K., Steedly, D., Dellaert, F.: Tectonic SAM: exact, out-of-core, submap-based SLAM. In: *Proc. IEEE International Conference on Robotics and Automation*, pp. 1678-1685 (2007)
18. Frese, U.: Treemap: An $O(\log n)$ algorithm for indoor simultaneous localization and mapping. *Autonomus Robots* 21, 103-122 (2006)