Iterated SLSJF: A Sparse Local Submap Joining Algorithm with Improved Consistency

Shoudong Huang, Zhan Wang and Gamini Dissanayake
The University of Technology, Sydney, Australia
{sdhuang,zhwang,gdissa}@eng.uts.edu.au
Udo Frese
University of Bremen, Germany
ufrese@informatik.uni-bremen.de

Abstract

This paper presents a new local submap joining algorithm for building large-scale feature based maps. The algorithm is based on the recently developed Sparse Local Submap Joining Filter (SLSJF) and uses multiple iterations to improve the estimate and hence is called *Iterated SLSJF (I-SLSJF)*. The input to the I-SLSJF algorithm is a sequence of local submaps. The output of the algorithm is a global map containing the global positions of all the features as well as all the robot start/end poses of the local submaps.

In the submap joining step of I-SLSJF, whenever the change of state estimate computed by an Extended Information Filter (EIF) is larger than a predefined threshold, the information vector and the information matrix is recomputed as a sum of all the local map contributions. This improves the accuracy of the estimate as well as avoids the possibility that the Jacobian with respect to the same feature gets evaluated at different estimate values, which is one of the major causes of inconsistency for EIF/EKF algorithms. Although the computational cost of I-SLSJF is higher than that of SLSJF, the algorithm can still be implemented efficiently due to the exactly sparseness of the information matrix. The new algorithm is compared with EKF SLAM and SLSJF using both computer simulation and experimental examples.

1 Introduction

Feature based simultaneous localization and mapping (SLAM) is the process of building a feature map of an environment while concurrently generating an estimate for the location of the robot. Local submap joining ([Tardos *et al.*, 2002][Williams, 2001] [Castellanos *et al.*, 2007][Ni *et al.*, 2007]) provides an efficient way to build large-scale maps. In local submap joining, a sequence of small sized local submaps are built (e.g. using Extended Kalman Filter (EKF) SLAM [Dissanayake *et al.*, 2001]) and then combined into a large-scale global map.

In the EKF sequential map joining [Tardos *et al.*, 2002], the resulting covariance matrix of the map is fully correlated and thus the EKF map fusion process is computationally demanding. Overall computational savings are achieved due to the fact that the frequency of global map updates is reduced.

In our recent work [Huang *et al.*, 2008a; 2008b], it was demonstrated that local submap joining can be achieved through the use of a sparse information filter. The proposed map joining filter, Sparse Local Submap Joining Filter (SLSJF), combines the advantages of the local submap joining algorithms and the sparse representation of SLAM to substantially reduce the computational cost of the global map construction.

Local submap based strategies can also improve the consistency of SLAM [Castellanos *et al.*, 2007][Huang and Dissanayake, 2007]. Simulation results show that SLSJF is more consistent as compared with a single EKF SLAM for many cases. However, SLSJF may still produce inconsistent estimate in some scenarios (e.g. when closing very large loops) due to the fact that linearization assumptions used to derive EIF equations may be violated if the errors in the state estimate is large.

In SLSJF, all the robot start/end poses are in the global state vector and there is no prediction step within the EIF. Thus the estimation problem can also be formulated as a least squares problem and can be solved more accurately by using multiple iterations at each map fusion step.

In this paper, we propose the Iterated SLSJF (I-SLSJF) algorithm. The algorithm starts from the EIF but performs multiple linearized least squares iterations at each map fusion step whenever necessary. In particular, at each map fusion step, if the change of state estimate before and after update is larger than a threshold,

then the information matrix and the information vector are recomputed as a sum of all the local submap contributions using the new estimate as linearization point for the Jacobians. This process is able to further improve the consistency. Although I-SLSJF requires more computational cost as compared with SLSJF, it is still very efficient due to the sparseness of the information matrix.

The paper is organized as follows. Section 2 briefly review the key points of SLSJF algorithm. The new I-SLSJF map joining algorithm is described in detail in Section 3. Simulation and experiment results are provided in Section 4. Section 5 discusses some related work. Section 6 concludes the paper.

2 A brief review of SLSJF

This section briefly review the main ideas of the SLSJF algorithm. SLSJF essentially combines a sequence of maps of a small local region [1] generated using any conventional SLAM algorithm to build a global map of the environment in a computationally efficient manner.

2.1 Use of local submaps as observations

When fusing the local maps into the global map, SLSJF treats each local map as an observation. Since the local map provides a consistent estimate of the relative positions from robot start pose to the local features and the robot end pose, this map can be treated as

> "*an observation made from the robot start pose to all the features in the local map and a virtual robot located at the robot end pose*".

The "observation value" is the local map state estimate and the "observation noise" is zero-mean Gaussian with covariance matrix equal to the local map covariance matrix.

2.2 Sparse Information Matrix

By including all the features and all the robot start/end poses of the local maps in the global map state vector, the local map joining problem becomes a large-scale static estimation problem with only "local" information (each local map only involves some "nearby objects" — the features and the robot start/end poses involved in the local map). When Extended Information Filter (EIF) is used to solve the estimation problem, a non-zero off-diagonal element of the information matrix (a "link" between the two related objects) occurs only when the two objects are within the same local map. Since the size of each local map is limited, any object will only have links with its "nearby objects" no matter how many (overlapping) local maps are fused (Fig. 1). This results in an exactly sparse information matrix (similar to Smoothing and Mapping (SAM) [Dellaert and Kaess, 2006] and full SLAM [Thrun *et al.*, 2005]).

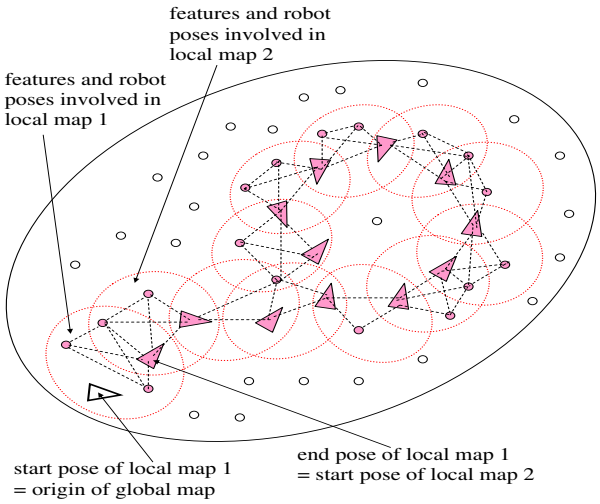


Figure 1: In local map joining, each object (feature or robot pose) is only linked to its "nearby objects" (features and robot poses that share the same local map with it). The final global map state vector of I-SLSJF contains the locations of all the shaded objects.

Since all the objects involved in the local maps are included in the global state vector, no marginalization is required during the map joining process and thus the information matrix will stay exactly sparse all the time. As most of the robot poses are marginalized out during the local map building process, the dimension of the global state vector is much less than that of SAM [Dellaert and Kaess, 2006] and full SLAM [Thrun *et al.*, 2005].

2.3 Consistency of SLSJF

The SLSJF algorithm does not contain any approximations (such as sparsification [Thrun *et al.*, 2004]) that can lead to estimator inconsistency. Moreover, SLSJF is a local map based strategy, which can improve the consistency of SLAM as compared with single EKF SLAM for many cases [Castellanos *et al.*, 2007][Huang and Dissanayake, 2007]. It was shown in [Huang *et al.*, 2008b] that SLSJF is able to generate consistent maps for large scale simulations. However, as the case with all EKF/EIF based estimation algorithms, it is still possible that inconsistencies occur in SLSJF due to errors introduced by the linearization process, especially when robot travels very large loops.

[1] In this paper, "local submap" and "local map" are both used to describe such maps.

In SLSJF, all the robot start/end poses are in the global state vector and there is no prediction step within the EIF. Thus the map joining problem can also be formulated as a least squares problem. A more accurate solution can be obtained by using multiple iterations in each map fusion step, without any impact on the sparseness of the information matrix. This motivate the I-SLSJF algorithm in this paper.

3 The I-SLSJF algorithm

The state vector in I-SLSJF is the same as that in SLSJF [Huang *et al.*, 2008a][Huang *et al.*, 2008b]. The difference between the two algorithms is the map fusion process.

3.1 The input and output

The input to the I-SLSJF is a sequence of local submaps constructed by some SLAM algorithm. A local map is denoted by

$$(\hat{X}^L, P^L) \tag{1}$$

where $\hat{X}^L$ (the superscript 'L' stands for the local map) is an estimate of the state vector

$$\begin{array}{rcl} X^L & = & (X_r^L, X_1^L, \cdots, X_n^L) \\ & = & (x_r^L, y_r^L, \phi_r^L, x_1^L, y_1^L, \cdots, x_n^L, y_n^L) \end{array} \tag{2}$$

and P^L is the associated covariance matrix. The state vector X^L contains the robot final pose X_r^L (the subscript 'r' stands for the robot) and all the local feature positions $X_1^L, \cdots, X_n^L$, as typically generated by conventional EKF SLAM. The coordinate system of a local map is defined by the robot pose when the building of the local map is started, i.e. the robot starts at the coordinate origin of the local map.

It is assumed that the robot starts to build local map $k+1$ as soon as it finishes local map k. Therefore the robot end pose of local map k (defined as the global position of the last robot pose when building local map k) is the same as the robot start pose of local map $k+1$ (Fig. 1).

The output of I-SLSJF is a global map. The global map state vector contains all the feature positions and all the robot end poses of the local maps (see Fig. 1). The global map result is given in the form of a global state estimate, an information vector and an information matrix.

3.2 State vector of the global map

For convenience, the origin of the global map is chosen to be the same as the origin of local map 1 (Fig. 1).

After local maps 1 to k are fused into the global map, the global state vector is denoted as $X^G(k)$ (the superscript 'G' stands for the global map) and is given by

$$\begin{array}{rcl} & & X^G(k) \\ & = & (X_1^G, \cdots, X_{n_1}^G, X_{1e}^G, \\ & & X_{n_1+1}^G, \cdots, X_{n_1+n_2}^G, X_{2e}^G, \\ & & \cdots\cdots \\ & & X_{n_1+\cdots+n_{k-1}+1}^G, \cdots, X_{n_1+\cdots+n_{k-1}+n_k}^G, X_{ke}^G) \end{array} \tag{3}$$

where $X_1^G, \cdots, X_{n_1}^G$ are the global positions of the features in local map 1; $X_{n_1+1}^G, \cdots, X_{n_1+n_2}^G$ are the global positions of those features in local map 2 but not in local map 1; $X_{n_1+\cdots+n_{k-1}+1}^G, \cdots, X_{n_1+\cdots+n_{k-1}+n_k}^G$ are the global positions of the features in local map k but not in local maps 1 to $k-1$. $X_{ie}^G = (x_{ie}^G, y_{ie}^G, \phi_{ie}^G)$ $(1 \le i \le k)$ is the global position of the robot end pose of local map i, which is also the robot start pose of local map $i+1$. Here the subscript 'e' stands for robot 'end pose'.

When fusing local map $k+1$ into the global map, the features that are present in local map $k+1$ but have not yet been included in the global map, $X_{n_1+\cdots+n_k+1}^G, \cdots, X_{n_1+\cdots+n_k+n_{k+1}}^G$, together with the robot end pose of local map $k+1$, $X_{(k+1)e}^G$, are added into the global state vector $X^G(k)$ in (3) to form the new state vector $X^G(k+1)$.

3.3 Local map fusion as a least squares problem

Suppose local map j is given by $(\hat{X}_j^L, P_j^L)$. Since the local map provides a consistent estimate of the relative positions from robot start pose to the local features and the robot end pose, this map can be treated as an observation of the true relative positions with a zero-mean Gaussian noise whose covariance matrix is P_j^L.

To state it clearly, suppose the features involved in local map j are $X_{j1}^G, \cdots, X_{jn}^G$, then the local map state estimate $\hat{X}_j^L$ can be regarded as an observation of the true relative positions from the robot start pose $X_{(j-1)e}^G$ to the features $X_{j1}^G, \cdots, X_{jn}^G$ and the robot end pose X_{je}^G. That is,

$$\hat{X}_j^L = H_j(X^G(k)) + w_j \tag{4}$$

where $H_j(X^G(k))$ is the vector of relative positions given by

$$\begin{pmatrix} (x_{je}^G - x_{(j-1)e}^G)\cos\phi_{(j-1)e}^G + (y_{je}^G - y_{(j-1)e}^G)\sin\phi_{(j-1)e}^G \\ (y_{je}^G - y_{(j-1)e}^G)\cos\phi_{(j-1)e}^G - (x_{je}^G - x_{(j-1)e}^G)\sin\phi_{(j-1)e}^G \\ \phi_{je}^G - \phi_{(j-1)e}^G \\ (x_{j1}^G - x_{(j-1)e}^G)\cos\phi_{(j-1)e}^G + (y_{j1}^G - y_{(j-1)e}^G)\sin\phi_{(j-1)e}^G \\ (y_{j1}^G - y_{(j-1)e}^G)\cos\phi_{(j-1)e}^G - (x_{j1}^G - x_{(j-1)e}^G)\sin\phi_{(j-1)e}^G \\ \vdots \\ (x_{jn}^G - x_{(j-1)e}^G)\cos\phi_{(j-1)e}^G + (y_{jn}^G - y_{(j-1)e}^G)\sin\phi_{(j-1)e}^G \\ (y_{jn}^G - y_{(j-1)e}^G)\cos\phi_{(j-1)e}^G - (x_{jn}^G - x_{(j-1)e}^G)\sin\phi_{(j-1)e}^G \end{pmatrix}$$

and w_j is the zero-mean Gaussian "observation noise"

whose covariance matrix is P_j^L (when $j = 1$, $x_{(j-1)e}^G = 0, y_{(j-1)e}^G = 0, \phi_{(j-1)e}^G = 0$).

So the problem of fusing local maps 1 to k is to estimate the global state $X^G(k)$ using all the local map information (4) for $j = 1, \cdots, k$. This problem can be formulated as a weighted least squares problem:

$$\min_{X^G(k)} \sum_{j=1}^{k} (\hat{X}_j^L - H_j(X^G(k)))^T (P_j^L)^{-1} (\hat{X}_j^L - H_j(X^G(k))). \tag{5}$$

The least squares problem can be solved iteratively using linearizations. In fact, the following linear equation can be used to compute the new estimate $\hat{X}_{new}^G(k)$ when the previous estimate $\hat{X}_{old}^G(k)$ is available:

$$\begin{aligned} &\left[\sum_{j=1}^{k} \nabla H_j^T (P_j^L)^{-1} \nabla H_j\right] \hat{X}_{new}^G(k) \\ &= \sum_{j=1}^{k} \nabla H_j^T (P_j^L)^{-1} [\hat{X}_j^L - H_j(\hat{X}_{old}^G(k)) + \nabla H_j \hat{X}_{old}^G(k)] \end{aligned} \tag{6}$$

where ∇H_j is the Jacobian of the function H_j with respect to $X^G(k)$ evaluated at $\hat{X}_{old}^G(k)$.

In (6), the matrix

$$I(k) = \sum_{j=1}^{k} \nabla H_j^T (P_j^L)^{-1} \nabla H_j$$

is called information matrix. Note that the information matrix is an exactly sparse matrix, so the state estimate $\hat{X}_{new}^G(k)$ can be obtained by solving a sparse linear equation, which can be done efficiently by sparse Cholesky decomposition. The inverse of the information matrix is the covariance matrix of the state estimate.

Same as SLSJF, I-SLSJF fuses the local maps sequentially to build a global map, in a manner similar to [Tardos *et al.*, 2002][Williams, 2001], using the structure presented in Algorithm 1.

Algorithm 1 Overall structure of I-SLSJF

1: Set local map 1 as the *global map*
2: For $k = 2 : p$ (p is the total number of local maps), fuse *local map* k into the *global map*
3: End

The steps used in fusing local map $k+1$ into the global map are listed in Algorithm 2.

3.4 Data Association

Data association here refers to finding the features in local map $k + 1$ that are already included in the global map and their corresponding indices in the global state vector. It can be performed using the same procedure as that in SLSJF [Huang *et al.*, 2008b]. When the estimation error is too large (e.g. when closing large loops), some global localization techniques (such as [Paz *et al.*, 2005]) is necessary for finding the match between the local map and the global map.

Algorithm 2 Fuse local map $k + 1$ into global map

1: Data association
2: Initialization using EIF
3: Update using EIF
4: Use least squares to do smoothing when necessary

3.5 Initialization using EIF

The initial values of the global positions of all unmatched features and the robot end pose of local map $k + 1$ are computed (using $\hat{X}_{ke}^G$ and the local map state estimate $\hat{X}_{k+1}^L$) and inserted to $\hat{X}^G(k)$ to form a new state vector estimate $\hat{X}^G(k)$. The dimensions of $i(k)$, $I(k)$ and L_k are increased by adding zeros to form a new information vector $i(k)$ and a new information matrix $I(k)$.

3.6 Update the global map using EIF

Algorithm 3 Update using EIF

1: Compute the information matrix and information vector using EIF
2: Reorder the global map state vector when necessary
3: Compute the Cholesky Factorization of $I(k + 1)$
4: Recover the global map state estimate $\hat{X}^G(k + 1)$

Compute the information matrix and information vector

The information matrix and information vector are computed using EIF formula

$$\begin{aligned} I(k+1) &= I(k) + \nabla H_{k+1}^T (P_{k+1}^L)^{-1} \nabla H_{k+1} \\ i(k+1) &= i(k) + \nabla H_{k+1}^T (P_{k+1}^L)^{-1} [\hat{X}_{k+1}^L \\ &\quad - H_{k+1}(\hat{X}^G(k)) + \nabla H_{k+1} \hat{X}^G(k)] \end{aligned} \tag{7}$$

where ∇H_{k+1} is the Jacobian of the function H_{k+1} with respect to $X^G(k)$ evaluated at $\hat{X}^G(k)$.

Reorder the global map state vector when necessary

The purpose of reordering the global state vector is to make the computation of Cholesky factorization and the state vector recovery (Steps 3 and 4 in Algorithm 3) and the covariance submatrix recovery (a step in the Data Association algorithm) more efficient. Many different strategies for reordering are available. One popular strategy that can be applied is the Approximately

Minimal Degree (AMD) reordering which has been used in [Dellaert and Kaess, 2006][Kaess *et al.*, 2007].

Once the state vector is reordered, the corresponding information matrix $I(k+1)$ and information vector $i(k+1)$ are reordered accordingly. For notational simplicity, they are still denoted as $I(k+1)$ and $i(k+1)$.

Compute the Cholesky factorization of $I(k+1)$
The Cholesky factorization of $I(k+1)$ is needed to recover the state vector and the covariance submatrix (for data association). Note that $I(k+1)$ is a sparse positive definite matrix.

State vector recovery
Because the global map is maintained as an information vector and an information matrix, the global state estimate $\hat{X}^G(k+1)$ is not directly available. The state vector estimate $\hat{X}^G(k+1)$ can be recovered by solving the sparse linear equation

$$I(k+1)\hat{X}^G(k+1) = i(k+1) \tag{8}$$

which can be done by first solving $L_{k+1}Y = i(k+1)$ and then solving $L_{k+1}^T\hat{X}^G(k+1) = Y$ where L_{k+1} is the Cholesky factorization of $I(k+1)$.

3.7 Smoothing using least squares

The steps for smoothing using the least squares method are listed in Algorithm 4.

Algorithm 4 Smoothing using least squares

1: Recompute the information matrix $I(k+1)$ and the information vector $i(k+1)$
2: Compute the Cholesky Factorization of $I(k+1)$
3: Recover the global map state estimate $\hat{X}^G(k+1)$
4: Repeat the above process until $\hat{X}^G(k+1)$ converges.

When recomputing the information matrix $I(k+1)$ and the information vector $i(k+1)$, the new state estimate is used to compute the Jacobians. Formulas

$$I(k+1) = \sum_{j=1}^{k+1} \nabla H_j^T (P_j^L)^{-1} \nabla H_j \tag{9}$$

and

$$\begin{aligned} & i(k+1) \\ &= \sum_{j=1}^{k} \nabla H_j^T (P_j^L)^{-1} [\hat{X}_j^L - H_j(\hat{X}_{old}^G(k)) + \nabla H_j \hat{X}_{old}^G(k)] \end{aligned} \tag{10}$$

are used to compute the information matrix and the information vector.

The steps required for computing the Cholesky factorization of $I(k+1)$ and recovering of the global map state estimate are the same as those described in Section 3.6.

3.8 Consistency improvement

In SLSJF, an EIF is applied in the map fusion process. EIF formula (7) is equivalent to a one-step linearized least squares with the information matrix and information vector incrementally computed.

There are two major reasons why I-SLSJF is more consistent as compared with SLSJF. Firstly, since multiple iterations are used and the Jacobians are evaluated at the final estimate, the state estimate obtained in I-SLSJF is the optimal solution of the least squares problem (5). Secondly, since the information matrix and information vector are recomputed as a sum of all the local map contributions, it avoids the scenarios where the Jacobian with respect to the same feature be evaluated at different estimate values, which is one of the major causes of inconsistency for EIF/EKF algorithms [Huang and Dissanayake, 2007].

4 Simulation and experiment results

In this section, simulation and experiment results are presented to illustrate the consistency improvement of SLSJF.

4.1 Simulation results

The $150 \times 150m^2$ simulation environment used contains 2500 features arranged in uniformly spaced rows and columns. The robot started from the left bottom corner of the square and followed a big loop as shown in Fig. 2(a). A sensor with a field of view of 180 degrees and a range of 6 meters (the small semi-circle seen near the bottom in Fig. 2(b)) was simulated to generate relative range and bearing measurements between the robot and the features. There were 8241 robot poses in total and 40524 measurements were made from the robot poses. The robot observed 612 features in total (Fig. 2(a)).

In order to compare the performance of EKF, SLSJF and I-SLSJF, the data association was assumed in all the algorithms.

Fig. 2(b) shows the map generated by a single EKF SLAM. It is obvious that the map is inconsistent. Fifty small sized local maps were built by conventional EKF SLAM using the odometry and measurement information. Fig. 2(c) shows the global map generated by fusing all the 50 local maps using SLSJF. Close examination (e.g. Fig. 2(d)) shows that the feature position estimates computed by SLSJF methods are inconsistent.

Fig. 2(e) shows the global map generated by fusing all the 50 local maps using I-SLSJF. Close examination (e.g. Fig. 2(f)) shows that the feature position estimates computed by I-SLSJF appear to be consistent.

For this simulation, the total processing time (in MATLAB) for fusing all the 50 local maps is 2.1 seconds for SLSJF and 6.6 seconds for I-SLSJF (building the 50 local maps takes 14.6 seconds), while the processing time

95%-interval	I-SLSJF	SLSJF	EKF
[1132,1326]	1433	1665	1753

Table 1: NEES of the different estimates for the simulation data. For SLSJF and I-SLSJF, the estimates of the last robot pose and the feature positions are first extracted from the results and then the NEES is computed. The dimension of state is 1227 (one robot pose and 612 features)

for the single EKF SLAM algorithm is 432 seconds. How much extra computational cost for I-SLSJF as compared with SLSJF is problem dependent and also depends on the threshold for the linearized least squares iterations.

In order to quantitatively compare the consistency of the three estimates, the normalised estimation error squared (NEES) of the three estimates are computed. Table 1 shows the corresponding NEES values for the estimates from the three algorithms, as well as the (two-sided) 95% confidence interval. It can be seen that none of the three NEES fall in the 95% confidence interval, but the NEES of I-SLSJF is the closest to the interval. A more proper comparison can be done by conducting Monte Carlo runs and use the average NEES to compare the consistency [Bailey *et al.*, 2006].

4.2 Experimental results

The map joining algorithm was also applied to the DLR-Spatial-Cognition Data Set which was made available at *http://www.sfbtr8.spatial-cognition.de/insidedataassociation/data.html*

For this data set, artificial landmarks with white/black circles are placed on the ground. A robot equipped with a camera was moved around in the building and a sequence of images were taken (Fig. 3(a)). The robot trajectory and the structure of the building is shown in Fig. 3(b). The image data has been preprocessed and the relative position of the observed landmarks with respect to the observation point are provided. The odometry information is also available from the data set. The correct data association is given in the data set and it is not performed here in this paper. In this data set, there are $p = 3298$ robot poses, $n = 576$ landmarks and $m = 14309$ measurements.

Fig. 3(c) shows the map obtained by a single conventional EKF SLAM using the odometry and observation data. Two hundred local maps were built by EKF SLAM using the same data. Fig. 3(d) shows the global map obtained by joining the 200 local maps using SLSJF. Fig. 3(e) shows the global map obtained by joining the 200 local maps using I-SLSJF. Comparing with the robot trajectory and map in Fig. 3(b), it appears that the I-SLSJF result is more consistent than that of EKF SLAM and SLSJF.

Fig. 3(f) shows all the non-zero elements of the sparse information matrix obtained by I-SLSJF in black. There are 86879 non-zero elements and 2975621 zero elements in the 1750×1750 information matrix. For this data set, the total processing time for fusing all the 200 local maps is 12 seconds for SLSJF and 48 seconds for I-SLSJF (building the 200 local maps takes 9 seconds), while the single EKF SLAM algorithm takes 296 seconds.

Evaluating the consistency of an estimator on a real data-set is more difficult, if, as here, no ground-truth is available. However, some insight can be gained by comparing to the non-linear maximum likelihood (ML) estimate which is supposedly consistent. Usually an estimate is accepted as consistent if the χ^2 distance to the ground truth is less than the 95% level of the χ^2 distribution. So we can look at the χ^2 distance of an estimate to the ML estimate as an indicator of the additional error introduced by a more efficient algorithm.

Table 2 shows the χ^2 errors from different algorithms. In theory, the exact minimum of the χ^2 error should be $2m - 2n = 27466$ (the dimension of the measurements minus the dimension of the state) on average. This indicates that the measurement covariances in the data-set are overconfident by a factor of $56871.5/27466 = 2.07$. The theoretical 95%-bound is obtained from the χ^2 distribution with $2n = 1152$ degrees of freedom corrected by this factor [Press *et al.*, 1992, §15.6]. Table 2 shows that the additional error introduced by I-SLSJF is by a factor of 20 below the 95% bound so clearly acceptable. The EKF error is about twice as high. The SLSJF error is actually more than three times worse than EKF, clearly showing the benefit of I-SLSJF.

5 Related work

The sparse representations of SLAM recently proposed in the literature (e.g. [Thrun *et al.*, 2004][Dellaert and Kaess, 2006][Walter *et al.*, 2007][Wang *et al.*, 2007][Huang *et al.*, 2006]) make use of different state vectors and/or apply different strategies for marginalizing out robot poses.

In SAM [Dellaert and Kaess, 2006], incremental SAM (iSAM) [Kaess *et al.*, 2007], Tectonic SAM [Ni *et al.*, 2007] and full-SLAM [Thrun *et al.*, 2005], all the robot poses are included in the state vector and no marginalization is needed. However, the dimension of the state vector is very high, especially when the robot trajectory is long. The Sparse Extended Information Filter (SEIF) presented in [Thrun *et al.*, 2004] approximates the information matrix by a sparse one using sparsification, but this leads to inconsistent estimates [Walter *et al.*, 2007].

The Exactly Sparse Extended Information Filter (ESEIF) developed by [Walter *et al.*, 2007], the Decoupled SLAM (D-SLAM) algorithm [Wang *et al.*, 2007], and the

$\min_x \chi^2(x)$	95%-bound	$\chi^2(x_{\text{I-SLSJF}})$	$\chi^2(x_{\text{EKF}})$	$\chi^2(x_{\text{SLSJF}})$
56871.5	59421.7	56996.6	57149.9	57837.4
0.0	2550.2	125.1	278.4	965.9

Table 2: Non-linear χ^2 error of the different estimates (first row) and difference to the exact minimum (second row). The χ^2 errors were obtained by running Gauss-Newton on the full non-linear SLAM problem while fixing the landmarks to the respective estimate.

D-SLAM map joining algorithm [Huang *et al.*, 2006] all result in some information loss due to the marginalization of robot poses. In SLSJF and I-SLSJF, the robot start and end poses of the local maps are never marginalized but kept in the global state vector. Thus all the information from local maps is preserved.

If each local map is treated as one integrated observation, then SLSJF has some similarity to iSAM [Kaess *et al.*, 2007]. The role of local maps in SLSJF is also similar to the "star nodes" in the Graphical SLAM [Folkesson and Christensen, 2007]. However, in the Graphical SLAM, the poses are first added in the graph and then "star nodes" are made. While in SLSJF and I-SLSJF, most of the robot poses are marginalized out during the local map building steps. Those robot poses are never present in the global state vector.

The I-SLSJF has some similarity to the Tectonic SAM algorithm [Ni *et al.*, 2007]. Tectonic SAM is also an efficient submap based approach and the state vector reordering and Cholesky factorization are used in solving the least-square problem. The submap fusion in Tectonic SAM uses a divide-and-conquer approach, which is more efficient than the sequential map joining in I-SLSJF when data association is assumed. The major difference between Tectonic SAM and SLSJF is that in Tectonic SAM, all the robot poses involved in building the local maps are kept and the dimension of the global state vector is much higher than that of SLSJF.

Similar to [Tardos *et al.*, 2002][Williams, 2001], there is no requirement on the structure of the environments for I-SLSJF to be applicable. This is different from the efficient treemap SLAM algorithm [Frese, 2006] where the environment has to be "topological suitable". Another difference between I-SLSJF and the treemap SLAM algorithm is that the covariance submatrix recovery and data association have been ignored in the treemap SLAM implementations available to date [Frese, 2006][Frese and Schröder, 2006][Frese, 2007].

6 Conclusion

This paper presented a robust local map joining algorithm – I-SLSJF. By treating each local map as an observation and including robot start/end poses in the global state vector, the map joining problem is formulated as a least squares problem. EIF combined with the linearized least squares approach are used in the map fusion step. As compared with SLSJF, the consistency is improved with the price of some extra computational cost. Both simulation and experimental results show the improved consistency of the proposed map joining algorithm.

The I-SLSJF is still computationally efficient due to the sparseness of the information matrix. In the future, we plan to extend the I-SLSJF algorithm to 3D local map joining to efficiently solve the large-scale 3D SLAM. A more interesting and challenging problem is to investigate the problem of joining non-feature based local maps.

Acknowledgments

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

References

[Bailey, 2002] T. Bailey. Mobile Robot Localization and Mapping in Extensive Outdoor Environment. Ph.D. Thesis. Australian Centre of Field Robotics, University of Sydney. 2002.

[Bailey *et al.*, 2006] T. Bailey, J. Nieto, J. Guivant, M. Stevens and E. Nebot. Consistency of the EKF-SLAM Algorithm. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3562-3568, Beijing, China, October 9-15, 2006.

[Castellanos *et al.*, 2007] J.A. Castellanos, R. Martinez-Cantin, J.D. Tardos and J. Neira. Robocentric map joining: Improving the consistency of EKF-SLAM. *Robotics and Autonomous Systems*, 2007, vol. 55, 21-29.

[Dellaert and Kaess, 2006] F. Dellaert and M. Kaess. Square root SAM: Simultaneous localization and mapping via square root information smoothing. *International Journal of Robotics Research*, 25(12):1181-1203, 2006.

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

[Estrada *et al.*, 2005] C. Estrada, J. Neira and J.D. Tardos, Hierarchical SLAM: real-time accurate mapping of large environments. *IEEE Transactions on Robotics*, 21(4):588-596, 2005.

[Eustice *et al.*, 2006] R. M. Eustice, H. Singh, J. J. Leonard and M. R. Walter. Visually mapping the RMS Titanic: Conservative covariance estimates for SLAM information filters. *International Journal of Robotics Research*, 25(12): 1223-1242, 2006.

[Folkesson and Christensen, 2007] J. Folkesson and H.I. Christensen. Closing the loop with Graphical SLAM. *IEEE Transactions on Robotics*, 23(4):731-741, 2007.

[Frese, 2006] U. Frese. Treemap: An O(log n) algorithm for indoor simultaneous localization and mapping. *Autonomous Robots*, 21(2):103-122, 2006.

[Frese, 2007] U. Frese. Efficient 6-DOF SLAM with Treemap as a generic backend. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4814-4819, Rome, Italy, 10-14 April 2007.

[Frese and Schröder, 2006] U. Frese and L. Schröder. Closing a million-landmarks loop. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5032-5039, Beijing, China, October 9 - 15, 2006.

[Guivant and Nebot, 2001] J. E. Guivant and E. M. Nebot. Optimization of the simultaneous localization and map building (SLAM) algorithm for real time implementation. *IEEE Transactions on Robotics and Automation*, 17:242-257, 2001.

[Huang *et al.*, 2006] S. Huang, Z. Wang, and G. Dissanayake. Mapping large-scale environments using relative position information among landmarks. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2297-2302, Orlando, Florida, USA, 15-19 May 2006.

[Huang *et al.*, 2008a] Shoudong Huang, Zhan Wang and Gamini Dissanayake. Exact state and covariance submatrix recovery for submap based sparse EIF SLAM algorithms. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1868–1873, Pasadena, California, May 19-23, 2008.

[Huang *et al.*, 2008b] Shoudong Huang, Zhan Wang and Gamini Dissanayake. Sparse Local Submap Joining Filter for Building Large-Scale Maps. *IEEE Transactions on Robotics*, 2008 (accepted, to appear). available at: http://services.eng.uts.edu.au/˜sdhuang/publications.htm

[Huang and Dissanayake, 2007] S. Huang and G. Dissanayake, Convergence and consistency analysis for Extended Kalman Filter based SLAM. *IEEE Transactions on Robotics*, 23(5):1036-1049, 2007.

[Kaess *et al.*, 2007] M. Kaess, A. Ranganathan, F. Dellaert. iSAM: Fast incremental Smoothing and Mapping with efficient data association. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1670-1677, Rome, Italy, 10-14 April 2007.

[Neira and Tardos, 2001] J. Neira and J.D. Tardos. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890-897, 2001.

[Ni *et al.*, 2007] K. Ni, D. Steedly, and F. Dellaert. Tectonic SAM: Exact, out-of-core, submap-based SLAM. In *Proceedings of the 2007 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1678-1685, Rome, Italy, 10-14 April 2007.

[Paz *et al.*, 2005] L. M. Paz, P. Pinies, J. Neira, and J. D. Tardos. Global localization in SLAM in Bilinear Time. In *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 655-661, Edmonton, Alberta, Canada, August 2-6, 2005.

[Press *et al.*, 1992] W.H. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery. *Numerical Recipes, Second Edition*. Cambridge University Press, Cambridge, 1992.

[Tardos *et al.*, 2002] J. D. Tardos, J. Neira, P. M. Newman and J. J. Leonard. Robust mapping and localization in indoor environments using sonar data. *International Journal of Robotics Research*, 21(4):311-330, April 2002.

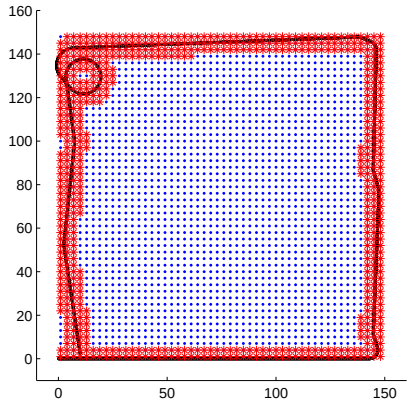
[Thrun *et al.*, 2004] S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani, H. Durrant-Whyte, Simultaneous localization and mapping with sparse extended information filters. *International Journal of Robotics Research*, 23:693-716, 2004.

[Thrun *et al.*, 2005] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2005.

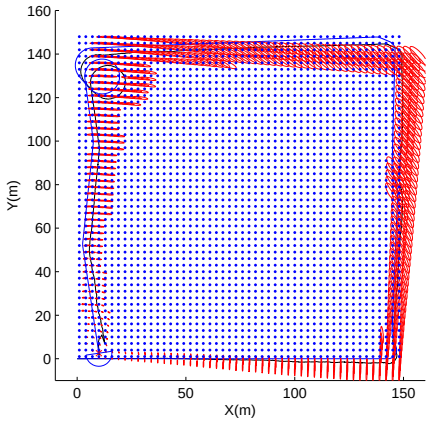
[Walter *et al.*, 2007] M. R. Walter, R. M. Eustice and J. J. Leonard. Exactly sparse Extended Information Filters for feature-based SLAM. *International Journal of Robotics Research*, 26(4):335-359, 2007.

[Wang *et al.*, 2007] Z. Wang, S. Huang and G. Dissanayake. D-SLAM: A decoupled solution to simultaneous localization and mapping. *International Journal of Robotics Research*, 26(2):187-204, 2007.

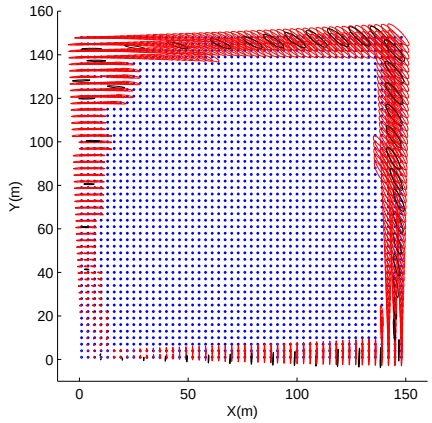
[Williams, 2001] S. B. Williams. Efficient Solutions to Autonomous Mapping and Navigation Problems. PhD thesis, Australian Centre of Field Robotics, University of Sydney, 2001.



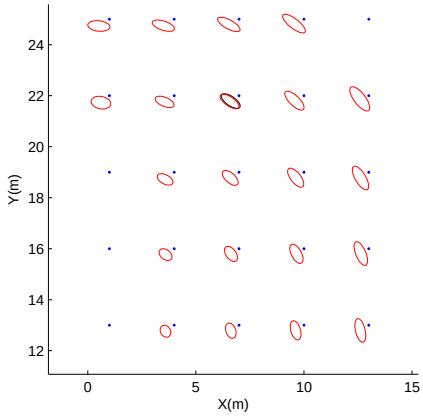
(a) The robot trajectory and the feature observed (red: the features observed, black: the true robot trajectory)



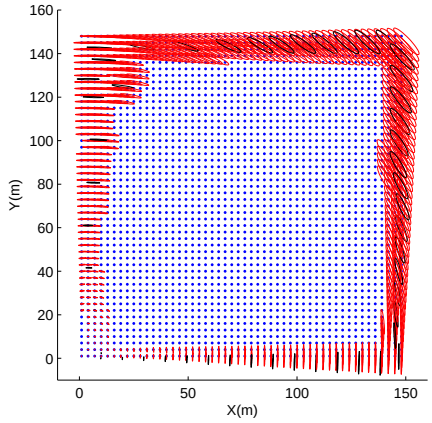
(b) Robot trajectory estimate and map obtained by a single EKF SLAM (red: 2σ covariance ellipses of the features estimate, black: the estimated robot trajectory, blue: the true robot trajectory)



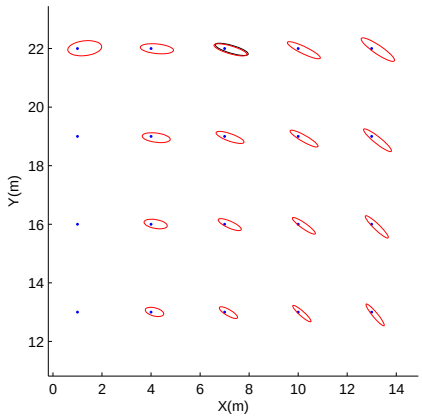
(c) The map obtained by fusing 50 local maps by SLSJF



(d) A close look at the estimate of the some features at the lower-left corner of the map (c): dots are true feature positions, 2σ covariance ellipses are from SLSJF



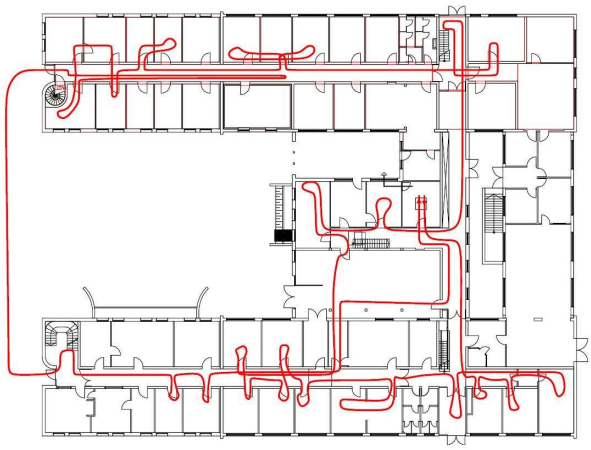
(e) The map obtained by fusing 50 local maps by I-SLSJF



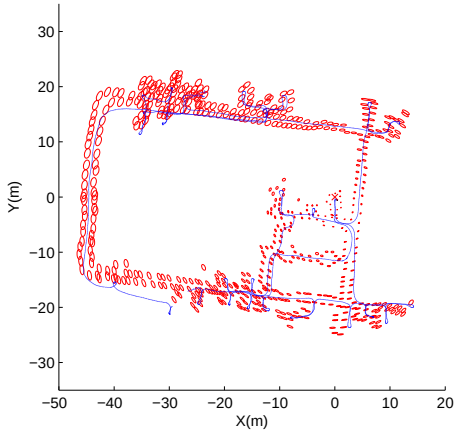
(f) A close look at the estimate of the some features at the lower-left corner of the map (e): dots are true feature positions, 2σ covariance ellipses are from I-SLSJF

Figure 2: Simulation results

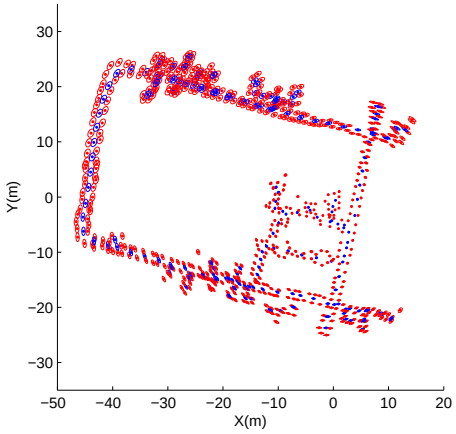
(a) The robot equipped with camera



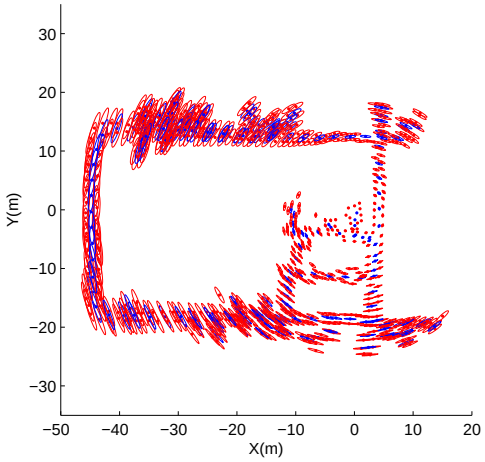
(b) Structure of the environment and the robot trajectory



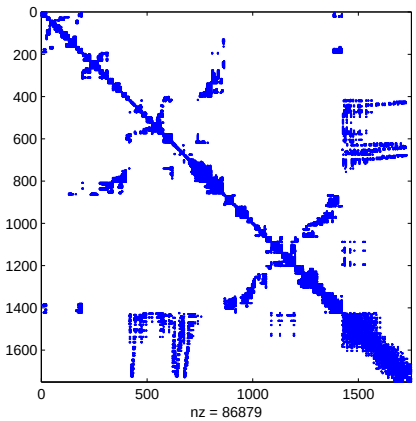
(c) EKF SLAM result



(d) Map obtained by joining 200 local maps using SLSJF



(e) Map obtained by joining 200 local maps using I-SLSJF



(f) The sparse information matrix obtained by I-SLSJF (86879 non-zero elements in a 1750 × 1750 matrix)

Figure 3: The results using DLR-Spatial-Cognition Data Set.