Implementation Issues and Experimental Evaluation of D-SLAM

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. D-SLAM algorithm first described in [1] allows SLAM to be decoupled into solving a non-linear static estimation problem for mapping and a three-dimensional estimation problem for localization. This paper presents a new version of the D-SLAM algorithm that uses an absolute map instead of a relative map as presented in [1]. One of the significant advantages of D-SLAM algorithm is its $O(N)$ computational cost where N is the total number of features (landmarks). The theoretical foundations of D-SLAM together with implementation issues including data association, state recovery, and computational complexity are addressed in detail. Evaluation of the D-SLAM algorithm is provided using both real experimental data and simulations.

Key words: Decoupled SLAM, Extended Information Filter, Sparse Matrix, Computational Complexity

1 Introduction

Simultaneous localization and mapping (SLAM) is the process of building a feature based map of an environment while concurrently generating an estimate for the location of the robot. The SLAM problem has been the subject of extensive research in the past few years, most of which make use of estimation-theoretic techniques (see for example [2], [3], [4], [5], [6], [7] and the references therein).

In traditional SLAM, the state vector contains the location of the robot and all the feature locations. Some convergence properties of the traditional SLAM algorithm using Extended Kalman Filter are proved in [2]. However, traditional SLAM algorithms lead to a heavy computation burden for large scale problems. Many researchers have exploited the special structure of the SLAM algorithm in order to reduce the computational effort required in the SLAM process thereby make large scale SLAM more tractable. For example, in [3], a compressed algorithm is presented to store and maintain all the information gathered in a local area, and then the information is transferred to the rest of the global map. In a recent publication [7], Thrun et al. used the Extended Information Filter to exploit the relative sparseness of the information matrix to reduce the computational effort required in SLAM.

Another way to reduce the computational complexity is to decouple the mapping and localization processes in SLAM. Different groups of researchers have been discussing the possibility of the decoupling. Most of them have made use of the idea of constructing a relative map using the observation information. For example, Newman [4] introduced a relative map in which the map state contains the relative locations among the features. Csorba et al. [8], Deans and Herbert [9], and Martinelli [10] have made use of relative map where the map state only contains distances among the features, which are invariants under shift and rotation. However, all the above approaches have redundant elements in the state vector of the relative map. If no further constraint is applied, it may result in inconsistent map. If constraints are applied, the computation complexity will be increased dramatically. Moreover, how to extract the information about the relative map from the observations and the possible information loss in the decoupling of localization and mapping have not been fully addressed.

In our recent research work [1], a novel decoupled SLAM algorithm, D-SLAM using compact relative maps, is proposed. The state vector for the mapping in D-SLAM is a $2n-3$ dimensional vector containing distances and angles among the features (where n is the total number of features). It is shown that the new formulation retains the significant advantage of being able to improve the location estimates of all the features from one local observation. When Extended Information Filter is applied, D-SLAM results in a sparse information matrix.

This paper provides a D-SLAM algorithm where the state vector for mapping is the absolute locations of the features ($2n$ dimension for n features). The new algorithm is easier to implement than the D-SLAM algorithm using relative map, yet maintains the sparseness of the information matrix and the resulting computational savings. Some discussion on the implementation issues and further evaluation of D-SLAM using experimental data is presented in this paper. The paper is organized as follows. In Section 2, the key idea of D-SLAM and the details of the mapping and localization algorithms are provided. Section 3 addresses some implementation issues in D-SLAM including data association, state recovery and computational complexity. Experimental and simulation results are presented and compared with the results using traditional SLAM in Section 4. Section 5 concludes the paper and addresses future research directions.

2 D-SLAM Algorithm

In traditional SLAM, the state vector contains both the robot location (consisting of the position and orientation of the robot) and the feature locations. In the D-SLAM algorithm proposed below, the state vector for the mapping only contains the absolute locations of the features. The state vector for the localization only contains the robot location. The key step is to recast the measurement vector such that the information about the map contained in the measurements is relatively separated from the information about the robot location. In this section, we first briefly review the recasting, then discuss in detail the procedure of the mapping and localization process in D-SLAM using absolute map.

2.1 New measurements used in D-SLAM

We assume that the robot observes more than one feature at a time. Suppose robot observes m features $f_1, \cdots, f_m$ at a particular time. The original measurements (used in traditional SLAM) are the measured range and bearing of each observed feature:

$$z_{old} = [r_1, \theta_1, \cdots, r_m, \theta_m]^T. \tag{1}$$

It contains Gaussian noise with zero mean and covariance matrix

$$R_{old} = diag[\sigma_{r_1}^2, \sigma_{\theta_1}^2, \cdots, \sigma_{r_m}^2, \sigma_{\theta_m}^2]. \tag{2}$$

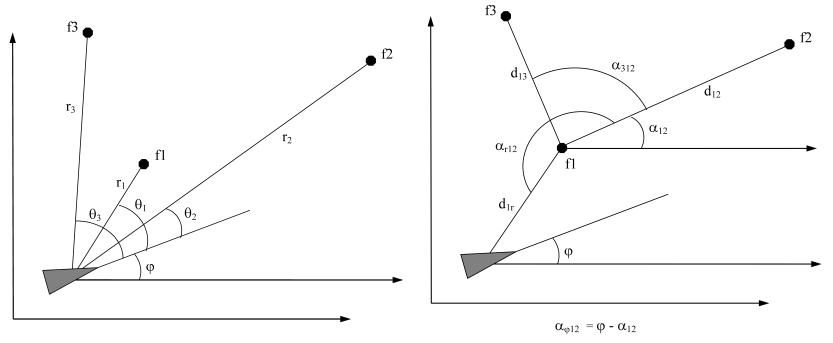
New measurement vector used in D-SLAM is

$$z_{new} = \begin{bmatrix} z_{rob} \\ z_{map} \end{bmatrix} = \begin{bmatrix} \alpha_{r12} \\ d_{1r} \\ \alpha_{\phi 12} \\ --- \\ d_{12} \\ \alpha_{312} \\ d_{13} \\ \vdots \\ \alpha_{m12} \\ d_{1m} \end{bmatrix} = \begin{bmatrix} atan2\left(\frac{-\tilde{y}_1}{-\tilde{x}_1}\right) - atan2\left(\frac{\tilde{y}_2-\tilde{y}_1}{\tilde{x}_2-\tilde{x}_1}\right) \\ \sqrt{(-\tilde{x}_1)^2 + (-\tilde{y}_1)^2} \\ -atan2\left(\frac{\tilde{y}_2-\tilde{y}_1}{\tilde{x}_2-\tilde{x}_1}\right) \\ --- \\ \sqrt{(\tilde{x}_2 - \tilde{x}_1)^2 + (\tilde{y}_2 - \tilde{y}_1)^2} \\ atan2\left(\frac{\tilde{y}_3-\tilde{y}_1}{\tilde{x}_3-\tilde{x}_1}\right) - atan2\left(\frac{\tilde{y}_2-\tilde{y}_1}{\tilde{x}_2-\tilde{x}_1}\right) \\ \sqrt{(\tilde{x}_3 - \tilde{x}_1)^2 + (\tilde{y}_3 - \tilde{y}_1)^2} \\ \vdots \\ atan2\left(\frac{\tilde{y}_m-\tilde{y}_1}{\tilde{x}_m-\tilde{x}_1}\right) - atan2\left(\frac{\tilde{y}_2-\tilde{y}_1}{\tilde{x}_2-\tilde{x}_1}\right) \\ \sqrt{(\tilde{x}_m - \tilde{x}_1)^2 + (\tilde{y}_m - \tilde{y}_1)^2} \end{bmatrix} \tag{3}$$

where

$$\begin{pmatrix} \tilde{x}_i \\ \tilde{y}_i \end{pmatrix} = \begin{pmatrix} r_i \cos\theta_i \\ r_i \sin\theta_i \end{pmatrix}, \quad i = 1, \cdots, m. \tag{4}$$

The physical meaning of the new measurement vector is shown in Figure 1(b) with that of the original measurements shown in Figure 1(a).



(a) Original measurements used in traditional SLAM

(b) New measurements used in D-SLAM

Fig. 1. Measurements used in traditional SLAM and D-SLAM

The noise on z_{rob} and z_{map} are assumed to be Gaussian with zero mean; the covariance matrices R_{rob} and R_{map} can be obtained by (2), (3) and (4) using Jacobian of the functions evaluated at the measurement value r_i, θ_i. This kind of assumption and approximation using linearization have been used in all the Extended Kalman Filter (or Extended Information Filter) related literature.

In the new measurement vector z_{new}, z_{rob} depends on the robot pose and features f_1, f_2 while z_{map} contains information about distances and angles among features which are independent of the coordinate system, namely invariant under shift and rotation. The part z_{map} carries the maximal amount of information of the map that can be extracted from the observations. **In D-SLAM, the key idea is to use only z_{map} in the mapping.**

However, z_{rob} and z_{map} are not independent. Therefore, the estimation process need to be formulated carefully in order that statistically consistent estimates are obtained. In the next two subsections, details of the mapping and localization algorithms in D-SLAM with absolute map are provided.

2.2 Mapping in D-SLAM

State vector: The state vector in mapping contains the locations of the features:

$$X = (X_1, \cdots, X_n)^T = (x_1, y_1, x_2, y_2, \cdots, x_n, y_n)^T. \qquad (5)$$

For convenience, we choose the initial robot coordinate system as the coordinate system, where the origin is the initial robot position and the x-axis is along the initial robot heading.

Since all the features are assumed to be stationary, there is no prediction step and the mapping problem is a non-linear static estimation problem. Extended Information Filter (e.g. [11] [7]) is used to derive the formulas. The relation between estimated state vector $\hat{X}(k)$ and information vector $i(k)$ is

$$i(k) = I(k)\hat{X}(k) \tag{6}$$

where $I(k)$ is the information matrix which is the inverse of the covariance matrix.

Phase I: The robot is stationary at its initial position

In this phase, the robot location is perfectly known. The original measurements (the range r_i and bearing θ_i) will be used to initialize and/or update feature f_i. The details are omitted.

Phase II: The robot is away from its initial position

Measurement model: Suppose the robot observes m features $f_1, \cdots, f_m$ and f_1, f_2 are old features. The model of the new measurement for mapping is

$$z_{map} = [d_{12}, \alpha_{312}, d_{13}, \cdots, \alpha_{m12}, d_{1m}]^T = H_{map}(X) + w_{map} \tag{7}$$

where

$$H_{map}(X) = \begin{pmatrix} \sqrt{(x_2-x_1)^2+(y_2-y_1)^2} \\ atan2\left(\frac{y_3-y_1}{x_3-x_1}\right) - atan2\left(\frac{y_2-y_1}{x_2-x_1}\right) \\ \sqrt{(x_3-x_1)^2+(y_3-y_1)^2} \\ \cdots \\ atan2\left(\frac{y_m-y_1}{x_m-x_1}\right) - atan2\left(\frac{y_2-y_1}{x_2-x_1}\right) \\ \sqrt{(x_m-x_1)^2+(y_m-y_1)^2} \end{pmatrix} \tag{8}$$

and w_{map} is the new measurement noise whose covariance matrix R_{map} can be computed by (2), (3) and (4).

Initialize new features: Suppose the current estimation of the location of features f_1, f_2 are $\hat{X}_1 = (\hat{x}_1, \hat{y}_1)$ and $\hat{X}_2 = (\hat{x}_2, \hat{y}_2)$. They can be used together with d_{1i}, α_{i12} in z_{map} to initialize the location of new feature f_i as follows:

$$\begin{aligned} \alpha_{12} &= atan2(\tfrac{\hat{y}_2-\hat{y}_1}{\hat{x}_2-\hat{x}_1}) \\ \hat{x}_i &= \hat{x}_1 + d_{1i}\cos(\alpha_{12}+\alpha_{i12}) \\ \hat{y}_i &= \hat{y}_1 + d_{1i}\sin(\alpha_{12}+\alpha_{i12}). \end{aligned} \tag{9}$$

Update (old and new) features: When new features are observed, the dimension of the information vector and information matrix will be increased by adding zeros for the new features. We still denote the new information vector as $i(k)$, the new information matrix as $I(k)$, and the new state estimation as $\hat{X}(k)$.

The formulas for the update of the information vector and the information matrix using the measurement z_{map} are as follows:

$$\begin{aligned} I(k+1) &= I(k) + \nabla H_{map}^T R_{map}^{-1} \nabla H_{map} \\ i(k+1) &= i(k) + \nabla H_{map}^T R_{map}^{-1} [z_{map}(k+1) - H_{map}(\hat{X}(k)) + \nabla H_{map} \hat{X}(k)] \end{aligned} \tag{10}$$

where ∇H_{map} is the Jacobian of the function H_{map} evaluated on the current state estimation $\hat{X}(k)$.

2.3 Localization in D-SLAM

State vector: The state vector used in localization is the three dimensional robot location:

$$X_r = (x_r, y_r, \phi_r)^T. \tag{11}$$

Localization is only needed when the robot is away from its initial position. We can obtain two estimates of the robot location. The first estimate is from the process model plus the priori knowledge of the robot location. The details are the same as those in the traditional SLAM and are omitted here. The second estimate is from the measurements.

Measurement model: Suppose the robot observes m features $f_1, f_2, \cdots, f_m$, among which $f_1, \cdots, f_{m_1}, m_1 \leq m$ are features that have been previously seen. Part of the original measurement vector z_{old} that involves these old features is used for localization

$$z_{loc} = H_{loc}(X_1, \cdots, X_{m_1}, X_r) + w_{loc}. \tag{12}$$

Estimate from measurement: An estimate of X_r can be obtained by z_{loc} and the current estimates of $f_1, \cdots, f_{m_1}$ and their corresponding covariance matrix (a submatrix of the whole covariance matrix).

Combine two estimates using Covariance Intersection: Close examination of the estimation process reveals that the two estimates generated above are not independent. In some cases, for example in an indoor robot equipped with a laser sensor, the estimate from measurement itself may provide a sufficiently accurate robot location. In our simulation, we combine the two estimates using Covariance Intersection (CI) [6], which facilitates combining two correlated pieces of information when the extent of correlation itself is unknown.

As in the case of D-SLAM using compact relative map [1], although z_{map} in (7) and z_{loc} in (12) are not independent, the observation information is not reused. This is because the information about the robot location obtained from the localization process will never be used in the mapping process.

3 Implementation issues

3.1 Data association

Data association refers to the process of associating the observations to the corresponding features. As in the traditional SLAM, many data association algorithms can be applied in the proposed D-SLAM algorithm. Generally speaking, batch data association algorithms (e.g [12]) are more robust than the standard maximum likelihood approach but the computational cost is higher.

In our simulation and experiment, we follow the standard maximum likelihood approach described in [2]. Due to erroneous feature detections caused by moving objects or measurement noise, two feature lists are maintained. One list stores features that are confirmed to be valid, and the other stores potential features yet to be validated. Mahalanobis distance between the newly observed features and the features in the two lists are computed in order to decide about the association.

Note that the recovery of feature location estimation and part of the associated covariance matrix is needed for the data association.

3.2 Recovery of the feature locations in D-SLAM

Recovery of the feature location estimation is not only needed in data association, but also needed in the map update and robot localization. When the number of features is small, the recovery can be simply obtained by (6) using the inverse of the information matrix. However, when the number of features is large, the computational cost of the inversion of the information matrix will be high. So it is crucial to find more efficient ways of the recovery.

We first consider which part of the map states is needed in the D-SLAM algorithm. (a) For mapping: as we can see in (10), by using the information vector, it is not necessary to compute the inverse of the information matrix $I(k+1)$ in the update step. However, the current state estimation of the features involved in the current observation is still needed to compute ∇H_{map} and $\nabla H_{map}\hat{X}(k)$. (b) For localization: in order to obtain the *Estimate from measurement*, the estimation of the old features $f_1, f_2, \cdots, f_{m_1}$ and their corresponding covariance matrix are needed. (c) For data association: only the estimates and the covariance matrices of features in the vicinity of robot (the vicinity here is defined in terms of the range of the sensor used for making observations) are needed.

In other words, we only need the estimation of the features within the sensor range of the current robot location and its corresponding covariance matrix. Since the Jacobian ∇H_{map} in (10) is sparse and there is no prediction step in the mapping process, the information matrix $I(k+1)$ is an exactly sparse matrix with the number of non-zero elements related to the sensor range. In fact, links (by link, we mean the non-zero off-diagonal element in the information matrix) between two features are established only if they are both involved in the same measurements at a particular time. The result is that links exist only between the features that are in the vicinity of each other. This exact sparseness makes it possible to reduce the computational cost of the map recovery significantly.

3.3 Computational complexity

Let N be the number of features in the map. Two dimensional D-SLAM requires the storage of the information vector with dimension $2N$, the recovered state vector with dimension $2N$, the sparse information matrix with non-zero elements $O(N)$, and the submatrix of the covariance matrix corresponding to the currently observed features $O(1)$. The storage requirements are therefore of $O(N)$.

Updating the information matrix and the information vector requires the Jacobian ∇H_{map} as well as $\nabla H_{map}\hat{X}(k)$. Thus it is necessary to recover the current estimate of map state vector $\hat{X}(k)$. This can be done by solving a set of sparse linear equations, using few iterations requiring $O(N)$ operations as a good initial guess of $\hat{X}(k)$ is always available.

Once the Jacobian is computed, updating the information matrix and the information vector requires constant time as the Jacobian is always sparse and as a prediction step is not necessary.

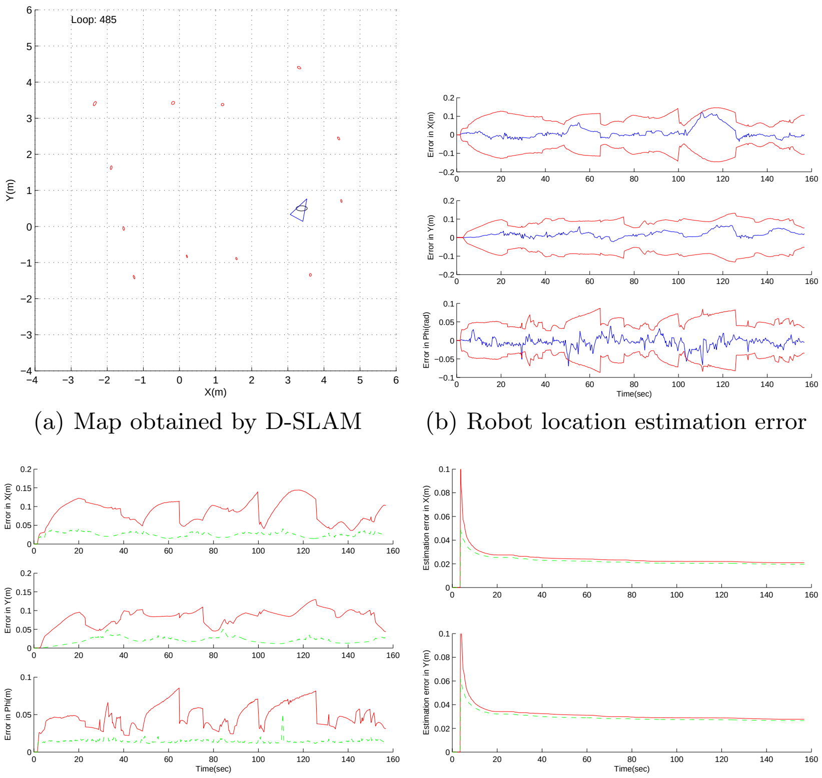
For data association, locations as well as the uncertainty of the features in the vicinity of the robot are required. The vicinity here is defined in terms of the range of the sensor. This requires $O(N)$ operations to evaluate. The desired columns of the covariance matrix associated with these features can be obtained by solving a constant number of sparse linear equations with the aid of a good initial guess, which also requires $O(N)$ operations. Once the locations of the observed features and the corresponding covariance matrix are available, localization can be performed in constant time. Overall cost of D-SLAM is, therefore, $O(N)$.

4 Evaluation of D-SLAM

4.1 Experimental Evaluation with a Pioneer robot in an office environment

The Pioneer 2 DX robot in our lab is used for the implementation. It is equipped with a laser range finder with a field of view of 180 degrees and an angular resolution of 0.5 degree to produce the relative range and bearing measurements between the robot and the features. We run the pioneer in our laboratory where we put twelve laser reflector strips in a $8 \times 8m^2$ area. The standard software, Player, is used to collect the control and sensor data from the robot. Then we run the D-SLAM algorithm in Matlab with the collected data.

In order to evaluate the robot and feature location estimation, we need the true value of the states. Here we use the traditional SLAM estimation as the truth. Figure 2(a) is the map obtained from D-SLAM. Figure 2(b) is the robot location estimation from D-SLAM with respect to traditional SLAM estimation. Figures 2(c) and 2(d) show the 2σ bound obtained from D-SLAM and traditional SLAM for the estimation of robot location and feature 9.



(a) Map obtained by D-SLAM

(b) Robot location estimation error

(c) 2σ bounds of robot location estimation (solid line is from D-SLAM; dashed line is from traditional SLAM)

(d) 2σ bounds of feature 9 estimation (solid line is from D-SLAM; dashed line is from traditional SLAM)

Fig. 2. D-SLAM implementation: map and estimation error

Figure 2(b) shows that the D-SLAM estimation is consistent. The map (Figure 2(a)) is almost as good as that of the traditional SLAM in this small area, as can be seen more clearly in feature 9 estimation in figure 2(d). In this figure, the 2σ bound from D-SLAM is very close to that from traditional SLAM. The slight difference comes from the fact that no information about robot location is fused into the map.

It can be seen from figures 2(b) and 2(c) that the localization result using CI is conservative. The reason is that CI applies conservative combination of the two estimates under the situation of not knowing their correlation [6]. The risk in it lies in the data association. The maximum likelihood method used in data association may fail when the robot estimation uncertainty is large. In D-SLAM, this failure may occur more frequently compared with traditional SLAM algorithms.

4.2 Evaluation of D-SLAM in simulation with a large number of features

In simulation, we ran D-SLAM algorithm in a much larger area, so as to further verify its convergence and illustrate its properties.

The environment used is a 40 meter square region. We put 196 features arranged in uniformly spaced rows and columns. The interval between two features is 3 meters. The robot starts from the left bottom corner and follows a random trajectory. Robot speed is 20cm/s and turn rate is 0.1rad/s. A sensor with a field of view of 180 degrees and a range of 5 meters is simulated to generate relative range and bearing measurements between the robot and the features.

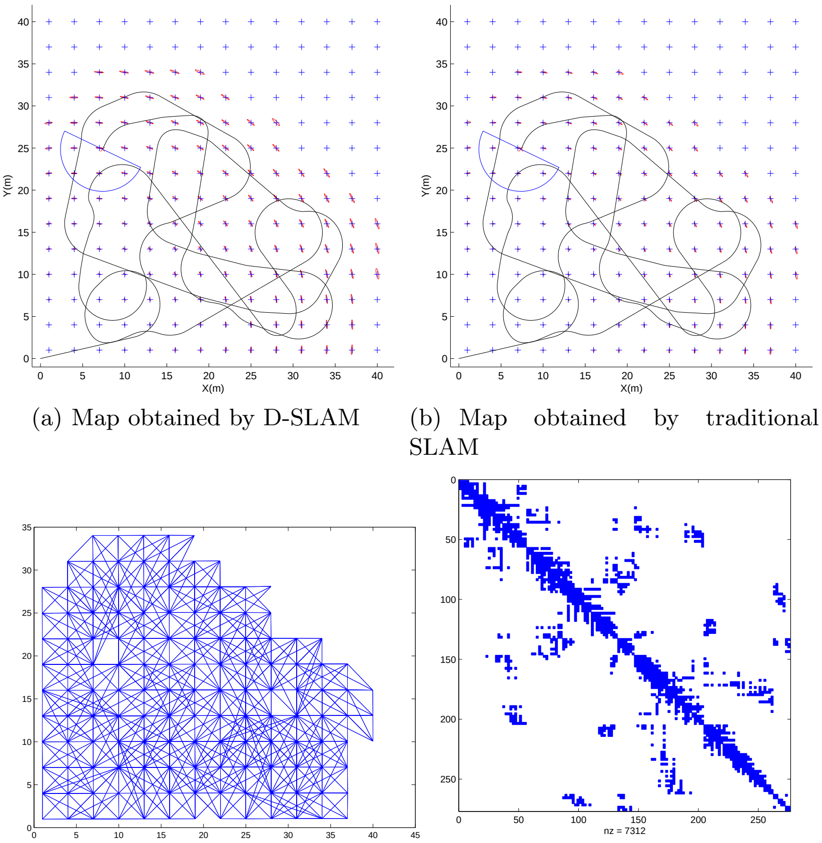
Figure 3(a) and 3(b) show the maps obtained from D-SLAM and traditional SLAM. It can be seen that the uncertainty of the feature location estimates are more conservative in D-SLAM, compared with the traditional SLAM estimator. This information loss is expected.

Figure 3(c) shows the links among the features in the information matrix. This figure demonstrates more clearly that links only exist among features within sensor range. Figure 3(d) shows the non-zero elements in the information matrix obtained by the D-SLAM algorithm. Non-zero off-diagonal elements are caused by closing loops.

5 Conclusions

In this paper, we proposed a new SLAM algorithm: D-SLAM using absolute map. We addressed some key implementation issues and provided experimental verification for D-SLAM. The convergence of D-SLAM is verified by both real experimental data and simulations. Although the robot location is not incorporated in the state vector used in mapping, correlations among the features are still preserved in the information matrix. Therefore, the estimation uncertainty of the feature locations that are far away from the initial location of the robot is significantly reduced as the “loop is closed”. A significant advantage of D-SLAM is that the information matrix associated with the mapping is exactly sparse resulting in a significant reduction in computational complexity.

Besides the $O(N)$ computational cost, D-SLAM also has the following potential advantages: (1) since the mapping problem is treated as a static estimation problem, the multi-robot SLAM problem can be a simple extension, provided data association issues can be resolved; (2) some recent results have also shown that the large error in the robot orientation introduces inconsistency in traditional SLAM [14] [15]. D-SLAM does not have the robot location in the state vector used for mapping thus may be more robust than traditional SLAM.



(a) Map obtained by D-SLAM (b) Map obtained by traditional SLAM

(c) Links in information matrix (d) Sparse information matrix obtained by D-SLAM

Fig. 3. D-SLAM simulations: Maps and Sparse Information Matrices

D-SLAM, however, results in some information loss because not all the information from the process model and observations is used for the mapping and localization. Preliminary analysis suggests that the extent of the information loss is related to the ratio between process noise and observation noise. It is seen from the experimental results that in many practical scenarios, with the availability of high rate scanners such as the SICK laser, the information loss is not a significant drawback.

Our ongoing research includes the detailed analysis of the information loss in D-SLAM, the verification using data from large outdoor environments, and multi-robot D-SLAM. Active D-SLAM problem where the robot trajectory is actively chosen on-line is our future research topic.

References

1. Wang Z, Huang S, Dissanayake G (2005) "Decoupling Localization and Mapping in SLAM Using Compact Relative Maps", in *Proceedings of IROS 2005*, Edmonton, Canada
2. Dissanayake G, Newman P, Clark S, Durrant-Whyte H, and Csorba M (2001) "A solution to the simultaneous localization and map building (SLAM) problem", *IEEE Trans. on Robotics and Automation* 17:229-241
3. Guivant JE, Nebot EM (2001) "Optimization of the simultaneous localization and map building (SLAM) algorithm for real time implementation", *IEEE Trans. on Robotics and Automation* 17:242-257
4. Newman P (2000) "On the Structure and Solution of the Simultaneous Localization and Map Building Problem", PhD thesis, Australian Centre of Field Robotics, University of Sydney, Sydney
5. Castellanos JA, Neira J, Tardos JD (2001) "Multisensor fusion for simultaneous localization and map building", *IEEE Trans. on Robotics and Automation* 17:908-914
6. Julier SJ, Uhlmann JK (2001) "Simultaneous localization and map building using split covariance intersection", in *Proceedings of IROS 2001*
7. Thrun S, Liu Y, Koller D, Ng AY, Ghahramani Z, Durrant-Whyte H (2004) "Simultaneous Localization and Mapping with Sparse Extended Information Filters", *International J. of Robotics Research* 23:693-716
8. Csorba M, Uhlmann JK, Durrant-Whyte H (1997) "A suboptimal algorithm for automatic map building", in *Proceedings of 1997 American Control Conference.* pp 537-541, USA
9. Deans MC, Hebert M (2000) "Invariant filtering for simultaneous localization and map building", in *Proceedings IEEE International Conference on Robotics and Automation.* pp 1042-1047
10. Martinelli A, Tomatics N, Siegwart R (2004) "Open challenges in SLAM: An optimal solution based on shift and rotation invariants", in *Proceedings IEEE International Conference on Robotics and Automation.* pp 1327-1332
11. Maybeck P (1979) "Stochastic Models, Estimation, and Control", Academic, New York
12. Bailey T (2002) "Mobile Robot Localization and Mapping in Extensive Outdoor Environment", PhD thesis, Australian Centre of Field Robotics, University of Sydney, Sydney
13. Pissanetzky S (1984) "Sparse Matrix Technology". Academic, London
14. Frese U (2005), "A Discussion of Simultaneous Localization and Mapping", *Autonomous Robots* (to appear). Available online http://www.informatik.uni-bremen.de/˜ufrese
15. Castellanos JA, Neira J, Tardos JD (2004) "Limits to the consistency of EKF-based SLAM", *5th IFAC Symp. on Intelligent Autonomous Vehicles, IAV'04*, Lisbon, Portugal