2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Linear SLAM: A Linear Solution to the Feature-based and Pose Graph SLAM based on Submap Joining

Liang Zhao, Shoudong Huang and Gamini Dissanayake

Abstract— **This paper presents a strategy for large-scale SLAM through solving a sequence of linear least squares problems. The algorithm is based on submap joining where submaps are built using any existing SLAM technique. It is demonstrated that if submaps coordinate frames are judiciously selected, the least squares objective function for joining two submaps becomes a quadratic function of the state vector. Therefore, a linear solution to large-scale SLAM that requires joining a number of local submaps either sequentially or in a more efficient Divide and Conquer manner, can be obtained. The proposed Linear SLAM technique is applicable to both feature-based and pose graph SLAM, in two and three dimensions, and does not require any assumption on the character of the covariance matrices or an initial guess of the state vector. Although this algorithm is an approximation to the optimal full nonlinear least squares SLAM, simulations and experiments using publicly available datasets in 2D and 3D show that Linear SLAM produces results that are very close to the best solutions that can be obtained using full nonlinear optimization started from an accurate initial value. The C/C++ and MATLAB source codes for the proposed algorithm are available on OpenSLAM.**

I. INTRODUCTION

Recently, nonlinear optimization techniques have become popular for SLAM. Following the seminal work [1], many efficient SLAM solutions have been obtained [2]–[6] by exploiting the sparseness of the information matrix and the different approaches for solving the sparse linear equations. However, since SLAM is formulated as a high dimensional nonlinear optimization problem, local minima is an important issue. Although many SLAM algorithms appear to work well for most of the practical datasets (e.g. [7]–[9]), there is no guarantee that the algorithm can converge to the global optimum and having an accurate initial value is very critical, especially for large-scale SLAM problems.

Local submap joining has shown to be an efficient way to build large-scale maps [10]–[16]. The idea of many map joining algorithms such as Sparse Local Submap Joining Filter (SLSJF) [13] is to treat each local map as an integrated observation and use it in the map joining step. However, the traditional map joining problems are still nonlinear estimation or nonlinear optimization problems solved by EKF/EIF [10]–[13] or nonlinear least squares [14]–[17].

This paper provides a new map joining algorithm which only requires solving linear least squares and performing coordinate transformations. The method can be applied to both feature-based SLAM and pose graph SLAM, 2D and 3D. There is no assumption on the covariance matrices of the local maps.

The paper is organized as follows. Section II explains the process of using linear least squares to solve the problem of joining two feature-based maps. Section III explains how to use the linear method in the sequential and Divide and Conquer local submap joining. Section IV provides the linear algorithm for joining two pose graphs. In Section V, some simulation and experimental results using publicly available datasets are given to demonstrate the effectiveness of the proposed algorithm. Section VI discusses the related work and some insights on the proposed algorithm. Finally, Section VII concludes the paper.

II. LINEAR SOLUTION TO JOINING TWO FEATURE-BASED MAPS

A. Traditional Way of Joining Two Maps

Joining two maps is a basic step in traditional map joining (such as SLSJF [13]). It is shown in Fig. 1(a).

Map 1 is built in the coordinate frame defined by its start pose $\mathbf{P}_0$, it contains a number of features and the robot end pose $\mathbf{P}_1$. Map 2 is built in the coordinate frame defined by its start pose $\mathbf{P}_1$, it contains a number of features and the robot end pose $\mathbf{P}_2$.[1]

The end pose of Map 1 is the same as the start pose of Map 2 (both are $\mathbf{P}_1$). The global map coordinate frame is defined by $\mathbf{P}_0$, the same as Map 1. Joining these two maps to get Map 12 can be formulated as a nonlinear optimization problem [17].

B. The New Way of Joining Two Maps

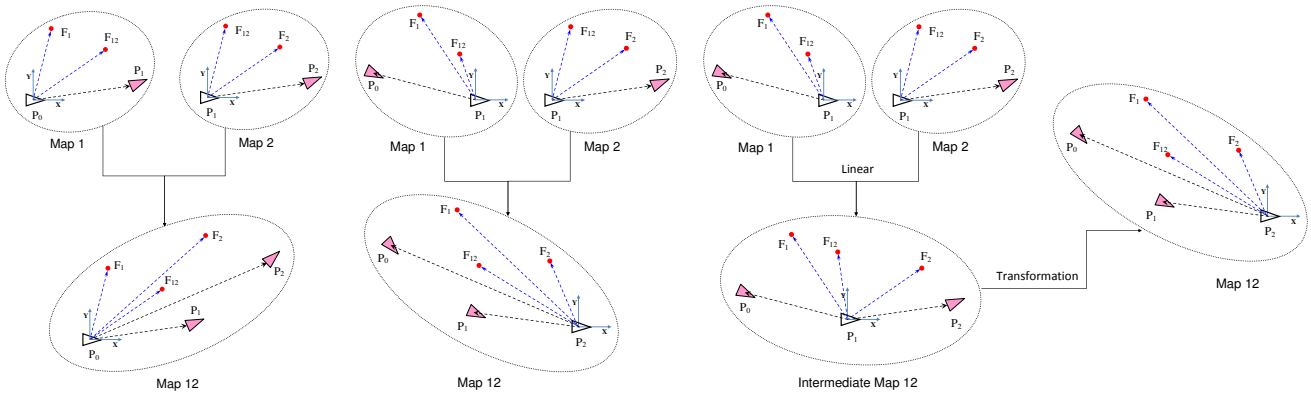
In this paper, we propose to build and join two maps differently as shown in Fig. 1(b).

Different from traditional map joining, now Map 1 is built in the coordinate frame defined by its end pose $\mathbf{P}_1$, it contains a number of features and the robot start pose $\mathbf{P}_0$ as its map estimates. Map 2 is still built in the coordinate frame defined by its start pose $\mathbf{P}_1$, it contains a number of features and the robot end pose $\mathbf{P}_2$.

Different from traditional map joining, now the coordinate frame of Map 12 is defined by $\mathbf{P}_2$, the robot end pose of Map 2.

L. Zhao, S. Huang and G. Dissanayake are with the Centre for Autonomous Systems, Faculty of Engineering and Information Technology, University of Technology, Sydney, NSW 2007, Australia. {Liang.Zhao-1, Shoudong.Huang, Gamini.Dissanayake}@uts.edu.au

[1]Map 1 and Map 2 can either be small local maps built using the local odometry and observation information, or be larger maps as the result of combining a number of small local maps. There can be some other robot poses in Map 1 and Map 2 (such as a map built from least squares SLAM without marginalizing any poses), they are not shown in the figure for simplification.



(a) Traditional way (b) Proposed new way (c) Linear way to build Map 12

Fig. 1. The traditional way and the proposed new way of building and joining two maps.

In the following, we will show that although the joining of Map 1 and Map 2 to get Map 12 in Fig. 1(b) is still a nonlinear optimization problem, it is equivalent to a linear least squares optimization problem plus a (nonlinear) coordinate transformation.

C. Nonlinear Method of Joining M^{L_1} and M^{L_2} to Build $M^{G_{12}}$

Suppose Map 1 and Map 2 in Fig. 1(b) are denoted by M^{L_1} and M^{L_2} and are given by

$$M^{L_1} = (\hat{\mathbf{X}}^{L_1},\ I^{L_1}), \quad M^{L_2} = (\hat{\mathbf{X}}^{L_2},\ I^{L_2}) \tag{1}$$

where $\hat{\mathbf{X}}^{L_1}$ and $\hat{\mathbf{X}}^{L_2}$ are the estimates of the state vectors $\mathbf{X}^{L_1}$ and $\mathbf{X}^{L_2}$ of each map, and I^{L_1} and I^{L_2} are the associated information matrices.

The state vectors $\mathbf{X}^{L_1}$ and $\mathbf{X}^{L_2}$ are defined as [2]

$$\begin{aligned}\mathbf{X}^{L_1} &= [\mathbf{P}_0^{L_1},\ \mathbf{X}_{F_1}^{L_1},\ \mathbf{X}_{F_{12}}^{L_1}]\\ \mathbf{X}^{L_2} &= [\mathbf{P}_2^{L_2},\ \mathbf{X}_{F_2}^{L_2},\ \mathbf{X}_{F_{12}}^{L_2}].\end{aligned} \tag{2}$$

Note that M^{L_1} is in the coordinate frame of $\mathbf{P}_1$, and M^{L_2} is also in the coordinate frame of $\mathbf{P}_1$. Here $\mathbf{P}_1$ is the end pose of M^{L_1} (also the start pose of M^{L_2}). It defines the coordinate frame of both M^{L_1} and M^{L_2} and thus is not included in the state vectors.

In the state vectors $\mathbf{X}^{L_1}$ and $\mathbf{X}^{L_2}$ in (2), $\mathbf{X}_{F_1}^{L_1}$ and $\mathbf{X}_{F_2}^{L_2}$ represent the features only appear in M^{L_1} or M^{L_2}, respectively, while $\mathbf{X}_{F_{12}}^{L_1}$ and $\mathbf{X}_{F_{12}}^{L_2}$ represent the common features appear in both the two maps.

In the state vector $\mathbf{X}^{L_1}$ in (2), the start pose $\mathbf{P}_0$ of M^{L_1} is presented by

$$\mathbf{P}_0^{L_1} = [\mathbf{t}_0^{L_1},\ \mathbf{r}_0^{L_1}] \tag{3}$$

where $\mathbf{t}_0^{L_1}$ is the translation vector, and $\mathbf{r}_0^{L_1}$ is/are the rotation angle/angles of pose $\mathbf{P}_0^{L_1}$. Similarly, in the state vector $\mathbf{X}^{L_2}$ in (2), the end pose $\mathbf{P}_2$ of M^{L_2} is presented by

$$\mathbf{P}_2^{L_2} = [\mathbf{t}_2^{L_2},\ \mathbf{r}_2^{L_2}]. \tag{4}$$

[2]To simplify the notations, the 'transpose's in the state vectors are sometimes omitted in this paper. For example, $\mathbf{X}^{L_1}, \mathbf{P}_0^{L_1}, \mathbf{X}_{F_1}^{L_1}, \mathbf{X}_{F_{12}}^{L_1}$ are all column vectors and the rigorous notation should be $\mathbf{X}^{L_1} = [(\mathbf{P}_0^{L_1})^T,\ (\mathbf{X}_{F_1}^{L_1})^T,\ (\mathbf{X}_{F_{12}}^{L_1})^T]^T$.

Now the state estimates $\hat{\mathbf{X}}^{L_1}$ and $\hat{\mathbf{X}}^{L_2}$ can be written clearly as

$$\begin{aligned}\hat{\mathbf{X}}^{L_1} &= [\hat{\mathbf{t}}_0^{L_1},\ \hat{\mathbf{r}}_0^{L_1},\ \hat{\mathbf{X}}_{F_1}^{L_1},\ \hat{\mathbf{X}}_{F_{12}}^{L_1}]\\ \hat{\mathbf{X}}^{L_2} &= [\hat{\mathbf{t}}_2^{L_2},\ \hat{\mathbf{r}}_2^{L_2},\ \hat{\mathbf{X}}_{F_2}^{L_2},\ \hat{\mathbf{X}}_{F_{12}}^{L_2}].\end{aligned} \tag{5}$$

Our goal is to join M^{L_1} and M^{L_2} to get the map $M^{G_{12}}$, where $M^{G_{12}}$ is in the coordinate frame of $\mathbf{P}_2$.

The state vector of $M^{G_{12}}$ containing pose $\mathbf{P}_0$, pose $\mathbf{P}_1$ and all the features is defined as

$$\begin{aligned}\mathbf{X}^{G_{12}} &= [\mathbf{P}_0^{G_{12}},\ \mathbf{P}_1^{G_{12}},\ \mathbf{X}_{F_1}^{G_{12}},\ \mathbf{X}_{F_2}^{G_{12}},\ \mathbf{X}_{F_{12}}^{G_{12}}]\\ &= [\mathbf{t}_0^{G_{12}},\ \mathbf{r}_0^{G_{12}},\ \mathbf{t}_1^{G_{12}},\ \mathbf{r}_1^{G_{12}},\ \mathbf{X}_{F_1}^{G_{12}},\ \mathbf{X}_{F_2}^{G_{12}},\ \mathbf{X}_{F_{12}}^{G_{12}}]\end{aligned} \tag{6}$$

where all the variables in the state vector $\mathbf{X}^{G_{12}}$ are in the coordinate frame of $\mathbf{P}_2$.

Similar to [17], the map joining problem is to minimize the following objective function

$$\begin{aligned}f(\mathbf{X}^{G_{12}}) &= \|e_1\|^2_{I^{L_1}} + \|e_2\|^2_{I^{L_2}}\\ &= \left\| \begin{array}{c} R_1\,(\mathbf{t}_0^{G_{12}} - \mathbf{t}_1^{G_{12}}) - \hat{\mathbf{t}}_0^{L_1} \\ r^{-1}(R_0 R_1^T) - \hat{\mathbf{r}}_0^{L_1} \\ R_1\,(\mathbf{X}_{F_1}^{G_{12}} - \mathbf{t}_1^{G_{12}}) - \hat{\mathbf{X}}_{F_1}^{L_1} \\ R_1\,(\mathbf{X}_{F_{12}}^{G_{12}} - \mathbf{t}_1^{G_{12}}) - \hat{\mathbf{X}}_{F_{12}}^{L_1} \end{array} \right\|^2_{I^{L_1}}\\ &+ \left\| \begin{array}{c} -R_1\,\mathbf{t}_1^{G_{12}} - \hat{\mathbf{t}}_2^{L_2} \\ r^{-1}(R_1^T) - \hat{\mathbf{r}}_2^{L_2} \\ R_1\,(\mathbf{X}_{F_2}^{G_{12}} - \mathbf{t}_1^{G_{12}}) - \hat{\mathbf{X}}_{F_2}^{L_2} \\ R_1\,(\mathbf{X}_{F_{12}}^{G_{12}} - \mathbf{t}_1^{G_{12}}) - \hat{\mathbf{X}}_{F_{12}}^{L_2} \end{array} \right\|^2_{I^{L_2}}\end{aligned} \tag{7}$$

where $\|e_i\|^2_{I^{L_i}} = e_i^T I^{L_i} e_i$ $(i = 1, 2)$ denotes the weighted norm of vector e_i with the given information matrix I^{L_i}.

In (7), $R_0 = r(\mathbf{r}_0^{G_{12}})$ and $R_1 = r(\mathbf{r}_1^{G_{12}})$ are the rotation matrices of pose $\mathbf{P}_0^{G_{12}}$ and $\mathbf{P}_1^{G_{12}}$ in the global state vector $\mathbf{X}^{G_{12}}$, respectively. And $r(\cdot)$ and $r^{-1}(\cdot)$ are the angle-to-matrix and matrix-to-angle functions. For 2D scenarios, $r^{-1}(R_0 R_1^T) = \mathbf{r}_0^{G_{12}} - \mathbf{r}_1^{G_{12}}$ and $r^{-1}(R_1^T) = -\mathbf{r}_1^{G_{12}}$.

The problem of minimizing (7) is a nonlinear least squares problem. Solving it we can obtain the estimate of Map 12

together with its information matrix as

$$M^{G_{12}} = (\hat{\mathbf{X}}^{G_{12}},\ I^{G_{12}}). \tag{8}$$

D. Linear Method to Build $M^{G_{12}}$

If we define new variables as follows (they are actually the 7 distinct (nonlinear) functions in (7))

$$\begin{aligned}
\bar{\mathbf{t}}_0^{G_{12}} &= R_1\,(\mathbf{t}_0^{G_{12}} - \mathbf{t}_1^{G_{12}}) \\
\bar{\mathbf{r}}_0^{G_{12}} &= r^{-1}(R_0 R_1^T) \\
\bar{\mathbf{t}}_2^{G_{12}} &= -R_1\,\mathbf{t}_1^{G_{12}} \\
\bar{\mathbf{r}}_2^{G_{12}} &= r^{-1}(R_1^T) \\
\bar{\mathbf{X}}_{F_1}^{G_{12}} &= R_1\,(\mathbf{X}_{F_1}^{G_{12}} - \mathbf{t}_1^{G_{12}}) \\
\bar{\mathbf{X}}_{F_2}^{G_{12}} &= R_1\,(\mathbf{X}_{F_2}^{G_{12}} - \mathbf{t}_1^{G_{12}}) \\
\bar{\mathbf{X}}_{F_{12}}^{G_{12}} &= R_1\,(\mathbf{X}_{F_{12}}^{G_{12}} - \mathbf{t}_1^{G_{12}})
\end{aligned} \tag{9}$$

and define a new state vector as

$$\begin{aligned}
\bar{\mathbf{X}}^{G_{12}} &= [\bar{\mathbf{t}}_0^{G_{12}},\ \bar{\mathbf{r}}_0^{G_{12}},\ \bar{\mathbf{t}}_2^{G_{12}},\ \bar{\mathbf{r}}_2^{G_{12}},\ \bar{\mathbf{X}}_{F_1}^{G_{12}},\ \bar{\mathbf{X}}_{F_2}^{G_{12}},\ \bar{\mathbf{X}}_{F_{12}}^{G_{12}}] \\
&= g(\mathbf{X}^{G_{12}})
\end{aligned} \tag{10}$$

then the nonlinear least squares problem to minimize the objective function (7) becomes a linear least squares problem to minimize the following objective function

$$\bar{f}(\bar{\mathbf{X}}^{G_{12}}) = \left\| \begin{bmatrix} \bar{\mathbf{t}}_0^{G_{12}} - \hat{\mathbf{t}}_0^{L_1} \\ \bar{\mathbf{r}}_0^{G_{12}} - \hat{\mathbf{r}}_0^{L_1} \\ \bar{\mathbf{X}}_{F_1}^{G_{12}} - \hat{\mathbf{X}}_{F_1}^{L_1} \\ \bar{\mathbf{X}}_{F_{12}}^{G_{12}} - \hat{\mathbf{X}}_{F_{12}}^{L_1} \end{bmatrix} \right\|_{I^{L_1}}^2 + \left\| \begin{bmatrix} \bar{\mathbf{t}}_2^{G_{12}} - \hat{\mathbf{t}}_2^{L_2} \\ \bar{\mathbf{r}}_2^{G_{12}} - \hat{\mathbf{r}}_2^{L_2} \\ \bar{\mathbf{X}}_{F_2}^{G_{12}} - \hat{\mathbf{X}}_{F_2}^{L_2} \\ \bar{\mathbf{X}}_{F_{12}}^{G_{12}} - \hat{\mathbf{X}}_{F_{12}}^{L_2} \end{bmatrix} \right\|_{I^{L_2}}^2. \tag{11}$$

This linear least squares problem can be written in a compact form as

$$\text{minimize}\ \ \bar{f}(\bar{\mathbf{X}}^{G_{12}}) = \|A\,\bar{\mathbf{X}}^{G_{12}} - \mathbf{Z}\|_{I_Z}^2 \tag{12}$$

where $\mathbf{Z}$ is the constant vector combining the state estimates of the two maps

$$\mathbf{Z} = [\hat{\mathbf{X}}^{L_1},\ \hat{\mathbf{X}}^{L_2}]. \tag{13}$$

I_Z is the corresponding information matrix given by

$$I_Z = \begin{bmatrix} I^{L_1} & 0 \\ 0 & I^{L_2} \end{bmatrix}. \tag{14}$$

$\bar{\mathbf{X}}^{G_{12}}$ is the state vector represents the global map defined in (10). The coefficient matrix A is sparse and is given by

$$A = \begin{bmatrix} I & & & & & & \\ & I & & & & & \\ & & & I & & & \\ & & & & & & I \\ & & I & & & & \\ & & & I & & & \\ & & & & & I & \\ & & & & & & I \end{bmatrix} \tag{15}$$

where I is the identity matrix with the size corresponding to the different variables in the state vector $\bar{\mathbf{X}}^{G_{12}}$.

The optimal solution $\hat{\bar{\mathbf{X}}}^{G_{12}}$ to the linear least squares problem (12) can be obtained by solving the sparse linear equation

$$A^T I_Z A\ \bar{\mathbf{X}}^{G_{12}} = A^T I_Z \mathbf{Z}. \tag{16}$$

And the corresponding information matrix can be computed as

$$\bar{I}^{G_{12}} = A^T I_Z A. \tag{17}$$

From (10), we have $\mathbf{X}^{G_{12}} = g^{-1}(\bar{\mathbf{X}}^{G_{12}})$. Note that the function $g^{-1}(\cdot)$ has a closed-form as follows:

$$\begin{aligned}
\mathbf{X}^{G_{12}} &= g^{-1}(\bar{\mathbf{X}}^{G_{12}}) \\
&\Rightarrow \begin{cases}
\mathbf{t}_0^{G_{12}} = \bar{R}_2\,(\bar{\mathbf{t}}_0^{G_{12}} - \bar{\mathbf{t}}_2^{G_{12}}) \\
\mathbf{r}_0^{G_{12}} = r^{-1}(\bar{R}_0 \bar{R}_2^T) \\
\mathbf{t}_1^{G_{12}} = -\bar{R}_2\,\bar{\mathbf{t}}_2^{G_{12}} \\
\mathbf{r}_1^{G_{12}} = r^{-1}(\bar{R}_2^T) \\
\mathbf{X}_{F_1}^{G_{12}} = \bar{R}_2\,(\bar{\mathbf{X}}_{F_1}^{G_{12}} - \bar{\mathbf{t}}_2^{G_{12}}) \\
\mathbf{X}_{F_2}^{G_{12}} = \bar{R}_2\,(\bar{\mathbf{X}}_{F_2}^{G_{12}} - \bar{\mathbf{t}}_2^{G_{12}}) \\
\mathbf{X}_{F_{12}}^{G_{12}} = \bar{R}_2\,(\bar{\mathbf{X}}_{F_{12}}^{G_{12}} - \bar{\mathbf{t}}_2^{G_{12}})
\end{cases}
\end{aligned} \tag{18}$$

where $\bar{R}_2 = r(\bar{\mathbf{r}}_2^{G_{12}})$, $\bar{R}_0 = r(\bar{\mathbf{r}}_0^{G_{12}})$ are the rotation matrices of pose $\mathbf{P}_2$ and pose $\mathbf{P}_0$ in the state vector $\bar{\mathbf{X}}^{G_{12}}$.

Now the optimal solution to the nonlinear least squares problem (7) can be obtained by

$$\hat{\mathbf{X}}^{G_{12}} = g^{-1}(\hat{\bar{\mathbf{X}}}^{G_{12}}). \tag{19}$$

The corresponding information matrix $I^{G_{12}}$ can also be obtained by

$$I^{G_{12}} = \nabla^T\,\bar{I}^{G_{12}}\,\nabla \tag{20}$$

where ∇ is the Jacobian of $\bar{\mathbf{X}}^{G_{12}}$ with respect to $\mathbf{X}^{G_{12}}$, evaluated at $\hat{\mathbf{X}}^{G_{12}}$

$$\nabla = \frac{\partial g(\mathbf{X}^{G_{12}})}{\partial \mathbf{X}^{G_{12}}}\Big|_{\hat{\mathbf{X}}^{G_{12}}} \tag{21}$$

The result $(\hat{\mathbf{X}}^{G_{12}},\ I^{G_{12}})$ obtained this way is exactly the same as the one in (8) in Section II-C by using nonlinear least squares. We have used a number of numerical examples to confirm the equivalence of the two solutions.

Remark 1. For an intuitive explanation, in fact, $\bar{\mathbf{X}}^{G_{12}} = g(\mathbf{X}^{G_{12}})$ in (9) and (10) is the coordinate transformation function, transforming pose $\mathbf{P}_0$, $\mathbf{P}_1$ and feature $\mathbf{F}$ in the coordinate frame of $\mathbf{P}_2$, into $\mathbf{P}_0$, $\mathbf{P}_2$ and $\mathbf{F}$ in the coordinate frame of $\mathbf{P}_1$. Thus $\mathbf{X}^{G_{12}} = g^{-1}(\bar{\mathbf{X}}^{G_{12}})$ in (18) has the closed-form formula which is another coordinate transformation, from $\mathbf{P}_0$, $\mathbf{P}_2$ and $\mathbf{F}$ in the coordinate frame of $\mathbf{P}_1$, back to $\mathbf{P}_0$, $\mathbf{P}_1$ and $\mathbf{F}$ in the coordinate frame of $\mathbf{P}_2$. The linear way to solve the map joining problem presented in Section II-D is equivalent to solving a linear least squares problem plus a coordinate transformation, which can be illustrated in Fig. 1(c). The fact that the two maps are built in the same coordinate frame is the key to make this linear map joining approach possible.

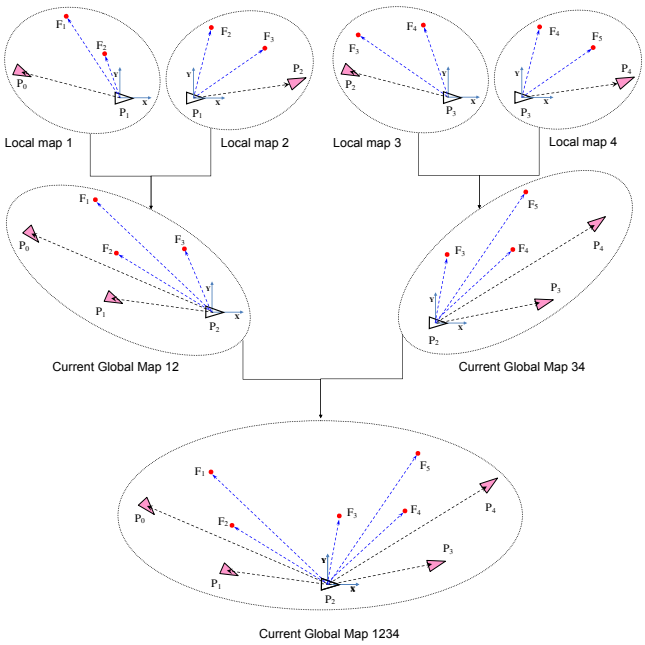


Fig. 2. The proposed Divide and Conquer map joining.

III. JOINING A SEQUENCE OF LOCAL MAPS

Based on the linear solution to joining two maps as shown in Fig. 1(b) and Section II-D, we can use either sequential map joining or Divide and Conquer map joining to solve large-scale SLAM problems.

A. Proposed Divide and Conquer Map Joining

The new Divide and Conquer map joining process we proposed is illustrated in Fig. 2. Please note: (1) local map 1 and local map 2 are built in the same coordinate frame, local map 3 and local map 4 are built in the same coordinate frame; (2) The global map 12 obtained by joining local map 1 and local map 2 is in the coordinate frame of the end pose of local map 2, while the global map 34 obtained by joining local map 3 and local map 4 is in the coordinate frame of the start pose of local map 3. Thus the global map 12 and global map 34 are in the same coordinate frame and can be joined together using the linear method in Section II-D.

B. Proposed New Sequential Map Joining

The new sequential map joining process we proposed can be illustrated by changing local map 3 built in the coordinate frame of its starting pose in Fig. 2 and sequentially joining every local map. Please note: (1) the first local map is built in the coordinate frame of its end pose instead of its starting pose; (2) the global map 12 after joining local map 1 and local map 2 is in the coordinate frame of the end pose of local map 2. Thus the result can be fused with local map 3 using the linear method in Section II-D.

Remark 2. Local map 1 in Fig. 2 can be simply built by performing a nonlinear least squares using all the observation and odometry data with state vector defined as the robot poses and feature positions in the coordinate frame of robot pose $\mathbf{P}_1$. Alternatively, we can first build the local map in the coordinate frame of $\mathbf{P}_0$, and then apply a coordinate transformation.

IV. LINEAR SOLUTION TO JOINING TWO POSE GRAPHS

The approaches proposed in Section II and Section III can also be applied to the pose graph SLAM. In this section, we explain the process of joining two pose graphs using linear least squares. The process can also be illustrated using Fig. 1(b) (by simply replacing all the features by poses).

A. Local Pose Graphs

Suppose there are two pose graphs

$$M^{L_1} = (\hat{\mathbf{X}}^{L_1},\ I^{L_1}), \quad M^{L_2} = (\hat{\mathbf{X}}^{L_2},\ I^{L_2}) \tag{22}$$

where $\hat{\mathbf{X}}^{L_1}$ and $\hat{\mathbf{X}}^{L_2}$ are the estimates of the state vectors $\mathbf{X}^{L_1}$ and $\mathbf{X}^{L_2}$ of each pose graph defined as

$$\begin{aligned} \mathbf{X}^{L_1} &= (\mathbf{X}^{L_1}_{P_{L1}},\ \mathbf{X}^{L_1}_{P_{L12}}) \\ \mathbf{X}^{L_2} &= (\mathbf{X}^{L_2}_{P_{L2}},\ \mathbf{X}^{L_2}_{P_{L12}}) \end{aligned} \tag{23}$$

and I^{L_1} and I^{L_2} are the associated information matrices. Both M^{L_1} and M^{L_2} are in the same coordinate frame of $\mathbf{P}_1$.

In the state vector $\mathbf{X}^{L_1}$ and $\mathbf{X}^{L_2}$ in (23), $\mathbf{X}^{L_1}_{P_{L12}}$ and $\mathbf{X}^{L_2}_{P_{L12}}$ are the common poses between M^{L_1} and M^{L_2}, while $\mathbf{X}^{L_1}_{P_{L1}}$ and $\mathbf{X}^{L_1}_{P_{L2}}$ are the poses that appear in only one of the pose graphs. All the relative poses can be defined similarly as (3) in Section II-C.

B. Joining Two Pose Graphs by Linear Optimization

Joining these two pose graphs M^{L_1} and M^{L_2} to build the global graph $\bar{M}^{G_{12}} = (\hat{\bar{\mathbf{X}}}^{G_{12}},\ \bar{I}^{G_{12}})$ can be solved the same as the linear method of joining two feature-based maps in Section II-D.

Suppose the global state vector $\bar{\mathbf{X}}^{G_{12}}$ is defined as

$$\bar{\mathbf{X}}^{G_{12}} = (\bar{\mathbf{X}}^{G_{12}}_{P_{L1}},\ \bar{\mathbf{X}}^{G_{12}}_{P_{L2}},\ \bar{\mathbf{X}}^{G_{12}}_{P_{L12}}) \tag{24}$$

where $\bar{\mathbf{X}}^{G_{12}}_{P_{L12}}$ are the common poses, $\bar{\mathbf{X}}^{G_{12}}_{P_{L1}}$ and $\bar{\mathbf{X}}^{G_{12}}_{P_{L2}}$ are the poses that appear in each pose graph, respectively. All the variables in the global state vector $\bar{\mathbf{X}}^{G_{12}}$ are in the coordinate frame of $\mathbf{P}_1$. Let $\mathbf{Z}$ be the constant vector combining the state estimates of the two pose graphs M^{L_1} and M^{L_2} as in (13), I_Z is the corresponding information matrix of $\mathbf{Z}$ obtained from the information matrices of each pose graph as in (14). Also because all these two pose graphs and the global pose graph are in the same coordinate frame of $\mathbf{P}_1$, joining the two pose graphs can be solved as a linear least squares problem with the same form as (12). And the coefficient matrix A is

$$A = \begin{bmatrix} I & & \\ & & I \\ & I & \\ & & I \end{bmatrix}. \tag{25}$$

Then the estimate of the state vector $\hat{\hat{\mathbf{X}}}^{G_{12}}$ and the corresponding information matrix $\bar{I}^{G_{12}}$ of the global graph $\bar{M}^{G_{12}}$ can be solved by using (16) and (17).

Remark 3. The transformation of the global pose graph from $\bar{M}^{G_{12}}$ to $M^{G_{12}}$ in the coordinate frame of $\mathbf{P}_2$ can be done the same way as described in Section II-D, thus the process is not detailed here.

Remark 4. It should be mentioned that, for the feature-based SLAM problem, the only common pose between two local maps defines the coordinates of both two local maps as shown in Fig. 1(b), thus this pose is not in the state vectors of the two local maps. So there is no common pose in the observations $\mathbf{Z}$ in (13) in Section II-D. When doing linear local map joining, there is no need to care about wraparound of the rotation angles. However, for the pose graph map joining problem, there can be many common poses between two local graphs, and wraparound must be considered. In this paper, because it is a linear least squares, we only wraparound the angle on one of the two observations of a common pose in $\mathbf{Z}$ to make the difference between the two angles in the observations fall into $(-\pi, \pi]$.

V. EXPERIMENTAL RESULTS

In the experiments, publicly available 2D and 3D simulation and real datasets of feature-based and pose graph SLAM have been used to check the validity and accuracy of the proposed Linear SLAM algorithm. Both sequential and Divide and Conquer map joining are implemented using this linear approach (please refer to the source code on OpenSLAM under the project "Linear SLAM"). While the results presented below are all from the Divide and Conquer implementation which is computationally less expensive [10]. The computational costs for the Linear SLAM algorithm using different datasets are listed in Table I.

A. 2D and 3D Feature-based SLAM Datasets

For 2D feature-based SLAM, the Victoria Park dataset [18] and the DLR dataset [19] are used. Here we used the local map datasets available on OpenSLAM under project 2D-I-SLSJF. In Table I, VicPark200 means the 200 local maps built from the Victoria Park dataset; DLR200 means the 200 local maps built from the DLR dataset, VicPark6898 means the 6898 local maps built from the Victoria Park dataset and DLR3298 means the 3298 local maps built from the DLR dataset. The results of Linear SLAM algorithm using the four datasets are shown in Fig. 3(a) to Fig. 3(d), respectively. As comparison, the results from both SLSJF [13] and I-SLSJF [17] are also shown in the figures.

The result of I-SLSJF is equivalent to that of performing nonlinear least squares optimization using the local maps as observations [17]. Since each local map in the VicPark6898 and DLR3298 datasets were built using the observations of features from one pose together with the odometry to the next pose, the I-SLSJF results using these two datasets are equivalent to those using full least squares optimization with all the observations and odometry information.

TABLE I
COMPUTATIONAL COSTS* OF LINEAR SLAM ALGORITHM (IN SECONDS)

Feature-based		Pose Graph	
Dataset	time	Dataset	time
VicPark200	1.49	Intel943	1.77
DLR200	0.77	Manhattan3500	6.87
VicPark6898	18.01	City10000	31.37
DLR3298	6.44	Sphere2500	23.48
3D Simu870	7.94	ParkingGarage1661	16.47

*Implemented in Matlab, run on an Intel E8400@3.0GHz CPU, code is not optimized. Time for building local maps in VicPark200 and DLR200 datasets are not included.

TABLE II
COMPARISON OF χ^2 OF MAP JOINING RESULTS FOR DIFFERENT ALGORITHMS

Dataset	Nonlinear Least Squares	SLSJF	Linear SLAM
VicPark200	3643	3644	3645
DLR200	11164	14603	12577
VicPark6898	9012	9013	9020
DLR3298	27689	28316	28373

As shown in Fig. 3(a) to Fig. 3(d), the results of Linear SLAM look better than those of SLSJF, while nearly the same as the optimal I-SLSJF results. As shown in Table I, the computational cost of Linear SLAM is very cheap. This is mainly because no iteration is needed in the algorithm.

The final χ^2 errors are also compared in Table II. The χ^2 of Linear SLAM results are similar to those from the SLSJF results, both of which are very close to the globally optimal nonlinear least squares solutions.

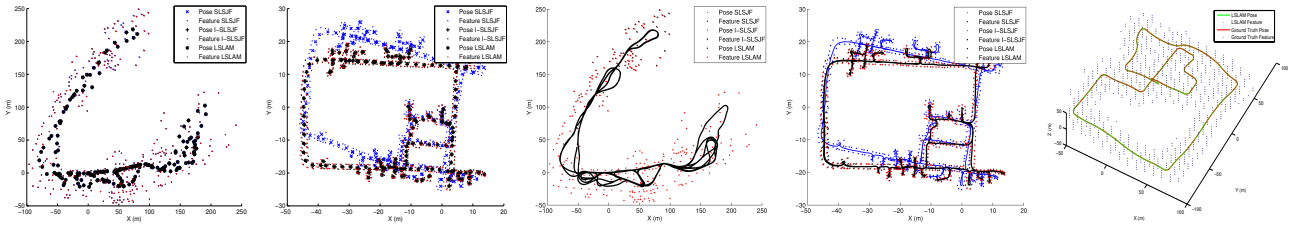
For the 3D feature-based SLAM, a simulation is done with 871 poses trajectory and uniformly distributed features in the environment. The Linear SLAM algorithm is applied with 870 local maps and the result is shown in Fig. 3(e).

B. 2D and 3D Pose Graph SLAM Datasets

For 2D pose graph SLAM, experimental dataset Intel, simulation datasets Manhattan and City10000 are used to test the proposed Linear SLAM algorithm. The results are shown in Fig. 4(a) to Fig. 4(c). For 3D pose graph SLAM, simulation dataset Sphere and experimental dataset Parking Garage are used. The results of Linear SLAM are shown in Fig. 4(d) to Fig. 4(e). As comparison, G2O [4] are also used to get the full nonlinear least squares SLAM results for these five pose graph datasets. As shown in Fig. 4, the results of Linear SLAM using the five pose graph datasets are all very close to the G2O results.

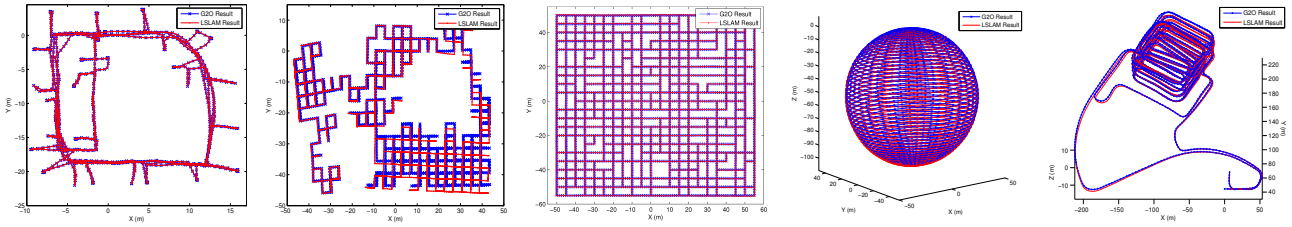
VI. RELATED WORK AND DISCUSSION

The Linear SLAM algorithm proposed in this paper is based on the idea of map joining, which has been a strategy applied by many research works [10]–[16]. Map joining has shown to be able to improve the efficiency of SLAM as well as reduce the linearization errors. In this paper, we demonstrate that map joining can actually be implemented in a way such that only linear least squares and nonlinear



(a) VicPark200 (b) DLR200 (c) VicPark6898 (d) DLR3298 (e) 3D Simu870

Fig. 3. 2D and 3D feature-based SLAM results of Linear SLAM (LSLAM). Fig. 3(a) and Fig. 3(b) are the results of joining the 200 local maps for 2D Victoria Park and DLR datasets. Fig. 3(c) and Fig. 3(d) are the results of Victoria Park and DLR datasets by joining 6898/3298 local maps. All the 2D results are compared with SLSJF and I-SLSJF. Fig. 3(e) is the result of a 3D simulation with 870 local maps.



(a) Intel (b) Manhattan (c) City10000 (d) Sphere (e) Parking garage

Fig. 4. 2D and 3D pose graph results of Linear SLAM (LSLAM). Fig. 4(a) to Fig. 4(c) are the results of Intel, Manhattan and City10000 2D pose graph datasets. Fig. 4(d) and Fig. 4(e) are the results of Sphere and Parking garage 3D pose graph datasets. All results are compared with those of G2O.

coordinate transformations are needed. Thus there is no need to compute initial value and there is no local minima issue.

The key to make the map joining problem a linear problem is due to the fact that the two maps to be fused are in the same coordinate frame. Basically, when joining two local maps built at different coordinate frames, the estimation/optimization problem is nonlinear, which has been used in most of the existing map joining algorithms. But when joining two local maps with the same coordinate frames, the problem can be linear, which is used in this paper.

Different from traditional local submap joining, the final global map obtained from this Linear SLAM algorithm is not in the coordinate frame of the first local map. For Divide and Conquer map joining, the coordinate frame of the final global map is the same as the coordinate frame of the last two maps that were fused. For sequential map joining, the coordinate frame of the final global map is the last robot pose in the last local map. In order to compare the result with traditional local submap joining, a coordinate transformation is needed. This is somewhat similar to the robocentric mapping [20] where the map is transformed into the current robot pose frame in order to reduce the linearization error in the EKF framework.

Using Divide and Conquer strategy makes the algorithm more efficient as compared with the sequential map joining, which is the major improvement of [21] over [13]. The main reason is that the size of the current global map to be combined with the next local map keeps increasing in the sequential map joining.

In the proposed Linear SLAM framework, the pose graph SLAM can be solved in the same way as feature-based SLAM. The only difference is the following. In feature-based SLAM, there is no common pose in the two maps to be fused apart from the one which serves as the (common) coordinate frame, so there is no wraparound problem for the robot orientations. But in pose graph SLAM, there might be a number of common poses in the two pose graphs to be fused. Thus the wraparound of robot orientation angles need to be considered. Because we only fuse two local maps at a time, the wraparound issue can be dealt with very easily which is different from that in the linear approximation approach proposed in [22] [23]. Some other limitations of the linear approximation method in [23] are: (1) it can only be applied to 2D pose graph SLAM; (2) it requires special structure on the covariance matrices.

Similar to SLSJF, the sparseness of the information matrix is maintained in the proposed linear map joining algorithm as the coefficient matrix A in (15)(25) are always sparse. Also similar to SLSJF, there is no information loss or information reuse in the Linear SLAM algorithm. The differences between the results in our Linear SLAM and the results of globally optimal solution to the nonlinear least squares SLAM come from two reasons: (i) we summarized the local map information as the local map estimate together with its uncertainty (information matrix), instead of using the original odometry and observation information, which is the same as many map joining algorithms; (ii) we fuse two maps at a time instead of fusing all the local maps together in one go using optimization, thus the quality is not as good as that of I-SLSJF. In general, the more accurate the local maps are, the closer the Linear SLAM results from the globally optimal solution. Thus we recommend to use nonlinear optimization techniques (and robust back-end if necessary) to build high quality small local maps and then apply our linear map joining algorithm. It is clear that fusing local maps in different coordinate frames optimally using

linear approach is impossible. Thus if we want to apply smoothing after we join all the local maps together, just like in I-SLSJF, the problem becomes nonlinear.

Data association is not considered in the experimental results presented in this paper. However, since our Linear SLAM approach is using linear least squares and the associated information matrices are always available, many of the data association methods suggested for EIF based or optimization based SLAM algorithms (such as SLSJF [13], ESEIF [24] and iSAM [5]), including covariance submatrix recovery, can be applied with the proposed Linear SLAM algorithm.

VII. CONCLUSION

This paper demonstrated that submap joining can be implemented in a way such that only linear least squares and coordinate transformations are needed. There is no assumption on the local map covariance matrices and the approach can be applied to both feature-based and pose graph SLAM, 2D and 3D. This new approach avoids the issues of local minima and inaccurate initial value which exist in almost all the existing nonlinear optimization based SLAM algorithms.

Simulation and experimental results using publicly available datasets demonstrated the accuracy and efficiency of the algorithms. Although the results look very promising, they are still not equivalent to the optimal solution to SLAM based on nonlinear least squares starting from accurate initial values. Like other submap joining algorithms, how different the results are from the optimal solution depends on many factors such as the quality of the original sensor data as well as the quality of local maps.

The results in this paper again show that SLAM is not very far from a linear problem. We are in the process of revisiting some other SLAM problems, such as SLAM with line features, range-only SLAM, and monocular SLAM to work out how the proposed Linear SLAM algorithm can be applied to these different SLAM problems to improve the reliability and efficiency of the SLAM algorithms. We are also planning to apply this Linear SLAM algorithm in active SLAM and investigate the possibility of using this Linear SLAM algorithm to help in the SLAM front-end to solve the robust back-end problem.

ACKNOWLEDGMENT

This work is supported in part by Australian Research Council (ARC) Linkage Project LP0884112 and ARC Discovery Project DP120102786.

REFERENCES

[1] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots*, vol. 4, pp. 333-349, 1997.

[2] F. Dellaert and M. Kaess, "Square root SAM: Simultaneous localization and mapping via square root information smoothing," *International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181-1203, 2006.

[3] K. Konolige, G. Grisetti, R. Kummerle, B. Limketkai and R. Vincent, "Efficient sparse pose adjustment for 2D mapping," In *Proc. International Conference on Intelligent Robots and Systems (IROS)*, pp. 22-29, 2010.

[4] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige and W. Burgard, "G2O: A general framework for graph optimization," In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3607-3613, 2011.

[5] M. Kaess, A. Ranganathan, and F. Dellaert. "iSAM: Incremental smoothing and mapping". *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365-1378, 2008.

[6] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *International Journal of Robotics Research*, vol. 31, pp. 217-236, 2012.

[7] E. Olson, J. Leonard, and S. Teller, "Fast iterative optimization of pose graphs with poor initial estimates," In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2262-2269, 2006.

[8] G. Grisetti, C. Stachniss, and W. Burgard, "Non-linear constraint network optimization for efficient map learning," *IEEE Transations on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 428-439, 2009.

[9] S. Huang, Y. Lai, U. Frese and G. Dissanayake, "How far is SLAM from a linear least squares problem?" In *Proc. International Conference on Intelligent Robots and Systems (IROS)*, pp. 3011-3016, 2010.

[10] L. M. Paz, J. D. Tardos and J. Neira, "Divide and Conquer: EKF SLAM in O(n)," *IEEE Transactions on Robotics,* vol 24, no. 5, pp. 1107-1120, 2008.

[11] 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*, vol. 21, no. 4, pp. 311-330, 2002.

[12] S. B. Williams, "Efficient solutions to autonomous mapping and navigation problems," Ph.D. Thesis, Australian Centre of Field Robotics, University of Sydney, 2001.

[13] S. Huang, Z. Wang, and G. Dissanayake, "Sparse local submap joining filter for building large-scale maps," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1121-1130, 2008.

[14] M. Bosse, P. M. Newman, J. J. Leonard and S. Teller, "SLAM in large-scale cyclic environments using the atlas framework," *International Journal of Robotics Research*, vol. 23, no. 12, pp. 1113-1139, 2004.

[15] B. Kim, M. Kaess, L. Fletcher, J. Leonard, A. Bachrach, N. Roy and S. Teller, "Multiple relative pose graphs for robust cooperative mapping," In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3185-3192, 2010.

[16] K. Steedly, D. Ni and F. Dellaert, "Tectonic SAM: Exact, out-of-core, submap-based SLAM," In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1678-1685, 2007.

[17] S. Huang, Z. Wang, G. Dissanayake and U. Frese, "Iterated SLSJF: A sparse local submap joining algorithm with improved consistency," In *Proc. Australiasan Conference on Robotics and Automation*, 2008.

[18] 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*, vol. 17, pp. 242-257, 2001.

[19] J. Kurlbaum and U. Frese, "A benchmark data set for data association," Technical Report, University of Bremen, available online: http://www.sfbtr8.uni-bremen.de/reports.htm, Data available online http://radish. sourceforge.net/.

[20] 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*, vol. 55, no. 1, pp. 21-29, 2007.

[21] C. Cadena and J. Neira, "SLAM in O(log n) with the Combined Kalman-Information Filter," *Robotics and Autonomous Systems*, vol. 58, no. 11, pp. 1207-1219, 2010.

[22] J. L. Martinez, J. Morales, A. Mandow and A. Garcia-Cerezo, "Incremental closed-form solution to globally consistent 2D range scan mapping with two-step pose estimation," In *Proc. The 11th IEEE International Workshop on Advanced Motion Control*, pp. 252-257, 2010.

[23] L. Carlone, R. Aragues, J. Castellanos and B. Bona, "A linear approximation for graph-based simultaneous localization and mapping," In *Proc. Robotics: Science and Systems (RSS)*, 2011.

[24] M. R. Walter, R. M. Eustice and J. J. Leonard, "Exactly sparse extended information filters for feature-based SLAM," *The International Journal of Robotics Research*, vol. 26, no. 4, pp. 335-359, 2007.