Robotics: Science and Systems 2021
Held Virtually, July 12–16, 2021

Feature-Based SLAM: Why Simultaneous Localisation and Mapping?

Liang Zhao, Zhehua Mao, and Shoudong Huang
Centre for Autonomous Systems, Faculty of Engineering and Information Technology
University of Technology Sydney, Australia.
Email: Liang.Zhao@uts.edu.au

***Abstract*—In this paper, we first prove an interesting result for point feature based SLAM. "*When the covariance matrices of feature observation errors are isotropic, the robot poses and feature positions obtained in each Gauss-Newton iteration (when solving a reformulated least squares optimisation based SLAM) are independent of the feature positions in the previous step*". That is, even if we reset the feature positions to different random values before each iteration, the results after the iteration never change. Building on this finding, we propose an algorithm to solve the robot poses only ("localisation") and show that the algorithm generates exactly the same robot poses in each iteration as the Gauss-Newton method (SLAM). The optimal feature positions can be easily recovered using a closed-form formula after the optimal robot poses are obtained.**

Similarly, when the covariance matrices of odometry translation errors are also isotropic, we can prove that the SLAM results are independent of both the feature positions and the robot positions. Thus, we can have an "rotation-only algorithm" which generates the same robot rotations as the full SLAM. Again, the optimal robot positions and the optimal feature positions can be computed from the obtained optimal robot rotations using a closed-form formula. We have used multiple 2D and 3D SLAM datasets to demonstrate our research findings. The video shows the interesting convergence results can be found at https://youtu.be/j1T8toqGtDE. We expect the findings in this paper can help SLAM researchers to further understand the special structure of the SLAM problems and to further develop more efficient and reliable SLAM algorithms.

I. Introduction

Simultaneous Localisation and Mapping (SLAM) has been a key problem in robotics for about 30 years [4]. In the state-of-the-art approaches for feature-based SLAM, given all the relative odometry measurements and the relative feature observations, under the assumption of (independent) Gaussian noises, the feature-based SLAM problem is formulated as a nonlinear least squares (NLLS) problem over all the robot poses and feature positions [6]. Similarly, state-of-the-art approaches for pose-graph SLAM have formulated the problem as a NLLS problem over all the robot poses given the relative pose measurements [20]. Based on the better and better understanding of the special structrue of the high dimensional SLAM problems, a lot of solvers for SLAM have been developed for 2D/3D feature-based SLAM and pose-graph SLAM (e.g. [14][13][2][12][1][19]). In particular, some recently developed algorithms can efficiently recover certifiably globally optimal solutions to the SLAM problems under certain conditions using techniques ranging from semidefinite relaxation and Riemannian Staircase procedure [21][3], sparse-bounded sum of squares programming [17], and complex number representation [7].

Researchers have shown that the SLAM problems become considerably easier to solve when the noise covariance matrices are *isotropic*. For example, in the tree-based network optimizer [8][20], good convergence results from bad initial values are reported for large pose-graphs when the noise covariance matrices are isotropic. In [10], an unexpected convergence of vanilla Gauss-Newton (GN) algorithm to the optimal solution from random initial guesses in high-dimensional feature-based SLAM problems has been reported, when the noise covariance matrices are isotropic. In [5], the convergence of GN algorithm in pose-graph SLAM is analysed where a conservative estimate for the basin of attraction of the Maximum Likelihood estimate in pose-graph SLAM was derived under the assumption of isotropic noise covariance matrices. In [11][24][25], the objective function in SLAM is studied when the noise covariance matrices are isotropic. They show that in general a 2D m-step feature-based SLAM problem is equivalent to a NLLS problem over only m variables (i.e., the robot orientations in m poses), in particular, they develop clear conditions and algorithms that can be guaranteed to find the globally optimal solution to the 1-step and 2-step SLAM when the orientation information in odometry is not used. Similarly, some dimension reduction results for pose-graph SLAM has been presented in [23]. However, the dimension reduction results presented in [25][23] cannot be used to obtain an efficient SLAM algorithm for multiple steps SLAM since the resulting reduced dimensional problem is very complicated and no longer has the sparse structure.

Some modern SLAM solvers have also assumed the isotropic noise covariance matrices. For example, in SE-sync [21], the noises on the relative translations are assumed to be Gaussian with isotropic covariance matrices, and the pose-graph optimisation problem is efficiently solved through semidefinite relaxation using Rienannian staricase optimisation on the Stiefel manifold. Similarly, Cartan-sync in [3] uses the same isotropic covariance matrices assumption and improved SE-sync by using the Cartan motion group and introducing a novel preconditioner to accelerate the algorithm. The isotropic noise covariance matrices assumption is also used in [7], where a more compact unit complex number representation instead of the matrix representation of $SO(2)$

is used to speed up the semidefinite relaxation and Rienannian staricase optimisation for 2D pose-graph and 2D feature-based SLAM problems.

Recently in medical image computing, [18] proposed a direct 3D multi-image registration method which simultaneously optimises the relative poses of 3D Ultrasound images and the intensities of the fused 3D image. It is shown that the relative poses can be directly solved without considering the intensities of the fused image. Inspired by [18], in this paper, we prove that when the covariance matrices of feature observation errors are isotropic, the robot poses and feature positions in each iteration of GN algorithm is independent of the feature positions in the previous step. We prove the result based on Schur complement and we also propose an algorithm to solve the robot poses only without having features in the state vector. Once the optimal robot poses are obtained, optimal feature positions can be obtained directly from a closed-form formula. Similarly, when the covariance matrices of odometry translation errors are also isotropic, the robot poses and feature positions in each GN iteration are independent of both feature positions and robot positions in the previous step. Thus, a rotation-only algorithm can be used to obtain the optimal robot rotations. Again, the optimal robot positions and feature positions can be computed using a closed-form formula. The results of both the pose-only and rotation-only algorithms are shown to be identical to the corresponding full NLLS SLAM algorithm in each iteration step. These new properties provide us some further insights into the structure of SLAM problems. We expect these new findings can help to further develop more efficient SLAM solvers.

This paper is organised as follows. In Section II, the NLLS formulation of SLAM is stated clearly. Section III derives an alternative SLAM formulation when covariance matrices of feature observation errors are isotropic. In Section IV, it is proved that the results in each iteration of GN is independent of the features and we propose a pose-only algorithm which is equivalent to the full NLLS SLAM algorithm. In Section V, we prove similar results when the odometry translation error covariance matrices are also isotropic. In Section VI, examples are presented to illustrate the results proved in this paper. Some discussions on the relations with existing results are given in Section VII. Finally Section VIII concludes the paper.

Notations: Throughout the paper, superscript T and -1 stand for, respectively, the transposition and the inverse of a matrix; I denotes the identity matrix with compatible dimension; $\mathbf{0}$ represents the zero vector with compatible dimension; and weighted 2-norm $\|e\|_C^2 = e^T C e$, where C is a positive definite matrix and e is a vector. The symbol $\mathrm{diag}(C_1, \dots, C_n)$ denotes a block-diagonal matrix whose diagonal blocks are $C_1, \dots, C_n$.

II. Problem Formulation

We consider 2D or 3D point feature based SLAM problems. Suppose n point features $\{f_i\}_{i=1}^n$ are observed from a sequence of $m+1$ robot poses $\{r_i\}_{i=0}^m$. We use Z_k^i to denote the observation made from pose r_i to feature f_k. We use O_i $(1 \leq i \leq m)$ to denote the odometry measurement between pose r_{i-1} and pose r_i. Both the observations and odometry are corrupted by zero-mean Gaussian noises with covariance matrices $P_{Z_k^i}$ and P_{O_i}, respectively. X_{f_k} denotes the position of feature f_k. $X_{r_i} = \{R_i, t_i\}$ denotes the rotation matrix and translation vector of robot pose r_i. The coordinate frame is defined by the robot pose r_0. That is, $X_{r_0} = \{I, \mathbf{0}\}$ where I is the 2×2 (2D) or 3×3 (3D) identity matrix.

The NLLS SLAM formulation [6] uses the odometry and observation information to estimate the state containing all the robot poses and all the feature positions

$$\mathbf{X} \triangleq \{X_{r_1}, \cdots, X_{r_m}, X_{f_1}, \cdots, X_{f_n}\} \tag{1}$$

and minimises the objective function

$$\begin{aligned} f(\mathbf{X}) =& \sum_{i=0}^{m} \sum_{j=1}^{n_i} \|Z_{k_{ij}}^i - H^{Z_{k_{ij}}^i}(\mathbf{X})\|^2_{P^{-1}_{Z^i_{k_{ij}}}} \\ &+ \sum_{i=1}^{m} \|H^{O_i}(\mathbf{X})\|^2_{P^{-1}_{O_i}} \end{aligned} \tag{2}$$

where $O_i = \{O_i^t, O_i^R\}$ is the odometry from pose r_{i-1} to pose r_i, O_i^t is the translation part while O_i^R is the rotation part $(1 \leq i \leq m)$, $Z^i_{k_{ij}}$ are observations (assume n_i features are observed from robot pose r_i and k_{ij} is the global index of the j-th feature observed from pose r_i), and P_{O_i} and $P_{Z^i_{k_{ij}}}$ are the corresponding covariance matrices.

In the above least squares SLAM formulation, $H^{Z_k^i}(\mathbf{X})$ and $H^{O_i}(\mathbf{X})$ are the corresponding functions relating Z_k^i and O_i to the state $\mathbf{X}$. An odometry measurement is a function of two poses $X_{r_{i-1}}$ and X_{r_i} and is given by

$$H^{O_i}(\mathbf{X}) = \begin{bmatrix} O_i^t - R_{i-1}^T(t_i - t_{i-1}) \\ d_{SO}(O_i^R, R_{i-1}^T R_i) \end{bmatrix}, \tag{3}$$

where $d_{SO}(\star, \bullet)$ means the distance function on the Lie group $SO(2)$ or $SO(3)$. One example is $\|\log(\star^\top \bullet)^\vee\|$ where $^\vee$ means the inverse of the skew-symmetric operator.

A single feature observation is a function of one pose rotation and translation X_{r_i} and one feature position X_{f_k} which is given by

$$H^{Z_k^i}(\mathbf{X}) = R_i^T(X_{f_k} - t_i). \tag{4}$$

In particular, since $X_{r_0} = \{R_0, t_0\} = \{I, \mathbf{0}\}$, the odometry function from pose r_0 to pose r_1 is given by

$$H^{O_1}(\mathbf{X}) = \begin{bmatrix} O_1^t - t_1 \\ d_{SO}(O_1^R, R_1) \end{bmatrix} \tag{5}$$

and the observation function from pose r_0 to feature f_k is given by

$$H^{Z_k^0}(\mathbf{X}) = X_{f_k}. \tag{6}$$

III. Alternative Formulation when Covariance Matrices of Feature Measurements are Isotropic

The NLLS in (2) can be simplified when the covariance matrix of the feature measurement error $P_{Z^i_{k_{ij}}}$ is isotropic for every i and j. A matrix is isotropic means it is proportional to the identity matrix (sometimes also called spherical [23]). For SLAM with range and bearing observations, the covariance matrix $P_{Z^i_{k_{ij}}}$ can be approximated by isotropic matrix and the SLAM result are not affected much in real-world scenarios (as seen in the experiments in Section VI).

When the covariance $P_{Z^i_{k_{ij}}}$ are isotropic positive definite matrices, we have

$$\begin{aligned} &\|Z^i_{k_{ij}} - H^{Z^i_{k_{ij}}}(\mathbf{X})\|^2_{P^{-1}_{Z^i_{k_{ij}}}} \\ =&\|Z^i_{k_{ij}} - R_i^T(X_{f_{k_{ij}}} - t_i)\|^2_{P^{-1}_{Z^i_{k_{ij}}}} && (7) \\ =&\|(X_{f_{k_{ij}}} - (R_i Z^i_{k_{ij}} + t_i))\|^2_{P^{-1}_{Z^i_{k_{ij}}}} . && (8) \end{aligned}$$

Here we have used the property that any isotropic matrix P commutes any rotation matrix R, thus $R^T P R = P$. The expression in (8) is easier to deal with than that in (7) because there is no product of variables in (8).

Thus, when $P_{Z^i_{k_{ij}}}$ are all isotropic, the objective function in NLLS SLAM formulation (2) can be written as

$$\begin{aligned} f_1(\mathbf{X}) =&\sum_{i=0}^{m}\sum_{j=1}^{n_i} \|X_{f_{k_{ij}}} - (R_i Z^i_{k_{ij}} + t_i)\|^2_{P^{-1}_{Z^i_{k_{ij}}}} \\ &+ \sum_{i=1}^{m} \|H^{O_i}(\mathbf{X})\|^2_{P^{-1}_{O_i}}. && (9) \end{aligned}$$

Remark 1: Please note that although the objective function $f_1(\mathbf{X})$ in (9) equals to the objective function $f(\mathbf{X})$ in (2) when the covariance $P_{Z^i_{k_{ij}}}$ are isotropic, the two equivalent NLLS problems (minimising $f(\mathbf{X})$ and minimising $f_1(\mathbf{X})$) have different nonlinear structures. We will prove in the next section that when solving the reformulated NLLS problem (minimising $f_1(\mathbf{X})$), the result in each iteration step is independent of features.

In the special case when no odometry information exists, the objective function in NLLS SLAM formulation (9) can be simplified as

$$f_0(\mathbf{X}) = \sum_{i=0}^{m}\sum_{j=1}^{n_i} \|X_{f_{k_{ij}}} - (R_i Z^i_{k_{ij}} + t_i)\|^2_{P^{-1}_{Z^i_{k_{ij}}}} . \qquad (10)$$

IV. Solving Poses in SLAM is Independent of Features

The alternative formulation of the feature-based SLAM problem in (9) is a standard NLLS problem. In this section, we will prove that when GN iteration is used to solve this NLLS problem, solving the robot poses is independent of the features.

We first consider the case when no odometry exists. That is, finding $\mathbf{X}$ defined in (1) such that the objective function $f_0(\mathbf{X})$ defined in (10) is mimimised.

A. Gauss-Newton Iteration

Let $\mathbf{X} = \{\mathbf{X}_r, \mathbf{X}_f\}$ where $\mathbf{X}_r = \{X_{r_1}, \cdots, X_{r_m}\}$ contains all the robot poses and $\mathbf{X}_f = [X^T_{f_1}, \cdots, X^T_{f_n}]^T$ contains all the features in the state (1) respectively; $\mathbf{Z} = [..., (Z^i_{k_{ij}})^T, ...]^T$ be the measurement vector which contains all the feature observations in (10); $\mathbf{X}^Z_f = [..., X^T_{f_{k_{ij}}}, ...]^T$ containing all the corresponding features in the state $\mathbf{X}_f$ w.r.t. the measurement vector $\mathbf{Z}$; $H^Z(\mathbf{X}_r) = [..., (R_i Z^i_{k_{ij}} + t_i)^T, ...]^T$; and $\Sigma_Z^{-1} = \text{diag}(..., P^{-1}_{Z^i_{k_{ij}}}, ...)$ be the weight which defined by combining all the covariance matrices of feature measurements. The least squares optimisation problem in (10) is to seek $\mathbf{X}$ such that

$$f_0(\mathbf{X}) = \|F^Z(\mathbf{X})\|^2_{\Sigma_Z^{-1}} = (F^Z(\mathbf{X}))^T \Sigma_Z^{-1} F^Z(\mathbf{X}) \qquad (11)$$

is minimised, where

$$F^Z(\mathbf{X}) = \mathbf{X}^Z_f - H^Z(\mathbf{X}_r). \qquad (12)$$

A solution to (11) can be obtained by using the standard GN iteration, which starts with an initial estimate $\mathbf{X}(0)$ and iterating with $\mathbf{X}(k+1) = \mathbf{X}(k) \oplus \Delta$[1], where the update vector Δ is the solution to[2]

$$J^T \Sigma_Z^{-1} J \Delta = -J^T \Sigma_Z^{-1} F^Z(\mathbf{X}(k)) \qquad (13)$$

where J is the linear mapping represented by the Jacobian matrix $\partial F^Z / \partial \mathbf{X}$ evaluated at $\mathbf{X}(k)$.

One important fact is that the Jacobian of $F^Z(\mathbf{X})$ w.r.t. features is a constant matrix. Let A be the matrix indicating which feature is observed in each observations. If a feature observation $Z^i_{k_{ij}}$ (row) is corresponding to feature X_{f_k} (column), the corresponding element in A is 1. An example of the matrix A can be shown as

$$A = \begin{array}{c} \\ Z^i_k \\ Z^i_{k+1} \\ Z^{i+1}_k \\ Z^{i+1}_{k+1} \end{array} \begin{array}{c} \begin{array}{cccc} ... & X_{f_k} & X_{f_{k+1}} & ... \end{array} \\ \begin{pmatrix} \mathbf{0}^T & 1 & 0 & \mathbf{0}^T \\ \mathbf{0}^T & 0 & 1 & \mathbf{0}^T \\ \mathbf{0}^T & 1 & 0 & \mathbf{0}^T \\ \mathbf{0}^T & 0 & 1 & \mathbf{0}^T \end{pmatrix} \end{array}.$$

From the fact that $F^Z(\mathbf{X})$ is linear in $\mathbf{X}_f$ and $H^Z(\mathbf{X}_r)$ is independent of $\mathbf{X}_f$, we can easily obtain the following lemma.

Lemma 1: The Jacobian of function $F^Z(\mathbf{X})$ in (12) w.r.t. robot poses $\mathbf{X}_r$ is independent of $\mathbf{X}_f$. The Jacobian of function $F^Z(\mathbf{X})$ w.r.t. features $\mathbf{X}_f$ is a constant matrix $J_f = A \otimes I$ and $\mathbf{X}^Z_f = J_f \mathbf{X}_f$, where I is the 2×2 (2D) or 3×3 (3D) identity matrix and $\otimes$ is the Kronecker product.

B. The Independence of Poses and Features

We first prove the following theorem.

Theorem 1: In each step of GN iteration for minimising the NLLS SLAM objective function in (10) (or (11)), the result of robot poses and features is independent of the values of features $\mathbf{X}_f$ in the previous step.

Proof. The linear equation (13) can be solved in a similar way as in Bundle Adjustment problems [16]. Suppose (13) is rewritten as

$$\begin{bmatrix} U & W \\ W^T & V \end{bmatrix} \begin{bmatrix} \Delta_r \\ \Delta_f \end{bmatrix} = \begin{bmatrix} \mathbf{b}_r \\ \mathbf{b}_f \end{bmatrix} \qquad (14)$$

[1] $\oplus$ is a special plus operator since rotation is involved.

[2] Here Δ is different in each iteration step k. However, we are omitting the "(k)" to simplify the equations. Similarly, we are omitting many "(k)"s in the proof of Theorem 1.

where Δ_r and Δ_f are the robot and feature parts of the update of state vector $\Delta = [\Delta_r^T, \Delta_f^T]^T$ in (13). $U = J_r^T \Sigma_Z^{-1} J_r$, $V = J_f^T \Sigma_Z^{-1} J_f$, $W = J_r^T \Sigma_Z^{-1} J_f$, $\mathbf{b}_r = -J_r^T \Sigma_Z^{-1} F^Z(\mathbf{X})$, and $\mathbf{b}_f = -J_f^T \Sigma_Z^{-1} F^Z(\mathbf{X})$, where $J_r = -\partial H^Z / \partial \mathbf{X}_r$ and $J_f = \partial \mathbf{X}_f^Z / \partial \mathbf{X}_f$ are the Jacobians w.r.t. robot poses and features respectively in the Jacobian matrix $J = [J_r, J_f]$ in (13).

Then, the whole linear system (14) can be solved by first solving the update of reduced robot system using Schur complement

$$(U - WV^{-1}W^T)\Delta_r = \mathbf{b}_r - WV^{-1}\mathbf{b}_f \tag{15}$$

and then performing back-substitution to get the feature update

$$V\Delta_f = \mathbf{b}_f - W^T\Delta_r. \tag{16}$$

It should be mentioned that, the linear system solved using Schur complement in Bundle Adjustment is only for reducing the computational complexity [16][22], Bundle Adjustment problems do not have the independent properties proved in this paper.

Now we prove that solving the update of robot pose in (15) is independent of the features $\mathbf{X}_f$.

In fact, we can rewrite (15) as

$$\begin{aligned} &(J_r^T \Sigma_Z^{-1} J_r - J_r^T \Sigma_Z^{-1} J_f (J_f^T \Sigma_Z^{-1} J_f)^{-1} J_f^T \Sigma_Z^{-1} J_r)\Delta_r \\ =& - J_r^T \Sigma_Z^{-1} (\mathbf{X}_f^Z - H^Z(\mathbf{X}_r)) \\ &+ J_r^T \Sigma_Z^{-1} J_f (J_f^T \Sigma_Z^{-1} J_f)^{-1} J_f^T \Sigma_Z^{-1} (\mathbf{X}_f^Z - H^Z(\mathbf{X}_r)). \end{aligned} \tag{17}$$

From Lemma 1, J_f is a constant and J_r does not depend on $\mathbf{X}_f$, thus the coefficient matrix on the left hand side of (17) is independent of $\mathbf{X}_f$. Again from Lemma 1, back substitute $\mathbf{X}_f^Z = J_f \mathbf{X}_f$ to the second term on the right hand side in (17), we have

$$J_r^T \Sigma_Z^{-1} J_f (J_f^T \Sigma_Z^{-1} J_f)^{-1} J_f^T \Sigma_Z^{-1} J_f \mathbf{X}_f = J_r^T \Sigma_Z^{-1} \mathbf{X}_f^Z. \tag{18}$$

Thus, (17) can be rewritten as

$$J_r^T V_f J_r \Delta_r = J_r^T V_f H^Z(\mathbf{X}_r) \tag{19}$$

where

$$V_f = \Sigma_Z^{-1} - \Sigma_Z^{-1} J_f (J_f^T \Sigma_Z^{-1} J_f)^{-1} J_f^T \Sigma_Z^{-1}. \tag{20}$$

As (19) is independent of feature $\mathbf{X}_f$, we have proved that the update of robot poses Δ_r solved by (15) is independent of the features $\mathbf{X}_f$. Thus the updated robot pose $\mathbf{X}_r + \Delta_r$ is also independent of features $\mathbf{X}_f$.

After the update of robot poses Δ_r is calculated using (19), the updated features can be obtained by calculating the feature update Δ_f using Δ_r and (16)

$$\mathbf{X}_f + \Delta_f = \mathbf{X}_f + V^{-1}\mathbf{b}_f - V^{-1}W^T\Delta_r. \tag{21}$$

The last term $V^{-1}W\Delta_r$ on the right hand side of (21) is independent of features $\mathbf{X}_f$. The second term $V^{-1}\mathbf{b}_f$ can be rewritten as

$$\begin{aligned} &- (J_f^T \Sigma_Z^{-1} J_f)^{-1} J_f^T \Sigma_Z^{-1} (J_f \mathbf{X}_f - H^Z(\mathbf{X}_r)) \\ =& - \mathbf{X}_f + V^{-1} J_f^T \Sigma_Z^{-1} H^Z(\mathbf{X}_r). \end{aligned} \tag{22}$$

Back substitute (22) into (21), the updated features can be calculated by

$$\mathbf{X}_f + \Delta_f = V^{-1}(J_f^T \Sigma_Z^{-1} H^Z(\mathbf{X}_r) - W^T \Delta_r) \tag{23}$$

which is also independent of features $\mathbf{X}_f$.

Now, we have proved in each iteration, the result of both features and robot poses are independent of features. Thus the proof of Theorem 1 is completed. Q.E.D.

Now comes to our main result. Since the first term in (9) is the only term contains features $\mathbf{X}_f$, thus we can easily extend the proof of Theorem 1 to prove the following theorem for the NLLS SLAM problem in (9).

Theorem 2: In each step of GN iteration for minimising the NLLS SLAM objective function in (9), the result of robot poses and features is independent of the values of features $\mathbf{X}_f$ in the previous step.

Remark 2: It should be noted that, Theorem 2 holds only for the alternative formulation of NLLS SLAM problem in (9), not for the original formulation in (2), even when the covariances of feature observation noises are isotropic. In the original formulation of NLLS SLAM problem in (2), the Jacocian w.r.t. robot poses is dependent on features (see function $H^{Z_k^i}(\mathbf{X})$ in (4)). That means the similar statement as in Lemma 1 does not hold.

C. Linear Solution to Features

After the optimal solution of the robot poses $\hat{\mathbf{X}}_r$ is obtained, the SLAM problem in (11) becomes a linear least squares problem to minimise

$$\|F^Z(\mathbf{X}_f)\|^2_{\Sigma_Z^{-1}} = (J_f \mathbf{X}_f - H^Z(\hat{\mathbf{X}}_r))^T \Sigma_Z^{-1} (J_f \mathbf{X}_f - H^Z(\hat{\mathbf{X}}_r)). \tag{24}$$

Then, the optimal solution of features $\hat{\mathbf{X}}_f$ can be obtained by solving the linear system

$$V\hat{\mathbf{X}}_f = J_f \Sigma_Z^{-1} H^Z(\hat{\mathbf{X}}_r). \tag{25}$$

This step is the same as what has been proposed in many related work such as [25]. The foundamental reason is that when all the robot rotations are fixed, the SLAM problem of minimising (2) is a linear least squares problem which has a closed-form solution.

D. An Equivalent Pose-Only Algorithm

Applying Theorem 2, we can have an algorithm that only solve robot poses but is equivalent to the original GN SLAM algorithm in each step. The details are given in Algorithm 1.

As solving robot poses is independent of the features, we only need to initialise robot poses using odometry, and then update robot poses iteratively using (19). Since the objective function (9) is not evaluated in each iteration anymore, the algorithm is identified as converged when $\Delta_r(k)$ is close to zero. After the optimal solution of the robot poses $\hat{\mathbf{X}}_r$ is obtained, the optimal solution of features $\hat{\mathbf{X}}_f$ can be obtained by using (25).

Remark 3: It should be mentioned that, giving the data association, V_f in (20), and V and J_f in (25) are constant, which

Algorithm 1 An equivalent pose-only algorithm

```
Input: Feature measurements $\{Z^i_{k_{ij}}\}$, odometry $\{O_i\}$
Output: Optimal robot poses $\hat{\mathbf{X}}_r$ and features $\hat{\mathbf{X}}_f$
Initialise $\mathbf{X}_r(0)$ using $\{O_i\}$
Pre-calculate $V$, $J_f$ and $V_f$
for $k = 1$; $k <= \tau_k$ & $\|\Delta_r(k)\|^2 > \tau_{\Delta_r}$; $k++$ do
    Evaluate $J_r$, $H^Z(\mathbf{X}_r)$ at $\mathbf{X}_r(k)$
    Solve linear system (19) to get $\Delta_r(k)$
    Update robot poses $\mathbf{X}_r(k+1) = \mathbf{X}_r(k) \oplus \Delta_r(k)$
end for
Solve linear system (25) to get features $\hat{\mathbf{X}}_f$
```

can be calculated prior to the iteration. Thus, in each iteration, only the Jacobian w.r.t. robot poses J_r and $H^Z(\mathbf{X}_r)$ need to be evaluated using the current robot poses and observations $\mathbf{Z}$, which makes the algorithm efficient.

Since Algorithm 1 computes $\mathbf{X}_r$ using linear system (19) which is equivalent to (15), its result is exactly the same as the robot poses obtained from the GN method in every iteration step. The feature $\hat{\mathbf{X}}_f$ is computed from (25) after the algorithm converges, thus the final result of robot poses and features are identical to the final result of GN.

In summary, we have proved that "*for the GN iteration algorithm for solving the NLLS SLAM problem (9), there is an equivalent pose-only iteration algorithm that produces exactly the same results in each iteration step*".

V. Solving Rotations in SLAM is Independent of Translations

When the covariance matrices of the translation part of the odometry errors are also isotropic, we can prove that for GN SLAM algorithm, there is an equivalent rotation-only algorithm that can obtain exactly the same result in each iteration. Some details are given below.

When the covariance of odometry P_{O_i} has the format

$$P_{O_i} = \mathrm{diag}(P_{O_i^t}, P_{O_i^R}) \tag{26}$$

with $P_{O_i^t}$ being an isotropic positive definite matrix and $P_{O_i^R}$ being any positive definite matrix, the odometry term $\|H^{O_i}(\mathbf{X})\|^2_{P^{-1}_{O_i}}$ in (2) becomes

$$\begin{aligned} &\|O_i^t - R_{i-1}^T(t_i - t_{i-1})\|^2_{P^{-1}_{O_i^t}} + \|d_{SO}(O_i^R, R_{i-1}^T R_i)\|^2_{P^{-1}_{O_i^R}} \\ =\ &\|(t_i - t_{i-1}) - R_{i-1}O_i^t\|^2_{P^{-1}_{O_i^t}} + \|d_{SO}(O_i^R, R_{i-1}^T R_i)\|^2_{P^{-1}_{O_i^R}}. \end{aligned} \tag{27}$$

Thus, when $P_{Z^i_{k_{ij}}}$ and P_{O_i} are all isotropic, the objective function in NLLS SLAM formulation (2) can be written as

$$\begin{aligned} f_2(\mathbf{X}) =& \sum_{i=0}^{m}\sum_{j=1}^{n_i} \|(X_{f_{k_{ij}}} - t_i) - R_i Z^i_{k_{ij}}\|^2_{P^{-1}_{Z^i_{k_{ij}}}} \\ &+ \sum_{i=1}^{m} \|(t_i - t_{i-1}) - R_{i-1}O_i^t\|^2_{P^{-1}_{O_i^t}} \\ &+ \sum_{i=1}^{m} \|d_{SO}(O_i^R, R_{i-1}^T R_i)\|^2_{P^{-1}_{O_i^R}}. \end{aligned} \tag{28}$$

The second term in (28) is about the relative translation of the odometry, it has a very similar form to the first term about feature observation which is the same as that in (9). Thus, following the same line as the proofs of Theorem 1 and Theorem 2, we can prove the following theorem.

Theorem 3: In each step of GN iteration for minimising the NLLS SLAM objective function in (28), the result of robot poses and features is independent of the values of features $\mathbf{X}_f$ as well as the robot translations $\mathbf{X}_t = [t_1^T, ..., t_m^T]^T$ in the previous step.

Proof. Assume that no measurements of the orientations of the robots exist, the objective function of (28) can be simplified as:

$$\begin{aligned} f_3(\mathbf{X}) =& \sum_{i=0}^{m}\sum_{j=1}^{n_i} \|(X_{f_{k_{ij}}} - t_i) - R_i Z^i_{k_{ij}}\|^2_{P^{-1}_{Z^i_{k_{ij}}}} \\ &+ \sum_{i=1}^{m} \|(t_i - t_{i-1}) - R_{i-1}O_i^t\|^2_{P^{-1}_{O_i^t}}. \end{aligned} \tag{29}$$

We first prove that solving (29) by GN method, the result in each iteration is independent of features $\mathbf{X}_f$ and the robot translations $\mathbf{X}_t$ in the previous step.

By separating the rotations from translations and features in the state vector, let $\mathbf{X} = \{\mathbf{X}_R, \mathbf{X}_{tf}\}$, where $\mathbf{X}_R = \{R_1, \cdots, R_m\}$ contains all the rotations of the robot poses and $\mathbf{X}_{tf} = [\mathbf{X}_t^T, \mathbf{X}_f^T]^T$ contains all the translations of the robot poses $\mathbf{X}_t$ and the features $\mathbf{X}_f$; let $\mathbf{X}_{tf}^{ZO^t} = [..., (X_{f_{k_{ij}}} - t_i)^T, ..., (t_i - t_{i-1})^T, ...]^T$ contain all the corresponding translations and features in the state $\mathbf{X}_{tf}$ w.r.t. the feature observation vector $\mathbf{Z}$ and the translation of odometry measurement vector $\mathbf{O}^t = [..., (O_i^t)^T, ...]^T$; $H^{ZO^t}(\mathbf{X}_R) = [..., (R_i Z^i_{k_{ij}})^T, ..., (R_{i-1}O_i^t)^T, ...]^T$; and $\Sigma^{-1}_{ZO^t} = \mathrm{diag}(..., P^{-1}_{Z^i_{k_{ij}}}, ..., P^{-1}_{O_i^t}, ...)$ be the weight which defined by combining all the covariance matrices of feature measurements and translation of odometry measurements. The least squares optimisation problem in (29) is to seek $\mathbf{X}$ such that

$$f_3(\mathbf{X}) = \|F^{ZO^t}(\mathbf{X})\|^2_{\Sigma^{-1}_{ZO^t}} = (F^{ZO^t}(\mathbf{X}))^T \Sigma^{-1}_{ZO^t} F^{ZO^t}(\mathbf{X}) \tag{30}$$

is minimised, where

$$F^{ZO^t}(\mathbf{X}) = \mathbf{X}_{tf}^{ZO^t} - H^{ZO^t}(\mathbf{X}_R). \tag{31}$$

Similar method as shown in Section IV-B can be used to solve (30), the linear equation in each GN iteration can be rewritten as

$$\begin{bmatrix} \bar{U} & \bar{W} \\ \bar{W}^T & \bar{V} \end{bmatrix} \begin{bmatrix} \Delta_R \\ \Delta_{tf} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_R \\ \mathbf{b}_{tf} \end{bmatrix}, \tag{32}$$

where Δ_R and Δ_{tf} are the rotation and translation-feature parts of the update of state vector $\mathbf{X}$. $\bar{U} = J_R^T \Sigma_{ZO^t}^{-1} J_R$, $\bar{V} = J_{tf}^T \Sigma_{ZO^t}^{-1} J_{tf}$, $\bar{W} = J_R^T \Sigma_{ZO^t}^{-1} J_{tf}$, $\mathbf{b}_R = -J_R^T \Sigma_{ZO^t}^{-1} F^{ZO^t}(\mathbf{X})$, and $\mathbf{b}_{tf} = -J_{tf}^T \Sigma_{ZO^t}^{-1} F^{ZO^t}(\mathbf{X})$, where $J_R = -\partial H^{ZO^t} / \partial \mathbf{X}_R$ and $J_{tf} = \partial \mathbf{X}_{tf}^{ZO^t} / \partial \mathbf{X}_{tf}$ are the Jacobian matrices of $F^{ZO^t}(\mathbf{X})$ w.r.t. the rotation and translation-feature respectively. Similar to Lemma 1, J_{tf} is also constant and $\mathbf{X}_{tf}^{ZO^t} = J_{tf} \mathbf{X}_{tf}$. Therefore, using Schur complement, the update of robot rotation can be solved by

$$J_R^T V_{tf} J_R \Delta_R = J_R^T V_{tf} H^{ZO^t}(\mathbf{X}_R), \tag{33}$$

where

$$V_{tf} = \Sigma_{ZO^t}^{-1} - \Sigma_{ZO^t}^{-1} J_{tf} (J_{tf}^T \Sigma_{ZO^t}^{-1} J_{tf})^{-1} J_{tf}^T \Sigma_{ZO^t}^{-1}. \tag{34}$$

In addition, similar to 23, the updated robot translations and features can be calculated as

$$\mathbf{X}_{tf} + \Delta_{tf} = \bar{V}^{-1} (J_{tf}^T \Sigma_{ZO^t}^{-1} H^{ZO^t}(\mathbf{X}_R) - \bar{W}^T \Delta_R). \tag{35}$$

Since both (33) and the right hand side of (35) are independent of $\mathbf{X}_{tf}$, in each step of GN iteration for minimizing (29), the result is independent of the values of features $\mathbf{X}_f$ and the robot translations $\mathbf{X}_t$ in the previous step. The conclusion can be extended to the NLLS problem in (28) since the third term in (28) is independent of features and the robot translations. Therefore, the proof of Theorem 3 is completed. Q.E.D.

After the optimal solution of the robot rotation $\hat{X}_R$ is obtained, the SLAM problem in (29) becomes a linear least squares problem and the optimal solution of robot translations and features can be obtained by solving the linear system

$$\bar{V} \hat{\mathbf{X}}_{tf} = J_{tf} \Sigma_{ZO^t}^{-1} H^{ZO^t}(\hat{\mathbf{X}}_R). \tag{36}$$

Therefore, similar to Algorithm 1, when the covariance matrices of both feature observations and odometry translations are isotropic, we can have a "rotation-only algorithm" that only solves the rotations of the robot poses $\mathbf{X}_R = \{R_1, ..., R_m\}$, and its results in each iteration step are identical to that of GN for solving (28). The "rotation-only algorithm" is summarised in Algorithm 2.

VI. Experimental Results

In this section, the theorems proposed in this paper will be numerically demonstrated using different 2D and 3D feature-based SLAM datasets. Then, the proposed pose-only algorithm and rotation-only algorithm are validated and compared to the NLLS SLAM solved using GN iterations.

Algorithm 2 An equivalent rotation-only algorithm

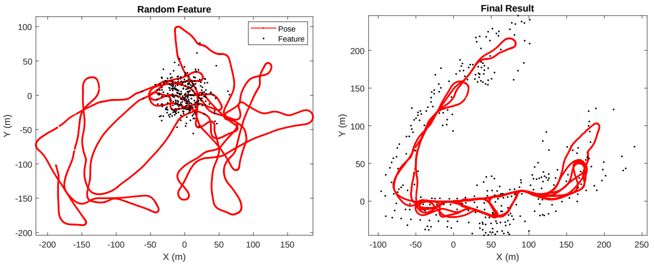
```
Input: Feature measurements $\{Z_{k_{ij}}^i\}$, odometry $\{O_i\}$
Output: Optimal robot rotations $\hat{\mathbf{X}}_R$, robot positions $\hat{\mathbf{X}}_t$ and features $\hat{\mathbf{X}}_f$
Initialise $\mathbf{X}_R(0)$ using $\{O_i^R\}$
Pre-calculate $\bar{V}$, $J_{tf}$ and $V_{tf}$
for $k = 1; k <= \tau_k \ \& \ \|\Delta_R(k)\|^2 > \tau_{\Delta_R}; k++$ do
    Evaluate $J_R$, $H^{ZO^t}(\mathbf{X}_R)$ at $\mathbf{X}_R(k)$
    Solve linear system (33) to get $\Delta_R(k)$
    Update robot rotations $\mathbf{X}_R(k+1) = \mathbf{X}_R(k) \oplus \Delta_R(k)$
end for
Solve linear system (36) to get robot positions $\hat{\mathbf{X}}_t$ and features $\hat{\mathbf{X}}_f$
```

A. 2D Feature-Based SLAM

For 2D feature-based SLAM, the Victoria Park dataset [9] and DLR dataset [15] are used. Identity matrices are used as the covariance of errors of feature measurements and odometry, to make the covariance used in the SLAM algorithms isotropic. For GN based NLLS SLAM, the initialisation is done by first initialising robot poses using odometry, and then using the initialised poses and the first observation of each feature to initialise feature positions as shown in Fig. 3(a) and Fig. 4(a).

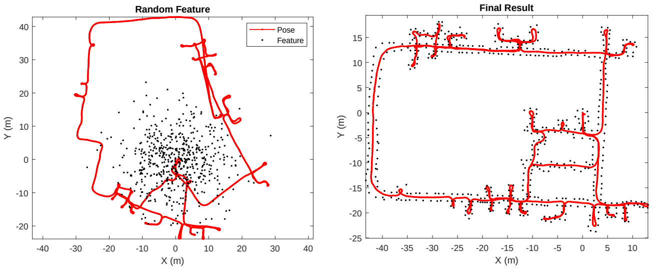
1) NLLS SLAM with Random Features: GN for NLLS SLAM using the alternative formulation described in (9) (*Full-NLLS*) is first implemented. To validate Theorem 2, we also modify the algorithm by resetting the features in the state vector as random values at the beginning of each iteration. This modified algorithm is called *Random-F*. Examples of the random features used are shown in Fig. 1(a) for Victoria Park dataset and Fig. 2(a) for DLR dataset. Visual inspection shows that after the state is updated in each iteration, the poses and features obtained in *Random-F* are the same as the ones obtained from *Full-NLLS* at the same iteration (see the video). The final results of *Random-F* are shown in Fig. 1(b) and Fig. 2(b), which are the same as those from *Full-NLLS* (Fig. 3(b) and Fig. 4(b)).

To quantatatively confirm the results in each iteration are the same for *Random-F* and *Full-NLLS*, we compare the objective function values and the squares of 2-norm of the update of robot poses $\|\Delta_r\|^2$ at each iteration of the two algorithms. For *Random-F*, we evaluate the objective function at the end of each iteration after the state vector is updated. And then features are reset to random values to be used in the next iteration. Here we only compare the update of robot poses, since the updates of features from the two algorithms are suppose to be different because of the random features used in *Random-F*. As we can see from Fig. 5, for both Victoria Park and DLR datasets, the objective function values and the squares of 2-norm of the update of robot poses for *Random-F* at each iteration are exactly the same as the ones for *Full-NLLS*. Thus, we can see from the results that giving random features at each iteration does not change the result



(a) Random Feature (b) *Random-F* Result

Fig. 1. NLLS SLAM with random features at the beginning of each iteration (*Random-F*) using Victoria Park dataset: (a) An example of random features and (b) *Random-F* result. Red lines are the robot trajectories and black dots are the features.



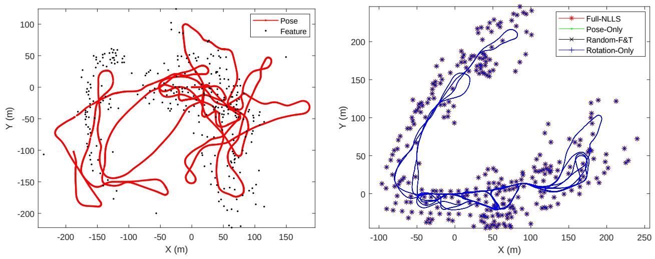
(a) Random Feature (b) *Random-F* Result

Fig. 2. NLLS SLAM with random features at the beginning of each iteration (*Random-F*) using DLR dataset.

at all, which demonstrates the independence result described in Theorem 2.

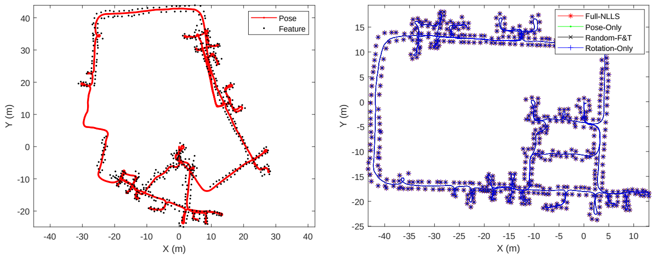
2) The Proposed Pose-Only Algorithm: The proposed algorithm which only solves robot poses (*Pose-Only*) as shown in Algorithm 1 is also implemented. The results using Victoria Park and DLR datasets are compared to *Full-NLLS* and *Random-F*. Since features are not considered in *Pose-Only*, only robot poses are in the state vector and it can be initialised by just using odometry. For *Pose-Only*, the features in the results are computed by the linear method in (25) after the optimal robot poses are obtained. The final results obtained from *Full-NLLS*, *Random-F* and *Pose-Only* are are exactly the same as shown in Fig. 3(b) and Fig. 4(b).

The objective function value and the squares of 2-norm of the update of robot poses at each iteration from *Pose-Only* are also compared to those from *Full-NLLS* and *Random-F*. Here, since the objective function does not need to be evaluated in *Pose-Only* (see Algorithm 1) and features are not in the state vector, for comparison purpose, in each iteration, after the robot poses are updated, we compute the linear solution to the features as described in (25) by considering the updated poses as constant, and then compute the objective function in (9) using both robot poses and features. The features are only used in this way to calculate the objective function value at each iteration. They are not needed in *Pose-Only* algorithm. As shown in Fig. 5, the squares of 2-norm of the updates of robot poses at each iteration for *Full-NLLS*, *Random-F* and *Pose-Only* are exactly the same, which numerically validate that *Pose-Only* is equivalent to *Full-NLLS* in (9). The objective function value at each iteration for *Pose-Only* is different from (slightly smaller than) those for the other two algorithms. This is because the features obtained by using the linear method in (25) is optimal w.r.t. the updated robot poses at each iteration, which is similar to the method proposed in [13]. Thus, the features calculated in this way are different from those in the state vector in *Full-NLLS* at the beginning, but will become the same when the algorithms converge.



(a) Initial Guess (b) All the results are the same

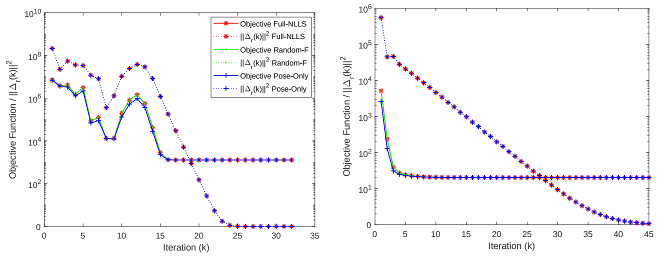
Fig. 3. Comparison of different algorithms using Victoria Park dataset: (1) *Full-NLLS*; (2) Proposed pose-only algorithm (*Pose-Only*); (3) NLLS with random feature and random robot positions (*Random-F&T*); and (4) Proposed rotation-only algorithm (*Rotation-Only*). Note that the last step of *Pose-Only* (*Rotation-Only*) is to compute the features (and robot positions) by solving a linear system.



(a) Initial Guess (b) All the results are the same

Fig. 4. Comparison of different algorithms using DLR dataset: (1) *Full-NLLS*; (2) Proposed pose-only algorithm (*Pose-Only*); (3) NLLS with random feature and random robot positions (*Random-F&T*); and (4) Proposed rotation-only algorithm (*Rotation-Only*).

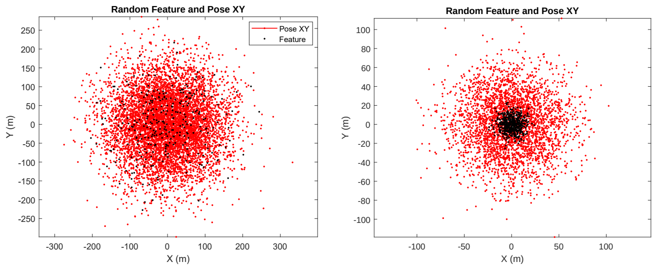
3) SLAM with Random Features and Random Robot Positions, and Rotation-Only Algorithm: To validate Theorem 3, the GN method for the NLLS SLAM defined in (28) (*Full-NLLS*) is implemented but given random features and random robot positions at the beginning of each iteration (*Random-F&T*). Examples of random features and random robot positions used are shown in Fig. 6. The results of Victoria Park dataset and DLR dataset, together with the objective function and the squares of 2-norm of the update of robot rotation $\|\Delta_\Phi\|^2$ at each iteration are shown in Fig. 3(b), Fig. 4(b) and Fig. 8, respectively. The rotation-only algorithm proposed in Section V which only solves a reduced robot rotation system (without considering features and robot positions in the state vector) as shown in Algorithm 2 is also implemented (*Rotation-Only*) and validated using the two



(a) Victoria Park Dataset

(b) DLR Dataset

Fig. 5. The objective fucntion and update of robot poses at each iteration for (1) GN SLAM (*Full-NLLS*), (2) NLLS with random features (*Random-F*) and (3) proposed pose-only algorithm (*Pose-Only*): (a) Victoria Park dataset and (b) DLR dataset. The objective fucnion does not need to be evaluated in *Pose-Only* and only calculated for comparison. The update of robot poses at each iteration for different algorithms are exactly the same.



(a) Victoria Park Dataset

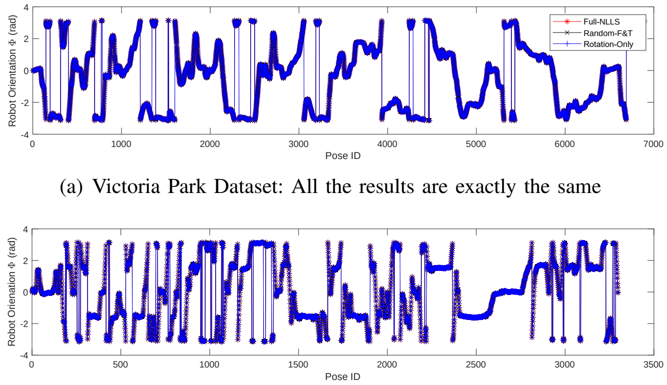
(b) DLR Dataset

Fig. 6. Examples of random features and random robot positions used before each iteration in *Random-F&T*.

datasets. Similar to *Pose-Only*, the final result from *Rotation-Only* is obtained by first optimising the robot rotations and then calculating the features and robot positions linearly by using the optimal robot rotations. Results from both *Random-F&T* and *Rotation-Only* are exactly the same as the one obtained from *Full-NLLS* for (28). And similar to the results in Section VI-A2, the squares of 2-norm of the update of robot rotation at each iteration for all these three algorithms are the same, with the objective function value of *Rotation-Only* smaller than those of the other two algorithms before the algorithms converge (see Fig. 8).

B. 3D Feature-Based SLAM

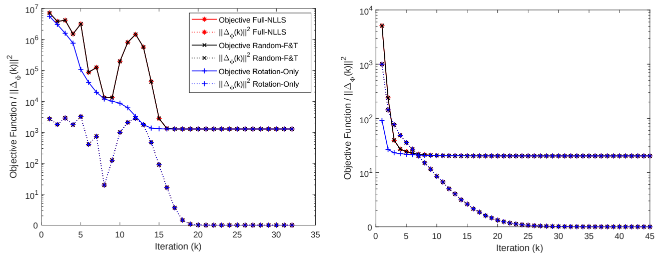
For 3D feature-based SLAM, a simulated dataset is used with a trajectory of 870 poses and uniformly distributed features in the environment as used in [26]. Both *Full-NLLS* in (9) and the proposed *Pose-Only* algorithm (for 3D scenario) are performed. The initial guess obtained from odometry and first feature observations for *Full-NLLS* is shown in Fig. 9(a). The same initial guess of robot poses is used for *Pose-Only*. And the final results from both algorithms are shown in Fig. 9(b). The obtained poses and features from *Full-NLLS* are exactly the same as the optimal poses obtained from *Pose-Only* and the optimal features obtained from the linear method (25) using the optimal poses. We further compared the robot poses at each iteration from the two algorithms. The maximum absolute difference between the two sets of robot poses at the same iteration is less than 10^{-9}, which means the robot poses obtained from *Full-NLLS* are exactly the same as the ones from *Pose-Only* at each iteration. These validate Theorem 2 in 3D scenario and validate that only solving robot poses using *Pose-Only* in the 3D scenario is also equivalent to *Full-NLLS*.



(a) Victoria Park Dataset: All the results are exactly the same

(b) DLR Dataset: All the results are exactly the same

Fig. 7. Robot rotation results from different SLAM algorithms: (1) *Full-NLLS*; (2) NLLS with random features and random robot positions (*Random-F&T*); and (3) The proposed rotation-only algorithm (*Rotation-Only*).



(a) Victoria Park Dataset

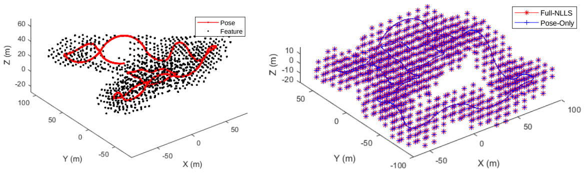
(b) DLR Dataset

Fig. 8. The objective fucntion values and the squares of 2-norm of the update of robot rotation at each iteration for (1) GN NLLS SLAM (*Full-NLLS*), (2) NLLS with random features and robot positions (*Random-F&T*), and (3) proposed rotation-only algorithm (*Rotation-Only*). The squares of 2-norm of the update of robot rotations at each iteration for different algorithms are exactly the same.

VII. Discussions

In this section, we discuss the relations between the results of this paper and some related results in the literature.

In [10], some surprising convergence results have been presented for 2D GN SLAM algorithm (for SLAM problem minimising (2)) with Victoria Park dataset when isotropic covariance matrices are used. It has been empirically shown that starting from random initial values for robot poses and features, the algorithm can converge to the correct solution 80% of the time. In this paper, what we have proved is that as long as the initial values of robot rotations used are the same, the GN SLAM algorithm (for minimising (28)) with arbitrarily reset robot translations and feature positions in each step will generate exactly the same results, since the GN results are independent of the robot translations and feature positions. Our conclusion holds for both 2D and 3D cases.



(a) Initial Guess (b) The two results are the same

Fig. 9. Comparison of different algorithms using simulated 3D dataset: (1) *Full-NLLS*; (2) Proposed pose-only algorithm (*Pose-Only*).

Please note although the objective functions in (2) and (28) are mathematically equal (when covariance matrices are isotropic), the two GN algorithms for solving these two problems behave differently because of the different nonlinear structures. As a result, the independence property proved in this paper does not hold for GN SLAM algorithm that minimising (2).

In [24][25], it has been shown that the m-step 2D feature-based SLAM problems can be obtained by solving a NLLS problem over only m variables (i.e., the robot orientations in m poses). However, when marginalising the variables of robot positions and feature positions, the resulting low dimensional NLLS problem of robot orientations is very complicated and no longer sparse, although the low dimensional NLLS problem is equivalent to the original NLLS problem. This makes the solving of the reduced dimensional problem very inefficient except for very small problems such as 1-step and 2-step SLAM problems. In this paper, we use formula derived from Schur complement to compute the same rotations as those in the GN algorithm in a much more efficient way. That is, the results of the proposed rotation-only algorithm is identical to that of GN for solving problem (28) in each iteration step. Thus we achieve dimension reduction at the algorithm level instead of the problem level as in [25]. Again, our results hold for both 2D and 3D.

In [13], the separable structure of SLAM is utilised in a different way. In each iteration of SLAM solver such as g2o, the whole state vector including both the robot poses and feature positions are solved using standard operation (such as in GN). Then, in some of the selected steps, the obtained robot positions and feature positions are ignored and replaced by the robot positions and feature positions calculated using the obtained robot rotations, in order to generate a better state for the next iteration. In this paper, we only compute the robot rotations using a formula derived from Schur complement, which is shown to be independent of the robot position and feature values. After obtaining the optimal robot rotations, we compute the optimal robot positions and feature positions only once (as in Algorithm 2).

In some modern algorithms for solving pose-graph SLAM such as [21] and planar pose-graph and feature-based SLAM such as [7], the original SLAM problem is first reformulated as a rotation only problem, and then semidefinite relaxation and Riemannian staircase optimisation are applied. Similarly, in the proof of the region of attraction for GN in [5], the pose-graph SLAM problem is also transferred into an orientation only problem first to facilitate the analysis. This is closely related to our finding that "solving rotations is independent of translations". Thus we believe the underlining property of the SLAM with isotropic covariance matrices has been implicitly used to generate some of the state-of-the-art results.

VIII. Conclusions and Future Works

In this paper, we have proved some interesting properties for 2D and 3D point feature-based SLAM with isotropic covariance matrices. First, when the covariance matrices for feature observation errors are isotropic, we prove that when solving a reformulated nonlinear least squares SLAM problem, the result in each Gauss-Newton (GN) iteration is completely independent of the feature values. Thus a pose-only algorithm is proposed to generate the same results as the original GN method. Second, when the covariance matrices for the translation part of the odometry errors are also isotropic, we prove that the result in each GN iteration is also completely independent of the robot translations. Thus, a rotation-only algorithm can be used to generate the same results as the original GN method. Experimental results using 2D and 3D datasets confirmed the properties.

These new properties provide us more insights into the fundamental nonlinear structure of the SLAM problems with isotropic covariance matrices. In this paper, we have focused on clearly presenting these properties to the SLAM community for further investigations and exploiting, and leave the development of efficient SLAM solvers using these properties to our future work.

We are also in the process of analysing other related problems with the aim of identifying similar properties, such as pose-graph SLAM, occupancy grid map based SLAM problems, multiple 3D point sets registration problems, etc. We are expecting to see some good applications of these properties in robotics in the near future.

References

[1] GTSAM. https://bitbucket.org/gtborg/gtsam.

[2] S. Agarwal, K. Mierle, and Others. Ceres solver. http://ceres-solver.org.

[3] J. Briales and J. Gonzalez-Jimenez. Fast and global SE (d)-synchronization. *IEEE Robotics and Automation Letters*, 2(4):2127–2134, 2017.

[4] C. Cadena, L Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J.J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[5] L. Carlone. Convergence analysis of pose graph optimization via gauss-newton methods. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 965–972, 2013.

[6] F. Dellaert and M. Kaess. Square Root SAM: Simultaneous localization and mapping via square root information smoothing. *The International Journal of Robotics Research*, 25(12):1181–1204, 2006.

[7] T. Fan, H. Wang, M. Rubenstein, and T. Murphey. CPL-SLAM: Efficient and certifiably correct planar graph-based SLAM using the complex number representation. *IEEE Transactions on Robotics*, 36(6):1719–1737, 2020.

[8] G. Grisetti, S. Grzonka, C. Stachniss, P. Pfaff, and W. Burgard. Efficient estimation of accurate maximum likelihood maps in 3D. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3472–3478, 2007.

[9] 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(3):242–257, 2001.

[10] S. Huang, Y. Lai, U. Frese, and G. Dissanayake. How far is SLAM from a linear least squares problem? In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3011–3016, 2010.

[11] S. Huang, H. Wang, U. Frese, and G. Dissanayake. On the number of local minima to the point feature based SLAM problem. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2074–2079, 2012.

[12] V. Ila, L. Polok, M. Solony, and P. Svoboda. SLAM++-a highly efficient and temporally scalable incremental slam framework. *The International Journal of Robotics Research*, 36(2):210–230, 2017.

[13] K. Khosoussi, S. Huang, and G. Dissanayake. A sparse separable slam back-end. *IEEE Transactions on Robotics*, 32(6):1536–1549, 2016.

[14] R. Kuemmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3607–3613, 2011.

[15] J. Kurlbaum and U. Frese. A benchmark data set for data association. Technical Report, University of Bremen, Data available online:. http://radish.sourceforge.net/.

[16] M.I. Lourakis and A.A. Argyros. SBA: A software package for generic sparse bundle adjustment. *ACM Transactions on Mathematical Software*, 36(1):2:1–2:30, 2009.

[17] J.G. Mangelson, J. Liu, R.M. Eustice, and R. Vasudevan. Guaranteed globally optimal planar pose graph and landmark SLAM via sparse-bounded sums-of-squares programming. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 9306–9312, 2019.

[18] Z. Mao, L. Zhao, S. Huang, Y Fan, and A. Lee. Direct simultaneous multi-image registration. *ArXiv*, abs/2105.10087, 2021.

[19] S.M. Nasiri, R. Hosseini, and H. Moradi. Novel parameterization for gauss–newton methods in 3-D pose graph optimization. *IEEE Transactions on Robotics*, 2020.

[20] E. Olson, J. Leonard, and S. Teller. Fast iterative optimization of pose graphs with poor initial estimates. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2262–2269, 2006.

[21] D.M. Rosen, L. Carlone, A.S. Bandeira, and J.J. Leonard. Se-sync: A certifiably correct algorithm for synchronization over the special euclidean group. *The International Journal of Robotics Research*, 38(2-3):95–125, 2019.

[22] B. Triggs, P.F. McLauchlan, R.I. Hartley, and A.W. Fitzgibbon. Bundle adjustment — a modern synthesis. In *Proceedings of the International workshop on Vision Algorithms*, pages 298–372, 2000.

[23] H. Wang, G. Hu, S. Huang, and G. Dissanayake. On the structure of nonlinearities in pose graph SLAM. In *Proceedings of Robotics: Science and Systems*, 2012.

[24] H. Wang, S. Huang, U. Frese, and G. Dissanayake. The nonlinearity structure of point feature SLAM problems with spherical covariance matrices. *Automatica*, 49(10): 3112–3119, 2013.

[25] H. Wang, S. Huang, K. Khosoussi, U. Frese, G. Dissanayake, and B. Liu. Dimensionality reduction for point feature SLAM problems with spherical covariance matrices. *Automatica*, 51:149–157, 2015.

[26] L. Zhao, S. Huang, and G. Dissanayake. Linear SLAM: Linearising the slam problems using submap joining. *Automatica*, 100:231–246, 2019.