Anchor selection for SLAM based on graph topology and sub-modular optimization

Yongbo Chen, Liang Zhao, Yanhao Zhang, Shoudong Huang, and Gamini Dissanayake

***Abstract*—This paper considers simultaneous localization and mapping (SLAM) problem for robots in situations where accurate estimates for some of the robot poses, termed anchors, are available. These may be acquired through external means, for example, by either stopping the robot at some previously known locations or pausing for a sufficient period of time to measure the robot poses with an external measurement system. The main contribution is an efficient algorithm for selecting a fixed number of anchors from a set of potential poses, that minimizes estimated error in the SLAM solution. Based on a graph-topological connection between the D-optimality design metric and the tree-connectivity of the pose-graph, the anchor selection problem can be formulated approximately as a sub-matrix selection problem for reduced weighted Laplacian matrix, leading to a cardinality-constrained sub-modular maximization problem. Two greedy methods are presented to solve this sub-modular optimization problem with a performance guarantee. These methods are complemented by Cholesky decomposition, approximate minimum degree permutation, order re-use, and rank-1 update that exploit the sparseness of the weighted Laplacian matrix. We demonstrate the efficiency and effectiveness of the proposed techniques on public-domain datasets, Gazebo simulations, and real-world experiments.**

***Index Terms*—SLAM, Anchor selection, Reduced weighted Laplacian matrix, Sub-modular optimization, Rank-1 update**

I. Introduction

MANY state estimation problems, such as simultaneous localization and mapping (SLAM) [1], point cloud registration [2], and structure from motion (SFM) [3], can be formulated as two/three dimensional (2D/3D) *pose-graph optimization*, in which the variables to be estimated are poses sampled along the robot trajectory, and each edge imposes a noisy relative measurement on a pair of poses [4]. In traditional SLAM frameworks, robot poses and surrounding maps are typically estimated with respect to the robot starting position, which is regarded as the origin of the world coordinate frame. In other words, only the first pose contains global ground truth information. As the robot travels away from the origin, the estimation error accumulates at a rate that depends on sensor accuracy, frequency of loop closure, trajectory length, and so on. However, if the ground truths of some poses, which we refer as anchors[1], are available, the accuracy of the estimated parameters will be significantly improved, because the intuitive benefits of the anchors are similar to introducing the 'zero-uncertainty' loop closures between the anchors and the origin. Practically, this can be achieved in a number of external means. In indoor environments, it may be possible to stop the robot at previously known places to obtain an accurate global robot pose. Outdoors, it may be possible to pause for a sufficient period of time to obtain an accurate Global Positioning System (GPS) fix[2]. Hence, adding anchors is a practical and efficient way to strengthen current SLAM frameworks.

The idea of introducing multiple anchors is necessary especially in certain tasks that require high-quality mapping and localization. The electronic total station (ETS) is a conventional surveying instrument used for mapping. For the same environment, even with the state-of-the-art SLAM methods and the most accurate on-board sensors, it is still extremely difficult to achieve the typical accuracy of the ETS ($\pm(1.5\text{mm}+2\text{ppm})$ in position and between $0.5'$ and $1'$ in orientation in ranges longer than 1.5km [6])[3]. Using predefined landmarks or mechanical location structure set by external manual measurements like ETS, intuitively, at least the accuracy of the poses, which observe these landmarks, will be greatly improved.

In fact, some researchers have already applied similar ideas to improve the accuracy of mapping and localization in some competitions and industrial projects. For example, the well-known DARPA Subterranean Challenge [7] is a competition requiring multiple robots, including both drones and ground vehicles, where the aim is to accomplish a variety of mapping, navigation, and search missions in man-made tunnels, natural caves, and underground structures. Because the environment is mostly featureless with many self-similar structures and the task spans several kilometers, this competition is challenging and may cause a large drift using the traditional SLAM methods. To solve this, the team from California Institute of Technology uses several calibrated base stations in the tunnel corners to fuse to the SLAM method thus achieving reliable and highly-accurate multiple robots navigation [8]. Another

Manuscript received October 4, 2020; revised February 6, 2021; accepted April 7, 2021. Date of publication; date of current version. This work was supported by the Australia Research Council (ARC) Discoverygrant deformable SLAM Project DP120102786. This paper was recommendedfor publication by Associate Editor and Editor F. Chaumette upon evaluation of the reviewers' comments. (Corresponding author: Yongbo Chen.)

The authors are with Centre for Autonomous Systems, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: Yongbo.Chen@uts.edu.au, Liang.Zhao@uts.edu.au, Yanhao.Zhang@student.uts.edu.au, Shoudong.Huang@uts.edu.au, Gamini.Dissanayake@uts.edu.au).

This article has supplementary material provided by the authors and colorversions of one or more figures available at https://doi.org.

Digital Object Identifier.

[1] Here an anchor means a pose with known global location and orientation. Different from the origins in multiple sub-maps [5], the anchored poses considered in this paper have their global information, which brings more information to the whole SLAM system.

[2] In this paper, the anchors are defined as the known poses (positions and orientations). It is also possible to extend our framework to the cases when only some positions or orientations of the poses are known.

[3] The symbol $'$ means the unit of the degree, minute.

example is the Alibaba autonomous car, also called "Road Reforming", which uses many intelligent perception base stations to help the localization and planning tasks for low-cost on-board sensors [9]. In short, the anchors and the similar technologies have shown some outstanding performances in improving the robustness and accuracy of the SLAM method.

Clearly, the number and the geometric placement of the anchors have an impact on the estimation accuracy of the SLAM framework. The external measurements of the anchors often require substantial manual operations (e.g. applying ETS systems), high-costs (e.g. building base stations), or affecting the original tasks (e.g. stopping at some place for a long time to achieve GPS fix). Therefore, it is desirable to reduce the number of anchors as much as possible while maintaining the estimation accuracy. Naturally, this leads to the following two important questions regarding *anchor selection* problem. (1) *Given a set of poses that can potentially serve as anchors, how to select the best sub-set of anchors with a given number that results in the most accurate SLAM solution?* (2) *How many anchors are sufficient (and where to put them) in order to achieve a certain level of accuracy?* This paper focuses on the first question in the context where the SLAM problem is formulated as a 2D/3D pose graph optimization. The second question will be considered in the future.

The Theory of Optimal Experimental Design (TOED), including A-, D-, E-, and T-optimality, on the Fisher information matrix (FIM) is widely used to assess the level of the uncertainty of the estimated parameters in SLAM [10], [11]. Even though the monotonicity of the optimal design metrics depends on the representation of uncertainty and the representation of the orientation of the robot pose, the D-optimality metric, which has been shown to be the most popular metric, preserves monotonicity in most commonly-used representations [12]. Thus, the anchor selection problem in this paper is formulated as that of maximizing the D-optimality metric of the FIM.

Recent works [13], [14] show that the D-/T-optimality metrics of 2D/3D pose-graph SLAM have a close relationship with the graph topology (tree-connectivity and weighted node degree) of the measurement network. In [14], regarding the pose graph as the network, the sub-modular optimization method has been applied to select the relative pose measurements (edges) to get more accurate SLAM results. These operations are similar to the relative information marginalization without adding additional global information. In our paper, instead of selecting the edges, the optimal or sub-optimal anchors are selected to add the global information (from the selected anchors) to improve the SLAM accuracy. Some connections and differences between the edge selection and the anchor selection will be discussed in Section IV-C.

Because of the widely applications of pose graph, the anchor selection method can be used in the high-accurate 2D/3D mapping and the collaborative SLAM (CSLAM) tasks to improve the accuracy of the estimation result. For the landmark setting in the 2D/3D mapping task[4], it is achieved by two different approaches: twice-trajectory approach and redundant landmarks approach. The twice-trajectory approach tries to execute the same path twice. The first iteration is used to build the initial anchor-free pose graph, select the optimal anchors, and set the landmarks. Based on the selected anchors, the second iteration is executed by following the same trajectory again with the aid of the calibrated landmarks. The redundant landmarks approach is to place many unknown marks along the trajectory, implement the SLAM method along the path only once, generate the pose graph, select the anchors, and pick out the corresponding marks to be calibrated. After the calibrating of the selected marks, the mark information (mark position/pose information) can be used in the SLAM algorithm to compute a more accurate SLAM result without executing the trajectory twice. For the CSLAM task, viewing all pose graphs as a whole, the initial poses of the different robots can be regarded as the anchors and selected based on the anchor selection method. More details are given in Section VI.

A. Contributions

This paper presents a novel sub-modular based method for the problem of selecting a fixed number of anchored poses in pose graph SLAM to maximize the D-optimality design criterion. The main contributions of this paper are listed below.

- **Graph topology**: The approximation from the D-optimality metric based anchor selection problem to the sub-matrix selection of the weighted Laplacian matrix, which greatly reduces its computational complexity and makes it computationally solvable.
- **Sub-modular**: The proof of the non-negative non-normalized non-monotone sub-modular property of the sub-matrix selection problem.
- **Solution**: The efficient greedy-based methods, using lazy evaluation, Cholesky decomposition, approximate minimum degree permutation (AMDP), order re-use, and rank-1 update techniques, which achieve better running-time ability compared with the standard greedy method.
- **Performance bound**: The performance bound, which compares the obtained solution with the optimal solution, offers a certifiable tool to know the quality of the solution.
- **Application**: The applications of the anchor selection framework in the 2D/3D mapping task and the CSLAM trajectory assignment task.

B. Outline

The rest of the paper is organized as follows: In Section II, we review the related works on the pose-graph SLAM with one and multiple anchors, the TOED, and the sub-modularity. The pose-graph SLAM with multiple anchors and its related FIM are formulated in Section III. Based on an approximation of the D-optimality metric, the anchor selection problem is

[4] The landmark presented in this paper means a pre-measured and observable marked feature with the known global location. In the redundant landmarks approach, before calibration, the positions/poses of the initial introduced marks are unknown, and then they will be known after using our proposed method and the additional calibration.

transformed into a sub-matrix selection problem, which is proved to be a non-normalized non-negative non-monotone sub-modular optimization problem over a cardinality-fixed constraint, in Section IV. In Section V, the normal and random greedy methods are applied to obtain a near-optimal solution with a performance guarantee. By exploiting the matrix sparseness, a high-efficiency lazy-greedy-based method, using Cholesky decomposition, AMDP, order re-use, and rank-1 update technologies, is presented to quickly select the anchored poses. Furthermore, some additional application scenarios, including the landmark setting in 2D/3D mapping and the CSLAM trajectory assignment with known initial positions, are presented in Section VI. Simulation and experimental results are presented to illustrate the proposed technique in Section VII and Section VIII respectively. Conclusions and future work are presented in Section IX.

C. Notations

Throughout this paper, unless otherwise stated, bold lowercase, and bold uppercase letters are reserved for vectors and matrices, respectively. Sets are shown by uppercase letters. $\boldsymbol{S}_1 \succeq \boldsymbol{S}_2$ means matrix $\boldsymbol{S}_1 - \boldsymbol{S}_2$ is positive semidefinite. The Kronecker product is denoted by $\otimes$. $\det(\star)$ and $tr(\star)$ represent the determinant and the trace of the matrix $\star$, respectively. $SO(n)$ (special orthogonal group) is defined as: $SO(n) \triangleq \{\boldsymbol{R} \in \mathbb{R}^{n\times n} : \boldsymbol{R}^\top \boldsymbol{R} = \boldsymbol{I}_{n\times n},\ \det(\boldsymbol{R}) = 1\}$. The squared vector norm is $\| \star \|_2^2 = \star^\top \cdot \star$ for a vector $\star$. $|\star|$ means the cardinality of the set $\star$. $\star \times \bullet$ means the direct product group of the groups $\star$ and $\bullet$.

II. Related work

The classical pose-graph SLAM with one anchor (the first robot pose) is a difficult non-convex optimization problem, of which the (globally optimal) solution is the maximum-likelihood (ML) estimate for the unknown poses [15]. The traditional approach to solve pose-graph SLAM is to use highly-efficient iterative nonlinear optimization methods such as Gauss-Newton (GN) [16], Levenberg-Marquardt (LM) [17], Powell's dogleg [18], or the gradient descent [19] to obtain locally optimal solutions. These methods are available as mature, highly-optimized toolboxes, such as g2o [20], GTSAM [21], SLAM++ [22], and ceres [23]. However, it is possible to obtain a wrong estimate using these iterative techniques, because of the local minimum caused by the non-convexity of the problem and a poor initial guess. A set of works shows that the duality gap of the general pose-graph SLAM problems in practical applications is close to zero [24], which implies that we can obtain the globally optimal solutions for many pose-graph SLAM problems via convex relaxations. Some further works also explore the separable structure of the SLAM between the linear part (translation) and the non-linear core (orientation), which distinguishes the pose-graph SLAM problem from a generic nonlinear least-squares problem. For the iterative techniques, in [25], the authors propose a scalable and efficient Newton-based method using the variable projection algorithm to take advantage of the separability. For the convex-based techniques, the authors in [26] decouple the rotation and translation parts and build rotation-only ML estimation using semidefinite relaxation and Riemannian optimization methods.

The anchor selection problem is based on the pose-graph SLAM with multiple anchors. Both iterative and convex-based techniques for the classical pose-graph SLAM with one anchor can be extended to the multiple anchors case. One approach is to consider the state vector corresponding to the anchored poses as constant values and remove the corresponding rows and columns of the Jacobian and Hessian matrices. This is easy to be implemented in the iterative optimization techniques but difficult to be applied in the convex-based algorithms, because the anchored poses make the relaxation process more complicated. The other approach for solving the pose-graph SLAM with multiple anchors is to add very accurate measurements between the origin and the anchored poses, corresponding to edges with very large weights, to the pose graph. This approximate approach is suitable for both iterative methods and convex-based methods. It is easy to implement and does not require modifying the mature toolboxes. So, in this paper, we use the approximate approach to deal with the pose-graph SLAM with multiple anchors.

The metrics in TOED provide us a tool to evaluate the uncertainty of the SLAM results with one or more anchors. These metrics are originally used in active SLAM [11] and belief space planning [27]. The uncertainty representations on $SE(3)$ based on the Gaussian noise have been considered in [28]. In references [14], [29], based on the block-isotropic Gaussian noise, the popular D-optimality metric shows its intrinsic connection with the graph structure of the measurement network, especially *tree-connectivity*. The similar but more complicated 3D conclusions on both the D-optimality metric and the T-optimality metric are presented in [13] based on the assumption of isotropic Langevin noise for rotation and block-isotropic Gaussian noise for translation. By the tight lower bound of the D-optimality metric [13], the measurements of the pose-graph network can be selected efficiently to improve the accuracy of the result in active SLAM [30] and CSLAM [31]. Similarly, in this paper, using the same noise assumption as in [13], the TOED metric and their tight lower bounds are used to evaluate the SLAM estimate uncertainty in our anchor selection framework.

In this paper, the anchor selection problem is solved by the sub-modular optimization method, whose objective function belongs to the sub-modular (set) functions. The sub-modular function is a set function of which the value has the property that, with the growth of the size of the input set, the difference in the incremental value of the function that a single element makes when added to an input set decreases. The sub-modular functions are first used for matroid optimization, transportation problems, and Boolean polynomials in [32]. The greedy solutions of the monotone increasing sub-modular functions commonly show the near-optimal performance with worst-case guarantees [33]. In fact, sub-/super-modular set functions are naturally discrete analogues of convex/concave continuous functions and inherit their desirable properties [34].

Sub-modularity has played an essential role in network-related combinatorial optimization problems, including network synchronization with nonlinear dynamics [35], network leader selection [36], and sensor scheduling [37]. The most related works of this paper belong to the sensor and measurement selection in robotics. In [38], using sub-modular optimization, the authors aim to select a subset of sensors (satisfying certain budget constraints) from a given set so as to minimize the trace of the steady state *a priori* or *a posteriori* error covariance produced by a Kalman filter for estimating the states of linear dynamical systems. In [39], the authors apply sub-modular optimization to choose a subset from a finite set of possible placements to optimize some real-valued controllability and observability metrics of the network. In [40], based on super-modularity and monotonicity of conditional entropy, the authors solve an NP-hard problem of sensor scheduling for the stochastic process estimation. As a combinatorial problem, our anchor selection problem is also solved by the sub-modular optimization techniques.

III. Pose-graph SLAM with multiple anchors

In this section, we present the problem formulation of the pose-graph SLAM with multiple anchors.

A. Graph Preliminaries of pose-graph SLAM

The pose-graph SLAM problem with no anchor can be represented as a weighted weakly-connected directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \omega)$, where $\mathcal{V} = \{1, 2, \cdots, n_p\}$, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ and $|\mathcal{E}| = m$ [25]. Each node $P_i = (\boldsymbol{x}_i, \boldsymbol{R}_i)$ denotes a robot pose, and each edge $e_k = (i, j) \in \mathcal{E}$ represents the k-th relative measurement between two robot poses P_i and P_j. We assign a positive weight $w : \mathcal{E} \to \mathbb{R} > 0$ to each edge. The set of poses, which gets measurements based on pose P_i, is denoted as V_i^+ (out), satisfying $(i, j) \in \mathcal{E} \Leftrightarrow j \in V_i^+$.

The incidence matrix of $\mathcal{G}$ is denoted by $\boldsymbol{A} \in \{-1, 0, 1\}^{n_p \times m}$. Its i-th and j-th elements in the k-th column satisfy $a_{ik} = -1$ and $a_{jk} = 1$, if the k-th edge is $e_k = (i, j) \in \mathcal{E}$. The rest elements of the incidence matrix $\boldsymbol{A}$ are 0. The weighted Laplacian matrix $\boldsymbol{L}_\omega \triangleq \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^\top$, where $\boldsymbol{\Sigma}$ is the diagonal matrix whose diagonal elements are the weight values of the graph edges. The weights represent the inverse values of the variances of the measurement errors and show the uncertainty and reliability of these measurements.

The multiple anchored poses are defined as $\mathcal{N}_a \subseteq \mathcal{V}$. $\mathcal{V} \backslash \mathcal{N}_a$ means the set of all the unknown poses. The incidence matrix $\boldsymbol{A}$ after anchoring to the anchored poses $\mathcal{N}_a$, $|\mathcal{N}_a| = N$, called as reduced incidence matrix $\boldsymbol{A}(\mathcal{V} \backslash \mathcal{N}_a) \in \{-1, 0, 1\}^{(n_p - N) \times m}$, is obtained simply by removing the rows corresponding to the anchored poses in $\boldsymbol{A}$. Its corresponding reduced weighted Laplacian matrix is $\boldsymbol{L}_\omega(\mathcal{V} \backslash \mathcal{N}_a) \triangleq \boldsymbol{A}(\mathcal{V} \backslash \mathcal{N}_a)\boldsymbol{\Sigma}^*(\mathcal{V} \backslash \mathcal{N}_a)\boldsymbol{A}^\top(\mathcal{V} \backslash \mathcal{N}_a)$, where $\boldsymbol{\Sigma}^*(\mathcal{V} \backslash \mathcal{N}_a)$ is the submatrix of the diagonal matrix $\boldsymbol{\Sigma}$ generated by deleting the rows and columns corresponding to the edges connecting two anchors.

B. Synchronization on $\mathbb{R}^n \times SO(n)$ with multiple anchors

2D/3D pose-graph SLAM, belonging to synchronization problem on $\mathbb{R}^n \times SO(n)$, is the problem of estimating the values of a set of n_p unknown poses $P_1, \cdots, P_{n_p} \in \mathbb{R}^n \times SO(n)$ given m noisy relative rotations $\boldsymbol{H}_{ij} \in SO(n)$ and relative translations $\boldsymbol{p}_{ij} \in \mathbb{R}^n$. For the pose-graph edge $(i, j) \in \mathcal{E}$, we assume that the noisy relative measurements follow [26]:

$$\begin{aligned} \boldsymbol{p}_{ij} &= \boldsymbol{R}_i^\top(\boldsymbol{x}_j - \boldsymbol{x}_i) + \boldsymbol{y}_{ij}, \quad & \boldsymbol{y}_{ij} &\sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{ij}) \\ \boldsymbol{H}_{ij} &= \boldsymbol{Z}_{ij}\boldsymbol{R}_j\boldsymbol{R}_i^\top, & \boldsymbol{Z}_{ij} &\sim Lang(\boldsymbol{I}_{n\times n}, \kappa_{ij}), \end{aligned} \tag{1}$$

where $\boldsymbol{y}_{ij} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{ij})$, $\boldsymbol{\Sigma}_{ij} = \delta_{ij}^2 \boldsymbol{I}_{n\times n}$ means the random vector $\boldsymbol{y}_{ij}$ following the isotropic Gaussian distribution, $Lang(\boldsymbol{I}_{n\times n}, \kappa_{ij})$ means the isotropic Langevin distribution with mean $\boldsymbol{I}_{n\times n}$ and concentration $\kappa_{ij} \geq 0$, and its probability density function (PDF) $f_{ij} : SO(n) \to \mathbb{R}^+$ [41] is:

$$\begin{aligned} f_{ij}(\boldsymbol{Z}_{ij}) &= \frac{1}{c_n(\kappa_{ij})} \exp\left(\kappa_{ij} \text{ trace}(\boldsymbol{Z}_{ij})\right), \\ c_2(\kappa_{ij}) &= I_0(2\kappa_{ij}), \\ c_3(\kappa_{ij}) &= \exp(\kappa_{ij})\left(I_0(2\kappa_{ij}) - I_1(2\kappa_{ij})\right), \\ I_v(2\kappa_{ij}) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} \exp\left(2\kappa_{ij}\cos(\theta)\right)\cos(v\theta)d\theta, \end{aligned} \tag{2}$$

where $c_n(\kappa_{ij})$, $n = 2, 3$ is a normalization constant such that f_{ij} has unit mass. $I_v(2\kappa_{ij})$, $v = \{0, 1, 2, \cdots\} \in \mathbb{Z}$ means the modified Bessel functions [42].

Given a set of noisy measurements $\boldsymbol{p}_{ij}$ and $\boldsymbol{H}_{ij}$, the pose-graph SLAM problem with no anchor is to obtain the ML estimand for the poses $\boldsymbol{P} = \{P_1, \cdots, P_{n_p}\} \in \{\mathbb{R}^n \times SO(n)\}^{n_p}$:

$$\max_{\boldsymbol{P}} \sum_{e_k \in \mathcal{E}} (\kappa_{ij} tr(\boldsymbol{H}_{ij}\boldsymbol{R}_i\boldsymbol{R}_j^\top) - \frac{\|\boldsymbol{p}_{ij} - \boldsymbol{R}_i^\top(\boldsymbol{x}_j - \boldsymbol{x}_i)\|_2^2}{2\delta_{ij}^2}). \tag{3}$$

For the SLAM with multiple anchors, the ground truths of the anchored poses are known, so the problem is to estimate poses $\boldsymbol{P'} = \{P_{i'_1},\ P_{i'_2},\ \cdots\}$, $\{i'_1,\ i'_2,\ \cdots\} = \mathcal{V} \backslash \mathcal{N}_a$.[5]

$$\max_{\boldsymbol{P'}} \sum_{e_k \in \mathcal{E}} (\kappa_{ij} tr(\boldsymbol{H}_{ij}\boldsymbol{R}_i\boldsymbol{R}_j^\top) - \frac{\|\boldsymbol{p}_{ij} - \boldsymbol{R}_i^\top(\boldsymbol{x}_j - \boldsymbol{x}_i)\|_2^2}{2\delta_{ij}^2}). \tag{4}$$

C. Fisher information matrix

As a tool to evaluate the uncertainty of the SLAM result, the FIMs $\boldsymbol{\mathcal{I}} = \boldsymbol{\mathcal{I}}_{nD}$, $n = 2, 3$ of the 2D/3D pose-graph SLAM result based on the log-likelihood function (3) are shown by the following two propositions[6] [13].

Proposition 1. *[13] For the 2D case of the pose-graph SLAM problem* (3)*, given the specific 2D basis in [13], the FIM is:*

$$\boldsymbol{\mathcal{I}}_{2D} = \left[\begin{array}{c:c} \boldsymbol{L}_w^{\mathbb{R}^2} & {\triangle_w^{2D}}^\top \\ \hdashline \triangle_w^{2D} & \boldsymbol{L}_w^{SO(2)} + diag\{\psi_1, \cdots, \psi_{n_p}\} \end{array}\right], \tag{5}$$

where $\boldsymbol{L}_w^{\mathbb{R}^2}$ is the sub-FIM corresponding to the Euclidean space $\mathbb{R}^2$, satisfying $\boldsymbol{L}_w^{\mathbb{R}^2} = \boldsymbol{L}_{w_\mathbb{R}} \otimes \boldsymbol{I}_{2\times 2}$. $\boldsymbol{L}_{w_\mathbb{R}}$ is the weighted Laplacian matrix, of which weight value $w_{ij}^{\mathbb{R}}$ for (i, j)-th edge is δ_{ij}^{-2}; $\boldsymbol{L}_w^{SO(2)} + diag\{\psi_1, \cdots, \psi_{n_p}\}$ is the sub-FIM corresponding to the $SO(2)$ Lie group, satisfying $\boldsymbol{L}_w^{SO(2)} =$

[5] In the case when both two poses of an edge are anchored, the edge needs to be deleted from $\mathcal{E}$, which results in a sub-set of $\mathcal{E}$ to be used in the objective function.

[6] Commonly, it is noted that the edge weights $w_{ij}^{\mathbb{R}}$, $w_{ij}^{SO(3)}$, and $w_{ij}^{SO(2)}$ are all much larger than 1, which will be used in Appendix B in supplementary material [43].

$\boldsymbol{L}_{w_{SO(2)}} \otimes \boldsymbol{I}_{d\times d}$, *where* $d = \frac{n(n-1)}{2} = 1$. $\boldsymbol{L}_{w_{SO(2)}}$ *is the weighted Laplacian matrix, of which weight value* $w_{ij}^{SO(2)}$ *for* (i,j)*-th edge is* $2\kappa_{ij}\frac{I_1(2\kappa_{ij})}{I_0(2\kappa_{ij})}$. $\psi_i = \sum_{j\in V_i^+}\delta_{ij}^{-2}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2,\ i = 1,2,\cdots,n_p$. $\boldsymbol{E} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$. *The* (i,i_1)*-th block of the* $SO(2)$ *by* $\mathbb{R}^2$ *coupling sub-matrix* $\triangle_w^{2D}$ *corresponding to the* (n_p+i,i_1)*-th block of the FIM is:*

$$(\triangle_w^{2D})_{i,i_1} = \begin{cases} \sum_{j\in V_i^+}\delta_{ij}^{-2}(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{E} & i = i_1 \\ \delta_{ii_1}^{-2}(\boldsymbol{x}_{i_1} - \boldsymbol{x}_i)^\top \boldsymbol{E} & (i,i_1)\in\mathcal{E} \\ \boldsymbol{0}_{1\times 2} & \textit{else}. \end{cases} \quad (6)$$

Proposition 2. *[13] For the 3D case of the pose-graph SLAM problem* (3)*, given the specific 3D basis in [13], the FIM is:*

$$\boldsymbol{\mathcal{I}}_{3D} = \left[\begin{array}{c:c} \boldsymbol{L}_w^{\mathbb{R}^3} & {\triangle_w^{3D}}^\top \\ \hdashline \triangle_w^{3D} & \boldsymbol{L}_w^{SO(3)} + \textit{diag}\{\boldsymbol{\Psi}_1,\cdots,\boldsymbol{\Psi}_{n_p}\} \end{array}\right], \quad (7)$$

where $\boldsymbol{L}_w^{\mathbb{R}^3}$ *is the sub-FIM corresponding to the Euclidean space* $\mathbb{R}^3$*, satisfying* $\boldsymbol{L}_w^{\mathbb{R}^3} = \boldsymbol{L}_{w_\mathbb{R}} \otimes \boldsymbol{I}_{3\times 3}$. $\boldsymbol{L}_{w_\mathbb{R}}$ *is the same as that in Proposition 1;* $\boldsymbol{L}_w^{SO(3)} + \textit{diag}\{\boldsymbol{\Psi}_1,\cdots,\boldsymbol{\Psi}_{n_p}\}$ *is the sub-FIM corresponding to the* $SO(3)$ *Lie group, satisfying* $\boldsymbol{L}_w^{SO(3)} = \boldsymbol{L}_{w_{SO(3)}} \otimes \boldsymbol{I}_{d\times d}$*, where* $d = \frac{n(n-1)}{2} = 3$. $\boldsymbol{L}_{w_{SO(3)}}$ *is the weighted Laplacian matrix, of which weight value* $w_{ij}^{SO(3)}$ *for* (i,j)*-th edge is* $\frac{1}{3}\frac{\kappa_{ij}^2(2I_0(2\kappa_{ij})-I_1(2\kappa_{ij})-2I_2(2\kappa_{ij})+I_3(2\kappa_{ij}))}{2I_0(2\kappa_{ij})-2I_1(2\kappa_{ij})}$. $\boldsymbol{\Psi}_i$ *satisfies:*

$$\begin{aligned} \boldsymbol{\Psi}_i &= \begin{bmatrix} \psi_i^{11} & \psi_i^{12} & \psi_i^{13} \\ \psi_i^{12} & \psi_i^{22} & \psi_i^{21} \\ \psi_i^{13} & \psi_i^{21} & \psi_i^{33} \end{bmatrix},\ i = 1,2,\cdots,n_p \\ \psi_i^{kl} &= \sum_{j\in V_i^+}\delta_{ij}^{-2}(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{R}_i \boldsymbol{I}_{3\times 3}^{k,l}\boldsymbol{R}_i^\top(\boldsymbol{x}_i - \boldsymbol{x}_j); \end{aligned} \quad (8)$$

where $\boldsymbol{I}_{3\times 3}^{k,l} = \boldsymbol{E}_k\boldsymbol{E}_l,\ k,\ l = 1,2,3$, $\boldsymbol{E}_k$ *and* $\boldsymbol{E}_l$ *are the orthogonal bases of Lie group* $SO(3)$. *Let* $\varsigma_{ij}^k = (\delta_{ij}^{-2}(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{R}_i\boldsymbol{E}_k\boldsymbol{R}_i^\top)^\top$, $k = 1,2,3$*, we have the* (i,i_1)*-th block of the* $SO(3)$ *by* $\mathbb{R}^3$ *coupling sub-matrix* $\triangle_w^{3D}$ *corresponding to the* (n_p+i,i_1)*-th block of the FIM:*

$$\begin{aligned} &(\triangle_w^{3D})_{i,i_1} = \\ &\begin{cases} \begin{bmatrix} \sum_{j\in V_i^+}\varsigma_{ij}^1 & \sum_{j\in V_i^+}\varsigma_{ij}^2 & \sum_{j\in V_i^+}\varsigma_{ij}^3 \end{bmatrix}^\top & i = i_1 \\ \begin{bmatrix} -\varsigma_{ii_1}^1 & -\varsigma_{ii_1}^2 & -\varsigma_{ii_1}^3 \end{bmatrix}^\top & (i,i_1)\in\mathcal{E} \\ \boldsymbol{0}_{3\times 3} & \textit{else}. \end{cases} \end{aligned} \quad (9)$$

For the conventional SLAM problem, the first pose P_1 is anchored. The corresponding columns and rows of the FIM are deleted, which leads to a positive reduced matrix $\boldsymbol{\mathcal{I}}(\mathcal{V}\backslash\{1\})$.

For the SLAM with multiple anchors, because the information for the known anchors with ground truths are infinite (without any uncertainty), the rows and columns of the FIM corresponding to the anchored poses are deleted, defined as $\boldsymbol{\mathcal{I}}(\mathcal{V}\backslash\mathcal{N}_a)$. It should be noted that, the final estimated poses $\boldsymbol{P'}$ will be different from the corresponding sub-part of the SLAM solutions $\boldsymbol{P}$ with only the first pose as the anchor. Meanwhile, with the same number of the anchors, the different anchors also result in the different solutions. Hence, with the value changing of the obtained poses, the elements of the parts of the reduced FIM $\boldsymbol{\mathcal{I}}(\mathcal{V}\backslash\mathcal{N}_a)$, including $\triangle_w^{nD}, n = 2,3$, ψ_i and $\boldsymbol{\Psi}_i$, are also changed, because they are dependent on the estimated poses $\{P_{i'_1},\ P_{i'_2},\ \cdots\}$. It means that $\boldsymbol{\mathcal{I}}(\mathcal{V}\backslash\mathcal{N}_a)$ is no longer a sub-matrix of the original matrix $\boldsymbol{\mathcal{I}}$. For the case with given set $\mathcal{N}_a$ of the anchors, in order to get its correct corresponding FIM $\boldsymbol{\mathcal{I}}(\mathcal{V}\backslash\mathcal{N}_a)$, the anchored SLAM problem is first solved using the iterative methods or the convex-based methods, and then Proposition 1 and Proposition 2 are applied by removing the rows and columns of the anchored poses.

IV. Anchor selection optimization problem

In this part, we will formulate the anchor selection problem, prove its sub-modularity, and discuss its relations with edge selection and node-edge selection problems.

A. Optimization problem formulation

The purpose to select the anchored poses is to minimize the uncertainty of the SLAM result. Based on the D-optimality metric, given the fixed anchor number $|\mathcal{N}_a| = N$, the anchor selection problem is formulated as:

$$\begin{aligned} \underset{\mathcal{N}_a}{\text{maximize}} \quad & f_{cost}(\mathcal{N}_a) \triangleq \log(\det(\boldsymbol{\mathcal{I}}(\mathcal{V}\backslash\mathcal{N}_a))) \\ \text{subject to} \quad & |\mathcal{N}_a| = N. \end{aligned} \quad (10)$$

As discussed in the last paragraph in Section III, different anchors $\mathcal{N}_a$ will result in different SLAM solutions and different reduced FIMs $\boldsymbol{\mathcal{I}}(\mathcal{V}\backslash\mathcal{N}_a)$. Thus, to compute the accurate objective functions in (10) for a given candidate choice $\mathcal{N}_a$, we need to first obtain the estimated result by solving the whole pose-graph optimization process, then use the result to compute its corresponding reduced FIM, and finally get the objective function value $f_{cost}(\mathcal{N}_a)$. In solving general combinatorial optimization problems, the evaluation of the objective functions needs to be performed many times and is one of the most computational-costly steps. Hence, it is computational unacceptable for directly solving the anchor selection based on the formulation (10).

Therefore, if we can find a way to avoid the complexity involved in computing the pose-graph SLAM solution, the computation of the objective function value of the anchor selection problem can be much faster and become computationally solvable. The following proposition shows that the D-optimality metric is close to the tree-connectivity of the pose-graph, which indicates that the graph structure of the SLAM measurement network plays a key role in evaluating the uncertainty of the SLAM result. Meanwhile, we note that the tree-connectivity of the pose-graph is independent of the computing of the SLAM solution, so we aim to replace the original D-optimality objective function with the tree-connectivity in order to avoid solving the pose-graph optimization problem when evaluating the objective function.

Proposition 3. *[13] Considering 2D/3D pose-graph SLAM in* (3)*,* $\delta \triangleq \max\limits_{i=2,3,\cdots,n_p}\sum_{j\in V_i^+}\delta_{ij}^{-2}\|\boldsymbol{x}_j - \boldsymbol{x}_i\|^2$ *(2D) or* $\delta \triangleq \max\limits_{i=2,3,\cdots,n_p}\frac{1}{2}\sum_{j\in V_i^+}\delta_{ij}^{-2}\|\boldsymbol{x}_j - \boldsymbol{x}_i\|^2$ *(3D) and* $\lambda_1 =$

$\lambda_{min}(\boldsymbol{L}_w^{SO(n)})$, *define* $\varepsilon = \log(\det(\boldsymbol{\mathcal{I}})) - \log(\det(\boldsymbol{L}_w^{\mathbb{R}^n})) - \log(\det(\boldsymbol{L}_w^{SO(n)}))$. *Then we have,*

$$0 \leq \varepsilon \leq d(n_p - 1)\log(1 + \delta/\lambda_1), \tag{11}$$

Based on Proposition 3, when $\delta/\lambda_1 \to 0^+$, $\log(\det(\boldsymbol{\mathcal{I}}))$ will get close to $\log(\det(\boldsymbol{L}_w^{\mathbb{R}^n})) + \log(\det(\boldsymbol{L}_w^{SO(n)}))$. From [13], [14], in many real-world pose graph SLAM problems, the translation part δ is relatively small. So, we can approximate the anchor selection problem (10) by the following problem:

$$\begin{aligned} \underset{\mathcal{N}_a}{\text{maximize}} \quad & f_{obj}(\mathcal{N}_a) \triangleq \log(\det(\boldsymbol{\mathcal{L}}(\mathcal{V}\backslash\mathcal{N}_a))) \\ \text{subject to} \quad & |\mathcal{N}_a| = N, \end{aligned} \tag{12}$$

where $\boldsymbol{\mathcal{L}} = \text{diag}\{\boldsymbol{L}_{w_{\mathbb{R}}} \otimes \boldsymbol{I}_{n\times n}, \boldsymbol{L}_{w_{SO(n)}} \otimes \boldsymbol{I}_{d\times d}\}$, thus $f_{obj}(\mathcal{N}_a) = n\log(\det(\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash\mathcal{N}_a))) + d\log(\det(\boldsymbol{L}_{w_{SO(n)}}(\mathcal{V}\backslash\mathcal{N}_a)))$.

It is important to notice that the new problem (12) is independent of the final SLAM solution. So we can compute the objective function of the anchor selection problem without performing the SLAM optimization process.

B. Cardinality-fixed sub-modular optimization

In this part, our goal is to prove the sub-modularity of the approximate formulation (12). Before this, let's consider:

Definition 1. *[14] Let $\mathcal{T}_{\mathcal{G}}$ be the set of all spanning trees of $\mathcal{G}$. The weighted number $t_w(\mathcal{G})$ of $\mathcal{G}$, called tree-connectivity [14], is defined as:*

$$t_w(\mathcal{G}) \triangleq \sum_{\mathcal{T}\in\mathcal{T}_{\mathcal{G}}} \mathbb{V}(\mathcal{T}),\ \mathbb{V}(\mathcal{T}) = \prod_{e\in\mathcal{E}(\mathcal{T})} w(e), \tag{13}$$

where $w(e)$ means the weighted value of the edge e, $\mathbb{V}(\mathcal{T}) : \mathcal{T} \to \mathbb{R}_+$ is the value of a spanning tree $\mathcal{T} \in \mathcal{T}_{\mathcal{G}}$, $\mathcal{E}(\mathcal{T})$ represents the set of all edges in $\mathcal{T}$.

A theorem about the tree-connectivity is as follows:

Theorem 1. *(Weighted Matrix-Tree Theorem [14]). For $\mathcal{G} = (\mathcal{V},\ \mathcal{E},\ w)$ with $w : \mathcal{E} \to \mathbb{R}^+$, we have $t_w(\mathcal{G}) = \det(\boldsymbol{\mathcal{L}}_w^{\mathcal{G}})$, where $\boldsymbol{\mathcal{L}}_w^{\mathcal{G}}$ is the reduced weighted Laplacian matrix of $\mathcal{G}$ removing one row and one corresponding column.*

For the selection of the only one anchored pose, we have the following theorem:

Theorem 2. *If only one pose is anchored, the objective function in* (12) *is independent of the choice of the anchored pose.*

Proof: Assuming the i-th pose or the j-th pose is anchored, the objective function in (12) is decided by the reduced weighted Laplacian matrices $\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash\{i\})$, $\boldsymbol{L}_{w_{SO(n)}}(\mathcal{V}\backslash\{i\})$, $\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash\{j\})$, and $\boldsymbol{L}_{w_{SO(n)}}(\mathcal{V}\backslash\{j\})$. Based on Theorem 1, the determinant of the reduced weighted Laplacian matrix with one anchor is equal to the weighted number of spanning trees of its corresponding weighted graph (tree-connectivity) [45]. Because the weighted pose graph is the same, the weighted number of its spanning trees is constant. So we have: $\det(\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash\{i\})) = \det(\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash\{j\}))$ and $\det(\boldsymbol{L}_{w_{SO(n)}}(\mathcal{V}\backslash\{i\})) = \det(\boldsymbol{L}_{w_{SO(n)}}(\mathcal{V}\backslash\{j\}))$. The proof is completed. ■

Without loss of generality, for the multiple anchor selection problem, we can first anchor the first pose P_1 (similar to the classical SLAM problems), and then choose the rest anchors. Then, we have the following theorem using some preliminaries about the sub-modularity shown in Appendix A in supplementary material [43].

Theorem 3. *When more than one anchor is chosen, the optimization problem* (12) *is a non-negative non-normalized non-monotone sub-modular optimization with a cardinality-fixed constraint.*

Proof: See Appendix B in supplementary material [43]. ■

Remark 1. *Benefiting from the approximation using graph topology from the optimization problem* (10) *to the problem* (12)*, Theorem 3 is related to the sub-modular property of the entropy maximization problem for selecting the most informative subset from a set of correlated random variables in [45]. However, in [45], the random variables correspond to a linear Gaussian estimation problem, such as linear sensor network and compass-SLAM [14], and is essentially different from the common highly-nonlinear SLAM problem. Besides the differences of the focused problem setting, the anchor selection problem uses the anchored poses as the variables and needs to operate multiple rows and columns, instead of one row and one column. Theorem 3 is also related to the related work [46] using a determinant function, instead of using the log-determinant function, for the sub-matrix selection of the real symmetric positive semidefinite matrix, which shows the multiplicative sub-modular, instead of classical sub-modular, and it is applied in the problem related to the complete graph without connecting with the estimation problems. Besides, compared with these two references [45], [46], our anchor selection is not a monotone increasing optimization.*

C. Discussion on the connections and differences between anchor (nodes) selection, edge selection, and node-edge selection

The approach in this paper is inspired by the novel edge selection reference [14] and its related application [31]. The k edge selection problem (k-ESP) aims to select the optimal edges to be added/deleted for a given pose graph with fixed nodes to maximize the information gain or minimize the information lost. They also use the tree-connectivity to replace the D-optimality design metric of the FIM as the objective function. However, there are some important differences between the anchor selection and the edge selection problems.

The key difference is in the information aspect. The edge selection is similar to the information marginalization, which means only the local relative information from measurements is added/removed. However, in the anchor selection, the global information of the anchored poses is added. As shown in Fig. 1, we consider a base pose graph and one factor $\frac{1}{2\delta_{ij}^2}\|\boldsymbol{p}_{ij} - \boldsymbol{R}_i^\top(\boldsymbol{x}_j - \boldsymbol{x}_i)\|^2$ in the objective function. For simplification, only the weighted Laplacian matrix corresponding to the translation part $\boldsymbol{L}_{w_{\mathbb{R}}} \otimes \boldsymbol{I}_{n\times n}$, which is a sub-matrix of $\boldsymbol{\mathcal{L}}$ in (12), is considered. Figures in the first row show the generating process from the incident matrix and the diagonal matrix with weights to $\boldsymbol{L}_{w_{\mathbb{R}}} \otimes \boldsymbol{I}_{n\times n}$ corresponding to the considered factor. For the anchor selection method, even though pose P_j is regarded as the constant value and removed from the state vector, the measurement between P_i and P_j remains, and the uncertainty of pose P_i is still limited by this measurement (i, j).

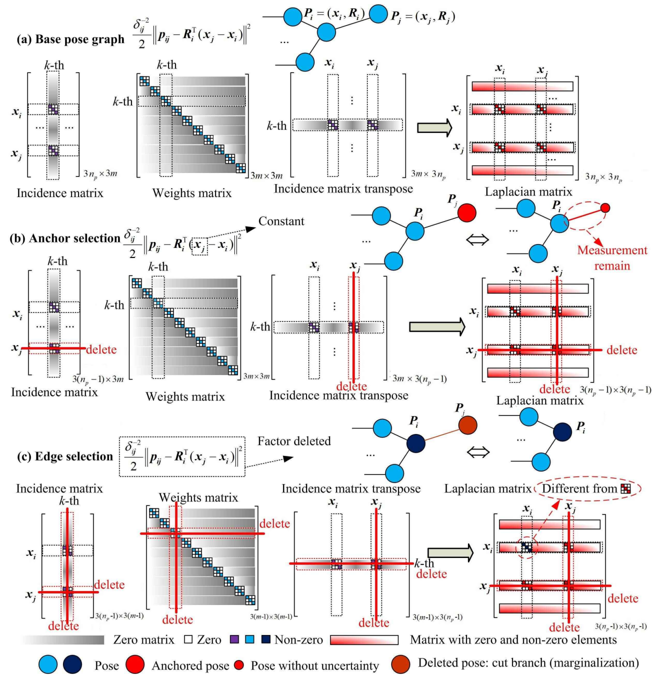


Fig. 1: The differences of changing the reduced weighted Laplacian matrices (translation only) calculated by incidence matrix and weights diagonal matrix: (a) Base pose graph; (b) Anchor selection: the anchored pose P_j is deleted; (c) Edge selection: both pose P_j and the edge (i, j) are deleted, which also resulting in a different P_i block in the weighted Laplacian matrix.

Commonly, in edge selection [14], we do not add or delete the nodes. In [47], the edge selection is formulated as a data exchange problem in the loop closure detection of CSLAM. Considering its corresponding problem in our formulation, it means to select all the edges connected to those poses by adding some additional measurements to these edges but without anchoring the poses themselves. Similar to the common edge selection problem, the main difference between our anchor selection and the data exchange problem is also from the information aspect. The added measurements will make the connections between these poses and their connected poses stronger, which greatly strengthens the local structure of the pose graph. In other words, the uncertainty levels of these poses connected by these additional edges trend to converge and these poses can be considered as a local unit from the information point of view. However, because no global information is added, their uncertainty level may be still large if the other poses in the whole pose graph are weakly connected with this new local unit. The impact of this local unit on the global pose graph is limited (Fig. 2). In some special situations, if edge selection results in the deletion of one pose P_j, which corresponds to deleting all the edges connected to this pose, then this situation is still different from anchoring pose P_j. This is because some useful information from the measurements (i, j) are marginalized, and the uncertainty of pose P_i increases. In this case, P_j is not estimated which is different from the case of anchoring P_j where the ground truth of P_j is known. This operation in edge selection may cause the sparser pose graph structure, as shown in Fig. 3.

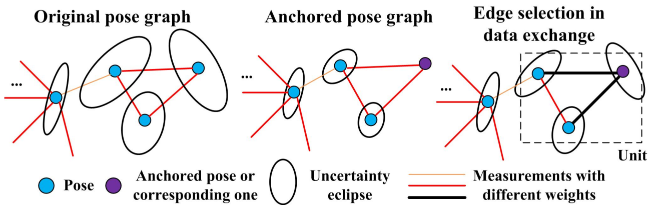

Original pose graph
Anchored pose graph
Edge selection in data exchange
...
Unit
Pose
Anchored pose or corresponding one
Uncertainty eclipse
Measurements with different weights

Fig. 2: The first and second figures are the original pose graph and the anchored pose graph with one anchor. The last figure is the corresponding edge selection case shown in data exchange [47]. For the last figure, because the measurements connected to the purple pose become very strong, the accuracy of the local poses will improve. The three poses can be considered as a local unit. Their accuracy will be decided by the other poses connect to this local unit. Especially when the local structure is weakly connected with other poses, no matter how accurate the added measurements (black measurements with large weights) are, the pose estimate accuracy is still poor and limited.

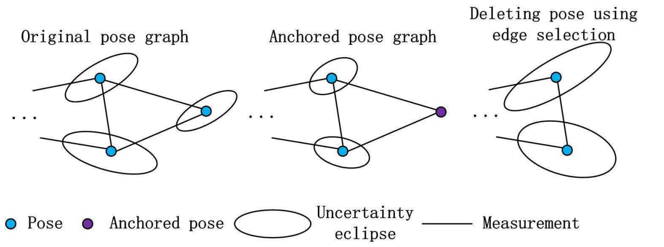


Fig. 3: The first figure is the original pose graph. The second figure is the anchored pose graph with an anchor in purple. Because the uncertainty of the purple anchor is reduced to zero, the connected measurements limit the related poses and the uncertainty of these poses also reduces. The last figure is to delete a pose using edge selection, which means to delete all the edges connected to a pose. Because the purple pose does not need to be estimated, the related measurements are also ignored in the pose graph. With the sparser pose-graph structure, the accuracy of the rest poses reduces.

Meanwhile, because of using different variables in the optimization, the details of the problem formulation are also different. Commonly, in most applications, like sensor scheduling [14], [48], the candidate set for the edge selection is limited to make sure that no poses are deleted/added, which means the dimension of the state vector corresponding to the poses keeps as constant and only the values of the FIM corresponding to the poses connected the selected edge are changed. However, for the anchor selection problem, the dimension of the FIM will certainly reduce. The edge selection problem discussed in [14] is proved to be a normalized, monotone increasing sub-modular maximization problem, which shows different properties compared with ours. More details of the edge selection problem are shown in [13], [14].

Some other works such as [30] focus on the node-edge selection problem related to active SLAM. In active SLAM, the additional trajectory, which introduces the new poses and measurements to the base pose graph, is selected optimally to reduce the uncertainty of the SLAM solution. Similar to the edge selection problem and different from the anchor selection problem, it does not introduce any global ground truth information. Moreover, because of introducing more poses, the dimension of the FIM will become larger, which is different from the ones corresponding to edge selection (reduce or keep dimension) and anchor selection (reduce dimension).

V. Greedy methods

As a non-negative non-normalized non-monotone sub-modular maximization problem with a cardinality-fixed constraint, the anchor selection problem (12) is in general NP-hard[7]. To find its optimal solution is computational-costly for a large-scale problem. The greedy-based optimization methods can be used to solve it with some performance guarantee. In this section, we will present several greedy-based methods and discuss the near optimal performance of the obtained solutions.

A. Natural greedy algorithm

A natural greedy algorithm for this problem starts with an empty set, and then an element is added to maximize the marginal gain at the k-th iteration. A formal statement of the 1-step greedy algorithm is given in Algorithm 1 [32].

Algorithm 1 1-step greedy algorithm

```
Require: The new weighted Laplacian matrix $\boldsymbol{\mathcal{L}}$; the fixed anchored poses number $N$
Ensure: The sub-optimal anchored poses set $\mathcal{N}_a$
$\mathcal{N}_a \leftarrow \emptyset$;
while $|\mathcal{N}_a| \le N$ do                          ▷ Cardinality-fixed constraint
    $s^*_{n_p-k} = \max_{s_{n_p-k}\in\mathcal{V}\setminus\mathcal{N}_a} f_{obj}(\mathcal{N}_a \cup \{s_{n_p-k}\})$;
    $\mathcal{N}_a \leftarrow \mathcal{N}_a \cup \{s^*_{n_p-k}\}$;
    $k \leftarrow k+1$;
return $\mathcal{N}_a$                                    ▷ Selected set with good performance
```

At the k-th iteration, based on the log-determinant values of $\boldsymbol{L}_{w_{\mathbb{R}}}$ and $\boldsymbol{L}_{w_{SO(n)}}$, the computational complexity of computing the objective function value $f_{obj}(\mathcal{N}_a \cup \{s_{n_p-k}\})$ is $O((n_p-k+1)^3)$. At each iteration, every possible candidate anchor will be evaluated by the log-determinant function, so the computational complexity of the greedy method will be $O((n_p-N+1)^4+\cdots+(n_p)^4)$, which can be simplified as:

$$\begin{aligned} &O((n_p-N+1)^4+\cdots+(n_p)^4)\\ =&O\left((1^4+\cdots+(n_p)^4)-(1^4+\cdots+(n_p-N)^4)\right)\\ =&O(f_o(n_p)-f_o(n_p-N)), \end{aligned} \tag{14}$$

where $f_o(\star) = \frac{\star(\star+1)(6\star^3+9\star^2+\star-1)}{2}$, so we can get $O(f_o(n_p)-f_o(n_p-N)) \approx O(6N^5-30N^4n_p+60N^3n_p^2-60N^2n_p^3+30Nn_p^4)$. When $n_p >> N$, the computational complexity will be about $O\left(30Nn_p^4\right)$.

B. Random greedy and continuous-double-greedy algorithms

Because $f_{obj}(\mathcal{N}_a)$ is a non-monotone sub-modular optimization function, instead of monotone increasing function, the natural greedy algorithm does not have a performance guarantee. For this problem, the combination of the random greedy method and the continuous-double-greedy method has a performance guarantee [58].

1) Random greedy algorithm: The random greedy algorithm is very similar to the natural greedy algorithm [49]. The main difference is to randomly select among the top N highest-scoring elements, instead of selecting the best one. The specific random greedy algorithm is shown in Algorithm 2[8].

Algorithm 2 Random greedy algorithm

```
Require: The weighted Laplacian matrix $\boldsymbol{\mathcal{L}}$; the fixed anchored poses number $N$
Ensure: The sub-optimal anchored poses set $\mathcal{N}_a$
$\mathcal{N}_a \leftarrow \emptyset$;
while $|\mathcal{N}_a| \le N$ do                          ▷ Cardinality-fixed constraint
    Select a subset $\mathcal{M}^*_k$ from $\mathcal{V}\setminus\mathcal{N}_a$ satisfying $\mathcal{M}^*_k = \max_{\mathcal{M}_k} \sum_{s_{n_p-k}\in\mathcal{M}_k,|\mathcal{M}_k|=N} f_{obj}(\mathcal{N}_a \cup \{s_{n_p-k}\})$;
    $s^*_{n_p-k} \leftarrow Uniform(\mathcal{M}^*_k)$    ▷ Randomly select one element from $\mathcal{M}^*_k$ following uniform distribution
    $\mathcal{N}_a \leftarrow \mathcal{N}_a \cup \{s^*_{n_p-k}\}$;
    $k \leftarrow k+1$;
return $\mathcal{N}_a$                                    ▷ Selected set with good performance
```

The computational complexity of the random greedy algorithm is similar to the natural greedy algorithm. In fact, this simple algorithm might be considered as a natural substitute for the classical greedy algorithm. For the cardinality-fixed monotone increasing sub-modular optimization problem, it retains the same tight guarantee with the natural/lazy one. For the non-monotone sub-modular optimization problem, it gives a $(1-\frac{N}{en_p})/e-\varepsilon$ approximation, where ε is an arbitrarily small positive constant [49], however, the classical greedy algorithm does not have any performance guarantee.

2) Continuous-double-greedy method: An alternative method is the continuous-double-greedy method [50]. It relies on an intuitive continuous extension of the sub-modular objective function, called the multi-linear extension. The multi-linear extension is a continuous expected function and its optimized solution is obtained using the gradient-based update considering the cardinality-fixed constraints. For more details, please refer to [49].

3) Discussion on the selection of random greedy algorithm and continuous-double-greedy method: Based on [49], for a non-monotone sub-modular optimization problem, the combination using the best solution of the random greedy algorithm (first sub-equation in (15)) and continuous-double-greedy method (second sub-equation in (15)) can reach an approximation factor ξ of:

$$\max\left\{\frac{1-\frac{N}{en_p}}{e}-\varepsilon, \left(1+\frac{n_p}{2\sqrt{(n_p-N)N}}\right)^{-1}-o(1)\right\}. \tag{15}$$

The approximation factor is defined as $\frac{\mathbb{E}\{f_{obj}(greed)\}}{f_{obj}(OPT)} \ge \xi$, where $f_{obj}(OPT)$ is the optimal objective function value and $\mathbb{E}\{f_{obj}(greed)\}$ means the expectation of the objective function value of the solution using the random greedy method. For the general anchor selection problem, the number of the anchored poses N is relatively small compared with the total number of poses n_p (usually larger than 10^3). Based on the approximation factor (15), when $N/n_p \to 0^+$, $\frac{1-\frac{N}{en_p}}{e}-\varepsilon$ will

[7] This claim of the computational complexity can be proved by following the reduction approach shown in [45]. We can specifically set the weights of the rotation part to be equal to the ones of the translation part. In this way, we can simplify the problem into the case with only one graph. Then, this formulation follows the sub-variable selection of the linear Guassian estimation problem. The log function will not affect this claim. This linear problem can be further bounded by the well-known NP-hard problems: *Clique* problem and *Stable set* problem. So in general our anchor selection problem is also NP-hard.

[8] In the last iteration of the random greed method, so as to get the better result, we pick out the best candidate anchor instead of the random one.

be bigger than $\left(1+\frac{n_p}{2\sqrt{(n_p-N)N}}\right)^{-1}-o(1)$, which means the random-greedy algorithm has a better guarantee in the low N situation. So in this paper, we only use the random greedy algorithm instead of the combination of two algorithms. So as to verify this choice, a simulation with $n_p = 10^3$ poses to show the approximation guarantees changing with the number of anchored poses is illustrated in Fig. 4.

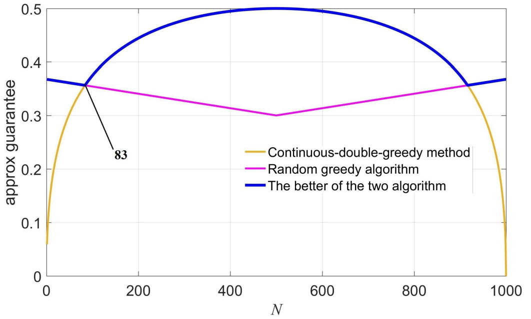


Fig. 4: Approximation guarantees of random greedy algorithm and continuous-double-greedy method.

In Fig. 4, the yellow, pink, and blue lines are the approximation factors for the continuous-double-greedy method, the random greedy method, and the combination of both two methods using the approximation equation (15) respectively. Hence, we can see that when $N \leq 83$, the random greedy algorithm achieves better approximation guarantee ($\xi = \frac{1-\frac{N}{en_p}}{e} - \varepsilon$), it verifies our choice to use the random greedy algorithm only.

C. Improved greedy-based algorithm

In this subsection, multiple techniques are used to reduce the computational complexity of the greedy-based algorithm.

1) Sparse Cholesky decomposition: It is well known that the computational complex of calculating the log-determinant function of a $n_m \times n_m$ dense matrix is $O(n_m^3)$. With the growth of n_m, its computational time becomes unacceptable for the general hardware system. Based on sparse Cholesky decomposition, for the positive definite matrix, we have:

$$\begin{aligned}\log(\det(\boldsymbol{L}_{w_{\mathbb{R}}})) &= \log(\det(\boldsymbol{C}_1\boldsymbol{C}_1^\top)) \\ &= 2\log(\det(\boldsymbol{C}_1)) = 2\sum_i \log(\boldsymbol{C}_1)_{i,i},\end{aligned} \tag{16}$$

where $\boldsymbol{C}_1$ is the lower triangular Cholesky factor of $\boldsymbol{L}_{w_{\mathbb{R}}}$, $(\boldsymbol{C}_1)_{i,i}$ is the i-th diagonal element of $\boldsymbol{C}_1$. Commonly, using (16), the computational complexity of calculating the log-determinant function of a $n_m \times n_m$ sparse matrix will be much smaller than $O(n_m^3)$ for the dense one. This formulation is also suitable for the other weighted Laplacian matrix $\boldsymbol{L}_{w_{SO(n)}}$.

2) Lazy evaluation: We also use lazy evaluations to speed up the algorithm [52]. Instead of computing the objective function for each candidate element $s_{n_p-k} \in \mathcal{V} \setminus \mathcal{N}_a$, the lazy-greedy algorithm only needs to evaluate a part of them. It keeps the upper bounds of the objective functions corresponding to the candidate elements and sorts them in decreasing order. In each iteration, the lazy-greedy algorithm only needs to evaluate the element on top of the list and updates its upper bound. Because of the sub-modularity, for the objective function (12), $\Delta_k = f_{obj}(\mathcal{N}_a \cup \{s_{n_p-k}\}) - f_{obj}(\mathcal{N}_a)$ is the upper bound of the following functions $\Delta_{k+1} = f_{obj}((\mathcal{N}_a \cup \{s^*_{n_p-k}\}) \cup \{s_{n_p-k-1}\}) - f_{obj}(\mathcal{N}_a \cup \{s^*_{n_p-k}\})$, $\Delta_{k+2}, \cdots$. Many elements on the later part of the list do not need to be evaluated, because their upper bounds are too small.

3) Order re-use: In the Cholesky decomposition process, in order to speed up the decomposition, the AMDP is used to permute the rows and columns of the symmetric sparse matrix $\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V})$. Because the structures of $\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V})$ and $\boldsymbol{L}_{w_{SO(n)}}(\mathcal{V})$ are similar, the order vector $\boldsymbol{p}$ obtained by $\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V})$ can also be used in the Cholesky decomposition process of the matrix $\boldsymbol{L}_{w_{SO(n)}}(\mathcal{V})$. We also find that, in every evaluation for the candidate anchors, only several rows and columns of the Laplacian matrix $\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V})$ are different. Hence, the AMDP only needs to be computed once for the initial matrix $\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash\{i\})$, and then the obtained order vector $\boldsymbol{p}$ can be re-used in all Laplacian matrices. So as to keep the dimension of the Laplacian matrix and use the same order vector $\boldsymbol{p}$, the following simple technique is used.

In the k-th iteration, we evaluate the objective function corresponding to the case that the i-th pose is anchored. Assuming the Laplacian matrix $\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash\mathcal{N}_a)$ before anchoring the i-th pose satisfies:

$$\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash\mathcal{N}_a) = \left[\begin{array}{c:c:cc} \boldsymbol{L}^{1,1} & \boldsymbol{l}_1 & \boldsymbol{L}^{1,2} & \\ \hdashline \boldsymbol{l}_1^\top & l & \boldsymbol{l}_2^\top & \}_{i-th} \\ \hdashline \boldsymbol{L}^{2,1} & \underbrace{\boldsymbol{l}_2}_{i-\text{th}} & \boldsymbol{L}^{2,2} & \end{array}\right], \tag{17}$$

we have the new Laplacian matrix $\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash(\mathcal{N}_a \cup \{i\}))$ after anchoring the i-th pose:

$$\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash(\mathcal{N}_a \cup \{i\})) = \begin{bmatrix} \boldsymbol{L}^{1,1} & \boldsymbol{L}^{1,2} \\ \boldsymbol{L}^{2,1} & \boldsymbol{L}^{2,2} \end{bmatrix}. \tag{18}$$

Then, we compute the function $\log(\det(\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash(\mathcal{N}_a \cup \{i\}))))$ after deleting the i-th row and column. However, by this way, because of the decrease of the matrix dimension, the order vector $\boldsymbol{p}$ is also changed, which makes the re-use operation complex. So the following matrix $\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash(\mathcal{N}_a \cup \{i\}))^*$ is used to replace $\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash(\mathcal{N}_a \cup \{i\}))$:

$$\begin{aligned}&\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash(\mathcal{N}_a \cup \{i\}))^* \triangleq \left[\begin{array}{c:c:cc} \boldsymbol{L}^{1,1} & \mathbf{0} & \boldsymbol{L}^{1,2} & \\ \hdashline \mathbf{0} & 1 & \mathbf{0} & \}_{i-th} \\ \hdashline \boldsymbol{L}^{2,1} & \underbrace{\mathbf{0}}_{i-\text{th}} & \boldsymbol{L}^{2,2} & \end{array}\right], \\ &\log(\det(\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash(\mathcal{N}_a \cup \{i\}))^*)) \\ &= \log(\det(\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash(\mathcal{N}_a \cup \{i\})))) + \log(\det(1)).\end{aligned} \tag{19}$$

Using (19) to replace (18), the dimensions of all Laplacian matrices in the greedy method keep constant, so the order vector $\boldsymbol{p}$ can be used for multiple times without additional operations. The computational time of the AMDP will be reduced. Meanwhile, benefiting from the order re-use, the rank-1 update can be used in the following sub-section.

4) Rank-1 update: Based on two reduced Laplacian matrices (17) and (19), let

$$\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash\mathcal{N}_a) = \begin{bmatrix} \boldsymbol{C}_{11} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{c}_{21} & c_{22} & \mathbf{0} \\ \boldsymbol{C}_{31} & \boldsymbol{c}_{32} & \boldsymbol{C}_{33} \end{bmatrix} \begin{bmatrix} \boldsymbol{C}_{11}^\top & \boldsymbol{c}_{21}^\top & \boldsymbol{C}_{31}^\top \\ \mathbf{0} & c_{22} & \boldsymbol{c}_{32}^\top \\ \mathbf{0} & \mathbf{0} & \boldsymbol{C}_{33}^\top \end{bmatrix} \tag{20}$$

be the Cholesky factorization of the matrix $\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash\mathcal{N}_a)$ and let

$$\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash(\mathcal{N}_a\cup\{i\}))^* = \begin{bmatrix}\bar{\boldsymbol{C}}_{11} & \boldsymbol{0} & \boldsymbol{0}\\ \bar{\boldsymbol{c}}_{21} & \bar{c}_{22} & \boldsymbol{0}\\ \bar{\boldsymbol{C}}_{31} & \bar{\boldsymbol{c}}_{32} & \bar{\boldsymbol{C}}_{33}\end{bmatrix}\begin{bmatrix}\bar{\boldsymbol{C}}_{11}^\top & \bar{\boldsymbol{c}}_{21}^\top & \bar{\boldsymbol{C}}_{31}^\top\\ \boldsymbol{0} & \bar{c}_{22} & \bar{\boldsymbol{c}}_{32}^\top\\ \boldsymbol{0} & \boldsymbol{0} & \bar{\boldsymbol{C}}_{33}^\top\end{bmatrix} \tag{21}$$

be the Cholesky factorization of the matrix $\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash(\mathcal{N}_a\cup\{i\}))^*$, we have: $\bar{\boldsymbol{C}}_{11}=\boldsymbol{C}_{11}$, $\bar{\boldsymbol{C}}_{31}=\boldsymbol{C}_{31}$, $\bar{\boldsymbol{c}}_{21}=\boldsymbol{0}$, $\bar{c}_{22}=1$, $\bar{\boldsymbol{c}}_{32}=\boldsymbol{0}$,

$$\begin{aligned}\boldsymbol{C}_{31}\boldsymbol{C}_{31}^\top+\boldsymbol{c}_{32}\boldsymbol{c}_{32}^\top+\boldsymbol{C}_{33}\boldsymbol{C}_{33}^\top&=\bar{\boldsymbol{C}}_{31}\bar{\boldsymbol{C}}_{31}^\top+\bar{\boldsymbol{C}}_{33}\bar{\boldsymbol{C}}_{33}^\top\\ \Rightarrow\ \bar{\boldsymbol{C}}_{33}\bar{\boldsymbol{C}}_{33}^\top&=\boldsymbol{c}_{32}\boldsymbol{c}_{32}^\top+\boldsymbol{C}_{33}\boldsymbol{C}_{33}^\top.\end{aligned} \tag{22}$$

Because $\boldsymbol{c}_{32}$ is a non-zero vector and $\boldsymbol{C}_{33}\boldsymbol{C}_{33}^\top$ is a full-rank positive matrix, we can find that the equation (22) to obtain $\bar{\boldsymbol{C}}_{33}\bar{\boldsymbol{C}}_{33}^\top$ belongs to rank-1 update of a sparse Cholesky factorization. So the objective function can be written as:

$$\begin{aligned}&\log(\det(\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash(\mathcal{N}_a\cup\{i\}))^*))\\ &=\log(\det(\underbrace{\boldsymbol{C}_{11}\boldsymbol{C}_{11}^\top}_{\text{Re-use}}))+\log(\det(\overbrace{\underbrace{\boldsymbol{c}_{32}\boldsymbol{c}_{32}^\top}_{\text{Re-use}}+\underbrace{\boldsymbol{C}_{33}\boldsymbol{C}_{33}^\top}_{\text{Re-use}}}^{\text{Rank 1 update}})).\end{aligned} \tag{23}$$

In every iteration, we need to traverse all possible candidate anchor $\{i\}\in\mathcal{V}\backslash\mathcal{N}_a$ for the objective function $\log(\det(\boldsymbol{L}_{w_{\mathbb{R}}}(\mathcal{V}\backslash(\mathcal{N}_a\cup\{i\}))^*))$. Using (23), in the same iteration, the sparse Cholesky decomposition can be performed only once and use it in every evaluation, which greatly reduces the computational cost. Because of the order re-use and the re-use of the Cholesky decomposition result, based on "CSparse" package, the whole rank-1 update can be finished very quickly.

Based on these technologies, the improved method is summarized by the pseudo-code in Algorithms 3 and 4.

D. Computational complexity analysis

Due to the technologies presented in Section V-C, the computational complexity of the improved method is reduced. At the k-th iteration, because of the sparseness, the computational complexity of the rank-1 update of the Cholesky factorization of the n_p-k+1 dimension matrix $\boldsymbol{c}_{32}\boldsymbol{c}_{32}^\top+\boldsymbol{C}_{33}\boldsymbol{C}_{33}^\top$ is generally lower than $O((n_p-k+1)^2)$. Without considering the lazy evaluation, in each iteration, the dimensions of the matrix $\boldsymbol{c}_{32}\boldsymbol{c}_{32}^\top+\boldsymbol{C}_{33}\boldsymbol{C}_{33}^\top$ change from n_p-k+1 to 1. For one iteration, the whole computational complexity is $O((n_p-k+1)^2+\cdots+1^2)$. Assuming that the equivalent efficiency of the lazy evaluation can be simplified as: $\omega\in(0,1)$, the computational complexity of the whole method will be:

$$\sum_{k=2}^{N}\omega O((n_p-k+1)^2+\cdots+1^2)\approx O(\frac{\omega N}{3}n_p^3). \tag{24}$$

Our new method is much faster than the original 1-step greedy algorithm $O(\frac{\omega N}{3}n_p^3) << O\left(30Nn_p^4\right)$. This conclusion will be validated by the numerical simulation in Section VII-B.

Algorithm 3 Improved lazy greedy algorithm using sparse Cholesky decompositon, order re-use, and rank-1 update

```
Require: Two weighted Laplacian matrices $L_{w_\mathbb{R}}$ and $L_{w_{SO(n)}}$; the fixed anchored poses number $N$
Ensure: The sub-optimal anchored poses set $\mathcal{N}_a$
//Based on Theorem 2
$\mathcal{N}_a \leftarrow \{1\}$;
//Choose a fill-reducing permutation heuristic $p$
$p \leftarrow COLAMD(L_{w_\mathbb{R}}(\mathcal{V}\backslash\mathcal{N}_a))$;   ▷ e.g., Column approximate minimum degree
//Sparse Cholesky factor based on $p$
$C_1 \leftarrow SparseCholesky\,(L_{w_\mathbb{R}}(\mathcal{V}\backslash\mathcal{N}_a)(p,p))$;
$C_2 \leftarrow SparseCholesky\left(L_{w_{SO(n)}}(\mathcal{V}\backslash\mathcal{N}_a)(p,p)\right)$;
$f_{obj}(\mathcal{N}_a) \leftarrow 2\cdot n\cdot\sum_i\log(C_1)_{i,i}+2\cdot d\cdot\sum_i\log(C_2)_{i,i}$;
$f_{value} \leftarrow f_{obj}(\mathcal{N}_a)$;   ▷ Apply to compute upper bound for lazy evaluation
//1-step improved greedy algorithm
while $|\mathcal{N}_a| \leq N$ do   ▷ Cardinality-fixed constraint
    $C_1 \leftarrow SparseCholesky\,(L_{w_\mathbb{R}}(\mathcal{V}\backslash\mathcal{N}_a)^*(p,p))$;
    $C_2 \leftarrow SparseCholesky\left(L_{w_{SO(n)}}(\mathcal{V}\backslash\mathcal{N}_a)^*(p,p)\right)$;
    for $j = |\mathcal{N}_a|+1:1:n_p$ do
        //Lazy evaluation
        if $\Delta_m > f_{index}$ then break;
        else
            $m = Order(j)$;▷ Lazy evaluation for largest several solutions $f_{obj}(\mathcal{N}_a\cup\{s_m\})$;
            $f_{obj}(\mathcal{N}_a \cup \{s_m\}),\ s_m \in \mathcal{V}\setminus\mathcal{N}_a \leftarrow Rank1\_reuse(C_1, C_2)$;   ▷ Algorithm 4
            $\Delta_m \leftarrow f_{value} - f_{obj}(\mathcal{N}_a\cup\{s_m\})$;   ▷ Upper bounds
            if $\Delta_m < f_{index}$ then $f_{index} \leftarrow \Delta_m$;
            $\mathcal{S} \leftarrow \mathcal{S}\cup\{s_m\}$;
    end
    $s_k^* \leftarrow \max_{s_m\in\mathcal{S}} f_{obj}(\mathcal{N}_a\cup\{s_m\})$;
    $Order \leftarrow sort(\Delta_m, \text{all } m)$;   ▷ Sort in ascending order
    $f_{value} \leftarrow f_{obj}(\mathcal{N}_a\cup\{s_k^*\})$;
    $\mathcal{N}_a \leftarrow \mathcal{N}_a\cup\{s_k^*\}$;
    $k \leftarrow k+1$;
return $\mathcal{N}_a$.   ▷ Selected set with good performance
```

Algorithm 4 Objective function using order re-use and rank-1 update technology ($Rank1\_reuse$)

```
Require: Cholesky factorization of the matrices $L_{w_\mathbb{R}}(\mathcal{V}\backslash\mathcal{N}_a)^*$ and $L_{w_{SO(n)}}(\mathcal{V}\backslash\mathcal{N}_a)^*$ : $C_1$ and $C_2$; the order of potential anchor $\{s_k\}$
Ensure: The objective function $f_{obj}(\mathcal{N}_a\cup\{s_k\})$
$C_{11} \leftarrow C_1(1:s_k-1, 1:s_k-1), c_{32} \leftarrow C_1(s_k, s_k+1:end), C_{33} \leftarrow C_1(s_k+1:end, s_k+1:end)$;
$C_{11}^* \leftarrow C_2(1:s_k-1, 1:s_k-1), c_{32}^* \leftarrow C_2(s_k, s_k+1:end), C_{33}^* \leftarrow C_2(s_k+1:end, s_k+1:end)$;
$v \leftarrow cs\_etree(C_{33}, 'col')$;   ▷ Return the elimination tree of $C_{33}^\top C_{33}$.
$f_{obj}(\mathcal{N}_a\cup\{s_k\}) \leftarrow 2n\sum_j\log(C_{11})_{jj}+2d\sum_j\log(C_{11}^*)_{jj}+2n\sum_j\log(c_{32}c_{32}^\top+C_{33}C_{33}^\top)_{jj}+2d\sum_j\log(c_{32}^*{c_{32}^*}^\top+C_{33}^*{C_{33}^*}^\top)_{jj}$;   ▷ Rank-1 update.
return $f_{obj}(\mathcal{N}_a\cup\{s_k\})$.
```

E. Discussion of near optimal performance

In this subsection, we discuss the near optimal performances of the greedy-based methods. Because of the sub-modularity and Cauchy Interlacing Theorem [51], the obtained solution of this optimization problem is limited by some bounds.

Theorem 4. *The objective function $f_{obj}\left(\mathcal{N}_a\right)$ has upper/lower* *bounds:*

$$\begin{aligned}&\mathcal{F}_{LB} \le f_{obj}\left(\mathcal{N}_a\right) \le \mathcal{F}_{UB}\\ &\mathcal{F}_{LB} = n\sum_{i=1}^{n_p-N} \log\lambda_i(\boldsymbol{L}_{w_{\mathbb{R}}}) + d\sum_{i=1}^{n_p-N}\log\lambda_i(\boldsymbol{L}_{w_{SO(n)}})\\ &\mathcal{F}_{UB} = n\sum_{i=N+1}^{n_p}\log\lambda_i(\boldsymbol{L}_{w_{\mathbb{R}}}) + d\sum_{i=N+1}^{n_p}\log\lambda_i(\boldsymbol{L}_{w_{SO(n)}}),\end{aligned} \tag{25}$$

where $\lambda_1(\boldsymbol{L}_{w_{\mathbb{R}}}), \cdots, \lambda_{n_p}(\boldsymbol{L}_{w_{\mathbb{R}}})$ *and* $\lambda_1(\boldsymbol{L}_{w_{SO(n)}}), \cdots, \lambda_{n_p}(\boldsymbol{L}_{w_{SO(n)}})$ *are the eigenvalues of* $\boldsymbol{L}_{w_{\mathbb{R}}}$ *and* $\boldsymbol{L}_{w_{SO(n)}}$ *respectively sorted by the increasing order.*

Proof: See Appendix C in supplementary material [43]. ■

Corollary 1. *The objective function* $f_{obj}\left(\mathcal{N}_a\right)$ *has bounds:*

$$\sum_{i=1}^{n_d(n_p-N)}\log\lambda_i(\boldsymbol{\mathcal{L}}) \le f_{obj}\left(\mathcal{N}_a\right) \le \sum_{i=n_dN+1}^{n_dn_p}\log\lambda_i(\boldsymbol{\mathcal{L}}), \tag{26}$$

where $n_d = n + d$, $n_p \ge 1$, *the eigenvalues of* $\boldsymbol{\mathcal{L}}$ *are sorted by the increasing order and named as* $\lambda_1(\boldsymbol{\mathcal{L}}), \cdots, \lambda_{n_dn_p}(\boldsymbol{\mathcal{L}})$.

Proof: It can be directly obtained by Theorem 4. ■

Based on the bounds (25), using the random greedy algorithm, we can get the tighter bounds for the optimal solution:

Theorem 5. *Let* $\mathcal{U}_{opt}$, $\mathcal{U}_{greedy}$, *and* $\mathcal{U}^*_{greedy}$ *be the optimal value of* (12), *the objective function value obtained by the lazy greedy algorithm, and the mathematical expectation of the objective function value achieved by the random greedy algorithm, respectively. The following inequality holds:*

$$\begin{aligned}&\mathcal{U}_{LB} \le \mathcal{U}_{opt} \le \mathcal{U}_{UB},\ where\\ &\mathcal{U}_{LB} = \max\left\{\mathcal{U}_{greedy}, \mathcal{F}_{LB}\right\}\\ &\mathcal{U}_{UB} = \min\left\{\left(\frac{1-\frac{N}{en_p}}{e} - \varepsilon\right)^{-1}\mathcal{U}^*_{greedy}, \mathcal{F}_{UB}\right\}.\end{aligned} \tag{27}$$

Proof: The value $\mathcal{U}_{greedy}$ obtained from the lazy greedy algorithm is naturally the lower bound of the optimal solution $\mathcal{U}_{opt}$. Meanwhile, because of the approximation factor of the random greedy algorithm, we have that $\left(\frac{1-\frac{N}{en_p}}{e} - \varepsilon\right)^{-1}\mathcal{U}^*_{greedy}$ is the upper bound of the optimal solution $\mathcal{U}_{opt}$. Combining Theorem 4, we can finally get (27). The proof is completed. ■

Based on Theorem 5, we have the bounds to ensure the near-optimal performance of the solution. All these bounds in Theorem 4, Theorem 5, and Corollary 1 are the extensions of the results in [45] where only an upper bound is proven. Because of the randomness, the natural greedy algorithm commonly has a better performance than the random greedy algorithm. So in practice, the natural greedy algorithm is used to get a good result and meanwhile the random greedy algorithm is used to get the bounds if it is required.

VI. Two important applications

We present the landmark setting in 2D/3D mapping task and the trajectory assignment for CSLAM based on the proposed method, since they are two important applications in robotics.

A. Landmarks setting in 2D/3D mapping

In some highly-accurate mapping tasks, the pose-graph optimization is widely used as an important tool to improve the accuracy and the loop-closure detecting ability. However, in the large-scale feature-less SLAM problem, like DARPA Subterranean Challenge [7], it is very difficult to obtain a very accurate global pose estimation in pose-graph SLAM using only one anchor by fixing the initial pose. For example, the state-of-the-art technologies, including RTAB-map [53] and Cartographer [54], can only reach 10∼50cm accuracy in 100∼300m trajectory using MIT Stata Center data set.

The landmark setting with ground truth is a simple and efficient way to improve the accuracy of the pose-graph SLAM. A landmark will offer accurate global information to the nearby poses in the whole pose-graph, which is similar to anchoring these poses. Fig. 5 presents an illustration on the reduced weighted Laplacian matrices of the base pose graph, the anchored pose-graph, and the pose-feature graph with anchored landmarks. The rows and columns of the Laplacian matrix corresponding to the anchored poses are deleted, which is equal to the infinite information saving in the diagonal block, as shown in the red diagonal sub-matrix. In (c), the added landmark commonly brings some new information to the connected pose and its corresponding diagonal elements will add a new yellow diagonal sub-matrix. When the weight of the measurement between landmark and the pose tends to infinity, the information increment of anchoring the landmark will approach the one of anchoring the pose.

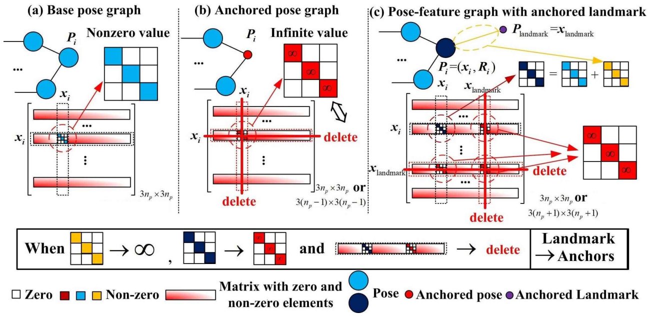


Fig. 5: The information involving in the pose P_i corresponding to different graphs: (a) Base pose graph and its corresponding weighted Laplacian matrix; (b) Pose graph with the anchored pose P_i and its weighted Laplacian matrix deleting the corresponding rows and columns; (c) Pose-feature graph with the anchored feature $P_{landmark}$ and its weighted Laplacian matrix deleting the rows and columns corresponding to $P_{landmark}$.

However, the landmark setting usually needs many external calibration operations with some precise instruments, such as ETS, and high human resource cost, so a good landmark setting should use fewer landmarks and yet obtain high quality map. In the following, we will explain how to apply our anchor selection method to perform the landmark setting using twice-trajectory approach and redundant landmarks approach.

Firstly, we propose the twice-trajectory approach to apply our anchor selection method to the landmark setting. As shown in Fig. 6, following a designed path, the robot first collects the data based on the on-board sensors, including laser data, camera data, and so on, without using anchors. Based on this collected data, we can output its corresponding pose graph and select the optimal anchors using the anchor selection framework. Then, the landmarks are set based on the selected anchors. The robot moves back to the origin, follows the path, and collects the related data with these calibrated landmarks again. Because of the similar environment and path, the measurements, especially the pose-graph structure, of two data collection processes will be similar[9]. Based on the static landmarks set by the selected anchors, the occupancy grid map is built with better accuracy using the SLAM tool-pack, like Cartographer[10]. The anchor selection process becomes the preparation to choose the positions of the landmarks and further improves the accuracy of the pose-graph result.

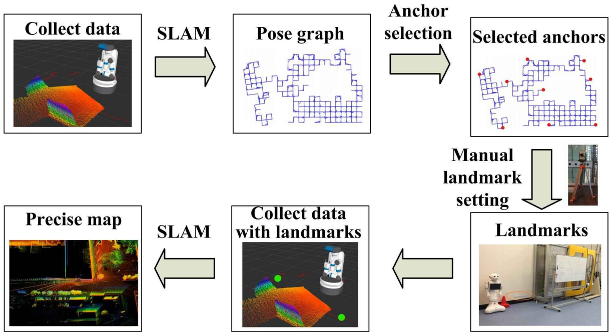


Fig. 6: Landmark setting operations using the twice-trajectory approach.

The specific operations of the other way, called as redundant landmarks approach, are shown as follows. First, before the SLAM task, many redundant marks with unknown global location are placed along the trajectory. An ideal setting way is to ensure that each pose in the obtained pose graph can measure only one set mark. In practice, the marks can be placed as uniform as possible without considering whether each pose can detect the marks. Second, the SLAM task is performed by following the designed path. During this process, all detected marks are tracked, mapped, and involved in the SLAM method. Then, based on the obtained pose-graph[11], our proposed anchor selection method is used to pick out the optimal anchors. Following, the marks detected by these selected poses need to be calibrated and the calibrated results will be regarded as the additional measurements and be introduced into the original SLAM measurement network. Finally, with the help of these additional calibrated measurements, the SLAM problem is solved again to get a better solution. Compared with the twice-trajectory approach, this approach only needs to follow the designed path once without repeating. The disadvantage is that many more marks need to be placed because of the large number of the robot poses[12].

B. Trajectory assignment in CSLAM

CSLAM [31] is a framework to use several robots to finish the SLAM task cooperatively in an unknown space, which means that every robot first obtains a sub-map using the classical SLAM framework based on its initial pose and finally a global map is obtained by fusing all the sub-maps. A natural question is, how to assign the trajectories for different robots such that they can obtain a better SLAM result?

In some CSLAM tasks, the initial poses of the robots are known in a common global frame, which means the initial robot poses can be regarded as multiple anchors of the pose-graph SLAM problem. In this situation, the trajectory assignment problem for the robots becomes a similar problem to find the best anchored poses. The anchor selection for CSLAM can be performed using the following steps. Firstly, only one robot[13] is used to follow the complete designed path including all places which need to be visited by all robots. After the pose graph is built, the optimal anchored poses can be selected using the anchor selection method, given the number of the robots as the number of the anchors. Then, the static initial poses of the robots are calibrated using some additional measurements. The path is divided into multiple parts by the anchored poses and each one between every two adjacent anchors is assigned to the corresponding robot[14]. Finally, the robots move from their own origins to follow the assigned paths and complete the CSLAM tasks.

VII. Simulation results

In this section, we will evaluate the performance of the proposed method using some well-known datasets with real and synthetic scenarios. All simulations are performed using python (Section VII-F3) and MATLAB (other sub-sections) on a Dell E5570 laptop with an Intel Core i5-6300U 2.40 GHz processor and 8 GB of RAM running Ubuntu 16.04.6 LTS (Section VII-F3) and Windows 7 (other sub-sections).

A. Different number of anchored poses

To show the relationship between the accuracy of the pose-graph SLAM result and the number of the anchored poses, based on Intel dataset (1728 poses) [26], we increase the number of the anchored poses, choose them by the improved greedy method, and finally get the estimated results using the

[9] Because of the sensor noise and the slight environment change, even though the robot follows the same path twice, practically, we know that it is impossible to obtain the exactly same pose graphs in two trajectories. The corresponding differences between two pose graphs will cause the potential deterioration of the proposed method's performance. However, because the main structures of two different pose graphs are highly similar, the affect of the performance deterioration is limited. Meanwhile, the twice-trajectory method will introduce the additional cost in the robot motion and the additional calibration operations. Compared with the operations to improve the used SLAM method, these costs are commonly acceptable especially for the industrial users, who do not want to deeply revise the SLAM methods.

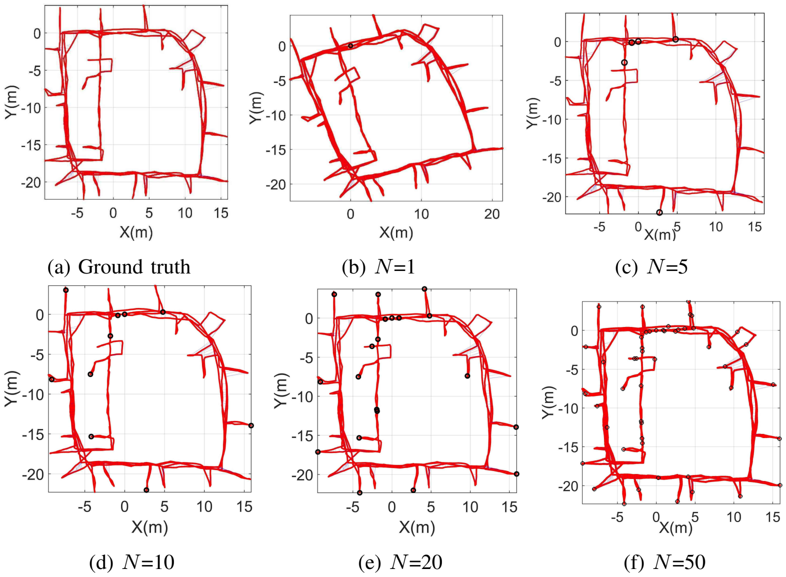
[10] It is noted that our anchor selection framework can be used not only for Cartographer, but also applied in different SLAM methods using pose-graph optimization, including ORB-SLAM2 [1], RTAB-Map [53], and so on.

[11] Most well-known mature laser/visual SLAM methods, like cartographer and ORB-SLAM2, use the pose-graph optimization. They also have ability to track some un-calibrated marks, like the 'landmark' interface in Cartographer.

[12] Because of its disadvantage, the implement of the redundant landmarks approach is relatively difficult, in our latter simulation and experiment, if we do not specifically mention the way to realize the landmark setting, the default way is the twice-trajectory approach.

[13] Because of the resource and scenario constraints, in some tasks, multiple robots need to be used to perform the CSLAM task. We can perform CSLAM task using multiple robots with unknown initial poses to replace the task using only one robot with only one initial pose.

[14] In order to avoid some paths of some robots are much shorter than the others, the potential pose set for anchors can be constrained such that the lengths of different paths are all similar.



(a) Ground truth (b) N=1 (c) N=5

(d) N=10 (e) N=20 (f) N=50

Fig. 7: Estimated results using the Intel dataset (red line) based on the best selected anchors (black circles) obtained by the greedy-based method.

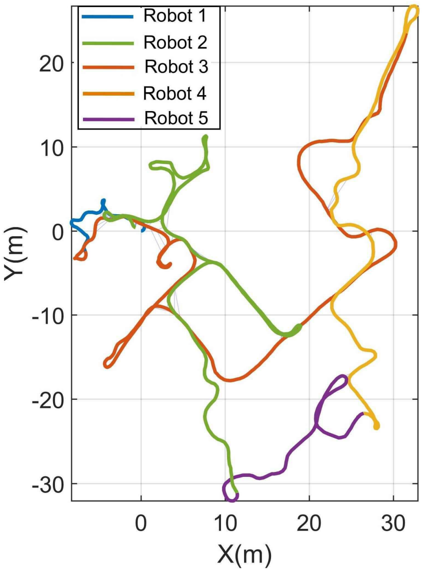


Fig. 8: Five robots perform coorperative SLAM using CSAIL dataset.

anchored poses, as shown in Fig. 7. The SLAM solution used as ground truth is first obtained using SE-sync [26], which is a state-of-the-art algorithm, with outstanding computational efficiency (comparable speed to other highly optimized libraries, like GTSAM [21] and g2o [20]) and certifiable global optimality. Then, random noises for the translation and rotation measurements obeying the isotropic Gaussian distribution and the isotropic Langevin distribution are generated using the normrnd MATLAB function and the Acceptance-Rejection Method respectively [55]. The precisions of the translation measurements δ_{ij} are between 9.5094×10^1 to 3.3203×10^2 with a mean value 1.5309×10^2 and concentrations of the rotation measurements κ_{ij} are between 1.1551×10^2 to 2.7815×10^2 with a mean value 1.4224×10^2. These noises are added to the relative measurements of the edge data using the measurement model (1). As mentioned in Section II, the approximation method, by adding very accurate measurements between the origin and the anchored poses, is applied in the SE-sync method to solve the new anchored pose graph SLAM without revising the toolbox. For the estimated result, we compute the absolute trajectory error (ATE), which means the mean value of root squared error of the position of the estimated poses with respect to that of ground truth.

Fig. 7a is the ground truth of the Intel dataset. It can be seen from Fig. 7b that the SLAM result is poor when only one anchor is applied. The estimated results tend to be more accurate if the number of the anchored poses N increases from 5 to 50, which are shown from Fig. 7c to Fig. 7f. Their ATEs are 2.3727 m (Fig. 7b, 1 anchor), 2.9514×10^{-1} m (Fig. 7c, 5 anchors), 1.7361×10^{-1} m (Fig. 7d, 10 anchors), 7.5231×10^{-2} m (Fig. 7e, 20 anchors), 2.7199×10^{-2} m (Fig. 7f, 50 anchors) respectively. These results show that our method greatly improves the result accuracy by selecting more anchors. This simulation also illustrates that multiple anchors are necessary for the large-scale SLAM task with high-level noisy data, since it is sometimes difficult to achieve the accuracy requirement using one anchor, especially when limited sensors can be accessed. Instead of revising the used SLAM technology or presenting the new SLAM framework, our framework serves as an additional tool for the used SLAM method to set the anchors and improve the final results.

B. *Comparison with the normal greedy method*

So as to verify the efficiency of the tools in Section V-C, we compare our proposed method with the normal greedy method. Because our method will not change the result obtained by the normal greedy method, in this part, we only show the advantage in terms of the computational efficiency to verify the analyses of the computational complexity shown in (24). In Table I, we compare them by using several well-known real and synthetic datasets, including 2D: CSAIL, intel, manhattan, KITTI, city10000, and ais2klinik; 3D: tinyGrid3D, smallGrid3D, torus3D, cubicle, grid3D, and rim [26].

In Table I, the 'greedy method' represents Algorithm 1 only using the sparse Cholesky decomposition as shown in Section V-C1. The 'improved method' means that all speed-up tools shown in Section V-C are applied. Based on Theorem 2, if only one anchor is introduced, its choice will not affect the accuracy of the final SLAM results. Hence, we use the first pose as the first anchor and, for the first anchor, the result only shows the computational time to compute the log-determinant function of its FIM. Under the circumstances with the same final results, our method is much faster than the normal greedy method. Even for some large datasets containing more than 10^4 poses, e.g. city10000, ais2klinik, and rim datasets, the proposed method can solve them within several minutes. So as to further show the efficiency of different tools in Section V-C, using Intel and CSAIL datasets, we apply these tools one by one, and then show the computational time changing with their applications in Fig. 9.

In Fig. 9, the results show that all additional operations, presented in Section V-C, can greatly reduce the computational time. In fact, the sparse Cholesky decomposition has the largest effect (more than two orders of magnitude, like 13.39

TABLE I: Comparison of computational time using different SLAM datasets

			Mean computational time in 10 runs[s]				
	Dataset	# Poses	N=1	N=5	N=10	N=20	N=50
Greedy method (2D)	CSAIL	1045	0.0584	3.6349	7.8533	16.1398	40.4455
	Intel	1728	0.0602	15.4858	34.5448	72.0651	183.0900
	manhattan	3500	0.0585	86.7457	187.9391	401.7045	1110.5978
	KITTI	4541	0.0571	57.2313	139.8398	294.8835	727.6991
	city10000	10000	0.0606	1298.3256	2927.0964	6117.0702	15535.6554
	ais2klinik	15115	0.0773	991.3454	2132.4369	4578.9760	11764.0283
Greedy method (3D)	tinyGrid3D	9	0.0892	0.0209	-*	-	-
	smallGrid3D	125	0.0840	0.2126	0.4268	0.7817	1.4366
	torus3D	5000	0.0606	595.9520	1347.1887	2710.1122	6289.2731
	cubicle	5750	0.0811	537.1241	1196.3851	2211.4586	6077.1242
	grid3D	8000	0.0571	5618.1917	10152.7144	>20000	>20000
	rim	10195	0.0870	2582.7873	5372.1122	11421.1211	>20000
Improved method (2D)	CSAIL	1045	0.0144	0.7732	0.9865	1.2732	1.7657
	Intel	1728	0.0175	3.0832	4.0620	4.9781	5.8554
	manhattan	3500	0.0322	7.5656	13.1060	16.8376	21.3019
	KITTI	4541	0.0215	10.1819	14.0718	18.5387	28.9056
	city10000	10000	0.0948	63.2231	64.1240	142.4568	180.8735
	ais2klinik	15115	0.0508	100.2318	122.2698	169.7197	238.3487
Improved method (3D)	tinyGrid3D	9	0.0930	0.0475	-	-	-
	smallGrid3D	125	0.0705	0.1965	0.2387	0.2762	0.3965
	torus3D	5000	0.0716	70.6634	73.1187	83.2794	94.5362
	cubicle	5750	0.0590	76.8378	78.3145	84.4654	102.0602
	grid3D	8000	0.0871	1216.2893	1804.8911	1953.7712	2107.2340
	rim	10195	0.2131	290.2561	295.7529	308.2754	321.0302

* '-' means the computation is infeasible, because the pose number is smaller than the anchor number.

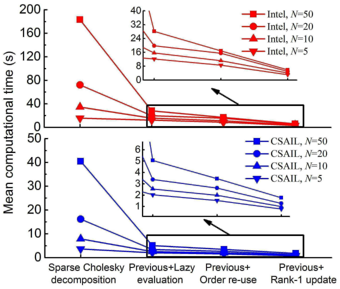


Fig. 9: Computational time reduced with the applications of speed-up technologies

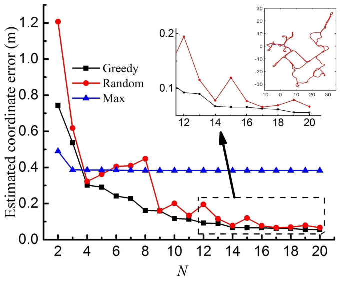


Fig. 10: Comparison with two methods using CSAIL dataset

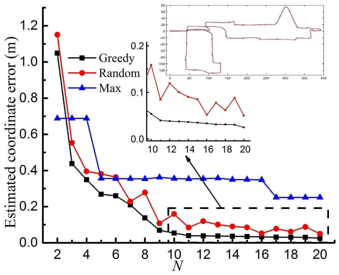


Fig. 11: Comparison with two methods using FR079 dataset.

seconds to 0.02 seconds for Intel dataset) to reduce the computational time, benefiting from the high sparsity of the reduced weighted Laplacian matrices.

C. Comparison with two other methods

For further comparison in terms of the estimated accuracy, our method, named 'Greedy', is compared with two methods, including maximal node degree method and random selection, using CSAIL [26], Fr-clinic, and FR079 datasets [56]. The maximal node degree method, named 'Max', means to choose the nodes which have the maximal weighted node degrees. Because these nodes are strongly connected to the whole network, they may perform the leading roles in the improvement of the result accuracy. The random method, named 'Random', is a cheap but mediocre and unstable approach. For the random method, we run $N_e = 100$ times and select the best result to compare with others. The comparisons of the estimated coordinate errors (same as ATE) are presented in Fig. 10 to Fig. 12 with the ground truth shown in the top-right corner.

We can see that our greedy-based method has the best performance and its performance is stable for different datasets and anchor numbers. Because of the randomness, the accuracy of the results from the random method is not very stable and shows a fair performance. For the maximal node degree method, we find a very interesting phenomenon that, in some stages, the estimated accuracy remains constant with the number growth of the anchors. This is because, in many pose graphs, the poses with large node degrees commonly locate in the similar positions and these anchors can only affect a small range of poses, which cannot make the accuracy of other poses improve. Based on the data in Fig. 10, Fig. 11, and Fig. 12, we define $RE = \text{mean}_{N=2,\cdots,20}\text{Others}/\text{Ours}$ and $RE1 = \max_{N=2,\cdots,20}\text{Others}/\text{Ours}$ by the ATE, where Others and Ours are the ATEs of the poses using different number of anchors corresponding to different methods, respectively. Their results are shown in Table II.

TABLE II: Comparison with the ATEs of two other methods

Superiority	Average (RE)		Maximal ($RE1$)	
Dataset	Random	Max	Random	Max
CSAIL	144.96%*	370.78%	276.87%	687.47%
Fr-clinic	186.16%	260.93%	292.53%	569.81%
FR079	198.06%	604.59%	312.00%	1040.98%

* When the value is bigger than 100%, it means that our method is better.

In Table II, we can see that our method has more than 40% and 176.87% advantage compared with other methods w.r.t the mean and maximal accuracy of the estimated results. Our method can provide a good and stable anchor selection result.

D. Approximation performance verification

In this paper, we approximate the original problem (10) to the new sub-matrix selection problem. To verify the perfor-

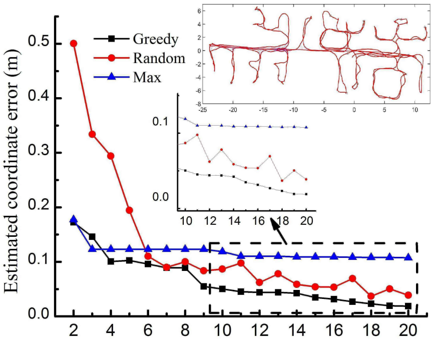


Fig. 12: Comparison with two methods using Fr-clinic dataset.

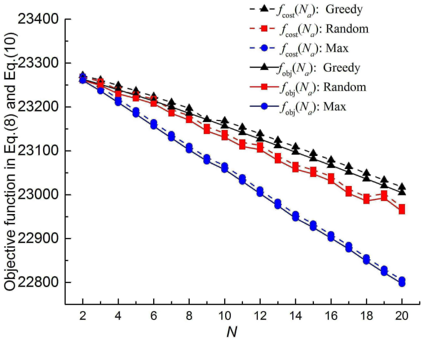


Fig. 13: Approximation performance and comparison with two methods using FR079 dataset.

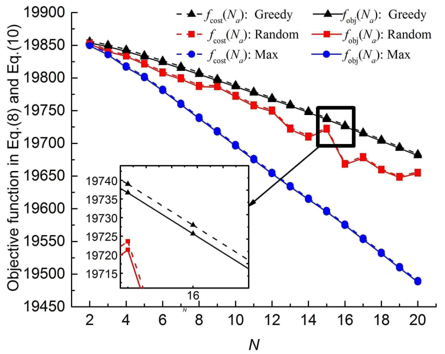


Fig. 14: Approximation performance and comparison with two methods using CSAIL dataset.

mance of the approximation (12) and compare the uncertainty level of the obtained results using different methods, we apply our Greedy, Random, and Max methods to CSAIL and FR079 datasets and present their objective functions.

The approximation and comparison results are shown in Fig. 13 and Fig. 14. It is shown that the obtained objective functions $f_{cost}(\mathcal{N}_a)$, used in (12), are very close to the objective functions of (10) $f_{obj}(\mathcal{N}_a)$. The result not only shows the high approximate performance of the new objective function, which justifies the application of problem (12) to approximate problem (10), but also validates that our method achieves smaller estimate uncertainty (larger D-optimality metric) compared with the other methods.

E. Performance bounds

In this part, we verify the tightness of our presented performance bounds. We use multiple methods to select anchors and then output the objective functions of the CSAIL dataset, shown in Fig. 15. Based on Theorem 5, we can get the lower and upper bounds of the random greedy method and show that the lazy greedy method locates in the bounds, which means that the optimal solution locates within a smaller area. This area is shown as the shaded area, called "Optimal area".

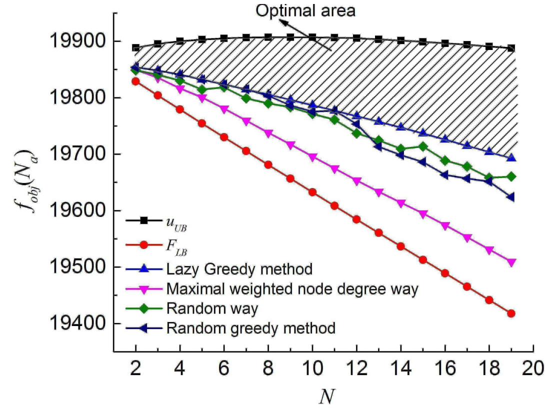


Fig. 15: Bounds from Theorem 5 using CSAIL dataset.

F. Simulation for potential applications

In this subsection, we present the applications of anchor selection in the CSLAM trajectory assignment problem and the landmarks setting of the mapping problems.

1) CSLAM trajectory assignment problem: Based on the anchor selection method, we use 5 robots to share the trajectory of the real CSAIL dataset (1045 poses). Assuming that the initial poses of the five robots are known, the trajectory assignment result is shown in Fig. 8. The final mean coordinate error $RE2 = \sum_{i=1}^{n_p} \|\boldsymbol{x}_i^{opt} - \boldsymbol{x}_i^G\|/n_p$ reduces from 2.1658m (only one robot) to 0.2905m (5 robots with known initial poses), where $\boldsymbol{x}_i^{opt}$ and $\boldsymbol{x}_i^G$ are the optimal solution obtained by the GN method using 5 robots and the ground truth of the dataset respectively. The results show that our method can be used in the CSLAM trajectory assignment problem with the known initial poses to improve the result accuracy.

So as to further validate this application, we present a CSLAM simulation using five quad-rotor unmanned aerial vehicles (UAVs). At first, only one UAV is used to get the initial pose graph. As shown in Fig. 16, in a 7m×7m×1m environment with several regular obstacles (cubics, cylinders, and spheres), a UAV moves from the first pentagram $(0.0, 2.0, 0.2)$ with a velocity 0.1m/s, passes several pre-defined way-points (blue pentagrams), and meanwhile performs the pose-graph SLAM task. In each simulation, the positions of the features are randomly generated. In the moving process of the UAV, the features will be detected when they locate in the sensor range of the UAV (1.5m) and the relative pose measurements are obtained based on the common features detected from two poses. So as to make the measurements follow the noise assumption shown in (1), using the singular value decomposition, the noise-free relative rotation and translation measurements are obtained from the 3D features observations, and then, the random noises are sampled by $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{ij})$ and $Lang(\boldsymbol{I}_{n\times n}, \kappa_{ij})$. The parameters $\kappa_{ij} = 10^3 N_v$ and $\delta_{ij} = 50 N_v$ for visual odometry are set to be proportional to the feature number N_v, which are visible both from P_i and P_j [1]. The noise parameters δ_{ij}^{-2} and κ_{ij} for the control input are 1.6×10^5 and 800 in every step (The simulation time step Δt is set as 1s.). In Fig. 16, the yellow star trajectory is the pose-graph SLAM result using only one UAV with no additional anchors.

Then, based on the generated pose-graph, using multiple anchor selection methods, the starting points of the UAVs are picked out and the trajectories between the adjacent starting points are assigned to the corresponding UAVs. Following, using the external global measurement tools to obtain the initial poses of the UAVs, the collaborative pose-graph SLAM tasks for five UAVs are solved. By randomly generating the different features, we run this simulation 20 times using all methods. One of the results are shown in Fig. 16. The red, black, and blue point trajectories are the estimated results using the random method, the greedy-based method, and the maximal node degree method, respectively. Using the proposed greedy-based method, the trajectory assignment results with five trajectories (blue: 1-st UAV; green: 2-nd UAV; red: 3-rd UAV; Orange: 4-th UAV, and magenta: 5-th UAV) and the generated pose-graph with relative measurements (blue lines) and poses (colorful points) are shown in the top-left of Fig. 16.

The statistical results of the estimated coordinate errors using three methods are presented by the box chart in Fig. 17. The RE compared with the random method and the 'Max' method are 128.49% and 302.92%. The $RE1$ compared with the random method and the 'Max' method are respectively 327.23% and 660.54%. In 20 times, our method shows a consistently better performance over the other methods, which verifies the practicality of our framework in the CSLAM task.

2) Landmarks setting problem in feature-based SLAM: For a feature-based SLAM simulation, we aim to obtain high-quality estimates of 100 randomly-generated features in a 120m × 120m environment by adding 4 landmarks with known locations (since the first pose is also known, this corresponds to 5 anchors), shown in Fig. 19. The sensor range is 15m. Firstly, without adding landmarks (only 1 anchor), we follow the designed path and collect the measurement data. Using the optimization-based feature-based SLAM, the estimated result is relatively poor especially in the boxed areas, which makes the mean error of the mapped features very large (about 0.217m). Based on the pose-graph generated by the relative pose measurements, using our presented method, the optimal anchored poses are selected and the corresponding 4 landmarks are set near the selected anchors, as shown in Fig. 19. We perform the feature-based SLAM by following the path again. The ATE of the final estimated results with the aid of the fixed landmarks is reduced to 0.0097m.

Compared with the unstable random method, the advantage of our method will become more obvious, if the distribution of the poses and the measurements is relatively uneven. In order to show this phenomenon, we change the size of the sampling area to generate the uniformly distributed features. We can apply a variable ξ to control the size of the feature area to be 120ξm×120ξm. After changing the variable $\xi < 1$, we can randomly generate the features in the changed area, and then, we select the anchors by using two methods (the random method and the presented method) and obtain the features using SLAM with landmarks. Finally, the ratios of their estimated results changing with the variable ξ are shown in Fig. 18. Because of the randomness of the features, we test 20 different feature datasets and get the average result.

Our method achieves better performance when ξ is smaller based on Fig. 18. Using 5 anchors, we also show the estimated result with red covariance ellipse using different methods when ξ is small ($\xi = 0.6$) in Fig. 20. The advantage of our method is obvious as shown from the estimated ellipses of the poses.

3) Landmarks setting problem using Cartographer and Fetch simulator: An occupancy grid mapping task using the Fetch simulator and Google Cartographer is implemented to further verify the practicality of our framework (Fig. 6) in a well-known Garage map (Fig. 21). The data are collected from the environment by following a designed path (lower half part) and an initial map is generated using Cartographer, which is shown in Fig. 22. Based on the generated pose-graph, the optimal anchored poses are selected and then the simulated landmarks are set in front of the robot poses (1m away, in front of the robot) (Fig. 23). After we get these landmarks, our following work is to involve these special measurements in the mapping process. There is an interface in Cartographer to track some landmarks without knowing their global locations. In order to anchor them, at the first pose $\boldsymbol{x}_0$, we publish simulated measurements between all selected landmarks and robot poses with large weights, which means the global locations of these landmarks are almost known. Then, after publishing these landmarks, the landmark measurements will be only obtained when the robot locates near to these landmarks. Finally, we can generate a new map using Cartographer with landmarks. The mean errors of relative distance of some corner points A, B, and C, which are pointed out in Fig. 24, in the new map reduce from 0.32m, 0.42m, and 0.78m to 0.21m, 0.31m, and 0.39m, respectively. We also generate a larger map based on this environment. The comparison between the results without using the landmarks, using our method with 5 landmarks, and using the random method with 5 landmarks are shown in Fig. 25 and Table III. These results show that our method helps to find the efficient landmarks and generate an accurate occupancy grid map.

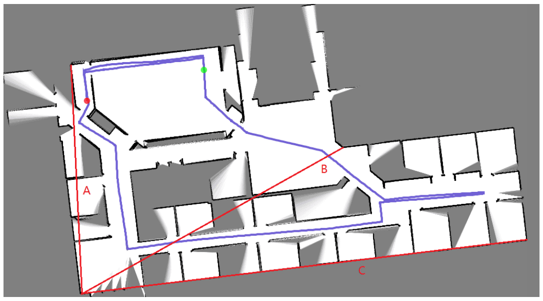


Fig. 24: New map based on five landmarks.

TABLE III: Length comparison of some line segments in obtained maps

Line	Ground truth	Our method	Random	No landmarks
A	42.10 m	**41.90 m**	41.88 m	41.06 m
B	28.86 m	28.54 m	**28.64 m**	28.36 m
C	34.91 m	**34.80 m**	35.05 m	35.07 m
D	19.73 m	19.49 m	**19.55 m**	19.47 m
E	24.28 m	**23.99 m**	23.77 m	23.68 m
F	19.16 m	**19.33 m**	18.70 m	16.70 m
G	26.98 m	**26.89 m**	25.66 m	23.66 m
H	24.59 m	**24.46 m**	23.58 m	21.58 m

VIII. Experimental Results

In this section, some experimental results are presented to validate the practicality of our theoretical developments and evaluate the performance of the method in a real lab environment (UTS Tech Lab). The experiments are performed using C++ on a MSI GL62VR laptop running Ubuntu 16.04.6 LTS as well as on the real Fetch robot. The laser reflectors,

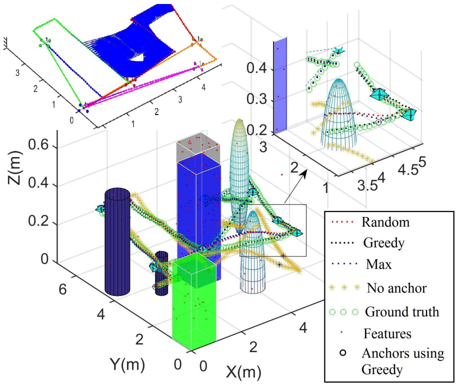


Fig. 16: Estimated result for collaborative pose-graph SLAM.

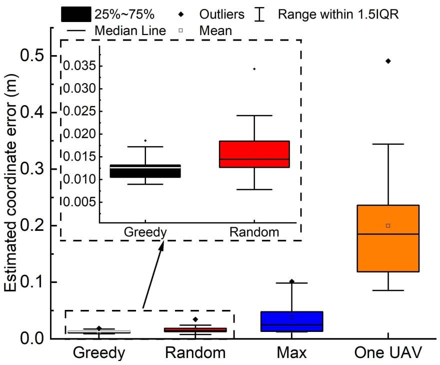


Fig. 17: Box chart of the estimated coordinate errors using different methods.

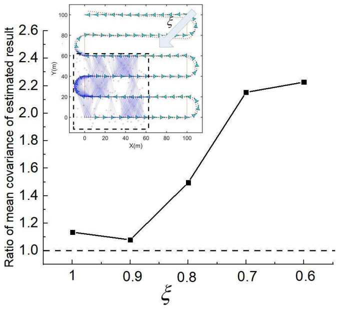


Fig. 18: Ratio of mean ATE of SLAM results changes with ξ (whether it is an relatively uniform pose-graph) using random method and the presented method. The ratio is defined as $\frac{\text{Random}}{\text{Ours}}$. So when values are larger than 1, it means our method can obtain better result (smaller error).

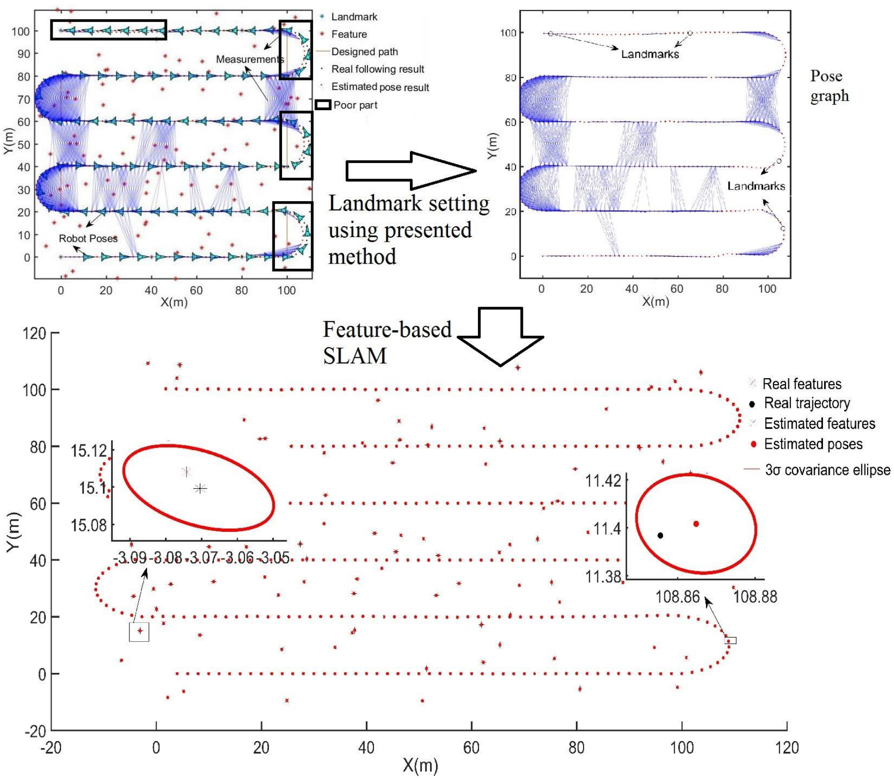


Fig. 19: Operations to obtain good SLAM result using the presented method.

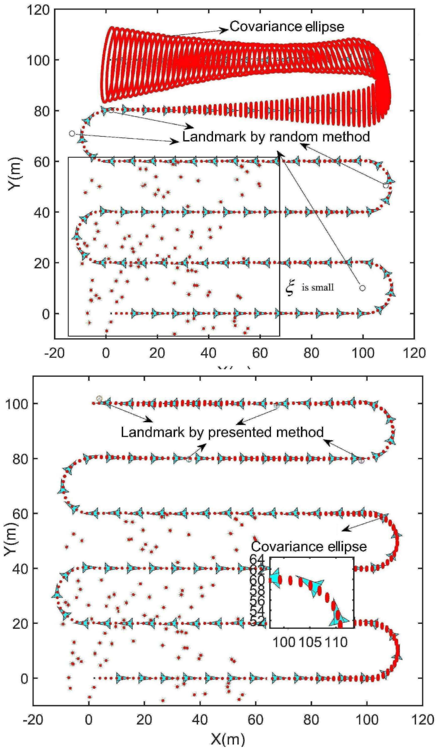


Fig. 20: Comparison of feature-based SLAM result using random method and the presented method, when $\xi = 0.7$. (The features are only sampled in the small square.).

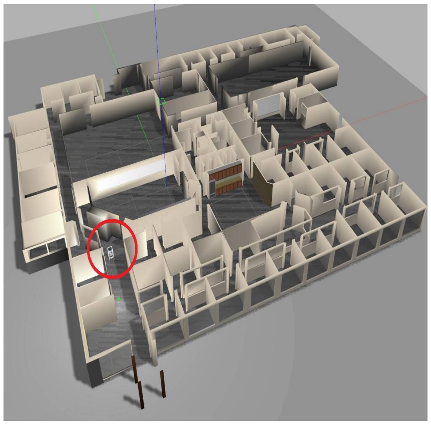


Fig. 21: Fetch simulator in the Willow Garage map.

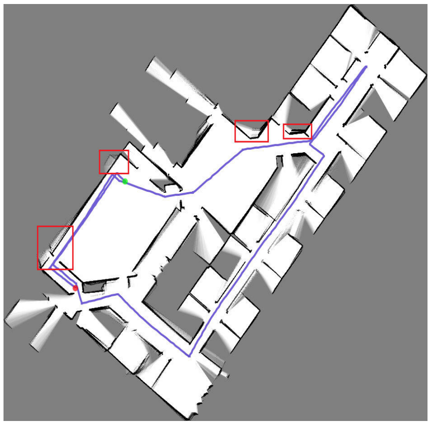


Fig. 22: Cartographer result without using anchors (some poor parts in red boxes).

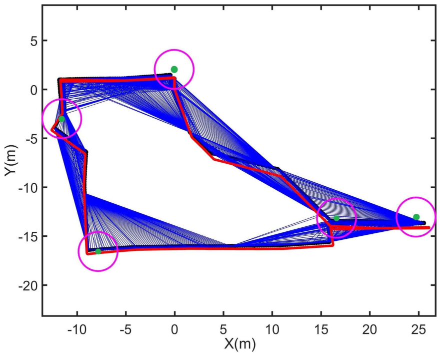
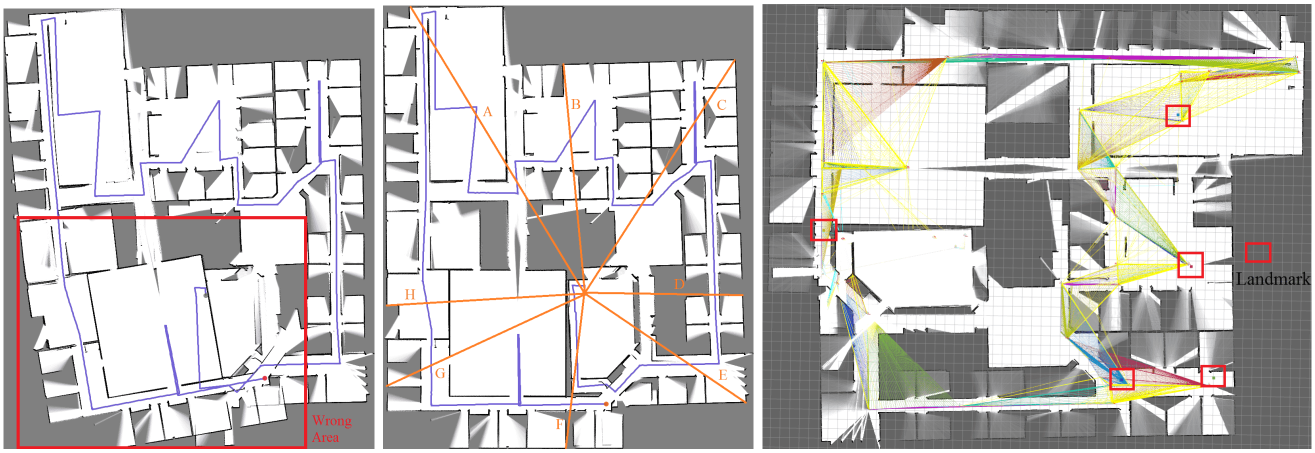


Fig. 23: Pose graph corresponding to Cartogaprapher result (black points and blue edges), ground truth (red points), and selected points for landmarks (green points and magenta circles).

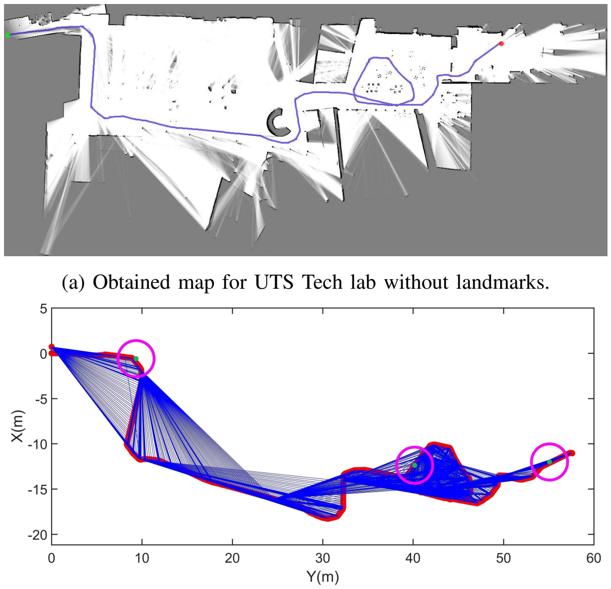


(a) Result without using landmarks. (b) Result with 5 landmarks (our method). (c) 5 selected landmarks using our method.

Fig. 25: Comparison results for the large maps.

which can increase the intensity of the laser scan, are regarded as the identified landmarks for Fetch robot.

We first control the Fetch robot to collect data and implement a mapping process using Cartographer without using the reflectors as landmarks. In a relatively large environment, the SLAM result is not accurate using Cartographer only in Fig. 26a. Using the greedy-based anchor selection method, 3 key poses are selected with the largest increment of the tree-connectivity, which is shown in Fig. 26b, and then, set the landmarks with the laser reflectors by the theodolite. Based on the similar trajectory and landmarks, we can achieve a better Cartographer result (Fig. 27). Using the same collected data, we also output the result without landmarks. The distances of four line segments shown in Fig. 27 are measured by a tape measure and compared with the results of pure Cartographer. Table IV shows that the results of our method are closer to the actual measurements. Photographs of the data collection process using the Fetch robot are shown in Fig. 28.


(a) Obtained map for UTS Tech lab without landmarks.

(b) Pose graph (red points and blue edges) and selected points (green points and magenta circles) for landmarks using Fetch robot.

Fig. 26: Cartographer result without using landmarks and obtained landmarks

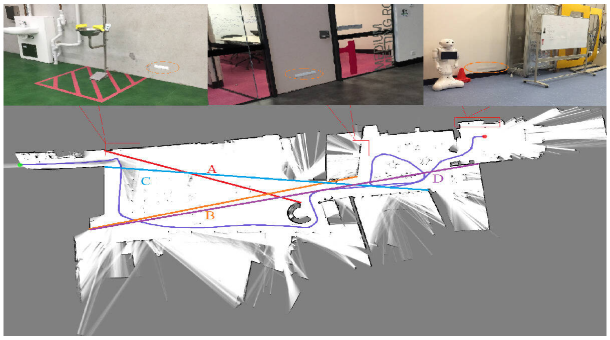


Fig. 27: Obtained occupacy grid map based on 3 landmarks.

TABLE IV: Length comparison and error percentage of some line segments in obtained occupacy grid maps

Line	Ground Truth	Using Landmarks	No Landmarks
A	26.345m	**25.92m (1.61%)**	24.48m (7.08%)
B	33.885m	**34.11m (0.66%)**	34.38m (1.46%)
C	40.623m	**40.45m (0.43%)**	40.41m (0.52%)
D	48.823m	**48.09m (1.50%)**	47.59m (2.53%)

(a) Theodolite. (b) Data collection. (c) Fetch robot.

Fig. 28: Experiments using Fetch robot.

With the help of the landmarks, the map error reduces a lot, which shows the efficiency of the proposed method in the real mapping task in the lab environment.

IX. Conclusion and Future Work

This paper presents a sub-modular optimization framework to solve the anchor selection problem in the 2D/3D pose-graph SLAM problem with multiple anchors. Based on the graph topology, the anchor selection problem using the original D-optimality metric is approximated as the sub-matrix selection problem for the weighted Laplacian matrix. Then, this new sub-matrix selection problem is proved to be a non-negative non-normalized non-monotone sub-modular optimization with the cardinality-fixed constraint. In order to solve it, the natural greedy method and the random greedy method are presented. The performance bounds of the optimal solution are presented and discussed. We also propose multiple tools to improve the running-time ability of the presented method. Several applications using our method are put forward. Finally, the simulation and experimental results illustrate that our method is able to select more efficient anchors and build more accurate maps compared with some other methods.

The results presented in this paper are only the first step towards the use of graph-based anchor selection in areas such as landmark setting, CSLAM, and so on. We are trying to use the sub-modular optimization based anchor selection method and its related graph topology in non-rigid structure from motion [57], deformable visual SLAM, planning, and decision making pipelines. In the landmark setting application, a more intuitive idea is to extend our framework into feature-based SLAM, which will be a good potential research direction in the future. Further studies on finding the optimal number and the best distribution of the anchors to reach the given level of accuracy, will be another interesting research direction.

Acknowledgment

We would like to thank Dr. Kasra Khosoussi, who has answered many of our queries about his research [14]. We also would like to thank Dr. Jiaheng Zhao and Mr. Brenton Leighton, who have helped us to finish our experimental data collection. We thank Mr. Brian Lee for his valuable insights that considerably improved this work. Finally, we would like to thank the anonymous reviewers for their constructive feedback, which greatly improves the overall quality of the paper.

References

[1] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, “ORB-SLAM: A versatile and accurate monocular SLAM system,” *IEEE Trans. on Robot.*, vol. 31, no. 5, pp. 1147-1163, Oct. 2015.

[2] O. Ozyesil, N. Sharon, and A. Singer, “Synchronization over Cartan motion groups via contraction,” *SIAM J. on Applied Algebra and Geometry*, vol. 2, no. 2, pp. 207-241, Apr. 2018.

[3] L. Liu, T. Zhang, B. Leighton, L. Zhao, S. Huang, and G. Dissanayake, “Robust global structure from motion pipeline with parallax on manifold bundle adjustment and initialization,” *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 2164-2171, Apr. 2019.

[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 Trans. on Robot.*, vol. 32, no. 6, pp. 1309-1332, Dec. 2016.

[5] L. Zhao, S. Huang, and G. Dissanayake, “Linear slam: Linearising the SLAM problems using submap joining,” *Automatica*, vol. 100, pp. 231-246, Feb. 2019.

[6] L. Flexline, “TS02/06/09.(2008)”. Leica Geosystems Accessed, Feb.10 2017.

[7] “DARPA Subterranean (SubT) Challenge,” [Online]. Available: https: //www.subtchallenge.com.

[8] K. Ebadi, Y. Change, M. Palieri, A. Stephens, A. H. Hatteland, E. Heiden, A. Thakur, N. Funabiki, B. Morrell, S. Wood, L. Carlone, and A. Agha-mohammadi, “LAMP: Large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments,” in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2020.

[9] S. Dai, “Alibaba joins Baidu, Tencent in China’s autonomous car race,” South China Morning Post, 16-Apr-2018. [Online]. Available: https://www.scmp.com/tech/china-tech/article/2141954/alibaba-confirms-self-driving-car-tests-joining-baidu-and-tencent.

[10] R. A. Bailey and P. J. Cameron, “Combinatorics of optimal designs,” *Surveys in Combinatorics*, vol. 365, pp. 19-73, 2009.

[11] S. Huang, N. M. Kwok, G. Dissanayake, and Q. P. Ha, “Multi-step look-ahead trajectory planning in SLAM: Possibility and necessity,” in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 1091-1096, May 2005.

[12] M. Rodríguez-Arévalo, J. Neira, and J. Castellanos, “On the importance of uncertainty representation in active SLAM,” *IEEE Trans. on Robot.*, vol. 34, no. 3, pp. 829-834, Jun. 2018.

[13] Y. Chen, S. Huang, L. Zhao, and G. Dissanayake, “Cramér-rao bounds and optimal design metrics for pose-graph SLAM,” *IEEE Trans. on Robot.*, vol. 37, no. 2, pp. 627-641, Apr. 2021.

[14] K. Khosoussi, M. Giamou, G. S. Sukhatme, S. Huang, G. Dissanayake, and J. P. How, “Reliable graph topologies for SLAM,” *Inter. J. Robot. Res.*, vol. 38, no. 2-3, Mar. 2019.

[15] L. Carlone and G. C. Calafiore, “Convex relaxations for pose graph optimization with outliers,” *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 1160-1167, Jan. 2018.

[16] L. Carlone, “A convergence analysis for pose graph optimization via gauss-newton methods,” in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 965-972.

[17] K. Konolige, G. Grisetti, R. Kümmerle, W. Burgard, B. Limketkai, and R. Vincent, “Efficient sparse pose adjustment for 2D mapping,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 22-29.

[18] D. M. Rosen, M. Kaess, and J. J. Leonard, “RISE: An incremental trust-region method for robust online sparse least-squares estimation,” *IEEE Trans. on Robot.*, vol. 30, no. 5, pp. 1091-1108, Jun. 2014.

[19] E. Olson, J. Leonard, and S. Teller, “Fast iterative alignment of pose graphs with poor initial estimates,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, May 2006, pp. 2262-2269.

[20] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “G2o: A general framework for graph optimization,” in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 3607-3613.

[21] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, “iSAM2: Incremental smoothing and mapping using the Bayes tree,” *Int. J. Robot. Res.*, vol. 31, no. 2, pp. 216-235, Feb. 2012.

[22] V. Ila, L. Polok, M. Solony, and P. Svoboda, “SLAM++-A highly efficient and temporally scalable incremental SLAM framework,” *Int. J. Robot. Res.*, vol. 36, no. 2, pp. 210-230, Feb. 2017.

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

[24] L. Carlone, G. Calafiore, C. Tommolillo, and F. Dellaert, “Planar pose graph optimization: Duality, optimal solutions, and verification,” *IEEE Trans. on Robot.*, vol. 32, no. 3, pp. 545-565, Jun. 2016.

[25] K. Khosoussi, S. Huang, and G. Dissanayake, “A sparse separable SLAM back-end,” *IEEE Trans. on Robot.*, vol. 32, no. 6, pp. 1536-1549, Dec. 2016.

[26] 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,” *Int. J. Robot. Res.*, vol. 38, no. 2-3, pp. 95-125, Mar. 2018.

[27] V. Indelman, L. Carlone, and F. Dellaert, “Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments,” *Int. J. Robot. Res.*, vol. 34, no. 7, pp. 849-882, Jun. 2015.

[28] T. D. Barfoot and P. T. Furgale. “Associating uncertainty with three-dimensional poses for use in estimation problems,” *IEEE Trans. on Robot.*, vol. 30, no.3, pp. 679-693, Jan. 2014.

[29] K. Khosoussi, S. Huang, and G. Dissanayake, “Novel insights into the impact of graph structure on SLAM,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 2707-2714.

[30] Y. Chen, L. Zhao, K. M. B. Lee, C. Yoo, and R. Fitch, “Broadcast your weaknesses: cooperative active Pose-Graph SLAM for multiple robots,” *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2200-2207, Apr. 2020.

[31] M. Giamou, K. Khosoussi, and J. P. How, “Talk resource-efficiently to me: Optimal communication planning for distributed loop closure detection,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Sep. 2018, pp. 3841-3848.

[32] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. “An analysis of approximations for maximizing submodular set functions-I,” *Math. Programming*, vol. 14, pp. 265-294, 1978.

[33] A. Krause and D. Golovin, Submodular function maximization, 2014.

[34] L. Lovász. Submodular functions and convexity. Math. Programming - State of the Art, pp. 235-257, 1983.

[35] A. Clark, B. Alomair, L. Bushnell, and R. Poovendran, “Toward synchronization in networks with nonlinear dynamics: A submodular optimization framework,” *IEEE Trans. on Auto. Control*, vol. 62, no. 10, pp. 5055-5068, Oct. 2017.

[36] E. Mackin and S. Patterson, “Submodular optimization for consensus networks with noise-corrupted leaders,” *IEEE Trans. on Auto. Control*, vol. 64, no. 7, pp. 3054-3059, July 2019.

[37] S. T. Jawaid and S. L. Smith, “On the submodularity of sensor scheduling for estimation of linear dynamical systems,” in *American Control Conf.*, Jun. 2014, pp. 4139-4144.

[38] H. Zhang, R. Ayoub, and S. Sundaram, “Sensor selection for Kalman filtering of linear dynamical systems: Complexity, limitations and greedy algorithms,” *Automatica*, vol. 78, pp. 202-210, Dec. 2017.

[39] T. H. Summers, F. L. Cortesi, and J. Lygeros, “On submodularity and controllability in complex dynamical networks,” *IEEE Trans. on Control of Net. Sys.*, vol. 3, no. 1, pp. 91-101, Mar. 2016.

[40] V. Tzoumas, N. A. Atanasov, A. Jadbabaie, and G. J. Pappas, “Scheduling nonlinear sensors for stochastic process estimation,” in *American Control Conf.*, Jun. 2017, pp. 580-585.

[41] N. Boumal, A. Singer, P. A. Absil, and V. D. Blondel, “Cramér-Rao bounds for synchronization of rotations,” *Information and Inference: A J. of the IMA*, vol. 3, no. 1, pp. 1-39, Mar. 2013.

[42] Wolfram, (2001), Modified Bessel function of the first kind: Integral representations. http://functions.wolfram.com/03.02.07.0007.01.

[43] Y. Chen, L. Zhao, Y. Zhang, S. Huang, and G. Dissanayake, “Anchor selection for SLAM based on graph topology and sub-modular optimization (Supplementary Material),” *IEEE Trans. on Robot.*, 2020.

[44] S. Chaiken and D. J. Kleitman, “Matrix tree theorems,” *J. of combinatorial theory, Series A*, vol. 24, no. 3, pp. 377-381, 1978.

[45] C. Ko, J. Lee, and M. Queyranne, “An exact algorithm for maximum entropy sampling,” *Oper. Res.*, vol. 43, no. 4, 684-691, Aug. 1995.

[46] A. K. Kelmans and B. N. Kimelfeld, “Multiplicative submodularity of a matrix’s principal minor as a function of the set of its rows and some combinatorial applications”, *Discrete Math.*, vol. 44, pp. 113-116, 1983.

[47] Y. Tian, K. Khosoussi, M. Giamou, J. P. How, and J. Kelly, “Near-optimal budgeted data exchange for distributed loop closure detection”, in *Robotics: Science and Systems (RSS)*, Jun. 2018, pp. 1-12.

[48] L. Carlone and S. Karaman, “Attention and anticipation in fast visual-inertial navigation,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Jul. 2017, pp. 3886-3893.

[49] J. Gillenwater. Maximization of non-monotone submodular functions. 2014.

[50] G. Calinescu, C. Chekuri, M. Pal, and J. Vondrak, “Maximizing a submodular set function subject to a matroid constraint”, Int. *Conf. on Integer Programming and Combinatorial Optim.*, 2007.

[51] G. H. Golub and L. C. F. Van, Matrix computations, vol. 3. JHU Press, 2012.

[52] M. Minoux, (1978) “Accelerated greedy algorithms for maximizing submodular set functions,” *Optimization techniques*, Springer, pp. 234-243.

[53] M. Labbe and F. Michaud, “Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and longterm online operation,” *J. of Field Robot.*, vol. 36, no. 2, pp. 416-446, Oct. 2019.

[54] W. Hess, D. Kohler, H. Rapp, and D. Andor, “Real-time loop closure in 2D LIDAR SLAM,” in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2016, pp. 1271-1278.

[55] B. D. Flury. “Acceptance-rejection sampling made easy,” *SIAM Review*, vol. 32, no.3, pp. 474-476, Sep. 1990.

[56] L. Carlone, R. Aragues, J. Castellanos, and B. Bona, “A fast and accurate approximation for planar pose graph optimization,” *Int. J. Robot. Res.*, vol. 33, no. 7, pp.965-987, May 2014.

[57] Y. Chen, L. Zhao, Y. Zhang, and S. Huang, “Dense Isometric Non-Rigid Shape-From-Motion Based on Graph Optimization and Edge Selection,” *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5889-5896, Oct. 2020.

[58] N. Buchbinder, M. Feldman, J. S. Naor, and R. Schwartz, “Submodular maximization with cardinality constraints”, in *Proc. 25 annual ACM-SIAM symposium on Discrete algorithms*, Jan. 2014, pp. 1433-1452.

Yongbo Chen received the bachelors in engineering from Beijing Institute of Technology (BIT), Beijing, China, in 2012. He is now a Ph.D. candidate with the Centre for Autonomous Systems (CAS), Faculty of Engineering and Information Technology (FEIT), University of Technology Sydney (UTS), Sydney, Australia. His research interests include UAV motion/mission planning, mobile robots simultaneous localization and mapping (SLAM), and non-rigid structure from motion (NRSfM).

Liang Zhao received the Ph.D. degree in photogrammetry and remote sensing from Peking University, Beijing China, in January 2013. From 2012–2014, he was a Postdoctoral Research Fellow in CAS, FEIT, UTS, Australia. From 2014–2016, he worked as a Postdoctoral Research Associate in the Hamlyn Centre for Robotic Surgery, Department of Computing, Faculty of Engineering, Imperial College London, United Kingdom. He is currently a Senior Lecturer in CAS, UTS since July 2016. His current research interests include surgical robotics, SLAM, monocular SLAM, aerial photogrammetry, optimization techniques in mobile robot localization and mapping and image guide robotic surgery.

Yanghao Zhang received the bachelors in science from Northeastern University (NEU), Shenyang, China in 2014. He then received the masters in engineering from the same university in 2017. He is now a Ph.D. candidate with the Centre for Autonomous Systems, Faculty of Engineering and Information Technology, University of Technology Sydney (UTS), Sydney, Australia. His research interests include visual SLAM and deformation reconstruction.

Shoudong Huang received the Ph.D. degree in automatic control from Northeastern University, Shenyang, China, in 1998. He is currently an Associate Professor with the Centre for Autonomous Systems, Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, Australia. His research interests include mobile robots simultaneous localization and mapping (SLAM), robot path planning and control.

Gamini Dissanayake graduated in mechanical/production engineering from University of Peradeniya, Peradeniya, Sri Lanka, and received the B.Sc. (Eng), M.Sc. degree in machine tool technology and the Ph.D. degree in mechanical engineering from University of Birmingham, Birmingham, U.K., in 1981 and 1985, respectively. He is the James N Kirby Distinguished Professor of mechanical and mechatronic engineering with UTS, Sydney, Australia. His research interests include SLAM, navigation systems, dynamics, and control of mechanical systems, cargo handling, optimization, and path planning.