“© 2020 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

Broadcast your weaknesses: cooperative active pose-graph SLAM for multiple robots

Yongbo Chen[1], Liang Zhao[1], Ki Myung Brian Lee[1], Chanyeol Yoo[1], Shoudong Huang[1] and Robert Fitch[1]

***Abstract*—In this paper, we propose a low-cost, high-efficiency framework for cooperative active pose-graph simultaneous localization and mapping (SLAM) for multiple robots in three-dimensional (3D) environments based on graph topology. Based on the selection of weak connections in pose graphs, this method aims to find the best trajectories for optimal information exchange to repair these weaknesses opportunistically when robots move near them. Based on tree-connectivity, which is greatly related to the D-optimality metric of the Fisher information matrix (FIM), we explore the relationship between measurement (edge) selection and pose-measurement (node-edge) selection, which often occurs in active SLAM, in terms of information increment. The measurement selection problem is formulated as a submodular optimization problem and solved by an exhaustive method using rank-1 updates. We decide which robot takes the selected measurements through a bidding framework where each robot computes its predicted cost. Finally, based on a novel continuous trajectory optimization method, these additional measurements collected by the winning robot are sent to the requesting robot to strengthen its pose graph. In simulations and experiments, we validate our approach by comparing against existing methods. Further, we demonstrate online communication based on offline planning results using two unmanned aerial vehicles (UAVs).**

***Index Terms*—Cooperative Active SLAM, Path Planning for Multiple Mobile Robots or Agents, Multi-Robot Systems.**

I. Introduction

MULTI-ROBOT systems can overcome the resource and performance limitation of a single robot, which is useful for applications such as environmental monitoring [1], exploration [2], perception [3], and active SLAM [4]. In many such applications, the environment is unknown and the robots must both collaboratively localize themselves and map the environment through SLAM. For most tasks, it is desirable, if not necessary, to maintain low localization uncertainty.

We are interested in cooperative active SLAM, where the aim is to choose multi-robot trajectories such that the measurements obtained can optimally reduce the uncertainty of the SLAM results. In our view, a practical active SLAM framework must also allow for performing other tasks, which has been overlooked so far in previous work (see, e.g. [5], [6]). In addition, existing communication systems are subject to many challenges, including energy constraints, bandwidth, and range limitations [7]. To this end, we propose a cooperative active SLAM framework that requires only a small amount of information exchange in improving the pose-graph SLAM result.

Manuscript received: September, 11, 2019; Revised November, 11, 2019; Accepted January, 20, 2020.

This paper was recommended for publication by Editor Editor A. Name upon evaluation of the Associate Editor and Reviewers' comments.

[1]Y. Chen, L. Zhao, K. M. B. Lee, C. Yoo, S. Huang and R. Fitch are with Centre for Autonomous Systems (CAS), Faculty of Engineering and Information Technology (FEIT), University of Technology Sydney, Ultimo, NSW, 2007 Australia Yongbo.Chen@student.uts.edu.au

Digital Object Identifier (DOI): see top of this page.

A. Related work

Popular frameworks in this area include model predictive control (MPC) [6], which can only reach a locally optimal solution, and the partially observably Markov decision process (POMDP) [8], which is known to be computationally intractable. These frameworks aim to choose the best future trajectory using A-, D-, or T-optimality metrics from optimal experiment design theory [9]. Two main categories of such strategies are *sampling-based* and *dynamic-based* methods.

Sampling-based approaches [10], [11], [12] discretize the belief space with randomized waypoints to realize a finite problem space with acceptable performance bounds. In [13], a probabilistic roadmap (PRM) is used to generate candidate paths with multi-robot constraints in both obstacle-free and obstacle-unknown environments. In [14], the authors introduce topological belief space planning (BSP), which uses the topological properties of the underlying factor graph over the future posterior beliefs to direct the search for an optimal multi-robot active SLAM solution. The main advantages of sampling-based methods are computational efficiency and the decoupling of path planning and information metric computation. Due to dynamic constraints, however, it is challenging to apply these methods to real robots directly.

In dynamic-based methods, cooperative active SLAM is usually formulated as a stochastic constrained optimal control problem that is commonly solved over a constant time-horizon [15], [16]. One example is extended Kalman filter (EKF)-based active SLAM for multiple robots using the A-optimality metric for trajectory selection [6], minimizing both the localization error and the corresponding uncertainty bounds. Its main limitation is that it is centralized and computationally expensive. To reduce running-time, [5] presents a multi-sensor active SLAM method to decentralize the entire control task to achieve linear computational complexity and provide sub-optimality guarantees. The main idea is that each robot receives the planning results of other robots with higher priority and combines them in solving its own problem. To improve practicality from the computation and communication perspectives, the authors of [17] propose a method for anytime distributed information gathering that provides sub-optimality guarantees. Even though dynamic-based methods consider

motion models, the scenarios considered are usually very simple. Typical problem settings involve a 2D obstacle-free environment with high-performance multi-hop communication, or all-to-all communication. These settings limit practicality for real multi-robot systems.

B. Overview of the approach

The aim of our cooperative active pose-graph SLAM framework is to collaboratively plan trajectories that reduce pose uncertainty in order to assist the completion of initial tasks. We mostly retain the designed path of the original task, while executing additional trajectories as needed to resolve weak connection points in the pose graph. Compared to the MPC framework [5], our framework is suitable for various tasks because the robot trajectory is modified minimally and only a small amount of communication data is required.

As shown in Fig. 1, suppose that a group of robots are initially tasked with visiting several pre-defined waypoints in an unknown indoor environment. Throughout operation, when the pose uncertainties are large, the weak connections in their pose graphs are quickly identified and sent as potential targets to the common communication network. When several robots are available in the vicinity of the potential targets, these robots compute the cost of obtaining the measurement that correspond to the potential targets. The most suitable robot is found using a bidding framework based on the computed costs, and a dynamically feasible trajectory is found between the current pose of the winning robot and the target while considering the resource limitation. Then, the winning robot visits the target by following the planned trajectory. The target measurements are obtained and sent to the requesting robot to build a stronger pose graph. The winning robot then resumes its original task.

C. Paper organization and contributions

This paper presents an active pose-graph SLAM framework for multi-robot systems using graph topology and submodular optimization. The problem formulation and the graph topology analysis are presented and discussed in Sec. II and Sec. III respectively. In Sec. IV, a bidding-based cooperative active SLAM method using continuous refinement trajectory planning is presented. Finally, in Sec. V, simulations and experiments are presented to verify the practicality of the proposed framework and its communication efficiency, running time, and estimation accuracy. Compared with our previous work on single robot active SLAM based on a combination of tree-connectivity and node degree [18], and other frameworks [6], [5], the main contributions of this paper are:

- ◇ Exploration of the informational relationship between measurement selection and pose-measurement selection, which leads to the application of the 1-ESP problem.
- ◇ Novel efficient bidding framework for finding optimal cooperative targets to visit and communication policies with respect to low data transmission.
- ◇ Providing an applied trajectory optimization mechanism using support vector machine (SVM) and continuous-time refinement in cooperative active SLAM.
- ◇ Application of multiple speed-up techniques in the proposed framework, including fast covariance recovery, the branch and bound method, and rank-1 updates.

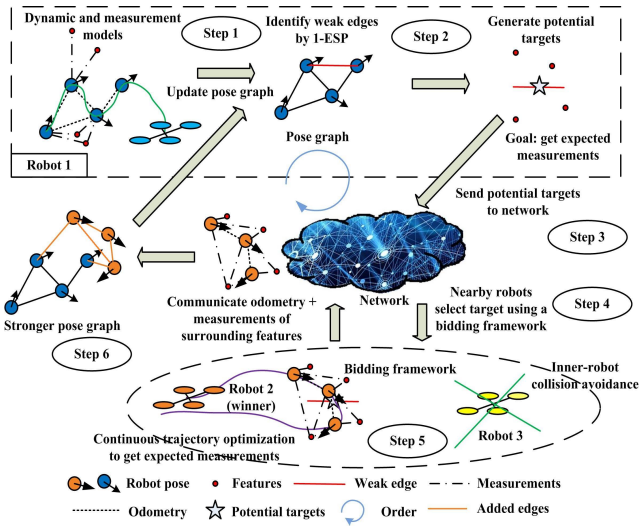


Fig. 1. Structure of our cooperative active SLAM method

II. Problem Formulation

A. Graph preliminaries

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ be a weighted, weakly-connected directed graph over n_p nodes, $\mathcal{V} = \{1, \cdots, n_p\}$, and m_p edges, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, $|\mathcal{E}| = m_p$. In the pose-graph SLAM problem, every node $i \in \mathcal{V}$ of the graph corresponds to a robot pose P_i, and every edge $e_k = (i_k, j_k) \in \mathcal{E}$ denotes the k-th relative measurement between two robot poses P_{i_k} and P_{j_k}. We assign positive weights $w : \mathcal{E} \to \mathbb{R} > 0$ to each edge, such that the weight matrix $\boldsymbol{\Sigma} = \text{diag}\{\omega(e_1), \omega(e_2), \cdots, \omega(e_{m_p})\}$ represents the covariance matrix of the measurements.

We denote the set of successor poses of pose i by V_i^+. In other words, $j \in V_i^+$ iff $(i, j) \in \mathcal{E}$. Let $\widetilde{\boldsymbol{A}} \in \{-1, 0, 1\}^{n_p \times m_p}$ be the incidence matrix of $\mathcal{G}$, defined by $\widetilde{\boldsymbol{A}}_{ik} = -1$ (resp. 1) if $e_k = (i, j)$ (resp. (j, i)) for some $j \in \mathcal{V}$, and $\widetilde{\boldsymbol{A}}_{ik} = 0$ otherwise. For an arbitrary choice of $i \in \mathcal{V}$, let $\boldsymbol{A} \in \{-1, 0, 1\}^{(n_p-1) \times m_p}$ be the matrix obtained by removing i-th row from $\widetilde{\boldsymbol{A}}$. The reduced weighted Laplacian matrix of $\mathcal{G}$ is then given by $\boldsymbol{\mathcal{L}}_w^{\mathcal{G}} = \boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^\top$.

B. Synchronization on $\mathbb{R}^n \times SO(n)$ and robot models

Pose-graph SLAM is an instance of a synchronization problem on $\mathbb{R}^n \times SO(n)$, $n = 2,\ 3$. The aim is to estimate the values of n_p unknown poses $P = \{P_1, \cdots, P_{n_p}\}$, $P_i \in \mathbb{R}^n \times SO(n)$, given m_p noisy measurements of relative rotations $\boldsymbol{D}_{ij}$ and relative translations $\boldsymbol{p}_{ij}$. These measurements are modelled as [19]:

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

where $\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma}_{ij})$, $\boldsymbol{\Sigma}_{ij} = \delta_{ij}^2 \boldsymbol{I}_{n\times n}$ is the multivariate isotropic Gaussian distribution, and $Lang(\boldsymbol{I}_{n\times n}, \kappa_{ij})$ is the isotropic Langevin distribution with mean $\boldsymbol{I}_{n\times n}$ and concentration $\kappa_{ij} \geq 0$.

Given a set of noisy measurements $\boldsymbol{p}_{ij}$ and $\boldsymbol{D}_{ij}$, the aim of the pose-graph SLAM problem is to obtain a maximum-likelihood estimate of the poses $P_i = (\boldsymbol{x}_i, \boldsymbol{R}_i), i = 1 \cdots n_p$:

$$\max_{P_1 \cdots P_{n_p}} \sum_{(i,j)\in\mathcal{E}} \kappa_{ij} tr(\boldsymbol{D}_{ij}\boldsymbol{R}_i\boldsymbol{R}_j^\top) - \frac{\delta_{ij}^{-2}}{2}\|\boldsymbol{p}_{ij} - \boldsymbol{R}_i^\top(\boldsymbol{x}_j - \boldsymbol{x}_i)\|_2^2. \tag{2}$$

In this paper, we use the SE-sync algorithm [19] to solve (2). It is a state-of-the-art algorithm, with outstanding computational efficiency and certifiable global optimality. With the computational enhancements featured in its latest version, it exhibits comparable speed to other highly optimized libraries, like GTSAM [20] and g2o [21].

The dynamic model of the robot corresponding to the measurement model (1) is defined as:

$$\begin{aligned} \boldsymbol{x}_{i+1} &= \boldsymbol{x}_i + \boldsymbol{R}_i \triangle \boldsymbol{x}_{i(i+1)} + \widetilde{\boldsymbol{y}}_{i(i+1)},\ \widetilde{\boldsymbol{y}}_{i(i+1)} \sim \mathcal{N}(\boldsymbol{0}, \widetilde{\boldsymbol{\Sigma}}_{ij}) \\ \boldsymbol{R}_{i+1} &= \widetilde{\boldsymbol{Z}}_{i(i+1)} \triangle \boldsymbol{R}_{i(i+1)} \boldsymbol{R}_i,\ \widetilde{\boldsymbol{Z}}_{i(i+1)} \sim Lang(\boldsymbol{I}_{n\times n}, \widetilde{\kappa}_{ij}), \end{aligned} \tag{3}$$

where $\widetilde{\boldsymbol{\Sigma}}_{ij} = \widetilde{\delta}_{ij}^2 \boldsymbol{I}_{n\times n}$, $\triangle \boldsymbol{x}_{i(i+1)} = (v_r \Delta t,\ \boldsymbol{0}_{1\times 2})^\top$ and $\triangle \boldsymbol{R}_{i(i+1)}$ are the control inputs, v_r is the robot velocity, and Δt is the time step. Many unknown features will be observed, which can be used to compute the relative pose measurements when they are within the sensor range R_s.

Assumption 1: All SLAM results are in a common frame of reference. This can be achieved if, e.g., the first poses of all robots are known to each other, or the robots measure common landmarks during the SLAM process.

C. Cooperative active pose-graph SLAM based on D-optimality metric

We are interested in finding a D-optimal solution to the active SLAM problem, where the log-determinant of the covariance matrix $\boldsymbol{C}$ is minimized. It is, however, challenging to compute $\log(\det(\boldsymbol{C}))$ directly, because it is a large, dense matrix. An efficient alternative is to use the sparse FIM $\boldsymbol{\mathcal{I}}$ via a Cramer-Rao lower bound (CRLB)[1] [22]:

$$\log(\det(\boldsymbol{C})) \approx \log(\det(\boldsymbol{\mathcal{I}}^{-1})) = -\log(\det(\boldsymbol{\mathcal{I}})). \tag{4}$$

For 3D pose-graph SLAM (1), the FIM is of the form [26]:

$$\boldsymbol{\mathcal{I}}_{3D} = \left[\begin{array}{c|c} \boldsymbol{L}_w^{\mathbb{R}^3} & \triangle_w^{3D\top} \\ \hline \triangle_w^{3D} & \boldsymbol{L}_w^{SO(3)} + \text{diag}\{\boldsymbol{\Psi}_1, \cdots, \boldsymbol{\Psi}_{n_p}\} \end{array}\right]. \tag{5}$$

Here, $\boldsymbol{L}_w^{SO(3)} = \boldsymbol{\mathcal{L}}_w^{SO(3)} \otimes \boldsymbol{I}_{3\times 3}$ and $\boldsymbol{L}_w^{\mathbb{R}^3} = \boldsymbol{\mathcal{L}}_w^{\mathbb{R}^3} \otimes \boldsymbol{I}_{3\times 3}$, $\boldsymbol{\mathcal{L}}_w^{SO(3)}$ and $\boldsymbol{\mathcal{L}}_w^{\mathbb{R}^3}$ are the weighted Laplacian matrices of the rotation and translation graphs prior to the Kronecker product. It is important to note that $\boldsymbol{\Psi}_i$ and $\triangle_w^{3D}$ depend on the solution of (2), while the others depend on the graph topology only. The full expressions are presented in [26], [27]. The weights $w(e_k) : (i,\ j) \in \mathcal{E}$ for rotation (resp. translation) graph is $\omega_{ij}^{SO(3)} = \frac{\kappa_{ij}^2(2I_0(2\kappa_{ij}) - I_1(2\kappa_{ij}) - 2I_2(2\kappa_{ij}) + I_3(2\kappa_{ij}))}{2I_0(2\kappa_{ij}) - 2I_1(2\kappa_{ij})}$ (resp. $\omega_{ij}^{\mathbb{R}^3} = \delta_{ij}^{-2}$), where $I_p(\cdot)$ is the p-th modified Bessel functions [22].

[1]Because of the non-flatness of $\mathbb{R}^n \times SO(n)$, the exact CRLB is of the form $\boldsymbol{C} \succeq \boldsymbol{\mathcal{I}}^{-1}$ + curvature terms, where curvature terms are caused by higher order terms. As an asymptotic bound, these curvature terms are negligible for small errors.

We now define the cooperative active pose-graph SLAM problem. Suppose we are given g robots, and the dynamic model (3). The aim of cooperative active SLAM is to select actions for each robot such that the D-optimality criterion of the predicted FIM is maximized:

$$\begin{aligned} &\max_{u_v, v=1,\cdots,g} \sum_{v=1}^{g} \log(\det(\boldsymbol{\mathcal{I}}_v)), \\ &s.t. \quad \boldsymbol{\mathcal{I}}_v = f_{pre}(u_v),\ Collision\text{-}free, \end{aligned} \tag{6}$$

where $u_v, v = 1, \cdots, g$ are the candidate actions of the $v-$th robot, $\boldsymbol{\mathcal{I}}_v$ is the predicted FIM after performing action u_v, and $f_{pre}(\star)$ is the FIM prediction function with zero-innovation. Zero-innovation means that the FIM is updated using the predicted robot poses based on the motion equation (3) without the effect of the noise [23]. $Collision\text{-}free$ constraints ensure that the solution trajectory is collision free.

III. Graph topology

In this section, we present links between graph-topology and the D-optimality metric which allow an approximate formulation of (6). We then present new results that relate the edge selection problem to the node-edge selection problems.

A. Graph topology analysis for the D-optimality metric

Interestingly, there is a strong link between the FIM and the topology of the pose graph. Specifically, the FIM can be calculated based on evaluating the *spanning trees* of the pose graph. A spanning tree of a graph $\mathcal{G}$ is a subgraph that is a tree and covers all nodes of $\mathcal{G}$. First, we need the following definitions of weighted spanning trees:

Definition 3.1: (Weighted value of a spanning tree [24]) Let $\mathcal{T}_\mathcal{G}$ be the set of all spanning trees of $\mathcal{G}$. The value $\mathbb{V} : \mathcal{T}_\mathcal{G} \to \mathbb{R}_+$ of a spanning tree $\mathcal{T} \in \mathcal{T}_\mathcal{G}$ is defined by:

$$\mathbb{V}(\mathcal{T}) = \prod_{e\in\mathcal{E}(\mathcal{T})} w(e). \tag{7}$$

where $\mathcal{E}(\mathcal{T})$ represents the set of edges in $\mathcal{T}$.

The weighted number of spanning trees, also called *tree-connectivity* [24], is defined as:

$$t_w(\mathcal{G}) \triangleq \sum_{\mathcal{T}\in\mathcal{T}_\mathcal{G}} \mathbb{V}(\mathcal{T}). \tag{8}$$

A classical result in graph theory is that tree-connectivity is related to the reduced weighted Laplacian matrix as follows:

Theorem 3.1: (Weighted Matrix-Tree Theorem [25]). For a simple weighted graph $\mathcal{G} = (\mathcal{V},\ \mathcal{E},\ w)$ with $w : \mathcal{E} \to \mathbb{R}^+$, we have $t_w(\mathcal{G}) = \det\left(\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{A}^\top\right) = \det(\boldsymbol{\mathcal{L}}_w^\mathcal{G})$.

Further, the determinant of the reduced weighted Laplacian matrix (equivalently, the tree-connectivity), is a close approximation of the D-optimality metric. The precise statement is as follows.

Theorem 3.2 ([26]): Consider the nD ($n = 2, 3$) pose-graph SLAM problem with isotropic Gaussian translation noise and isotropic Langevin rotation noise. Define $\xi \triangleq \max_{i=1,2,\cdots,n_p} \frac{1}{n-1} \sum_{j\in V_i^+} \delta_{ij}^{-2} \|\boldsymbol{x}_j - \boldsymbol{x}_i\|_2^2$, and let $\lambda_{min}(\boldsymbol{L}_w^{SO(n)})$ be the minimal eigenvalue of $\boldsymbol{L}_w^{SO(n)}$. Then, we have

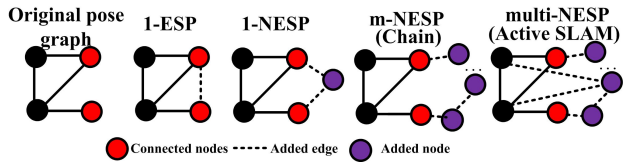


Fig. 2. Original pose graph, measurement-selection problem (1-ESP) and corresponding pose-measurement selection prblem (1-NESP, m-NESP and multi-NESP) with the same connected nodes

the following bound on the approximation error $\varepsilon = \log(\det(\boldsymbol{\mathcal{I}_{nD}})) - \log(\det(\boldsymbol{L}_w^{\mathbb{R}^n})) - \log(\det(\boldsymbol{L}_w^{SO(n)}))$:

$$0 \leq \varepsilon \leq n_p n(n-1)\log(1+\xi/\lambda_{min}(\boldsymbol{L}_w^{SO(n)}))/2. \tag{9}$$

In practical applications, ε is relatively small. We have:

$$\begin{aligned}&\log(\det(\boldsymbol{\mathcal{I}_{nD}})) \approx \log(\det(\boldsymbol{L}_w^{\mathbb{R}^n})) + \log(\det(\boldsymbol{L}_w^{SO(n)}))\\ &\Rightarrow \log(\det(\boldsymbol{\mathcal{I}}_{nD})) \approx n\log(t_w(\mathcal{G}^{\mathbb{R}^n})) + d\log(t_w(\mathcal{G}^{SO(n)})).\end{aligned} \tag{10}$$

where $d = \frac{n(n-1)}{2}$. So we can solve (6) approximately, by maximizing the weighted tree-connectivity of the pose graph.

B. Relationship between measurement selection and pose-measurement selection

A key idea behind our framework is that each robot selects its own weak connection point as a potential target measurement for the other robots. In this section, we discuss the relationship between the measurement selection problem [24], and the pose-measurement selection problem (Fig. 2), which is the case for the active SLAM problem, from an information-theoretic perspective.

In the k measurement selection problem, or, equivalently, the k edge selection problem ($k-$ESP) [24], the aim is to find k additional edges to a given base graph that leads to the highest increase in tree connectivity of the new pose graph.

In this paper, we only consider 1-ESP, the simplest instance of $k-$ESP. The 1-ESP problem is defined as:

$$\max_{e\in\mathcal{E}_c} n\log(t_w(\mathcal{G}^e_{\mathbb{R}^n})) + d\log(t_w(\mathcal{G}^e_{SO(n)})), \tag{11}$$

where $\mathcal{G}^e_{\mathbb{R}^n} = \mathcal{G}^{\mathbb{R}^n} \cup e^{\mathbb{R}^n}$, $\mathcal{G}^e_{SO(n)} = \mathcal{G}^{SO(n)} \cup e^{SO(n)}$ and e is the new edge in $\mathcal{G}$ with $e^{\mathbb{R}^n}$ (resp. $e^{SO(n)}$) the corresponding new edge in translation (resp. rotation) graph.

While active SLAM belongs to the multiple node-edge section problem (multi-NESP), it is not a k-ESP, because we are selecting both nodes and incident edges, rather than only edges. This raises the following questions:

Problem 1: Suppose we add an edge to a base pose graph $\mathcal{G}^0$, as in 1-ESP. Can we find an equivalent addition of a node with two edges, as in 1-NESP, or approximate the difference in terms of tree-connectivity?

Problem 2: Can we find m nodes and $m+1$ or more than m edges, as in m-NESP or multi-NESP, such that the increase in tree-connectivity is approximately equal to the one of the case adding a single edge?

Solving Problems 1 and 2 implies that we can pose the active SLAM problem as a 1-ESP problem, which is easier to formulate and solve. The following theorems answer Problems 1 and 2.

Theorem 3.3: Let $\mathcal{G}^e$, $\mathcal{G}^e_{\mathbb{R}^n}$ and $\mathcal{G}^e_{SO(n)}$ be the pose, translation and rotation graphs after adding a new edge with weight ω_1 to $\mathcal{G}^0$, and let $\mathcal{G}^n$, $\mathcal{G}^n_{\mathbb{R}^n}$ and $\mathcal{G}^n_{SO(n)}$ be the ones obtained by adding a new node and two edges to $\mathcal{G}^0$, whose weights both equal to ω_2. If $2\omega_1 \geq \omega_2 \geq \omega_1 > 1$, we have the following upper and lower bounds on $t_w(\mathcal{G}^n)$:

$$\begin{aligned}&lb \leq n\log(t_w(\mathcal{G}^n_{\mathbb{R}^n})) + d\log(t_w(\mathcal{G}^n_{SO(n)})) \leq ub,\\ &lb = n\log(t_w(\mathcal{G}^e_{\mathbb{R}^n})) + d\log(t_w(\mathcal{G}^e_{SO(n)})),\\ &ub = (n+d)\log(\omega_2) + (n+d)\log(2) + lb.\end{aligned} \tag{12}$$

Proof: Based on Definition 3.1, there are two kinds of spanning trees $\mathcal{T}$ of the graph $\mathcal{G}^e$: either the new edge e is in $\mathcal{E}(\mathcal{T})$, or not. If $e \notin \mathcal{E}(\mathcal{T})$, $\mathbb{V}(\mathcal{T})$ remains unchanged. Otherwise, we have $\sum_{\mathcal{T}:e\in\mathcal{E}(\mathcal{T})} \mathbb{V}(\mathcal{T}) = \omega_1 t_w(\mathcal{G}^1)$, where $\mathcal{G}^1$ is the the pose graph after adding the same edge, but with unit-weight. Proceeding similarly, it can be shown that:

$$\begin{aligned}lb &= n\log(t_w(\mathcal{G}^0_{\mathbb{R}^n}) + \omega_1 t_w(\mathcal{G}^1_{\mathbb{R}^n}))\\ &+ d\log(t_w(\mathcal{G}^0_{SO(n)}) + \omega_1 t_w(\mathcal{G}^1_{SO(n)}))\\ &\leq n\log(t_w(\mathcal{G}^n_{\mathbb{R}^n})) + d\log(t_w(\mathcal{G}^n_{SO(n)}))\\ &= n\log(2\omega_2 t_w(\mathcal{G}^0_{\mathbb{R}^n}) + \omega_2^2 t_w(\mathcal{G}^1_{\mathbb{R}^n}))\\ &+ d\log(2\omega_2 t_w(\mathcal{G}^0_{SO(n)}) + \omega_2^2 t_w(\mathcal{G}^1_{SO(n)}))\\ &\leq n\log(2\omega_2) + n\log(t_w(\mathcal{G}^0_{\mathbb{R}^n}) + \omega_1 t_w(\mathcal{G}^1_{\mathbb{R}^n})) +\\ &d\log(2\omega_2) + d\log(t_w(\mathcal{G}^0_{SO(n)}) + \omega_1 t_w(\mathcal{G}^1_{SO(n)}))\\ &= (n+d)\log(\omega_2) + (n+d)\log(2) + lb = ub\end{aligned} \tag{13}$$

where $\mathcal{G}^0_{SO(n)}$ (resp. $\mathcal{G}^0_{\mathbb{R}^n}$) is the rotation (resp. translation) graph of $\mathcal{G}^0$, and $\mathcal{G}^1_{SO(n)}$ (resp. $\mathcal{G}^1_{\mathbb{R}^n}$) is the rotation (resp. translation) graph of $\mathcal{G}^1$. ■

Theorem 3.4: Let $\mathcal{G}^m$, $\mathcal{G}^m_{\mathbb{R}^n}$ and $\mathcal{G}^m_{SO(n)}$ be the pose, translation and rotation graphs after adding a chain graph with m new nodes and $m+1$ edges to $\mathcal{G}^0$, whose weights all equal to ω_3. If $(m+1)\omega_1 \geq \omega_3 \geq \omega_1 \geq 1$, we have the following upper and lower bounds on $t_w(\mathcal{G}^m)$:

$$\begin{aligned}&lb_1 \leq n\log(t_w(\mathcal{G}^m_{\mathbb{R}^n})) + d\log(t_w(\mathcal{G}^m_{SO(n)})) \leq ub_1,\\ &lb_1 = n\log(t_w(\mathcal{G}^e_{\mathbb{R}^n})) + d\log(t_w(\mathcal{G}^e_{SO(n)})),\\ &ub_1 = m(n+d)\log(\omega_3) + (n+d)\log(m+1) + lb_1.\end{aligned} \tag{14}$$

Theorem 3.5: Let $\mathcal{G}^{m*}$, $\mathcal{G}^{m*}_{\mathbb{R}^n}$ and $\mathcal{G}^{m*}_{SO(n)}$ be the pose, translation and rotation graphs obtained by adding a general graph with m new nodes and more than $m+1$ edges, whose weights all equal ω_3. If $\mathcal{G}^m$ is a sub-graph of $\mathcal{G}^{m*}$, $t_w(\mathcal{G}^{m*})$ has the following lower bound:

$$\begin{aligned}&n\log(t_w(\mathcal{G}^m_{\mathbb{R}^n})) + d\log(t_w(\mathcal{G}^m_{SO(n)})) \leq\\ &n\log(t_w(\mathcal{G}^{m*}_{\mathbb{R}^n})) + d\log(t_w(\mathcal{G}^{m*}_{SO(n)})).\end{aligned} \tag{15}$$

The proofs are given in the supplementary material [27].

Theorems 3.3, 3.4 and 3.5 imply that solving the 1-ESP problem gives a performance bound on the solutions of the corresponding 1-NESP, m-NESP and multi-NESP problems. This is further discussed in [27]. An important observation is that, in Theorems 3.3 and 3.4, the upper bounds ub and ub_1 are better approximations of the tree-connectivity in 1-NESP and m-NESP than lb and lb_1 in most real-world datasets.

IV. WEAKNESS BIDDING FRAMEWORK

A. Bidding framework and communication

Our weakness bidding framework consists of the following steps. Throughout operation, each robot checks the trace of the covariance of its pose, which is calculated in a computationally efficient manner (Sec. IV-B). If the trace of covariance of the last pose is greater than a threshold, we generate candidate edge sets (i.e. the weak connection points) by solving 1-ESP discussed in Sec. IV-C. The weak connection points, which are located on the selected edges, are communicated to other robots in a multi-hop manner.

Upon receiving the weak connection points, other robots generate potential paths to these targets efficiently with RRT-connect [18] if they are within a set distance. The costs of these potential paths are communicated as a 'quoted price' in the bidding framework. The robot with the lowest quoted price wins the bid, plans a safe trajectory (Sec. IV-D), and takes the target measurement.

For the communication network[2], suppose for example that only one robot publishes its weakness. All robots in the broadcast range will receive the weak connection points and also broadcast this target in its own broadcast range. Being such a multi-hop network, the week connection points will be transferred to all robots in a chained manner.

B. Fast covariance recovery to trigger weak edge selection

To limit the deviation from the robots' original task, active SLAM is activated only when the uncertainty of a robot's position is larger than a set threshold ($\text{trace}(\boldsymbol{C}_{ii}) \geq Threshold_1$).[3] We measure the uncertainty in terms of the trace of the position covariance of i-th pose $\boldsymbol{C}_{ii}$, which is a block diagonal element of the overall covariance matrix $\boldsymbol{C}$. The CRLB implies we may approximate the covariance matrix $\boldsymbol{C}$ by inverting the FIM $\boldsymbol{\mathcal{I}}_{nD}$. Fortunately, we only need the block diagonal elements corresponding to the last pose, which can be calculated efficiently using a recursive formula [28] given the square root information matrix $\boldsymbol{H}$ with $\boldsymbol{\mathcal{I}}_{nD} = \boldsymbol{H}^\top \boldsymbol{H}$. The specific operations, which are similar to [28], are shown in the supplementary material [27].

C. Measurement selection based on submodular optimization, branch and bound method and rank-1 update

When the uncertainty of a robot is large, we solve 1-ESP to find a weak connection in its pose graph which is then sent to the communication network. Note that the k-ESP is a monotone submodular maximization problem subject to a cardinality constraint (Theorem 4, [24]).

As the simplest instance of the k-ESP problem, 1-ESP can be solved by an exhaustive search over all candidate edges in $\mathcal{E}_c$. We limit the lengths of candidate edges, which are considered to be straight line paths for simplicity, to be less than a threshold. This limits the number of communicated additional active SLAM paths and is defined as $Threshold_2 = 2R_s + 3v_r \triangle t$.[3] It constrains the trajectory length, and reduces the approximation loss of converting active SLAM to 1-ESP in Theorem 3.5.

Given a pose graph, the natural way of selecting the candidate set $\mathcal{E}_c$ is to limit the distance between each pair of poses. Specifically, we obtain the candidate edge set $\mathcal{E}_c$ by incrementally testing whether the distances between the last pose and the previous old poses is lower than $Threshold_2$. To do so more efficiently, we use a bounding box method to separate the old poses into several parts, which is similar to the idea of the branch and bound method. We incrementally update the bounding box of the poses, when a bounding box has more than N_b robot poses[4]. This allows us to compare the distance between a new robot pose P_{new} and 8 vertices of the box instead of every old pose.

Based on this exhaustive method, the optimal edge for 1-ESP (11) can be found by exhaustively computing the objective function (11) for every candidate edge $e_c \in \mathcal{E}_c$, leading to a new pose graph with nodes $\mathcal{V}$ and edges $\mathcal{E} \cup \{e_c\}$. For computational efficiency, the 1-ESP (11) is re-written in a matrix form as:

$$\begin{aligned} \max_{e_c \in \mathcal{E}_c} \; & f(e_c) = n \log(\det(\boldsymbol{L}_w^{\mathbb{R}^n} + \boldsymbol{a}_c \sigma_c \boldsymbol{a}_c^\top)) \\ & + d \log(\det(\boldsymbol{L}_w^{SO(n)} + \boldsymbol{a}_c \kappa_c \boldsymbol{a}_c^\top)), \end{aligned} \tag{16}$$

where $\boldsymbol{a}_c$ is a column in the incidence matrix corresponding to new added edge, and σ_c and κ_c are edge wegihts in translation and rotation graphs respectively. $f(e_c)$ can be calculated by a rank-1 update to the reduced weighted Laplacian matrix setting $\bar{\boldsymbol{a}}_c = \boldsymbol{a}_c \sigma_c^{\frac{1}{2}}$, $\hat{\boldsymbol{a}}_c = \boldsymbol{a}_c \kappa_c^{\frac{1}{2}}$, $\boldsymbol{L}_w^{\mathbb{R}^n} = \boldsymbol{H}^{\mathbb{R}^n} \boldsymbol{H}^{\mathbb{R}^n \top}$ and $\boldsymbol{L}_w^{SO(n)} = \boldsymbol{H}^{SO(n)} \boldsymbol{H}^{SO(n)\top}$.[5] Overall, our 1-ESP method is summarized in Algorithm 1.

D. Trajectory optimization

In our bidding framework, each robot generates a 'quoted price' of taking a measurement in terms of trajectory length required. We perform trajectory optimization to not only predict trajectory length, but also to use it for navigation if the bid is successful. In this section, we briefly present the trajectory optimization component. Additional details are provided in the supplementary material [27].

1) Initial Trajectory Generation: We use RRT-connect to generate several initial connected paths from the current pose $\boldsymbol{x}_s$ to the target $\boldsymbol{x}_m$. The shortest one is then used in the bidding step. Fig. 3 shows an example of a shortest path generated by RRT-connect (black line). We select several waypoints $\{\boldsymbol{x}_k^{r,i}\} = \{x_{x,k}^{r,i}, x_{y,k}^{r,i}, x_{z,k}^{r,i}\}, k = 1, \cdots, N_w$ of each robot from the paths to generate a safe corridor (colored regions) for the continuous refinement stage.

2) SVM-based corridor generation: We use 'safe corridors' to represent the free space during the continuous refinement stage. We divide the corridor generation into two stages for horizontal and vertical directions.

[2]A demonstration of this broadcast communication is shown in https://github.com/cyb1212/SM-RAL.git.

[3]$Threshold_1$ and $Threshold_2$ are user-defined constants. We discuss why we set $Threshold_2 = 2R_s + 3v_r \triangle t$ in [27].

[4]N_b is a user-defined constant, which should nominally satisfy $N_b \gg 8$.

[5]The 'SuiteSparse' library [29] provides efficient subroutines for sparse rank-1 Cholesky update including cs_etree and $CholeskyUpdate$ in Algorithm 1.

Algorithm 1: 1-ESP

Input: The candidate edge set $\mathcal{E}_c$ and the edges in base pose graph $\mathcal{E}$
Output: The optimal edge e_c^*

$\boldsymbol{p}_{order} = AMDP(\boldsymbol{L}_w^{\mathbb{R}^n})$; //Column approximate minimum degree permutation
$\boldsymbol{H}^{\mathbb{R}^n} = Cholesky(\boldsymbol{L}_w^{\mathbb{R}^n}(\boldsymbol{p}_{order}, \boldsymbol{p}_{order}))$;
$\boldsymbol{H}^{SO(n)} = Cholesky(\boldsymbol{L}_w^{SO(n)}(\boldsymbol{p}_{order}, \boldsymbol{p}_{order}))$;
$\boldsymbol{v} \leftarrow cs\_etree(\boldsymbol{H}^{\mathbb{R}^n}, 'col')$; //Return the elimination tree of $\boldsymbol{H}^{\mathbb{R}^n\top}\boldsymbol{H}^{\mathbb{R}^n}$ for rank-1 update $CholeskyUpdate$
$f^* \leftarrow 0$
for *all* e_c *in* $\mathcal{E}_c$ **do**
  Generate $\boldsymbol{a}_c$ based on e_c;
  $\bar{\boldsymbol{a}}_c \leftarrow \sigma_c^{\frac{1}{2}}\boldsymbol{a}_c$ and $\hat{\boldsymbol{a}}_c = \kappa_c^{\frac{1}{2}}\boldsymbol{a}_c$;
  $\bar{\boldsymbol{H}}^{\mathbb{R}^n} \leftarrow CholeskyUpdate(\boldsymbol{H}^{\mathbb{R}^n}, \bar{\boldsymbol{a}}_c, \boldsymbol{v})$, $\bar{\boldsymbol{H}}^{SO(n)} \leftarrow CholeskyUpdate(\boldsymbol{H}^{SO(n)}, \bar{\boldsymbol{a}}_c, \boldsymbol{v})$;
  $f \leftarrow d \cdot 2\sum_i \log(\bar{\boldsymbol{H}}^{\mathbb{R}^n})_{i,i} + n \cdot 2\sum_i \log(\bar{\boldsymbol{H}}^{SO(n)})_{i,i}$;
  if $f > f^*$ **then**
    $f^* = f$, $e_c^* = e_c$;
  end
end
return e_c^*

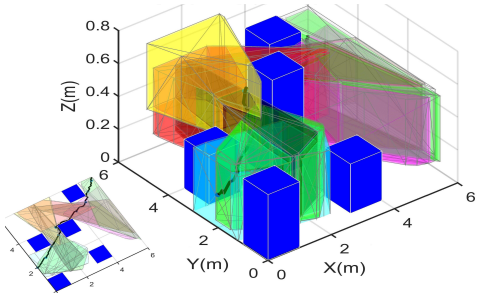


Fig. 3. An example showing safe corridors (every colored corridor is obtained by one selected waypoint) generated by the RRT-connect method (black line) and the modified hard-margin SVM quadratic program.

For the horizontal direction, corridor generation becomes a 2D problem of finding several hyperplanes that divide the waypoints and the features on the obstacles. The fundamental idea is to firstly use the SVM formulation to get the slopes of the hyperplanes. Then, we choose the y-intercept such that the hyperplane intersects the nearest feature point, or a point between the waypoint and the nearest feature point.

For the vertical direction, we first limit the z-axis of the corridor to $[x_{z,k}^{r,i} - h_{lim}, x_{z,k}^{r,i} + h_{lim}]$ in order to minimize change in altitude, and then test for safety. If some parts of the obstacles are located in the corridor, the range will be reduced by a constant value h_r and tested again until the corridor becomes safe. For example, if we use an occupancy grid map, h_r can be set as the grid size.

A large safe corridor $\mathcal{P}$ is generated. An example is shown in Fig. 3. Compared with the method in [31], our novel two stages method can generate a robust and larger corridor.

3) Bezier curve-based continuous refinement: Given the safe corridors generated as per the previous section, it remains to plan a smooth trajectory for the robot. Each $\mu = (x,\ y,\ z)$ component of the robot trajectory $\boldsymbol{x}(t)$ is represented as a Bezier polynomial with N control points as in [32]. We minimize snap $J = \int_{t_0}^{t_N} \|\dddot{\boldsymbol{x}}(t)\|_2^2\, dt$ to ensure a smooth trajectory. Rewriting in a matrix form, the trajectory optimization

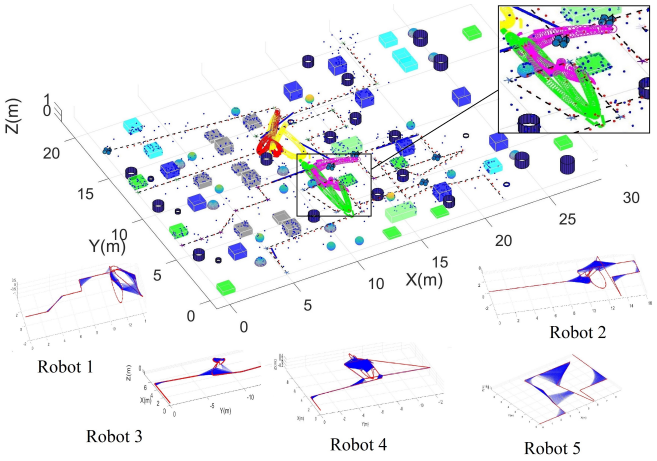


Fig. 4. Active SLAM trajectory (top, colored curves), resulting pose graphs and relative measurements (bottom) in a simulated environment. Blue lines show relative measurements, and the red circles show the estimated robot poses at last step. (an enlarged version of this figure is shown in the supplementary material [27])

problem including safe corridor and start and goal constraints can be solved using quadratic programming, similar to [31]. The problem is convex, and we use the interior point method to solve it.

V. Simulations and Experiments

A. Simulation

In this section, we present a simulation study of the entire cooperative active SLAM framwork in various scenarios. We use MATLAB on a Dell E5570 laptop.

1) Simulations using 5 robots: We consider a 30m $\times$ 20m $\times$ 1m environment with many obstacles (Fig. 4). The original tasks of the robots are to visit several waypoints (blue pentagrams) with velocity $v_r = 0.1$m/s. Each robot is equipped with an omnidirectional sensor with range R_s =1.5m. We set $\Delta t = 1$s, and the noise parameters to $\widetilde{\delta}_{ij}^{-2} = 10^6$, $\widetilde{\kappa}_{ij} = 10^5$, $\delta_{ij}^{-2} = 10^5 N_v$ and $\kappa_{ij} = 500 N_v$, where N_v is the number of common features visible in both P_i and P_j, similar to [18], [33]. The length of additional active SLAM trajectory is limited to be below 8m. To avoid excessive planning iterations, each robot is allowed to respond to other robots' requests twice only. The maximum number of active SLAM processes and maximum additional trajectory length constraint are user-defined parameters.

The trajectories and estimation results with 5 robots using our method is shown in Fig. 4. It can be seen that the 5, 3, 2 and 1-st robots move to get the measurements to repair the weaknesses of the 1, 2, 3 and 4-th robots respectively. These new measurements help the requesting robots to build stronger pose graphs. Results show that, using our method, 5 robots can cooperatively perform SLAM with good accuracy. Meanwhile, only the measurements along the colored additional trajectories are communicated, which leads to low communication cost.

2) Comparison with two existing methods: We compare our method with two previous methods, RE [6] and DAIA [5], and the baseline case of performing the original task only, referred to as the non-active method (Non). For all cases, we

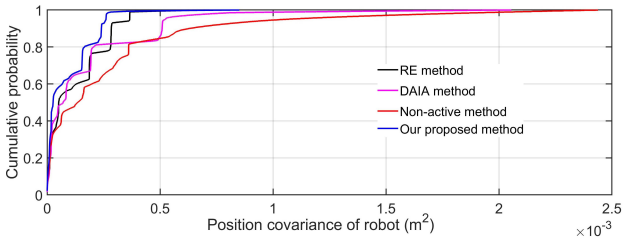


Fig. 5. CDF of position covariance for one trajectory based on 5 simulations. Greater position covariance means worse performance. I.e., better performance corresponds to a CDF approaches 1 in the y-axis as quickly as possible. Blue line (our method) shows good performance.

TABLE I
COMPARISON OF TRAJECTORY LENGTH AND COMMUNICATION DATA

	Communicated data[6]		Trajectory lengths	
Algorithm	Mean (MB)	STD (MB)	Mean (m)	STD (m)
Non	-	-	-	-
Ours	0.855	0.078	35.6	4.46
RE	25.203	0.7312	39.63	1.34
DAIA	12.543	0.5334	40.72	1.31

consider the same task as in Sec. V-A1. Because the compared methods do not consider other tasks or maintaining the original trajectory, we limit the comparison to periods when active SLAM is being performed. Meanwhile, because these methods do not have a triggering mechanism similar to ours, we trigger these methods at randomly selected times. We also modified these methods to use an optimization framework instead of the original filter framework. We use identical parameters for all cases.

Similar to [18], we examine the performance of each method in terms of robot position covariance using a cumulative distribution function (CDF) over 5 simulations to account for randomness, as shown in Fig. 5. Table I shows a statistical summary of the total additional trajectory lengths and the communicated data for all robots.

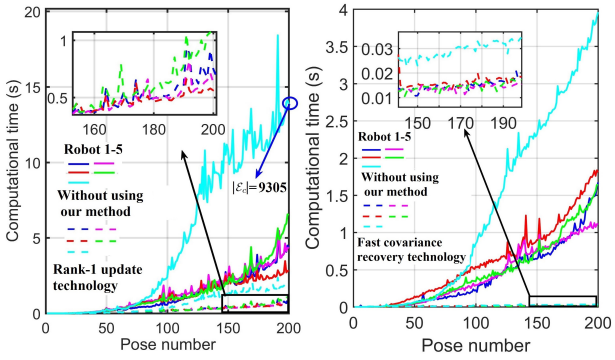
Fig. 5 and Table I show that our method has the best performance in terms of uncertainty reduction and average trajectory length. We also verified that the position covariance reduction is consistent for other tasks. Further, our framework uses communication efficiently because it only communicates weak connection points for bidding, and the measurements in the additional trajectory. In comparison, other methods are more communication intensive because they must communicate the entire previous pose graphs. One shortcoming we observed is that the additional trajectory length can exceed the threshold, because our trajectory optimization method (Sec. IV-D) cannot limit the length of the final generated trajectory. Addressing this limitation is an important future research direction.

Table II shows a comparison of our method and previous methods [6], [5] in terms of planning environment, trajectory type, dimensionality, communication architecture and suitability for other tasks such exploration and search. We find that

[6]Because we assume that the features are already detected and matched by a data association method, the communicated data seems to be small. However, a real system will require transmitting a series of images, leading to a proportional increase in data usage: $25.203O(1) \gg 12.543O(1) \gg 0.855O(1)$ (from first column of Table I). This shows our method can lead to a significant advantage.

TABLE II
COMPARISON OF CHARACTERISTICS

Characteristics	Ours	RE [6]	DAIA [5]
Environment	Obstacle	Obstacle-free	Obstacle-free
Trajectory	Smooth	Smooth	Smooth
Dimension	3D	2D	2D
Suitable for other tasks	Yes	No	No
Communication	Multi-hop Highly distributed	All-to-all	Multi-hop Distributed



(a) (b)

Fig. 6. Comparison of computation time. (a) Rank-1 update (dashed) vs. full log determinant (solid). (b) Fast covariance recovery (dashed) vs. full inversion (solid).

our method is robust, widely applicable and communication-efficient, which are desirable properties for practical robotic systems.

3) Effect of efficient linear-algebraic operations: We examine the improvement in computation time from using the proposed linear-algebraic operations, namely the rank-1 update technique for log determinant computation, and fast covariance recovery. Fig. 6a shows the computation time of optimal edge selection using our rank-1 update method and a naive log-determinant calculation of the full FIM. It can be seen that the rank-1 update method can select the optimal edge with less computational time. Moreover, there is a greater improvement with increasing number of poses. Fig. 6b shows a comparison of the computation time of our fast covariance recovery with full inversion of the FIM. It can be seen that fast covariance recovery incurs less computation cost than full inversion. Again, the speed-up is more significant with increasing number of poses. This allows fast examination of the uncertainty of robot poses.

B. Experiments using two robots

In this section, we present an experimental demonstration in a 4m $\times$ 4m $\times$ 2m environment with two quad-rotor UAV platforms and an online communication system implemented in C++. We run our framework in MATLAB in an offline manner with predictive simulation, on a desktop PC with an Intel(R) Core(TM) i7-4790K CPU @ 4.0GHz. The resulting trajectory is then followed by the two UAVs, which communicate through the desktop computer in real time. The final executed trajectories are shown in Fig. 7, and data usage is shown in Fig. 8. The time required for simulation during the planning process is less than 0.45s (2.2Hz), which indicates

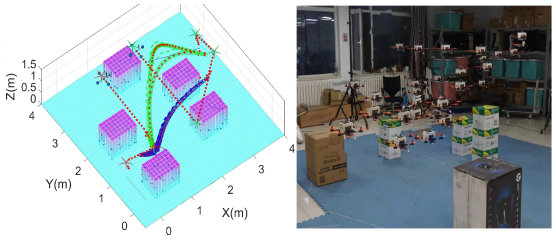


Fig. 7. Offline active SLAM result (The red and blue points respectively mean estimated result and detected features. The planning parts are shown by the green and blue lines.) and following trajectory (composited)

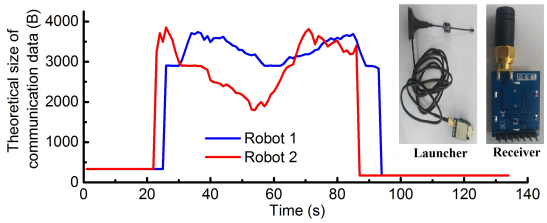


Fig. 8. Theoretical sizes of the received data using launcher and receiver

the possibility of online operation through integration with a SLAM system front-end.

VI. Conclusions and Future Work

This paper presents a cooperative active pose-graph SLAM framework for multiple robots with low communication cost, high computational efficiency and high estimation accuracy in an unknown 3D environment. Using convex quadratic programming, our framework generates smooth trajectories instead of rough and simple tree-shape paths. Simulations and experiments show that our method has good performance in terms of computation and communication efficiency and accuracy. In the future, we would like to conduct real-time experiments to further evaluate our method's practicality, using a combination of our previous work [18] and mature SLAM frameworks [33].

References

[1] M. Dunbabin and L. Marques, "Robots for environmental monitoring: significant advancements and applications," *IEEE Robot. Autom. Mag.*, vol. 19 , no. 1, 24-39, 2012.

[2] P.M. Dames, M. Schwager, D. Rus, and V. Kumar, "Active magnetic anomaly detection using multiple micro aerial vehicles," *IEEE Robot. and Autom. Letters*, vol. 1, no. 1, pp. 153-160, 2015.

[3] S. Saeedi, M. Trentini, M. Seto, and H. Li, "Multiple-robot simultaneous localization and mapping: A review," *J. Field Robot.*, vol. 33, no. 1, pp. 346, 2016.

[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. 13091332, 2016.

[5] N. Atanasov, J. L. Ny, K. Daniilidis, and G. J. Pappas, "Decentralized active information acquisition: theory and application to multi-Robot SLAM," in *Proc. Int. Conf. Robot. Autom.*, pp. 4775-4782, 2015.

[6] M. Kontitsis, E. A. Theodorou, and E. Todorov, "Multi-Robot active SLAM with relative entropy optimization," *American Control Conference*, 2013, pp. 2757-2764.

[7] M. Giamou, K. Khosoussi, and J. P. How, "Talk resource-Efficiently to me: optimal communication planning for distributed loop closure detection," in *Proc. Int. Conf. Robot. Autom.*, 2018, pp. 3841-3848.

[8] T. Regev and V. Indelman, "Multi-Robot decentralized belief space planning in unknown environments via efficient re-evaluation of impacted paths," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 5591-5598.

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

[10] Y. Kantaros, B. Schlotfeldt, N. Atanasov, and G.J. Pappas, "Asymptotically Optimal Planning for Non-myopic Multi-Robot Information Gathering," in *Robotics: Science and Systems*, 2019.

[11] A. Viseras, and R. Garcia, "DeepIG: Multi-Robot Information Gathering with Deep Reinforcement Learning," *IEEE Robot. and Autom. Letters*, vol. 4, no. 3, pp.3059-3066, 2019.

[12] G. Best, J. Faigl, and R. Fitch, "Multi-robot path planning for budgeted active perception with self-organising maps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 3164-3171, 2016.

[13] V. Indelman, "Cooperative multi-robot belief space planning for autonomous navigation in unknown environments," *Auton. Robots*, vol. 42, no. 1, pp. 353373, 2018.

[14] A. Kitanov and V. Indelman, "Topological Multi-Robot Belief Space Planning in Unknown Environments", in *Proc. Int. Conf. Robot. Autom.*, 2019, pp.5726-5732.

[15] M. Lauri, E. Heinnen, and S. Frintrop, "Multi-robot active information gathering with periodic communication," in *Proc. Int. Conf. Robot. Autom.*, 2017, pp. 851-856.

[16] R. Khodayi-mehr, Y. Kantaros, and M.M. Zavlanos, "Distributed state estimation using intermittently connected robot networks," *IEEE Trans. on Robot.*, vol. 35, no. 3, pp. 709- 724, 2019

[17] B. Schlotfeldt, D. Thakur, N. Atanasov, and G.J. Pappas, "Anytime planning for decentralized multirobot active information gathering," *IEEE Robot. and Autom. Let.*, vol. 3, no. 2, pp.1025-1032, 2018.

[18] Y. Chen, S. Huang, R. Fitch, L. Zhao, and D. Yang, "On-line 3D active pose-graph SLAM based on key poses using graph topology and submaps," in *Proc. Int. Conf. Robot. Autom.*, pp. 169-175, 2019.

[19] 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, pp. 95-125, 2019.

[20] 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.216235, 2012.

[21] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Proc. Int. Conf. Robot. Autom.*, pp. 3607-3613, 2011.

[22] N. Boumal, A. Singer, P.-A. Absil, and V.D. Blondel, "Cramer-Rao bounds for synchronization of rotations," *Information and Inference: A Journal of the IMA*, vol. 3, no. 1, pp.1-39, 2014.

[23] S. Huang, N. M. Kwok, G. Dissanayake, and G. Fang, "Multi-step look-ahead trajectory planning in SLAM: Possibility and necessity," in *Proc. Int. Conf. Robot. Autom.*, pp. 1091-1096, 2005.

[24] K. Khosoussi, M. Giamou, G.S. Sukhatme, S. Huang, G. Dissanayake, and J.P. How, "Reliable graph topologies for SLAM," *Int. J. Robot. Res.*, vol. 38, no. 2-3, pp. 260-298, 2019.

[25] M. Mesbahi, and M. Egerstedt, "Graph theoretic methods in multiagent networks," vol. 33. Princeton University Press, 2010.

[26] Y. Chen, S. Huang, L. Zhao, and G. Dissanayake, "Cramér-Rao bounds and optimal design metrics for pose-graph SLAM," 2018. https://github.com/cyb1212/A-submittedpaper

[27] Y. Chen, L. Zhao, S. Huang, R. Fitch, K.M.B. Lee, and C. Yoo, "Broadcast your weaknesses: cooperative active pose-graph SLAM for multiple robots," (Supplementary Material), 2019. https://github.com/cyb1212/SM-RAL

[28] V. Ila, L. Polok, M. Solony, P. Smrz, and P. Zemcik, "Fast covariance recovery in incremental nonlinear least square solvers," in *Proc. Int. Conf. Robot. Autom.*, pp. 4636-4643, 2015.

[29] A. Davis, SUITESPARSE: A suite of sparse matrix software, available at http://faculty.cse.tamu.edu/davis/suitesparse.html.

[30] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C.J. Taylor, and V. Kumar, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3D complex environments," *IEEE Robot. and Autom. Let.*, vol. 2, no. 3, pp. 1688-1695, 2017.

[31] W. Honig, J. A. Preiss, T. S. Kumar, G. S. Sukhatme, and N. Ayanian, "Trajectory planning for quadrotor swarms," *IEEE Trans. on Robot.*, vol. 34, no. 4, 856-869, 2018.

[32] K. I. Joy, "Bernstein polynomials," On-Line Geometric Modeling Notes, vol. 13, 2000.

[33] R. Mur-Artal and J.D. Tards, "Orb-slam2: An open-source slam system for monocular, stereo, and RGB-D cameras," *IEEE Trans. on Robot.*, vol. 33, no. 5, pp. 1255-1262, 2017.