Swarm-SLAM: Sparse Decentralized Collaborative Simultaneous Localization and Mapping Framework for Multi-Robot Systems

Pierre-Yves Lajoie, Giovanni Beltrame

***Abstract*—Collaborative Simultaneous Localization And Mapping (C-SLAM) is a vital component for successful multi-robot operations in environments without an external positioning system, such as indoors, underground or underwater. In this paper, we introduce Swarm-SLAM, an open-source C-SLAM system that is designed to be scalable, flexible, decentralized, and sparse, which are all key properties in swarm robotics. Our system supports inertial, lidar, stereo, and RGB-D sensing, and it includes a novel inter-robot loop closure prioritization technique that reduces communication and accelerates convergence. We evaluated our ROS-2 implementation on five different datasets, and in a real-world experiment with three robots communicating through an ad-hoc network. Our code is publicly available: https://github.com/MISTLab/Swarm-SLAM**

***Index Terms*—SLAM, Multi-Robot Systems, Collaborative Perception, Swarm Intelligence**

I. Introduction

COLLABORATIVE perception is an important problem for the future of robotics. The shared understanding of the environment it provides is a prerequisite to many applications from autonmous warehouse management to subterranean exploration. One of the most powerful tools for robotic perception is Simultaneous Localization And Mapping (SLAM) which tightly couples the geometric perception of the environment with state estimation [1]. In addition to producing high-quality maps of the robot surroundings, it provides localization estimates that are essential for planning and control. However, single-robot SLAM estimates are local in the individual robot reference frame. Therefore, when multiple robots operate in GPS-denied environments, they do not share situational awareness unless they manage to connect, or merge, their local maps. To solve this problem, Collaborative SLAM (C-SLAM) searches for inter-robot map links and uses them to combine the local maps into a shared global understanding of the environment.

One of the main practical challenges in C-SLAM is resource management [2], in particular considering the severe communication and computation limitations of mobile robots. Those limitations need to be addressed to achieve real-time performance, especially when a large number of robots work together. While very effective in some scenarios, centralized

This work was partially supported by a Vanier Canada Graduate Scholarships Award and by the Canadian Space Agency.

Department of Computer and Software Engineering, Polytechnique Montréal, Montreal, Canada, {pierre-yves.lajoie, giovanni.beltrame}@polymtl.ca

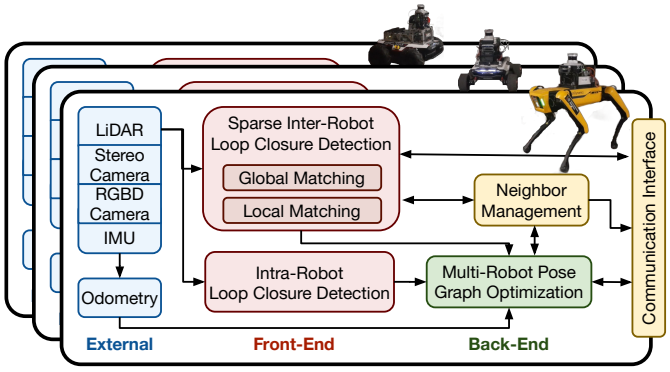


Fig. 1: Swarm-SLAM Overview

Issue w/ traditional C-SLAM

C-SLAM solutions, which rely on a single server for computation, suffer from a communication bottleneck between the robots and the server, which limits their scalability. Besides, due to networking coverage challenges in large indoor, subterranean or underwater environments, robots cannot realistically maintain a stable connection to a central server. Thus, decentralized solution relying only on sporadic and intermittent communication between the robots are better suited for large-scale deployment.

While collaborative perception within small teams of autonomous robots is currently challenging, we believe it useful to look forward to very large teams, or swarms of robots and start tackling the problems specific to this scale of deployment. Prior works on swarm robotics have identified a few key properties required for swarm compatibility [3] such as: communication and sensing must be local to the robot neighborhood, and robots should not rely on a centralized authority or global knowledge. In the specific case of C-SLAM [4], additional properties such as flexibility and scalability must be considered.

What the paper proposes!

In this paper, we propose novel techniques assembled in a complete resource-efficient C-SLAM framework compliant with these key swarm compatibility properties. Our approach is fully decentralized, supports different types of sensors (stereo cameras, RGB-D cameras, and lidars), relies only on sporadic connectivity, and requires significantly less communication than previous techniques. To reduce the required data exchanges, we introduce a novel budgeted approach to select candidate inter-robot loop closures based on algebraic connectivity maximization, adapted from recent work on pose graph sparsification [5]. This preprocessing of place recogni-

tion matches allows us to achieve accurate C-SLAM estimates faster and using fewer communication resources. Moreover, we leverage ROS 2 [6] and recent advances in robotic software engineering, to make our framework compatible with ad-hoc networks.

In summary, this paper offers the following **contributions**:

- A sparse budgeted inter-robot loop closure detection algorithm under communication constraints based on algebraic connectivity maximization;
- A decentralized approach to neighbor management and pose graph optimization suited for sporadic inter-robot communication;
- A swarm-compatible open-source framework based on ROS 2 which supports lidars, as well as stereo or RGB-D cameras;
- An extensive evaluation of the overall system performance on datasets and in a real-world experiment.

The rest of this paper is organized as follows: Section II presents some context and related work; Section III provides an overview of the system; Section IV presents the front-end mechanisms including our novel inter-robot loop closure prioritization; Section V details the estimation process in the back-end; Section VI presents experimental results; and Section VII offers conclusions and suggests future works.

II. Background and Related Work

A. Collaborative SLAM

C-SLAM systems can usually be divided into two categories: centralized and decentralized. Centralized systems rely on a remote base station to compute the global SLAM estimates for all the robots. The base station can be a server cluster with computing capabilities unattainable on mobile platforms, and transmit the optimized localization and map estimates back to the robots. However, in those systems, the robots need a reliable permanent connection with the base station, and the scalability is severely limited by the communication bottleneck to the central server [7]. Such stringent networking constraints are often unrealistic, especially in large environments. Decentralized approaches, relying only on sporadic communication links between robots and without any need for a central authority, are preferred in those scenarios. However, decentralized systems are limited by the onboard computation and communication capabilities of the robots, and they require more sophisticated data management and bookkeeping strategies to obtain accurate SLAM estimates [2].

Similar to single robot SLAM systems, C-SLAM contains two parts commonly named front-end and back-end. The front-end is in charge of feature extraction and data association, while the back-end performs state and map estimation [1].

1) Front-End: One of the most important step in C-SLAM, is the detection and computation of inter-robot loop closures. Inter-robot loop closures correspond to common features or places previously visited by two or more robots. Those shared features between the individual robots maps act as stitching points to merge the local maps together and obtain a shared (global) reference frame. Similar to their intra-robot versions, inter-robot loop closures are geometric constraints between two pose graph vertices. By connecting vertices from different robots pose graphs, we can merge them and compute a global estimate for all robots.

In this work, we focus on indirect loop closures in the sense that we want to find connections anywhere between individual robot maps, rather than direct observations of a robot by another during an encounter [8]. Indirect loop closures have the advantage of connecting maps in multiple points to further reduce the estimation error, and do not require additional sensing (e.g., UWB ranging sensor [9]). The crux of the challenge is to find those indirect inter-robot links among large robot maps in a resource-efficient manner.

Since the communication cost of sharing entire maps is usually prohibitive, indirect inter-robot loop closure detection can be performed in two stages [10], [11]. In the first stage, compact global descriptors of images [12] or lidar scans [13], are shared between the robots for place recognition. Similarity scores are computed between the global descriptors from both robots to recognize places, or overlaps, between their respective maps. Thus, recognized places correspond to candidate loop closures for the second stage. Place recognition is followed by geometric verification to compute the loop closures 3D constraints. For each candidate with high global descriptors similarity, the corresponding local descriptors such as 3D keypoints or scans are transmitted to compute the geometric registration between the two robots frames. The resulting measurement linking a pose from both robots is then added to their pose graph. Matching compact descriptors first is advantageous given that local descriptors are usually larger and thus more costly to communicate and compare [14].

2) Back-End: The role of the C-SLAM back-end is to estimate the most likely poses and map from the noisy measurements gathered by all robots. To this end, Choudary et al. [15] propose the distributed Gauss-Seidel (DGS) technique which allows robots to converge to a globally consistent local pose graph by communicating only the pose estimates involved in inter-robot loop closures, and therefore preserving the privacy of their whole trajectories. Tian et al. [16] significantly improve on that approach and provide a certifiably correct distributed solver for pose graph optimization. This technique performs multiple exchanges between the robots until they converge to globally consistent local solutions. Tian et al. [17] extend this approach to support asynchronous updates during the iterative optimization process, thus avoiding complex synchronization and bookkeeping. In a different vein, recent work by Murai et al. [18] laid the foundation for larger-scale multi-robot collaborative localization based on Gaussian Belief Propagation.

One of the main challenges in both single-robot and collaborative SLAM is the frequent occurrence of erroneous measurements among inter-robot loop closures due to perceptual aliasing [19]. While many techniques exist for the single-robot problem, Lajoie et al. [11] first combined DGS with Pairwise Consistency Maximization (PCM) [20], which computes the maximal clique of pairwise consistent inter-robot measurements, to perform robust and distributed pose graph optimization. More recently, Yang et al. [21] introduced the Graduated Non-Convexity (GNC) algorithm, a general-purpose approach

TABLE I: Open-Source C-SLAM Frameworks

	Sensor	Decentralized	Robust	Sporadic	Sparse
DSLAM [10]	s	✓			
DOOR-SLAM [11]	s,l	✓	✓	✓	
COVINS [23]	s		✓		✓
Kimera-Multi [22]	s	✓	✓	✓	
Disco-SLAM [24]	l	✓	✓	✓	
LAMP 2.0 [25]	l		✓		✓
maplab 2.0 [26]	s,d,l		✓		
Swarm-SLAM	s,d,l	✓	✓	✓	✓

for robust estimation on various problems including pose graph optimization. GNC was integrated with [16] in a robust distributed C-SLAM solver (D-GNC) [22].

3) Open-Source C-SLAM Systems: Many open-source C-SLAM systems have been proposed in the recent years. Ckieslewski et al. [10] introduce DSLAM, which uses CNN-based global descriptors for distributed place recognition, and DGS for estimation. DOOR-SLAM [11] robustified the approach by integrating PCM for outlier rejection and adapted it for sporadic inter-robot communication. DiSCo-SLAM [24] extends those ideas to lidar-based C-SLAM using ScanContext global descriptors [13]. Kimera-Multi [22] integrates D-GNC and incorporates semantic data in the resulting maps.

In an other line of work, centralized C-SLAM system have also evolved considerably. COVINS [23] presents an efficient solution tailored for visual-inertial sensing, which is sparsified by removing redundant keyframes. The lidar-based system LAMP 2.0 [25] introduces a centralized Graph Neural Network-based prioritization mechanism to predict the outcome of pose graph optimization for each inter-robot loop closure candidates. The multi-modal maplab 2.0 [26] supports heterogeneous groups of robots with different sensor setups.

In contrast, Swarm-SLAM combines the latest advances from previous frameworks and introduce a new sparse inter-robot loop closure prioritization to further reduce communication. Additionally, unlike previous techniques, Swarm-SLAM leverages recent advances with ROS 2 and robotic software to provide a seamless C-SLAM integration with ad-hoc, sporadic, networking.

Remark. *ROS 2 offers many possibilities for C-SLAM that were not possible using ROS 1. Most importantly, there is no more centralized master node in ROS 2. This allows us to build truly decentralized systems with dynamic network discovery.*

Table I offers a comparison of the various systems based on key desirable properties, for the supported sensors we use s for stereo cameras, l for lidar and d for RGB-D cameras. We refer the reader to [2] for a thorough survey on C-SLAM.

B. Graph Sparsification for C-SLAM

The ever-growing map and pose graph during long-term operations is an important memory and computation efficiency challenge in both single-robot and multi-robot SLAM [27]. One favored solution is graph sparsification, which aims to approximate the complete graph with as few edges as possible, mainly by removing redundant edges that are not providing new information during the estimation process. To this end, Doherty et al. [5] formulate the graph sparsification of single robot pose graphs as a *maximum algebraic connectivity augmentation* problem, and solve it efficiently using a more tractable convex relaxation.

In this paper, instead of sparsifying the pose graph after all the measurements have already been computed, we aim to preemptively sparsify the inter-robot loop closure candidates generated by the place recognition module. This way, we can prioritize the geometric verification of inter-robot loop closures that will approximate the full pose graph, thus avoiding wasting resources on redundant measurements. Importantly, unlike other work maximizing the determinant of the information matrix [28], we leverage the results from [5] and focus on the algebraic connectivity of the pose graph which has been shown to be a key measure of estimation accuracy [29], [30] (i.e., higher algebraic connectivity is associated with lower estimation error). Solving a similar problem, Denniston et al. [31] prioritize loop closure candidates based on point cloud characteristics, the proximity of known beacons, and the information gain predicted with a graph neural network. Interestingly, Tian et al. [32] explore spectral sparsification in the C-SLAM back-end to reduce the required communication during distributed pose graph optimization.

III. System Overview

As described in Fig. 1, Swarm-SLAM is composed of three modules. First, to enable decentralization, the neighbor management module continuously tracks which robots are in communication range (i.e., neighbors that can be reached reliably). Robots publish heartbeat messages at a fixed rate such that network connectivity can be evaluated periodically. To make the system scalable (see Property 1), the other modules query the neighbor management module to determine which robots, if any, are available for every operation.

Property 1. ***Scalable****. The number of robots using the framework is not predetermined and it does not require connectivity maintenance during the whole mission. This makes the system suited for random rendezvous between robots and tolerant to disconnections. Communication and computation budgets are set to fit the available bandwidth and computation power onboard the individual robots.*

The front-end takes as input an odometry estimates (obtained using an arbitrary technique) along with synchronized sensor data (see Property 2). Upon reception, the front-end extracts global (e.g. compact learned representation) and local descriptors (e.g. 3D keypoints). Global descriptors allow us to identify candidate place recognition matches (i.e., loop closures) between the robots, then local descriptors are used for 3D registration.

Property 2. ***Flexible****. The framework supports multiple sensors (i.e., stereo cameras, RGB-D cameras, lidars) and is decoupled from the odometry source.*

This indirect inter-robot loop closure detection process, leverages inter-robot communication to search for loop closures anywhere in the robot maps, instead of limiting itself to

direct observations. It is enabled whenever one or more other robots are within communication range (see Assumption 1).

Assumption 1. *The robots can transmit data to their neighbors during rendezvous. However, rendezvous and network connectivity can be sporadic during the mission.*

In the back-end, the resulting intra-robot and inter-robot loop closure measurements are combined with the odometry measurements into a pose graph. Local pose graphs are transmitted to the robot selected to perform the optimization and the resulting estimates are sent back to the respective robots. This process is decentralized (see Property 3) since the robot performing the optimization is the result of a simple negociation between the neighbors during a rendezvous.

Property 3. ***Decentralized****. All computation is performed onboard the robots without any central authority and they rely only on peer-to-peer communication.*

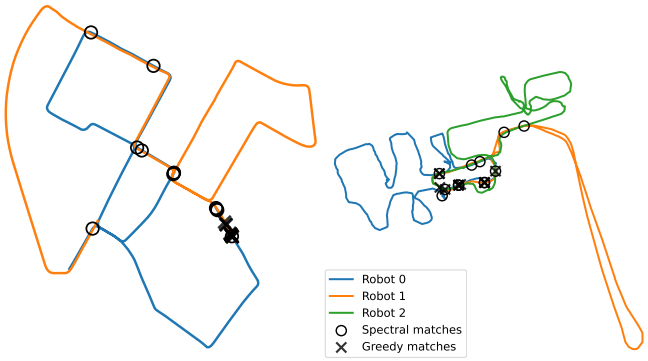
Current pose estimates, resulting from the whole process, are made available periodically in the form of ROS 2 messages for a minimally invasive integration into existing robotic systems. Mapping data for planning or visualization can also be queried at the cost of additional computation and communication. For debugging purposes, we provide a minimal visualization tool which opportunistically collects mapping data from robots in communication range. Overall, we divided place recognition, geometric verification and pose graph optimization into modular and decoupled processes (ROS 2 nodes) with clear data interfaces to enable researchers to leverage Swarm-SLAM to easily test new ideas in each subsystems.

IV. Front-End

The main task of the Swarm-SLAM front-end is indirect inter-robot loop closure detection, which is one of the most challenging problem in C-SLAM. Similar to many comparable techniques (e.g. [10], [11], [22]), we adopt a two stage approach in which global matching generate candidate place recognition matches that are verified using local features in the latter stage. This approach allows us to reduce communication and computation since global matching is done using compact descriptors that are inexpensive to transmit and compare. Expensive local matching is performed only on the most promising candidate matches. To further increase the front-end resource-efficiency, we introduce a novel sparsification algorithm to select the global matching candidates that would maximize the pose graph algebraic connectivity, and in turn the decrease the estimation error. The proposed front-end approach leads to better accuracy using less communication and computation (see Fig. 4).

A. Global Matching

For each keyframe, compact descriptors, that can be compared with a similarity score, are extracted from sensor data and broadcast to neighboring robots. When two robots meet, we perform simple bookkeeping to determine which global descriptors are already known by the other robot and which ones need to be transmitted. We use ScanContext [13] as global descriptors of lidar scans and the recent CNN-based CosPlace [33] for images. We use nearest neighbors based on cosine similarity for descriptor matching. We also support NetVLAD [12] and our system can be easily extended to other compact descriptor extraction algorithms that satisfy Assumption 2.



(a) 2 robots KITTI-00 (b) 3 robots S3E Laboratory

Fig. 2: Visualization of 10 first loop closure candidates selected by the Spectral and Greedy Approaches. We can see that the Spectral approach selects candidates in different regions of the pose graph to better increase the estimation accuracy, while the Greedy approach tends to select redundant candidates in high similarity regions.

Assumption 2. *Place recognition is performed using compact global descriptors, and geometric verification is computed using larger collections of local features.*

Once matches are computed, Swarm-SLAM offers two candidate prioritization mechanisms: a trivial greedy prioritization algorithm, used in prior work [10], [11], [22], [24], and a novel spectral approach. To perfom the candidate prioritization, we define the multi-robot pose graph as:

$$\mathcal{G} = (V, \mathcal{E}^{\text{local}}, \mathcal{E}^{\text{global}}) \tag{1}$$

$$V = (V_1, \dots, V_n) \tag{2}$$

$$\mathcal{E}^{\text{local}} = (\mathcal{E}_1^{\text{local}}, \dots, \mathcal{E}_n^{\text{local}}) \tag{3}$$

$$\mathcal{E}^{\text{global}} = (\mathcal{E}_{\text{fixed}}^{\text{global}}, \mathcal{E}_{\text{candidate}}^{\text{global}}) \tag{4}$$

where V are the vertices from every n robots pose graphs, each vertex corresponding to a keyframe; $\mathcal{E}^{\text{local}}$ are the local pose graphs edges such as odometry measurements and intra-robot loop closures; and $\mathcal{E}^{\text{global}}$ are the global pose graph edges corresponding to inter-robot loop closures. $\mathcal{E}^{\text{global}}$ is further divided between $\mathcal{E}_{\text{fixed}}^{\text{global}}$ which contains the fixed measurements that have already been computed, and the candidate inter-robot loop closures $\mathcal{E}_{\text{candidate}}^{\text{global}}$ on which the prioritization is performed. Detailed measurements (i.e., pose estimates) are not required for our proposed candidate prioritization mechanism. Therefore, fixed measurements, both local and global, are undirected unweighted edges between two vertices, and candidates edges contain an additional weight value corresponding to their respective similarity score. This reduced multi-robot pose graph can be built directly from the global matching information and does not require any additional inter-robot communication.

The number of edges to select B at each time step is set by the user. This budget should reflect the communication and computation capacities of the robots. The common candidate prioritization approach widely used prior works is a basic *greedy prioritization* in which the top B candidates with the highest similarity scores are selected.

In our proposed *spectral prioritization* process, we frame pose graph sparsification as a candidate prioritization problem, and leverage recent work on spectral sparsification [5]. Unlike Doherty et al. [5], which remove existing, redundant, measurements from an existing single-robot pose graph while retaining estimation accuracy, we perform sparsification on the candidate inter-robot matches before computing the corresponding 3D measurements, reducing resource usage for the costly inter-robot geometric verification of redundant candidates, and achieving better accuracy (see Property 4).

Property 4. ***Sparse***. *The framework prioritizes communication using algebraic connectivity maximization to acheive better localization accuracy with fewer data exchanges. At every stage, it requires less communication than comparable techniques.*

As shown in [30], the algebraic connectivity of the pose graph controls the worst-case error of the solutions of the *Maximum Likelihood Estimation* problem (see Problem 2). The pose graph algebraic connectivity corresponds to the second-smallest eigenvalue λ_2 of the *rotational weighted Laplacian* with entries for pairs of vertices (i, j) defined as:

$$L_{ij} = \begin{cases} \sum_{e \in \delta(i)} \kappa_e, & i = j, \\ -\kappa_{ij}, & \{i, j\} \in \mathcal{E}, \\ 0, & \{i, j\} \notin \mathcal{E}. \end{cases} \tag{5}$$

where κ_{ij} denotes the edge weight and $\delta(i)$ is the set of edges incident to vertiex i. In our proposed approach, we use $\kappa_e = 1 \; \forall \; e \in (\mathcal{E}^{\text{local}}, \mathcal{E}^{\text{global}}_{\text{fixed}})$, and $\kappa_e = s \; \forall \; e \in \mathcal{E}^{\text{global}}_{\text{candidate}}$ where $s \in [0, 1]$ is the matching similarity score. This approach forgoes the need to communicate additional information regarding the existing edges' confidence.

For our purposes, we leverage the property that the Laplacian L can be expressed as the sum of subgraph Laplacians corresponding to each of its edges to define the augmented pose graph Laplacian as follows:

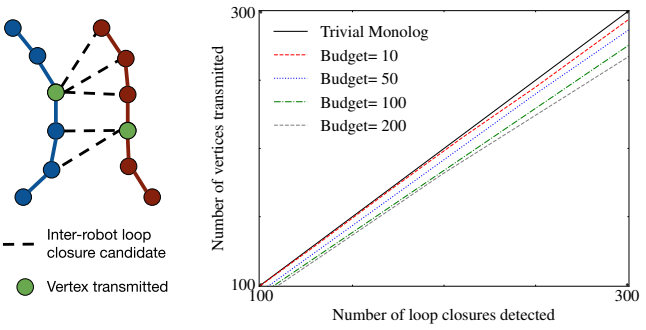
$$L(\omega) \triangleq L^{\mathcal{E}^{\text{local}}} + L^{\mathcal{E}^{\text{global}}_{\text{fixed}}} + \sum_{e \in \mathcal{E}^{\text{global}}_{\text{candidate}}} \omega_e L_e \tag{6}$$

where $\omega_e \in \{0, 1\}$ is the binary variable which determines the prioritization of candidate edge e.

Therefores, according to our previously stated goal, we aim to select the subset $\mathcal{E}^\star \subseteq \mathcal{E}^{\text{global}}_{\text{candidate}}$ of fixed budgeted size $|\mathcal{E}^\star| = B$ which maximizes the algebraic connectivity $\lambda_2(L(\omega))$:

Problem 1. *Candidate prioritization via Algebraic Connectivity Maximization*

$$\begin{aligned} &\max_{\omega_e \in \{0,1\}} \lambda_2(L(\omega)) \\ &|\omega| = B. \end{aligned} \tag{7}$$



(a) Vertex cover problem illustration

(b) Vertex cover with different budgets versus trivial monolog. We can see higher gains with higher candidate budget.

Fig. 3: Vertex cover problem for the minimization of local features communication during geometric verification. The results were obtained on the 2 robots KITTI-00 dataset.

Problem 1 is NP-Hard [34] due to the integrality constraint on ω_e. Therefore, as in [5], we relax the integrality constraints and, when necessary, we round the optimization result to the nearest solution in the feasible set of Problem 1. We solve the relaxed problem using the simple and computationally inexpensive approach developed in [5]. It is important to note that this approach requires the pose graph to be connected, so we first perform greedy prioritization up until at least one inter-robot loop closure exists between the local pose graphs. We also use the greedy solution as initial guess for the algebraic connectivity maximization.

In Fig. 2, we present a visualization of the approach results in comparison of the ones obtained with the standard greedy approach. We can see that the candidates selected using our spectral technique are more evenly distributed along the pose graph while the greedy candidates are mostly concentrated in high-similarity areas. Our selected candidates are therefore less redundant and more informative to the estimation process.

B. Local Matching

Once the inter-robot loop closure candidates are selected, the next step is to perform local matching (i.e., geometric verification). This step leverages larger collections of local features, such as ORB [35] or point clouds, to compute the 3D relative pose measurement between the candidate's two vertices. We use the RTAB-Map library [36] for visual feature extraction and geometric verification, and TEASER++ [37] for robust pointcloud registration. To avoid computing the same loop closure twice, the standard approach consists of a trivial monolog [38] in which only one of the two vertices is transmitted per candidate, and the receiving robot is in charge of generating the relative pose measurement. Once the measurement is obtained, it is shared back to the other robot.

However, to further reduce the communication burden of geometric verification, we follow [38] and formulate the vertices local features sharing problem as a vertex cover problem. As illustrated in Fig. 3a, when two or more inter-

robot loop closure candidates share a vertex in common, only the common vertex needs to be transmitted to effectively compute all the associated relative pose measurements. Thus, by computing the minimal vertex cover, optimally for bipartite graphs and approximately with 3 robots or more, we obtain an exchange policy which avoids redundant communication. The benefits of vertex cover minimization are shown in Fig. 3b. As expected, the larger the candidate budget is (i.e., number of loop closures candidates to select in global matching), the larger the communication savings are, since more vertices are likely to be part of multiple loop closure candidates.

C. Inter-Robot Communication

It is worth noting, that both for the spectral matching and the vertex exchange policy, a *broker* needs to be dynamically elected among the robots in communication range. The broker then computes the matches and sends requests for the vertices to be transferred. In our current implementation, the broker is simply the robot with the lowest ID, but it could be elected through a more informed decentralized negotiation (e.g. based on the available computation resources onboard each robots).

As mentioned above, the user of Swarm-SLAM can control the communication bandwidth by setting a candidate budget. They can also specify the rate at which the computation loops run in the different processes, allowing for example the user to dictate the number of candidates to select per second.

V. Back-End

The role of the back-end is to gather the odometry, intra- and inter-robot loop closure measurements from the front-end in a pose graph, and then estimate the most likely map and poses based on those noisy measurements. The *Maximum Likelihood Estimation* (MLE) problem we solve is:

Problem 2 (Pose Graph Optimization MLE on $\mathrm{SE}(3)$)**.**

$$\min_{\substack{\boldsymbol{t}_i \in \mathbb{R}^3 \\ \boldsymbol{R}_i \in \mathrm{SO}(3)}} \kappa_{ij} \|\boldsymbol{R}_j - \boldsymbol{R}_i \bar{\boldsymbol{R}}_{ij}\|_F^2 + \sum_{(i,j)\in\mathcal{E}} \tau_{ij} \|\boldsymbol{t}_j - \boldsymbol{t}_i - \boldsymbol{R}_i \bar{\boldsymbol{t}}_{ij}\|_2^2. \tag{8}$$

where $(\boldsymbol{R}_i, \boldsymbol{t}_i)$ are the rotation and translation estimate of pose i; $(\bar{\boldsymbol{R}}_{ij}, \bar{\boldsymbol{t}}_{ij})$ are the relative measurements between two poses i and j; κ_{ij} and τ_{ij} are rotation and translation information parameters related to each measurement confidence. The edge set for the multi-robot pose graph estimation includes all local (intra-robot) and global (inter-robot) measurements: $\mathcal{E} = (\mathcal{E}^{\text{local}}, \mathcal{E}^{\text{global}})$.

As mentioned above, unlike other recent systems [11], [28] based on distributed pose graph optimization, we opt for a simpler decentralized approach. Similar to the front-end, a robot is dynamically elected to perform the computation among the robots in communication range. The other robots share their current pose graph estimates with the elected robot and receive the updated estimates once the computation is completed. Importantly, any robot can be elected through negotiation to perform the pose graph optimization during a sporadic encounter between the robots. Swarm-SLAM performs the pose graph optimization using the Graduated Non-Convexity [21] solver, with the robust Truncated Least Square loss, available in the GTSAM library [39].

To ensure convergence to a single global localization estimate after multiple sporadic rendezvous without enforcing a central authority, we introduce an anchor selection process to keep track of the current global reference frame. During pose graph optimization, the anchor usually corresponds to a prior which assigns a fixed value to the first pose of the graph. This anchor then becomes the reference frame of the resulting estimate. In the beginning, all robots are within their own local reference frames where the origin corresponds to their first pose (i.e., initial position and orientation). Then, when some robots meet for the first time (e.g. robots 0, 4 and 5), we choose the first pose of the robot with the lowest ID (e.g. robot 0) as the anchor. Therefore, as a result of the estimation process, the involved robots estimates share the same reference frame (e.g. robot 0's first pose). In subsequent rendezvous (e.g. robots 2, 3 and 4), the anchor is selected based on the reference frame with the lowest ID (e.g. robot 4's first pose is selected as the anchor since its reference frame is robot 0's). After a few rendezvous, the robots converge to a single global reference frame without requiring rendezvous including all robots (e.g. after the second rendezvous, robots 2 and 3 are also within robot 0's reference frame). This means that Swarm-SLAM can scale to large groups of robots, through iterative estimation among smaller groups of robots.

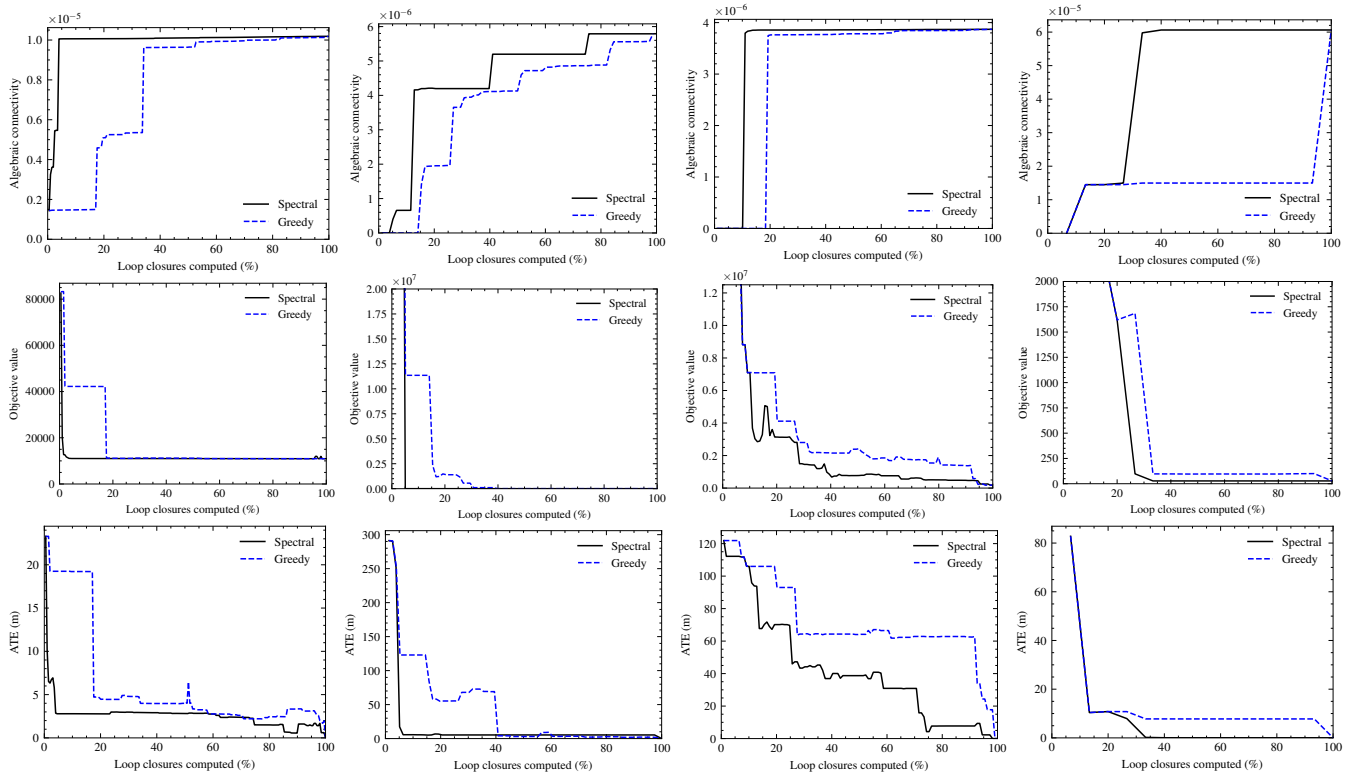
We motivate our preference for a decentralized approach, in terms of communication and computation loads, with an evaluation and comparison against state-of-the-art distributed approaches presented in Table III and discussed in Section VI-A. In a nutshell, performing the optimization on a single robot leads to more accurate estimates with less communication and computation.

TABLE II: Datasets Information

	# Robots	Sensors	Total Length (m)	Size (GB)
KITTI 00	2	stereo	3835	26.4
KITTI-360 09	5	lidar	10714	26.9
GrAco Ground	3	lidar	1427	72.9
M2DGR Gate	3	lidar	484	8.3
S3E Square	3	stereo & lidar	2286	19.1
S3E College	3	stereo & lidar	3335	31.7
S3E Playground	3	stereo & lidar	1232	8.7
S3E Laboratory	3	stereo & lidar	468	10.3

VI. Experimental Results

To evaluate the effectiveness of our proposed solutions for the ongoing challenges in Collaborative Simultaneous Localization and Mapping, we conducted extensive experiments on several public datasets, as well as in a real-world deployment. Our experiments involved three robots exploring and mapping an indoor environment and communicating via ad-hoc networking. We specifically evaluated our key contributions to



(a) KITTI 00 (b) KITTI-360 09 (c) GrAco Ground (d) S3E Square

Fig. 4: Comparison between *Greedy* and *Spectral* prioritization of candidate inter-robot loop closures. For each datasets, we show that the *spectral* approach outperforms the *greedy* one in terms of algebraic connectivity (higher is better), final objective value of the pose graph optimization (lower is better), and the *Absolute Translation Error* (ATE) (lower is better). We can see that the *spectral* prioritization converges to the best estimate with less inter-robot loop closures than the *greedy* prioritization which tends to prioritize redundant measurements.

inter-robot loop closure detection and decentralized C-SLAM estimation. Additionally, we present detailed statistics of the communication and computation load during our real-world experiment, providing insight into the system's performance and resource requirements.

A. Dataset Experiments

As detailed in Table II, we tested Swarm-SLAM on eight sequences from five different datasets. To demonstrate the flexibility of our framework, we used IMUs, stereo cameras, lidars, or a combination as inputs. First, we tested on the widely known autonomous driving KITTI 00 stereo sequence [40] which we split into two parts to simulate a two-robots exploration. Second, we split the very large (∼10km) KITTI360 09 lidar sequence [41] into 5 parts that contain a large number of loop closures, making it particularly well suited for inter-robot loop closure detection analysis. Third, we experimented on the first three overlapping lidar sequences of the very recent GrAco dataset [42] acquired with custom ground robots on a college campus. Fourth, we evaluate our system on the three lidar Gate sequences of the M2DGR dataset [43]. Fifth, we tested on four sequences of the recent C-SLAM-focused S3E dataset [44]. To avoid tracking failures and obtain more robust results on S3E sequences, we combined lidar-IMU odometry and stereo camera-based inter-robot loop closure detection, highlighting the versatility of Swarm-SLAM. Overall, we chose the sequences with the most trajectory overlaps to obtain more loop closures, and with available GPS ground truth (except for S3E Laboratory). For simplicity and robustness, we used off-the-shelf RTAB-Map [36] modules to compute and provide the required odometry input to Swarm-SLAM, but this could be replaced by any other odometry software.

To better evaluate the inter-robot loop closure detection, we consider the worst-case scenario in which the robots are within communication range only at the end of their trajectories, such that they have to find loop closures between their whole maps at once instead of processing them incrementally. This scenario, analog to multiple robots exploring different parts of an environment and meeting back at the end, is among the most challenging in terms of communication and computation load, and therefore benefits the most from our novel spectral candidate prioritization mechanism. We refer the reader to our open-source implementation for all the parameters and configuration details of the experiments.

1) Inter-Robot Loop Closure Detection Evaluation: In Fig. 4, we compare the greedy and spectral inter-robot loop closure candidate prioritization techniques with respect to algebraic connectivity, PGO objective value, and Absolute Translation Error (ATE). Each approach is used to priori-

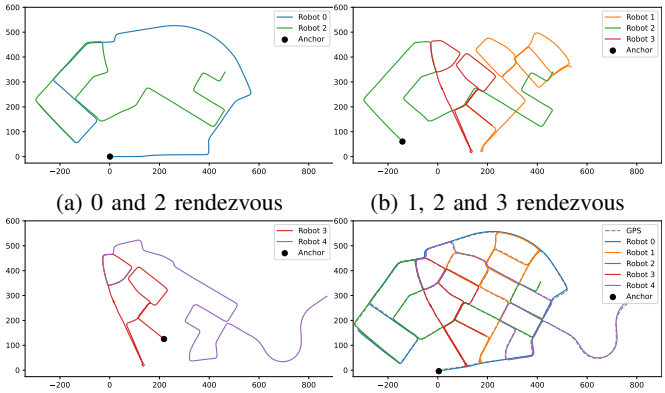
TABLE III: C-SLAM Estimates Evaluation on Public Datasets.

		Communication (kB)			Time (s)			ATE (m)		
	# Robots	Ours	DGS+PCM	D-GNC	Ours	DGS+PCM	D-GNC	Ours	DGS+PCM	D-GNC
KITTI 00	2	**280.00**	18730.96	6144.12	**20.11**	191.89	62.38	**2.17**	14.18	4.13
KITTI-360 09	5	**484.91**	1743.89	4224.48	196.93	79.54	**58.22**	**4.02**	6.83	7.33
GrAco Ground	3	**105.82**	2052.05	806.00	**8.06**	44.67	20.57	**6.19**	47.91	13.33
M2DGR Gate	3	**51.29**	215.34	696.56	**1.42**	4.79	7.62	**0.70**	20.56	3.11
S3E Square	3	**80.97**	1000.44	288.24	**6.05**	44.06	13.11	**4.20**	11.30	25.57
S3E College	3	**150.37**	3105.39	1529.84	**11.96**	116.34	19.06	**3.57**	32.94	4.36
S3E Playground	3	**43.93**	653.13	840.88	**1.08**	22.50	7.63	**3.54**	11.96	4.97

tize the computation of loop closures from the same set of candidates with a budget of 1. We plot each metric against the percentage of loop closures computed within the set of candidate (x-axis). Given enough time, both techniques will go through all the possible matches (i.e., 100% of loop closures computed). We expect that a better prioritization will need to select fewer candidates to converge to the best possible accuracy. We compute the ATE in Fig. 4 with the evo package [45], against the final pose graph estimate containing all possible inter-robot loop closures, and thus constitutes the best estimate we can achieve.

On the first row, we can see that, as intended, our spectral prioritization is correctly maximizing the algebraic connectivity of the pose graph. On the second row, we can see that PGO objective value converges faster to its final value using our approach. On the third row, as expected, we can see that our spectral prioritization decreases the error faster than the greedy prioritization. In summary, our experiments show that our technique requires the computation of fewer inter-robot loop closures to significantly reduce the estimation error (ATE), and thus approximate the final pose graph. This result also confirms our hypothesis that algebraic connectivity is a relevant metric for inter-robot loop closure candidates prioritization.

2) Decentralized C-SLAM Evaluation: In Table III, we present the estimates computed in the back-end on all the sequences for which GPS latitude and longitude data is available as ground truth. We report the computation time on a AMD Ryzen 7 CPU and the total communication required in kB. We compared against two state-of-the-art approaches: the Distributed Gauss-Seidel (DGS) pose graph optimization [15] combined with Pairwise Consistency Maximization (PCM) [20] for outlier rejection as used in [11]; and a distributed implementation of Graduated Non-Convexity (D-GNC) [22] based on the RCBD solver [16]. Our approach systematically obtains the most accurate ATE while the other methods often fail to obtain reasonable estimates. Swarm-SLAM also consistently outperforms the distributed optimization approaches in terms of required communication and computation time. Interestingly, on KITTI-360 09 which is our largest dataset, DGS+PCM and D-GNC require less computation time than our approach, but fail to obtain a comparable accuracy and require over 20x more data transmission. While distributed approaches should benefit from the division of labour on large problems, more research is required to obtain the same levels of accuracy, robustness, and communication bandwidth as our simpler approach.

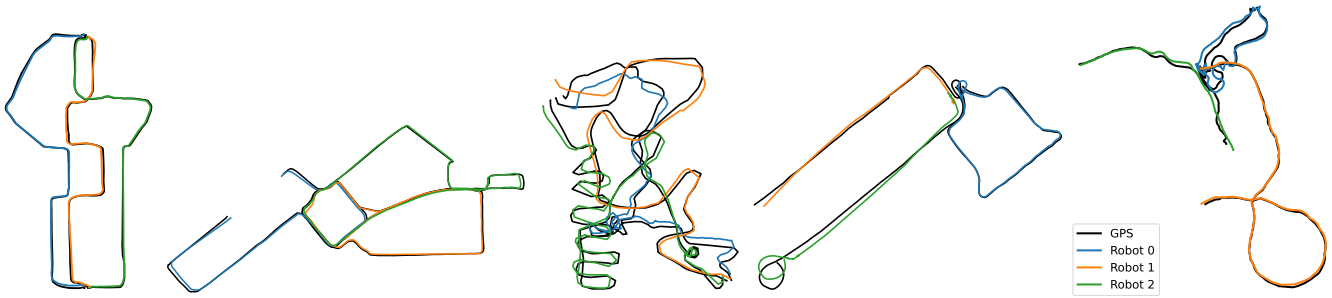


(a) 0 and 2 rendezvous (b) 1, 2 and 3 rendezvous

(c) 3 and 4 rendezvous (d) All robots rendezvous

Fig. 5: Reference frame convergence via sporadic rendezvous. From (a) to (c) we trajectory estimates from successive rendezvous between different groups of robots: {0,2}, then {1,2,3}, and finally {3,4}. After the three rendezvous, thanks to our anchor selection mechanism, all estimates are within robot 0 reference frame which acts as the global reference frame. For comparison, we also include the result of a rendezvous with all robots and the GPS ground truth (d).

In Fig. 5, we show the Swarm-SLAM resulting estimates on the KITTI360 09 sequence from four different rendezvous, defined as a sporadic encounter in which a subset of robots are within communication range of each other. Our simple anchor selection scheme ensures that by choosing the current first pose estimate from the robot with the lowest reference frame ID (i.e., first poses of (a) robot 0, (b) robot 2, (c) robot 3), we can propagate the global reference frame among the team of robots. In other words, we are able to converge to a single global reference frame through successive estimations between subsets of robots, without enforcing connectivity maintenance or a central authority. This decentralized approach improves the scalability of the system by relying only on local interactions among neighboring robots. We present the visualization of the Swarm-SLAM solutions on the remaining dataset sequences in Fig. 6.



(a) S3E Square (b) S3E College (c) S3E Playground (d) GrAco Ground (e) M2DGR

Fig. 6: Swarm-SLAM trajectory estimates on various dataset sequences compared with GPS ground truth.

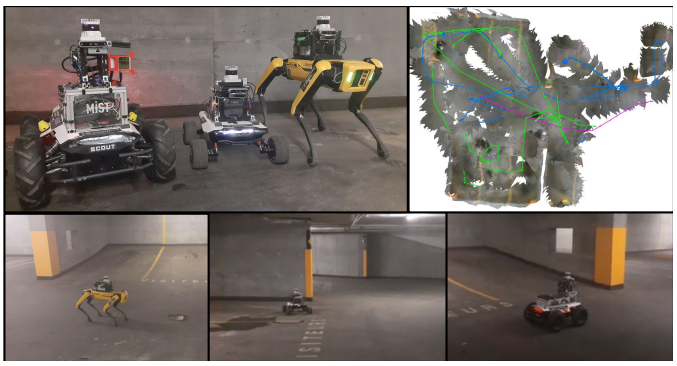


Fig. 7: Swarm-SLAM experiment with 3 robots, equipped with lidars and RGB-D cameras, simultaneously exploring an indoor parking lot, and achieving shared situational awareness via collaborative perception. A visualization of the resulting map and pose graphs is showed in the top right corner.

TABLE IV: Real-World Experiment Statistics

# Robots	3	Length (m)	475.42
# Keyframes	3103	Total comm. (MB)	94.95
# Inter-robot loop cl.	67	# Outliers	10
PGO time (s)	5.52 ± 7.11	Sparsification time (s)	2.71 ± 2.39

B. Real-World Experiments

To assess the viability of Swarm-SLAM on resource-constrained platforms, we deployed the system in an indoor parking lot and gathered statistics regarding the computation time and communication load. As shown in Fig. 7, we performed an online real-world demonstration with 3 different robots (Boston Dynamics Spot, Agilex Scout, and Agilex Scout Mini), all equipped with an NVIDIA Jetson AGX Xavier onboard computer, an Intel Realsense D455 camera, an Ouster lidar OS0-64, a VectorNav VN100 IMU, and a GL-iNet GL-S1300 OpenWrt gateway for ad hoc networking. We used lidars and IMUs for odometry and the RGB-D cameras for inter-robot loop closure detection. We started the exploration with the first two robots, and the third robot, which was initially out of range, joined the experiment later. Upon arrival within communication range, the third robot was dynamically discovered by the first two robots via ROS 2, and included in the estimation process. At the cost of extra communication, our minimal visualization tool, running on a laptop connected to the ad hoc network, gathered the resulting estimates from the robots in real time, along with mapping data that would not be transmitted otherwise.

As stated in Table IV, our robots travelled a total of 475 meters during the experiment and produced a total of 3103 keyframes that needed to be matches and verified in the search for inter-robot loop closures. The process resulted in 67 loop closures, including 10 that were rejected by the GNC optimizer (i.e., due to high residual factors). The large number of outliers is attributable to the many similar-looking sections of the parking lot. Keyframe images from different parts of the parking lot are erroneously matched by the place recognition system leading to spurious loop closure measurements (i.e., perceptual aliasing). In total, Swarm-SLAM required the transmission of only 94.95 MB of data between the robots, not including the visualization. The communication load is mostly attributable to the front-end and thus dependent on the number of keyframes. In Table IV, we also report the average sparsification and pose graph optimization times. We can observe that the sparsification time, while being non-negligible, is lower than the pose graph optimization. To mitigate the PGO computation time, we implemented it within a separate thread such that we can continue to accumulate additional global and local measurements while waiting for the next estimate. Also, we continuously provide to users an up-to-date pose estimate, published at a fixed rate, completing PGO estimates with the latest odometry measurements.

VII. Conclusions And Future Work

In this paper, we presented Swarm-SLAM, a comprehensive open-source C-SLAM framework that is designed to comply with essential properties of swarm robotics. We have introduced a novel inter-robot loop closure prioritization technique based on algebraic connectivity maximization that reduces communication and speed-up convergence. Our system has been rigorously evaluated on five different public datasets, as well as in a real-world experiment, demonstrating its effectiveness and suitability for multi-robot operations in environments without an external positioning system. By leveraging ROS 2, we have shown that Swarm-SLAM is adapted to ad-hoc networking with sporadic or intermittent local communication

between neighboring robots. While our framework's scalability is still limited by the underlying single-robot SLAM efficiency and onboard resource constraints, our decentralized framework can scale to large groups of robots by performing successive C-SLAM estimation with smaller subsets of robots and still converge to globally consistent estimates.

In future work, we plan to extend Swarm-SLAM to support additional sensing modes, beyond lidar, stereo and RGB-D, such as higher-level data representations (e.g. topological or semantic maps). We also aim to investigate collaborative domain calibration and/or uncertainty estimation in place recognition [46] to reduce the prevalence of measurement outliers among inter-robot loop closures, and therefore increase the overall accuracy and resilience of C-SLAM. Additionally, we aim to integrate and deploy a Swarm-SLAM-enabled autonomous exploration system in large unknown environments and improve our visualization tools for better human-multi-robot interactions.

Finally, we hope that our open-source framework will be useful as a testbed for the research and development of new methods and techniques in place recognition, inter-robot loop closure detection, multi-robot pose graph optimization, and other challenging open-problems in C-SLAM.

REFERENCES

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.

[2] P.-Y. Lajoie, B. Ramtoula, F. Wu, and G. Beltrame, "Towards Collaborative Simultaneous Localization and Mapping: A Survey of the Current Research Landscape," *Field Robotics*, vol. 2, no. 1, pp. 971–1000, Mar. 2022.

[3] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: A review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, Mar. 2013.

[4] M. Kegeleirs, G. Grisetti, and M. Birattari, "Swarm SLAM: Challenges and Perspectives," *Frontiers in Robotics and AI*, vol. 8, 2021.

[5] K. J. Doherty, D. M. Rosen, and J. J. Leonard, "Spectral Measurement Sparsification for Pose-Graph SLAM," *arXiv:2203.13897 [cs]*, Mar. 2022.

[6] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot Operating System 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, May 2022.

[7] S. Saeedi, M. Trentini, M. Seto, and H. Li, "Multiple-Robot Simultaneous Localization and Mapping: A Review," *Journal of Field Robotics*, vol. 33, no. 1, pp. 3–46, 2016.

[8] B. Kim, M. Kaess, L. Fletcher, J. Leonard, A. Bachrach, N. Roy, and S. Teller, "Multiple relative pose graphs for robust cooperative mapping," in *2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 3185–3192.

[9] Y. Cao and G. Beltrame, "VIR-SLAM: Visual, inertial, and ranging SLAM for single and multi-robot systems," *Autonomous Robots*, vol. 45, no. 6, pp. 905–917, Sep. 2021.

[10] T. Cieslewski, S. Choudhary, and D. Scaramuzza, "Data-Efficient Decentralized Visual SLAM," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 2466–2473.

[11] P.-Y. Lajoie, B. Ramtoula, Y. Chang, L. Carlone, and G. Beltrame, "DOOR-SLAM: Distributed, Online, and Outlier Resilient SLAM for Robotic Teams," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1656–1663, Apr. 2020.

[12] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1437–1451, Jun. 2018.

[13] G. Kim and A. Kim, "Scan Context: Egocentric Spatial Descriptor for Place Recognition Within 3D Point Cloud Map," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 4802–4809.

[14] D. Tardioli, E. Montijano, and A. R. Mosteo, "Visual data association in narrow-bandwidth networks," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2015, pp. 2572–2577.

[15] S. Choudhary, L. Carlone, C. Nieto, J. Rogers, H. I. Christensen, and F. Dellaert, "Distributed mapping with privacy and communication constraints: Lightweight algorithms and object-based models," *The International Journal of Robotics Research*, vol. 36, no. 12, pp. 1286–1311, Oct. 2017.

[16] Y. Tian, K. Khosoussi, D. M. Rosen, and J. P. How, "Distributed Certifiably Correct Pose-Graph Optimization," *IEEE Transactions on Robotics*, pp. 1–20, 2021.

[17] Y. Tian, A. Koppel, A. S. Bedi, and J. P. How, "Asynchronous and Parallel Distributed Pose Graph Optimization," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5819–5826, Oct. 2020.

[18] R. Murai, J. Ortiz, S. Saeedi, P. H. J. Kelly, and A. J. Davison, "A Robot Web for Distributed Many-Device Localisation," Feb. 2022.

[19] P.-Y. Lajoie, S. Hu, G. Beltrame, and L. Carlone, "Modeling Perceptual Aliasing in SLAM via Discrete–Continuous Graphical Models," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1232–1239, Apr. 2019.

[20] J. G. Mangelson, D. Dominic, R. M. Eustice, and R. Vasudevan, "Pairwise Consistent Measurement Set Maximization for Robust Multi-Robot Map Merging," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 2916–2923.

[21] H. Yang, P. Antonante, V. Tzoumas, and L. Carlone, "Graduated Non-Convexity for Robust Spatial Perception: From Non-Minimal Solvers to Global Outlier Rejection," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1127–1134, Apr. 2020.

[22] Y. Tian, Y. Chang, F. Herrera Arias, C. Nieto-Granda, J. P. How, and L. Carlone, "Kimera-Multi: Robust, Distributed, Dense Metric-Semantic SLAM for Multi-Robot Systems," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2022–2038, Aug. 2022.

[23] P. Schmuck, T. Ziegler, M. Karrer, J. Perraudin, and M. Chli, "COVINS: Visual-Inertial SLAM for Centralized Collaboration," in *2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, Oct. 2021, pp. 171–176.

[24] Y. Huang, T. Shan, F. Chen, and B. Englot, "DiSCo-SLAM: Distributed Scan Context-Enabled Multi-Robot LiDAR SLAM With Two-Stage Global-Local Graph Optimization," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1150–1157, Apr. 2022.

[25] Y. Chang, K. Ebadi, C. E. Denniston, M. F. Ginting, A. Rosinol, A. Reinke, M. Palieri, J. Shi, A. Chatterjee, B. Morrell, A.-a. Aghamohammadi, and L. Carlone, "LAMP 2.0: A Robust Multi-Robot SLAM System for Operation in Challenging Large-Scale Underground Environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9175–9182, Oct. 2022.

[26] A. Cramariuc, L. Bernreiter, F. Tschopp, M. Fehr, V. Reijgwart, J. Nieto, R. Siegwart, and C. Cadena, "Maplab 2.0 – A Modular and Multi-Modal Mapping Framework," Dec. 2022.

[27] G. Kurz, M. Holoch, and P. Biber, "Geometry-based Graph Pruning for Lifelong SLAM," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2021, pp. 3313–3320.

[28] Y. Tian, K. Khosoussi, and J. P. How, "A resource-aware approach to collaborative loop-closure detection with provable performance guarantees," *The International Journal of Robotics Research*, vol. 40, no. 10-11, pp. 1212–1233, Sep. 2021.

[29] K. Khosoussi, M. Giamou, G. S. Sukhatme, S. Huang, G. Dissanayake, and J. P. How, "Reliable Graphs for SLAM," *The International Journal of Robotics Research*, vol. 38, no. 2-3, pp. 260–298, Mar. 2019.

[30] K. J. Doherty, D. M. Rosen, and J. J. Leonard, "Performance Guarantees for Spectral Initialization in Rotation Averaging and Pose-Graph SLAM," *arXiv:2201.03773 [cs]*, Jan. 2022.

[31] C. E. Denniston, Y. Chang, A. Reinke, K. Ebadi, G. S. Sukhatme, L. Carlone, B. Morrell, and A.-a. Agha-mohammadi, "Loop Closure Prioritization for Efficient and Scalable Multi-Robot SLAM," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9651–9658, Oct. 2022.

[32] Y. Tian and J. P. How, "Spectral Sparsification for Communication-Efficient Collaborative Rotation and Translation Estimation," Oct. 2022.

[33] G. Berton, C. Masone, and B. Caputo, "Rethinking Visual Geo-localization for Large-Scale Applications," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. New Orleans, LA, USA: IEEE, Jun. 2022, pp. 4868–4878.

[34] D. Mosk-Aoyama, "Maximum algebraic connectivity augmentation is NP-hard," *Operations Research Letters*, vol. 36, no. 6, pp. 677–679, Nov. 2008.

[35] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 International Conference on Computer Vision*, Nov. 2011, pp. 2564–2571.

[36] M. Labbé and F. Michaud, "RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of Field Robotics*, vol. 36, no. 2, pp. 416–446, Mar. 2019.

[37] H. Yang, J. Shi, and L. Carlone, "TEASER: Fast and Certifiable Point Cloud Registration," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 314–333, Apr. 2021.

[38] M. Giamou, K. Khosoussi, and J. P. How, "Talk Resource-Efficiently to Me: Optimal Communication Planning for Distributed Loop Closure Detection," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 3841–3848.

[39] F. Dellaert et al., "Georgia Tech Smoothing And Mapping (GTSAM)," http://gtsam.org/.

[40] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. Providence, RI: IEEE, Jun. 2012, pp. 3354–3361.

[41] Y. Liao, J. Xie, and A. Geiger, "KITTI-360: A Novel Dataset and Benchmarks for Urban Scene Understanding in 2D and 3D," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2022.

[42] Y. Zhu, Y. Kong, Y. Jie, S. Xu, and H. Cheng, "GRACO: A Multimodal Dataset for Ground and Aerial Cooperative Localization and Mapping," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 966–973, Feb. 2023.

[43] J. Yin, A. Li, T. Li, W. Yu, and D. Zou, "M2DGR: A Multi-Sensor and Multi-Scenario SLAM Dataset for Ground Robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2266–2273, Apr. 2022.

[44] D. Feng, Y. Qi, S. Zhong, Z. Chen, Y. Jiao, Q. Chen, T. Jiang, and H. Chen, "S3E: A Large-scale Multimodal Dataset for Collaborative SLAM," Oct. 2022.

[45] M. Grupp, "evo: Python package for the evaluation of odometry and slam." https://github.com/MichaelGrupp/evo, 2017.

[46] P.-Y. Lajoie and G. Beltrame, "Self-supervised domain calibration and uncertainty estimation for place recognition," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 792–799, Feb. 2023.