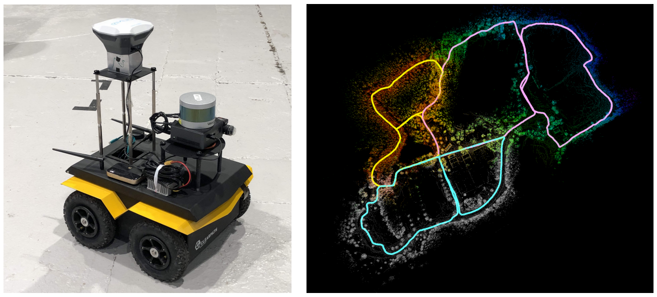
DiSCo-SLAM: Distributed Scan Context-Enabled Multi-Robot LiDAR SLAM with Two-Stage Global-Local Graph Optimization

Yewei Huang[1], Tixiao Shan[2], Fanfei Chen[3] and Brendan Englot[1]

***Abstract*—We propose a novel framework for distributed, multi-robot SLAM intended for use with 3D LiDAR observations. The framework, DiSCo-SLAM, is the first to use the lightweight Scan Context descriptor for multi-robot SLAM, permitting a data-efficient exchange of LiDAR observations among robots. Additionally, our framework includes a two-stage global and local optimization framework for distributed multi-robot SLAM which provides stable localization results that are resilient to the unknown initial conditions that typify the search for inter-robot loop closures. We compare our proposed framework with the widely used distributed Gauss-Seidel (DGS) approach, over a variety of multi-robot datasets, quantitatively demonstrating its accuracy, stability, and data-efficiency.**

***Index Terms*—Multi-Robot SLAM, Distributed Robot Systems, Range Sensing.**



(a) Our Jackal UGV instrumented with LiDAR, IMU, and a GPS used for ground truth.

(b) SLAM-derived map, with pink, yellow and blue lines indicating the estimated trajectories of three robots.

Fig. 1: Our mobile UGV platform (a) and a representative three-robot SLAM result using the proposed DiSCo-SLAM framework, over a new outdoor SLAM dataset gathered using our UGV (b).

I. Introduction

Simultaneous localization and mapping (SLAM) is a fundamental capability in robot navigation, in which a mobile robot maps an unknown environment, while using relative measurements of that environment as the basis for localizing itself. Although many successful single-robot SLAM solutions have been proposed, fast and accurate scene reconstruction with a robot team remains an open problem. In multi-robot SLAM, a group of robots traverse an unknown environment and build a map cooperatively, by exchanging information. Cooperative robot teams have the potential to be more efficient than a single robot in time-sensitive tasks, such as search-and-rescue, infrastructure inspection, household services, and logistical and transportation applications.

Single-robot SLAM solutions incorporating 3D LiDAR have become increasingly prevalent in recent years, since 3D LiDAR scans provide high-resolution point clouds spanning a large volumetric field of view, and are robust to a wide range of weather and lighting conditions. However, many robot teams are equipped with either cameras or 2D LiDAR scanners, since visual features and 2D LiDAR scans are lightweight and can be exchanged between robots under low-bandwidth conditions. Multi-robot SLAM has been successfully implemented with 3D LiDAR, but such frameworks have typically required a centralized server [1], or have not been applied in online experiments with 3D LiDAR [2], [3].

In this paper, we propose a distributed multi-robot SLAM framework intended for real-time use with 3D LiDAR, and with a two-stage global-local graph optimization procedure designed for robust compatibility with smoothing-and-mapping optimizers [4]. We use Scan Context [5], a lightweight LiDAR descriptor for loop closure detection, to compactly represent LiDAR scans and permit a low-bandwidth exchange of information with other robots. As descriptors are received from other robots, the recipient robot performs a radius search and requests the full set of point features from the best-matched keyframe. Unlike intra-robot loop closures, inter-robot loop closures often lack access to an accurate initial guess from odometry information. Since a widely-used optimization method, the distributed Gauss-Seidel (DGS) approach, is not reliable with a poor initial guess, we propose a two-stage global and local optimization strategy. In the global step, a factor graph containing a robot's local-frame to global-frame transformations is optimized. In the subsequent local step, a local pose graph encompassing (1) local odometry, (2) intra-robot constraints, and (3) inter-robot constraints related to the local robot is optimized.

We validate our framework using several multi-robot 3D LiDAR datasets, including a unique dataset gathered with our Jackal unmanned ground vehicle (UGV) (Fig. 1(a)), capable of tightly-coupled lidar inertial odometry [6], equipped with a single-band RTK GNSS receiver for ground truth.

In summary, the novel contributions of our work include:

- The first use of the Scan Context descriptor in multi-robot LiDAR SLAM, for data-efficient communication.

Manuscript received: September 9, 2021; Revised December 1, 2021; Accepted December 15, 2021.

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

This work was supported in part by NSF grants IIS-1652064 and IIS-1723996, and ONR grant N00014-21-1-2161.

[1] Y. Huang and B. Englot are with Stevens Institute of Technology, Hoboken, NJ, USA 07030. {yhuang85, benglot}@stevens.edu

[2] T. Shan is with SRI International, Princeton, NJ, USA 08540.

[3] F. Chen is with Exyn Technologies, Philadelphia, PA, USA 19146.

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

- A two-stage global-local optimization for each robot; optimizing a global coordinate transformation graph ensures a high quality initial guess for stably and accurately optimizing a robot's local pose graph.
- A unique and publicly available large-scale three-robot SLAM dataset with tightly-coupled 3D lidar and inertial sensing, and GPS ground truth, gathered for this study.

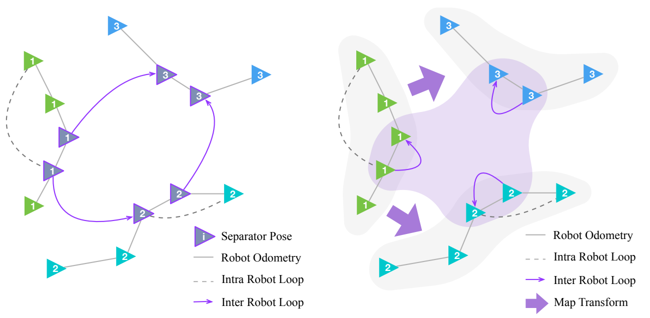
The rest of our paper is organized as follows. After a review of the relevant background literature in Section II, the definition of distributed multi-robot SLAM and our proposed two-stage optimization for distributed multi-robot SLAM are presented in Section III. In Section IV, a framework for distributed multi-robot SLAM using the Scan Context LiDAR descriptor is presented. Experimental results are given in Section V, with conclusions in Section VI.

II. Related Work

It is pivotal for a multi-robot SLAM algorithm, especially for distributed multi-robot SLAM [7], [8], whose communication bandwidth is limited, to establish accurate inter-robot measurement constraints, or loop closures. In single-robot SLAM, odometry measurements are frequently available to provide an accurate initial guess. In multi-robot SLAM, inter-robot constraints must be derived from perceptual data.

Since feature descriptors are easy to transfer and query, many vision-based methods [9], [10], [11], [12], [13] extract features from visual imagery and find potential data association using Bag of Words (BoW). Feature extractions are also widely used in 3D LiDAR SLAM methods. LeGO-LOAM [14] and [15] extract point features representing edge and planar structures to scale down the data while ensuring high performance. Segmap [16] extracts segments from LiDAR point clouds and describes them with CNN-based features. Recent research [17] also discusses the possibility of using visual features for LiDAR place recognition. DARE-SLAM [18] extracts visual features from 2D occupancy grid maps as geometric verification for inter-robot loop closures. We choose Scan Context [5] as our feature descriptor since it is lightweight and robust. Scan Context describes the raw LiDAR point cloud by projecting the scan onto a low-resolution 2D plane, which is easily searched and exchanged.

Although features enable robust performance in many cases, they cannot distinguish among repeated scenes in the environment. Many works address this issue through selectivity in accepting inter-robot loop closures. To ensure the correctness of inter-robot data association, [2], [19], [20] use a smart rendezvous approach; robots only exchange the present poses and observations when they meet. These meeting places are where inter-robot loop closures typically occur. However, a rendezvous approach only provides rough position information and may introduce uncertainties into subsequent optimization steps. Recent methods introduce an outlier rejection procedure to detect erroneous loop closures derived from feature matching. Agarwal [21] adds a scale factor to the covariance matrix to dynamically adjust the influence of each measurement. Pairwise consistent measurement set maximization (PCM) [22], proposed for multi-robot map merging, checks the consistency of inter-robot measurements. It has been adopted by others for outlier rejection in multi-robot SLAM [3], and we also use it.



(a) DGS method: separator poses are where robots rendezvous or share common observations - they are included in each robot's optimization.

(b) Illustration of our method, where a global coordinate transformation graph (purple) and local pose graph (gray) are optimized by each robot.

Fig. 2: Differences between DGS and our proposed method. Green, cyan and blue triangles indicate the poses of three different robots.

Another critical problem is the optimization of inter-robot constraints. Centralized methods [1], [9], [10], [11], [12], which collect all messages from local robots using a global server, can easily handle this task by optimizing all measurements in one factor graph. Distributed methods maintain several graphs across robots, making it harder to resolve ambiguities arising among them. In the widely-used distributed Gauss-Seidel (DGS) approach [2], each robot optimizes its own graph, considering other robots only when there are overlapping constraints. DDF-SAM [23] optimizes a local pose graph and constrains these local graphs using a constrained factor graph (CFG) containing shared landmarks among robots. DDF-SAM 2.0 [24] uses a decision tree in both local and neighbor graphs to avoid double-counting measurements. Tian [25] proposes Riemannian Block-Coordinate Descent (RBCD), which achieves a better convergence than the DGS method by solving a rank-restricted relaxation of the pose graph optimization problem [26]. We adopt a two-stage optimization approach that is tailored to LiDAR SLAM with a data-efficient descriptor, with few restrictions on the prerequisites for relative pose estimation, and, unlike other works [2], [3], we perform local optimization in a single unified step that does not address rotation/translation separately.

III. Problem Formulation and Approach

Let $\mathbf{X} = \{\mathbf{x}_0, \cdots, \mathbf{x}_t\}$ be a set of 6DoF robot poses from time 0 to time t, with $\mathbf{X} \subset \mathrm{SE}(3)$. C contains all constraints between robot poses. For each pair of poses defining a constraint $\langle i, j\rangle \in C$, we define error $\mathbf{e}_{ij}$ between observed transformation $\mathbf{z}_{ij} \in \mathrm{SE}(3)$ and expected transformation $\hat{\mathbf{z}}_{ij}$ as:

$$\mathbf{e}_{ij}(\mathbf{x}_i, \mathbf{x}_j, \mathbf{z}_{ij}) = \mathbf{z}_{ij} - \hat{\mathbf{z}}_{ij}(\mathbf{x}_i, \mathbf{x}_j), \tag{1}$$

$$\hat{\mathbf{z}}_{ij}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j. \tag{2}$$

A single-agent SLAM problem can be represented as a nonlinear least squares problem [27]:

$$\mathbf{X}^* = \arg\min_{\mathbf{X}} \sum_{\langle i,j\rangle \in C} \mathbf{F}_{ij} \tag{3}$$

$$\mathbf{F}_{ij} = \mathbf{e}_{ij}^T \Omega_{ij} \mathbf{e}_{ij}, \tag{4}$$

where $\mathbf{F}_{ij}$ denotes the negative log-likelihood function of one constraint between $\mathbf{x}_i$ and $\mathbf{x}_j$, and Ω_{ij} is the covariance. The SLAM system aims to find a set of robot poses which minimizes the total observation error.

We define a set of n robots: $\mathcal{N} = \{1, 2, \cdots, n\}$. $\forall \alpha \in \mathcal{N}$, $\mathbf{X}_\alpha$ denotes the poses of robot α and $\mathbb{X} = \{\mathbf{X}_\alpha | \alpha \in \mathcal{N}\}$ is a set containing the poses of all n robots. $\mathbb{C} = \{\langle i, j\rangle | \mathbf{x}_i, \mathbf{x}_j \in \mathbb{X}\}$ is the set of all constraints. For the multi-agent SLAM problem, we wish to solve the following equation:

$$\mathbb{X}^* = \underset{\mathbb{X}}{\arg\min}\{\mathbf{F}_{intra}(\mathbb{X}) + \mathbf{F}_{inter}(\mathbb{X})\}. \tag{5}$$

The cost function consists of two parts: intra-robot terms and inter-robot terms. $C_\alpha \subset \mathbb{C}$ indicates the set of intra-robot constraints between poses of robot α, and $C_{\alpha\beta} \subset \mathbb{C}$ indicates inter-robot constraints between poses of robots α and β.

$$\mathbf{F}_{intra}(\mathbb{X}) = \sum_{\alpha \in \mathcal{N}} \sum_{\langle i,j\rangle \in C_\alpha} \mathbf{F}_{ij} \tag{6}$$

$$\mathbf{F}_{inter}(\mathbb{X}) = \sum_{\alpha,\beta \in \mathcal{N}, \alpha \neq \beta} \sum_{\langle i,j\rangle \in C_{\alpha\beta}} \mathbf{F}_{ij} \tag{7}$$

In the distributed case, each robot optimizes its own contributions to the objective. For robot α, we have:

$$\mathbb{X}^*_\alpha = \underset{\mathbb{X}_\alpha}{\arg\min}(\overbrace{\sum_{\langle i,j\rangle \in C_\alpha} \mathbf{F}_{ij}}^{\mathbf{F}_{intra}} + \overbrace{\sum_{\beta \in \mathcal{N}, \alpha\neq\beta} \sum_{\langle i,j\rangle \in C_{\alpha\beta}} \mathbf{F}_{ij}}^{\mathbf{F}_{inter}}) \tag{8}$$

$$\mathbb{X}_\alpha = \mathbf{X}_\alpha \cup \left\{ \mathbf{x}_j \,\middle|\, \begin{matrix} \mathbf{x}_j \in \mathbf{X}_\beta, \langle i, j\rangle \in C_{\alpha\beta}, \\ \forall \beta \in \mathcal{N}, \beta \neq \alpha. \end{matrix} \right\}, \tag{9}$$

where $\mathbb{X}_\alpha$ contains all poses related to robot α. $\forall \mathbf{x} \in \mathbb{X}_\alpha$, the robot pose $\mathbf{x}$ consists of two parts, the rotation $\mathbf{R}$ and the translation $\mathbf{t}$. Since the rotation $\mathbf{R} \in SO(3)$ is a non-convex component [2], Eq. (8) may fall into local minima instead of converging to a global minimal solution.

To address this, the widely used distributed Gauss-Seidel (DGS) method [2] (illustrated in Fig. 2(a)) rewrites $\mathbb{X}_\alpha$ as two subsets: $\mathcal{R}_\alpha$, containing the rotations of all poses, and $\sqcup_\alpha$, containing the translations of all poses. Additionally, DGS entails a two-stage optimization process. For robot α, the DGS method first approximates the rotation $\mathcal{R}_\alpha$:

$$\mathcal{R}^*_\alpha = \underset{\mathcal{R}_\alpha}{\arg\min}(\sum_{\langle i,j\rangle \in C_\alpha} \mathbf{G}_{ij} + \sum_{\beta \in \mathcal{N}, \alpha\neq\beta} \sum_{\langle i,j\rangle \in C_{\alpha\beta}} \mathbf{G}_{ij}). \tag{10}$$

$\mathbf{G}_{ij}$ is the negative log-likelihood function only considering a robot's rotation:

$$\mathbf{G}_{ij} = [\mathbf{C}_{ij} - \hat{\mathbf{C}}_{ij}(\mathbf{R}_i, \mathbf{R}_j)]^T \omega_R [\mathbf{C}_{ij} - \hat{\mathbf{C}}_{ij}(\mathbf{R}_i, \mathbf{R}_j)]. \tag{11}$$

Similar to $\mathbf{z}_{ij}$ and $\hat{\mathbf{z}}_{ij}$, $\mathbf{C}_{ij}$ and $\hat{\mathbf{C}}_{ij}$ are the observed and expected relative rotation between $\mathbf{R}_i$ and $\mathbf{R}_j$. We rewrite Ω_{ij} as $\begin{bmatrix} \omega_R & 0 \\ 0 & \omega_t \end{bmatrix}$, where ω_R is the rotation block of Ω_{ij}. Then, the method performs a full-state graph optimization via the Gauss-Newton method, with the optimized rotation guess $\mathcal{R}^*_\alpha$, to solve Eq. (8). However, the full-state optimization step requires a good rotation approximation, while the first step is still solving a non-convex problem. So DGS may require a long time to converge with a poor initial guess.

Inspired by the DGS method, we propose a two-stage global-local graph optimization. The initial global optimization step solves the transformation among robots. $\forall$ pairs of separator poses $\langle \alpha_i, \beta_j\rangle \in C_{\alpha\beta}$, with $\mathbf{x}_{\alpha_i} \in \mathbf{X}_\alpha$, $\mathbf{x}_{\beta_j} \in \mathbf{X}_\beta$, we define the local robot frame for robot α as $\underset{\rightarrow}{\mathcal{F}}_\alpha$. Let ${}_\alpha\mathbf{x}_{\alpha_i}$ be the pose at time i of robot α in its local coordinates, while ${}_\beta\mathbf{x}_{\alpha_i}$ is $\mathbf{x}_{\alpha_i}$ in robot β's local coordinates. $\exists \mathbf{T}_{\beta\alpha}$, transforming ${}_\alpha\mathbf{x}_{\alpha_i}$ to ${}_\beta\mathbf{x}_{\alpha_i}$ gives us:

$${}_\beta\mathbf{x}_{\alpha_i} = \mathbf{T}_{\beta\alpha} \cdot {}_\alpha\mathbf{x}_{\alpha_i} = {}_\beta\mathbf{x}_{\beta_j} \cdot {}_\beta\mathbf{z}_{\beta_j\alpha_i} \tag{12}$$

$$\mathbf{T}_{\beta\alpha} = {}_\beta\mathbf{x}_{\beta_j} \cdot {}_\beta\mathbf{z}_{\beta_j\alpha_i} \cdot ({}_\alpha\mathbf{x}_{\alpha_i})^T. \tag{13}$$

Once there are inter-robot loop closures between robot β and robot α, $T_{\beta\alpha}$ can be determined. $\mathbb{T}_{\beta\alpha} = \{\mathbf{T}^{(1)}_{\beta\alpha}, ..., \mathbf{T}^{(m)}_{\beta\alpha}\}$ is the set of estimations of $\mathbf{T}_{\beta\alpha}$ obtained from m inter-robot loop closures. Let us next assume the global frame $\underset{\rightarrow}{\mathcal{F}}_g$ is aligned with the local frame of the first robot $\underset{\rightarrow}{\mathcal{F}}_1$ (${}_g\mathbf{x}_\alpha = {}_1\mathbf{x}_\alpha$). Let $\mathbb{T}$ be the set of transformations from any robot frame to the global frame:

$$\mathbb{T} = \{\mathbf{T}_{g\alpha} \mid \forall \alpha \in \mathcal{N}, \alpha \neq g\}. \tag{14}$$

We aim to minimize the total transformation error between local robot frames with the Levenberg-Marquardt method:

$$\mathbb{T}^* = \underset{\mathbb{T}}{\arg\min} \sum_{\alpha,\beta \in \mathcal{N}} \mathbf{e}^T_{\beta\alpha} \mathbf{\Omega}_{\beta\alpha} \mathbf{e}_{\beta\alpha}. \tag{15}$$

$\mathbf{e}_{\beta\alpha}$ is the error of one transformation, and $\forall \mathbf{T}^{(i)}_{\beta\alpha} \in \mathbb{T}_{\beta\alpha}$:

$$\mathbf{e}_{\beta\alpha}(\mathbf{T}_{g\beta}, \mathbf{T}_{g\alpha}) = \mathbf{T}^{(i)}_{\beta\alpha} - \hat{\mathbf{T}}^{(i)}_{\beta\alpha}(\mathbf{T}_{g\beta}, \mathbf{T}_{g\alpha}). \tag{16}$$

Next, all inter-robot constraints are transformed to local robot frames, and the local graph optimization step is performed. Let us suppose there are inter-robot constraints between robot α and robot β. To perform the local optimization of robot α, we should transform the separator poses of robot β into local coordinates. Accordingly, $\forall \mathbf{x}_{\beta_j} \in \mathbf{X}_\beta \cup \mathbb{X}_\alpha$:

$${}_\alpha\mathbf{x}_{\beta_j} = \hat{\mathbf{T}}^T_{g\alpha} \cdot \hat{\mathbf{T}}_{g\beta} \cdot \mathbf{x}_{\beta_j}. \tag{17}$$

Finally, consider ${}_\alpha\mathbb{X}_\alpha$ as the initial value for separator poses. For any two sets of separator poses, $\langle \alpha_i, \beta_j\rangle$, $\langle \alpha_k, \gamma_l\rangle$, a virtual intra-robot loop closure ${}_\alpha\mathbf{z}_{\beta_j\gamma_l}$ can be computed by Eq. (21).

We perform pose graph optimization using the Levenberg-Marquardt method on Eq. (8) with a modified inter-robot term, $\mathbf{F}_{inter}$:

$$\mathbf{F}_{inter} = \sum_{\substack{\alpha,\beta,\gamma \in \mathcal{N}, \\ \alpha\neq\beta, \alpha\neq\gamma}} \sum_{\substack{\langle \alpha_i,\beta_j\rangle \in C_{\alpha\beta}, \\ \langle \alpha_k,\gamma_l\rangle \in C_{\alpha\gamma}}} \mathbf{e}^T_{vir} \mathbf{\Omega}_{vir} \mathbf{e}_{vir} \tag{18}$$

$$\mathbf{e}_{vir} = \mathbf{z}_{\alpha_i\beta_j} \cdot {}_\alpha\mathbf{z}_{\beta_j\gamma_l} \cdot (\mathbf{z}_{\alpha_k\gamma_l})^T - \hat{\mathbf{z}}_{\alpha_i\alpha_k}(\mathbf{x}_{\alpha_i}, \mathbf{x}_{\alpha_k}) \tag{19}$$

This two-stage global and local optimization ensures a high-quality initial guess for the local optimization step, which results in faster convergence. Furthermore, the introduction of local robot frames contributes to the numerical stability and consistency of the estimates and error covariances of the individual mobile robots. Thus, the SLAM systems of the local

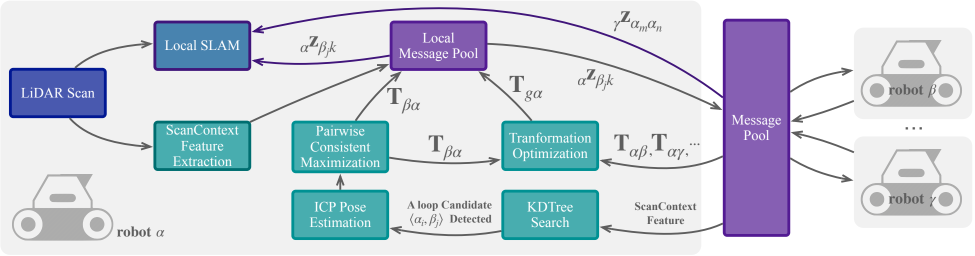

LiDAR Scan
Local SLAM
$_\alpha\mathbf{z}_{\beta_j k}$
Local Message Pool
$_\gamma\mathbf{z}_{\alpha_m\alpha_n}$
$_\alpha\mathbf{z}_{\beta_j k}$
ScanContext Feature Extraction
$\mathbf{T}_{\beta\alpha}$
$\mathbf{T}_{g\alpha}$
Pairwise Consistent Maximization
$\mathbf{T}_{\beta\alpha}$
Tranformation Optimization
$\mathbf{T}_{\alpha\beta}, \mathbf{T}_{\alpha\gamma}, \dots$
Message Pool
ICP Pose Estimation
A loop Candidate $\langle\alpha_i,\beta_j\rangle$ Detected
KDTree Search
ScanContext Feature
robot α
robot β
...
robot γ

Fig. 3: Overview of the architecture of DiSCo-SLAM, our proposed distributed multi-robot SLAM system, with robot α as the local robot.

robots are only dealing with minor numerical changes in poses due to odometry drifts, which is beneficial for applications in active SLAM, path planning, and exploration.

IV. Algorithms

An overview of our distributed multi-robot SLAM system is shown in Figures 2 and 3. Per the architecture shown in Fig. 3, once the system receives a LiDAR scan, the local SLAM thread and feature extraction thread are activated simultaneously. We adopt LIO-SAM [6] as our local SLAM framework, and alternatively use LeGO-LOAM [14] when tightly-coupled LiDAR-inertial data is unavailable. Scan Context [5], a lightweight spatial feature descriptor for 3D LiDAR, is used to describe and match features. Then, mobile robots exchange scan context features and perform scan matching. Once a potential inter-robot loop closure candidate is detected, incremental pairwise consistent measurement set maximization (PCM) [22] is performed to remove outliers, as in [3]. A two-stage optimization is performed by each robot, first establishing a global-to-local coordinate transformation, which informs a subsequent local pose graph optimization. Finally, coordinate transformations are exchanged among robots for further optimization.

A. Scan Context Feature Description and Matching

Scan Context (SC) [5] describes a LiDAR scan by projecting the scan onto a 2D plane, where the z-coordinate value of each 3D point is encoded in the intensity of the corresponding 2D point. The 2D scan image is then divided into grid cells according to a specified number of sectors N_s and rings N_r. For the Velodyne VLP-16 Lidar employed in our work, we use $N_s = 60$ and $N_r = 16$. The value of each grid cell I is the maximum intensity of all points captured in the cell. Finally, a ring key feature of dimension N_r is extracted by counting the non-zero values of each ring.

A ring key KD tree is then built for loop closure candidate search. All SC features of the top N matched ring key features are further compared to identify the best loop closure candidate. The SC features are shifted along the sector axis to ensure rotation invariance. The shifting angle also serves as an initial rotation guess for the ICP scan-matching process when there is no coordinate transformation history.

B. Incremental PCM

Unlike single-robot SLAM, for which incremental odometry measurements are frequently available to support an accurate initial guess, inter-robot loop closure candidates in multi-robot SLAM are only estimated by feature matching. Incremental pairwise consistent measurement set maximization (PCM) is introduced to avoid the acceptance of erroneous loop closures, which may result from different environmental regions with similar appearance, or objects in the environment arranged in repeating patterns.

Algorithm 1: Global-local optimization for robot α

```
Global: History of Separator Poses $\mathbf{X}_s$, Global Graph $_g$, Local Graph $_l$
Input: New Separator Pose Pair $\langle\mathbf{x}_{\alpha_i}, \mathbf{x}_{\beta_j}\rangle$, Scan
        Matching Measurement $\mathbf{z}_{\beta_j\alpha_i}$
Output: Optimized Coordinate Transformation $\mathbf{T}$,
         Optimized Poses $\mathbf{X}^*$
$\mathbf{T}_{\beta\alpha} \leftarrow \mathbf{z}_{\beta_j\alpha_i} \cdot {}_\beta\mathbf{x}_{\beta_j} \cdot ({}_\alpha\mathbf{x}_{\alpha_i})^T$
UpdateGraph($_g$, $\mathbf{T}_{\beta\alpha}$)
$\mathbf{T} \leftarrow$ OptimizeGraph($_g$)
# Calculate separator pose under local coordinate
$_\alpha\mathbf{X}_s \leftarrow$ TransformSeparatorPose($\mathbf{T}$, $\mathbf{X}_s$)
# Search for nearest inter-robot constraint
$_\alpha\mathbf{x}_k \leftarrow$ RadiusSearch($_\alpha\mathbf{X}_s$, $_\alpha\mathbf{x}_{\beta_j}$)
# Calculate virtual observation
$_\alpha\mathbf{z}_{\beta_j k} \leftarrow$ Between($_\alpha\mathbf{x}_{\beta_j}$, $_\alpha\mathbf{x}_k$)
UpdateGraph($_l$, $\mathbf{z}_{\beta_j\alpha_i}$, $_\alpha\mathbf{z}_{\beta_j k}$)
$\mathbf{X}^* \leftarrow$ OptimizeGraph($_l$)
$\mathbf{X}_s \leftarrow \mathbf{X}_s \cup \mathbf{x}_j$
return $\mathbf{X}^*$, $\mathbf{T}$
```

PCM [22] checks the consistency of inter-robot constraints. A loop closure is accepted if any two inter-robot constraints $\mathbf{z}_{\beta_j\alpha_i}$ and $\mathbf{z}_{\beta_l\alpha_k}$ meet the following condition:

$$\left\|(\mathbf{z}_{\beta_l\beta_j} \cdot \mathbf{z}_{\beta_j\alpha_i} \cdot \mathbf{z}_{\alpha_i\alpha_k}) \cdot \mathbf{z}_{\beta_l\alpha_k}{}^{-1}\right\|_2^2 < \epsilon. \tag{20}$$

$\mathbf{z}_{\beta_l\beta_j}$ is the intra-robot transformation of robot β between timestamps l and j; $\mathbf{z}_{\beta_j\alpha_i}$ is the inter-robot transformation relating timestamp j of robot β and timestamp i of robot α; ϵ is a small threshold (in our experiments to follow, we choose $\epsilon = 5$). To ensure robust real-time localization, we use a lazy initialization: incremental PCM will not be performed until there is a designated number of loop closure candidates. In our case, incremental PCM is performed after more than five loop closures are detected.

C. Two-Stage Global and Local Optimization

As mentioned in Sec. III, the robots in our proposed framework perform a two-stage global and local optimization. Alg. 1 presents the process of global and local optimization when a new inter-robot constraint pair is detected. In the global step, coordinate transformations between robots are treated as measurements. As shown in Eq. (15), only transformations from local robots to the global coordinate frame are optimized. The covariance matrices of these measurements are linearly related to the timestamp, since each robot's dead reckoning error grows as time accumulates.

After global optimization, all separator poses from other robots are transformed to the local coordinate frame according to the latest coordinate transformation matrices. Then, a euclidean distance based radius search is performed to find the nearest inter-robot constraint. During the radius search, separator poses whose timestamps are too close to the present timestamp are discarded to avoid optimizing an ill-posed graph. Present separator pose ${}_\alpha\mathbf{x}_{\beta_j}$ and nearest separator pose ${}_\alpha\mathbf{x}_k$ are converted to a virtual intra-robot loop closure:

$$ {}_\alpha\mathbf{z}_{\beta_j k} = {}_\alpha\mathbf{x}_{\beta_j}{}^{T} \cdot {}_\alpha\mathbf{x}_k . \tag{21} $$

Finally, the virtual observations are added to the local pose graph and the local pose graph is optimized.

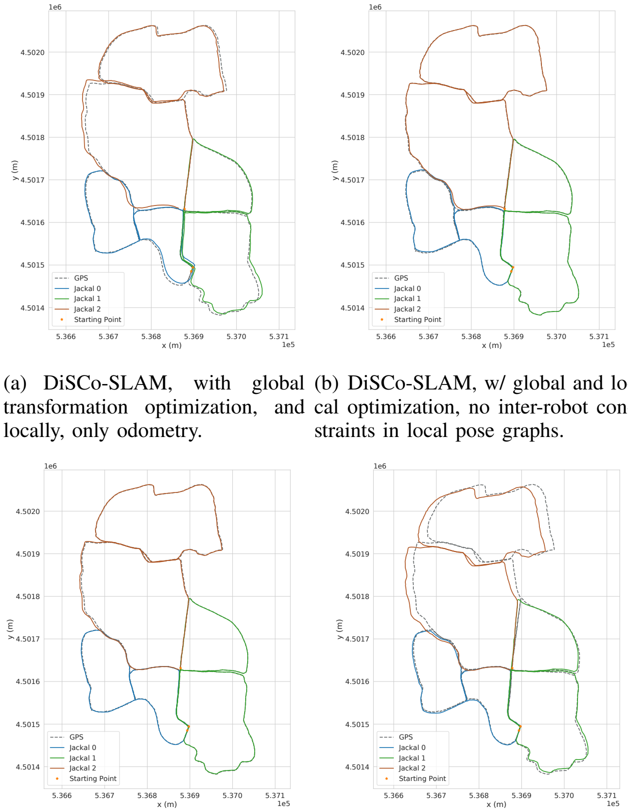
D. Message Passing Between Robots

When two robots rendezvous, they share their past SC features (which each robot stores in a last in, first out buffer) as well as coordinate transformations between themselves and other robots. The shared coordinate transformations are added to the global factor graph for each robot's global optimization step. Each recipient robot searches for neighbors of the shared SC features in its respective KD tree. If an SC feature match is found, the recipient robot will query its neighbor for a feature point cloud containing edge and planar features, and its corresponding pose in the neighbor robot's local coordinates. The feature point cloud is then used for scan matching. If an inter-robot loop closure is detected, the recipient robot will send the resulting transformation to the related robot when it is feasible to do so. Our use of SC permits data-efficient communication, even when there are no communication constraints and message-passing can occur at all times. An overview of sizes/quantities of messages passed in a practical use-case is given in Sec. V-D.

V. Experiments and Results

A. Experimental Setup

In the experiments to follow, we present results using four customized datasets configured for compatibility with 3D LiDAR-based, distributed multi-robot SLAM. We refer to the four datasets as (1) the KITTI 08 dataset, (2) the KITTI 00 dataset, (3) the Stevens dataset, and (4) the Park dataset. We adapt our first two datasets from the KITTI Vision Benchmark raw data sequences 08 and 00 [28]. In sequence 08, 100Hz raw inertial measurement unit (IMU) data is recorded with suitable temporal consistency to be used with LIO-SAM [6]. We use LeGO-LOAM [14] to run sequence 00 with LiDAR scans only. Since both LIO-SAM and LeGO-LOAM are configured to work with a 16-channel LiDAR, we downsample the KITTI LiDAR data from 64 beams to 16. We have modified sequences 08 and 00 into a synthetic two-robot dataset and a synthetic three-robot dataset respectively, where time-stamps have been adjusted to incorporate overlap and rendezvous.



(a) DiSCo-SLAM, with global transformation optimization, and locally, only odometry.

(b) DiSCo-SLAM, w/ global and local optimization, no inter-robot constraints in local pose graphs.

(c) DiSCo-SLAM as proposed, full global and local optimization.

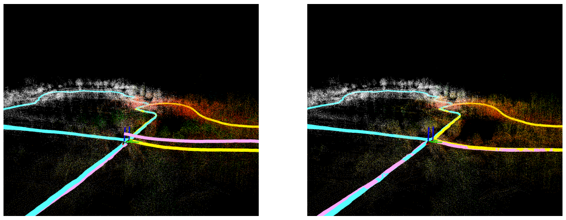
(d) DGS optimization with PCM.

Fig. 4: Representative trajectory estimation results over the three-robot Park dataset, with different multi-robot SLAM configurations.

The Stevens dataset is adapted from a dataset previously gathered on our campus [14] with a Clearpath Jackal UGV equipped with a Velodyne VLP-16 LiDAR, and only LiDAR data is used for LeGO-LOAM [14]. The dataset from this earlier paper has been modified into a two-robot dataset which includes overlap and rendezvous. Lacking RTK-GPS ground truth in this dataset, we use satellite imagery to qualitatively evaluate our experimental results.

Finally, the Park dataset was gathered specifically for this study, using a single Clearpath Jackal UGV, which is pictured in Fig. 1(a). Our UGV is equipped with a Velodyne VLP-16 LiDAR, a MicroStrain 3DM-GX5-25 IMU, and a Single-band RTK GNSS receiver, to both apply LIO-SAM [6] and evaluate it using RTK-GPS derived ground truth information. The data was collected in a suburban park environment. To generate a synthetic three-robot dataset from this single-robot mission, we rewrote the timestamps of each synthetic robot's trajectory to achieve a meaningful synchronization of three intersecting robot trajectories. Each robot executes a "figure-eight" trajectory comprised of two large loops; the individual robots will accumulate errors along these loops, but there is sufficient overlap among robots that inter-robot constraints can alleviate these errors. This dataset has been made freely available, along with our optimization framework[1]. DiSCo-SLAM's application to this dataset is shown in our video.

[1] https://github.com/RobustFieldAutonomyLab/DiSCo-SLAM



(a) No inter-robot constraints (as in Fig. 4(b)).

(b) With inter-robot constraints (as in Fig. 4(c)).

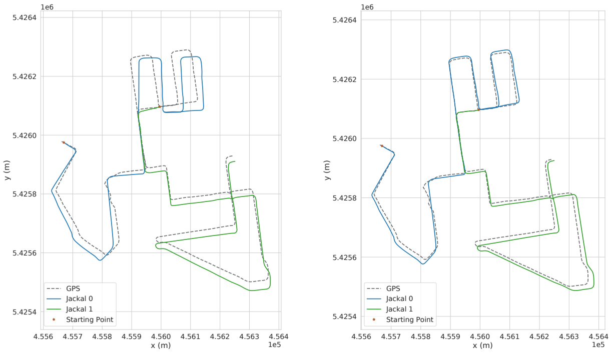
Fig. 5: Park dataset, at a location where there is overlap among all three robot trajectories. The "jackal 2" robot from the plots of Fig. 4 (pink) is completing a large loop with no intra-robot constraints.

In all four datasets, KITTI (08 and 00), Stevens, and Park, each robot trajectory includes intra-robot and inter-robot loop closures, and each robot encounters at least one rendezvous with every other robot. When GPS is available, it is used as ground truth for quantitative analysis, and not used for SLAM. We project the GPS measurements onto Universal Transverse Mercator (UTM) coordinates and perform a coordinate transformation to facilitate comparisons with our SLAM results. All experimental comparisons are performed using playback of previously gathered data on a desktop computer equipped with an Intel i9-9900K CPU, 62.7 GB memory using the robot operating system (ROS) in Ubuntu Linux 18.04. All robot threads run concurrently on the same processor, and do not utilize GPUs.

B. Performance of Two-Stage Optimization

In this section, we perform comparisons of multiple configurations of our proposed DiSCo-SLAM two-stage optimization framework, using the three-robot Park dataset. Displayed in Fig. 4 are multi-robot SLAM results for increasing levels of optimization. These comparisons are intended to display the efficacy of the proposed global and local optimization procedures in combination, versus their standalone performance. In Fig. 4(a), only the global optimization step is applied, and neither inter-robot nor intra-robot loop closure constraints are included in the local optimization step; it is produced using odometry only. Accordingly, overlap and rendezvous among robots fails to inform local pose-graph constraints, and thus results in a visible buildup of localization error in the result. The inclusion of intra-robot loop closures into the local optimization procedure, seen in Fig. 4(b), noticeably improves robot localization performance. Further improvements are apparent in Fig. 4(c), where inter-robot constraints are also added to each vehicle's local pose graph, including them in the local optimization step. A further inspection of the benefits of utilizing inter-robot constraints in local pose graphs can be seen in Fig 5. There, one can clearly note the higher consistency in pose estimates across overlapping robots, when leveraging these constraints.

Quantitative pose estimation error metrics for a representative SLAM execution trace over the Park dataset are listed in Table I. To compute these errors relative to ground truth information, GPS data is collected at a rate of 5Hz, while we generate one LiDAR keyframe per second. We match each keyframe pose with the nearest GPS pose according to their timestamps. The estimated trajectories are transformed from local SLAM coordinates into UTM coordinates using the starting points of the “jackal 1" and “jackal 2" trajectories as geometric constraints (the jackal 1 and jackal 2 trajectories are denoted in the plots of Fig. 4). The result in Table I with both intra- and inter- robot constraints added to the local pose graph achieves the highest accuracy.



(a) DGS optimization with PCM.

(b) DiSCo-SLAM, with full global and local optimization.

Fig. 6: Optimization result on the KITTI 08 dataset.

TABLE I: Root Mean Square Error (RMSE) w.r.t. GPS

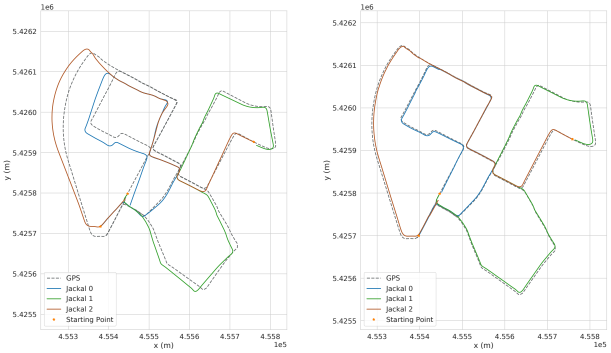
Dataset	Configure	X (m)	Y (m)	Total (m)
Park	DiSCo-Odometry	4.87	3.35	5.91
	DiSCo-Local	1.39	1.11	1.78
	Full DiSCo-SLAM	**1.31**	**0.52**	**1.43**
	DGS with PCM	11.84	5.38	13.00
KITTI 08	Full DiSCo-SLAM	**5.57**	**4.39**	**7.09**
	DGS with PCM	14.81	19.47	24.46
KITTI 00	Full DiSCo-SLAM	**3.48**	**5.61**	**6.60**
	DGS with PCM	14.54	14.78	20.73

C. Comparison with Distributed Gauss-Seidel (DGS)

Due to its data-efficiency and relevance to real-time multi-robot SLAM applications, we next compare our method against DGS optimization with PCM (summarized in Fig. 2 and Eqs. (10)-(11)), which comprises the back-end of DOOR-SLAM [3], a framework that has supported distributed multi-robot SLAM across different platforms and sensing modalities, including LiDAR. We test both DGS with PCM and DiSCo-SLAM with the same front-end on four datasets, (1) our modified KITTI 08 dataset, (2) our modified KITTI 00 dataset, (3) the Stevens campus dataset and (4) the Park dataset. The RMSE with respect to GPS is given in Table I.

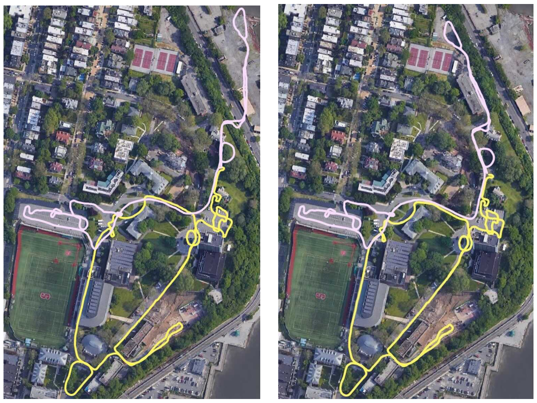
Fig. 6 shows representative results optimized by DGS over the KITTI 08 dataset. We transform the trajectories to align them with the GPS data according to the starting point of both robots, although the GPS data undergoes a small amount of erroneous drift in this dataset. The rotation angles are incorrectly estimated by DGS at several corners in Fig. 6(a) where turns occur, while there are no significant errors in the result of DiSCo-SLAM when turning corners in Fig. 6(b).

The KITTI 00 dataset is challenging since only LiDAR scans are used, and the LiDAR frame spacing is larger than our VLP-16 datasets. Both DiSCo-SLAM and DGS with PCM achieve low accuracy since the LiDAR frame rate is low, and thus fewer inter-robot loop closures are detected. The



(a) DGS optimization with PCM. (b) DiSCo-SLAM, with full global and local optimization.

Fig. 7: Optimization result on the KITTI 00 dataset.



(a) DGS optimization w/ PCM. (b) DiSCo-SLAM, with full global and local optimization.

Fig. 8: Optimization result on the Stevens campus dataset.

optimized robot trajectories of DGS with PCM (Fig. 7(a)) align well where there are inter-robot loop closures. However, their rotation estimation falls into local minima and introduces errors. Fig. 7(b) shows the result for DiSCo-SLAM. Since drift accumulates locally due to a lack of intra-robot loop closures, we lower the threshold for PCM and model the covariances of inter-robot loop closure measurements as Cauchy distributions. Although the resulting trajectory aligns with GPS well for most parts, errors occur along the z-axis of "jackal 1" at the ending point due to the lack of intra-robot loop closures.

For the Stevens dataset, GPS measurements along the path are not available, so we transform the trajectory estimates into UTM coordinates and project them onto satellite imagery. Fig. 8 shows the optimized trajectories using DiSCo-SLAM (Fig. 8(b)), and DGS with PCM (Fig. 8(a)). The yellow robot trajectory, aligned with the global frame, is well-optimized in both methods. DGS's estimate of the pink robot trajectory drifts as time accumulates, while this robot's trajectory estimate in our DiSCo-SLAM method aligns with the roadways depicted in the imagery.

The Park dataset was gathered with GPS signal available throughout, so we compared our method and the DGS method using the GPS data as ground truth. Figs. 4(d) and 4(c) show results from the Park dataset using DGS optimization and our method, respectively. For this dataset, we run multiple trials to examine the robustness of our method. After each trial, the SLAM trajectory estimates are compared against the GPS data, following the same procedure described in Sec. V-B. Table III shows the RMSE with respect to GPS ground truth data, across 40 trials of re-playing the same recorded dataset, using both methods. Although the lowest error in all the trials for DGS and our method is close, our method offers a more stable, consistent output. The DGS method's rotation optimization step often hinders convergence to a global minimum under the infrequent arrival of inter-robot constraints, as evidenced by the buildup of drift for "jackal 2" in Fig. 4(d).

Because our RTK-GPS data only covers two translational degrees of freedom, we also compare relative pose estimation error. Since all of our multi-robot datasets are obtained by dividing single-robot datasets into several parts, we use the coincidence of the ending point of one robot's trajectory and the starting point of the next as the basis for quantifying the rotational and translational errors across a representative inter-robot "rendezvous point" from each dataset, which are captured in Table II for all of our datasets. Although DiSCo-SLAM is not always superior, its worst-case performance is well below the levels occasionally reached by DGS.

TABLE II: Relative pose estimation error at traj. connecting points

Dataset	**Configure**	**Roll**	**Pitch**	**Yaw**	**Total (°)**	**X**	**Y**	**Z**	**Total (m)**
Park	DiSCo-SLAM	0.30	0.50	0.60	0.83	0.52	0.95	1.36	1.74
	DGS with PCM	**0.12**	**0.01**	**0.22**	**0.25**	**0.16**	**0.04**	**0.03**	**0.16**
	DiSCo-SLAM	0.85	**2.56**	**1.75**	**3.21**	**0.28**	**0.16**	**0.23**	**0.40**
	DGS with PCM	**0.55**	3.40	3.64	5.01	1.61	1.32	0.45	2.13
KITTI08	DiSCo-SLAM	**1.09**	**0.99**	**0.45**	**1.54**	**3.60**	**4.28**	**0.27**	**5.6**
	DGS with PCM	1.79	1.25	14.03	14.19	22.58	7.82	0.37	23.90
KITTI00	DiSCo-SLAM	**0.38**	2.99	**1.10**	**3.20**	**6.38**	**1.94**	**1.21**	**6.78**
	DGS with PCM	11.38	**1.57**	14.84	18.76	14.92	26.35	14.15	33.42
	DiSCo-SLAM	0.56	7.43	**0.62**	7.48	2.08	**6.74**	14.00	15.67
	DGS with PCM	**0.20**	**0.40**	0.75	**0.87**	**0.67**	11.35	**1.49**	**11.47**
Stevens	DiSCo-SLAM	3.22	5.07	**0.26**	6.01	**0.32**	**0.99**	**0.23**	**1.07**
	DGS with PCM	**1.61**	**3.63**	4.28	**5.84**	0.50	1.28	0.99	1.70

TABLE III: RMSE w.r.t GPS for 40 trials

RMSE	Min (m)	Max (m)	Mean (m)	STD (m)
DiSCo-SLAM	**1.01**	**2.51**	**1.52**	**0.34**
DGS with PCM	2.87	14.37	6.23	2.72

D. *Communication and Computational Efficiency*

To quantify the bandwidth requirements of the proposed SLAM framework, we have examined the sizes of the messages sent between robots during execution of the datasets. The results for Velodyne VLP-16 and VLP-64 LiDAR are shown in Tables IV and V respectively, which catalog the mean, minimum, and maximum size of each type of message exchanged, as well as the total quantity of each type of message exchanged, between robots during their execution of the trajectories. We assume there is no maximum communication range, so that messages can be exchanged between robots at any time. While a single laser scan from the Velodyne VLP-16 is 1.04 MB, the message size needed for our DiSCo-SLAM method for each LiDAR keyframe is around 200 KB.

Table VI shows the computation time of each key step of DiSCo-SLAM, using the computer described in Sec. V.A. The groupings of rows correspond to the groupings given in Tables IV and V (i.e., the algorithmic steps in one grouping yield the messages in the other). Feature description and matching are, by far, the most frequently performed steps, serving as an efficient filtering mechanism that permits costly point cloud matching to be invoked less frequently.

TABLE IV: Data Sizes of Messages Sent (VLP-16)

Message Info	**Mean (kB)**	**Min (kB)**	**Max (kB)**	**No. Total Msgs.**	
				Stevens	Park
SC Feature & Local Pose	4.08	4.04	4.12	3936	4222
Feature Cloud (Edge)	9.65	5.08	15.90	37	837
Feature Cloud (Planar)	71.31	54.74	85.98	37	837
Feature Cloud (Other)	70.91	50.58	83.99	37	837
Coordinate Transformation	0.70	0.70	0.70	3	711
Inter-Robot Loop Closure	0.12	0.12	0.12	3	72

TABLE V: Data Sizes of Messages Sent (VLP-64)

Message Info	**Mean (kB)**	**Min (kB)**	**Max (kB)**	**No. Total Msgs.**	
				KITTI00	KITTI08
SC Feature & Local Pose	15.75	15.75	15.75	1134	2333
Feature Cloud (Edge)	30.60	16.76	42.62	198	12
Feature Cloud (Planar)	309.70	242.39	383.03	198	12
Feature Cloud (Other)	89.18	69.46	115.96	198	12
Coordinate Transformation	0.70	0.70	0.70	130	8
Inter-Robot Loop Closure	0.12	0.12	0.12	24	7

TABLE VI: Processing Time for Each Algorithmic Step (ms)

Subroutine	**Park**		**KITTI08**		**KITTI00**		**Stevens**	
	Mean	Max	Mean	Max	Mean	Max	Mean	Max
SC Feature Description	<1	9	<1	6	<1	6	<1	10
SC Feature Matching	61	188	34	67	2	68	24	90
Cloud Scan Matching	193	614	112	147	19	147	314	677
Incremental PCM	55	346	7	10	<1	10	5	20
Global Optimization	4	12	<1	1	<1	1	<1	<1
Local Optimization	7	36	2	14	<1	14	7	20

VI. Conclusions

In this paper we have presented DiSCo-SLAM, a distributed multi-robot SLAM framework for 3D LiDAR observations, which requires a relatively low communication bandwidth for message passing. In DiSCo-SLAM, LiDAR scans are efficiently described using Scan Context descriptors and shared between robots. We also propose a two-stage global-local graph optimization procedure that offers robust output for relatively large scale multi-robot SLAM problems with limited occurrences of rendezvous, finding transformations relating robots that may be distant from one another. We compare our optimization strategy with the widely used distributed Gauss-Seidel method, showing the relative stability of our method.

References

[1] R. Dubé, A. Gawel, H. Sommer, J. Nieto, R. Siegwart and C. Cadena, "An online multi-robot SLAM system for 3D lidars," *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Syst.*, 1004–1011, 2017.

[2] S. Choudhary, L. Carlone, C. Nieto, J. Rogers, H. Christensen, and F. Dellaert, "Distributed trajectory estimation with privacy and communication constraints: a two-stage distributed Gauss-Seidel approach," *Proc. IEEE Int. Conf. Robotics Automation*, 5261–5268, 2016.

[3] P. 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 Lett.*, 5(2):1656–1663, 2020.

[4] F. Dellaert, "Factor Graphs and GTSAM: A Hands-on Introduction," Georgia Institute of Technology, Tech. Rep. No. GT-RIM-CP&R-2012-002, 2012.

[5] G. Kim and A. Kim, "Scan Context: Egocentric spatial descriptor for place recognition within 3D point cloud map," *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Syst.*, 4802–4809, 2018.

[6] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti and D. Rus, "LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping," *Prof. IEEE/RSJ Int. Conf. Intelligent Robots Syst.*, 5135-5142, 2020.

[7] S. Saeedi, M. Trentini, M. Seto, H. Li "Multiple-robot simultaneous localization and mapping: A review" *J. Field Robotics*, 33:3-46, 2016.

[8] M. Kegeleirs, G. Grisetti, M. Birattari, "Swarm SLAM: Challenges and Perspectives," *Frontiers in Robotics and AI*, 8:23, 2021.

[9] L. Riazuelo, J. Civera and J. Montiel, "C2tam: A cloud framework for cooperative tracking and mapping," *Robotics and Autonomous Syst.*, 62(4):401–413, 2014.

[10] I. Deutsch, M. Liu and R. Siegwart, "A framework for multi-robot pose graph SLAM," *Proc. IEEE Int. Conf. Real-time Computing Robotics*, 567–572, 2016.

[11] M. Karrer, P. Schmuck and M. Chli, "CVI-SLAM — Collaborative visual-inertial SLAM," *IEEE Robotics and Automation Lett.*, 3(4):2762–2769, 2018.

[12] P. Zhang, H. Pengfei, B. Ding and S. Shang, "Cloud-based framework for scalable and real-time multi-robot SLAM," *Proc. IEEE Int. Conf. Web Services*, 147–154, 2018.

[13] Y. Chang, Y. Tian, J. How and L. Carlone, "Kimera-Multi: a System for Distributed Multi-Robot Metric-Semantic Simultaneous Localization and Mapping," *Proc. IEEE Int. Conf. Robotics Automation*, 11210–11218, 2021.

[14] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain," *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Syst.*, 4758-4765, 2018.

[15] H. Ye, Y. Chen, M. Liu "Tightly coupled 3d lidar inertial odometry and mapping" *Proc. IEEE Int. Conf. Robotics Automation*, 3144–3150, 2019.

[16] R. Dubé, A. Cramariuc, D. Dugas, J. Nieto, R. Siegwart and C. Cadena "SegMap: 3d segment mapping using data-driven descriptors" *Proc. Robotics: Science and Syst.*, 2018.

[17] L. Di Giammarino, I. Aloise, C. Stachniss and G. Grisetti "Visual Place Recognition using LiDAR Intensity Information" *arXiv preprint arXiv:2103.09605*, 2021.

[18] K. Ebadi, M. Palieri, S. Wood, C. Padgett, A. Agha-mohammadi "DARE-SLAM: Degeneracy-Aware and Resilient Loop Closing in Perceptually-Degraded Environments" *J. Intelligent & Robotic Syst.*, 102, 2021.

[19] M. Lazaro, L. Paz, P. Pinies, J. Castellanos and G. Grisetti, "Multi-robot SLAM using condensed measurements," *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Syst.*, 1069–1076, 2013.

[20] W. Wang, N. Jadhav, P. Vohs, N. Hughes, M. Mazumder and S. Gil, "Active rendezvous for multi-robot pose graph optimization using sensing over Wi-Fi," *arXiv preprint arXiv:1907.05538*, 2019.

[21] P. Agarwal, G. Tipaldi, L. Spinello, C. Stachniss and W. Burgard, "Robust map optimization using dynamic covariance scaling," *Proc. IEEE Int. Conf. Robotics Automation*, 62–69, 2013.

[22] J. Mangelson, D. Dominic, R. Eustice and R. Vasudevan, "Pairwise consistent measurement set maximization for robust multi-robot map merging," *Proc. IEEE Int. Conf. Robotics Automation*, 2916–2923, 2018.

[23] A. Cunningham, M. Paluri and F. Dellaert, "DDF-SAM: Fully distributed SLAM using constrained factor graphs," *Proc. IEEE/RSJ Int. Conf. Intelligent Robots Syst.*, 3025–3030, 2010.

[24] A. Cunningham, V. Indelman and F. Dellaert, "DDF-SAM 2.0: Consistent distributed smoothing and mapping," *Proc. IEEE Int. Conf. Robotics Automation*, 5220–5227, 2013.

[25] Y. Tian, K. Khosoussi, D. Rosen and J. How, "Distributed certifiably correct pose-graph optimization," *IEEE Trans. Robotics*, doi: 10.1109/TRO.2021.3072346, 2021.

[26] D. Rosen, L. Carlone, A. Bandeira and J. Leonard, "SE-Sync: A Certifiably Correct Algorithm for Synchronization over the Special Euclidean Group," *Int. J. Robotics Res.*, 38(2-3):95–125, 2019.

[27] G. Grisetti, R. Kümmerle, C. Stachniss and W. Burgard "A tutorial on graph-based SLAM," *IEEE Intelligent Transportation Syst. Mag.*, 2(4):31–43, 2010.

[28] A. Geiger, P. Lenz, C. Stiller and R. Urtasun, "Vision meets Robotics: The KITTI Dataset," *Int. J. Robotics Res.*, 32(11):1231-1237, 2013.