Centralized Multi-robot Collaborative LiDAR SLAM Utilizing Loop Closure Selection

1st Ensong Liu
School of Electronics
Peking University
Beijing, China
ensongliu@pku.edu.cn

2nd Haozhen Li
School of Electronics
Peking University
Beijing, China
pkuimlhz@pku.edu.cn

3rd Sijiang Li
School of Electronics
Peking University
Beijing, China
pkulsj@pku.edu.cn

4th Xiang Cheng
School of Electronics
Peking University
Beijing, China
xiangcheng@pku.edu.cn

***Abstract*—Multi-robot systems have attracted much attention due to their high efficiency and robustness. Multi-robot Simultaneous Localization and Mapping (SLAM) is the cornerstone of collaboration, such that each robot can localize and build a map of the workspace. One of the key challenges of this problem lies in utilizing the shared information between robots adequately so that the system can retain the correct information and reject the outliers. Here we propose a centralized multi-robot collaborative LiDAR SLAM framework for robots, each only equipped with a 2D LiDAR without the need for additional sensors. With each robot able to run simple LiDAR odometry onboard, a central server with greater computational capacity collects the information from robots, optimizes their poses globally, and updates their poses through communication. For a large number of loop closures in multi-robot SLAM systems, it is catastrophic to build wrong data associations between pose nodes in the global factor graph. We resort to a loop closure selection method to ensure the correctness and consistency of global optimization. Throughout the verification process on the real Automated Mobile Robot (AMR) platform, the framework exhibits better performance on localization and mapping.**

***Index Terms*—Multi-robot System, Collaborative Localization, SLAM, Multi-robot SLAM**

I. Introduction

With the fleets of robotics and artificial intelligence, the application of robots is becoming increasingly widespread. Simultaneous Localization and Mapping (SLAM) is the fundamental of other upper-level intelligent techniques, that the robot localizes in and builds a map of the environment simultaneously. Mainstream SLAM technology can be categorized into Visual SLAM, Laser SLAM, and multi-sensor fusion SLAM, which utilize cameras, Light Detection And Ranging (LiDAR), or both respectively. Due to the complexity of real-world environments and the diversity of tasks, current SLAM technology focuses more on robustness, which is called “the Robust-Perception Age” [1].

Multi-robot SLAM (MR-SLAM) has the advantage of faster task execution and greater robustness when facing single-robot failure. Moreover, in the era of ubiquitous intelligence, multi-robot collaboration is increasingly important in realizing the full potential of intelligent systems. While single-robot SLAM is challenging enough, moving it to a multi-robot platform adds a new level of challenge, including multi-robot loop closure, task allocation, communication mechanism design and so on [2]. With multiple robots exploring more efficiently, it is more possible for one robot to visit where the robots have visited before, which is called loop closure, including the inter-robot and the intra-robot. However, not all of these data associations are beneficial to the optimization because the registration may produce wrong pose transformation results due to the complexity of real-world environments.

With this motivation, in this study, we propose a multi-robot collaborative SLAM framework based on 2D LiDAR which can use several robots’ sensors to optimize the poses globally and construct a comprehensive map of the operating environment. Our key contribution is a flexible, scalable, and robust centralized system for multi-robot collaborative LiDAR SLAM. To achieve this we

- Develop a multi-robot back-end based on Hector SLAM, that consists of keyframe management, loop closure detection, and global optimization.
- Propose a loop closure selection module to avoid incorrect associations between poses in graph optimization, so as to maximize the capability of the multi-robot networked communication.
- Perform experiments on the real AMR to demonstrate the performance of the proposed framework, evaluating the localization accuracy and mapping quality in a complex and challenging corridor environment.

II. Related Work

LiDAR has the advantage of high resolution and a wide field of view, including 2D LiDAR and 3D LiDAR. Although 3D LiDAR with multiple laser lines has richer and denser structure information, 2D LiDAR is widely applied in indoor environments due to its flexibility and low cost. The core idea of LiDAR SLAM is scan matching, which can be categorized into scan-to-scan [3] [4], scan-to-map [5], and pixel-accurate matching [6], with higher computation consumption and lower accumulation of local error correspondingly. We select a 2D LiDAR scan-to-map method, Hector SLAM [5], as our fundamental single-robot SLAM algorithm for its high flexibility and independence from reliance on additional sensors.

This work was supported in part by the National Key R&D Program of China under Grant No. 2021ZD0112700.

2023 China Automation Congress (CAC) | 979-8-3503-0375-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/CAC59555.2023.10450881

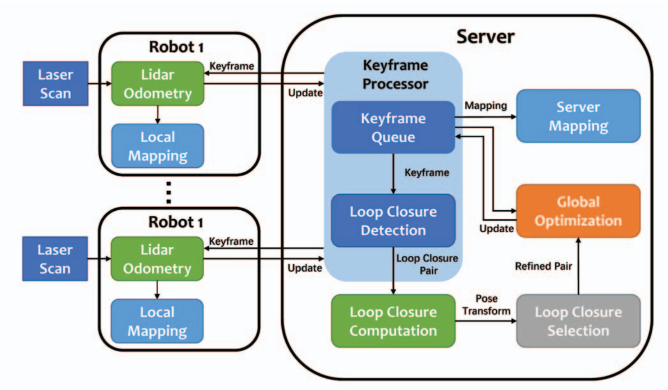


Fig. 1: Overview of the system architecture. The multi-robot system can be divided into robots and the server. The robot processes laser scan onboard and generates LiDAR odometry keyframes. The server collects keyframes from multiple robots, computes loop closure transformations, and performs global optimization.

Multi-robot SLAM is an extension of regular single-robot SLAM exploiting several collaborating robots to map the unknown environment. For multi-robot SLAM, it is essential to design the system architecture, which can be centralized or distributed. A centralized system has the advantage of collecting data from all the robots and processing them in an efficient and consistent way [7] [8]. Instead, in a distributed manner, the robots exchange information directly amongst each other and the system can scale better to large numbers of robots [9] [10]. For the benefit of processing data effectively, we design a centralized multi-robot architecture, where a server collects information from all the robots and optimizes the global cost function.

Meanwhile, for more data associations introduced by multiple robots, it is essential to decide on one association to retain or reject [11] [12]. Different loop closure candidates would make different influences on SLAM based on the inherent graph property, sensor data redundancy and other pertinent factors [11]. The loop closure transformations are obtained by feature matching between two images or pointclouds, e.g. pointcloud registration. The wrong transformations are catastrophic for the optimization, so we propose the consistency restrictions to select the effective ones.

Compared with current MR-SLAM systems mainly on 3D LiDAR or vision, 2D LiDAR MR-SLAM systems mostly focus on building maps efficiently [13], rather than improving localization accuracy ,mapping quality and system robustness through collaboration. This work proposes a 2D LiDAR MR-SLAM system, which is centralized to process redundant data efficiently and utilizes loop closure selection to improve robustness and accuracy.

III. Approach

A. System Overview

The centralized multi-robot SLAM system overview is as Fig. 1 implemented on Robot Operating System (ROS), consists of a real-time single-robot front-end running on the robot, generating keyframes at regular intervals, and a multi-robot back-end on the server, collecting multiple robots' data and processing data globally. The keyframe processor gathers information from different robots and produces potential loop closure pairs to the pointcloud registration module. The pose transformations computed by registration are checked by the loop closure selection module so correct transformations are retained to fuel the optimization and wrong transformations are rejected to avoid influencing the system. The global factor graph collects LiDAR odometry factors from the keyframe queue and refined loop closure factors from the loop closure selection, optimizes multi-robot data associations globally, and updates the result of every robot.

B. LiDAR Odometry

We choose Hector SLAM as the fundamental localization method in our multi-robot SLAM system. Hector SLAM is a flexible and scalable LiDAR odometry based on 2D LiDAR, which utilizes the scan-to-map method to localize in the environment. The main idea is to seek the rigid transformation $\boldsymbol{\xi} = (p_x, p_y, \psi)$ that maximizes the alignment of the LiDAR pointcloud with the occupancy grid map (1)

$$\boldsymbol{\xi}^* = \arg\min_{\xi} \sum_{i=1}^{n} [1 - M(\boldsymbol{S}_i(\boldsymbol{\xi}))]^2 \tag{1}$$

Here, $\boldsymbol{S}_i(\boldsymbol{\xi})$ are the world coordinates of scan endpoint $\boldsymbol{s}_i = (s_{i,x}, s_{i,y})^T$, which transform the endpoint in the LiDAR coordinate according to the estimated pose. The function $M(\boldsymbol{S}_i(\boldsymbol{\xi}))$ returns the map value at the coordinate given by $\boldsymbol{S}_i(\boldsymbol{\xi})$, representing the occupation probability.

Given some starting estimate of $\boldsymbol{\xi}$, e.g. the estimated transform last timestamp, we aim to estimate $\Delta\boldsymbol{\xi}$ which optimizes the error measure according to

$$\sum_{i=1}^{n} [1 - M(\boldsymbol{S}_i(\boldsymbol{\xi} + \Delta\boldsymbol{\xi}))]^2 \to 0 \tag{2}$$

This equation can be further transformed by first-order Taylor expansion, then solved by the Gauss-Newton method finally. Additionally, Hector SLAM utilizes the bilinear interpolation method to allow continuous and differentiable representation. To overcome the optimization stuck in local optimal, a multi-resolution map representation scheme is employed, which is similar to image pyramid approaches used in computer vision. However, without the loop closure detection and back-end module, this method still suffers from accumulated error with the robot running for a long time or in a large-scale environment.

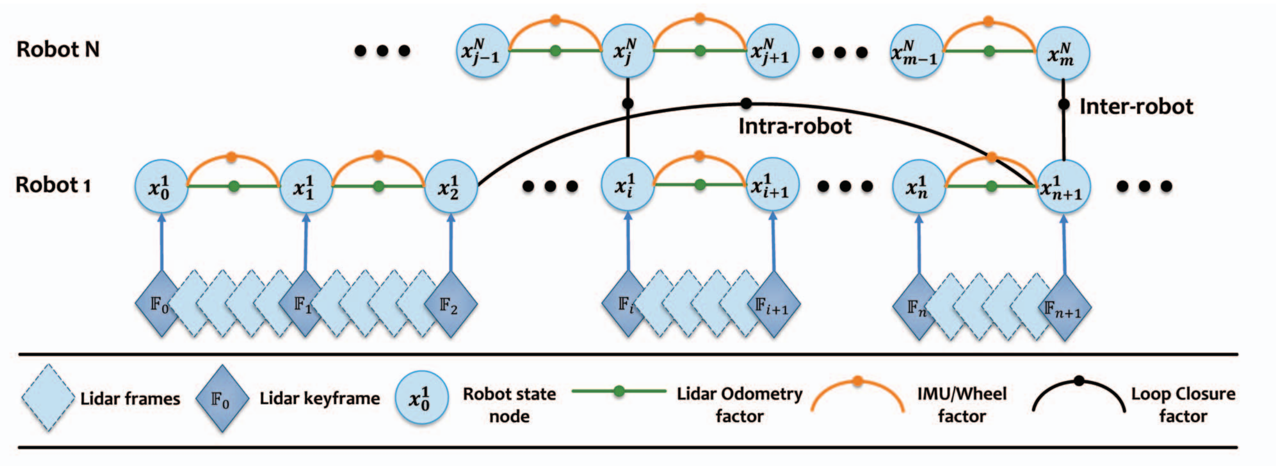


Fig. 2: Multi-robot SLAM Factor Graph Model. The system input receives a 2D laser scan, an IMU, and a wheel odometry. Three types of factors are introduced to the constructed factor graph: (a) LiDAR odometry factor, (b) wheel odometry/IMU integration factor, and (c) loop closure factor. The generation of these factors is discussed in Section III.

C. Keyframe Processor

Based on Hector SLAM above, we develop a multi-robot back-end module to eliminate accumulated error. In order to improve the efficiency of data processing, the estimated poses are only saved as keyframes if the robot occurs a significant movement over a predefined threshold. The keyframe format is defined as a message in ROS, including the robot's ID, key scan, and key pose. The processor collects multiple robots' keyframes and inserts them into the global pose graph. Specifically, keyframes generated from different robots are defined as below, for the robot n

$$\mathbf{X^n} = \{\boldsymbol{x}_0^n, ..., \boldsymbol{x}_t^n\}, \quad n \in \{1, 2, ..., N\} \tag{3}$$

where N is the number of robots. They are reorganized into the global keyframe queue $\mathbf{X}^{\mathcal{G}}$ according to their timestamps, recording their source robots as $\boldsymbol{\mathcal{I}}^{\mathcal{G}}$ at the same time

$$\mathbf{X}^{\mathcal{G}} = \{\boldsymbol{x}_0^{\mathcal{G}}, ..., \boldsymbol{x}_t^{\mathcal{G}}\}, \quad \boldsymbol{\mathcal{I}}^{\mathcal{G}} = \{id(\boldsymbol{x}_0^{\mathcal{G}}), ..., id(\boldsymbol{x}_t^{\mathcal{G}})\} \tag{4}$$

The function id returns the source robot of the input keyframe, e.g. $id(\boldsymbol{x}_t^n) = n$. The notation superscript, global frame $\boldsymbol{\mathcal{G}}$, is omitted for simplification in the following sections.

Then the processor generates potential loop closure pairs based on Euclidean distance between nodes. This approach generates candidates from nodes that lie within a certain distance d from the most recent node. Meanwhile, the nodes are required to be distant in timestamp, intra-robot loop closure, or be from different robots, inter-robot loop closure. Under these conditions, it is possible to build more loop closure associations and eliminate accumulated error through long-term operation. When a new state node $\boldsymbol{x}_{i+1}$ is added to the factor graph, we first search the graph and find the prior nodes that are close to $\boldsymbol{x}_{i+1}$ in Euclidean space or are not from the same robot of $\boldsymbol{x}_{i+1}$. The returned nodes are computed to get the loop closure transformations and further checked. In this paper, the search distance for loop closure is set to be $4m$ from a new state $\boldsymbol{x}_{i+1}$ based on the LiDAR parameteters.

D. Loop Closure Computation

While there are numerous ways to perform pointcloud registration, such as classical methods based on geometry [14] [15] or novel methods using deep learning [16]. For 2D laser scan in this paper, we choose classical registration methods, Normal Distribution Transform (NDT) and Iterative Closest Point (ICP), as a two-stage loop closure computation module due to their flexibility and efficiency. ICP has the advantage of high precision at the point-to-point level but optimization results highly depend on the initial value given. Instead, NDT could still work out without a favorable initial estimate due to its probability representation. Therefore, for loop closure computation, we first evaluate transformations produced by NDT, utilizing fitness-score of the algorithm in PCL (Point Cloud Library). The lower the fitness-score, the better the registration effect. If the result is good enough, it would be used as the initial estimate for ICP. Otherwise, ICP uses pose transformations between keyframes as the initial value. Throughout this two-stage registration, we can take full advantage of the benefits of NDT and ICP such that improve the effectiveness of loop closure computation.

Algorithm 1 Two-stage Loop Closure Computation

Input: Pointcloud $\boldsymbol{P}_i, \boldsymbol{P}_j$ of timestamp i, j,
Odometry Pose $\boldsymbol{x}_i, \boldsymbol{x}_j$ of timestamp i, j

Output: Transformation $\boldsymbol{z}_{ij}$, Fitness-score f

$\boldsymbol{z}_{ij}^{NDT}, f_{ij}^{NDT} \leftarrow NDT(\boldsymbol{P}_i, \boldsymbol{P}_j, \boldsymbol{I})$

if $f_{ij}^{NDT} < f_{threshold}^{NDT}$ **then**

$\quad \boldsymbol{z}_{ij}, f_{ij} \leftarrow ICP(\boldsymbol{P}_i, \boldsymbol{P}_j, \boldsymbol{z}_{ij}^{NDT})$

else

$\quad \boldsymbol{z}_{ij}, f_{ij} \leftarrow ICP(\boldsymbol{P}_i, \boldsymbol{P}_j, \boldsymbol{x}_i \boldsymbol{x}_j^{-1})$

end if

E. Loop Closure Selection

Although registration algorithms could give fine alignment between pointclouds, wrong transformations may still occur due to the limitations of registration methods and the complexity of real-world environments, especially for 2D LiDAR. It would be catastrophic for global optimization if wrong loop closure is added, even worse than only odometry without a back-end. Therefore, for a large number of inter-robot and intra-robot constraints, a loop closure is accepted to participate in optimization if any constraint $\boldsymbol{z}_{ij}$ and corresponding fitness f_{ij} meet the following condition:

$$f_{ij} < \alpha,\ \left\|\boldsymbol{z}_{ij}\boldsymbol{x}_j - \boldsymbol{x}_i\right\|_2 < \begin{cases} |j-i|\boldsymbol{\epsilon}, & \boldsymbol{\mathcal{I}}(i) = \boldsymbol{\mathcal{I}}(j) \\ \boldsymbol{\delta}, & \boldsymbol{\mathcal{I}}(i) \neq \boldsymbol{\mathcal{I}}(j) \end{cases} \tag{5}$$

$\boldsymbol{z}_{ij}$ is the transformation from timestamp j to timestamp i; $\boldsymbol{x}_i$ is the keyframe pose $\in$ SE(2) at timestamp i; $\boldsymbol{\epsilon}$ and $\boldsymbol{\delta}$ are threshold vectors contaning three directions (x, y, ψ) for intra-robot case and inter-robot case respectively. This condition ensures optimization only retains the transformations whose fitness-score is under the pre-defined threshold α and consistency check is passed. The consistency check means that loop closure transformations should be consistent with odometry poses within a certain range. For the intra-robot case, the larger the interval between two nodes, the greater the range of corrections allowed by loop closure detection. This is because accumulated error is the cumulative result of tiny errors in the front-end pose estimation, rather than estimation failures. We assume accumulated error grows linearly with time and set a unit accumulated error vector $\boldsymbol{\epsilon}$, together deciding the correction threshold, $|j-i|\boldsymbol{\epsilon}$. The parameter $\boldsymbol{\epsilon}$ value is related to the distance d between keyframes, which are set $(0.008, 0.008, 0.001)$ and $1.5m$. For the inter-robot case, we set the constraint $\boldsymbol{\delta}$ as a fixed threshold $(1.5, 1.5, 0.3)$.

F. Global Optimization

The global optimization builds a factor graph of multiple robots' information as Fig. 2, that edges represent the pose constraints and nodes represent the estimated poses. The constraints include laser odometry factors, wheel odometry integration factors, and inter-robot/intra-robot loop closure factors. This centralized factor graph can represent multiple robots' information consistently and efficiently, so we can process each robot's information equally.

Let $\mathbf{X} = \{\boldsymbol{x}_0, ..., \boldsymbol{x}_t\}$ be a set of 3DoF robot poses from multiple robots, with $\mathbf{X} \subset$ SE(2). C contains all inter-robot and intra-robot constraints between robot poses. For each pair of poses defining a constraint $\langle i, j\rangle \in C$, we define error $\boldsymbol{e}_{ij}$ between observed transformation $\boldsymbol{z}_{ij} \in$ SE(2) and expected transformation $\hat{\boldsymbol{z}}_{ij}$ as:

$$\boldsymbol{e}_{ij}(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{z}_{ij}) = \boldsymbol{z}_{ij} - \hat{\boldsymbol{z}}_{ij}(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{6}$$

$$\hat{\boldsymbol{z}}_{ij}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^{-1}\boldsymbol{x}_j \tag{7}$$

(a) Our AMR platform (b) 2D LiDAR

(c) Real-world Environment

Fig. 3: Experiment platform and real-world environment.

For LiDAR odometry constraints, we define error e_i between odometry transformation $\Delta\boldsymbol{\xi}_i \in$ SE(2) and expected transformation $\hat{\boldsymbol{z}}_i$

$$\boldsymbol{e}_i(\boldsymbol{x}_{i-1}, \boldsymbol{x}_i, \boldsymbol{\xi}_i) = \Delta\boldsymbol{\xi}_i - \hat{\boldsymbol{z}}_i(\boldsymbol{x}_{i-1}, \boldsymbol{x}_i) \tag{8}$$

$$\hat{\boldsymbol{z}}_i(\boldsymbol{x}_{i-1}, \boldsymbol{x}_i) = \boldsymbol{x}_{i-1}^{-1}\boldsymbol{x}_i \tag{9}$$

The SLAM problem can be represented as a nonlinear least squares problem

$$\mathbf{X}^* = \arg\min_{\mathbf{X}} \Big\{ \sum_{\langle i,j\rangle \in C} \boldsymbol{F}_{ij}^{closure} + \sum_i \boldsymbol{F}_i^{lidar} \Big\} \tag{10}$$

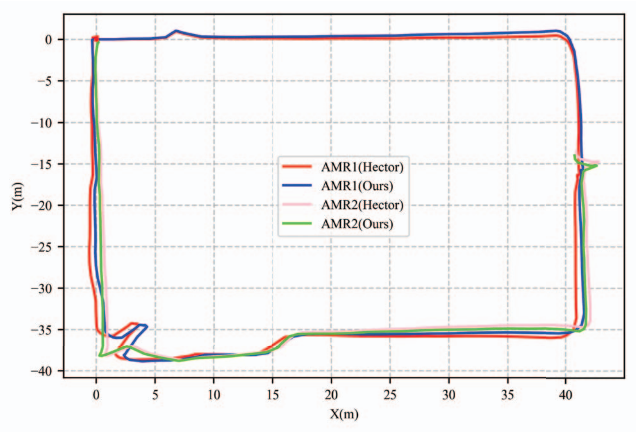
$$\boldsymbol{F}_{ij}^{closure} = \boldsymbol{e}_{ij}^T \Omega_{ij} \boldsymbol{e}_{ij}, \quad \boldsymbol{F}_{ij}^{lidar} = \boldsymbol{e}_i^T \Omega_i \boldsymbol{e}_i \tag{11}$$

where $\boldsymbol{F}_{ij}$ denotes the negative log-likelihood function of one loop closure constraint between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, and Ω_{ij} is the covariance that is related to the fitness-score of registration between the nodes. Correspondingly, $\boldsymbol{F}_i^{lidar}$ represents the cost function of LiDAR odometry constraints. The SLAM system aims to find a set of multiple robot poses which minimize the total error. For SLAM back-end optimization, there have been several open-source libraries. We implement the global multi-robot optimization module using the gtsam library [17].

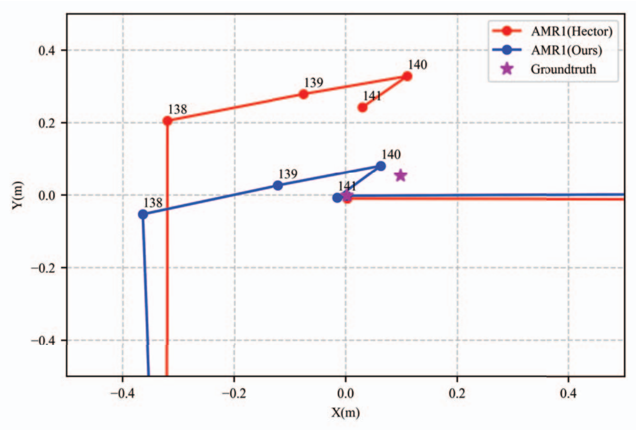
IV. Experiment

A. Experimental Setup

For validation, we collect datasets using the real AMR, which is equipped with a 2D LiDAR and a wheel odometry. The field of view of the LiDAR is $180°$ and the scanning frequency is 15Hz, correspondingly 1024 points. The mapping

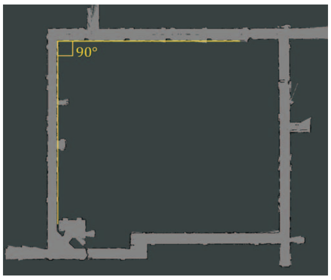


(a) Localization trajectories of two AMRs.

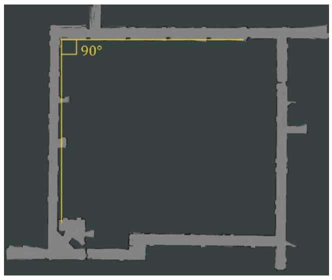


(b) End-to-end error evaluation.

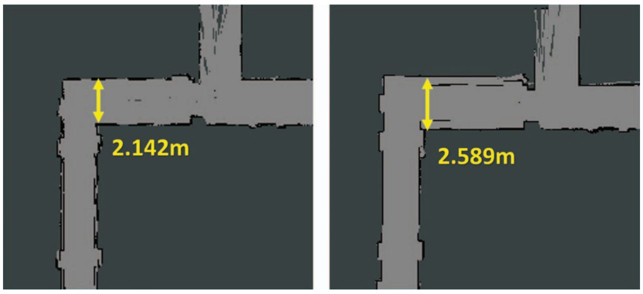
Fig. 4: Differences between Hector SLAM and our proposed method on localization results.



(a) Our Method: the map of long corridor built by multiple robots is consistent and accurate. The corridor is straight according to the reference line.



(b) Hector SLAM: due to accumulated error, when the robot returns to its origin, the map is overlapped. Meanwhile, the long corridor is skewed.



(c) The width of the corridor. We measure the width of the corridor around the origin using the tape ruler, whose ground truth is $2.141m$. Our method result is $2.142m$ (left), in contrast, the Hector slam result is $2.589m$ (right).

Fig. 5: Differences between Hector SLAM and our proposed method on mapping results.

environment is chosen as the 3rd floor of No.2 Science Building in Peking University, which is challenging due to its long corridor and loop structure. Two robots are running in the environment under WiFi communication network, starting from the same point in the opposite direction and encountering in the middle location. One AMR completes a full round around the corridor and returns to the starting point, while the other runs half a round. For the experiment, we only use the 2D Lidar without other sensors, afterwards, the wheel odometry and IMU are optional.

TABLE I: End-to-end transformation error

Keyframe	Trans(m)		Rota(rad)	
ID	Hector	Ours	Hector	Ours
140	0.2749	**0.0443**	0.0363	**0.0099**
141	0.2454	**0.0179**	0.0361	**0.0095**

B. Localization Results

We compare the proposed multi-robot framework with the original single-robot Hector SLAM on mapping efficiency, mapping quality, and localization accuracy. The comparison of localization trajectory is as Fig. 4. In real indoor environments, the ground truth of localization is hard to get, so we use end-to-end translation and rotation error to demonstrate localization accuracy, that true transformations can be computed by the manual pointcloud registration. For AMR1, it generates 141 keyframes, that the $140th$ and the $141th$ can be aligned with the $1st$ successfully. From Fig. 4b zooming in on Fig. 4a around the origin, we can intuitively prove estimated poses by our proposed method are better than single-robot Hector SLAM. The relative translation and rotation error of all methods when the AMR1 returns to the starting point is shown in Table I, all evaluation metrics are better than the Hector SLAM method. The centimeter-level positioning of the proposed method results in better mapping quality below.

C. Mapping Results

As Fig. 5b, due to long-time operation, large accumulated error results in the overlapped map and skewed corridor according to the vertical reference line. In contrast, the mapping result of the proposed method with back-end is more consistent and accurate (Fig. 5a). To further analyze mapping results

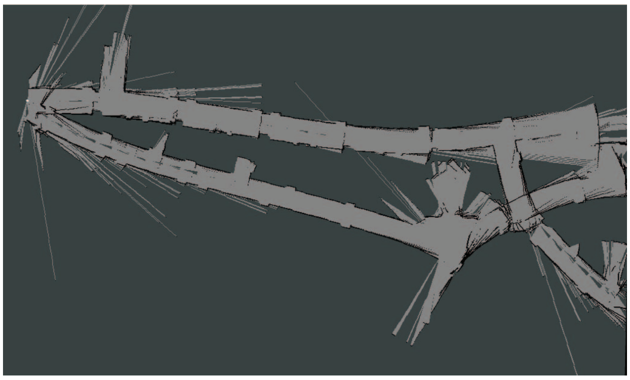


Fig. 6: The mapping result without loop closure selection. Wrong loop closure transformations between nodes make the built map distorted severely.

quantitatively, we measure the width of the corridor in the built map when AMR1 returns to the origin (Fig. 5c). Utilizing the "measurement" tool in Rviz, the result of Hector SLAM is $2.589m$, instead, ours is $2.142m$. The ground truth obtained through manual measurement is $2.141m$, demonstrating good mapping results of our proposed method. Furthermore, the mapping efficiency is significantly improved with the assistance of multiple robots.

D. Loop Closure Selection Analysis

For a large number of loop closures in the multi-robot SLAM system, we propose the loop closure selection in III-E to avoid using wrong loop closure transformations. The key to solving loop closure detection lies in leveraging accurate data associations rather than introducing as many candidates as possible, as only a few accurate loop closure associations can already significantly improve accuracy. In this paper, the number of refined loop closure pairs is $3-5$ every time, compared with hundreds of candidates. Due to the challenging long corridor environment, there are many similar scans in the keyframe queue which may generate wrong transformations between pose nodes. Without the loop closure selection, the localization and mapping process fails as Fig 6 because of wrong data associations in the factor graph. The result shows that the multi-robot system resorts to the loop closure selection to improve accuracy of localization and robustness of mapping.

V. Conclusions

In this paper, we have presented a centralized multi-robot collaborative SLAM framework for 2D LiDAR observations, which improves localization accuracy, mapping efficiency, and optimization robustness. The centralized architecture could process information from each robot equally and optimize them together efficiently. For more data associations in a multi-robot system, we present the loop closure selection method to retain correct transformations and reject wrong ones such that the system could take full advantage of the redundant data. We compare the result with original single-robot Hector SLAM, showing the accuracy and robustness of our method. Furthermore, the exploration of multiple robots proves to be beneficial to improve mapping efficiency and build a more consistent global map.

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, 2016.

[2] 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.

[3] E. Olson, "M3rsm: Many-to-many multi-resolution scan matching," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 5815–5821.

[4] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, vol. 2, no. 9. Berkeley, CA, 2014, pp. 1–9.

[5] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," in *2011 IEEE international symposium on safety, security, and rescue robotics*. IEEE, 2011, pp. 155–160.

[6] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 1271–1278.

[7] P. Schmuck and M. Chli, "Ccm-slam: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams," *Journal of Field Robotics*, vol. 36, no. 4, pp. 763–781, 2019.

[8] C. Forster, S. Lynen, L. Kneip, and D. Scaramuzza, "Collaborative monocular slam with multiple micro aerial vehicles," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 3962–3970.

[9] 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, 2020.

[10] 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, 2021.

[11] 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, 2022.

[12] 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)*, 2018.

[13] P. Huang, L. Zeng, K. Luo, J. Guo, Z. Zhou, and X. Chen, "Colaslam: Real-time multi-robot collaborative laser slam via edge computing," in *2021 IEEE/CIC International Conference on Communications in China (ICCC)*. IEEE, 2021, pp. 242–247.

[14] M. Magnusson, "The three-dimensional normal-distributions transform: an efficient representation for registration, surface analysis, and loop detection," Ph.D. dissertation, Örebro universitet, 2009.

[15] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611. Spie, 1992, pp. 586–606.

[16] X. Chen, T. Läbe, A. Milioto, T. Röhling, J. Behley, and C. Stachniss, "Overlapnet: A siamese network for computing lidar scan similarity with applications to loop closing and localization," *Autonomous Robots*, pp. 1–21, 2022.

[17] F. Dellaert and M. Kaess, "Georgia tech smoothing and mapping (gtsam)," *2019*, 2019.