

Article

Distributed Multi-Robot SLAM Algorithm with Lightweight Communication and Optimization

Jin Han ¹, Chongyang Ma ², Dan Zou ¹, Song Jiao ², Chao Chen ² and Jun Wang ^{2,*}

¹ Intelligent Science and Technology Academy Limited of CASIC, Beijing 102202, China; hanjinxxx@126.com (J.H.); 13910279390@163.com (D.Z.)

² College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100013, China; 2023210509@buct.edu.cn (C.M.); 2022200791@buct.edu.cn (S.J.); chenchao@buct.edu.cn (C.C.)

* Correspondence: wangjunrob@buct.edu.cn

Abstract: Multi-robot SLAM (simultaneous localization and mapping) is crucial for the implementation of robots in practical scenarios. Bandwidth constraints significantly influence multi-robot SLAM systems, prompting a reliance on lightweight feature descriptors for robot cooperation in positioning tasks. Real-time map sharing among robots is also frequently ignored in such systems. Consequently, such algorithms are not feasible for autonomous multi-robot navigation tasks in the real world. Furthermore, the computation cost of the global optimization of multi-robot SLAM increases significantly in large-scale scenes. In this study, we introduce a novel distributed multi-robot SLAM framework incorporating sliding window-based optimization to mitigate computation loads and manage inter-robot loop closure constraints. In particular, we transmit a 2.5D grid map of the keyframe-based submap between robots to promote map consistency among robots and maintain bandwidth efficiency in data exchange. The proposed algorithm was evaluated in extensive experimental environments, and the results validate its effectiveness and superiority over other mainstream methods.

Keywords: SLAM; multi-robot systems; distributed robot system



Citation: Han, J.; Ma, C.; Zou, D.; Jiao, S.; Chen, C.; Wang, J. Distributed Multi-Robot SLAM Algorithm with Lightweight Communication and Optimization. *Electronics* **2024**, *13*, 4129. <https://doi.org/10.3390/electronics13204129>

Academic Editor: Andrea Bonci

Received: 20 August 2024

Revised: 28 September 2024

Accepted: 6 October 2024

Published: 21 October 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

SLAM (simultaneous localization and mapping) is a fundamental technology for mobile robot navigation in unknown environments. Many outstanding single-robot SLAM algorithms have been proposed in the last decade, e.g., LOAM [1] and LIO-SAM [2]. However, in specific tasks, such as rescue, planetary exploration, and logistics applications, multi-robot SLAM is more efficient than single-robot SLAM. Compared with single-robot SLAM, multi-robot SLAM is still a challenging issue for most scenes.

Multi-robot SLAM relies on information interaction to build a consistent map when cooperative robot teams traverse an unknown environment. According to the interaction manner, multi-robot SLAM systems can be divided into the following two types: centralized and decentralized approaches. Centralized approaches compute the global SLAM through a remote base station that can interact with each team robot [3,4]. Decentralized approaches rely only on the communication between robots to build a consistent map, such that it does not need any central authority [5,6]. Due to the limitations of networking coverage, robots cannot always maintain a stable connection to the base station. Thus, the decentralized solution is better suited for large-scale environments. However, in both multi-robot SLAM approaches, a consistent map is derived from the constraints of inter-robot loop closures. For the optimization of inter-robot constraints in the distributed approach, it maintains several graphs across robots. Each robot optimizes the localization with respect to other robots only when there are overlapping constraints.

The optimization of inter-robot constraints is a critical problem for multi-robot SLAM. Traditional multi-robot SLAM methods optimize the robot's entire map and trajectory based on current and historical inter-robot loop closures. However, the computation burden rises with increases in trajectory and inter-robot loop closures, making conventional algorithms unsuitable for large-scale scenes. Moreover, besides cooperative localization, multi-robot SLAM also needs to generate an accurate and robust map that can be used for collaborative planning and navigation in practice tasks. However, to overcome the limitation of communication bandwidth, most current multi-robot SLAM approaches only transmit lightweight descriptors between robots. Each robot can calculate its global pose in real time but cannot obtain the map information of other robots. Thus, the map built by these methods can only be applied to navigation tasks after the completion of the SLAM process. This limits the application of these multi-robot SLAM methods to tasks such as exploration and rescue in fully autonomous scenarios.

In this paper, a distributed multi-robot SLAM framework with a sliding window-based global graph optimization procedure is designed. This framework handles the computation cost and constraints of the inter-robot loop closure sequence. In particular, we transmit a 2.5D grid map of the keyframe-based submap between robot teams to achieve low-bandwidth information exchange. We use RING [7], a general non-learning framework with roto-translation invariance for inter-loop closure detection and estimation of a coarse relative pose. Then, this result is used as the initial value of the Iterative Closest Point (ICP) to generate a more accurate relative pose between two robots. Subsequently, pose graph optimization is used to update the poses of robots in the sliding window. Finally, the global map of each robot is updated based on the updated consistent robot poses. Moreover, the global grid map of each robot can be directly used for distributed multi-robot planning and navigation. The contributions of our work are summarized as follows:

1. A 2.5D grid map of the keyframe-based submap is used for inter-machine communication, reducing bandwidth while achieving a balance between bandwidth and the robustness of map fusion.
2. Sliding window optimization is used to update the poses of robots, ensuring the efficiency of pose optimization and enhancing the real-time performance of the algorithm.
3. An extensive evaluation of the system's overall performance is conducted under different conditions through a real-world experiment.

2. Related Work

Multi-robot SLAM systems typically include two essential components, namely inter-loop closure detection and relative pose estimation. For inter-loop closure, various methods employ place descriptors such as NetVLAD [8], Bag of Words (BoW) [9], and AnyLoc [10] for visual-based approaches and Scan Context [11], PointNetVLAD [12], and DiSCO [13] for laser-based approaches to identify the same location. Subsequently, relative pose estimation is carried out using Perspective-n-Point (PnP) for image feature points and Iterative Closest Point (ICP) for laser point clouds.

DOOR-SLAM [5] is a fully distributed multi-robot SLAM system featuring a robust outlier rejection mechanism. It utilizes NetVLAD descriptors in its inter-loop closure module, while its pose graph optimization module integrates a distributed outlier rejection model to achieve accurate and robust collaborative positioning and mapping. DiSCO-SLAM [14] is the first method to use a LiDAR-based global descriptor, i.e., scan context [11], for distributed place recognition. The lightweight nature of the scan context descriptor facilitates real-time applications. DRACo-SLAM [15] estimates multi-robot system trajectories in GPS-free environments using scene descriptors and dynamic event-triggered communication. Kimera-Multi [16] integrates D-GNC and semantic data into its generated maps. Tian et al. [17] improved Kimera-Multi to make it suitable for large-scale practical deployments, with a particular focus on handling intermittent and unreliable communications. Bozzi et al. [18] proposed a solution for dynamically adjusting the trajectories of vehicle platoons, ensuring that optimal spacing is maintained between adjacent vehicles traveling at cru-

ing speed on a straight road. Multi-modal maplab 2.0 [19] supports heterogeneous robot groups with diverse sensor configurations. Swarm-SLAM [20] implements an innovative budgeted method for selecting inter-robot loop closures by maximizing algebraic connectivity. DCL-SLAM [21] evaluates the performance of LiDARiris [22], M2DP [23], and scan context, ultimately selecting the most effective and rotation-invariant LiDAR-Iris for loop closure detection.

However, previous systems were mainly designed for multi-robot positioning, ignoring map sharing between robots. Thus, they are not feasible in multi-robot exploration with a high dependency on maps. Compared with the sharing of a complete single-robot map, incremental map construction and merging only need to share newly added parts of maps and put little pressure on the communication bandwidth. SMMR-Explore [24] is a submap-based multi-robot SLAM framework that only shares the grid map, eliminating the transfer of place recognition descriptors and raw sensor data. This framework saves 30% of the total communication traffic, and the aligned global map can be directly used for the exploration task. Nevertheless, SMMR-Explore transmits 2D laser scans between robots, easily leading to incorrect loop closure detection and limiting its application to rough and complex roads. Motivated by this limitation, we utilize 2.5D submap interaction among robots to achieve low-bandwidth information exchange and the construction of a global map in each robot. The communication protocol for the 2.5D submap incorporates timestamps to accurately pinpoint the generation time of the data packet, ensuring precise synchronization. It also includes the pose of the last scan, detailing the sensor's position and orientation, which are essential for relative localization and environmental reconstruction. The protocol further specifies the dimensions of the submap on the x and y axes and a one-dimensional array that stores the height values of each voxel point, facilitating the construction of three-dimensional terrain models. Additionally, it integrates RTK data from the last scan to provide real-time kinematic positioning information. The 2.5D global map contains the height information for the robot, enabling more reasonable and safe navigation in complex terrain. In particular, a sliding window-based global graph optimization procedure is designed to handle the increased computation cost as the scale of the environment expands.

3. Approach

Each robot individually performs a SLAM task. Within the communication range, robots interchange 2.5D submaps to facilitate collaboration. The shared submaps are used to determine whether the robots are encountering a common scene. Subsequently, the relative positions and orientations between the robots are calculated by aligning the observations of the identical scene. Finally, leveraging the computed relative poses of the robot team, a consistent map of the explored area is obtained based on the transformation and integration of individual submaps.

3.1. Inter-Robot Place Recognition

This research employs 2.5D submaps to facilitate lightweight inter-robot communication. For inter-robot place recognition, RING [7], which represents LiDAR scans in the form of Radon sinograms, is employed. To create a translation-invariant descriptor, RING applies a 1D discrete Fourier transform to each row along the d axis, obtaining the magnitude spectrum in the frequency domain. The resultant spectra of the query-scan RING and map-scan RING are denoted as S_Q and S_M , respectively. S_Q can be expressed as

$$S_Q = |\mathcal{F}(\mathcal{D}_Q(\theta, d))|, \quad (1)$$

where $\mathcal{F}(\cdot)$ is the discrete Fourier transform operator. Subsequently, RING applies cross-correlation along the θ axis between S_Q and S_M to obtain the orientation invariance. The

max correlation value ($C(S_Q, S_M)$) of the circular cross-correlation results is considered as the metric between the query scan and map scan as follows:

$$C(S_Q, S_M) = \max_{\theta_i} \sum S_Q(\theta_i, \omega) \cdot S_M(\theta_i + \alpha, \omega), \quad (2)$$

where α denotes an orientation of $f(x, y)$ and ω is the discrete frequency. Then, the most similar map scan, which should be acquired in the same place as the query scan, can be achieved. Notably, our algorithm also leverages the Pairwise Consistency Maximization (PCM) [25] approach. This approach identifies whether the loop-closure pairs satisfy the criteria for establishing inter-robot associations.

3.2. Relative Pose Optimization

To calculate the relative transform between the robot teams, we designed a sliding window-based optimization module. In contrast to the traditional optimization method, the proposed optimization method focuses on optimizing the estimated trajectory and map within a time window. It integrates the relative transform of the candidate loop-closure pair provided by the place recognition module with odometry data and the pose of the first keyframes in a sliding window as follows:

$$\bar{\mathbf{X}}_{t-L,t} = \arg \min_{\mathbf{X}_{t-L,t}} (F_{inter}(\mathbf{X}_{t-L,t}) + E_o(\mathbf{X}_{t-L,t}) + E_f(\mathbf{X}_{t-L,t})), \quad (3)$$

where $F_{inter}(\mathbf{X}_{t-L,t})$ is the cost function of inter-loop closure. E_o is the residual error, which is calculated based on the constraint of odometry data. $E_f(\mathbf{X}_{t-L,t})$ is the residual error linked with the pose of the first keyframes in the sliding window. Since the sliding window-based optimizer mainly optimizes the partial path, $E_f(\mathbf{X}_{t-L,t})$ is used to prevent discontinuity at path junctions. In particular, we also incorporate Gaussian noise to simulate observation uncertainty in the real world. The sliding window-based optimizer ensures the effective utilization of computational resources while providing sufficient context information for precise pose inference.

After optimization, the last 10 keyframes of the odometry and the associated loop closures are retained in the sliding window. We emphasize the reuse and maintenance of effective loop closure and odometry factors in the optimization process. This strategy helps mitigate potential negative impacts from suboptimal optimizations on subsequent pose tracking. It ensures the coherence and stability of pose estimation throughout the entire process.

3.3. Efficient, Consistent Map

Based on the relative pose matrices, the proposed approach subsequently transforms the local maps from each robot's coordinate system into the global coordinate. For the maintenance of the global map, an increasing scope of the area mapped by the robots leads to substantial storage requirements. Voxel filtering is a voxel-based method for point-cloud filtering. It divides the point-cloud space into a series of cubic units at a predefined resolution. Each unit encompasses one or multiple points, enabling noise and outlier removal without compromising the geometric integrity of the point cloud. Therefore, we apply voxel filtering to the global map to efficiently reduce spatial occupancy.

To address unavoidable, dynamic obstacles in the environment, which can impact backend loop closure detection and global map construction, we adopt a strategy based on relative motion to remove dynamic points. The current point cloud is denoted as P_t , and the point cloud of the previous frame is expressed as P_{t-1} . The transformation matrix between consecutive frames is represented as T_{t-1}^t , which can be calculated using the consecutive relative poses. Using this transformation matrix, the point cloud from the previous frame can be transformed into the coordinate system of the current frame as follows:

$$P'_{t-1} = T_{t-1}^t \cdot P_{t-1}. \quad (4)$$

Then, the dynamic points can be recognized by calculating the disparity between the transformed point cloud (P'_{t-1}) and the current frame's point cloud (P_t).

Notably, dynamic points are further removed in this paper. In a multi-robot exploration process, an area might be surveyed by different robots at varying times. The presence of dynamic objects in this region during the passage of certain robots can lead to disparate observation outcomes. In these specific areas, we can achieve clearer removal based on the disparate observations from different robots. The criteria are defined as follows:

$$\bar{A}(p) = \prod_{i=1}^N A^i(p), \quad (5)$$

where $\bar{A}(p)$ is the comprehensive property of the point that appears in the area scanned by multiple robots and $A^i(p)$ is the property in robot i , which is equal to 0 or 1. Equation (5) means that only if one robot observes the area without dynamic objects can the dynamic points generated by other robots be removed. On the other hand, regarding the loss of sensor data and communication interruptions between multiple robots, the proposed method includes mechanisms to handle such challenges effectively. Following the initialization of multiple robots through RTK, their coordinates are unified into their respective North–East–Up (NEU) coordinate frames. Even if there is a disruption in information transmission among the robots, upon the re-establishment of communication, the missing submaps can be seamlessly integrated into the global map. Furthermore, the corresponding poses can be directly incorporated into the factor graph for optimization, facilitated by the unified coordinate system.

4. Experiments

4.1. Real Environment

4.1.1. Implementation Details

The proposed algorithm is evaluated by a self-built dataset (https://drive.google.com/file/d/1urfKVAkVlkRyDMFjhatv-Jexst7vnjsp/view?usp=drive_link, accessed on 28 September 2024) using three sensor-identical AGILE·X vehicles operating in an industrial park. In this experiment, the first and third robots are wheeled robots. The specific parameters are outlined as follows: the model is SCOUT 2.0, with dimensions of 930 mm × 699 mm × 349 mm (Length × Width × Height), a maximum speed of 1.5 m/s, and a four-wheel independent drive system. It supports rapid secondary development through a standard protocol and CAN bus. The second robot is a BUNKER tracked robot with dimensions of 1023 mm × 778 mm × 400 mm (length × width × height), a maximum speed of 1.2 m/s, left and right independent drive, and tracked differential steering. The communication interface includes a standard CAN/232 serial port. Each AGILE·X vehicle is equipped with an onboard computer and a paired display. The system runs Ubuntu 20.04 and is used to operate our multi-robot SLAM program in real time. All vehicles are on the same local area network and communicate through the Robot Operating System (ROS 1). The mapping results are displayed via Rviz and can be directly interacted with by a human operator. The program can be controlled through a wireless keyboard and mouse connected to the onboard computer to start or stop the operation. The robots run on three pre-defined trajectories by the control handler. NOVATEL OEM7 RTK and RoboSense Lidar-16 are integrated into the three mobile robots, as shown in Figure 1. The LiDAR, IMU, and GPS data are recorded during the movements of three robots.



Figure 1. The three mobile robots equipped with RoboSense Lidar-16, NOVATEL OEM7 RTK, and IMU in an industrial park.

The proposed algorithm is implemented through modification based on the open-source code of MR-SLAM (https://github.com/MaverickPeter/MR_SLAM accessed on 28 September 2024). We rely on ISAM2 from the GTSAM library to implement the sliding window-based optimization. We adopted DLIO [26] as the front end of our proposed algorithm. A primary AGILE-X vehicle is chosen as the global reference, and the coordinates of the other vehicles are transformed into its coordinate frame when inter-loop closure is detected. To evaluate the performance of the proposed algorithm, an ablation study is designed and implemented. We show how the design of each novelty in the proposed system affects the accuracy and efficiency. The data obtained by NOVATEL OEM7 RTK are used as the ground truth to calculate the localization errors of the multi-robot SLAM algorithms. Trajectory error analysis is performed using the EVO tool (<https://github.com/MichaelGrupp/evo>, accessed on 28 September 2024). The accuracy of the generated maps is obtained by comparing the actual geodetic distances of five predefined landmark sets with the distances of those landmarks in the generated map, as shown in Figure 2. In experiments, all algorithms are executed on machines powered by 13th i7 processors with 16 GB RAM, all operating under Ubuntu 20.04 LTS.



Figure 2. The experimental environment in an industrial park and five sets of landmarks used to evaluate the accuracy of mapping.

4.1.2. Results and Discussion

Table 1 lists the errors of robot localization and mapping obtained under different conditions. The data in the first row show the results of the condition under which both the proposed sliding window optimizer and 2.5D submap-based communication strategy are disabled. The traditional global optimizer and 2D submap-based communication strategy are utilized. Compared with other results, this condition generates the largest errors. The second row lists the performance achieved after enabling the sliding window optimizer when inter-loop closure is detected. Compared with data in the first row, the errors of robot localization are significantly reduced with efficient optimization. In addition, the accuracy of mapping is also improved with more accurate robot localization. However, 2D submap-based transmission results in incorrect inter-loop closure under the first two conditions and leads to map alignment failure. Thus, three sets of landmarks cannot be determined, and errors are unable to be calculated. The third row presents the results achieved using the 2.5D submap as the transmission information. The accuracy of inter-loop closure is efficiently improved by adding one more dimension of information. However, the accuracy of robot localization and mapping is not better than in the second row of data. The last row shows the results achieved with the two proposed modules enabled. The results indicate that the condition under which the two proposed modules are enabled in the multi-robot SLAM framework achieves the best accuracy.

Table 1. Errors of the robot trajectory and map generated under different conditions in an ablation study. SWO denotes sliding window optimization.

SWO	2.5 D	Mapping Error [m]						Robot Localization Error [m]			
		1–2	3–4	5–6	7–8	9–10	AE	Robot1	Robot2	Robot3	AE
✗	✗	—	0.34	—	0.28	—	0.31	2.78	5.65	6.08	4.84
✓	✗	—	0.16	—	0.14	—	0.15	1.02	0.99	4.54	2.18
✗	✓	0.14	0.12	0.15	0.16	0.07	0.13	0.88	1.86	0.92	1.22
✓	✓	0.14	0.10	0.04	0.13	0.01	0.08	0.62	0.64	0.44	0.57

To validate the accuracy of our localization results, we incorporated statistical analysis into the positioning errors, as listed in Table 2. We conducted multiple tests across several trajectories of different lengths, and the test results are used as our sample data for statistical analysis. The analysis results indicate that the sample mean error is 0.5497 and the sample variance is 0.0237. With a confidence level of 0.95, the confidence interval is [0.4396, 0.6597]. The significance test employed a t-test, with a t-statistic of -0.4182 and a p value of 0.6856. The results of the significance test support the hypothesis that the mean is 0.57, demonstrating that our test results are accurate.

Table 2. The trajectory errors of each robot in multiple tests across different segments of various trajectories.

Traj	1	2	3	4	5	6	7	8	9	10	AE
Robot1	0.29	1.20	0.40	0.22	0.33	0.58	0.79	0.72	0.56	0.99	0.61
Robot2	0.53	0.44	0.68	0.41	0.59	0.75	0.65	1.16	0.72	0.46	0.64
Robot3	0.29	0.28	0.34	0.30	0.28	0.43	0.51	0.48	0.49	0.61	0.41
AE	0.37	0.64	0.47	0.31	0.40	0.59	0.65	0.78	0.59	0.69	0.55

Figure 3 shows comparisons between trajectories estimated under four conditions and ground-truth trajectories. The sequence of results in Figure 3a–d corresponds to the sequence in Table 1. The results show that the method that integrates the sliding window optimizer and the 2.5D submap-based communication strategy achieves the best

performance. Incorrect estimation of relative pose appears in Figure 3a,b when using 2D submaps. In particular, three trajectories in Figure 3a generate large deviations due to the use of the global optimization method. However, in Figure 3b, only the trajectory of the third robot generates a large deviation by using the proposed sliding window optimizer. This result also occurs in Figure 3c,d. Additionally, comparison of Figure 3a–d further indicates that 2.5D submaps outperform 2D submaps. Furthermore, to validate our low-bandwidth information exchange, we compared the data sizes of 3D, 2.5D, and 2D submaps required for each communication. The data sizes of 2.5D and 2D submaps are similar, at 71.99 KB and 70.18 KB, respectively, whereas the data size of 3D submaps is 1.30 MB.

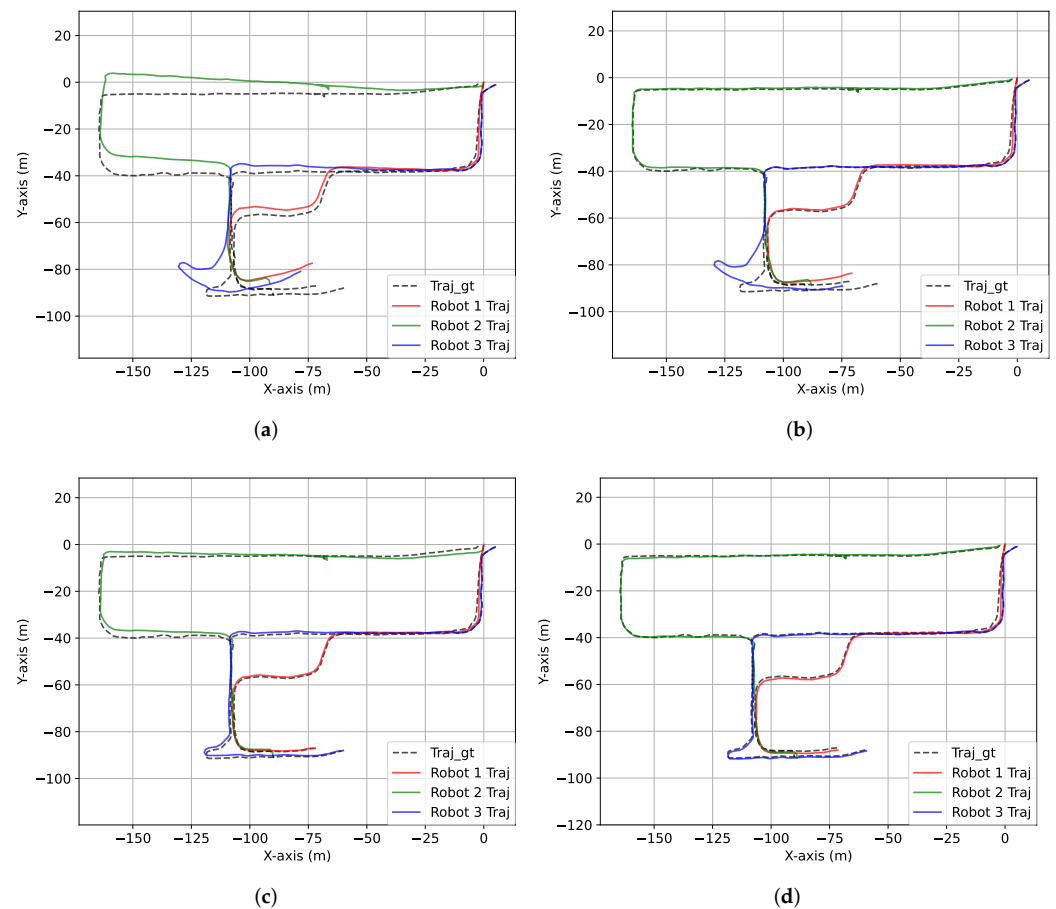


Figure 3. Performances of robot trajectories under different conditions in an ablation study. (a) Both sliding window optimization and 2.5D submap are disabled; (b) only sliding window optimization is enabled; (c) only 2.5D submap is enabled; (d) both sliding window optimization and 2.5D submap are enabled.

The efficiency of the sliding window optimizer was also manifested in the experiments. The time consumption of the optimization process under the four conditions is shown in Figure 4. The time consumption of the global optimizer increases with an increase in the number of inter-loop closures. The real-time performance of the global optimizer is also affected in larger scenarios. However, the proposed sliding window optimizer requires limited stationary time consumption. Comprehensive consideration of error and time consumption indicates that the proposed sliding window optimizer is more accurate and efficient than the traditional global optimizer.

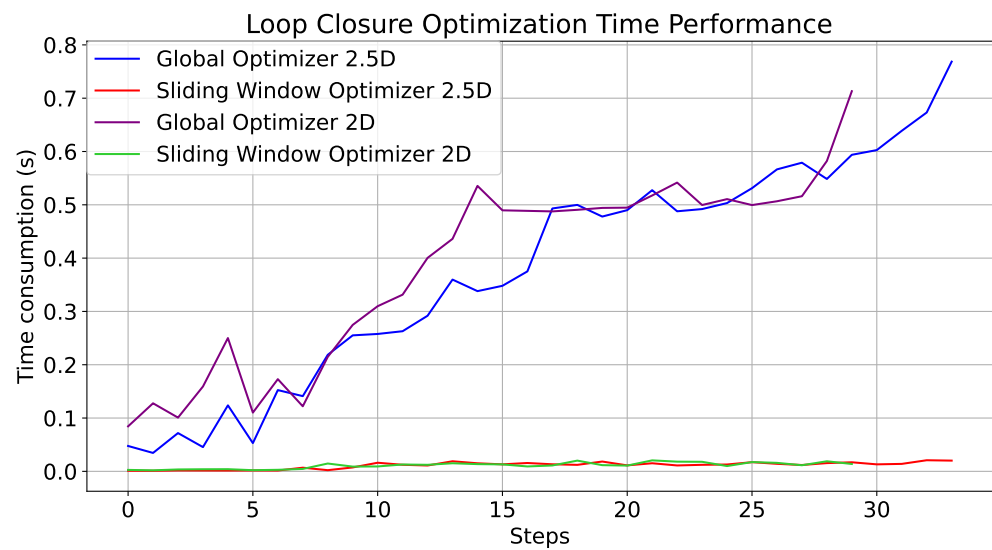


Figure 4. Per-step time consumption curves under different conditions in an ablation study.

4.2. Public Dataset

Furthermore, the performance of the proposed method is compared with DiSCO-SLAM [14] on a public dataset. We adapt the dataset from the raw data of the KITTI Vision Benchmark (sequence 08) [27]. In sequence 08, raw inertial measurement unit (IMU) data sampled at 100 Hz are recorded with temporal coherence, rendering them compatible with integration with the LIO-SAM and DLIO frameworks. Two robots are recorded in sequence 08. Each robot trajectory in the dataset includes intra-robot and inter-robot loop closures. When GPS signals are accessible, the GPS data are used as ground truth for quantitative analysis but not integrated into the SLAM process. Trajectory error analysis is also performed using the EVO tool.

Figure 5 shows the performances of DiSCO-SLAM and the proposed method on KITTI-08. It shows that the proposed method achieves better performance than DiSCO-SLAM. The trajectories of two robots estimated by the proposed method almost coincide with the ground truth. However, the trajectory of robot 2 estimated by DiSCO-SLAM generates a significant deviation.

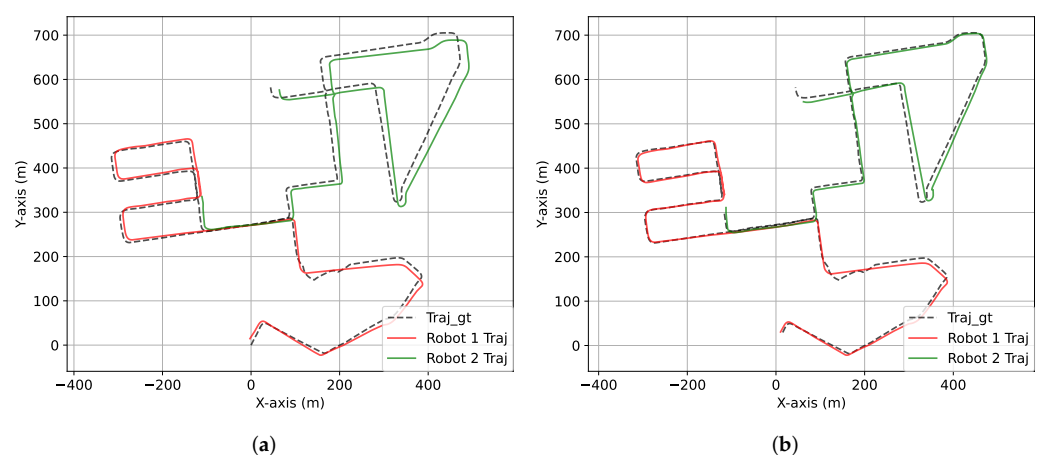


Figure 5. Performances of DiSCO-SLAM and the proposed method on KITTI-08. (a) DiSCO-SLAM with two robots; (b) the proposed method with two robots.

Furthermore, based on the quantitative analysis provided by the EVO tool, the average trajectory errors of robot 1 and robot 2 generated by DiSCO-SLAM are 4.21 m and 8.68 m, respectively. In contrast, our proposed method yields average trajectory errors of robot 1 and robot 2 of 3.0 m and 3.98 m, respectively. These results indicate that our proposed method demonstrates superior performance compared to DiSCO-SLAM. DiSCO-SLAM uses scan context to extract descriptors for place recognition. However, scan context can only provide the initial rotation estimate for ICP, lacking an initial translation estimate, which leads to larger matching errors and affects the accuracy of ICP's relative pose estimation. In contrast, our method uses the RING descriptor for place recognition, which can provide both initial rotation and translation estimates, enabling ICP to solve the relative pose more quickly and accurately.

On the other hand, the 2.5D grid map used in our proposed method has several limitations compared to the 3D map. Incomplete height information can lead to the loss of terrain details and possible penetration effects when objects are occluded. Additionally, the insufficient representation of vertical details reduces navigation precision and increases the difficulty in merging data from different robots. In future work, to address these issues, the multi-sensor data fusion method is a potential solution to improve the consistency and completeness of the map, and deep learning technology is also a promising method to predict and fill in missing height information.

5. Conclusions

In this paper, a novel distributed, multi-robot SLAM framework incorporating sliding window-based global optimization is proposed. The sliding window-based optimizer ensures the efficiency of the global optimization process while avoiding the impact of incorrect inter-loop closure on early results. In addition, a 2.5D keyframe-based submap is transmitted between robots to promote map consistency among robots and maintain bandwidth efficiency in data exchange. The lightweight communication and optimization methods enhance the practicality of the multi-robot SLAM algorithm in large scenarios. Extensive experiments demonstrate the validity and superiority of the proposed algorithm. However, our approach still has certain limitations. Specifically, compared to a 3D map, our 2.5D submap lacks rich spatial geometric structures, which could negatively affect subsequent path planning and obstacle avoidance during navigation. In future work, we will focus on addressing these limitations and enhancing our method to support multi-robot exploration tasks.

Author Contributions: Conceptualization, D.Z.; methodology, J.H. and C.M.; software, S.J. and C.C.; validation, C.M. and S.J.; writing—original draft preparation, J.H. and C.M.; writing—review and editing, J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by Beijing Natural Science Foundation (L233029) and in part by National Natural Science Foundation of China under Grant 62303040.

Data Availability Statement: Data will be made available upon request to the corresponding author.

Conflicts of Interest: Jin Han and Dan Zou was employed by the Intelligent Science and Technology Academy Limited of CASIC. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Zhang, J.; Singh, S. LOAM: Lidar Odometry and Mapping in real-time. In Proceedings of the Robotics: Science and Systems Conference (RSS), Berkeley, CA, USA, 12–16 July 2014; pp. 109–111.
2. Shan, T.; Englot, B.; Meyers, D.; Wang, W.; Ratti, C.; Daniela, R. LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020; IEEE: New York, NY, USA, 2020; pp. 5135–5142.

3. Dubé, R.; Gawel, A.; Sommer, H.; Nieto, J.; Siegwart, R.; Cadena, C. An online multi-robot SLAM system for 3D LiDARs. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 1004–1011.
4. Zhang, P.; Wang, H.; Ding, B.; Shang, S. Cloud-Based Framework for Scalable and Real-Time Multi-Robot SLAM. In Proceedings of the 2018 IEEE International Conference on Web Services (ICWS), San Francisco, CA, USA, 2–7 July 2018; pp. 147–154.
5. Lajoie, P.Y.; Ramtoula, B.; Chang, Y.; Carlone, L.; Beltrame, G. DOOR-SLAM: Distributed, Online, and Outlier Resilient SLAM for Robotic Teams. *IEEE Robot. Autom. Lett.* **2020**, *5*, 1656–1663. [\[CrossRef\]](#)
6. Chang, Y.; Tian, Y.; How, J.P.; Carlone, L. Kimera-Multi: A System for Distributed Multi-Robot Metric-Semantic Simultaneous Localization and Mapping. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation, Xi'an, China, 30 May–5 June 2021; pp. 11210–11218.
7. Lu, S.; Xu, X.; Yin, H.; Chen, Z.; Xiong, R.; Wang, Y. One RING to Rule Them All: Radon Sinogram for Place Recognition, Orientation and Translation Estimation. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; pp. 2778–2785.
8. Arandjelovic, R.; Gronat, P.; Torii, A.; Pajdla, T.; Sivic, J. NetVLAD: CNN architecture for weakly supervised place recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 5297–5307.
9. Jégou, H.; Douze, M.; Schmid, C.; Pérez, P. Aggregating local descriptors into a compact image representation. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 3304–3311.
10. Keetha, N.; Mishra, A.; Karhade, J.; Jatavallabhula, K.M.; Scherer, S.; Krishna, M.; Garg, S. AnyLoc: Towards Universal Visual Place Recognition. *arXiv* **2023**, arXiv:2308.00688. [\[CrossRef\]](#)
11. Kim, G.; Kim, A. Scan Context: Egocentric Spatial Descriptor for Place Recognition Within 3D Point Cloud Map. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 4802–4809.
12. Uy, M.; Lee, G. PointNetVLAD: Deep Point Cloud Based Retrieval for Large-Scale Place Recognition. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 4470–4479.
13. Xu, X.; Yin, H.; Chen, Z.; Li, Y.; Wang, Y.; Xiong, R. DiSCO: Differentiable Scan Context with Orientation. *IEEE Robot. Autom. Lett.* **2021**, *6*, 2791–2798. [\[CrossRef\]](#)
14. Huang, Y.; Shan, T.; Chen, F.; Englot, B. DiSCO-SLAM: Distributed Scan Context-Enabled Multi-Robot LiDAR SLAM With Two-Stage Global-Local Graph Optimization. *IEEE Robot. Autom. Lett.* **2022**, *7*, 1150–1157. [\[CrossRef\]](#)
15. McConnell, J.; Huang, Y.; Szenher, P.; Collado-Gonzalez, I.; Englot, B. DRACo-SLAM: Distributed Robust Acoustic Communication-efficient SLAM for Imaging Sonar Equipped Underwater Robot Teams. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; pp. 8457–8464. [\[CrossRef\]](#)
16. Tian, Y.; Chang, Y.; Herrera Arias, F.; Nieto-Granda, C.; How, J.P.; Carlone, L. Kimera-Multi: Robust, Distributed, Dense Metric-Semantic SLAM for Multi-Robot Systems. *IEEE Trans. Robot.* **2022**, *38*, 2022–2038. [\[CrossRef\]](#)
17. Tian, Y.; Chang, Y.; Quang, L.; Schang, A.; Nieto-Granda, C.; How, J.P.; Carlone, L. Resilient and Distributed Multi-Robot Visual SLAM: Datasets, Experiments, and Lessons Learned. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 1–5 October 2023; pp. 11027–11034. [\[CrossRef\]](#)
18. Bozzi, A.; Zero, E.; Sacile, R.; Bersani, C. Real-time Robust Trajectory Control for Vehicle Platoons: A Linear Matrix Inequality-based Approach. In Proceedings of the 18th International Conference on Informatics in Control, Automation and Robotics, Paris, France, 6–8 July 2021.
19. Cramariuc, A.; Bernreiter, L.; Tschopp, F.; Fehr, M.; Reijgwart, V.; Nieto, J.; Siegwart, R.; Cadena, C. maplab 2.0—A Modular and Multi-Modal Mapping Framework. *IEEE Robot. Autom. Lett.* **2023**, *8*, 520–527. [\[CrossRef\]](#)
20. Lajoie, P.Y.; Beltrame, G. Swarm-SLAM: Sparse Decentralized Collaborative Simultaneous Localization and Mapping Framework for Multi-Robot Systems. *IEEE Robot. Autom. Lett.* **2024**, *9*, 475–482. [\[CrossRef\]](#)
21. Zhong, S.; Qi, Y.; Chen, Z.; Wu, J.; Chen, H.; Liu, M. DCL-SLAM: A Distributed Collaborative LiDAR SLAM Framework for a Robotic Swarm. *IEEE Sensors J.* **2024**, *24*, 4786–4797. [\[CrossRef\]](#)
22. Wang, Y.; Sun, Z.; Xu, C.Z.; Sarma, S.E.; Yang, J.; Kong, H. LiDAR Iris for Loop-Closure Detection. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 25–29 October 2020; pp. 5769–5775.
23. He, L.; Wang, X.; Zhang, H. M2DP: A novel 3D point cloud descriptor and its application in loop closure detection. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Republic of Korea, 9–14 October 2016; pp. 231–237.
24. Yu, J.; Tong, J.; Xu, Y.; Xu, Z.; Dong, H.; Yang, T.; Wang, Y. SMMR-Explore: SubMap-based Multi-Robot Exploration System with Multi-robot Multi-target Potential Field Exploration Method. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 8779–8785.

25. Mangelson, J.G.; Dominic, D.; Eustice, R.M.; Vasudevan, R. Pairwise Consistent Measurement Set Maximization for Robust Multi-Robot Map Merging. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation, Brisbane, Australia, 21–25 May 2018; pp. 2916–2923.
26. Chen, K.; Nemiroff, R.; Lopez, B.T. Direct LiDAR-Inertial Odometry: Lightweight LIO with Continuous-Time Motion Correction. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 3983–3989. [\[CrossRef\]](#)
27. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The KITTI dataset. *Int. J. Rob. Res.* **2013**, *32*, 1231–1237. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.