

DOI: 10.37943/VNCP6721

Abdirakhman Onabek

Bachelor's Degree, Engineering, Robotics Department
abdirakhman.onabek@nu.edu.kz, orcid.org/0009-0008-1619-6518
Julius Maximilian University of Würzburg, Germany

Zaki Al-Farabi

Bachelor's Degree, Engineering, Robotics Department
zaki.alfarabi@nu.edu.kz, orcid.org/0009-0001-6300-4119
Nazarbayev University, Kazakhstan

Ilyas Umurbekov

Bachelor's Degree, Engineering, Robotics Department
ilyas.umurbekov@nu.edu.kz, orcid.org/0009-0003-1147-3196
Nazarbayev University, Kazakhstan

Temirlan Galimzhanov

Master's Degree, Engineering, Robotics Department
temirlan.galimzhanov@nu.edu.kz, orcid.org/0009-0004-4967-576X
Nazarbayev University, Kazakhstan

Ilyas Tursynbek

PhD, Assistant Professor
iliyas.tursynbek@nu.edu.kz, 0000-0003-3753-9969
Astana IT university, Kazakhstan

Zhanat Kappasov

PhD, Associate Professor, Robotics Department
zhkappasov@nu.edu.kz, orcid.org/0000-0003-3262-3993
Nazarbayev University, Kazakhstan

**EFFICIENT MULTI-ROBOT SLAM IN A LABYRINTH ENVIRONMENT: A
CENTRALIZED APPROACH USING ALGEBRAIC CONNECTIVITY
AUGMENTATION**

Abstract: The visual system is a central component of mobile robot navigation, as it provides essential information about the surrounding environment for motion planning and coordination. However, navigation requires the robot to interpret sensory data through localization and mapping processes, such as Simultaneous Localization and Mapping (SLAM) frameworks. Mapping such environments with a single robot remains a significant challenge due to limited coverage and the substantial time required to map large areas. Multi-robot SLAM offers a promising alternative by distributing the exploration task across several robots, yet practical implementations still struggle with issues such as inconsistent map merging and the overall computational complexity of generating a unified global map. The objective of this study is to implement and evaluate a multi-robot SLAM system in which robots operate independently, and maps are merged on a centralized server in a real labyrinth setting to determine whether collaborative mapping can be practically achieved using mobile turtlebot robots. The system integrates frontier-driven exploration with a two-stage inter-robot loop closure detection process, enabling robots to identify and validate overlapping regions during mapping. Global pose graph optimization is performed on a centralized server that aggregates data from all robots, while pose graph sparsification based on effective resistances reduces the amount of data that must be exchanged. Experiments were first conducted in simulation and later in a physical labyrinth constructed for testing. The results demonstrate that two robots were able to autonomously explore distinct regions, detect inter-robot loop closures, generate sparsified pose graphs, and merge their individual maps into a coherent global representation. The study provides an end-to-end

implementation and establishes a foundation for scaling multi-robot SLAM to more complex scenarios.

Keywords: Robotics; Robot perception; Sensing; Optimization.

1. Introduction

1.1 Problem Definition

Mobile robots rely heavily on perception systems to interpret their surroundings and make navigation decisions in real time. Sensors such as lidar, cameras, and radar provide complementary forms of environmental information, but effective autonomous navigation requires more than sensing alone. Robots must integrate this information into a coherent representation of the environment while continuously estimating their own motion, a process commonly addressed through Simultaneous Localization and Mapping (SLAM). Traditionally, SLAM is performed using a single robot, but recent advancements have shown that employing multiple robots can significantly improve the efficiency of mapping. Although single-robot SLAM frameworks are well established, their performance in large and complex indoor environments is limited by restricted sensor coverage, accumulated odometric drift, and the extended time required to explore and map an entire area. This paper explores the hypothesis that multi-robot SLAM can reduce the time required to map complex environments, such as labyrinths, by leveraging collaboration between multiple Turtlebots.

1.2 Literature Review

Multi-robot SLAM (MR-SLAM) seeks to improve exploration efficiency by allowing multiple robots to build maps simultaneously, but this advantage introduces additional challenges in relative pose estimation, synchronization, communication, and global map consistency. Saeedi et al. [1] provide a broad review of these issues and show that the main difficulty in MR-SLAM is not only local map generation, but also the reliable fusion of independently built maps into a consistent global representation. This review is valuable as a general foundation, although it is conceptual and does not propose a specific implementation strategy for low-cost indoor systems. More recent studies have addressed this limitation by improving loop closure detection [2], robust distributed map alignment [3], and optimization under uncertain measurements [4].

Centralized collaborative SLAM methods attempt to preserve map accuracy by separating local estimation from global optimization. Deutsch et al. [5] propose a multi-robot pose graph SLAM framework in which robots perform local mapping while a central workstation handles global consistency. The main strength of this approach is accurate global map alignment with reduced local computational burden, making it highly relevant to the present study. However, its dependence on reliable communication and a central processing unit limits robustness in fully distributed scenarios. Similar trade-offs appear in practical SLAM systems such as Karto SLAM, which achieves strong mapping accuracy through graph optimization and ROS-based navigation but remains more suitable for stable, well-connected robotic deployments than for lightweight collaborative operation [6]. Cloud-supported solutions have also been explored to reduce on-board computation, although they introduce additional network dependence [7].

Decentralized and distributed MR-SLAM methods improve scalability by reducing reliance on a single fusion center. Kegeleirs et al. [8] discuss Swarm SLAM as a decentralized strategy based on local robot interactions and simplified environmental representations. Its main advantage is robustness and fault tolerance in communication-limited settings, but this often comes at the expense of map precision and strict global consistency. Lajoie and Beltrame [9] extend this direction by proposing a sparse decentralized collaborative SLAM framework that reduces communication through selective constraint sharing. This is particularly important because communication cost is a major limitation in collaborative mapping. However, as the system becomes more decentralized, loop closure management and consistency enforcement become more complex. Recent distributed LiDAR SLAM systems such as DCL-SLAM [10] and SKiD-SLAM [11] further demonstrate that collaborative mapping can be achieved in swarm or resource-

constrained settings, but they also confirm that distributed robustness usually requires greater coordination complexity.

A key component of collaborative mapping is reliable loop closure detection and place recognition. Kim G. and Kim A. [12] show that compact descriptors can efficiently identify loop closure candidates in point-cloud maps, which is useful for reducing front-end matching cost. Similarly, Zhou et al. [2] propose a 3D point-cloud descriptor to improve loop closure detection performance. These methods strengthen the front end of SLAM, but descriptor similarity alone is insufficient for reliable map fusion in repetitive environments; geometric verification is still required. This limitation is particularly important in labyrinth-like environments, where similar corridors or intersections may increase the risk of false matches. More advanced systems such as Kimera-Multi [3] show that robust distributed multi-robot SLAM is possible, but such systems are computationally more complex than the graph-based fusion pipeline considered in this work.

Graph-based optimization remains the dominant back-end formulation in modern SLAM. Kümmerle et al. [13] provide the g2o framework, which is widely used for nonlinear optimization of sparse pose graphs and is therefore highly relevant to collaborative map fusion. However, optimization quality depends not only on the solver, but also on the structure of the graph itself. Spielman and Srivastava [14] show that graph sparsification can reduce graph complexity while preserving important structural properties. This idea has been further developed for SLAM through spectral measurement sparsification [15], algebraic-connectivity maximization [16], and learning-based sparsification strategies [17]. These methods are especially important in collaborative SLAM because they reduce communication and optimization cost without discarding the most informative constraints. Although GO-SLAM [18] is not specifically a multi-robot framework, it further illustrates the importance of strong global optimization for maintaining consistency in large-scale reconstruction.

Exploration efficiency is another important aspect of MR-SLAM. Yamauchi [19] introduced frontier-based exploration as a practical method for guiding a robot toward unexplored regions, and later work improved its efficiency for occupancy-grid-based graph SLAM [20]. These methods are effective for autonomous coverage, but they do not solve map merging on their own. Therefore, they are best combined with pose-graph optimization and loop closure validation rather than used as a complete collaborative mapping solution.

Overall, the literature shows a clear trade-off between accuracy, scalability, and communication cost in multi-robot SLAM (Table 1). Centralized methods generally provide stronger global map consistency, but they rely on communication infrastructure and shared processing resources. In contrast, decentralized methods are more scalable and robust to failures, although they usually require more complex coordination between robots. Existing studies also show that reliable place recognition, efficient graph optimization, and effective exploration strategies each improve different parts of the mapping process, but no single approach resolves all of these challenges equally well. Therefore, the reviewed works support the use of independently exploring robots with centralized map fusion and selective retention of informative constraints for labyrinth mapping.

While the individual components of centralized pose-graph optimization and loop closure detection are well-documented in existing literature, the novelty of this study lies in the synergistic integration of spectral graph sparsification with algebraic connectivity-based prioritization specifically within a centralized multi-robot framework for labyrinthine environments. Unlike standard C-SLAM implementations that may struggle with redundant or noisy constraints in repetitive corridors, our approach utilizes effective resistance sampling to preserve the spectral properties of the global pose graph. This ensures that only the most "structurally informative" loop closures are communicated and optimized. By combining this with a two-stage verification process, the system achieves a level of robustness to odometry drift that exceeds traditional Bayesian fusion, providing an efficient end-to-end pipeline for low-cost robotic platforms in constrained indoor spaces.

While the individual components of this system, such as frontier exploration, pose - graph optimization, and descriptor matching, are well - established, the core scientific contribution of this work is the novel synthesis of algebraic connectivity-based sparsification within a centralized multi-robot pipeline specifically optimized for labyrinthine environments. Unlike existing C-SLAM frameworks that treat all constraints equally, our approach leverages spectral graph properties to prioritize loop closures that most significantly improve global map rigidity. This specific integration addresses the 'perceptual aliasing' problem inherent in repetitive labyrinth corridors, providing a communication-efficient method for achieving global consistency without the computational overhead of full graph optimization.

Table 1. Classification of Multi-Robot SLAM Methods

Method	Framework	Optimization	Advantages	Disadvantages
General MR-SLAM Challenges [1]	–	–	Faster mapping via collaboration	Requires solving relative pose estimation, synchronization, and loop closure
Multi-robot Pose Graph SLAM (TSLAM) [5]	Centralized	Global optimization (g2o)	High accuracy, network delay tolerance, trust factor for smooth transitions	Requires reliable communication infrastructure, central workstation dependence
Karto SLAM [6]	Centralized	Centralized graph optimization	High mapping accuracy, integrates navigation (DWA)	Poor adaptability to decentralized or low-cost systems, needs stable communication
Swarm SLAM [8]	Decentralized	Distributed, no central optimization	Fault tolerance, suitable for unreliable networks, scalable exploration	Lower map precision, simplified environment representation
Sparse Decentralized Collaborative SLAM (C-SLAM) [9]	Decentralized	Distributed Gauss-Seidel (DGS) optimization	Communication-efficient through sparsification, suitable for large-scale scenarios	Complex loop closure management, depends on inter-robot cooperation
RTAB-Map [21]	Semi-centralized	Local optimization with memory management	Supports multiple sensors, good scalability with large datasets	Performance highly dependent on sensor choice and tuning

2. Methods and Materials

This study develops and evaluates a collaborative multi-robot SLAM system for labyrinth mapping using two low-cost TurtleBot platforms. The proposed approach is based on independent local exploration by each robot and centralized map fusion on a server. In contrast to direct map merging, the system first estimates spatial relationships between the robots through loop closure detection and pose graph alignment, and only then combines the local maps into a single global representation. This design is intended to improve global consistency while preserving the practical advantages of parallel exploration.

2.1 Study design and objective

The main purpose of this study is to investigate whether a multi-robot SLAM system can map a labyrinth environment more effectively than a single-robot system while maintaining structural consistency in the final map. For this reason, the experimental design includes both a single-robot baseline and a collaborative two-robot scenario.

Each mobile operates as an independent SLAM agent equipped with a 2D LiDAR sensor and wheel odometry. During exploration, each robot builds its own local map and trajectory estimate. Instead of assuming a known relative position between robots, the proposed system allows the robots to begin exploration independently and determines map alignment later through centralized processing. This is important because unknown relative initialization is one of the main practical difficulties in collaborative SLAM.

The system utilizes a hybrid SLAM architecture. Each TurtleBot employs a feature-based EKF-SLAM as a local front-end. The EKF-SLAM handles high-frequency state estimation x_k by integrating wheel odometry and LiDAR-derived landmarks in real-time. Periodically, the EKF-SLAM outputs 'key-poses' and their associated local landmark maps to a centralized pose-graph back-end. On the server, these poses become nodes in a global graph, where edges represent odometry constraints and inter-robot loop closures. This separation allows the robots to remain computationally lightweight while the server performs global non-linear optimization using the g2o framework to correct accumulated drift.

The approach is built upon a feature-based EKF-SLAM framework to estimate the robot's trajectory and the locations of observed landmarks simultaneously. The system state at time k is represented by the vector x_k :

$$x_k = [x_r, y_r, \theta_r, m_1, m_2, \dots, m_n]^T \quad (1)$$

where (x_r, y_r, θ_r) denotes the robot's pose and m_i represents the coordinates of the i -th landmark. When a landmark is detected, the measurement z_k (typically range and bearing) is modeled by the observation function h :

$$z_k = h(x_k) + v_k \quad (2)$$

where v_k represents measurement noise. The difference between the actual observation and the predicted observation y_k is calculated as $y_k = z_k - h(x_k)$. The overall workflow of the proposed collaborative multi-robot SLAM system is illustrated in Figure 1.

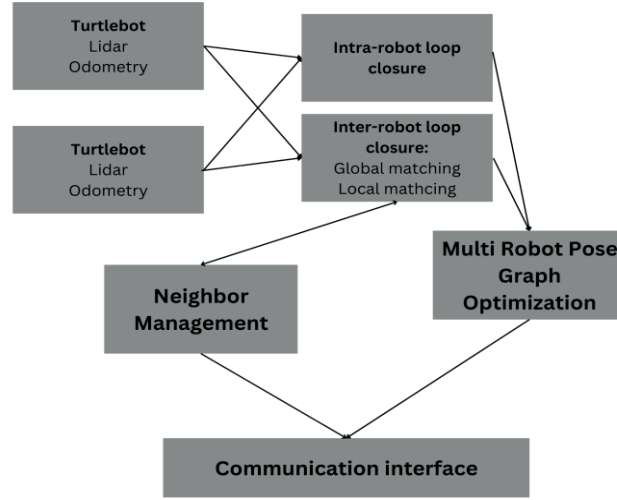


Figure 1. Processing pipeline of the proposed centralized collaborative multi-robot SLAM system [8].

The mathematical description of motion trajectories and the state space are following. We define the robot's state at time t as $q(t) \in C$, where C is the configuration space. For a mobile robot in a 2D environment, $q(t) = [x(t), y(t), \theta(t)]^T$. A motion trajectory T is defined as a time-indexed sequence of states over the interval $[0, T]$:

$$T = \{q(t) \mid t \in [0, T]\}$$

In practice, the trajectory is discretized into N steps with a sampling interval Δt , denoted as

$$Q = \{q_0, q_1, \dots, q_n\}.$$

The evolution of the trajectory is governed by the non-linear discrete-time kinematic model:

$$q_{k+1} = f(q_k, u_k) + w_k$$

Where $u_k = [v_k, \omega_k]^T$ represents control inputs and $w_k \sim N(0, Q_k)$. For a differential drive system:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} v_k \cos(\theta_k) \Delta t \\ v_k \sin(\theta_k) \Delta t \\ \omega_k \Delta t \end{bmatrix}$$

2.2 Proposed collaborative SLAM method

The proposed method follows a centralized collaborative SLAM architecture inspired by earlier pose-graph-based multi-robot approaches [5], while incorporating communication-aware constraint selection during map fusion [9, 15, 16]. As shown in Figure 1, the proposed method consists of sequential local and centralized stages. Each robot performs local SLAM independently using its onboard LiDAR and odometry data and generates a local occupancy map together with

its local pose trajectory. During this stage, the robots explore the labyrinth autonomously using frontier-based exploration, where new targets are chosen at the boundary between explored and unexplored space [19-20]. Because the robots start from different initial positions, they are able to cover different parts of the environment in parallel, reducing redundant traversal.

To combine the independently generated maps, the centralized server searches for overlap between the robot trajectories through inter-robot loop closure detection. This process is performed in two stages. First, compact global descriptors derived from LiDAR observations are compared in order to identify candidate matches efficiently [12]. Second, these candidate matches are geometrically verified using more detailed local information so that false correspondences can be rejected before map merging. This two-stage procedure is particularly important in labyrinth environments, where similar-looking corridors may otherwise produce incorrect matches.

After loop closures have been verified, the centralized server aggregates odometry constraints, local loop closures, and inter-robot loop closures into a single global pose graph. The graph is then optimized using the g2o framework [13], which refines the relative positions of both robots in a common frame. To reduce unnecessary communication and optimization cost, the system applies graph sparsification before final map fusion, retaining only the most informative constraints for reliable global alignment [14-16]. Once the optimized global poses are obtained, the local maps are transformed into the shared reference frame and fused into a single map of the labyrinth.

2.3 Evaluation approach

The proposed method is evaluated through comparison of single-robot and two-robot mapping performance. The assessment focuses on whether the two robots can explore the labyrinth independently, whether overlap between their trajectories can be detected reliably, whether centralized pose graph optimization produces a coherent merged map, and whether sparsification reduces graph complexity without degrading the structural quality of the final result.

Accordingly, the study aims to demonstrate three main outcomes. First, collaborative exploration should improve practical coverage of the environment by allowing both robots to operate in parallel. Second, centralized map fusion should reduce the inconsistency that would otherwise appear if local maps were merged directly from uncertain initial alignment. Third, selective retention of informative graph constraints should make the system more suitable for bandwidth-constrained operation without sacrificing map quality. The following performance metrics exist:

- Trajectory Accuracy: Measured via Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) against ground-truth data from the Gazebo environment.
- Map Fidelity: Quantified using the Root Mean Square Error (RMSE) of occupancy grid alignment between the merged multi-robot map and the single-robot baseline.
- Sparsification Efficiency: Evaluated through the Graph Compression Ratio, defined as the number of edges in the sparsified graph E_s versus the original graph E_o .
- Exploration Productivity: Measured as the Area Coverage Rate m^2/s , comparing the time required for two robots to reach 95% environmental coverage versus a single-robot deployment.

3. Results and Discussion

The experimental validation was carried out in two stages: simulation in Gazebo and testing in a real labyrinth environment. The purpose of the experiments was to assess whether the proposed collaborative SLAM pipeline could produce a structurally consistent merged map under both controlled and real operating conditions. The evaluation therefore focused on qualitative comparison of map consistency, including alignment of overlapping regions, continuity of walls and corridors, absence of duplicated structures, and robustness to odometry drift.

Three configurations were considered in the comparison. The first was a single-robot SLAM baseline, used as a reference for the expected structure of the map. The second was a two-robot Bayesian fusion baseline, in which the maps were merged directly using an estimated relative pose without pose graph optimization or inter-robot loop closure validation. The third was the

proposed collaborative multi-robot SLAM method, in which local maps were aligned through inter-robot loop closure detection, pose graph optimization, and graph sparsification before final map fusion.

3.1 Design and implementation results in Simulation

To begin with, we implemented an SLAM mapping algorithm in the Gazebo environment for one TurtleBot, enabling autonomous exploration and map generation using its laser scanner and odometry data. Utilizing the files we already had from one turtlebot, we created custom launch files that initialize two independent SLAM nodes and deployed two TurtleBots in Gazebo, each beginning SLAM from different starting positions on the same map (Figure 2).

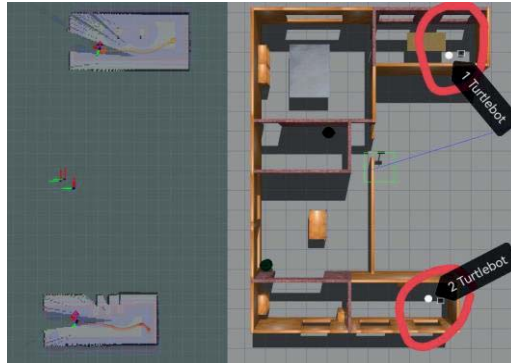


Figure 2. Exploration initialization of two turtlebots

The exploration method was based on key mathematical tools integrated through the `slam_toolbox` library, which provided functions for mapping, localization, coordinate transformations, and navigation. Using these built-in capabilities, the system identified unexplored regions, planned paths, and performed autonomous exploration by leveraging occupancy grids, gradient-based frontier detection, and frame transformations. This approach follows the classical frontier-based exploration method introduced by Yamauchi [19], which enables robots to systematically expand map coverage by selecting goal points at unexplored boundaries. The environment is represented as a 2D occupancy grid with values 0 (free), 100 (obstacle), and -1 (unknown). Frontiers between explored and unexplored areas are detected using image gradients computed with Sobel kernels, which are a simple mathematical matrix designed to highlight edges (transitions) in an image:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (3)$$

These kernels approximate the partial derivatives $\frac{\partial I}{\partial x}$ and $\frac{\partial I}{\partial y}$, where $I(x, y)$ represents the grid. The gradient magnitude, $\sqrt{I_x^2 + I_y^2}$, shows the strength of transitions, and the gradient direction, $\left(\frac{I_y}{I_x}\right)$, provides the orientation of these changes. High-gradient regions indicate transitions between unexplored and free areas, which are ideal candidate locations for the robot to explore next. To navigate toward frontiers, poses are expressed in homogeneous coordinates $p = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$ and transformed using a matrix $T = \begin{bmatrix} \cos \theta & -\sin \theta & t_x \\ \sin \theta & \cos \theta & t_y \\ 0 & 0 & 1 \end{bmatrix}$. This transformation allows the robot to compute the target pose in the global map frame while accounting for its own position. A global costmap removes unsafe or obstructed goals. Combining frontier detection with these transformations enables systematic exploration. The use of gradient-based frontier detection

provides an efficient approximation of frontier extraction commonly employed in frontier-based exploration frameworks [19].

Two TurtleBots were initialized at separate spawn points (Figure 2) to reduce early overlap and support collaborative mapping. Their movements, which were recorded through odometry (Figures 3-4), showed individual trajectories, coverage efficiency, and obstacle avoidance. The final maps from both robots aligned with the single-robot baseline.

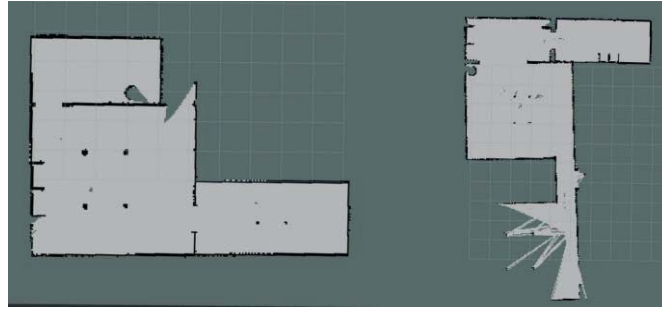


Figure 3. Exploration map of two turtlebots

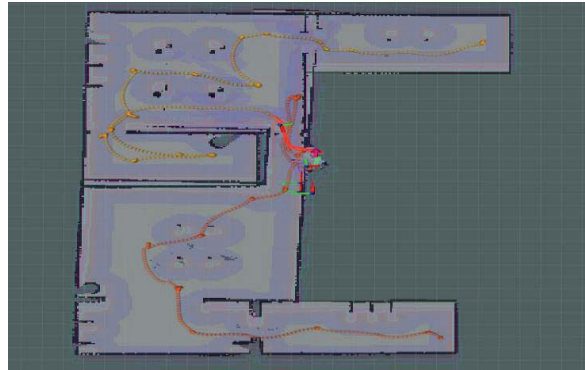


Figure 4. Exploration completion and map merging for two turtlebots

We established a two-TurtleBot SLAM system using Gazebo, RViz, and a C-SLAM-inspired implementation. Each robot built its own map, which was merged either through known initial poses or feature-based alignment. LIDAR and odometry data were also combined into point clouds to improve mapping accuracy. Refinement then focused on improving alignment robustness to avoid map overlap, robot collisions, or inconsistent global frames. Map merging accuracy is critical in multi-robot SLAM, as misalignment leads to navigation errors. In this regard, we tested multiple configurations to improve consistency and evaluate merging strategies. The system was optimized to ensure stable autonomous mapping in complex environments.

Before exchanging pose graphs, they were sparsified to reduce communication load using effective resistance. A unit current was placed across each edge to measure its contribution to connectivity. Edges with higher effective resistance had a higher sampling probability [14]. Selected edges were then reweighted to preserve the spectral structure of the original graph (Figure 5).

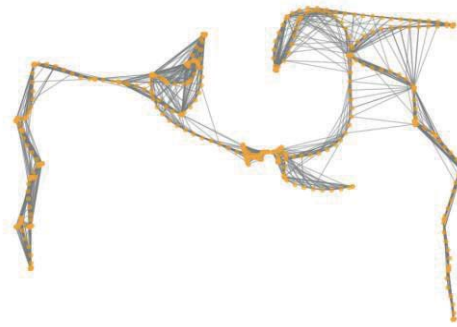


Figure 5. Algebraic connectivity maximization

Moreover, the use of effective resistance naturally emphasized the preservation of long-range constraints and structurally critical links in the pose graph, which were often vital to ensuring global consistency across multiple robot trajectories. The sparsified pose graphs were transmitted to the centralized server, where global consistency was enforced through pose graph optimization. Figures 6 and 7 demonstrate the graph sparsification by effective resistances, and Figure 8 shows the pose graphs overlayed on top of the map in Gazebo so that intra- and inter-robot loop closures are easy to identify. Effective resistance naturally preserves long-range constraints crucial for global consistency in multi-robot trajectories [18]. This approach preserved long-range constraints while reducing computational and communication overhead.

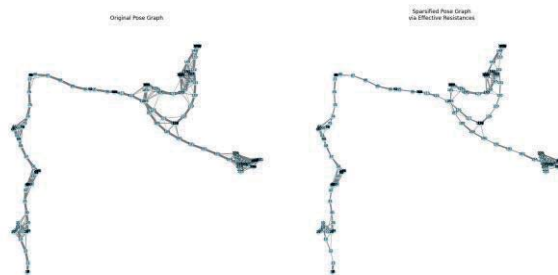


Figure 6. Turtlebot 1 graph sparsification

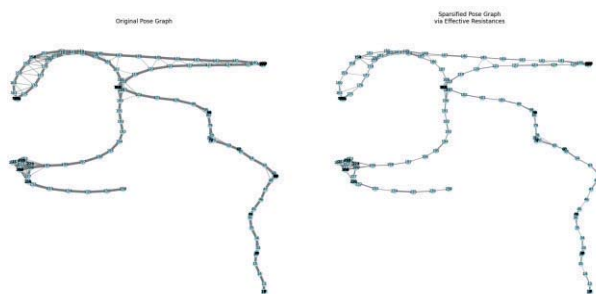


Figure 7. Turtlebot 2 graph sparsification

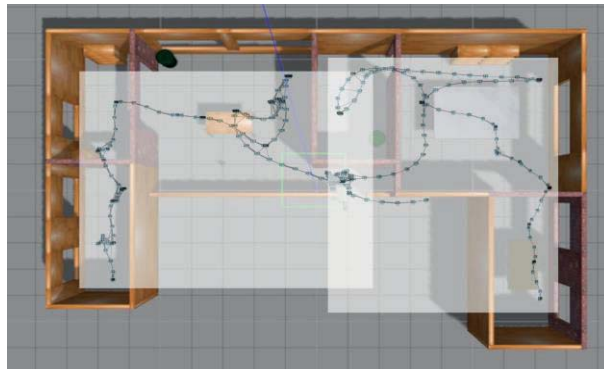


Figure 8. Pose graphs of 2 Turtlebots overlaid on top of the map

3.2 Testing and evaluation of Real-world results

The experimental evaluation of our multi-robot SLAM system was carried out first in simulation and later in a real physical labyrinth. In simulation, each robot generated its own local map independently, while mapping data were transmitted to a centralized server for fusion when communication was available. After validating the approach in Gazebo, we applied the same system in a real labyrinth constructed for testing (Figure 9). The real environment introduced additional challenges, such as sensor noise, lighting variations, and irregular physical obstacles that were not present in the simulation.

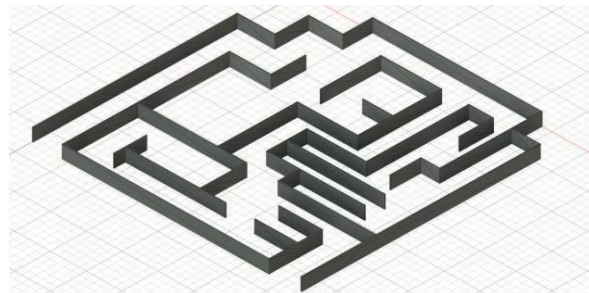


Figure 9. 3D model of the labyrinth in the lab

Real-world testing was carried out with two TurtleBots, and their generated map can be seen in Figure 10. The robots performed autonomous exploration using occupancy-grid gradient analysis, where grid cells were assigned values 0 (free), 100 (obstacle), and -1 (unknown). Sobel gradient operators were used to detect frontiers separating explored and unexplored regions, enabling efficient goal selection and path planning throughout the maze.

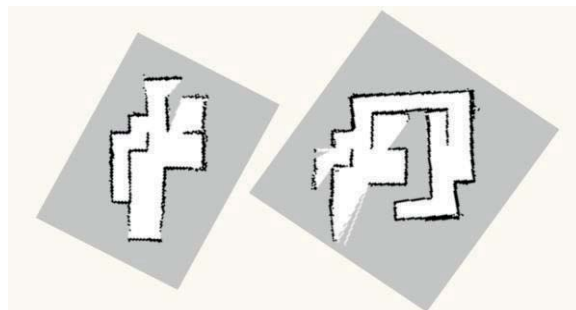


Figure 10. Map received from the experiment

Inter-robot loop closure candidates were identified through descriptor matching (Figure 11), ensuring that overlapping areas between the robots' trajectories were consistently aligned.

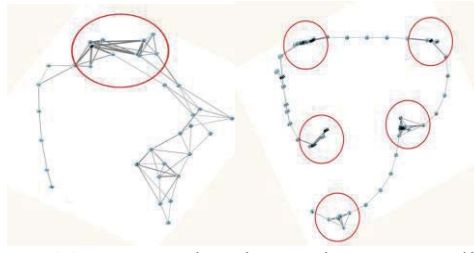


Figure 11. Inter-robot loop closure candidates

Figure 12 shows the sparsified pose graphs produced via effective resistance sampling. Each edge was assigned a sampling probability proportional to its contribution to global connectivity, allowing the system to retain essential loop closures while greatly reducing communication overhead.

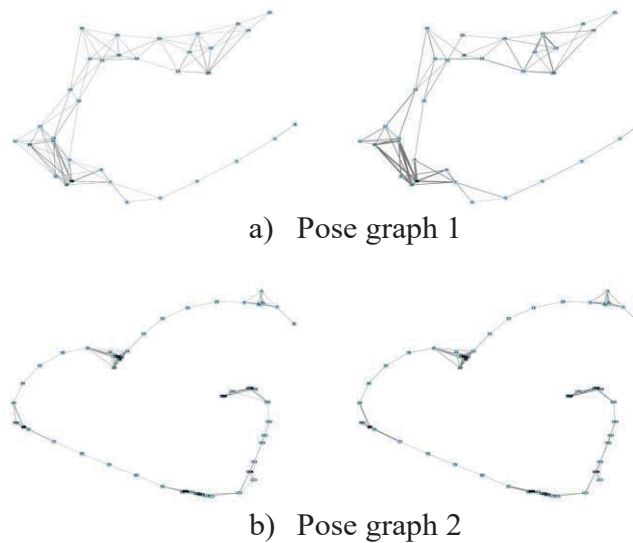


Figure 12. Sparsified pose graphs

Then, pose graph merging through algebraic connectivity maximization produced accurate relative transformations between the two robots, allowing their individual maps to merge into a single coherent global map (Figure 13). By using all available inter- and intra-robot loop closures rather than relying on initial pose estimates, the method achieved consistent alignment even under real-world sensor noise.

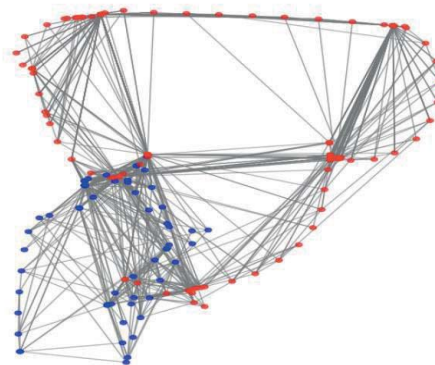
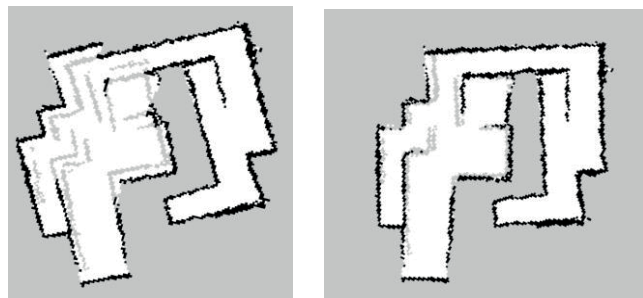


Figure 13. Algebraic connectivity maximization

3.3 Comparison with baseline methods

When the two maps were merged, significant differences appeared between our method and the traditional Bayesian fusion approach. The Bayesian fusion baseline refers to direct occupancy grid merging based on an estimated or known relative pose between robots, without performing pose graph optimization or inter-robot loop closure detection. In this approach, grid cell probabilities are combined using standard Bayesian update rules, assuming that the initial alignment between maps is sufficiently accurate. As a result, accumulated odometry drift is not explicitly corrected during map merging. Figure 14 (a) shows a snippet of the Bayesian fusion process, which relies heavily on initial pose estimates; this dependency resulted in noticeable distortions when odometry drift accumulated, both in simulation and in real experiments (Figure 15). In contrast, our merging pipeline first aligns the robots' maps through centralized pose graph merging using algebraic connectivity maximization, and only then fuses the occupancy grids. This allowed us to compensate for drift and sensor noise, producing a consistent global map, which can be seen in Figure 14 (b). The two-stage loop closure detection procedure, global descriptor matching followed by geometric registration, ensured reliable estimation of inter-robot constraints while keeping the computational load low.

Overall, our experimental results confirm the hypothesis that multi-robot SLAM can map complex environments more efficiently than single-robot approaches. The proposed multi-robot SLAM system with centralized fusion and effective graph sparsification proved particularly suitable for bandwidth-constrained scenarios, enabling robots to share only the most informative data while maintaining mapping accuracy.



a) Map from Bayesian Fusion b) Map from Algebraic connectivity

Figure 14. Comparison of resulting maps



Figure 15. Inconsistency resulted from the Bayesian fusion method

4. Conclusion

This work demonstrates that a multi-robot SLAM system with independently operating robots and centralized map fusion, implemented using a C-SLAM-inspired framework, is a feasible and structurally consistent alternative to single-robot mapping in complex labyrinth environments. Across both simulated and real-world trials, the two TurtleBots explored the labyrinth in parallel, and their locally built maps were successfully merged into a single coherent global representation, qualitatively supporting the hypothesis that multi-robot collaboration can improve mapping efficiency without sacrificing structural consistency. Because the trajectory-accuracy, map-fidelity, and exploration-time metrics defined in Section 2.3 were not measured in

this study, this efficiency and accuracy advantage has been demonstrated only qualitatively and has not yet been statistically validated; quantifying it is identified below as a priority for future work. The proposed two-stage loop closure detection process proved effective for identifying inter-robot correspondences, addressing one of the major challenges in multi-robot SLAM. Pose graph sparsification using effective resistances visibly reduced the number of retained constraints (Figures 6, 7, and 12) while preserving the structural properties required for reliable optimization, consistent with the theoretical guarantees reported by Spielman and Srivastava [14] and Doherty et al. [16]. Centralized pose graph optimization produced visibly more consistent global maps than direct Bayesian fusion under sensor noise and odometry drift (Figures 14–15), although, as discussed in Section 3.4, this comparison was likewise qualitative rather than metric-based. Gradient-based frontier detection supported autonomous exploration of the labyrinth, and the map merging pipeline produced coherent global maps even when robots began from unknown relative positions. Two key limitations should be noted. First, no quantitative performance metrics (ATE, RPE, RMSE, compression ratio, or coverage rate, as defined in Section 2.3) were collected in either simulation or the physical experiment; the comparisons in Section 3 are therefore based on visual and structural inspection of the resulting maps rather than statistical measures, and the statements above regarding speed and accuracy should be interpreted as feasibility observations rather than validated performance gains. Second, the controlled testing environment limits generalizability: while the labyrinth introduces structural complexity, it does not fully capture the variability of unstructured or dynamic environments, and computational cost is expected to increase with environmental size and complexity, so real-time performance at larger scale may require further optimization.

The results in Sections 3.1–3.3 indicate that centralized pose graph optimization combined with algebraic-connectivity-based sparsification produced more structurally consistent merged maps than direct Bayesian occupancy-grid fusion, both in simulation and in the physical labyrinth. This outcome is consistent with the underlying theory rather than incidental: because Bayesian fusion combines occupancy grids under the assumption that the estimated relative pose between robots is already correct, any accumulated odometry drift is propagated directly into the fused map, which explains the distortions visible in Figure 15. In contrast, the proposed pipeline first resolves the relative transformation between robots through pose graph optimization constrained by verified inter-robot loop closures, so odometry drift is corrected before the occupancy grids are combined. This mirrors the rationale of centralized pose-graph SLAM frameworks such as Deutsch et al. [5], and helps explain why our merged maps (Figure 14b) show fewer discontinuities than the Bayesian baseline (Figure 14a), even though both approaches used the same underlying odometry and LiDAR data.

The benefit of graph sparsification observed in Figures 6, 7, and 12 is likewise explainable from the effective-resistance sampling criterion itself: edges with high effective resistance correspond to constraints that carry disproportionate influence on the graph's algebraic connectivity, so preferentially retaining them, as proposed by Spielman and Srivastava [14] and later adapted for pose-graph SLAM by Doherty et al. [15–16], preserves the spectral structure of the graph while discarding redundant, low-information constraints. This is consistent with the visual observation that the sparsified graphs in this study retained long corridor-spanning loop closures rather than short, locally redundant ones, and it explains why sparsification did not visibly degrade the coherence of the final merged maps.

The two-stage loop closure procedure, coarse descriptor matching followed by geometric verification, addresses the perceptual aliasing risk that is specific to labyrinth-like environments, where repeating corridors and intersections are geometrically similar and can easily be confused by descriptor matching alone [2, 12]. Descriptor matching without a verification stage would be expected to accept some of these false candidates as genuine inter-robot loop closures, corrupting the global pose graph; requiring geometric agreement between candidate matches before they are

added to the graph is therefore an important factor in why no severe map corruption from incorrect loop closures was observed during either the simulated or real-world trials.

Comparing the simulated and physical experiments also clarifies the practical limits of the approach. The transition from Gazebo to the physical labyrinth introduced sensor noise, lighting variation, and irregular obstacles (Section 3.2) that are absent from simulation, and the physical results (Figures 9–13) show visibly less regular corridor geometry than the simulated maps (Figures 2–8), even though both were successfully merged. This suggests that the system's robustness to real-world sensory imperfection, while sufficient for a two-robot labyrinth of this size, has not been characterized under harsher conditions (e.g., low-feature corridors, moving obstacles, or degraded LiDAR returns), consistent with the scalability concerns raised in Section 4 regarding unstructured or dynamic environments.

Overall, the outcomes of this work show that multi-robot SLAM with independent robot operation and centralized fusion is a strong alternative to single-robot approaches for mapping constrained environments. The methods developed here form a solid basis for extending multi-robot SLAM into more demanding applications.

Although this study focuses on a two-robot deployment in a controlled labyrinth, the underlying architecture is designed with theoretical scalability in mind. The use of effective resistance for graph sparsification is a key enabler for larger swarms $N > 2$, as it prevents the centralized server from becoming a computational bottleneck by selectively filtering the influx of pose-graph constraints. However, scaling to larger or more unstructured environments would introduce new challenges:

1. **Communication Latency:** A centralized approach relies heavily on network stability. As N increases, the bandwidth requirements for the central server may necessitate a transition to a "hierarchically centralized" or fully distributed model.

2. **Environmental Complexity:** In unstructured or dynamic settings (unlike the rigid walls of our labyrinth), the frontier-based exploration and descriptor matching would require more robust feature sets to avoid false inter-robot loop closures.

Future work will focus on testing the system's resilience in "open-world" scenarios with varying robot counts to identify the exact point of diminishing returns in centralized fusion

Acknowledgement This research was funded by the Science Committee of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No. AP23485994)

References

- [1] Saeedi, S., Trentini, M., Seto, M., & Li, H. (2015). Multiple-Robot Simultaneous Localization and Mapping: A review. *Journal of Field Robotics*, 33(1), 3–46. <https://doi.org/10.1002/rob.21620>
- [2] Zhou, R., He, L., Zhang, H., Lin, X., & Guan, Y. (2022). NDD: a 3D point cloud descriptor based on normal distribution for loop closure detection. 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1328–1335. <https://doi.org/10.1109/iros47612.2022.9981180>
- [3] Tian, Y., Chang, Y., Arias, F. H., Nieto-Granda, C., How, J. P., & Carlone, L. (2022). Kimera-Multi: Robust, Distributed, dense Metric-Semantic SLAM for Multi-Robot systems. *IEEE Transactions on Robotics*, 38(4), 2022–2038. <https://doi.org/10.1109/tro.2021.3137751>
- [4] McGann, D., Rogers, J.G., & Kaess, M. (2022). Robust Incremental Smoothing and Mapping (riSAM). 2023 IEEE International Conference on Robotics and Automation (ICRA), 4157–4163. <https://doi.org/10.48550/arXiv.2209.14359>
- [5] Deutsch, I., Liu, M., & Siegwart, R. (2016). A framework for multi-robot pose Graph SLAM. 2016 IEEE International Conference on Real-time Computing and Robotics (RCAR), 567–572. <https://doi.org/10.1109/rcar.2016.7784092>

- [6] Zhao, J., Liu, S., & Li, J. (2022). Research and Implementation of Autonomous Navigation for Mobile Robots Based on SLAM Algorithm under ROS. *Sensors*, 22(11), 4172. <https://doi.org/10.3390/s22114172>
- [7] Li, B., Mi, Z., Guo, Y., Yang, Y., & Obaidat, M. S. (2019, May 30). Partial computing offloading assisted cloud point registration in multi-robot SLAM. *arXiv.org*. <https://arxiv.org/abs/1905.12973>
- [8] Kegeleirs, M., Grisetti, G., & Birattari, M. (2021). Swarm SLAM: Challenges and Perspectives. *Frontiers in Robotics and AI*, 8, 618268. <https://doi.org/10.3389/frobt.2021.618268>
- [9] Lajoie, P., & Beltrame, G. (2023). Swarm-SLAM: Sparse decentralized collaborative simultaneous localization and mapping framework for Multi-Robot systems. *IEEE Robotics and Automation Letters*, 9(1), 475–482. <https://doi.org/10.1109/lra.2023.3333742>
- [10] Zhong, S., Qi, Y., Chen, Z., Wu, J., Chen, H., & Liu, M. (2023). DCL-SLAM: a distributed collaborative LiDAR SLAM framework for a robotic swarm. *IEEE Sensors Journal*, 24(4), 4786–4797. <https://doi.org/10.1109/jsen.2023.3345541>
- [11] Kim, H., Choi, J., Kim, J., Yang, G., Cho, D., Lim, H., & Cho, Y. (2025, May 13). SKID-SLAM: Robust, lightweight, and distributed Multi-Robot LIDAR SLAM in Resource-Constrained field environments. *arXiv.org*. <https://arxiv.org/abs/2505.08230>
- [12] Kim, G., & Kim, A. (2018). Scan Context: Egocentric spatial descriptor for place recognition within 3D Point Cloud Map. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 4802–4809. <https://doi.org/10.1109/iros.2018.8593953>
- [13] Kummerle, R., Grisetti, G., Strasdat, H., Konolige, K., & Burgard, W. (2011). G2O: A General Framework for graph Optimization. 2011 IEEE International Conference on Robotics and Automation, 3607–3613. <https://doi.org/10.1109/icra.2011.5979949>
- [14] Spielman, D. A., & Srivastava, N. (2011). Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6), 1913–1926. <https://doi.org/10.1137/080734029>
- [15] Doherty, K. J., Rosen, D. M., & Leonard, J. J. (2022). Spectral Measurement sparsification for Pose-Graph SLAM. 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 01–08. <https://doi.org/10.1109/iros47612.2022.9981584>
- [16] Doherty, K., Papalia, A., Huang, Y., Rosen, D., Englot, B., & Leonard, J. (2024, March 28). MAC: Graph sparsification by maximizing algebraic connectivity. *arXiv.org*. <https://arxiv.org/abs/2403.19879>
- [17] Wickman, R., Zhang, X., & Li, W. (2022). A Generic Graph Sparsification Framework using Deep Reinforcement Learning. 2022 IEEE International Conference on Data Mining (ICDM), 1221–1226. <https://doi.org/10.1109/icdm54844.2022.00158>
- [18] Zhang, Y., Tosi, F., Mottocchia, S., & Poggi, M. (2023). GO-SLAM: Global Optimization for Consistent 3D Instant Reconstruction. 2023 IEEE/CVF International Conference on Computer Vision (ICCV), 3704–3714. <https://doi.org/10.1109/iccv51070.2023.00345>
- [19] Yamauchi, B. (1997). A frontier-based approach for autonomous exploration. *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97*, 146–151. <https://doi.org/10.1109/cira.1997.613851>
- [20] Orsulic, J., Miklic, D., & Kovacic, Z. (2019). Efficient dense frontier detection for 2-D graph SLAM based on occupancy grid submaps. *IEEE Robotics and Automation Letters*, 4(4), 3569–3576. <https://doi.org/10.1109/lra.2019.2928203>
- [21] Labbé, M., & Michaud, F. (2018). RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of Field Robotics*, 36(2), 416–446. <https://doi.org/10.1002/rob.21831>