Iraqi Journal for Computer Science and Mathematics

Volume 5 | Issue 4 Article 24

2024

Enhancing Multi-Robot SLAM: Centralized LiDAR-Based Loop Closure Detection Approach

Basma Ahmed Jalil
Department of Computer Engineering, Faculty of Engineering, Mosul University, Mosul 41002, Iraq,
basma.a.jalil@gmail.com

Ibraheem Kasim Ibraheem
Department of Electrical Engineering, College of Engineering, University of Baghdad, Baghdad 10001, Iraq AND Department of Electronics and Communication Engineering, Uruk University, Baghdad, Iraq,
ibraheem.i.k@ieee.org

Recommended Citation

Jalil, Basma Ahmed and Ibraheem, Ibraheem Kasim (2024) "Enhancing Multi-Robot SLAM: Centralized LiDAR-Based Loop Closure Detection Approach," *Iraqi Journal for Computer Science and Mathematics*: Vol. 5: Iss. 4, Article 24.
DOI: https://doi.org/10.52866/2788-7421.1219
Available at: https://ijcsm.researchcommons.org/ijcsm/vol5/iss4/24

IRAQI JOURNAL FOR COMPUTER SCIENCE AND MATHEMATICS 2025;5:310–325

RESEARCH ARTICLE

Enhancing Multi-Robot SLAM: Centralized LiDAR-Based Loop Closure Detection Approach

Basma Ahmed Jalil [a,*], **Ibraheem Kasim Ibraheem** [b,c]

[a] Department of Computer Engineering, Faculty of Engineering, Mosul University, Mosul 41002, Iraq
[b] Department of Electrical Engineering, College of Engineering, University of Baghdad, Baghdad 10001, Iraq
[c] Department of Electronics and Communication Engineering, Uruk University, Baghdad, Iraq

ABSTRACT

The loop closure detection is crucial for global mapping and route correction in multi-robot simultaneous localization and mapping (SLAM). However, including loop closure detection algorithms in MR-SLAM increases the computational complexity and the required resources on the robot board and at the base station. In this paper, An Enhanced Multi-Robot Fast Localization Odometry and Mapping (EMR-FLOAM) to deal with computation complexity issue. The EMR-FLOAM algorithm addresses computational complexity and resource requirements by utilizing a two-stage non-iterative distortion compensation technique, resulting in optimized code and accelerated localization and map construction processes. The simulation of the proposed work has been tested on the outdoor dataset tailored for multi-robot systems and the indoor environment constructed by the Gazebo simulator. The simulation results have been compared with Enhanced Multi Robot An improved Localization odometry and Mapping (EMR-ALOAM) algorithm and it was noted that EMR-FLOAM shows an enhancement in reducing the error drift, but the error is still larger than that of EMR-ALOAM, while the computational complexity in EMR-FLOAM is smaller than that of EMR-ALOAM.

Keywords: Scan Context descriptor, Centralized SLAM, Loop closure detection, LiDAR, Multi-Robot SLAM (MR-SLAM)

1. Introduction

Odometry data integration is essential for accurate route prediction and global map development in the dynamic field of SLAM. Odometry tracks movement, but loop closure solves drift by recognizing previously studied terrain. Lengthy procedures need this since even slight error can substantially change a robot's course [1].

In Multi Robot Simultaneous Localization and Mapping (MR-SLAM) scenario, robots collaborate to comprehend the environment and construct an accurate global map to achieve the specified task. Consequently, it is crucial to address the revisited area information acquired by either a single robot or multiple robots to rectify the accumulated errors resulting from the multi-robot operation. This correction is essential for aligning the local maps from individual robots, ensuring the creation of a consistent, drift-free global map [2]. This introduces two important concepts in multi-robot loop closure detection: intra-loop closure detection and inter-loop closure detection. Intra-loop closure detection concentrates on identifying loop closures within the trajectory of a single robot, aiming to minimize drift errors as the robot navigates through the scene. On the contrary, inter-loop closure detection is concerned with identifying loop closures across distinct robots. It entails locating areas of overlap between maps generated by different robots. This process facilitates map fusion, enabling the creation of a consistent global map that integrates information from various robots [3].

In recent times, 3D LiDAR sensors have become widely employed in MR-SLAM applications, spanning

Received 8 March 2024; accepted 28 April 2024.
Available online 3 January 2025

* Corresponding author.
E-mail addresses: basma.a.jalil@gmail.com (B. A. Jalil), ibraheem.i.k@ieee.org (I. K. Ibraheem).

https://doi.org/10.52866/2788-7421.1219

diverse fields like intelligent vehicles, warehouses, and mines detection. This is attributed to the sensor's capability to exhibit robustness against illumination changes and furnish high-resolution data across a broad volumetric field of view. These capabilities significantly enhance its effectiveness in providing a more accurate understanding of the environment, enabling seamless cooperation among robots to achieve tasks effectively.

This work is an extension of our prior research [4], which introduced a centralized MR-SLAM structure based on a LiDAR sensor. The earlier work encountered challenges with accumulated errors, causing deviations in robot trajectories and resulting in the construction of an inconsistent global map. In response, this study introduces a novel loop closure procedure aimed at improving the estimated trajectories of the robots and ensuring the creation of a more accurate global map. Additionally, a consideration has been given to computation time, recognizing its significance as a crucial factor in some applications. The proposed loop closure procedure consists of two essential components. The first part is executed directly on the robots, utilizing the ScanContext descriptor [5]. The second part, managed at the central station, involves identifying overlapping areas within the local map. This identification is achieved through the extraction of Fast Point Feature Histogram (FPFH) descriptors [6]. Following this, Singular-Value Decomposition (SVD) [7] and FAST_VGICP [8] are employed to determine the transformation within these overlapped areas.

2. Related work

In the context of MR-SLAM, the front-end operations, encompassing feature extraction and loop closure detection, and the back-end tasks involving optimization and estimation may not necessarily be carried out entirely on a single robot. This depends on the specific architecture employed in multi-robot systems [9]. Accordingly, we will review relevant literature concerning loop closure detection in MR-SLAM, considering the variations in system architecture.

Distributed Loop Closure Detection: In this architecture, individual robots perform feature extraction and share information for both intra- and inter-loop closure detection, as well as individual state estimation. An object-based module utilizing Persistent Feature Histogram (PFHRGB) [10] for inter-loop closure detection has been proposed in [11] employing the Distributed Gauss-Seidel (DGS) algorithm [12] for optimization. However, the challenge of storing numerous object models to account for intra-class variations exists. DiSCo-SLAM [13] addresses the efficient use of communication resources by employing compact ScanContext descriptors [14] for inter-loop closure detection. The Perspective Consistency Maximization (PCM) method [15] and a two-stage global and local optimization further improve accuracy. Swarm-SLAM [16] prioritizes sparse inter-robot loop closure based on algebraic connectivity maximization and uses optimization methods like Graduated Non-Convexity (GNC) and Georgia Tech Smoothing and Mapping (GTSAM). DCL-SLAM [17] employs iris descriptors[18] derived from LiDAR data, utilizing Hamming distance and KD-tree methods for inter-loop closure detection. Table 1 shows brief details for the above work, including the features utilized in inter- and intra-loop detection algorithms, the optimization method, the outlier rejection module, and the source code links. Despite the performance advantages of this MR-SLAM architecture, it still faces several challenges, including communication overhead, data fusion intricacies, consistency and synchronization issues, and algorithmic complexities.

Centralized Loop Closure Detection: In this method, the central station is responsible for inter-loop closure detection, and in some designs, it may also handle intra-loop closure detection, depending on the application or available resources. Detecting loop closures by finding overlapped areas between maps has gained popularity in centralized LiDAR-based MR-SLAM [20]. The place recognition module

Table 1. Brief details for the distrusted MR-SLAM.

Works	Intra-Loop features	Inter-Loop features	Optimization method	Outlier rejection	Source code
[19]	PFHRGB	PFHRGB	DGS	Not support	https://cognitiverobotics.github.io/distributed-mapper/
[13]	Corner and surface features	Scan context	DGS	PCM	https://github.com/RobustFieldAutonomyLab/DiSCo-SLAM
[16]	FPFH	Scan context	GNC and GTSAM	PCM	https://github.com/MISTLab/Swarm-SLAM
[17]	Edge and surface features	LiDAR Iris	DGS	PCM	https://github.com/PengYu-Team/DCL-SLAM

Table 2. Brief details for the centerlized MR-SLAM.

Works	Sensor	Local SLAM	Inter-Loop method	Outlier rejection	Optimization method	3D or 2D
[23]	LiDAR	Pose-Graph SLAM	Segment matching	No	iSAM2	3D
[24]	Camera	Pose-Graph SLAM	SURF feature matching	PCM	iSAM	2D
[25]	LiDAR, camera	LOAM SLAM	Scan matching	ICM	GTSAM	3D
[22]	LiDAR, camera	LeGO-LOAM	ORB feature matching	PCM	GTSAM	2D
[26]	LiDAR, camera	LOCUS or Hovermap [29]	Egin value-based matching	ICM	GTSAM + Leven-berg Marquard	3D
Our work	LiDAR	FLOAM	FPFH matching	Confidence score	GTSAM	3D

is utilized to detect the overlapped areas. Once the place recognition matches are determined, geometric estimation is performed to find the relative pose between the corresponding places. However, in LiDAR-based MR-SLAM, finding the overlapped area with 3D point clouds is a challenge due to the dense nature of the data and the lack of expressive features for place recognition. Compact global point cloud descriptors [21] have been developed to address the above challenge by providing a condensed representation of the point cloud data and enabling efficient comparison.

Another popular approach recently used for loop closure in this architecture involves extracting features from point clouds. These features are not only utilized in place recognition but also provide initial estimates for later geometric alignments, as seen in DARE-SLAM [22], which employs the Iterative Closest Point (ICP) algorithm [8]. However, ICP relies on good initial guesses, which are not readily available between the local maps of different robots in multi-robot operations. Therefore, using ICP directly may not be suitable in such scenarios. A place recognition method presented in [23] is based on a segment-matching algorithm. Each robot independently calculates the odometry factors, scan matching factors, and the segmented point cloud. All this information is then transmitted to the central station for the processes of place recognition and pose graph optimization.

The robustness of loop closure detection algorithms depends heavily on the robot's environment. In some cases, these algorithms may struggle to recognize reliable loop closure candidates. For example, when different places are incorrectly identified as the exact location, it can result in inaccurate loop closure detection, a phenomenon known as the perceptual aliasing problem. This issue has been addressed in [24], where a selection algorithm was introduced to filter out false-positive loop closure candidates and choose reliable loop closures for accurate map merging. Another example is that environments with long corridors and tunnels may experience perceptual degradation due to their symmetrical and featureless walls, a challenge addressed in LAMP [25], which uses Incremental Consistent Measurement (ICM). Derived works from [25], namely LAMP 2.0 [26] and [27] have been presented to tackle large-scale multi-robot SLAM in subterranean environments. In these works, the loop closure detection algorithm relies on the Graph Neural Network (GNN) [28] to predict unreliable loop closures.

Additionally, a prioritization system is employed to prioritize feature-rich loop closures and filter out feature-poor loop closure candidates. However, LAMP and LAMP 2.0 have not been tested for indoor environments such as warehouse applications and don't support 3D mapping. Table 2 provides concise details for some of the surveyed works for the centralized MR-SLAM, including the used sensor, local SLAM used approach for inter-loop closure detection, optimization method, outlier rejection module, source code link, and its support for 3D mapping.

Despite the large amount of research on MR-SLAM, it still suffers from some limitations, such as the need for a large amount of training data to recognize revisited places, computational complexity, and spatial distribution issues, as clarified in Table 3. For example, the work [1] suffers from training and computational complexity issues but does not suffer from the spatial distribution issue.

In this work, we propose a centralized LiDAR-based MR-SLAM system with a novel loop closure detection

Table 3. Some limitations in the reveiwed works.

Works	Required training	Computation complexity on the robot board	Spatial distribution issue
[19]	✓	✓	✘
[13]	✘	✓	✘
[17]	✘	✓	✘
[16]	✓	✓	✘
[23]	✓	✓	✓
[24]	✘	✓	✘
[25]	✘	✓	✓
[22]	✘	✓	✓
[26]	✘	✓	✓
Our works (EMR-FLOAM)	✘	✘	✘

procedure. The approach carefully addresses the computational complexity and time required to build a 3D consistent global map by distributing computations between the robots and the central station. Specifically, intra-loop closure detection is implemented on the robot itself using Scan Context descriptors. At the same time, a robust map fusion algorithm can identify overlapped areas (place recognition) between local maps at the central station, thereby preventing the need for an inter-loop closure detection algorithm.

This approach streamlines the fusion process by directly recognizing common regions in the local maps without explicitly relying on inter-loop closure detection. Table 2 provides brief details for the reviewed centralized MR-SLAM and our work, including the sensor used, local SLAM method, inter-loop closure approach, outlier rejection method, source code, and map style.

The contributions of this work are outlined as follows:

1. Development of a novel centralized MR-SLAM based solely on LiDAR sensor data, supporting the creation of 3D maps.

This contribution is significant as it addresses the need for efficient 3D mapping in centralized MR-SLAM. LiDAR-based SLAM is crucial for accurate 3D mapping, especially in applications like autonomous vehicles, where precise spatial information is required; providing a centralized MR-SLAM solution that relies on LiDAR data is valuable.

2. FLOAM SLAM features to address MR-SLAM speed and computational complexity issues.

This approach addresses a significant challenge in MR-SLAM: computational complexity. Implementing FLOAM SLAM methods in real-time MR-SLAM systems could improve efficiency by decreasing computational complexity and accelerating local map generation. Robotic applications need rapidity and effectiveness.

3. A pioneering technique is adopted to resolve the loop closure problem.

Loop closure detection is vital to making sure maps are accurate when robots try to figure out where they are and build a map at the same time (i.e., SLAM). Our new way does this by using robots that can scan their surroundings and an intelligent way to combine maps to recognize places they've seen before. You've pushed this technology forward by introducing a fresh approach to finding these loops with MR-SLAM.

This research looks at problems with MR-SLAM, like the difficulty of computations, creating 3D maps, and finding loop closures. Making progress in these areas could help make systems that map and locate using multiple robots work better in various uses.

3. Theoretical foundations

The following theoretical postulates underpin the suggested technique.

3.1. Registration and odometry estimation

Environmental resilience and location estimate accuracy are hallmarks of LiDAR-based SLAM. For 3D mapping, its three-dimensional viewpoint is essential. Traditional odometry estimation uses the Iterative Closest Point (ICP) approach to align scans using point cloud distance reduction. Processing several points makes this method computationally inefficient. Numerous examples include FLOAM [30], LeGO-LOAM [31], SA-LOAM [32], and more.

In our work, we utilize FLOAM for odometry estimation and intra-loop closure detection. FLOAM stands out due to its non-iterative two-stage distortion compensation method. It identifies edge and plane characteristics within point clouds based on their level of smoothness, where high smoothness indicates edges and low smoothness indicates planes.

$$Smoothness(\alpha) = \frac{1}{n, \|p_i\|} \left\| \sum_{j=1, j\neq i}^{n} (p_j - p_i) \right\| \quad (1)$$

In this context, p_i represents the target point, 'n' denotes the number of adjacent points, and p_j stands for the source point in the same ring. Given that 3D LiDAR operates at a frequency exceeding 10 Hz, with very short intervals between consecutive scans, the first stage of distortion compensation assumes constant angular and linear velocity to estimate motion and correct distortion within a brief timeframe. The second stage involves reevaluating distortion after the pose estimation process, and the recalculated undistorted features are then updated in the final map. The two-stage processes are described below. To estimate the transformation between two consecutive scans using linear interpolation over a small period of time Δt:

$$T_k(\Delta t) = T_{k-1} \cdot \exp(\Delta t \cdot \xi) \quad (2)$$

T_{k-1} represents the transformation at the previous frame, ξ denotes the twist vector representing instantaneous motion, which contains the vector of angular velocities and linear velocities, and the $\exp(\cdot)$ function is used to convert the twist vector into a transformation matrix. The undistorted features are

expressed as follows.

$$\tilde{p}_k = \left\{ T_k(\Delta t) p_k^{m,n} | p_k^{m,n} \in p_k \right\} \quad (3)$$

Which is used in the final pose estimation as below equation:

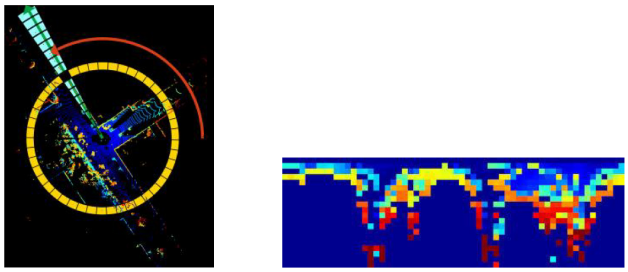
$$T_k^* = min_{T_k} \sum (w1 d_{ei}) + \sum (w2 d_{pj}) \quad (4)$$

where T_k^* represents the current pose estimation, where $w1$ and $w2$ denote the weights used to balance the matching process further, which can be calculated based on the local smoothness defined previously. The variable d_{ei} indicates the separation between a specific edge feature (the i-th one) and the line sub-map. Similarly, d_{pj} shows how far apart a particular plane feature is (in this case, j-th in sequence) from its corresponding plane sub-map. The primary goal of this section is to introduce a non-iterative two-stage distortion compensation method as an alternative to the computationally inefficient iterative method. Additionally, scan-to-submap matching is employed for pose estimation. These two steps significantly reduce computational complexity, leading to faster processing and improved overall system performance.

3.2. Intra-Loop closure detection and optimization

Existing LiDAR-based loop closure detection methods grapple with two primary challenges. Firstly, they strive to develop descriptors that attain rotational invariance, irrespective of changes in viewpoint. Secondly, these methods address the issue of noise handling concerning spatial descriptors. Next, managing the noise levels is yet another focus area for these spatial identifiers. This is because a point cloud's resolution can fluctuate based on distance, and its associated normals are typically speckled with cacophony [5]. Current methods primarily rely on histograms [33, 34], and [35] to tackle the mentioned challenges. While the histogram method provides a random gauge of the scenario, it lacks in portraying an elaborate layout or structure of surroundings. This constraint makes it difficult to develop a unique enough descriptor for location recognition, which may result in false positives.

According to [33], a two-phase matching technique is used in conjunction with Scan Context descriptors in this study. By providing an effective bin encoding method while maintaining the point cloud's underlying structure, the Scan Context descriptor converts a full 3D scan point cloud into a matrix. At the same time, in order to achieve a reasonable search duration when matching, the two-phase matching method is employed. This synergy produces a stable loop closure algorithm that is independent of the LiDAR perspective.



(a) (b)

Fig. 1. Scan Context Description: (a) rings and sectors construction (b) how the sectors and rings are represented in a 2D matrix.

3.2.1. Scan context

Shape Context, proposed by [36], is a model for scan contexts that uses geometric encoding to capture the shape of a point cloud surrounding a local key point. Partitioning the 3D scan into Nr rings and Ns sectors is shown in Fig. 1a. Afterwards, each bin is given a single real value based on the z-coordinate of the point cloud inside that bin. This real value, denoted as v_{ij} serves as the pixel value when translating the bin to a pixel, as depicted in 1b. Ultimately, a scan context, denoted as SC, is represented as a $Nr\ x\ Ns$ matrix, as follows:

$$SC = (v_{ij}) \in R^{NrxNs},\ v_{ij} = max_{p \in p_{ij}} z(p) \quad (5)$$

Here, p_{ij} represents a group of points found within the specific grid compartment determined by ring (i) and sector (j). The function $z(\cdot)$ is used to tell us about this point's exact position (p) in terms of its vertical or 'z-coordinate'.

3.2.2. Loop closure detection

After converting the 3D LiDAR scan to the Scan Context matrix, the similarity score can be calculated between the source (SC^s) and the candidate (SC^c) Scan Context matrixes based on the distance measurement. Both SC^s and SC^c are compared in a column-wise manner. Or put another way, a cosine distance (c_j^s, c_j^c) is deployed to ascertain the gap between two column vectors at an identical index. The question of this 'distance' can be framed as follows:

$$d(SC^s, SC^c) = \frac{1}{N_s} \sum_{j=1}^{N_s} \left(1 - \frac{c_j^s c_j^c}{\|c_j^s\| \|c_j^c\|} \right) \quad (6)$$

However, revisiting the same location from a different direction or corner caused shifts in some columns in SC^c. As demonstrated in the equation, the minimal distance was determined utilizing all column-shifted

scan contexts to solve this issue:

$$D(SC^s, SC^c) = min_{n\in|Ns|}d(SC^s, SC^c) \quad (7)$$

$$n^* = argmin_{n\in|Ns|}d(SC^s, SC^c_n) \quad (8)$$

Where SC^c_n represents the candidate scan context shifted by n columns from the source SC^s scan context. This shift information may be an excellent starting point for localization refinement like ICP.

3.2.3. Search algorithm

In the realm of context-place recognition searching, three primary approaches are commonly employed: pairwise similarity scoring, sparse optimization, and nearest-neighbor search [37]. The search method hierarchically blends pairwise scoring and closest neighbor search to reach a realistic search time. As a rotation-invariant descriptor retrieved from a scan context, the ring key information is crucial to the search process. Each row in the scan context, denoted as undergoes encoding into a single real value through the function $\Theta(r)$. These resulting values are then represented by an N_r-dimensional vector as follows:

$$\mathrm{K} = \{\Theta(r_1), \ldots, \Theta(r_{N_r})\}, \text{ where } \Theta : r_i - > R \quad (9)$$

The ring key attains rotation invariance through its encoding function's reliance on the occupation ratio, which remains consistent regardless of viewpoint. Moreover, its inherent simplicity, lacking detailed information, streamlines the search process for potential candidates or loops. The source's ring key finds comparable keys and obtains their scan indexes using vector K to build a KD tree. Their scan context is compared to the source's operating distance (6) to discover the top comparable keys. After accepting a candidate, the nearest place is revisited.

$$c^* = argmin_{ck\in C}D(SC^s, SC^{ck}), s.t\ D < t \quad (10)$$

In this context, C represents a set of indexes of candidates extracted from the KD tree, where c^* is the index of the identified loop, and t is a predetermined acceptance threshold. The pose graph is optimized using Incremental Smoothing and Mapping (ISAM) once coil closure detection is finished. This optimization stage is essential for fixing the robot's trajectory drift and making the produced local map more accurate and high-quality. By including ISAM, a more accurate and consistent representation of the environment is created, which helps to fix any path-related mistakes or inconsistencies that may have built up over time.

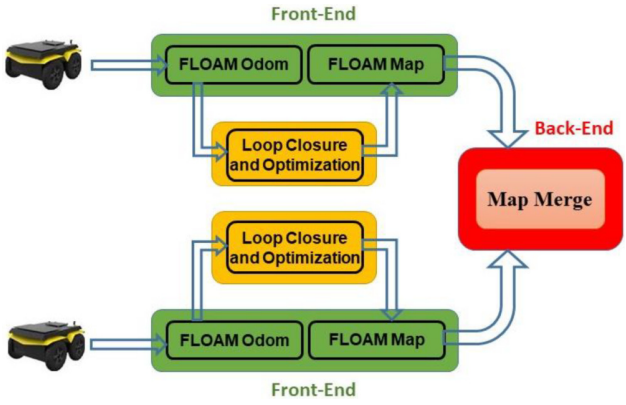


Fig. 2. EMR-FLOAM block diagram.

4. The proposed algorithm

Fig. 2 depicts the proposed algorithm diagram. Some robots in this setup estimate odometry, detect intra-loop closures, and optimize and construct local maps. After that, the locally produced maps are forwarded to the central station to identify inter-loop closures and implement fusion. The system accurately identifies candidates for intra-loop closure and performs place recognition between local maps by distributing computations between the robots and the central station. By taking this route, we can speed up the process of creating a world map while using our resources more efficiently.

The core idea behind map fusion algorithms is to find the places where local maps overlap, match them, and then figure out how to alter them such that they create a coherent global map. It is possible to think of the overlapping areas as inter-loop closures or places that are visited by numerous robots [20]. At the central station, local maps are collected, outliers are removed, and the maps are down-sampled to reduce computations. The following steps are then implemented to detect the overlapped areas:

- Surface Normal Estimation: As a preprocessing step, vectors perpendicular to the surface are determined in the vicinity of each point by performing least-squares plane fitting estimation for each point neighborhood to determine the orientation of the estimated normal.
- Keypoint Detection: To enhance matching efficiency, key points are extracted from the point clouds to reduce the number of points in each map. Since the point cloud contains only coordinate information without additional data, the Harris 3D keypoint detector is employed for this purpose.
- Descriptor Computation: Fast Point Feature Histogram (FPFH) descriptors are computed for their

Algorithm 1. Matching approach using k-nearest matches validated with reciprocal matching.

Input: $D1$, $D2$ set of descriptors
Output: set of matches between $D1$, $D2$

```
function matchReciprocalK (D1, D2)
    M = {}
    for all di ∈ D1 do
        N ← k-nearest neighbors of di in D2
        for all dj ∈ N do
            N′ ← k-nearest neighbors of dj in D1
            if di ∈ N′ then
                M ← M ∪ {(di, dj)}
            end if
        end for
    end for
    return M
endfunction
```

effectiveness in handling geometric information and superior processing speed.

- Matching Process: The objective is to identify descriptors from all sets of detected descriptors that describe the same place (inter-loop closure). This task poses a challenge as the descriptors may be less descriptive than desired. Consequently, the correct match may not always be the closest descriptor but rather the k-nearest descriptor. Reciprocal matching is utilized in this work to find the matching relationship in bidirectional (d(i) ↔ d(j)) and provide mutual agreement as clarified in Algorithm 1.
- Pair-Transform Estimation: The Random Sample Consensus (RANSAC) technique finds matching inlier descriptors and estimates their transform using Singular-Value Decomposition. FAST_VGICP refines estimated transform.
- Global transform estimation: An approximation of the transition between regional maps and a worldwide standard is provided by the map-merge graph. The match graph, similar to a pose graph, is formed, and the most significant connected component is determined, excluding matches with confidences below 1.0. After selecting the global reference frame from the maximum spanning tree, only one path is supplied from the nodes to it. Finally, tree and pairwise transforms estimate the global transform.
- Map Merging: When the maps and their transforms are obtained, the 3D global map is constructed by combining them. This can be expressed mathematically by the equation below:

$$M_{global} = \mathrm{U}_i T_i(M_i), \text{ s.t } T_i > Conf._{Th} \tag{11}$$

U_i represents the union operation, combining the transformed local maps, $T_i(M_i)$ denotes the application of the transformation T_i to the local map M_i. The pairwise transformation T_i should satisfy the confidence threshold $Conf._{Th}$.

Place recognition occurs during the map fusion stage, eliminating the necessity for inter-loop closure algorithms. This omission stems from the reliance on the map-merge graph. Unlike graph-based SLAM, where loop closures are crucial, they are typically non-essential for optimal performance in standard setups of map-merging graphs. In SLAM graphs, measurements linked with edges are influenced by Gaussian noise. However, map-merging graphs rely on pairwise estimates derived from the geometry of entire maps, encompassing a significant number of points. These pairwise estimates tend to be precise, covering extensive sections of the environment. Evaluation metrics, such as confidence thresholds, are in place for these pairwise estimates, ensuring the utilization of robust estimates within the graph [38].

5. Simulation results

The proposed work was implemented on a laptop featuring an Intel(R) Core(TM) i7-7500U CPU clocked at 2.70 GHz (with a boost up to 2.90 GHz), accompanied by 12.0 GB of RAM and running on a 64-bit operating system. For virtualization, VirtualBox software was utilized and configured with Workstation Pro settings, including four processors and 8 GB of memory. This virtual environment facilitated the installation of Ubuntu 20.04 and ROS (Robotic Operating System) Noetic version. Initially, the system's performance and accuracy were assessed using real-world data, comparing it with other LIDAR-based SLAM methods. Afterwards, a Gazebo-created virtual environment was used to mimic the implementation. This dual method allowed virtual testing and real-world validation to thoroughly evaluate the proposed technology.

5.1. Real world dataset

Our method was compared against cutting-edge LiDAR-based MR-SLAM systems using a real-world dataset of outdoor surrounds [39]. This dataset was chosen since it was the first open-source multi-robot dataset. Six LiDAR-equipped robots generate a complex and diverse real-world scenario ideal for system evaluation and benchmarking.

5.2. Use the outdoor environment dataset to assess the suggested algorithm

A. Testing without Loop Closure Detection: To compare results, the proposed system was tested

Table 4. Details of the selected dataset.

File name	Size (GB)	Length (m)	LiDAR data topic
Acl_Jackal.bag	8.0	784.998	/acl_jackal/LiDAR_points
Sparkal1-001.bag	10.86	1225.951	/sparkal1-001/LiDAR_points

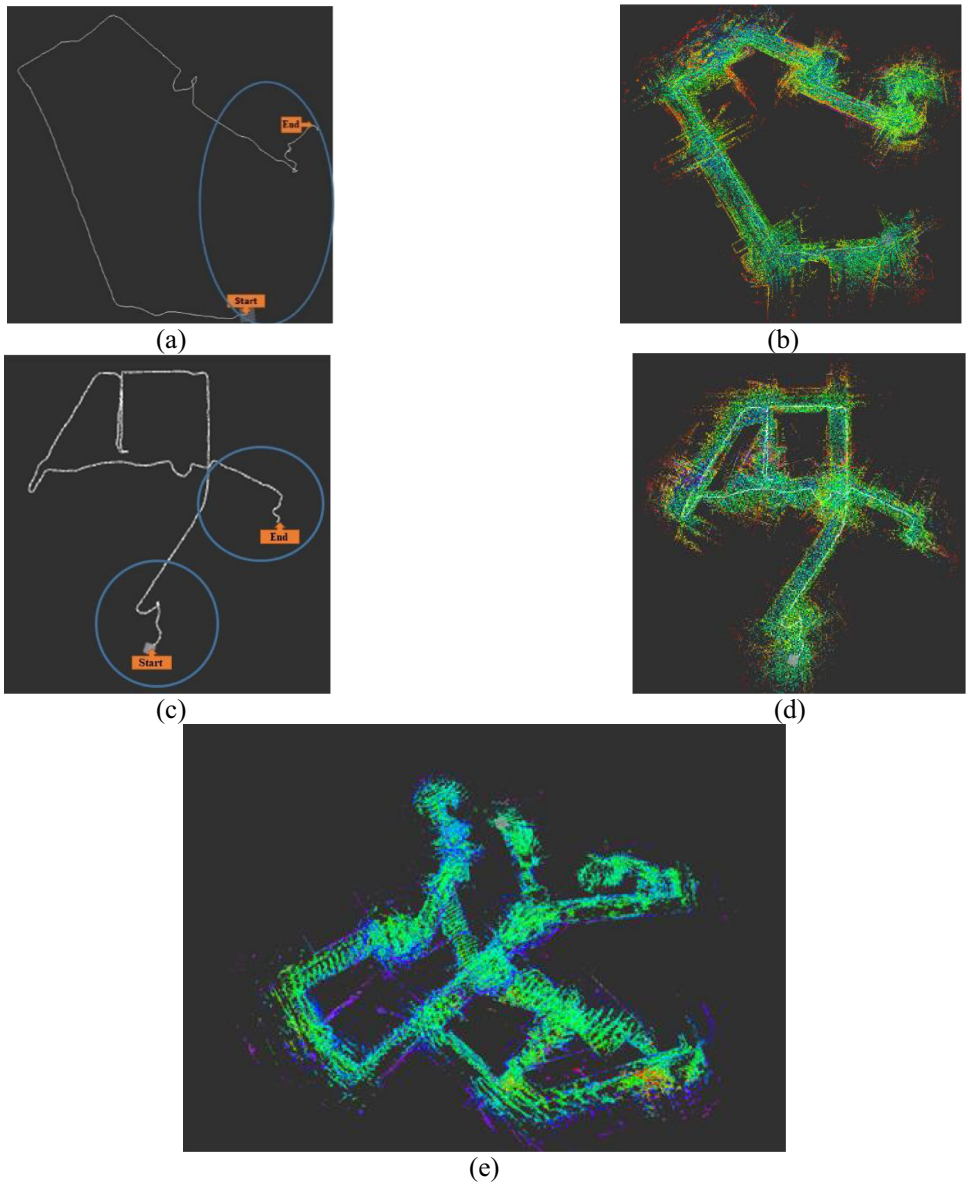


Fig. 3. Path Estimation, Local Maps, and Global Map Produced by MR-FLOAM: (a & c) illustrate the estimated trajectories by FLOAM for each robot, (b & d) depict the local maps built by FLOAM for each robot, (e) the global map constructed by the MR-FLOAM.

against A-LOAM, a LiDAR-based SLAM solution. Table 4 shows that two robots evaluated dataset sequences 1 and 4. These sequences were chosen to improve review and introduce tough loop closure conditions. This technique lets us compare our system against A-LOAM in detail.

We first employed MR-FLOAM and MR-ALOAM SLAM systems individually without a loop closure method. The paths, local maps, and global maps created by each system are shown in Figs. 3 and 4. Table 5 shows each system's computation time and difficulty. The table also displays how much RAM each system requires to store its global map, indicating its size. This early study lays the groundwork for evaluating system performance without loop closure detection. The formula calculates each robot's Root

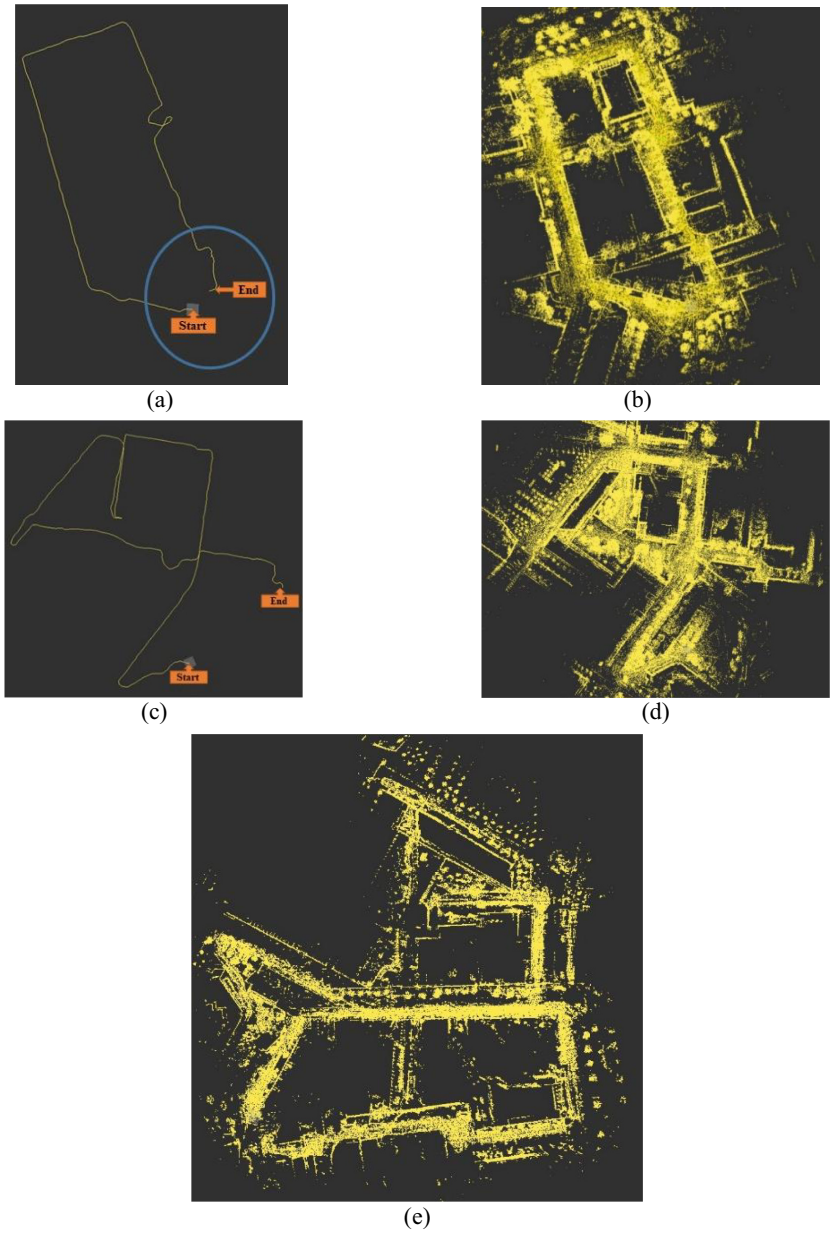


Fig. 4. Path Estimation, Local Maps, and Global Map Produced by ALOAM: (a & c) illustrate the estimated trajectories by ALOAM for each robot, (b & d) depict the local maps built by ALOAM for each robot, (e) the global map constructed by the MR-ALOAM.

Mean Square Error (RMSE), which may be used to assess path estimate precision and accuracy:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{est} - x_{gt})^2 + (y_{est} - y_{gt})^2 + (z_{est} - z_{gt})^2} \tag{12}$$

where $(x_{est}, y_{est}, z_{est})$ signifies the predicted posture coordinate, and (x_{gt}, y_{gt}, z_{gt}) represents the ground truth coordinates. The RMSE value for each robot is clarified in Table 6.

Table 5. Average computing time and global map size for each algorithm.

MR-implementation method	Average computation time	Global map size
MR-ALOAM	103665.5735 ms	3688037 point
MR-FLOAM	8173.136616 ms	1096696 point

B. Testing with Loop Closure Detection: In this section, we reintroduce the previously mentioned systems with the integration of our proposed intra-loop

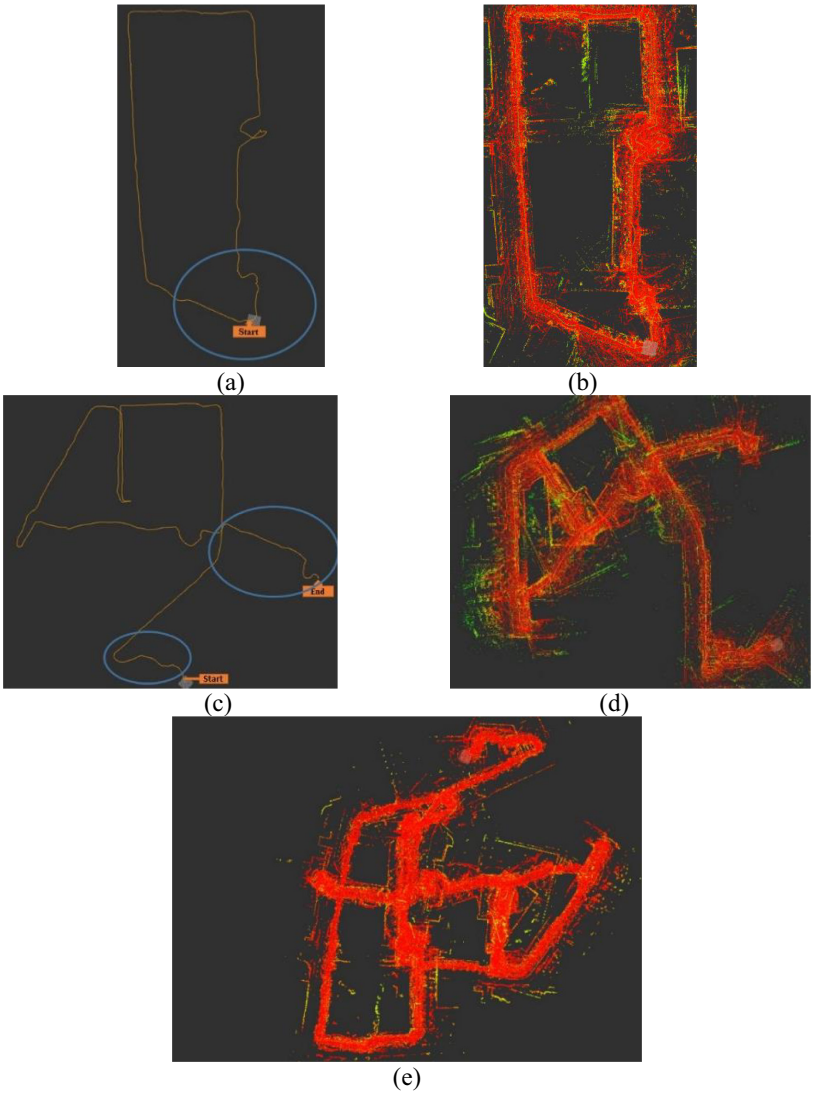


Fig. 5. Path Estimation, Local Maps, and Global Map Produced by Enhanced FLOAM:(a & c) illustrate the estimated trajectories of the robots using enhanced FLOAM, (b & d) depict the local maps generated by enhanced FLOAM for each robot, (e) the global map constructed by the enhanced MR-FLOAM.

Table 6. RMSE values for each robot in both algorithms.

MR-implementation method	ROBOT1	ROBOT2
MR-ALOAM	8.685816 (m)	2.914255 (m)
MR-FLOAM	13.122269 (m)	18.340151 (m)

detection approach, paying close attention to the computation time and the size of the global map. Furthermore, we evaluate the values of the RMSE for each robot to assess the effectiveness of our proposed loop closure detection method. The simulation outcomes for the enhanced multi-robot Fast Localization Odometry and Mapping (EMR-FLOAM) and the enhanced multi-robot an Improved Localization Odometry and Mapping (EMR-ALOAM) are elucidated in Figs. 5 and 6, as well as Tables 7 and 8.

Table 7. Average computing time and the map size with loop closure algorithm.

MR-implementation method	Average computation time	Global map size
EMR-ALOAM	87017.23542 ms	2772359 points
EMR-FLOAM	43213.38074 ms	841450 points

Table 8. RMSE value for each robot in both enhanced approaches.

MR-implementation method	ROBOT1	ROBOT2
EMR-ALOAM	8.903713 (m)	2.709052 (m)
EMR-FLOAM	9.446425 (m)	13.120242 (m)

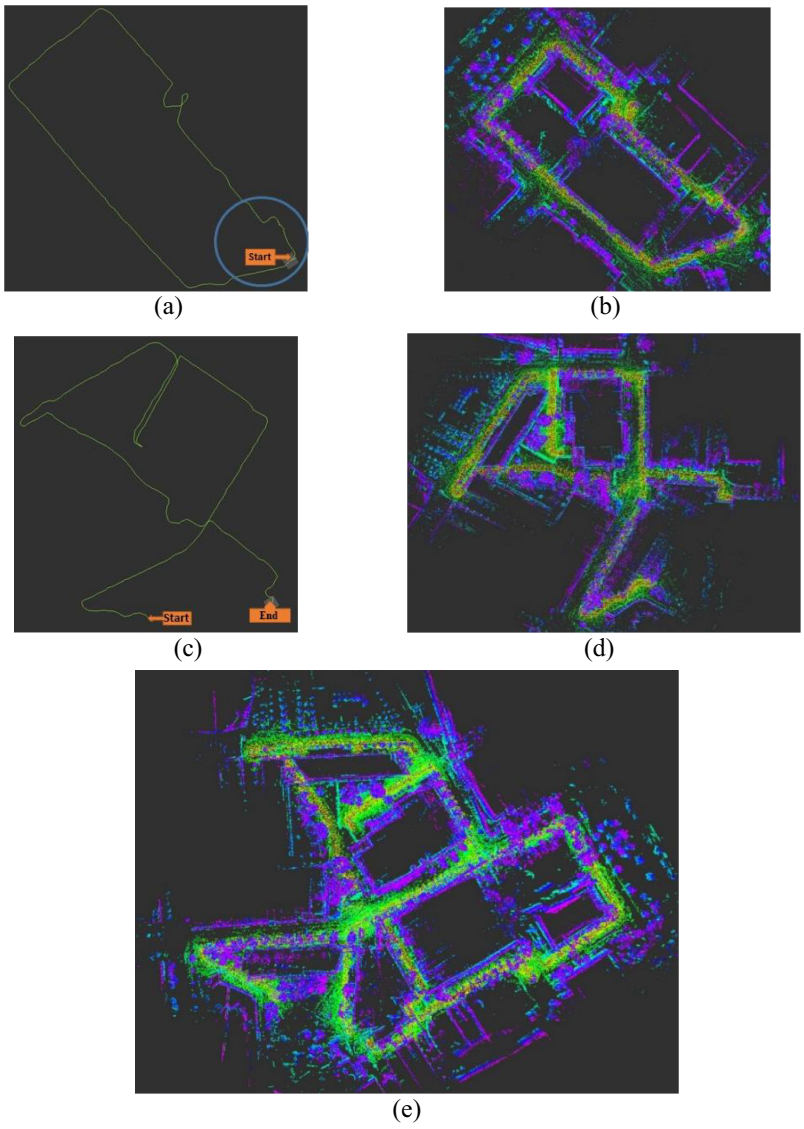


Fig. 6. Path Estimation, Local Maps, and Global Map Produced by Enhanced ALOAM: (a & c) illustrate the estimated trajectories of the robots using enhanced ALOAM, (b & d) depict the local maps generated by enhanced ALOAM for each robot, (e) the global map constructed by the enhanced MR-ALOAM.

5.3. Evaluation of the proposed algorithm in the simulated world

A virtual warehouse environment has been constructed using the Gazebo simulator, featuring two simulated robots designed to navigate within the environment. These robots are equipped with Velodyne sensor simulation (VLP-16), gathering information in point cloud format. Fig. 7 provides a visual depiction of the simulated environment and the robots. The proposed system will undergo testing in two phases, outlined as follows:

A. Testing the Proposed Algorithm without Including Intra Loop Closure Algorithm: This section focuses on utilizing only the overlapped area for detecting the revisited location. The local maps exhibit a small overlapping region marked with a red circle, as depicted in Fig. 7. The green and blue circles indicate the areas that will be detected by Robot 1 and Robot 2, respectively.

Robot 1 will commence its journey from point (1, 0, 0) and navigate back to the same point, thereby introducing a loop closure challenge. In contrast, Robot 2 will initiate its path from point (1, 1, 0) without returning to the initial position. Fig. 8 illustrates the generated local maps, the inferred paths for each robot, and the resultant global map. Table 9 provides details on the RMSE values, computing time, and local map size for each robot.

Fig. 7. Warehouse simulated environment.

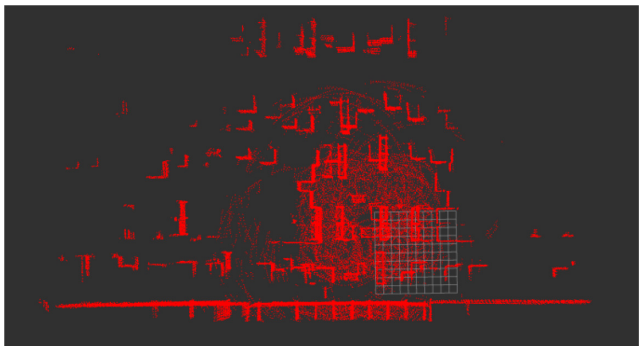


Fig. 9. Inconsistent global map.

Table 9. RMSE, local map size, and the total computing time average to construct the global map.

	RMSE	Map size	Computing time
Robot1	0.014373	16274 points	252.090525
Robot2	0.015164	25970 points	

B. Testing the Proposed System, including the Intra Loop Closure Algorithm: In this phase, each robot has been equipped with an intra-loop detection algorithm. The effectiveness of the algorithm has been evaluated in the environment illustrated in Fig. 7, featuring a small overlapped area. The resulting global map is presented in Fig. 9.

As depicted in Fig. 9, there were inaccuracies in the merging of the local maps. Consequently, adjustments were made to the simulated environment to enhance information sharing among the robots, thereby expanding the overlapped area, as illustrated in Fig. 10. Fig. 11 displays the local maps and the resulting global map. Additionally, Table 10 presents the corresponding RMSE values and computing times.

6. Disccusion of the simulation results

6.1. Path estimation and map construction

Our proposed MR-FLOAM system demonstrates faster path estimation when compared to MR-ALOAM, leading to accelerated global map construction, as detailed in Table 5. While both global and local maps generated by MR-LOAM exhibit intricate details, they come at the cost of a more significant memory allocation (map size). Notably, in the absence of the loop closure detection algorithm, the Root Mean Square Error (RMSE) values for each

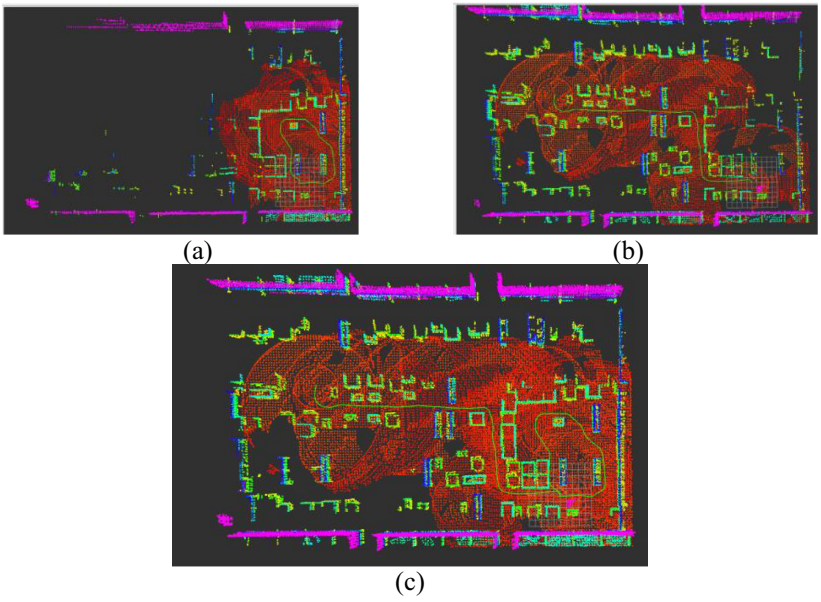


Fig. 8. Local maps and the global map: (a) and (b) represent the local maps and the estimated path for robot1 and robot2, respectively, and (c) illustrates the global merged map.

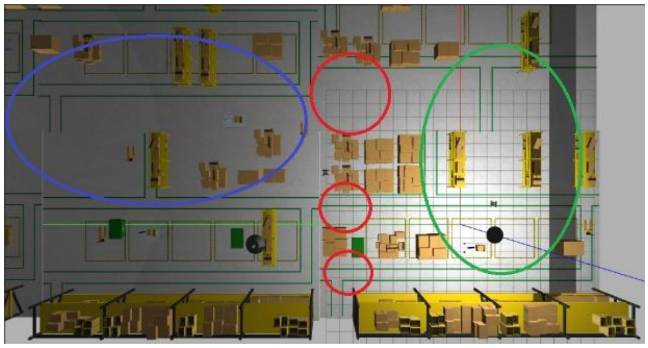


Fig. 10. Modified warehouse environment, red circle represents the overlapped area, and green and blue circles represent the detected area by robot1 and robot2, respectively.

robot's path experience a noticeable increase. This increase is particularly evident in Robot1's trajectory, as indicated in Table 6. The rise in RMSE values signifies a drifting phenomenon in Robot1's path, which is further illustrated by the unclosed trajectory in Figs. 3a and 4a. Consequently, this drift leads to inconsistencies in local and global map construction, as depicted in Figs. 3b, 3e, 4b, and 4e.

6.2. Drift correction and map improvement

Figs. 5a, 5b, 6a, and 6b shed light on the improvement in path estimation and local maps, with drift correction becoming evident in both MR-FLOAM and MR-ALOAM following loop closures. The RMSE values in both systems have decreased, indicating successful drift correction, as demonstrated in Table 8. While the enhanced MR-ALOAM exhibits lower RMSE values, signifying higher precision, the enhanced MR-FLOAM still offers the advantage of reduced computing time, as shown in Table 6. It's noteworthy that both systems have witnessed improvements in their global and local maps. However, it's worth mentioning that the enhanced MR-ALOAM produces a global map with more detailed information, necessitating a more significant memory allocation, as outlined in Table 7. Conversely, the improved MR-FLOAM requires less memory to store the global map while maintaining essential information content.

Table 10. RMSE, local map size, and the total computing time average to construct the global map.

	RMSE(m)	Map size	Computing time(ms)
Robot1	0.008618	39206 points	616.3846
Robot2	0.009492	61329 points	

6.3. Robustness in simulated environments

In the simulated environment tests, our proposed MR-FLOAM consistently generates a global map, even in scenarios with small overlapped areas, as depicted

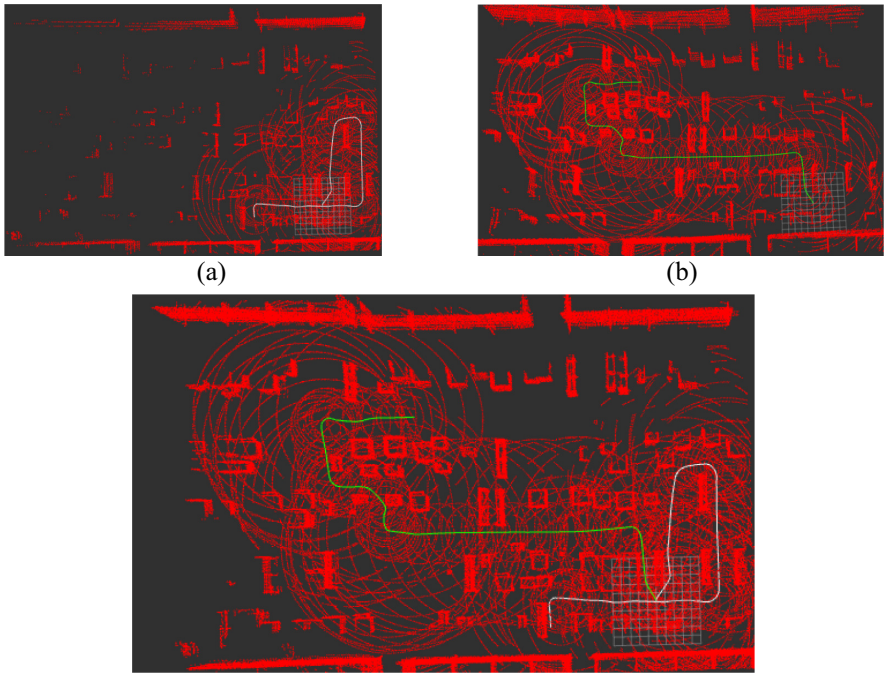


Fig. 11. Simulation result after enhancement, (a) and (b) represent the local maps and the estimated paths for robot1 and robot2, respectively and (c) represent the complete global map.

in Fig. 8c. In the second part of the test, despite incorporating the intra-loop closure detection algorithm, the system produces an inconsistent global map, as shown in Fig. 9. Consequently, we modified the environment to increase the overlapped area, resulting in Fig. 10. This modification led to a consistent global map, as observed in Fig. 11c. Furthermore, Robot1's trajectory achieved loop closure, as evidenced in Fig. 11a, accompanied by a low RMSE value, as detailed in Table 10.

7. Limitation and future directions

The proposed centralized LiDAR-based MR-SLAM system has shown promising results; however, we have identified several areas that present challenges and offer opportunities for further advancement:

1. Perceptual Aliasing: The robustness of our loop closure detection algorithm is heavily influenced by the environmental context, with symmetrical and featureless structures presenting a particular challenge. To address this, future research will look into more robust feature descriptors or machine learning classifiers (SegMatch, Range++) that can enhance discrimination between visually similar environments, thereby mitigating the perceptual aliasing issue.
2. Memory Requirements: The high precision of our enhanced MR-ALOAM system comes with significant memory demands for storing detailed global maps. To save space and keep map details safe, upcoming projects will look into ways to shrink and store data more efficiently.
3. Challenges with 3D Point Clouds: It's tough to spot areas that overlap in dense clouds. To get better at finding places and combining maps, we're going to work on making better ways to show features.
4. Sharing the computing work between robots and the central station makes things simpler but can lead to delays because of slow communications. Future studies will work on improving how computing tasks are divided and how devices talk to each other to make the system work better and faster in real-time.

We're working on making our method better so that when robots or areas barely touch, they can still connect better. We're also adding extra details in our maps to help robots know more about where they are, which makes them more competent in deciding and makes the maps better, too.

We hope to use this method for smart cars, automated storage places, and digging sites. We want to see how well our method works in these different places. We'll keep trying new things in the world of many robots working together.

8. Conculsion

This study expands upon our prior work [4] by presenting a new method for identifying loop closures in centralized multi-robot LiDAR-based SLAM. We successfully decreased the robot's path prediction drift by applying a strategy that takes into account both intra- and inter-loop closures and by using scan context descriptors for intra-loop detection. As a result, trajectory mapping is now more precise. Our approach (EMR-FLOAM) has helped generate a consistent global map, and its performance in pose estimation has been especially noteworthy. Computability and optimization of resources were key design goals for the suggested method because of their importance in some applications. Cases where careful allocation of computing resources is critical have shown promising results for our approach. The work done here is a significant advance for SLAM, which relies on centralized multi-robot LiDAR. We have proven that our method works and highlighted how crucial it is for robotic mapping and navigation systems to have accurate data and well-managed resources. Our research provides strong evidence that robotic perception and mapping technology can continue to advance.

Funding

None.

Acknowledgement

None.

Conflicts of interest

The author declares no conflict of interest.

References

1. Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, and Ian Reid, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, 2016. doi: 10.1109/TRO.2016.2624754.
2. J. Pippine, "DARPA subterranean (SubT) challenge," Auvsi Xponential 2018, 2018.

3. S. A. S. Mohamed, M. H. Haghbayan, T. Westerlund, J. Heikkonen, H. Tenhunen, and J. Plosila, "A survey on odometry for autonomous navigation systems," *IEEE Access*, vol. 7, pp. 97466–97486, 2019. doi: 10.1109/ACCESS.2019.2929133.
4. B. Ahmed Jalil and I. Kasim Ibraheem, "Multi-Robot SLAM using fast LiDAR odometry and mapping," *Designs*, vol. 7, no. 5, 2023. doi: 10.3390/designs7050110.
5. G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3D point cloud map," *IEEE International Conference on Intelligent Robots and Systems*, pp. 4802–4809, 2018.
6. R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D Registration," *IEEE International Conference on Robotics and Automation*, pp. 3212–3217, 2009. doi: 10.1109/ROBOT.2009.5152473.
7. G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numer. Math.*, vol. 14, no. 5, pp. 403–420, 1970. doi: 10.1007/BF02163027.
8. P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239–256, 1992. doi: 10.1109/34.121791.
9. M. A. Abdulgalil, M. M. Nasr, M. H. Elalfy, A. Khamis, and F. Karray, "Multi-robot SLAM: An overview and quantitative evaluation of MRGS ROS framework for MR-SLAM," pp. 165–183m, 2019. doi: 10.1007/978-3-319-78452-6_15.
10. R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Aligning point cloud views using persistent feature histograms," *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pp. 3384–3391, 2008. doi: 10.1109/IROS.2008.4650967.
11. S. Choudhary, L. Carlone, C. Nieto, J. Rogers, H. I. Christensen, and F. Dellaert, "Distributed mapping with privacy and communication constraints: Lightweight algorithms and object-based models," *International Journal of Robotics Research*, vol. 36, no. 12. pp. 1286–1311, 2017. doi: 10.1177/0278364917732640.
12. S. Choudhary, L. Carlone, C. Nieto, J. Rogers, H. I. Christensen, and F. Dellaert, "Distributed trajectory estimation with privacy and communication constraints: A two-stage distributed Gauss-Seidel approach," in Proceedings - IEEE International Conference on Robotics and Automation, vol. 2016-June, pp. 5261–5268, 2016. doi: 10.1109/ICRA.2016.7487736.
13. 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 Robot. Autom. Lett.*, vol. 7, no. 2, pp. 1150–1157, 2022. doi: 10.1109/LRA.2021.3138156.
14. H. Wang, C. Wang, and L. Xie, "Intensity scan context: Coding intensity and geometry relations for loop closure detection," in 2020 IEEE International Conference on Robotics and Automation (ICRA, pp. 2095–2101), 2020. doi: 10.1109/ICRA40945.2020.9196764.
15. R. V. Joshua G. Mangelson, Derrick Dominic, and Ryan M. Eustice, "Pairwise consistent measurement set maximization for robust multi-robot map merging," *IEEE International Conference on Robotics and Automation*, pp. 2916–2923, 2018. doi: 10.1109/ICRA.2018.8460217.
16. G. B. Pierre-Yves Lajoie, "Swarm-SLAM: Sparse decentralized collaborative simultaneous localization and mapping framework for multi-robot systems," *Artic*. arxiv, pp. 1–11, 2023.
17. S. Zhong, Y. Qi, Z. Chen, J. Wu, H. Chen, and M. Liu, "DCL-SLAM: A distributed collaborative LiDAR SLAM framework for a robotic swarm," *Artic*. arxiv, 2022. doi: https://doi.org/10.48550/arXiv.2210.11978.
18. Y. Wang, Z. Sun, C. Z. Xu, S. E. Sarma, J. Yang, and H. Kong, "LiDAR iris for loop-closure detection," in IEEE International Conference on Intelligent Robots and Systems, pp. 5769–5775, 2020. doi: 10.1109/IROS45743.2020.9341010.
19. S. Choudhary, L. Carlone, C. Nieto, J. Rogers, H. I. Christensen, and F. Dellaert, "Distributed mapping with privacy and communication constraints: Lightweight algorithms and object-based models," *Int. J. Rob. Res.*, vol. 36, no. 12, pp. 1286–1311, 2017. doi: 10.1177/0278364917732640.
20. P.-Y. Lajoie, B. Ramtoula, F. Wu, and G. Beltrame, "Towards collaborative simultaneous localization and mapping: a survey of the current research landscape," F. Robot., vol. 2, no. 1, pp. 971–1000, 2022. doi: 10.55417/fr.2022032.
21. M. A. Uy and G. H. Lee, "PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4470–4479. doi: 10.1109/CVPR.2018.00470.
22. K. Ebadi, M. Palieri, S. Wood, C. Padgett, and A. akbar Aghamohammadi, "DARE-SLAM: Degeneracy-aware and resilient loop closing in perceptually-degraded environments," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 102, no. 1, 2021. doi: 10.1007/s10846-021-01362-w.
23. R. Dube, A. Gawel, H. Sommer, J. Nieto, R. Siegwart, and C. Cadena, "An online multi-robot SLAM system for 3D LiDARs," *IEEE International Conference on Intelligent Robots and Systems*, vol. 2017-Septe. pp. 1004–1011, 2017. doi: 10.1109/IROS.2017.8202268.
24. H. Do, S. Hong, and J. Kim, "Robust loop closure method for multi-robot map fusion by integration of consistency and data similarity," *IEEE Robotics and Automation Letters*, vol. 5, no. 4. pp. 5701–5708, 2020. doi: 10.1109/LRA.2020.3010731.
25. K. Ebadi *et al.*, "LAMP: Large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments," in Proceedings - IEEE International Conference on Robotics and Automation, 2020, pp. 80–86. doi: 10.1109/ICRA40945.2020.9197082.
26. L. C. Yun Chang, K. Ebadi, C. E. Denniston, M. Fadhil Ginting, A. Rosinol, A. Reinke, M. Palieri, J. Shi, A. Chatterjee, B. Morrell, Ali-akbar, and A.-M., "LAMP 2. 0: A robust multi-robot SLAM system for operation in challenging large-scale underground environments," *IEEE Robot. Autom. Lett.*, 2022.
27. A. A. C. E. Denniston, Y. Chang, A. Reinke, K. Ebadi, G. S. Sukhatme, L. Carlone, and B. Morrell, "Loop closure prioritization for efficient and scalable multi-robot SLAM," *IEEE Robotics and Automation Letters*, vol. 7, no. 4. p. 8, 2022. doi: 10.1109/LRA.2022.3191156.
28. J. Zhou *et al.*, "Graph neural networks: A review of methods and applications," arXiv:1812.08434 [cs, stat], 2019, vol. 1. pp. 57–81, 2020. doi: 10.1016/j.aiopen.2021.01.001.
29. E. Jones, J. Sofonia, C. Canales, S. Hrabar, and F. Kendoul, "Applications for the Hovermap autonomous drone system in underground mining operations," in *Journal of the Southern African Institute of Mining and Metallurgy*, vol. 120, no. 1, pp. 49–56, 2020. doi: 10.17159/2411-9717/862/2020.
30. H. Wang, C. Wang, C. L. Chen, and L. Xie, "F-LOAM: Fast LiDAR Odometry and Mapping," IEEE International Conference on Intelligent Robots and Systems. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4390–4396, 2021. doi: 10.1109/IROS51168.2021.9636655.
31. T. Shan and B. Englot, "LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain," IEEE International Conference on Intelligent Robots and Systems. pp. 4758–4765, 2018.

32. L. Li *et al.*, “SA-LOAM: Semantic-aided LiDAR SLAM with Loop Closure,” in 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 7627–7634, May 2021. doi: 10.1109/ICRA48506.2021.9560884.
33. N. Muhammad and S. Lacroix, “Loop closure detection using small-sized signatures from 3D LIDAR data,” in 9th IEEE International Symposium on Safety, Security, and Rescue Robotics, SSRR 2011, pp. 333–338, 2011. doi: 10.1109/SSRR.2011.6106765.
34. W. Wohlkinger and M. Vincze, “Ensemble of shape functions for 3D object classification,” 2011 IEEE International Conference on Robotics and Biomimetics, ROBIO. pp. 2987–2992, 2011. doi: 10.1109/ROBIO.2011.6181760.
35. M. Himstedt, J. Frost, S. Hellbach, H. J. Bohme, and E. Maehle, “Large scale place recognition in 2D LIDAR scans using Geometrical Landmark Relations,” IEEE International Conference on Intelligent Robots and Systems. pp. 5030–5035, 2014. doi: 10.1109/IROS.2014.6943277.
36. F. Mokhtarian and S. Abbasi, “Matching shapes with self-intersections: Application to leaf classification,” IEEE Transactions on Image Processing, vol. 13, no. 5. pp. 653–661, 2004. doi: 10.1109/TIP.2004.826126.
37. H. Zhang, F. Han, and H. Wang, “Robust multimodal sequence-based loop closure detection via structured sparsity,” in *Robotics: Science and Systems*, vol. 12, 2016. doi: 10.15607/rss.2016.xii.043.
38. J. H¨orner, “Automatic Point Clouds Merging,” 2018.
39. Y. Tian *et al.*, “Resilient and Distributed Multi-Robot Visual SLAM: Datasets, Experiments, and Lessons Learned,” 2023. doi: https://doi.org/10.48550/arXiv.2304.04362.