

Article

A SLAM Method Based on Multi-Robot Cooperation for Pipeline Environments Underground

Dongfeng Lu, Yunwei Zhang *, Zewu Gong and Tiannan Wu

Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China

* Correspondence: zhangyunwei72@gmail.com

Abstract: SLAM (simultaneous localization and mapping) technology has recently shown considerable forward progress; however, most of the mainstream SLAM technologies are currently based on laser- and vision-based fusion strategies. However, there are problems (e.g., a lack of geometric structure, no significant feature points in the surrounding environment, LiDAR degradation, and the longitudinal loss of constraints, as well as missing GPS signals within the pipeline) in special circumstances (e.g., in underground pipelines and tunnels), thus making it difficult to apply laser or vision SLAM techniques. To solve this issue, a multi-robot cooperation-based SLAM method is proposed in this study for pipeline environments, based on LIO-SAM. The proposed method can effectively perform SLAM tasks in spaces with high environmental similarity (e.g., tunnels), thus overcoming the limitation that existing SLAM methods have been poorly applied in pipeline environments due to the high environmental similarity. In this study, the laser-matching part of the LIO-SAM is removed, and a high-precision differential odometer, IMU inertial navigation sensor, and an ultrasonic sensor, which are not easily affected by the high similarity of the environment, are employed as the major sources of positioning information. Moreover, a front-to-back queue of two robots is trained in the pipeline environment; a unique period-creep method has been designed as a cooperation strategy between the two robots, and a multi-robot pose constraint factor (ultrasonic range factor) is built to constrain the robots' relative poses. On that basis, the robot queue can provide a mutual reference when traveling through the pipeline and fulfill its pose correction with high quality, thus achieving high positioning accuracy. To validate the method presented in this study, four experiments were designed, and SLAM testing was performed in common environments, as well as simple and complex urban pipeline environments. Next, error analysis was conducted using EVO. The experimental results suggest that the method proposed in this study is less susceptible to environmental effects than the existing methods due to the benefits of multi-robot cooperation. This applies to a common environment (e.g., a room) and can achieve a good performance; this means that a wide variety of piping environments can be established with high similarity. The average error of SLAM in the pipeline was 0.047 m, and the maximum error was 0.713 m, such that the proposed method shows the advantages of controllable cumulative error, high reliability, and robustness with an increase in the scale of the pipeline and with an extension of the operation time.

Keywords: multi-robot cooperation; periodic creep; pipeline environment; laser degradation; factor graph optimization



Citation: Lu, D.; Zhang, Y.; Gong, Z.; Wu, T. A SLAM Method Based on Multi-Robot Cooperation for Pipeline Environments Underground. *Sustainability* **2022**, *14*, 12995. <https://doi.org/10.3390/su142012995>

Academic Editors: Jianjun Ma, Mingfeng Lei, Yu Liang and Yuexiang Lin

Received: 5 September 2022

Accepted: 9 October 2022

Published: 11 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the ongoing advancement of technology, robots have become more and more intelligent and are now widely used in many fields, such as disease diagnosis, customer services, pipeline inspection, and so on [1]. Pipeline transmission is a vital means of transporting energy and materials (e.g., oil, natural gas, and urban water supplies) due to its characteristics of large transmission volume, low loss rates, good continuity, and controllability. Nevertheless, most of the current transmission pipelines are buried deep

underground and have hidden hazards (e.g., aging and corrosion); the inspection and maintenance of the underground pipelines also have problems (e.g., complex network, harsh environment, high maintenance cost, and loss of information regarding the distribution pipeline). The use of robots with a wide variety of sensors takes on great practical significance for carrying out environmental sensing and pipeline safety inspections. The ability of pipeline robots to precisely locate themselves in the pipeline is recognized as a critical technological aspect when performing maintenance operations [2].

SLAM (simultaneous localization and mapping) aims to solve the problem of locating and mapping a robot in an unknown environment. Multi-sensor information fusion technology endows the system with high robustness and accuracy and has become the main research direction in this field. In general, fusion methods are classified into three types, comprising EKF-based, factor graph optimization-based, and factor graph optimization plus EKF-based multi-sensor information fusion methods. Most mainstream SLAM open-source algorithm frameworks that are based on multi-sensor information fusion employ laser or visual sensor fusion strategies. Figure 1 illustrates the technological progress of the frameworks.



Figure 1. The technological progress of the SLAM mainstream algorithm.

The core of LOAM (Lidar Odometry and Mapping) [3] is to use two threads of the optimized front-end odometer and the back-end diagram to build a high-precision laser SLAM system. In addition, VLOAM (Visual-lidar Odometry and Mapping) [4] facilitates the distortion and matching of the point cloud using higher-frequency visual odometry, thus increasing the accuracy of the algorithm. Lego-LOAM (Lightweight and Ground-Optimized Lidar Odometry and Mapping) [5] is an optimized SLAM algorithm that uses ground and plane clustering. The LIO-SAM (Lidar Inertial Odometry via Smoothing and Mapping) [6] algorithm introduces the IMU pre-integration factor and the GPS factor to Lego-LOAM. The front-end odometer uses a tightly coupled IMU fusion method to replace the original inter-frame odometer, thereby making it easier to use. The back end incorporates GPS observation, based on Lego-LOAM. Moreover, the front end and back end are coupled to increase the accuracy of the system. LVI-SAM (Lidar-Visual-Inertial Odometry via Smoothing and Mapping) [7] introduces a vision component to LIO-SAM to increase the system's robustness. LIC-FUSION (LiDAR-Inertial-Camera) [8] is based on the MSCKF framework and combines a radar-inertial-vision SLAM algorithm with sensor non-reference calibration. LIC-FUSION2.0 [9] introduces a sliding window and planar tracking algorithms to LIC-FUSION. R2LIVE (Robust, Real-time, LiDAR-inertial-visual tightly-coupled state Estimator and mapping) [10] integrates the data from LiDAR, IMU, and vision to become tightly coupled. The R3LIVE (Robust, Real-time, RGB-colored, LiDAR-Inertial-Visual tightly-coupled state Estimation and mapping) [11] method comprises two subsystems, including LiDAR-inertial odometry (LIO) and visual-inertial odometry (VIO). LIO (FAST-LIO) builds global map geometry (using a 3D point-cloud map) through LiDAR and inertial sensor measurements. VIO adopts vision-inertial sensor data to create map textures (with 3D point-cloud colors) while integrating visual data directly and effectively by minimizing frame-to-map optical errors. For indoor SLAM tasks, a previous study [12] proposed a method to extract the overlapping regions between the respective local maps, based on geometric features, using a multi-robot collaboration. The core of this method is to extract the geometric features first, then perform a similarity evaluation, and finally, complete the overlapping parts to merge the local maps for constructing the global map.

LiDAR offers certain advantages (e.g., high precision and non-dependence on light). However, using LiDAR in underground pipeline environments can cause a loss of constraints on pipeline orientation since robots often move along the axis of the pipeline

(Figure 2). Furthermore, the highly similar environment in the pipeline and the lack of significant geometric features [13–16] can cause extremely poor results or may even result in failed map construction when using laser or vision sensors.



Figure 2. Schematic diagram of underground pipeline.

An example of this is that when a robot passes through a long, straight pipe (Figure 3), t_1 and t_2 are used to represent the radar measurements of the robot at two different moments in the environment that show high similarity, respectively. It is suggested that despite the two different moments, there is little difference in the number of radar measurements, which is most likely to cause the unavailability of the radar data. From the point of view of the robot, it cannot pinpoint its position within this environment. Moreover, the direction of the pipe is not restricted, due to the similar height structure throughout the pipe, meaning that the robot cannot achieve self-localization. Likewise, it is difficult for vision sensors to obtain useful feature points or reference points in scenes that are highly similar, such that relying on vision sensors may also cause localization failure.

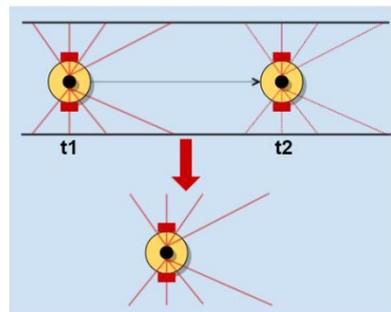


Figure 3. Schematic diagram of robot environment perception within the pipeline.

Previous authors [13] have suggested that LiDAR scan-matching performs poorly in environments without rich geometry (e.g., long hallways or large rooms). LiDAR scan-matching often loses confidence in estimating a particular position along the direction of the hallways or walls. To overcome this limitation, a mm-wave fixed antenna array radar is proposed as a low-cost aid to maintaining full positional information in the above situations. Localization is performed using a scan-matching algorithm that was developed using a correlative method.

In an earlier study [14], a factor graph-based framework was constructed for tightly coupled LiDAR/odometry/inertial measurement unit fusion for structural degradation in underground spaces. Moreover, a high-precision inertial measurement unit/odometry pre-integration model was built, while the factor graph algorithm was adopted to estimate the mobile robot motion states and sensor parameters.

A novel global localization algorithm, based on an environmental difference evaluation and correlation scan-matching fusion, has been proposed in the literature [15]. In this process, the surrounding space was evaluated using an a priori insight into the environment, based on a linear fit. Subsequently, the corresponding evaluation and positioning results from the correlation scan-matching were modified through dynamic selection, based on a position-updating strategy.

The authors of [16] aimed to solve the SLAM problem in the high-similarity environments found in tunnels. The authors have suggested that the robot is capable of advancing within the tunnel, not more than straight, but rather toward the walls on both sides. On that basis, the robot calculates the angular and distance differences according to the previous moment used for localization, using a segmentation algorithm.

Furthermore, for this type of degradation, which is caused by environmental similarity, two commonly used methods are proposed in the existing research. One method is to reduce the localization error problem caused by this factor via determining the direction and extent of laser degradation and filtering, or by compensating for the fusion of degradation information. In this regard, Ji Zhang presented a degenerate treatment of optimization problems at IROS 2016 [17]. He has suggested that the optimization-based state estimation problem can be expressed as a least-squares problem, which can also be linearized for nonlinear problems, while a state estimation problem comprises multiple constraints. He defined a degradation factor, D , to determine the degradation direction. On that basis, the error can be reduced by rounding off the information in the degradation direction during the optimization. In the literature [18], the ESKF was employed for localization, and the degradation direction of the laser was determined. The singular value decomposition (SVD) was performed on the constraint matrix of the laser beam; the eigenvector corresponding to the smallest singular value is the degradation direction. In the literature [19], the degraded directions were examined in a tunnel environment, and the problem of localization in a single geometric environment (e.g., a tunnel) was solved using a UWB (ultra-wide band) and LiDAR fusion, which is considered a further advance in the literature [18] in terms of examining the confidence of LiDAR localization calculations. Since degraded information is difficult to filter out or compensate for accurately in the real environment, the application of the above methods cannot satisfy the actual needs.

The other common method is to combine a high-precision differential odometer and IMU to solve problems with other sensors. A photoelectric tachometer sensor odometry wheel or inertial measurement unit (IMU) voyage position projection has usually been employed for positioning [13], to ensure that the positioning results are not affected by the surrounding environment. The combination of the IMU and odometer solves the problems of cumulative error and the unreliability of a single sensor, to a certain extent. In the literature [14], the tightly coupled fusion method of factor graph-based odometers, inertial measurement units, and LiDAR was adopted for the environmental sensing of underground spaces or tunnels. However, in a pipeline environment, the problem of tire slippage can easily occur due to the conveyed medium and the pipeline wall, thus causing a significant increase in system errors. The premise is that the position information after fusing the IMU and odometer data is reliable within a short time, while the premise for distance is that the accuracy of IMU should be sufficiently high. Based on the two integral points required for location information and the actual noise, the cumulative error tends to increase with the extension of the operation of the system, thus making the robot's estimated position beyond the pipeline a constrained space, with low overall robustness and reliability.

To solve the above problems, a multi-robot cooperation-based pipeline detection method is proposed in this study, based on LIO-SAM. This method is applicable to wheeled robots and can efficiently perform SLAM tasks in spaces with high environmental similarity (e.g., pipelines and tunnels). The primary idea is to remove the LIO-SAM laser-matching part with the high-precision differential odometry, the IMU inertial navigation sensor, and the ultrasonic sensor as the major sources of localization data. In the pipeline environment, two robots are formed into a front-to-back queue, and a unique periodic creeping method is developed as a cooperation strategy between the two robots. A high-quality multi-robot pose constraint factor (ultrasonic range factor) is constructed to restrain the relative positioning of robots, such that the robot queue can provide references for each other when moving through the pipeline, thereby achieving high-quality pose correction, and also achieving higher positioning precision. The proposed method overcomes the limitation

that the existing SLAM technique has a poor success rate in pipeline environments due to the high environmental similarity.

The chapters of this study are organized as follows. In Section 2, the method and model of this study are primarily explained. Section 3 is the experimental part of the paper, in which the method proposed in this study is simulated, and the relevant comparative experiments are performed. Finally, Section 4 provides a discussion and a summary of this study.

2. Methods and Models

2.1. Robot Queue

In the pipeline environment, the Leader and Follower robots are formed into a front-to-back queue (Figure 4). The specific configuration is presented as follows.

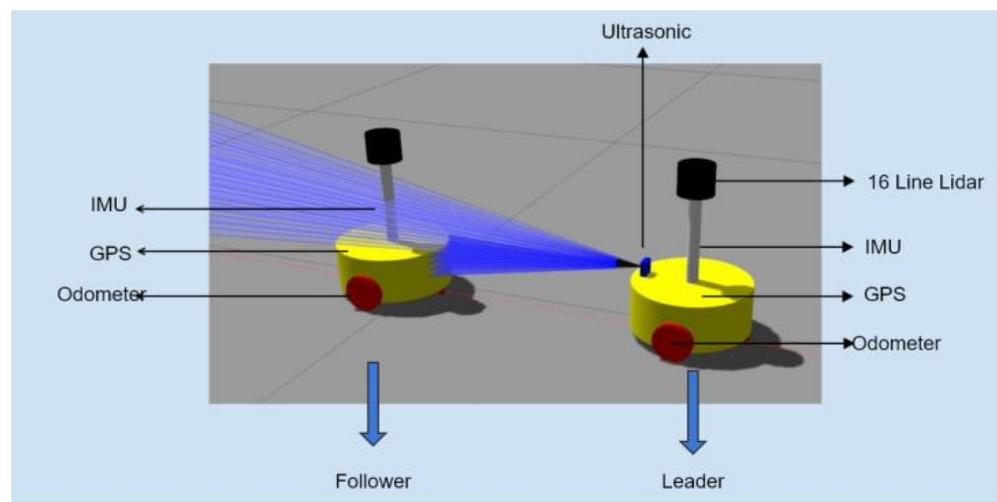


Figure 4. Multi-robot queue.

1. The Leader and the Follower use broadcasting to communicate with each other, to exchange data and information.
2. Each respective robot is equipped with a high-precision wheeled odometer and inertial navigation sensors to establish the basic positioning of its movement.
3. The Leader carries ultrasonic sensors to observe the Follower and then acquires the observation value of the relative distance between the Leader and the Follower. The Leader receives the observation value and then sends it to the Follower in real time.
4. Since GPS signals can be received at the entrance and exit locations of the pipe, both robots are equipped with GPS receivers, thus further reducing cumulative errors when using GPS signals at the entry and exit locations.
5. A 16-line LiDAR is installed on the Leader for point-cloud projection, using its fused poses to complete the 3D reconstruction of the pipeline environment.

2.2. Cycle Creep Cooperation Strategy

The robot queues move along the axis of the pipeline, in accordance with a cyclic peristaltic collaborative strategy. The respective peristaltic cycle comprises two stages, including extension and contraction. For ease of description, an example of moving forward is illustrated in Figure 5.

- A. Stage 1 is the extension stage. At this stage, the Leader plays the role of the mover, and the Follower plays the role of the reference point. The Leader moves forward and stops when the distance to the Follower is D_{max} , while the Follower position remains stationary.
- B. Stage 2 is the contraction stage. At this stage, the role of the Leader is converted to a reference, while the role of the Follower is transformed into a mover. The Leader is

stationary, and the Follower's movement is transformed into the movement direction of the Leader at the minimum contraction distance, D_{min} .

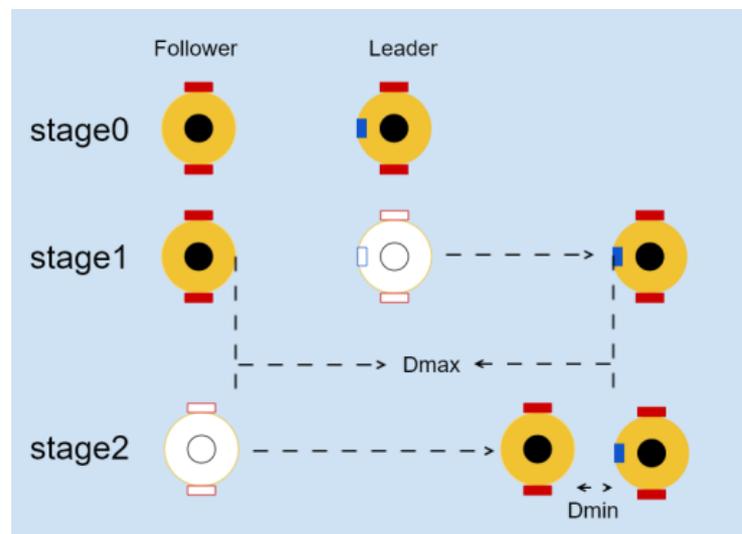


Figure 5. Diagram of creep-forward cooperation.

During the whole crawling cooperation cycle, the wheeled odometer and the IMU data of the mover are first fused, to obtain its estimated position and attitude. Subsequently, the relative translation between the mover and reference is obtained and combined with ultrasonic observations to develop an ultrasonic factor. Next, the ultrasonic factor is added as a multi-robot position constraint factor to the factor graph of the movers for information fusion, so as to correct the estimated position of the movers. In addition, the reference is stationary, and it is not necessary to correct its position. This, therefore, ensured that during a collaborative cycle, the Leader and the Follower both have the opportunity to correct their poses and can then achieve higher positioning accuracy, since the Leader and the Follower become, alternately, the mover and the reference. GPS signals are added to the factor graph as a special priority for fusion when the robot receives them at the inlet and exit points of the pipe, thus minimizing the cumulative errors. To achieve reliable communication and reduce the effect of unexpected interference on the formation, it is necessary to adjust and limit the maximum extension distance, D_{max} , in accordance with the actual hardware parameters, especially when the pipe bends, as well as to appropriately reduce the maximum extension distance.

In addition, robot queues do not always move in a straight line when the cyclic creep cooperation strategies are implemented. If the queue moves in a different direction, the reliability of ultrasonic observation and the robot-following problem should be considered.

1. Ultrasonic observation reliability problem.

At the back end of the Leader, pointing toward the Follower, the ultrasonic sensor is deployed and is used for observation of the Follower, to obtain the relative distance between the Leader and the Follower. However, when a Leader changes its direction of motion, the Follower may be out of its measuring range, thus resulting in there being no ultrasonic measurement signal or ongoing measurement of other objects. As a result, an unreliable ultrasonic observation is caused, thus increasing the system error and generating positioning errors.

To solve the above problem, a tripod head should be installed for the Leader's ultrasonic sensor, and the rotation of the tripod head should be controlled by solving the deflection angle of the Follower in an ultrasonic tripod-head coordinate system. Thus, the detection direction of the ultrasonic sensor always points to the center point of the Follower, to ensure the reliability of ultrasonic observation. The ultrasonic detection direction correction is illustrated in Figure 6. The Leader and the Follower move in the same direction

in Figure 6a, and it is not necessary to correct the ultrasonic sensing direction. When the Leader changes its direction in Figure 6b, we expect the ultrasonic Leader sensor to always point to the Follower's center, and the ultrasonic sensing direction should be corrected. The ultrasonic sensing direction is corrected in Figure 6c to point to the Follower center.

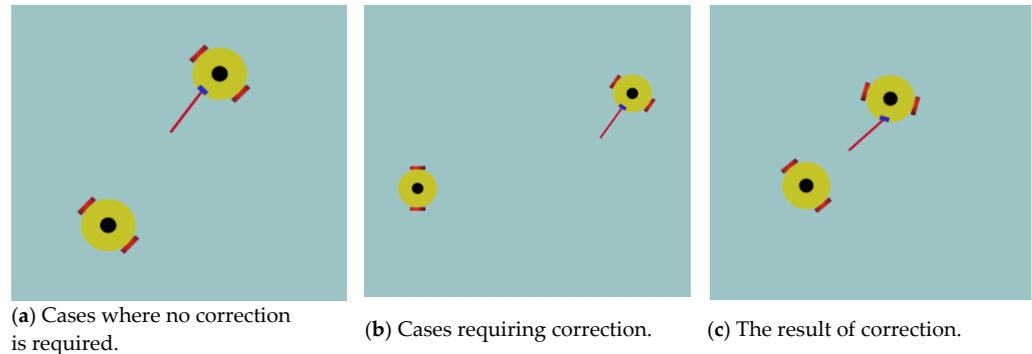


Figure 6. Ultrasonic detection angle correction diagram.

The world coordinate system is set as $\{W\} - O_W X_W Y_W$, while the ultrasonic-tripod head coordinate system on the Leader is set as $\{U\} - O_U X_U Y_U$. The two-dimensional plane position of the Follower under $\{W\}$ in the homogeneous coordinates is ${}^W P_F = [x_w \ y_w \ 1]^T$. Next, the homogeneous coordinates ${}^U P_F = [x_U \ y_U \ 1]^T$ of the Follower's position under $\{U\}$ can be obtained from Equation (1):

$${}^U P_F = {}^U_W T {}^W P_F. \quad (1)$$

The position relationship of the Follower with regard to the Leader coordinate system $\{U\}$ is assumed, as presented in Figure 7. α in the figure is expressed by Equation (2):

$$\alpha = \arctan \frac{y_U}{x_U}. \quad (2)$$

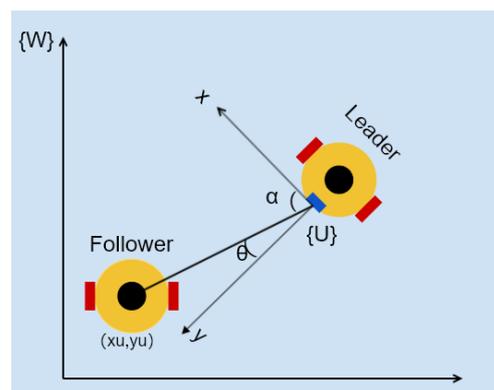


Figure 7. Position of the Follower with regard to the $\{U\}$ coordinate system.

Clockwise rotation is defined as positive, while counterclockwise rotation is defined as negative. Subsequently, the angle α by which the tripod head should rotate when the ultrasonic detection direction is corrected is written as in Equation (3):

$$\begin{cases} \theta = 90^\circ - \alpha (\alpha \geq 0) \\ \theta = -\alpha - 90^\circ (\alpha < 0) \end{cases} \quad \alpha \in [-180^\circ, 180^\circ]. \quad (3)$$

2. Robot following problem.

In complex environments, when a navigator changes its direction as a navigator, the following problem should be solved to avoid disrupting robot queues, since the navigator then cannot follow. The circumscribed area at the center of the Leader is denoted as the Leader's following domain, and the radius of the Leader's following domain is D_{\min} from the previous creeping cooperation strategy. In general, the following problem is to determine the speed and direction of the movement of the Follower at the respective moment, thus making the Follower reach the edge of the Leader's next domain quickly. It is assumed that at moment t , $(x_L^{(t)}, y_L^{(t)})$ and $(x_F^{(t)}, y_F^{(t)})$ are the coordinates of the Leader's and the Follower's two-dimensional plane position under $\{W\}$, respectively, then the velocity $v_F^{(t)}$ of motion at the center of the Follower at moment t is calculated by Equation (4):

$$v_F^{(t)} = k_p(\sqrt{(x_L^{(t)} - x_F^{(t)})^2 + (y_L^{(t)} - y_F^{(t)})^2} - D_{\min}) \quad (4)$$

where k_p is a positive scale factor, suggesting that $v_F^{(t)}$ is proportional to the distance between the Leader and the Follower, which can be selected experimentally in accordance with the robot motion parameters. The value of $v_F^{(t)}$ increases when the Leader and the Follower are far apart. As the distance between the Leader and the Follower decreases, the value of $v_F^{(t)}$ decreases. When the Follower reaches the edge of the Leader's following domain, $v_F^{(t)}$ takes the value of 0, thus avoiding collision between the robots.

Furthermore, when the Follower moves the tripod head toward the Leader, the magnitude and direction of the angular velocity should be constantly adjusted to correct the direction of motion. On that basis, the Follower can constantly be close to the Leader (Figure 8).

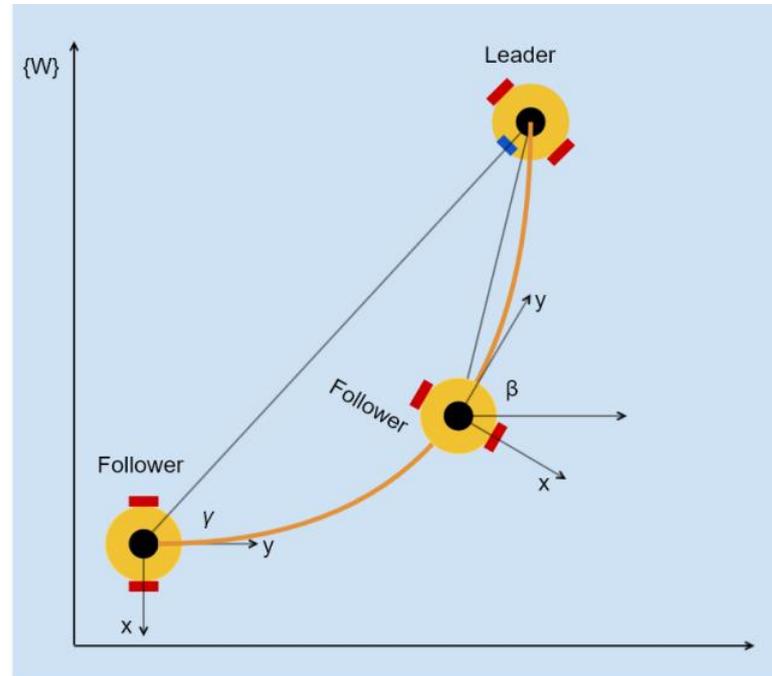


Figure 8. Robot following model.

In Figure 8, γ represents the angle of the line between the Follower and the Leader with the horizontal coordinate of the world coordinate system $\{W\}$; β represents the angle of the front direction of the Follower, with the horizontal coordinate that of the world coordinate system $\{W\}$. The Follower's angular velocity $\omega_F^{(t)}$ at time t is expressed by Equation (5):

$$\begin{cases} \gamma = \arctan \frac{y_L^{(t)} - y_F^{(t)}}{x_L^{(t)} - x_F^{(t)}} \\ \omega_F^{(t)} = k_\omega (\gamma - \beta) \end{cases} \quad (5)$$

where k_ω denotes a positive scale factor, thus suggesting that $\omega_F^{(t)}$ is proportional to the deviation angle $(\gamma - \beta)$ of the Follower pointing to the Leader, which can be selected experimentally, in accordance with the robot motion parameters. The larger the deviation angle of the Follower to the Leader, the larger $\omega_F^{(t)}$ becomes and takes on a larger value. As the deviation angle from the Follower to the Leader decreases, the value of $\omega_F^{(t)}$ becomes progressively smaller. When the Follower points toward the Leader, the deviation angle is 0 and $\omega_F^{(t)}$ takes the value of 0.

The motion control parameters of the left and right wheeled motors of the Follower robot at the respective time instant can be obtained by substituting x and y into the motion equation of the wheeled robot. The Follower is commanded to reach the destination position, according to its expected velocity and direction. The control algorithm model is illustrated in Figure 9.

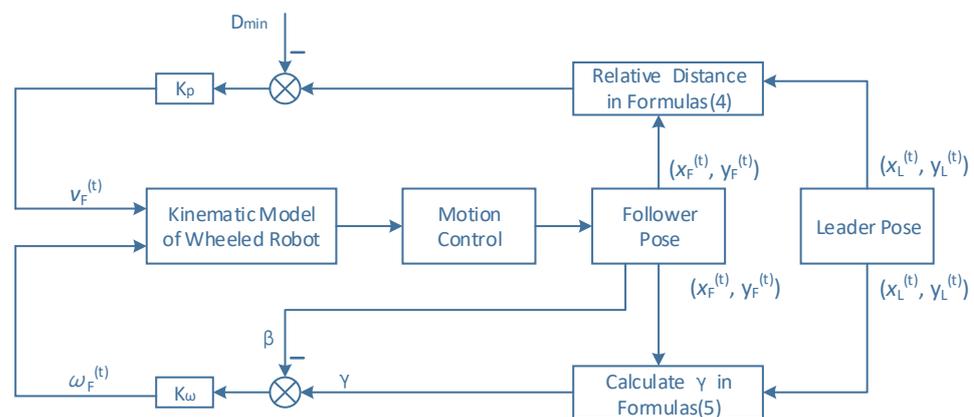


Figure 9. Robot-following control algorithm model.

2.3. SLAM Algorithm Model

In this study, based on the cycle creep cooperation strategy proposed previously, factor graph optimization is adopted to fuse the multi-sensor information regarding Leader and Follower robots to build a model of the SLAM algorithm for multi-robot cooperation.

2.3.1. Factor Graph Structure

The factor graph of the system comprises two factor graphs (including ares1 and ares2), representing the factor graphs of the two robots, i.e., the Leader (blue) and the Follower (green), respectively (Figure 10).

The above four main factors include the wheeled odometer factor, the pre-integration factor of IMU, and the GPS factor, as well as the multi-robot pose constraint factor (ultrasonic range factor).

- Wheeled odometer factor.

Wheeled odometry contains the location (position and attitude) and velocity (forward and angular velocity) information, which is acquired via an optical high-precision encoder system. The wheeled odometry factors are employed as the prior factors of both robots for factor graph optimizations.

- IMU pre-integration factor.

The optimization of the factor graph frequently adjusts the state quantities in the respective frames, thus making the IMU observations integrate repeatedly, leading to an increase in the system’s computational effort. As a result, the pre-integration of IMU is conducted in the optimization to avoid the problem of repeated integration [20–22]. The IMU pre-integration factor is obtained from IMU measurement integration between adjacent key-frames and is inserted into the factor graph as a constraint between two key-frames, to be optimized jointly with the Odom prior factor. For more information regarding the pre-integration factor of the IMU, please refer to the literature [20].

- GPS factor.

The GPS factor is acquired by the GPS receiver at the inlet and outlet locations of the pipeline, and it is inserted as a special a priori factor into the key-frame at the inlet and outlet locations of the pipeline for joint optimization with other factors.

- Multi-robot pose constraint factor (ultrasonic range factor).

The multi-robot pose constraint factor is adopted to constrain the pose between movers and references. It is a critical factor in achieving high-quality pose correction for the robots in this study. The core purpose is to build a multi-robot pose constraint factor by solving the relative translations between the estimated active pose and the reference pose and combining them with the relative distance observation. When robots collaborate with each other, the relative distance between robots is limited by D_{max} , due to communication issues and other factors. The ultrasonic sensors show the advantages of small size and low cost for such a relative position measurement over short distances. Moreover, they are not sensitive to ambient light and electromagnetic fields, while satisfying the measurement precision demands. Accordingly, the ultrasonic sensors mounted on the Leader are adopted to obtain the relative distance observations between the mover and the reference, such that a multi-robot pose constraint factor called the ultrasonic RANGE factor can be constituted.

If it is assumed that the relative translation between the active and reference points is obtained by solving for d_t at a certain time t , and the ultrasound observation is r_t , the optimized function of the ultrasound RANGE factor at time t is expressed in Equation (6):

$$f_{range} = \operatorname{argmin} |d_t - r_t|. \tag{6}$$

The above factor is inserted into the mover factor graph at the key-frame. In addition, the pose of the mover is continuously corrected by optimizing the factor graph, in accordance with the reference.

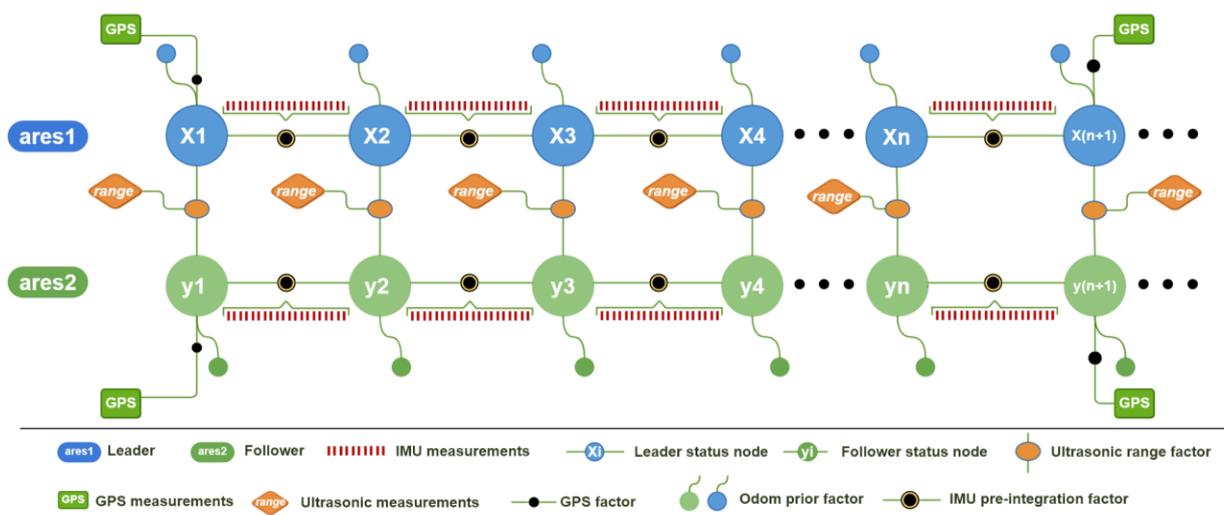


Figure 10. The system factor graph.

2.3.2. Factor Graph Optimization Solution

The factor graph optimization problem is solved using iSAM2. With the Leader as an example, the state with the highest probability is found by maximizing the product of all factors in factor graph solving, which is expressed in Equations (7) and (8).

$$\phi(x_i) = \psi(x_i, z_i) \propto P_i(z_i|x_i). \tag{7}$$

$$\phi(x) = \phi(x_1)\phi(x_2)\phi(x_1, x_2)\phi(x_2, y_1)\phi(x_2, y_2) \tag{8}$$

$$X^* = \underset{X}{\operatorname{argmax}} \phi(x) = \underset{X}{\operatorname{argmax}} \prod_i \phi_i(x_i). \tag{9}$$

In Equations (7) and (8), it is assumed that x_i is the system state variable and z_i is the observed variable. Thus, $\phi(x_i)$ is adopted to denote $P(z_i|x_i)$ since each observed variable is independent of the other. As a result, all the conditional probabilities are in the form of products and are decomposable. In the factor graph, the respective decomposition term is a factor. The products of all the factors can be expressed as a factor graph. It is assumed that y is the state variable of the robot follower, such that $\phi(x_1)$ and $\phi(x_2)$ can represent the a priori factor. $\phi(x_1, x_2)$ expresses the IMU factor. $\phi(x_2, y_1)$ and $\phi(x_2, y_2)$ denote the ultrasonic factor. The target here is to maximize the product of these factors, which is expressed in Equation (9).

Sensor noise generally follows a Gaussian distribution, such that all the factors satisfy an exponential model:

$$\phi_i(x_i) = \exp\left(-\frac{1}{2} \|f_i(x_i)\|_{\Sigma_i}^2\right) \tag{10}$$

If the observed values of the respective sensor types are Z , the error function of each individual sensor is expressed as follows:

$$\begin{cases} f_{\text{prior}}(x_i) = x_i - Z \\ f_{\text{imu}}(x_i, x_{i+1}) = (x_{i+1} - x_i) - Z \\ f_{\text{ultrasonic}}(x_{i+1}, y_i) = (x_{i+1} - y_i) - Z \end{cases} \tag{11}$$

Thus, the factorial graph solution is converted to a nonlinear least-squares problem, as follows:

$$X^* = \underset{X}{\operatorname{argmin}} \prod_i -\log(\phi_i(x_i)) = \underset{X}{\operatorname{argmin}} \sum_i \|f_i(x_i)\|_{\Sigma_i}^2 \tag{12}$$

2.3.3. Algorithm Flow

The respective creeping collaborative cycle corresponds to a local factor graph containing two parts (the expansion and contraction phases) (Figure 11). In the factor graph, the operating frequency uses Odom as the operating master frequency. Figure 12 shows the algorithm flow for processing the respective key-frame.

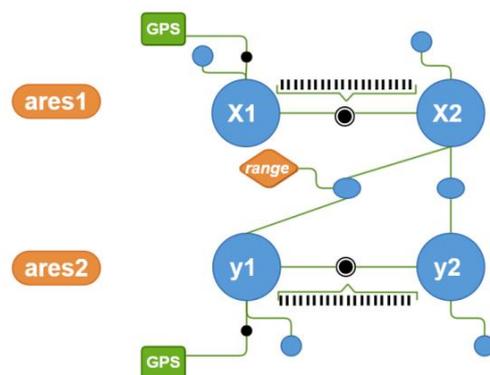


Figure 11. The local factor graph.

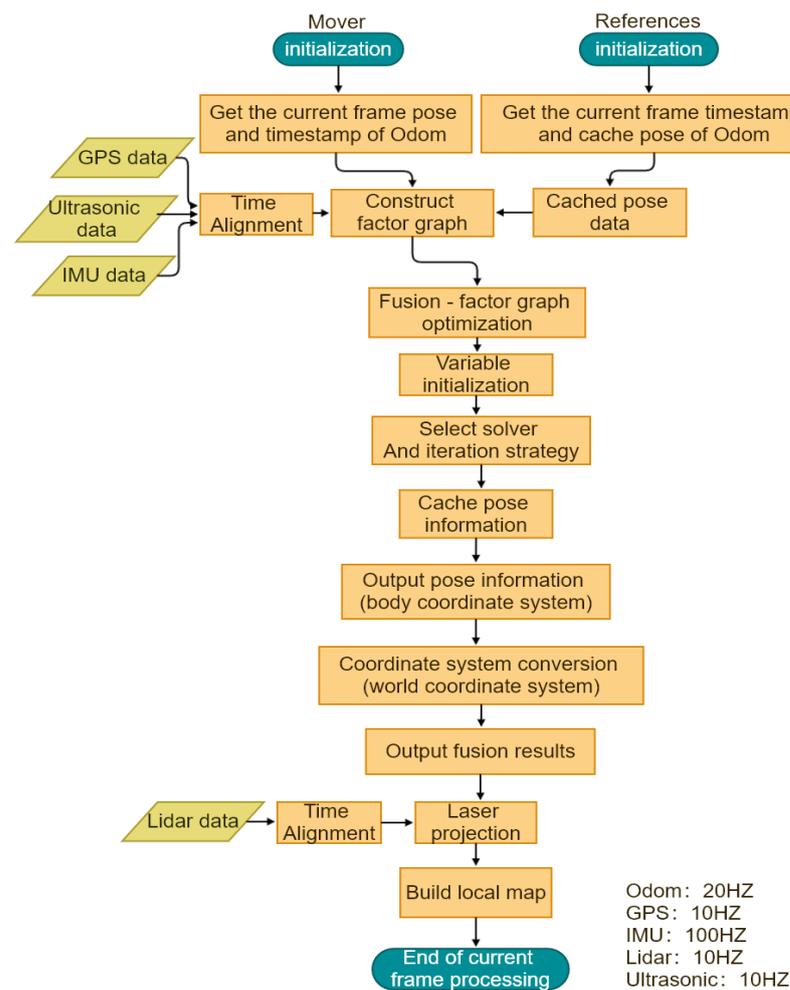


Figure 12. Algorithm flow chart.

- The Leader plays the role of the mover in the expansion phase, while the Follower plays the role of the reference. The mover (Leader) moves a tripod head and the position changes. In this case, the reference (Follower) is stationary, and the position stays the same. The Follower and the Leader in this process obtain the current frame poses of their respective Odoms with, the current and previous frame timestamps. The pose information is employed as a priori factor to construct the respective factor graphs, and the current frame timestamp and the previous frame timestamp are adopted to perform the temporal alignment for the respective sensor data. The reference (Follower) is in the stationary state and sends its cached information regarding its pose to the active (Leader). The leader uses the IMU data pre-integration between the current frame timestamp and the previous frame timestamp as a constraint to generate the IMU factor insertion factor graph, while it employs the reference (Follower) published pose information as a reference point to address the relative displacement between its Odom pose and the reference (Follower) pose. Moreover, it uses the ultrasonic measurement value and the relative translation amount to form an ultrasonic RANGE factor to be inserted into its factor graph to find the optimal solution, then the final pose information is output. As the active robot (Follower) moves forward, the position changes, the reference (Leader) is stationary, and its position remains unchanged.
- Likewise, in the contraction phase, the role of the Follower is the mover, and the Leader is the reference at that moment. The Follower and the Leader in this process obtain the current frame pose of their respective Odom, with the current and previous frame timestamp. Position information serves as a priori factor to construct respective

factor graphs, and the current frame timestamp and previous frame timestamp are adopted to perform a temporal alignment for the respective sensor data. The reference (Leader) is stationary, and it sends its buffered information regarding its own position and ultrasound measurements to the active one (Follower). The Follower takes the pre-integration of IMU data between the current frame timestamp and previous frame timestamp as a constraint, to generate the IMU factor insertion factor graph. Moreover, it takes the reference position information, when released by the Leader as a reference point, thus solving the relative displacement between its own Odom position and the reference (Leader) position. Furthermore, it uses ultrasonic measurement value and relative translation quantity to form the ultrasonic RANG factor, to be inserted into its factor graph for the optimal solution, final position, as well as final position.

- Furthermore, during the robot forward queue, point cloud projection is performed by the Leader, mounted on 16-line LiDAR using the final pose information output by fusion, to complete the 3D reconstruction of the pipeline.

3. Experiment

3.1. Experimental Design

The proposed SLAM method was validated experimentally through computer simulation in this study, based on the constraints of the real environment of the pipeline. The experimental environments were implemented using ubuntu 20.04 and the ROS system (noetic). Two simulated Leader and Follower robots were generated, respectively, under two namespaces and using two-wheeled differential drive robots with a structure as presented in Figure 4. It was modeled as a cylindrical body, with a diameter of 150 mm, a height of 70 mm, a diameter of the robot tire of 70 mm, and a width of 30 mm. In addition, the Leader, versus the Follower, released ultrasound topics and LiDAR topics for fusion and map construction, as well as sensor data registration under Gazebo simulation conditions. Four aspects were devised in the experiment as follows. SLAM testing in a common environment. To verify the applicability of the method in this study to the common environment, we incorporate the following variables:

1. SLAM testing in a simple pipeline environment. To verify the applicability and benefits of this study's method in the pipeline environment, SLAM testing was performed, and the method in this study was compared with the LIO-SAM algorithm, the FAST-LIO2.0 algorithm, and the conventional algorithms fusing IMU and Odom.
2. SLAM testing in a complex urban pipeline environment is used to verify the applicability of the method in this study in a larger-scale and more complex urban pipeline environment. The pipeline model was derived from the actual natural gas pipeline network distribution map of Kunming city; the local pipeline network was modeled using Solidworks and was then converted to the URDF format.
3. EVO evaluation analysis. The SLAM algorithm evaluation tool, EVO, was adopted to quantitatively assess and analyze the SLAM results of the method used in this study, in a complex urban pipeline environment.

3.2. SLAM Testing in General Environments

The common setting for testing is a room with multiple objects, having different geometrical characteristics (Figure 13).

Only the proposed method in this study was verified since the applicability of the LIO-SAM algorithm, FAST-LIO2.0 algorithm, and Odom IMU fusion method were verified in common environments. In the following scenario, the two robots were trained in a queue, as expressed in the previous section, and environmental data were collected in the room according to a creeping collaborative periodic strategy. The fusion effect in real situations was simulated by adding some Gaussian noise to the Leader and Follower's sensors, respectively, and the wheel slip of the robot was simulated at corners and walls.

The SLAM results under normal circumstances are presented in Figure 14.

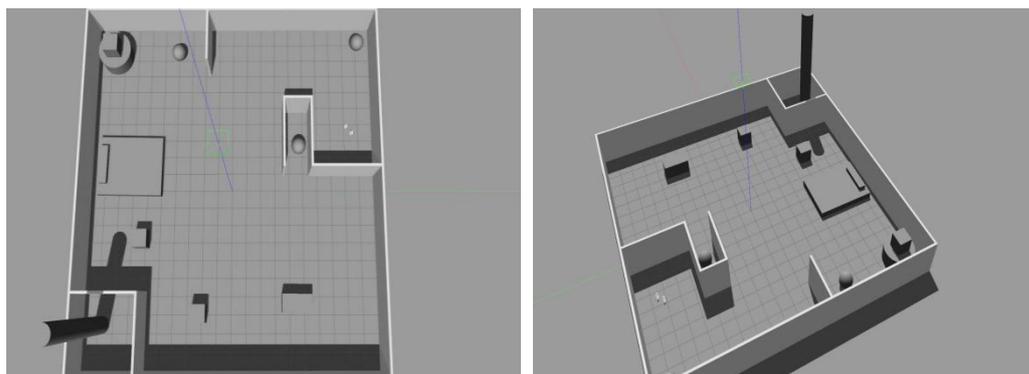


Figure 13. Room model.

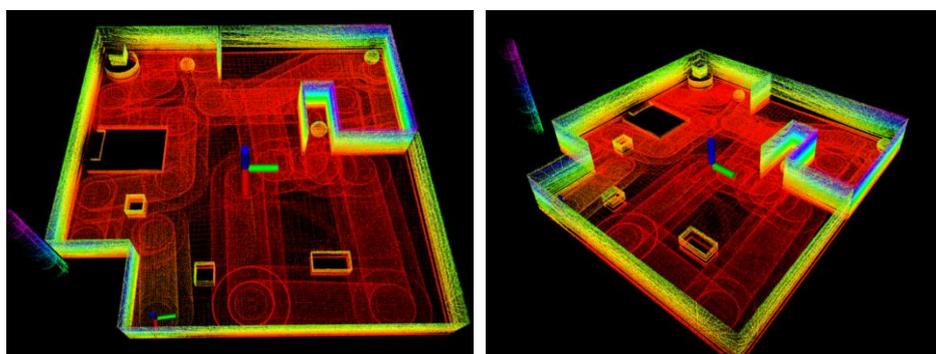


Figure 14. The SLAM test results of the proposed method in the common environment.

As depicted in Figure 14, this method is capable of completing its positioning in the room and the 3-D reconstruction of the room in the common environment, as well as closely recreating the map and indicating the geometric features and depth information of the room. The result suggests that the method used in this study is capable of completing the SLAM task in an ordinary environment; it also applies to a common environment.

3.3. SLAM Test in Simple Pipeline Environments

A pipe with two long straight bends, four-way bends, and 90-degree bends was used to represent a common pipe environment containing a variety of typical pipe configurations. This model was trained using SolidWorks and then converted to a URDF model and imported to the Gazebo simulation environment for testing. The pipe model is illustrated in Figure 15. In the SLAM test, Gaussian noise was added to the robot sensors, and wheel slip was simulated at the pipe turns.

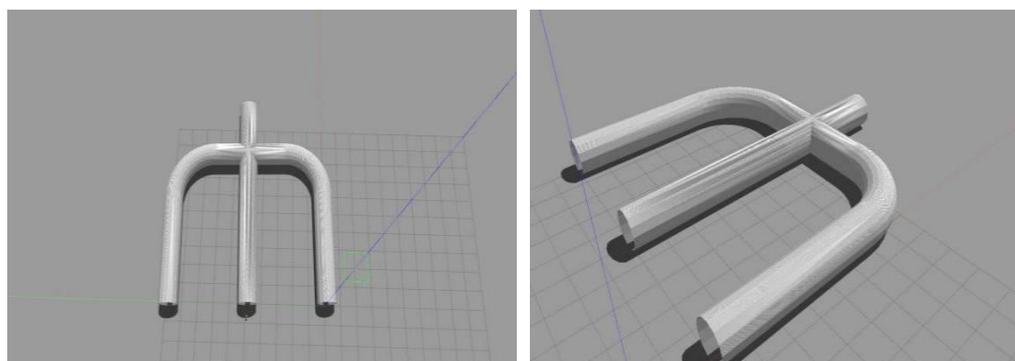


Figure 15. Simple pipeline model diagram.

The method of this study is based on the improvement of LIO-SAM, therefore the feasibility of LIO was first tested in a common pipeline setting. After the change of the profile and after performing the corresponding calibration, the test was performed (Figure 16). The LIO-SAM system emitted a warning after entering the pipeline that there are not enough feature points (Figure 16a); therefore, a system error occurs after a certain operating period. From the point of view of the error message prompt, the cause is an error due to insufficient variable constraints, while mathematically, the system is in an indeterminate state (Figure 16b). As shown from the test results, the laser matching in LIO-SAM fails in the context of laser degradation scenarios, such as long corridors or pipelines. Since there are no significant geometric features in the pipeline, the results of the LiDAR scan are consistent everywhere, resulting in the robot not being able to determine its location and thereby causing the SLAM task to fail.

```

/home/ldf/catkin_ws/src/LIO-SAM-master/launch/run.launch http://localhost:11311
ekf_gps (robot_localization/ekf_localization_node)
lio_sam_featureExtraction (lio_sam/lio_sam_featureExtraction)
lio_sam_imageProjection (lio_sam/lio_sam_imageProjection)
lio_sam_inuPreIntegration (lio_sam/lio_sam_inuPreIntegration)
lio_sam_mapOptimization (lio_sam/lio_sam_mapOptimization)
lio_sam_rviz (rviz/rviz)
navsat (robot_localization/navsat_transform_node)
robot_state_publisher (robot_state_publisher/robot_state_publisher)

auto-starting new master
process[master]: started with pid [16907]
ROS_MASTER_URI=http://localhost:11311

setting /run_id to 24f835d8-f9c9-11ec-82c7-000c29d60087
process[rosout-1]: started with pid [16920]
started core service [/rosout]
process[lio_sam_inuPreIntegration-2]: started with pid [16937]
process[lio_sam_imageProjection-3]: started with pid [16938]
process[lio_sam_featureExtraction-4]: started with pid [16940]
process[lio_sam_mapOptimization-5]: started with pid [16956]
process[robot_state_publisher-6]: started with pid [16969]
process[ekf_gps-7]: started with pid [16972]
process[navsat-8]: started with pid [16996]
[ INFO] [1656740285.746291212]: ----> Feature Extraction Started.
[ INFO] [1656740285.820930855]: ----> Map Optimization Started.
[ INFO] [1656740285.871891819]: ----> IMU PreIntegration Started.
[ WARN] [1656740388.257667587]: Not enough features! Only 15 edge and 100 planar features available.

ldf@ldf-virtual-machine: ~/catkin_ws
process[lio_sam_featureExtraction-4]: started with pid [18159]
process[lio_sam_mapOptimization-5]: started with pid [18176]
process[robot_state_publisher-6]: started with pid [18186]
process[ekf_gps-7]: started with pid [18197]
process[navsat-8]: started with pid [18204]
process[lio_sam_rviz-9]: started with pid [18215]
[ INFO] [1656740646.448199069]: ----> Feature Extraction Started.
[ INFO] [1656740646.464072354]: ----> Image Projection Started.
[ INFO] [1656740646.511337666]: ----> Map Optimization Started.
[ INFO] [1656740646.555641891]: ----> IMU PreIntegration Started.
[ WARN] [1656740678.165545410]: Not enough features! Only 15 edge and 100 planar features available.
terminate called after throwing an instance of 'gtsam::IndeterminantLinearSystemException'
what():
Indeterminant linear system detected while working near variable
8646911284551352381 (Symbol: x61).

Thrown when a linear system is ill-posed. The most common cause for this error is having underconstrained variables. Mathematically, the system is underdetermined. See the GTSAM Doxygen documentation at http://borg.cc.gatech.edu/ on gtsam::IndeterminantLinearSystemException for more information.
lio_sam_inuPreIntegration-2 process has died [pid 18156, exit code -6, cmd /home/ldf/catkin_ws/devel/lib/lio_sam/lio_sam_inuPreIntegration --name=lio_sam_inuPreIntegration --log=/home/ldf/.ros/log/fbf9c80c-f9c9-11ec-82c7-000c29d60087/lio_sam_inuPreIntegration-2.log].
log files: /home/ldf/.ros/log/fbf9c80c-f9c9-11ec-82c7-000c29d60087/lio_sam_inuPreIntegration-2*.log
lio_sam_inuPreIntegration-2 restarting process
process[lio_sam_inuPreIntegration-2]: started with pid [19196]
[ INFO] [1656740731.119515394]: ----> IMU PreIntegration Started.

```

(a) Information on LIO-SAM runtime.

(b) LIO-SAM error messages.

Figure 16. LIO-SAM test results within a simple pipeline environment.

Subsequently, the feasibility of FAST-LIO2.0 was verified in this setting. FAST-LIO 2.0 [23] is more complete, compared with FAST-LIO [24], and an incremental kd-Tree (ikd-Tree) value was proposed on top of it, thus enabling kd-Tree to incrementally remove or add point clouds. By taking advantage of ikd-Tree, FAST-LIO2.0 no longer extracts edge and plan features similar to LOAM, but instead, directly aligns the respective 3D points with the map, such that it can operate more stably in some scenarios where manual feature extraction is more challenging. Figure 17 presents the results of the tests. FAST-LIO2.0 is capable of reconstructing part of the map (Figure 17a); however, the reduction was poor, and the point cloud map started to distort with the extension of the running time. The invalid point alert shown in Figure 17b continued to appear during the run until the system was stopped. This figure shows that FAST-LIO2.0 can achieve a stable performance in some environments where manual feature extraction is more challenging, whereas it fails in environments with high similarity and almost no geometrical features (e.g., pipes),

especially in long, straight pipes; the SLAM task failure problem is due to the occurrence of insufficient feature points.



(a) Information on FAST-LIO2.0 runtime.

(b) FAST-LIO2.0 error messages.

Figure 17. FAST-LIO2.0 test results under a simple pipeline environment.

To avoid the sensitivity of LiDAR to a high similarity environment, a feasibility test under a normal pipeline environment was performed using the IMU Odom fusion method, which is not easily affected by the environment and the methods used in this study. Figure 18 presents the test results of using factor graphs to fuse Odom and IMU, along with the test results of the method used in this study.

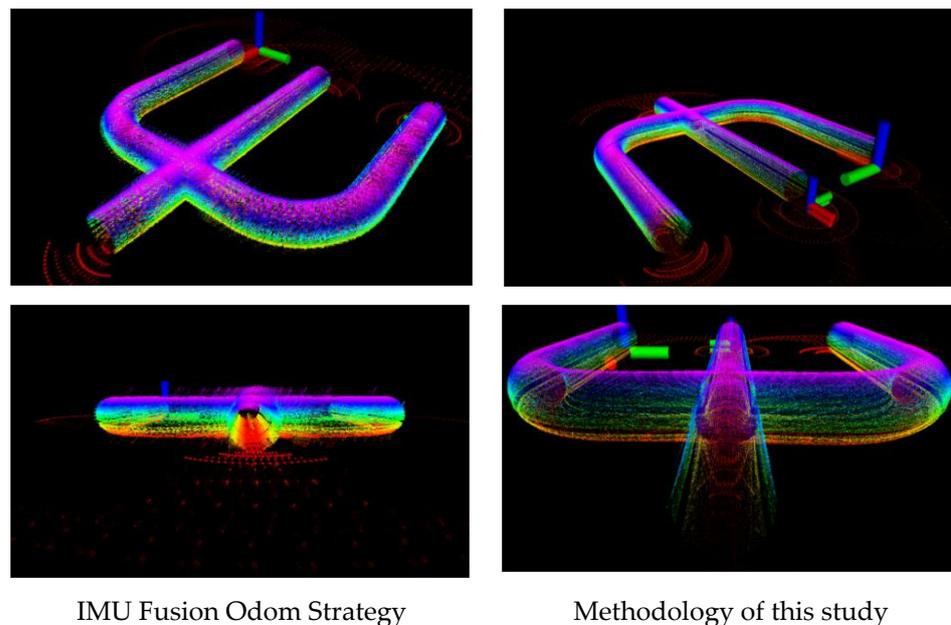


Figure 18. Comparison results between the IMU Odom fusion strategy and the SLAM test of this study method, in a simple pipeline environment.

As depicted in the figure, the IMU fusion Odom strategy can work stably in a normal pipeline environment, whereas the SLAM task is not performed well, with more severe map ghosting and considerable noise points at the corners and in four directions. This is because the point cloud is repeatedly projected to the same location but at different positions. Moreover, after the tire slip occurs at the corner by the single robot, a point clouds overlay occurs. Therefore, the SLAM task is not completed well, although the Odom IMU fusion method can work stably in a normal pipeline environment. Notably, when the pipe size increases, and the number of bends increases, the cumulative error will increase in size, thus resulting in considerable numbers of noisy points and ghosting in the point-cloud map, such that a lower confidence level or even unusable results are achieved. The test results of the method used in this study, given in Figure 18, suggest that compared with the test result of the LIO-SAM, FAST-LIO2.0, and IMU Odom fusion methods, the

method in this study can work stably in a simple pipeline environment, and the SLAM tasks are achieved well. In the localization of the simulated robot wheel slip (e.g., curves), it can effectively correct the errors generated by the slip of the wheeled odometers of the individual robots, and there are no issues (e.g., map ghosting and many noise points). The results after 3D reconstruction suggest that the structure of the pipeline is highly restored, and the directional and depth information are clearly visible. The test results reveal that the method in this study benefits from the advantages of multi-robot cooperation, and the two robots are capable of alternately using each other as a reference in a creeping collaborative cycle to make high-quality corrections of their poses, thus significantly reducing the effect of errors when the robots' wheeled odometers slip, allowing LiDAR to perform point-cloud projection in the correct positions. Furthermore, there are no problems (e.g., map ghosting and many noisy points in the results), and the robots' poses are more accurate.

In this section, the experiments were performed to verify the feasibility of LIO-SAM, FAST-LIO2.0, the IMU Odom fusion strategy, and the current method for SLAM tasks in a simple pipeline. The following conclusions were drawn through the above experimental tests, as listed in Table 1.

Table 1. Feasibility verification conclusions of the respective algorithm on a simple pipeline.

Algorithm	Operation Status	Completion Degree	Noise	Map Ghosting
LIO-SAM	Failure	Incomplete	×	×
FAST-LIO2	Failure	Incomplete	×	×
IMU Fusion Odom Strategy	Success	Completed	More	Serious
Method of this study	Success	Completed	Less	None

3.4. SLAM Test in a Complex Urban Pipeline Environment

The pipeline model was derived from an actual distribution map of the Kunming natural gas pipeline network; the local pipeline network was modeled using Solidworks and was then converted to a URDF format model and imported into the Gazebo simulation environment. The pipeline model is presented with the original map in Figure 19. Compared with the previous simple pipeline environment, the scale and complexity of the pipeline increase significantly, including environments (e.g., long straight pipes with no significant geometrical characteristics) and scenarios where the pipeline direction is unconstrained, as well as various pipeline structures (e.g., tees, crossovers, and turns).

1. Types of piping include a long straight pipe of about 2230 m, one tee pipe, two four-way pipes, and nine multi-angle bends.
2. The total length of the pipeline is about 4000 m.
3. The pipe's inner diameter is 1016 mm.

During the feasibility test in the previous section, both the method in this study and the single robot using the IMU fusion Odom strategy can accomplish the SLAM task in the simple pipeline, whereas there is a certain gap between the two methods in terms of accuracy, as well as of error. Thus, SLAM tests were performed in this section for the above two methods in simulated real city pipelines, and the two methods were compared. The test comparisons are presented in Figure 19.

The results in Figure 19 suggest that in the single robot SLAM results, using the IMU Fusion Odom strategy for the real city simulation pipeline, the 3D map has considerable noise and multiple map ghosts, thus causing severe distortion of the local results. There was no significant noise and map ghosting in the SLAM results of this study, and the overall results were better than those of IMU fusion Odom, being higher than the IMU fusion Odom strategy in terms of robustness and reliability. As the pipeline size increases and the running time is extended, the method in this study still maintains a good SLAM effect, with the advantages of cumulative controllable error and high confidence and robustness. In the 3D pipeline map, the color does not represent the difference in the results, but simply a different coordinate system, selected in rviz.

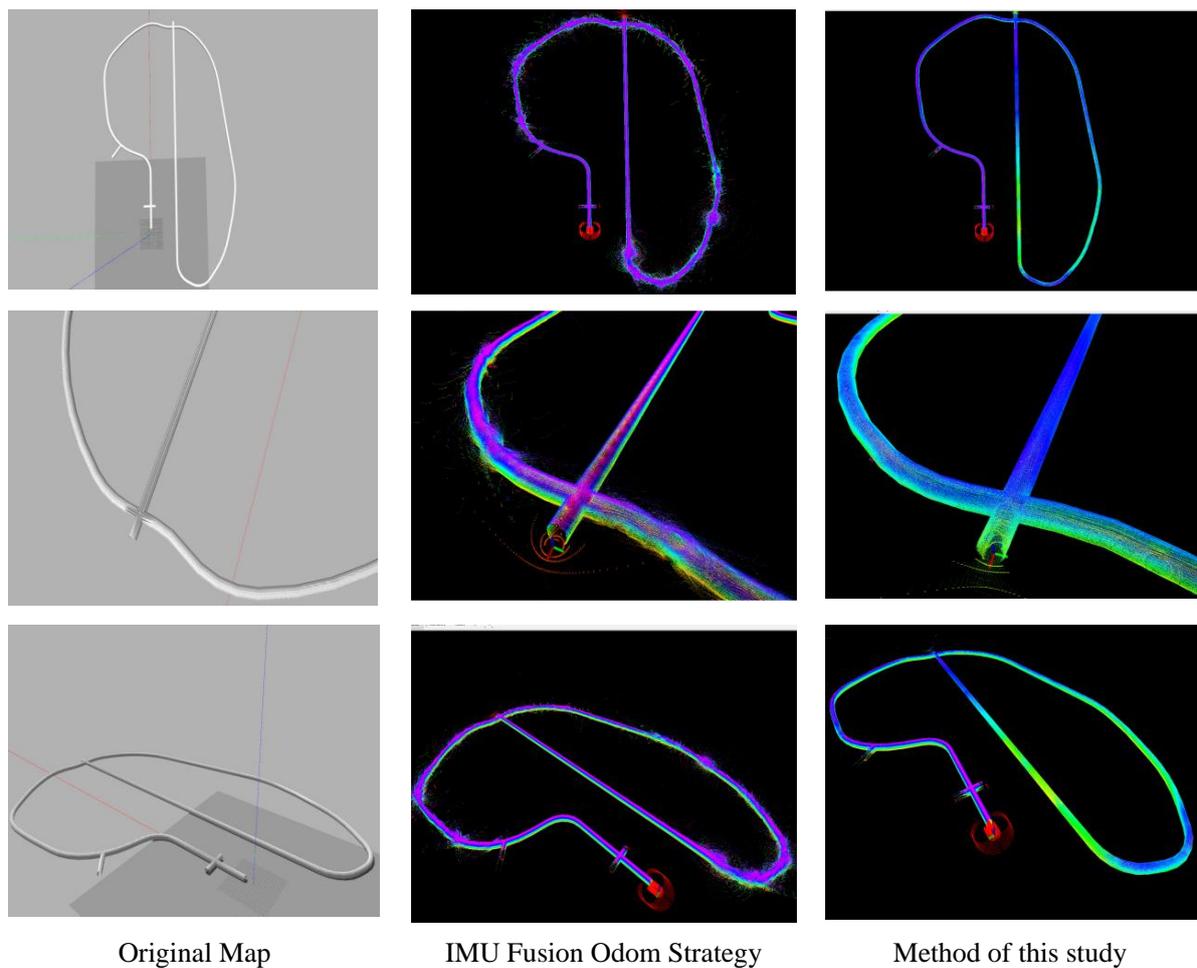
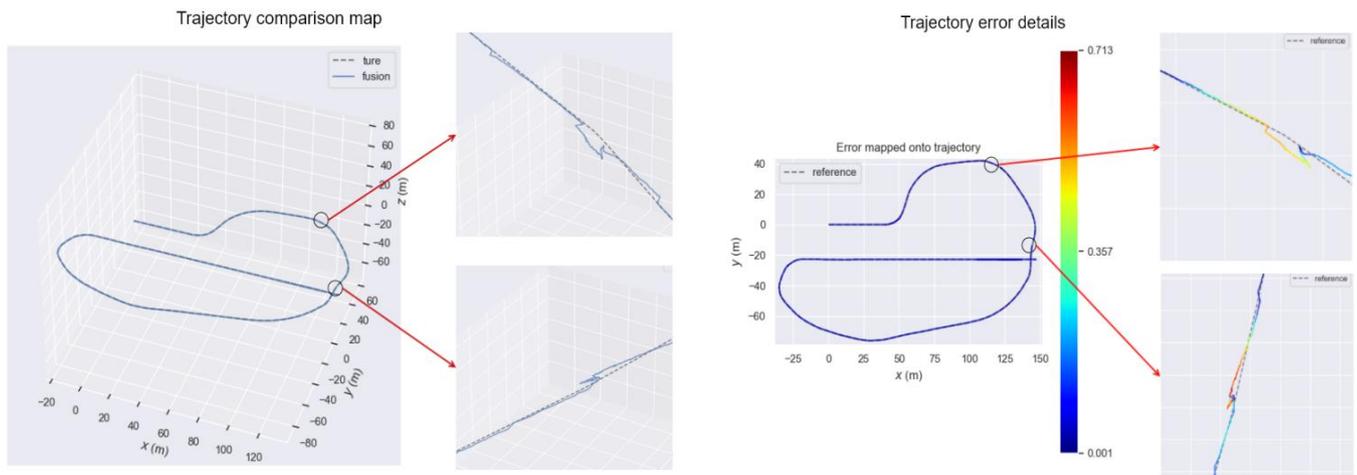


Figure 19. SLAM test results of the proposed method and the IMU fusion Odom strategy in a complex urban pipeline environment.

3.5. EVO Review Analysis

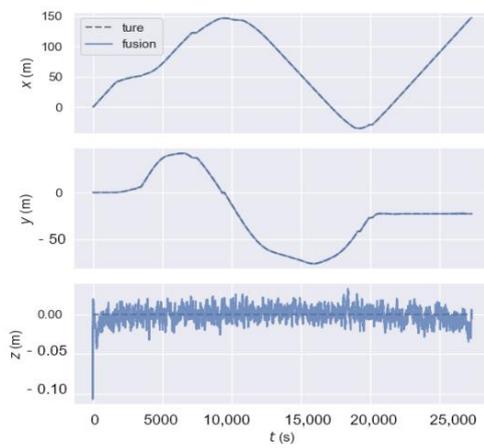
For the in-depth quantitative analysis of the SLAM effect of the method in this study, the EVO SLAM algorithm evaluation tool was employed for a quantitative evaluation and analysis of previous SLAM testing experiments of the methodology in a complex urban pipeline environment. The true value in the experiment was obtained by extracting the Leader's pose information before adding noise to the sensor, while the compared trajectory was the final Leader's fusion output result, and the data were in TUM format. The EVO evaluation analysis results are presented in Figure 20.

Figure 20a depicts a visual comparison of the trajectories, suggesting that the true value is consistent with the fused effect. The visual information of the trajectory error in Figure 20b and the APE (absolute pose error) histogram in Figure 20d suggests that the average error of the method in this study is 0.047 m, and the maximum error is 0.713 m for SLAM when used in the pipeline. The average and maximum errors are within a reasonable range, compared with the 1016 mm inner diameter of the pipe and 4000 m total length, respectively. The maximum error occurred at the corners of larger arcs, where the wheels of both robots may have slipped at the same time. However, the comparison of the pipeline 3D map in Figure 19 indicated that no map distortion or ghosting occurred at the corners. The results of the analysis reveal that the method in this study can excellently perform SLAM tasks in a complex urban pipeline environment. with good reproduction of the pipeline and low error, and it is capable of overcoming the problem of map ghosting caused by the tire slip at the corner.

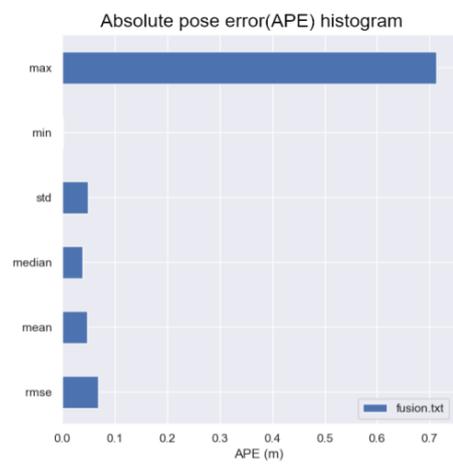


(a) Trajectory comparison map.

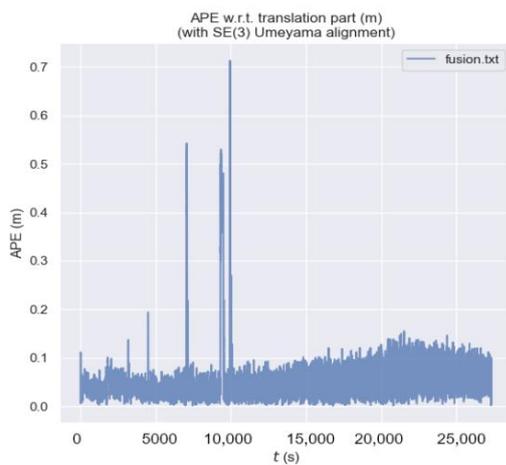
(b) Trajectory error details.



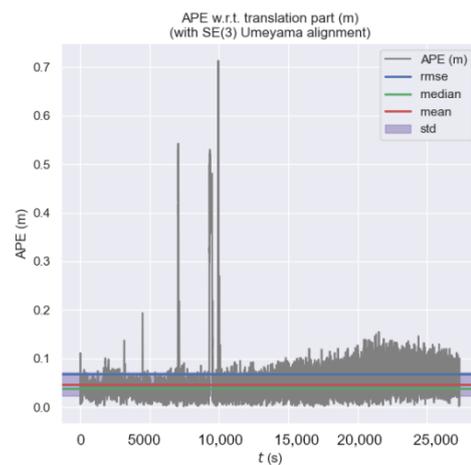
(c) Track comparison map (comparison in details).



(d) Absolute pose error (APE) histogram.



(e) Absolute trajectory error information.



(f) Absolute trajectory error details.

Figure 20. EVO evaluation results of the method in this study.

The results of the comparison of the IMU fusion Odom strategy and the methods in this study by the EVO evaluation tool are presented in Figure 21.

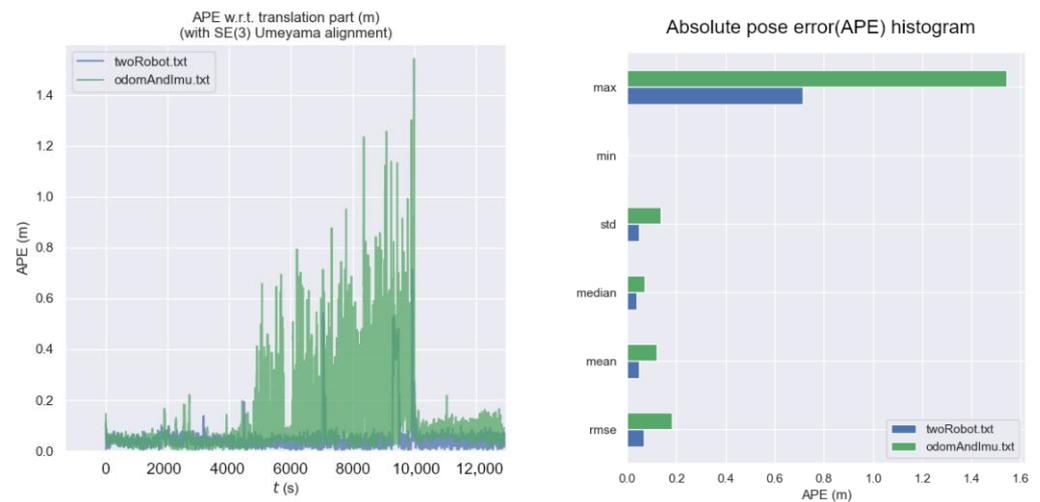


Figure 21. The results of the error analysis of the method in this study, compared with the IMU fusion Odom strategy.

As depicted in Figure 21, the maximum error when using the IMU fusion Odom strategy was nearly 1.54 m. The comparison of the above figure with the SLAM result graph in Figure 19 indicates that the map ghosting problem and considerable noise occurred at the bend, due to the possible occurrence of tire skidding at the bend. As a result, the maximum absolute error was more than 2.2 times that of the proposed method, and the average value was relatively high. The average value was nearly 2.5 times that reported using this method. The specific comparison result parameters generated by EVO are listed in Table 2.

Table 2. EVO analysis results.

Algorithm	RMSE (Root Mean Square Error)	Mean (Mean Absolute Error)	Median (Median Error)	Std. (Standard Deviation)	Min (Minimum Error)	Max (Maximum Error)
IMU Fusion Odom Strategy	0.1810	0.1199	0.0718	0.1355	0.0011	1.5434
Method used in this study	0.0681	0.0472	0.0391	0.049	0.0008	0.7131

4. Conclusions and Discussion

Using mainstream laser or visual SLAM techniques in certain scenarios (e.g., underground pipes) can easily cause a failure of map-building since there are no significant internal aggregated features and no longitudinal constraints on the LiDAR scans. Besides this, GPS signals cannot be received in underground pipes, such that accurate localization is unlikely to be possible. The above problem is solved in this study by proposing a multi-robot cooperation-based SLAM method for pipeline environments. The proposed method is capable of efficiently performing SLAM tasks in environments with high-similarity characteristics (e.g., pipelines and tunnels). The core of the proposed method is to remove the laser-matching part of LIO-SAM and employ three devices (including a high-precision differential odometer, an inertial navigation sensor IMU, and an ultrasonic sensor) that are not easily affected by the high similarity of environments to be the major positioning data sources. Two robots are trained in a front-to-back queue in the pipeline environment, and a single-period creep is designed as a cooperation strategy between the two robots. Moreover, a multi-robot pose constraint factor (the ultrasonic range factor) is constructed to constrain the robots' relative pose. On that basis, the robot queue can provide a mutual reference when traveling through the pipeline; it obtains a high-quality pose correction, thus achieving high positioning accuracy. The robot queue overcomes the disadvantage that the existing SLAM method is not effective in the pipeline environment due to its high environmental similarity.

Four aspects of the experiments were designed to validate the proposed method in this study, including SLAM testing in a common environment and SLAM testing in a simple pipeline environment, as well as SLAM testing in a complex urban pipeline environment and EVO evaluation and analysis. The experimental results suggest that laser-matching in LIO-SAM cannot be applied to laser degradation scenarios (e.g., long corridors or pipelines) since there are no significant geometrical features in the pipeline and the LiDAR scans are consistent everywhere. As a result, the robot cannot determine where it is, thus causing SLAM task failure. Even though FAST-LIO2.0 can work stably in some environments where manual feature extraction is more difficult, it also fails in environments with high similarity and almost no geometric features, such as pipes, especially in long straight pipes; as a result, the SLAM task failure problem occurs, due to insufficient feature points. Although the IMU Odom fusion strategy can work stably in a normal pipeline environment, the SLAM task is not completed well. Notably, when the pipeline scale increases and the number of turns increases, the cumulative error will become larger and larger, causing considerable amounts of noisy points and ghosting in the point-cloud map, which will cause lower confidence in the results, or they may even be unusable. The method in this study can operate stably in various pipeline environments; the robot position is more accurate, and the SLAM task is completed well. In particular, it can effectively correct the errors generated by the slip of the wheeled odometers of the individual robot at specific locations (e.g., the bends that simulate robot wheel slip), without problems (e.g., map ghosting and many noise points). The structure of the pipeline is significantly restored after a 3D reconstruction of the model, and the directional and depth information of the pipe are observed. The EVO evaluation results data indicated that the average error of SLAM in the pipe was 0.047 m, and the maximum error was 0.713 m. This error is within a reasonable range relative to the geometric parameters of the robot and the pipe in the experimental environment.

In brief, the following conclusions were drawn from the research work of this study.

1. Due to the specificity of the pipeline environment, some major SLAM methods (e.g., LIO-SAM, FAST-LIO2.0, and the IMU Odom fusion strategy) cannot achieve better performance and effectiveness in the pipeline setting, and they do not apply to the pipeline context.
2. Given the advantage of multi-robot cooperation, compared with existing methods, the method used in this study is not easily affected by the environment. Besides its applicability to ordinary environments (e.g., rooms), the method in this study achieves better performance in various pipeline environments with high similarity. Moreover, with the increase in pipeline size and operation time, the method in this study can still maintain a good SLAM effect, with the advantages of cumulative controllable error, high confidence, and robustness. Based on the above result, training robots in queues and designing efficient queue cooperation strategies takes on a critical significance for improving SLAM task performance and efficiency.
3. The multi-robot pose constraint factor (e.g., ultrasonic RANGE factor) is built using a multi-robot cooperation method. This method is capable of effectively reducing the covariance, and it is a critical factor for increasing the efficiency of SLAM and enhancing the robustness of the system. It is not limited to the ultrasonic method. The ultrasonic method can be replaced with other observation methods (e.g., laser, Bluetooth, and infrared) according to the actual environment and accuracy requirements, to satisfy the needs of different scenarios.

Author Contributions: Data curation, T.W.; Formal analysis, D.L.; Investigation, Y.Z.; Methodology, D.L., Y.Z. and Z.G.; Software, D.L.; Writing—original draft, D.L.; Writing—review & editing, D.L. and Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 51365019).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Restrictions apply to the availability of these data. Data was obtained from [Yunnan Petrochina Kunlun Gas Co.] and are available [from the authors/at 20202204163@stu.kust.edu.cn] with the permission of [Yunnan Petrochina Kunlun Gas Co.].

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cai, X.; Ning, H.; Dhelim, S.; Zhou, R.; Zhang, T.; Xu, Y.; Wan, Y. Robot and its living space: A roadmap for robot development based on the view of living space. *Digit. Commun. Netw.* **2021**, *7*, 505–517. [[CrossRef](#)]
2. Wang, Z.; Cao, Q.; Luan, N.; Zhang, L. Precision location technology of pipeline robot based on multi-sensor data fusion. *Robot* **2008**, *30*, 238–241.
3. Zhang, J.; Singh, S. LOAM: Lidar Odometry and Mapping in Real-Time. In Proceedings of the Robotics: Science and Systems, Berkeley, CA, USA, 13–15 July 2014.
4. Zhang, J.; Singh, S. Visual-lidar Odometry and Mapping: Low-drift, Robust, and Fast. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015.
5. Shan, T.; Englot, B. LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018.
6. Shan, T.; Englot, B.; Meyers, D.; Wang, W.; Ratti, C.; Rus, D. LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021.
7. Shan, T.; Englot, B.; Ratti, C.; Rus, D. LVI-SAM: Tightly-coupled Lidar-Visual-Inertial Odometry via Smoothing and Mapping. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021.
8. Zuo, X.; Geneva, P.; Lee, W.; Liu, Y.; Huang, G. Lic-fusion: Lidarinertial-camera Odometry. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 4–8 November 2019.
9. Zuo, X.; Yang, Y.; Geneva, P.; Lv, J.; Liu, Y.; Huang, G.; Pollefeys, M. LIC-Fusion 2.0: LiDAR-Inertial-Camera Odometry with Sliding-Window Plane-Feature Tracking. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021.
10. Lin, J.; Zheng, C.; Xu, W.; Zhang, F. R2LIVE: A robust, real-time, LiDAR-inertial-visual tightly-coupled state estimator and mapping. *IEEE Robot. Autom. Lett.* **2021**, *6*, 7469–7476. [[CrossRef](#)]
11. Lin, J.; Zhang, F. R3LIVE: A Robust, Real-time, RGB-colored, LiDAR-Inertial-Visual tightly-coupled state Estimation and mapping package. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022.
12. Zhu, Z.; Jiang, W.; Yang, L.; Luo, Z. Indoor multi-robot cooperative mapping based on geometric features. *IEEE Access* **2021**, *9*, 74574–74588. [[CrossRef](#)]
13. Zhang, A.; Atia, M.M. Comparison of 2D Localization Using Radar and LiDAR in Long Corridors. In Proceedings of the 2020 IEEE SENSORS, Rotterdam, The Netherlands, 25–28 October 2020.
14. Bai, S.; Lai, J.; Lü, P.; Ji, B.; Zheng, X.; Fang, W.; Cen, Y. Mobile robot localization and perception method for subterranean space exploration. *Jiqiren/Robot* **2022**, *44*, 463–470. [[CrossRef](#)]
15. Zeng, L.; Guo, S.; Xu, Z.; Zhu, M. An indoor global localization technique for mobile robots in long straight environments. *IEEE Access* **2020**, *8*, 209644–209656. [[CrossRef](#)]
16. Tardioli, D.; Villarroel, J.L. Odometry-less Localization in Tunnel-like Environments. In Proceedings of the 2014 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Espinho, Portugal, 14–15 May 2014.
17. Zhang, J.; Kaess, M.; Singh, S. On Degeneracy of Optimization-based State Estimation Problems. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016.
18. Zhen, W.; Zeng, S.; Soberer, S. Robust Localization and Localizability Estimation with a Rotating Laser Scanner. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017.
19. Zhen, W.; Scherer, S. Estimating the Localizability in Tunnel-like Environments using LiDAR and UWB. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019.
20. Forster, C.; Carlone, L.; Dellaert, F.; Scaramuzza, D. IMU Preintegration on Manifold for Efficient Visual-inertial Maximum-a-posteriori Estimation. In Proceedings of the Robotics: Science and Systems, Rome, Italy, 13–17 July 2015.
21. Forster, C.; Carlone, L.; Dellaert, F.; Scaramuzza, D. On-manifold preintegration for real-time visual-inertial odometry. *IEEE Trans. Robot.* **2017**, *33*, 1–21. [[CrossRef](#)]
22. Qin, T.; Li, P.; Shen, S. VINS-Mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Trans. Robot.* **2018**, *34*, 1004–1020. [[CrossRef](#)]
23. Xu, W.; Cai, Y.; He, D.; Lin, J.; Zhang, F. FAST-LIO2: Fast direct LiDAR-inertial odometry. *IEEE Trans. Robot.* **2022**, *38*, 2053–2073. [[CrossRef](#)]
24. Xu, W.; Zhang, F. FAST-LIO: A fast, robust LiDAR-inertial odometry package by tightly-coupled iterated kalman filter. *IEEE Robot. Autom. Lett.* **2021**, *6*, 3317–3324. [[CrossRef](#)]