RESEARCH ON SIMULTANEOUS LOCALIZATION AND MAPPING METHOD FOR ORCHARDS BASED ON SCAN CONTEXT AND NDT-ICP FUSION SCHEME
/
基于扫描上下文和 NDT-ICP 融合方案的果园同步定位与绘图方法研究

Zhen QIN [1)], Hongxia WANG [1)], Pengcheng LV *[2)] *
[1)] School of Information and Control Engineering of Qingdao University of Technology, Qingdao, China
[2)] Shandong University of Technology, Collage of Agricultural Engineering and Food Science, ZiBo, China
Tel: +86 13964379460*; E-mail: wslpc1999@163.com*
Corresponding author: Pengcheng LV
DOI: https://doi.org/10.35633/inmateh-73-54

Keywords: *Loopback Detection; Scan Context; Point Cloud Alignment; SLAM*

ABSTRACT
Simultaneous localization and mapping (SLAM) is one of the key technologies for agricultural robots to build maps and localize in complex orchard environments and realize unmanned autonomous operations. Due to the complexity of the orchard environment, the single canopy feature and the diffuse reflection of light caused by the leaves, etc., the map construction process of the orchard environment leads to mismatch and increases the cumulative error of the map construction. Aiming at the above problems, this paper proposes a navigation map construction method for orchard environment based on the fusion of Scan Context and NDT-ICP. The method firstly searches the Ring key quickly to get the candidate frames, and scores the similarity between the candidate frames and the current frame, and effectively detects the loopbacks by two-stage searching algorithm to reduce the false matches in the map of orchard environment. Meanwhile, a point cloud alignment method based on the fusion of normal distribution transform coarse alignment and iterative nearest point exact alignment is used to reduce the cumulative error of the orchard environment map. The results show that the improved algorithm compensates the drift of the point cloud map with higher mapping accuracy, better real-time performance, lower resource utilization, higher overlap between the trajectory estimation and the real trajectory, smoother loops, and a 4% reduction in CPU occupancy. In the complex orchard environment, the root mean square error and standard deviation of the trajectories of this paper's algorithm are 0.57 m and 0.19 m, which are 68% and 83% higher than those of the loop detection algorithms in the Lightweight Ground Optimized Lidar Trajectory Measurement and Multivariate Terrain Mapping (LeGO-LOAM), respectively. Accurate map construction and low drift pose estimation can be performed. The research algorithm effectively reduces the influence of mis-matching and large cumulative error in the process of map construction in the orchard environment, meets the demand for high-precision environmental mapping in the orchard environment, and provides technical support for promoting unmanned operation in the orchard environment.

摘要
同步定位与绘图（SLAM）是农业机器人在复杂果园环境中构建地图并进行定位、实现无人自主作业的关键技术之一。由于果园环境的复杂性、树冠的单一性和树叶对光线的漫反射等特点，果园环境的地图构建过程中会出现不匹配现象，增加了地图构建的累积误差。针对上述问题，本文提出了一种基于 Scan Context 和 NDT-ICP 融合的果园环境导航地图构建方法。该方法首先快速搜索环键得到候选帧，并对候选帧与当前帧的相似度进行评分，通过两阶段搜索算法有效检测回环，减少果园环境地图中的虚假匹配。同时，采用基于正态分布变换粗配准和迭代最近点精确配准融合的点云配准方法，降低果园环境地图的累积误差。结果表明，改进后的算法可以弥补点云图的漂移，具有更高的映射精度、更好的实时性、更低的资源利用率、更高的轨迹估计与真实轨迹重合度、更平滑的循环以及降低 4% 的 CPU 占用率。在复杂果园环境中，本文算法的轨迹均方根误差和标准偏差分别为 0.57 米和 0.19 米，比轻量级地面优化激光雷达轨迹测量和多元地形测绘（LeGO-LOAM）中的环路检测算法分别高 68%和 83%。该算法有效降低了果园环境地图构建过程中的误匹配和大累积误差的影响，满足了果园环境高精度环境测绘的需求，为推进果园环境无人化作业提供了技术支撑。

INTRODUCTION
In recent years, with China's strong support for intelligent agriculture, China's first unmanned farm landed, which further promoted the rapid development of intelligent agricultural equipment technology.

The application of mobile robots in precision agriculture is becoming more and more widespread, autonomous mobile robots have a variety of functions such as environment perception, dynamic decision-making, path planning and autonomous navigation, and can perform specific tasks in unknown environments. Agricultural robotics technology has gained rapid development with the advancement of computers, LiDAR and other technologies. Instant localization and map construction is one of the key technologies of agricultural robot navigation technology, and it is also an important basis for automatic navigation. The continuous development of sensor technology, especially the rapid development of three-dimensional LiDAR in recent years, makes the instant localization and map construction technology having a great degree of improvement in map building accuracy and robustness. According to the type of sensor, SLAM technology can be divided into two categories: visual SLAM and laser SLAM. Among them, visual SLAM obtains environmental information with the help of visual sensors, low cost, simple structure, but large arithmetic capacity, easily affected by light, not applicable to farmland and other environments with obvious changes in light, laser SLAM technology is relatively mature, accurate ranging, less affected by light, and LIDAR compared to cameras, ultrasonic, infrared sensors, etc. has the advantages of strong anti-jamming ability, high accuracy, wide measurement range, etc., and is more suitable for orchard environments. At present, agricultural robots equipped with LiDAR to achieve local positioning and navigation method cannot provide global positioning information, Real Time Kinematic (RTK) global positioning method is mainly realized through the three-dimensional map building and repositioning. The main algorithms to achieve map building are Lidar Odometry and Mapping (LOAM) and Lightweight and Ground Optimized Lidar Odometry and Mapping (LeGO-LOAM). The LOAM algorithm, which lacks loopback detection and back-end map optimization, operates in the complex environment of the orchard with a serious lack of accuracy. Based on this, LeGO-LOAM algorithm adds loopback detection and graph optimization part. However, agricultural scenarios are a typical unstructured environment, which is characterized by inconspicuous features, uneven terrain, and dynamic changes of objects, which brings new challenges to the construction of maps in such scenarios, and the loopback detection and graph optimization part inevitably introduces problems such as map mismatch and large cumulative errors in orchard environments *(Wang et al., 2022)*.

Loopback detection, also known as closed-loop detection, is the ability of a robot to recognize that it has arrived at a scene so that the map closes the loop, usually using 3D descriptor matching, such as Fast Point Feature Histogram (FPFH), Fast Laser Region of Interest Transform (FLIRT) and Signature of Histograms of Orientation (SHOT) algorithms *(Davison et al., 2007)*. If the loopback detection is successful, it can significantly reduce the cumulative error and help the robot realize the obstacle avoidance navigation work more accurately and quickly. Therefore, loopback detection has become one of the most popular methods in recent years in the field of robotics research. Loop detection has become one of the hotspots in robotics research in recent years. With the development of deep learning, semantic information can be easily obtained from point cloud images, which can be used to assist SLAM loopback detection *(Pire et al., 2017)*. For example, SegMatch, a 3D point cloud segmentation matching based method, and SUMA++, a surf-based method, achieve amazing accuracy in highway scenes by dynamic removal *(Zhou et al., 2021)*.

In addition to loopback detection, point cloud alignment is also an important step in robot obstacle avoidance navigation. There are two commonly used 3D LiDAR point cloud alignment methods, Iterative Closest Point (ICP) and Normal Distribution Transform (NDT). Among them, the ICP algorithm is slower but more accurate, while the NDT algorithm is faster but less accurate. To address the problems of mis-matching of orchard environment maps and large cumulative errors introduced by the LeGO-LOAM algorithm, the Scan Context algorithm is used to optimize the LeGO-LOAM map detection module, taking into account the characteristics of Scan Context's high accuracy, low cost, high efficiency and robustness, and the NDT -ICP point cloud alignment algorithm to optimize the global map obtained from the loop to improve the accuracy, real-time and robustness of the map construction in the orchard environment, and the performance of the algorithm is evaluated with the KITTI dataset 00 sequential data and the field map construction test in the vineyard as an example to validate the performance of the proposed map construction method for the agricultural robots' localization and navigation, environment map construction and unmanned driving.

MATERIALS AND METHODS

LeGO-LOAM Algorithm

LeGO-LOAM is based on the improvements made to the LOAM algorithm in 2018. As illustrated in Fig. 1, the collected point cloud of the orchard environment is initially clustered and segmented, with the ground point cloud being isolated. Concurrently, a limited number of point cloud clusters are excluded *(Zhang et al.,*

2014). A two-step Levenberg-Marquardt (L-M) optimization method is employed to resolve the six-degree-of-freedom transformation between continuous frames. In the initial stage, the ground point cloud is employed to determine the planar transformation parameters *(Rösmann et al., 2015)*. Subsequently, the edge and surface points within the segmented point cloud are matched to generate the bitmap transformation matrix, which is subsequently processed and aligned. Finally, the motion estimation drift correction is performed, and the final pose estimate is generated *(Xue et al., 2023)*.

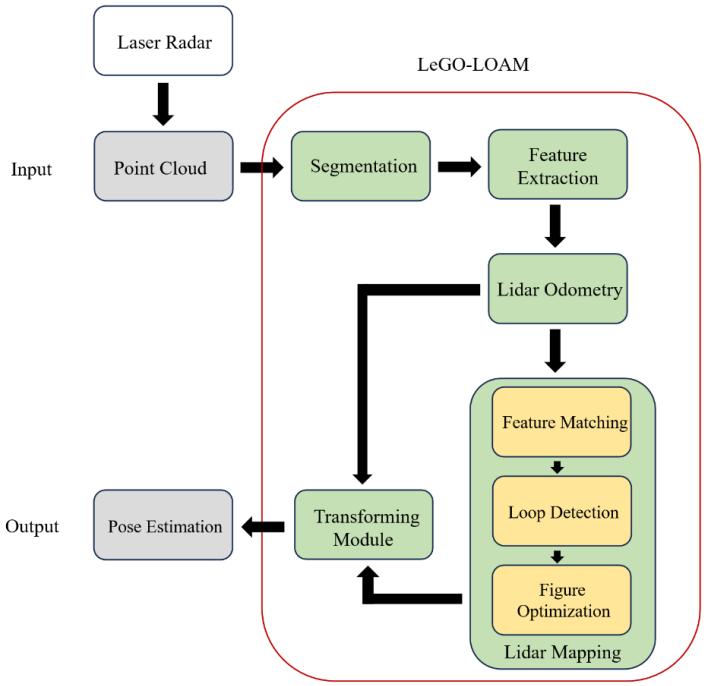
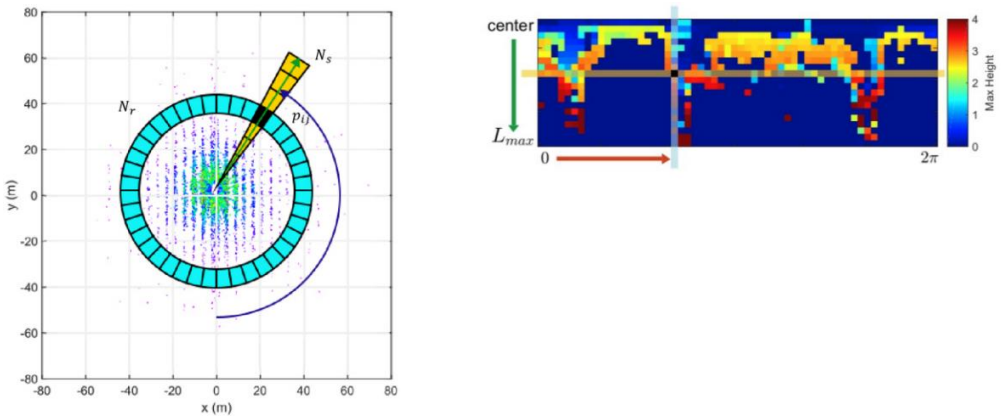


Fig. 1 – LeGO-LOAM algorithm flow

The LeGO-LOAM method cyclic algorithm is computationally intensive, inefficient in detection, and still produces significant cumulative errors in mapping large scenes at long distances *(Shan et al., 2018)*.

Scan Context Loopback Detection Method

Scan context converts a 3D point cloud to 2.5D by dimensionality reduction and uses a search algorithm to match point cloud data from current and historical frames to enable loopback detection *(Shan et al., 2020)*. Figure 2 shows the structure diagram of scan context global descriptor. For a frame of point cloud data scanned by LiDAR, the top view is obtained from the top of 3D point cloud (as shown in Figure 2a), and the polar coordinate system is established with the LiDAR position as its origin. Twenty rings are divided outward from origin, and each ring is divided into 60 equal parts, namely 1200 grids, taking the maximum height (Z value) of the points in each grid as the grid value. Then the top view is expanded radially into 20 rows and 60 columns of rectangular images (Figure 2b), the average values of each row and each column are calculated respectively, and the two vectors—ring key and sector are obtained as global descriptors *(Dijkstra et al., 1959)*.



(a) Top view of one frame of point cloud (b) Rectangular image expanded from the top view

Fig. 2 – Construction diagram of scan context global descriptor

The Scan Context loopback detection algorithm constructs a rectangular image using the scanned point cloud data, constructs a KD tree using the loopback key vectors, performs a nearest-neighbor search to find similar frames and their loopback key transitions that may be looped back to the current frame, calculates the similarity score, and filters out the similar frames that have higher scores; it then calculates the minimum offset and the similarity score on a sector-by-sector basis, and selects the frame that has the highest similarity score as a looped back frame, solves the attitude relationship between the current frame and the looped back frame, and realizes the loopback detection *(Seet et al., 2004)*.This method is highly effective in detecting trajectory drift and can significantly reduce the accumulated errors in map construction and localization, enabling orchard robots to perform tasks such as obstacle avoidance navigation with greater accuracy and speed *(Fox et al., 1997)*.

Improved Algorithm Principle

The ICP loopback detection method based on Euclidean distance is mainly adopted in the LeGO-LOAM algorithm, in this paper, the fusion algorithm of Scan Context and NDT-ICP is adopted instead of the ICP loopback detection method based on Euclidean distance in the LeGO-LOAM algorithm, and the positional constraints computed by the NDT-ICP are added into the GTSAM for the global positional optimization, at this time, we obtain The global map is more complete. The block diagram of the algorithm is shown in Fig. 3.

The system reads reads the point cloud data collected by LiDAR and projects the points as depth images. Then, the ground plane of the depth map is estimated, ground points are extracted, and each frame of the point cloud Pt is divided into different clusters labeled as segmentation points by the point cloud segmentation module. At the same time, the point cloud that cannot be clustered is re-clustered by obtaining three features: the label of the point, the row and column index in the depth map, and the distance value. After that, the feature extraction module is used to extract the edge feature point nF_e and the plane feature point nF_p based on the roughness *c*. The ranging module performs a two-step L-M optimization to obtain the attitude transformation matrix through the above two feature points to realize the spatial constraints between consecutive frames of the point cloud. The LiDAR mapping module matches the features in (F_e^t, F_p^t) with the surrounding point cloud Q^{-t-1} and further sends its own position to GTSAM for map optimization, at which point the radar estimates the pose update and optimizes the current map.

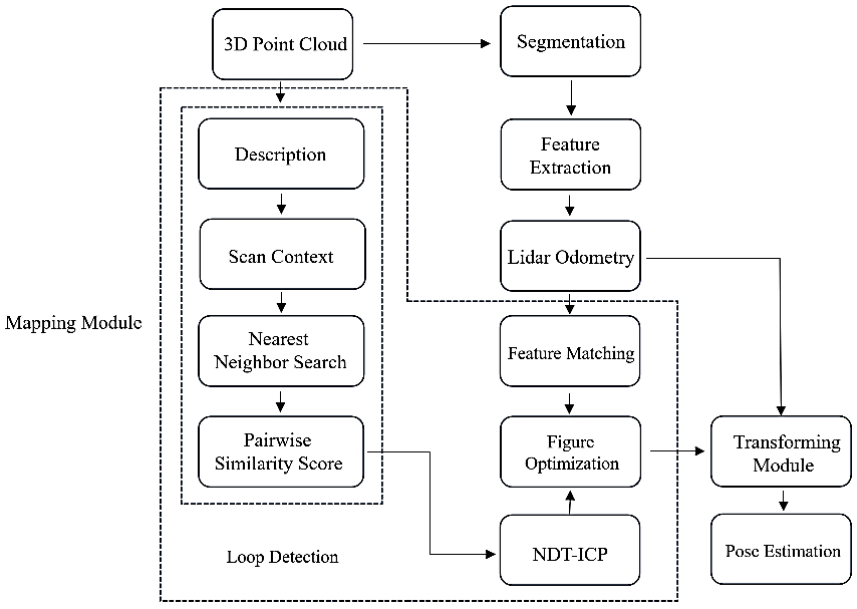


Fig. 3 – System architecture diagram of the improved algorithm in this paper

In order to obtain the relative position matrix between the matched point cloud frame and the loopback point cloud frame for back-end graph optimization, after confirming the loopback position, this paper chooses NDT-ICP to align the point cloud between the matched point cloud frame and the candidate point cloud frame of the orchard, and calculate the alignment score by the NDT-ICP method. If the alignment score is less than a given threshold, the repositioning is considered successful and the attitude constraints between the repositioned frame and the current frame are obtained. The constraints are added to GTSAM for map optimization and point cloud map update.

The transform fusion module fuses the position and location estimates from the LiDAR rangefinder module and the LiDAR mapping module and outputs the final position and location estimates.

Divide the orchard matching point cloud frame space into a number of identical cubes and satisfy that there are at least 5 points in the cube, the mathematical expressions for the mean μ and covariance matrix C of the points in the cube of each matching point cloud frame are:

$$C = \frac{1}{m}\sum_{k=1}^{m} I_k^q \tag{1}$$

$$\sum = \frac{1}{m-1}\sum_{k=1}^{m} \left(I_k^q - \mu\right)\left(I_k^q - \mu\right)^{\mathrm{T}} \tag{2}$$

where: m is the number of point clouds in the spatial cube of the matched point cloud frame; k is the point cloud serial number, k = 1, 2, ..., m; I_k^q is the point cloud in the spatial cube of the matched point cloud frame of the orchard.

Probability density of each point location in the frame cube of the orchard matching point cloud:

$$p\left(I^q\right) = \exp\left(-\frac{\left(I^q - \mu\right)^{\mathrm{T}} \sum_{k}^{-1} \left(I^q - \mu\right)}{2}\right) \tag{3}$$

Mapping the matched point cloud frames to the loopback detection frame coordinate system, each point mapping a transformed normal distribution:

$$p\left(I_k^{q'}\right) \sim \exp\left(-\frac{\left(I_k^{q'} - \mu_k\right)^{\mathrm{T}} \sum_{k}^{-1} \left(I_k^{q'} - \mu_k\right)}{2}\right) \tag{4}$$

where:

$I_k^{q'}$ is the set of point clouds after I_k^q mapping; μ_k is the mean value of the point cloud set $I_k^{q'}$.

Summing the probability densities of each point, the mathematical expression for the parameters of the coordinate transformation is evaluated as

$$s(p) = \sum_{k} \exp\left(-\frac{\left(I_k^{q'} - \mu_k\right)^{\mathrm{T}} \sum_{k}^{-1} \left(I_k^{q'} - \mu_k\right)}{2}\right) \tag{5}$$

Optimize $s(p)$ using the Hessian matrix method. Then, remap to the loopback detection frame coordinate system until the convergence condition is satisfied. The optimized loopback point set is $I_k^{c^{*'}}$ and the matched point set I^q. The error between the orchard origin point set $I_k^{c^{*'}}$ under the transformation matrix (R, t) and the orchard target point set I^q is denoted by $E(R,t)$. Then the problem of solving the optimal transformation matrix is transformed into the optimal solution (R,t) satisfying min$E(R,t)$. where $E(R,t)$ is called the objective function, which represents the degree of difference between the two point sets. This objective function is expressed as

$$E(R,t) = \sum_{k=1}^{n} \left\| I_k^q - I_k^{c^{*'}} \cdot R + t \right\|^2 \tag{6}$$

where:

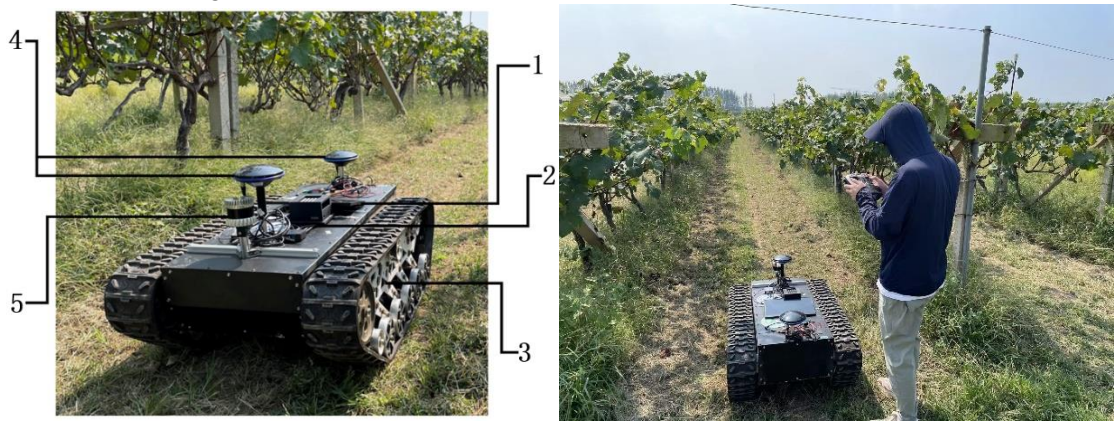
$I_k^{c^{*'}}$ is the optimized loopback point set of the candidate point cloud frame I^c; n is the number of points in the point set I_k^q; R is the rotation parameter; t is the translation parameter.

In order to minimize the objective function, the optimal transformation matrix, i.e., R and t is solved. first, the corresponding closest point in the I^q point set is calculated for each point in the orchard's target point set, $I^{c^*\prime}$. The rotation parameter R and the translation parameter t are obtained using SVD decomposition so that the transformation matrix $E(R,t)$ is minimized. Using the rotation parameter R and translation parameter t obtained in the previous step for the point set $I^{c^*\prime}$, new transformed point set $I^{c^*\prime\prime}$ is obtained. If the transformed point set $I^{c^*\prime\prime}$ and I^q point set satisfy the requirement of the objective function, i.e., the average distance between the two point sets is less than a given threshold, then the iterative computation is stopped; otherwise, recalculate the new point set $I^{c^*\prime\prime}$ as the new point set $I^{c^*\prime}$ and continue iterating until the convergence condition is satisfied, and then obtain the optimal solution (R, t) with the relative positional matrix.

Test Equipment and Environment

At present, orchard robots are available in three main categories: legged, wheeled, and tracked. Each type has its own set of advantages and disadvantages in different environments. The legged inspection robot exhibits strong terrain adaptability, yet its intricate structure and sophisticated control system present significant technical challenges. Wheeled inspection robots are distinguished by their high speed, high efficiency, and low motion noise, yet they are constrained by complex terrain. The tracked inspection robot is equipped with a robust traction force, high applicability, and excellent stability in challenging terrain, such as outdoor, sandy, and muddy areas.In this paper, a tracked kinematic chassis was used to design an orchard robot for data acquisition and field trials in a complex outdoor orchard environment, as shown in Fig. 4(a). The hardware system mainly consists of a 3D LiDAR (Ouster-64), Global navigation satellite systems real-time kinematic (GNSS-RTK), and Jetson AGX Xavier with Ubuntu 20.04 operating system. Ouster-64 is a 64-line 3D LiDAR developed by Ouster Inc. in the United States. The GNSS-RTK used in this paper was developed by New Coordinate Intelligent Equipment Co. Ltd. in Zibo, Shandong Province, China, and the Jetson AGX Xavier was manufactured by NVIDIA Corporation in Santa Clara, California, USA.

The robot fixes the Ouster 64-line LiDAR at the centre of the crawler chassis through an aluminium radar bracket, and the centre of the LiDAR is 1 m away from the ground, which ensures that the LiDAR is not obstructed and guarantees the accuracy of the SLAM construction. In addition, two GNSS antennas are screwed to the strong magnetic suction cups and fixed in the forward direction and backward direction of the crawler chassis respectively, with a distance of 1 meter, to ensure that there is no obstruction above the antennas, so as to ensure that they can receive good positioning signals, and at the same time, ensure that the corresponding centres of the two GNSS antennas form the same line with the central axis of the crawler chassis, so as to ensure that the GNSS-RTK can collect the position of the orchard robots in real time. GNSS-RTK can collect the position information of the orchard robot in real time, and the accuracy reaches centimetre level. Finally, the Jetson AGX Xavier is installed and connected to the robot's underlying control driver board via a serial cable, to the GNSS-RTK positioning receiver via a USB cable, and to the LiDAR adapter box via an Ethernet cable. The tracked chassis of the orchard robot is powered by a bucking module that reduces the 48V voltage of the LiDAR to 24V and the 48V voltage of the Jetson AGX Xavier to 5V. Figure 5 shows the hardware connection diagram of the orchard robot.



(a) Orchard robots (b) Orchard environment

Fig. 4 – Field orchard environmental experiment

1. External Monitor; 2. Jetson AGX Xavier; 3. Crawler-Type Mobile Chassis; 4. Dual-Antenna GNSS-RTK; 5. 64-Line 3D LiDAR

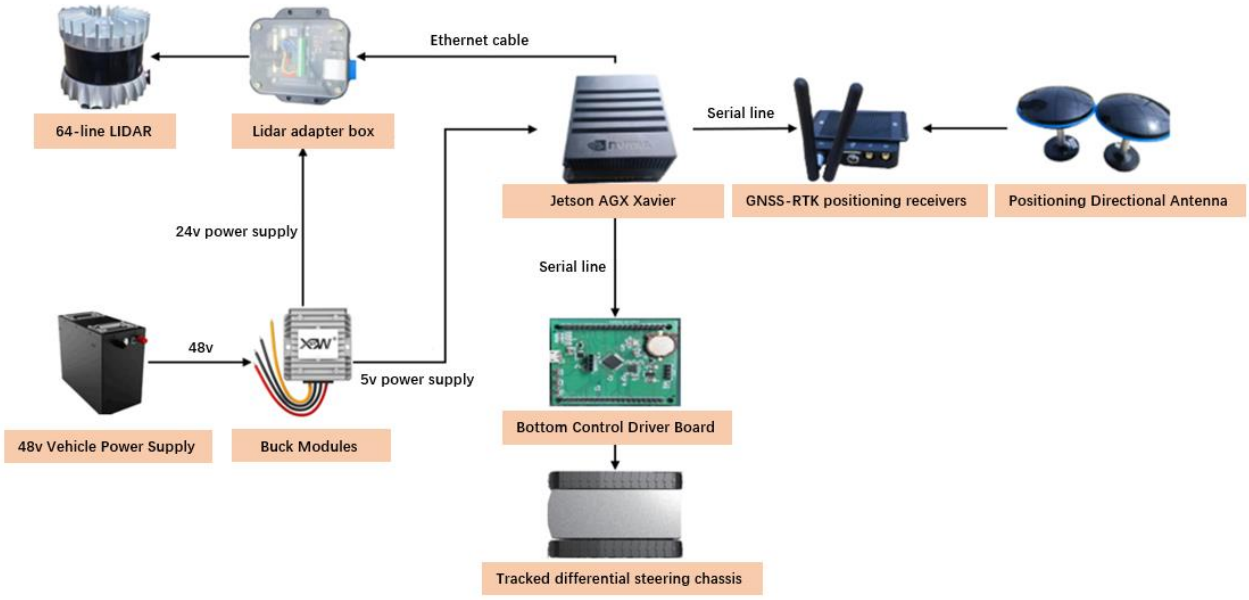


Fig. 5 – Hardware Connection Diagram of Orchard Work Robot

The data used for the experiments in this paper are the KITTI dataset in the field of automated driving and the dataset collected in the field of standardized plantation vineyard environment. Among them, the orchard environment dataset has a total length of 1707.635 m and a duration of 1166 s. The KITTI dataset 00 sequence, which has a total length of 3724.187 m, lasts for 570 s.

As shown in Fig. 4(b), in order to study the SLAM approach for orchard robots in orchard environments, we have used the KITTI dataset 00 sequence in the Grapevine Base in Huantai County, Zibo City, Shandong Province (N37.07420° and E117.91777°) in Zibo City, Shandong Province, where a 132-m × 144-m rectangular plot was selected for data collection. The cultivation pattern of the grape base is the ridge planting method, which means that grapes are planted in a single row on each ridge with a row spacing of 3 m. The cultivation pattern of the grape base is the ridge planting method. To collect the data, we need to scan the orchard environment with a LIDAR-equipped robot. The manually controlled orchard robot travels through the orchard at a speed of 0.5-1 m/s, and the 3D LIDAR records the point cloud data at a frequency of 10 Hz.

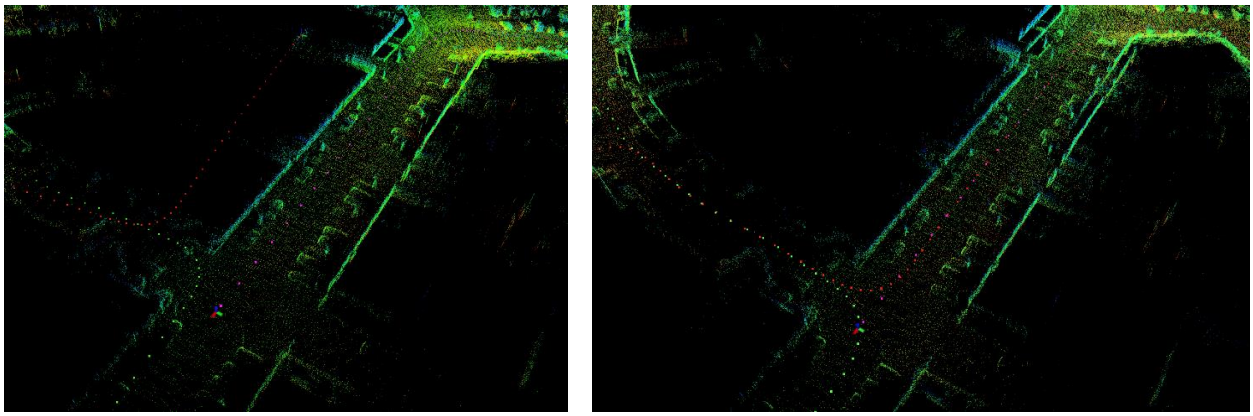
RESULTS

Experiment details

The experimental equipment used for the test was a Shenzhou laptop, its CPU model is I7-12650H, memory is 16G, the graphics chip is NVIDIA Geforce GTX4060, the system environment is Ubuntu20.04, ROS noetic, PCL 1.8, GTSAM 4.0.0, Python 3.6. The test dataset uses the KITTI dataset 00 sequence as well as our field-collected standardized vineyard environment dataset.

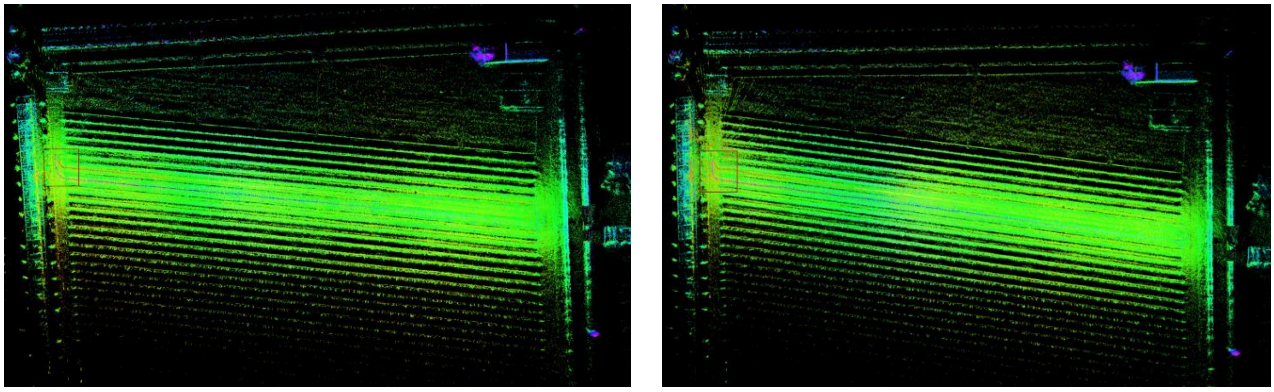
In the loopback detection experiment on the KITTI dataset, the vineyard point cloud map generated by the orchard robot based on the LeGO-LOAM algorithm is depicted in Fig. 6(a). Upon the orchard robot's return to the starting point, the end point of the radar odometer fails to form a closed loop with the starting point, resulting in a cumulative error in position estimation that causes the robot trajectory to deviate and the loopback detection to fail in this instance. The point cloud generated by this paper based on the scanning context fusion NDT-ICP loopback detection algorithm is shown in Fig. 6(b).

The robot is able to recognize the scene of the previously reached location after building the map and successfully performs loopback detection. Furthermore, we conducted a comparative analysis between LeGO-LOAM and the proposed algorithm on a standardized vineyard environment dataset, as illustrated in Fig. 7(a) and Fig. 7(b). This analysis revealed that the endpoints of the vineyard point clouds generated by LeGO-LOAM do not form a closed loop with the starting point, resulting in a failed loopback. In contrast, the vineyard point clouds generated by this paper's algorithm are successfully looped back, as illustrated by the red box.



(a) Loopback detection test failed (b) Loopback detection test successful

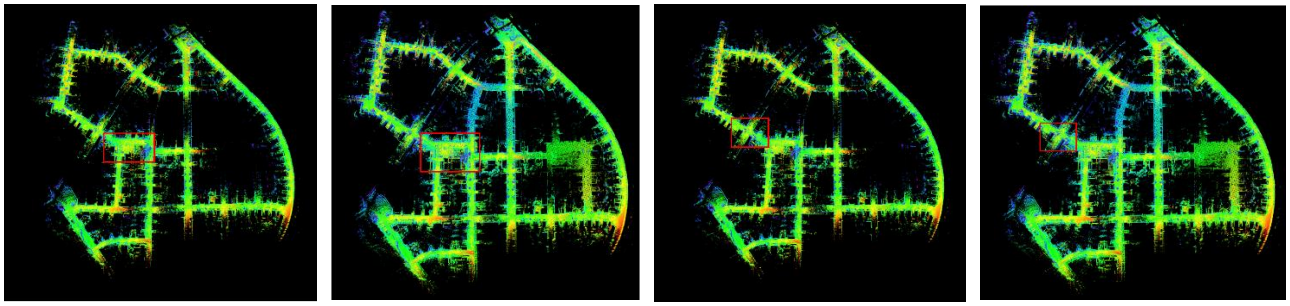
Fig. 6 – KITTI dataset loopback detection test



(a) Loopback detection test failed (b) Loopback detection test successful

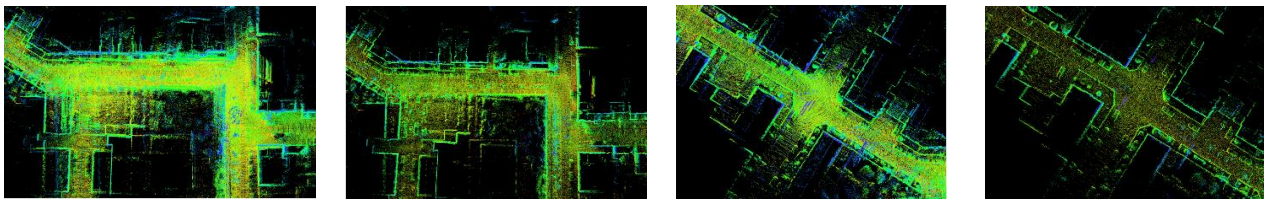
Fig. 7 – Vineyard dataset loopback detection test

Figure 8 shows the global point cloud map of KITTI dataset 00 sequence constructed by LeGO-LOAM and the improved algorithm in this paper. Figure 9 shows the local zoomed-in point cloud map of KITTI dataset 00 sequence constructed by LeGO-LOAM and the improved algorithm in this paper. It can be seen that when the LeGO-LOAM algorithm is used for mapping, the drift is serious, the effect is poor and the mapping is incomplete. When the algorithm in this paper is used for mapping, the mapping results are more complete and clearer than the LeGO-LOAM algorithm. Figures 8(b), (d) and 9(b), (d) show that it just makes up for the phenomenon of point cloud map drift.



(a) (b) (c) (d)

Fig. 8 – Global Map



(a) (b) (c) (d)

Fig. 9 – Local zoom map

Test Analysis

In order to evaluate the effectiveness of the mapping method in this paper in the vineyard, the continuously acquired latitude and longitude information from GNSS is used as the trajectory truth in this paper to compare with the trajectory generated by the algorithm. The Root Mean Square Error (RMSE) and the Standard Deviation (STD) are often used as evaluation metrics for algorithms, so we use this as a reference.

The mathematical expression for root mean square error (RMSE) is:

$$RMSE = \sqrt{\frac{\sum_{a=1}^{d} X_a - X'^{\,2}}{d}} \tag{7}$$

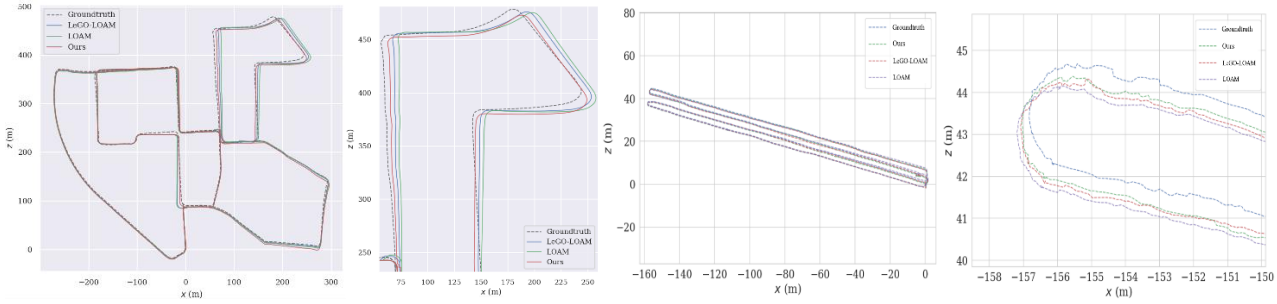
where: X' is the true value; X_a is the measured value; d is the number of positions; a is the positional ordinate, $a = 1, 2, ..., d$.

The mathematical expression for standard deviation (STD) is:

$$std = \sqrt{\frac{\sum_{a=1}^{d} X_a - \bar{X}^{\,2}}{d-1}} \tag{8}$$

where $\bar{X}$ is the mean value.

Three algorithms for motion trajectory estimation for the orchard test: LOAM, LeGO-LOAM and Ours. as shown in Fig. 10, the algorithm in this paper gives the best results, which are basically consistent with the real trajectory (ground truth).



(a) (b) (c) (d)

Fig. 10 – Trial traces versus real-value traces

(a). Global trajectory of the KITTI dataset; (b). Localized trajectories for the KITTI dataset; (c). Global trajectory of the Vineyard dataset; (d). Localized trajectories for the Vineyard dataset.

Table 1 gives the estimated path length, time consumption, CPU occupancy, RMSE, and STD for LOAM, LeGO-LOAM, and the algorithm in this paper (Ours), reflecting the real-time robustness and stability of the algorithms.

Table 1

Test error comparison between different algorithms

Scene	Algorithm	Length (m)	Time (s)	CPU Rate / %	RMSE (m)	STD (m)
Kitti00	LOAM	3736.586	556.362	70	7.92	4.88
	LeGO-LOAM	3731.182	572.168	65	5.46	2.41
	Ours	3724.187	540.872	62	2.11	0.72
Vineyard	LOAM	1719.361	1181.271	77	2.88	1.31
	LeGO-LOAM	1712.846	1173.984	74	1.76	1.12
	Ours	1707.635	1166.323	71	0.57	0.19

In comparison to the LeGO-LOAM algorithm, the algorithm presented in this paper demonstrates a 4% reduction in CPU occupancy on average, an improvement in real-time performance, and a reduction in elapsed time of approximately 5%. The root mean square error and standard deviation of the algorithm presented in this paper in the KITTI 00 dataset are 2.11 meters and 0.72 meters, respectively. These values are reduced by 61% and 70%, respectively, in comparison to the LeGO-LOAM algorithm.

In the Vineyard dataset, the root mean square error and standard deviation of this algorithm are 0.57 m and 0.19 m, respectively, representing a reduction of 68% and 83%, respectively, in comparison to the LeGO-LOAM algorithm.

The experiments demonstrate that in a large-scale, standardized planting vineyard scenario, the orchard robot is capable of successfully looping back to detect multiple times, thereby greatly reducing the cumulative error and enhancing the efficiency of our algorithm. Furthermore, an examination of the KITTI dataset reveals that the algorithm has been optimized for real orchard scenarios. This is due to the fact that the algorithm is not constrained by the dimensions of the scene in question. The experimental results serve to illustrate the viability of our augmented algorithm in the context of orchard scenes.

CONCLUSIONS

This study addresses the context of complex orchard environments where simultaneous localization and mapping (SLAM) in large-scale scenarios often leads to loopback failures. To address these challenges, this paper presents a method for constructing navigation maps for orchard environments based on scanning context and NDT-ICP fusion. The method initially identifies potential frames by rapidly searching for ring keys, and then assesses the degree of similarity between candidate frames and the current frame. This approach enables the effective detection of loopbacks through the implementation of a two-stage search algorithm, thereby reducing the occurrence of false matches in the orchard environment map.

Furthermore, a point cloud alignment method based on the fusion of normal distribution transform coarse alignment and iterative nearest point exact alignment is employed to reduce the cumulative errors in the orchard environment map, and field experiments were conducted in the vineyard. The following conclusions were drawn from the study:

1. An improved SLAM algorithm incorporating Scan Context and NDT-ICP proposed in this paper has been tested on datasets and experimental test results show that the improved algorithm has higher mapping accuracy and lower time consumption and resource usage.

2. The mapping and position estimation performance of LOAM, LeGO-LOAM, and the enhanced algorithm of this paper were evaluated using the KITTI dataset 00 sequence with the vineyard dataset. The results demonstrated that the algorithm proposed in this paper exhibited enhanced performance in terms of reducing the drift of point cloud maps, improving the accuracy of the motion trajectory estimation, and providing a more seamless loop. Additionally, the estimated trajectory length was found to be more closely aligned with the real trajectory length. In comparison to the LeGO-LOAM algorithm, the CPU occupancy is reduced by 4% on average, the real-time performance is enhanced, and the time consumption is reduced by approximately 5%. In the KITTI 00 dataset, the root mean square error and standard deviation of the enhanced algorithm are 2.11 meters and 0.72 meters, respectively. In comparison to the LeGO-LOAM algorithm, these two values are reduced by 61% and 70%, respectively. In the vineyard dataset, the root mean square error and standard deviation of our enhanced algorithm are 0.57 m and 0.19 m, respectively, representing a reduction of 68% and 83%.

3. The prevailing approach to SLAM is largely reliant on mathematical models and the design of intricate algorithmic processes to process sensor data and generate localization maps. In the future, there is potential to explore the integration of deep learning and SLAM. This could enable the full exploitation of the strengths of deep learning in vision and semantics, thereby enhancing the localization and mapping accuracy of the SLAM system. Furthermore, the application of a deep learning model to extract and analyse the features of the sensor data could lead to improvements in the performance and applicability of mapping and localization.

ACKNOWLEDGEMENT

The project supported by the R&D of low-speed unmanned autonomous navigation controller Project under Grant (No.2022TSGC1175).

REFERENCES

[1] Chang L., Shan L., Jiang C., Dai Y., et al. (2021) Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment. *Auton. Robot*. 45, 51–76.

[2] Davison A., Reid I., Molton N., Stasse O., et al. (2007). MonoSLAM: Real-time single camera SLAM. *IEEE Trans. Pattern Anal. Mach.Intell.* 29, 1052–1067.

[3] Dijkstra, E. W., et al. (1959) A note on two problems in connexion with graphs. *Numerische Mathematik*. 1, 269–271.

[4] Fox, D., Burgard, W., & Thrun, S., et al. (1997) The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*. 4(1), 23-33.

[5] Pire T., Fischer T., Castro G., De Cristóforis P., Civera J., Berlles J., et al. (2017) S-PTAM: Stereo parallel tracking and mapping. *Robot.Auton*. Syst. 93, 27–42.

[6] Rösmann C., Hoffmann F., Bertram T., et al. (2015) Timed-elastic-bands for time-optimal point-to-point nonlinear model predictive control. In *Proceedings of the 2015 European Control Conference (ECC)*, Linz, Austria, 15–17 July. 3352–3357.

[7] Shan T., Englot B., et al. (2018) LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain*. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 4758-4765.

[8] Shan T., Englot B., Meyers D., et al. (2020) LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping. *IEEE/RSJ international conference on intelligent robots and systems (IROS)*. 5135-5142.

[9] Seet B., Liu G., Lee B., Foh C., Wong K., Lee K., et al. (2004) A-STAR: A mobile ad hoc routing strategy for metropolis vehicular communications. In *Proceedings of the Networking 2004: Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Third International IFIP-TC6 Networking Conference*, Athens, Greece, 9–14 May; Springer: Berlin/Heidelberg, Germany. 989–999.

[10] Wang N., Han Y., Wang Y., et al. (2022) Research progress on full-coverage operational planning for agricultural robots (农业机器人全覆盖作业规划研究进展). *Transactions of the Chinese Society for Agricultural Machinery*, Vol. 53, Issue S1: 1-19.

[11] Xue G, Li R, Zhang Z., et al. (2023) Research Status and Development Trend of SLAM Algorithm Based on 3D LiDAR (基于 3D 激光雷达的 SLAM 算法研究现状与发展趋势). *Information and Control*, vol. 52, Issue (01): 18-36.

[12] Zhou Z., Cao J., Di S., et al. (2021) Overview of 3D LiDAR SLAM algorithms (3D 激光雷达 SLAM 算法综述). *Chinese Journal of Scientific Instrument*, vol. 42, Issue (09): 13-27.

[13] Zhang J., Singh S., et al. (2014) LOAM: Lidar odometry and mapping in real-time. Robotics: *Science and Systems Conference*. 2(9): 1-9.