RESEARCH ON SLAM AND PATH PLANNING METHOD FOR INSPECTION ROBOT IN ORCHARD ENVIRONMENT
/
果园环境中巡检机器人的 SLAM 和路径规划方法研究

Pengcheng LV, Minhui ZHANG, Lili YI*
Shandong University of Technology, Collage of Agricultural Engineering and Food Science, ZiBo, China
Corresponding author: Lili YI; Tel: +86 18553308656*; E-mail: yili0001@sdut.edu.cn*
DOI: https://doi.org/10.35633/inmateh-73-18

Keywords: *Orchard robots; Autonomous navigation; SLAM; Path planning*

ABSTRACT

Orchard robots play a crucial role in agricultural production. Autonomous navigation serves as the foundation for orchard robots and eco-unmanned farms. Accurate sensing and localization are prerequisites for achieving autonomous navigation. However, current vision-based navigation solutions are sensitive to environmental factors, such as light, weather, and background, which can affect positioning accuracy. Therefore, they are unsuitable for outdoor navigation applications. LIDAR provides accurate distance measurements and is suitable for a wide range of environments. Its immunity to interference is not affected by light, colour, weather, or other factors, making it suitable for low objects and complex orchard scenes. Therefore, LiDAR navigation is more suitable for orchard environments. In complex orchard environments, tree branches and foliage can cause Global Positioning System (GNSS) accuracy to degrade, resulting in signal loss. Therefore, the major challenge that needs to be addressed is generating navigation paths and locating the position of orchard robots. In this paper, an improved method for Simultaneous Localization and Mapping (SLAM) and A-star algorithm is proposed. The SLAM and path planning method designed in this study effectively solves the problems of insufficient smoothness and large curvature fluctuation of the path planned in the complex orchard environment, and improves the detection efficiency of the robot. The experimental results indicate that the method can consistently and accurately fulfil the robot's detection needs in intricate orchard environments.

摘要

果园机器人在农业生产中发挥着至关重要的作用。自主导航是果园机器人和生态无人农场的基础。准确的感知和定位是实现自主导航的先决条件。然而，目前基于视觉的导航解决方案对光线、天气和背景等环境因素非常敏感，会影响定位精度。因此，它们不适合户外导航应用。激光雷达可提供精确的距离测量，适用于各种环境。它的抗干扰能力不受光线、颜色、天气或其他因素的影响，因此适用于低矮物体和复杂的果园场景。因此，激光雷达导航更适合果园环境。在复杂的果园环境中，树枝和树叶会导致全球定位系统（GNSS）精度降低，造成信号丢失。因此，需要解决的主要挑战是生成导航路径和定位果园机器人的位置。在本文中，我们提出了一种改进的同步定位与地图构建（SLAM）方法和 A-star 算法。本研究设计的 SLAM 和路径规划方法有效解决了复杂果园环境下规划路径的平滑度不够和曲率波动较大的问题，提高了机器人的巡检效率。实验结果表明，该方法能稳定、准确地满足机器人在复杂果园环境中的巡检需求。

INTRODUCTION

As China vigorously develops smart agriculture, the use of agricultural robots can effectively solve the challenges of rural labour shortage and labour intensity. China has issued a series of policy documents to support smart agriculture, such as the "14th Five-Year Plan" Robotics Industry Development Plan and the "14th Five-Year Plan" National Agricultural and Rural Informatization Development Plan, in which it is clearly stated that it is necessary to develop precision agriculture and support and encourage the Agricultural robot research and development and application, in order to accelerate the process of agricultural modernization. With the increasing diversity and complexity of the application scenarios of orchard robots, the autonomous navigation performance of the robot has put forward higher requirements. Robot navigation technology mainly includes **S**imultaneous **l**ocalization **a**nd **m**apping technology (SLAM) and path planning technology. SLAM is the process by which a moving robot determines its own position and creates a map through sensors carried in the surrounding environment.

Path planning technology creates the optimal navigation path for the robot to reach the target location according to different task objectives and requirements, according to the robot's operating width, minimum turning radius, operating strips and other parameters, and with the goal of minimizing operating energy consumption, the robot automatically plans a full-coverage operating path without repetitive omissions within the operating area, which is the basic requirement for autonomous robot operation *(Wang et al., 2022)*.

The orchard inspection robot belongs to the modern intelligent mobile robot. Its main function is to familiarize with the orchard environment, find a suitable path to reach the target point, and complete the whole task. Robot autonomous navigation is the basis for realizing robot intelligence, and it is also the key to realize robot autonomous walking to reach the destination and complete specific tasks. In the process of autonomous navigation, the robot must ensure its own safety and not cause damage to the surrounding environment.

Simultaneous localization and mapping, as one of the key technologies in the field of mobile robotics, has made great progress through the development of recent years *(Davison et al., 2007)*. The SLAM system is divided into vision SLAM and LIDAR SLAM according to the different sensors *(Pire et al., 2017)*. Due to the camera's sensitivity to light, vision SLAM technology is difficult to land in the actual production of orchards, and the application scene is limited *(Zhou et al., 2021)*. On the other hand, laser SLAM uses LiDAR (laser radar) as the main sensor, which is characterized by high robustness and high accuracy, and is more friendly to long lifecycle SLAM systems and suitable for operation in complex outdoor scenes such as orchards *(Xue et al., 2023)*. In recent years, many laser SLAM algorithms have been proposed. Zhang et al. proposed the classic and representative LOAM (LiDAR Odometry and Mapping) in 2014, which opens up the precedent of feature extraction for 3D point clouds and is useful for subsequent algorithms *(Zhang et al., 2014)*. The algorithm extracts edge and plane points in the point cloud according to the curvature size, and the algorithm extracts edge points and plane points in the point cloud according to the curvature size, and participates in frame-to-frame matching (Scan-to-Scan) and frame-to-map matching (Scan-to-Map), which improves the efficiency of point cloud alignment, but due to the lack of loopback detection and back-end optimisation, the cumulative error is larger in large-scale scenes. In 2018, LeGo-LOAM (Lightweight and Ground-Optimized LiDAR Odometry and Mapping) algorithm was proposed by Shan et al *(Shan et al., 2018)*. This algorithm optimises the front-end odometry, introduces the concept of keyframes, and adds loopback detection to LOAM, which greatly reduces the computation amount. LIO-SAM is the sequel of LeGO-SLAM authors, which implements tightly coupled laser inertial odometry on the basis of graph optimization framework, and adopts factor graph optimization to realize the fusion of introducing GPS factor and loopback detection factor, and this algorithm is compared with LOAM and LIOM algorithms horizontally, and the end-to-end error under the park dataset is 0.04 m, which is better than the 121.74 m of LOAM algorithm and the 34.6 m of LIOM, and it has better global consistency, although this algorithm is more consistent than LOAM algorithm, and it has more accuracy *(Shan et al., 2020)*.

The purpose of path planning is to plan an optimal collision-free path from the start point to the end point in a map environment. Dijkstra et al. proposed the shortest path planning algorithm which uses breadth-first search to search for paths *(Dijkstra et al., 1959)*. Seet et al. proposed the A-star algorithm, which reduces the search nodes and improves the efficiency of path search by using a heuristic evaluation function *(Seet et al., 2004)*. Fox et al. proposed the DWA, which dynamically samples velocities in the robot's sampling space based on the robot's kinematic model and the current motion parameters and selects the optimal trajectory *(Fox et al., 1997)*. To address the problem of insufficient DWA evaluation functions, Chang et al. proposed an improved DWA algorithm based on Q-learning, which modifies and extends the evaluation functions by adding two more evaluation functions in order to improve the navigation performance and to achieve higher navigation efficiency and success rate in complex and unknown environments *(Chang et al., 2021)*. Rösmann et al. proposed a time-elastic segmentation algorithm based on multi-objective optimization, which ensures that the robot outputs smooth trajectories while satisfying kinematic constraints *(Rösmann et al., 2015)*.

In this paper, an inspection robot navigation system designed for complex scenes in orchards, as shown in Figure 1, is introduced. A SLAM system based on multiline LiDAR is designed to meet the robot's inspection needs in outdoor scenes such as orchards. Firstly, the LiDAR scanned images of the vineyard environment are collected, and the original point cloud of the vineyard is subjected to data processing, and then features such as edges and corners are extracted from the scanned images, and the point cloud is matched using the NDT-ICP algorithm. Finally, these features are used to construct a 3D point cloud map of the environment. The 3D point cloud map is converted to a 2D raster map, and the improved A-star algorithm is used for global path planning to improve the navigation efficiency and help the robot quickly plan the optimal path. Finally, the robot is realized to complete the high-precision navigation inspection task in the complex scene of the orchard.

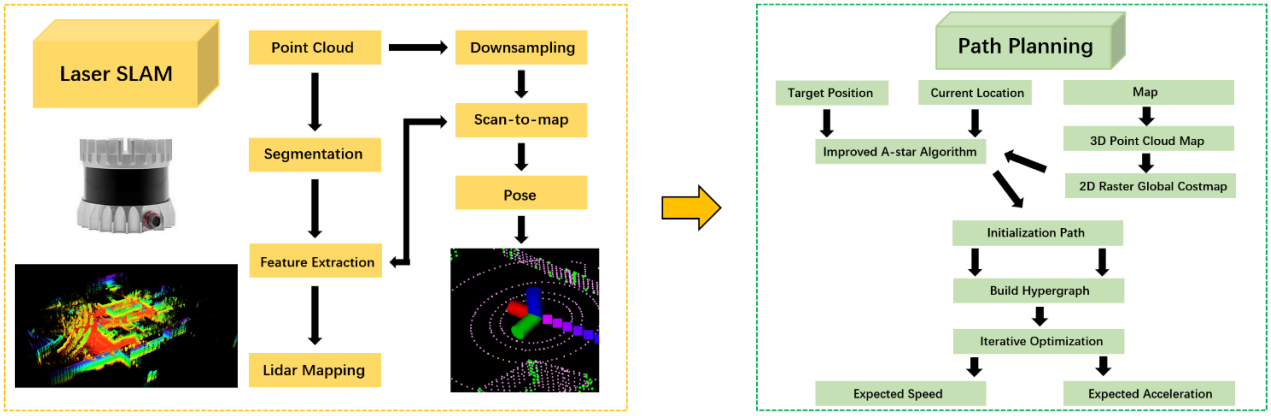


Fig. 1 – System framework

MATERIALS AND METHODS

Radar point cloud information acquisition

In order to study the SLAM and path planning methods for inspection robots in orchard environments, a 132-meter × 144-meter rectangular plot in a vineyard in Huantai County, Zibo City, Shandong Province (N37.07420°, E117.91777°) was selected for data acquisition. Figure 2 depicts the actual vineyard. The grapes in the orchard are planted using the ridge planting method, which involves the planting of grapes in single rows on each ridge, with a distance of three meters between the rows.

Fig. 2 – The Grapevine

The map query function of Bigemap GIS Office software was used to locate the test site, as shown in Figure 3.

Fig. 3 – The test site

An orchard robot was designed for the purpose of data collection and field experiments. This robot fixes the Ouster 64-line lidar to the centre of the crawler chassis by means of a radar bracket with an aluminium profile, and the centre of the LIDAR is located at a distance of 1 m from the ground, which ensures that the LIDAR is not obstructed and ensures that the accuracy of the SLAM construction is guaranteed. In addition, the two GNSS antennas are screwed to the strong magnetic suction cups and fixed in the forward direction and backward direction of the crawler chassis at a distance of 1 m to ensure that there is no obstruction above the antennas to ensure that good positioning signals are received.

Concurrently, it is imperative to guarantee that the corresponding centers of the two GNSS antennas align with the central axis of the crawler chassis. This configuration enables the GNSS-RTK to obtain the position data of the orchard robot in real time, and the accuracy reaches centimetre level. Finally, the AI processor is installed, which is connected to the robot's underlying control driver board via serial cable, to the GNSS-RTK positioning receiver via USB cable, and to the LiDAR adapter box via Ethernet cable. The tracked chassis of the orchard robot is powered by a buck module that steps down the 48 V to 24 V for the LIDAR and 48 V to 5 V for the AI processor.

Fig. 4 – Tracked mobile platform for orchard robots

1. External Monitor; 2. Artificial Intelligence (AI) Processors;
3. Crawler-Type Mobile Chassis; 4. Dual-Antenna GNSS-RTK; 5. 64-Line 3D LiDAR.

Figure 5 shows the hardware connection diagram of the orchard robot.

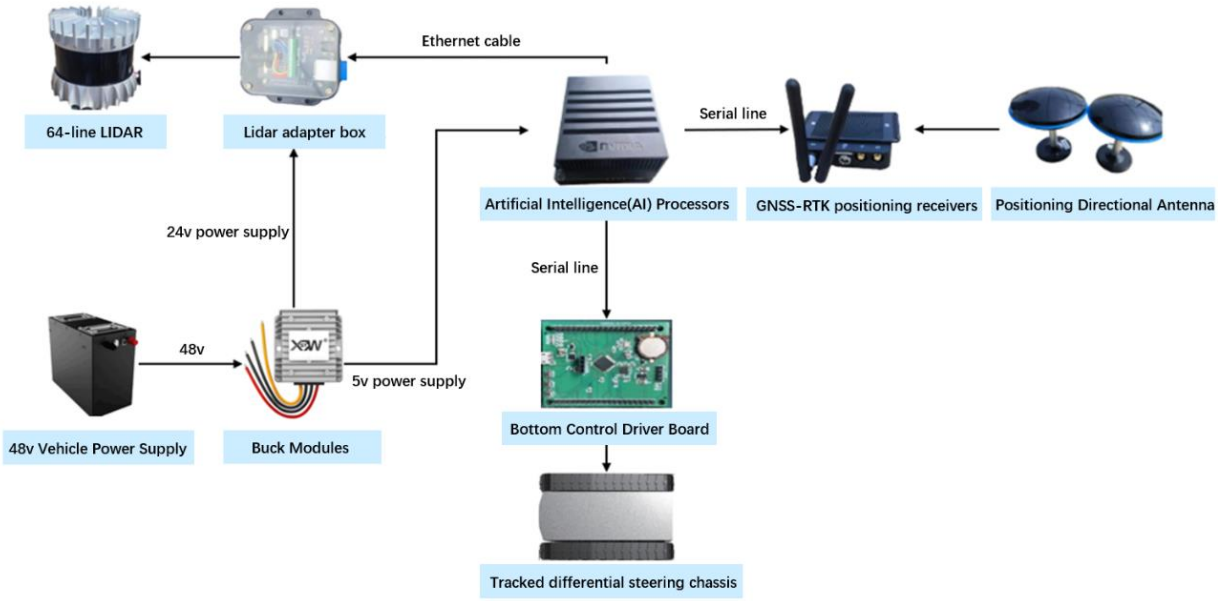


Fig. 5 – Hardware Connection Diagram of Orchard Work Robot

Multi-Line LiDAR-Based SLAM Algorithm Design and verification

In outdoor orchard environments, the position of the sun changes all the time from sunrise to sunset, and the intensity of the sunlight varies over time, which can cause vision sensors to misidentify objects or locate them inaccurately. However, laser sensors do not require an external light source because they emit their own laser beams, which are not affected by lighting problems. Therefore, a lightweight, ground-optimized laser ranging and mapping method, the structure of which is schematically shown in Fig. 6, was adopted.

The LeGo-LOAM algorithm consists of four main parts, which are point cloud segmentation, feature extraction, radar odometry, and map construction. The loopback detection used in the LeGo-LOAM algorithm is relatively simple, and the loopback matching process uses the Iterative Closest Point (ICP) and Euclidean distance combination, which is established for low drift cases, and when running in scenes with large distance ranges, it accumulates large errors and produces large scale drifts. To address this problem, new spatial constraints are added to the loopback detection process, a constraint relationship is established between the current frame point cloud data and the first three frames of point cloud data, and the bit position of the current frame is recursively solved from the first three frames of point cloud data to reduce the accumulation of error, and finally the current point cloud is mapped to the global map based on graph optimization to complete the establishment of a high-precision map.

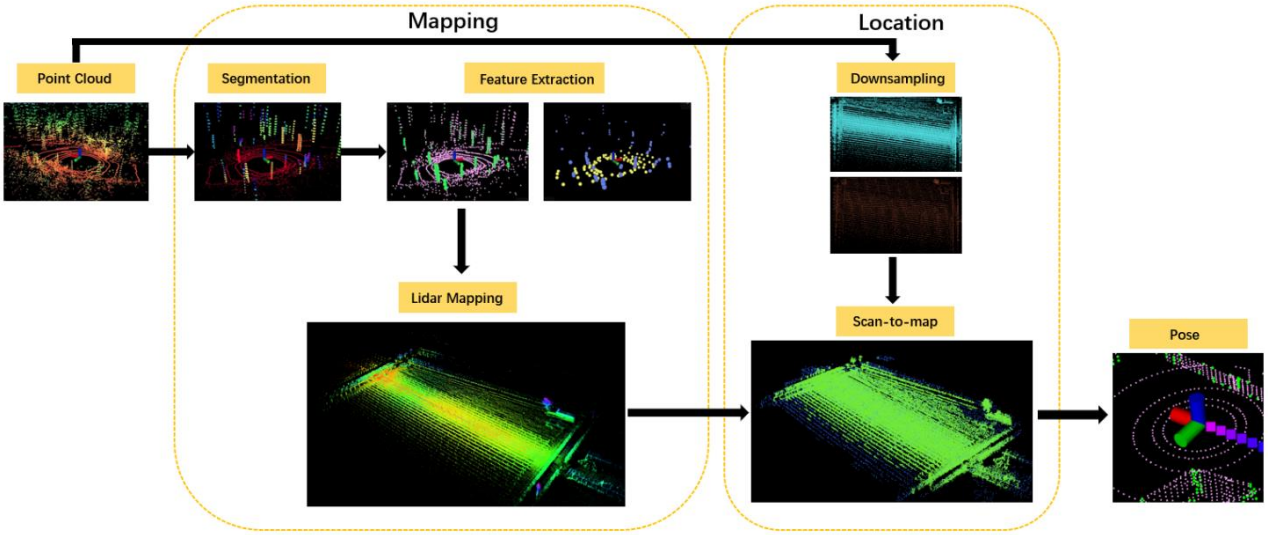


Fig. 6 – Multi-line LiDAR-based SLAM system structure diagram

The algorithm improvement process is as follows:

1) The scanned point cloud data is clustered and ground segmented to segment out the non-ground points, while the point cloud data is filtered. In the process of feature matching, the distance image level is evenly divided into equal sub-images, and the plane smoothness of the points in each row of each sub-image is calculated, and the plane smoothness is used as an index to extract the feature information of the current frame, and the feature points are divided into edge points and planar points according to the plane smoothness as an index, and the formula for calculating the plane smoothness is shown in equation (1):

$$c = \frac{1}{|S| \cdot \|r_i\|} \left\| \sum_{j \in S, j \neq i} r_j - r_i \right\| \tag{1}$$

where S is the set of 5 points before and after point i in the same direction; r_i is the Euclidean distance from point i to LIDAR in the distance image; r_j is the Euclidean distance from point j to LIDAR in the distance image.

2) The extracted feature points are traversed to distinguish the edge points from the plane points, and the constraint relations from the edge points to the straight line and the plane points to the plane are constructed respectively. The distance from an edge point to a straight line is shown in equation (2):

$$D_\varepsilon = \frac{\left| \left(\overline{X}^L_{(k+1,i)} - \overline{X}^L_{(k,j)} \right) \times \left(\overline{X}^L_{(k+1,i)} - \overline{X}^L_{(k,l)} \right) \right|}{\left| \overline{X}^L_{(k,j)} - \overline{X}^L_{(k,l)} \right|} \tag{2}$$

where D_ε is the shortest distance from the edge point to the straight line; $\overline{X}^L_{(k,i)}$ is the coordinate of point i in the kth complete scan in the radar coordinate system; $\overline{X}^L_{(k,i)}$ is the coordinate of point j in the kth complete scan in the radar coordinate system; $\overline{X}^L_{(k+1,i)}$ is the coordinate of point i in the k+1st complete scan in the radar coordinate system; and $\overline{X}^L_{(k+1,i)}$ is the coordinate of point j in the k+1st complete scan in the radar coordinate system.

The distance from the plane point to the plane is shown in equation (3):

$$D_{\mathrm{H}}=\frac{\left|\left(\bar{X}_{(k+1,i)}^{L}-\bar{X}_{(k,j)}^{L}\right)\cdot\left(\bar{X}_{(k,j)}^{L}-\bar{X}_{(k,n)}^{L}\right)\times\left(\bar{X}_{(k,l)}^{L}-\bar{X}_{(k,m)}^{L}\right)\right|}{\left|\left(\bar{X}_{(k,j)}^{L}-\bar{X}_{(k,\eta)}^{L}\right)\times\left(\bar{X}_{(k,j)}^{L}-\bar{X}_{(k,m)}^{L}\right)\right|} \tag{3}$$

Where D_H is the shortest distance from the plane point to the plane; $\bar{X}_{(k,m)}^{L}$ is the coordinate of point m in the kth complete scan under the radar coordinate system; $\bar{X}_{(k+1,m)}^{L}$ is the coordinate of point m in the k+1th complete scan under the radar coordinate system.

3) Calculate the distance between the current frame point cloud data and the previous three frames point cloud data by the point to line and point to plane distance formula, when there are two frames of point cloud data with close distance, match the edge points or plane points in them, for the edge points, match them by the segmented segmentation point labels; for the plane points, match them by the segmented plane point labels. Comparing the difference between the description vectors of the two frames of point cloud data, the similarity of the two frames of data is calculated, and the similarity calculation process is shown below:

$$\cos(\theta)=\frac{\sum_{i=1}^{n}\left(x_i\times y_i\right)}{\sqrt{\sum_{i=1}^{n}\left(x_i\right)^2}\times\sqrt{\sum_{i=1}^{n}\left(y_i\right)^2}} \tag{4}$$

where $cos(\theta)$ is the cosine of the two vectors; n is the dimension of the vector.

The range of cosine value is [-1,1], when $cos(\theta)$ is closer to 1, it represents that the direction of the two vectors is closer, the higher the similarity of the two vectors; when $cos(\theta)$ is closer to -1, it represents that the direction of the two vectors tends to the opposite direction, the lower the similarity of the two vectors.

4) When the similarity between two frames of data that are close to each other is high, the feature similarity matching is recognized to be completed, and the current frame of point cloud data is used to project the bit position through a frame of point cloud data at the previous moment. The error function is minimized by calculating the rotation matrix and translation matrix between the two point clouds. Assuming that the two point clouds are $X=\{x_1,x_2,\ldots,x_{np}\}$, $p=\{p_1,p_2,\ldots,p_{nx}\}$, and the error function is shown in equation (5):

$$E(R,t)=\frac{1}{n}\sum_{i=1}^{n}\left\|x_i-Rp_i-t\right\|^2 \tag{5}$$

where R is the rotation matrix; t is the translation matrix.

The rotation matrix R and the translation matrix t satisfy the following relationship:

$$x_i=Rp_i+t \tag{6}$$

Calculate the centre of mass of the two sets of point clouds as shown in equations (7) and (8):

$$u_x=\frac{1}{n}\sum_{i=1}^{n}X_i \tag{7}$$

$$u_p=\frac{1}{n}\sum_{i=1}^{n}P_i \tag{8}$$

The two sets of point clouds are moved towards the origin to remove the centre of mass, and the coordinates of the two point clouds after the removal of the centre of mass are shown in Eqs. (9) and (10):

$$X=\left\{X_i-u_i\right\} \tag{9}$$

$$P=\left\{P_i-u_p\right\} \tag{10}$$

Set the matrix W:

$$W=\sum_{i=1}^{n}x_i p_i^T \tag{11}$$

Obtained by SVD decomposition:

$$W=U\sum V^T \tag{12}$$

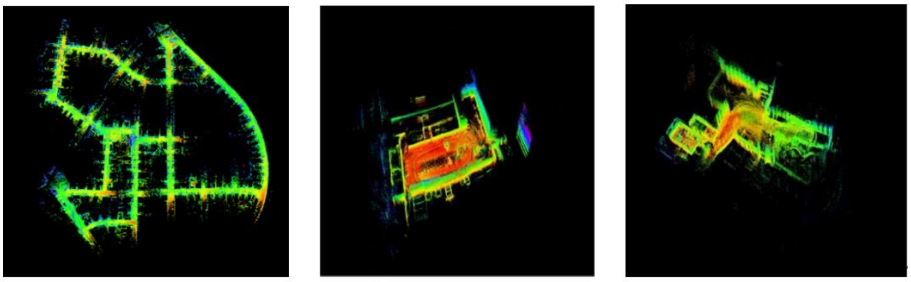
The rotation matrix R and the translation matrix t can be obtained:

$$R = UV^T \tag{13}$$

$$t = u_x - Ru_P \tag{14}$$

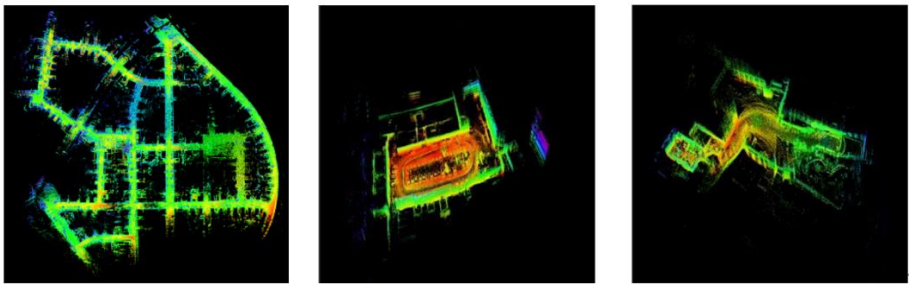
After aligning the two point cloud frames by the obtained rotation and translation matrices, each feature set is bound to the LiDAR's position to generate a local feature point cloud map, and the local point cloud data is fused to the global map, and the final global map is obtained through optimization.

In this paper, the feasibility, reliability and accuracy of the improved LeGo-LOAM algorithm are verified by the KITTI dataset, and the verification experiments are designed as follows: the same dataset is run using the LeGo-LOAM algorithm and the improved LeGo-LOAM algorithm, and the LeGo-LOAM algorithm and the improved LeGo-LOAM algorithm are compared with the improved LeGo-LOAM algorithm in terms of the graph building effect, the running time, and the CPU occupancy rate. LeGo-LOAM algorithm and the improved LeGo-LOAM algorithm are compared in three aspects: building effect, running time and CPU utilization. In order to ensure the accuracy of the test, this paper conducts several tests and takes the average value. The map constructed by LeGo-LOAM algorithm is shown in Figure 7, and the map constructed by the improved LeGo-LOAM algorithm is shown in Figure 8. The comparison results of the two algorithms are shown in Table 1.



(a) (b) (c)

Fig. 7 – Maps constructed by the LeGo-LOAM algorithm



(a) (b) (c)

Fig. 8 – Maps constructed by the improved LeGo-LOAM algorithm

As can be seen from the comparison of Fig. 7 and Fig. 8, the maps constructed by the improved LeGo-LOAM algorithm are clearer compared to the maps constructed by the LeGo-LOAM algorithm, which reduces the error generated by the accumulation, and handles the edge contours better *(Wei et al., 2023)*.

As can be seen from Table 1, the two algorithms have roughly the same building time, and the CPU occupancy rate of the improved LeGo-LOAM algorithm is slightly higher than that of the LeGo-LOAM algorithm, but the difference is very small. Comparing the two algorithms in terms of building time, building effect and CPU occupancy, the improved LeGo-LOAM algorithm is better than LeGo-LOAM algorithm.

Table 1

Algorithm comparison test results

Data set	Algorithm	Time (min)	CPU Occupancy (%)
First data set	LeGo-LOAM	1.50	24
	Ours	1.50	29
Second data set	LeGo-LOAM	1.25	23
	Ours	1.24	25
Third data set	LeGo-LOAM	2.43	31
	Ours	2.44	36

Inspection Robot Path Planning System

The global path planning takes the 2D raster cost map converted from the 3D point cloud map as input, and takes the shape of the robot, the dynamics model and the motion performance into consideration when planning the path, and adopts the multi-constraint Bessel curve fusion to improve the A-star path planning algorithm to plan the optimal path from the robot's current position to the desired target position, and in this process, it is necessary to continuously adjust the robot's position in the process of motion, so that new paths can be generated by each iteration, and through the continuous iteration, the optimal path of the global path in the process of navigation of the robot in the orchard is obtained, as shown in Fig. 9.

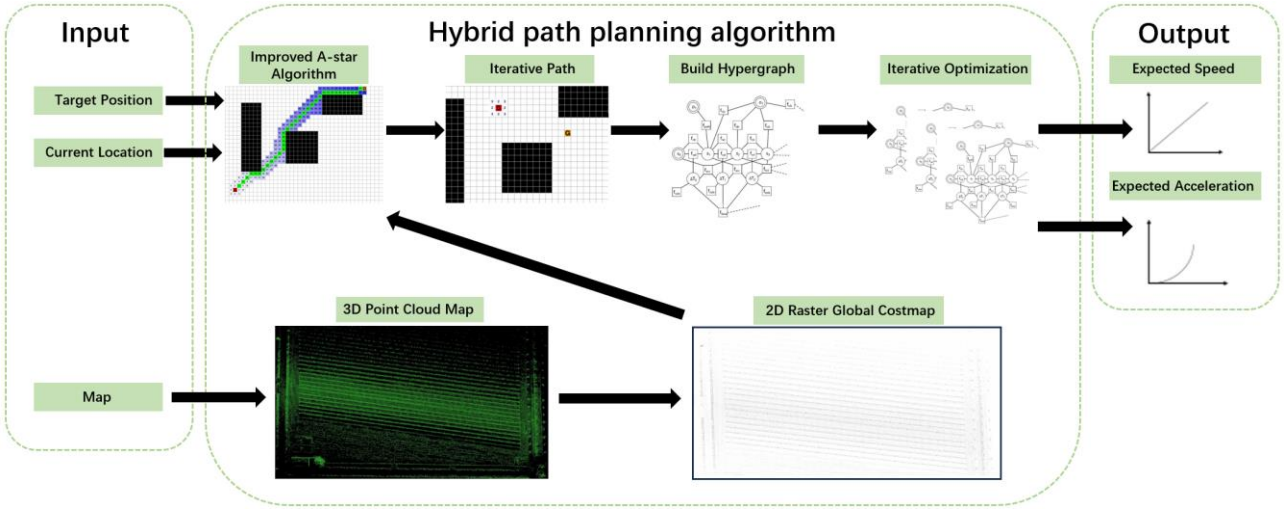


Fig. 9 – Flowchart of the route planning system

The advantage of A-star algorithm as a common heuristic search algorithm is that it can find the optimal solution quickly and also ensure the correctness of the optimal solution when pathfinding and path planning work.

The cost function $f(n)$ of the A-star algorithm is denoted as:

$$f(n) = g(n) + h(n) \tag{15}$$

Here, a simplified version of the vineyard simulation map is drawn using MATLAB to test the improved A-star full-coverage path planning algorithm. The map size is 30*30 raster units as shown in Fig.10.

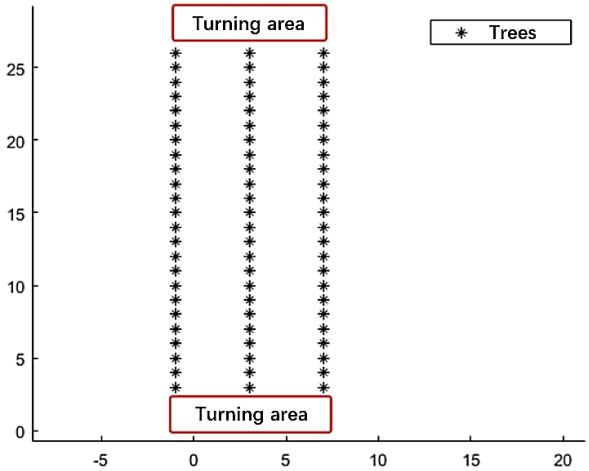


Fig. 10 – Simulated Orchard Environment Map

The traditional A-star algorithm global path planning work is shown in Fig.11, and the A-star full-coverage path planning algorithm for global path planning work is shown in Fig.12.

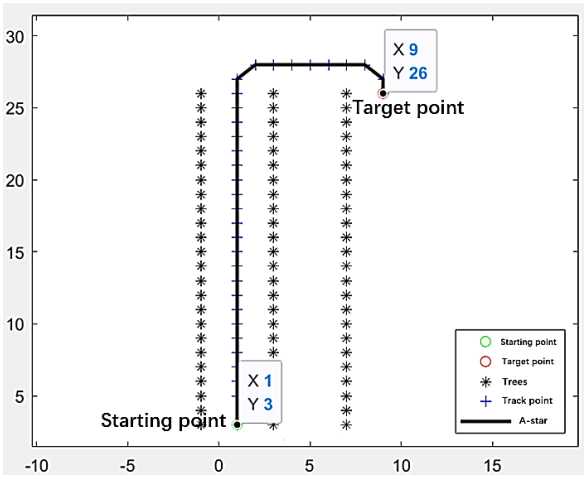


Fig. 11– Simulation of Traditional A-star Algorithm Planning

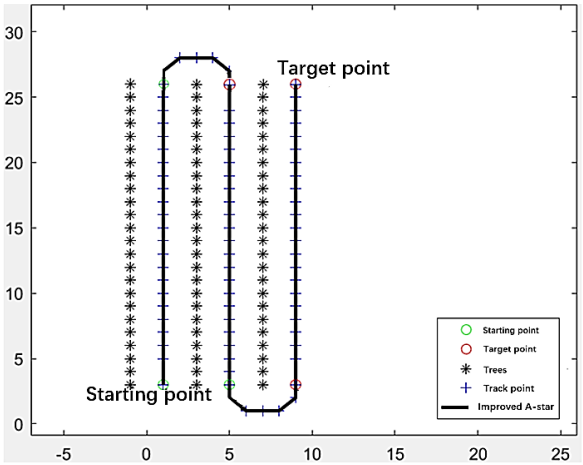


Fig. 12 – Simulation of Improved A-star Full Coverage Path Planning Algorithm

Relative to Fig. 11, it can be seen from Fig. 12 that the improved A-star full-coverage path planning algorithm can plan a collision-free, full-coverage shortest path.

However, the improved path trajectory in Fig. 12 has more turns and folds, which can cause the robot to change direction during travel and thus lose control, and the efficiency is greatly reduced, so Bessel curves will be used to plan the paths for the improved A-star algorithm under kinematic constraints to make them closer to the theoretical travel paths.

The shape of the Bezier curve is determined by the control points. Different Bezier curves can be created by adjusting the position and number of control points.

When n=1, the first order Bessel curve is a straight line with the expression.

$$P^1(t) = (1-t)P_0 + tP_1 = \sum_{i=0}^{1} C_1^i (1-t)^{1-i} t^i P_i \tag{16}$$

When n=2, the second-order Bessel curve is parabolic and at most the first-order derivative is not constant, with the expression.

$$P^2(t) = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2 = \sum_{i=0}^{2} C_2^i (1-t)^{2-i} t^i P_i \tag{17}$$

The formula for the third-order Bessel curve when n=3 , with at most the second-order derivatives not being constant, is expressed as.

$$P^3(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3 = \sum_{i=0}^{3} C_3^i (1-t)^{3-i} t^i P_i \tag{18}$$

When n≥3, the curve is a higher order Bessel curve. The first to third order Bessel curve images are shown in Fig. 13.

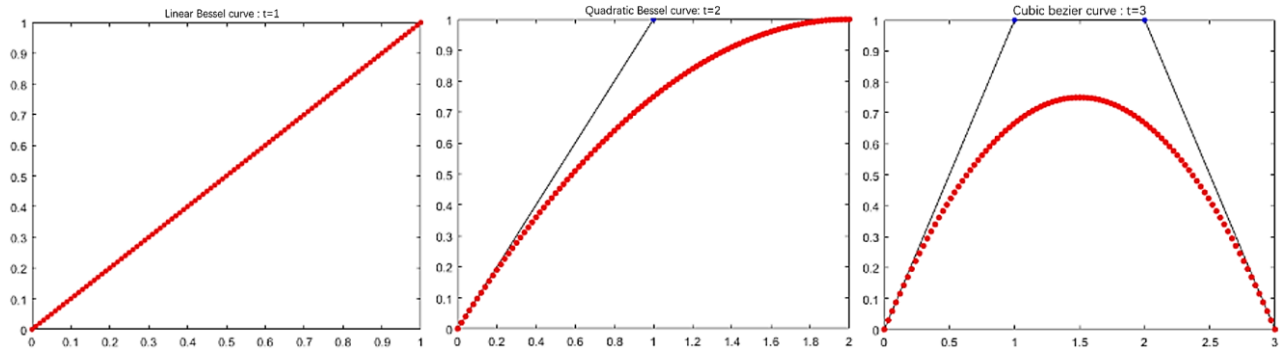


Fig. 13 – First- to third-order Bessel plots

When the trajectory is smoothed, it should be ensured that the robot kinematic constraints are satisfied to ensure that the cost is minimized when the robot trajectory is tracked. The curvature continuity constraint is of paramount importance in ensuring smooth turning during robot travel. In order to avoid unstable phenomena such as violent turns during trajectory tracking by the robot, it is essential that the curvature changes of adjacent path segments on the path are continuous. This is achieved by ensuring that the third-order Bessel curve, which is the basis of the curvature continuity constraint, has continuous second-order derivatives.

The flowchart of the improved A-star algorithm based on constraint Bessel curves is shown in Fig. 14.

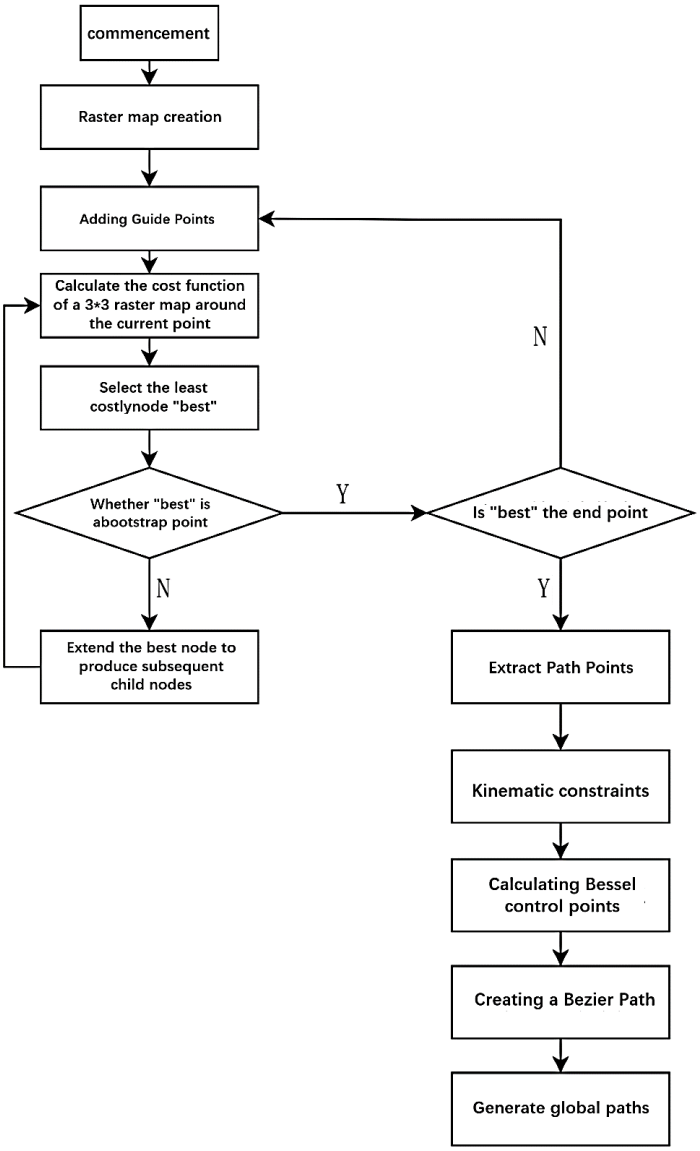


Fig. 14 – Flowchart of Improved A-star Algorithm Based on Constraint Bessel Curve

Through Figure 15, it can be found that the A-star algorithm with added Bezier curve optimization is smoother in the ground head turnaround region, which is more suitable for the orchard scene operation, while the global path optimized by the unconstrained Bezier curve has an abrupt change of curvature in the connection between the straight line and the turnaround path, which is not smooth.

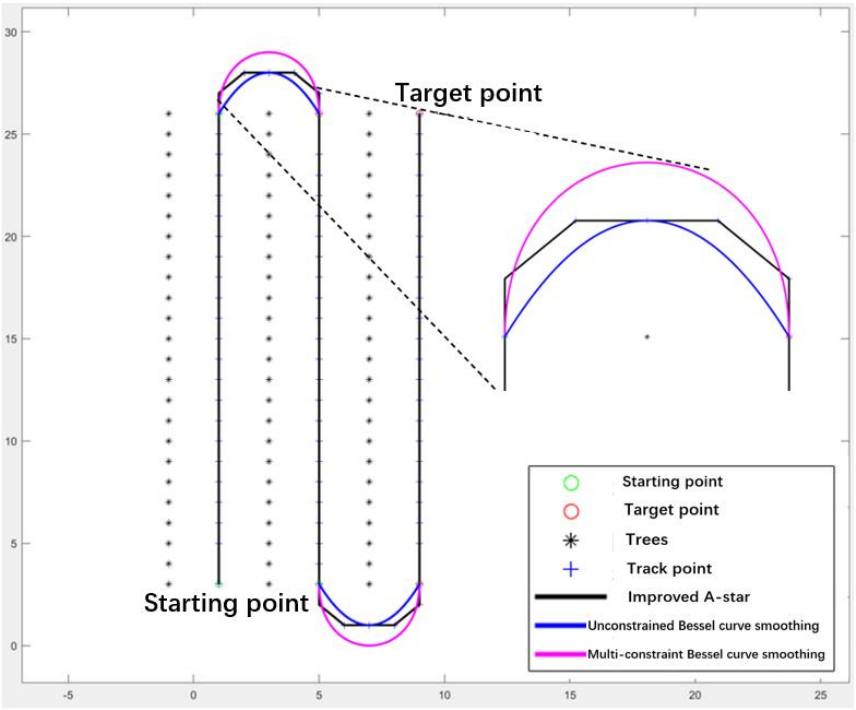


Fig. 15 – A-star Algorithm before and after Bessel Curve Optimization

RESULTS

Details of SLAM algorithm testing based on multi-line LiDAR

Before the orchard robot works, it establishes a three-dimensional point cloud map for the orchard and transforms it into a two-dimensional raster map that can be used for navigation, and generates an optimized global path according to the two-dimensional raster map in the Robot Operating System (ROS) to start the unmanned inspection operation at the starting point. In the process of robot driving, the artificial intelligence processor receives the data from the localization receiver through the serial communication, and analyses and transforms it into the localization point information under the absolute coordinate system, and receives the localization point information through the control module, and after calculation, it sends the linear velocity and angular velocity information to the bottom control driver board of the tracked vehicle, so as to let the vehicle carry out the inspection operation along the globally planned path.

First, the inspection operation area is selected, because the path in the vineyard is mainly divided into two parts, one is the straight line path in the inspection area, and the other is the turnaround path in the ground turnaround area, so two rows of vineyard are selected as the test area, as shown in Figure 16.

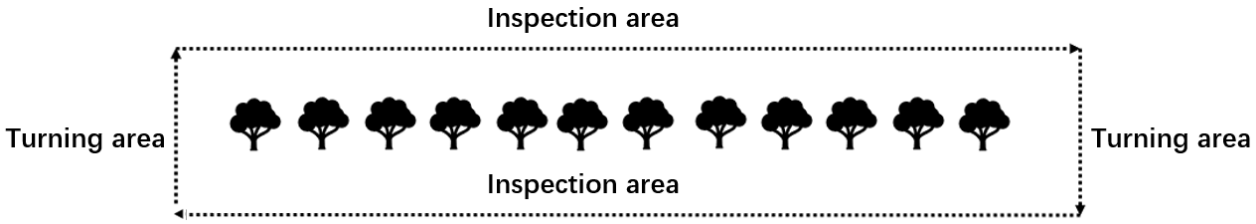


Fig. 16 – Schematic diagram of the inspection operation area

Second, as shown in Figure 16, the orchard inspection robot is controlled to loop back at 1.0 m/s according to the rectangular frame, record the 3D point cloud data at 10 Hz, and run the improved SLAM algorithm in order to construct a high-precision 3D point cloud map of the vineyard, as shown in Figure 17.

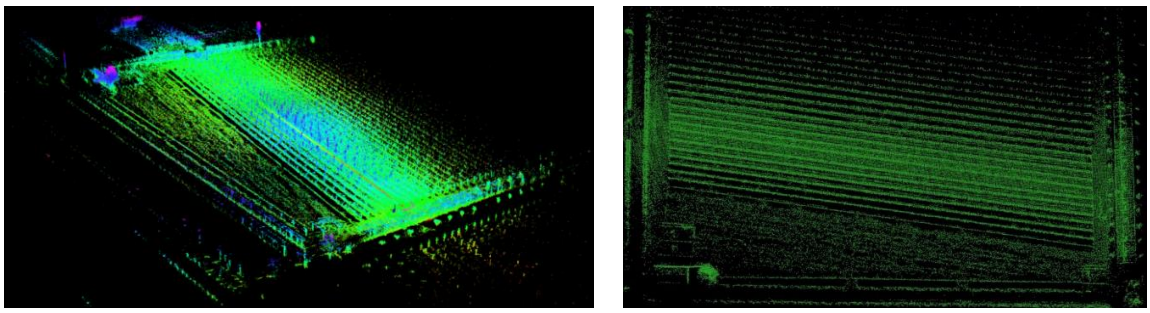


Fig. 17 – High-precision 3D point cloud map of the vineyard

Finally, the map conversion function package is launched to generate and update the 2D raster map from the high precision map in real time. The generated 2D raster map is shown in Figure 18.

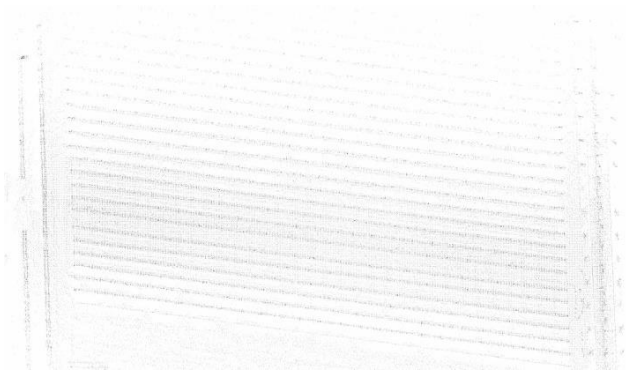


Fig. 18 – Vineyard 2D Grid Map

The 3D point cloud map constructed by the improved SLAM algorithm is clearer and denser than the map constructed by the LeGo-LOAM algorithm, the data is more complete, the processing of the edge contour is better, and the converted 2D raster map can satisfy the robot's navigational accuracy requirements in the orchard environment.

Inspection Robot Path Planning System Test Details

When launching the Global Path Planning feature package, you need to import a 2D raster map of the test area. The map information is obtained by subscribing to the map information through the ROS theme. Then specify the starting and target points of the orchard robot and add guidance points. The global path is generated by the comb-type full-coverage algorithm and the improved A-star path planning algorithm with multi-constraint Bessel curves, and the global path is displayed using MATLAB, as shown in Fig. 19.

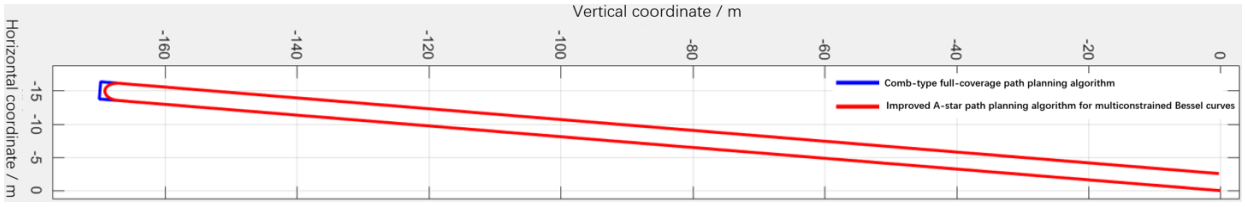


Fig. 19 – Improved A-star path planning algorithm for multi constrained Bessel curves

From Fig. 19, it can be seen that the A-star algorithm with the addition of Bessel curve optimization is smoother in the ground-head turning region, which is more suitable for the operation of the orchard scene. Next, the navigation parameters are set, the sampling time is set to 0.1s, the traveling speed is set to 1m/s, the basic forward-looking distance is set to L_{d0}=1m, the gain coefficient of pure tracking algorithm speed control is set to K_{v}=1, and the integration adjustment coefficient is set to K_i=0.5, and the initial lateral bias and heading error are not set, and the startup command is inputted into the ROS terminal to control the orchard robot to start trajectory tracking and to record the Robot position information.

Save the driving trajectory point data in txt format and import the txt format file into MATLAB to display the trajectory as shown in Fig. 20.

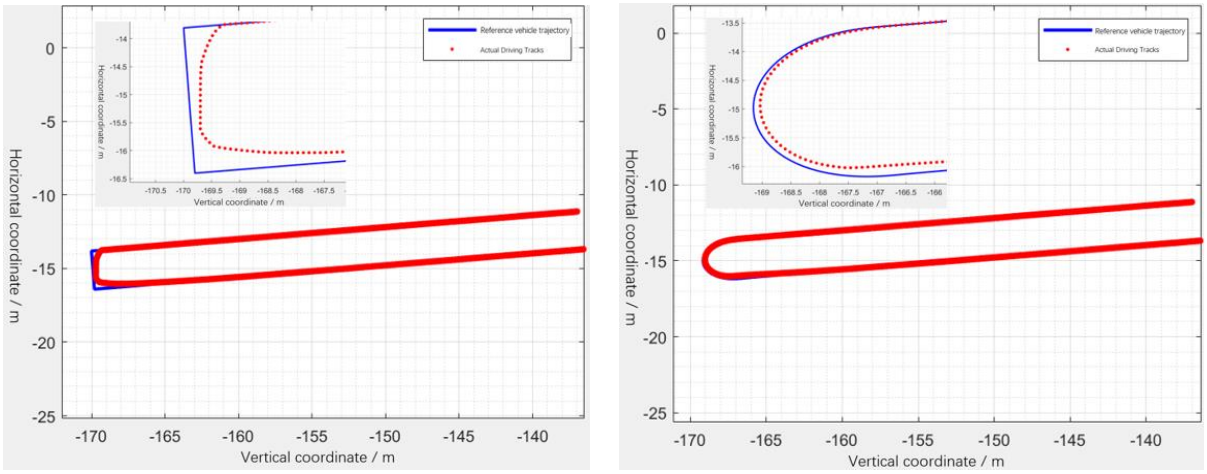


Fig. 20 – Comparison chart of tracking effect

The lateral deviations of the sampled points are saved in txt format, and then the txt format file is imported into MATLAB to display the trajectory tracking errors of the two planning algorithms, as shown in Fig. 21 and Table 2.

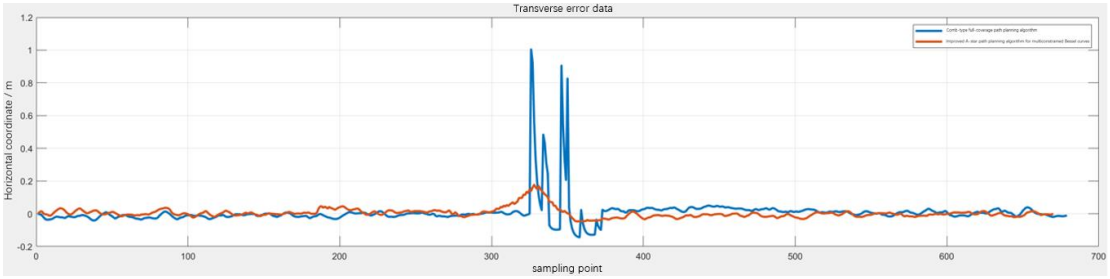


Fig. 21 – Comparison of lateral tracking deviation between two algorithms for path planning

Table 2

Lateral deviation of two planning paths under a pure tracking algorithm

Route planning methodology	Different test paths	Average lateral deviation/m	Maximum lateral deviation/m
Comb-type full coverage	linear path	0.0162 m	0.0505 m
	U-turn path	0.2580 m	1 m
	Overall path	0.0300 m	1 m
Improved A-star for multiconstrained Bessel curves	linear path	0.0157 m	0.0457 m
	U-turn path	0.1081 m	0.1768 m
	Overall path	0.0176 m	0.1768 m

Experimental analysis shows that the path generated by the improved A-star algorithm based on constrained Bessel curves meets the requirements of full-coverage trajectory smoothing, and experiments show that the average lateral deviation of the straight line path during trajectory tracking is 0.0157 m, while the average lateral deviation of the trajectory tracking turn path is 0.1081 m, and the robot travels smoothly and the steering amplitude fluctuates less to meet the requirements of the robot autonomous unmanned operation.

CONCLUSIONS

This paper presents a novel autonomous detection and navigation system for orchard robots that integrates SLAM (Simultaneous Localization and Mapping) and path planning algorithms. Field tests in vineyards have validated the efficacy of the algorithms proposed in this paper. The research outcomes are of paramount importance in reducing labour costs and fostering the advancement of precision agriculture.

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] Wei Y., et al. (2023) *Research on orchard robot navigation system based on fusion of LiDAR and millimeter wave radar (基于激光雷达和毫米波雷达融合的果园机器人导航系统研究). Shandong Agricultural University*.

[12] 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.

[13] 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.

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