RESEARCH ON THE CONSTRUCTION METHOD OF HIGH PRECISION 3D POINT CLOUD MAP FOR AGRICULTURAL ENVIRONMENTS

/

农业环境高精度三维点云地图构建方法研究

Bo ZHAO[1], Suchun LIU[1], Xiufeng Zhao[2], Licheng ZHU[1*], Tianfu ZHANG[1], Zhenhao HAN[1], Weipeng ZHANG[1]

[1] National Key Laboratory of Agricultural Equipment Technology, China Academy of Agricultural Mechanization Science Group Co., Ltd, Beijing 100083, China;

[2] Nong'an County Agricultural Mechanization Technology Promotion Station, Changchun, Jilin 130200, China

Tel: +86-13601287800; E-mail: zhulicheng@caams.org.cn; Correspondent author: Licheng ZHU

DOI: https://doi.org/10.35633/inmateh-72-35

***Keywords:** Point cloud high-definition map; driverless; semantic segmentation*

ABSTRACT

In agricultural operation scenarios, the diversity of farmland terrain, crops and other forms, as well as uncertain factors such as weather changes and crop growth during agricultural operation, can have an impact on the construction of high-precision maps. To address these challenges and analyze operational scenarios based on the characteristics of agricultural scenarios, this paper proposes a point cloud map construction algorithm for plant point removal and locatability estimation. Based on the existing Simultaneous Localization and Mapping (SLAM) framework, plant point removal and locatability estimation are improved. Firstly, Red, Green, Blue (RGB) images and Near Infrared (NIR) images are fused to identify and remove plant point clouds, preserving effective inter frame matching information, reducing the impact of dynamic points on inter frame matching, and achieving high front-end motion estimation accuracy. Then, the localization estimation method based on learning is used to determine the motion estimation status and determine whether to execute the backend optimization algorithm. Finally, the back-end optimization algorithm based on Factor graph is designed, and the Factor graph, constraint relationship and optimization function are constructed to optimize the pose of all frames. The optimized map construction algorithm reduces the re projection errors between field roads, paths, and crop rows by 10.27%, 20.76%, and 14.36% compared to before optimization. To verify the actual operational effectiveness of the point cloud map construction algorithm, the hardware part of the multi-sensor information collection system was designed, and sensor internal and external parameter calibration were also carried out. A map information collection vehicle was built and field experiments were conducted. The results showed that the positioning error of the point cloud map construction method proposed in this paper is less than 0.5°, and the cumulative error of 30 m translation is less than 12 cm, which meets the actual operational requirements.

摘要

在农业作业场景中，农田地形、作物等形态的多样性，以及农业作业过程中的天气变化、作物生长等不确定因素，都会对高精度地图的构建产生影响。为了应对这些挑战，并根据农业场景的特点分析操作场景，本文提出了一种用于植物点去除和定位估计的点云地图构建算法。在现有 SLAM 框架的基础上，对植物点去除和可定位性估计进行了改进。首先，将 RGB 图像和近红外图像融合，识别和去除植物点云，保留了有效的帧间匹配信息，减少了动态点对帧间匹配的影响，实现了较高的前端运动估计精度。然后，使用基于学习的局部估计方法来确定运动估计状态，并确定是否执行后端优化算法。最后，设计了基于因子图的后端优化算法，构造了因子图、约束关系和优化函数，对所有帧的姿态进行优化。与优化前相比，优化后的地图构建算法将田间道路、路径和作物行之间的重新投影误差分别降低了 10.27%、20.76%和 14.36%。为了验证点云地图构建算法的实际操作有效性，设计了多传感器信息采集系统的硬件部分，并进行了传感器内外参数校准。建造了地图信息采集车，并进行了野外试验。结果表明，本文提出的点云地图构建方法定位误差小于0.5°，30m 平移累积误差小于12cm，满足实际操作要求。

INTRODUCTION

The need for unmanned and refined agriculture has promoted the development of intelligent technologies such as driverless, robots and sensors (*Kim et al., 2022*). The use of unmanned technology in agricultural fields requires high-precision maps to provide vehicles with relevant information about the environment and this information helps self-driving vehicles perceive their surroundings and make navigation decisions. The highly unstructured nature of agricultural scenarios and the complexity of semantic information bring great challenges to the construction of high-precision maps. To cope with these challenges, agricultural high-precision point cloud map construction is needed, i.e., to construct a map containing environmental information through multiple point cloud datasets. The point cloud data in the map can provide useful semantic information for robots or other intelligent devices to support their autonomous localization and navigation in unknown environments (*Qi et al., 2017; Wang et al., 2022*).

Numerous scholars have carried out research in various aspects of point cloud map construction. Ji et al. developed a LiDAR-based point cloud acquisition system for farmland environment (*Ji et al., 2019*), aiming to realize the stable and reliable information acquisition of farmland environment point cloud and positional attitude of farm machinery. The system employs multi-sensor data acquisition software to achieve accurate and consistent global point cloud data acquisition. Chris et al. proposed a methodology using high-resolution LiDAR point cloud data for classification and segmentation to detect linear vegetation elements in agricultural landscapes (*Lucas et al., 2019*). Due to the ground and trees are surrounded by foliage, thorns, and vines, and sensors often experience extreme motion, Guilherme et al. proposed an end-to-end trunk diameter estimation method that is based on semantic segmentation and LiDAR odometry with map building (*Chen et al., 2020*). Poor feature descriptiveness and insufficient point cloud alignment accuracy due to the lack of highly distinguishable high-level structures of points, lines, and surfaces, therefore, Dong et al. proposed a point cloud method based on rotated surface contour features for farmland surface point clouds (*Dong et al., 2020*).

Aiming at the problems of highly unstructured scene and complex environment semantic information in the construction of high-precision point cloud semantic map for agriculture, this paper proposes a method of multimodal map construction based on multi-sensors. By designing a multi-sensor information acquisition device and proposing a point cloud map construction algorithm and a multi-modal semantic segmentation algorithm, high-precision point cloud semantic maps are established to provide accurate data support for unmanned operation of agricultural machines.

MATERIALS AND METHODS

Data Acquisition Vehicle Construction

The overall design of the data acquisition vehicle is shown in Figure. 1, which consists of the information acquisition device, the modified bracket and the vehicle underpan, respectively. The Livox-mid70 non-repeating scanning lidar, Intel RealSense D455 depth camera and MER2-507-23GM NIR near-infrared optical camera are used for data acquisition, respectively. The collection vehicle uses the Agile HUNTER2.0 chassis, which features a wheel drive system, Ackermann turning mode, and independent suspension system. This setup provides a reliable platform with precise control and smooth movement.

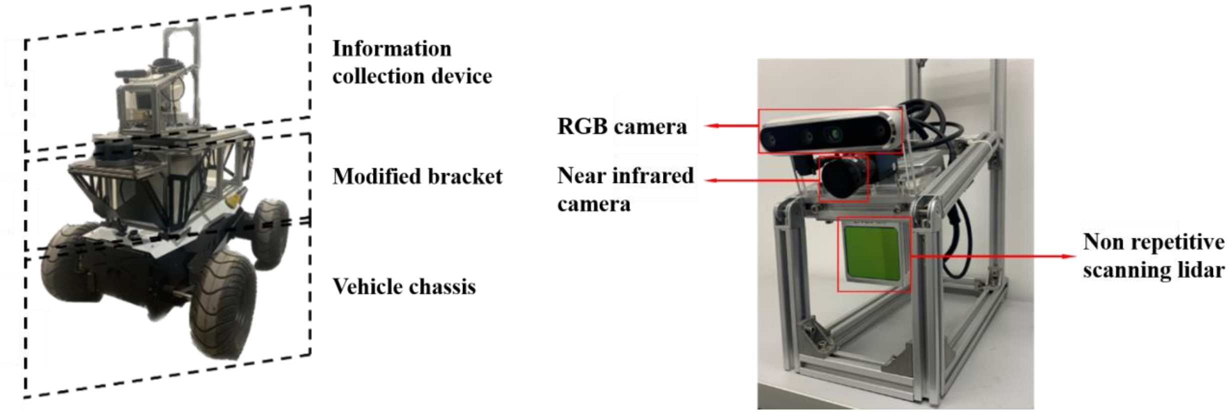

Fig. 1 - The information collection vehicle

Principle of Point Cloud Map Construction

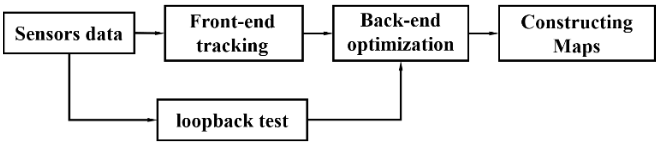

Fig. 2 - Process of the classic SLAM algorithm

The construction of a point cloud map relies heavily on the use of the SLAM (Simultaneous Localization and Mapping) technique (*Zeng et al., 2022; Johnson et al., 1999; Li et al., 2021*). Its algorithmic process is exemplified in Figure. 2 and is employed to achieve simultaneous localization and map construction by continually estimating and updating the robot movements and surroundings for independent localization and navigation. The key stages of the SLAM technology used during point cloud map creation encompass point cloud data acquisition, motion estimation, point cloud alignment, point cloud feature extraction, point cloud filtering and optimization, and map construction.

During point cloud data acquisition, several sensors are employed to gather point cloud data within the environment and then process and fuse it to enhance the completeness of environmental information. Front-end tracking utilizes camera acquired image frames captured during motion at distinct times to solve the camera position transformation between neighboring domains through feature matching. This facilitates image frame fusion to reconstruct the map (*Tombar et al., 2010; Johnson et al., 1999*). However, the estimation of the position at the frontend over an extended period will lead to error accumulation during the moving process, leading to offset phenomenon. To decrease the accumulation of mistakes, loopback detection and backend optimization techniques are introduced. Loopback detection aims to establish whether the same position is achieved again, whereas backend optimization improves the noise problem in the position estimation (*Han et al., 2022; Han et al., 2022; Xu et al., 2022*). After detecting feedback loops and optimizing the camera positions obtained from motion estimation, it is possible to generate globally consistent trajectories and their corresponding map forms.

POINT CLOUD MAP CONSTRUCTION ALGORITHM

Algorithm Framework

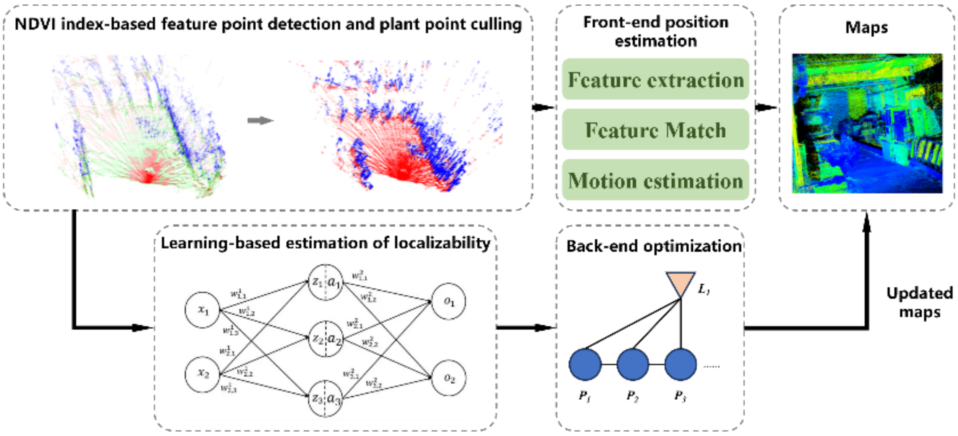

Fig. 3 - Point cloud map construction algorithm framework

In the frontend part of the algorithm, edge and feature points are first extracted by extracting scene information such as non-vegetation points (e.g. ground, obstacles and debris). Feature point matching is performed using the nearest neighbor method, whilst the Iterative Closest Point (ICP) algorithm estimates motion to determine the interframe locality matching. During the backend optimization phase is utilized the optimization algorithm that is rooted in factor graph (*Vizzo et al., 2022; Wang et al., 2021*). This optimization algorithm is used to enhance the position and feature points of all frames by constructing factor graph, constraint relationships and optimization functions, thereby improving the accuracy of the map. Following the generation of a high-precision 3D map, execution of the map building algorithm results in an accurate map being created. The complete algorithm framework is depicted in Figure 3.

Plant Point Culling Based on Image Information

This paper utilizes the Normalized Difference Vegetation Index (NDVI) to separate plant and non-plant regions (*Ao et al., 2021; Tian et al., 2021*). By combining the RGB and NIR images, a multispectral image with red, green, blue, and near-infrared band information can be obtained. Segregation of plant and non-plant regions can be accomplished through the use of the $NDVI$. The normalized vegetation index, calculated using the NIR and red bands in a multispectral image, can be expressed through Equation (1).

$$NDVI = \frac{NIR - Re\,d}{NIR - Re\,d} \tag{1}$$

where NIR denotes the pixel value in the near-infrared band and Red denotes the pixel value in the red band. As shown in Figure 4, the non-vegetative points can be segmented by extracting non-vegetative data from the camera image and mapping it onto the point cloud.

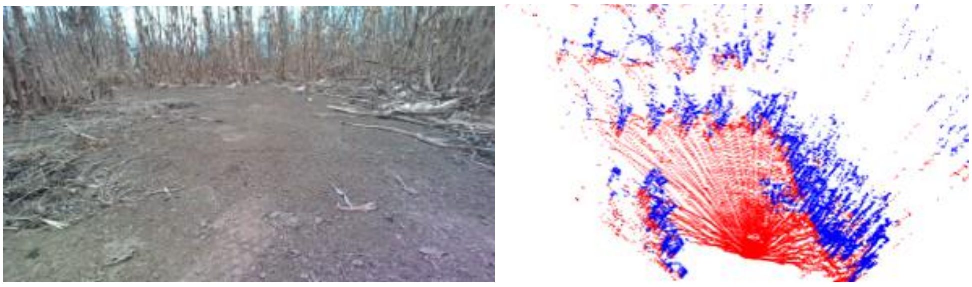

Fig. 4 - Plant spot removal in field experiments

Learning-Based Estimation of Localizability

Conventional algorithms may degrade, which can result in a significant deviation from the estimated bit position. To detect such a situation, an end-to-end method for estimating bit position is proposed. This method helps identify whether the conventional approach is degraded or not.

- *Mathematical modeling*

3D LIDAR point cloud scanning data is analyzed to forecast the feasibility of accomplishing localizations in the existing surroundings by employing scan matching methods. The LiDAR point cloud data is exploited to supply an approximation of localization, which is described as the identification across six degrees of freedom. The LiDAR point cloud data is exploited to supply an approximation of localization, which is described as the identification across six degrees of freedom. Abbreviations will be clarified whenever initially employed.

$$D_k = (d_x, d_y, d_z, d_\varphi, d_\theta, d_\psi)^T \tag{2}$$

x, y, z – the translation coordinate;

φ, θ, ψ – the Euler angles for roll, pitch, and yaw; [°]

For practical purposes, the localization of orientation is considered in binary terms. Therefore, for all $i \in \{x, y, z, \varphi, \theta, \psi\}$, there exists $d_i \in \{0,1\}$, where 0 indicates successful localization and 1 indicates unsuccessful localization. Localization is accomplished through interframe matching of the current and previous frame point clouds.

The ability to localize at time *k* is determined by scanning data from the current point cloud$s_k \in R^{n_k \times 3}$. Therefore, the function $sk \to dk$ can be approximated $\widetilde{D}_k(\Theta, s_k)$, where $\Theta \in R^p$ is the P trainable parameters in the network. These parameters are obtained by minimizing the supervised classification loss $\underset{\theta}{\operatorname{argmin}}(\widetilde{D}(\Theta, S), T)$ in the training set S, which includes the point cloud frames $s_i \in S$ and corresponding labels $t_i \in T$.

- *Training set generation: sampling and alignment*

Generating the data for training occurs in two stages: initially, the alignment error e_k is calculated; subsequently, this error is mapped to d_k through the implementation of a thresholding operation.

- Alignment Error

Obtain a set of child point clouds adjacent to the parent point cloud s_p through Monte Carlo sampling to calculate the expected alignment error. Align the child point clouds with the parent point cloud to obtain the average alignment residuals of the point cloud distribution. The mean absolute error is then calculated to obtain the alignment error e_p of the s_p.

A group of *M* sub-point clouds s_p are randomly sampled using Monte Carlo within the vicinity of $s_{c,j} \in \{1, \dots, M\}$. To determine its positional attitude $T_{p,c,j}$ relative to the main point cloud, each sub-point cloud requires six perturbations in the direction $i \in \{x, y, z, \varphi, \theta, \psi\}$ using a Gaussian function $N(0, \sigma 2i)$. The alignment of each sub-point cloud with the main point cloud is achieved using the ICP algorithm with point-to-plane implementation, resulting in the transform matrix $T_{p,c,j}$. The sub-point cloud's alignment quality to the parent point cloud is assessed by computing the mean absolute error between each sub-point cloud and its aligned point cloud on the parent point cloud. Subsequently, the mean absolute error of all child point clouds is averaged to obtain the alignment error e_p of the parent point cloud s_p.

$$e_p = \frac{1}{M} \sum_{j=1}^{M} \left| \gamma\left(\tilde{T}_{p,c,j}{}^{-1} - T_{p,c,j}\right) \right| \tag{3}$$

The alignment error γ for each sub point cloud is determined by aligning it with the parent point cloud using ICP. The $T_{p,c,j}$ attitude of each sub point cloud is converted to $\tilde{T}_{p,c,j}$ in relation to the parent point cloud, as indicated in Equation (3), and is then utilized as input for the ICP to compute the alignment error γ.

- *Localizability estimation*

Once the 6D alignment error has been calculated, it must be compared with a preset threshold to determine whether localization can be successful in a particular direction. If the alignment error exceeds the preset threshold, positioning in that direction is considered to have failed and the corresponding Localizability label is set to 1. Conversely, it is set to 0. Positioning is considered successful if the translation error is less than 10 cm and the rotation error is less than 2°.

Frontend position estimation

Frontend pose estimation consists of three stages: feature extraction, feature matching and motion estimation.

- *Feature extraction*

A 3D curvature based feature extraction method is used to extract points with distinct geometric features. The point cloud is meshed and divided into a number of small cubes, and the center point pi within each small cube is computed. For each point p_i, its normal vector n_i is computed. the curvature tensor C_i of each point p_i is computed, where C_i is computed from the normal vectors of all points in its neighborhood, and the calculation formula is as Equation (4).

$$C_i = \frac{1}{N}\sum_{j=1}^{N}(n_j - n_i)(n_j - n_i)^T \tag{4}$$

N – the number of points in the neighborhood of p_i;

n_j – the normal vector of the j^{th} neighborhood point.

For every p_i, determine the eigenvalues λ_1 and λ_2 of its curvature. These eigenvalues are accessible from the eigenequations of the curvature tensor C_i as shown in Equation (5). In the equation for the unit matrix, the symbol I represents the unit matrix.

$$\det(C_i - \lambda I) = 0 \tag{5}$$

To calculate the curvature index K_i of a point p_i, λ_1 and λ_2 are used in the following manner.

$$K_i = \frac{\lambda_i}{\lambda_1 - \lambda_2} \tag{6}$$

By defining K_e and K_f thresholds, it is possible to filter the feature information contained in the point cloud. This filtering process includes both edge points and feature points.

$$f(k) = \begin{cases} Edge\ points & K_i > K_e \\ Feature\ points & K_f < K_i < K_e \\ Noise\ points & other \end{cases} \tag{7}$$

- *Feature match*

The matching of feature points utilizes the nearest neighbor algorithm. p_i^1 in the first frame point cloud represents the i^{th} feature point and p_j^2 in the second frame refers to the j^{th} feature point. The Euclidean distance serves as the distance metric between these points, and it is computed by measuring the distance between the two point cloud frames, as illustrated in Equation (8).

$$d_{i,j} = \left|p_i^1 - p_j^2\right| \tag{8}$$

where |·| denotes the Euclidean paradigm. Find the feature point p_i^1 that is closest to p_i^2 in the second frame of the point cloud.

$$k = \text{argmin} j d_{i,j} \tag{9}$$

If the square of p_k^2 is too distant from p_i^2, it will be removed. The ultimate matching of feature points obtained can be presented as a set $M = \{(i,k)|i = 1,\dots,n_1, k = 1,\dots,n_2\}$, where (i,k) indicates that the first frame point cloud's ith feature point has been successfully matched with the second frame point cloud's kth feature point.

- *Motion estimation*

After preprocessing the point cloud frames through feature extraction, dynamic objects are excluded, and static point features are simultaneously acquired. These characteristics facilitate interframe matching with ICP algorithm, leading to improved results by removing dynamic objects and higher efficacy for real-time applications.

Assume that frame point clouds P and Q are given, and let M be the set of feature points obtained by feature matching. The objective of the ICP algorithm is to transform point cloud P to Q in the same coordinate system. Define the transformation matrix T as shown in Equation (10), which transforms the point cloud P into the coordinate system.

$$P' = TP \tag{10}$$

P' – the transformed point cloud P.

The objective of the ICP algorithm is to minimize the distance between P' and Q by iteratively refining the transformation matrix T. The average distance between P' and Q serves as the basis for the ICP algorithm. The least squares method is utilized to solve T such that the average distance between P' and Q is minimized. Technical term abbreviations are explained upon their first usage. The equation used to calculate the transformation matrix T is as follows:

$$T = (Q_M^T P_M)^T (Q_M^T P_M)^{-1} \tag{11}$$

Q_M, P_M – the feature point sets after feature matching in point clouds Q and P, respectively;

$Q_M^T P_M$ – the correspondence between feature points;

$Q_M^T P_M$ – the covariance matrix between feature points.

The optimal rigid body transformation matrix T between point clouds P and Q is obtained by solving the least squares solution.

Back-end optimization

The graph optimization algorithm is a highly effective back-end optimization approach for the issue of creating precise point cloud maps utilizing LiDAR in agricultural settings. This approach involves modeling successive frames of LiDAR-obtained point clouds and incorporating landmark extraction from point cloud frames where localization fails, thereby constraining pose estimation. In the factor graph represented in Figure 5, nodes portray each frame replete with landmarks, while edges depict the constraint relationships between them. The nodes symbolize the state variables of the LiDAR point cloud frames and landmarks.

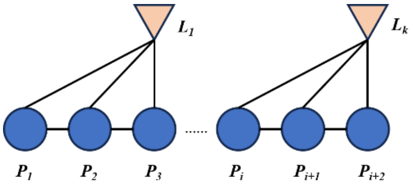

Fig. 5 - Factor diagram

The motion constraint relationship between neighboring frames and the previous frame of landmarks is established through the motion relationship. Equation (12) illustrates the motion constraint relation between neighboring frames, where $T_{i.j}$ represents the motion relation between frame *i* and frame *j* and T_k represents the motion relation between the signpost and frame *k*.

$$z_{i,j} = \begin{bmatrix} R_{i,j} & t_{i,j} \\ 0^T & 1 \end{bmatrix} = \begin{bmatrix} T_{i,j} & 0 \\ 0^T & 1 \end{bmatrix} \tag{12}$$

where the rotation and translation of the i^{th} frame with respect to the j^{th} frame are represented by $R_{k,i}$ and $t_{i,j}$, respectively. Additionally, the bit-position transformation matrix of the i^{th} frame with respect to the j^{th} frame is denoted as $T_{i,j}$. Similarly, the motion relationship between the road sign and the first three frames can be expressed in the same manner as shown in Equation (13), where $R_{k,i}$ and $t_{k,j}$ are the rotation and translation of the roadmap with respect to frame i, respectively, and T_k is the bit-pose transformation matrix of the roadmap.

$$z_{k,j} = \begin{bmatrix} R_{k,i} & t_{k,i} \\ 0^T & 1 \end{bmatrix} = \begin{bmatrix} T_k & 0 \\ 0^T & 1 \end{bmatrix} \tag{13}$$

With this factor graph, a function to be optimized is defined that represents the state of each node as a vector and optimizes the minimization error over all constraint relations. This optimization problem can be expressed as minimizing the objective function $f(x)$ as follow:

$$f(x) = \sum_{k=1}^{N} \sum_{i,j\in N_k} e_{ij}^k(x)^T \, \Omega_{ij}^k e_{ij}^k(x) + \sum_{m=1}^{M} \sum_{i,j\in N_m} e_{ij}^m(x)^T \Omega_{ij}^m e_{ij}^m(x) \tag{14}$$

N_k – the set of the k^{th} node and its adjacent nodes;

N_m – the set of the m^{th} signpost and its associated nodes;

$e_{ij}^k(x)$, $e_{ij}^m(x)$ – the error vectors between nodes *i* and *j*;

Ω_{ij}^k, Ω_{ij}^m – the corresponding information matrices.

Analysis of point cloud map construction algorithms

To assess the dependability of the point cloud mapping algorithm put forward in this paper, front-end position estimation comparison experiments and back-end optimization comparison experiments were conducted on field roads, paths, and crop rows. The algorithm's performance is evaluated using the reprojection error as the index, where a smaller error indicates a position estimation that is closer to the actual position and hence a higher accuracy.

The reprojection error is calculated using the following formula:

$$ereproj = \frac{1}{n}\sum_{i=1}^{n}\frac{|p_i'-p_i|_2}{\sqrt{w^2+\hbar^2}} \tag{15}$$

p_i – the location of the *ith* point in the actual world;
p_i' – the position of the reprojected point;
w, h – the dimensions of the image;
n – the count of points within the point cloud.

- *Analysis of point cloud map construction algorithms*

To evaluate the effectiveness of the Plant Point Rejection method in estimating motion, this experiment compares it with the ground extraction-based algorithm. The positioning accuracies of the motion estimation method based on ground extraction and the method based on plant point rejection are compared in three scenarios: a large field road, a small field road and a crop row. This is done by comparing the reprojection errors of both methods.

Figure 6 shows the accuracy control plot of the experiments. The plant point culling method (A) has a significantly reduced reprojection error compared to the ground extraction-based method (B) for the three experimental scenarios (1, 2, 3) by 10.27%, 20.76% and 14.36% respectively. This suggests that the method based on plant point culling is more precise and reliable in motion estimation.

The experiments simulated different pavement conditions such as flat pavement, raised pavement and pothole pavement. The results show that the vegetation point rejection-based method is better able to adapt to the variations of different pavement conditions with greater robustness and reliability than the ground extraction-based method. This is because the method is better able to remove non-ground points, including raised objects and low-lying areas with large height variations.

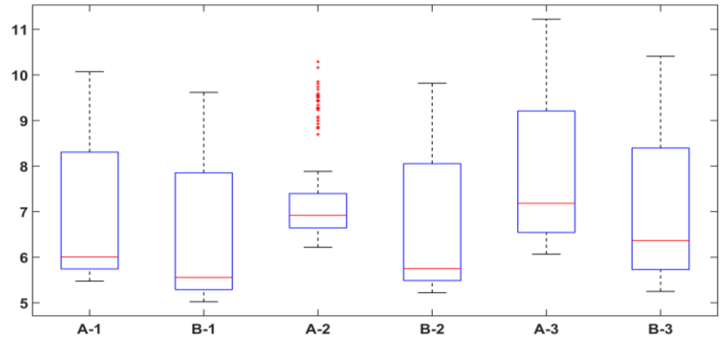

Fig. 6 - Comparison of motion estimation accuracy

- *Experiment of back-end optimization comparison*

To evaluate the effectiveness of the loanability estimation inspired back-end optimization algorithm, this experiment compares it with the distance and time based back-end optimization algorithm. The sensor data were collected in three scenarios: field road, field path and crop row, and the collected data were fed into the loanability estimation inspired back-end optimization algorithm and the distance and time based back-end optimization algorithm respectively for processing, and their accuracy and robustness were compared by comparing the reprojection errors of the two methods.

The experimental results showed that the reprojection error of the loanability estimation-inspired back-end optimization compared to the distance and time-inspired back-end optimization was reduced by 19.3%, 15.53% and 10.95% for three scenarios: large field roads, small field roads and crop rows. The results show that the heuristic back-end optimization algorithm based on localization estimation can significantly improve the localization accuracy and trajectory accuracy of the robot.

RESULTS

Example of Map Construction

To evaluate the performance of the proposed point cloud map construction approach, a test was conducted at the Xiaowangzhuang Agricultural Machinery Experimental Station of the China Agricultural Machinery Institute.

The point cloud map for the experimental field is shown in Figure 7(a). The green dots signify non-vegetative points, which include open areas, rocks on the land surface, and other comparable features. Meanwhile, blue dots indicate obstacles surrounding them, such as cornstalks or weeds. Figure 7(b) shows the performance of the point cloud map construction algorithm between crops rows in the experimental field. The algorithm has the function of localization estimation and can detect the failure of front-end pose tracking. Once tracking failure is detected, the localization estimation algorithm will be modified to improve the robustness and stability of the algorithm. Figure 7(c) illustrates an instance of point cloud map construction for field edges and field tracks. In this scenario, the luxuriant vegetation may lead to occlusion and increased noise in the point cloud data, thereby complicating the construction and localization of the point cloud map. Within the point cloud map, the geometric shape of the pathway and the distribution of surrounding vegetation are distinctly observable. The width and curvature of the pathway, as well as the potential presence of obstacles, are clearly discernible. Figure 7(d) illustrates a field pathways where point cloud mapping has been employed. The road is characterized by street trees situated on either side and a tarmac surface. The point cloud map provides clear visualization of the geometry and features of road, while also presenting the distribution and structure of the street trees as point clouds in space.

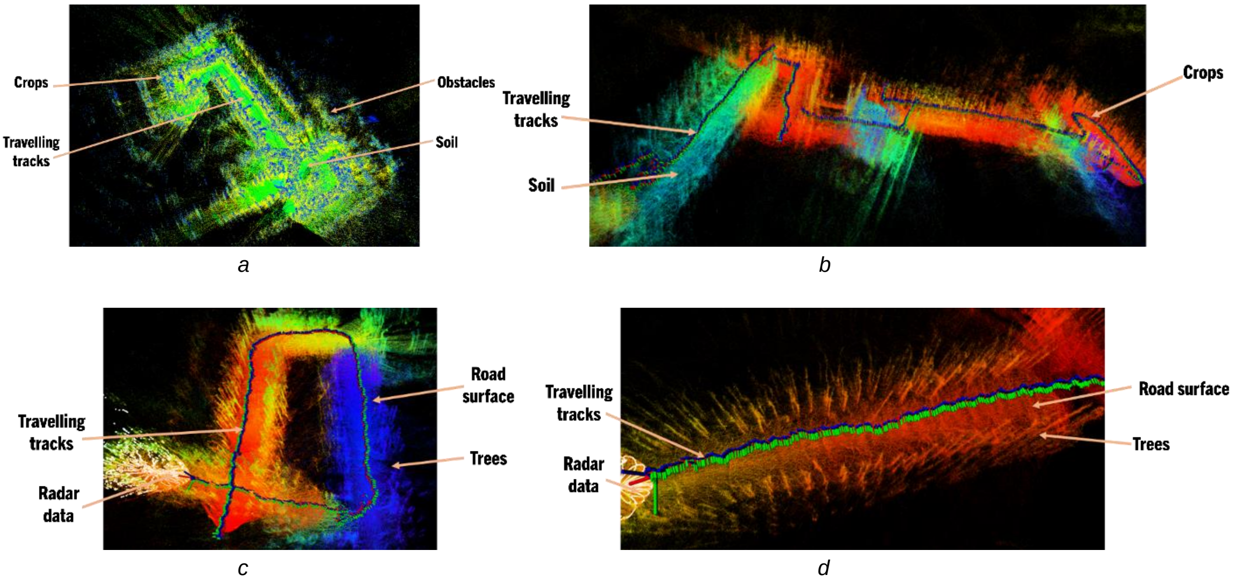

Fig. 7 - Example of point cloud map construction

a - example of point cloud mapping in experimental fields; b - example of point cloud mapping between crop row; c - example of point cloud mapping along field edges and field tracks; d - example of point cloud mapping on field pathway

Localization Accuracy Evaluation

In this experiment, the localization error in point cloud maps serves as the evaluation metric. The point cloud map localization error specifically signifies the distance difference between the algorithmically obtained localization result (P_{pos}) and the true position (P_{gt}) during the localization process. This difference is quantified using the Euclidean distance. Given the point cloud map localization result as P_{pos} and the true position as P_{gt}, the point cloud map localization error (E) is calculated as follows:

$$E_{pos} = \left\| P_{pos} - P_{gt} \right\| \tag{16}$$

Point cloud map localization accuracy refers to the statistical properties of localization errors, typically quantified using Root Mean Square Error (RMSE). RMSE represents the square root of the average of the squared values of all localization errors. Let n be the number of localization error samples, $E_{pos,i}$ and be the localization error of the i^{th} sample. The calculation formula for point cloud map localization accuracy is given:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(E_{pos,i}\right)^2} \tag{17}$$

In assessing the performance of the mapping algorithm in terms of rotation and translation, high-precision gyroscope data and Global Navigation Satellite System (GNSS) data were employed to obtain accurate rotation and translation information. The gyroscope recorded acceleration data, allowing for the derivation of rotation angles, which were then compared with the pose transformations calculated by the mapping algorithm to assess rotational accuracy. Positional information obtained through GNSS, and the translation vectors calculated by the mapping algorithm were compared to evaluate the algorithm's precision in translation. The data collection vehicle traversed along predefined paths, collecting three-dimensional point cloud data. Simultaneously, the gyroscope recorded the vehicle's rotational information, while GNSS captured the vehicle's positional data.

Table 1

Rotating positioning accuracy.

Localization Accuracy / °	Between Crop Rows	Field Edges	Field Tracks	Field Pathways
Roll Angle	0.429	0.405	0.284	0.261
Pitch Angle	0.249	0.250	0.206	0.175
Yaw Angle	0.179	0.155	0.149	0.151

The gyroscope was mounted at the central position of the vehicle, maintaining a fixed relative orientation to the vehicle body. In this gyroscope setup, the x-axis pointed in the vehicle's forward direction, the y-axis pointed to the right, and the z-axis pointed upward. Thus, the rotation angle of the vehicle's front around the x-axis was referred to as the Roll Angle (θ_x), rotation around the y-axis as the Pitch Angle (θ_y), and rotation around the z-axis as the Yaw Angle (θ_z). These angles were used to describe the vehicle's orientation and pose information. A detailed presentation of the positioning accuracy data is provided in Table 1.

In the experiments, it is observed that in densely populated areas such as field edges and between crop rows, the accuracy is relatively lower, yet it remains within an error margin of approximately 0.5°. This discrepancy can be attributed to increased interference in motion estimation as the density of plants rises. Despite the removal of a significant portion of plant points during initial data processing, the presence of dense vegetation introduces more occlusion. Consequently, the availability of feature points for accurate estimation diminishes, making feature matching more challenging. In contrast, relatively open areas such as field pathways and large field pathways exhibit higher positioning accuracy due to reduced interference in motion estimation caused by limited occlusion.

Following the validation of the algorithm's rotational accuracy, the subsequent step involves evaluating its translational precision. The GNSS positioning is employed as the ground truth in the experiment, with each scenario involving a 30-meter drive and error calculation performed every ten meters. The vehicle's position information is compared with GNSS positioning data, and translational errors are calculated. By comparing the differences between the vehicle's positioning and the actual GNSS positioning, the accuracy and precision of point cloud pose estimation are assessed. Detailed experimental results are presented in Table 2.

Table 2

Cumulative error.

Error (cm)	Orientation	10 m	20 m	30 m
Field Pathways	x	1.98	3.96	5.03
	y	1.95	2.83	4.85
Field Tracks	x	3.31	5.87	8.25
	y	1.93	3.85	6.87
Field Edges	x	3.84	7.73	11.45
	y	2.76	5.68	8.13
Between Crop Rows	x	4.15	7.53	11.68
	y	2.87	5.63	8.73

In the scenarios of field pathways and field tracks, relative to the more complex terrains of field edges and between crop rows, the topography is relatively flat and open. Additionally, there is comparatively less vegetation obstruction and reflection interference. Therefore, in these two scenarios, the mapping algorithm exhibits higher translational accuracy, with cumulative errors over a 30 m travel distance measuring 5.03 cm and 8.25 cm, respectively. In contrast, in the scenarios of field edges and between crop rows, the distribution density of plants is higher, and the height and morphology of vegetation are more complex. This complexity results in more noise and errors in the point cloud data detected by the lidar sensor. Cumulative errors over a 30 m distance in these scenarios are 11.45 cm and 11.68 cm, respectively. Despite the challenges posed by higher vegetation density and complexity, the cumulative errors in these scenarios remain within 12 cm.

CONCLUSIONS

(1) Considering the environmental characteristics of typical agricultural machinery operation scenes, a comprehensive plan for semantic point cloud map construction was devised. In hardware design, a data acquisition device incorporating Lidar, RGB, and NIR sensors was designed. Furthermore, a calibration board was created for calibrating both visible and non-visible light cameras, and an algorithmic workflow for multi-sensor joint intrinsic and extrinsic parameter calibration was established. In terms of algorithmic design, an advanced algorithm for high-precision point cloud map construction within the SLAM framework was proposed. Additionally, a Transformer-based semantic segmentation algorithm was introduced.

(2) The algorithm, built upon the SLAM framework, addresses the challenges posed by dynamic points interference in agricultural machinery operation scenes through a plant point removal preprocessing method. An optimization algorithm inspired by locatability estimation was proposed for the backend, addressing the limitations of time or distance-based heuristics. Experimental evaluations conducted in field pathways, small pathways, and between crop rows showcased the effectiveness of the optimized mapping algorithm, with the median reprojection error Euclidean distances reduced by 10.27%, 20.76%, and 14.36%, respectively.

(3) A high-precision map information collection vehicle was assembled, equipped with field information collection devices and deployed with mapping and semantic segmentation algorithms. Experimental trials conducted in field pathways, small pathways, field edges, and between crop rows demonstrated favorable results. At 30 meters, the average rotational localization errors were 0.429°, 0.405°, 0.284°, and 0.261°, with minimum cumulative errors ranging from 1.93 cm to 4.15 cm at 10 meters, 2.83 cm to 7.73 cm at 20 meters, and 4.85 cm to 11.68 cm at 30 meters.

ACKNOWLEDGEMENT

The work was sponsored by the National Key R&D Program Project of China (2021YFD2000105).

REFERENCES

[1] Ao, Z., Sun, Y., Xin, Q. (2021). Constructing 10-m NDVI Time Series from Landsat 8 and Sentinel 2 Images Using Convolutional Neural Networks. *IEEE Geoscience and Remote Sensing Letters*, Vol. 18, pp. 1461-1465, Guangzhou/China.

[2] Chen, S., Nardari, G., Lee, E., Chao, Q., Xu, L., Roseli, A., Vijay, Kumar. (2020). SLOAM: Semantic Lidar Odometry and Mapping for Forest Inventory. *IEEE Robotics and Automation Letters*, Vol. 5, pp. 612-619, United States.

[3] Dong, N., Chi, R., Du, Y., Wen, C., Zhang, Z. (2020). Research on field surface point cloud Registration based on rotary Surface profile (基于旋转曲面轮廓特征的农田地表点云配准研究). *Transactions of the Chinese Society for Agricultural Machinery*, Vol. 51, pp. 325-332, Beijing/China.

[4] Han, B., Luo, L., Liu, X., Shen, H. (2022). Feature Descriptor Enhancement for Loop Detection Based on Metric Learning (基于度量学习的回环检测描述子提升算法). *Pattern Recognition and Artificial Intelligence*, Vol. 35, pp. 51-61, Zhejiang/China.

[5] Han, C., Chen, M., Huang, Y., Zhao, M., Du, Q., Liang, Q. (2022). Laser SLAM loopback detection based on global feature descriptor (基于全局特征描述子的激光 SLAM 回环检测方法). *Journal of Shanghai Jiao Tong University*, Vol. 10, pp. 1379-1387, Shanghai/China.

[6] Ji, Y., Xu, H., Zhang, M, Li, S., Cao, R., Li, H. (2019). Design of farmland environment point cloud acquisition system based on laser radar (基于激光雷达的农田环境点云采集系统设计). *Transactions of the Chinese Society for Agricultural Machinery*, Vol. 50, pp. 1, Beijing/China.

[7] Johnson, A. E., Hebert, M. (1999). Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE transactions on pattern analysis and machine intelligence*, Vol. 21, pp. 433-449, USA.

[8] Kim, L., Wu, Z. (2022). Practice path of Promoting Agricultural Modernization with Agricultural Science and technology Modernization (以农业科技现代化促进农业现代化的实践路径). *Journal of Drainage and Irrigation Machinery Engineering*, Vol. 40(10), pp. 1056-1064, Jiangsu/China.

[9] Li, X., Wei, P., He J., Li, M., Zhang, M., Wen, B. (2021). Field plant point cloud registration method based on Kinect V3 depth sensor (基于 Kinect V3 深度传感器的田间植株点云配准方法). *Transactions of the Chinese Society of Agricultural Engineering*, Vol. 37, pp. 45-52, Guangxi/China.

[10] Li, X., He, W., Zhu, S., Li, Y., Xie, T. (2021). A review of synchronous localization and map construction methods based on environmental semantic information (基于环境语义信息的同步定位与地图构建方法综述). *Chinese Journal of Engineering Science*, Vol. 43, pp. 754-767, Zhejiang/China.

[11] Lucas, C., Bouten, W., Koma, Z., Kissling, W., Seijmonsbergen, A. (2019). Identification of Linear Vegetation Elements in a Rural Landscape Using LiDAR Point Clouds. *Remote Sensing*, Vol. 11, pp. 292, Netherlands.

[12] Newcombe, R. A., Lovegrove, S. J., Davison, A. J. (2011). DTAM: Dense tracking and mapping in real-time. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2320-2327, United Kingdom.

[13] Qi, F., Zhou, X., Wu, Z., Zhang, X. (2017). Path and method of infrastructure engineering in the process of agricultural modernization (农业现代化过程中基础设施工程化路径与方法). *Transactions of the Chinese Society of Agricultural Engineering*, Vol.33, pp. 16-25, Beijing/China.

[14] Tian, X., Zhang, M., Yang, C., Ma, J. (2021). FusionNDVI: A Computational Fusion Approach for High-Resolution Normalized Difference Vegetation Index [J]. *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 59, pp. 5258-5271, Hubei/China.

[15] Tombari, F., Salti, S., Di, S. L. (2010). Unique Signatures of Histograms for Local Surface Description. *European Conference on Computer Vision*, Vol. 11, pp. 356-369, Italian.

[16] Vizzo, I., Guadagnino, T., Mersch, B., Wiesmann, L., Behley, J., Stachniss, C. (2023). KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way[J]. *IEEE Robotics and automation letters*, Vol. 8, pp. 1029-1036, Germany.

[17] Wang, X., Yuan, S., Jia, W. (2022). Current situation and development of agricultural mechanization in hilly and mountainous areas (丘陵山区农业机械化现状与发展). *Journal of Drainage and Irrigation Machinery Engineering*, Vol. 5, pp. 535-540, Jiangsu/China.

[18] Wang, Y., Lou, Y., Zhang, Y., Song, W., Huang, F., Tu, Z. (2021). A Robust Framework for Simultaneous Localization and Mapping with Multiple Non-Repetitive Scanning Lidars. *Remote Sensing*, Vol. 13, pp. 2015, Hubei/China.

[19] Xu, X., Li, N., Yao, Y. (2022) Laser Radar SLAM Algorithm Based on fast loopback detection in Outdoor environment (基于深度学习的室内动态场景下的 VSLAM 方法). *Journal of Chinese Inertial Technology*, Vol. 30 pp. 716-722, Jiangsu/China.

[20] Zeng, Q., Luo, Y., Sun, K., Li, Y., Liu, J. (2022). Review on the development of SLAM technology based on vision and its integration with inertia (视觉及其融合惯性的 SLAM 技术发展综述). *Journal of Nanjing University of Aeronautics and Astronautics*, Vol. 54, pp. 1007-1020, Jiangsu/China.