

Article

Phenotypic Trait Acquisition Method for Tomato Plants Based on RGB-D SLAM

Penggang Wang ^{1,2,3}, Yuejun He ^{1,2,3,*}, Jiguang Zhang ^{4,*}, Jiandong Liu ^{1,2,3}, Ran Chen ^{1,2,3} and Xiang Zhuang ^{1,2,3}

¹ North China Institute of Aerospace Engineering, Langfang 065000, China; wangpg@stumail.nciae.edu.cn (P.W.); liujd@stumail.nciae.edu.cn (J.L.); cran199@stumail.nciae.edu.cn (R.C.); zx1033676206@stumail.nciae.edu.cn (X.Z.)

² National & Regional Joint Engineering Research Center for Aerospace Remote Sensing Application Technology, Langfang 065000, China

³ Aerospace Remote Sensing Information Processing and Application Collaborative Innovation Center of Hebei Province, Langfang 065000, China

⁴ Institute of Automation of Chinese Academy of Sciences, Beijing 100090, China

* Correspondence: yuejunhe1981@nciae.edu.cn (Y.H.); jiguang.zhang@ia.ac.cn (J.Z.)

Abstract

The acquisition of plant phenotypic traits is essential for selecting superior varieties, improving crop yield, and supporting precision agriculture and agricultural decision-making. Therefore, it plays a significant role in modern agriculture and plant science research. Traditional manual measurements of phenotypic traits are labor-intensive and inefficient. In contrast, combining 3D reconstruction technologies with autonomous vehicles enables more intuitive and efficient trait acquisition. This study proposes a 3D semantic reconstruction system based on an improved ORB-SLAM3 framework, which is mounted on an unmanned vehicle to acquire phenotypic traits in tomato cultivation scenarios. The vehicle is also equipped with the A* algorithm for autonomous navigation. To enhance the semantic representation of the point cloud map, we integrate the BiSeNetV2 network into the ORB-SLAM3 system as a semantic segmentation module. Furthermore, a two-stage filtering strategy is employed to remove outliers and improve the map accuracy, and OctoMap is adopted to store the point cloud data, significantly reducing the memory consumption. A spherical fitting method is applied to estimate the number of tomato fruits. The experimental results demonstrate that BiSeNetV2 achieves a mean intersection over union (*mIoU*) of 95.37% and a frame rate of 61.98 FPS on the tomato dataset, enabling real-time segmentation. The use of OctoMap reduces the memory consumption by an average of 96.70%. The relative errors when predicting the plant height, canopy width, and volume are 3.86%, 14.34%, and 27.14%, respectively, while the errors concerning the fruit count and fruit volume are 14.36% and 14.25%. Localization experiments on a field dataset show that the proposed system achieves a mean absolute trajectory error (mATE) of 0.16 m and a root mean square error (RMSE) of 0.21 m, indicating high localization accuracy. Therefore, the proposed system can accurately acquire the phenotypic traits of tomato plants, providing data support for precision agriculture and agricultural decision-making.

Keywords: ORB-SLAM3; plant phenotypic traits; semantic segmentation; tomato plant; unmanned vehicle

Academic Editor: Francesco Marinello

Received: 23 June 2025

Revised: 15 July 2025

Accepted: 21 July 2025

Published: 22 July 2025

Citation: Wang, P.; He, Y.; Zhang, J.;

Liu, J.; Chen, R.; Zhuang, X.

Phenotypic Trait Acquisition

Method for Tomato Plants Based on

RGB-D SLAM. *Agriculture* **2025**, *15*,

1574. [https://doi.org/10.3390/](https://doi.org/10.3390/agriculture15151574)

[agriculture15151574](https://doi.org/10.3390/agriculture15151574)

Copyright: © 2025 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the continuous expansion of modern agricultural production, it has become increasingly difficult and costly to rely solely on manual labor for tasks such as pest and disease monitoring, phenotypic trait acquisition, and yield estimation. As a result, integrating artificial intelligence technologies into various aspects of agricultural production [1–3] has emerged as a key research focus in the field of smart agriculture. Three-dimensional reconstruction of crops enables a comprehensive understanding of plants' phenotypic traits, which reflect the current growth status and cultivation conditions of the plants [4]. These phenotypic characteristics facilitate the extraction of various plant parameters [5–7] and support automated monitoring.

With the advancement of smart agriculture, numerous studies have been conducted on three-dimensional (3D) reconstruction methods for crops. Traditional 3D reconstruction techniques include multi-view stereo (MVS) [8], structure from motion (SfM) [9], and time-of-flight (ToF) [10]. Both MVS and SfM reconstruct 3D structures from two-dimensional images captured from multiple viewpoints. Bao et al. [11] employed MVS to reconstruct 3D models of sorghum plants and successfully extracted their phenotypic traits. Van et al. [12] combined MVS with the A* algorithm on an autonomous robot to enable automatic cucumber harvesting. Zhu et al. [13] applied SfM to reconstruct 3D models of rice and maize and further achieved high-fidelity visualization using clustering and fitting methods. Liu et al. [14] proposed a combined SfM-MVS approach for measuring the morphological parameters of potato tubers. ToF technology is typically integrated into scanning devices such as LiDAR and depth cameras. Shi et al. [15] utilized LiDAR for 3D reconstruction of maize and achieved real-time phenotypic trait measurement. Marinello et al. [16] used a Kinect depth camera to estimate the volume and mass of grape bunches. Thapa et al. [17] employed an LMS 511 LiDAR scanner to acquire phenotypic parameters such as the leaf area and leaf inclination angles for maize and sorghum. However, traditional 3D reconstruction methods are often sensitive to the lighting conditions or dependent on expensive scanning equipment, making them challenging to apply in outdoor environments. Over the past decade, visual simultaneous localization and mapping (vSLAM) has undergone rapid development. Owing to its low implementation cost and reliable reconstruction performance in outdoor settings, it has gradually become a mainstream approach for 3D plant reconstruction in smart agriculture.

Numerous studies have employed visual SLAM to acquire the phenotypic trait parameters of plants and guide agricultural development. Fan et al. [18] proposed a portable RGB-D SLAM system to reconstruct large-scale forests, obtaining tree location and height information. Dong et al. [19] utilized a depth camera combined with the ORB-SLAM2 system to capture and track the point cloud data of fruits in orchards. Pierzchala et al. [20] integrated multiple sensors, including stereo cameras and LiDAR, to map forest environments, providing precise tree location data for forestry vehicles. Kim et al. [21] developed a framework named P-AgSLAM to extract distinctive morphological features of maize in cornfields. Lobefaro et al. [22] achieved high-precision spatiotemporal matching and data association in pepper fields by combining RGB-D SLAM with visual place recognition. Nellithimaru et al. [23] integrated deep learning with visual SLAM and used stereo cameras to capture regions of interest, enabling stable grape counting in vineyards. Fan et al. [24] reconstructed maize crops using RGB-D cameras and estimated the maize stalk diameters by analyzing the convexities and projections in the point clouds. Wang et al. [25] improved ORB-SLAM2 and combined it with the YOLOv4 and DeepSORT algorithms to localize and count melons. Islam et al. [26] proposed the AGRI-SLAM system, which integrates image enhancement techniques to enable real-time environmental reconstruction under a wider range of agricultural conditions, including low-light scenarios. Tang et al. [27] employed a multi-sensor fusion SLAM system to conduct fruit tree localization

experiments in pear and persimmon orchards, achieving high localization accuracy and constructing global maps of the orchards through 3D reconstruction. In previous studies, SLAM has been utilized to construct 3D maps for acquiring crop phenotypic traits. However, these maps typically represent only geometric features and exhibit limited expressiveness for regions of interest. By incorporating semantic information into the map, the representation of such regions can be enhanced, thereby improving both the expressiveness and the usability of the 3D map.

Integrating semantic or instance segmentation into SLAM to construct semantic point cloud maps that emphasize the phenotypic trait information of crops has become a research hotspot in agriculture. Xiong et al. [28] incorporated the semantic segmentation network BiSeNetV1 into VINS-RGBD to achieve 3D semantic reconstruction of citrus trees and estimated the fruit yield through conditional filtering and spherical fitting. Liu et al. [29] proposed a fruit counting framework using Faster R-CNN for fruit detection followed by semantic SfM for reconstruction and counting. Yuan et al. [30] combined the PP-LiteSeg-T network with VINS-RGBD to map strawberry fields and count strawberry fruits. Giang et al. [31] employed ORB-SLAM3 alongside a semantic segmentation network to generate semantic point cloud maps of sweet peppers, facilitating the detection of the lowest branch pruning points. Liu et al. [32] integrated ORB-SLAM3 with semantic segmentation to classify mature strawberries, guiding harvesting activities. Xie et al. [33] proposed a multi-object semantic segmentation network, MM-ED, combined with ORB-SLAM3 to build semantic point cloud maps of unstructured lawns, providing a reference and guidance for weeding operations. Pan et al. [34] designed a novel semantic mapping framework that employs the 3D-ODN detection network to extract object information from point clouds. The experimental results demonstrated that the system enables high-precision understanding of agricultural environments. Although these studies enrich point cloud representation by incorporating semantic segmentation and highlighting regions of interest, the data acquisition is performed manually, which is highly challenging in large-scale agricultural plantations. The use of embedded devices such as unmanned aerial vehicles (UAVs) or unmanned ground vehicles (UGVs) to assist with data collection can alleviate labor demands and reduce long-term operational costs.

Integrating unmanned aerial vehicles (UAVs) or unmanned ground vehicles (UGVs) enables more flexible 3D reconstruction of crops, thereby improving the efficiency of plant phenotypic trait acquisition. The ultimate goal of smart agriculture is to combine intelligent equipment to support automated agricultural production. UAVs can rapidly acquire phenotypic trait data for small-scale fields, such as the plant height [35] or canopy coverage [36]. Sun et al. [37] utilized UAVs for 3D reconstruction of apple trees to obtain parameters including the tree height and canopy volume. Sarron et al. [38] performed 3D reconstruction of mango orchard trees using UAVs, achieving accurate mango yield estimation. UGVs are more suitable for detailed 3D reconstruction of shorter plants and provide finer mapping results. Siebers et al. [39] employed a UGV combined with LiDAR to generate precise 3D point clouds of grapevine rows for rapid phenotypic information extraction and conducted preliminary assessment of the grape nutritional status. Qiu et al. [40] designed a LiDAR-equipped mobile robot to reconstruct maize plants and measured phenotypic parameters such as the plant height and row spacing. Wang et al. [41] developed a 3D reconstruction robot for greenhouse environments and validated the system in an eggplant cultivation base, successfully predicting the fruit counts. Nguyen et al. [42] developed DairyBioBot, a fully automated phenotyping platform equipped with LiDAR and RTK sensors, to reconstruct perennial ryegrass and estimate its field biomass. Qi et al. [43] proposed a CNN-based loop closure detection method tailored for intelligent embedded agricultural devices and suitable for RGB-D SLAM. The system was validated in

greenhouse environments and demonstrated high accuracy and real-time performance in loop detection.

Ma et al. [44] employed an improved YOLOv5 model to detect tree trunks and pedestrians, integrating it into a visual SLAM system to assist intelligent agricultural equipment in achieving more precise localization. These studies primarily focus on embedded devices for crop 3D reconstruction, but most rely on high-cost sensors such as LiDAR and are conducted offline. Therefore, employing lightweight visual SLAM for crop 3D modeling can reduce the investment costs while enabling real-time acquisition of phenotypic traits and growth status.

To address the challenges encountered in the application of 3D modeling technologies in smart agriculture, this study proposes an ORB-SLAM3-based system integrated with the lightweight semantic segmentation network BiSeNetV2. The system aims to overcome the lack of semantic information in point cloud maps and the limited real-time performance of existing approaches. A two-stage filtering method is employed to remove noise from the semantic point cloud, and OctoMap is used for efficient storage, thereby addressing the issue of large-scale point cloud data management. The entire system is deployed on an unmanned ground vehicle equipped with the A* algorithm and validated in a tomato plantation. The experimental platform demonstrates the potential to significantly reduce labor requirements. The main contributions of this study are as follows:

1. Develop a 3D reconstruction robotic platform tailored for greenhouse environments, enabling accurate extraction of crop phenotypic traits from 3D maps. The platform was validated in a tomato plantation.
2. Design a reconstruction system optimized for low-profile crops, integrating a lightweight semantic segmentation module to enhance the representation of regions of interest, while maintaining real-time performance by segmenting only keyframes.
3. Propose a two-stage filtering method to enhance the point cloud accuracy and adopt OctoMap for efficient storage on embedded devices.
4. A tomato fruit semantic segmentation dataset with precise annotations is constructed, providing valuable data support for future research on tomato plants.

2. Materials and Methods

2.1. Study Area and Data Acquisition

In this study, a tomato cultivation base located in Gu'an County, Langfang City, was selected as the study area for data acquisition. Gu'an County is situated in the central part of the North China Plain (39°26'–39°45' N, 116°09'–116°32' E) and has a semi-humid continental monsoon climate, characterized by four distinct seasons, abundant sunlight, and rich thermal resources. The average annual temperature is approximately 12.1 °C, and the annual precipitation ranges from 500 to 600 mm, providing favorable conditions for agricultural development. In recent years, Gu'an County has actively promoted modern agriculture, with a continuously expanding scale of protected vegetable cultivation. As one of the major crops, tomatoes are widely grown with diverse management practices, making the region a representative sample area for studying crop phenotypic traits.

On 18 April 2025, image data of tomato plants were collected at the tomato cultivation base in Gu'an County using an Intel RealSense D435i depth camera (Intel, Bangkok, Thailand) and a Nikon Z30 camera (Nikon, Tokyo, Japan) equipped with a 50–250 mm f/4.5–6.3 lens. The image acquisition covered tomato plants and fruits of varying maturity levels and under varying lighting conditions. The high-resolution images clearly captured the morphological structures of the plants, providing a reliable data foundation for subsequent tasks such as semantic segmentation, feature extraction, and 3D reconstruction of tomato plants.

Images captured by the D435i and Nikon Z30 cameras were jointly used to train the semantic segmentation network. In the self-constructed tomato dataset, the D435i was used to acquire wide-view images of tomato plants, while the Nikon Z30 was used to capture close-up images. A total of 300 wide-view and 300 close-up images were collected. All the images were uniformly resized and cropped to a resolution of 640×480 pixels and annotated using the Labelme software (version 5.4.1). The final dataset consisted of 600 annotated images. Sample images from the dataset are shown in Figure 1.

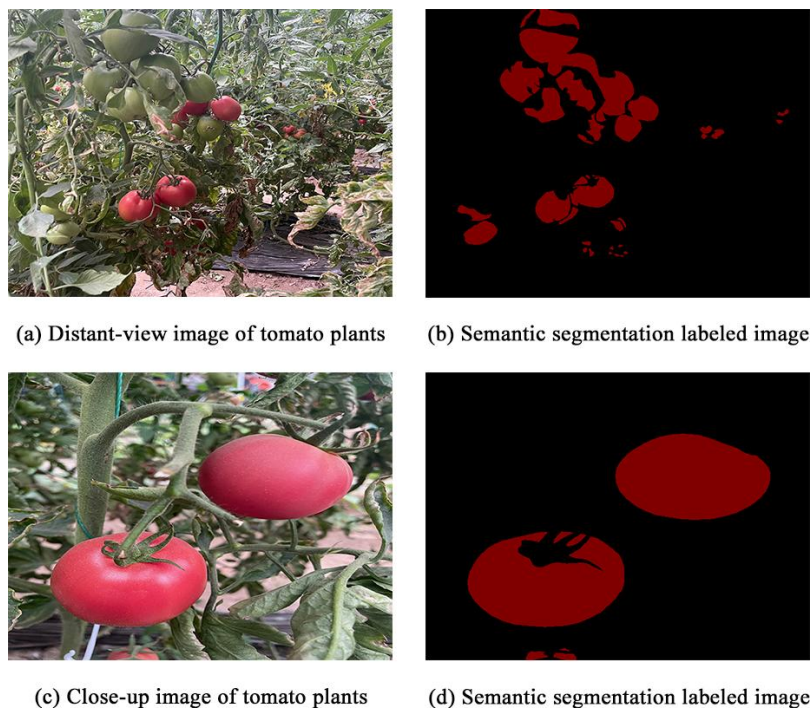


Figure 1. Examples of dataset images and annotations: (a) represents a wide-view image captured by the D435i, and (c) represents a close-up image captured by the Nikon Z30. Moreover, (b,d) show the corresponding semantic annotation maps for (a,c), respectively.

The system was tested in real-world scenarios using a UGV platform, as shown in Figure 2a, while Figure 2b presents the corresponding visualization results during the experiment. The UGV platform consists of a sensor module, a chassis module, and a control module. The sensor module includes an Intel RealSense D435i depth camera, a 2D LiDAR, and an RTK positioning system. The chassis module supports keyboard and joystick control, with a maximum payload capacity of 50 kg and a maximum operating speed of 0.8 m/s. The control module is PC-based, running Ubuntu 20.04 and ROS. Equipped with the A* algorithm, the UGV performs autonomous navigation in the field to collect data and conduct 3D reconstruction. During the experiment, RTK was used to obtain the ground-truth trajectory of the UGV; the Intel RealSense D435i was used to capture RGB and depth images; a monitor was used to visualize the experimental results; and a LiDAR sensor was employed for obstacle avoidance and path planning.

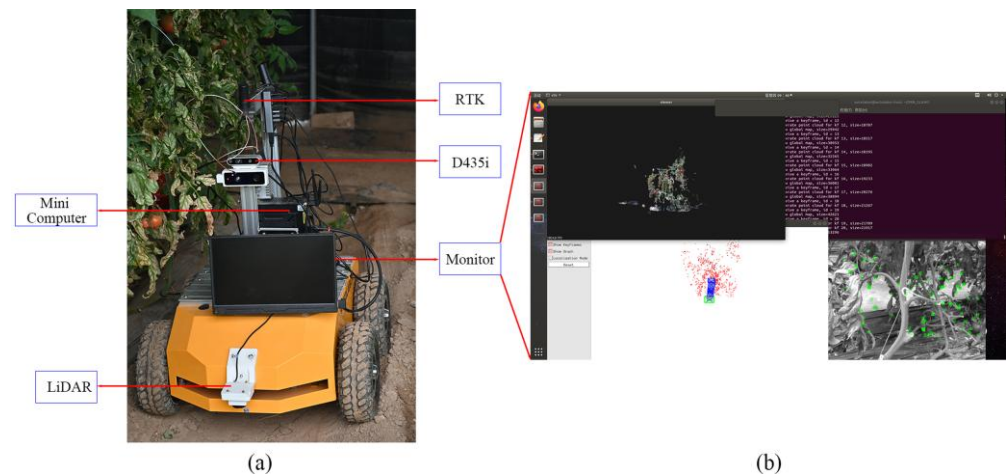


Figure 2. UGV data acquisition and experimental platform. (a) Hardware platform; (b) Example of runtime interface results.

2.2. Improved ORB-SLAM3 System

This study performs 3D semantic reconstruction of tomato plants based on an improved ORB-SLAM3 system. Compared to ORB-SLAM2, ORB-SLAM3 incorporates inertial measurement unit (IMU) input to assist in camera pose estimation and introduces a multi-map system, resulting in improved reconstruction quality and localization accuracy. Based on the practical application scenario, we further enhanced ORB-SLAM3. The specific workflow is illustrated in Figure 3.

Since the 3D reconstruction results of ORB-SLAM3 consider only geometric information and lack semantic expressiveness, we integrated a semantic module into the system to enhance the representation of the point cloud map. Considering that the system runs on embedded devices, measures are necessary to reduce the computational and storage burdens. As the majority of the computational load originates from the semantic segmentation module's inference, we selected the lightweight semantic segmentation network BiSeNetV2 and performed segmentation only on keyframe images. For the point cloud map storage, OctoMap was employed to significantly reduce the storage requirements, making it suitable for large-scale agricultural 3D reconstruction.

It is worth noting that this study adopted a “segment-then-reconstruct” approach to generate semantic point clouds. This strategy is based on the maturity of current image semantic segmentation techniques, which can provide more detailed and accurate semantic information at the image level. By generating point clouds from segmented semantic images, the complexity of the subsequent point cloud processing is effectively reduced, thereby improving the computational efficiency. Moreover, this module is easier to integrate in practical applications, demonstrating favorable engineering feasibility.

The system first performs tracking and pose estimation on each input frame and determines whether the frame qualifies as a keyframe based on the motion information. The keyframe selection is guided by the pose change and tracking quality between the current frame and the reference keyframe. Specifically, a frame is marked as a keyframe when the translation or rotation relative to the previous keyframe exceeds the predefined thresholds, or when the number of successfully tracked map points drops significantly. For each selected keyframe, semantic segmentation is performed, and the pixel-level semantic labels are projected into the 3D space using the corresponding depth map, pose information, and camera intrinsics to generate a 3D semantic point cloud. The resulting semantic point cloud is then processed through a two-stage filtering pipeline to remove noise and redundant information, and it is subsequently integrated into an OctoMap to construct the 3D semantic map. The system retains the local optimization and loop closure modules of

ORB-SLAM3 to ensure the global consistency and accuracy of the reconstructed map. The detailed workflow of the improved ORB-SLAM3 system is presented in Section 1 of Appendix A.

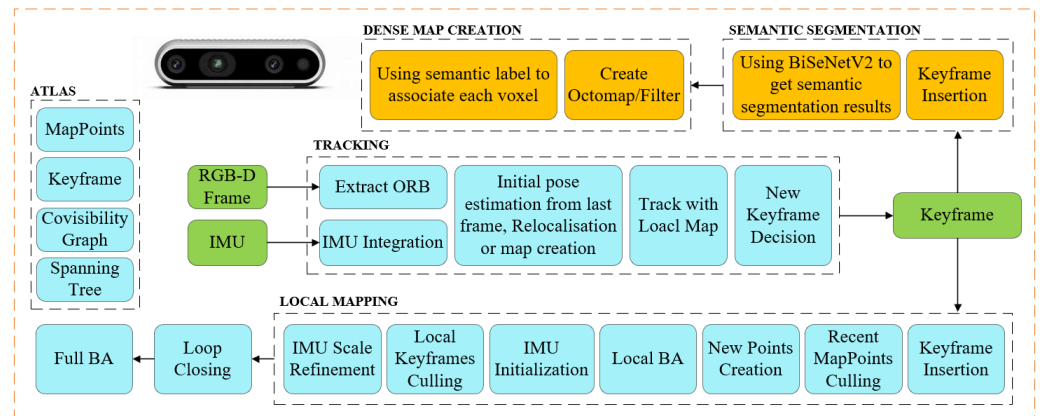


Figure 3. Workflow diagram of the improved ORB-SLAM3 system. The blue components represent the original modules of the ORB-SLAM3 system, the orange components indicate the modules we added or modified, and the green components denote key data generated during system operation.

2.3. Semantic Segmentation Module

In this study, we employed the lightweight bilateral segmentation network BiSeNetV2 for the pixel-level semantic segmentation of tomato plant images. The network architecture is illustrated in Figure 4. BiSeNetV2 is particularly well-suited for real-time 3D semantic reconstruction of low-height crops in agricultural scenarios due to its efficient network design and excellent segmentation performance.

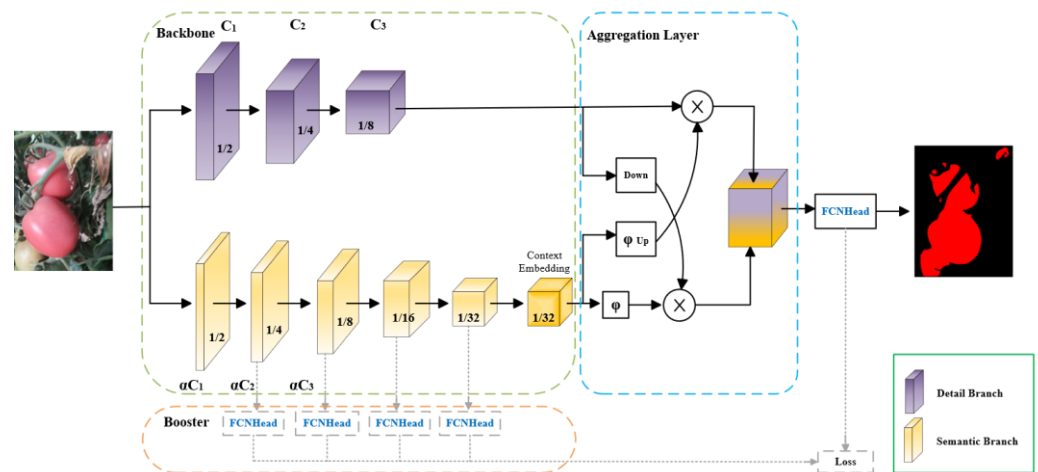


Figure 4. Workflow diagram of the BiSeNetV2 semantic segmentation network.

The core architecture of the BiSeNetV2 network consists of three main components: the bilateral segmentation network, the guided aggregation layer, and an enhanced training strategy. This design allows the network to maintain a high inference speed while meeting the accuracy requirements of semantic segmentation. As shown in Figure 4, the green box represents the bilateral segmentation network, which is composed of a detail branch (highlighted in purple) and a semantic branch (highlighted in yellow). The detail branch adopts a wide and shallow structure to effectively extract high-resolution boundary and texture information, which is beneficial for delineating structurally complex regions such as the leaves and fruits of tomato plants. In contrast, the semantic branch

employs a deep and narrow architecture, incorporating depthwise separable convolutions, fast downsampling, and global average pooling to capture rich contextual semantics. This enhances the network's ability to interpret complex scenes involving background clutter and occlusion. The dual-branch structure enables the complementary fusion of semantic and geometric features, resulting in robust segmentation performance even under conditions of dense crop distribution or leaf occlusion.

The blue box represents the guided aggregation layer, which employs a bidirectional attention mechanism to efficiently fuse the detail and semantic features extracted by the bilateral segmentation network. Compared with traditional operations such as concatenation or element-wise addition, this module minimizes the information loss during the fusion process. As a result, it offers improved discriminative capability when dealing with challenges such as blurred plant edges and leaf occlusions in tomato plants.

BiSeNetV2 also incorporates an enhanced training strategy, indicated by the orange box in the figure. This strategy embeds lightweight FCNHead auxiliary heads at different levels of the semantic branch to guide the network in learning multi-scale features more effectively. During training, these auxiliary heads help improve the convergence speed and robustness of the network. However, they are completely removed during inference, thus introducing no additional computational overhead into the semantic segmentation process and ensuring the overall efficiency of the system.

The semantic segmentation network integrated into the SLAM system performs segmentation only on selected keyframes to avoid compromising the system's real-time performance. BiSeNetV2 accurately and efficiently segments the keyframe images, and the semantic segmentation results are used to assign semantic labels to the point clouds generated during mapping, thereby constructing a structurally clear and semantically explicit 3D semantic point cloud map. Its excellent inference speed and segmentation accuracy ensure the real-time capability and precision of the semantic map construction, facilitating subsequent tasks such as fruit identification and region extraction in tomato plants.

2.4. Two-Stage Filtering Method

Abnormal points may appear in the 3D reconstruction map due to factors such as lighting variations and dynamic objects. The presence of these outliers degrades the quality of the final point cloud map and increases the errors in estimating the phenotypic shape of tomato plants from the point cloud data. Therefore, to enable the point cloud map to more accurately represent tomato plants, we applied a two-stage filtering process on the tomato point cloud, consisting of pass-through filtering followed by statistical filtering.

In the first stage, pass-through filtering is applied to the point cloud of tomato plants along the vertical direction to remove a large number of non-plant points in the background. This process can be expressed by Equation (1), where x , y , and z represent the coordinates of points in the point cloud, and (X_{min}, X_{max}) , (Y_{min}, Y_{max}) and (Z_{min}, Z_{max}) denote the filtering ranges along the three coordinate axes, respectively.

$$\begin{cases} X_{min} < x < X_{max} \\ Y_{min} < y < Y_{max} \\ Z_{min} < z < Z_{max} \end{cases} \quad (1)$$

Since pass-through filtering only provides an approximate range of valid tomato plant points, a secondary filtering step is necessary to improve the accuracy of the local point cloud. In the second stage, statistical filtering is applied to the remaining point cloud. Statistical filtering removes outliers by analyzing the distances between neighboring points within each point's local neighborhood. Specifically, for each point, $P_i(x_i, y_i, z_i) (i = 1, 2, \dots, n)$, the average distance d_i to its k nearest neighbors, $P_j(x_j, y_j, z_j) (j = 1, 2, \dots, k)$, k is calculated using Equation (2). Then, the mean μ and

standard deviation δ of these average distances are computed, assuming they follow a Gaussian distribution. The valid range of the point cloud distances is defined as $[\mu - k\delta, \mu + k\delta]$. For each point P_i , if its average distance d_i lies within this interval, it is retained; otherwise, it is removed. By tuning the parameters of the statistical filter, a balance between mapping accuracy and filtering speed for tomato plant reconstruction is achieved.

$$d_i = \frac{1}{k} \sum_{j=1}^k \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}, i \in (1, 2, \dots, n) \quad (2)$$

2.5. OctoMap Construction

In this study, the map constructed by ORB-SLAM3 is a dense point cloud, which requires substantial storage space. This poses a challenge for embedded devices with limited storage capacity, making it difficult to store large-scale 3D reconstruction results in agricultural scenarios. To address this issue, we employed OctoMap to convert and store the dense point cloud. Maps constructed with OctoMap offer flexible storage and can be incrementally updated, making them well-suited for large-scale agricultural applications.

The principle of representing dense point clouds using OctoMap is illustrated in Figure 5. OctoMap models the 3D space as a cube, which can be recursively subdivided into eight smaller cubes of equal size. This process begins from a root node, where each node can expand into eight child nodes, and the subdivision continues iteratively. Based on this principle, the entire dense point cloud is progressively partitioned so that each node represents a smaller spatial region, until a predefined resolution is reached. By adjusting the threshold parameters, OctoMaps of different resolutions can be generated to meet varying precision requirements. Section 2 in Appendix A describes the algorithmic process of OctoMap.

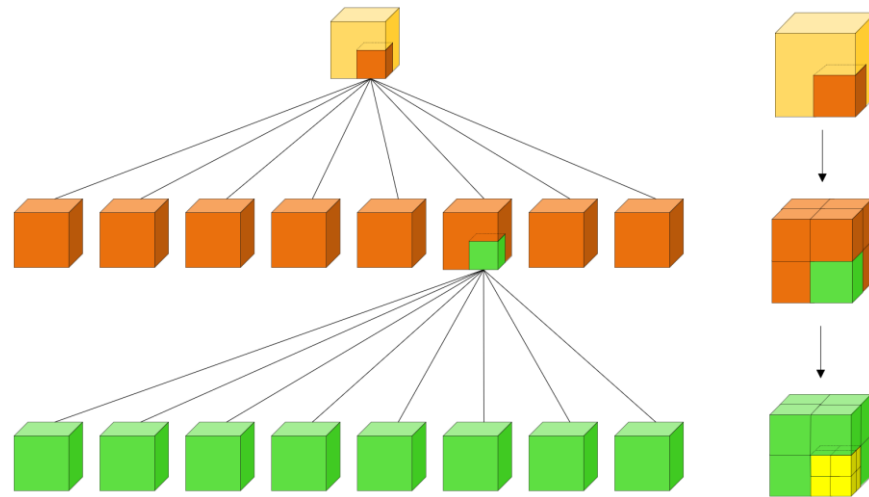


Figure 5. Schematic diagram of the OctoMap principle.

In the final OctoMap, each node contains information indicating whether the corresponding space is occupied. The occupancy status of a node is represented using a log-odds formulation, as defined in Equation (3), where α denotes the log-odds and p represents the probability of occupancy. By rearranging Equation (3), we obtain Equation (4), which shows that as α varies within the range $(-\infty, +\infty)$, the corresponding probability p lies within the interval $[0, 1]$.

$$\alpha = \text{logit}(p) = \log\left(\frac{p}{1-p}\right) \quad (3)$$

$$p = \text{logit}^{-1}(\alpha) = \frac{1}{1 + \exp(-\alpha)} \quad (4)$$

When all eight child nodes within a block are consistently marked as either occupied or free, the root node corresponding to that block will cease further subdivision. When the α value of a node exceeds the defined threshold, the node is considered occupied and can be visualized, thereby being regarded as a stable node.

In this study, the improved ORB-SLAM3 system integrates with OctoMap through the mapping thread, which outputs dense semantic point clouds. When the mapping thread processes keyframes and generates the corresponding dense semantic point clouds, these point clouds are immediately passed to the OctoMap module for spatial voxelization and incremental map updates. Compared to directly storing point clouds, this representation significantly reduces the memory consumption, improves the processing efficiency, and enhances the system's deployability on resource-constrained platforms. As a result, it increases the feasibility of performing large-scale 3D semantic reconstruction in agricultural environments.

3. Results and Discussion

3.1. Semantic Segmentation Experiments on Tomato Plants

The performance of semantic segmentation directly affects the accuracy of constructing the semantic point cloud map. Therefore, it is necessary to evaluate the performance of the BiSeNetV2 network. We selected several classic semantic segmentation networks—ICNet [45], PSPNet [46], SegFormer [47], DeepLabv3+ [48], and BiSeNetV1 [49]—to compare their segmentation performance with BiSeNetV2. The training dataset consisted of 600 uniformly cropped images of tomato plants, with annotations only on the tomato fruits. During training, the dataset was split into training, validation, and test sets at a ratio of 7:2:1. Figure 6 presents segmentation examples produced by different networks on our tomato dataset. The results indicate that all the selected networks can clearly segment the tomato fruits; however, segmentation inaccuracies occur in some regions due to environmental factors such as lighting variations.

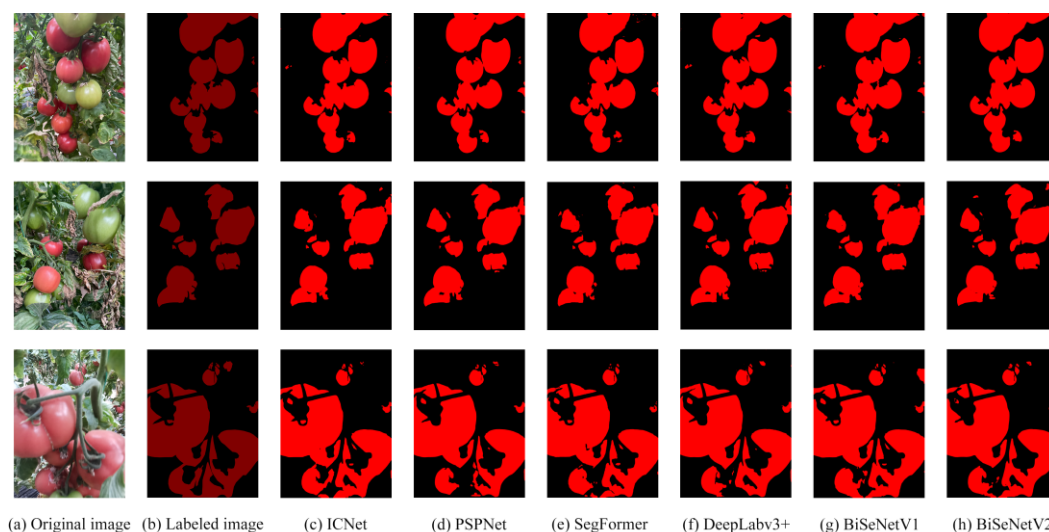


Figure 6. Examples of the segmentation results from different semantic segmentation networks: (a) original images; (b) manually annotated images; and (c–h) segmentation results from the ICNet, PSPNet, SegFormer, DeepLabv3+, BiSeNetV1, and BiSeNetV2 networks, respectively.

To quantitatively evaluate the semantic segmentation results, we selected the *mIoU*, *mAcc*, and *FPS* as evaluation metrics for the different semantic segmentation networks.

The $mIoU$ represents the average ratio of the intersection to the union between the ground truth and predicted regions for each class, expressed by Equation (5), where TP denotes true positives, FP false positives, FN false negatives, and k is the number of classes. The $mAcc$ indicates the mean classification accuracy across all the classes and is defined by Equation (6), where i denotes the i -th class. FPS represents the number of images processed per second by the semantic segmentation network, calculated by Equation (7), where $frameNum$ is the total number of processed images and $elapsedTime$ is the total processing time.

$$mIoU = \frac{1}{k+1} \sum_{i=0}^k \frac{TP}{FN + FP + TP} \quad (5)$$

$$mAcc = \frac{1}{k+1} \sum_{i=0}^k \frac{TP_i}{TP_i + FN_i} \quad (6)$$

$$FPS = \frac{frameNums}{elapsedTime} \quad (7)$$

The experiments were conducted on a Ubuntu 20.04 operating system, using an Intel® Core™ i9-14900KF CPU and an NVIDIA RTX 4080S GPU. PyTorch 1.11 was employed for network training. The training process used a fixed number of iterations, with the maximum set uniformly to 20,000 iterations, and the learning rate was set to 1×10^{-3} . Table 1 presents the final experimental results.

Table 1. Performance comparison of various semantic segmentation networks.

Model	Encoder	$mIoU$ (%)	$mAcc$ (%)	FPS
ICNet	PSPNet50	93.51	96.19	60.12
PSPNet	ResNet50	95.26	97.39	53.48
SegFormer	Mix Transformer B0	94.44	96.84	57.97
DeepLabv3+	ResNet50	92.21	96.41	51.63
BiSeNetV1	ResNet18	93.77	96.58	61.59
BiSeNetV2	-	95.37	97.38	61.98

The experimental results reveal a trade-off between segmentation accuracy and processing speed across different semantic segmentation networks on the tomato dataset. PSPNet and BiSeNetV2 achieved $mIoU$ scores of 95.26% and 95.37%, respectively, with both networks reaching $mAcc$ values above 97%, indicating strong performance in terms of the segmentation accuracy. BiSeNetV2 attained the highest inference speed at 61.98 FPS , demonstrating well-balanced performance between accuracy and real-time processing. Although ICNet and BiSeNetV1 showed slightly lower accuracy compared to PSPNet and BiSeNetV2, their FPS values reached 60.12 and 61.59, respectively, indicating solid real-time capabilities. DeepLabv3+ achieved relatively high $mAcc$ but exhibited the lowest $mIoU$ and FPS , suggesting that while its performance was balanced, it was not particularly outstanding in this experiment. Overall, BiSeNetV2 demonstrated the best balance between high segmentation accuracy and fast inference speed, making it well-suited to meet the system's requirements for both segmentation precision and real-time performance.

3.2. Three-Dimensional Semantic Reconstruction of Tomato Plants

To validate the proposed system, we deployed it on an unmanned vehicle and conducted field experiments. In the tomato cultivation base, the plants were grown using a string-hanging method and exhibited high planting density. In this study, 3D semantic point cloud maps were constructed for both individual tomato plants and entire rows. For the purpose of dataset construction and fruit count estimation, only tomato fruits were annotated during the experiments. To facilitate the direct extraction of the predicted values based on the point cloud coordinates, the DepthMapFactor parameter was set to 1000 during the 3D reconstruction process. The resulting 3D semantic point cloud maps generated by the system are shown in Figure 7.



(a) Three-dimensional semantic reconstruction result of a single tomato plant



(b) Three-dimensional semantic reconstruction result of a single row of tomato plants

Figure 7. Three-dimensional semantic reconstruction results of tomato plants: (a) shows the 3D semantic reconstruction of a single tomato plant; and (b) presents the reconstruction of an entire row of tomato plants. In the semantic point cloud, tomato fruits are highlighted in red.

Figure 7 shows the 3D semantic reconstruction results of the tomato plants. As observed, the system effectively distinguishes between the tomato plants and their fruits. However, the reconstructed plant boundaries are not clearly defined. This is mainly attributed to lighting variations and sensor measurement errors, which introduce a large number of outliers into the point cloud map. These outliers negatively impact the accuracy of the subsequent phenotypic trait estimation. Therefore, filtering is required to remove the noise and improve the overall mapping accuracy.

Considering the sources of noise and the morphological characteristics of tomato plants, we applied a two-stage filtering approach combining pass-through filtering and statistical filtering. First, pass-through filtering was used in the vertical direction to remove a large number of non-tomato and background points. Then, statistical filtering was applied to eliminate the remaining outliers and refine the boundaries of the tomato plant point cloud. Figure 8 illustrates the results at different stages of the filtering process.

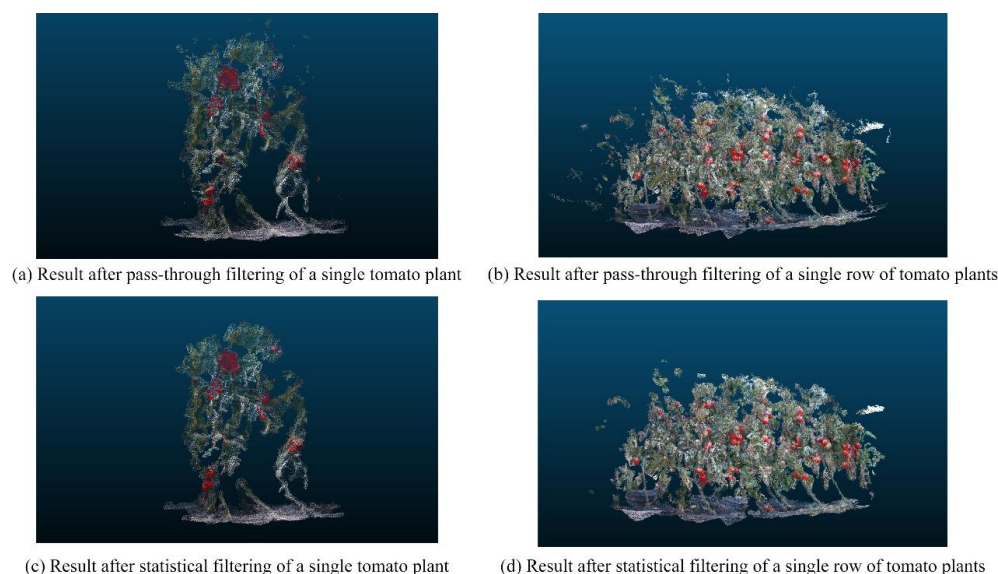


Figure 8. Filtering results of the tomato plant point clouds: (a,c) show the results of pass-through filtering and statistical filtering for a single tomato plant, respectively; and (b,d) show the results of pass-through filtering and statistical filtering for a single row of tomato plants, respectively.

As shown in Figure 8, the pass-through filtering method effectively removes non-tomato outliers such as ground and sky points in the spatial domain, ensuring the independence and accuracy of the tomato plant structure in the point cloud map. Furthermore, statistical filtering is applied in the local density domain to eliminate outliers. The figure illustrates that statistical filtering successfully suppresses local anomalies caused by sensor noise or environmental interference, thereby improving the overall clarity and consistency of the point cloud map.

Traditional point cloud maps require large amounts of storage space, which poses a challenge for embedded systems with limited storage capacity, making it difficult to store large-scale 3D farmland maps. To address this issue and enable scalable 3D modeling in agricultural scenarios, we adopted OctoMap to compress and store the point cloud data. Figure 9 shows the visualization results of an OctoMap constructed from the point cloud of a single row of tomato plants, with the base resolution set to 0.05 m.

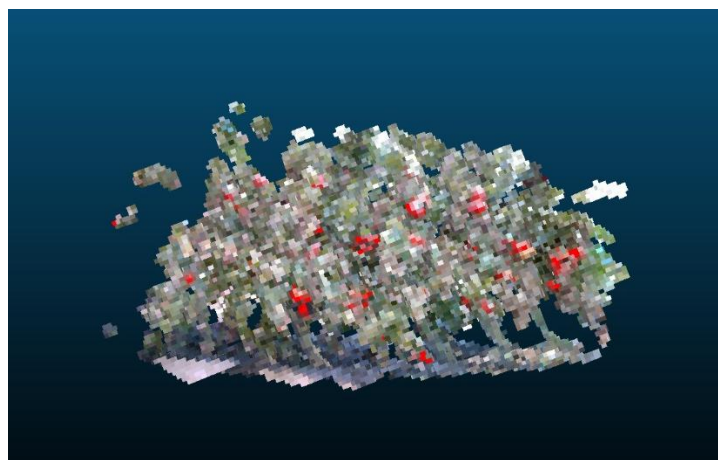


Figure 9. Visualization result of the OctoMap.

To quantitatively evaluate the storage efficiency of OctoMap for point cloud compression, we selected three individual tomato plants and two rows of tomato plants and assessed the storage consumption of their 3D semantic point clouds. Since both filtering

operations and OctoMap construction contribute to storage reduction, we conducted a staged quantitative analysis of the memory usage. The experimental results are presented in Table 2.

As shown in the table, the filtering process not only improves the quality of the point clouds but also reduces the memory consumption by an average of 9.92%. Furthermore, the use of OctoMap significantly reduces the storage size of the point cloud maps, achieving an average memory savings of 96.70%. Therefore, the filtering process combined with OctoMap construction effectively addresses the storage limitations of embedded devices in large-scale farmland scenarios.

Table 2. Evaluation results of the memory consumption for filtering operations and OctoMap storage.

Sample	Semantic Point Cloud Map (MB)	After Filtering (MB)	Memory Saving After Filter (%)	Building OctoMap (MB)	OctoMap Memory Saving (%)
The first tomato plant	7.73	6.96	9.96	0.32	95.86
The second tomato plant	8.11	7.49	7.64	0.40	95.07
The third tomato plant	8.49	7.74	8.83	0.35	95.88
The first row of tomato plants	73.63	64.35	12.60	1.32	98.21
The second row of tomato plants	56.78	50.79	10.55	0.86	98.49
Mean	-	-	9.92	-	96.70

3.3. Phenotypic Trait Acquisition of Tomato Plants

Accurate 3D models of tomato plants can be obtained through 3D semantic reconstruction. To evaluate the accuracy of the system's 3D reconstruction results and extract phenotypic traits, we selected the 3D semantic point clouds of 10 tomato plants for partial phenotypic feature extraction during the experiments. The ground truth phenotypic traits of the tomato plants were obtained through manual measurements. All the experimental results were quantitatively evaluated using the relative error, which is defined by Equation (8), where N_p denotes the predicted value from the system and N_t denotes the ground truth measurement.

$$\delta = \frac{|N_p - N_t|}{N_t} \times 100\% \quad (8)$$

3.3.1. Acquisition of Plant Height, Canopy Width, and Plant Volume

Acquiring phenotypic traits such as the plant height and canopy width of tomato plants enables an understanding of their growth status and guides optimal planting density. Therefore, we estimated parameters including the plant height, canopy width, and volume. The predicted values were obtained using the filtered semantic point clouds of the tomato plants, where H_p , W_p , and V_p denote the plant height, canopy width, and volume of the 3D model, respectively. In the semantic point cloud, H_{max} , W_{max} , and L_{max} represent the maximum values of the plant model along the y -axis, x -axis, and z -axis,

respectively, while H_{min} , W_{min} , and L_{min} represent the corresponding minimum values. Therefore, H_p , W_p , and V_p can be calculated according to Equation (9).

$$\begin{cases} H_p = H_{max} - H_{min} \\ W_p = \frac{(W_{max} - W_{min}) + (L_{max} - L_{min})}{2} \\ V_p = H_p \times W_p \times W_p \end{cases} \quad (9)$$

To enhance the validity of the experimental evaluation, the manual measurement of the tomato canopy width was conducted along directions consistent with the x -axis and z -axis of the system's 3D reconstruction coordinate frame. Since the tomato plants in the orchard are cultivated using a string-hanging method, for convenient evaluation, the height measurement and prediction of the tomato plants were defined from the plant base to the position of the fifth fruit cluster, representing the ground truth and predicted plant height, respectively. Table 3 presents the experimental results comparing the ground truth measurements and system predictions for the plant height, canopy width, and volume of the tomato plants.

Table 3. Measured and predicted values of the plant height, canopy width, and volume for 10 tomato plants.

Tomato Plant	Plant Height (cm)			Canopy Width (cm)			Volume (dm ³)		
	Predicted Value	Measured Value	Relative Error	Predicted Value	Measured Value	Relative Error	Predicted Value	Measured Value	Relative Error
1	116.9	120.5	2.99	62.5	55.7	12.21	456.64	373.85	22.15
2	123.3	120.7	2.15	56.6	49.6	14.11	395.00	296.94	33.02
3	142.3	138.6	2.67	59.1	50.6	16.80	497.03	354.87	40.06
4	120.7	125.7	3.98	55.6	48.6	14.40	373.13	296.90	25.68
5	136.8	132.4	3.32	57.0	53.4	6.74	444.46	377.55	17.72
6	125.4	130.7	4.06	55.2	51.1	8.02	382.10	341.29	11.96
7	140.1	133.5	4.94	54.4	50.2	8.37	414.61	336.43	23.24
8	133.7	130.5	2.45	52.6	46.8	12.39	369.92	285.83	29.42
9	122.4	126.5	3.24	48.0	40.6	18.23	282.01	208.52	35.24
10	138.6	135.8	2.06	51.8	45.4	14.10	371.90	279.91	32.86
Mean	-	-	3.86	-	-	14.34	-	-	27.14

Analysis of Table 3 indicates that the average relative error for predicting the tomato plant height through 3D reconstruction is 3.86%, demonstrating high prediction accuracy. This is attributed to the clear and well-defined boundaries for both manual measurement and prediction of the plant height, which are less affected by external factors. However, the average relative error for the canopy width prediction is 14.34%. This larger error is mainly due to the susceptibility of tomato leaves to external influences during 3D reconstruction, causing leaf positions to shift and resulting in inaccurate mapping boundaries. Additionally, unavoidable manual intervention during physical measurements contributes to the error. The plant volume, calculated based on the height and canopy width, accumulates errors from both parameters, leading to an average relative error of 27.14% in the volume prediction. Despite these errors, the average relative errors for all the phenotypic parameters remain within acceptable ranges, thereby validating the feasibility of obtaining plant phenotypic traits using 3D reconstruction.

3.3.2. Acquisition of Tomato Fruit Count and Volume

We obtained the number and volume of the tomato fruits from the filtered semantic point cloud. Since the semantic point cloud contained only the segmented tomato fruits, a conditional filter with constrained color thresholds was applied to extract the tomato fruit points. The extraction result is shown in Figure 10a. The extracted point cloud was then imported into CloudCompare software, where the RANSAC Shape Detection tool was used for spherical fitting. Based on the empirically measured radius range of typical tomato fruits, the minimum radius was set to 0.023 and the maximum radius to 0.030. The final fitting result is shown in Figure 10b, in which the number of fitted spheres corresponds to the number of tomato fruits predicted by the system.



Figure 10. Results of the tomato fruit point cloud extraction and spherical fitting: (a) shows the extracted point cloud of the tomato fruits, and (b) shows the result after spherical fitting of the tomato fruit point cloud.

To verify the feasibility of fruit counting and indirectly evaluate the accuracy of the 3D map, we conducted fruit count experiments on three individual tomato plants and three rows of tomato plants. A quantitative assessment was performed on the spherical fitting results of the tomato fruits. The experimental results are presented in Table 4, where the ground truth number of tomato fruits was obtained through manual counting.

Analysis of Table 4 shows that, after multiple trials, the average relative error of the predicted fruit count was 14.36%, indicating a high level of prediction accuracy. During the experiments, the completeness of the reconstruction for occluded fruits was relatively low due to occlusions caused by leaves and neighboring fruits. As a result, these occluded fruits were excluded from the spherical fitting process, contributing to the prediction error. In addition, residual noise points in the extracted tomato fruit point cloud may have led to overlapping fitted spheres during the fitting process, introducing further errors.

Table 4. Comparison between the ground truth and predicted number of tomato fruits.

Sample	Predicted Number	Ground Truth Number	Relative Error (%)
The first tomato plant	7	6	16.67
The second tomato plant	9	12	25.00
The third tomato plant	10	11	9.09
The first row of tomato plants	46	52	11.54
The second row of tomato plants	39	44	11.36
The third row of tomato plants	47	56	12.50
Mean	-	-	14.36

After harvesting, tomato fruits need to be graded, and the fruit volume is an important evaluation criterion in this process. We estimated the fruit volume based on the results obtained from the spherical fitting experiments with the tomato fruits. Specifically,

the volume of each fitted sphere was taken as the predicted volume of the corresponding tomato fruit, which can be calculated using Equation (10), where r denotes the radius of the fitted sphere. Fruits that were omitted during the spherical fitting were excluded from the statistics. Since the actual volume of tomato fruits is difficult to measure directly, we approximated each fruit as an ellipsoid. A vernier caliper was used to measure the length, width, and height of the fruit, and the approximated actual volume was calculated accordingly. This value was considered the ground truth volume, as given by Equation (11), where a , b , and c represent half of the measured length, width, and height of the tomato fruit, respectively.

$$V_p = \frac{4}{3}\pi r^3 \quad (10)$$

$$V_r = \frac{4}{3}\pi abc \quad (11)$$

Since each tomato plant bears multiple fruits, to reduce the repetitive work, we conducted the fruit volume estimation experiment on all the tomato fruits from only one selected plant. The experimental results are presented in Table 5.

Table 5. Comparison between the measured and predicted volumes of the tomato fruits.

Tomato Fruit	Predicted Radius (mm)	Predicted Volume (cm ³)	Measured Length (mm)	Measured Width (mm)	Measured Height (mm)	Measured Volume (cm ³)	Relative Error (%)
1	23.31	53.05	51.9	49.8	51.0	69.01	23.13
2	25.83	72.19	56.6	55.8	51.7	85.49	15.56
3	28.70	99.02	56.0	56.9	53.9	89.93	10.11
4	26.16	74.99	51.8	49.7	52.1	70.23	6.78
5	24.61	62.43	52.9	55.9	47.8	74.01	15.65
6	30.50	118.85	68.1	65.1	57.4	133.24	10.80
7	28.47	96.66	54.8	60.2	62.9	108.65	11.04
8	31.80	134.70	66.8	73.2	49.4	126.48	6.50
9	34.53	172.46	68.5	68.8	54.3	133.99	28.71
Mean	-	-	-	-	-	-	14.25

As shown in Table 5, the average relative error in terms of the tomato fruit volume prediction was 14.25%, indicating that the system can estimate the fruit volume with reasonably high accuracy. Since tomato fruits are irregularly shaped spheres, some error is inevitable during ground truth volume measurement. To mitigate this, multiple measurements were taken and averaged to reduce the variability. During the sphere fitting process, outliers and partial occlusions may cause the estimated radius to be larger or smaller than the actual value. To address this issue, the parameter Min support points per primitive was adjusted during the experiment to minimize the impact of such errors.

3.4. Localization Accuracy Experiment

The SLAM system performs localization by estimating the camera pose through the tracking thread and detecting loop closures via the loop detection thread. The accuracy of the system's localization directly affects the quality of the constructed 3D map, which in turn influences the accuracy of the phenotypic trait prediction for tomato plants. To address various challenges encountered in agricultural scenarios, we selected two datasets from distinct environments to evaluate the localization accuracy of the proposed system. Since data collection by an unmanned vehicle in farmland inevitably encounters occlusions caused by leaves, the first dataset used was the publicly available Rgbd_kidnap dataset, which includes a period of intentional occlusion during recording to simulate

camera obstruction by leaves in real-world scenarios. The second dataset was collected in the field by our team, featuring an actual trajectory with loop closures. During the experiment, we used RTK positioning to obtain the ground truth trajectory.

For the quantitative evaluation of the experimental results, we employed the *ATE* and *RMSE*. The *ATE* represents the absolute difference between the ground truth trajectory and the predicted trajectory, which can be expressed by Equation (12). Here, F_i denotes the absolute trajectory error at the i -th time step, Q_i is the ground truth pose at the i -th time step, P_i is the system's predicted pose at the i -th time step, and S is the transformation matrix between P_i and Q_i . The function $trans(X)$ extracts the translational component of X , and n represents the total number of frames used for the trajectory evaluation. Since an *ATE* value is computed for each time step, we used the mATE, which is the average of *ATE* values across all the time steps, to quantitatively assess the predicted trajectory.

$$F_i = Q_i^{-1}SP_i$$

$$ATE(i) = \sqrt{\frac{1}{n} \|trans(F_i)\|^2} \quad (12)$$

Additionally, we employed the *RMSE* of the *ATE* to quantify the deviation of the predicted trajectory. the *RMSE* is sensitive to outliers in the predicted trajectory and can be calculated using Equation (13).

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Q_i - P_i)^2} \quad (13)$$

Since the semantic segmentation module does not affect the system's localization accuracy, we conducted localization accuracy experiments using ORB-SLAM3 and ORB-SLAM2. The experimental results are presented in Table 6, where *Std* denotes the standard deviation of the *ATE*.

Table 6. Localization accuracy experiment results. *RMSE* represents the root mean square error of the *ATE*, mATE denotes the mean value of the *ATE*, and Std refers to the standard deviation.

Module	Rgbd_kidnap Dataset			Real-World Dataset		
	mATE(m)	RMSE(m)	Std	mATE (m)	RMSE(m)	Std
ORB-SLAM2	0.09	0.10	0.04	0.45	0.49	0.19
ORB-SLAM3	0.17	0.37	0.33	0.16	0.21	0.11

The predicted trajectories under occlusion conditions using the public dataset are shown in Figure 11. As illustrated in Figure 11, both systems lost tracking due to intentional occlusion. However, after the occlusion was removed, both systems quickly relocalized and resumed trajectory prediction. Compared with ORB-SLAM2, the ORB-SLAM3 system was able to redetect loop closures more rapidly after occlusion recovery. Analysis of Table 6 shows that the *ATE* and *RMSE* of the trajectories predicted by ORB-SLAM2 are lower than those of ORB-SLAM3. This is because the trajectory evaluation tool *evo* only compares the aligned portions of the predicted and ground truth trajectories based on the timestamps. Any unaligned or missing segments in the predicted trajectory are excluded from the evaluation. As a result, ORB-SLAM2 appears to achieve better performance in terms of the *evo*-based statistical metrics.

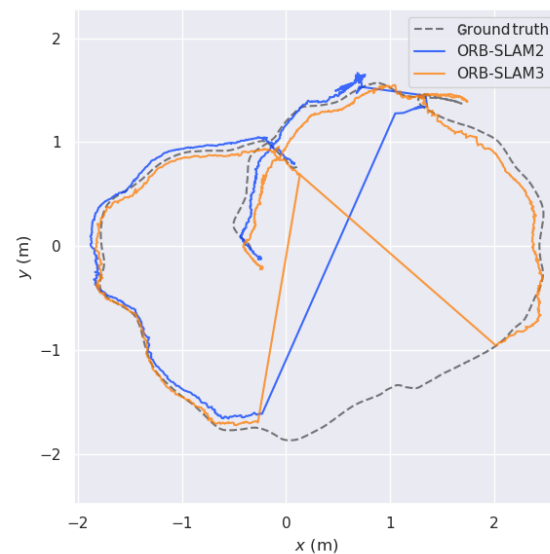


Figure 11. Experimental results on the Rgbd_kidnap public dataset.

Figure 12 presents the line chart of the *ATE* between the predicted trajectory and the ground truth at each time interval for both systems. As shown in the figure, the *ATE* for both systems increases significantly during the occlusion period. The predicted trajectory from ORB-SLAM3 exhibits a larger Std, but the *ATE* shows relatively small fluctuations outside the occlusion period, indicating stable prediction performance. To visually demonstrate the deviation between the predicted and ground truth trajectories, Figure 13 shows the deviation heatmaps of both systems on the Rgbd_kidnap public dataset.

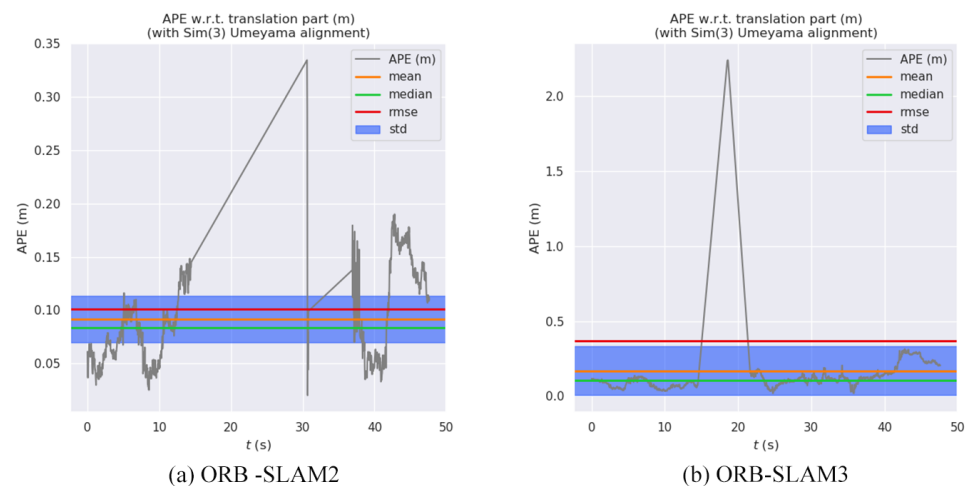


Figure 12. The *ATE* variation over time on the Rgbd_kidnap public dataset. The gray line represents the *ATE* curve over time; the orange line indicates the mean *ATE*; the green line represents the median *ATE*; the red line denotes the RMSE of the *ATE*; and the purple shaded area indicates the standard deviation of the *ATE*.

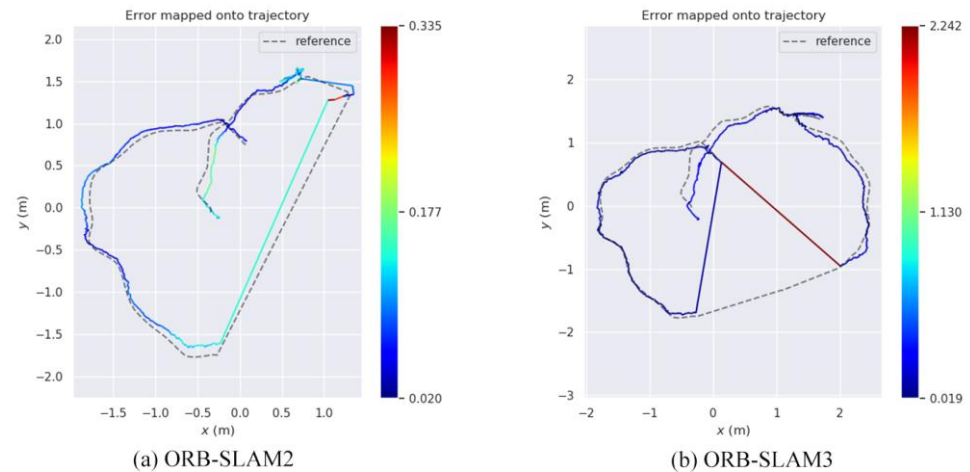


Figure 13. Trajectory deviation heatmaps on the Rgbd_kidnap public dataset.

Analysis of Table 6 and Figure 14 shows that both systems are capable of correctly recognizing loop closures in scenarios with looping paths. The mATE and RMSE values of ORB-SLAM3 were 0.16 m and 0.21 m, respectively, which are 0.29 m and 0.28 m lower than those of ORB-SLAM2. This demonstrates that ORB-SLAM3 achieves more accurate trajectory predictions than ORB-SLAM2 in standard field environments. In large-scale farmland scenarios, the accumulated drift errors can significantly affect the system performance. The loop closure thread plays a critical role in detecting such loops and correcting the accumulated errors. During the experiments, ORB-SLAM3 produced significantly better optimized trajectories after loop closure than ORB-SLAM2. These results indicate that ORB-SLAM3 offers higher robustness and can effectively reduce the risk of collisions or crop damage during real-world operation.

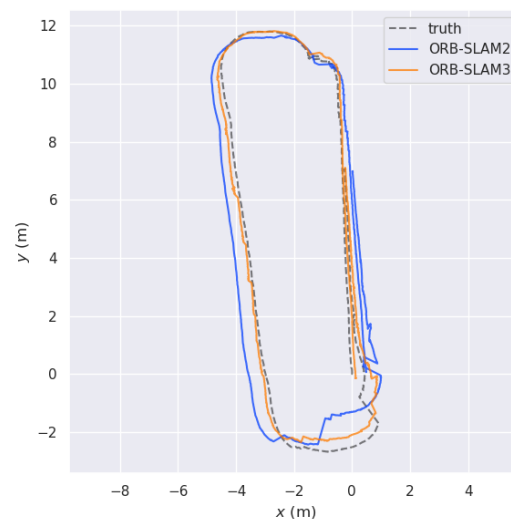


Figure 14. Experimental results on the field dataset.

Figure 15 illustrates the ATE curves of the predicted trajectories for both systems on the field dataset. As shown in the figure, ORB-SLAM3 maintained a more stable ATE curve throughout the operation, with significantly lower error magnitudes compared to ORB-SLAM2. The mean, median, RMSE, and Std of the trajectory errors were all smaller for ORB-SLAM3, indicating higher accuracy and stability in its pose estimation. These results suggest that ORB-SLAM3 exhibits greater robustness to noise during trajectory prediction and demonstrates overall superior performance compared to ORB-SLAM2.

Furthermore, Figure 16 presents deviation heatmaps that visually depict the differences between the predicted and ground truth trajectories for both systems.

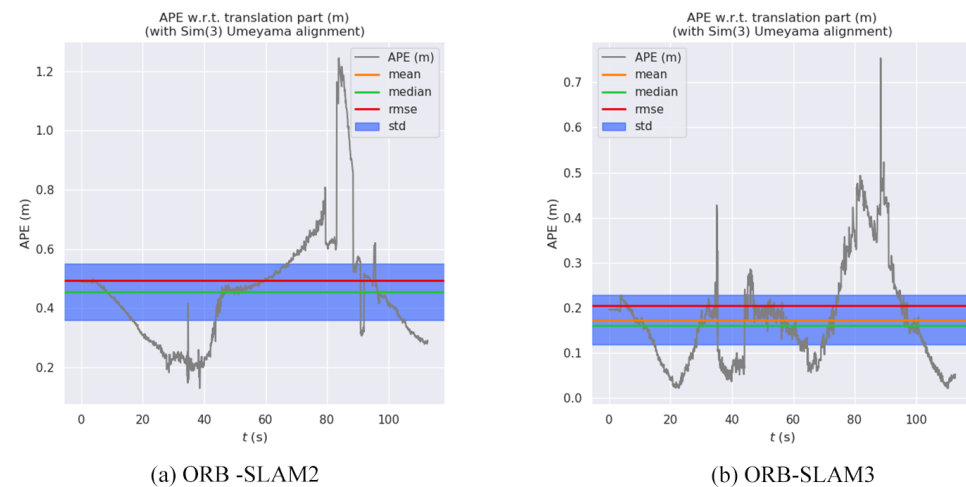


Figure 15. ATE variation over time on the field dataset.

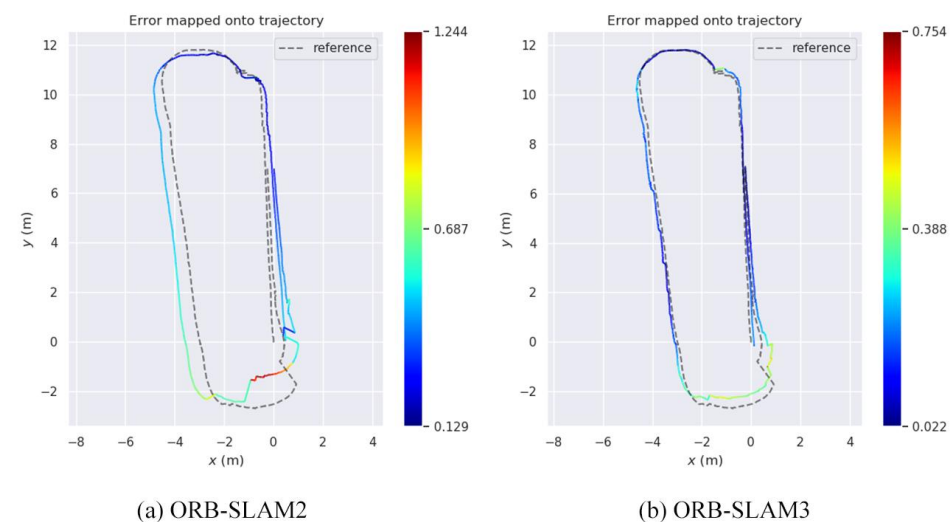


Figure 16. Trajectory deviation heatmaps on the field dataset.

4. Conclusions

This study proposes an ORB-SLAM3 system integrated with the lightweight semantic segmentation network BiSeNetV2, which is embedded on an unmanned vehicle for phenotypic trait acquisition of tomato plants in a greenhouse environment. To validate the feasibility of the proposed method, field experiments were conducted in a tomato cultivation base. The integration of BiSeNetV2 enables real-time semantic point cloud reconstruction, achieving a segmentation accuracy of 95.37% *mIoU* and a processing speed of 61.98 *FPS* on the tomato dataset. Outlier points in the point cloud were removed using a combination of pass-through and statistical filtering, and the processed data were stored using OctoMap, resulting in an average storage space reduction of 96.70%. Analysis of the point cloud allowed extraction of the plant height, canopy width, and plant volume, with relative errors of 3.86%, 14.34%, and 27.14%, respectively, compared to the ground truth measurements. The number and size of the tomato fruits were estimated via spherical fitting, yielding relative errors of 14.36% and 14.25%, respectively. In practical greenhouse scenarios, the system demonstrated robust localization performance, achieving an *mATE* of 0.16 m and an *RMSE* of 0.21 m on the field dataset.

Although the proposed system has achieved promising results in acquiring phenotypic traits of tomato plants, several limitations remain. Since the system relies on an RGB-D camera for data acquisition, it is sensitive to lighting variations—strong illumination or shadowed regions can introduce depth errors, thereby reducing the accuracy of the point cloud. In real-world operating environments, issues such as leaf overlap and fruit occlusion may occur, leading to incomplete semantic segmentation. These segmentation gaps can subsequently result in inaccuracies in the phenotypic parameter estimation.

Future work will explore the application prospects of the system across various other greenhouse fruits and vegetables to evaluate its generalizability. Meanwhile, efforts will be made to optimize the methods for acquiring accurate ground truth values for plant phenotypic traits.

Author Contributions: Writing—original draft preparation, P.W.; conceptualization, Y.H.; methodology, J.Z.; visualization, J.L.; supervision, R.C.; resources, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Science and Technology Innovation Project Fund of North China Institute of Aerospace Engineering, No. YKY-2024-81.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors upon request.

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

Appendix A

1. Improved ORB-SLAM3 system

Algorithm A1 Pipeline of the Improved ORB-SLAM3 System

Input: RGB-D **image** stream $I_t = \{I_{rgb}, I_{depth}\}$, camera intrinsics K

Output: Optimized camera trajectory T , semantic OctoMap M

- 1: Initialize ORB-SLAM3 system **with configuration file and** RGB-D mode
- 2: Load pretrained semantic segmentation model
- 3: Initialize OctoMap M **for** semantic point cloud fusion
- 4: **while** I_t has **next** frame **do**
- 5: Acquire RGB-D frame: $(I_{rgb}, I_{depth}) \leftarrow I_t$
- 6: Estimate current camera pose $T_t \leftarrow \text{Tracking}(I_{rgb}, I_{depth})$
- 7: **if** $\text{IsKeyFrame}(I_t)$ **then**
- 8: Insert keyframe into local **map**
- 9: Perform semantic segmentation **on** RGB **image**:
 $S_t \leftarrow \text{SemanticSegmentor}(I_{rgb})$
- 10: **Generate** semantic point cloud:
 $P_t \leftarrow \text{BackProject}(S_t, I_{depth}, T_t, K)$
- 11: Apply two-stage filtering **on** P_t :
 (a) Semantic + geometric constraints
 (b) Statistical **or** voxel-based denoising
- 12: Integrate filtered point cloud into OctoMap:
 $M \leftarrow M \cup P_t$
- 13: **end if**
- 14: **if** LocalMapping **is active** **then**
- 15: Run local bundle adjustment **and** **map** optimization
- 16: **end if**
- 17: **if** LoopClosing **is triggered** **then**
- 18: Detect **loop** closure

```

19:         if Loop is confirmed then
20:             Perform pose graph optimization
21:         end if
22:     end if
23:     Update visualization with trajectory and semantic map
24: end while
25: Save final trajectory T
26: Save semantic OctoMap M to disk

```

2. OctoMap Building Method

Algorithm A2 OctoMap Construction Algorithm

Input: Semantic point cloud $P = \{p_i = (x_i, y_i, z_i, c_i)\}$,
corresponding sensor pose T ,
occupancy octree O (resolution r)

Output: Updated semantic occupancy **map** O

```

1: for each point  $p_i \in P$  do
2:     Transform point to world coordinates:
         $p_{i\_world} \leftarrow T \oplus p_i$ 
3:     Compute 3D ray from sensor origin to  $p_{i\_world}$ 
4:     for each voxel  $v_j$  along the ray (excluding endpoint) do
5:         Update occupancy of  $v_j$  as free space in octree  $O$ 
6:     end for
7:     Find leaf node  $v_i$  corresponding to  $p_{i\_world}$ 
8:     Update occupancy of  $v_i$  as occupied in octree  $O$ 
9:     Update semantic label of  $v_i$  using Bayesian fusion:
         $label(v_i) \leftarrow Fuse(label(v_i), c_i)$ 
10: end for

```

References

1. Sampaio, G.S.; Silva, L.A.; Marengoni, M. 3D reconstruction of non-rigid plants and sensor data fusion for agriculture phenotyping. *Sensors* **2021**, *21*, 4115.
2. Zahid, A.; Mahmud, M.; He, L.; Heinemann, P.; Choi, D.; Schupp, J. Technological advancements towards developing a robotic pruner for apple trees: A review. *Comput. Electron. Agric.* **2021**, *189*, 106383.
3. Krul, S.; Pantos, C.; Frangulea, M.; Valente, J. Visual SLAM for indoor livestock and farming using a small drone with a monocular camera: A feasibility study. *Drones* **2021**, *5*, 41.
4. Ma, X.; Wei, B.; Guan, H.; Yu, S. A method of calculating phenotypic traits for soybean canopies based on three-dimensional point cloud. *Ecol. Inform.* **2022**, *68*, 101524.
5. Liu, F.; Hu, P.; Zheng, B.; Duan, T.; Zhu, B.; Guo, Y. A field-based high-throughput method for acquiring canopy architecture using unmanned aerial vehicle images. *Agric. For. Meteorol.* **2021**, *296*, 108231.
6. Rossi, R.; Costafreda-Aumedes, S.; Leolini, L.; Leolini, C.; Bindi, M.; Moriondo, M. Implementation of an algorithm for automated phenotyping through plant 3D-modeling: A practical application on the early detection of water stress. *Comput. Electron. Agric.* **2022**, *197*, 106937.
7. Wu, D.; Yu, L.; Ye, J.; Zhai, R.; Duan, L.; Liu, L.; Wu, N.; Geng, Z.; Fu, J.; Huang, C.; et al. Panicle-3D: A low-cost 3D-modeling method for rice panicles based on deep learning, shape from silhouette, and supervoxel clustering. *Crop J.* **2022**, *10*, 1386–1398.
8. Shu, F.; Lesur, P.; Xie, Y.; Pagani, A.; Stricker, D. SLAM in the field: An evaluation of monocular mapping and localization on challenging dynamic agricultural environment. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–8 January 2021.
9. Malambo, L.; Popescu, S.; Murray, S.; Putman, E.; Pugh, N.; Horne, D.; Richardson, G.; Sheridan, R.; Rooney, W.; Avant, R. et al. Multitemporal field-based plant height estimation using 3D point clouds generated from small unmanned aerial systems high-resolution imagery. *Int. J. Appl. Earth Obs. Geoinf.* **2018**, *64*, 31–42.
10. Yin, X.; Noguchi, N.; Choi, J. Development of a target recognition and following system for a field robot. *Comput. Electron. Agric.* **2013**, *98*, 17–24.

11. Bao, Y.; Tang, L.; Breitzman, M.; Fernandez, M.; Schnable, P. Field-based robotic phenotyping of sorghum plant architecture using stereo vision. *J. Field Robot.* **2019**, *36*, 397–415.
12. Van Henten, E.J.; Hemming, J.; Tuijl, B.A.J.; Kornet, J.G.; Meuleman, J.; Bontsema, J.; Os, E.A. An autonomous robot for harvesting cucumbers in greenhouses. *Auton. Robot.* **2002**, *13*, 241–258.
13. Zhu, F.; Thapa, S.; Gao, T.; Ge, Y.; Walia, H.; Yu, H. 3D reconstruction of plant leaves for high-throughput phenotyping. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; IEEE: New York, NY, USA, 2018.
14. Liu, J.; Xu, X.; Liu, Y.; Rao, Z.; Smith, M.; Jin, L.; Li, B. Quantitative potato tuber phenotyping by 3D imaging. *Biosyst. Eng.* **2021**, *210*, 48–59.
15. Shi, Y.; Wang, N.; Taylor, R.K.; Raun, W.R. Improvement of a ground-LiDAR-based corn plant population and spacing measurement system. *Comput. Electron. Agric.* **2015**, *112*, 92–101.
16. Marinello, F.; Pezzuolo, A.; Donato, C.; Sartori, L. Kinect 3D reconstruction for quantification of grape bunches volume and mass. *Eng. Rural Dev.* **2016**, *15*, 876–881.
17. Thapa, S.; Zhu, F.; Walia, H.; Yu, H.; Ge, Y. A novel LiDAR-based instrument for high-throughput, 3D measurement of morphological traits in maize and sorghum. *Sensors* **2018**, *18*, 1187.
18. Fan, Y.; Feng, Z.; Mannan, A.; Khan, T.; Shen, C.; Saeed, S. Estimating tree position, diameter at breast height, and tree height in real-time using a mobile phone with RGB-D SLAM. *Remote Sens.* **2018**, *10*, 1845.
19. Dong, W.; Roy, P.; Isler, V. Semantic mapping for orchard environments by merging two-sides reconstructions of tree rows. *J. Field Robot.* **2020**, *37*, 97–121.
20. Pierzchała, M.; Giguère, P.; Astrup, R. Mapping forests using an unmanned ground vehicle with 3D LiDAR and graph-SLAM. *Comput. Electron. Agric.* **2018**, *145*, 217–225.
21. Kim, K.; Deb, A.; Cappelleri, D.J. P-AgSLAM: In-Row and Under-Canopy SLAM for Agricultural Monitoring in Cornfields. *IEEE Robot. Autom. Lett.* **2024**, *9*, 4982–4989.
22. Lobefaro, L.; Malladi, M.; Vysotska, O.; Guadagnino, T.; Stachniss, C. Estimating 4D Data Associations Towards Spatial-Temporal Mapping of Growing Plants for Agricultural Robots. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 1–5 October 2023; IEEE: New York, NY, USA, 2023.
23. Nellithimaru, A.K.; Kantor, G.A. Rols: Robust object-level slam for grape counting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, CA, USA, 16–17 June 2019.
24. Fan, Z.; Sun, N.; Qiu, Q.; Li, T.; Feng, Q.; Zhao, C. In situ measuring stem diameters of maize crops with a high-throughput phenotyping robot. *Remote Sens.* **2022**, *14*, 1030.
25. Wang, J.-Y.; Lin, T.-T. Application of a visual-based autonomous drone system for greenhouse muskmelon phenotyping. In Proceedings of the 2023 ASABE Annual International Meeting, Omaha, NE, USA, 9–12 July 2023; American Society of Agricultural and Biological Engineers: St. Joseph, MI, USA, 2023.
26. Islam, R.; Habibullah, H.; Hossain, T. AGRI-SLAM: A real-time stereo visual SLAM for agricultural environment. *Auton. Robot.* **2023**, *47*, 649–668.
27. Tang, B.; Guo, Z.; Huang, C.; Huai, S.; Gai, J. A fruit-tree mapping system for semi-structured orchards based on multi-sensor-fusion SLAM. *IEEE Access* **2024**, *12*, 162122–162130.
28. Xiong, J.; Liang, J.; Zhuang, Y.; Hong, D.; Zheng, Z.; Liao, S.; Hu, W.; Yang, Z. Real-time localization and 3D semantic map reconstruction for unstructured citrus orchards. *Comput. Electron. Agric.* **2023**, *213*, 108217.
29. Liu, X.; Chen, S.; Liu, C.; Shivakumar, S.; Das, J.; Taylor, C. Monocular camera based fruit counting and mapping with semantic data association. *IEEE Robot. Autom. Lett.* **2019**, *4*, 2296–2303.
30. Yuan, Q.; Wang, P.; Luo, W.; Zhou, Y.; Chen, H.; Meng, Z. Simultaneous Localization and Mapping System for Agricultural Yield Estimation Based on Improved VINS-RGBD: A Case Study of a Strawberry Field. *Agriculture* **2024**, *14*, 784.
31. Giang, T.T.H.; Ryoo, Y.-J. Pruning points detection of sweet pepper plants using 3D point clouds and semantic segmentation neural network. *Sensors* **2023**, *23*, 4040.
32. Liu, T.; Chopra, N.; Samtani, J. Information system for detecting strawberry fruit locations and ripeness conditions in a farm. *Biol. Life Sci. Forum* **2022**, *16*, 22.
33. Xie, X.; Yan, Z.; Zhang, Z.; Qin, Y.; Jin, H.; Zhang, C.; Xu, M. Construction of Three-Dimensional Semantic Maps of Unstructured Lawn Scenes Based on Deep Learning. *Appl. Sci.* **2024**, *14*, 4884.
34. Pan, Y.; Hu, K.; Cao, H.; Kang, H.; Wang, X. A novel perception and semantic mapping method for robot autonomy in orchards. *Comput. Electron. Agric.* **2024**, *219*, 108769.

35. Che, Y.; Wang, Q.; Xie, Z.; Zhou, L.; Li, S.; Hui, F.; Wang, X.; Li, B.; Ma, Y. Estimation of maize plant height and leaf area index dynamics using an unmanned aerial vehicle with oblique and nadir photography. *Ann. Bot.* **2020**, *126*, 765–773.
36. Cheng, M.; Jiao, X.; Liu, Y.; Shao, M.; Yu, X.; Bai, Y.; Wang, Z.; Wang, S.; Tuohuti, N.; Liu, S.; et al. Estimation of soil moisture content under high maize canopy coverage from UAV multimodal data and machine learning. *Agric. Water Manag.* **2022**, *264*, 107530.
37. Sun, G.; Wang, X.; Ding, Y.; Lu, W.; Sun, Y. Remote measurement of apple orchard canopy information using unmanned aerial vehicle photogrammetry. *Agronomy* **2019**, *9*, 774.
38. Sarron, J.; Malézieux, É.; Sané, C.; Faye, É. Mango yield mapping at the orchard scale based on tree structure and land cover assessed by UAV. *Remote Sens.* **2018**, *10*, 1900.
39. Siebers, M.H.; Edwards, E.J.; Jimenez-Berni, J.A.; Thomas, M.R.; Salim, M.; Walker, R.R. Fast phenomics in vineyards: Development of GROVER, the grapevine rover, and LiDAR for assessing grapevine traits in the field. *Sensors* **2018**, *18*, 2924.
40. Qiu, Q.; Sun, N.; Bai, H.; Wang, N.; Fan, Z.; Wang, Y.; Meng, Z.; Li, B.; Cong, Y. Field-based high-throughput phenotyping for maize plant using 3D LiDAR point cloud generated with a Phenomobile. *Front. Plant Sci.* **2019**, *10*, 554.
41. Wang, P.; Luo, W.; Liu, J.; Zhou, Y.; Li, X.; Zhao, S.; Zhang, G.; Zhao, Y. Real-time semantic SLAM-based 3D reconstruction robot for greenhouse vegetables. *Comput. Electron. Agric.* **2025**, *237*, 110582.
42. Nguyen, P.; Badenhorst, P.E.; Shi, F.; Spangenberg, G.C.; Smith, K.F.; Daetwyler, H.D. Design of an unmanned ground vehicle and LiDAR pipeline for the high-throughput phenotyping of biomass in perennial ryegrass. *Remote Sens.* **2020**, *13*, 20.
43. Qi, H.; Wang, C.; Li, J.; Shi, L. Loop closure detection with CNN in RGB-D SLAM for intelligent agricultural equipment. *Agriculture* **2024**, *14*, 949.
44. Ma, Z.; Yang, S.; Li, J.; Qi, J. Research on slam localization algorithm for orchard dynamic vision based on YOLOD-SLAM2. *Agriculture* **2024**, *14*, 1622.
45. Li, G.; Liu, Z.; Ling, H. ICNet: Information conversion network for RGB-D based salient object detection. *IEEE Trans. Image Process.* **2020**, *29*, 4873–4884.
46. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
47. Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J.M.; Luo, P. SegFormer: Simple and efficient design for semantic segmentation with transformers. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 12077–12090.
48. Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.
49. Yu, C.; Wang, J.; Peng, C.; Gao, C.; Yu, G.; Sang, N. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.