2022 Joint 12th International Conference on Soft Computing and Intelligent Systems and 23rd International Symposium on Advanced Intelligent Systems (SCIS&ISIS) | 978-1-6654-9924-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/SCISISIS55246.2022.10001991

3D Semantic Point Clouds Construction based on ORB-SLAM3 and ICP Algorithm for Tomato Plants

Truong Thi Huong Giang
Department of Electrical Engineering
Mokpo National University
Mokpo, Korea
tthgiang@ttn.edu.vn

Young-Jae Ryoo
Dept. of Electrical and Control Eng.
Mokpo National University
Mokpo, Korea
yjryoo@mokpo.ac.kr

Dae-Young Im
Components & Materials R&D Group
Korea Institute of Industrial Technology
Gwangju, Korea
dylim@kitech.re.kr

***Abstract*—One or two RGB-D images cannot provide enough information to detect cut-off points in pruning systems. 3D semantic point clouds constructed from many RGB-D images represent real tomato plants and help the system find the cut-off points correctly. We proposed a method to create 3D semantic point clouds based on ORB-SLAM3, ICP (iterative closet point) algorithm, and semantic segmentation neural network. RGB-D images are converted to semantic images by the semantic segmentation neural network. Each pair of camera poses which is estimated by ORB-SLAM3, and an RGB-D semantic image is used to create a 3D point cloud. The ICP method is applied to stick and refine these point clouds to construct a full 3D semantic point cloud.**

Keywords—3D semantic point clouds, ORB-SLAM3, ICP, cut-off points, semantic segmentation neural network, tomato plants

I. INTRODUCTION

Tomato plants are popular plants on smart farms. Many previous kinds of research on tomatoes, such as finding their disease through their leaves [1] or autonomous tomato harvesting [2]. In our research, we focus on autonomous pruning on tomato plants. In this process, not only suckers but also old branches should be removed. In some complicated procedures, one or two suckers below the first flower should not be removed. The rules of pruning are complex and could be changed. They require many complicated operations such as counting, identifying the branches, suckers, and flowers, comparing their heights, and detecting which should be removed. There are many occlusions or hidden parts in 2D images, so 2D RGB-D images can not satisfy these requirements. Therefore, a 3D point cloud built from many RGB-D images can eliminate many occlusions and depict many parts that could be hidden in a 2D image.

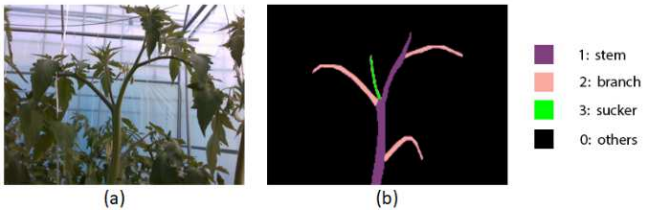

Fig. 1. Tomato plant parts. (a) is an RGB tomato plant, (b) is semantic tomato plant images

Many background parts were removed in Semantic RGB-D images, as in Fig. 1. A semantic RGB-D image is an image created from the semantic RGB image and its depth information. Therefore, 3D point clouds constructed by semantic RGB-D images, called 3D semantic point clouds, have fewer points than 3D normal point clouds made from normal RGB-D images. This leads to less time in execution in building 3D point clouds. In addition, the system only needs information on stems, branches, suckers, and other important parts to process further operations in detecting cut-off points. So, we use 3D semantic point clouds instead of 3D normal point clouds. We use the semantic segmentation neural network proposed in our previous research [3] for tomato parts recognition and create semantic RGB-D images. The output of this process is like the image (b) in Fig. 1.

ORB-SLAM families are Open-Source SLAM systems[4]–[6]. The input of ORB-SLAM families are images from the stereo, RGB-D, or monocular cameras, and the output is the estimated camera trajectory. In these systems, Bundle Adjustment is used to optimize the camera poses. ORB-SLAM3 is the newest version with multi-map SLAM and uses both pin-hole and fisheye lens models camera [6]. We have tried to construct 3D point clouds based on the camera poses of keyframes of ORB-SLAM3. However, the result is not good because of cumulative drift. A loop detection function decreases this problem in ORB-SLAM3, but a tomato plant is not large enough to detect a loop. On the other hand, we need precise 3D point clouds to perform the pruning action precisely. Therefore, we apply the ICP algorithm to stick these point clouds more accurately [7]. The idea of this algorithm is that iteration to find alignment. The two closet point clouds would be aligned quickly after a few iterations because most of their points are the same.

II. PROPOSED METHOD

We use two types of images as inputs of the system. Stereo fish-eye images are used as the input of ORB-SLAM3 to get the position of the fish-eye camera at each step. The proposed method is presented in Fig. 2. RGB-D images are used as the semantic segmentation neural network input to get the semantic images. The position of the RGB-D camera can be detected by the poses of the fish-eye camera and the transformation matrix between them. Then a 3D semantic point cloud can be constructed at each step. Finally, these point clouds are combined by the ICP module to get the tomato plant's final 3D semantic point cloud.

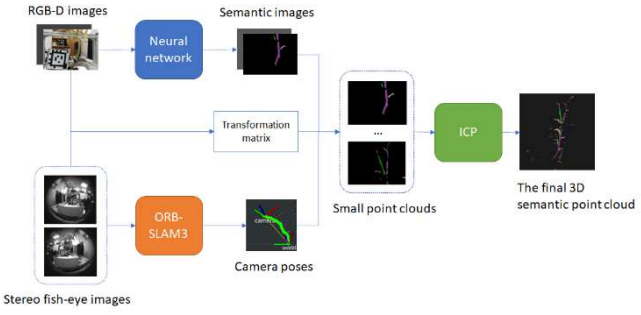

Fig. 2. Proposed method for 3D semantic point clouds construction

The details of the ICP module are presented in Fig. 3. We used 3D semantic point clouds to compare in ICP algorithm. Suppose the RGB-D images do not capture the tomato plants or small parts of tomato plants. In that case, the semantic images do not have much information, and the corresponding 3D semantic point clouds have a small number of points. Therefore, in the ICP module, we compare the number of points in 3D semantic point clouds to a threshold t. If this number is smaller than t, we use the 3D normal point clouds to compare in the ICP algorithm. After many experiments, we saw that if a semantic point cloud has more than 400 points, it ensures good results. So we choose t equal 400.

Another critical thing that we calculate is the position of each new point cloud when inserted into the list of point clouds. When comparing the new point cloud with the previous point clouds, we also compare it with the latest point cloud and the point clouds having the same spatial position. This makes the result correct when the camera moves back to the previous post.

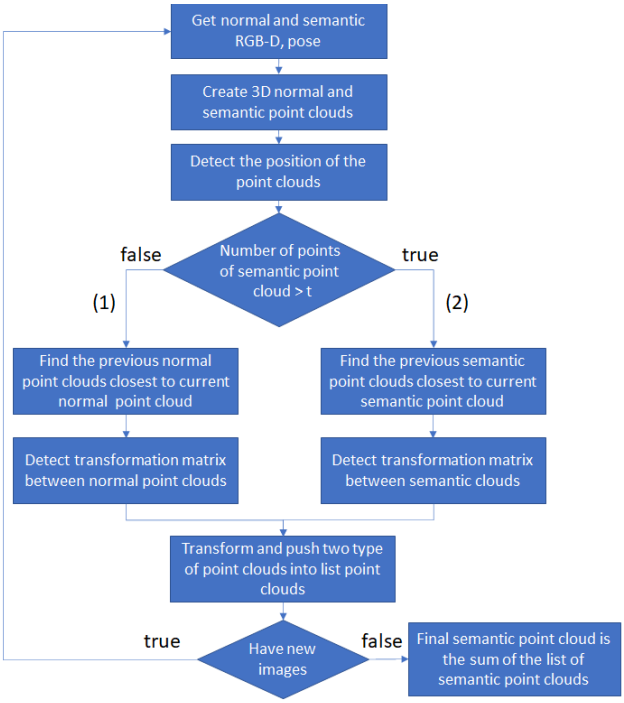

Fig. 3. Flow chart of the ICP module

III. Experiments and Results

A. Experiments

We use an Intel RealSense D435 camera and Intel RealSense T625 to capture RGB-D images and stereo fish-eye images. The cameras were hand-held and moved around the tomato plant to get images.

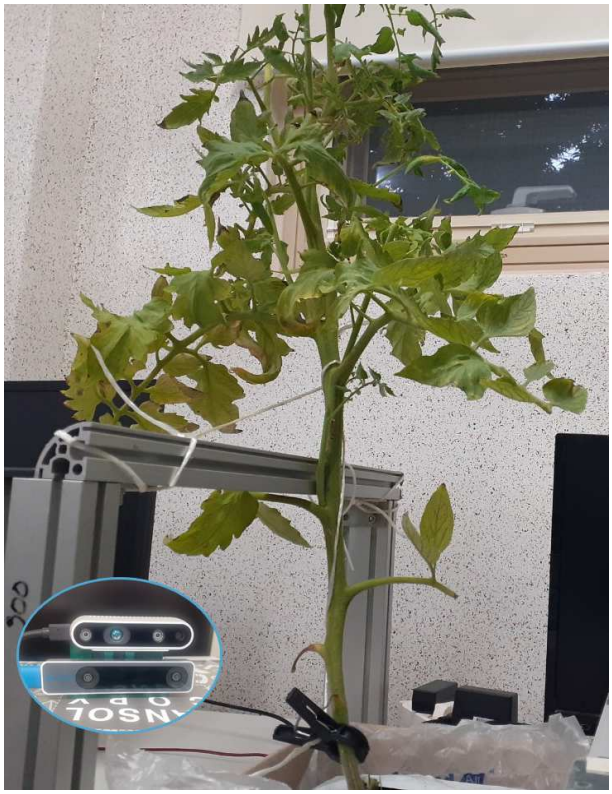

Fig. 4. Cameras and tomato plant used in experiment

We installed ORB-SLAM3 and ROS Noetic under Ubuntu 20.4. We wrote three ROS nodes. One ROS node was written in python to run the semantic segmentation neural network model, which would return the semantic images. The second node was written in C++ to get the camera poses from ORB-SLAM3 and publicize them. The final node was written in C++ as well. It created point clouds from sematic images and camera poses. Every when there was a new point cloud, we used the ICP algorithm to find the transformation matrix from the previous point cloud to the latest point cloud.

We focus on execution time and the visual result of 3D point clouds. The time we calculate in this experiment is the time of each step, from getting the images to the combined semantic point clouds. It includes the time semantic segmentation neural network prediction, detecting features and estimating the camera poses, building 3D point clouds, and iteration of ICP modules. We created three programs to prove that our proposed method has better performance. The first program named P1 used the proposed method. The second program, named P2, used a similar method. However, in the ICP module, we always use thread (1) in the block diagram in Fig. 3. We do not use semantic point clouds in the iterations of the ICP algorithm. The third program, P3, created 3D point clouds without an ICP module.

B. Results

Fig. 5 and Fig. 6 show two viewpoints of the results from P1 and P2 respectively. The result of P2 has errors at the sucker at the top of the figure, while the result of P1 is correct.

There are many errors in Fig. 7, the output of P3, because of cumulative drift. That is the reason why we need the ICP module to get better results.

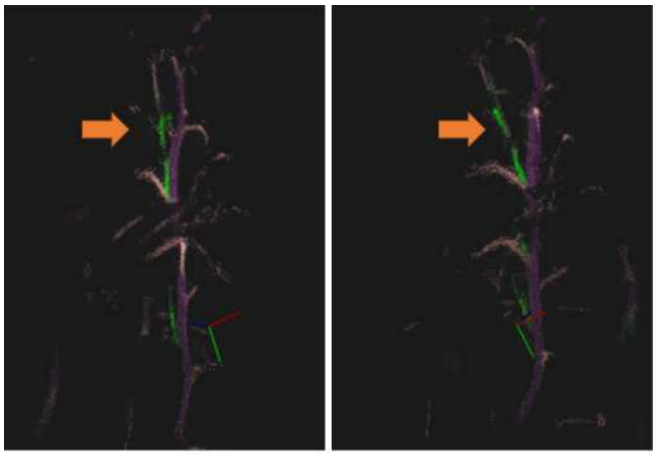

Fig. 5. 3D semantic point cloud output of P 1.

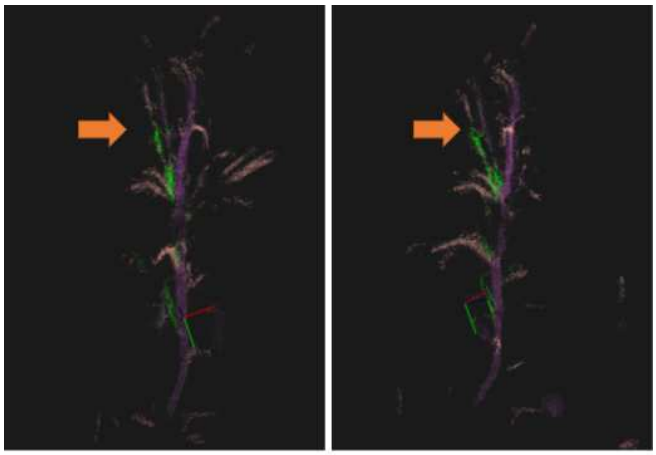

Fig. 6. 3D semantic point cloud output of P2.

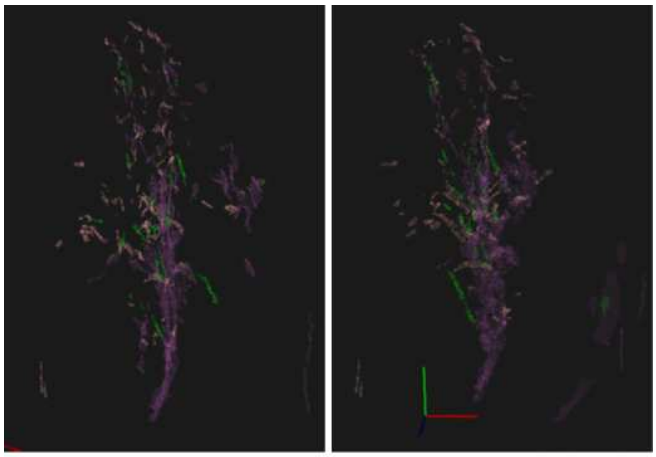

Fig. 7. 3D semantic point cloud output of P3.

Table 1 also shows that the execution time of P1 is smaller than P2. Semantic point clouds consist of only tomato plant part objects, while there are many types of objects in normal point clouds. This increases the execution time and has more errors when using normal point clouds in P2. P3 did not use ICP modules but it took 830 ms, slightly less than the time of P1 which use the ICP modules at 925 ms. It means that the long execution time is not in ICP modules.

TABLE 1. TIME EXECUTION OF THE TWO PROGRAMS

Name	Execution time (ms)
P1	925
P2	2,299
P3	830

IV. CONCLUSION

We proposed a method to create 3D semantic point clouds, which could help us prune tomato plants. This method used the result of ORB-SLAM3 as the input of our proposed program using the ICP algorithm on 3D point clouds. This program also applies a semantic segmentation neural network to create semantic RGB-D images, leading to better results and time-saving. We will continue to study this subject to decrease the execution time and complete the whole system with the cut-off point detection algorithm.

ACKNOWLEDGMENT

This work was supported by Korea Institute of Planning and Evaluation for Technology in Food, Agriculture and Forestry(IPET) and Korea Smart Farm R&D Foundation(KosFarm) through Smart Farm Innovation Technology Development Program, funded by Ministry of Agriculture, Food and Rural Affairs(MAFRA) and Ministry of Science and ICT(MSIT), Rural Development Administration(RDA) (421032-04).

REFERENCES

[1] A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park, "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition," *Sensors (Switzerland)*, vol. 17, no. 9, 2017.

[2] Y. Zhao, L. Gong, C. Liu, and Y. Huang, "Dual-arm Robot Design and Testing for Harvesting Tomato in Greenhouse," *IFAC-PapersOnLine*, vol. 49, no. 16, pp. 161–165, 2016,.

[3] T. T. H. Giang, T. Q. Khai, D. Im, and Y. Ryoo, "Fast Detection of Tomato Sucker Using Semantic Segmentation Neural Networks Based on RGB-D Images," *Sensors*, vol. 22, no. 14, p. 5140, Jul. 2022.

[4] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, 2015.

[5] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, 2017.

[6] C. Campos, R. Elvira, J. J. G. Rodriguez, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, 2021,.

[7] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes." IEEE Transactions ort Pattern Analysis and Machine Intelligence, p. vol. 14, no. 2, pp. 239–256, 1992.