

Visual LiDAR Odometry Using Tree Trunk Detection and LiDAR Localization

K. W. Park¹, S. Y. Park^{1*}

¹ Graduate School of Electronic and Electrical Engineering, Kyungpook National University, Daegu 41566, South Korea,
- kevin2760@naver.com, syark@knu.ac.kr

KEY WORDS: SLAM, Tree Trunk, Mapping, Stereo Camera, LiDAR.

ABSTRACT:

This paper presents a method of visual LiDAR odometry and forest mapping, leveraging tree trunk detection and LiDAR localization techniques. In environments like dense forests, where smooth GPS signals are unreliable, we employ camera and LiDAR sensors to accurately estimate the robot's position. However, forested or orchard settings introduce unique challenges, including a diverse mixture of trees, tall grass, and uneven terrain. To address these complexities, we propose a distance-based filtering method to extract data composed solely of tree trunk information from 2D LiDAR. By restoring arc data from the LiDAR sensor to its circular shape, we obtain position and radius measurements of reference trees in the LiDAR coordinate system. Then, these values are stored in a comprehensive tree trunk database. Our approach combines visual-based SLAM and LiDAR-based SLAM independently, followed by an integration step using the Extended Kalman Filter (EKF) to improve odometry estimation. Utilizing the obtained odometry information and the EKF, we generate a tree map based on observed trees. In addition, we use the tree position in the map as the landmark to reduce the localization error in the proposed SLAM algorithm. Experimental results show that the loop-closing error ranges between 0.3 to 0.5 meters. In the future, it is expected that this method will also be applicable in the fields of path planning and navigation.

1. INTRODUCTION

In recent years, mobile robots have been primarily used in different agricultural applications. The development of sensors, software applications, and communication techniques has encouraged active participation from researchers and companies to advance the utilization of autonomous agricultural robots to save on labor and costs. Map generation of forest environments is one of the important parts of the forest-related technologies. In the view of the complexity of the forest environment, achieving precise localization and navigation of robots is a primary challenge that needs to be addressed for successful mission execution. SLAM (simultaneous localization and mapping) (Dissanayake et al, 2001) technology has played an important role in solving problems related to indoor localization. SLAM-based robots are equipped with various sensors to accurately detect information about the surroundings such as image sensors, 2D and 3D LiDAR, radar, GPS, wheel encoder and IMU (Aguilar et al, 2020). GPS is widely used as a positioning sensor for robots to estimate their location. However, in areas with mountainous terrain and dense tree coverage, the GPS signal can be distorted, leading to inaccuracies in position estimation (Garforth and Webb, 2019). Cameras play a vital role in localization, mapping, and mobile navigation in outdoor agricultural environments, primarily due to their affordability. However, the camera's performance can be significantly influenced by external environmental factors, such as shadows or backlighting. While LiDAR sensors may be more expensive compared to image sensors, they are widely recognized as one of the most commonly employed sensors in outdoor agricultural environments. LiDAR operates as an active system that measure the time it takes for emitted light to travel to objects and provide

range and angle measurements for detected features. Another alternative is to use wheel odometry that estimates the position of a mobile robot from attached encoders of its wheels. However, it has drawbacks such as wheel slip or loss of traction when encountering uneven or varying friction on the floor. The integration of various sensors for mapping purposes not only enhances the robustness of the map but also enables mobile robots to accurately determine their position and orientation at each time step. Previous studies have investigated the individual use of GPS (Han, J. h et al, 2020), image sensors (S. Liu et al, 2022), and LiDAR (Li, S. et al, 2020) sensors for mapping applications. Furthermore, researchers are actively exploring sensor fusion techniques that combine different sensor modalities, such as camera and GPS (Cho, Y., et al, 2020), 3D LiDAR and GPS (Mao, Wenju et al, 2022), as well as camera and 3D LiDAR (H. Kurita et al, 2022). This work presents a method for accurate mapping of forest environments using 2D LiDAR, stereo camera and wheel encoder. A flow diagram of the proposed method is shown in Fig. 1. We apply distance-based filtering to the LiDAR to extract only tree trunk information and restore it with position and radius data. The localization information of tree trunks is managed in real-time through a database. Thus, a tree map is generated using loosely coupled odometry from a LiDAR and a stereo camera. In addition, we use the tree position in the map as the landmark to reduce the localization error in the proposed SLAM algorithm. The structure of our paper is as follows. We introduce our work in five chapters: Chapter 2 provides a brief introduction to related research similar to this paper. Chapter 3 explains how the proposed methods are developed. Chapter 4 presents the experiments conducted. Finally, in Chapter 5, we describe the conclusions.

* Corresponding author

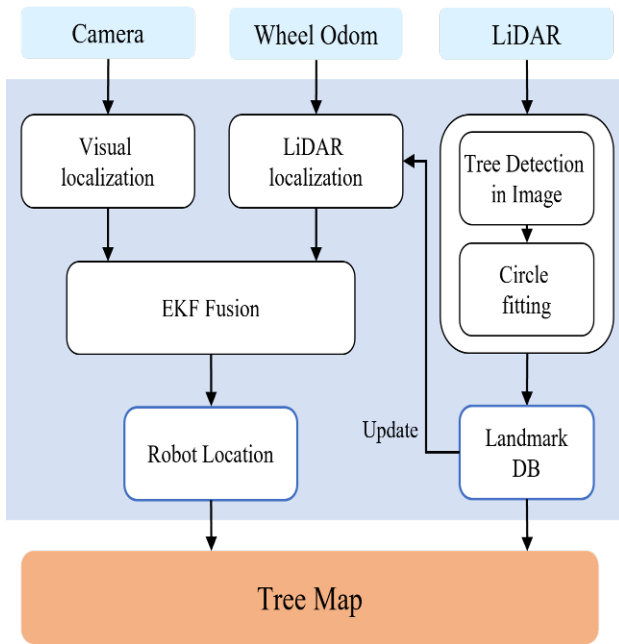


Figure 1. End-to-End System Flowchart

2. APPROACH

2.1 Multi-Sensor Calibration

There is the disparity in coordinate systems between the scan information of a 3D space captured by a 2D LiDAR and the 2D image planes obtained from a camera. This can be enabled through geometric correction to accurately project the 3D point capture by the 2D LiDAR onto the left camera of a stereo camera system. Fig. 2 illustrates the process, where the point $P(X,Y,Z)$ in the 3D space is projected onto the 2D camera image, resulting in the conversion to a point (u,v) . This demonstrates the transformation relationship between the 3D coordinates of the LiDAR and the camera. To achieve this transformation, a PnP algorithm is utilized to obtain a transformation matrix that accounts for the disparities in coordinate systems. The transformation matrix is derived using the information of the 3D points obtained from the LiDAR and the corresponding 2D image points. The internal parameters of the camera were obtained through calibration in advance.

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & f_{x0} & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1)$$

where f = focal length
 R = Rotation matrix $\in \mathbb{R}^{3 \times 3}$
 t = translation matrix $\in \mathbb{R}^{3 \times 1}$
 u, v = camera coordinate point
 X, Y, Z = LiDAR coordinate point
 s = scale factor

2.2 Generate tree trunk from LiDAR data

LiDAR is a sensor that is widely used for robotic systems for environmental perception and self-localization. To enable robots to operate effectively in forested areas and generate tree maps, the detection of trees becomes crucial. However, traditional 2D LiDAR systems, which emit laser beams in all directions to measure angles and distances from objects within a 360° field of view, often encounter challenges in distinguishing

trees from other forest elements such as fences, stones, and ground. These elements share similar characteristics in atypical forest environments, making accurate tree detection difficult. To address this issue, we propose a handcrafted tree detection algorithm using 2D LiDAR. This algorithm employs a filtering-based approach to extract tree trunk information from the LiDAR data. It is also able to operate efficiently in low-power and low-cost embedded environments. Moreover, since the algorithm relies solely on 2D LiDAR data, it also has the advantage that sophisticated camera- LiDAR synchronization is not required. The tree detection algorithm follows a step-by-step procedure to extract tree data in unstructured environments. Firstly, to focus on tree-related information, the algorithm selectively considers data within a 100° angle of view from the camera and within a 5-meter distance. Next, among LiDAR data, consecutive points based on distance are grouped. This grouping process allows for a coherent representation of objects and aids in identifying tree structures accurately. Finally, through filtering based on the width and length of the trees, tree column LiDAR data candidates are obtained. By setting a threshold of 0.55 meters for tree width, the algorithm determines viable candidates for tree column LiDAR data. Fig. 3 provides an example illustrating the selection of tree trunk data by applying distance-based filtering to the LiDAR sensor data. Overall, this approach ensures that only relevant tree information is extracted, leading to a refined dataset focused on tree structures.

2.3 Circle Fitting

To create a tree map, it is necessary to restore the occluded area of the tree trunk. The previously entered data through tree detection filtering is grouped and recognized as a single tree trunk. However, an error occurs because points of overlapping wooden pillars with similar distances are recognized as one pillar. In order to separate overlapping trees, we applied radius median filtering to filter only data within a specific distance centered on the median point of the grouped points. Through this, we separated overlapping trees by removing noise data. After that, the grouped points are restored to the wooden column using KASA circular fitting (Kåsa, 1976), and the location and radius of the central point of the wooden column are calculated. LiDAR is a sensor that has a high laser and can recognize objects at a relatively long distance. However, when recognizing a distant object, the circle fitting causes a large error due to the small number of corresponding points. The circle fitting error was improved by limiting the circle fitting to be performed only when the number of points recognized as trees was more than a certain number. At this time, the radius used for median filtering is 0.5m, and the number of points used for circle fitting conditions is 40.

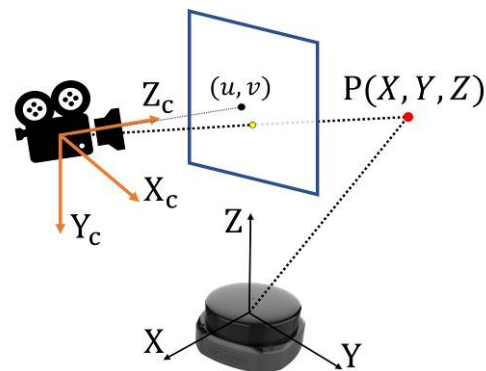


Figure 2. The 3D coordinate transformation relationship between 2D LiDAR and stereo cameras

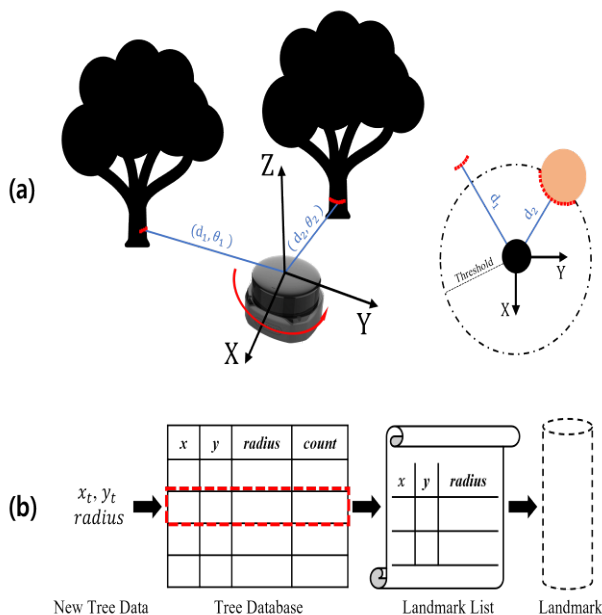


Figure 3. Landmark extraction process with (a) distance-based filter and (b) tree database

2.4 Combination of LiDAR SLAM and visual SLAM

We use a robot equipped with a stereo camera and a 2D LiDAR. As the SLAM algorithms, we employ VINS-stereo (Qin et al., 2018) and Cartographer (Hess et al., 2016) to utilize each sensor. However, each SLAM algorithm has its own advantages and disadvantages. Stereo camera-based SLAM calculates odometry by comparing and computing features between the surrounding environment in the images. Although the estimation of robot direction is accurate, the generated map may have a different scale from the actual object size. Cartographer captures the distance between the robot and surrounding environment by scanning lasers, obtaining 3D data of the environment. It can achieve precise SLAM but accuracy may decrease in environments that the ground is not planar. We assume that the robot moves on a planar ground. To combine the strengths of each sensor and compensate for their weaknesses, the EKF(Extended Kalman Filter) (Bellantoni and Dodge, 1967) is used to combine them in a loosely coupled manner.

2.5 Tree trunks as T-SLAM landmarks

The data received in LIDAR frame was filtered and restored as a tree trunk composed of information on the position of a tree, and the pose of the robot was estimated based on EKF between sensors. However, applying the tree data as-is for mapping the estimated position of the robot prevents the creation of an accurate tree map due to the accumulation of localization errors. In uneven terrain, the estimation of robot position using sensors is prone to accumulating localization errors, which arise from LIDAR sensor limitations in detecting terrain features beyond trees and the occurrence of slip phenomena between the wheels and the running floor. The accumulation of localization errors in the robot results in the fragmentation of the same tree into multiple segments in physical space, thereby impeding loop closure on the map. Moreover, recording all the detected tree data in each frame onto a 3D map leads to a proportional increase in data volume with operation time and mission space size, consequently hampering efficiency.

To address this issue, we introduce a tree DB(database) to generate only the most reliable trees by considering the

observed trees as the same entity, as illustrated in Figure 3(b). The tree DB is a data space that stores and manages location information and radius information of tree trunks recovered through circle fitting during the process of SLAM. The process of unifying tree location information using a DB is as follows. Initially, the location information and radius information of the tree trunk recovered through the tree observed for the first time is stored in the tree DB. The tree observed in the subsequent frame is considered to be the same tree if it falls within a specific range determined by comparing it to the tree stored in the tree DB. If it is recognized as the same tree, the position and radius of the existing tree are updated based on weighted averaging. The number of times a tree is recognized is also calculated and stored in the database. If it is not recognized as the same tree, new tree information is added to the DB. The weight averaging method is as follows.

$$\begin{cases} x_t = \frac{c_{t-1} * x_{t-1} + x_t}{c_{t-1} + 1} \\ y_t = \frac{c_{t-1} * y_{t-1} + y_t}{c_{t-1} + 1} \\ c_t = c_{t-1} + 1 \end{cases}, \quad (2)$$

During the robot's continuous movement, weight-based updates for tree positions are performed using the tree data obtained for each frame. If a tree falls outside the observation range of the robot, the update for that tree in the tree DB is halted. Subsequently, when the robot observes a new tree, the newly obtained tree data is not immediately stored in the tree DB. Instead, it is compared with the values stored in the existing tree DB based on distance. If the new tree falls within a certain range, it is recognized as the same tree, and the tree position stored in the tree DB is updated based on weighted averaging. Conversely, if the new tree falls outside the range, it is identified as a new tree and added as a new entry in the DB. The tree DB, where tree locations are updated using weight averaging, is employed to generate a tree map based on the recognition count for each tree. When the recognition count of a tree stored in the DB exceeds a specified threshold, the location information of that tree is considered highly reliable and updated in the landmark list. Tree data stored in the list is used for visualizing the tree map. At this time, the radius used for median filtering is 0.5m and the recognition count threshold is 20.

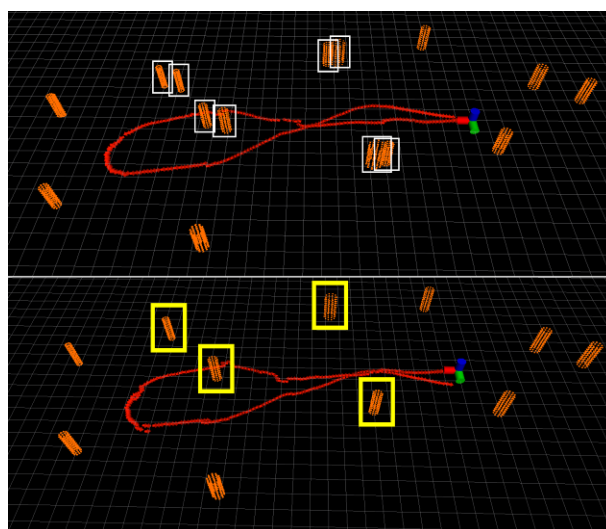


Figure 4. Example of tree DB usage for tree trunk position unification

Fig. 4 presents an illustrative example of unifying the locations of identical trees using a tree DB and generating a tree map. In this particular example, the robot detects the same tree on two occasions and follows a path back to its starting point. For the first tree observed while in motion, the tree's location information, acquired through frame-to-frame comparison, is updated in the tree DB. In Figure 4(a), the positional information of the same tree observed during the robot's return journey is calculated. However, due to cumulative odometry errors, the tree is visualized at a different position from what is stored in the tree DB. The trees observed are marked with white bounding boxes. In Figure 4(b), the location information of the tree observed during the process is compared with the stored value in the tree DB. If the observed tree falls within a certain range of the stored value, it is recognized as the same tree, and the location in the existing tree DB is updated accordingly. The unified tree is highlighted with a yellow bounding box.

Additionally, the restored position of tree trunks in the list can serve as landmarks in the global coordinate system, indicating the tree location. These tree landmarks are converted into landmark messages containing the tree location, pose, and unique ID of each tree. They are then incorporated as an input to LiDAR SLAM. The tree location is derived from the information in the list, and individual IDs are assigned based on location information. Since the tree trunk orientation cannot be discerned as it always faces the sky, it remains constant regardless of the robot's observation direction or position relative to the tree trunk. By employing landmarks as inputs in LiDAR SLAM, errors in loop closure are corrected, enhancing the robot's localization performance and map creation capabilities during the optimization phase of LiDAR SLAM.

2.6 Tree map Generation

Accurate tree maps are essential for enabling robots to successfully carry out missions and navigate their environments. As part of the preceding processes, the calculation of 3D robot position estimates and tree locations from a list of landmarks was carried out, which are essential for map generation. However, it should be noted that the landmark list stores the location of observed tree data in the LiDAR coordinate system. In order to create a 3D map using tree data, LiDAR coordinate system data is converted into robot coordinate system. Subsequently, the trees are mapped to the EKF odometry location to ensure precise alignment within the map. Equation 3 is the formula for this. $\mathbf{P}_L = (x_L, y_L, 0.1)^T$ represents a LiDAR data point defined in the 2D LiDAR coordinate system. \mathbf{T}_L^B is a 4x4 matrix representing the transformation between the LiDAR sensor and the base. The calibration between these two sensors is mechanically established. \mathbf{T}_B^W is a 4x4 matrix representing the transformation between the base and the world coordinate systems. Finally, $\mathbf{P}_W = (x_W, y_W, z_W, 1)^T$ denotes the \mathbf{P}_L point, which is measured by the 2D LiDAR based on the world coordinate system.

$$(\mathbf{P}_W = \mathbf{T}_B^W \times \mathbf{T}_L^B \times \mathbf{P}_L), \quad (3)$$

The resulting \mathbf{P}_W point represents the position of the tree trunk within the world coordinate system. To visualize the tree map, a point cloud is generated using the position and radius information from the landmark DB at the corresponding location. In this visualization, the tree trunk is assumed to have a height of 1 meter. The circumference of the circle, centered on the location of the tree trunk, is represented by 10 samples. Additionally, the height of each tree trunk is augmented from 0 meters to 1 meter at intervals of 0.1 meters, resulting in a total of 110 points per tree trunk for representation.

3. EXPERIMENTS

3.1 Configuration and environment

To evaluate the performance of the proposed method, we acquired the data by equipping a robot with a stereo camera and LiDAR sensor, as depicted in Fig. 5. The specification of each sensor utilized in the mobile robot platform are shown in Table 1. The stereo camera employed is ZED2 which offers a field of view of 110 degrees. To capture image input, the camera is configured to obtain color images for each lens, with a resolution of (672x376) pixels and 10 fps. The 2D LiDAR we applied is RPLIDAR S2, with operates at a 0.12° angle discrimination rate and 20 fps. For this experiment, we use Clearpath Husky UGV robot platform, capable of operating on uneven terrain and accommodating experiments of varying sizes and paths. The main embedded board is NVIDIA Jetson Xavier NX, responsible for collecting data from the sensors, executing the tree detection algorithm, and simultaneously operating the SLAM algorithm for tree map. Camera and LiDAR are connected to the embedded board via a USB connection, and the UGV communicates wheel encoder data through a serial connection.

The robot operates at speeds ranging from 0.1 m/s to 0.3 m/s during data collection. We conduct the experiment in the forest area of the school campus, as shown in Figure 6. Both experimental spaces measure 20 meters in width and 20 meters in length.

3.2 Experiment

Since we do not have exact GT(Ground-Truth) information for each trajectory in the forest, we establish a prerequisite. The prerequisite assumes that a trajectory starting and ending at the same line ensures loop closure. To fulfill this prerequisite, we create a start line by aligning the chess board on the side of the wheel, as shown in Fig. 6 (a). The robot proceeds forward to collect data, exploring the environment until it returns to the same location beside the chessboard, marking the end of data collection. We calculate the arrival point from the data and measure the distance by comparing the start point and end point, as depicted in Figure 6(b). The error value represents the Euclidean distance of each point in the xy plane.

Type	Robot hardware component
Robot Platform	CLEARPATH Husky UGV
Embedded board	NVIDIA Jetson-Xavier NX
2D LiDAR	SLAMTEC RPLIDAR-S2
Stereo Camera	STEREOLABS ZED 2

Table 1. Robot hardware spec sheet



Figure 5. Description of mobile robot platform used in the experiment

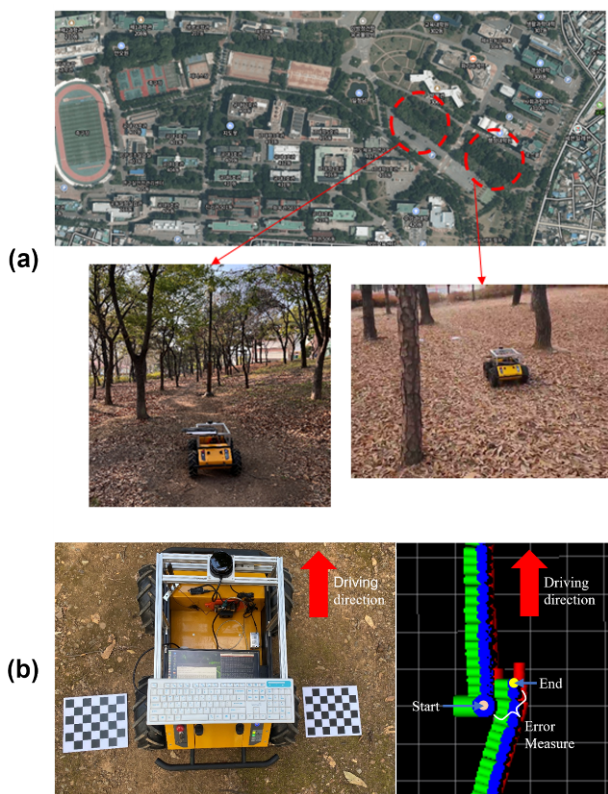


Figure 6. (a) Test field (b) loop closure error measurement from start to end point

Figure 7 shows the tree map results for multiple trajectories. The robot's path is denoted by a red line, while the observed tree trunks are represented as orange cylinders with a height of 1m. Figure 7(a) illustrates an experiment conducted to compare the effectiveness of tree landmarks. The robot's path is designed to enable multiple observations of the same tree. The tree map demonstrates the trees being managed as a list, ensuring the absence of duplicated instances of the same tree visualized in different locations. By utilizing the tree trunks as landmarks, the loop closure error is reduced to 0.56 meters, compared to 0.656 meters without tree landmarks. The loop closure errors, reported from top to bottom, are: 0.56 meters, 0.36 meters, 0.348 meters, and 0.413m, respectively. It is found that that the loop closure errors range from 0.3 to 0.5 meters.

4. CONCLUSIONS

We applied stereo camera and 2D LIDAR to organize the platform. To address the complexities of the forest, we propose a distance-based filtering method to extract data consisting solely of tree trunk information from 2D LiDAR. We also estimate odometry using EKF and generate a tree map based on observed trees. Experimental results demonstrate that the loop-closure error ranges between 0.3 to 0.5 meters.

Prior work documents the effectiveness of mapping using multiple sensors and reducing loop-closing errors. Other projects have explored the use of GNSS for improved localization performance. However, these studies have not focused on the cost-performance trade-off. The sensors and platform used in these studies are expensive for customers to purchase. In this study, we apply an algorithm that offers relative low cost while maintaining high performance. We find that mapping in the forest using stereo cameras and 2D LiDAR allows for the detection of tree trunks while operating. We also use the location of detected tree trunks to improve loop-closing accuracy.

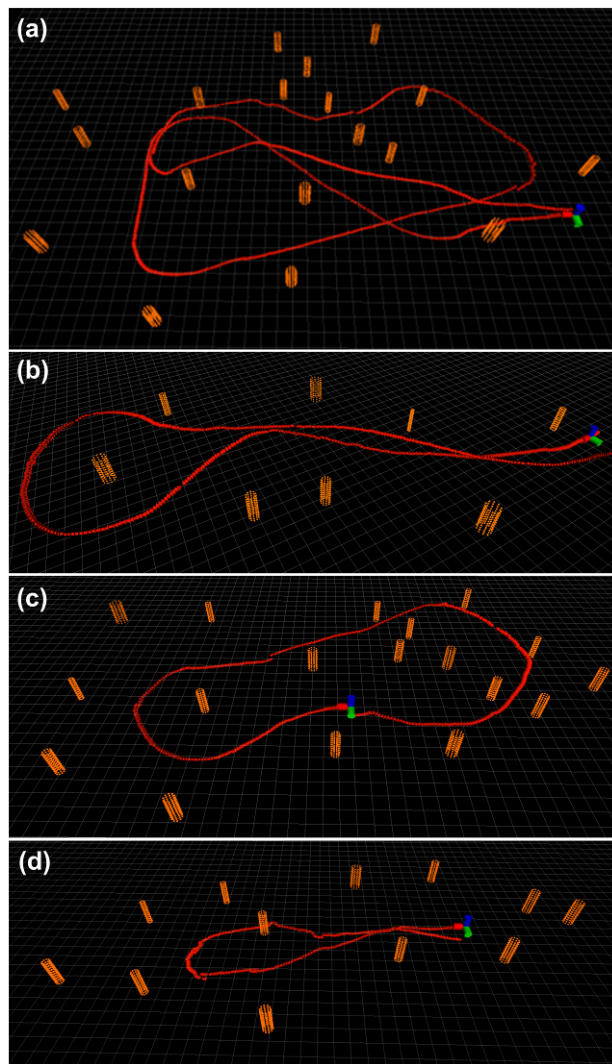


Figure 7. Examples of tree map visualized with tree landmarks

Although our study results show improvements in loop closure, there is limited space available for experimentation. Therefore, it is necessary to conduct experiments in larger and more expansive areas to effectively address loop-closing errors.

ACKNOWLEDGEMENTS

This work was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education (No. 2021R1A6A1A03043144)

This work has supported by the Development of SLAM Technique for Orchard Environment based on Stereo Camera-IMU Fusion, grant funded by the Electronics and Telecommunications Research Institute (ETRI) in 2023

REFERENCES

- Dissanayake G, Newman P, Clark S, Durrant-Whyte H, Csorba M., 2001. A solution to the simultaneous localization and map building (slam) problem. *IEEE Transactions on robotics and automation*, 17(3), 229-241.
- Aguiar, A.S.; dos Santos, F.N.; Cunha, J.B.; Sobreira, H.; Sousa, A.J., 2020. Localization and Mapping for Robots in Agriculture and Forestry: A Survey. *Robotics*, 9(4), 97

Garforth, J., and Webb, B. 2019. Visual appearance analysis of forest scenes for monocular SLAM, *International Conference on Robotics and Automation (ICRA)*, 1794–1800.

Han, J. h., Park, C. h., Kwon, J. H., Lee, J., Kim, T. S., Jang, Y. Y., 2020. Performance Evaluation of Autonomous Driving Control Algorithm for a Crawler-Type Agricultural Vehicle Based on Low-Cost Multi-Sensor Fusion Positioning. *Applied Sciences*. 10(13):4667.

S. Liu, X. Wang, S. Li, X. Chen, and X. Zhang, 2022. Obstacle avoidance for orchard vehicle trinocular vision system based on coupling of geometric constraint and virtual force field method. *Expert Systems with Applications*, vol. 190, pp. 1–13.

Li, S., Kang, D., Lee, H., An, S., Kwon, W., & Chung, Y., 2020. Development of Unmanned Driving Technologies for Speed Sprayer in Orchard Environment. *IEMEK Journal of Embedded Systems and Applications*. 15(6), 269–279.

Cho, Y., Woo, S. Y., Seong, S. H., Hong, H. G., Yun, H., Oh, J. Y., Kim, J., Kim, D. W., 2020. A Study on Modular Agricultural Robotic Platform for Upland. *Journal of Korea Robotics Society*. The Korea Robotics Society.

Mao, Wenju, Heng Liu, Wei Hao, Fuzeng Yang, and Zhijie Liu, 2022. Development of a Combined Orchard Harvesting Robot Navigation System. *Remote Sensing* 14, no. 3: 675.

H. Kurita, M. Oku, T. Nakamura, T. Yoshida, and T. Fukao, 2022. Localization Method Using Camera and LiDAR and its Application to Autonomous Mowing in Orchards. *J. Robot. Mechatron.*, Vol.34 No.4, pp. 877-886.

I. Kåsa, 1976. A circle fitting procedure and its error analysis. *IEEE Transactions on Instrumentation and Measurement*, 25(1), 8-14.

T. Qin, P. Li and S. Shen, 2018. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Transactions on Robotics*, 34(4), 1004-1020.

W. Hess, D. Kohler, H. Rapp and D. Andor., 2006. Real-time loop closure in 2D LIDAR SLAM. *IEEE International Conference on Robotics and Automation (ICRA)*, 1271- 1278.

Bellantoni, J. F. and Dodge. K. W., 1967. A Square Root Formulation of the Kalman-Schmidt Filter. *AIAA journal*, 5(7), 1309–1314.