2950 Niles Road, St. Joseph, MI 49085-9659, USA
269.429.0300 fax 269.429.3852 hq@asabe.org www.asabe.org

An ASABE Meeting Presentation
DOI: https://doi.org/10.13031/aim.202200999
Paper Number: 2200999

Simulation of an In-field Phenotyping Robot: System Design, Vision-based Navigation and Field Mapping

Zhengkun Li[a], Rui Xu[a], Changying Li[a, b], Longsheng Fu[c]
a. Bio-Sensing and Instrumentation Laboratory, College of Engineering, University of Georgia, Athens, Georgia, USA
b. Phenomics and Plant Robotics Center, University of Georgia, Athens, Georgia, USA
c. College of Mechanical and Electronic Engineering, Northwest Agriculture and Forestry University, Yangling, Shaanxi, China

Written for presentation at the
2022 ASABE Annual International Meeting
Sponsored by ASABE
Houston, Texas
July 17–20, 2022

ABSTRACT. *Cultivating high-yield and high-quality crops is important for addressing the growing demand in food and fiber from a growing population. In selective breeding programs, autonomous robotic systems have been gradually replacing the manual phenotypic trait measurements which are time-consuming and labor-intensive with relatively low temporal and spatial resolutions. In this paper, we present a Robot Operating System (ROS)-based phenotyping robot "MARS-PhenoBot" and demonstrate its reliable capacities including vision-based navigation and crop mapping through ROS-Gazebo simulation. MARS-PhenoBot is a solar-powered modular platform with a four-wheel steering and four-wheel driving configuration. Each wheel module is equipped with an independent suspension mechanism, making it adaptable to uneven field terrain. We developed a navigation strategy that relies on the crop row features extracted from images and exploits the prior knowledge of crop arrangement to autonomously navigate in cotton fields without any explicit map. For field mapping, a 2D map containing the distribution of weeds and cotton plants was developed based on a multi-object tracking algorithm and 2D-to-3D projection transform. The tracking algorithm that combines YOLO detector and DeepSORT tracker was applied to identify each plant in the field. In addition, a 3D mapping approach based on RTAB-Map SLAM method is deployed in our mapping architecture to generate the 3D map of cotton field. The simulation results show that our proposed navigation and crop mapping algorithms are capable of automated robotic phenotyping for field crops. The methodology developed in this study is scalable and can be easily deployed to the real agricultural robots to perform automated phenotyping tasks for crop field.*

Keywords. *Phenotyping robot; Navigation; Multi-object tacking; ROS; Gazebo; Simulation; Agriculture*

1. Introduction

The growing world population, climate change, degradation and loss of available land, and the increasing occurrence of new pests and diseases all threaten the world's food supply. It has been reported that rapidly rising demand in global food production requires a doubling of crop production yields by 2050 (Tilman et al., 2011). Recently, high-throughput and high-resolution phenotyping technology has been emphasized to product agricultural crops with high quantity and superior quality. This technology exploring how plants respond to environmental and genetic perturbations by identify and assess both simple and complex plant traits including, but not limited to, plant height, biomass, flowering time and grain yield (Shakoor et al., 2017). However, those phenotypic traits used to be measured manually by field technicians in the field, which is expensive because it is labor-intensive and prone to human measurement errors (Mahlein, 2016).

With the application of robotic technologies in agriculture, autonomous phenotyping platforms are usually designed as unmanned ground vehicles (UGV) or unmanned aerial vehicles (UAV) to complete specific collecting tasks for phenotypic traits with various sensors (Qiu et al., 2018). These phenotyping systems focus on either imaging the whole plant or the canopy using various cameras (such as RGB camera, hyperspectral camera, and thermal cameras)(Benet et al., 2018; Jiang et al., 2018; Ludovisi et al., 2017) or conducting plant 3D reconstruction with different range sensors (including stereo cameras, Time-of-Flight of light sensors, LiDAR sensors, and Computed Tomography sensors)(Bao et al., 2018; Bao et al., 2019; Mueller-Sim et al., 2017; Virlet et al., 2016). Varying the scale of phenotyping platforms allows for the precise and consistent data collection of single leaves/plant organs, individual plants, field plots, and full fields as required (Shakoor et al., 2017). UAV is typically applied to field scale monitoring, biomass estimation and identification of crop diseases over large growing regions. UGV can measure more detailed growing traits for single plants including, but not limited to, plant height, plant diameter, and plant architectureare.

In this study, we focus more on the mobile planforms which allow proximal sensing on individual plants for more detailed observations and fine-grained phenotypic traits. Its advantages in flexible design, large sensor payload, and high sensor resolution attract researchers and companies to customize platforms for specific phenotyping tasks. Some examples of such systems include "RoboHortic" (Cubero et al., 2020), Robotanist (Mueller-Sim et al., 2017), Vinobot (Shafiekhani et al., 2017), and "PATHoBot" (Smitt et al., 2020), along with more scalable modular platforms such as Thorvald (Grimstad et al., 2017) and MARS (Xu and Li, 2022). Compared with traditional agricultural machinery, these lightweight agricultural robots can be autonomous and intelligent, so they are less dependent on labor. Also, they create little or no soil compaction and are less limited by field conditions than heavy tractors. The advanced algorithms are deployed on these platforms to perform field phenotyping tasks, such as plant counting (Zhang et al., 2020), stem traits estimation (Xiang et al., 2020; Young et al., 2018), and leaf traits estimation (Vijayarangan et al., 2018).

Field phenotyping robots need to work in an uncontrolled environment or semi-structured environment, which requires higher intelligence and autonomy in navigation. One of the common solutions is to drive the robots using global navigation satellite system (GNSS) and inertial measurement unit (IMU) and avoid obstacles using the imaging sensors or LiDAR. However, accurate navigation relies on the costly high-performance sensors, for example, RTK-GNSS, which limits the further extensive application of field robot. Several researchers are interested in the relatively cheap and flexible camera sensors. Vision-based guidance systems are developed for controlling the robot to navigate the robot in the field (Ahmadi et al., 2021; Ahmadi et al., 2020; Li et al., 2018). These approaches exploit the crop-row structure in the fields as the control signal for the robot. Several studies aim to improve the robustness of crop-row detection, such as the tolerance to weeds (English et al., 2014), small crops (Winterhalter et al., 2018), and changeable illumination (Jeon et al., 2011).

Agricultural field mapping is another important goal for phenotyping robots and has become increasingly important to monitor fields and manage future activities. Most of the current field mapping research relies on remote sensing technologies to capture the whole field with a large view, focusing on the measurement or estimation of the integral properties, such as soil health and nutrition, biomass, disease, and stress resistance (Atefi et al., 2021). For the ground mobile robot, the limited views can only cover its surrounding area. Existing approaches are limited without considering the spatial connections between the current local scene to others. The current approaches are far from meeting the needs of phenotyping tasks targeting long time-span and large-scale areas (Chen et al., 2021). Thus, the large-scale mapping technologies is necessary for deploying phenotyping tasks in the field. With the advancement of simultaneous localization and mapping (SLAM), Visual SLAM technology has the most potential to complete a large-scale phenotyping task. It can construct accurately a large-scale scene, track the position of the vision system in the global map, and estimate the motion trajectory. Several V-SLAM frameworks such as ORB-SLAM2, RTAB-Map have been used to generate the 3D mapping for agricultural tasks (Dong et al., 2020; Fan et al., 2018; Habibie et al., 2017; Nellithimaru and Kantor, 2019).

In this study, we present a Robot Operating System (ROS)-based phenotyping robot, the "MARS-PhenoBot," with which we attempt to develop pipelines to support the robot's autonomous navigation and perform field mapping simultaneously. Only a cost-efficient camera is used to perceive the surrounding environment as the information source to the navigation and mapping pipeline. The proposed pipelines are tested in a Gazebo simulator to evaluate the performance. The specific objectives are to 1) develop a phenotyping robot to perform field phenotyping tasks, 2) design a vision-based navigation

algorithm and evaluate its efficacy; 3) explore the field mapping pipeline for 3D crop mapping in the field.

2. Methodology

2.1 Phenotyping robot "MARS-PhenoBot" design

In our previous work, we developed a universal robotic system architecture and the resulting modular agricultural robotic system (MARS) (Xu and Li, 2022). It allows various combinations of the hardware and software modules to create different robot configurations for specific agricultural tasks. Based on the modular design concept and architecture, MARS-PhenoBot was developed for cotton field phenotyping and management (Figure 1(a)). MARS-PhenoBot is a set of hardware modules and robot configurations using low-cost, off-the-shelf parts and 3D printed parts. It consists of frame module, wheel module, power module, sensing module, and controller module. The robot frames are made of aluminum extrusions, which allows to maintain a flexible frame dimension in both width and height via sliding mechanisms for adapting the different crops planting situation. Four steering wheels and four driving wheels were configured as the wheel modules for maximizing the maneuverability and payload. It can support the robot's movement along any direction to meet the unstructured field environment. Independent suspension mechanisms are used for the connection between the frame and each wheel module, increasing the trafficability on the uneven terrain. The power module includes the solar panel, charger controller, battery management system, and other electronic components that protect the battery and load. Its solar panels can be easily mounted to the top of robot's frame, and its area can be customized according to the real power of robot. Three cameras (called *FrontCam*, BackCam, and *BottomCam*, respectively) are equipped as the sensing module of MARS-PhenoBot. Among the three cameras, *FrontCam* and *BottomCam* are the RGB-D cameras, Kinectv2 (Microsoft, Redmond, WA, USA), while the *BackCam* is general RGB cameras (Logitech, Lausanne, Switzerland) The tilt angle $\theta_{FrontCam}$ and $\theta_{BackCam}$ are the angles between the z axis of camera and the horizontal lines, respectively. The *BottomCam* is pointed at the ground to observe more details of field. A high-computational onboard computer is used as the controller module to communicate with other hardware and run the control algorithm and perception algorithm.

The control software of MARS-PhenoBot is developed based on Robot Operating System (ROS) architecture, as illustrated in Figure 1(b). It can control robot to perform various operations with controlling algorithms based on the observation of robot itself and surrounding environment. In the hardware layer, there are three subsystems: a solar power system, a mobile platform system, and a sensing system. The charger controller in solar power system is used to collect the battery state and charging state as well as receive the controlled signals from the ROS master to realize the power management. The mobile platform with four-steering and four-driving configuration could be controlled based on the kinematics and motion controller we developed (Xu and Li, 2022). The data fused the encoders data from each motor and the inertial measurement unit (IMU) can be used to estimate the robot's location and pose. In the sensing system, the three cameras would perform in different pipeline. Both *FrontCam* and *BackCam* are used to navigate along the crop row in vison-based pipeline, while *BottomCam* works on crops detection, counting and crop distribution map generation. Additionally, *FrontCam* was used for a 3D mapping pipeline to generate the 3D field map.

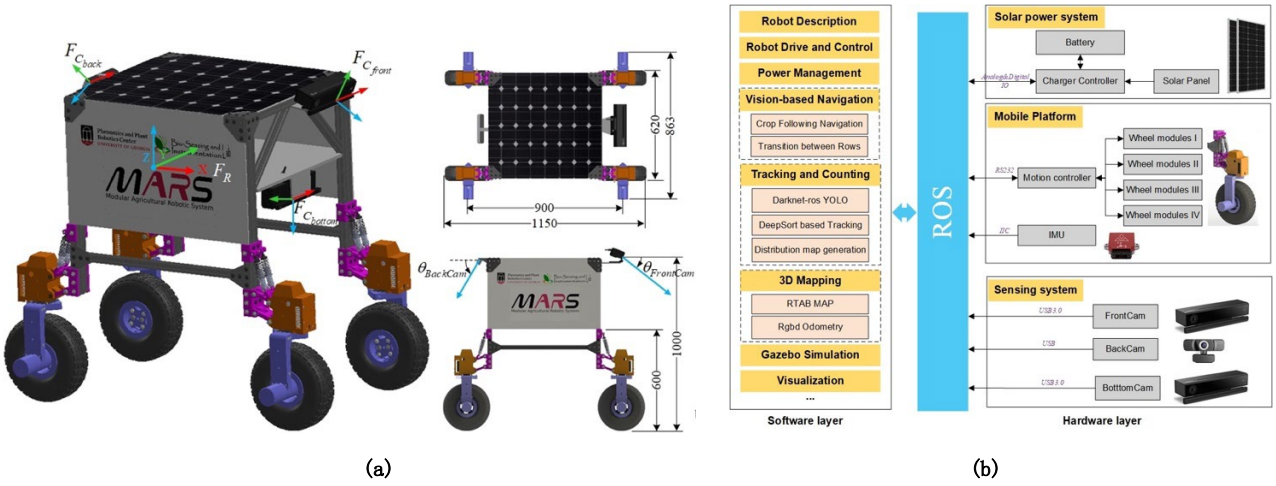


Figure. 1. Overview of phenotyping robot "MARS-PhenoBot": (a) the mechanical design, (b) ROS based software architecture

2.2 Simulation world generation

Before demonstrating in the physical world, the algorithms must be evaluated and optimized in the simulator first. One of the challenges is to create a high-fidelity interaction environment for the robot, which is fairly important for narrowing the gap between the simulation and reality. In our study, the Gazebo simulator was chosen to simulate the phenotyping robot in a cotton field for evaluating the developed algorithm. One of the preliminary works is to generate the simulation environment. There are two challenges: rich high-fidelity model and easy-configuration generator. The realistic models ensure the validity from the simulation to the real world, while the world generator allows to configure and modify various simulation environment fast and easily.

In Gazebo simulator, SDFormat (Simulation description format), an XML format is used to describe objects and environments for robot simulators, visualization, and control. In the description files, all the elements (or named models) including robots, lights, sensors, and static objects, are described with various tags. For example, the cotton plant models, can be described with "visual" tags to define the visual elements, including the pose, geometry, materials, shadows, etc. Our study aimed to develop a script to generate the world description files through customizing different tags for objects.

Figure 2 shows the proposed Gazebo world generation pipeline. It allows the user can easily change the components in simulation and focuses on algorithm evaluation rather than the trivial environmental definition. Generally, a planting field can be regarded as the combination of various types and numbers of plants and soli with a specific distribution rule. We build a field model library to support the diversity of the simulation world. there are two sources to acquire the 3D models: one is mesh files generated from the high-resolution point cloud, and another is the modified models based on the open-source model libraries (such as such as RenderHub, Turbosquid, and CGTrader). In our previous works (Sun et al., 2021), FARO LiDAR (FARO Technology, USA) is used to collect the high-resolution point cloud from the cotton field and greenhouse. Then, the mesh of single plant or soil could be extracted with the CloudCompare or MeshLab software, and the processed output would be further modified with the CAE software such as SolidWorks and Blender.

After generating the sufficient models in libraries, we developed a Python-based script to generate the Gazebo world files with user-defined configuration files. The configuration files describe the basic parameters of the field, including the used model list and the distribution rules of plants. Considering the more realistic application, noise is added in the location of each plant. For example, adding the gaussian noise to the cotton plants' location is closer to mechanized planting situation. Additionally, the pose of each plant (cotton plants and weeds) was randomized to increase the diversity of scene.

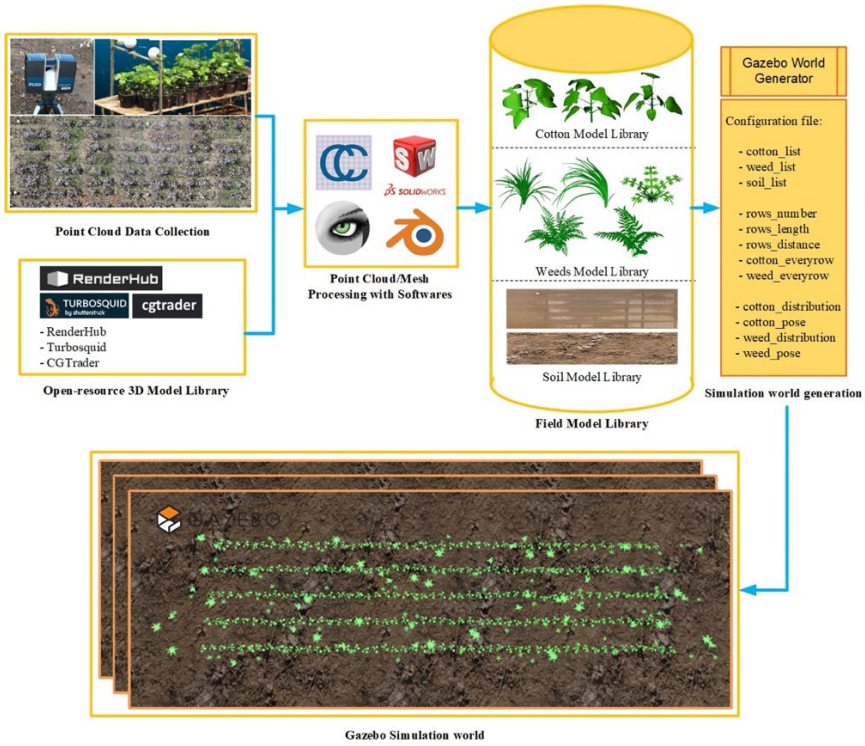


Figure. 2. Gazebo simulation world generation pipeline

2.3 Vision-based navigation for cotton field

We propose to use a vision-based navigation pipeline that suitable the crop row planting field. The pipeline relies on the visual features extracted form images and exploits the prior knowledge of crop arrangement to autonomously navigate in cotton fields without any explicit map. The main idea is to deploy the line-following algorithm to follow the path of crop

row based on the image processing result of camera view, and then register the switch to the next row through integrating the prior knowledge of cotton field and the heading of line-following stage.

2.3.1 Scheme for Autonomous Navigation in Cotton Fields

In cotton fields, cottons are arranged along multiple parallel straight lines with the mechanized planting pattern. We took advantage of such an arrangement to enable a mobile robot to navigate autonomously when performing the phenotyping task in the field.

Figure 3 illustrates the vision-based navigation in the entire field. The field can be divided into multiple same segments and each segment consists of three stages: line-following with *FrontCam* (stage ①, cyan lines), line-following with *BackCam* (stage ②, yellow lines), and transition between rows (stage ③, white lines). This sequence of behaviors is repeated to navigate along all the rows in the field. We accomplish this by using a vision-based navigation algorithm that integrates all these behaviors and leads the robot to navigate autonomously and collect the phenotyping data.

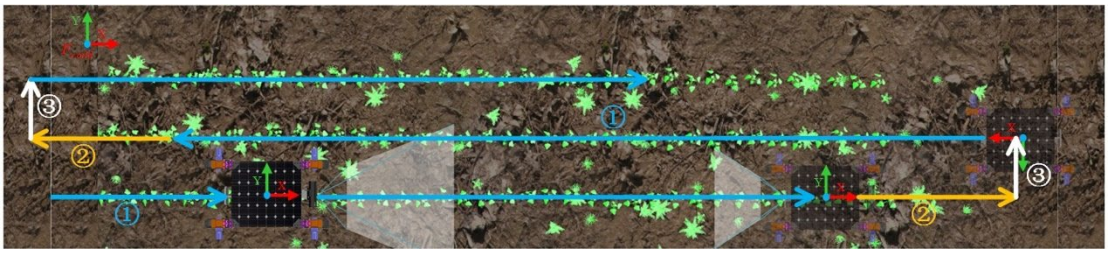


Figure. 3. Scheme for navigation in a cotton field: the robot enters the field and navigates along a cotton row with *FrontCam*' view (①, cyan lines), approaches the end of the row and navigates along a cotton row with *BackCam*' view (②, yellow lines), transits to the next crop row (③, white lines).

Our scheme to navigate in the cotton field is illustrated in algorithm 1. It relies on the image data from *FrontCam* and *BackCam* as well as the heading data from the IMU sensor (line 1). We developed a customized line-following controller based on PID algorithm. It uses the crop-row detection results based on the probabilistic Hough transform as the feedback variables to directly control the angular velocity of the robot. The detailed illustrations of the line-following controller and crop row detection algorithm are in sections 2.3.2 and 2.3.3, respectively. The navigation scheme also requires the field arrangement information (number of the rows) to determine the stopping time.

Algorithm. 1. Field navigation scheme

```
1: Initialize (FrontCam, BackCam, IMU, RowNumber, CountRow)
2: repeat                                   ▶ Control loop
3:   CountRow +=1
4:   CropLine_FrontCam ← CropRowDetection (FrontCam)
5:   if IsEmpty (CropLine_FrontCam) then
6:     CropLine_BackCam ← CropRowDetection (BackCam)
7:     if IsEmpty (CropLine_BackCam) then   ▶ Stage III: Transition between rows
8:       RobotRotation (180°, CountRow, IMU)
9:       repeat
10:          RobotOffset (FrontCam)
11:          CropLine_FrontCam ← CropRowDetection (FrontCam)
12:      until CropLine_FrontCam             ▶ Enter next rows
13:    else                                  ▶ Stage II: Follow crop rows with BackCam
14:      LineFllowing (CropLine_BackCam)
15:  else                                    ▶ Stage I: Follow crop rows with FrontCam
16:    LineFllowing (CropLine_FrontCam)
17: until CountRow == RowNumber
```

In the navigation control loop (line 2), we first deployed the customized crop row detection algorithm to the images captured from *FrontCam* (line 4). If it detects a crop line, the robot enters stage ① and navigates along the crop row with the line-following controller (line 16). As it approaches the end of the row (stage ②), the FrontCam no longer detects the crops, since it is tilted to look forward and thus, the robot has no path to follow. At this position, the BackCam can replace the FrontCam to recompute the detected crops to provides the references for the navigation controller (line 6 and line 14). This setup guides the robot to remain in the crop row until the BackCam no longer detects the row. Then the robot enters the transition stage ③ to move to the next crop row. In the transition stage, the robot first rotates 180° (line 8) and then moves among the orientation perpendicular to the crop row (line 10) until the next row is detected by the *FrontCam*. The rotational

action relies on the real-time heading update based on the IMU data to ensure the accurate rotation angle. After the robot rotates, movement to the next rows is estimated using the heading data during the entire line-following stage (stage ① and ②). First, the heading data is smoothed with a first order Kalman filter, and then the smoothed data is averaged to obtain the moving direction. After rotation, there is the obvious risk that the robot mistakes the current row as the next one if it only relies on the crop row detection. Our strategy was to skip the first several interactions and narrow the region of interest (ROI) to ensure the correct row are detected.

During the navigation in the entire cotton field, the robot repeatedly operated the control loop (line 3 – line 16) in each crop row until the accumulated number of rows is equal to the pre-defied number of field rows.

2.3.2 Line-following Controller for Crop Row

Our vision-based navigation pipeline relied on visual serving control to guide the robot to follow crop rows, as illustrated in Figure 4. It can be regarded as a line-following algorithm on the basis of the feedback control for the detected crop row. The crop row detection process can be regarded as a sensor that constantly generates signals that describe the error offset from the reference, including the angular offset θ and the horizontal offset L of the detected line. the angular offset θ is the angle of the detected crop line in the vertical orientation, while the horizontal offset L is calculated by the difference between the central point of image and the central point of detected line. The signal as the input to a PID (proportion, integration, and differentiation) controller to control the robot's velocity and movement.

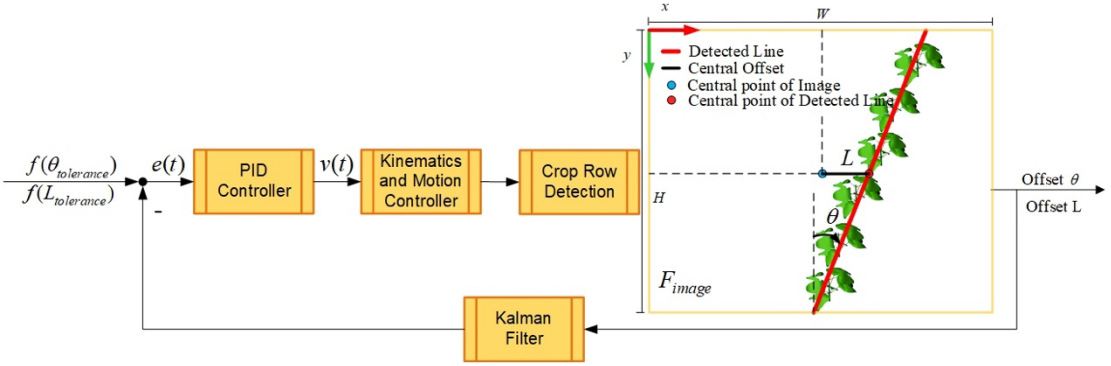


Figure. 4. The proposed visual servoing controller based on the line-following algorithm. It uses a PID controller to control the robot's velocity for eliminating the offset which is calculated by image processing approach

In real applications, the detected offset is not stable, relying on the image quality (e.g., illumination) and the robustness of crop row detection algorithm. The unstable signals cause the too high-frequency velocity adjustment rather than follow the path stably. Two tracks are used in the feedback control: a Kalman filter for smoothing the offset signal and the offset tolerance regulation to constrain the velocity adjustment. The Kalman filter algorithm can estimate the error and generate closer to real measurement, while offset tolerance method allows robot to adjust the velocity only when the offset is large enough. In another word, the control goal is to ensure the offset in the offset tolerances. Once the control error is calculated, a PID controller will be used to output the control the velocity. Because the system responses directly depend on the coefficients of controller, the reasonable parameters estimation is significant. In our work, we set the linear velocity of the robot as a constant value on the x axis, and only the z component of the angular velocity is controlled with a P controller.

2.3.3 Crop Row detection

To navigate along a crop row, we extracted the line along which the crops are arranged and then calculate the offset error for the line-following controller. The crop row detection algorithm was developed based on the image processing technologies and can work on both the *FrontCam* and *BackCam* with minor changes.

Figure 5 illustrates an example of image processing sequence working on the captured image. As shown in Fig 5(a), the view of the *FrontCam* covers multiple crop rows and several weeds in the Gazebo simulation. As the *FrontCam* mounted on the central line of the robot, the central crop row in the image is the one need to follow. We performed a homography transformation for the original image (Fig 5b) to obtain an orthonormal perspective to ensure the rows were parallel in the image (Fig 5c). Then, the blur process was performed to remove salt and pepper noise to improve the signal-to-noise rate (Fig 5(d)) to suppress noise and preserve key information like contours and edges. The HSV (hue, saturation, value) threshold segmentation algorithm was used to segment the plants (cotton plants and weeds) from the soil background in the HSV color space (Fig 5(e)). The resulting image was subjected to the open and closed operations in morphological filtering to extend the pixels of the same crop rows, which integrated the separate plants of the same row with an integral object (Fig 5(f)).

Next, the probabilistic Hough transform is used to find all lines on the basis of processed image. It can filter the nonconforming detected line with the pre-defined parameters: the detected resolution in pixels and angle, the minimum number of intersecting points to detect a line, the minimum length of detected line, and the maximum distance allowed to

connected points in the same line. Fig 5(g) shows that all detected lines satisfied predefined conditions: all the crop rows were detected successfully, and there is not only one line in the same crop.

Thus, the final line followed could be extracted further on the basis of the detected lines. In a 2D plane, a line can be defined with a point and a vertical angle. For the followed line, we calculated the central points of the middle lines set as the intersection point and average the vertical angles of all the detected lines as the vertical angle. The final crop row detection result is shown in Fig 5(h): the red line is the followed line which is as the control signal for the line-following controller, and the white lines presents the following crop.

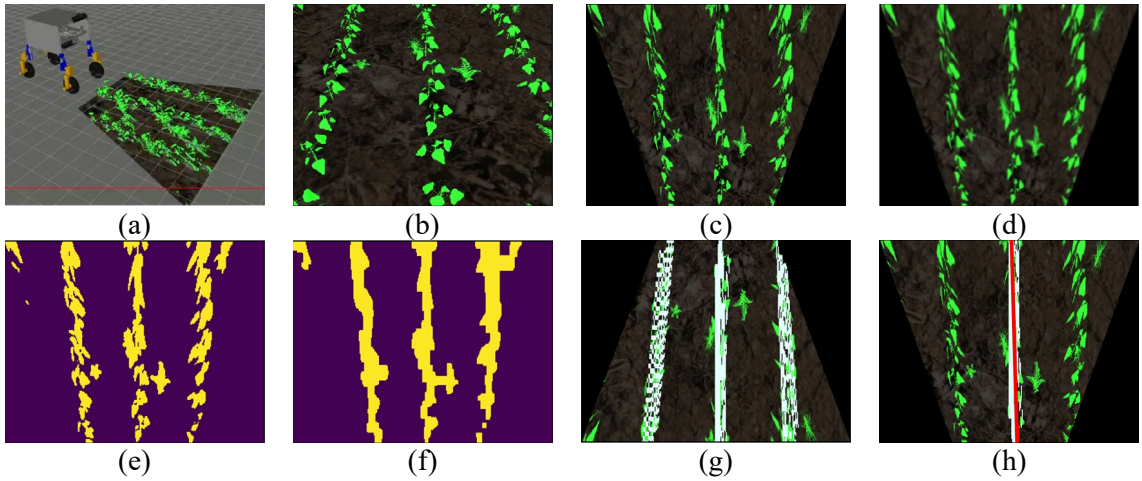


Figure. 5. The sequence of crop row detection process. (a)Gazebo Simulation, (b) Original Image, (c) Homography Transform, (d) Blur, (e)HSV Threshold, (f) Morphological Processing, (g) Probabilistic Hough Transform, (h) Crop Row Detection

2.4 Field Mapping for plant phenotyping

Compared with the UAV (Unmanned Aerial Vehicle), the UGV (Unmanned Ground Vehicle) is closer to the ground and plants and can acquire more details for individual-level morphological information with the same equipment. Our goal in this stage was to develop a field mapping pipeline for the MARS-PhenoBot to support plant phenotyping analysis. Two subfunctions are considered: 2D distribution map of crops and 3D mapping of the entire field. The mapping pipeline can work in parallel with vision-based navigation, which means the field map can be generated continually when the robot navigates in the field.

Our 2D distribution map was developed based on current advances in counting technologies applied to the yield estimation. Benefiting from the rapidly developing object detection and multiple objects tracking technologies, the counting technologies have improved significantly in accuracy, such as seedling counting, flower counting, and fruit counting, However, they only focus on the estimation of partial field or the entire field without any distribution information and limit further applications, such as the density map of flowers or fruits. In our study, we further extended the counting outcome to generate the distribution map by combining the location data from the robot.

Generating an accurate field map requires three important steps: perceive each plant, locate each plant, and analyze the properties of each plant. For a mobile platform, the limited view of the camera merely covers plants in the views. Among the successive image sequences, the same plants may appear in multiple frames. Thus, it is necessary to track each plant and give a special label for each single plant tracked over the image sequences. Once the tracking outcomes were calculated in the image sequences, the next step was to build the transform relationship between the image frame and the world frame, which requires mapping the 2D image frame to the camera frame, and then transforming the coordinates to the world frame from the camera frame. After the transformation, the 3D coordinates relative to the world frame of each plant could be calculated. Finally, the various properties of each plant could be further analyzed with a specific approach, such as plant types and its corresponding phenotyping traits. One of the benefits is it can build a detailed growing information library for each individual plant rather than that of a population in an area.

Figure 6 illustrates a specific case of our proposed the concept of distribution map. In our work, a DeepSORT (Simple Online and Realtime Tracking) (Wojke et al., 2017) based two-stage tracking architecture was used to track all cotton plants and weeds. DeepSORT is a tracking-by-detection approach, treating the multi-object tracking as a data association problem. In the detection stage, a lightweight end-to-end YOLO (You Look Only Once) (Bochkovskiy et al., 2020) based detector is used to detect crops and weeds for each image frame. In the data association stage, on the basis of the Kalman filter prediction and Hungarian algorithm matching, we increased an additional pre-trained Convolutional Neural Network (CNN) to provide appearance data to reduce the switching of the object's identities. Once it matches the detections, the information (ID number, class, and bounding boxes) of the tracked objects in each frame would be updated.

The *BottomCam* (Kinectv2 RGB-D camera) is used to collect the RGB image and depth image to register the point cloud.

The number of registered point clouds equals the image resolution which means any registered point can correspond to a single pixel. Thus, the 3D coordinate corresponding to each pixel in the image can be easily indexed. Furthermore, based on the frame transform matrix from the camera frame to the map frame, the registered point cloud can be updated to the world frame.

In each interaction, the real-time updated RGB image sequences are input into the DeepSORT based tracking workflow, and the information (ID number, class, and bounding boxes) of the tracked objects would be updated. We calculated the central pixels of all tracked objects in the image frame and then indexed their 3D coordinates in the world frame. When the robot works on the field, the transformed tracked information (ID number, class, 3D coordinates in the world frame) can record and update continuously until we stop the pipeline. For the same plant, it obviously can generate the transformed tracked information multiple times. We defined sets for the tracked objects that have the same ID number and average the items in the sets as the final update. For example, a cotton plant was tracked in three iterations and generated three corresponding, transformed, and tracked data with the same ID number. All the transformed tracked data was processed further to get final data. Finally, all plant sets were recorded and updated to the 2D crop/weed distribution map.

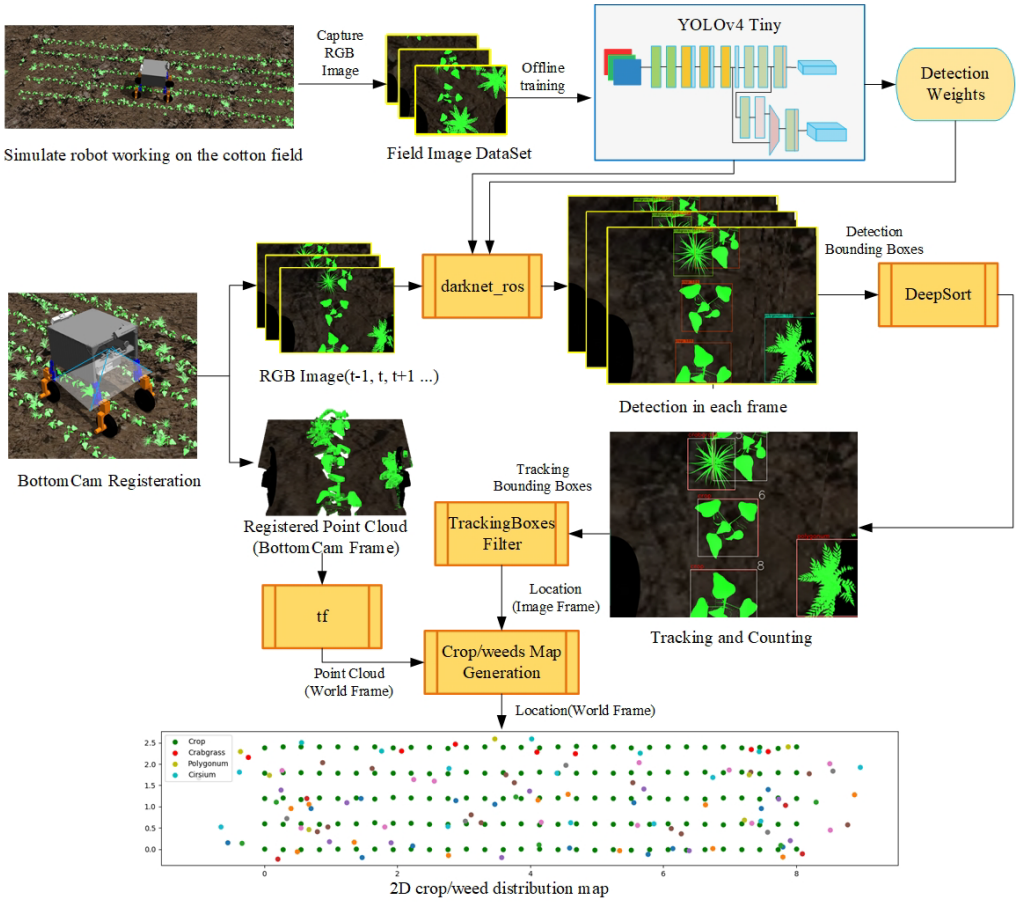


Figure. 6. The simplified flowchart of the generated 2D Crop/weed distribution map

Our proposed field mapping pipeline was deployed on ROS, as illustrated in Figure 7. Except for the 2D distribution map workflow, the SfM-based (Structure-from-Motion) a 3D mapping approach was also integrated to build the 3D reconstruction of the field map.

In ROS, the workflow can be divided into multiple independent nodes, and these nodes were into a graphed and communicated with one another using streaming topics, RPC services, and the parameter server. Also, the URDF (Unified Robot Description Format) file was used to define every part of the robot by links and joints, with each link and joint defining the various properties of the robot. The coordinate transform from one joint to another can be achieved easily with the tf node.

In the 2D distribution map workflow (left part), *BottomCam* was driven with the Openni_kinect driver and registered the RGB image and point cloud. *BottomCam* possesses a smaller view and can observes more details information with few disturbances by the background. The registered RGB image was input to the darknet_ros node to generate the 2D bounding box of detected plants. A customized lightweight YOLOv4-tiny detector was trained with the image dataset collected from the Gazebo simulation world. The DeepSORT-based tracking node described the detected bounding box data and published

the tracking box result. The map generation node combined the registered point clouds data, transform tree, and the tracking box results to generate the 2D distribution map.

In the 3D mapping workflow (right part), an open-source lidar and visual SLAM (Simultaneous Localization and Mapping) library, RTAB-Map (Real-Time Appearance-Based mapping) (Labbe and Michaud, 2019), was deployed to generate an accurate 3D map for the entire field. Because it had a larger view, we choose the *FrontCam* as the data source for capturing more accessible visual data. For more precise odometry estimation, an extended Kalman filter was used to fuse multiple source data: wheel odometry, IMU, and visual odometry. The visual odometry was generated from the registered RGB-D data of the *FrontCam* with the RTAB-Map visual odometry node. Additionally, the RTAB-Map library also provided the visualization tool, rtabmapviz, to visualize and optimize the real-time, updated mapping data.

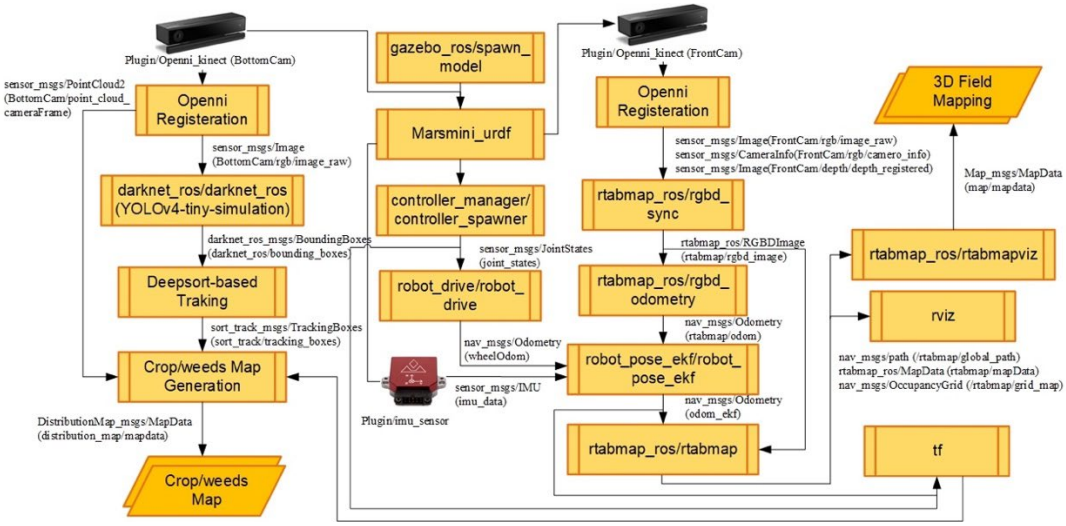


Figure. 7. The ROS node diagram of the proposed field mapping pipeline

3. Experiments and Results

In this section, we conducted Gazebo simulations to evaluate the vision-based navigation pipeline and field mapping pipeline. The Gazebo worlds were generated with our Gazebo world generator (section 2.2). Figure 8 shows the configurations of the Gazebo worlds. It consists of three types of cotton models and a soil model. We set 5 crops rows on the soil model with the 8 meters in length and 0.6 meters in distance between rows. In a single crop row, 40 cotton plants models are almost evenly distributed on the crop row (gaussian noise is considered). The type of a cotton plant models was selected randomly among the pre-defined model list, and their orientations were set randomly to increase the diversity of the field. Besides, three typical types of weeds (crabgrass, polygonum, and Cirsium) in cotton field were also considered for the robustness test.

Figure. 8. A case of Gazebo world for algorithm performance evaluation.

The MARS-PhenoBot is described with URDF file and loaded in to the two Gazebo worlds. The two tilt angles ($\theta_{FrontCam}$ and $\theta_{BackCam}$) of the *FrontCam* and *BackCam* were set as 60°and 30°. Additionally, the linear velocity of the robot was constant during the simulations, and was set to 0.2 m/s. With these basic parameters, the robot can be simulated in the Gazebo simulator and perform the navigation pipeline and field mapping pipeline simultaneously. All data from the running ROS system was recorded to rosbag files for analysis.

3.1 Performance of vision-based navigation

Performance evaluation: Our proposed vision-based navigation pipeline was tested in the simulated cotton fields. For

the crop-row following controller, we set the offset tolerance parameters the $\theta_{tolerance}$ and $L_{tolerance}$ as 8 degree and 30 pixels, respectively. Also, the PID controller is only pre-defined as a P-control mode with the proportional coefficient of 0.1. Based on the observation of the simulation, the robot successfully navigated the entire fields, relying only on visual crop features without any explicit input, such as GPS or landmarks. We assume the cotton field as a 2D plane, so the crop arrangement was able to display the x and y coordinates of the cotton plants. The location of the cotton plants and the robot were updated by the global Gazebo simulator, which can be regarded as the ground truth data. To evaluate navigation accuracy, we visualized the moving trajectory and analysis the path-track errors and heading errors during the whole.

Figure 9 illustrates the performance evaluation of vision-based navigation in a weedless cotton field. It can be observed that the robot was able to follow each of the rows successfully until the end and transition to the next one until it covered the whole field. The robot can track crop rows with a cross-track error less than 5 cm and heading error less than 1 degree for the most the path. Considering the geometric shape and volume of cotton plants, the cross-track errors are almost negligible when the robot's heading is stable. At the end of the row, the robot was able to transition to the next within an average maneuvering space of 1.19 m. Thus, our proposed vision-based navigation pipeline allows the robot to enter the next row using a limited maneuvering space, which is often a critical requirement while navigating in a field.

We also observed that errors suddenly increase when the robot enters to new rows but will decrease quickly once the following controller works. The cross-track error of the final row was larger than that of the other paths because the error tolerance of offset L was set to a large value to decrease adjustments to the frequency of velocity. According to our observation, our navigation scheme has an excellent performance in error tolerance and can work well if the cross-track error and heading error are lower than 20 cm and 10 degrees.

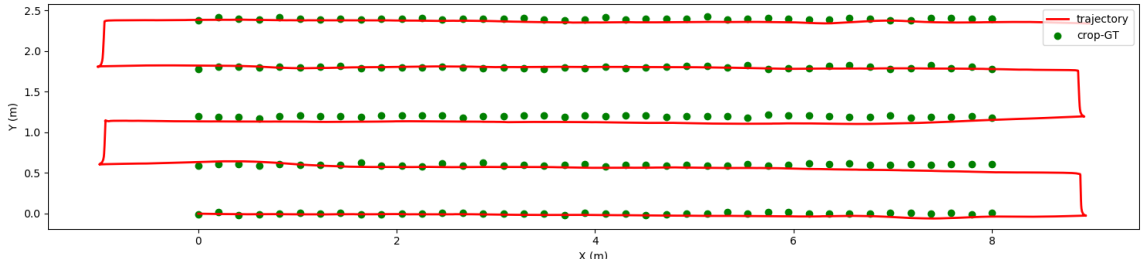


Figure. 9. Performance evaluation of vision-based navigation in cotton field.

Navigation pipeline Analysis: The proposed vision-based navigation pipeline was developed based on prior knowledge that the crop rows were arranged in multiple parallel lines. It exploits the row structure inherent in the crop fields to guide the robot along the crop row without the need for explicit localization systems, GNSS, or a map of the environment. The simulation experiments demonstrate the feasibility to use only on the local observations from the on-board cameras to navigate. Our pipeline allows the robot to successfully navigate through the crop fields row by row and cover the entire field.

However, there are still some limitations and further improvement for the vision-based navigation pipeline. First, the crop row segmentation relies on the HSV threshold algorithm. Although most the previous studies demonstrate its feasibility in real applications, it remains difficult to distinguish objects whose color are similar to the crops, for example, the weeds. Furthermore, the illumination variation also makes it risky for segmenting crops. Second, the offset signal extracted from the segmented has noticeable, high-frequency noise, which may mislead the controller to perform a mistaken velocity adjustment. Third, the setting of many pre-defined parameters requires expert experience, such as the HSV threshold, the constrains of Hough transform, and the coefficients of PID controller. For another new scenario, these parameters may need to tune for the best performance. Finally, failure in row transition is possible. The simulation results showed a high risk at the point of entering the next row because of suddenly increased track-path and heading errors.

One of our future studies will seek to improve the robustness of crop rows extraction. Conventional image processing approaches limited by imaging qualities, while recent advances in CNNs based feature extraction appear excellent generalization. We aim to develop a lightweight CNNs-based crop-row feature extractor and directly predict the offset parameters with these features using a Multi-Layered Perceptron. Furthermore, another potential study might seek to to decrease the pre-defined parameters and improve the level of generalization.

3.2 Performance of field Mapping

Once the robot starts to navigate in the cotton field, the field mapping pipeline is launched at the same time. We will evaluate the pipeline from the performance of each node: YOLOv4-tiny based detector, 2D distribution map based on DeepSORT and 3D mapping based RTAB-Map.

YOLOv4 tiny based detector: Considering the tradeoff between the detection accuracy and inference speed in a limited computational unit, we choose the lightweight YOLOv4-tiny as the detector. YOLOv4-tiny is the compressed version of YOLOv4, with the compressed CSP backbone with a total of 29 pretrained convolutional layers. Besides, the number of

YOLO layer is reduced to two and less anchor boxes for prediction are used.

Transfer learning was used to increase the learning efficiency and generalization as the training dataset was small. In this study, the YOLOv4-tiny model was initialized by weights pretrained on MS COCO dataset (https://cocodataset.org/), and then fine-tuned on our training set. Our dataset had 354 images collected from the Gazebo simulator. These images contained four types of objects: cotton plants, crabgrass, polygonum, and cirsium. We annotated the images with an online annotation tool (Roboflow) and then augmented them to 3x images with rotations, flips, shears, and exposures. The dataset was divided into three sections, including a training set, a testing set, and a valid dataset with the proportion of 7:2:1. Table. 1 shows the best performance of the trained weights. Because of the relative simplicity of the detected objects and background, the average accuracy rose to 99.49%.

Table. 1. Performance of YOLOv4-tiny model for cotton plants and weeds detection

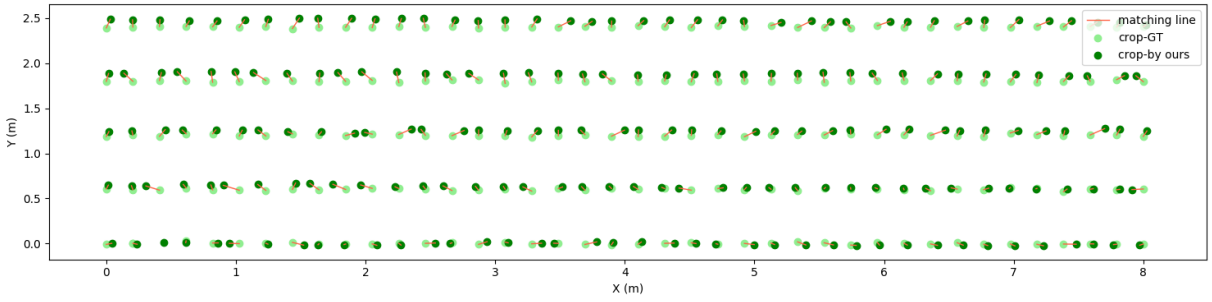
	Class	AP	Recall	F1 score	mAP@0.5
YOLOv4-tiny model	Cotton plant	99.98%	0.99	0.98	99.46%
	Crabgrass	99.83%			
	Polygonum	99.94%			
	Cirsium	98.08%			

2D distribution map based on DeepSORT architecture: Our proposed 2D distribution maps rely on the tracking outcome and the coordinate transform of the registered point cloud. DeepSORT is an open-source detection-based tracking architecture which uses SORT (Kalman filter) (Bewley et al., 2016) and ReID (identification model) to link the bounding boxes and tracks. We directly use the original architecture and deploy our customized YOLOv4-tiny detector to generate the bounding boxes.

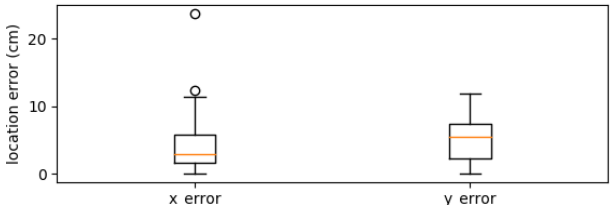
We test the 2D distribution map generation approach in the simulated cotton fields. For more intuitive comparison and analysis, we only considered the cotton plants as the tracking objectives. Figure 10(a) shows a case of the 2D crop distribution map generated by our proposed pipeline: green dots present the tracked cotton plants, light-green dots present the ground truth, and the tomato-red lines are the matching lines between corresponding plants. For each tracked plants (green dots), it requires to be detected, tracked, and filtered in image sequences, and finally the estimated location can be generated. It was observed that all the tracked plants are successfully matched with the corresponding ground-truth. That means our YOLOv4-tiny detector can successfully detect the cotton plants with 100% accuracy and the DeepSORT based tracker also can track 200 cotton plants successfully with 100% success rate.

Figure 10(b) shows the x component and y component of the location error: the mean location errors along the crop (x component) was 3.90 centimeters with the standard deviation of 3.21 cm, while the error perpendicular to crop rows (y component) was 5.09 centimeters with the standard deviation of 3.06 cm. Most of the Euclidean distance between the corresponding crops is less than 9 cm. Considering the volume and shape of cotton plants, the location error is perfectly acceptable in real application. For the more detailed error source analysis of the mapping pipeline, we discuss in the next section.

Obviously, it is difficult to achieve the 100% of success rate for detector and tracker in the complex real environment. But once the tracking works well, our proposed 2D distribution map generation pipeline can work with high accuracy.



(a)



(b)

Figure. 10. Performance evaluation of our proposed tracking-based pipeline: (a) 2D crop distribution map, (b) boxplot of location error

3D mapping： We are exploring the accurate point cloud registration for the entire cotton field with the MARS-PhenoBot. The implementation of real-time and online 3D mapping relies on the RTAB-Map library. It implements loop closure detection with a memory management approach, limiting the size of the map so that loop closure detections are always processed under a fixed time limit, thus satisfying online requirements for long-term and large-scale environment mapping.

Figure 11 illustrates the 3D map of the cotton field generated by our proposed mapping pipeline. In order to decrease the overlap point clouds of the same plant, we narrowed the camera's view to only observe plants of the current followed row. Discrete point clouds were filtered with the CloudCompare software. It was observed that the high-quality point clouds of the field and plants were generated. The leaves and stems of a plant can be easily recognized, which means 3D mapping results are accessible to use for further phenotyping analysis.

However, it still exists several limitations in this mapping pipeline. The accurate 3D mapping relies on the low drift odometry. The drift is difficult to eliminate, although we fused the multiple sensor data including the RGB-D camera, IMU, and the wheel encoder. That causes the location drift of all point clouds. For example, at the left top corner of the 3D map, the area with large drift even caused a breach. RTAB-Map provides loop closure detection to correct odometry drift, but the robot continued to explore new areas, which made it difficult to occur enough overlapped area to trigger the loop closure. Additionally, there is a lack of sufficient discriminative features in the simulation environment caused by the limited types of models. The limited view only covered a small area of crops, making its feature similarities were insufficient to exceed critical condition. Thus, the offline processing technologies to calibrate the mapping outcome should be further developed.

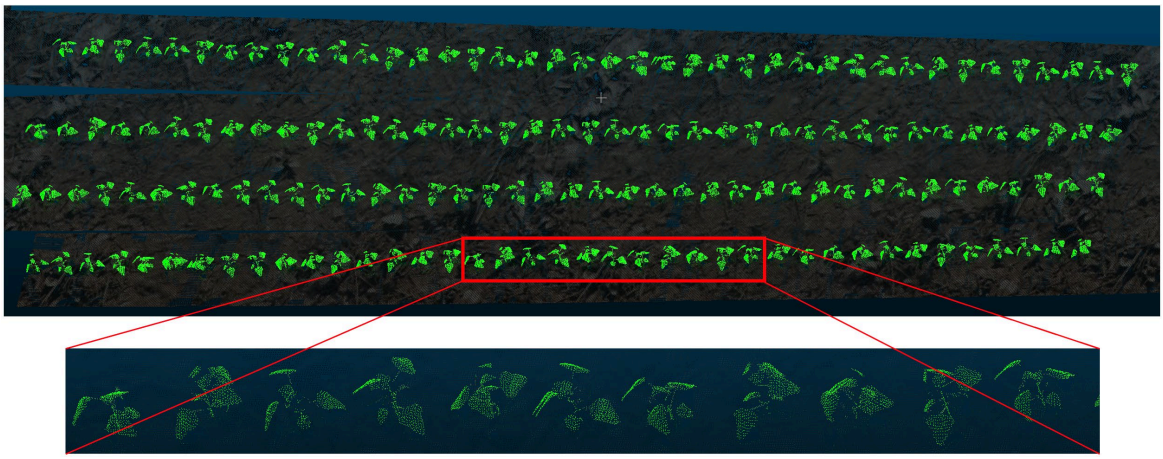


Figure. 11. 3D mapping result of field mapping pipeline

Mapping pipeline Analysis: Without the absolute global localization (e.g., GNSS), using only local data is limited for updating accurate odometry and location. In our field mapping pipeline, both the 2D distribution map and 3D map rely on accurate odometry estimation, especially for the 3D mapping, during which more than a centimeter-level drift can cause the low-quality points clouds of plants.

Figure 12 illustrates the odometry drift along the navigation trajectory. The red line presents the ground-truth trajectory of the robot, which is based on the simulator's real-time update. While the blue one is generated based on the odometry estimation. The primary drift occurs in the transition stage, when the velocity and heading of the robot are changed drastically. The odometry is sensitive to the noise. Once the drift occurs, its errors can accumulate over the time. For example, in the first transition stage, the pose of the robot is estimated with a slightly offset; then, the accumulated location error is up to 0.05 meter at the second transition stage, even up to maximum value of 0.12 meter at the fourth transition. Without the loop closure occurs, the odometry drift are not eliminated. One of the potential improvements is to combine the global localization information to improve the odometry estimation, for example, fusing GNSS or other localization system (e.g., WiFi, Bluetooth or UWB).

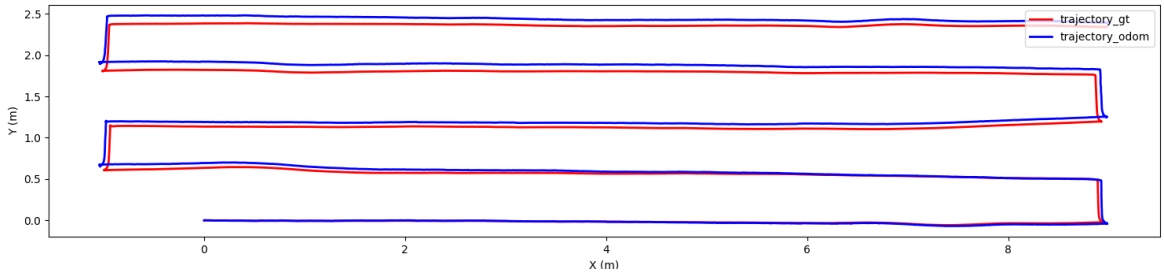
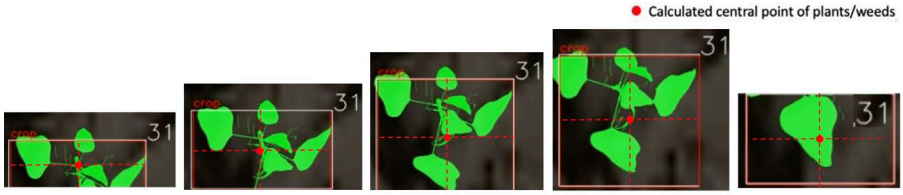


Figure. 12. Analysis of odometry drift along the navigation trajectory

In the 2D mapping pipeline, another error source of the plants' location is the central offset between the tracked plants and the ground truth. Figure. 13 illuminates the how the central offset occurs. with the movement of the robot, the location of a same plant in the image frame also changes accordingly. When the plant appears or disappear in the camera view, only part of the plant is in the image. If the partial plant is detected and regard it as a whole plant, then the corresponding central point of the tracking bounding box occurs offset. Thus, the tracking times of plants is extremely significant to the successful mapping. There are several accessible approaches to achieve including, but not limit more lightweight and accurate detector and tracker, higher-performance computation equipment, and slower movement of the robot.



(a) **(b)** **(c)** **(d)** **(e)**

Figure. 13. The illustration of tracking offset. (a) - (e) shows the tracking bounding boxes and the corresponding central points of a same plant in an image sequence

4. Conclusions

This study presents a Robot Operating System (ROS)-based phenotyping robot "MARS-PhenoBot" and demonstrates its reliable capacities including vision-based navigation and field mapping through ROS-Gazebo simulation. The proposed vision-based navigation pipeline exploits the row structure inherent in the crop fields to guide the robot along the crop row without the need for explicit localization systems, such as GNSS or a map of the environment. It handles the switching to new crop rows combining the prior knowledge of crop arrangement. At the same time, the field mapping approach for generating 2D and 3D maps was also evaluated. In a weedless cotton field, a YOLO-based detector and DeepSORT-based tracker achieved nearly a 100% success rate. The local error of the most plants in the 2D distribution map is lower than 9 cm, demonstrating its feasibility to be used in crop mapping. The 3D mapping technology was also evaluated on our customized robot, but the mapping quality is not satisfied because odometry drift. In our future work, we will develop the alignment and matching approaches to improve the mapping quality. The methodology developed in this study is scalable and can be deployed to the real agricultural robots to perform automated phenotyping tasks for crop field.

Acknowledgements

This study is supported by NSF Growing Convergent Research and Georgia Cotton Commission.

References

Ahmadi, A., M. Halstead, and C. J. a. p. a. McCool. 2021. Towards autonomous crop-agnostic visual navigation in arable fields.

Ahmadi, A., L. Nardi, N. Chebrolu, and C. Stachniss. 2020. Visual servoing-based navigation for monitoring row-crop fields. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE.

Atefi, A., Y. F. Ge, S. Pitla, and J. Schnable. 2021. Robotic Technologies for High-Throughput Plant Phenotyping: Contemporary Reviews and Future Perspectives. *Frontiers in Plant Science* 12.

Bao, Y., D. S. Shah, and L. Tang. 2018. 3D Perception-Based Collision-Free Robotic Leaf Probing for Automated Indoor Plant Phenotyping. *Transactions of the ASABE* 61(3):859-872.

Bao, Y., L. Tang, M. W. Breitzman, M. G. Salas Fernandez, and P. S. Schnable. 2019. Field-based robotic phenotyping of sorghum plant architecture using stereo vision. *Journal of Field Robotics* 36(2):397-415.

Benet, B., C. Dubos, F. Maupas, G. Malatesta, and R. Lenain. 2018. Development of autonomous robotic platforms for sugar beet crop phenotyping using artificial vision. In *AGENG Conference 2018*.

Bewley, A., Z. Ge, L. Ott, F. Ramos, and B. Upcroft. 2016. Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*. IEEE.

Bochkovskiy, A., C.-Y. Wang, and H.-Y. M. Liao. 2020. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Chen, M. Y., Y. C. Tang, X. J. Zou, Z. F. Huang, H. Zhou, and S. Y. Chen. 2021. 3D global mapping of large-scale unstructured orchard integrating eye-in-hand stereo vision and SLAM. *Computers and Electronics in Agriculture* 187:16.

Cubero, S., E. Marco-Noales, N. Aleixos, S. Barbe, and J. Blasco. 2020. RobHortic: A Field Robot to Detect Pests and Diseases in Horticultural Crops by Proximal Sensing. *Agriculture-Basel* 10(7).

Dong, W. B., P. Roy, and V. Isler. 2020. Semantic mapping for orchard environments by merging two-sides reconstructions of tree rows. *Journal of Field Robotics* 37(1):97-121.

English, A., P. Ross, D. Ball, P. Corke, and Ieee. 2014. Vision Based Guidance for Robot Navigation in Agriculture. In *IEEE International Conference on Robotics and Automation (ICRA)*. Hong Kong, PEOPLES R CHINA: Ieee.

Fan, Y., Z. Feng, A. Mannan, T. U. Khan, C. Shen, and S. Saeed. 2018. Estimating tree position, diameter at breast height, and tree height in real-time using a mobile phone with RGB-D SLAM. *Remote Sensing* 10(11):1845.

Grimstad, L., K. Skattum, E. Solberg, G. Loureiro, and P. J. From. 2017. Thorvald II configuration for wheat phenotyping. In *Proceedings of the IROS Workshop on Agri-Food Robotics: Learning from Industry*.

Habibie, N., A. M. Nugraha, A. Z. Anshori, M. A. Ma'sum, and W. Jatmiko. 2017. Fruit mapping mobile robot on simulated agricultural area in Gazebo simulator using simultaneous localization and mapping (SLAM). In *2017 International Symposium on Micro-NanoMechatronics and Human Science (MHS)*. IEEE.

Jeon, H. Y., L. F. Tian, and H. P. Zhu. 2011. Robust Crop and Weed Segmentation under Uncontrolled Outdoor Illumination. *Sensors* 11(6):6270-6283.

Jiang, Y., C. Li, J. S. Robertson, S. Sun, R. Xu, and A. H. Paterson. 2018. GPhenoVision: A Ground Mobile System with Multi-modal Imaging for Field-Based High Throughput Phenotyping of Cotton. *Scientific Reports* 8(1).

Labbe, M., and F. Michaud. 2019. RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation. *Journal of Field Robotics* 36(2):416-446.

Li, J. B., R. G. Zhu, and B. Q. Chen. 2018. Image detection and verification of visual navigation route during cotton field management period. *International Journal of Agricultural and Biological Engineering* 11(6):159-165.

Ludovisi, R., F. Tauro, R. Salvati, S. Khoury, G. Mugnozza Scarascia, and A. Harfouche. 2017. UAV-Based Thermal Imaging for High-Throughput Field Phenotyping of Black Poplar Response to Drought. *Front Plant Sci* 8:1681.

Mahlein, A. K. 2016. Plant Disease Detection by Imaging Sensors - Parallels and Specific Demands for Precision Agriculture and Plant Phenotyping. *Plant Dis* 100(2):241-251.

Mueller-Sim, T., M. Jenkins, J. Abel, and G. Kantor. 2017. The Robotanist: a ground-based agricultural robot for high-throughput crop phenotyping. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE.

Nellithimaru, A. K., and G. A. Kantor. 2019. ROLS: Robust Object-level SLAM for grape counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*.

Qiu, R., S. Wei, M. Zhang, H. Li, H. Sun, G. Liu, and M. Li. 2018. Sensors for measuring plant phenotyping: A review. *International Journal of Agricultural and Biological Engineering* 11(2):1-17.

Shafiekhani, A., S. Kadam, F. B. Fritschi, and G. N. DeSouza. 2017. Vinobot and Vinoculer: Two Robotic Platforms for High-Throughput Field Phenotyping. *Sensors (Basel)* 17(1):214.

Shakoor, N., S. Lee, and T. C. Mockler. 2017. High throughput phenotyping to accelerate crop breeding and monitoring of diseases in the field. *Current Opinion in Plant Biology* 38:184-192.

Smitt, C., M. Halstead, T. Zaenker, M. Bennewitz, and C. J. a. p. a. McCool. 2020. PATHoBot: A Robot for Glasshouse Crop Phenotyping and Intervention.

Sun, S. P., C. Y. Li, P. W. Chee, A. H. Paterson, C. Meng, J. Y. Zhang, P. Ma, J. S. Robertson, and J. Adhikari. 2021. High resolution 3D terrestrial LiDAR for cotton plant main stalk and node detection. *Computers and Electronics in Agriculture* 187:10.

Tilman, D., C. Balzer, J. Hill, and B. L. Befort. 2011. Global food demand and the sustainable intensification of agriculture. *Proc Natl Acad Sci U S A* 108(50):20260-20264.

Vijayarangan, S., P. Sodhi, P. Kini, J. Bourne, S. Du, H. Sun, B. Poczos, D. Apostolopoulos, and D. Wettergreen. 2018. High-throughput robotic phenotyping of energy sorghum crops. In *Field and Service Robotics*. Springer.

Virlet, N., K. Sabermanesh, P. Sadeghi-Tehran, and M. J. Hawkesford. 2016. Field Scanalyzer: An automated robotic field phenotyping platform for detailed crop monitoring. *Funct Plant Biol* 44(1):143-153.

Winterhalter, W., F. V. Fleckenstein, C. Dornhege, and W. Burgard. 2018. Crop row detection on tiny plants with the pattern hough transform. *IEEE Robotics and Automation Letters* 3(4):3394-3401.

Wojke, N., A. Bewley, D. Paulus, and Ieee. 2017. SIMPLE ONLINE AND REALTIME TRACKING WITH A DEEP ASSOCIATION METRIC. In *24th IEEE International Conference on Image Processing (ICIP)*. Beijing, PEOPLES R CHINA: Ieee.

Xiang, L., L. Tang, J. Gai, and L. Wang. 2020. PhenoStereo: a high-throughput stereo vision system for field-based plant phenotyping - with an application in sorghum stem diameter estimation</i&gt.

Xu, R., and C. Li. 2022. A modular agricultural robotic system (MARS) for precision farming: Concept and implementation. *Journal of Field Robotics*.

Young, S. N., E. Kayacan, and J. M. Peschel. 2018. Design and field evaluation of a ground robot for high-throughput phenotyping of energy sorghum. *Precision Agriculture* 20(4):697-722.

Zhang, Z., E. Kayacan, B. Thompson, and G. Chowdhary. 2020. High precision control and deep learning-based corn stand counting algorithms for agricultural robot. *Autonomous Robots* 44(7):1289-1302.