IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

P-AgNav: Range View-Based Autonomous Navigation System for Cornfields

Kitae Kim[1], Aarya Deb[1], and David J. Cappelleri[1,2]

***Abstract*—In this paper, we present an in-row and under-canopy autonomous navigation system for cornfields, called the Purdue Agricultural Navigation System or P-AgNav. Our navigation framework is primarily based on range view images from a 3D light detection and ranging (LiDAR) sensor. P-AgNav is designed for an autonomous robot to navigate in the corn rows with collision avoidance and to switch between rows without GNSS assistance or pre-defined waypoints. The system enables robots, which are intended to monitor crops or conduct physical sampling, to autonomously navigate multiple crop rows with minimal human intervention, thereby increasing crop management efficiency. The capabilities of P-AgNav have been validated through experiments in both simulation and real cornfield environments.**

***Index Terms*—Robotics and Automation in Agriculture and Forestry, Agricultural Automation**

I. INTRODUCTION

PRECISION agriculture employs Internet of Things (IoT) technologies to optimize resource use and improve crop yields for efficient farm management [1]–[4]. Agricultural robotics has been deployed to automate various agricultural tasks that were traditionally labor-intensive and time-consuming, such as weed removal, crop monitoring, physical sampling, and data collection for optimal health and productivity [5]–[10]. In order to monitor inside the rows and under cornfield canopy, reasonable-sized Unmanned Ground Vehicles (UGVs) of a reasonable size that can traverse between the corn rows are advantageous compared to other robotic platforms widely used in the agriculture sector, such as tractors and drones [11], due to the cluttered hanging leaves and narrow spaces between the rows. When the under-canopy traversable ground robots are able to navigate autonomously in the fields and perform assigned tasks on site, not only does the amount of intensive manual labor decrease, but productivity also increases. There are numerous traditional navigation and path planning algorithms for mobile robots with obstacle avoidance modules [12]. However, in agricultural settings, particularly in the rows and under the canopies of cornfields, classical navigation methods are not suitable. This is because agricultural elements, such as hanging leaves or weeds, that were not present in the pre-defined map during the early growth stages can be identified as obstacles, leading the robot to unnecessarily generate new paths to detour around these unexpected obstacles or causing it to fail in navigation. This means that conventional path planning methods are neither ideal nor optimal for in-row and under-canopy navigation. Therefore, we are motivated to design an autonomous navigation system specifically for in-row and under-canopy agricultural environments.

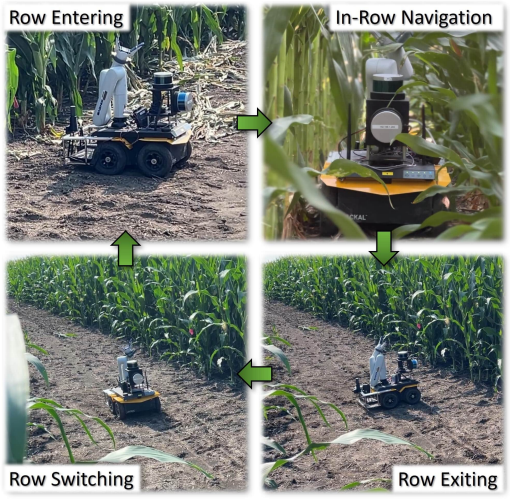


Fig. 1: Our robot, P-AgBot, navigates under the canopies in multiple rows in real cornfields using the proposed P-AgNav in this work. P-AgNav is designed for autonomous multi-row crop monitoring and physical sampling in cornfields. P-AgNav handles all the necessary stages of navigation, which include row entering (top-left), in-row navigation (top-right), row exiting (bottom-right), and row switching (bottom-left), without relying on GNSS assistance or pre-defined waypoints.

Many studies on autonomous navigation system design in agricultural settings have been conducted, focusing on two main approaches: Global Navigation Satellite System (GNSS)-based and camera-based systems. GNSS-based autonomous navigation systems [13]–[16] are widely utilized on over-canopy platforms to generate paths with relatively high accuracy in the global coordinate system. These systems are beneficial when platforms are not interfered with by their surroundings or are in the clear sky during operations. However, it is difficult to obtain reliable GNSS signals in rows and under the canopies of cornfields because the signals are

Manuscript received: September 4, 2024; Revised January 8, 2025; Accepted February 1, 2025.

This paper was recommended for publication by Editor Hyungpil Moon upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the IoT4Ag Engineering Research Center funded by the National Science Foundation (NSF) under the NSF Cooperative Agreement Number EEC-1941529.

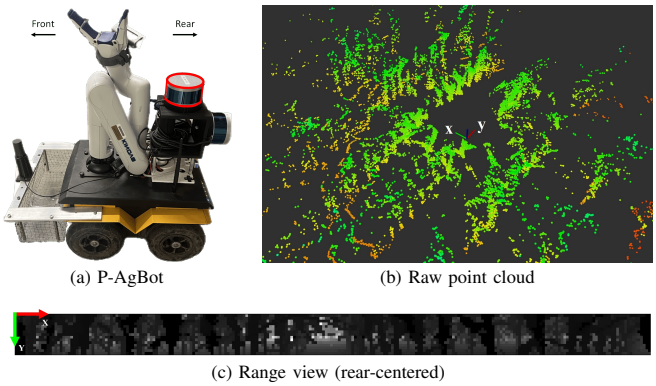
[1]Kitae Kim, Aarya Deb, and David J. Cappelleri are with the School of Mechanical Engineering, Purdue University, West Lafayette, IN USA.

[2] David J. Cappelleri is also with the Weldon School of Biomedical Engineering (By Courtesy), Purdue University, West Lafayette, IN USA. {kim3686,deb8,dcappell}@purdue.edu

This paper has a supplementary material available at https://youtu.be/AMmThS29mvM, provided by the authors.

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.



(a) P-AgBot (b) Raw point cloud

(c) Range view (rear-centered)

Fig. 2: (a) P-AgBot is a platform for our proposed navigation system, with a physical size of 40 cm in width and 65 cm in length, including a corn leaf storage box on the front side. The horizontally mounted 3D LiDAR (highlighted in red) data is used with P-AgNav. (b)(c) Examples of different representations (raw point cloud and range view) from the highlighted 3D LiDAR while traversing under the canopy of an arbitrary row in a real cornfield. The image coordinates, X and Y, in the range views from (b) are indicated with red and green arrows in (c), respectively, as summarized in Table I.

often interfered with or blocked by the plants, leading to a loss of accuracy. Camera-based systems [17]–[20] generally extract features from input images, and the outputs are used to control velocities accordingly to follow the rows. These methods have the advantage of obtaining and utilizing various features through images. In [19], an autonomous navigation system for under-canopy farm robots based on a machine learning method using low-cost camera data was described. However, overhanging leaves and corn ears often cause varying illumination conditions in rows, reducing the reliability of their methods. Furthermore, due to the limited field of view of the cameras, it is difficult to cover all required navigation scenarios and thus fully automate navigation across multiple rows. Our previous work in [7] proposed a 2D light detection and ranging (LiDAR)-based navigation method that detects left and right-side rows and publishes control velocities that balance the distances from the robot to both rows. It enables the robot to navigate safely within the rows. Even though 2D LiDAR has cost advantages, the system introduced in [7] has limitations on multi-row navigation because it is designed to navigate only a single row.

In this paper, we introduce the Purdue Agricultural Navigation System (P-AgNav), a 3D LiDAR range view (RV)-based autonomous navigation system for ground robots that are designed to perform in-row tasks under canopies in multiple rows in cornfields and overcome the limitations of previous studies. While 3D LiDARs generally provide richer spatial information, they come with increased costs compared to 2D LiDARs. To clarify this trade-off, we highlight the specific limitations of the 2D LiDAR system. The 2D LiDAR system in [7] has difficulties in capturing vertical spatial features critical for interpreting under-canopy environments and reduces the accuracy in detecting complex obstacles in real-world conditions. In contrast, the proposed navigation system enables the ground robot to traverse within rows and switch between rows autonomously with minimal collisions with corn plants.

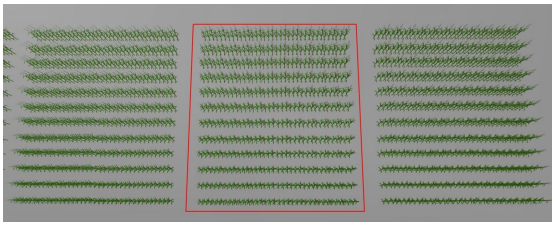


Fig. 3: An illustration of a simulation environment in ROS Gazebo that aligns with the assumed planting structures in traditional cornfields. A red box in this figure depicts a plot containing multiple rows of the same variety of corn plants. Each row in the plot is aligned with the corresponding row in the opposite plot.

This type of operation is needed for long-term and long-range autonomous field operations. Fig. 1 illustrates each stage of P-AgNav in a real cornfield with our robot. A range view, one of the various 3D LiDAR representations, is an image processed from a 3D LiDAR point cloud and is the main sensor data used in P-AgNav. The proposed navigation method is evaluated in both simulation and real cornfield environments and demonstrates the robustness of our method with minimal human intervention while traversing multiple corn rows. Our framework has three primary contributions:

- P-AgNav is able to reliably handle all navigation scenarios, including in-row and out-row navigation and row switching. Specifically, P-AgNav uses only a single 3D LiDAR and does not require GNSS assistance or predefined waypoints.
- P-AgNav minimizes the number of collisions with plants. It prevents disturbances to plant growth.
- P-AgNav extends the use of range views, mostly employed in autonomous cars [21]–[25], to agricultural robots and environments.

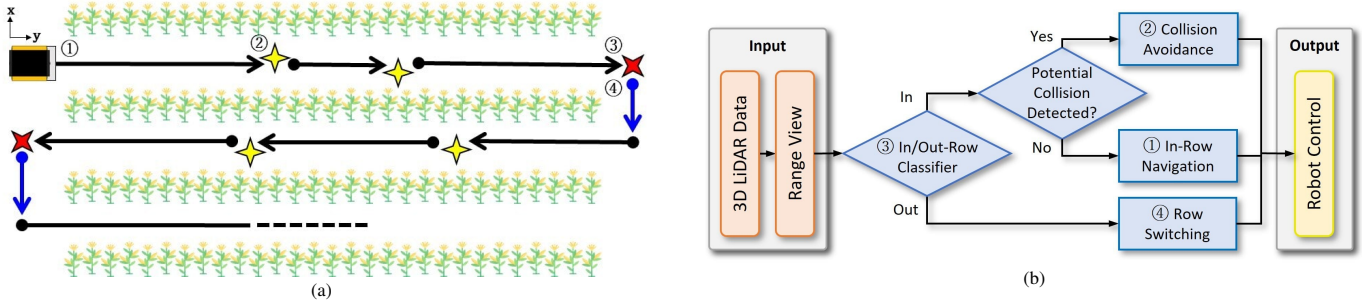
II. SYSTEM OVERVIEW

In this section, we first describe the robotic platform we used to demonstrate our system. Then, we briefly introduce the concept of range view, a fundamental data component of our proposed system. Finally, we introduce an overview of our proposed framework.

A. System Hardware: P-AgBot

Our hardware system, called P-AgBot [26], has been designed for in-row and under-canopy crop monitoring and physical sampling, mainly in cornfields. In order to achieve the tasks, P-AgBot consists of a small 4-wheel UGV from Clearpath Robotics, a variety of sensors, and a robotic arm, as illustrated in Fig. 2a. In this study, a horizontal 3D LiDAR (Velodyne VLP-16, highlighted in red in Fig. 2a), mounted on the back side of the robot, is the only sensor utilized for P-AgNav. When designing the navigation framework, we considered that the robotic arm mounted in front of the horizontally mounted LiDAR partially obstructs its field of view by approximately 9°.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.



(a) (b)

Fig. 4: (a) An overview diagram of the entire P-AgNav process: ① In-row navigation (black arrows), ② Collision avoidance (yellow markers), ③ In/Out classifier (red markers), and ④ Row switching (blue arrows). (b) The proposed pipeline of P-AgNav: Our pipeline processes input from a 3D LiDAR and converts the data into RV format. As a result of P-AgNav, the pipeline outputs the appropriate control velocity for the robot, adapting to the current navigation stage.

B. LiDAR Representation: Range View

The 3D LiDAR sensor, the only sensor utilized by our system, captures 3D spatial information with high precision and accuracy in the form of point clouds. Compared to cameras and 2D LiDARs, 3D LiDARs have distinct advantages to use in agricultural environments. The 3D LiDAR is independent of lighting conditions and has a 360° horizontal field of view that is wider than most cameras, making the systems possible to understand the current surrounding situations at once. In addition, the 3D LiDAR is able to scan 3D spatial information rather than just a single plane. The 3D LiDAR is particularly beneficial to use for robust navigation under the canopies of cornfields, our main agricultural environment, where random illumination variations exist due to various factors, such as overhanging leaves, corn ears, and changing weather conditions.

The key lies in effectively and efficiently processing the point cloud data from the 3D LiDAR to guarantee safe and robust autonomous navigation in row-based cornfields. There are a variety of 3D LiDAR data representations [27], *i.e.*, raw point clouds, voxel views, and range views (RVs). RV is a 2D image representation of 3D LiDAR data. Each pixel in the RV image corresponds to a specific laser beam of the 3D LiDAR and its value represents the range measured by the beam. A raw point cloud and the corresponding RV in a real cornfield are illustrated in Fig. 2b and 2c, respectively. Although raw point clouds and voxel representations are widely used, they are unordered and have complexities of $\mathcal{O}(N \cdot d)$, where N is the number of points in the point cloud that is typically on the order of 10^5 to 10^6, and d is the dimensionality of each point's data attributes. Similarly, voxel-based methods [28]–[30] organize the point cloud into a 3D grid structure, simplifying spatial queries and feature extraction. However, voxel representations also introduce high memory and computational costs due to the need to maintain a dense grid, resulting in the complexities of $\mathcal{O}(V \cdot d)$, where V represents the number of occupied voxels. The complexity of voxel-based methods grows rapidly with increasing point-cloud resolution.

Unlike 3D representations, RVs have a more compact data complexity of $\mathcal{O}(H \cdot W)$, where H and W are the height and width of the images, respectively. This lower complexity makes RVs more efficient for real-time processing on robots with limited computational power. In addition to the lightweight complexity, RVs are also advantageous for extracting valuable features from the robot's surroundings for diverse objectives, such as semantic segmentation [21], [24], [25] and scene understanding [21], [23]. In this paper, we propose a navigation algorithm that leverages RVs to extract features representing the robot's surroundings and provides appropriate controls in real time. In this work, we set H and W to 16 and 540, respectively, since this configuration minimizes the sparsity and maximizes the RV resolution for our 16 channel 3D LiDAR sensor. Before RVs are put into the system, image smoothing is applied to the RVs, a basic technique in vision processing to reduce small variations caused by noise. This allows the system to focus more on understanding the overall scene rather than on minor variations in the RVs.

C. Framework Overview

In order to efficiently perform agricultural tasks such as crop monitoring or physical sampling, the robot must be able to navigate autonomously and cover the field row by row. We established several assumptions regarding the planting structure in traditional cornfields to design our system, as shown in Fig. 3. We assumed that corn plants are arranged in straight and parallel rows. Each plot contains multiple rows, and there is sufficient open space between plots to allow the robot to switch rows. In addition, each row is aligned with the corresponding row in the opposite plot.

The proposed framework and an overview diagram of P-AgNav are illustrated in Fig. 4. P-AgNav consists of the following four stages:

1) **In-Row Navigation**: At the initial position, the robot navigates in the row and under the canopy.
2) **Collision Avoidance**: While in-row navigation, if the robot detects potential collisions with any part of corn plants, the robot stops driving and rotates toward a safer direction.
3) **In/Out-Row Classifier**: Once the robot exits the currently traversing row, the system recognizes that the robot is outside the row and commands the robot to switch to the next row.
4) **Row Switching**: The robot transitions to the next row.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

The four stages (①-④ in Fig. 4a) are iterated until the robot completes navigating the user-defined number of crop rows. Upon completing the task, the robot returns to its initial position and terminates the operation.

III. SYSTEM DESIGN

This section provides a detailed discussion of the approaches for each module of P-AgNav, all of which are driven by 3D LiDAR RV images. Since the robotic arm mounted on the front side of the robot partially occludes the LiDAR view, as explained in Section II, we used rear-centered RVs, where the center of the images is aligned exactly with the rear of the robot, as shown in Fig. 2c. Given the properties of RVs and the sensor setup of P-AgBot, we can state that the centerline of the RVs, $\mathrm{X} = \frac{W}{2}$, is consistently aligned with the robot's heading. This statement is one of the key features exploited by the system. Table I summarizes the necessary mathematical notations.

TABLE I: Summary of Notations

Notation	Description
R_t	Range view image from horizontal 3D LiDAR at time t
X, Y	Image coordinates of R_t
H, W	Height and width of R_t
$R_t\,(\mathrm{X}, \mathrm{Y})$	Pixel value at coordinates (X, Y) in R_t
v_t	Linear control velocity of the robot at time t
ω_t	Angular control velocity of the robot at time t

A. In-Row Navigation

The main objective of this module is to follow the row. To navigate inside the row without an explicit map, the system extracts relevant valuable features from R_t and utilizes these features to calculate and publish control velocities. In this module, it is important to identify the features that effectively indicate whether the robot is aligned well with the direction of the row for safe navigation. A key feature in this module is that when the robot is in a row, the pixels corresponding to a row entry (where the robot has already entered at the beginning of the navigation or after switching the row) show relatively higher (bright) $R_t\,(\mathrm{X}, \mathrm{Y})$, while other pixels in R_t have relatively lower (dark) $R_t\,(\mathrm{X}, \mathrm{Y})$. The module estimates the robot's relative heading to the row entry behind it by focusing on areas with high $R_t\,(\mathrm{X}, \mathrm{Y})$ in R_t. The outputs of the system, v_t and ω_t, are determined by the estimation of the robot's relative orientation to the row direction. For instance, if the direction of the row is perfectly aligned with the robot's heading, the robot does not need to adjust its steering and will proceed solely with linear velocity ($v_t \neq 0, \omega_t = 0$). If not, the robot requires the angular velocity to follow the row direction ($v_t \neq 0, \omega_t \neq 0$).

Our system creates a binary mask of R_t to identify the pixel candidates of the row entry in R_t. The binary mask is generated by applying a threshold based on the highest $R_t\,(\mathrm{X}, \mathrm{Y})$ in a user-defined threshold range, which is tuned based on the ratio between the minimum and maximum pixel values in the region of interest within R_t. This process masks the candidates of the row entry, which is the region of interest for the in-row navigation module. A blob detection algorithm is applied to the binary mask, not only to filter out noise but also to accurately extract the features of the row entry in R_t. The blob detection algorithm, one of the traditional techniques in computer vision, is widely used to locate areas of interest in images [31]. This algorithm identifies clusters of neighboring pixels that share similar characteristics, such as color, area, convexity, and circularity. In our system, the blob detector filters the binary mask image on the basis of color and area criteria. Consequently, the relative position of the row entry from the centerline of R_t, which is aligned with the robot's orientation, in the X coordinates at time t is calculated and denoted as $x_{r,t} = x_t - x_d$. Here, x_t represents the absolute X value of the row entry blob, and $x_d = \frac{W}{2}$ denotes the absolute X value of the centerline in R_t. An example of the generated blob in R_t is illustrated in Fig. 6a.

To calculate the control velocity v_t and ω_t, based on the relative location of the entry of the passed row in R_t, our system utilizes a model predictive controller (MPC). The objective of MPC is to find a solution that not only satisfies the state and input constraints but also minimizes the associated MPC cost J so that the controller aims to keep the system as close as possible to the desired trajectory. In our system, the state of MPC represents the robot's position $\mathbf{x}_t$ in RVs. The proposed system desires the robot to navigate as close as possible to the centerline of the row. To achieve the desired trajectory, we designed the MPC cost function J for this stage, as given by Eq. 1.

$$\begin{aligned} &\min_{\mathbf{x}_{1:T+1}, \omega_{1:T}} J(\mathbf{x}_{1:T+1}, \omega_{1:T}) \\ &J = \sum_{t=1}^{T} \left(\mathbf{x}_{t+1} - x_t\right)^2 + \left(\omega_t - \omega_{t-1}\right)^2 \end{aligned} \tag{1}$$

In Eq. 1, J represents the summation of two components over the prediction horizon T. The first component is the squared error between the predicted future states $\mathbf{x}_{t+1}$ and their desired state x_t. The second component represents the squared difference between consecutive angular velocities ω_t and ω_{t-1}, which is added to make the trajectory smoother. The state of the system is updated with state and control constraints, as described in Eq. 2. The bound of $\mathbf{x}_t$ is expressed by $[\mathbf{x}_{\min}, \mathbf{x}_{\max}]$, and the bound of ω_t is expressed by $[\omega_{\min}, \omega_{\max}]$ in Eq. 2.

$$\begin{aligned} &\mathbf{x}_{t+1} = \mathbf{x}_t + \omega_t \cdot dt \cdot \frac{W}{2\pi}, \quad \mathbf{x}_1 = x_d \\ &\mathbf{x}_{\min} \leq \mathbf{x}_t \leq \mathbf{x}_{\max}, \quad \omega_{\min} \leq \omega_t \leq \omega_{\max} \end{aligned} \tag{2}$$

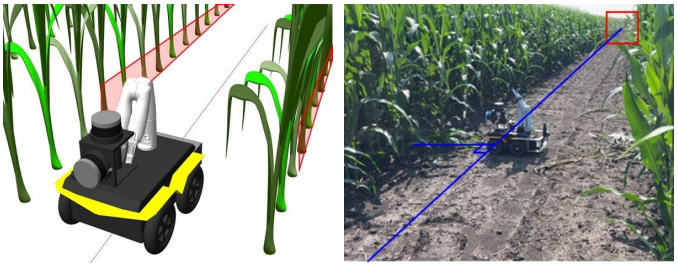
$$v_t = \min\left(\frac{c}{|\omega_t|}, v_{\max}\right), \quad c = \text{const.} \tag{3}$$

With the results from the MPC and an additional mathematical relationship between v_t and ω_t, which is defined in Eq. 3, the proposed system updates the control velocities v_t and ω_t for the in-row navigation.

B. Collision Avoidance

During in-row and out-row navigation task operations, the robot occasionally faces potential collisions arising from unexpected circumstances. These can result from unintended

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.



(a) (b)

Fig. 5: Visual representations of the regions of interest for collision detection and row switching modules. (a) Collision detection: In Sec. III-B, this region (red boxes) is defined as the lower part of the corn plants in the robot's forward-facing space. (b) Row switching: Blue lines indicate that the directions to the regions of interest (red box), which represent the open spaces in two opposite directions outside the rows, are always perpendicular to the direction to the row exit at the robot.

robot movements caused by insufficient traction between the wheels and the ground due to various conditions, such as mud or downed corn plants, as well as sensor occlusions from overhanging leaves. The system recognizes these situations and activates the collision avoidance module, which adjusts control velocities to prevent collisions.

Since P-AgNav is designed to move only forward, the risks of collisions exist only at the front of the robot. Regarding plant safety, it is more important to avoid corn stalks than corn leaves because hanging leaves can typically withstand impacts due to their flexibility. These leaves typically emerge near the upper section of the stalks. Taking these factors into account, the module defines the region of interest (RoI) for collision detection as the lower part of the corn plants located in the robot's forward-facing area, as illustrated in Fig. 5a. The RoI allows the system to focus primarily on the corn stalks. The RoI is assigned to a specific region in R_t, corresponding to the lower part of the forward-facing point cloud data projected into the R_t, and the module identifies the closest area to the robot among the RoI in R_t based on its $R_t\,(\mathrm{X}, \mathrm{Y})$ value. If $R_t\,(\mathrm{X}, \mathrm{Y})$ of the closest area is lower than a user-defined threshold, the system assesses the situation as a collision risk and activates the collision avoidance mode. The robot avoids potential collisions by performing the curvilinear motion in the direction that reduces the risk of collision ($v_t = v_r, \omega_t \neq 0$). Here, v_r denotes a small linear velocity value used for this navigation stage. The direction of rotation is determined by the relative X position of the closest area with respect to the centerline in R_t. For example, if the closest area is found on the front-right side of the robot, the robot rotates counterclockwise, otherwise, it rotates clockwise.

The other case to consider is the need to recover the robot's orientation when the robot's heading becomes misaligned with the row direction beyond a certain degree, regardless of its proximity to the crops. This misalignment can lead the robot into plant collisions in the narrow row. To monitor whether the robot heading aligns with the row direction or not, we employ $x_{r,t}$, the same feature used in Section III-A. If the robot's heading is not well aligned with the row direction, the robot performs an in-place rotation toward the direction that aligns itself with the row ($v_t = 0, \omega_t \neq 0$). The system uses the relative X position of the passed row entry with respect to the centerline in R_t, to guide the correction.

C. In/Out-Row Classifier

When the robot completes navigating the row, it exits the row and places itself outside of the row. In order to autonomously switch to the next row, the navigation system requires an in-row and out-row classifier. From the perspective of RV, one of the key differences between in-row and out-row is the density of plants surrounding the robot. Fig. 3 clearly shows that the open spaces between the plant plots are relatively empty compared to the in-row areas. This characteristic is reflected in R_t by the density of pixels with lower $R_t\,(\mathrm{X}, \mathrm{Y})$. If the number of pixels in R_t with lower values, which correspond to the pixels from objects close to the robot, is below a heuristic threshold, the classifier determines that the robot is outside the rows.

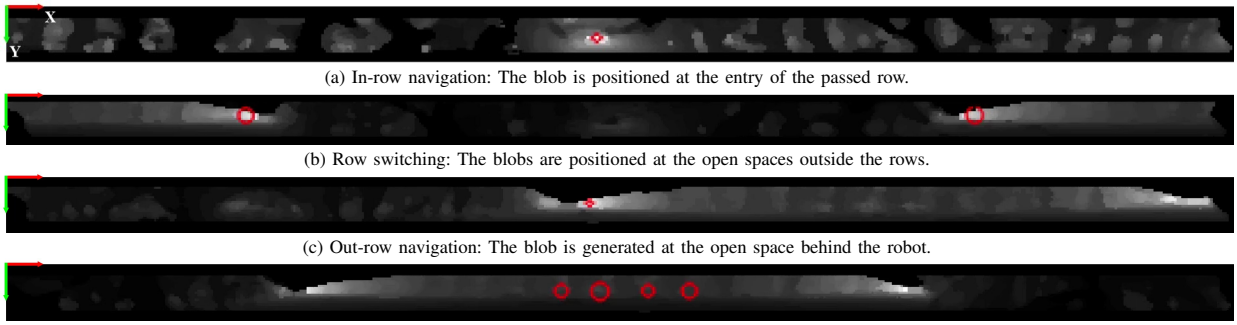
D. Row Switching

This module consists of the following four sub-steps.

1) 1st Rotation: P-AgNav calculates the rotation required for the robot to drive safely to the next-row area. The system defines the RoI in R_t as the 90° position clockwise and counterclockwise from the row exit blob coordinate x_t. The specific angle, 90°, comes from the fact that the directions to open spaces in two opposite directions outside the rows are always perpendicular to the direction of the row exit at the robot, as shown in Fig. 5b. Based on field characteristics, the open spaces on the left and right of the robot are placed at $\mathrm{X} = x_t \pm \frac{W}{4}$ in R_t, as expressed in Fig. 6b. This step aims to align the robot's heading with the direction of the RoI. According to its objective, P-AgNav calculates the distance between the robot's heading ($X = x_d$) and either $\mathrm{X} = x_t + \frac{W}{4}$ or $\mathrm{X} = x_t - \frac{W}{4}$, depending on the next target row, in R_t. This distance in R_t corresponds to the degree of angle required by which the robot needs to rotate. After the sub-step of the 1st rotation, the robot's orientation becomes parallel in the virtual line between the two open spaces, as shown in the bottom-left image of Fig. 1.

2) Out-Row Navigation: The robot navigates through the out-row area while simultaneously detecting the next row to navigate. There are similarities in the environmental structures between the in-row and out-row navigations. As a result of the 1st rotation, the positional relationship between the robot and the open space behind it becomes comparable to the relationship between the robot and the row entry when the robot navigates in the row. Taking advantage of these similarities, the robot is controlled using a concept similar to that of the in-row navigation and collision avoidance modules, as introduced in Section III-A and III-B, respectively. The out-row area is regarded as an in-row-like area with a greater width. The key difference in this sub-step is that a blob is generated from the open space, serving as the RoI for out-row navigation, and MPC is used to align the blob with the centerline in the RVs. An example of the generated blob from the open space in R_t is shown in Fig. 6c.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.



(a) In-row navigation: The blob is positioned at the entry of the passed row.

(b) Row switching: The blobs are positioned at the open spaces outside the rows.

(c) Out-row navigation: The blob is generated at the open space behind the robot.

(d) Next-row entering: Multiple blobs are generated from the rows on the opposite side of the plots. These include both paired and unpaired rows to the target next row.

Fig. 6: Examples of the range view image R_t and its blobs at each stage. The blobs, illustrated as red circles, represent the regions of interest in R_t at each stage. X and Y are illustrated with red and green arrows in each R_t.

An additional function in this sub-step, not included in the in-row navigation, is the detection of the next row to navigate. It is essential to switch the rows autonomously without GNSS assistance or pre-defined waypoints. The system uses a sequential feature between RVs. While traversing outside the row after the sub-step of the 1st rotation, the system focuses on the sequential feature from one side of the robot in the RVs, with the side chosen based on the target next row. The sequential feature used in this sub-step is as follows. The pixels corresponding to the side spaces in the RVs are unoccupied when the robot is at the row entrances. However, these pixels become occupied by corn plants between the current row and the next row, while the robot is on the way to the next row. Therefore, in the system, we define that the robot has reached the next row when these pixels complete the sequence of being unoccupied, then occupied, and finally unoccupied again.

3) 2nd Rotation: The 2nd rotation is required to navigate the next row once the robot has reached it. It uses the same method for the 1st rotation, which calculates the rotation angle through the robot heading and the direction of the RoI. The only change in this sub-step is that RoI is redefined from the open-space pixels to the next-row pixels in RVs.

4) Next-Row Entering: At the beginning of the row entry, multiple blobs are generated in RVs if the system uses the in-row navigation module because the robot is not yet in the row and the LiDAR sensor scans multiple rows on the opposite side of the plots at the same time. The initial stage, when the robot begins navigating in the field, is also classified under the sub-step *Next-Row Entering*, as the robot is initially positioned at the row entry. Fig. 6d represents a result of the multi-row detection. In such situations, it is possible for the robot to lose its navigation direction by following one of the blobs generated from the rows on the opposite side of the plots, which are not paired to the row where the robot is currently located. Therefore, it is important to correctly identify the blob from the paired row in the opposite plot corresponding to the current row in this sub-step. From the robot's point of view, the paired blob is the only one whose direction is always perpendicular to the virtual line connecting the open spaces on both sides, as detailed in Section III-D1 and Fig. 5b. This structural characteristic is utilized to identify the paired blob and to track it until the open spaces on the sides disappear from the LiDAR's field of view during the row entering.

If the next row is detected slightly late or early, it is possible that the robot does not align properly with the next row and crashes into the plants. To deal with these situations, the system allows the robot to track the centerline of the row by using the position of the paired row in the RVs, and the system publishes the control velocities for driving toward the centerline of the row during this sub-step. It is the same method as the in-row navigation module, described in Sec. III-A. This method prevents collisions and enables the robot to enter the next row safely.

E. System Termination

Once the robot has completed navigating the user-defined number of rows with the four main stages of P-AgNav, the system commands the robot to return to the starting row. The system counts the number of crop rows on its side that the robot has passed to successfully end the operation on the initial row, using the same method described in Section III-D4.

IV. EXPERIMENTAL RESULTS

The experiments are conducted to demonstrate the capabilities of our proposed system, P-AgNav, without GNSS assistance or pre-defined waypoints, across various cornfield scenarios, including both in simulation and in real cornfield environments. The simulation and the real cornfield environments are referred to as SIM and ACRE (Agronomy Center for Research and Education at Purdue), respectively. Table II summarizes the specification of experimental environments and Fig. 1 and 3 briefly show ACRE and SIM, respectively. The simulated P-AgBot with the same sensor configuration as our real robot is deployed in the ROS Gazebo environment. In SIM, three different types of 3D corn plant models are used to mimic real-world diversity. The real robot is equipped with an Intel Core i5 4570T CPU on board. ROS Gazebo does not support physical attributes for 3D models, meaning that the simulated robot in SIM considers avoiding not only the corn stalks but also hanging leaves for successful navigation, unlike in ACRE. For safe and precise crop monitoring and physical sampling, it is preferable to navigate at a slow speed for safety rather than at a higher speed. Thus, in the experiments, we

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE II: Specifications of Experimental Environments

Environment	Row width [m]	Row length [m]	Crop interval [m]	Plot spacing [m]
SIM	0.80	10.0	0.25	1.20
ACRE	0.73	12.0	0.22	3.05

TABLE III: Navigation Performance in SIM

# rows	Total distance [m]	Traversal time [min]	# collisions	# human interventions	Avg. distance per collision [m]
4	46.4	24	0	0	-
6	70.6	34	0	0	-

set the maximum linear and angular velocities of P-AgBot to $v_{\max} = 0.1\,\text{m/s}$ and $\omega_{\max} = 0.05\,\text{rad/s}$, respectively.

The experiments are designed to show that the navigation system is capable of navigating multiple rows autonomously while minimizing the number of collisions with our robot, P-AgBot. During the tests, a supervisor monitors the robot's operation and intervenes if the robot requires any manual recovery. In ACRE, corn plants are grown in each of the six rows of each plot according to the same nutrient management practices. Therefore, we designed each experimental scenario to begin in the first row in a group under the same nutrient conditions and navigate the row and adjacent rows, selecting fewer than six rows with the same nutrient conditions. When the robot is configured to navigate multiple rows, the robot maneuvers the rows in a zig-zag pattern. We evaluated the performance of P-AgNav using two criteria: collisions with crops and human interventions. Human interventions were required not only to prevent collisions with crops in some cases but also for various unforeseen yet unavoidable situations in the testbed, to be explained later.

The navigation results from SIM underscore the overall capabilities of the proposed system. In contrast to real cornfields, the simulation environments do not contain naturally occurring obstacles, *i.e.*, piled-up downed corn plants or muddy terrain. This means that SIM is efficient in validating the overall effectiveness of the proposed methods in P-AgNav. In SIM, hanging leaves are static models that need to be avoided. Consequently, we set the related thresholds of the proposed system more conservatively in SIM than in ACRE. Table III describes that the robot successfully followed each row and switched to the next row without any collisions or human intervention in SIM. These results highlight two key points. First, the robot is able to traverse inside the rows, avoiding potential collisions with cluttered corn plants, with P-AgNav. Secondly, the robot successfully performed the row switching and demonstrated good performance in identifying both the open spaces between the plots and the next rows, as well as in precisely entering the next target rows.

We summarize the results of ACRE experiments in Table IV, using the validated P-AgNav system in SIM. Even if the robot completed navigating multiple rows with P-AgNav, a few human interventions were required in some instances at ACRE due to the diverse field conditions. Across 25 trials with a total driving distance of 834.5 m, the (mean, standard deviation) of the number of collisions and human interrupts were $(0.92, 1.41)$ and $(1.88, 1.64)$, respectively. The average traversing distance per collision was 37.93 m. The cases that caused the supervisor to intervene are described below.

TABLE IV: Navigation Performance in ACRE

Corn type	# rows	Total distance [m]	Traversal time [min]	# collisions	# human interventions	Avg. distance per collision [m]
Short	4	58.4	25	0	0	-
	4	58.0	19	0	2	-
	6	87.0	37	3	3	29.0
Tall	4	58.3	23	0	0	-
	4	58.1	23	1	3	58.1
	6	88.2	38	3	3	29.4

(a) (b) (c)

Fig. 7: Unforeseen challenges for the autonomous navigation in the real cornfields (ACRE): (a) Empty spaces due to missed plants, (b) Hard roots protruding from the ground (a red circle), (c) Piled-up downed corn plants.

Collisions primarily occurred during two specific events. First, in the middle of the row, there are empty spaces where a couple of plants are missing along the side of the row, as shown in Fig. 7a. These empty spaces occur if the corn plants were accidentally missed during planting by tractors or if farm supervisors removed the plants. When the robot passes the row with these spaces, the system, which normally identifies potential collisions from the row, is unable to detect potential collisions from these spaces. This is because the system assumes that each row is composed of dense corn plants. Consequently, the robot moves toward the empty space and collides with the row where the empty space is located. The second event occurs when the overhanging leaves from the crops continuously occlude the LiDAR sensor for a longer duration than usual. This event results in the system being unable to publish timely control velocities for collision avoidance. In addition to the collision events, human interventions were also required in several challenging scenarios, such as navigating through muddy terrain and encountering hard roots protruding from the ground (Fig. 7b) and piled-up downed corn plants (Fig. 7c). These challenges caused the robot to become stuck, disrupting its autonomous navigation. P-AgBot can traverse over one or two downed corn plants, but it is challenging to navigate through piles of plants. Throughout the experiments in both simulation and real cornfield environments, our proposed navigation system, P-AgNav, is validated to handle all scenarios, including in-row and out-row navigation, using a single LiDAR sensor. The experimental results at ACRE highlight the complexity of real-world agricultural environments and the need for further system refinements to handle such unforeseen challenges effectively.

V. CONCLUSION

In this paper, we introduced P-AgNav, an autonomous navigation framework based solely on a single LiDAR sensor,

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

specifically designed to support automated crop monitoring and physical sampling in row-based cornfields. By utilizing LiDAR's range view images, our system enables mobile robots to navigate in the rows and under the canopies of cornfields, effectively avoiding potential collision and also facilitating autonomous row switching, only using a single LiDAR. The capabilities of our proposed system are validated in the experiments both in the simulation and real cornfield environments. Ultimately, we aim to integrate our P-AgNav system with our crop sampling module which can pick or cut target crops or leaves in cornfield environments for precision agriculture applications, to increase productivity and reduce the intensive manual labor in the farms.

During the future growing seasons, we plan to enhance P-AgNav's applicability to a wider range of agricultural environments. This will be achieved by adapting the P-AgNav system to diverse crop types and planting patterns. The crop spacing and patterns are known after planting. Thus, the P-AgNav approach can be modified accordingly to handle these situations prior to deployment, as needed. In addition, our goal is also to integrate the P-AgNav system with our crop sampling module. This integration will enable physical sampling tasks with autonomous navigation, such as cutting corn leaves and ears, which we expect to increase productivity and reduce intensive manual labor on farms.

REFERENCES

[1] R. P. Sishodia, R. L. Ray, and S. K. Singh, "Applications of remote sensing in precision agriculture: A review," *Remote sensing*, vol. 12, no. 19, p. 3136, 2020.

[2] C. R. Kagan, D. P. Arnold, D. J. Cappelleri, C. M. Keske, and K. T. Turner, "Special report: The internet of things for precision agriculture (iot4ag)," *Computers and Electronics in Agriculture*, vol. 196, p. 106 742, 2022.

[3] A. Dechemi, D. Chatziparaschis, J. Chen, *et al.*, "Robotic assessment of a crop's need for watering: Automating a time-consuming task to support sustainable agriculture," *IEEE Robotics and Automation Magazine*, vol. 30, no. 4, pp. 52–67, 2023.

[4] F. A. Castiblanco, M. S. Basir, A. D. Balmos, J. V. Krogmeier, and D. R. Buckmaster, "Avena: An event-driven software framework for informed decisions and actions in cropping systems," Aug. 2024.

[5] A.-F. Jimenez, P.-F. Cardenas, A. Canales, F. Jimenez, and A. Portacio, "A survey on intelligent agents and multi-agents for irrigation scheduling," *Computers and Electronics in Agriculture*, vol. 176, p. 105 474, 2020.

[6] R. Manish, Z. An, A. Habib, M. R. Tuinstra, and D. J. Cappelleri, "Agbug: Agricultural robotic platform for in-row and under canopy crop monitoring and assessment," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, American Society of Mechanical Engineers, vol. 85451, 2021, V08BT08A017.

[7] K. Kim, A. Deb, and D. J. Cappelleri, "P-agbot: In-row & under-canopy agricultural robot for monitoring and physical sampling," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7942–7949, 2022.

[8] X. Liu, S. W. Chen, G. V. Nardari, *et al.*, "Challenges and opportunities for autonomous micro-uavs in precision agriculture," *IEEE Micro*, vol. 42, no. 1, pp. 61–68, 2022.

[9] G. Roggiolani, M. Sodano, T. Guadagnino, F. Magistri, J. Behley, and C. Stachniss, "Hierarchical approach for joint semantic, plant instance, and leaf instance segmentation in the agricultural domain," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 9601–9607.

[10] A. Deb, K. Kim, and D. J. Cappelleri, "Deep learning-based leaf detection for robotic physical sampling with p-agbot," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2023, pp. 8291–8297.

[11] B. Nazeri and M. Crawford, "Detection of outliers in lidar data acquired by multiple platforms over sorghum and maize," *Remote Sensing*, vol. 13, no. 21, p. 4445, 2021.

[12] B. Patle, A. Pandey, D. Parhi, A. Jagadeesh, *et al.*, "A review: On path planning strategies for navigation of mobile robot," *Defence Technology*, vol. 15, no. 4, pp. 582–606, 2019.

[13] T. Mueller-Sim, M. Jenkins, J. Abel, and G. Kantor, "The robotanist: A ground-based agricultural robot for high-throughput crop phenotyping," in *2017 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2017, pp. 3634–3639.

[14] T. C. Thayer, S. Vougioukas, K. Goldberg, and S. Carpin, "Multirobot routing algorithms for robots operating in vineyards," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 3, pp. 1184–1194, 2020.

[15] J. M. Bengochea-Guevara, J. Conesa-Muñoz, D. Andújar, and A. Ribeiro, "Merge fuzzy visual servoing and gps-based planning to obtain a proper navigation behavior for a small crop-inspection robot," *Sensors*, vol. 16, no. 3, p. 276, 2016.

[16] S. Kanagasingham, M. Ekpanyapong, and R. Chaihan, "Integrating machine vision-based row guidance with gps and compass-based routing to achieve autonomous navigation for a rice field weeding robot," *Precision Agriculture*, vol. 21, no. 4, pp. 831–855, 2020.

[17] I. D. García-Santillán, M. Montalvo, J. M. Guerrero, and G. Pajares, "Automatic detection of curved and straight crop rows from images in maize fields," *Biosystems Engineering*, vol. 156, pp. 61–79, 2017.

[18] S. K. Panda, Y. Lee, and M. K. Jawed, "Agronav: Autonomous navigation framework for agricultural robots and vehicles using semantic segmentation and semantic line detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2023, pp. 6272–6281.

[19] A. N. Sivakumar, S. Modi, M. V. Gasparino, *et al.*, "Learned Visual Navigation for Under-Canopy Agricultural Robots," in *Proceedings of Robotics: Science and Systems*, Virtual, Jul. 2021.

[20] A. Durand-Petiteville, E. Le Flecher, V. Cadenat, T. Sentenac, and S. Vougioukas, "Tree detection with low-cost three-dimensional sensors for autonomous navigation in orchards," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3876–3883, 2018.

[21] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "Rangenet++: Fast and accurate lidar semantic segmentation," in *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, IEEE, 2019, pp. 4213–4220.

[22] L. Fan, X. Xiong, F. Wang, N. Wang, and Z. Zhang, "Rangedet: In defense of range view for lidar-based 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 2918–2927.

[23] L. Kong, Y. Liu, R. Chen, *et al.*, "Rethinking range view representation for lidar segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2023, pp. 228–240.

[24] J. Kim, J. Woo, and S. Im, "Rvmos: Range-view moving object segmentation leveraged by semantic and motion features," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8044–8051, 2022.

[25] A. Ando, S. Gidaris, A. Bursuc, G. Puy, A. Boulch, and R. Marlet, "Rangevit: Towards vision transformers for 3d semantic segmentation in autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2023, pp. 5240–5250.

[26] K. Kim, A. Deb, and D. J. Cappelleri, "P-agslam: In-row and under-canopy slam for agricultural monitoring in cornfields," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 4982–4989, 2024.

[27] J. Xu, R. Zhang, J. Dou, Y. Zhu, J. Sun, and S. Pu, "Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 16 024–16 033.

[28] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019.

[29] Z. Liu, H. Tang, Y. Lin, and S. Han, "Point-voxel cnn for efficient 3d deep learning," *Advances in neural information processing systems*, vol. 32, 2019.

[30] H. Tang, Z. Liu, S. Zhao, *et al.*, "Searching efficient 3d architectures with sparse point-voxel convolution," in *European conference on computer vision*, Springer, 2020, pp. 685–702.

[31] T. Lindeberg, "Feature detection with automatic scale selection," *International journal of computer vision*, vol. 30, pp. 79–116, 1998.