

# LiDAR-Based Crop Row Detection Algorithm for Over-Canopy Autonomous Navigation in Agriculture Fields

Ruiji Liu<sup>1</sup>, Francisco Yandun<sup>1</sup> and George Kantor<sup>1</sup>

**Abstract**— Autonomous navigation is crucial for various robotics applications in agriculture. However, many existing methods depend on RTK-GPS systems, which are expensive and susceptible to poor signal coverage. This paper introduces a state-of-the-art LiDAR-based navigation system that can achieve over-canopy autonomous navigation in row-crop fields, even when the canopy fully blocks the interrow spacing. Our crop row detection algorithm can detect crop rows across diverse scenarios, encompassing various crop types, growth stages, weeds presence, and discontinuities within the crop rows. Without utilizing the global localization of the robot, our navigation system can perform autonomous navigation in these challenging scenarios, detect the end of the crop rows, and navigate to the next crop row autonomously, providing a crop-agnostic approach to navigate the whole row-crop field. This navigation system has undergone tests in various simulated agricultural fields, achieving an average of 2.98cm autonomous driving accuracy without human intervention on the custom Amiga robot. In addition, the qualitative results of our crop row detection algorithm from the actual soybean fields validate our LiDAR-based crop row detection algorithm’s potential for practical agricultural applications.

## I. INTRODUCTION

The escalating global population, labor shortage, and environmental challenges pose significant concerns regarding food shortage. These challenges emphasize the urgent need for sustainable agricultural practices and innovative solutions to ensure food security for future generations [1]. Agricultural robots have emerged as a sustainable solution to tackle these challenges in agri-food production, offering precision farming capabilities [2].

Autonomous navigation without damaging the crops plays a pivotal role in different robotics applications in agriculture. Existing technologies can safely drive the robot and perform various tasks on the pre-devised path by utilizing the Real Time Kinematic Global Positioning System (RTK-GPS) [3] [4]. However, deploying RTK-GPS devices on robots for various agriculture fields is unrealistic due to the expense of the RTK-GPS devices. In addition, the RTK network coverage and the obstacles above the GPS antenna will affect the accuracy of the receiving position messages. In the worst scenario, GPS malfunction can cause a complete failure in autonomous navigation. It leads to the development of precise crop row detection algorithms utilizing local sensors like cameras and LiDAR. By having precise crop row locations, the robot can locate itself and follow the crop rows, achieving resilient and cost-effective autonomous navigation independent of the robot’s global position. Nevertheless,



Fig. 1. Visualization of our custom Amiga robot working environments and robot configuration. (i) The Amiga robot is navigating in the soybean field. (ii) The side view of the Amiga robot with customized sensors and manipulators.

field conditions in reality, such as crop types, crop growth stages, terrains, discontinuities, and weeds presence, pose significant challenges to precise crop row detection and robot localization.

In this paper, we introduce a state-of-the-art LiDAR-based over-canopy crop row detection algorithm combined with the crop row following algorithm and the lane-switching algorithm. This autonomous navigation system enables agricultural robots to navigate on different crops and cover the entire field without human intervention. The main contributions of our work are summarized below:

- A novel and robust multi-crop-row detection algorithm that works in fields with different crop types, growth stages, weeds presence, and discontinued crop rows.
- A multi-crop-row autonomous following and lane-switching algorithm, enabling the robot full field coverage.
- Validation of our autonomous navigation system, incorporating crop row detection and following algorithm, on a full-scale Amiga robot in Gazebo simulated fields (corn, young soybean, and mature soybean). Additionally, we obtain qualitative results of our crop row detection algorithm using LiDAR data from real soybean fields.

## II. RELATED WORK

Autonomous agricultural robots serve various functions, such as seeding [5], harvesting [6], and crop monitoring [7], in modern farming practices to enhance productivity and sustainability. Full autonomous navigation for agricultural robots across different crop fields is also a popular research topic. Despite the widespread use of GPS with RTK corrections in outdoor farm robots for autonomous navigation [3], [4], its adoption as a general solution is hindered by

<sup>1</sup>Robotics Institutes, Carnegie Mellon University, US {ruijil, fyandun, gkantor}@andrew.cmu.edu

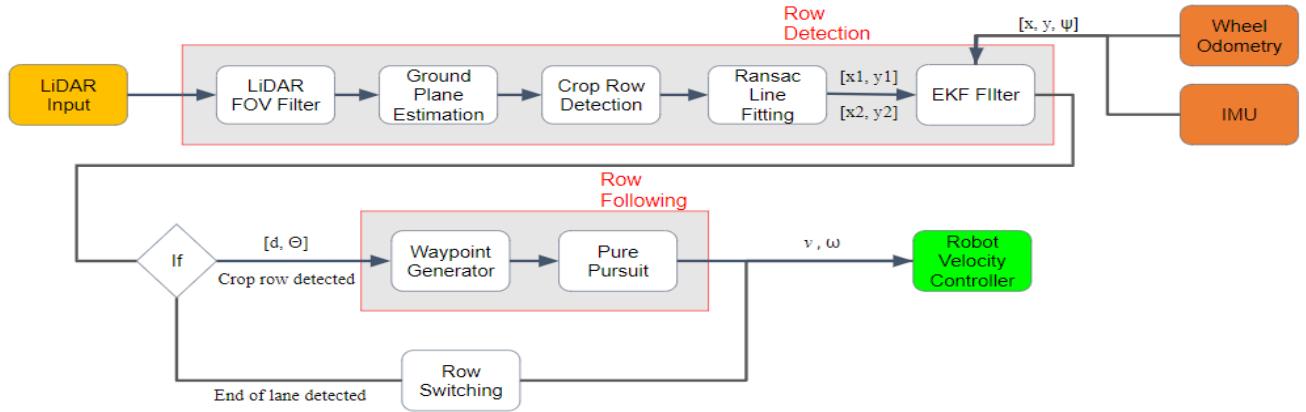


Fig. 2. Navigation system's workflow. The row detection algorithm takes LiDAR, wheel odometry, and IMU data as input and publishes predicted crop row positions in the form of  $[d, \phi]$ . The Row following algorithm takes the predicted crop row positions and follows the crop row by controlling the robot's linear velocity  $v$  and angular velocity  $w$ . If the end of the crop lane is detected, the robot will perform a lane-switching process and enter the next lane.

its high cost and signal coverage problem. Thus, researchers investigate the crop row detection algorithm to find a more robust and cheap solution for autonomous navigation in the agricultural field.

Previous research on crop row detection algorithms utilizes different methods and combinations of onboard sensors. Vision-based crop row detection algorithm extracts the texture of the crop row [8] or the color information from the RGB images [9] based on the computer vision method to generate crop row masks and detect the rows. LiDAR-based method extracts the line parameter from 2D point clouds [10]. The combination of different sensors, such as LiDAR and cameras, increases the robustness of the row detection algorithm for orchards planted in row structures [11]. In addition, the emergence of deep learning for image segmentation unveils a new area of crop row detection algorithm [12] [13]. However, most of these crop row detection algorithms are only tested on a limited type of crop fields and early stages of crops. Moreover, none of these algorithms provide a robust solution for over-canopy navigating on the crops when the canopy blocks the interrow spacing entirely, such as mature soybeans. Most existing methods only utilize the information from the RGB images but ignore the depth information. This results in their failure on crops with full canopy closure, since the RGB images provide no visual information to detect the

crop rows. To address this problem, we propose an innovative crop-agnostic crop row detection algorithm utilizing LiDAR data to acquire the 3D information of the crop rows and output precise crop row positions and orientations. Our crop row detection method has undergone tests in actual soybean fields and simulated fields(corn and soybean), including discontinuities and weeds presence in rows. Additionally, we present a crop row following algorithm integrated with a dead-reckoning lane-switching mechanism, allowing robots to navigate the entire field. For real-time evaluation, we implement the navigation system on the full-scale Amiga robot model and test it in Gazebo simulation environments. We have open-sourced this navigation system and simulation environments<sup>1</sup>.

### III. SYSTEM DESIGN

Fig. 1 provides an overview of the custom Amiga robot configuration and working environments. A full description of the custom Amiga configuration used in this paper can be found here [14]. Our crop row detection algorithm is based on the nature of over-canopy navigation for this robotics platform. Fig. 2 illustrates our navigation system workflow. In the following sections, we introduce brief information of the Amiga robot platform in Sec. III-A and the navigation strategy for autonomously navigating across the entire field in Sec. III-B.

#### A. Robotics Platform

The Amiga robot is an all-electric robot platform that accommodates various tasks, crops, and terrain. With a 250 lbs base weight, this platform provides a 1000 lbs payload for 8 hours on flat ground, combined with a maximum speed of 5.5 mph. The width and length of the Amiga platform can be adjusted by switching the beam across wheels. For navigation purposes, the width of the platform is set to 5 ft with a vertical clearance of 1m. This enables the Amiga robot to

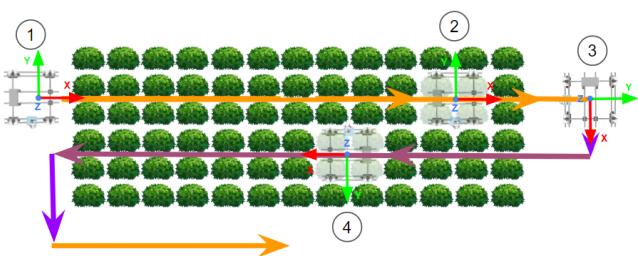


Fig. 3. Navigation strategy for navigating across the entire field. ① enter and follow the detected crop rows, ② detect the end of the crop row, ③ switch to the next crop row, ④ follow the new crop rows

<sup>1</sup>[https://github.com/Kantor-Lab/LiDAR\\_CropRowDetection](https://github.com/Kantor-Lab/LiDAR_CropRowDetection)

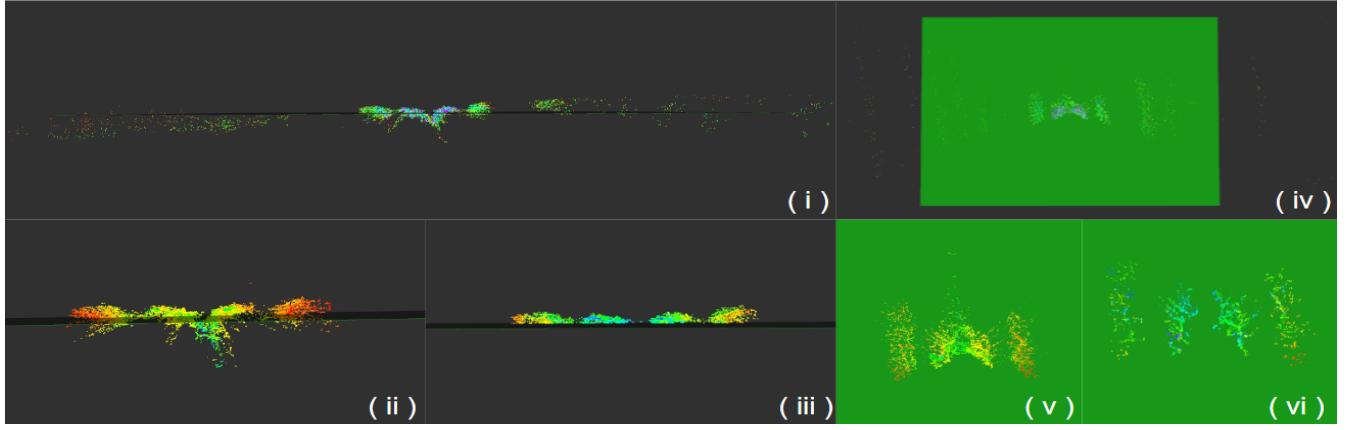


Fig. 4. Visualization of the point clouds from an actual mature soybean field in both side and top views after each filtering step. The green plane in all the images represents the virtual ground plane we created to filter point clouds. (i) Side view of the raw LiDAR point clouds. (ii) Side view of the LiDAR point clouds within the robot's dimension. (iii) Side view of the LiDAR point clouds after filtering out the points below the generated virtual plane. (iv) Top view of raw LiDAR point clouds. (v) Top view of the LiDAR point clouds in (ii). (vi) Top view of the LiDAR point clouds in (iii).

perform over-canopy navigation in row-crop fields with crop heights lower than 1m, such as corn, pepper, and soybean fields. The length of the platform is set to 4 ft to provide enough space for mounting sensors and manipulators. We deploy multiple sensors on the Amiga platform. For testing our navigation system, we install one GPS device with RTK correction for ground truth position, a Vectorsnav IMU, and a Velodyne VLP 16 LiDAR in the center front of the robot. We define the robot coordinate axis as follows: x for forward, y for leftward, and z for upward direction, as illustrated in Fig. 3.

### B. In Row Navigation Strategy

The ultimate goal of autonomous navigation in agriculture is to cover the entire crop field to perform various tasks. Crops are typically planted in parallel rows to facilitate efficient cultivation, irrigation, and harvesting. As a result, to cover the field, the robot needs to follow the crop row, detect the end of the row, and switch to the next unvisited crop lane autonomously. Fig. 3 represents our navigation strategy to enable whole-field-navigation. The robot starts at

the beginning of the crop rows and autonomously follows the crop row utilizing the LiDAR-based navigation system. Once the LiDAR detects the end of the lane (no more detected crops), the robot starts the lane-switching process. Assuming the crop rows are parallel, we utilize a dead reckoning control method that first turns the robot by 90 degrees (based on IMU reading), drives the robot forward based on the crop row distance and odometry reading, and turns the robot 90 degrees again to enter the next lane. Once the lane-switching process is finished, the LiDAR-based navigation system is started again, and the robot will autonomously navigate through the new lane.

## IV. AUTONOMOUS NAVIGATION IN ROW-CROP FIELDS

This autonomous navigation system consists of two parts: crop-row detection and crop-row following. Sec. IV-A describes how we extract the LiDAR point cloud data and perform crop row detection on the extracted results. The crop row following algorithm navigates the robots along the detected crop row as described in Sec. IV-B.

### A. Multiple Crop Row Detection Algorithm

We choose to utilize LiDAR point cloud data to perform crop row detection. LiDAR provides accurate 3D elevation and depth data that better describes crop row trends than cameras, especially when crop canopy covers the interrow spacing. The LiDAR is installed in the center front of the robot and has the same coordinate axis as the Amiga robot as described in III-A. The Lidar is only tilted down along its Y-axis(leftward) to obtain a sufficient amount of point clouds for crop row detection. Based on what we observe in real fields, we assume the crop rows are planted in a parallel fashion and the ground is flat.

Inspired by human visual depth perception in real crop fields and leveraging height differences between crops and interrow spacing, our crop row detection algorithm utilizes LiDAR data to provide sufficient 3D information on crops, enabling accurate detection of crop rows. The algorithm

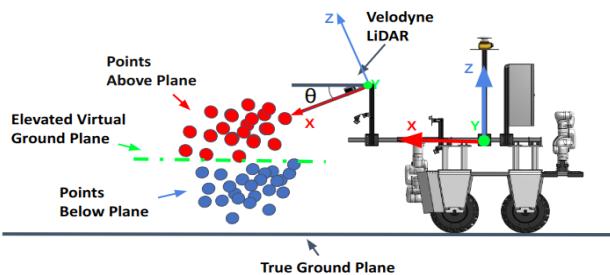


Fig. 5. Animated figure to illustrate the method of generating virtual ground plane. Both red and blue points represent the 3D points from the previous LiDAR FOV filter. After we generate and elevate the ground plane to intersect the centroid of the point clouds, We filter out the point clouds below the estimated ground plane (blue points) and obtain a simplified LiDAR dataset(red points)

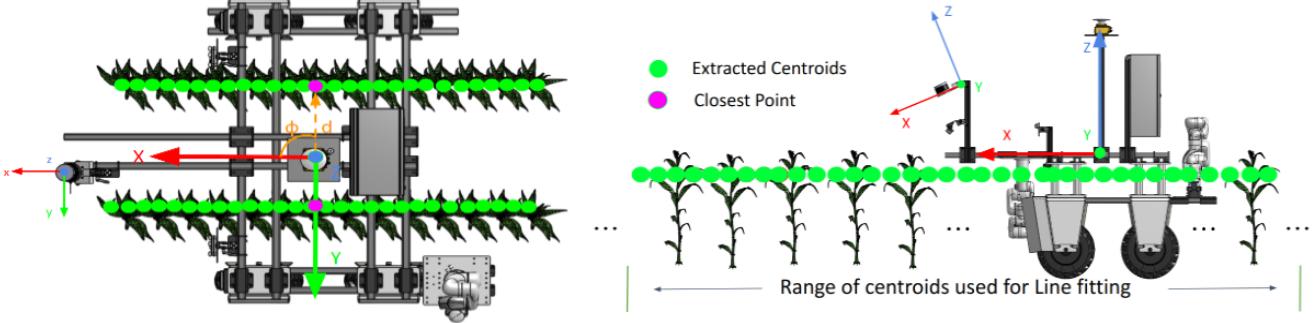


Fig. 6. Illustration of our algorithm extracting detected centroids for applying RANSAC line fitting algorithm and the state  $[d, \phi]$  for the EKF. As shown in the robot's top view (left), our algorithm extracts the centroids of the first row on the left and the first row on the right in the robot frame. Then, as shown in the robot's side view (right), we perform the RANSAC line fitting algorithm between the current farthest detected centroids and previous centroids in a certain range (e.g. 0.5m behind the robot).

comprises three key components. Firstly, we estimate the ground plane based on the known LiDAR tilted angle  $\theta$  by assuming a flat ground in the crop field. Elevating the generated ground plane to intersect the centroid of LiDAR point clouds and filtering out points below the plane leaves returns from the tops of the plants, making it easier to determine where the rows are, as shown in Fig. 4. Secondly, employing the filtered LiDAR data, we apply the K-means clustering algorithm to segment crop rows autonomously. The generated centroids of these segments represent the center of crop rows. We utilize robot odometry and IMU to map recently detected centroids into the current robot frame. This crop-row map facilitates the RANSAC line fitting algorithm, extracting 3D line locations for the first row on the left and the first right row on the right. Finally, we implement an Extended Kalman Filter (EKF) to enhance the accuracy and robustness of the extracted 3D crop row positions based on both current and previous robot odometry and crop row predictions. Detailed explanations of each step are provided in the subsequent subsections.

*1) LiDAR Data Preprocessor:* We employ a multi-step preprocessing approach to address the challenge of differentiating crop rows within raw LiDAR data. First, we apply a LiDAR FOV filter that only extracts the points within the

frontal  $120^\circ$  and 0 to 4m depth range. To exploit the height difference between crops and interrow spacing for crop row detection, we generate a virtual plane that intersects the centroid of the point clouds. As shown in Fig. 5, this virtual plane, parallel to the ground plane in the LiDAR frame, is generated based on the known LiDAR tilted angle  $\theta$  along the LiDAR Y-axis(leftward). We filter out the point clouds located below this plane as:

$$v_{\perp} = R_{y,\theta} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} ; \quad d = v_{\perp} \cdot \bar{P}_{x,y,z} \quad (1)$$

$$P_f = \{p \mid p_x v_{\perp,x} + p_y v_{\perp,y} + p_z v_{\perp,z} + d > 0\} \quad (2)$$

We determine the normal vector of the ground plane in the LiDAR frame  $v_{\perp}$  by multiplying the rotation matrix  $R_{y,\theta}$  of the ground plane in the LiDAR frame and the unit normal vector  $[0, 0, 1]^T$ . To elevate the ground plane to intersect the centroid of the LiDAR point clouds, we calculate the constant  $d$  along the normal vector of the ground plane by multiplying the  $v_{\perp}$  and the centroid of the point clouds  $\bar{P}_{x,y,z}$ . In the end, we filter out all the points below the elevated ground plane and obtain filtered LiDAR data  $P_f$ .

*2) K-means Clustering and Localization Integration:* We categorize LiDAR points into bins for each 1m depth range. For each bin, the K-means clustering algorithm is employed, and the centroid  $Q_i$  of each detected cluster is computed,  $i \in \{1, \dots, n\}$ , where  $n$  is the number of detected cluster. We store these centroids in each time step into  $Q_{\text{local}}$  within the current robot frame based on the robot odometry and IMU data. We apply the K-means filter that merges the centroids with Euclidean distance lower than 0.3m to detect the number of crop rows autonomously. The number of centroids  $n$  indicates the number of detected crop rows, and each centroid lies on the center line of the associated row.

*3) RANSAC Line Fitting:* Due to the sparsity of centroids detected in each time frame, we track centroids' local positions from previous frames to enhance the smoothness and accuracy of currently predicted crop rows, as shown in Fig. 6. Initially, the robot advances without guidance. Once the robot reaches the centroids' local positions that were

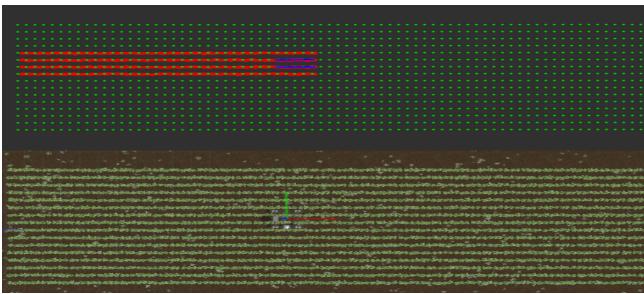


Fig. 7. Visualization of the LiDAR row detection algorithm (top) while robot operating in the simulated young soybean fields (bottom). In the visualization tool, green points represent the ground truth positions of crops in the global frame, red points denote detected centroids, and the two blue lines indicate predicted positions and orientations of the first crop row on the left and the first crop row on the right.

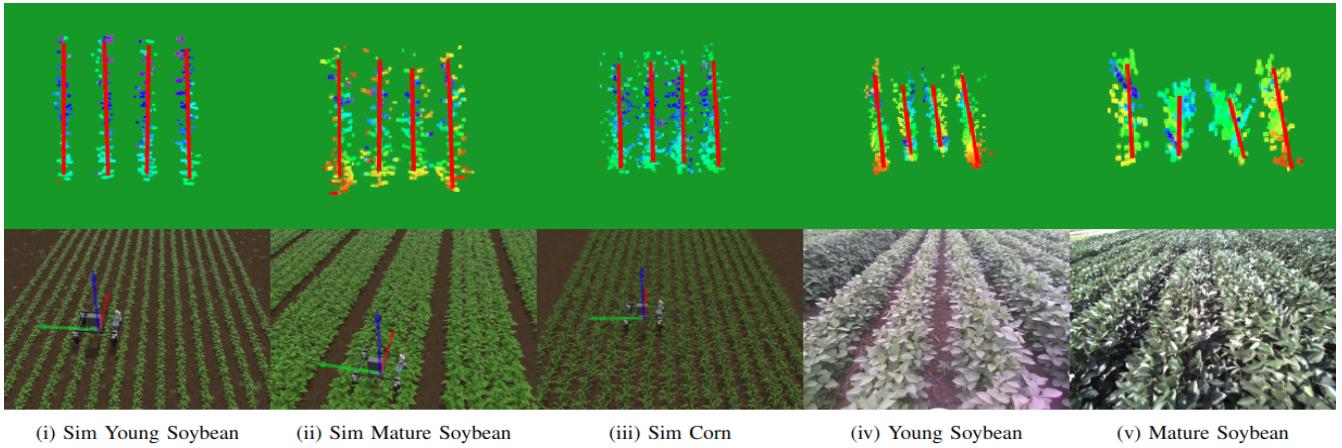


Fig. 8. Visualization of the point clouds after filtering the point below the generated virtual plane from simulated fields (young soybean(i), mature soybean(ii), and corn(iii)) and real fields (young soybean(iv) and mature soybean(v)). In each subfigure, the top one is the top view of the filtered point clouds, as humans can easily differentiate the crop rows(red lines). The bottom is the picture of the test field during the crop row detection experiments

detected at the robot's start, we extract the centroids' local positions from the farthest currently detected centroids to the centroids located  $0.5m$  behind the robot from  $Q_{\text{local}}$ . Notably, we exclusively extract the centroids' local positions of the first row on the left and the first row on the right in the robot's frame. We categorize and store centroids into two sets, each corresponding to a crop row. Then, we apply the RANSAC line fitting algorithm to each set to predict the positions and orientations of the crop rows, as shown in the two blue lines in Fig 7. The output prediction of each crop row is in the form of  $[x_1, y_1, x_2, y_2]$ , where  $[x_1, y_1]$  represent coordinates of the far end of the predicted line and  $[x_2, y_2]$  represent coordinates of the close end of the predicted line, both in the robot's frame.

*4) Extended Kalman Filter:* Assuming the crop row is straight, we can represent the crop row effectively using polar coordinates in the form of  $[d, \phi]$ . As illustrated in Fig. 6,  $d$  represents the closest distance between the robot and the crop row, and  $\phi$  is the angle from the robot heading orientation to the vector pointing to the closest point on the crop row.

To enhance the accuracy of the crop row tracking, We implement an Extended Kalman Filter (EKF) on both crop rows and utilize  $[d, \phi]$  as the state for the EKF. The robot

pose difference in each time step is calculated based on the robot odometry data, denoting as  $[\Delta X, \Delta Y, \Delta \psi]$ . In each time step  $k \in \{1, \dots, n\}$ , our EKF predicts the current state  $\mathbf{X}_k$  based on previous state  $\mathbf{X}_{k-1}$  and robot pose difference  $[\Delta X, \Delta Y, \Delta \psi]$  through the state prediction matrix as:

$$\mathbf{X}_k = \begin{bmatrix} \mathbf{d}_{k-1} - \Delta X \cos(\phi_{k-1}) - \Delta Y \sin(\phi_{k-1}) \\ \phi_{k-1} - \Delta \psi \end{bmatrix} \quad (3)$$

The covariance matrices in the EKF are tuned to represent the uncertainty associated with the state estimates. When the new crop row predictions from the RANSAC line fitting algorithm become available, the EKF incorporates the new measurements to update the predicted state and output the current state. The output state delivers a more accurate and smooth crop row prediction for the crop row following algorithm later.

#### B. Multiple Crop Row Following System

After the position and orientation of detected crop rows undergo filtering using an Extended Kalman Filter to attain higher accuracy and robustness based on robot odometry and centroids from previous frames, waypoints are generated in the center of the two filtered crop row predictions by implementing a path-planning algorithm. This approach explains the extraction of only the first left and the first right crop row positions in the crop row detection algorithm. Finally, we employ pure pursuit [15] to navigate the robot along the generated waypoints, effectively following the detected multiple crop rows. We opt for a lookahead distance set to 60% of the predicted lines from the EKF node. The application of pure pursuit enables the robot to follow the continuously generated waypoints adeptly, showcasing robust navigation ability along crop rows.

## V. EXPERIMENTAL RESULTS AND DISCUSSION

For the crop row detection algorithm, we conduct experiments in both the actual soybean fields and the Gazebo simulated fields. The navigation system is tested in the

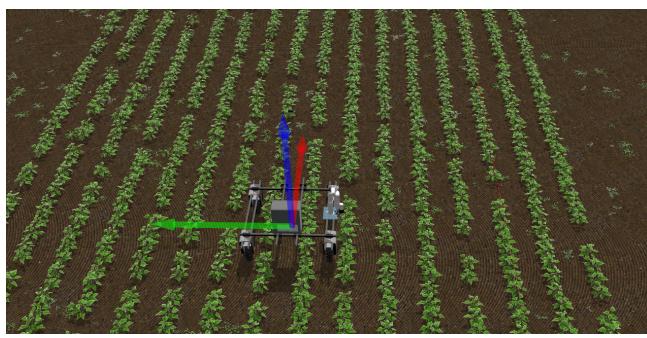


Fig. 9. Simulated young soybean fields with weeds presence among the crop rows and discontinuities within the rows.

simulated fields. The simulation environments are built in the Gazebo simulator with different crop types and growth stages(corn, young soybean, and mature soybean) by using the Cropercraft tool [16]. As shown in Fig. 9, we also implement two challenging scenarios: weeds between the rows and discontinuities within the crop rows. These scenarios are designed to rigorously test our navigation system, evaluating its performance under conditions that mimic challenges possibly faced in real agricultural fields. While testing the crop row detection algorithm, a human operates the Amiga robot to transverse the fields. The subsequent crop row following experiments are conducted without human intervention during navigation.

To simulate navigation in real agricultural fields realistically, We implement simulated fields and the Amiga robot on a 1:1 scale. In both simulation and reality, the LiDAR is positioned at a height of 1.5m above the ground and a tilted angle of 30°. We utilize the cuML package (GPU-accelerated) to ensure real-time operation on the K-means clustering algorithm while maintaining compatibility with CPU-only machines. The following subsections present and discuss the experiment results with evaluation metrics.

#### A. Multi-crop-row Detection and metric

We conduct experiments to test the robustness of the crop row detection algorithm against various field conditions. These experiments are performed with the Amiga robot in the Gazebo simulation environments (corn, young soybeans, and mature soybeans). All the simulated fields are constructed with 0.762m interrow spacing and a total dimensions of 50 m × 12 m.

During the experiments, a human operator controls the Amiga robot to traverse the agricultural field. Fig. 7 provides an example visualization of our crop row detection algorithm while the Amiga robot is controlled to follow the crop rows in the simulated corn field. The visualization tool provides information on detected crop positions in the current robot frame(red points), crop ground truth positions(green points), and predicted crop row positions and orientations(blue lines). The ground truth positions of the crop are derived from the crop model positions in the simulated fields. To assess the performance of the crop row detection algorithm, we employ Mean Absolute Error (MAE) with standard deviation metric to calculate the distance error and the angular error between the predicted crop positions and the crop ground truth positions.

Based on results in Table I, our method achieves an aver-

TABLE I  
CROP ROW DETECTION PERFORMANCE IN SIMULATED FIELDS.

Crop	Length	$\mu \pm \sigma$ of dist to crop rows	$\mu \pm \sigma$ of angular error
Corn	50m	$2.82 \pm 0.91\text{cm}$	$1.18 \pm 0.67^\circ$
Young Soybean	50m	$1.74 \pm 0.87\text{cm}$	$0.89 \pm 0.59^\circ$
Mature Soybean	50m	$4.12 \pm 1.31\text{cm}$	$1.29 \pm 0.85^\circ$

TABLE II  
CROP ROW FOLLOWING PERFORMANCE IN SIMULATED FIELDS.

Crop	Length	$\mu \pm \sigma$ of dist to desired path
Corn	50m	$3.18 \pm 1.71\text{cm}$
Young Soybean	50m	$1.26 \pm 0.46\text{cm}$
Mature Soybean	50m	$4.17 \pm 3.31\text{cm}$

age detection accuracy of 2.89cm when predicting the crop rows' positions across various simulated agricultural fields. Similarly, the prediction of crop row orientation achieves an average accuracy of 1.12° across the same fields. The mature soybean field poses the most challenging scenario for crop row detection due to the complete blocking of interrow spacing by its canopy. Most existing vision-based crop row detection algorithms fail for this reason [9]. In our simulated mature soybean field, we emulate this canopy property of the mature soybean, and we successfully detected the mature soybean crop rows with an average accuracy of 4.12cm. Due to the lack of the ground truth positions of the crop in the actual fields, we apply our crop row detection algorithm on the point clouds data from both simulated fields and actual fields to obtain qualitative results as shown in Fig. 8. These qualitative results showcase the feasibility of our crop row detection algorithm in detecting various crops in both simulated fields and actual fields. A potential limitation of our crop row detection approach arises when crops are in the germination stage, leading to significant gaps between each crop and minor height differences between crops and the ground. Despite this, our crop row detection approach demonstrates its ability to detect multiple crop rows in various challenging agricultural fields reliably.

#### B. Multi-crop-row Following and metric

To evaluate the robustness of the crop row following algorithm, we perform experiments in the same simulated fields. During these experiments, we activate both crop row detection and crop row following algorithms. Once the detection algorithm predicts the crop rows' positions and orientations, the crop row following algorithm (path planning and pure pursuit) will start to control the Amiga robot without human intervention.

Our three simulated fields (corn, young soybean, and mature soybean) present challenges of various crop types and growth stages encountered in actual agricultural fields. To evaluate the performance of autonomous navigation, a perfect case is established for comparison. We assume the optimal performance during row navigation is achieved by maintaining the robot's position at the center of the crop rows. Leveraging precise knowledge of the crop positions in the simulated fields, the desired path for autonomous navigation can be calculated. The robot path during the autonomous navigation is obtained from the filtered robot odometry message  $[x, y, \psi]$  [17]. We apply Mean Absolute Error (MAE) with standard deviation as the metric to evaluate the crop row following performance, quantifying the error

between robot odometry and the desired path. The results are presented in Table II.

A suboptimal crop row following algorithm, causing oscillations during navigation, can lead to failures in the crop row detection algorithm to track the crop rows accurately and thus make the whole system fail. As depicted in the results in Table II, our navigation system successfully achieves autonomous navigation within the crop rows with an average error of  $2.98\text{cm}$  from the center of the crop rows in all simulated fields without human intervention and without causing harm to the crops. This minor deviation from the ideal crop row path demonstrates the feasibility of our navigation system as a crop-agnostic autonomous navigation approach in various real-world crop fields.

## VI. CONCLUSION

In this paper, we present a novel row detection approach that integrates the Extended Kalman Filter (EKF), pure pursuit, and lane-switching algorithm to create an autonomous navigation system for agricultural robots in row-crop fields. This system facilitates independent robot navigation for diverse agricultural tasks, such as harvesting and crop monitoring, contributing to precision farming. Leveraging the observed height difference between crops and interrow spacing in real agricultural fields, our approach utilizes LiDAR data to extract the height information and accurately detect crop rows amidst challenging scenarios such as different crop types, discontinuities, weeds presence, and canopy obstructions. The navigation system incorporates the crop row following algorithm, enabling automated tracking of detected crop rows. Additionally, the lane-switching algorithm is implemented to transition the robot to the next lane upon detecting the end of the current lane. The navigation system is evaluated on the 1:1 scale Amiga robot in the Gazebo simulations with the challenging scenarios mentioned above. The crop row detection algorithm achieves an average detection accuracy of  $2.89\text{cm}$  between detected crop rows and the ground truth. The crop row following algorithm achieves an average driving accuracy of  $2.98\text{cm}$  between robot odometry and the center line of crop rows. As part of future work, we aim to enhance the robustness of the crop row perception algorithm by incorporating camera data and exploring the implementation of Model Predictive Control (MPC) for improving crop row following performance.

## VII. ACKNOWLEDGMENTS

We would like to thank Srinivasan Vijayarangan from Carnegie Mellon University and Aaron Berger from Harvard University for their help in our work. We would also like to thank Iowa State University for providing field research opportunities in the Agronomy farm. This work was supported by: NSF/USDA NIFA AIIRA AI Research Institute 2021-67021-35329.

## REFERENCES

- [1] C. Mouél and A. Forslund, “How can we feed the world in 2050? a review of the responses from global scenario studies,” *European Review of Agricultural Economics*, vol. 44, 2017.
- [2] J. P. Vasconez, G. A. Kantor, and F. A. Auat Cheein, “Human–robot interaction in agriculture: A survey and current challenges,” *Biosystems Engineering*, vol. 179, pp. 35–48, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1537511017309625>
- [3] R. Moeller, T. Deemyad, and A. Sebastian, “Autonomous navigation of an agricultural robot using rtk gps and pixhawk,” in *2020 Intermountain Engineering, Technology and Computing (IETC)*, 2020, pp. 1–6.
- [4] H. Lan, M. Elsheikh, W. Abdelfatah, A. Wahdan, and N. El-Sheimy, “Integrated rtk/ins navigation for precision agriculture,” *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*, pp. 4076–4086, 9 2019.
- [5] P. V. Santhi, N. Kapileswar, V. K. R. Chenchela, and C. H. V. S. Prasad, “Sensor and vision based autonomous agribot for sowing seeds,” in *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, 2017, pp. 242–245.
- [6] K. G. Fue, W. M. Porter, E. M. Barnes, and G. C. Rains, “An extensive review of mobile agricultural robotics for field operations: Focus on cotton harvesting,” *AgriEngineering*, vol. 2, no. 1, pp. 150–174, 2020. [Online]. Available: <https://www.mdpi.com/2624-7402/2/1/10>
- [7] M. Bayati and R. Fotouhi, “A mobile robotic platform for crop monitoring,” *Advances in Robotics & Automation*, vol. 7, no. 1, p. 1000186, 2018.
- [8] A. English, P. Ross, D. Ball, and P. Corke, “Vision based guidance for robot navigation in agriculture,” in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1693–1698.
- [9] A. Ahmadi, M. Halstead, and C. McCool, “Towards autonomous crop-agnostic visual navigation in arable fields,” *arXiv preprint arXiv:2109.11936*, 2021.
- [10] F. B. Malavazi, R. Guyonneau, J.-B. Fasquel, S. Lagrange, and F. Mercier, “Lidar-only based navigation algorithm for an autonomous agricultural robot,” *Computers and Electronics in Agriculture*, vol. 154, pp. 71–79, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168169918302679>
- [11] F. Rovira-Más, V. Saiz-Rubio, and A. Cuenca-Cuenca, “Augmented perception for agricultural robots navigation,” *IEEE Sensors Journal*, vol. 21, no. 10, pp. 11 712–11 727, 2021.
- [12] A. Ahmadi, M. Halstead, and C. McCool, “Towards autonomous visual navigation in arable fields,” in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 6585–6592.
- [13] A. N. Sivakumar, S. Modi, M. V. Gasparino, C. Ellis, A. E. B. Velasquez, G. Chowdhary, and S. Gupta, “Learned visual navigation for under-canopy agricultural robots,” in *Robotics: Science and Systems*, 2021.
- [14] D. Guri, M. Lee, O. Kroemer, and G. Kantor, “Hefty: A modular reconfigurable robot for advancing robot manipulation in agriculture,” 2024.
- [15] R. C. Conlter, “Implementation of the pure pursuit path tracking algorithm,” in *1990 Carnegie Mellon*. Carnegie Mellon University, January 1992.
- [16] C. Pierre, “cropcraft,” <https://github.com/Romea/cropcraft>, 2024.
- [17] T. Moore and D. Stouch, “A generalized extended kalman filter implementation for the robot operating system,” in *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*. Springer, July 2014.