

# Visual Servoing-based Navigation for Monitoring Row-Crop Fields

Alireza Ahmadi

Lorenzo Nardi

Nived Chebrolu

Cyrril Stachniss

**Abstract**—Autonomous navigation is a pre-requisite for field robots to carry out precision agriculture tasks. Typically, a robot has to navigate through a whole crop field several times during a season for monitoring the plants, for applying agrochemicals, or for performing targeted intervention actions. In this paper, we propose a framework tailored for navigation in row-crop fields by exploiting the regular crop-row structure present in the fields. Our approach uses only the images from on-board cameras without the need for performing explicit localization or maintaining a map of the field and thus can operate without expensive RTK-GPS solutions often used in agriculture automation systems. Our navigation approach allows the robot to follow the crop-rows accurately and handles the switch to the next row seamlessly within the same framework. We implemented our approach using C++ and ROS and thoroughly tested it in several simulated environments with different shapes and sizes of field. We also demonstrated the system running at frame-rate on an actual robot operating on a test row-crop field. The code and data have been published.

## I. INTRODUCTION

Autonomous agricultural robots have the potential to improve farm productivity and to perform targeted field management activities. In crop fields, agricultural robots are typically used to perform monitoring tasks [15], [19] or targeted intervention such as weed control [26], [17]. Several crops such as maize, sugar beet, sunflower, potato, soybean, and many others are arranged along multiple parallel rows in the fields as illustrated in Fig. 1. This arrangement facilitates cultivation, weeding, and other farming operations. For accomplishing such tasks, robots must be able to autonomously navigate through the crop-rows repeatedly in the field.

Currently, a popular solution for navigating autonomously in fields is to use a high-precision, dual-frequency RTK-GNSS receiver to guide the robot along pre-programmed paths. However, the high cost of these systems and vulnerability to outages has led to an interest in solutions using observations from on-board sensors. Such solutions typically use observations from a laser scanner or a camera to localize the robot in the environment and then navigate along crop rows, often with the help of a map. The crop field scenario poses serious challenges to such systems due to high visual aliasing in the fields and lack of reliable sensor measurements of identifiable landmarks to support localization and mapping tasks. Additionally, as the field is constantly changing due to the growth of plants, a map of the field needs to be updated several times during a crop season.

In this paper, we address navigation in row-crop fields only based on camera observations and by exploiting the row



Fig. 1: Robot navigation in a test row-crop field. Top-right: on-board camera image. The visual servoing based controller executes the velocity control that brings the crop row (red arrow) to the center of the camera image (green arrow). The blue box shows the sliding window used for tracking the row along which the robot navigates.

structure inherent in the field to guide the robot and cover the field. An example illustrating our robot navigating along a crop-row is shown in Fig. 1. We aim at controlling the robot without explicitly maintaining a map of the environment or performing localization in a global reference frame.

The main contribution of this paper is a novel navigation system for agricultural robots operating in row-crop fields. We present a visual servoing-based controller that controls the robot using local directional features extracted from the camera images. This information is obtained from the crop-row structure, which is continuously tracked through a sliding window. Our approach integrates a switching mechanism to transition from one row to the next one when the robot reaches the end of a crop-row. By using a pair of cameras, the robot is able to enter a new crop-row within a limited space and avoids making sharp turns or other complex maneuvers. As our experiments show, the proposed approach allows a robot to (i) autonomously navigate through row-crop fields without the maintaining any global reference maps, (ii) monitor the crops in the fields with a high coverage by accurately following the crop-rows, (iii) is robust to fields with different row structures and characteristics, as well as to critical user-defined parameters.

Note that the source code of our approach, the data from the real-world experiments as well as the simulated environment are available at: <http://github.com/PRBonn/visual-crop-row-navigation>.

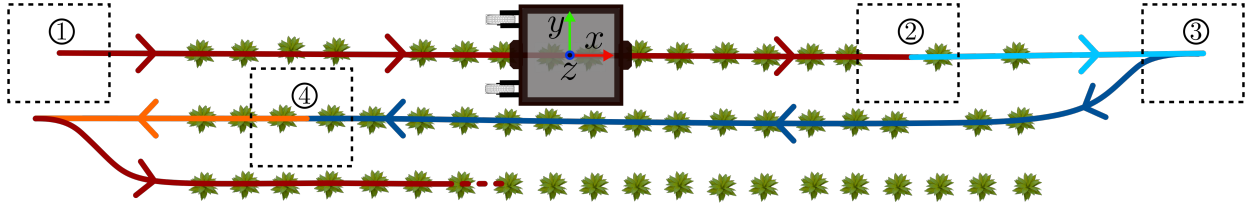


Fig. 2: Scheme for navigation in a crop field: the robot enters the field and navigates along a crop row (①), exits the row (②), transitions to the next crop row (③), and exits the row on the opposite side (④).

## II. RELATED WORK

Early autonomous systems for navigation in crop fields such as the one developed by Bell [3] or Thuilot *et al.* [22] are based on GNSS while others use visual fiducial markers [20] or artificial beacons [16]. More recently, agricultural robots equipped with a suite of sensors including GNSS receiver, laser-scanner, and camera have been used for precision agriculture tasks such as selective spraying [23] or targeted mechanical intervention [13]. Dong *et al.* [9] as well as Chebrolu *et al.* [5] address the issue of changing appearance of the crop fields and proposed localization systems, which fuse information from several on-board sensors and prior aerial maps to localize over longer periods of time. While most of these systems allow for navigation in crop fields accurately, they require either additional infrastructure or reference maps for navigation. In contrast to that, our approach only requires local observations of the crop rows from obtained from a camera.

Other approaches also exploit the crop row structure in the fields and developed vision based guidance systems for controlling the robot [4] and perform weeding operations in between the crop rows [1]. These methods require a reliable crop row detection system as they use this information for computing the control signal for the robot. As a result, several works focus on detecting crop rows from images under challenging conditions. Winterhalter *et al.* [24] propose a method for detecting crop rows based on hough transform and obtain robust detections even at an early growth stage when are plants are very small. English *et al.* [10] and Sogaard *et al.* [21] show reliable crop row detections in the presence of many weeds and under challenging illumination conditions. Other works such as [18], [12], [14] aim at estimating the stem locations of the plants accurately. While we use a fairly simple method for crop row detection in this paper, more robust methods for detection can be easily integrated in our approach for dealing with challenging field conditions with a high percentage of weeds. The proposed controller is independent of the used detection technique.

Traditionally, visual servoing techniques [11] are used for controlling robotic arms and manipulators. These techniques aim to control the motion of the robot by directly using vision data in the control loop. Cherubini *et al.* [6], [7], [27] propose visual servoing techniques for the controlling mobile robots along continuous paths. De Lima *et al.* [8] apply these visual servoing techniques to autonomous cars for following lanes in urban scenarios, whereas Avanzini *et al.* [2] control a platoon of cars using a guiding vehicle. We have built upon

these ideas to develop our controller for crop field navigation including a mechanism to transition from one row to the next row within the same framework.

## III. NAVIGATION ALONG ROW-CROP FIELDS

In this paper, we consider a mobile robot that navigates in row-crop fields to perform tasks such as monitoring of the crops or removing the weeds and has to cover the field row by row. Thus, the robot must be able to navigate autonomously along all the crop rows in the field.

### A. Navigation Along Crop Rows

In row-crop fields, crops are arranged along multiple parallel curves. We take advantage of such an arrangement to enable a mobile robot to autonomously navigate and monitor the crops in the field. The main steps to achieve this are illustrated in Fig. 2. The robot starts from one of the corners of the field (①), enters the first crop row and follows it until the end (②). As it reaches the end of a row, it needs to enter the next one and follow it in the opposite direction (③). This sequence of behaviors is repeated to navigate along all the crop rows in the field. We achieve this by using a visual-based navigation system that integrates all of these behaviors which leads the robot to autonomously navigate and monitor the crops in a row-crop field.

### B. Robotic Platform

We consider a mobile robotic platform that is able to navigate on crop fields by driving seamlessly forward and backwards. We equip this robot with two cameras  $\mathcal{F}_{C_{\text{front}}}$  and  $\mathcal{F}_{C_{\text{back}}}$  mounted respectively looking to the front and to the back of the robot as illustrated in Fig. 3. The cameras are symmetric with respect to the center of rotation of the robot denoted as  $\mathcal{F}_R$ . The cameras have a fixed tilt angle  $\rho$  and are positioned on the robot at a height  $t_z$  from the ground and with an horizontal offset  $t_y$  from  $\mathcal{F}_R$ . A camera image is illustrated in Fig. 3(c), where  $W$  and  $H$  are respectively its width and height in pixels.

## IV. OUR NAVIGATION APPROACH

We propose to use a visual-based navigation system that relies only local visual features and exploits the arrangement of the crops in fields to autonomously navigate in row-crop fields without requiring an explicit map. Our visual-based navigation system builds upon the image-based visual servoing controller by Cherubini *et al.* [6] and extends it by considering an image that presents multiple crop-rows and integrating a mechanism for switching to the next row.

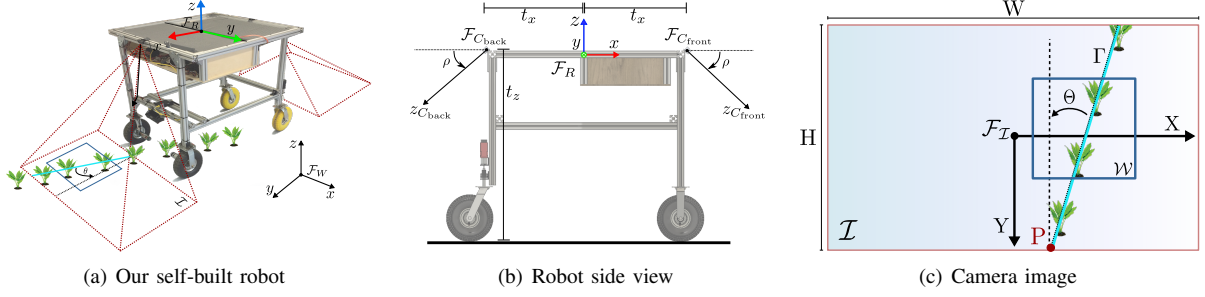


Fig. 3: Robot, frames and variables. The robot navigates by following the path (cyan) along the crop row. In (a),  $\mathcal{F}_W$  and  $\mathcal{F}_R$  are world and robot frames,  $\theta$  is the orientation of the robot in  $\mathcal{F}_W$ . In (b),  $\mathcal{F}_{C_{\text{front}}}$ ,  $\mathcal{F}_{C_{\text{back}}}$  are the front and back camera frames. The cameras are mounted at an offset  $t_x$  from the robot center  $\mathcal{F}_R$  and  $t_z$  above the ground, and with tilt  $\rho$ . In (c),  $\mathcal{F}_I$  is the image frame and  $s = [X, Y, \Theta]$  is the image features computed from the crop row.

#### A. Visual Servoing Controller

Visual servoing allows for controlling a robot by processing visual information. Cherubini *et al.* [6] propose an image-based visual servoing scheme that allows a mobile robot equipped with a fixed pinhole camera to follow a continuous path on the ground. It uses a combination of two primitive image-based controllers to drive the robot to the desired configuration.

We define the robot configuration as  $q = [x, y, \theta]^T$ . The control variables are the linear and the angular velocity of the robot  $u = [v, \omega]^T$ . We impose that the robot moves with a fixed constant translational velocity  $v = v^*$ . Thus, our controller controls only the angular velocity  $\omega$ .

The controller computes the controls  $u$  by minimizing the error  $e = s - s^*$ , where  $s$  is a vector of features computed on the camera image and  $s^*$  is the desired value of the corresponding features. The state dynamics are given by:

$$\dot{s} = J u = J_v v^* + J_\omega \omega, \quad (1)$$

where  $J_v$  and  $J_\omega$  are the columns of the Jacobian  $J$  that relate  $u$  to  $\dot{s}$ , the controller computes the controls by applying the feedback control:

$$\omega = -J_\omega^+ (\lambda e + J_v v^*), \quad \lambda > 0, \quad (2)$$

where  $J_\omega^+$  indicates the Moore-Penrose pseudo-inverse of  $J_\omega$ .

From the camera image, we compute an image feature  $s = [X, Y, \Theta]$  illustrated in Fig. 3(c) where  $P = [X, Y]$  is the position of the first point along the visible path and  $\Theta$  is the orientation of the tangent  $\Gamma$  to the path. We use uppercase variables to denote the quantities in the image frame  $\mathcal{I}$ . The controller computes a control  $u$  such that it brings  $P$  to the bottom center of the image  $s^* = [0, \frac{H}{2}, 0]$ . The desired configuration corresponds to driving the robot along the center of the path. The image feature and its desired position are illustrated in Fig. 1 (top-right).

The interaction matrix  $L_s$  allows for relating the dynamics of the image features  $s$  to the robot velocity in the camera frame  $u_c$ . The velocity in the camera frame  $u_c$  can be expressed as a function of the robot velocity  $u$  as  $u_c = {}^C T_R u$ , where  ${}^C T_R$  is the homogeneous transformation from  $\mathcal{F}_R$  to  $\mathcal{F}_C$ . Therefore, we can write the relation between the

image feature dynamics  $\dot{s}$  and the robot controls  $u$  as:

$$\dot{s} = L_s u_c = L_s {}^C T_R u. \quad (3)$$

#### B. Crop Row Detection for Visual Servoing

The visual-servoing approach described in the previous section allows the robot to follow a continuous path drawn on the ground. In fields, we can exploit the arrangement of the crops in rows to enable the robot to navigate using a similar visual-servoing scheme.

To navigate along a crop-row, we extract the curve along which the crops are arranged. To this end, for each new camera image we first compute the vegetation mask using the Excess Green Index (ExG) [25] often used in agricultural applications. It is given by  $I_{ExG} = 2I_G - I_R - I_B$  where  $I_R$ ,  $I_G$  and  $I_B$  correspond to the red, green and blue channels of the image. For each connected component in the vegetation mask, we compute a center point of the crop. We then estimate the path curve along which the robot should navigate by computing the line that best fits all the center points using a robust least-squares fitting method. This procedure allows for continuously computing a path curve in the image that the robot can follow using the visual-servoing controller described in Sec. IV-A. In this paper, we use a fairly straight-forward approach to detect the crop rows as our main focus has been on the design of the visual servoing controller and more sophisticated detection methods are easily implementable.

Typically, fields are composed by number of parallel crop-rows. Thus, multiple rows can be visible at the same time in the camera image. This introduces ambiguity to identify the curve that the robot should follow. This ambiguity may cause the robot to follow a different crop-row before reaching the end of the current one. If this is case, there is no guarantee that the robot will navigate through the whole field. To remove this ambiguity, we use a sliding window  $\mathcal{W}$  of fixed size in the image that captures the row that the robot is following, as illustrated on the bottom of Fig. 1. For every new camera image we update the position of the window  $\mathcal{W}$  by centering it at the average position of the crops detected in that frame. Updating this window continuously allows for tracking a crop row and ensures that the robot follows it up to its end.

---

**Algorithm 1** Crop row navigation scheme

---

```
1:  $\mathcal{W} \leftarrow \text{INITIALIZEWINDOW}$   $\triangleright$  Initialization.
2: repeat  $\triangleright$  Control loop.
3:    $\text{crops}_{\mathcal{P}} \leftarrow \text{DETECTCROPS}(\text{cam}_{\mathcal{P}})$ 
4:    $\text{crops}_{\mathcal{W}} \leftarrow \text{CROPSINWINDOW}(\text{crops}_{\mathcal{P}}, \mathcal{W})$ 
5:   if  $\text{ISEMPTY}(\text{crops}_{\mathcal{W}})$  then
6:     if  $\text{ISEMPTY}(\text{DETECTCROPS}(\text{cam}_{\mathcal{S}}))$  then
7:        $\mathcal{W} \leftarrow \text{SHIFTWINDOW}$   $\triangleright$  Enter next row.
8:     else  $\triangleright$  Exit row.
9:        $\text{SWITCHCAMERAS}(\text{cam}_{\mathcal{P}}, \text{cam}_{\mathcal{S}})$ 
10:       $\mathcal{W} \leftarrow \text{INITIALIZEWINDOW}$ 
11:       $\text{crops}_{\mathcal{P}} \leftarrow \text{DETECTCROPS}(\text{cam}_{\mathcal{P}})$ 
12:       $\text{crops}_{\mathcal{W}} \leftarrow \text{CROPSINWINDOW}(\text{crops}_{\mathcal{P}}, \mathcal{W})$ 
13:       $\text{FOLLOWCROW}(\text{crops}_{\mathcal{W}})$ 
14:       $\mathcal{W} \leftarrow \text{UPDATEWINDOW}(\text{crops}_{\mathcal{W}})$ 
15: until  $\text{ISEMPTY}(\text{crops}_{\mathcal{W}})$   $\triangleright$  Stop navigation.
```

---

### C. Scheme for Autonomous Navigation in Crop-Row Fields

The visual-based navigation system described in the previous section allows the robot to navigate by following a single crop row. To cover the whole field, the robot should be able to transition to the next row upon reaching the end of the current one. However, as the rows are not connected to each other, the robot has no continuous path curve to follow over the whole field. Therefore, we introduce a visual-based navigation scheme that allows the robot to follow a crop-row, to exit from it, and to enter the next one by exploiting both the cameras mounted on the robot and its ability to drive both in forward and backward directions. Our scheme to navigate in crop fields is illustrated in Alg. 1.

The visual-servoing controller described in section Sec. IV-A uses the image of one camera to compute the controls. We extend this approach to using both the front camera  $\mathcal{F}_{C_{\text{front}}}$  and the back camera  $\mathcal{F}_{C_{\text{back}}}$ . We set in turn the camera used by the visual-servoing controller, which we refer to as the primary camera  $\text{cam}_{\mathcal{P}}$ . Whereas, we denote the other camera as the secondary camera  $\text{cam}_{\mathcal{S}}$ . We define the size of the sliding window  $\mathcal{W}$  used by the controller and a shift offset to capture the next crop row based on the tilt angle of the camera  $\rho$  and an estimate of the distance between the crop rows  $\delta$ .

Our navigation scheme assumes that the starting position of the robot is at one of the corners of the field (see for example ① in Fig. 2). We initially set the camera looking in the direction of the field as the primary camera  $\text{cam}_{\mathcal{P}}$  and initialize the position of the window  $\mathcal{W}$  at the center of the image (line 1 of Alg. 1).

In the control loop (line 2), we first detect the centers of the crops  $\text{crops}_{\mathcal{P}}$  in the image of the primary camera (line 3) using the approach described in Sec. IV-B. We select the crops in the image that lie within the window  $\mathcal{W}$ ,  $\text{crops}_{\mathcal{W}}$  (line 4). The robot navigates along the crop row by computing the line that fits the  $\text{crops}_{\mathcal{W}}$  and follows it using the visual servoing controller (line 13). Then, it updates the position of the sliding window  $\mathcal{W}$  in the image at the average position of

the  $\text{crops}_{\mathcal{W}}$  (line 14). This corresponds to the robot following the red path in Fig. 2. When the robot approaches the end of the row (②), the primary camera does not see crops anymore as it is tilted to look forward.

In this position, the secondary camera can still see the crops belonging to current row (line 8). Therefore, we switch the primary and secondary camera, re-initialize the window, re-compute the detected crops, and drive the robot in the opposite direction to which the primary camera is facing. This setup guides the robot to exit the crop row (light blue path in Fig. 2) until it does not detect crops anymore in the window  $\mathcal{W}$  (③). At this point, the secondary camera also does not see any crops and the robot needs to enter in the next crop row. Therefore, we shift the sliding window in the direction of the next row (line 6) to capture the crops in it. By doing this, the robot starts tracking the next crop row and can navigate by following it (blue path in Fig. 2). When no crops are present in the sliding window (④), the robot switches the camera as in ②, exits the row and shift  $\mathcal{W}$  to start following the next one. This control loop repeats until the robot reaches the end of the field and can not see any crop row with both its cameras.

Note that our navigation scheme allows the robot to transition from one crop row to the next one only by switching the cameras and without requiring the robot to perform a complex maneuver to enter the next row. Furthermore, following our navigation scheme the robot requires a smaller space for maneuvering than the one that it would require to perform a sharp U-turn.

## V. EXPERIMENTAL EVALUATION

The experiments are designed to show the capabilities of our method for navigation in row-crop fields and to support our key claims, which are: (i) autonomous navigation through row-crop fields without the need of maintaining any global reference map or an external positioning system such as GNSS, (ii) monitoring the crops with a high coverage by accurately following the rows in fields with different row structures, (iii) robustness to fields with varying properties and to the input parameters provided by the user.

### A. Experimental Setup

In the experiments, we consider our self-built agricultural robot as well as a Clearpath Husky. Both robots are equipped with two monocular cameras placed in the front and the back of the robot with the camera tilt  $\rho$  set to  $75^\circ$ . Our agricultural robot uses a laptop as the main computational device along with a Raspberry Pi 3 as a device communication manager. We implemented our approach on a real robot using C++ and ROS. We also created a simulated version of the robot in Gazebo, which is built on a 1:1 scale and has the same kinematics as the real robot. We have generated several simulated crop fields of different shapes and sizes and evaluated our navigation system both on the simulated crop fields, as well as on the real robot. The experiments with the Clearpath Husky are provided here as it is a common



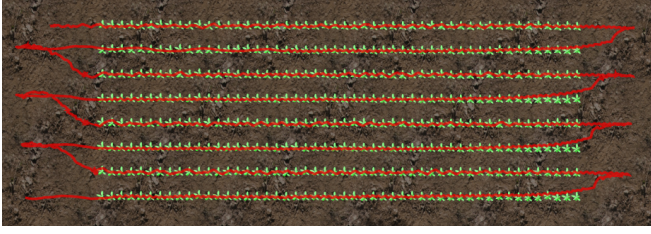


Fig. 4: Trajectory of the robot following our visual-based navigation scheme in a simulated field environment.

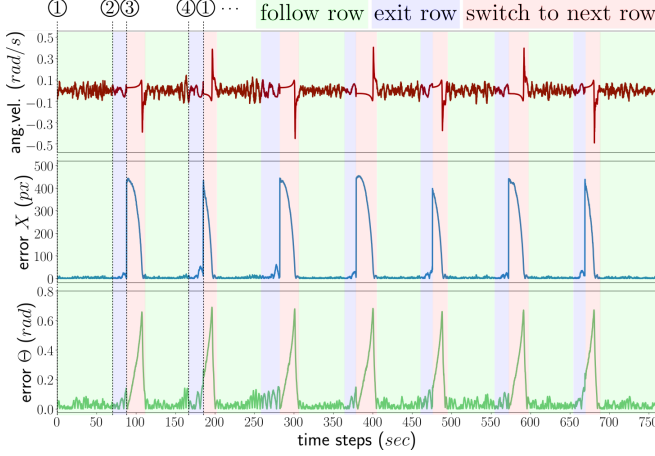


Fig. 5: Angular velocity control (top), error in  $X$  (middle) and error in  $\Theta$  (bottom) computed by the visual servoing controller to navigate along the trajectory illustrated in Fig. 4. We highlighted the steps of our navigation scheme as in Fig. 2 and the robot behaviors.

platform in the robotics community and thus easier for the reader to interpret the results.

### B. Navigation in Crop Fields

The first experiment is designed to show that we are able to autonomously navigate through crop fields using our navigation system. To this end, we use our simulated robot that navigates in a crop field environment in Gazebo. We consider a test field with a dimension of  $20\text{m} \times 10\text{m}$  composed by 8 rows as illustrated in Fig. 4. The rows have an average crop-row distance of 50 cm and a standard deviation of 5 cm. The crops were distributed along the row with a gap ranging from 5 cm to 15 cm to mimic real crop-rows. In our setup, the robot starts at the beginning of the first crop row in top left corner of the field. The goal consists of reaching the opposite corner by navigating through each row.

The trajectory along which the robot navigated is illustrated in Fig. 4. The robot was successfully able to follow each of the rows until the end and transition to the next ones until it covered the whole field. At the end of the row, the robot was able to transition to the next row within an average maneuvering space of around 1.3 m. Thus, our navigation scheme allows the robot to enter the next row using a limited maneuvering space which is often a critical requirement while navigating in a field.

In Fig. 5, we illustrate the error signals in  $X$  and  $\Theta$ , which was used by the visual servoing controller to compute the angular velocity  $\omega$  for the robot. Both, the error signals in  $X$

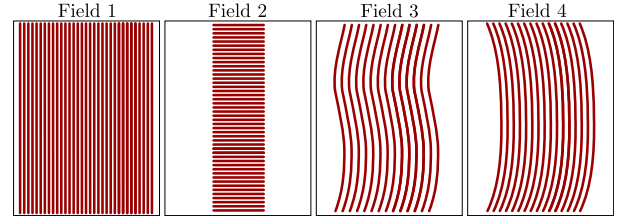


Fig. 6: Fields with different shapes used in the experiments. Field 1 is large and long; Field 2 is short and requires many turns in quick succession; Field 3 is S-shaped; Field 4 is parabola shaped.

Field 1	$0.651 \pm 0.826$ cm	2.28	100%
Field 2	$0.872 \pm 1.534$ cm	2.17	100%
Field 3	$0.814 \pm 1.144$ cm	2.46	100%
Field 4	$0.830 \pm 1.155$ cm	4.38	100%
Avg. and std. dev. distance to crop rows		Avg. missed crops per row	Percentage of visited rows

TABLE I: Field coverage analysis in the fields illustrated in Fig. 6 using our navigation scheme.

and  $\Theta$  show peaks at the times which correspond to the transition to the next crop row (see for example ③). This is the normal behavior whenever a new row is selected and the robot must align itself to the new row. The controller compensates for this error using large values of  $\omega$  as shown in Fig. 5 (top). Also, note that the direction of  $\omega$  is flipped at the end of each row since the robot alternates between a left and right turn to go through the crop rows. In this experiment, we demonstrated that our navigation scheme allows the robot to autonomously monitor a row-crop field by accurately following the crop-rows.

### C. Field Coverage Analysis

The second experiment is designed to evaluate the capability of a robot using our navigation scheme to cover fields of different shapes and lengths. We consider four different fields to evaluate our navigation scheme, which are shown in Fig. 6. Field 1 presents a typical scenario having long crop rows, whereas Field 2 is a short but wide, which requires the robot to turn several times in quick succession. Field 3 and 4 have curved crop rows which are typically found in real world fields. To evaluate the navigation performance, we consider the following metrics: (i) the average and standard deviation of the distance of the robot from the crop rows, (ii) the average number of crops per row missed by the robot and, (iii) the percentage of crop rows in the field missed by the robot during navigation.

Tab. I summarizes the performance of our navigation scheme for each of the four fields. The robot is able to navigate along all of the fields with an average distance from the crop rows of 0.8 cm and a standard deviation of 1.15 cm (without relying on any map or an external positioning system). This shows that the robot was able to follow the crop rows closely during traversal. The number of crops covered by the robot is computed by considering all of the crops that are within a range of 10 cm from the trajectory of the robot. This threshold ensures that the crops we are monitoring are visible in the center region of the image for our robot setup.

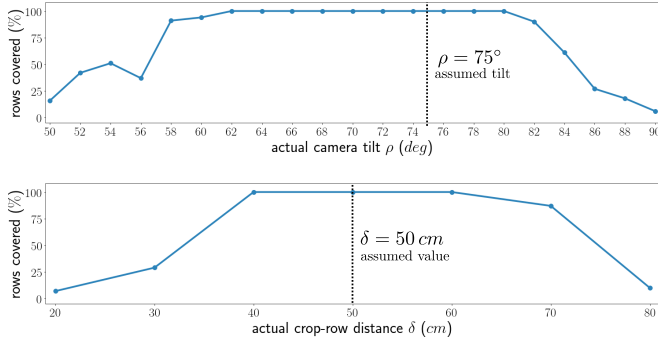


Fig. 7: Robustness of our navigation system to errors in camera tilt  $\rho$  (top) and crop-row distance  $\delta$  (bottom) with respect to the assumed values.

In all of the fields, the average number of plants missed per row is negligible with respect to the number of plants in a real crop row. These missed crops are the ones which the robot misses while entering the next row as shown in Fig. 4.

Finally, we also evaluate the number of crop rows in the field that were visited by the robot. We consider a row to be visited only if the robot misses less than 5 crops in a row (i.e. it does not take a shortcut at the end of the row). For each of the four fields, the robot was able to traverse all of the crop rows successfully. These results indicate that our system is able to monitor the crops in the field with a high coverage even in fields presenting different characteristics.

#### D. Robustness to User-Defined Parameters

In this experiment, we evaluate the robustness of our navigation scheme to the critical parameters which needs to be provided by the user. The first parameter is the camera tilt angle  $\rho$  which is used by the visual servoing controller for following the crop row. Another critical parameter is the crop row distance  $\delta$  which the navigation scheme uses for estimating the shift of the window  $\mathcal{W}$  at the end of the row. The crop row distance  $\delta$  may not be accurately measured or varies in different parts of the field. Therefore, to account for these errors, we analyzed the robustness of our navigation scheme to the camera tilt angle  $\rho$  and the row distance  $\delta$ .

To analyze the robustness of the navigation scheme, we use the percentage of the crop rows in the field traversed by the robot. This measure is indicative of how successful the robot is in transitioning to the next row. To test the first parameter, we fix the camera tilt  $\rho$  in the visual servoing controller to  $75^\circ$  and vary the actual tilt of the camera on the robot in the range from  $50^\circ$  to  $90^\circ$ . In Fig. 7 (top), we observe that the robot is able to traverse all the crop-rows in the field (100% coverage) for  $\rho$  in the range from  $62^\circ$  to  $80^\circ$ . This range corresponds to an error in the camera tilt varying from  $-13^\circ$  to  $+5^\circ$  with respect to the assumed value. Thus, we suggest to know the tilt parameter up to  $\pm 5^\circ$  which is easy to achieve in practice.

For the second parameter, we assume the crop row distance  $\delta$  to be 50 cm in the controller and compute the shift for the window  $\mathcal{W}$  based on this value. We evaluate the robustness of the system to this parameter by considering

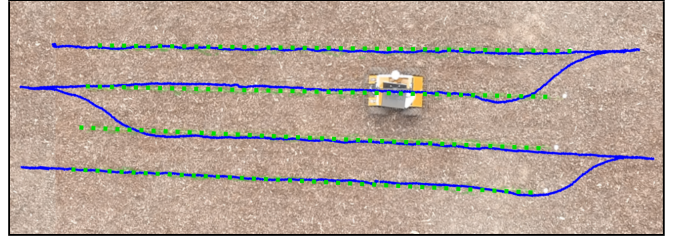


Fig. 8: Real robot following our visual-based navigation scheme to navigate in a test row-crop field, and the trajectory (blue) of the robot recorded with a RTK-GPS.

fields with a crop-row distances  $\delta$  that range from 20 cm to 80 cm. We observe in Fig. 7 (bottom) that the robot is able to perform all the transition to the next rows successfully for  $\delta$  varying from 40 cm to 60 cm. This corresponds to a difference of  $-10$  cm to  $+10$  cm from the assumed  $\delta$  which is a reasonable variation considering that most fields are sown with precision agricultural machines today. These results indicate that our system is robust to reasonable error which is expected in real-world scenarios.

#### E. Demonstration on Real Robot

In the last experiment, we demonstrate our navigation system running on a real robot operating in an outdoor environment. For this experiment, we used a *Clearpath Husky* robot equipped with two cameras arranged in the configuration shown in Fig. 1. All the computations were performed on a consumer grade laptop running ROS. We setup a test-field with 4 parallel rows each 15 meters on a rough terrain. The robot was able to successfully navigate along the crop-rows and switch to the correctly the end of each crop row. We recorded the robot's trajectory using a RTK-GPS system but only to visualize it by overlaying on a aerial image of the field as shown in Fig. 8. We observed that the robot traverses the rows by navigating close to the crops rows within a deviation of 4 cms and transition to the next rows within a average maneuvering length of 1.2 m at the start/end of each row.

## VI. CONCLUSION

In this paper, we presented a novel approach for autonomous robot navigation in crop fields, which allows an agricultural robot to carry out precision agriculture tasks such as crop monitoring. Our approach operates only on the local observations from the on-board cameras for navigation. Our method exploits the row structure inherent in the crop fields to guide the robot along the crop row without the need for explicit localization system, GNSS, or a map of the environment. It handles the switching to new crop rows as an integrated part of the control loop. This allows the robot to successfully navigate through the crop fields row by row and cover the whole field. We implemented and evaluated our approach on different simulated datasets as well as on a self-built agricultural robot. The experiments suggest that our approach can be used by agricultural robots in crop fields of different shapes and is fairly robust to the critical user defined parameters of the controller.

## REFERENCES

- [1] B. Åstrand and A. J. Baerveldt. A vision based row-following system for agricultural field machinery. *Mechatronics*, 15(2):251–269, 2005.
- [2] P. Avanzini, B. Thuilot, and P. Martinet. Accurate platoon control of urban vehicles based solely on monocular vision. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2010.
- [3] T. Bell. Automatic tractor guidance using carrier-phase differential gps. *Computers and Electronics in Agriculture*, 25(1-2):53–66, 2000.
- [4] J. Billingsley and M. Schoenfisch. The successful development of a vision guidance system for agriculture. *Computers and Electronics in Agriculture*, 16(2):147–163, 1997.
- [5] N. Chebrolu, P. Lottes, T. Laebe, and C. Stachniss. Robot Localization Based on Aerial Images for Precision Agriculture Tasks in Crop Fields. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019.
- [6] A. Cherubini, F. Chaumette, and G. Oriolo. An image-based visual servoing scheme for following paths with nonholonomic mobile robots. In *Proc. of the Intl. Conf. on Control, Automation, Robotics and Vision*, pages 108–113, 2008.
- [7] A. Cherubini, F. Chaumette, and G. Oriolo. A position-based visual servoing scheme for following paths with nonholonomic mobile robots. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2008.
- [8] D. de Lima and A. Victorino. A visual servoing approach for road lane following with obstacle avoidance. In *Proc. of the Intl. Conf. on Intelligent Transportation*, pages 412–417, 2014.
- [9] J. Dong, J.G. Burnham, B. Boots, G. Rains, and F. Dellaert. 4D Crop Monitoring: Spatio-Temporal Reconstruction for Agriculture. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2017.
- [10] A. English, P. Ross, D. Ball, and P. Corke. Vision based guidance for robot navigation in agriculture. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2014.
- [11] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Trans. on Robotics and Automation*, 8(3):313–326, 1992.
- [12] S. Haug, P. Biber, A. Michaels, and J. Ostermann. Plant stem detection and position estimation using machine vision. In *Workshop Proc. of Conf. on Intelligent Autonomous Systems (IAS)*, pages 483–490, 2014.
- [13] M. Imperoli, C. Potena, D. Nardi, G. Grisetti, and A. Pretto. An effective multi-cue positioning system for agricultural robotics. *IEEE Robotics and Automation Letters*, 3(4):3685–3692, Oct 2018.
- [14] F. Kraemer, A. Schaefer, A. Eitel, J. Vertens, and W. Burgard. From Plants to Landmarks: Time-invariant Plant Localization that uses Deep Pose Regression in Agricultural Fields. In *IROS Workshop on Agri-Food Robotics*, 2017.
- [15] K. Kusumam, T. Krajník, S. Pearson, T. Duckett, and G. Cielniak. 3d-vision based detection, localization, and sizing of broccoli heads in the field. *Journal of Field Robotics*, 34(8):1505–1518, 2017.
- [16] J. J. Leonard and H. F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Trans. on Robotics and Automation*, 7(3):376–382, 1991.
- [17] C.S. McCool, J. Beattie, J. Firn, C. Lehnert, J. Kulk, R. Russell, T. Perez, and O. Bawden. Efficacy of mechanical weeding tools: A study into alternative weed management strategies enabled by robotics (2). *IEEE Robotics and Automation Letters (RA-L)*, 2018.
- [18] H. S. Midtby, T. M. Giselsson, and R. N. Jrgensen. Estimating the plant stem emerging points (pseps) of sugar beets at early growth stages. *Biosystems Engineering*, 111(1):83 – 90, 2012.
- [19] A. D. Nakarmi and L. Tang. Within-row spacing sensing of maize plants using 3d computer vision. *Biosystems Engineering*, 125:54–64, 2014.
- [20] E. Olson. Apriltag: A robust and flexible visual fiducial system. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2011.
- [21] H. T. Sogaard and H. J. Olsen. Determination of crop rows by image analysis without segmentation. 38(2):141–158, 2003.
- [22] B. Thuilot, C. Cariou, L. Cordesses, and P. Martinet. Automatic guidance of a farm tractor along curved paths, using a unique cp-dgps. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, volume 2, pages 674–679. IEEE, 2001.
- [23] J. P. Underwood, M. Calleja, Z. Taylor, C. Hung, J. Nieto, R. Fitch, and S. Sukkarieh. Real-time target detection and steerable spray for vegetable crops. pages 26–30, 2015.
- [24] W. Winterhalter, F. Fleckenstein, C. Dornhege, and W. Burgard. Crop row detection on tiny plants with the pattern hough transform. *IEEE Robotics and Automation Letters (RA-L)*, 3(4):3394–3401, 2018.
- [25] D. Woebbecke, G. Meyer, K. Von Bargen, and D. A. Mortensen. Color indices for weed identification under various soil, residue, and lighting conditions. *Trans. of the American Society of Agricultural Engineers (ASAE)*, 38(1):259–269, 1995.
- [26] X. Wu, S. Aravecchia, and C. Pradalier. Design and implementation of computer vision based in-row weeding system. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 4218–4224, May 2019.
- [27] Yi Ma, J. Kosecka, and S. S. Sastry. Vision guided navigation for a nonholonomic mobile robot. *IEEE Transactions on Robotics and Automation*, 15(3):521–536, June 1999.