REGULAR ARTICLE

WILEY

Localization for precision navigation in agricultural fields-Beyond crop row following

Wera Winterhalter | Freya Fleckenstein | Christian Dornhege | Wolfram Burgard

Department of Computer Science, University of Freiburg, Freiburg, Germany

Correspondence

Freya Fleckenstein, Department of Computer Science, University of Freiburg, Freiburg 79110. Germany. Email: fleckenf@informatik.uni-freiburg.de

Abstract

The growing world population calls for more efficient and sustainable farming technologies. Automating agricultural tasks has great potential to improve farming technologies. A key requirement for full automation is the ability of agricultural vehicles to accurately navigate entire fields without damaging value crops. One important precondition for autonomous navigation is localization, that is, the ability of a vehicle to accurately estimate its pose relative to the crops. A majority of localization approaches detect crop rows to track the heading and lateral offset of the vehicle. This is sufficient to guide the vehicle along crop rows while driving inside the field. However, switching between rows requires a longitudinal pose estimate to determine when to turn at the end of the field. Additionally, at the end of the field sensor data contains less crop row structure and more noise from wild growing vegetation. This can lead to false-positive crop row detections. In this paper, we present a localization approach that goes beyond state-of-the-art crop row following algorithms by providing robust pose estimates not only inside the field but also at the end of the field. The underlying concept of our approach is to estimate the vehicle pose relative to a global navigation satellite system (GNSS)referenced map of crop rows. This allows us to fuse crop row detections with GNSS signals to obtain a pose estimate with the accuracy comparable to a row following approach in the heading and lateral offset, while at the same time maintaining at least GNSS accuracy along the row. Employing a GNSS-referenced map of crop rows poses several challenges. To relate the detected crop rows to those in the map, we propose a data association strategy that finds correspondences between two sets of lines, that is, crop rows. Furthermore, we improve the GNSS-based longitudinal pose estimate by detecting the end of the field from vegetation data. Additionally, we introduce a novel method to determine false-positive crop row detections to increase the overall robustness in particular in challenging scenarios at the end of the field. Extensive real-world experiments on three different types of crops demonstrate that our localization approach is well suited for fully autonomous navigation in entire fields.

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes. © 2020 The Authors. Journal of Field Robotics published by Wiley Periodicals LLC

1 | INTRODUCTION

Automation in agriculture has rapidly gained interest over the recent years as we need more and more food to sustain the growing world population. Despite the technological progress in agriculture over the last decades, there are still many tasks that involve a lot of hard manual labor or could be performed more efficiently. These range from manually planting small seedlings over manually harvesting delicate vegetables or fruits, for example, salad, asparagus or strawberries to large-scale application of herbicides or fertilizer. Automating these tasks is a big step towards sustainability as it will reduce the workload of farmers and at the same time enable them to increase yield and reduce the amount of resources used. All these tasks have one common denominator: Instead of uniformly performing the same task in the same way on the whole field, they require the ability to treat plants or parts of the field individually. With manual labor alone, individual treatment is time consuming and tedious and thus only performed if absolutely unavoidable. If these tasks could be performed autonomously by a robot that is able to work continuously, with only short breaks, irrespective of the time of day, individual treatment is possible.

A key ability for any robot performing such tasks is safe autonomous navigation in an agricultural environment. As soon as a robot has the ability to reliably traverse agricultural fields, the aforementioned tasks and many others can be built on top of the navigation system. An important requirement for such an autonomous system is its ability to localize itself, that is, to estimate its pose with respect to the crops in the field. A navigation system with this capability can steer the robot safely along a desired path through the field to execute its task without damaging any agricultural crop.

A widely used solution to this problem is to rely purely on high precision global navigation satellite system (GNSS) sensors. The GNSS signal allows to track the position of the robot with high precision relative to the global GNSS coordinate frame. To steer the robot safely along the target field, corresponding systems rely on the assumption that a predefined reference GNSS path is available. This path is usually generated by recording the trajectory of the vehicle during seeding. After seeding, the system can execute the same recorded GNSS path repeatedly. The major drawback of such a purely GNSS-based approach is that it does not know its pose relative to the crop or crop rows, which can easily change after seeding, for example, due to heavy rain or simply uneven growth. Thus, executing the same GNSS path as during seeding might not be safe. Also, to enable more individual treatment of plants on agricultural fields for higher sustainability, many applications like mechanical weeding or harvesting need the system to be aware of the vehicle pose relative to the crop. Thus, it is highly relevant to design agricultural systems so that they base their pose estimate not solely on high accuracy GNSS, but also take into account their pose relative to the crop.

Pose estimates relative to crops are computed by taking into account data from sensors that perceive the local environment around the robot, for example, from cameras or lidar sensors. State-of-the-art approaches that are designed to actively guide a robot along the crop rows of an agricultural field estimate a heading and lateral offset of the robot relative to the detected crop rows. These computations are usually performed in the frame of the perceived sensor data and then directly transformed into steering correction signals. In contrast to a purely GNSS-based system, such *row following* approaches are applied in high precision agriculture. Since they directly convert the sensor input into steering correction signals, there is no need for a global reference frame or map of the field. This is an advantage, since—in contrast to the GNSS-based approach—no prior knowledge of the field is required (for example no reference GNSS path).

However, as soon as the robot approaches the end of the field, the front facing sensor used to detect the crop rows will perceive less of the traversed field. Instead, it also captures parts of the area at the edges of the field where the vehicles can enter or leave crop rows, called headlands. The vegetation on the headlands varies for every field: no vegetation at all, grass or straw, or even bushes and trees. This additional vegetation can easily confuse a vegetationbased crop row detection algorithm. Also, the crop rows of neighboring fields might be detected. Both cause false-positive detections. Finally, when the robot reaches the end of the field, the crop rows of the traversed field are not visible in the sensor data anymore. In such a situation, a pure row following approach has to stop the robot since no reliable steering correction signals can be sent anymore. For these approaches, the robot has to be manually steered to leave the current row, turn at the headlands and reenter the adjacent row. A fully autonomous system has to be able to perform these maneuvers by itself. Thus, we employ an explicit localization algorithm in our navigation system.

The underlying concept of our method is a GNSS-referenced map of crop rows (*semantic map*). We specifically design our localization to combine the advantages of row following and GNSS-based approaches. Using such a semantic map allows us to relate crop row detections from local sensor data with global GNSS measurements. As a result of this the heading and lateral offset accuracy of our method are comparable to a row following approach. At the same time, our method maintains at least GNSS accuracy along a row. Additionally, information like odometry and measurements from an inertial measurement unit (IMU) can be easily integrated to further improve the pose estimate. A visualization of our approach is given in Figure 1.

There are other approaches addressing the problem of localizing and tracking the pose of an agricultural vehicle. Most of them are based on high precision GNSS data, some also use odometry and IMU measurements and very few integrate vision or lidar data as well. However, those localization algorithms are designed to provide accurate maps featuring a multitude of different properties of the field. These maps can then be analyzed to answer questions like: Should the farmer put more fertilizer on certain areas of the field? Is weeding or another kind of plant treatment necessary? What amount of yield is to be expected? Are the crops growing as fast as expected? Since the target application of these approaches is mapping and not autonomous navigation, they do not provide any evaluation on the localization accuracy relative to crops. It is unclear if those approaches would be suited to safely guide an agricultural vehicle across a field.

FIGURE 1 Beyond crop row following. These images illustrate the behavior of our localization approach when leaving (left two images) and re-entering (right two images) the field. The detected pattern (vertical red lines) is used as an input for our localization algorithm to correct heading and offset of the robot relative to a GNSS-referenced map (green lines). Additionally, on the left side, the detected end of field (horizontal red line) is used to correct the longitudinal pose estimate along the crop rows [Color figure can be viewed at wileyonlinelibrary.com]

In this paper, we present a localization that enables an agricultural vehicle to autonomously traverse entire fields, including leaving the crop rows, turning on the headlands and re-entering the field. We base our approach on the idea of crop row following, and use a GNSS-referenced map to go beyond the restrictions of naive row following. Such an approach has to deal with the following challenges.

First, the localization has to find correspondences between the detected crop rows and the rows of the semantic map to compute a correction measurement for the pose estimate. This is challenging since there are no unique features to correlate the detected rows with the rows in the semantic map. After finding the correspondences, we need to define how to correct the heading and offset of the robot using these correspondences.

Second, for the robot to safely transition between rows, we need to estimate the pose along the crop rows. If the longitudinal pose estimate is too far off, the robot might initiate a turning maneuver too early, that is, while still in the field, and crush valuable crops. If the robot starts turning too late, it might end up turning on a neighboring field, or leave the traversable area of the headlands. A longitudinal correction measurement cannot be derived from crop rows directly as they are geometrically represented as lines. Thus, additional information is needed to estimate the longitudinal position.

Third, the localization needs to handle incorrect or false-positive crop row detections. During transition between rows, the crop row structure is only partially visible in the sensor data and thus less information is available to compute their direction and offset (see Figure 1 right two images). This might lead to incorrect detections. Additional vegetation like grass or bushes might be visible in the sensor data, which can confuse the crop row detection algorithm and produce false-positives.

We tackle those challenges and thereby contribute to state-ofthe-art localization approaches for autonomous navigation in agricultural fields as follows:

 We present an approach that goes beyond the restrictions of naive row following by also localizing with respect to a GNSSreferenced map of crop rows.

- To this end, we introduce a method to integrate detected crop rows into the localization algorithm. This entails a data association approach for two sets of lines and defining an error measure for the resulting line to line correspondences. This data association relies on geometric relations between the lines of each set to find consistent correspondences. The localization computes the corrected pose estimate using our error measure between correspondences of detected and mapped rows.
- We employ the following modalities to estimate the pose along the crop rows: First, we use our GNSS-referenced semantic map to integrate GNSS signals into the longitudinal pose estimate. For narrow headlands, an accuracy of a standard GNSS, that is, within 3 m, might not suffice. Thus, we also detect the end of the field from vision or lidar data to further improve the longitudinal pose estimate.
- We propose a quality estimate for a set of crop rows. This quality states how well a detected set of crop rows is supported by the sensor data. If the support is not sufficient, we declare the detection to be a false-positive.
- We evaluate the results of our localization on a field that contains three different kinds of plants. Our evaluation shows that our proposed localization enables fully autonomous navigation in agricultural fields.

Since our approach is based on row following, a robust and accurate crop row detection method is crucial. We therefore investigate different state-of-the-art crop row detection algorithms and evaluate their detection accuracy to estimate a lower bound for the localization accuracy. In the following, we thus give a detailed overview of state-of-the-art agricultural localization approaches as well as state-of-the-art crop row detection methods. In the next sections, we describe different state-of-the-art crop row detection methods and explain how we integrate their results into our localization approach: First, we present our data association for two sets of lines and define the error measure for the resulting line to line correspondences. Second, we define the heading and lateral offset correction for the localization using our error measure. Then, we introduce our end of field detection and formulate how to integrate the longitudinal correction measurement from GNSS signals or the end of field detection into the localization. Afterward, we present our method for determining false-positive crop row detections. In the experimental evaluation section, we first investigate the performance of the described state-of-the-art crop row detection methods. The results of this evaluation give us a lower bound on the heading and lateral offset localization accuracy. We then evaluate the performance of our localization algorithm employing the best performing crop row detection algorithm. We compare our localization to pure row following and GNSS-based approaches. The results confirm that our approach enables autonomous navigation in agricultural fields—beyond crop row following.

2 | RELATED WORK

Automation in agriculture has gained interest in recent years. There are many approaches that estimate the pose of an agricultural ground vehicle. Most of these approaches use the pose estimate to create maps of the field. These maps are then analyzed offline to determine different properties of the field and its crops. In these scenarios, the vehicle is often manually steered to gather data of interest using a multitude of different sensor modalities. During the data recording, not only the sensor data, but also a highly accurate GNSS position of the vehicle is stored. In an offline processing step, state-of-the-art techniques are then used to compute maps of the traversed field. Equipping the vehicle with a high precision GNSS sensor to obtain the necessary accuracy is a key requirement for these approaches.

In the work of Hague et al. (2006), the GNSS-referenced data was used to evaluate the crop and weed density, respectively. This information can then be used to determine, if and which areas of the field require weed treatment. The high precision GNSS information, together with a robust data association in RGB images, can also be used to overlay the computed maps over time (Dong et al., 2017). Maps that stretch over multiple weeks or even months hold valuable information like the speed of crop growth. Baia et al. (2016) also use a high precision GNSS sensor to acquire information about different phenotypes of soybean and wheat. There are many more approaches that leverage high precision GNSS to create maps for the purpose of phenotyping (Mueller-Sim et al., 2017; Ruckelshausen et al., 2009; Underwood et al., 2017). However, these works also state some drawbacks of relying purely on a global GNSS signal for pose estimation: According to the work of Mueller-Sim et al., the precision of the GNSS signal is affected if the antenna is covered by plants. As a remedy, they propose to use lidar or vision data in addition to GNSS in future work. For similar reasons, Ruckelshausen et al. state that they will investigate the use of lidar data to increase the robustness of the pose estimate. Another requirement when deploying a high precision GNSS sensor for long-term navigation is a fixed local reference to align the GNSS reference path from one deployment to the next (Underwood et al., 2017; Watanabe, 2018). A recent work by Watanabe investigates how to remedy this dependency. However, the author's solution still requires a local reference marker. Here, the

acquired pose estimate depends on the accuracy of the GNSS position of the reference marker.

The findings from the above approaches suggest that under optimal conditions (no GNSS outages and the availability of a fixed infrastructure), autonomous navigation based solely on high precision GNSS is possible. Nevertheless, to achieve robust long-term autonomous navigation fusing additional information from lidar or vision instead of purely relying on high precision GNSS seems advisable.

Compared to research on pose estimation for mapping and phenotyping, research focusing on pose estimation for autonomous navigation in agriculture is scarce. Most approaches for localization in agriculture use high precision GNSS as the primary sensor. Some also leverage information from depth or color data. For example, Biber et al. (2012) use crop rows detected from lidar data to guide the autonomous vehicle along the crop rows. When turning at the end of the field, they switch to GNSS-based localization. Similarly, Bakker et al. (2011) have two separate algorithms for pose estimation: One based on the high precision GNSS signal and the other based on the crop rows detected by their algorithm presented in Bakker et al. (2008). They evaluate the accuracy of both approaches with respect to the crop rows. They reach centimeter accuracy with the high precision GNSS and errors up to 0.1 m for their row following approach while the image data contains only the field and no headland. These errors are within the bounds to guide a vehicle along crop rows. However, since the error increases when the vehicle approaches the headlands, this approach is restricted to pose estimation within the crop rows. In English et al. (2013), the authors use a particle filter to estimate the pose of the robot and fuse high precision GNSS signals with their crop row tracking. Later works improve the accuracy and robustness of their method (English et al., 2014, 2015). In their most recent work, the authors state that the high precision GNSS is the primary sensor for their localization algorithm (Ball et al., 2017). The information from the crop rows is used as redundancy measurement or in case of GNSS signal outages. This is also reflected in the evaluation, where localization accuracy is investigated with and without periods of GNSS outage. This approach has an average accuracy of approximately 0.2 m using a customer-grade high precision GNSS. During GNSS outages, the error slowly increases to 1.5 m after 2 min of GNSS outages. However, when evaluating the offset error while tracking the crop rows without GNSS corrections, the accuracy is around 0.1 m and stays below 0.2 m.

These works support that even localization approaches based on high precision GNSS need to take into account row-based information to achieve long-term robustness. Fusing information from both modalities into one pose estimate while maintaining an accuracy of up to 0.1 m relative to the crop, even when approaching the edge of the field, is a key requirement to enable safe guidance of an agricultural vehicle.

There has been some research on localization for navigation in agricultural fields that does not rely on high precision GNSS signals. These approaches are based on detecting the row structure of the field and estimating the pose of the vehicle relative to these

-WILEY

detections. In practice, this works well as long as the vehicle is driving in the field. Some approaches directly convert the extracted rows into steering signals for the vehicle (Baerveldt & Baerveldt, 2005). Furthermore, different sensor modalities like infrared cameras have been investigated to obtain better vegetation segmentation and thus higher robustness for the localization algorithm (Xaud et al., 2019). However, pure row following approaches that directly convert the perceived row structure into steering signals can only provide a steering input as long as a crop row is detected, that is, as long as the vehicle is driving in row. For turning maneuvers at the headlands, a different strategy is required. One possible solution is relying on the readings from the wheel odometry of the platform (Riggio et al., 2018). Another solution is placing markers at the end of each crop row. These markers serve as a reference for the platform, to initiate a turning maneuver and to properly realign with the crop row after turning (Libby & Kantor, 2011).

While placing markers at the end of each row is a viable solution for small research fields, this is tedious and impractical on large production fields. Riggio et al. show that turning solely based on odometry measurements is feasible. These results support that guiding an agricultural vehicle without high precision GNSS signals is possible.

Most recently, Chebrolu et al. presented a localization approach that uses the positions of plants on the field instead of crop rows (Chebrolu et al., 2019). They fuse odometry and plant positions detected from image data to localize on an agricultural field. Representing crops as individual landmarks instead of whole crop rows implicitly enables the localization to track the longitudinal pose estimate along the crop rows and reduces the maximum global localization error. The authors do not provide an evaluation of the lateral tracking accuracy within the crop rows relative to the crops. It remains unclear if a sufficiently low lateral localization error can be reached with this approach to safely navigate with larger agricultural vehicles that only have a clearance of few centimeters to the crop rows.

Similar to previously introduced approaches, we base our localization on row following. While the above approaches depend on artificial markers or are designed for specific plant types, a more general approach is preferable. Thus, we design our localization to detect the end of the field from sensor data instead of relying on markers. Furthermore, it does not depend on a particular crop row detection algorithm, but only takes a set of detected crop rows as input. Several crop row detection techniques that provide suitable input for our localization have been developed.

Most crop row detection approaches assume that the crop rows are sown in equidistant, straight and parallel lines (Åstrand & Baerveldt, 2005; Bakker et al., 2008; Marchant, 1996; Søgaard & Olsen, 2003; Winterhalter et al., 2018). For those based on vision data, perspective projection and distortion leads to nonparallel curves in image space. Performing a rectification and projecting from the image plane into three-dimensional (3D) space recovers the original geometry. Other works only assume that crop rows are sown in straight lines (English et al., 2014, 2015) or that they are sown WILEY

equidistantly (Kise et al., 2005; Montalvo et al., 2012). While either assumption might not be valid for all fields, they do hold on most fields as long as only a small local area is considered. A set of crop rows can then be described by providing their heading, an offset of one of the crop rows to a given reference point, and the interrow spacing.

Many approaches additionally assume that one of these parameters is given to reduce the dimensionality of the problem. Vidović et al. assume that the camera is always aligned with the crop rows so that the heading is fixed (Vidović et al., 2016). This assumption is only correct when driving within a row. To go beyond row following, we require an approach that also detects crop rows when the camera is not aligned with the crop rows. This is the case, that is, when turning towards and entering a field as illustrated in Figure 1 in the right two images.

Other approaches assume that the spacing between crop rows is known (Åstrand & Baerveldt, 2005; Bakker et al., 2008; Kise et al., 2005; Marchant, 1996; Montalvo et al., 2012; Søgaard & Olsen, 2003). Thus, they estimate only the offset and heading of crop rows. This is a valid assumption for crop-specific detection methods. However, different crop types are sown with different spacings, so that these approaches need user input to be flexible with respect to different crop types. Nevertheless, these approaches can be used for autonomous navigation within crop rows (Bakker et al., 2011). Other methods need the number of crop rows that are visible in the image (Leemans & Destain, 2006; Montalvo et al., 2012). Here, the number of crop rows is determined offline, depending on the sensor placement and the spacing between crop rows. Again, user input is needed to be flexible with respect to different crop types and sensor systems.

An approach that is independent of the number of visible crop rows and estimates the heading, spacing and offset of crop rows is presented in English et al. (2014). Here, the image is projected into 3D. Then, the authors sum the pixel values over the columns and determine the variance of these sums. Iteratively skewing the image by varying angles and computing this variance, the correct heading is determined as the corresponding skew with the highest variance. Peaks in the image column sums correspond to crop rows. This approach was extended using depth information from stereo cameras and training a support vector machine to estimate the offset of crop rows (English et al., 2015). Initially, the crop rows of the first image have to be labeled manually before each run. The algorithm then tracks the labeled crop rows and was used for in row localization by Ball et al. (2017).

Another approach that estimates heading, offset and spacing of crop rows assuming parallel, equidistant and straight lines is the Pattern Hough transform (Winterhalter et al., 2018). It is based on the Hough transform for single line detection (Hough, 1962). In contrast to extracting several individual lines with the Hough transform, the Pattern Hough transform extracts a pattern of parallel and equidistant lines jointly.

Overall, many crop row detection methods have been developed and successfully applied for autonomous driving in row. For our localization approach, we are interested in crop row detection methods that extract a set of crop rows and rely only on the assumption that crop rows are locally parallel and equidistant lines. Furthermore, they should produce reliable detections without prior knowledge about heading, offset or spacing of the rows. This is crucial for switching between rows when the sensor is not necessarily aligned with the row structure. For more flexibility of the localization, the underlying crop row detection should generalize well over different crop types. In the following section, we give examples of existing crop row detection methods that satisfy these requirements.

3 | CROP ROW DETECTION FOR LOCALIZATION

For our localization approach, we require an accurate estimate for the heading and lateral offset of crop rows relative to the autonomous vehicle. Since we fuse the detected rows into the localization also during turning maneuvers at the end of the field, the crop row detection needs to reliably extract crop rows including situations where the platform is not aligned with the rows. Furthermore, the detection results should be robust against noise introduced by weed not growing within the row structure as well as vegetation growing at the edges of the field. As discussed in the previous section, there is a wide variety of crop row detection approaches that satisfy these requirements. In this section, we briefly introduce a few examples of crop row detection approaches that can be used with our localization method.

3.1 | Feature maps

To be more flexible with respect to the sensor modality, we preprocess lidar or vision data into a feature map as presented in (Winterhalter et al., 2018). A feature map is a two-dimensional grid map, where each cell holds the weight corresponding to the likelihood of vegetation being present in this cell. Such a feature map is defined in the robot coordinate frame and located on the ground plane. Thus, geometric relations between plants are transferred to the feature map. This ensures that the parallel equidistant crop rows appear as parallel equidistant lines in the feature maps. This is particularly important for vision data, as the perspective projection and distortion in images lets the parallel crop rows appear as nonparallel curves. To extract features from vision data, we use the triangular greenness index (Constantin et al., 2015). The resulting feature maps provide an input to a crop row detection that is independent of the sensor modality. An example of a feature map is shown on the right of Figure 5.

3.2 | Crop row detection methods

A crop row detection suited for localization should determine a set of rows localized in metric space relative to the robot. As input, the crop row detection takes a feature map in the robot coordinate system as described in the previous section. We make use of the assumption that crops are sown such that they locally appear in parallel equidistant lines. Therefore, the goal is to extract a crop row pattern $p_{o,\theta,s}$ of parallel equidistant lines from a feature map. Here, o is the offset of the line closest to the origin of the robot frame, θ is the angle of the pattern lines, and s is the spacing between lines in the pattern (see Figure 2, right).

Many crop row detection methods use the Hough transform for single line detection (Hough, 1962). The Hough transform represents lines in the Hesse normal form,

$$I_{r,\theta} \coloneqq \{(x, y) \in \mathbb{R}^2 | r = x \cdot \cos(\theta) + y \cdot \sin(\theta)\},\tag{1}$$

where θ defines the normal vector $\vec{n}_{\theta} = (\cos(\theta), \sin(\theta))^T$ to the line $l_{r,\theta}$ and r is the signed distance of the line to the origin. An illustration of the Hesse normal form is shown in the left image of Figure 2. The Hough transform determines all possible lines $l_{r,\theta}$ discretized over the offset r and the orientation θ . It then computes the support histogram over all possible lines from a set of points, and thus detects the best fitting line l_{r^*,θ^*} . In our case, the set of points is given by the vegetation features in the feature map. Together with a given spacing s, a pattern is defined as p_{r^*,θ^*,s^*} . We call this method the *Line Hough* transform, which is often used for crop row detection (Åstrand & Baerveldt, 2005; Bakker et al., 2008; Leemans & Destain, 2006; Marchant, 1996).

Instead of assuming a given spacing, a second line can also be extracted parallel to the one detected by the Hough transform. Given the best fitting line l_{r^*,θ^*} extracted by the Hough transform, a second line $l_{\hat{r},\theta^*}$ with $\hat{r} \neq r^*$ is extracted in a similar way. These two lines together define a pattern $p_{r^*,\theta^*,|\hat{r}-r^*|}$. This results in the Dual Line Hough transform.

A different idea is to use all data to jointly estimate a whole pattern at once instead of extracting individual lines. One approach is to compute the support histogram over all patterns instead of over all lines. While in the Hough transform, the support of all possible discretized lines is computed, here, the support of



FIGURE 2 Models for line and pattern detection. Left: The Hesse normal form of a two-dimensional line (red) is defined by the angle θ of the normal to the line and the distance r to the origin. Right: The lines l_0 , l_1 , l_2 (red) are part of a pattern with parameters o (offset), θ (angle), s (spacing). The blue line depicts the distance r_1 of line l_1 from the origin, equal to $1 \cdot s + o$ [Color figure can be viewed at wileyonlinelibrary.com]

all possible discretized patterns $p_{r,\theta,s}$ is computed. Then, the best matching pattern over all data is determined as the maximum of the support histogram. This method is called the *Pattern Hough* transform (Winterhalter et al., 2018).

Another method that jointly estimates a whole pattern from all available data is a variant of the random sample consensus algorithm (RANSAC) (Fischler & Bolles, 1981), the *Pattern RANSAC*. It first defines a line $l_{r,\theta}$ by sampling two feature points in the feature map. It then samples a third feature point, which by its distance to the line defines a pattern spacing *s*. We evaluate the resulting pattern $p_{r,\theta,s}$ by computing the total weight of its support in the feature map. These steps are repeated iteratively for a fixed number of times. The result is the sampled pattern with the highest weight. Similar to the Pattern Hough transform, the Pattern RANSAC therefore finds the pattern that is supported by most data across the feature map.

All these crop row detection methods are potential choices as input for our localization. We investigate their accuracy and robustness in the evaluation section.

4 | LOCALIZATION-BEYOND CROP ROW FOLLOWING

To autonomously traverse a field, including turns at the headlands, a reliable localization with respect to the value crops is required. We present a robust localization that handles imperfect input data as provided by a crop row detection algorithm. In the following, we present two different localization approaches that use the extracted crop rows and additional sensor data to provide accurate position estimates not only within a field but also at the headlands when leaving or entering a field.

Both localization algorithms estimate the 2D pose of the robot in a global coordinate frame by iteratively integrating sensor data. The first approach is based on *gradient descent* (GD) to correct the current pose estimate with a new sensor observation. The second is an *Extended Kalman Filter* (EKF). Both algorithms have a prediction and a correction step. In the prediction step the current estimate \mathbf{x}_t is updated with the measurements of odometry and IMU \mathbf{u}_t to predict the state for the next time step $\hat{\mathbf{x}}_{t+1}$. During the correction step, the predicted state $\hat{\mathbf{x}}_{t+1}$ is corrected using the sensor observation. The result of the correction step is the state estimate \mathbf{x}_{t+1} for time step t + 1.

The GD algorithm directly tracks the robot pose, that is, the state estimate is defined as $\mathbf{x}_t = (\mathbf{x}_t \ y_t \ \theta_t)^T$, where \mathbf{x}_t and \mathbf{y}_t are the position and θ_t the heading of the robot in the global coordinate frame at each time step t. In the prediction step, the relative motion $\mathbf{u}_t = (\Delta \mathbf{x}_t \ \Delta \mathbf{y}_t \ \Delta \theta_t)^T$ measured by odometry for translation $\Delta \mathbf{x}_t$ and $\Delta \mathbf{y}_t$ and IMU for orientation $\Delta \theta_t$ between time step t and t + 1 is applied to the current state estimate \mathbf{x}_t to predict the state at time step t + 1 by

$$\hat{\mathbf{x}}_{t+1} \coloneqq \mathbf{x}_t \oplus \mathbf{u}_t.$$

We define \oplus as the operator that transforms a given pose by a relative motion.

The EKF tracks the pose of the vehicle as a Gaussian distribution. Thus, the state estimate \mathbf{x}_t is comprised of a pose estimate $\boldsymbol{\mu}_t$ and a covariance matrix Σ_t , which can be interpreted as a measure of uncertainty for the current pose estimate. Again, the pose estimate $\boldsymbol{\mu}_t = (\mathbf{x}_t \ \mathbf{y}_t \ \theta_t)^T$ contains the position $(\mathbf{x}_t \ \mathbf{y}_t)^T$ and the heading θ_t of the robot in the global coordinate frame.

The prediction of the state for the next time step $\hat{\mathbf{x}}_{t+1} = (\hat{\boldsymbol{\mu}}_{t+1}, \hat{\boldsymbol{\Sigma}}_{t+1})$ for a relative motion measurement $\boldsymbol{u}_t = (\Delta x_t \ \Delta y_t \ \Delta \theta_t)^T$ is defined as follows:

$$\hat{\mu}_{t+1} := \qquad \mu_t \oplus u_t$$

$$\hat{\Sigma}_{t+1} := \qquad F \cdot \Sigma_t \cdot F^T + Q, \text{ where}$$

$$F := \begin{pmatrix} 1 & 0 & f_x \\ 0 & 1 & f_y \\ 0 & 0 & 1 \end{pmatrix}, \text{ with } \begin{pmatrix} f_x \\ f_y \end{pmatrix} = \begin{pmatrix} -\sin \theta_t & -\cos \theta_t \\ \cos \theta_t & -\sin \theta_t \end{pmatrix} \cdot \begin{pmatrix} \Delta x_t \\ \Delta y_t \end{pmatrix}$$

$$Q := \begin{pmatrix} q_x & 0 & 0 \\ 0 & q_y & 0 \\ 0 & 0 & q_\theta \end{pmatrix}, \text{ with } \begin{pmatrix} q_x \\ q_y \\ q_\theta \end{pmatrix} = S \cdot \begin{pmatrix} | \Delta x_t | \\ | \Delta y_t | \\ | \Delta \theta_t | \end{pmatrix} + s.$$

The matrix S and the vector s are the parameters of the motion model. The matrix S introduces uncertainty that scales according to the size of the measured motion. The farther the robot traveled between time steps, the more uncertain is the prediction. The vector saccounts for uncertainty that is independent of motion, for example, measurement uncertainty from the odometry and IMU sensors.

Both algorithms then use sensor observations to correct the predicted pose estimate. These corrections are computed with respect to a semantic map of the field. A semantic map represents the field as a set of lines that correspond to the crop rows. To enable the fusion of GNSS signals into the localization, the semantic map is GNSS-referenced, that is, the GNSS coordinates of each crop row are known.

The correction step of the EKF requires the definition of a measurement prediction function $h(\mathbf{x}_t)$ and measurement noise covariance matrix R to update the state with a sensor measurement \mathbf{z}_t . With this information, one derives the residual $\mathbf{y} \coloneqq \mathbf{z}_t - h(\mathbf{x}_t)$ and the Jacobian matrix H of the measurement prediction $h(\mathbf{x}_t)$. The mathematical equations to update the EKF state are then given by:

$$\begin{split} \boldsymbol{\mu}_{t+1} &\coloneqq \quad \hat{\boldsymbol{\mu}}_t + \boldsymbol{K} \cdot \boldsymbol{y} \\ \boldsymbol{\Sigma}_{t+1} &\coloneqq (\boldsymbol{I} - \boldsymbol{K} \cdot \boldsymbol{H}) \cdot \hat{\boldsymbol{\Sigma}}_t \\ \boldsymbol{K} &\coloneqq \quad \hat{\boldsymbol{\Sigma}}_t \cdot \boldsymbol{H}^T \cdot \boldsymbol{U}^{-1} \\ \boldsymbol{U} &\coloneqq \boldsymbol{H} \cdot \hat{\boldsymbol{\Sigma}}_t \cdot \boldsymbol{H}^T + \boldsymbol{R}. \end{split}$$

In the following, we describe in detail how we compute the correction measurements and how we define the individual measurement prediction functions to incorporate the measurements into the GD and EKF.

4.1 | Lateral and orientation correction using the detected crop row pattern

Given our semantic map of crop rows as a set of lines M, and the detected crop row pattern in the robot frame, we first project

the detected pattern into the map frame using the predicted pose of the vehicle to get $p = (\theta_p, o_p, s_p)$.

To compute a heading and lateral offset correction from the set of map lines and the set of lines defined by the pattern, we first need to compute which line in the map corresponds to which line in the pattern, that is, we need to compute a data association between the map and the projected pattern *p*. Since we need to compare lines with lines, we first define a function to measure the distance between two lines that are not necessarily parallel.

4.1.1 | Line to line distance measure

We define the signed distance $d_q(l_1, l_2)$ between two lines l_1 and l_2 using a reference point q as follows:

$$d_q(l_1, l_2) := d(l_1, q) - d(l_2, q)$$

where d(l, q) is the signed Euclidean distance of a point q to the line l. This distance measure is agnostic to the orientation of the lines.

In the following, we always use the position of the robot as reference point q (and omit the parameter q). This means that we compute all offsets relative to the robot position, that is, two lines have a distance of zero, if and only if they have the same signed distance from the robot (see Figure 3).

4.1.2 | Data association between two sets of lines

Now that we know how to compute the distance between two lines, we use this measure to compute correspondences *C* between the set of map lines $\mathcal{M} = \{m_i\}_{i \in I}$ and the set of lines defined by the pattern $p = \{l_j\}_{j \in J}$. More formally, a correspondence $c \in C$ is a mapping from the set of pattern indices into the set of map



FIGURE 3 Line to line distance. The distance between two lines l_1 (red) and l_2 (green) is marked in blue. Geometrically, the computed distance corresponds to rotating one of the lines around the reference point (by less than 90°) until both lines are parallel and then computing their distance [Color figure can be viewed at wileyonlinelibrary.com]

indices, such that for each pattern line l_j , there is a corresponding map line $m_{c(j)}$. The best correspondence, that is, the best data association, is now defined as the correspondence c^* that minimizes the summed line-to-line distance over all pattern and map line pairs:

$$c^* = \operatorname{argmin}_{c \in C} \sum_{j \in J} d(I_j, m_{c(j)}).$$

However, we do not aim for just any arbitrary correspondence $c \in C$ as data association, but we require a mapping that is consistent with the geometric relation between lines of the same set, that is, *geometrically consistent* data associations. For two sets of lines we define geometrically consistent as follows. Given a match of lines (l_j, m_i) , then the neighboring lines have to be matched with the respective neighbors, that is (l_{j-1}, m_{i-1}) and (l_{j+1}, m_{i+1}) . We generate the set of consistent correspondences as follows.

We change the offset parameter of the pattern to shift the whole pattern without changing its orientation. For each line in the shifted pattern, the corresponding map line is its nearest neighbor using the line-to-line distance and a threshold of 0.1 m. This means that, if for any given pattern line the closest map line has a distance larger than 0.1 m, this pattern line does not have a reasonable match in the map and thus, the investigated correspondence is discarded as invalid.

Since randomly shifting the pattern offset does not necessarily yield a different correspondence mapping, we do not investigate all possible pattern offsets. Instead, we use the fact that the robot should track its pose relative to the line closest to it. Therefore, we compute the pattern line closest to the robot and use it as reference line l_r . For this reference line, we compute its *n* closest map lines $\{m_1, ..., m_n\} \subseteq \mathcal{M}$, where *n* can be chosen between 1 and $|\mathcal{M}|$. The signed distance $d_1, ..., d_n$ between each of these *n* map lines and the reference line then defines a set of candidate configurations $\{c_k\}_{k=1, ..., n}$ with a pattern offset shift equal to d_k for the *k*-th correspondences is then the subset of all valid candidate correspondences $\{c_k\}_{k=1, ..., n}$. A simplified example with two sets of parallel lines is given in Figure 4.

We compute the best correspondence c_{i^*} out of the geometrically consistent correspondences. If more than one configuration minimizes the objective function, we choose the one with the smallest pattern shift. The intuition behind this is that—if in doubt about the correct data association—we favor the data association closer to the predicted pose estimate. If no valid configuration can be found (that is all configurations have at least one unmatched pattern line), it is likely that the pattern detection was unsuccessful (for example the spacing is too narrow) and we do not correct the pose estimate with the detected pattern.

4.1.3 | Correction step in GD and EKF

The lateral correction Δ lat for the best configuration c_{i^*} is given by its corresponding pattern offset shift d_{i^*} . The angular correction $\Delta \theta$ is



FIGURE 4 Example data association between map lines (green) and pattern lines (red). Left: The pattern line closest to the robot (solid red line) is the reference line l_r . For n = 4, we compute its n nearest neighbors in the set of map lines { m_1, m_2, m_3, m_4 }. Right: Shifting the pattern offset by the respective signed distances $d_1, ..., d_4$, yields the candidate correspondences $c_1, ..., c_4$. Note that shifting the pattern offset by d_i translates the reference line l_r on the respective map line m_i . For c_1 and c_2 one of the pattern lines has no matching map line, that is, d > 0.1m. Therefore, these correspondences are invalid and do not belong to the set of geometrically consistent correspondences. For c_3 and c_4 all pattern lines perfectly match the map lines, yielding a summed distance score of zero, that is, both data associations are equally reasonable. Since c_3 corresponds to a smaller offset shift than c_4, c_3 is chosen as the best data association [Color figure can be viewed at wileyonlinelibrary.com]

the signed distance between the orientation of the reference line θ_{l_r} and the orientation of its matched map line θ_{m_l} :

$$\Delta \text{lat} := d_{i^*} = d(l_r, m_{i^*})$$
$$\Delta \theta := \theta_{l_r} - \theta_{m_{i^*}}$$

For the GD filter, we apply the correction measurement directly in the robot frame:

$$\mathbf{x}_{t+1} \coloneqq \hat{\mathbf{x}}_t \oplus \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \alpha_{\theta} \cdot \Delta \theta \end{pmatrix} \oplus \begin{pmatrix} \mathbf{0} \\ \alpha_{\mathsf{lat}} \cdot \Delta \mathsf{lat} \\ \mathbf{0} \end{pmatrix},$$

where the scaling parameters α_{lat} and α_{θ} determine how strongly we correct the predicted pose in the direction of the correction measurement.

For the correction step in the EKF, the measurement prediction h, its Jacobian H, the residual y and the measurement noise covariance matrix R are defined as follows:

$$\begin{split} h_{m_{i^*}}(\mathbf{x}_t) &:= \begin{pmatrix} n_{m_{i^*}} \cdot (\mathbf{x}_t - \mathbf{p}_{m_{i^*}}) \\ \theta_{m_{i^*}} - \theta_t \end{pmatrix} \\ \mathbf{y} &:= \begin{pmatrix} \Delta \text{lat} \\ \Delta \theta \end{pmatrix} \\ H &:= \begin{pmatrix} n_x & n_y & 0 \\ 0 & 0 & -1 \end{pmatrix} \\ R &:= \begin{pmatrix} \sigma_{\text{lat}}^2 & 0 \\ 0 & \sigma_{\theta}^2 \end{pmatrix}. \end{split}$$

Note that these definitions depend on the map line m_{i^*} that is matched to the reference pattern line according to the best configuration c_{i^*} . Here, $n_{m_{i^*}} = (n_x \ n_y)^T$ is the normal vector of m_{i^*} and p_{i^*} is an arbitrary point on the line. Thus, $n_{m_{i^*}}$ defines the direction in which the lateral correction measurement needs to be applied to the current pose estimate. The parameters σ_{lat} and σ_{θ} model the uncertainty of the measurement.

4.2 | Longitudinal correction

For longitudinal pose correction, we use GNSS signals as well as the detected end of the field. On the headlands, the extracted crop row pattern is usually not well supported by the feature map. We make use of this to determine the end of the field.

4.2.1 | End of field detection

To detect the end of the field, we compute the support of the pattern along all pattern lines. We first determine all cells Cin the feature map that are traversed by one of the pattern lines. We call each section of a pattern line going through a cell $c \in C$ a pattern section. We determine the support of this pattern section as follows (see also Figure 5 on the left for an illustration). For each cell $c \in C$, we compute the line l_c that is orthogonal to the pattern and goes through the center of c. We apply a one-dimensional Gaussian kernel along l_c , centered on c, to compute how well c is supported by the feature map. We denote the resulting value as s(c). To account for the fact that some plant centers might not be perfectly aligned with the pattern, we also compute the support of cells on l_c that are within a certain distance from c. We then choose the cell with the maximum support along l_c as the reference cell c_r for the corresponding pattern section and thus the support of c_r defines the support of this section.

We analyze the reference cells along each pattern line and choose the two outermost cells c_{\min}, c_{\max} in each direction with sufficient support and sufficient supported adjacent cells. We determine all lines \mathcal{L} through these cells that are orthogonal to the pattern. We then get both a minimum and maximum extension line by finding the two lines $I_{\min}, I_{\max} \in \mathcal{L}$ that include all other lines in \mathcal{L} between them, as shown in Figure 5 on the right. The maximum extension line defines the end of the field.

10 WILEY



FIGURE 5 Support of a pattern and end of field detection. Left: An example of computing the pattern support on a feature map (white and cyan cells) for one pattern line (black) is shown. For each cell *c* that is traversed by the line, we define the support as a one-dimensional Gaussian kernel in direction of l_c (dotted black line, orthogonal to the pattern line). Here, the window size of the kernel is five cells (blue border). We also determine the support for cells on l_c that are within a certain distance of *c* (here two cells). The cell with the maximum support is chosen as reference cell c_r . Its distance from the pattern line is denoted as $d(c_r)$. The resulting reference cells and their distance to the pattern line are marked in red. Right: The detected end of field and parameters for the pattern quality are shown. The pattern extensions (red lines) of the rows (black lines) are overlaid on a feature map (heat map). The detected end of field is defined as the pattern extension I_{max} . The supported pattern sections with sufficient adjacent supported cells are indicated in pink [Color figure can be viewed at wileyonlinelibrary.com]

4.2.2 | Correction step in GD and EKF

For the longitudinal correction, we need to define a reference line just as we did for the lateral and angular correction. Since we want the robot to track the closest crop row, we choose the map line m_r that is closest to the current pose estimate.

For the longitudinal correction measurement from GNSS data, we define a line m'_r that is parallel to the reference map line m_r and passes through the robot position. The GNSS measurement is then projected onto this line m'_r . The longitudinal correction measurement $\Delta \log_{GNSS}$ is computed as the signed distance between the projected GNSS and robot position (see Figure 6 on the left).

To further improve the longitudinal pose estimate of the robot, especially when it approaches the end of the field, we include the detected end of the field. The distance to the detected end of field d_{EOF} is the signed Euclidean distance of the robot position to the maximum extension of the pattern (see Figure 6, mid). Similarly, the expected distance to the end of field measurement e_{EOF} is the signed Euclidean distance of the projected current robot pose to the row endpoint of the reference line m_r (see Figure 6, right). The longitudinal correction with respect to the end of the field is then $\Delta \log_{\text{EOF}} \coloneqq d_{\text{EOF}} - e_{\text{EOF}}$.

For any longitudinal correction measurement Δ_{long} , we perform the correction step as follows. In the GD algorithm, we directly integrate the correction on the current estimate \hat{x}_t :



FIGURE 6 Longitudinal correction measurements. Left: We define a line m'_r (dotted green) that is parallel to the reference map line m_r and passes through the robot position (black). The received GNSS signal (red) is orthogonally projected onto m'_r . The signed distance between the robot position and the projection point is the longitudinal GNSS correction $\Delta \log_{GNSS}$ (blue). Middle: The detected end of field d_{EOF} (blue) is the signed Euclidean distance of the robot position (black) to the maximum extension line I_{max} (red). Right: The expected end of field e_{EOF} (blue) is the signed distance of the robot position (black) projected onto the reference map line m_r (green) to the end point of that line [Color figure can be viewed at wileyonlinelibrary.com]

11

$$\mathbf{x}_{t+1} \coloneqq \hat{\mathbf{x}}_t \oplus \alpha_{\text{long}} \cdot \Delta \text{long} \cdot \begin{pmatrix} b_x \\ b_y \\ 0 \end{pmatrix},$$

where $\mathbf{b} = (b_x \ b_y)^T$ is the normalized direction of the reference map line m_r and α_{long} models the uncertainty in the longitudinal position measurement.

For the EKF, we get the following update matrices:

$$\begin{aligned} h(\mathbf{x}_t) &\coloneqq \begin{pmatrix} x_t \\ y_t \end{pmatrix} \\ \mathbf{y} &\coloneqq \Delta \text{long} \cdot \mathbf{b} \\ H &\coloneqq \mathbf{I} \\ R &\coloneqq \begin{pmatrix} \sigma_x^2 & \mathbf{0} \\ \mathbf{0} & \sigma_y^2 \end{pmatrix}, \end{aligned}$$

where σ_x and σ_y account for the uncertainty of the position estimate. Intuitively, the predicted measurement is equal to the current robot position and the longitudinal correction measurement is applied in the direction of the reference map line m_r (see Figure 6, left).

4.3 | False-positive crop row detection

The robustness of a localization algorithm highly depends on the reliability of the input data. Therefore, we derived a measure that we call pattern quality that estimates how reliable the detected pattern is, that is, how well the detected pattern is supported by the feature map. With a reasonable threshold on this pattern quality, we are able to filter out false-positive crop row detections. This is necessary since all presented crop row detection algorithms always return the best pattern to any given input. This is independent of the structure of the given input, that is, even on a feature map containing only noise and no actual row structure, the crop row detection algorithms will still find a pattern that matches the data best.

Our pattern quality consists of several parameters that describe how well the feature map supports the detected pattern. In the following, we present each pattern quality parameter and in the end describe how we merge the parameters to obtain the pattern quality measure.

We already determined the support of the pattern in Section 4.2 and use the computations from there. First of all, we use the pattern extensions that we previously determined for the end of field detection: If only short parts of the pattern lines are supported, we cannot expect a reliable pattern orientation. Thus, if the minimum and maximum extension lines are close together, which means that the feature map supports only a short part of the extracted pattern, the pattern quality is low. Therefore, the distance between the extension lines defines the first parameter for the pattern quality.

Given the reference cells for all pattern sections, we compute their distances $d(c_r)$ to the corresponding pattern line (see Figure 5, left). We use the mean and standard deviation of the distances $d(c_r)$ as parameters for the pattern quality. Intuitively, a large distance means that most of the support is far from the pattern and thus this pattern section is not well supported. Similarly, a large standard deviation infers that the support is not consistently shifted by the same amount but can even be shifted to either side of the pattern line, that is, there is no clear linear structure close to the extracted pattern.

Another parameter for the pattern quality is how the support of the pattern is distributed over its individual lines. If all the support lies on a single line of the pattern, it is unlikely that it is actually a crop row, as we would expect to see more than one crop row at a time. The more equally distributed the support is among the pattern lines, the better the features support the pattern. To this end, we check for each line in the pattern if its reference cells contain sufficient support. We define the number of supported rows as the number of rows between the left- and right-most supported pattern lines.

For each of the quality parameters we compute the likelihood that the detected pattern corresponds to actual crop rows. We first clamp and then linearly normalize each quality parameter according to the expected range of each parameter. We also invert the likelihood for the mean and *SD* of the distances $d(c_r)$ as lower values represent a higher quality value. The overall likelihood that the pattern corresponds to rows, that is, the pattern quality, is then the product over all likelihoods. If the pattern quality is low, the detected pattern is likely a false-positive detection, that is, no row structure is visible in the sensor data, or the detection is wrong or inaccurate, for example, because there is too much noise from weeds or headlands vegetation. With a lower bound on the pattern quality, we are able to filter out these false detections, which are then not used to correct the pose estimate of the localization algorithm.

While filtering out false-positive pattern detections helps to avoid localization failure, there are other problems in row-based localization. One reason why the localization might be mislead is a pattern detected from crop rows of an (unmapped) neighboring field. This can happen when the robot is leaving the currently traversed field. To avoid such errors, we automatically disable pattern integration when the robot leaves the field and the field of view of its sensors does not contain the traversed field anymore. As soon as the robot is facing the current field, we start integrating crop row patterns again to relocalize the robot with respect to the crop rows.

Overall, we present a localization method that uses different sensor modalities and combines their strengths to provide accurate pose estimates in the field as well as on the headlands, while leaving or entering a field. In the experiments, we will show how the different modalities complement one another to achieve accurate localization results.

5 | EXPERIMENTAL EVALUATION

To evaluate our localization approach, we performed experiments on a production vegetable field in Eichstetten near Freiburg, Germany. It was kindly provided by a local farmer. The field contained several vegetable types in close proximity, even changing mid-row so that crop-specific algorithms are not applicable. We recorded data from six rows available for our experiments, containing Kohlrabi, Chinese Cabbage, and Pointed Cabbage, using our agricultural robot BoniRob (see Figure 12). It was driven at different speeds with a maximum of 0.4 m/s. We performed one run in the morning and a second run in the afternoon for comparison. In each run, we recorded sensor data, that is, odometry, IMU, GNSS, and camera images. The images were recorded with a PointGrey Blackfly color camera with five megapixels (BFLY-PGE-50S5C-C with a Sony IMX264 chip, 2/3", 3.45 µm) at a frame rate of 5 Hz. It is mounted centered in the front of the robot at a height of about one meter above the ground and tilted downwards by about 25°. The GNSS receiver is a u-blox EVK-7, and the IMU is an SBG ELLIPSE2-A-G4A3-B1. Note that our approach is independent of the exact sensors and mounting points, as long as the intrinsic and extrinsic calibration is known. The localization runs on a Pokini i2 computer with an Intel Core i7-4600U CPU and 16 GB RAM, which is integrated into the BoniRob. The computation time for the localization pipeline is dominated by the crop row detection that takes about 0.1 s per image. This is sufficient for online processing of the localization since each image of the camera is integrated with at least 5 Hz. In the next section, we present an evaluation to show the performance of various crop row detection methods as an input to our localization. It is followed by the section on results for the localization evaluation.

5.1 | Crop row detection

As the crop row detection is an important input to our localization approach, we evaluate several crop row detection methods. The evaluation on this data set extends our previous evaluation of the same crop row detection methods (Winterhalter et al., 2018). Of particular interest are the lateral and angular error, as they will be used in the correction for the localization.

We evaluate the performance of the crop row detection methods regarding robustness on the vegetable field data set featuring different crops. Robustness in our case means that a crop row detection algorithm is able to reliably produce crop row patterns that are close to the actual real-world situation. This is crucial for robust localization with respect to the crop rows.

We ran all four crop row detection algorithms presented in Section 3.2 on feature maps created from vision data. In all cases the same parameter sets were used independent of the crop and algorithm. The histograms for the Hough-based algorithms had an angular resolution of 0.57° (0.01 rad) and an offset and spacing resolution of 1 cm. As the Pattern RANSAC algorithm relies on an incremental improvement, we evaluated it with 2500, 5000, and 25,000 iterations and repeated each run five times to account for its probabilistic nature. We chose the number of iterations so that one variant is faster than the Pattern Hough transform, one takes a similar amount of time and one gets close to the optimal result when time is not an issue.

In the following, we first introduce our data set, then we present the method that we used for evaluation, before investigating the robustness of crop row detection in general. Afterward, we discuss data sets that are especially challenging in detail.

5.1.1 | Data set description

To ensure robustness on all crop types present, we split the recorded data of the vegetable field into parts that each contains only one crop type. As a result, we have three parts. In Figure 7 you can see representative image data of the crop types, namely *Kohlrabi* with small plants of about 10 cm height, *Chinese Cabbage* with large overlapping plants, and *Pointed Cabbage* with medium-sized plants of about 15 cm height. For Chinese Cabbage, the extracted feature maps are much denser than for the other crop types, as can be seen on the bottom of Figure 7 and in Table 1.

All parts of the data set were again split into in row motions, where the robot drives in the field aligned with crop rows, and transition motions, where the robot either leaves or enters the field. The covered distance for driving in row and the covered angle for transitions are listed in Table 1. When leaving or entering the field, crop rows are usually only partially visible and the robot is not necessarily aligned with the field, for example, when turning towards it (see Figure 1, right two images). Such scenarios present hard challenges to crop row detection algorithms as less data is available to detect crop row patterns, while at the same time the sensors see areas not part of the field that also contain vegetation. These situations are particularly important for autonomous navigation as they allow the robot to accurately leave and enter a field.

5.1.2 | Evaluation method

Robustness is crucial for guiding an autonomous vehicle. We define robustness by the percentage of successfully extracted patterns. A pattern was extracted successfully if the angular and lateral error are within reasonable thresholds for navigation. Here, the error of the spacing is not essential, as navigation relies on the angular and lateral errors. To determine the error of the extracted crop rows, we manually labeled crop row patterns in feature maps extracted from images. We quantify the error between the computed pattern and the labeled ground truth pattern by two measures: the angular error between the normal angles of the patterns, and the lateral error. The lateral error is defined as the minimum of the pairwise distance between all pattern and ground truth lines with respect to a given reference point. We project this reference point onto both lines of each pair. The distance between those projected points then defines the distance between the respective lines. The reference point is the position of the robot projected one meter in front of the robot.

We say that a pattern extraction was successful if the angular error is smaller than 10° and the lateral offset does not exceed 0.10 m. Both measures are crucial to control the angle and sideways tracking of a robot following a crop row (Åstrand & Baerveldt, 2005),



FIGURE 7 Image data and feature maps. Shown are examples for each plant type present in the data set [Color figure can be viewed at wileyonlinelibrary.com]

and thus also in a localization. The thresholds were determined in real-world experiments on our BoniRob.

As feature maps extracted during transitions potentially change in appearance quickly, we manually labeled crop rows and evaluated the crop row detection methods every 0.1 s in transition data sets. When driving in a row, we only performed the evaluation on feature maps every 5 s since the surroundings stay similar for a long time due to low speed. The number of feature maps for each data set is shown in Table 1.

TABLE 1 Data set properties for Run 1/Run 2

In row	Distance covered (m)	Number of feature maps	Mean vegetation density (%)	
Kohlrabi	153 m	86/83	1.54/1.57	
Chinese cabbage	246 m	126/130	9.14/9.20	
Pointed cabbage	110 m	55/60	2.78/1.97	
Transition	Angle covered (°)	Number of feature maps	Mean vegetation density (%)	
Kohlrabi	63/65	91/101	1.34/1.61	
Chinese cabbage	68/85	151/249	5.87/4.74	
D · · · · · ·	4.0.00	05 (4 4 0	1 0 1 /0 5 1	

Note: When driving in row, the angle does not change a lot, and during transition, the robot does not cover much distance. The vegetation density denotes the percentage of cells in a feature map that contain plant features

5.1.3 | Results

In Figure 8, the success rates for all algorithms on all data sets are shown. As a first observation, note that for all algorithms the performance does not notably differ among the two runs, except for the Line Hough and Dual Line Hough in the transition part of Kohlrabi. As the first run was done in the morning and the second in the afternoon, this suggests that the crop row detections are independent of the lighting conditions. Over all in row data sets, the pattern-based crop row extraction methods have a high success rate of at least 94%, whereas the success rate for the line-based methods is considerably lower on Chinese Cabbage with only 72% for the Line Hough. This confirms that employing all available data to jointly estimate a pattern is important for robustness. For driving in a row, any of the presented pattern-based crop row detection approach.

For the Chinese Cabbage in row data set, the success rate is in general lower than for the other two in row data sets. This is because the crops are larger and thus cover a larger part of the image, as can be seen in the vegetation density in Table 1. The resulting feature maps are more dense and noisy, allowing to shift the correct pattern in the lateral direction without decreasing its support (see Figure 7, mid). Due to this ambiguity, we observe overall higher lateral errors on the Chinese Cabbage in row data set compared to the other two plant types (see Figure 9). A method with fixed spacing that only fits a single line (Line Hough) is even more error-prone than methods with variable spacing that consider all available data.

FIGURE 8 Success rates of all crop row detection algorithms. A pattern detection is labeled as successful, if the angular error is below 10° and the lateral error is smaller than 0.1 m [Color figure can be viewed at wileyonlinelibrary.com]

14



All data sets pose greater challenges when the robot is transitioning as less data is visible overall and the images do not only capture the field, but also possible vegetation off the field. On Kohlrabi, the success rates drop to 40% for line-based methods and 66% for pattern-based methods. For Chinese Cabbage and Pointed Cabbage, the success rates are reasonably high with at least 80% for line-based methods and 83% for pattern-based methods. An 80% success rate is still sufficient for localization, but this demonstrates that our localization needs to deal with failure cases in the crop row detection.

To investigate the low success rates in the Kohlrabi transition data set, we take a closer look at the individual angular and lateral errors in Figure 10. It is obvious that the angular error is much too high for roughly 20% of the extracted patterns in the first run, which also leads to a high lateral error. The reason for this wrong pattern extraction is apparent in Figure 11 on the left. While turning toward a row of Kohlrabi, the camera first captures an image of the neighboring Chinese Cabbage. This results in a noisy and dense feature map, from which none of the evaluated algorithms is able to correctly estimate the crop row pattern. For the same reason, the lateral errors in the second run are too high. Furthermore, the spacing assumed for the Line Hough is not always correct, as the rows are not sown in exactly the same distances. Thus, the Line Hough uses a wrong spacing as shown in Figure 11 on the right and thereby estimates the offset incorrectly.

The results of our evaluation on this vegetable field data set are consistent with our previous findings on different crops (Winterhalter et al., 2018). However, during our experiments we discovered that a real agricultural field is sometimes quite different from a nicely groomed research field. One assumption for detecting the crop rows is that crop rows are locally equidistant. This has always been the case on the research fields on which we evaluated our algorithm. On the real-world field in Eichstetten, there was a larger distance between some of the crop rows to make space for the wheels of the tractor.



FIGURE 9 Evaluation of the lateral error when driving in row. Shown are individual errors sorted by size. The black horizontal line marks the threshold for a successful pattern extraction [Color figure can be viewed at wileyonlinelibrary.com]

WINTERHALTER ET AL.



FIGURE 10 Evaluation of angular and lateral error for Kohlrabi during transition. Shown are individual errors sorted by size. The black horizontal line marks the threshold for a successful pattern extraction [Color figure can be viewed at wileyonlinelibrary.com]

This fact is also reflected in our semantic map of the field (see Figure 12). We therefore had to restrict the detection window to the range of the inner three crop rows.

While crop row detection methods already provide valuable information to row-based localization directly, they can also be used to create a semantic map of crop rows as required by our localization approach. For example, a GNSS-referenced overhead image of the field can be acquired by a UAV within minutes, while a manual mapping run with the BoniRob could take hours. On this overhead image, any pattern-based crop row detection method can be used to determine the crop rows. As the image is GNSS-referenced, the GNSS coordinates of the crop rows can be extracted and used as a map for localization. We used this method to extract maps before. However, on the production field in Eichstetten it was not applicable since the rows were not equidistant. For the future, it might be helpful to extend the model used in crop row detection to also allow for periodically larger spacing. This would allow us to extract a semantic map from overhead images and to use a larger detection window in the localization. The latter might further increase the success rate of pattern detection during transition.

In the following, we use the patterns detected by the Pattern Hough transform to localize our robot on the vegetable field. The results will show that, even with a success rate of only 66% in the Kohlrabi transition data set, the localization is still accurate.

5.2 | Localization

In this section, we evaluate our approach for localization in agricultural fields. We designed our experiments to show that our approach satisfies the requirements to enable safe and robust autonomous navigation. First, we investigate the lateral and angular localization accuracy. Only if the lateral and angular pose estimate is sufficiently accurate, it can be used to safely guide a vehicle relative to crop rows. Second, we evaluate how filtering false-positive crop row detections affects the robustness of the localization. Third, we evaluate how different localization algorithms track the longitudinal pose estimate along the field. An accurate longitudinal pose estimate is crucial when leaving the field since this determines where the robot starts turning. If this estimate is too far off, it will either start



FIGURE 11 Example crop row detection results on the Kohlrabi transition data set. Left: While turning towards the row of Kohlrabi, the neighboring rows of Chinese Cabbage are visible resulting in a dense and noisy feature map. Patterns are extracted using the Line Hough (pink), Dual Line Hough (red), Pattern RANSAC 25000 (blue), and the Pattern Hough (black). None of the investigated algorithms find the correct pattern. Right: The spacing parameter of the Line Hough (pink) is incorrect. In contrast, the Pattern Hough (black) fits its pattern to all available data, thus estimating the offset and spacing correctly [Color figure can be viewed at wileyonlinelibrary.com]



FIGURE 12 Recording ground truth poses. Top: We used our agricultural robot BoniRob to record the data set. The ground truth markers were placed at the beginning, the mid and the end of each row. We mounted a downward pointing laser on the robot to mark its position and orientation on the markers in the field whenever it reached a marker position. Bottom: Pose (red arrow) and id of each marker for both data sets [Color figure can be viewed at wileyonlinelibrary.com]

turning too early, that is, in the field, or too late and thus leave the safe area of the headlands.

Our evaluation is performed on our vegetable field data set. We evaluate the localization accuracy relative to the actual crop rows. This allows us to relate the position of the robot to the crops in contrast to using an absolute position provided by external tracking systems such as high precision GNSS. Besides the orientation accuracy, for the positional accuracy we do not just determine the Euclidean distance but consider the lateral distance relative to the crop rows independent from the longitudinal displacement along a crop row. The main reason is that the required accuracy of the lateral offset is usually much higher than for the longitudinal position along a row. Depending on the vehicle, a lateral accuracy of few centimeters can be crucial to avoid driving over crops, while dozens of centimeters or even meters might be sufficient longitudinally. Considering the Euclidean distance only leads to meaningful results for navigation if a localization algorithm provides centimeter precision in both measures. In the following, we detail how we performed the

experiments and show how our methods are necessary to enable localization for the full traversal of agricultural fields.

5.2.1 | Evaluation method

To evaluate the localization algorithms on our vegetable field data set, we manually measured the orientation and position of the robot laterally and longitudinally relative to the rows. For measuring, we stopped the robot at critical points, that is, at the start and end of each row. We also measured the orientation and lateral position at points several meters into the field to investigate the tracking in the field. For this, we placed ground truth markers in the crop rows and determined their positions to the row. The pose of the robot was determined by down-projecting a coordinate system from a fixed laser. In this way, we determine the position and orientation of the robot accurately (see Figure 12). We estimate the attainable accuracy at 3° in orientation and 5 cm in position laterally and

-WILEY

longitudinally. Since these measurements are determined relative to naturally grown plants a more precise position cannot clearly be defined. A localization error within this magnitude is well suited for navigation so that higher accuracies are not required. We further define a pose estimation as successful, if the orientation error does not exceed 10° and the lateral error does not exceed 0.1 m. This corresponds to the success thresholds we already employed for evaluating the crop row detection methods. Again, these thresholds are based on real-world constraints of our BoniRob.

We evaluate our EKF-based localization (denoted as EKF) and the gradient-descent-based localization (denoted as GD). To investigate the influence of the longitudinal corrections on the localization accuracy, we consider different variants of the GD and EKF algorithm: In the first variant, we only correct with patterns from the crop row detection without any longitudinal correction (denoted as GD or EKF, respectively). In the second variant, we include longitudinal corrections from GNSS signals (denoted as EKF-GNSS or GD-GNSS). In the third, we include longitudinal corrections with GNSS and our end of field detection (denoted as EKF-GNSS EOF or GD-GNSS EOF). The first variant serves as a base line reflecting the performance of state-of-the-art row following approaches. The second and third variants show how longitudinal corrections improve localization accuracy. The third variant highlights the importance of including our end-of-field detections for longitudinal pose corrections. As an additional reference, we also compare the GD and EKF variants with GNSS localization, where we fuse odometry, IMU and GNSS, but do not integrate crop row patterns. To show the effect of our pattern quality filtering on the robustness of the localization algorithm, we also investigate a variant of the EKF-GNSS EOF localization, where all detected patterns (regardless of their quality score) are integrated (denoted as EKF-GNSS EOF ng).

We ran all localization algorithms on both runs of the data set and evaluate the orientation, lateral and longitudinal errors. We plot these errors at each marker position for each algorithm. As we collect measurements at markers at the start and end of each row as well as a marker within the row, the marker indices $\{1, 4, 7, ...\}$ are positions where the robot just entered a row; the indices $\{2, 5, 8, ...\}$ are within a row; and the indices $\{3, 6, 9, ...\}$ are positions at the end of a row (see Figure 12, bottom).

5.2.2 | Results

We first look at the input data available to the localization algorithms apart from the crop row patterns which were discussed thoroughly in the previous section. In Figure 13 we show the odometry and GNSS trajectories for both runs. The fused odometry and IMU trajectory shows the typical drift in orientation, resulting in a starshaped trajectory. In contrast, the raw GNSS measurements show better alignment to the field. Note that there are considerable jumps in the GNSS trajectory at the marker positions. These are caused by the GNSS drift while the robot was standing for measuring the position. WILEY-

Even the fused GNSS localization although close to the rows is not correctly aligned and following this signal is not sufficient for accurate navigation as can be seen in Figure 14. In contrast, both our EKF localization with end of field correction and the GD without any longitudinal correction are able to accurately follow the robot pose along the rows. This shows that integrating crop row patterns in the localization is essential. Additionally integrating end of field or GNSS information improves the longitudinal position estimate.

For a quantitative evaluation, we first examine the orientation error in the top row of Figure 15. The base line GNSS localization cannot recover the orientation precisely as here no sensor information of the field is integrated. The error in orientation is especially high at marker positions 4, 7, 10, 13, and 16. This is caused by the IMU and odometry drift of the input data during turning and re-entering at the end of the field. In contrast, the pattern-based localization algorithms accurately determine the orientation of the robot relative to the crop rows as almost all measurements showed small errors within the measurement accuracy. A measurable but still acceptable orientation error is observed at marker position 16 of Run 2. The resulting pose is illustrated in Figure 16 on the bottom right. At this position the localization algorithms did not yet integrate any patterns due to too low pattern quality. However, just after measuring at the marker position, the guality increases, patterns are integrated and the robot position is tracked correctly along the rows.

Similarly, the lateral errors for the GNSS localization are considerably higher than for localizations that integrate crop rows (see Figure 15, bottom). While the GNSS localization was still mostly successfully localized regarding orientation, the lateral errors frequently exceed the success threshold with maximum lateral error up to 1.34 m. In contrast, all pattern-based algorithms succeed in tracking the robot pose with almost all lateral errors below the threshold of 0.1 m. The maximum lateral error of 0.11 m for patternbased algorithms is much lower than for the GNSS localization. The impact of the lateral error of 0.11 m in Run 1 at marker position 7 is shown in Figure 16. As can be seen on the bottom left, the wheels of the robot are close to the crop rows, but still not on the wrong side of the crops as is the case for the GNSS localization.

These results confirm that using pattern-based localization algorithms is preferable over GNSS localization. The pattern-based algorithms track the pose estimate with respect to the crop rows and therefore enable safe and accurate localization in the field.

An important improvement to reach these results is filtering out false-positive crop row detections as described in Section 4.3. Our pattern quality determines whether the pattern detection was successful and the pattern should be used in the correction step of the localization algorithm. The evaluation of the crop row detection algorithms (Section 5.1) shows that this is not critical when driving in row since most of the patterns were detected correctly. During transition between rows the success rate decreases. Here, a robust measure for false-positive crop row detection is required.

We provide an evaluation of our false-positive detection on crop row patterns in Table 2. In row, more than 95% of all patterns were correctly classified as valid in both runs. At most 1.47% (in Run 2)





x in m

invalid patterns we incorrectly forwarded to the localization, which is not a problem for our localization.

Run 2 y in m 10 15 20

ß

5 10 15 20 25 30 35 40 45 50 55 60 65

The latter fact, that is, how many incorrect patterns are integrated by the localization, is more important than how many correct patterns are discarded, as these incorrect patterns can lead to instabilities or even divergence. Thus, as long as a sufficient amount of correct patterns are integrated to achieve a stable and precise enough pose estimate, it is preferable to exclude as many

80 85

70 75

95 100

90



FIGURE 14 Resulting localization trajectories: GNSS localization (black), GD (orange) and EKF- GNSS EOF (red) [Color figure can be viewed at wileyonlinelibrary.com]



FIGURE 15 Orientation and lateral errors. The estimated measurement accuracy is marked by the dashed black line and the success threshold by the dotted black line. Markers at the start and end of a row are highlighted by light gray background [Color figure can be viewed at wileyonlinelibrary.com]



FIGURE 16 Visualization of localization poses. The high lateral error of the GNSS localization (black) would cause the robot to drive over the crop row. In contrast, although in Run 1 at Marker 7 the EKF-GNSS EOF localization exceeds the success threshold of 0.1 m, it still tracks the correct row. In Run 2 at Marker 16, the EKF-GNSS EOF localization did not yet integrate any patterns to correct its orientation estimate due to too low pattern quality. Just after measuring the error, the pattern quality increases and the pose is tracked correctly [Color figure can be viewed at wileyonlinelibrary.com]

	Run 1				Run 2			
	In row (267 patterns) Classified as		Transition (337 patterns) Classified as		In row (273 patterns) Classified as		Transition (462 patterns) Classified as	
	Correct (%)	Wrong (%)	Correct (%)	Wrong (%)	Correct (%)	Wrong (%)	Correct (%)	Wrong (%)
Correct	98.13	1.12	48.66	39.17	95.97	2.56	58.01	24.46
Wrong	0.75	0.00	1.78	10.39	1.47	0.00	3.90	13.64
Precision	99.24		96.47		98.50		93.71	
Recall	98.87		55.41		97.40		70.34	

Note: The top of the table shows the confusion matrix and the bottom gives precision and recall figures.

false-positives as possible, even at the expense of discarding some correct patterns. The precision and recall numbers highlight this fact. We aim for a high precision even in the more challenging transition scenarios and consider a lower recall as an acceptable tradeoff.

During transition, at most 3.90% (in Run 2) invalid patterns are passed to the localization. This is reflected in a precision of more than 93% for the false-positive detection in both runs and settings. This results in a false-discovery rate of less than 7%, where incorrect patterns are integrated into the localization. The localization successfully handles these cases, as can be seen in Figure 15. Aiming for a high precision accordingly impacts the recall, which in transition for Run 1 goes down to 55%. While this means that our localization will disregard almost half of all correct patterns, it still receives sufficient information to correctly determine the robot pose.

To demonstrate the impact of our false-positive pattern detection, the results of running our EKF-GNSS EOF algorithm with and without filtering false-positive detections are shown in Figure 17. For both runs both localization variants initially show similar tracking accuracy, until the EKF-GNSS EOF nq variant integrates wrong patterns (Run 1 Marker 15 and Run 2 Marker 6 and Marker 9). Integrating incorrect



FIGURE 17 Orientation and lateral errors for the false-positive detection evaluation. The estimated measurement accuracy is marked by the dashed black line and the success threshold by the dotted black line. Markers at the start and end of a row are highlighted by light gray background [Color figure can be viewed at wileyonlinelibrary.com]

21

patterns finally results in irrecoverable divergence of the localization algorithm in both cases. These results confirm that incorporating the pattern quality to filter incorrect patterns during localization is an important improvement over state-of-the-art crop row following approaches. This enables us to go beyond crop row following as the robot is accurately localized while it is leaving and re-entering the field.

For safe navigation, especially when there is not much space to turn at the headlands, the robot needs an accurate longitudinal pose estimate during transition. Therefore, we developed a longitudinal correction as described in Section 4.2 and evaluate how it improves the longitudinal pose estimate of the presented localization algorithms. As can be seen in Figure 18, without any longitudinal correction the vanilla row-based localization algorithms overshoot the end of the field for more than 4.0 m. Since the longitudinal pose estimate is only based on odometry measures, and the odometry constantly overestimates the traveled distance, the longitudinal estimate is roughly correct when coming back to the starting side of the field (see Markers 6, 7, 12, 13, 18 and also Figure 14). When fusing GNSS measurements to correct the longitudinal pose estimate as done in the GNSS variants, the longitudinal error drops to under 3.0 m as expected from the GNSS measurement accuracy. Integrating the detected end of field further decreases the error to the range of 0.21-1.1 m. Whenever the longitudinal error at the end of the field did not improve, that is, in Run 1 at Marker 9 and in Run 2 at Marker 15, the detected end of field was not integrated because the pattern quality was too low. These results show that detecting the end of the field is a crucial improvement of the longitudinal pose estimate to enable safe turning at the headlands.

When comparing the performance of the GD-GNSS EOF algorithm with the EKF-GNSS EOF algorithm, both yield comparable results regarding orientation and lateral error. However, the EKFbased localization with a mean longitudinal error of 0.39 m slightly outperforms the GD localization with a mean longitudinal error of 0.45 m. However, we did not perform any parameter fine tuning for both algorithms. So there might be a better parameter configuration for the GD approach. On the other hand, this also means that the accuracy of both localization approaches is not sensitive to the choice of parameters.

One of the limitations of our approach is that it cannot distinguish between different crop rows. This means that the rows can only be matched onto the map using geometrical properties. If the field has equidistant crop rows, finding the correct data association becomes quite hard. If the wrong rows are matched, the localization might skip one row and therefore, during autonomous driving, the robot might start traversing a row adjacent to the target row. However, this only affects the efficiency of the execution of a highlevel task. Traversing a neighboring crop row does not harm the crops. We did not experience these skips in localization for our



FIGURE 18 Longitudinal errors. The estimated measurement accuracy is marked by the dashed black line. Markers at the start and end of a row are highlighted by light gray background [Color figure can be viewed at wileyonlinelibrary.com]

Eichstetten data set since the rows were not equally distributed and the periodically larger gaps yielded enough information to identify the correct crop rows.

The results confirm that pattern-based localization algorithms are preferable over a GNSS localization that disregards crop row information entirely. Pattern-based approaches track the crop rows accurately and guide the robot along the field. Furthermore, the experiments show that using a GNSS-referenced map to consistently fuse GNSS signals into our row-based localization improves the longitudinal pose estimate while maintaining the heading and lateral offset accuracy of pure row following approaches. We show that integrating the detected end of field into the longitudinal pose estimate decreases the error from 3.0 down to 1.1 m. This is especially valuable in production fields that might not offer much space at the headlands for turning maneuvers. The evaluation of our method for filtering out false-positive crop row detections indicates that ignoring the pattern quality and thus integrating all detected patterns leads to unrecoverable failure of the localization. Overall, the experiments demonstrate that our proposed localization approach is well suited for navigation in an agricultural field in row as well as during turning maneuvers.

6 | CONCLUSION

In this paper, we presented a novel localization approach for autonomous navigation in agricultural fields. Existing localization approaches for agricultural environments typically guide the vehicle along crop rows by estimating its heading and lateral offset relative to the crop rows. However, heading and lateral offset corrections from crop rows are not sufficient when the vehicle is required to switch between rows. The method presented in this paper overcomes this restriction by additionally estimating the longitudinal pose of the vehicle along the crop row using a GNSS-referenced map of rows. It fuses the heading and lateral pose information provided by a crop row detection algorithm with longitudinal measurements from GNSS signals and an end of field detection method. To this end, we introduced a data association method that computes consistent correspondences between the detected and mapped crop rows. We include a longitudinal pose correction using GNSS data and an end of field detection to improve the longitudinal pose estimate. We also presented a method to filter false-positive crop row detections to further increase the robustness of our approach.

In extensive experiments we demonstrated the advantages of employing our proposed method over pure row following or purely GNSS-based methods. We evaluated the heading, lateral and longitudinal errors of the investigated localization methods in a production vegetable field with three different crop types. Our experiments confirm that pure row following approaches produce sufficiently accurate lateral and angular pose estimates to guide a vehicle along crop rows. We showed that including GNSS signals and our end of field detection is crucial to determine when to initiate a turning maneuver. Furthermore, our results illustrate that filtering false-positive detections increases the robustness of row-based localization algorithms. In summary, the experiments confirm that our approach provides sufficient accuracy to enable autonomous navigation in entire agricultural fields.

ACKNOWLEDGMENT

We would like to thank Herbert Rinklin for allowing us to run our experiments on his production vegetable fields.

REFERENCES

- Åstrand, B., & Baerveldt, A.-J. (2005). A vision based row-following system for agricultural field machinery. *Mechatronics*, 15(2), 251–269.
- Baia, G., Ge, Y., Hussain, W., Baenziger, P. S., & Graef, G. (2016). A multi-sensor system for high throughput field phenotyping in soybean and wheat breeding. *Computers and Electronics in Agriculture (CEA)*, 128, 181–192.
- Bakker, T., van Asselt, K., Bontsema, J., Iler, J. M., & van Straten, G. (2011). Autonomous navigation using a robot platform in a sugar beet field. *Biosystems Engineering*, 109(4), 357–368.
- Bakker, T., Wouters, H., van Asselt, K., Bontsema, J., Tang, L., Müller, J., & van Straten, G. (2008). A vision based row detection system for sugar beet. *Computers and Electronics in Agriculture (CEA)*, 60(1), 87–95.
- Ball, D., Ross, P., English, A., Milani, P., Richards, D., Bate, A., Upcroft, B., Wyeth, G., & Corke, P. (2017). Farm Workers of the Future visionbased robotics for broad-acre agriculture. *Robotics & Automation Magazine*, 24(3), 97–107.
- Biber, P., Weiss, U., Dorna, M., & Albert, A. (2012). Navigation system of the autonomous agricultural robot BoniRob. International Conference on Intelligent Robots and Systems (IROS).
- Chebrolu, N., Lottes, P., Läbe, T., & Stachniss, C. (2019). Robot localization based on aerial images for precision agriculture tasks in crop fields. International Conference on Robotics and Automation (ICRA).
- Constantin, D., Rehak, M., Akhtman, Y., & Liebisch, F. (2015). Detection of crop properties by means of hyperspectral remote sensing from a micro UAV. Bornimer Agrartechnische Berichte, 88, 129–137.
- Dong, J., Burnham, J. G., Boots, B., Rains, G. C., & Dellaert, F. (2017). 4d crop monitoring: Spatio-temporal reconstruction for agriculture. International Conference on Robotics and Automation (ICRA).
- English, A., Ball, D., Ross, P., Upcroft, B., Wyeth, G., & Corke, P. (2013). Low cost localisation for agricultural robotics. *Australasian Conference on Robotics and Automation (ACRA)*.
- English, A., Ross, P., Ball, D., & Corke, P. (2014). Vision based guidance for robot navigation in agriculture. *International Conference on Robotics* and Automation (ICRA).
- English, A., Ross, P., Ball, D., Upcroft, B., & Corke, P. (2015). Learning crop models for vision-based guidance of agricultural robots. *International Conference on Intelligent Robots and Systems (IROS).*
- Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6), 381–395.
- Hague, T., Tillett, N. D., & Wheeler, H. (2006). Automated crop and weed monitoring in widely spaced cereals. *Precision Agriculture (PA)*, 7(1), 21–32.
- Hough, P. (1962). Method and means for recognizing complex patterns. U.S. Patent 3.069.654.
- Kise, M., Zhang, Q., & Más, F. R. (2005). A Stereovision-based crop row detection method for tractor-automated guidance. *Biosystems Engineering (BE)*, 90(4), 357–367.
- Leemans, V., & Destain, M.-F. (2006). Line cluster detection using a variant of the Hough transform for culture row localisation. *Image and Vision Computing*, 24(5), 541–550.

- Libby, J., & Kantor, G. (2011). Deployment of a point and line feature localization system for an outdoor agriculture vehicle. *International Conference on Robotics and Automation (ICRA).*
- Marchant, J. (1996). Tracking of row structure in three crops using image analysis. Computers and Electronics in Agriculture (CEA), 15(2), 161–179.
- Montalvo, M., Pajares, G., Guerrero, J., Romeo, J., Guijarro, M., Ribeiro, A., Ruz, J., & Cruz, J. (2012). Automatic detection of crop rows in maize fields with high weeds pressure. *Expert Systems with Applications*, 39(15), 11889–11897.
- Mueller-Sim, T., Jenkins, M., Abel, J., & Kantor, G. (2017). The robotanist: A ground-based agricultural robot for high-throughput crop phenotyping. International Conference on Robotics and Automation (ICRA).
- Riggio, G., Fantuzzi, C., & Secchi, C. (2018). A low-cost navigation strategy for yield estimation in vineyards. *International Conference on Robotics* and Automation (ICRA).
- Ruckelshausen, A., Biber, P., Dorna, M., Gremmes, H., Klose, R., Linz, A., Rahe, R., Resch, R., Thiel, M., Trautz, D., & Weiss, U. (2009). Bonirob: An autonomous field robot platform for individual plant phenotyping. *Precision Agriculture (PA)*, *9*, 841–847.
- Søgaard, H., & Olsen, H. (2003). Determination of crop rows by image analysis without segmentation. *Computers and Electronics in Agriculture (CEA)*, 38(2), 141–158.
- Underwood, J., Wendel, A., Schofield, B., McMurray, L., & Kimber, R. (2017). Efficient in-field plant phenomics for row-crops with an

autonomous ground vehicle. Journal of Field Robotics (JFR), 34(6), 1061-1083.

- Vidović, I., Cupec, R., & Hocenski, Ž. (2016). Crop row detection by global energy minimization. Pattern Recognition, 55, 68–86.
- Watanabe, K. (2018). ARIADNE with ambiguity resolution: Visual marker based rapid initialization of PPP-AR. International Conference on Intelligent Robots and Systems (IROS).
- Winterhalter, W., Fleckenstein, F. V., Dornhege, C., & Burgard, W. (2018). Crop row detection on tiny plants with the pattern hough transform. *Robotics and Automation Letters (RA-L)*, 3(4), 3394–3401.
- Xaud, M. F. S., Leite, A. C., & From, P. J. (2019). Thermal image based navigation system for skid-steering mobile robots in sugarcane crops. International Conference on Robotics and Automation (ICRA).

How to cite this article: Winterhalter W, Fleckenstein F, Dornhege C, Burgard W. Localization for precision navigation in agricultural fields—Beyond crop row following. *J Field Robotics*. 2020;1–23. https://doi.org/10.1002/rob.21995