Project Thesis
PRO-063

Mechanik und Meerestechnik

Autonomous crop row navigation during harvest for strawberry fields

Navigierung eines autonomen Roboter durch ein Erdbeerfeld während der Erntezeits

by
Caroline Kühl

Supervisors: Prof. Dr.-Ing. R. Seifried
D. Dücker, M.Sc.

Hamburg University of Technology (TUHH)
Institute of Mechanics and Ocean Engineering
Prof. Dr.-Ing. R. Seifried

Hamburg, October 2022

TUHH

Abstract

Mobile robots are used in the agricultural field in various applications such as harvesting, weeding and planting. The automation of traditionally manual tasks has become more appealing for farmers as they are increasingly faced with challenging economic and environmental conditions. Many vegetable and fruit growing farmers in Germany, however, are still dependent on the manual work.
A robotic transportation platform for harvested strawberries is presented that aims to improve the efficiency of the harvesting process and to facilitate the hard labor. As one of the most essential prerequisites, the robot has to be able to drive autonomously through the fields. In this thesis, an autonomous navigation system is proposed that can be applied to navigate in strawberry fields using the point cloud data from the ZED2i stereo camera as the only sensor data. After pre-processing the point cloud data, the height information is used to generate a grid map containing a two-dimensional projection of all points belonging to the crop rows. By applying image processing methods, the grid map is then used to detect the crop rows separately and to fit a line through the center row. Multiple cluster and line detection methods were implemented and compared with respect to accuracy and computational time.
In order to achieve a in particular robust navigation system, the methods were verified in real-world experiments. It was found that a clustering approach inspired by the sliding window object detection used in computer vision applications provides the most robust crop row detection. It is combined with a line fitting approach that is based on the method of least squares which requires the least computational time to achieve a reliable navigation system.

Keywords: agricultural automation; field robots; autonomous navigation; crop row detection; image segmentation; mapping; stereo vision

Contents

Chapter 1

Introduction

There is an increasing demand of autonomous mobile robots in the agricultural field as human workers are hard to find, especially considering the difficult circumstances caused by the Pandemic, that the minimum wage is raised and the population is constantly growing. Semi-automated vehicles and mobile robots are already being used to assist in a wide range of tasks such as harvesting, ploughing, seeding and weeding while achieving high precision performance. They contribute to decreasing the human's workload and to facilitating and possibly speeding up work processes. Even though there has already been a large technological progress in the agricultural sector for the last decades, there are still many tasks that need to be done by manual labor as they are hard to automate. Especially when harvesting delicate vegetables and fruits such as salads, berries or tomatoes the crops need to be handled with greatest care since they are highly sensitive to pressure and will degrade faster when they are bruised. While it is hard to mimic the high sensitivity of a human hand with agricultural machines or robotic end-effectors, it has become apparent that the use of robots during harvesting season is still beneficial for delicate and high value crops like strawberries which will be considered in this project thesis.

The picking of ripe strawberries is usually done by humans only. The procedure is depicted in the images in Figure 1.1. Human workers carry an empty wooden or plastic box with smaller cardboard container placed inside which can be seen in the right image. The pickers follow the strawberry rows and place the ripe fruits into the containers until they are filled. Sometimes small and simply constructed wheelbarrows are used to carry the box. As soon as the box is full, the farm workers have to carry the filled box to a trailer located at the edge of the field to exchange it for a new container with empty boxes. The harvesting season starts in May and usually ends in July or the beginning of August depending on the weather conditions and the type of strawberry plant that is used. The

(a) (b)

Figure 1.1: (a) The harvest worker follow the crop rows and pick ripe strawberries while carrying the harvested crops in containers. (b) A typical strawberry box contains cardboard containers that are filled with the ripe fruits.

harvesting procedure of one plant is repeated multiple times as fruits keep growing and ripening for a time period of four to six weeks. It was found that carrying the containers is not only a physically demanding task for the farm workers but also takes about 30 % of the working hours. Therefore, an autonomously driving transportation vehicle that carries full strawberry containers would already be an enormous assistance for the farmers. Such a mobile platform needs to be able to follow the harvest workers whilst picking in the fields to provide a nearby place where full containers can be placed and where empty containers and boxes can be stored. As soon as the total loading capacity of the transportation platform is reached, it needs to autonomously carry the filled boxes to the trailer and then return to the harvest workers with empty containers. This would reduce required time for the harvest process by approximately 30 % and would at the same time facilitate the physical labor.

1.1 Problem Statement

Considering the scenario of strawberry-picking during harvest season that was described in the previous section reveals that establishing an autonomously driving robotic platform would mean an undeniable necessary step to facilitate the hard work of strawberry pickers. While there are multiple approaches for field navigation to guide agricultural vehicles and mobile robots through the fields, a mobile transportation platform is needed that supports the workers in a collaborative work environment with human and robot interaction. Combining

the indispensable sensitiveness of the human hand with the advantages of a mobile robotic platform makes the task less physically exhausting and saves time. To accurately complete the just mentioned tasks the robot has to be able to precisely navigate in the fields by detecting the crop row structure which can be done by using a variety of different sensors and detection techniques. While the field structure and crop row features such as the color and height differ vastly for different crops, this thesis will only consider strawberry fields.
This project thesis contributes to this goal by presenting a navigation system that enables a mobile robotic platform to autonomously traverse through strawberry fields while taking into account the final goal to be part of a collaborative environment with human workers. Since the navigation system will be based on row detection, it is of great importance to choose a robust detection method and to guarantee a consistent high precision navigation. Considering all challenges mentioned above, the following requirement list sums up all important properties that the navigation system must satisfy.

Requirement List:

- robust: against noise such as weed, vegetation gaps and bumps on the driving lanes
- versatile: applicable under different environmental conditions such as lighting and weather conditions, soil and hilly or flat fields
- enables autonomous crop row detection of fully grown crops and row following
- minimal computational cost
- inexpensive: low-cost hardware components (e.g. sensors) to achieve an affordable product for farmers
- no damaging of plants while following the crop rows
- applicable in a real-time system with low speed (approximately $5\frac{\text{km}}{\text{h}}$)

To fulfil the entire operating principle of a transportation platform the robot not only needs to be able to traverse a field but also needs to be capable of swapping lanes and reaching waypoints such as trailers outside the field. Leaving and reentering the fields lead to new challenges as the robot reaches a point where it is not able to detect the row structure within the camera's range. This task, however, exceeds the scope of this thesis. It is always assumed that rows are visible on the camera image and as the robot will never be used on the outer rows of the field, at least one row next to the center row on each side respectively will always be present.

1.2 Thesis Structure

The following section provides a thoroughly overview over state-of-the-art agricultural field navigation approaches using different sensors and methods. Those methods will be evaluated considering the particular application of this thesis and, as a result of this, the most promising approach will be identified. The used hardware forming components of the robotic platform, the structure and main features of the considered strawberry fields and the currently used navigation approach is presented. In the subsequent section, the main idea of the chosen approach will be explained followed by a detailed description of all processing methods that have been implemented. The experimental evaluation of the presented approach will then be presented in Chapter 4 by describing the overall experimental setup and explaining the evaluation methods that were used. This is followed by an interpretation and critical discussion of the resulting evaluation parameters. Chapter 5 will then conclude the main findings of the project followed by a short outlook.

Chapter 2

State-of-the-Art

As automated navigation through agricultural fields has been studied and even applied in the past years there are numerous approaches that have been tested thoroughly. Although most fields have the typical row structure different crop types, environmental conditions and other factors make it impossible to get one perfectly working method for in-field navigation. Even if only strawberry fields are considered, as it is done in this project thesis, a broad band of different approaches can be found. The most appropriate sensors and detection techniques need to be chosen carefully and strongly depend on the requirements given by the use case. In the following section the available sensors will therefore be viewed followed by the crop row detection methods that have been successfully applied in other projects.

2.1 Literature Review

Fountas et. al give a detailed overview of the development of agricultural robots for research or commercial application in their research paper [FountasEtAl20]. In the paper, the robotic systems are classified according to the field operations they have been used for. The application area of agricultural robots is split into eight major operational tasks including weeding, seeding, disease and insect detection, crop scouting, spraying, harvesting, plant management and multi-purpose application. Finally, the authors conclude that the most research has been done on harvesting and weeding robots while seeding and disease detection has been studied at the least.

2.1.1 Sensors

The authors of the paper [FountasEtAl20] conclude that for autonomous navigation in the agricultural sector most robots are equipped with a Global Navigation Satellite System (GNSS) such as high-precision real-time kinematic Global Positioning System (RTK-GPS). Even though these sensors usually provide a sufficient accuracy their performance is affected by signal interruption and failure due to large changes in the appearance of the field. Additionally, accurate GPS in all agricultural fields is not guaranteed due to incomplete coverage. Despite those challenges, an automatically steered tractor is presented by Blackmore et. al [Blackmore04] that only relies on RTK-GPS signals. The tractor is able to follow a pre-defined path within a few centimeters offset but requires additional sensors for obstacle and people detection to become a practically applicable tool. As mentioned in the paper [Blackmore04], RTK-GPS can only be used when digital maps of the respective field are available for the navigation system to generate passable routes between the crop rows. If such maps are not available GPS markers need to be positioned to define the field boundaries, the existing row structure and the precise position of the individual rows. Those markers can be gained from manually driving a vehicle that is equipped with the appropriate sensors around the field to create a field map, by for example using remote control. Usually, this is done by generating a reference path whilst tracking the positional signal of the machinery that is used during planting or seeding. After the GPS markers have been generated the RTK-GPS navigation can be used for the whole season independent of the growth stage of the plants. The accuracy, however, is limited to the fact that the position of the rows stays unchanged.
When solely relying on GPS or GNSS, changes in the environment and shifted row positions are not considered. However, such changes can quickly occur due to heavy rains after seeding, unevenly growing plants or simply by single crops growing in between rows. For this reason, additional sensors are usually needed to take unexpected obstacles, human detection and out-of-row growing plants into account. They can also be used to overcome disturbances of GPS signals which particularly occur in under-canopy applications or at points in the field where the antenna gets covered by plants. The additional sensors are even necessary to achieve a robust long-term autonomous navigation system. Examples for these supplementary sensors are Light Detection and Ranging (LiDAR), color and stereo cameras. They are used to reconstruct the surrounding area of the mobile robot and to guarantee a consistent performance of the navigation system by combining global GPS measurements with local sensor information. The local measurements obtained by on-board sensors of the mobile robot can be added to the global GPS map to update the map during the season and to

record changes in the field.

Many of the sensors that can be used to increase the robustness of GPS-based navigation can also be used on their own and can provide a low-cost alternative to the rather expensive RTK-GPS sensors. The recorded data can in most cases not be used for global localization as the camera only detects parts of the entire field and doesn't know its global position in a field map but it can be used to determine its local position. Local localization or navigation refers to determining the robot's position not inside the global reference frame of the entire field but relative to the detected crop rows. It therefore focuses on accurately following crop rows without damaging plants and choosing the steering direction with respect to the local environment. The great advantage of this navigation approach is that it can be applied in many fields that don't need to be equipped with GPS markers or similar preparations.

In many projects local navigation of vehicles in fields is done by using color images recorded by inexpensive digital color cameras. The images are used to detect the vegetation in the images using visual differences, such as color and texture, between soil and vegetation. There is a large variety of different segmentation techniques that will be presented in Section 2.1.2 and have been studied extensively to even handle visually challenging environments. The segmentation results are used to determine the goal direction by transforming the information into the robot's coordinate frame and generating steering commands. The ongoing research in applying color cameras in the agricultural sector and the numerous available techniques demonstrate that there are many factors that need to be considered to avoid the system to fail. Image Segmentation is especially hard to realize due to the changing appearance of crops during the season, gaps between plants, weed, inconsistent illumination and many further sources of error.

Although digital cameras give very reliable results in indoor environments where constant lighting conditions are given, their performance suffers under varying illumination. Sensors that are directly providing distance measurements, such as time-of-flight (TOF) cameras and LiDAR sensors, are less sensitive to variable lighting conditions than the digital cameras which measure reflectance. Furthermore, those ranging-based sensors provide a larger range than digital cameras. 3D LiDAR systems and 3D TOF cameras can be used to generate point clouds that replicate the environment in three-dimensional space by points lying within the external surface of the scene. Point clouds can additionally contain the color information for every point. 3D LiDAR systems and 3D TOF are usually more cost-intensive compared to the previously mentioned sensors. Especially 3D LiDAR comes at relatively high costs compare to 2D LiDAR

but was proven by Zhang et al [ZhangEtAl13] to obtain more accurate results. LiDAR systems measure the time for the light transmitted by a laser to return to a receiver to measure the distance of surrounding objects. One of the other downsides of distance sensors is sensor occlusion which can easily happen due to leaves. Nonetheless, Velasquez et. al [VelasquezEtAl] where able to build a LiDAR based autonomous navigation system that runs on low-cost hardware and can navigate autonomously under the plant canopy. Their robot successfully navigated through under-canopy tracks of a typical field with only small gaps for 386.9 m without the need of intervention. Similarly, Gai et. al [GaiXiangTang21] considered the navigation under canopies which is especially the case during harvest season and for tall plants. Instead of using LiDAR systems, which has been increasingly used in the past due to decreasing sensor costs, the authors focus on a navigation system using a TOF-camera. This range sensor measures distance on the basis of the time of flight of modulated infrared light to get depth images of a scene. Gai et. al point out that the TOF cameras are more favorable than LiDAR when considering the low computational overheads at high acquisition rates and the wider vertical angle. The authors also mention that using TOF cameras outdoors remains almost unexplored because most commercially available TOF cameras are equipped with weak light sources which cannot provide high accuracy under direct sunlight. As a result of their research, Gai et al confirm the feasibility of a front-facing TOF camera for autonomous outdoor in-field navigation, since they achieved a mean absolute error (MAE) of 3.6 cm for mapping the crop rows and a MAE of 5 cm for inter-row positioning. Compared to LiDAR-based systems the TOF camera is less effected by sensor occlusion, weeds and leaves. The final system certainly requires further improvements to overcome limitations due to the maximum range, a narrow field of view in the horizontal direction, noisy measurement data and beyond that would benefit from sensor fusion.

Stereo vision cameras combine the advantages of range sensors and color cameras by exploiting the principle of human binocular vision. The sensor uses at least two cameras to perceive depth which enables it to generate point clouds and depth images similar to range sensors and is simultaneously able to detect colors which is more suitable for the detection of early stage and tiny plants. Kise et. al [KiseZhangRovira Más05] show a stereo vision based crop row detection method for tractor guidance that reaches a root mean square (RMS) error of lateral deviation of less than 0.05 m. Stereo vision cameras are no low-cost sensors but are usually less expensive than high-precision TOF or LiDAR systems. A further successfully working navigation system for agricultural tractors is presented by Hanawa et. al [HANAWAEtAl12] that was independent of the position and type of the two tested stereo cameras. Their approach uses distant image data to get 3D information from the surrounding area and accomplishes the goal of detecting

Table 2.1: There are many different sensors that can be used for autonomous navigation in agricultural fields. Those sensors generate various kinds of sensor data and are suitable in different environmental conditions. The comparison of the sensors demonstrates their advantages and disadvantages.

Sensor	**Evaluation**	**Measurement**
GNSS	+ most commonly used sensor + proven to reach a sufficient accuracy - malfunction for under-canopy use - signal failure can occur - requires previously generated map (preparation) - expensive when high accuracy is required - additional sensors for people detection needed - doesn't consider local environment - doesn't consider changes of the field	position
TOF	+ good for objects that are far away + wide angular view compared to LiDAR + less prone to weed and greens than LiDAR + less sensor occlusion than LiDAR + 3D features less error-prone to illumination inconsistency than color data + denser data than LiDAR - expensive sensor - weak light source not applicable for outdoor use	distance depth image point cloud
LiDAR	+ 3D features less error-prone to illumination inconsistency than color data + wider range than digital cameras - expensive sensor - sensor obstruction due to leaves and weed	distance depth image point cloud
Stereo Vision Camera	+ use color and 3D information - expensive sensor compared to color cameras - sensor obstruction due to leaves and weed - inaccurate measurements in the shade	distance depth image point cloud
Color Camera	+ low-cost sensor + many approaches to overcome weed noise, gaps etc. - only relies on color - plant color can change during season - usually crafted for particular crop appearance - Error-prone to weed noise, gaps, lighting conditions	color image

10 cm height differences and to stay in a tracking deviation of 5 cm. During the experiments the major challenge turned out to be the negative impact of shadow casted by the tractor which led to no distance images being generated in the shadows.

All sensors listed above present the most commonly used sensors for autonomous field navigation that can be used as standalone solutions. There is a variety of additional sensor that can be used to improve the crop row navigation and to achieve a more reliable system by also considers cases in which the main sensors fail. Merging different sensor data is called sensor fusion and is used to increase the safety of the system by embedding, for instance, an obstacle and people detection system. Van Henten et. al presented their experimental autonomous robotic platform Cropscout at the Field Robot Event 2004 which is built for research in precision agriculture for tasks such as weed and disease detection. In their paper [E.J. Van Henten, B.A.J. Van Tuijl, J. Hemming, V.T.J. Achten, J. Balendonck05] the authors describe the used navigation system that exploits the advantages of fusing sensor data. Here, sensors such as an inclinometer, pulse counter, an ultrasound and an infrared range sensor and whiskers are used to confirm the row detection based on color images in the visible spectrum recorded by a digital camera. The resulting system presents a viable solution that is applicable in real maize field and could obtain the first prize at the Field Robot Event since it was able to pass all test impeccably. The disadvantage of using multiple sensors is the increase of costs and the rising complexity of the system which can also raise the computing time. The Table 2.1 lists the most commonly used sensors for field navigation, the type of measurement data that is collected and their pros and cons considering the problem statement given in Chapter 1.1.

2.1.2 Methods

The choice of methods to be applied for a navigation system is limited by the available sensor data. The selection of sensor should therefore not only depend on the sensor characteristics but also on the different techniques applicable with a specific sensor type.

Using only visual feedback works really well for well-structured fields but can fail when gaps occur due to missing plants in the rows or when noise appears in the form of weed that covers the ground. Many methods have been presented to overcome these issues by applying pre-processing steps using filtering and clustering methods to get rid of noise and gaps. Such pre-processing is a key step in order to detect the crop row structure in two-dimensional color images. The

processing of the input images is usually divided into two main steps, the image segmentation followed by the line or pattern detection. In his master thesis [Aske Bay Jakobsen15], Aske Bay Jakobsen mentions the most commonly applied image processing steps starting with image segmentation. As a result of image segmentation, a black and white image is generated that distinguishes between vegetation and ground. Then, a line or pattern detection algorithm is applied on the binary image to detect the crop rows. As vegetation is expected to be green the excess green vegetation (ExG) index is often used that can be calculated from the 3-channels of an RGB image which represent the three colors red, blue and green. Many crop row extraction methods rely on the ExG index as a distinguishing feature for vegetation in color images. It is applied as a mask to accentuate any green in the image which is expected to highlight any vegetation in the image.
The ExG index is also embedded in the extraction method presented by Yue Hu and He Huang [HuHuang21] and by Jiang et. al [JiangZhao10] where it contributes to a robust crop row navigation system and can not only help to make plants more prominent in the image but to also suppresses shadows. After applying the ExG index, a threshold is set to generate a binary image. The threshold value can be calculated automatically by for instance using Otsu's method which takes into account changing color values due to illumination inconsistency and varying crop appearance. As an alternative to the RGB color space, other color spaces such as the YUV color space can be used for segmentation as presented by Bay Jakobsen [Aske Bay Jakobsen15]. He only uses the U and V layers to create a binary image which are the only layers containing color information in the YUV color space. This is done by using the grayscale images of both layers and turning them into binary images by again using a threshold value. Combining both resulting images using a logical AND operation then generates a clear separation between ground and vegetation.
Using color data has led to good results using further improvements such as a bounding box clustering method used by Yue Hu and He Huang [HuHuang21] or further changes in the processing steps as they have been presented in many further research papers [NichollsGreen], [Wang10], [SøgaardOlsen03], [RomeoEtAl12] to overcome the main challenges occurring from vegetation gaps, weed noise and poor lighting conditions. A purely vision-based approach is presented by Ahmadi et. al [AhmadiEtAl20] that is independent of expensive RTK-GPS sensors and is even able to switch to the next lane. In the project the ExG index is used to compute a vegetation mask to then compute the crop's center point for each mask. Afterwards, a least square fitting method is used to robustly compute the lines that best fit the center points. In the author's subsequent paper [AhmadiHalsteadMcCool] the approach was enhanced and reached an accuracy of 3.82 cm in five different crop fields using GPS measurements as the ground truth. The testing was done for various lighting conditions and is robust in scenes where large batches of weeds occur. The on-board sensors that

are used for visual navigation are two RGB-Depth (RGB-D) cameras that are placed in the front and in the back of the robot and are sufficient to navigate through the field without knowing the global or local position of the robot.

In addition to the normally used methods for image segmentation, machine learning approaches are increasingly used for crop row detection especially for fields where the crop rows are hard to detect or even when their appearance is not known before. Using such modern deep learning approaches can achieve a better segmentation than common image segmentation and makes the detection approaches more applicable for different crop types. The navigation system presented by English et. al [EnglishEtAl15] consists of a model that takes into consideration color, texture and three-dimensional structure measurements of a specific field that are provided by two front facing stereo vision cameras. The position of the crop rows is estimated by a Support Vector Machine (SVM) regression algorithm. The vehicle was tested on fields of four different types of crops. Even though only in the first two fields green vegetation segmentation was applicable, the algorithm could successfully detect the rows in all fields. Comparing the row following with the SVM to the path taken when using high-precision GNSS showed that the maximum RMS error was less than 3 cm. The authors conclude that the system could even be used as a low-cost solution by adding measurements of an inexpensive GPS sensor to the output of the system within a particle filter.
The navigation system in the paper written by Ponnambalam et. al [PonnambalamEtAl20] also exploits the advantages of using machine learning for crop row detection. They show that the model can adapt easily to changes in the crop appearance and can overcome the challenges of an unstructured outdoor environment by even handling uneven contours of crop fields and irregular spacing of plants. The segmentation is done using a convolutional neural network (CNN) that is able to perform a vegetation segment on input RGB images. As a result, the authors demonstrate an improved performance of crop row detection compared with other commonly used fitting methods. The major disadvantage of using deep learning methods is the large amount of data that is required to train the models. The data acquisition, preparation and labelling are very time consuming.

Mapping is a widely used approach for autonomous in-field navigation. Field maps can be used for crop scouting, disease detection, to record changes in the field over the season and to obtain and store other relevant information for the farmer. Mapping can be done by updating existing GNSS maps or by building a map from only local sensor data. The second approach is called occupancy grid map or feature map generation. It uses the sensor data to generate a grid map whose cells contain the probability of the space being occupied by an ob-

stacle. This map generation approach makes use of on-board sensors and can be used for path planning in rough terrain as it is presented by Annett Chilian and Heiko Hirschmüller [Annett ChilianHeiko HirschmÃ¼ller09]. Using a stereo vision camera, the authors show how they reconstruct the robot's environment for save navigation through unknown terrain using legged or wheeled robots. While exploring its environment, the robot is able to constantly update the map and to plan a suitable path. In their project, the mapping focused on computing the traversability of terrain by computing a danger value for the map's cells.
The feature map generation as part of the navigation system designed by Winterhalter et. al [WinterhalterEtAl18] focuses on building a reliable crop row detection algorithm for different crop types that is applicable in any crop row stage and particularly suited for small plants. The main advantage of the approach is that, after a two-dimensional feature map is generated, the map can be used as a generic input for the following processing pipeline independent of the used sensor. This leads to a particularly robust crop row detection as the used sensor can be chosen dependent on the given environmental conditions. To prove this concept, the authors created feature maps based on RGB images and on 3D point clouds provided by a 3D laser scanner and could achieve a robust navigation through fields which is especially applicable in precision agriculture. In April 2020, Winterhalter et. al published an article [WinterhalterEtAl21] to present their improved navigation system. They enhanced their previous approach by estimating the robot's position in a GNSS-referenced map of crop row locations by merging GNSS-based data with the computed feature map. Associating the uncertain measurements of the on-board sensor that are stored in the feature map to the known crop row structure in the global GNSS reference map is called data association.
An alternative way of using reference field maps is shown by Chebrolu et. al [ChebroluEtAl19]. The authors present a localization approach that uses a field map generated from camera recordings taken by an unmanned aerial vehicle (UAV). The approach was evaluated on a real field over various sessions in a period of time of multiple weeks. Exploiting plant specific features, the method performs data association to localize the robot in the aerial map by using visual data from an on-board ZED stereo camera. Continuously updating observations into the map allows the system to navigate throughout the entire season. The approach aims to resolve the issue of visual ambiguity due to the repetitive structure of fields and to cope with changes in the appearance of plants over time.

2.2 Strawberry fields

Strawberries are planted in different ways to increase the yield and protect the plants. In Figure 2.1(b) the plants are placed on racks which are usually used

for indoor farming. For this project fields with mounds are considered, because they are the most commonly used way of planting strawberries in Germany. Such a field is depicted in Figure 2.1(a). The image shows a typical strawberry field as it has been used for the experimental evaluation of this thesis. It represents the prospective work space of the robot. As fields with mounds can be located both indoors and outdoors, the navigation system is challenged with varying environmental conditions. The raised mounds serve as a base for the strawberry plants. Hence, a distinctive height profile is expected that undergoes minor negligible positional changes after planting. The vegetation is always significantly higher than the ground even before the fully grown stage is reached. Farmers use mounds to get improved conditions for their plants including drainage, increased air circulation around the plants and preventing diseases to spread. Black foil is used to cover the mounds especially before planting season begins to protect the soil and to prevent weed to grow. While the strawberry plants are placed on the loose and evenly humidified soil of the mounds the tracks between the crop rows gain less attention as weed or soil compaction doesn't harm the development of the plants. This aspect is where the strawberry fields vary the most in their appearance. While some farmers put straw on the tracks causing a uniform appearance, others hardly take care of weed removal. When the strawberry plants are fully grown, they tend to not only cover the soil of the mounds but their canopy of leaves also tends to hide large parts of the tracks in-between rows.

The crop row structure is predefined in the planting season and only slightly varies throughout the season due to environmental influences. The inter-row distance

(a) (b)

Figure 2.1: There are many different methods available to protect the strawberry plants for yield increase. The most commonly used methods of German strawberry farmers are to plant their strawberries on mounds (a). Alternatively, racks are used (b). Both methods help to protect the strawberries from insects, weeds and to improve water drainage.

is fixed to 1 m and the rows are always parallel. The mounds have a height of 20 cm and their base is up to 40 cm wide.

2.3 Robotic Platform

The mobile platform that was used for the project was chosen on the one hand to be able to traverse through possibly rough terrain as it is common in agricultural work environments and, on the other hand, to build a low-cost vehicle. The robot consists of four wheels and is actuated by a differential drive acting on both front wheels. The differential drive is a two-wheeled drive system with independent actuation of each wheel used to drive the robot forward. A DC motor controller provided by Roboteq is used. The height of the platform is chosen such that it allows the robot to drive over strawberry rows without damaging the fully grown plants planted on standard-sized mounds.

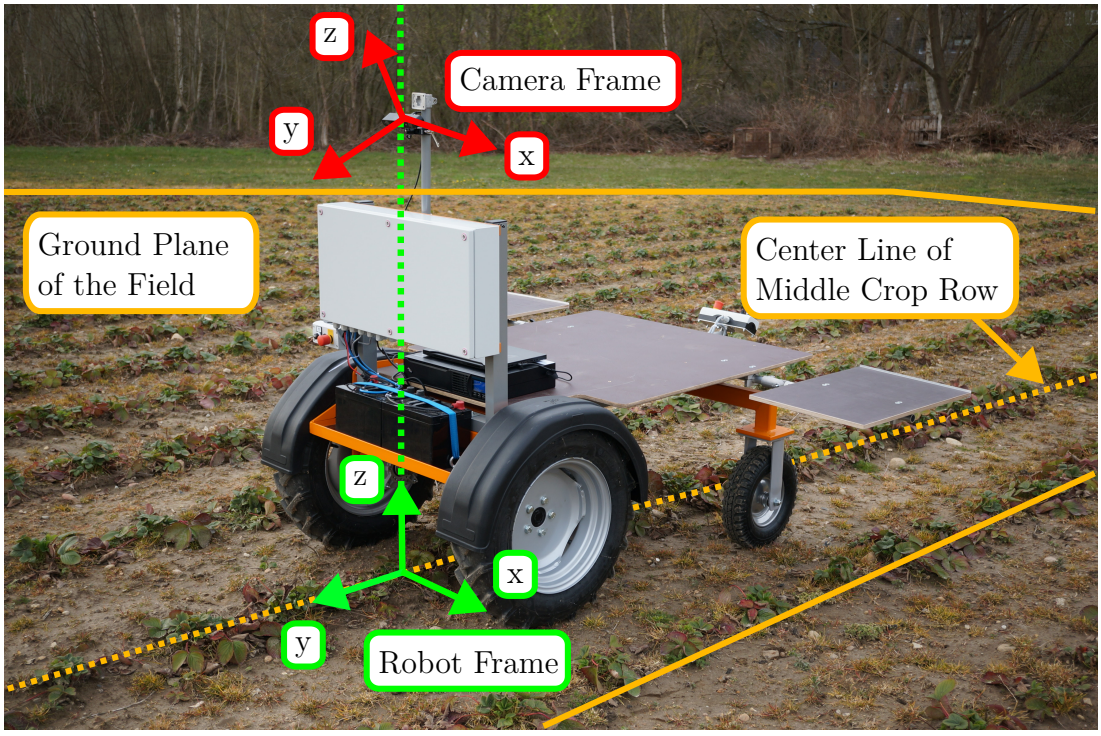


Figure 2.2: The transformation between camera frame (red), robot frame (green) and ground plane (orange) need to be considered in the navigation system.

Furthermore, each of the front wheels has a width of 27 cm. The total width of the robot which is measured at the front wheels 1.2 m. Those dimensions are chosen to enable the robot to drive over the crop row spaced at a standard distance of 1 m, as mentioned above. The measurements result in a maximum possible lateral offset of 26 cm for the robot to stay within the inter-row tracks.
The front side of the robot can be equipped with a sensor to collect data of the robot's environment and includes a box containing all required electrical components leaving the rest of the platform for storage purposes. To achieve a maximum storage space the platform can be extended with side panels to load further harvest boxes. For the project, the ROG Zephyrus G15 laptop by ASUS was used to run the code. While a laptop is not useful for processing in the final product, it is an appropriate device for testing and visualization for a first experimental validation of the output while running in the fields. To enable the communication between all included devices CANopen, a communication protocol, is used. The Controller Area Network based (CAN-based) communication system is built for embedded systems and is mainly used in automation.
The image in Figure 2.2 shows the way the camera is mounted on the robot. The camera's frame is marked in red with its y-axis pointing towards the viewing direction. The camera is located at the center of the front side of the robot and points to the ground with a pitch angle of 30 °. For accurate results, this angle, which is part of the extrinsic parameters of the camera, needs to be measured precisely as it is used to transform the measured point cloud data from the camera's coordinate frame to the robot's coordinate frame. The reference frame for all processing steps on the point cloud is always the robot's frame originated at the middle of the front side of the robot which is marked in green in Figure 2.2. The y-axis of the reference frame points towards the direction of travel. As can be seen in Figure 2.2, the x-y-plane of the robot's coordinate system is aligned with the ground of the field, highlighted in orange. Hence, the positive z-axis is perpendicular to the field's ground plane and points towards the sky. Both, the origin of the robot's frame and the origin of the camera's frame lie on the z-axis of the robot's frame. Unless otherwise stated, everything will be considered in the robot's coordinate frame.

2.4 Row Navigation Algorithm

The currently applied approach uses the point cloud data to extract required information of the surrounding row structure for navigation. First, the input cloud is cropped and a voxel grid filter is applied to reduce the amount of data to be processed. A detailed description of how those filters work is given in the following Chapter 3. The cloud is then rotated around the x-axis until its height reaches a minimum value to align the field's ground plane with the x-y-plane of

the robot's frame. A slice is extracted from the point cloud that is parallel to the robot's x-z-plane and is located 2 m in front of the robot. The slice has a thickness of 0.5 m in y-direction. All points that lie inside this slice are projected on the x-z-plane to generate a two-dimensional image. The side length of each square pixel is 0.0069 m. All pixels containing points independent of their y-position in the slice are black. The remaining pixels are white. This black and white image is then pre-processed to remove noise. To get the shape of the mounds as a discrete linear convolution only the highest z-value is chosen for each x-value as it can be seen in Figure 2.3. The x-coordinate for the maximum z-value is chosen as the location of the center row which is marked in red in the graph. The difference between the robot's x-position, which always lie in the center of the image, and the row position is used to calculate the lateral distance between robot and crop row. This value is used as the input command for the controller.

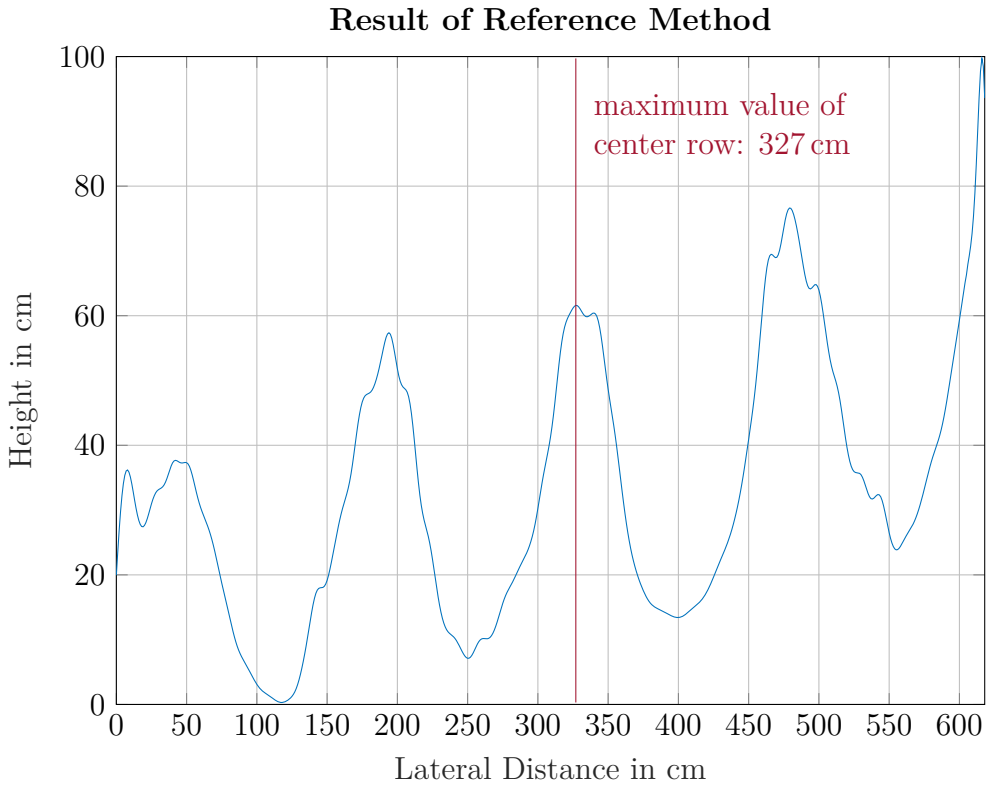


Figure 2.3: Result for the extracted height profile of the reference method that is currently used for autonomous field navigation.

Chapter 3

Methodology

Overall, there are various valid approaches that have been studied for autonomous crop row navigation that can be applied in different use cases. As the robot needs to collaborate with the pickers, on-board sensors are a requirement for safety reasons and will allow the robot to follow the workers during harvest. For this project, the robot is only required to perform a local navigation by determining its position relative to the surrounding crop rows to traverse through the field without damaging plants. GPS markers are not included in the current development status but will most probably be used for more advanced versions of the robot. They are necessary when the global position of the robot needs to be known especially when leaving the field environment. As this is not required at the current stage of technology and the platform should be a low-cost solution, the ZED2i camera from Stereolabs is selected as the only sensor needed for the navigation system.
During harvesting season, the inter-row space has strongly varying colors and texture and can even be covered by green leaves from the strawberry plants and other plant material. For this reason, it is unfeasible to build a reliable segmentation by only using the information stored in color images. Besides, the fields will always exhibit a clearly visible elevation profile as most strawberries are planted on top of mounds and as only fully grown strawberry plants are considered. The chosen stereo camera is an appropriate sensor to provide 3D environmental data and comes at a lower price than most range sensors. Purely relying on 3D data contributes to a reliable navigation system nearly independent of weed noise and gaps as the mound structure will always be present. It is also less error-prone to varying lighting conditions compared to color cameras.
The ZED2i camera provides an IP rating of 66 which guarantees a reliable performance in rough outdoor environments. The two digits stand for full protection against dust and similar particles including a vacuum seal that was tested

against continuous airflow and resilience to high pressure jets. Furthermore, stereo vision cameras are well suited for the detection over long distance and of moving objects. The ZED cameras are passive sensors which don't have any lasers or LEDs as active sensors have. Passive stereo vision can be much more affordable than other 3D machine vision technologies and is suitable for most ambient lighting conditions because they don't suffer from sunlight interferences since they are not using IR light. The ZED2i camera is also equipped with a polarizing filter that is placed in front of the camera to filter out sunlight. This helps to decrease reflections and increase color saturation making the sensor more suitable for outdoor applications. While the ZED2i camera provides a good sensor for the development of the navigation system due to its high-definition and robustness to challenging outdoor conditions lower-cost alternatives can be considered. The Intel RealSense Depth camera was utilized as an instance for such a sensor to test, if it achieves a similarly robust in-field navigation.
When taken into account the broad research that has been done on autonomous in-field navigation, it becomes obvious that there is only little research on using stereo vision cameras as the only sensor. Most papers using stereo vision cameras still rely on additional information from GNSS or only use the color images obtained by the camera. Using only 3D information from point clouds and depth images is usually done when using range sensors as a measurement device. Therefore, this project will contribute to a research field that needs to be investigated in more depth and will look into applications during harvesting season.
Using machine learning algorithm for a more reliable crop row detection is not applicable in this thesis as the data available at this time in point is not sufficient to train a neural net in order to get satisfactory results.

The pipeline of data processing is split into two main parts, the segmentation and the row detection. In the segmentation step the point cloud data is used as an input to segment the vegetation from the ground using height information. This step results in a binary grid map which is equivalent to a black and white image as its cells are either filled with 0 or 1 which stands for vegetation or ground respectively. In the second part image processing steps such as clustering and line detection are performed on the binary map to detect the three center crop rows. Finally, the results of the row detection is used to generate steering commands. The separate steps can be seen in Figure 3.1 which at the same time represents the core nodes of ROS which has been used as a framework for this project.

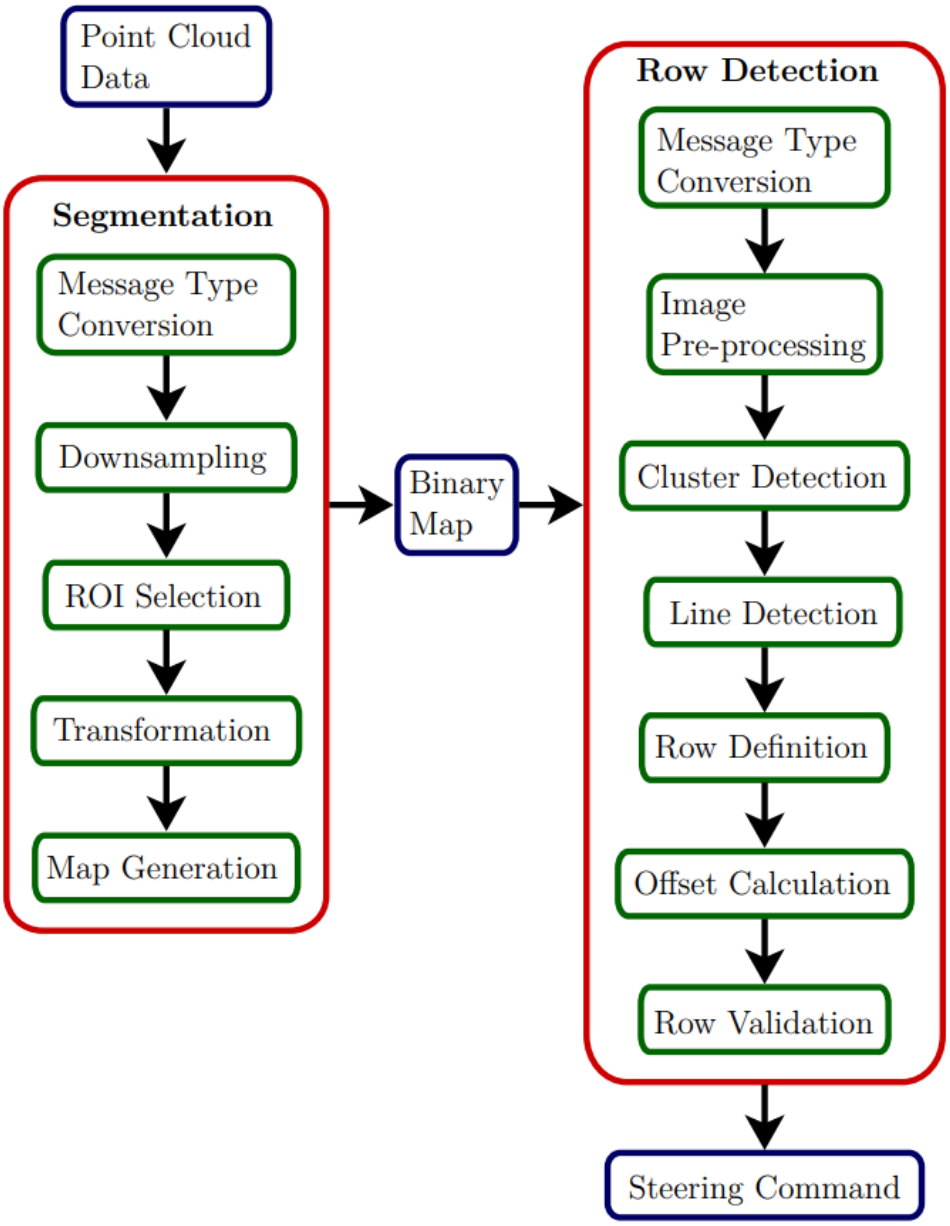


Figure 3.1: The input point cloud is processed to generate a two-dimensional binary grid map. This grid map is then used to detect crop rows to determine the steering commands for the robot. The nodes are displayed with a red, the topics with a blue and the processing steps with a green frame.

3.1 Segmentation

The input point cloud processing starts with first converting the ROS point cloud message into the points cloud message in the point cloud library as most processing steps make use of the library.

3.1.1 Pre-processing

Since the approach only relies on detecting the three center crop rows a region of interest (ROI) is selected in the next step. This is also done to reduce the size of the point cloud and to get rid of measured points that are far away as the probability of measurement errors increases with increasing distance to the camera. The point cloud library provides a passthrough filter which cuts off all points that are placed beyond a pre-defined range. The ROI includes points at a distance of 3 m in direction of travel and of -2 m and $+2$ m in x-direction. Points that are higher than the expected maximum plant height of 3 m will also be cut off with the passthrough filter. It can be assumed that they represent measurement mistakes or obstacles like humans, animals or other objects not belonging to the crop field structure.
Then, a downsampling step using a voxel grid filter is performed to reduce the data, which means reducing the number of points. It is an essential step to keep the computational cost as low as possible. The voxel grid filter divides the space in which the point cloud is placed into a three-dimensional grid with individual three-dimensional boxes as grid elements. The grid elements have a given side length also called the leaf size. The voxel grid filter iterates over all points in the cloud. All points located inside the same grid element will be approximated with their centroid. While this approach is more time-intensive than just choosing the center of the grid element, the resulting point cloud is a more accurate presentation for the input cloud. This downsampling method is applied after the passthrough filtering, because it is more time-intensive when comparing the processing steps on the same input data.

3.1.2 Alignment of Frames

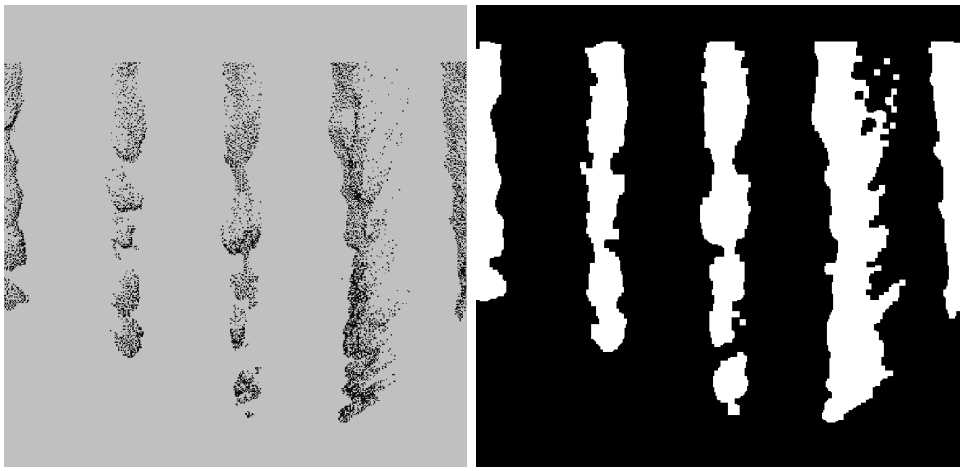
The extrinsic parameters of the camera are measured before navigating through the field to transform the point cloud into the robot's frame. This approach doesn't consider if the camera is not mounted correctly or if it changes its orientation over time. It also disregards hilly fields, bumps or similar unevenness of the ground. Since only the ROI is used for detection and the fields have a mostly even ground and no hills, the crop row navigation is still applicable while

neglecting uneven surfaces. The disturbances occurring due to such conditions are expected to be only of short periods of time and to only cause insignificant deviations. While a dynamic alignment of the point cloud is required to get more accurate results and to enable passing through rougher and hilly fields, further research need to be done in future work as it is out of the scope of this thesis. The alignment could possibly be done by applying a plane detection method to apply the correct transformation of the point cloud and by including inertial measurement unit (IMU) data.
To place the point cloud with the ground plane of the robot's frame the point with the smallest z-value is placed inside the x-y-plane and set to $z = 0$ as they are positioned on the surface of the ground in the inter-row space.

3.1.3 Map Generation

After the point cloud has been transformed and the ROI has been selected, a segmentation of ground and vegetation is performed. Since the strawberry plants are placed on mounds it can be assumed that points of the cloud that are at a predefined height from the ground plane belong to plant while the remaining points belong to the ground. That is why a height threshold is defined that determines if points are considered to be part of the ground or the crop row. While the threshold value can be chosen dynamically by calculating the z-range of the cloud. It has however become clear that a fixed threshold value results in a more accurate grid map because the mounds have a constant height and the impact of obstacles or noisy data will be reduced.
After all points have been classified as either vegetation or ground, the extracted points belonging to the vegetation are stored in an array. These points are transferred inside a grid map which represents the x-y-plane and contains the x-y-coordinates of the vegetation. After setting the resolution of the map and initializing an empty map only containing 0 entries, cells that comprise vegetation points are set to 1. The position of the points in the map only depends on the x and y-coordinates and doesn't take into account the z-value. All remaining cells containing 0 indicate ground. The grid map visualizes the crop row structure projected on the ground plane. In Figure 3.2 such a grid map is visualized. Figure 3.2(b) image demonstrate an Occupancy Grid map visualized by ROS which is generated from the field shown in the camera's color image in Figure 3.2(a).
The point cloud data is only used to extract height information for segmentation and is not used for further line fitting or clustering as this consumes more computational power than processing two-dimensional images such as the grid map.

(a)



(b) (c)

Figure 3.2: (a) A RGB image is recorded by the ZED2i camera while the robot drives through a strawberry field. (b) The information included in the point cloud data is used to extract row features to generate a binary grid map. All points belonging to vegetation are extracted from the cloud and projected on the map. (c) The grid map can be processed as a black and white image. White pixels represent crop rows.

3.2 Row Detection

In the second processing step the grid map is first converted into an OpenCV black and white image which can be seen in Figure 3.2(c). The cells of the grid map represent the pixels of the image and their values 0 and 1 stand for either white or black respectively. The OpenCV library is used for the image processing because it is a programming library that provides a variety of real-time computer vision methods. The goal of the second step is to estimate the center line of the center crop row to generate steering commands.

3.2.1 Pre-processing

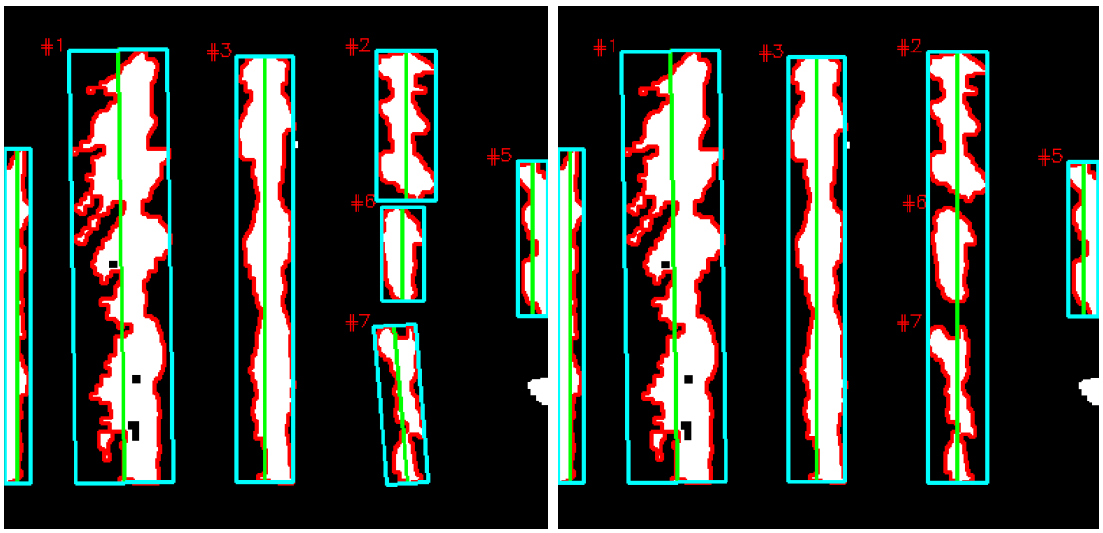
Pre-processing steps are used to unite clusters belonging to the same crop row and to get rid of noise represented by isolated white pixels which can occurs for example due to weeds or measurement errors. Dilation, a morphological filter, is applied that convolves the image with a kernel. The used kernel is square shaped with its anchor point being placed at the center. When the kernel is scanned over the image the maximum pixel value is computed for the pixels that overlap with the kernel resulting in bright regions to grow. This helps to merge clusters that belong to the same row but that are separated in the map due to vegetation gaps within rows. Erosion, the opposite filter of dilation, is applied to remove noise in the inter-row space to obtain a cleaner binary image.

3.2.2 Clustering

After the pre-processing of the grid map is completed the cluster detection is performed. Two clustering method, here referred to as contour clustering and window clustering, were implemented.

Contour Clustering
A connected-components analysis using the scikit-image library is used to detect clusters of white pixels and to filter out small noise regions in the binary input image. All white pixels that are connected with each other belong to the same group and are assigned with the same label. This is done by scanning all pixels of the image to identify regions of connected pixels. For each white pixel their four neighbors, that have already been scanned, are examined. A new label is assigned to the pixel if all four neighbors are black. If only one neighbor is white the white pixel is assigned the same label. In case more than one neighbor is white one of the labels is assigned to the current pixel and a note of equivalence is made. After the completion of the scan, a unique label is assigned to each

class and the equivalent label pairs are sorted into equivalence classes. During a second scan each label is replaced by the label assigned to its equivalence class. A mask is generated for all labels that contain a minimum number of pixels to only select the large clusters and to mask out noise.
The contour for each mask is detected in the next step. Then, for every contour a rectangle is positioned in such a way that it encloses the entire contour as well as its area reaches a minimum. An example image of such cluster detection can be seen in Figure 3.3(a). The angle of the rectangle defines the orientation of each cluster. If weed in between crop rows occurs or the robot traverses uneven ground rows possibly occur as one cluster in the feature map. If this happens, the width of the row is detected to be larger than the expected row width of 120 m. If the width is exceeded the cluster is invalid resulting in a failed detection that will be disregarded for steering. Rectangles that have a slope larger than 140 ° or an angle smaller than −40 ° are deleted as their orientation lies outside the maximum possible range. The slope of each detected rectangle is also compared to the average slope of the other rectangles. If the angle difference between the slope of a rectangle and the average slope of the remaining rectangles is larger than the maximum allowed angular deviation of 20 ° the cluster is neglected.



(a) (b)

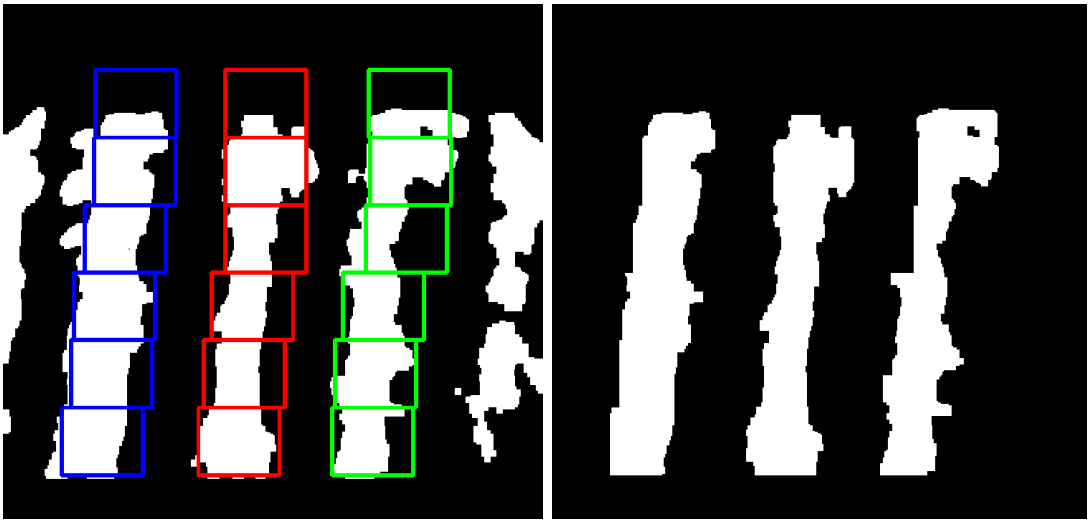
Figure 3.3: (a) All white pixels that are enclosed by the same contour line are detected as one cluster and displayed with their bounding rectangle of minimum area. (b) All detected clusters belonging to the same crop row are then merged and again displayed with their bounding rectangles of minimum area.

As multiple clusters can belong to the same crop row, if for example vegetation gaps occur in the gid map, the respective clusters need to be merged. This is done by comparing the offset of the center lines of each cluster. Using the center line including its orientation with respect to the robot's frame the distance between the robot and the related crop row can be computed. If the difference of the row's lateral position is smaller than a 25 cm the clusters are merged as they belong to the same row. This is depicted in Figure 3.3 where the clusters 2, 6 and 7 are merged. The center crop row is assumed to have a maximum lateral offset of 25 cm. If all clusters are located further away than this distance, the center crop cannot be detected. After the center row detection, the left and right crop rows are determined by choosing the clusters with the closest negative and positive offset respectively. The left and right cluster or the center cluster detection needs to succeed to apply the subsequent line detection method.

Window Clustering

The second clustering method is used to avoid false row association due to false crop row merging and separation that was observed when applying contour clustering. The method is inspired by the sliding window method that is mainly used in computer vision for object localization. This is done by moving a window formed by a rectangular region of a fixed height and width over an image. An object detection algorithm is then used to detect the object in the respective window. The idea of using multiple windows enclosing parts of the crop rows is used in several crop row detection approaches. García-Santillán et al. [García-SantillánEtAl18] obtain binary images from vegetation segmentation using the ExG index and then divide the image into 10 substrips. For each substrip and each of the four rows that are to be found a micro-ROI which is here referred to as "window" is placed in the image. The initial points of the crop rows are computed by getting four peaks of the Hough polar space that is calculated in the lower half of the binary image. The windows of predefined dimensions are then placed in each substrip such that they enclose the maximum number of white pixels and thus the strawberry plants. Following this approach even curved rows can be detected by applying appropriate line fitting like the least square technique for linear and quadratic polynomials. Similar approaches placing micro-ROIs in horizontal strips before fitting the row's center line in the binary image are presented in further reports [PonnambalamEtAl20], [ZhouEtAl21].

In this project the binary image is also split into horizontal strips as it is previously known that the crop rows will be present at each y-position, synonymous with the height of the image. Solely the x-position and slope of the row with respect



(a) (b)

Figure 3.4: (a) The windows are placed over the crop row such that they are placed at the average position of all white pixels belonging to the same row cluster. (b) All white pixels enclosed by the windows belonging to one row are considered to be part of the crop row cluster.

to the robot's frame varies. Each strip comprises a part of the clusters of all three guiding crop rows. These white pixels are localized for each strip and each row separately by selecting a search region of the image to then determine the center of the present white pixels and thus the position of the row. The dimensions of the search region are dependent on the crop row width, the inter-row distance and the maximum lateral distance to be expected. While in the approaches mentioned above the initial center point of the window for the first strip is determined applying laborious methods, here the previously known features of the strawberry fields are exploited. As the robot's wheels are always placed to the left and the right of the center crop row, the lateral distance and angle between robot and field frame is constrained. In the first strip, which is the horizontal region that is closest to the robot, the impact of the angular difference between robot and field frame on the lateral offset is negligible. It needs to be considered for the upper strips of the more distant regions, since it proportionally increases with the distance from the robot. This is considered by setting the base x-position of the search window to the x-coordinate of the x-position that was determined in the previous strip for the same crop row. In Figure 4.10(a) the positioning of the search windows can be seen. Afterwards, all white pixels enclosed by the windows of the same row are extracted and defined as the row cluster as it is displayed in Figure 4.10(c). The offset between the robot's frame or the previous window and the crop rows is assumed to be zero. This assumption is valid as the robot

is always placed between rows and will therefore not exceed a maximum offset. Crop rows connected by white pixels cannot be detected as one crop row as the width of the window is set invariably. The approach is also chosen to minimize the effect of isolated white pixels or small batches of white pixels occurring due to noise and to reduce the effect of overhanging shoots. No crop row is detected if no or too few white pixels are present in the image.

3.2.3 Line Detection

The rows of the considered fields are always straight, so that their center lines can be presented by the equation $y = mx + b$ of a straight line. In this thesis three different line fitting algorithms were implemented and evaluated in terms of computational time and correctness.

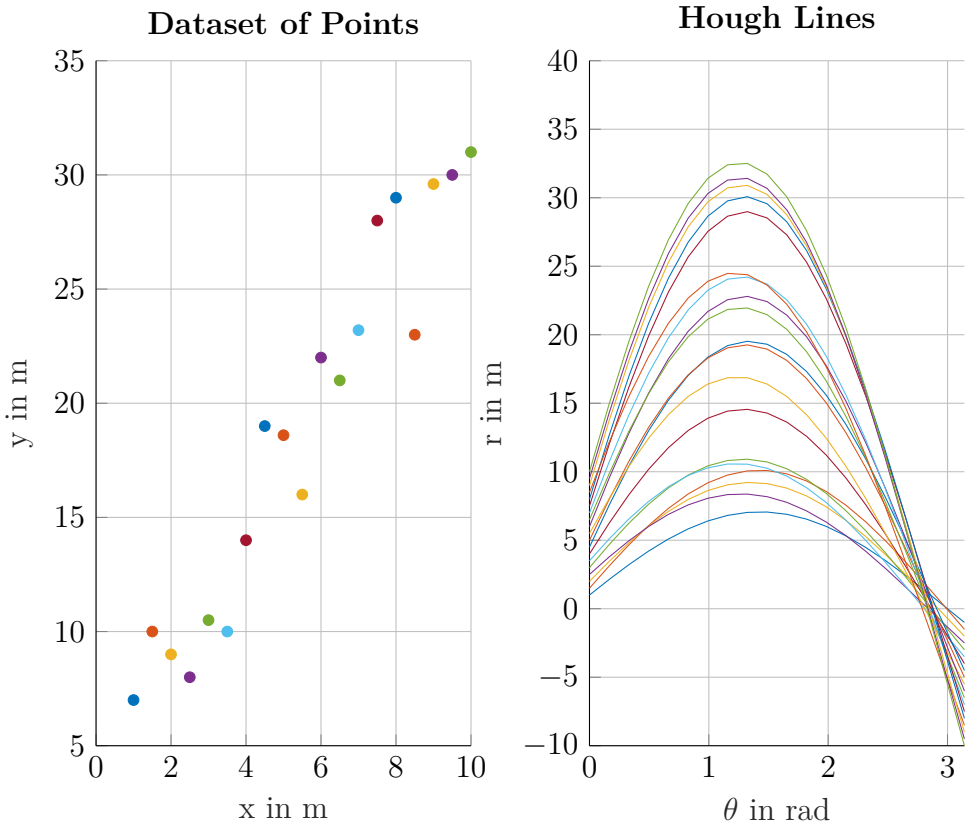


Figure 3.5: A noisy dataset of points is given as an input to the Hough Line method. For each point a family of lines can be found and plotted as a sinusoid. The intersection represents the common line of all points.

Hough Lines

The Hough Line Transform is a method that is used to detect straight lines in grayscale or, as is the case here, in a binary image. The lines are represented in the polar system and can be describes with the equation $r = x \cdot cos(\theta) + y \cdot sin(\theta)$. For each point $P = (x_p, y_p)$ a family of lines can be defined by the pair (r_θ, θ) representing all lines that pass through P. In the left image of Figure 3.5 a noisy dataset of points forming one line is shown. For each point of all considered points in the given image the family of lines can be plotted as a sinusoid in the θ-r-plane. For the crop row detection, the white pixels belonging to the same row are used as the point dataset. The sinusoid of the point dataset is shown in the right image of Figure 3.5. Using the OpenCV function "HoughLinesP" only points for $r > 0$ and $0 < \theta < 2\pi$ are considered. Plotting this sinusoid is done for every point in the dataset. An intersection of two plots indicates that two points have the same line. The number of intersections equals the number of points that are passed by the line. If this number of points exceeds a given threshold, the line is declared as a valid line. Here, a more efficient implementation, the Probabilistic Hough Line Transform, is used which gives the extremes of the detected lines as an output. It only uses a subset of the input data points that are randomly chosen to speed up the computations. Line segments that have a maximum allowed gap and segments with a minimum length are considered. All detected lines are then used to calculate the average line which is considered as the row's center line.

M-estimator

The "fitLine" method provided by the OpenCV library is based on the M-estimator which is an extremum estimator that searches for the zero of the estimating function. It iteratively fits the line using the weighted least squares algorithm. It minimizes the sum of the distance function between all considered points which are all white pixels of the crop row and the line. The distance functions, specified by the parameter "DistType", that are applied here are $\rho(r) = \frac{r^2}{2}$ (DistType = L2) and

$$\rho(r) = \begin{cases} \frac{r^2}{2} & \text{if} \quad r < C \\ C \cdot (r - \frac{C}{2}) & \text{otherwise} \end{cases} \qquad C = 1.345$$

(DisType = DIST_HUBER). The first is the simplest and the fastest least square method, whereas the second distance function results in a Random sample consensus (RANSAC) fit. RANSAC is an approach for parameter estimation that uses the minimum number of data points to estimate the desired solution. It is based on resampling as new the chosen set of data points is enlarged by new points until a suitable model is found. The process is repeated by a maximum number of iterations that needs to be chosen high enough to achieve a set of data

points with a sufficient number of inliers. In the following, the methods will be referred to as Least Square and RANSAC line method.

Bounding Rectangle
The third line detection method, the Rectangle line method, draws bounding rectangles of minimum area around the white pixels of each cluster as it was done when assigning the detected clusters to the respective rows. The center line of the rectangle defines the center line of the crop row.

3.2.4 Row Definition

The three center rows closest to the robot are used to navigate the robot along the rows. Each crop row is defined by two points that are located on the x-axis of the robot's frame and on the upper edge of the map image. The x-coordinate of the first point which is located on the x-axis of the image represents the lateral distance between the considered frames. Due to the perspective of the camera lens, defined by the position of the camera at the center of the robot, the cluster of the center row can be detected more reliable than the position of the outer rows. Therefore, only the center rows position is used for navigation. In case that the center row cannot be detected, the average line of both outer rows' center line is used to generate steering commands. If all three rows are detected, both calculated center lines are compared. As all rows are anticipated to be equally spaced and to be parallel to each other the calculated center lines should be identical. If significant divergences arise, the computed center line will be neglected for the steering commands as it probably originates from measurement mistakes, noise or false detection. The frequency of neglected measurements is tracked for safety to detect potential malfunction. If the detection has failed several times consecutively, the robot needs to be stopped before drifting out of the lane. The approach is based on a data association strategy similar as it was presented by Winterhalter et. al [WinterhalterEtAl21]. The authors project their detected row pattern into a previously defined map containing the entire crop row structure. Generally, data association aims at finding correspondences between uncertain measurements and known structures and features to assign the measurements to the known parts. In this project, each detected row is also associated with one of the three main guiding crop rows. The rows, however, are extracted separately and are not detected as a pattern with constant geometrical features such as the inter-row distance.

3.2.5 Steering Commands

Similarly to the previously used navigation approach presented in Section 2.4, the lateral offset between the robot's frame and the center row is used as the steering command. It is obtained by computing the difference between the robot's x-position and the center row's x-position 2 m in front of the robot. The offset value is mapped to a value between -1 and 1. If the distance is larger or smaller than -1 m the value is 1 or -1 respectively. This value is then sent to the motor controller which converts it into actuation commands for the two motors.

Chapter 4

Experimental Evaluation

For the development of the algorithm recordings of different strawberry fields taken by the ZED2i camera were used to do first improvements and tests. The recordings were taken while the robot was navigating through the strawberry fields during harvesting season either remotely controlled or by steering autonomously. In order to achieve a navigation algorithm that enables robust steering, recordings from fields at different stages throughout the strawberry season were used. Robustness is here stated as the capability of the algorithm to reliably produce crop row pattern that are close to the actual environment, despite of challenging environmental conditions such as weed, varying lighting conditions and gaps. This is crucial for autonomous row following and is this project's main objective. The algorithms of the different methods were applied not only on recordings from different strawberry fields but also during real-world experiments to verify and improve the approaches and for troubleshooting. To evaluate the localization approach, final experiments were performed on a strawberry field with mounds. The experiments took place after harvesting season on a field that was already planted for the following season and kindly provided by a regional farmer in Winsen (Luhe). Nonetheless, as can be seen in Figure 4.1, the plants were nearly fully grown and were hence suitable to replicate a field during harvesting season. The inter-row space was only covered by small weeds and vegetation gaps of one missing plant at the most were included in the experiment. Even though varying appearances due to different growth stages were considered, the navigation will be applied for harvesting only and does not need to perform well for early season.

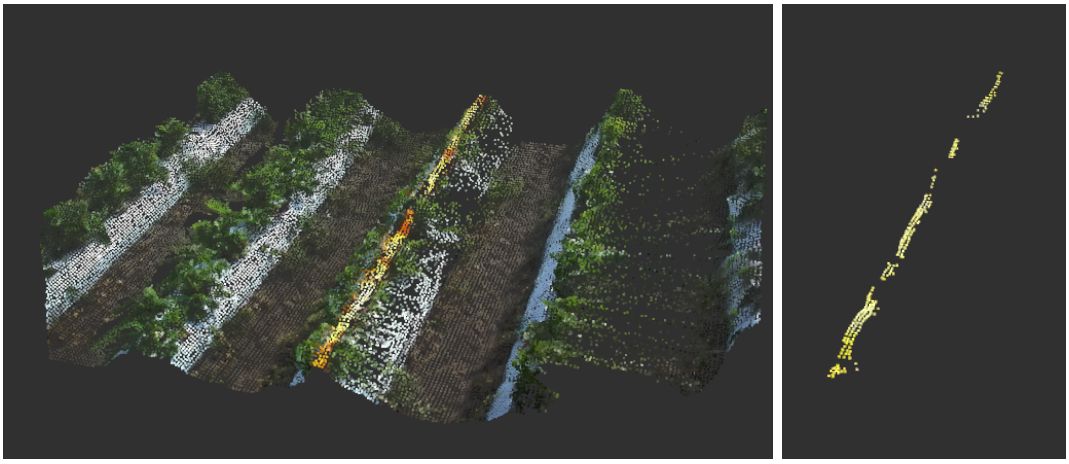
(a) (b)

Figure 4.1: (a) The field taken for the experiments has small weeds in the inter-row space and some gaps due to missing plants. The mounds are newly formed and guarantee a neat structure. (b) A lashing strap is placed along the center line on top of the crop row to be used as a ground truth reference.

Each time that the test environment is changed the parameters for the detection algorithm need to be tuned. This includes the z-height of the points that is extracted for the row detection and the rotation parameters of the point cloud such as the pitch angle which can change due to inaccurate orientation of the camera introduced while mounting. These parameters are adjusted to ensure that the camera frame's x-y-plane and the ground plane are parallel to each other. In the currently used navigation method that was explained in Section 2.4 the plane alignment is already implemented. The approach will here be referred to as reference navigation. When using the ZED2i camera, the point cloud also needs to be shifted 6 cm to the right as it is represented in the left camera frame. As describe in Section 2.3, the camera frame and the robot frame are assumed to be identical. All adapted parameters are set constant throughout the entire experiment such that testing the different approaches is not affected.

4.1 Evaluation Method

To get a direct comparison of the approaches a fixed section of the field with a track length of approximately 12 m was selected to autonomously run the robot using each of the algorithms one after the other. Additionally, the same section was run using the input data of the Intel RealSense camera to check if a low-cost stereo camera can achieve similar results. For this run the contour clustering and the RANSAC line detection methods were used. The reference navigation approach was also applied on the same experimental track to compare the steering commands for further comparison.



(a) (b)

Figure 4.2: (a) The point cloud generated by the ZED2i camera contains color information. (b) All points belonging to the orange lashing strap are extracted by only selecting points of its color.

As it can be seen in Figure 4.1(b), a bright orange lashing is used as a reference for the position of the center row. During the experiments, the strap with a length of 12 m and a width of 5 cm was placed along the center line on top of the middle row. The point cloud generated by the ZED2i camera is displayed in Figure 4.2(a). As the RGB values of the color of the strap significantly vary from the environment's colors, the points belonging to the strap in the recorded point cloud can be extracted as it is shown in Figure 4.2(b). The points are then projected on a black and white image which is pre-processed using morphological filters to get rid of wrongly extracted points. The RANSAC line fit approach is used to fit a line through the extracted pixels to attain a ground truth reference. The strap doesn't affect the crop row detection, since the approach doesn't take into account the color of the points. In addition, the strap was pushed down

such that it was always placed lower than the outer leaves of the plants.

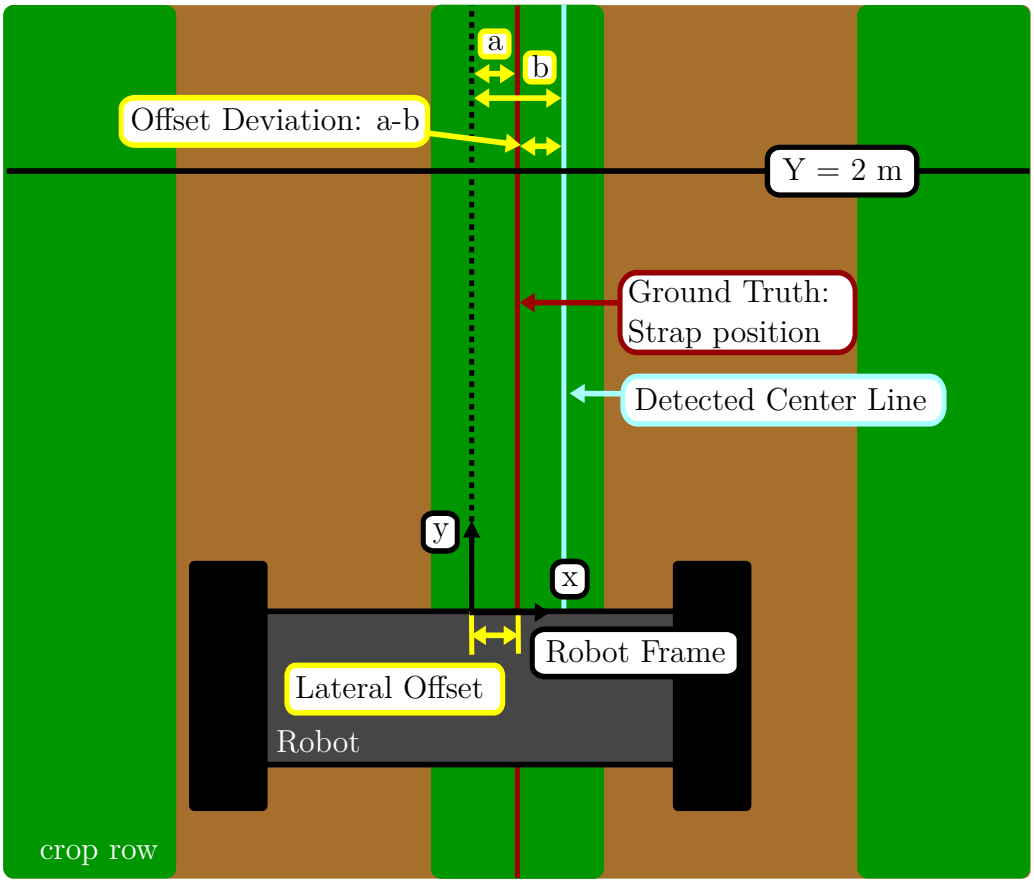


Figure 4.3: The performance of the tested approach is evaluated with respect to the offset deviation and the lateral offset. The lateral offset is the lateral distance between the center row and the robot frame as it is investigated in Section 4.3.1. The offset deviation is covered in Section 4.3.2 and is defined as the difference between the detected and the actual crop row position measured 2 m in front of the robot.

The lateral offset, illustrated in Figure 4.3, of the robot with respect to the row structure is used to evaluate the methods. It is desired to be as small as possible and should not exceed the maximum possible value of 26 cm to be counted as a valid run. The sketch is only given to illustrate the experimental setup and the evaluation measurements. It is only schematic and not true to scale.
The robustness of the methods is evaluated by the deviation of the lateral distance between the center row 2 m in front of the robot and the y-axis of the robot's frame. The deviation is obtained by subtracting the detected x-position of the center line from the ground truth x-position given by the lashing strap at

$y = 2\,\mathrm{m}$.
The number of frames that could not be used to successfully detect a row pattern or fit lines through the detected clusters is counted as an additional evaluation parameter for performance monitoring.
The computation time for the clustering methods was recorded while the experiment. It is a limiting factor for the speed of detection and thereby determines the maximum speed of the robot for navigation. Reducing computational time is not a main target of this thesis as the driving speed is not of great importance for a harvest transportation platform as it only operates at low speed while following harvest workers. It also needs to stay at a moderate speed when navigating to the drop-off point for safety reasons. Requiring high computational resources, however, also mean higher costs of the hardware components.

4.2 Limitations of the Evaluation Method

There are four main limiting factors that need to be considered when viewing the findings of the experiment. First, the accuracy of the ground truth measurements is limited by the precision of the used measurement equipment. Figure 4.4 shows the lateral distance between the robot and the strap which, as explained above, is localized by means of color information while standing still. Although the distance is expected to stay constant, the detection results in a peak-to-peak amplitude of 1 cm. This amplitude arises from measurement noise from the ZED2i camera.
The camera is also influenced by environmental influences such as varying lighting conditions and view angles that lead to changing RGB values of the points belonging to the strap.
Second, the placement and geometry of the lashing strap limit the reliable measurement of the lashing strap. Even if the strap is perfectly placed and fully detected, as it has a width of 5 cm multiple lines that fit through the extracted points can be found at different x-positions and varying slopes. While it was aimed to place the strap at the center of the middle row, the placement can only achieve a limited accuracy. The uneven surface formed by differently shaped plants prevent the strap being perfectly straight. In addition, leaves covering parts of the strap and shadows lead to parts of the strap not being fully detectable.
Third, the system is evaluated mainly using one test field. Varying appearances of the row structures and plant shapes on different fields or at various points in time were not considered.
Finally, the resolution of the grid map generated from projecting the extracted strap point cloud limits the accuracy of the localization of the strap. Here, a pixel size of 1 cm was chosen.

As the here investigated transportation platform doesn't require a highly precise navigation the used evaluation method is sufficient for evaluating in-field navigation. Nonetheless, further experiments using more precise evaluation methods are meaningful in the future to get more accurate evaluation results to further improve the system and especially with regards to the future implementation of tasks that demand higher precision.

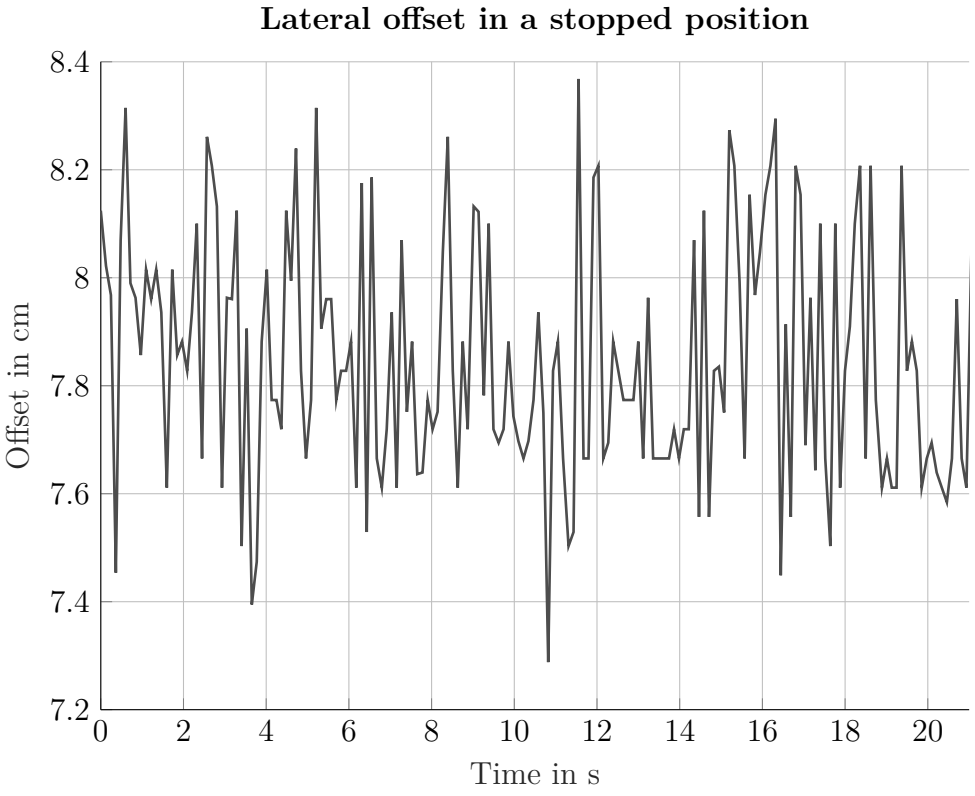


Figure 4.4: Lateral offset of the robot to the center row at $Y = 0\,\mathrm{m}$ while standing still.

4.3 Detection Results

The steering commands that are required for the autonomous navigation can only be generated when the center row has been localized with respect to the robot's frame. The row detection is dependent on multiple factors such as the clustering and the line detection. The center row can only be defined when at least either both outer rows or the center row have been successfully detected.
All tested methods allowed the robot to follow the chosen track along the lashing strap in about 75 s. Considering the length of the lashing strap of 12 m, the robot reached a speed of around $0.16\,\frac{\mathrm{m}}{\mathrm{s}}$. While applying the window clustering method

all three clusters and all lines could constantly be detected. No detection was identified as a failure when the window clustering was applied in the chosen test environment. When using the contour clustering method, the steering command could not always be correctly generated. The algorithm detects if the offset exceeds a maximum constant lateral distance of 30 cm to decide whether the detected value is valid. If this threshold is exceeded the steering command will be neglected. Given that a detection failure was only detected when applying the contour clustering method, it is confirmed that the determination of the lateral offset only depends on the cluster detection and is independent of the line method. As can be seen in Table 4.1, frames were neglected in all line methods when applying contour clustering. For one test run approximately 1100 frames were generated by the camera resulting in an average number of 1.2 neglected frames per 1000 processed frame.

Table 4.1: The number of detected center rows that resulted in a lateral offset exceeding the maximum expected threshold only occurred while using contour clustering. If this occurs the result is neglected.

Hough	RANSAC	Least Square	Rectangle
3	5	1	4

The reliable clustering output provided by the window clustering permitted a constantly successful line detection resulting in a failure rate of 0 neglected frames per 1000 processed frame. The cluster and line detection results for all line methods for the contour method is listed in Table 4.2. Large vegetation gaps and weeds are assumed to be the most common reason for non-detectable rows. Row clusters can get merged with unrelated clusters belonging to weed or other rows. If the total width is still within the accepted row width, the center line of the detected cluster can lie outside the expected location and will then cause a detection failure. As the experimental environment did not include large vegetation gaps, the false cluster or line detection are very likely to be caused by weed, measurement errors and disturbances causing the robot to jerk.

Table 4.2: Number of failed cluster and line detection that were detected for the contour clustering.

Object of Detection	Hough	RANSAC	Least Square	Rectangle
Center Cluster	3	0	1	2
Left Cluster	0	3	4	0
Right Cluster	2	3	4	0
Center Line	3	0	1	2
Left Line	0	3	4	0
Right Line	0	1	2	0

4.3.1 Lateral Offset

In the field the robot moves along the rows at a constant speed while constantly striving to minimize the lateral distance between its center and the middle crop row. To successfully navigate along the track the lateral offset should not exceed a maximum threshold to avoid plant damaging and to achieve efficient steering which results in moving along straight tracks and avoiding wavy lines. This can be examined by looking at the lateral offset between robot and center row measured at the origin of the robot frame.

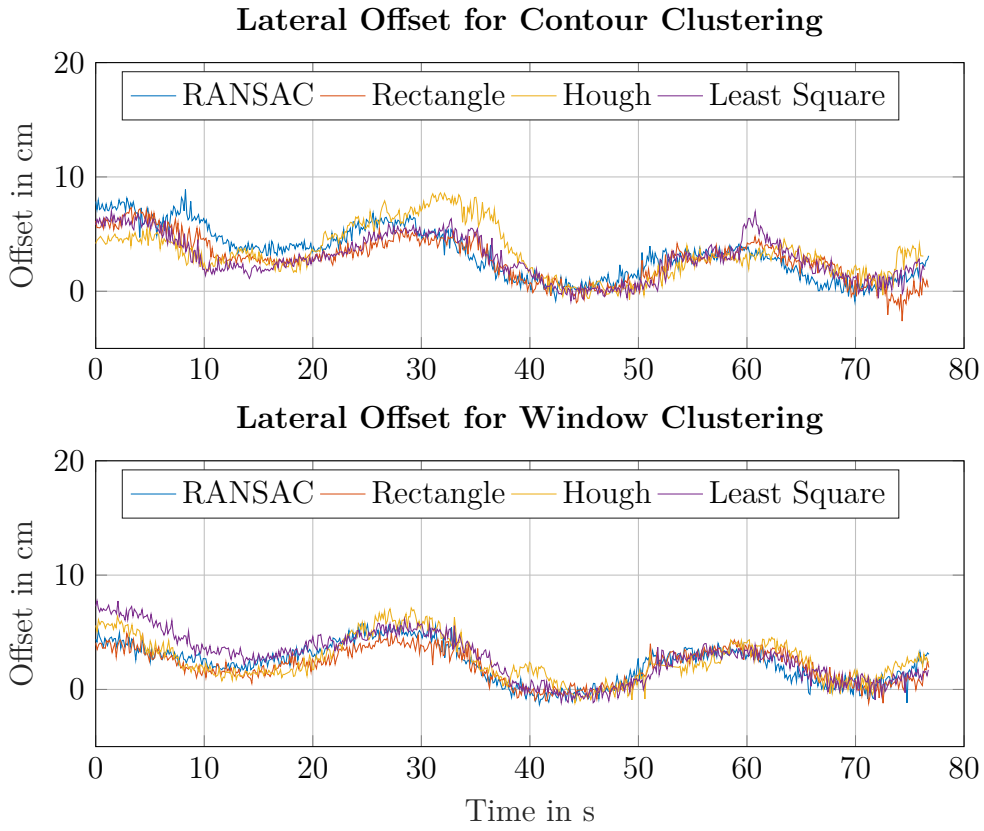


Figure 4.5: Lateral offset of the robot to the center row while navigating along the track using all approaches.

The results depicted in Figure 4.5 show a continuous oscillation and noise in the signal. The noisy signal arises due to noisy camera measurements that were already mentioned in Section 4.2. The oscillation is related to the motors responding to the steering commands. The peak-to-peak amplitude for all line detection methods is smaller than 10 cm which results in smooth navigation.
A maximum deviation of 9 cm was reached which stays within the maximally

allowed bound of 26 cm. All large deviation exceeding the given threshold are likely to be caused by sudden disturbances in the environment such as small bumps or slipping. For all runs, the offset caused by the disturbances didn't cause failure or could be identified as false detection and was rejected. By neglecting all outliers which, as mentioned before, was necessary when applying contour clustering, shows that all methods resulted in the same performance regarding the lateral offset. While the lateral distance stays in an acceptable range the mean value is 4 cm. The chosen experimental track needs to be chosen longer to investigate if the mean value of the lateral offset converges to 0. If that is not the case, the reason for the remaining constant offset error needs to be investigated as it is not desired.

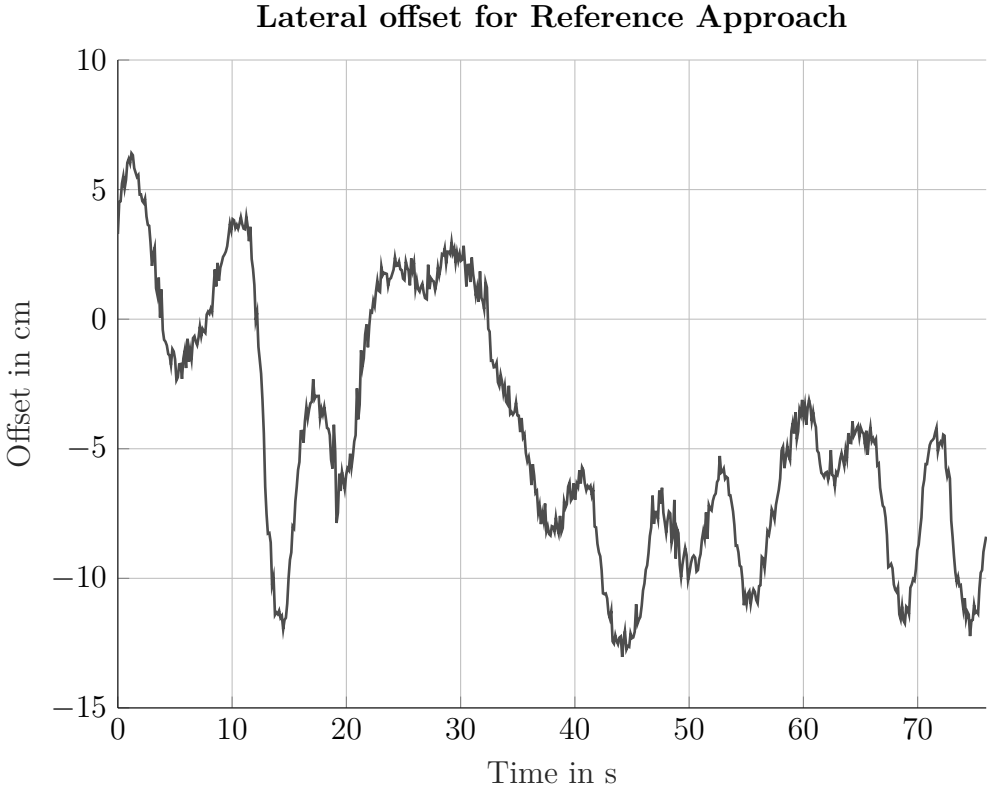


Figure 4.6: The lateral offset of the robot to the center row while navigating along the track using the reference approach.

The results for the lateral offset of the reference navigation approach are depicted in Figure 4.6. At time stamp 43 s the offset reaches an absolute peak value of 13 cm. In terms of efficient steering, the reference approach performs worse as the track reaches a peak-to-peak amplitude of 20 cm and the oscillation frequency is higher than before. This can be explained by the higher processing rate.

During the experiment the ZED2i camera provided an average publishing rate of nearly 10 point clouds per second. The here presented approach only uses every third point cloud to generate steering commands to reduce the processing time of the navigation system. This is a reasonable approach for systems only operating at low speeds. As a result, more than three steering commands are published per second if the row detection succeeds for all selected frames. The reference approach, on the contrary, uses all available point clouds and therefore achieves a three times faster response time.

4.3.2 Robustness and General Performance

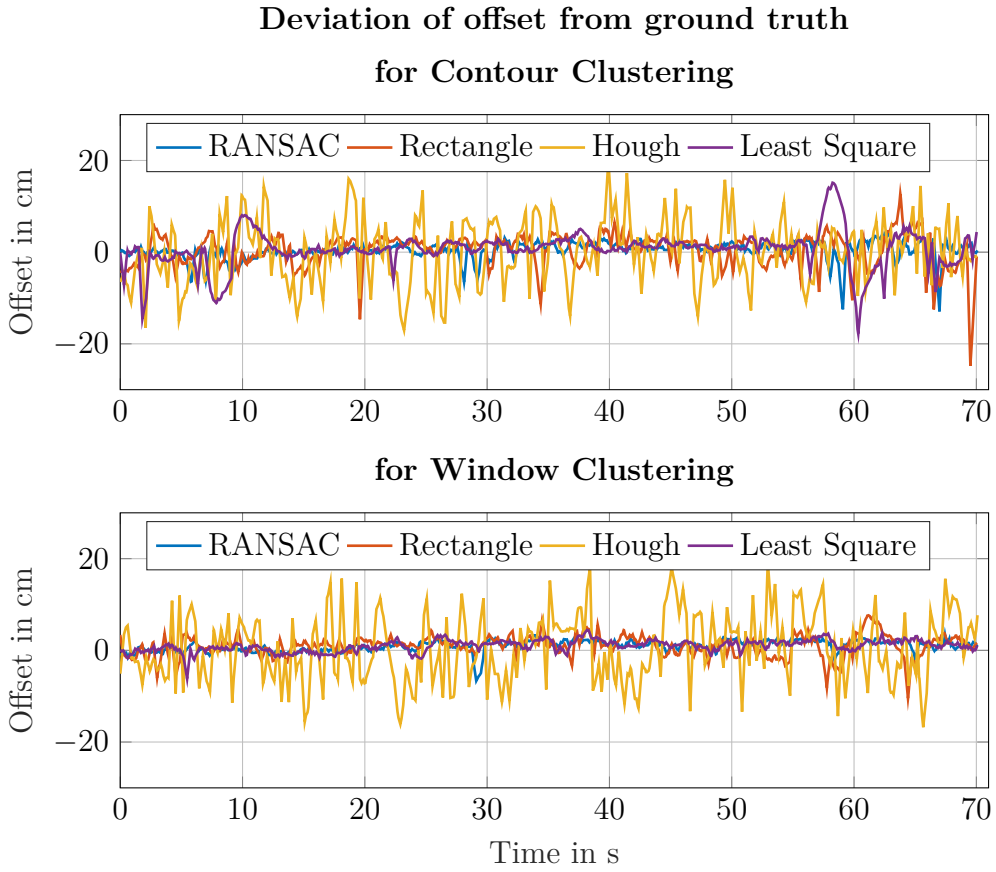


Figure 4.7: The deviation from the ground truth data can be used as a measurement for accuracy.

Figure 4.7 shows the deviation of the lateral distance measured 2 m in front of the robot from the ground truth data. This distance which, as previously said, is illustrated in Figure 4.3 is especially interesting for the evaluation of the approach as it is used to determine the value of the steering command. A constant

offset could not be detected which shows that the constant offset error mentioned before doesn't arise from a systematic error in the system but comes from the method itself. Looking at the deviation graph shows that the RANSAC, Rectangle and Least Square line detection methods resulted in only small deviation of less than $/pm10$ cm. Only the Hough method shows significantly higher and more frequently appearing deviations from the ground truth value. As the graph does not provide a sufficient basis for a comparison analysis of all methods, the error distribution of all measurements can be seen in Figure 4.8. It becomes evident that all methods apart from the Hough line detection method result in a Gaussian distribution. This can be observed for both clustering methods. The standard deviation

$$\sigma = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^{n} (x_i - \overline{x})} \tag{4.1}$$

is used to determine the dispersion of the estimated position of the robot with respect to the center row which also includes the method's precision. A small value is desired as it indicates that the value tends to be close to the mean value $\overline{x}$ of all estimated values. Here, the standard deviation with respect to the ground truth value was calculated for all line methods apart from the Hough method since its error distribution doesn't result in a Gaussian distribution. The results can be seen in Table 4.3.

Table 4.3: The standard deviation shows how much the estimated value is expected to deviate from the mean value.

Clustering Method	RANSAC	Least Square	Rectangle
Contour Clustering	1.99 cm	3.95 cm	3.67 cm
Window Clustering	1.49 cm	1.54 cm	2.4 cm

The RANSAC method achieves the best performance especially when combined with the window clustering method. It reached a standard deviation of 1.49 cm. While the Least Square Method achieved a significantly higher standard deviation of 3.95 cm for contour clustering, it attained a standard deviation of just 1.54 cm with window clustering. Thus, it can be concluded that the RANSAC and Least Square method performed equally well when applied with window clustering. The rectangle method achieved a higher standard deviation of 3.67 cm and 2.4 cm for contour and window clustering respectively. Once again, a better performance for window clustering was detected. As it was mentioned before, the result can be strongly impacted by outliers as they occurred for the Least Square and Rectangle method. It is assumed that they arise out of wrong cluster detection and association due to weeds. Even though the standard deviation for the Hough method wasn't calculated, it can still be seen from Figure 4.8 that its estimation are the most dispersed around the expected value.

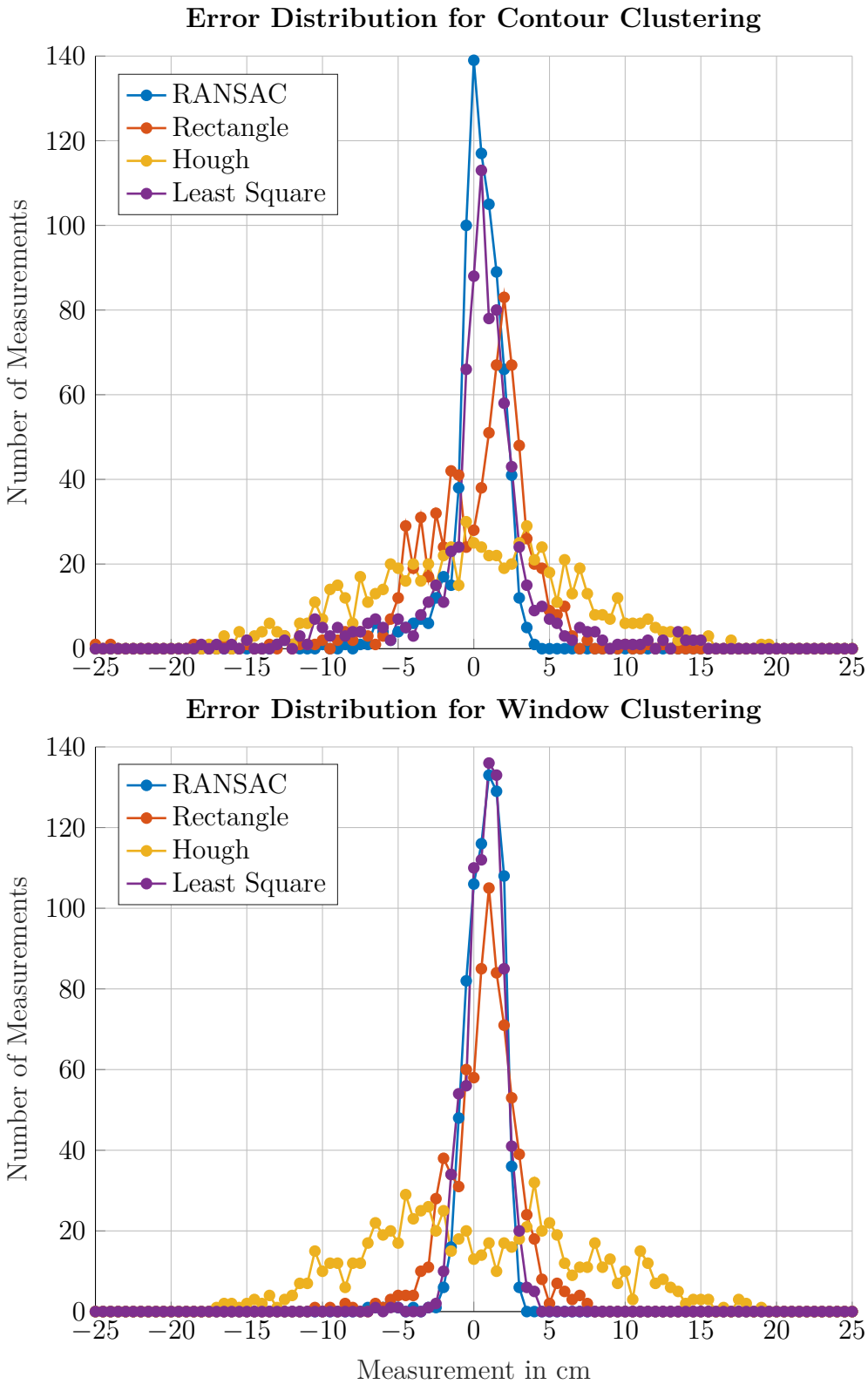


Figure 4.8: The distribution of the error is used to investigate the accuracy and precision of the approach. It can be used to calculate the standard deviation for each approach to get the dispersion of the detection result.

4.3.3 Computational Time

Comparing the clustering methods regarding computational time shows that the window clustering method is more than four times more time consuming than the clustering method using contours. The computationa time was measured using the given hardware described in Chapter 2.3 that was used for the field experiments. The average duration for the clustering step for one frame is 7.1 ms using contour clustering and 40.5 ms using window clustering. The computational time of the window clustering method could potentially be reduced by changing the size of the windows. For the experiments, 6 windows were used to gain a highly accurate result. While the computational time was sufficient to successfully navigate through the field, reducing the computational time is needed to compete with the other clustering method and especially when considering the development of an overall more complex system in the future that is capable of further technical tasks consuming more computational resource. When playing back the same ROS bagfile data using less numbers of windows such as 5, 4 and 3 windows per row or reducing the window's width didn't lead to a reduction of time consumption.

Table 4.4: Average duration for all tested line detection methods using recorded data from the field experiment.

Hough	RANSAC	Least Square	Rectangle
4.5 ms	27.2 ms	0.63 ms	0.88 ms

The duration for all line methods was evaluated when playing back data to get a direct comparison using the same input data. The total duration values that are listed in Table 4.4 might vary from the duration when running the experiment on the field. The proportion between the measured average duration will still be valid for the real-world application. As can be seen in Table 4.4, the line detection method using the RANSAC algorithm took the longest with a duration of 27.2 ms. The Hough line detection method only needs a sixth of this duration. The best results in terms of computational time could be achieved using the Least Square and Rectangle line detection as they required significantly less time.The Rectangle method only required 0.88 ms which is a fifth of the time required for the Hough computations. The Least Square method could achieve the best results with the lowest computational times of 0.63 ms resulting in more than a 25 % decrease compared to the Rectangle method. Downsampling the data could help to reduce the computational time for Hough and RANSAC line detection.

4.3.4 Clustering Comparison

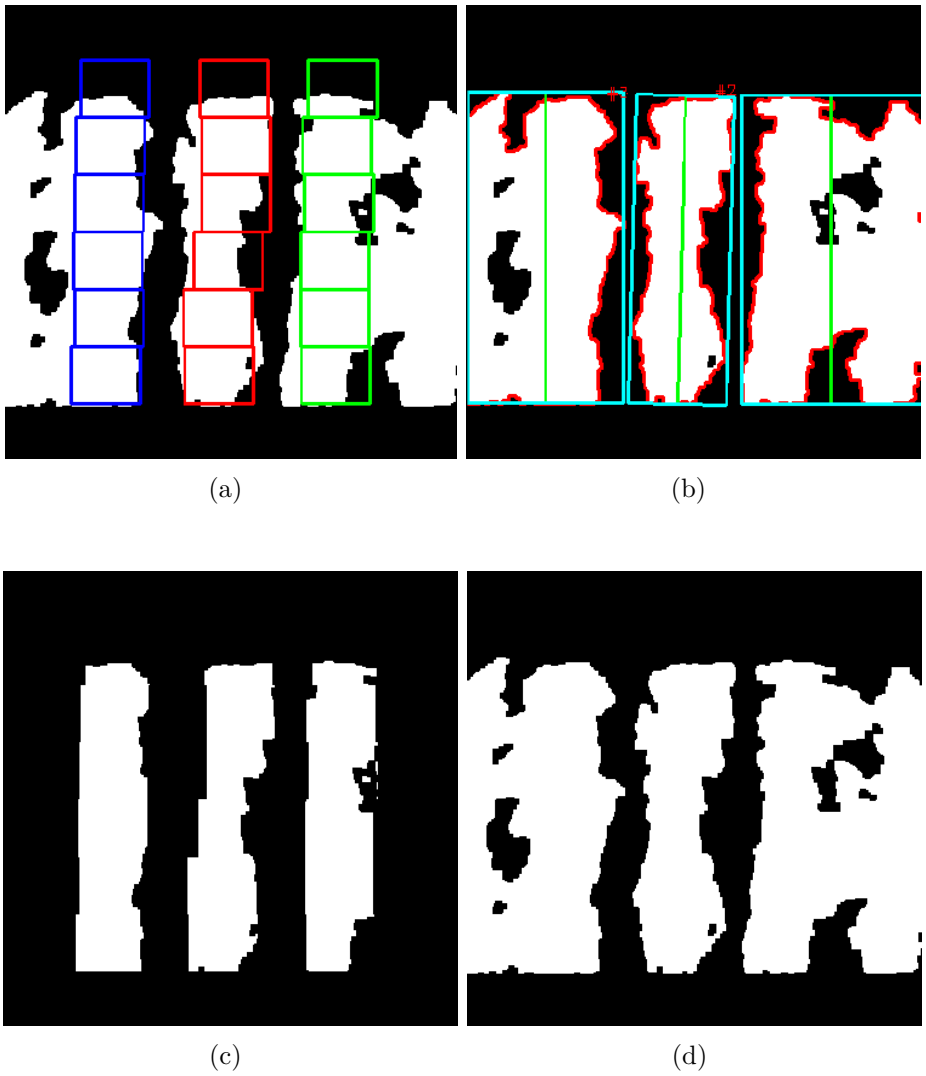
The real-world experiment shows that all approaches perform well on the chosen field. Nonetheless, testing the algorithms on different and more challenging data including higher weeds and larger gaps emphasize that the robustness of the algorithm still needs to be improved.

(a) (b)

Figure 4.9: (a) The top of the fully grown plants are mowed after harvesting season. The orange lashing strap is placed at the center of the middle row as a reference for the detected center line. (b) Differently shaped and sized plants, gaps, weed and wrongly positioned strawberry plants make the row detection more difficult and are recorded to test the approach's performance in challenging environments.

Such challenging experimental environment can be seen in the images of Figure 4.9. The data was recorded from a strawberry field after harvesting season. It contains overgrown inter-row tracks due to strawberry shoots and weeds, large gaps in the rows and damaged plants. While the large, bushy plants are characteristic for harvest time the structure of the field can be expected to be better visible during harvest season. As the plants were mowed and pushed

down it was challenging to extract the rows separately when using the contour clustering. Nevertheless, the row structure is still apparent and visible in the recorded point cloud data due to the mounds.



(a) (b)

(c) (d)

Figure 4.10: (a) The windows of the window clustering method have a pre-defined size and lateral offset. (c) Therefore, overgrown tracks are fremoved from the white pixels presenting the three guiding crop rows. (b) The contour clustering defines all white pixel clusters enclosed by one contour line (red) as belonging to the same cluster. If plant material or similar objects occur in the inter-row space the algorithm tends to merge clusters of multiple rows as it occured in (d) for both outer rows.

Challenging frames with large weeds in between the rows as it is depicted in Figure 4.10 were chosen to compare the clustering methods. While the contour clustering method associates all clusters that are enclosed by one contour to the same crop row, the window clustering method is more reliable in fields strongly overgrown with weeds. In Figure 4.10(a) the window clustering places the search window along the center of the rows leading to cropping off the plant material at the sides of the rows and in the inter-row space to obtain a cleaner cluster detection of the row structure. The contour clustering method depicted in Figure 4.10(b) fails in detecting the outer rows correctly. The rows get merged with their neighboring rows and weeds located further outwards. The result of the contour detection in Figure 4.10(b) shows that the outer rows were wrongly identified. In the here presented case, the steering commands can still be generated from the successful detection of the center crop row. If the center crop row gets merged with additional plant material or even further crop rows, the detection will fail. The Large gap didn't affect the navigation approach as the consistent mound structure always delivered a sufficient height for row detection. Due to time constraints, the clustering methods could not be compared quantitatively.

4.3.5 Intel RealSense Camera

The code was developed using the ZED2i camera as it provides a reliable output even in challenging outdoor environments. This makes it possible to more focus on the methods without being restricted by insufficient quality of sensor data. In the same experimental environment, the algorithm was run on the Intel RealSense Depth camera that also uses stereo vision to obtain depth information. This is done, as previously said, to also test less expensive sensors and to check if similar results can be attained. The recordings were taken on the same experimental track using the RANSAC line method and contour clustering for autonomous steering. The detailed evaluation for all methods using the Intel RealSense camera exceeds the scope of this project. Nonetheless, first results were collected to check if further testing is expected to pay off. The failure rate is $0,56$ neglected frames per 1000 processed frame which are less failures compared to when using the ZED2i camera.
Furthermore, Figure 4.11 shows the lateral offset between the robot and the center row. The run resulted in a peak-to-peak amplitude of 0.21 cm. This indicates a smoother steering as it was detected when using the ZED2i camera which was discussed in Section 4.3.1.

As the results are based on only one test run using the Intel RealSense camera, the findings cannot be used for a meaningful evaluation. In spite of this, the

promising results generally indicate the potential of the application of low-cost sensors.

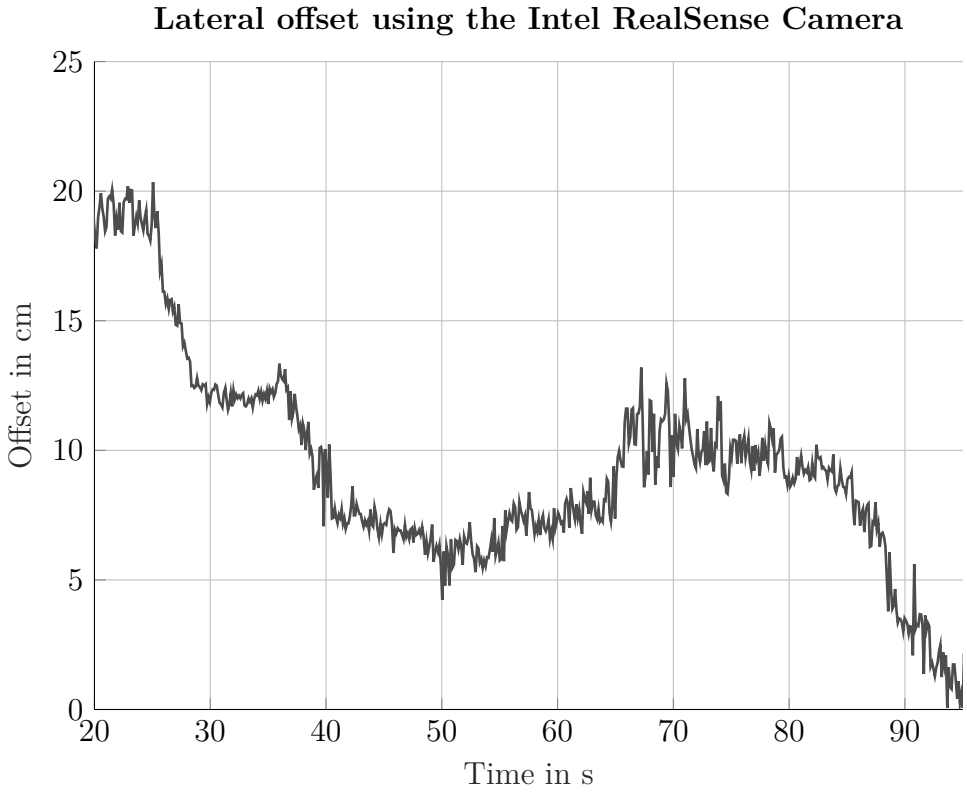


Figure 4.11: Lateral offset of the robot to the center row while navigating along the track using the Intel RealSense camera.

Chapter 5

Conclusion

In this paper, a novel localization approach for autonomous navigation in strawberry fields was presented. The aim was to establish a robust and low-cost mobile transportation platform that autonomously navigates along the crop rows to carry filled boxes of harvested strawberries. A mapping approach was chosen as it is well suited for further development enabling future tasks. It can easily be modified in order to build a more detailed representation of the immediate environment by inserting additional features of the surroundings. The map is generated using spatial information contained in point cloud data which is given by the ZED2i stereo vision camera as this is the most promising sensor data for crop row detection.
It was demonstrated experimentally that all evaluated approaches are suitable for autonomous in-field navigation. In all conducted experiments, the lateral distance between robot and center crop row was less than 10 cm indicating a smooth movement and an accurate navigation. While the reference navigation resulted in a curvier navigation path, the novel approach could achieve a much smoother motion along an almost straight path. The weaker performance of the reference method is due to a higher frequency of processed frames resulting from a lower processing time. A smoother way of driving could be achieved by applying a noise filter which reduces the sensitivity of the approach.
In addition, the accuracy and dispersion of all methods was investigated using the standard deviation of the estimated robot's position. The results could confirm that window clustering is the preferred clustering approach. A small standard deviation for the window clustering could be especially achieved by the RANSAC and Least Square method. Only the estimated values of the Hough method didn't result in a Gaussian distribution. In comparison, the Hough method reached the highest dispersion of detection results. A distinct accumulation of measurement results at the mean value could not be detected and, moreover, it had the most frequently occurring large deviations exceeding 10 cm.

When diving deeper into the experimental results, it becomes apparent that the window clustering method achieves significantly more robust crop row detection. This could especially be confirmed for fields with a large part of the inter-row space covered by vegetation. While the contour clustering allowed a successful navigation along the test track without interruptions, the detection of a few frames failed due to vegetation appearing between rows. The window clustering clearly achieves a more robust result as its cluster detection never failed but comes at a considerably higher computational cost that is nearly six times larger than the one measured for contour clustering. All line detection methods, except the Hough method, didn't result in significant performance differences. Nonetheless, comparing them with respect to their required computational time led to clear differences. The Rectangle and Least Square line detection methods required by far the shortest computational time making them more favorable.
Assuming that the source code for the window clustering method can be further improve with regards to computational time and considering all examined evaluation criteria, it can be concluded that window clustering is the preferred method. Combining this clustering approach with the fastest line detection based on the Least Square method results in a robust navigation approach that enables the robot to successfully navigate through a strawberry field containing fully grown plants placed on straight mound structure with equidistant rows. The presented approach could even be used with lower-cost sensors such as the Intel RealSense Depth camera replacing the currently used ZED2i stereo camera aiming for an overall inexpensive transportation platform. Further testing, however, is required to achieve a reliable evaluation and a quantitative comparison between using the different cameras to build a lower-cost transportation platform.
This project has faced up the challenged of building a field robot only using low-cost on-board sensor independent of field maps or GPS markers. It was shown that, while the robot is only able to acquire information of its direct environment, it is still able to robustly and smoothly follow the crop rows of a strawberry fields at a low speed. Regarding all findings, this project contributes greatly to the field of mobile field robots as it developed a robust navigation system that carries a high potential for the considered application. The approach, at the same time, represents a solid foundation for the continual development as the current functional status can easily be expanded by adding new software components. Nonetheless, it is important to note the limitations of the used evaluation method that were indicated in detail in Section 4.2.

Chapter 6

Outlook

While the project resulted in a successful implementation of a navigation system, in the following discussion various ideas are proposed that can be taken into consideration for future studies aiming to integrate the navigation system in an embedded platform. First, improvement measures are discussed that can be implemented to improve the current state of development especially with respect to robustness. Following this, aspects are given that are mandatory to employ the robot in the field. Finally, ideas are suggested with the objective of reaching more advanced future development levels.

For the purpose of increasing the robustness, additional sensors, such as a back-camera, are useful as a validation of the current detection results and in case the front camera's field of view is obstructed. Further real-world experiments are needed including varying environmental conditions, field structures throughout harvesting season and fields at different locations. Small changes can be done such as trying out different ROI dimensions, voxel sizes and various map resolutions. The currently selected point that is used to generate steering commands, located 2 m in front of the robot, can be moved to another position to test its impact on the performance. Additional features such as the angular deviation between the field frame and automated frame alignment can be introduced aiming for a safer and more robust performance enabling the work alongside humans.
The tests were only conducted on planar fields with constant transformation parameters of the point cloud. IMU data or planar segmentation can be used in the future to automatically adjust the transformation, especially the rotation parameters of the point cloud to always correctly align the camera and ground plane. During the development of the here presented approach a plane detection method was applied using a simple plane model segmentation to detect the ground plane and for dynamic frame alignment. The Kalman filter was used to stabilize the measurements. Detecting the normal vector of the field's ground

plane did not work sufficiently well, because the point cloud data included too much noise in form of vegetation covering the ground and unevenness. The measured part of the field didn't contain enough information to repetitively generate a reliable plane estimation. Achieving a successful plane alignment would however be highly beneficial to compensate for unevenness and to allow the navigation in hilly fields.
Generally, the approach was developed mainly focusing on the robustness of the navigation system while the computational time consumption had lower priority. Measurements to reduce the computational time, however, need to be considered especially when applying time-intense approaches such as the window clustering method. This can be done by code optimization, ROI reduction or downsampling in the sense of reducing the number of pixels which results from the resolution of the grid map.
The evaluation method needs to be improved to reduce its limitations that were mentioned in 4.2. Using alternative sensors and finding a way to get a more precise position of the ground truth reference is key to get more accurate ground truth data. Instead of using just one strap, multiple markers can be placed in the field to get a more accurate identification of the reference row position.
The presented robotic transportation platform can only be deployed in its work environment closely collaborating with human workers if it fulfils stringent safety requirements. To achieve such a target the robustness needs to be further increased and object detection and avoidance needs to be implemented. Avoiding collisions with obstacles such as baskets and workers requires additional sensors not only in the front but also at the side and back since the robot can be loaded from all sides. As the robot is built to follow the pickers, humans, however, will always be in the current camera's field of view and would prevent a successful detection of the center row's position. This needs to be considered in building the final navigation system by combining obstacle and row detection.
Future work addressing more challenging navigation outside the field can focus on Simultaneous Localization and Mapping (SLAM) which is a method for localization of mobile robots in an unknown environment. The robot constantly generates a map, called occupancy map, of its environment and its relative position to the detected surroundings. SLAM can be used to detect obstacles but also target positions and field structures. In the considered application, it is especially useful as soon as the robot needs to leave the structured field environment when for example switching between crop rows or navigating to a drop-off point for the picked strawberries. The generated grid map in the here presented approach can serve as a basis for the advanced mapping and could be incorporated in the new occupancy map.

Bibliography

[AhmadiEtAl20] Ahmadi, A.; Nardi, L.; Chebrolu, N.; Stachniss, C.: Visual servoing-based navigation for monitoring row-crop fields. In 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 4920–4926, IEEE, 2020.

[AhmadiHalsteadMcCool] Ahmadi, A.; Halstead, M.; McCool, C.: Towards autonomous crop-agnostic visual navigation in arable fields.

[Annett ChilianHeiko HirschmÃ¼ller09] Annett Chilian; Heiko HirschmÃ¼ller: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009: IROS 2009 ; 10 - 15 [i.e. 11 - 15] Oct. 2009, St. Louis, USA. Piscataway, NJ: IEEE, 2009.

[Aske Bay Jakobsen15] Aske Bay Jakobsen: Crop row naviagtion for autonomous field robot. Master's thesis, Technical University of Denmark, April 2015.

[Blackmore04] Blackmore, B.S.; Griepentrog, H.W.; Nielsen, H.; Nørremark, M.; Resting-Jeppesen, J. (Eds.): Development of a deterministic autonomous tractor. 2004.

[ChebroluEtAl19] Chebrolu, N.; Lottes, P.; Labe, T.; Stachniss, C.: Robot localization based on aerial images for precision agriculture tasks in crop fields. In 2019 International Conference on Robotics and Automation (ICRA), pp. 1787–1793, IEEE, 2019.

[E.J. Van Henten, B.A.J. Van Tuijl, J. Hemming, V.T.J. Achten, J. Balendonck05] E.J. Van Henten, B.A.J. Van Tuijl, J. Hemming, V.T.J. Achten, J. Balendonck: Cropscout : a mini field robot for research on precision agriculture: source in: Proceedings of the 2nd Field Robot Event 2004, Wageningen, June 17 & 18, 2004. Wageningen: Wageningen University, Farm Technology Group, 2005.

[EnglishEtAl15] English, A.; Ross, P.; Ball, D.; Upcroft, B.; Corke, P.: Learning crop models for vision-based guidance of agricultural robots. In 2015

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1158–1163, IEEE, 2015.

[FountasEtAl20] Fountas, S.; Mylonas, N.; Malounas, I.; Rodias, E.; Hellmann Santos, C.; Pekkeriet, E.: Agricultural robotics for field operations. Sensors (Basel, Switzerland), Vol. 20, No. 9, 2020.

[GaiXiangTang21] Gai, J.; Xiang, L.; Tang, L.: Using a depth camera for crop row detection and mapping for under-canopy navigation of agricultural robotic vehicle. Computers and Electronics in Agriculture, Vol. 188, p. 106301, 2021.

[García-SantillánEtAl18] García-Santillán, I.; Peluffo-Ordoñez, D.; Caranqui, V.; Pusdá, M.; Garrido, F.; Granda, P.: Computer vision-based method for automatic detection of crop rows in potato fields. In Á. Rocha; T. Guarda (Eds.) Proceedings of the International Conference on Information Technology & Systems (ICITS 2018), Vol. 721 of Advances in Intelligent Systems and Computing, pp. 355–366. Cham: Springer International Publishing, 2018.

[HANAWAEtAl12] HANAWA, K.; YAMASHITA, T.; MATSUO, Y.; HAMADA, Y.: Development of a stereo vision system to assist the operation of agricultural tractors. Japan Agricultural Research Quarterly: JARQ, Vol. 46, No. 4, pp. 287–293, 2012.

[HuHuang21] Hu, Y.; Huang, H.: Extraction method for centerlines of crop row based on improved lightweight yolov4. In 2021 6th International Symposium on Computer and Information Processing Technology (ISCIPT), pp. 127–132, IEEE, 2021.

[JiangZhao10] Jiang, G.; Zhao, C.: A vision system based crop rows for agricultural mobile robot. In 2010 International Conference on Computer Application and System Modeling (ICCASM 2010), pp. V11–142–V11–145, IEEE, 2010.

[KiseZhangRovira Más05] Kise, M.; Zhang, Q.; Rovira Más, F.: A stereovision-based crop row detection method for tractor-automated guidance. Biosystems Engineering, Vol. 90, No. 4, pp. 357–367, 2005.

[NichollsGreen] Nicholls, P.; Green, R.: Detecting tramways in crops for robot navigation.

[PonnambalamEtAl20] Ponnambalam, V.R.; Bakken, M.; Moore, R.J.D.; Glenn Omholt Gjevestad, J.; Johan From, P.: Autonomous crop row guidance using adaptive multi-roi in strawberry fields. Sensors (Basel, Switzerland), Vol. 20, No. 18, 2020.

[RomeoEtAl12] Romeo, J.; Pajares, G.; Montalvo, M.; Guerrero, J.M.; Guijarro, M.; Ribeiro, A.: Crop row detection in maize fields inspired on the human visual perception. TheScientificWorldJournal, Vol. 2012, p. 484390, 2012.

[SøgaardOlsen03] Søgaard, H.T.; Olsen, H.J.: Determination of crop rows by image analysis without segmentation. Computers and Electronics in Agriculture, Vol. 38, No. 2, pp. 141–158, 2003.

[VelasquezEtAl] Velasquez, A.E.B.; Higuti, V.A.H.; Gasparino, M.V.; Sivakumar, A.N.; Becker, M.; Chowdhary, G.: Multi-sensor fusion based robust row following for compact agricultural robots.

[Wang10] Wang, F.: Guidance line detection for strawberry field in greenhouse. In 2010 Symposium on Photonics and Optoelectronics, pp. 1–4, IEEE, 2010.

[WinterhalterEtAl18] Winterhalter, W.; Fleckenstein, F.V.; Dornhege, C.; Burgard, W.: Crop row detection on tiny plants with the pattern hough transform. IEEE Robotics and Automation Letters, Vol. 3, No. 4, pp. 3394–3401, 2018.

[WinterhalterEtAl21] Winterhalter, W.; Fleckenstein, F.; Dornhege, C.; Burgard, W.: Localization for precision navigation in agricultural fields—beyond crop row following. Journal of Field Robotics, Vol. 38, No. 3, pp. 429–451, 2021.

[ZhangEtAl13] Zhang, J.; Chambers, A.; Maeta, S.; Bergerman, M.; Singh, S.: 3d perception for accurate row following: Methodology and results. In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5306–5313, IEEE, 2013.

[ZhouEtAl21] Zhou, Y.; Yang, Y.; Zhang, B.; Wen, X.; Yue, X.; Chen, L.: Autonomous detection of crop rows based on adaptive multi-roi in maize fields. International Journal of Agricultural and Biological Engineering, Vol. 14, No. 3, pp. 217–225, 2021.

Erklärung

Ich, Caroline Kühl (Student des Maschinenbaus an der Technischen Universität Hamburg, Matrikelnummer 50975), versichere, dass ich die vorliegende Projektarbeit selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe. Die Arbeit wurde in dieser oder ähnlicher Form noch keiner Prüfungskommission vorgelegt.

Unterschrift	Datum