Techniques for Localization and Mapping in Precision Agriculture

Wera Winterhalter

Technische Fakultät
Albert-Ludwigs-Universität Freiburg

UNI FREIBURG

Dissertation zur Erlangung des akademischen Grades
Doktor der Naturwissenschaften

Betreuer: Prof. Dr. Wolfram Burgard

Techniques for Localization and Mapping in Precision Agriculture

Wera Winterhalter

Dissertation zur Erlangung des akademischen Grades Doktor der Naturwissenschaften
Technische Fakultät, Albert-Ludwigs-Universität Freiburg

Dekan	Prof. Dr.-Ing. Frank Balle
Erstgutachter	Prof. Dr. Wolfram Burgard
Zweitgutachter	Prof. Dr. Cédric Pradalier
Drittgutachter	Prof. Dr. Joschka Boedecker
Tag der Disputation	11. October 2023

Zusammenfassung

Der steigende Nahrungsbedarf einer ständig wachsenden Weltbevölkerung hat über die letzten Jahrzehnte die Industrialisierung der Landwirtschaft voran getrieben. Vor allem in den letzten Jahren wurden die negativen Einflüsse der industriellen Landwirtschaft auf die Umwelt immer offensichtlicher, so dass heutzutage nachhaltigere Landwirtschaftskonzepte wie *Präzisionslandwirtschaft* und Automatisierung von landwirtschaftlichen Aufgaben und Fahrzeugen im Fokus der Forschung stehen. Automation in der Landwirtschaft steigert nicht nur die Effizienz landwirtschaftlicher Techniken im Allgemeinen, sondern ermöglicht vielmehr auch die Anwendung nachhaltiger präzisionslandwirtschaftlicher Techniken in der Praxis. Ein wichtiger Teil vollständiger Automation ist die Fähigkeit des landwirtschaftlichen Fahrzeugs, verlässlich über ein Feld zu fahren, ohne die wertvollen Kulturpflanzen zu beschädigen, was wiederum eine genaue Schätzung der Pose des Fahrzeugs relativ zu den Kulturpflanzen erfordert.

Die Umgebung eines landwirtschaftlichen Feldes stellt traditionelle Lokalisierungs- und Kartierungstechniken vor einige neue Herausforderungen: Erstens, gibt es eine große Anzahl verschiedener Arten von Kulturpflanzen, deren Aussehen stark variieren kann. Diese müssen zuverlässig in lokalen Sensordaten erkannt und von wild wachsender Vegetation unterschieden werden. Zweitens, um Informationen von mehreren Sensoren verwenden zu können, müssen die erkannten Kulturpflanzen mit zuvor erkannten oder bereits kartierten Kulturpflanzen assoziiert werden. Datenassoziation basierend auf Pflanzendetektionen ist anspruchsvoll, da Pflanzen der gleichen Art schwierig zu unterscheiden sind und auf dem Feld in regelmäßigen Abständen gesät werden, was zu vielen uneindeutigen Situationen führt. Drittens, während des Wendens am Ende des Feldes sind die Kulturpflanzen nicht immer in den Sensordaten sichtbar, so dass sich Fehler in der Posenschätzung akkumulieren können. Eine weitere Herausforderung ist daher die Relokalisierung des Fahrzeugs, bevor es wieder in das Feld hineinfährt. Viertens, ist der Abstand zwischen den Reifen einer großen landwirtschaftlichen Maschine und benachbarten Pflanzenreihen sehr klein. Daher sind eine hoch genaue Ausrichtungs- und seitliche Positionsschätzung essentiell um eine präzise autonome Navigation zu gewährleisten, die die Kulturpflanzen nicht beschädigt.

In dieser Dissertation präsentieren wir neue Techniken, die diese Herausforderungen angehen, um genaue Lokalisierung und Kartierung für autonome Navigation auf landwirtschaftlichen Feldern zu ermöglichen. Erstens, stellen wir zwei neue Reihendetektionsmethoden vor, die darauf basieren, dass die Pflanzenreihen auf dem Feld als Menge von parallelen und äquidistanten Linien repräsentiert werden können. Diese Repräsentation ermöglicht die gleichzeitige Detektion aller sichtbaren Pflanzenreihen, wodurch unsere Methoden mit wild wachsender Vegetation und anderen schwierigen Situationen, in denen die Reihenstruktur nicht vollständig sichtbar ist, besser umgehen können. Zweitens,

präsentieren wir mehrere Techniken, die Posenschätzung aufgrund verschiedener Sensormodalitäten, einschließlich GPS und Reihendetektionen in lokalen Bild- und Tiefendaten, ermöglichen. Diese umfassen eine Datenassoziation, die die geometrischen Beziehungen zwischen Pflanzenreihen nutzt, eine Methode, die das Ende des Feldes erkennt um die Genauigkeit der Posenschätzung zu erhöhen, und die Formulierung eines Sensormodells, das die Informationen verschiedener Sensormodalitäten passend für widerspruchsfreie Posenschätzungen kombiniert. Drittens, zur Relokalisierung nach dem Wenden und um die Genauigkeit der Posenschätzung weiter zu verbessern, nutzen wir einzelne Pflanzenpositionen anstatt gesamten Pflanzenreihen als Landmarken. Um Posenschätzung anhand einzelner Pflanzenreihen zu ermöglichen, präsentieren wir eine neue Datenassoziationstechnik, die leichte Unregelmäßigkeiten in der Verteilung der Pflanzen entlang den Reihen ausnutzt und dadurch sogar in mehrdeutigen Situationen die richtige Assoziation findet.

Wir werten unsere Techniken in umfangreichen Experimenten auf Daten aus, die an drei unterschiedlichen Orten aufgenommen wurden und sieben verschiedene Kulturpflanzenarten in variierenden Wachstumsstadien zeigen. Unsere Ergebnisse bestätigen, dass unsere Methoden den Stand der Technik voranbringen, indem sie eine genaue Bestimmung der Pose relativ zu den Kulturpflanzen auf dem gesamten Feld ermöglichen. Des Weiteren wurden die vorgestellten Techniken in mehreren erfolgreichen autonomen Fahrten über das gesamte Feld genutzt, was die Eignung unserer Techniken für die praktische Anwendung auf landwirtschaftlichen Feldern weiter bestätigt.

Die vorgestellten Techniken ermöglichen genaue Lokalisierung und Kartierung für verlässliche und präzise autonome Navigation auf dem gesamten Feld und bringen dadurch Automation in der Landwirtschaft und nachhaltige Präzisionslandwirtschaft voran.

Abstract

With a growing world population the rising demand for food has been pushing the development of industrialized farming techniques over the past decades. In recent years increasing environmental awareness has spawned research into more sustainable farming concepts such as *precision agriculture* and automation of agricultural tasks and vehicles. Automation in agriculture not only has many advantages for farming techniques in general, but more importantly also opens the door to more sustainable precision farming applications. An important part of full automation is the ability of the vehicle to reliably traverse an agricultural field without damaging the valuable crops, which in turn requires an accurate pose estimate of the vehicle relative to the crops.

The agricultural environment of a crop field poses several challenges for traditional localization and mapping techniques: First, crops that have a large variety in shape and size need to be reliably detected in local sensor data and distinguished from wild growing vegetation. Second, to leverage information from multiple sensors, the detected crops need to be associated with previously detected or mapped crop features. Data association on crop features is challenging since the crops are hard to distinguish and almost uniformly distributed resulting in many highly ambiguous situations. Third, during turning maneuvers outside the field, the crops are not necessarily visible in the sensor data, so that errors in the pose estimate usually accumulate during these maneuvers. Thus, re-localizing the vehicle before it enters the field is another challenge for accurate pose estimation. Fourth, there is only little clearance between the wheels of a large agricultural vehicle and adjacent crop rows. Therefore, highly accurate heading and sideways tracking estimates are crucial to enable autonomous navigation without damaging the crops.

In this thesis we present novel techniques that tackle these challenges thus enabling accurate localization and mapping for autonomous navigation on agricultural fields. First, we introduce two novel crop row detection methods based on modeling crop rows as a set of parallel and equidistant lines. This model enables joint detection of all visible crop rows and thereby increases the robustness of the detection methods against wild growing vegetation as well as in challenging situations, where the crop row structure is not fully visible. Second, we present several techniques that enable pose estimation based on multiple sensor modalities including GPS and crop row detections from local vision or depth data. This includes data association based on geometric relations between crop rows, detecting the end of the field to improve the accuracy of the position estimate as well as the formulation of a sensor model that fuses information from different sensor modalities to obtain consistent pose estimates. Third, for re-localization after turning maneuvers and to further improve the accuracy of the pose estimate, we propose to detect and leverage individual plant positions as features instead of crop rows. To enable pose estimation relative to individual plant positions, we present a novel data association technique that leverages

slight irregularities in the distribution of the plants along the crop rows to determine the correct data association even in highly ambiguous situations.

We evaluate our techniques in extensive experiments on real-world data recorded in three different locations featuring seven crop types at varying growth stages. Our evaluation results confirm that our techniques contribute to the state of the art by enabling accurate pose estimation relative to crops across the entire field. Furthermore, our techniques were used in several successful autonomous navigation runs across entire crop fields, which showcases the suitability of our techniques for real-world application.

The techniques presented in this thesis enable accurate localization and mapping for reliable and precise autonomous navigation across entire crop fields – thereby contributing to automation in agriculture and sustainable precision farming applications.

Contents

Chapter 1

Introduction

Producing enough food to sustain a growing world population has become more and more challenging over the past decades. This has pushed development in the agricultural sector towards larger and larger monocultures to increase food production. The key advantage of having larger fields containing the same type of crop is that the same treatment can be performed on larger areas simultaneously and therefore more efficiently than different treatments on many smaller sized individual crop fields. Therefore, technological advances in agriculture have also been focused on increasing the size and thus range of agricultural machinery to perform a certain task more efficiently on a larger area. These tasks range from seeding, over watering, fertilizing and exterminating weeds or pests, up to harvesting and preparing the field before sowing the next crop. This large scale deployment of agricultural machinery is described as *industrial* agriculture, where a crop field is treated in a homogeneous manner. For example, in industrial agriculture, the whole field – including the crops – is sprayed uniformly with herbicide to exterminate weeds as shown on the left of Figure 1.1.

Nowadays the strain that industrial agriculture is putting on the environment becomes more and more evident, such as, for example, fertilizer that can contaminate the ground water or reduced biodiversity in large monocultures. Research towards more sustainable agricultural practices has therefore dramatically increased in recent years. The main challenge for alternative farming techniques is to reduce environmental stress while at the same time retaining as much of the production rate achieved with industrial methods as possible. One of the explored alternative farming concepts is based on introducing *precision* into the agricultural tasks performed by the machinery. For example, in *precision agriculture* instead of spraying the whole field with herbicide uniformly, the herbicide is only applied on the weed – with precision. While the weed is still exterminated, the amount of herbicide sprayed is dramatically reduced [Pretto et al., 2021]. This principle of precision agriculture, where any resource should only be applied with precision, can be transferred to most other tasks of cultivating a crop field. For example, detailed analysis of the soil at different locations of the field allows for targeted application of fertilizer, exactly where it is required – thereby reducing the amount of fertilizer brought into the environment. A similar argument holds for irrigation or the use of pesticides. The key idea of precision agriculture is that if resources are only applied in the right amount and at the right location, the same area of agricultural land can be cultivated but at the cost of much less resources while the size of the yield can be retained. Using less chemicals and

Figure 1.1: This figure shows an example for industrial weed treatment with a tractor on the left and the BoniRob, a robotic system developed by Deepfield Robotics for precision agriculture applications, on the right.

fertilizer only if and where required might even result in healthier crops and thus healthier food [Walter et al., 2017].

While differentiated treatment of different parts of the field provides a solution towards more efficient use of resources as well as more sustainable farming, it also involves higher production costs and higher risks: For precise treatment of a field the condition of the crops as well as the soil of the field has to be monitored closely. For example, different kinds of soil require different amounts of fertilizer for optimal crop growth. These soil properties can vary, even within smaller sized fields. Therefore, in precision agriculture the soil quality should be analyzed to determine if and how much fertilizer is required at each location on the field. This task of analyzing the soil adds to the work load of the farmer and therefore results in increased production costs. Some farmers also might not consider precision agriculture techniques due to the increased risk of a smaller yield or even loosing the entire yield, which can happen if reduction of resources is not performed correctly, for example, due to incorrect information about the soil quality. However, there is another obstacle that prevents a wide spread use of precision agriculture methods: Commonly used industrial agriculture machines are usually not well suited for precision agriculture applications as they are optimized for large scale and uniform treatment instead of precise treatment of smaller areas. One of the most prominent challenges to make precision agriculture more accessible and economically feasible is thus to design precision agriculture machinery that can perform differentiated treatment in an efficient manner.

In recent years an increasing amount of research has been focused on making the concept of precision agriculture more accessible and safer for production farming applications. This research includes investigating the use of automated vehicles for both, gathering the required detailed information about the condition of the field and its crops as well as performing the precise treatment after the acquired information has been evaluated. An example prototype for a vehicle specifically developed for precision agriculture applications is the BoniRob shown on the right of Figure 1.1. While fully autonomous traversal of crop fields is often required for the envisioned precision agriculture applications, it also has many advantages for the agricultural sector in general:

The main advantage of fully autonomous vehicles is that they considerably reduce the work load of industrial as well as precision agriculture. Since the vehicle performs the required tasks fully autonomously, the worker that would usually steer the vehicle along the crop field is free to perform other tasks. While fully autonomous vehicles might not be crucial for industrial farming, they can be an important stepping stone to make precision agriculture competitive.

Furthermore, fully autonomous vehicles can open the door to entirely new agricultural applications that are otherwise either not feasible at all or at least not economically feasible without full automation: For example, in industrial agriculture treatment such as spraying chemicals is usually performed at high speeds, which enables efficient treatment of large fields. However, precise treatment of local parts of the field or even on a per plant basis, such as removing individual weed plants, might only be possible at much lower speeds. Such a treatment is not economically feasible with a person steering the vehicle, since it would take far too much time to treat the entire field. However, with a fully autonomous system that can even be able to work through the night driving at much lower speeds can be considered.

Another precision agriculture application that might benefit from autonomous vehicles at lower driving speeds is harvesting high value crops. Driving at lower speeds could enable automated harvesting of crops that are usually gathered manually at high financial cost due to their fragile nature, such as, for example, strawberries [Xiong et al., 2020].

Another well researched precision agriculture application called phenotyping is not only valuable on production fields but also in crop science: In crop research, phenotyping describes the process of developing new crop varieties, new *phenotypes*, with special properties such as higher robustness against pests or increased yield. To find phenotypes with interesting properties many different phenotypes of the same crop are grown in small patches on research fields and closely monitored to detect and evaluate any special properties a certain phenotype might express. This research is usually based on a lot of manual work, including manually measuring properties of interest such as foliage coverage, plant growth as well as close visual inspection of the different phenotypes, to, e. g., determine a possible infestation with pests. Automated vehicles can help monitor these phenotypes by collecting detailed data on each patch of phenotypes on a regular basis [Gomez et al., 2021, Magistri et al., 2021, Smitt et al., 2021].

These are only a few examples that illustrate how fully autonomous navigation not only plays an important role for more efficient conventional agriculture, but how it can also open the door to more sustainable precision farming applications that are otherwise economically not feasible. For fully autonomous navigation in agriculture that is also feasible in practice a navigation system that steers the vehicle along the crop field needs to satisfy the following requirements: First, it needs to guide the vehicle along the crop rows of an agricultural field with high *precision* to avoid damaging the valuable crops by driving over them. Second, a navigation system for real-world application needs to be *fully autonomous* and should not require manual intervention, such as for example manually turning the vehicle at the end of each crop row. Third, the navigation system should *reliably* traverse entire crop fields, without experiencing critical errors that require manual intervention or stopping the vehicle for longer periods of time. One of the key

components to obtain such a precise and reliable fully autonomous navigation system is an accurate pose estimate of the vehicle. Within the navigation system this pose estimate is computed by a localization module that uses a map of the environment to localize the vehicle within this map. Such a map is usually obtained through mapping techniques that use pose estimates of the vehicle to fuse the observed sensor data into a globally consistent representation of the environment. While localization and mapping techniques are well researched for indoor and urban outdoor environments, an agricultural field poses several challenges for traditional localization and mapping approaches. In this thesis, we present techniques that tackle these challenges and thereby enable robust and accurate localization and mapping towards precise and reliable fully autonomous navigation on agricultural fields.

1.1 Challenges for Localization and Mapping in Precision Agriculture

While localization and mapping for autonomous navigation is well researched in the field of robotics in general, the focus often lies on indoor or urban outdoor environments [Cadena et al., 2016, Kümmerle et al., 2014, Levinson and Thrun, 2010, Pradalier and Sekhavat, 2002, Thrun et al., 2005, Winterhalter et al., 2015]. However, obtaining reliable and accurate pose estimates to enable fully autonomous navigation on agricultural fields poses novel challenges.

A technique commonly used in industrial agriculture applications to obtain pose information of the vehicle is based on high precision GPS signals. Such a localization approach might be well suited for industrial agriculture applications with the goal to guide the vehicle across the field to cover as much area of the field as efficiently as possible. However, the main disadvantage of this localization is that it is solely dependent on the accuracy of the received GPS signal, which can easily become a single point of failure in the navigation system. For example, during GPS outages the accuracy of the high precision GPS signal can drop considerably. In the best case, the signal outage is detected by the vehicle and it stops and waits until the signal reception improves, which makes the navigation system unreliable in areas that have problems with GPS signal reception. In the worst case, the signal outage is not detected and the vehicle continues traversing the field based on inaccurate position information, which usually results in imprecise driving behavior where the vehicle damages the value crops by driving over them. In order to overcome this single point of failure, additional local information obtained from sensors mounted on the vehicle is commonly used in other robotics applications, such as autonomous driving, to improve the accuracy and robustness of the pose estimate. For reliable and precise autonomous navigation on crop fields additional local sensor information should therefore also be used to estimate the pose of the vehicle in the agricultural setting.

In traditional localization or mapping approaches this local sensor information is compared to the mapped information of the environment to estimate the pose of the vehicle. If the raw sensor data is not discriminative enough to provide sufficient information about

Figure 1.2: This figure shows impressions of different crop research fields located at the crop science research station of ETH Zurich in Eschikon (left and right) as well as in Ancona, Italy (mid). The left two images show a sugar beet field with tiny plants that have just emerged, so that they are barely visible from an elevated point of view. The image in the middle shows the same type of crop at a later growth stage with clearly visible crop row structure on a sunny day in late spring. The image on the right shows canola crops at a very late growth stage on a cold and cloudy day in autumn. The foliage of the canola crops is overlapping into neighboring crop rows, covering the soil between crop rows and thereby occluding the crop row structure.

the pose of the vehicle, the raw data is often processed into more discriminative information by detecting unique *features*, also called *landmarks*, in the local sensor data. These features are then compared to the features in the map to estimate the pose of the vehicle within the map according to the observed features. These features are usually large, distinct objects in the environment that are easy to detect in the sensor data of the vehicle, for example, doors in a hallway of an indoor environment, or buildings, lamp poles or street signs in an urban setting. However, on an agricultural field as shown in Figure 1.2, such distinct structures are usually not present. While there might be larger structures like sheds or trees next to the field, they are only visible if the vehicle is in close proximity to those landmarks. However, most of the time the vehicle is driving in the field, where these landmarks are not within sensor range. Therefore, such landmarks at the edge of the field cannot be used for pose estimation during autonomous navigation. The only feature that is always present on an agricultural field are the crops themselves. Therefore, most agricultural localization approaches that use local sensor information detect the crops in the sensor data and estimate the pose of the vehicle relative to these crops. The fact that local-sensor-based pose estimation can only rely on the crops as features is the main reason why localization and mapping on agricultural fields is challenging:

First, in order to use the crops as features in localization or mapping they need to be detected in the sensor data. This is challenging not only because there is a large variety of different kinds of crops that come in a lot of different shapes and sizes, but even the same crop type has a very different appearance over its life cycle. This can also be seen in Figure 1.2. The two left most images show sugar beets in a very early growth stage, where the plants have just emerged and are barely visible in images captured from an elevated point of view (left most image). Detecting these tiny crops is challenging since it is hard to distinguish them from the soil. While detecting the same crop in a later growth stage as shown in the middle of Figure 1.2 might be more straightforward, a reliable crop

detection method needs to be able to detect these larger plants as well as the tiny plants. Another extreme example is shown on the right of Figure 1.2, where the crops are at a late growth stage so that the foliage of the plants covers almost the whole field. In these overgrown crop fields it is difficult to define an accurate location of the crop features since almost no contrasting soil is visible in the sensor data. While detecting a wide variety of crops at different growth stages is important, another requirement for a crop detection method intended for pose estimation is robustness against other wild growing vegetation. This vegetation is usually also not considered as features when creating the map of the field since it might not always be present and could even be removed between consecutive traversals of the same field. Therefore, these plants should not be detected by a crop detection method. Such wild growing vegetation can either be weeds found inside the crop field, which are usually removed by performing some kind of weed treatment, or grass or larger plants around the edges of the field (see left most and right images of Figure 1.2). While weeds inside the field are usually sparsely distributed, the vegetation outside the field can range from grass that covers the whole ground, over bushes and even trees. Being robust against all kinds of wild growing vegetation, while at the same time also detecting a large variety of crops reliably is quite challenging. Another important requirement for any outdoor perception method is reliability throughout the whole year. Therefore, it should be robust against changing lighting conditions that can range from a sunny summer day (middle of Figure 1.2) to a cloudy autumn day (right of Figure 1.2) as well as bad weather conditions such as rain or fog.

Second, the detected crop features need to be associated with the mapped crop features to facilitate estimating the pose of the vehicle within the map. This problem is commonly known as the *data association* problem and finding the correct association is crucial for robust localization or mapping results. In the literature, data association is usually either based on descriptors that encode unique information of each feature or based on geometrical properties and relations between the features, if no unique properties can be found. Finding unique information to distinguish individual plants of the same type or even individual crop rows containing the same crop type is hard. This is not only caused by the fact that crops of the same type look similar, but also by the fact that crops change their appearance over time as they grow. Therefore, the data association problem needs to be solved based on the geometrical properties and relations between the crops. Most data association approaches that use geometrical features require a sparse and unique distribution of features and usually do not scale well on large amounts of features. However, the crops on an agricultural field satisfy none of those requirements. They are usually sown in parallel and equidistant crop rows and even within the rows they are densely and evenly distributed (see Figure 1.2). Finding the correct data association on such highly ambiguous feature distributions is therefore a hard challenge.

Third, the crops can only be perceived and therefore used for pose estimation while the vehicle is inside the field. Besides mounting additional sensors at the back of the vehicle just for turning, the front mounted sensors of the vehicle capture less information containing the crops of the field and more information containing possibly wild growing vegetation outside the field as it approaches the end of the field. Nevertheless, for fully autonomous navigation the vehicle needs to perform turning maneuvers outside the field

Figure 1.3: This illustrates the spacing between adjacent crop rows on the example of sugar beet plants in an early growth stage. For comparison, the image on the right shows one of the wheels of the BoniRob, a large agricultural vehicle, next to the sugar beet crop rows. Wheels of agricultural vehicles are usually quite wide to improve grip on muddy soil as well as distribute the weight of the vehicle on a larger area to reduce soil compression. This leaves only few centimeters clearance between adjacent crop rows and the wheels of the vehicle.

and thus frequently leave and reenter the field. To facilitate turning outside of the field, a localization module needs to rely on additional sensor information to perform the turning maneuver, such as relative motion information obtained from wheel odometry and inertial measurement unit (IMU) sensors. Since tracking of the crops of the field is lost during these "blind" turning maneuvers, the pose estimate after turning might not be correct. Therefore, the autonomous vehicle needs to be re-localized after the turning maneuver according to the detected crop row structure. This again requires finding the correct data association between the crop row structure observed after turning and the mapped crop structure of the field. Solving the data association problem after turning is more challenging, since the pose estimate after turning might be inaccurate and therefore a larger amount of highly ambiguous data association matches needs to be considered.

For precise autonomous navigation on crop fields in general the pose estimate provided to the navigation system needs to be sufficiently accurate. Highly accurate heading and lateral offset estimates of the vehicle relative to the crop rows are crucial to ensure precise navigation within a crop field. This is due to the fact that there is usually only few centimeters clearance between the wheels of a large agricultural vehicle and adjacent crop rows as can be seen in Figure 1.3.

1.2 Contributions

In this thesis we present novel techniques that overcome the aforementioned challenges to enable localization and mapping for precise and reliable fully autonomous navigation in agriculture. To this end, we investigate how a large variety of crop types at different growth stages can be reliably detected, while also being robust against wild growing vegetation. We then determine how these detections can be used to obtain a full pose estimate of the vehicle in conjunction with information from other sensor modalities such as GPS, wheel odometry, and IMU to facilitate precise and reliable fully autonomous navigation

of entire crop fields. To further improve the robustness and accuracy of the obtained pose estimates, we also investigate how an autonomous vehicle can be re-localized after performing a turning maneuver at the end of the field in the context of a mapping framework. In the following we summarize the main contributions of this thesis.

Robust Crop Row Detection For precise and reliable autonomous navigation on crop fields using local crop row detections to obtain an accurate pose estimate of the vehicle is an integral part. In Chapter 3 we present two novel crop row detection methods that detect crop rows on a large variety of crop types at different growth stages. Both methods are based on our idea to model the crop rows as a set of parallel and equidistant lines, which we call a *Pattern*. This enables the crop row detection method to consider all available data and jointly extract the visible crop rows. Our experimental evaluation confirms that this results in robust and accurate crop row detections inside as well as outside of the field. The presented crop row detection was used in many successful autonomous runs across entire crop fields to provide accurate heading and lateral offset information of the vehicle relative to the crops.

Beyond Crop Row Following In order to determine a full pose estimate, including the heading, the lateral offset to the crop rows as well as the longitudinal position along the crop rows, local crop row detections need to be processed together with information from other sensor modalities. In Chapter 4 we therefore present several crucial techniques to facilitate fusion of these sensor modalities into one localization method that produces consistent pose estimates. Our key idea is that fusing local crop row detections with global sensor measurements from GPS signals requires a common reference frame in which the pose estimate is defined. We define this common frame by providing a GPS-referenced map of crop rows to the localization method. An integral part of our localization algorithm is the ability to associate the locally detected crop rows with the crop rows given in the GPS-referenced map. We thus introduce our novel *Crop Row data association* that analyses the geometric relations between sets of lines to determine the correct data association. We also present a novel method for detecting not only the crop row structure of the field, but also the location of the *End of the Field*. This allows us to not only correct the heading and lateral offset but also the longitudinal position of the vehicle according to local sensor information. Finally, we present the sensor model of the localization algorithm, where we split all sensor information into a heading, a lateral and a longitudinal component. Here, the lateral and the longitudinal direction are defined relative to the direction of the mapped crop rows. Splitting the sensor information in this fashion enables integrating the most accurate measurement for each component to obtain consistent pose estimates. Our experimental evaluation shows that the presented techniques result in localization methods that provide accurate and consistent pose estimates throughout the entire traversal of a crop field, including turning maneuvers outside the field. The techniques presented in this chapter have provided robust and accurate pose estimates to steer an agricultural vehicle with precision along entire crop fields during multiple fully autonomous runs – thereby enabling autonomous navigation beyond crop row following.

Data Association on Individual Plants While the local detections of the crop structure provide accurate information inside the field, sensors usually do not perceive the crop structure during the turning maneuver outside the field, since the vehicle is not facing the field for most of the time. This means that the autonomous vehicle cannot continue tracking the features and the error of the pose estimate accumulates during turning. However, for efficient navigation maneuvers, where the autonomous vehicle does not skip rows or traverses the same rows multiple times, the pose estimate of the vehicle needs to be corrected after turning. The key requirement to facilitate this re-localization is to determine which rows the vehicle is facing after the turning maneuver and therefore, association of the correct crop rows after turning. In Chapter 5, our key idea is that this can be facilitated by representing the crop structure not as crop rows (line features) but instead as individual plant positions (point features). We then leverage slight irregularities in the distribution of individual plants along the crop rows to determine the correct data association after turning. Partially inspired by our *Crop Row data association* from Chapter 4, we present a novel data association on individual plants. Our data association can handle large amounts of indistinguishable, densely and almost uniformly distributed features by leveraging geometrical relations between features to determine the correct data association. The key novelties of our approach are an efficient parameterized representation of possible sets of data association matches as well as a highly discriminative, but still robust strategy that enables counting matches in a continuous fashion. Furthermore, we also investigate how our data association approach can be applied in a mapping framework on real-world data to obtain GPS-referenced maps of individual crop positions from ground vehicle data. Our in-depth experimental evaluation shows that our data association approach on individual plants can indeed find the correct data association after turning on indistinguishable, densely and almost uniformly distributed crop features.

The techniques presented in this thesis not only enable robust detection of the crop row structure, but also facilitate integrating these detections with information from other sensor modalities into an accurate and consistent pose estimate well suited for reliable and precise fully autonomous navigation on crop fields. Furthermore, finding the correct data association on a per plant basis enables re-localizing an agricultural vehicle after performing a turning maneuver. While this enables localization and mapping applications for ground vehicles without the need of high precision GPS information, it can also improve the efficiency of fully autonomous navigation.

1.3 Publications

Parts of this thesis have been published in international peer-reviewed conferences and journals. We list these publications in chronological order below.

- N. Chebrolu, P. Lottes, A. Schaefer, W. Winterhalter, W. Burgard, and C. Stachniss. Agricultural robot dataset for plant classification, localization and mapping on sugar

beet fields. *International Journal of Robotics Research (IJRR)*, 36(10):1045–1052, 2017

- W. Winterhalter, F. Fleckenstein, C. Dornhege, and W. Burgard. Crop row detection on tiny plants with the pattern hough transform. *Robotics and Automation Letters (RA-L)*, 3(4):3394–3401, 2018

- F. Fleckenstein, W. Winterhalter, C. Dornhege, C. Pradalier, and W. Burgard. Smooth local planning incorporating steering constraints. In *International Conference on Field and Service Robotics (FSR)*, 2019

- A. Pretto, S. Aravecchia, W. Burgard, N. Chebrolu, C. Dornhege, T. Falck, F. Fleckenstein, A. Fontenla, M. Imperoli, R. Khanna, F. Liebisch, P. Lottes, A. Milioto, D. Nardi, S. Nardi, J. Pfeifer, M. Popović, C. Potena, C. Pradalier, E. Rothacker-Feder, I. Sa, A. Schaefer, R. Siegwart, C. Stachniss, A. Walter, W. Winterhalter, X. Wu, and J. Nieto. Building an aerial-ground robotics system for precision farming: an adaptable solution. *IEEE Robotics & Automation Magazine (RAM)*, 28(3): 29–49, 2021. IEEE Robotics & Automation Magazine Best Paper Award

- W. Winterhalter, F. Fleckenstein, C. Dornhege, and W. Burgard. Localization for precision navigation in agricultural fields—beyond crop row following. *Journal of Field Robotics (JFR)*, 38(3):429–451, 2021

1.4 Collaborations

Parts of this thesis are the result of joint work with other researchers. The supervisor of this thesis, Wolfram Burgard, contributed suggestions and ideas to all of its parts. Collaborations with other researchers are detailed in the following:

- In Chapter 2, the results of joint work with research colleagues from the Flourish project is presented. This research resulted in several publications [Chebrolu et al., 2017, Fleckenstein et al., 2019, Pretto et al., 2021]. The author of this thesis contributed to the work by Chebrolu et al. [2017] by extrinsically calibrating the sensors of the ground vehicle, the BoniRob, which was used to record the data set. In the work ofPretto et al. [2021] the presented crop row detection and localization algorithms used on the BoniRob for autonomous navigation are the main contribution of this author. The publication of Fleckenstein et al. [2019] presents a solution to obtain smooth local plans by considering steering constraints, which is crucial for efficient traversal of crop fields with large agricultural vehicles. Cédric Pradalier contributed to this work by providing the formulation of the steering constraints as an instantaneous center of rotation (ICR) path. Freya Fleckenstein, the main contributor of this work then translated the ICR paths into valid velocity paths by providing the idea of velocity roll-outs. Christian Dornhege provided input on experiment design and evaluation. The author of this thesis supported Freya Fleckenstein with data set collection and experiment execution.

- The crop row detection methods presented in Chapter 3 are part of joint work with Freya Fleckenstein and Christian Dornhege and resulted in the publication of Winterhalter et al. [2018]. Freya Fleckenstein provided the idea and implementation of the vegetation Feature Maps. These Feature Maps are used as the input to the crop row detection algorithms. The Pattern formulations and the crop row detection algorithms are the main contributions of this author. Christian Dornhege, Freya Fleckenstein and the author of this thesis performed the data collection and experimental evaluation presented in this thesis. Large parts of Chapter 3 have been previously published in the work by Winterhalter et al. [2018].

- The techniques presented in Chapter 4 are the result of joint work with Freya Fleckenstein and Christian Dornhege and have also been published in the work by Winterhalter et al. [2021]. Freya Fleckenstein contributed the definition and implementation of the quality measure on detected Patterns of crop rows. The other presented techniques including the *Crop Row data association*, the *End of the Field detection* as well as the definition of the sensor measurements and implementation of the localization algorithms are the main contributions of this author. Christian Dornhege, Freya Fleckenstein and the author of this thesis collected the data and performed the experimental evaluation presented in this thesis. Large parts of Chapter 4 are published in the work by Winterhalter et al. [2021].

- The data association technique on individual plants as well as all other techniques presented in Chapter 5 except for the SEP detection network are the sole contribution of this author and are previously unpublished. The SEP detection network was developed by Nina Pant for her Bachelor thesis, who was supervised by the author of this thesis and Nicolai Dorka.

Chapter 2

Basics and Definitions

In this chapter, we discuss topics that are relevant for the techniques as well as the experimental evaluation presented in this thesis. The agricultural setting is quite different from other more commonly researched indoor or urban outdoor scenarios. In the first section of this chapter, we therefore provide an introduction into autonomous navigation in agriculture. Solving the data association problem is an important part of any feature-based localization and mapping technique. In the second section, we thus give an overview of different terms and definitions that we will use throughout the thesis to efficiently introduce our novel data association techniques.

2.1 Precise Autonomous Navigation in Agriculture

In this section, we introduce the agricultural terms used in this thesis. We also explain in more detail the special requirements for the accuracy of the pose estimate to guide an autonomous vehicle across an agricultural field without damaging the crops. Furthermore, we discuss the hardware and software components that are relevant in an autonomous navigation system and their setup using the BoniRob as an example (see Figure 2.2). Finally, we also give a brief overview over the different experiment locations where we collected the data sets used in the experimental evaluation sections of this thesis.

2.1.1 Terms and Accuracy Requirements

For better understanding of the special requirements for precise autonomous navigation on agricultural fields, we give an overview of a crop field and the usually desired s-shaped path of the agricultural vehicle when traversing the field in Figure 2.1. First, the vehicle needs to approach the field and align itself with the crop rows to enter the field without damaging the crops. Second, the autonomous vehicle should track the crop rows and determine its pose relative to the crops with high precision in order to drive along the crop rows without driving over the crops. Third, the vehicle leaves the crop field, while still driving aligned with the crop row structure until it has completely left the field. If desired, the vehicle then performs a turning maneuver on the area just outside the field, to re-align itself with the next set of crop rows that should be traversed. These steps are then iterated to traverse the entire field. In this thesis, we distinguish between the second type of driving behavior, called driving *in-row*, where the vehicle is driving inside the field

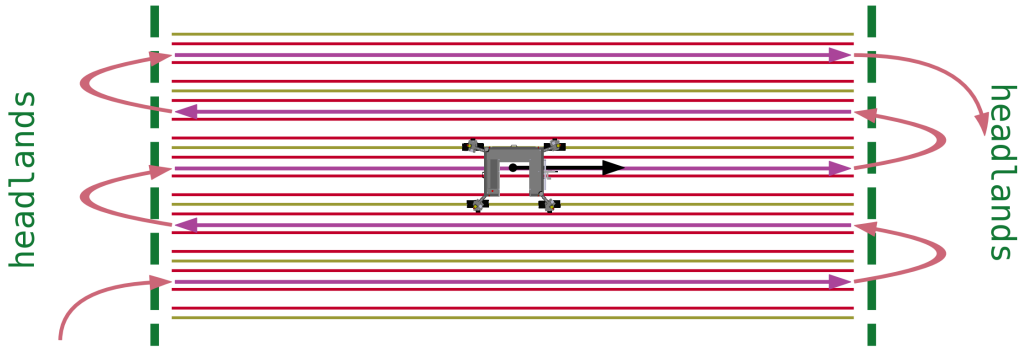

Figure 2.1: This figure shows an overview of an agricultural field. The crop row structure is visualized using red lines. In this example, the crops are sown in rows of three between adjacent wheel tracks (olive lines). The BoniRob is traversing the field aligned with the direction of the crop rows on the wheel tracks. The path that the vehicle should take to traverse the field is shown using purple and rose arrows. The purple arrows denote the in-row driving, transition maneuvers are highlighted in rose. The headlands are the area free of crops to the left and right side of the green dashed lines.

aligned with the crop row structure and the other driving maneuvers outside the field, called *transition* maneuvers. These transition maneuvers include turning outside the field as well as entering and leaving the field. The area just outside the field that serves as free space for the agricultural machines to perform transition maneuvers is called *headlands*.

Another important parameter on a crop field is the spacing between the crop rows, which we call *(crop) row spacing*. The value of this crop row spacing depends on the space requirements of the crop types cultivated on the field but is also restricted by the track width of agricultural vehicles. Therefore, depending on the space requirements of the crop type, usually two or three crop rows are sown between adjacent *wheel tracks*. We give an example for a set of three crop rows between the wheel tracks in Figure 2.1. Since most agricultural vehicles have a normed track width of $1.5\,\mathrm{m}$, crop rows are usually sown with a spacing of around $0.5\,\mathrm{m}$ for three crop rows and $0.75\,\mathrm{m}$ for two crop rows. Together with the fact that larger agricultural vehicles usually have large and wide wheels to prevent slippage on muddy soil and decrease soil compression, this usually leaves only little clearance of few centimeters between the wheels of the agricultural vehicle and adjacent crop rows as already shown in Figure 1.3.

In Chapter 1, we already mentioned the terms *heading* as well as *lateral* and *longitudinal* position of the agricultural vehicle. While the heading is commonly used in robotics and denotes the direction the autonomous platform is facing, we use the terms *lateral* and *longitudinal* in this thesis with a meaning specific to the agricultural environment using the direction of the crop rows on the field: The longitudinal direction is always parallel to the crop rows. The lateral direction is orthogonal to the longitudinal direction and therefore orthogonal to the direction of the crop rows. During evaluation, due to the different requirements for the accuracy of the lateral and longitudinal position of the vehicle, it is crucial to evaluate the position of the vehicle with respect to both directions separately. Usually, a longitudinal position accuracy of $1\,\mathrm{m}$ and below is sufficient to en-

sure autonomous turning maneuvers at the end of the field without damaging the crops. In contrast, due to the low clearance between the wheels of larger vehicles and adjacent crop rows, a high lateral and heading accuracy of $0.1\,\mathrm{m}$ and 10° respectively are crucial for precise autonomous navigation within the field.

2.1.2 Autonomous Navigation System

The navigation system of an agricultural vehicle usually consists of several hardware and software components to perform precise and reliable autonomous navigation on a crop field. In the following we give an overview of the most important components of such a navigation system and also briefly discuss important setup steps that enable precise autonomous navigation. As an example for such an autonomous navigation system, we reference the BoniRob.

In order to perceive the environment around an autonomous vehicle, at least one or multiple sensors that collect information about the surrounding structures and objects are mounted on the vehicle. The most common sensor modalities are vision or depth-based such as, for example, RGB cameras or LIDAR sensors. If the perceived sensor information is intended for the purpose of autonomous navigation, it is usually mounted so that it faces in the main driving direction of the vehicle to collect information about the area the vehicle is going to traverse. On the BoniRob, these sensors are a PointGrey Blackfly that provides RGB images at a resolution of five megapixels and a frame rate of $5\,\mathrm{Hz}$ and a Nippon-Signal FX-8 that captures depth information in form of a three-dimensional point cloud at $5\,\mathrm{Hz}$. Another purpose of such sensors is to provide information about the local environment to ensure the safety of the autonomous vehicle by detecting objects in the environment and analyzing the terrain. This information is then used to prevent collision with objects in the environment as well as damage to the platform by traversing terrain that is unsuited to capabilities of the platform. Sensors used to ensure safe driving behavior are therefore mounted so that they perceive as much of the surrounding environment as possible in order to minimize the blind spots of the vehicle. On the BoniRob, we use two Velodyne VLP-16 LIDAR sensors for this purpose capturing point clouds at $10\,\mathrm{Hz}$. We give an overview of the sensors and their mounting position on the BoniRob as well as examples for the sensor data used in this thesis in Figure 2.2.

Additionally to the local sensors that perceive the environment, an autonomous vehicle for outdoor environments is usually also equipped with a GPS sensor that provides global position information. While agricultural machines on production fields are often equipped with expensive high precision GPS receivers, we decided to use a standard precision GPS receiver, a u-blox EVK-7, with an accuracy of around $3\,\mathrm{m}$ running at $4\,\mathrm{Hz}$. We also measure changes in the orientation of the BoniRob using the IMU sensor SBG Ellipse2-A as well as the traversed distance using the wheel odometry provided by the BoniRob. Both sensors provide data at a high rate of $100\,\mathrm{Hz}$ for the IMU sensor and $20\,\mathrm{Hz}$ for the odometry measurements. To provide enough computing power for the navigation system we used two Pokini i2 with Intel Core i7-4600U CPU and 16 GB RAM. The first is used to run all required drivers and handle data flow and communication between the software modules of the navigation system. The second Pokini i2 is reserved for the navigation

Figure 2.2: This figure gives an overview of the sensor setup of the BoniRob platform. The left image shows the BoniRob. The sensors of the platform that are visible in this image are highlighted in orange. This includes one of the Velodyne VLP-16 sensors in the top left as well as the Nippon-Signal FX-8 and the PointGrey Blackfly. On the right, we show the two sensors and corresponding example sensor data that are used during the experimental evaluation in this thesis.

software to ensure stable online performance of all navigation algorithms.

One crucial requirement when handling different data streams from different sensors that are received at different frequencies is time synchronization. We therefore setup the navigation system to provide accurate time stamps for each sensor measurement of all sensors. Using the time stamps, we can synchronize the data from all sensors to obtain data tuples d_t for a certain point in time t containing one measurement of each sensor at approximately the same time stamp: $d_t = (C_t, I_t, o_t, i_t, g_t)$, where C_t is the corresponding point cloud information captured by the Nippon-Signal FX-8, I_t is the RGB image of PointGrey Blackfly, g_t the GPS position information and i_t and o_t are the relative motion measurements from the IMU and the wheel odometry.

Another important setup step is intrinsic and extrinsic sensor calibration. Both are essential to enable comparison and fusion of information obtained from different sensors. Intrinsic calibration of a sensor is required to properly interpret the raw sensor data of each specific sensor to, for example, rectify an RGB image and remove lens distortion effects. Extrinsic calibration usually describes the exact mounting position and orientation of the sensor on the vehicle. This information is also necessary to relate measurements from different sensors, for example, to consistently merge point cloud information from two different LIDAR sensors into one larger point cloud. The intrinsic and extrinsic calibration of the camera and LIDAR sensors of the BoniRob is explained in more detail

in the work by Chebrolu et al. [2017]. The PointGrey Blackfly was added at a later point, therefore it is not mentioned by Chebrolu et al. [2017]. We manually determined the extrinsic calibration parameters between the PointGrey Blackfly and the Nippon-Signal FX-8 using a calibration target visible in the data of both sensors.

The navigation modules of an autonomous system are usually divided into *perception*, *localization*, *path planning* and *execution*. Additionally, there usually also exists a – not necessarily online – *mapping* module. While the first four components need to run online at a stable rate to precisely and reliably navigate the vehicle, the mapping module is usually applied offline on manually recorded data to create a map of the environment. In general, the perception module is responsible for interpreting the local sensor data such as images and point clouds and detect objects of interest. In the context of localization and mapping, these objects of interest, called *landmarks* or *features*, are used to localize the vehicle by comparing the detected features with the features in the map. On crop fields, the main focus therefore lies on detecting the crop and sometimes also other plants such as weeds. The localization module then uses the map provided by the mapping module, the detections from the perception module as well as other sensor information such as GPS position, IMU and odometry measurements to estimate the pose of the vehicle at the current time step. An accurate pose estimate is crucial for any autonomous navigation system since the path planning and execution modules depend on this pose estimate. Based on the pose estimate of the localization module the path planner determines the path that the vehicle should follow and can also determine if a new plan is required. Similarly, the execution module sends motion commands to the vehicle based on the given path from the path planning module and the pose estimate of the vehicle. An overview of all modules of the navigation system can be found in the work by Pretto et al. [2021]. The path planning and execution modules are presented in detail in the publications by Fleckenstein et al. [2017] and Fleckenstein et al. [2019] respectively. The focus of this thesis is on the perception, localization and mapping modules of the autonomous navigation system.

While all data collection as well as experimental evaluation presented in this thesis was performed using the BoniRob, the developed techniques and algorithms are transferable to other agricultural machines, if they have a comparable sensor setup.

2.1.3 Experiment Locations

The experimental evaluation in this thesis uses many different data sets recorded with the BoniRob. Each data set contains time synchronized and calibrated data of all sensors as described above. In this section, we give a brief overview of the different data sets including location, season and crop types.

The majority of the data sets was recorded on different production fields in Eichstetten, which is close to Freiburg im Breisgau in Germany[1]. The other data sets were recorded in the course of the Flourish project on different fields either during consortium meetings at the crop science research station of ETH Zurich in Eschikon, Switzerland, or during the

[1]We thank Herbert Rinklin who kindly allowed us to perform our experiments and data collection on his vegetable production fields.

"Flourish Demo" on fields prepared for the presentation of the Flourish project in Ancona, Italy.

In Chapter 3, we evaluate on data recorded in all three locations on varying crop types as well as during different seasons: The sugar beet data sets were recorded in Ancona in late spring 2018. In the summer of 2016 we recorded the leek data set on a production field in Eichstetten and in autumn of the same year, we also recorded the canola and corn data sets in Eschikon. For the evaluation in Chapter 4 and Chapter 5, we again recorded data on the production vegetable field in Eichstetten in the summer of 2018. This vegetable field is different to the other crop fields presented in this thesis since it features three different crop types on the same field that even change mid row: kohlrabi, Chinese cabbage and sweetheart cabbage. Another difference of this field compared to the others is that it has irregular crop row spacing with a larger spacing between crop rows adjacent to the wheel tracks. Due to these two differences this production field is quite interesting for the evaluation of our presented techniques, since it provides further insight into the advantages of our presented techniques as well as additional challenges. Just two weeks later, we collected another data set on a field with sugar beets in Eschikon for our evaluation in Chapter 5. We give an overview of the recorded data including example images for each crop type in Figure 2.3.

Overall, we evaluate the techniques presented in this thesis on data collected at three different locations in three different countries during spring, summer and autumn on research as well as production fields featuring a total of seven different crop types.

2.2 Terms and Definitions for Data Association

Data association is an important part of feature-based pose estimation in localization and mapping. In this section, we therefore introduce terms and mathematical notation that are used throughout the thesis to efficiently define several different data association techniques. First, we formulate the general problem statement for data association. Second, we describe different data association techniques found in the literature. We extract and name different concepts and ideas from these techniques so that we can refer to them more easily throughout this thesis. Finally, we also discuss how the accuracy of a data association algorithm can be evaluated.

2.2.1 The Data Association Problem

In feature-based localization and mapping, locally detected features need to be compared to the features in the map to correct the pose estimate accordingly. For this comparison, knowledge about which detected feature corresponds to which mapped feature is crucial. A mapped and detected feature should only be associated if they describe the same location or object in the environment. However, the correct data association is often not known, for example, due to uncertainty in the pose estimate or indistinguishable features. Finding the correct data association between the set of detected features, also called *observations* or *observed features* F_o, and the *mapped features* F_m is the *data association*

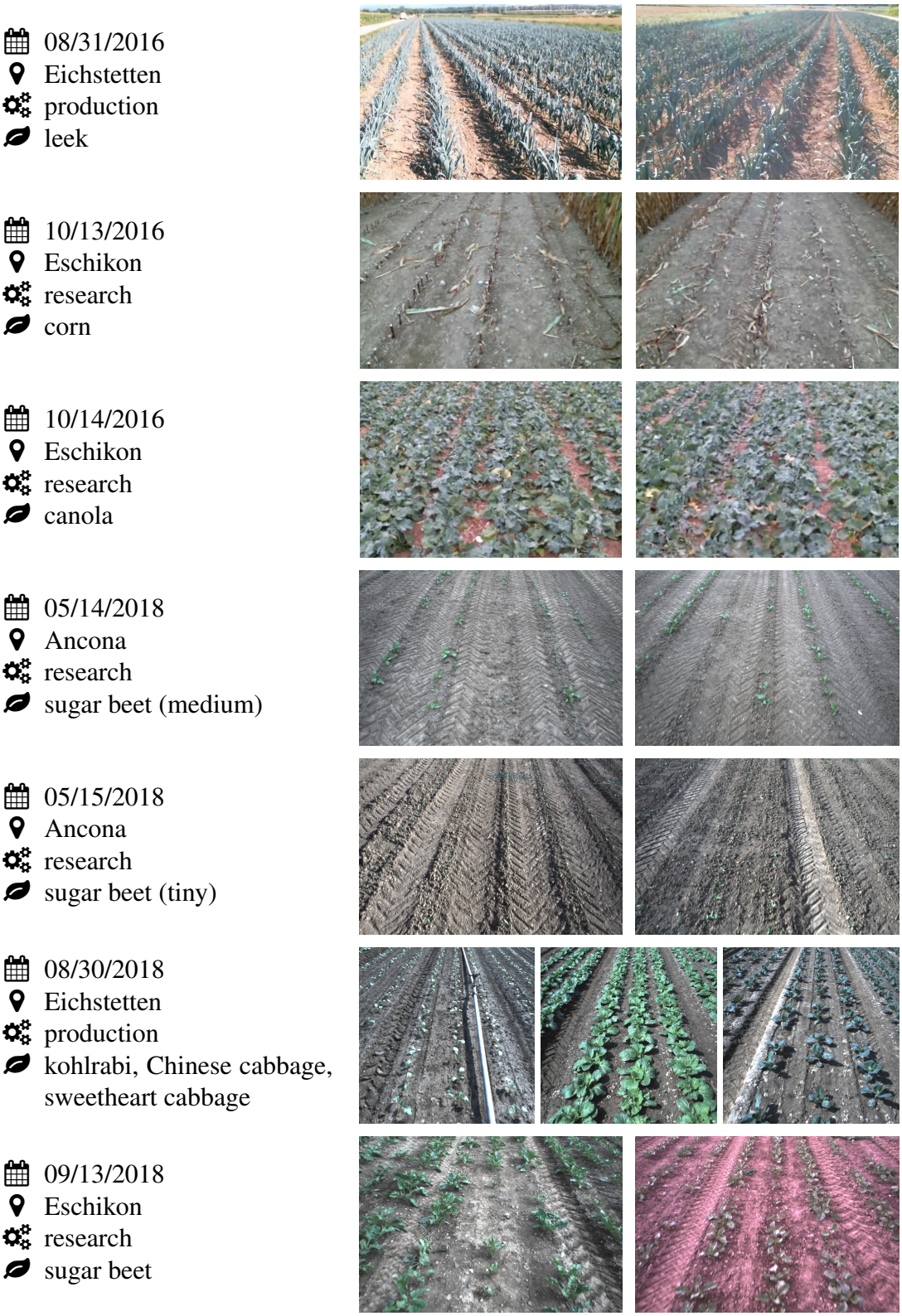

Figure 2.3: This figure gives an overview of the different crop fields on which we recorded data with the BoniRob for our experimental evaluation. On the right we show two to three images of each data set from different locations on the field.

problem. While there are many different notations and terms for the solution of the data association problem in the literature, we decided to describe the solution as a set of pairwise matches M, where the first element of a match is from the set of observed features F_o and the second element of a match is from the set of mapped features F_m:

$$M \;:=\; \{(o,m) \in F_o \times F_m \mid \forall\,(o',m') \in M\colon o = o' \Rightarrow m = m'\}$$

In order to ensure well defined correspondences between the observed and mapped features, a set of data association matches M can only match an observed feature once. This definition explicitly allows duplicate matches in the sense of assigning different observations to the same mapped feature as well as not matching an observed feature $o \in F_o$ by not having an element that contains this observation. Thus, the size $|M|$ of a set of data association matches M is always smaller or equal to the number of observed features $|F_o| := n_{F_o}$. Throughout this thesis, we use this representation of a set of matches M to describe a possible solution to the data association problem. Furthermore, we define the set of all possible sets of data association matches $\mathcal{M} \subset \mathcal{P}\,(F_o \times F_m)$. This set $\mathcal{M}$ contains all sets of data association matches for a given set of observed and mapped features F_o and F_m. Depending on the size n_{F_o} of F_o and the size n_{F_m} of F_m, the number of possible solutions to the data association problem, i. e., the size of $\mathcal{M}$ can become quite large due to its combinatorial complexity: $|\mathcal{M}| = (n_{F_m}+1)^{n_{F_o}}$.

Given a set of observed features F_o and set of mapped features F_m, the goal of a data association technique can then be formulated as finding the correct data set of matches $M^{c^*} \in \mathcal{M}$ between the observations in F_o and the mapped features of F_m. As stated before, a mapped and observed feature are associated if they are generated from the same instance in the environment. This can be, for example, the same tree in a park, the same edge or corner in a hallway, or the same plant on a crop field. Therefore, the correct data association M^{c^*} matches each observation to its true correspondence in the map, if such a correspondence exists in the map. If the observed feature is not within the map, the correct data association M^{c^*} does not contain a match for this observation. Solving the data association problem then amounts to finding the correct set of data association matches M^{c^*}, or at least a good approximation $M^* \in \mathcal{M}$ that is *close to* the correct data association M^{c^*}. In the context of defining the data association as a set of matches, we need to define a function c^* that measures how close a set of matches M is to the correct set of matches M^{c^*}. Such a function c^* could, for example, be defined as follows:

$$c^*\,(M) := \left|M \cap M^{c^*}\right| - \left|M \setminus M^{c^*}\right| \qquad\qquad \in \left[-\,|M|\,,\left|M^{c^*}\right|\right]$$

A set of matches $M \in \mathcal{M}$ is therefore close to the correct set of matches M^{c^*}, if it contains many elements of M^{c^*} and only few or no incorrect matches. The data association problem can then be formulated as follows:

$$M^* \;=\; \underset{M \in \mathcal{M}}{\operatorname{argmax}}\, c^*\,(M)$$

Since this correct data association M^{c^*} is unknown, we cannot directly compare the sets of data association matches $M \in \mathcal{M}$ to the correct data association M^{c^*}. Therefore, one of the major challenges of any data association approach is to find a good approximation or model of c^* that can reliably measure, if a certain set of data association matches M is close to the correct data association M^{c^*} without knowing M^{c^*}. Throughout our thesis, we call this model the *Target function* t.

Many techniques in the literature additionally also constrain the number of sets of data association matches that are considered to find the correct data association. This not only notably reduces the complexity of the search space $\mathcal{M}$, but it can also be used to shift the responsibility of finding the correct data association from the Target function t to the constraints by filtering out all incorrect matches. In this thesis, we call this subset of data association matches that fulfill certain constrains and are therefore considered for data association the *valid* set of matches $\mathcal{M}^{\text{valid}} \subset \mathcal{M}$.

A general approach for solving the data association problem can then be formulated as follows:

$$M^* \approx \underset{M \in \mathcal{M}^{\text{valid}} \subset \mathcal{M}}{\operatorname{argmax}} t(M)$$

In practice, both the constrains that define $\mathcal{M}^{\text{valid}}$ as well as the definition of the Target function t, are highly dependent on the application scenario. Therefore, there is no one solution to the general data association problem, which is why there are many different data association approaches tackling different challenges in the literature. In the following, we discuss different data association techniques presented in the literature and extract different properties and principles used by these techniques, so that we can more easily refer to them in this thesis.

2.2.2 Data Association Techniques

There are many different data association approaches found in the literature, each solving the data association problem for a specific scenario. However, some of the ideas presented in those approaches can be formulated into general data association principles, which can be useful tools for designing novel data association techniques for different scenarios such as associating detections of crops or crop rows on an agricultural field. In the following, we briefly introduce some of the more common data association techniques, explain how they translate into our formalization of the data association problem and name the concepts used in these techniques. We use some of these concepts to develop the novel data association techniques presented in this thesis. For a better overview, we distinguish between concepts that constrain the set of valid data associations, called *hard requirements*, and concepts for modeling the Target function, called *soft criteria*.

Since we are investigating data association for the purpose of associating indistinguishable crop features, we focus on data association techniques that are based on geometric information provided by the observed and mapped features. The main assumption of all geometry-based data association techniques is that a set of data association matches

M is close to the correct data association M^{c^*}, if it *explains the data well*. Defining what exactly this means is highly dependent on the application scenario and depending on the complexity of the features and their distribution also quite challenging. In general, sparse feature distributions are more likely to show unique geometric patterns and therefore usually require less intricate data association techniques. More densely and uniformly distributions have an increased chance of forming ambiguous patterns that are not easily resolved and therefore require more complex data association techniques. In order to compare the observed with the mapped features, in geometry-based data associations the current pose estimate is often used to project the observed features according to the pose estimate into the reference frame of the mapped features. Therefore, the accuracy of the pose estimate also plays an important role in finding the correct data association. In the following introduction of data association techniques, unless otherwise stated, we assume that the observations have been projected using the current pose estimate in order to facilitate a direct geometrical comparison between observed and mapped features.

The Nearest Neighbor (NN) Data Association The Nearest Neighbor data association strategy is one of the most commonly used geometry-based data association techniques. The key idea of this strategy is that a set of data association matches M *explains the data well*, if the distance between individual matches is small. This results in a rather strong hard requirement for a set of data association matches to be valid, the *Nearest Neighbor* requirement: A set of matches M is only valid, if all of the matches $(o, m) \in M$ only match an observed feature $o \in F_o$ to its closest correspondence in the map, i. e., only if $m \in F_m$ is the nearest neighbor of o. This Nearest Neighbor requirement is based on the assumption that all features are independent and thus that the matched mapped feature for each observed feature can be determined independently. Also, the vanilla NN strategy always assigns a match to all observations, even if the nearest neighbor is not actually close to the observation. This results in only one valid set of data association matches M^{NN}, which is the solution to the data association problem according to the NN strategy. Modeling the Target function t can be neglected since there is only one valid set of data association matches, i. e., $\mathcal{M}^{\text{valid}} := \{M^{\text{NN}}\}$.

In order to improve the results of the NN strategy many different approaches have been presented in the literature. One variant of the NN strategy introduces a distance threshold ε to define the maximum allowed distance between matched features to account for outliers or spurious detections and therefore increase the overall robustness of the NN data association. Throughout the thesis, we therefore always use the NN data association together with a distance threshold ε.

$$
\begin{aligned}
M^{\text{NN}} &:= \left\{ \left(o, m^{\text{NN}}(o)\right) \in F_o \times F_m \mid \left\| o - m^{\text{NN}}(o) \right\| < \varepsilon \right\} \\
m^{\text{NN}}(o) &:= \operatorname*{argmin}_{m \in F_m} \|o - m\|
\end{aligned}
$$

While the basic idea of the NN strategy that the closest mapped feature is the correct data association is valid in many situations, it is not true in cases where the current pose

estimate is inaccurate. Similarly, while the independence assumption of the NN strategy allows efficient data association even on larger feature sets, it often prevents the NN strategy from finding a good solution in case of ambiguous feature distributions. Overall, the NN strategy is a powerful data association on large feature sets with unambiguous feature distributions that finds the correct data association, if the pose estimate is sufficiently accurate.

The Joint Compatibility Branch and Bound (JCBB) Data Association In more challenging environments where the pose estimate is not always sufficiently accurate or features might be more ambiguously distributed, more complex data association techniques are necessary. Therefore, many data association techniques presented in the literature drop the independence assumption and explicitly consider dependency between features, for example, in form of *geometrical relations* or *joint compatibility* and consider a wider range of valid sets of data association matches instead of only the closest mapped feature to account for inaccuracies in the pose estimate.

One example for such a data association technique is the probabilistic approach presented by Neira and Tardos [2001] called *Joint Compatibility Branch and Bound* (JCBB) data association. The key idea of this approach is that, if the pose estimate as well as the observed and mapped features are modeled in a probabilistic framework, the dependency between features is also captured in this model. Based on this idea, the authors present a branch and bound technique that iteratively constructs the solution to the data association problem by efficiently searching a tree representation of all possible data association matches $\mathcal{M}$. Each node of the tree represents a different set of data association matches $M \in \mathcal{M}$. The root of the tree is the empty set. At the i-th depth the tree branches by matching one observation $o_i \in F_o$ to all possible mapped features F_m including one branch for not matching the observation. For example, the nodes at a depth of 1 represent all singleton sets of matches M that associate the first observation $o_1 \in F_o$ with one of the mapped features. The sets of matches M at depth 2 then contain two matches each, where the second element matches the second observation o_2.

Since performing a full search on this tree is computationally quite demanding, the authors propose the *joint compatibility* test jc, that, due to its iterative definition, allows to prune the search tree to a manageable size for many application scenarios with a reasonable amount of observed and mapped features. This joint compatibility test leverages the information contained in the probabilistic representation of the localization and mapping framework to determine whether the set of matches M at a certain node in the tree are all jointly compatible. If they are not compatible, the tree is pruned at this node and expanded otherwise. In the JCBB technique, passing the *joint compatibility* test is a hard requirement, since only these sets of matches are considered. Therefore, the set of valid matches $\mathcal{M}^{\text{valid}}$ is equal to all sets of jointly compatible matches.

When performing the tree search, the JCBB method keeps track of the best data association M found so far. Due to the strong hard requirement of *joint compatibility*, similarly to the NN strategy, all valid sets of matches $M \in \mathcal{M}^{\text{valid}}$ already *explain the data well*. Therefore, the authors define the best data association as simply the one with the most

matches. Thus, the Target function t of the JCBB method is defined by the soft criterion of *counting the number of matches*, leading to the following formulation of the JCBB method in our framework:

$$\begin{aligned} \mathcal{M}^{\text{valid}} &:= \{M \in \mathcal{M} \mid \text{jc}\,(M)\} \\ t\,(M) &:= |M| \\ M^{\text{JCBB}} &= \underset{M \in \mathcal{M}^{\text{valid}}}{\text{argmax}}\, t\,(M) \end{aligned}$$

In the publication by Neira et al. [2003] the authors also introduce a variant of the JCBB based on *geometric constraints* and thus called Geometric Constraints Branch and Bound (GCBB). This variant uses a geometric constraints test gc consisting of unary and binary geometric constraints to model geometric dependencies between features to prune the tree during the branch and bound search. In order to ensure joint compatibility of the considered sets of matches, the joint compatibility test is also performed whenever the algorithm reaches a leaf node of the tree. Formulating geometric constrains to define the valid set of data association matches is an interesting concept also used by other data association approaches found in the literature, for example in the work by Bailey et al. [2000]. In the context of our data association definitions, we call this concept the hard requirement of *geometric consistency*.

Another concept presented by Neira et al. [2003] is the principle of *locality*. The key idea of locality is that the number of valid sets of matches $\mathcal{M}^{\text{valid}}$ can be further reduced by only considering mapped features that are close enough to the current pose estimate so that they could have been observed. This notably reduces the number of mapped features considered during data association without affecting the result of the data association method, since mapped features that are far away would be anyways discarded because they always result in geometrically inconsistent or not jointly compatible matches. Using the hard requirement of *locality* is especially advantageous on maps that are notably larger than the feature detection range. Note that the local subset of considered mapped features depends on the current pose estimate. Thus, the main precondition for the principle of *locality* is that the current pose estimate is accurate up to a certain limit and therefore that the true pose of the vehicle is within these limits. If the current pose estimate is not accurate enough, the local subset of considered mapped features might not contain the detected features, which can result in incorrect data association. Using this observation, we can re-formulate the hard requirement of *locality* as follows: Only consider data association matches that result in a corrected pose estimate that is reasonably close to the current pose estimate, i. e., within the limits of the accuracy of the current pose estimate.

Positive and Negative Information Another common concept in feature-based localization and mapping is *positive* and *negative* evidence mentioned, for example, by Montemerlo et al. [2003]. The authors use the term *positive* evidence to describe features that have been observed, since this observation provides *positive* evidence of their existence. The counterpart to this are features that have not been observed although they are within

the *detection range* and therefore should have been detected. This "missing detection" hints towards the fact that this feature might actually not exist in the environment and therefore provides *negative* evidence. In the publication by Montemerlo et al. [2003] this information is used for feature management in the presented simultaneous localization and mapping approach, i. e., to decide whether an observed feature should be added to the map or not depending on the evidence. However, the idea of considering both, *positive* as well as *negative* evidence can also be beneficial to resolve ambiguous situations during data association. Many data association approaches found in the literature focus solely on how well the observed features are overlaid onto the mapped features to model the Target function t using only the *Positive Information* criterion such as, for example, the NN strategy as well as the JCBB and GCBB algorithms. Additionally, also considering *Negative Information* by penalizing sets of data association matches that leave mapped features within the *detection range* unmatched can improve the discriminative ability of a Target function t. This additional *Negative Information* is especially helpful in ambiguous situations where it is hard to determine the correct data association based on *Positive Information* alone. Depending on the application scenario, modeling the Target function t according to the *Positive Information* as well as the *Negative Information* criterion should be considered.

In summary, different data association techniques use different hard requirements to constrain the search space of all possible sets of data association matches $\mathcal{M}$ to a manageable subset of valid sets of matches $\mathcal{M}^{\text{valid}} \subset \mathcal{M}$. Constraining the number of considered data association matches not only increases computational efficiency of the data association method, but it is also used to create a small subset of high quality data association candidates. Depending on the application scenario different or even opposite hard requirements might be applicable. Based on the NN, the JCBB and the GCBB data association techniques, we defined the *Nearest Neighbor*, the *joint compatibility* as well as the *geometric consistency* requirements.

Similarly, different data association methods use different soft criteria to measure how well a set of matches *explains the data*. While the NN strategy is based on minimizing the distance between individual matches, the JCBB and GCBB focus on maximizing the number of matched features. Therefore, these techniques determine how close a set of matches is to the correct data association solely based on the soft criterion of *Positive Information*. Additionally considering the soft criterion of *Negative Information* to obtain a more discriminative Target function can improve data association results in ambiguous situations.

2.2.3 Accuracy of a Data Association Algorithm

In order to evaluate the performance of a data association algorithm, information about the correct data association is required. In an experimental evaluation, this information can be obtained by manually determining which detection originates from which object in the environment. To define evaluation methods on data association algorithms, we assume

that this ground truth information is available in form of a set of *ground truth* (GT) data association matches M^{GT}.

Our main observation is that localization or mapping techniques usually correct the pose estimate by rotating and translating the current pose estimate according to an isometric transform $T \in \mathcal{T}$ that can be described by a rotation parameter θ and a translation parameter t. The pose correction inferred by a set of data association matches $M \in \mathcal{M}$ can thus be described as the transform $T_M \in \mathcal{T}$ that overlays the projected observed features onto the corresponding mapped features according to the matches contained in M. Mathematically, we define this *inferred pose correction* T_M for any set of data association matches $M \in \mathcal{M}$ as follows:

$$T_M \;:=\; \underset{T \in \mathcal{T}}{\operatorname{argmin}} \frac{1}{|M|} \sum_{(o,m) \in M} \|T\,(o) - m\|^2$$

For point-based feature sets F_o and F_m this transform can be easily computed using *Principal Component Analysis*. The main advantage of this formulation is that it enables comparing the effect of different sets of data association matches on the pose estimate independent of the underlying localization or mapping application. To obtain a measure of accuracy, we use the GT data association matches M^{GT} to determine a *GT inferred pose correction* $T_{M^{\text{GT}}}$. Given the solution $M^* \in \mathcal{M}$ of the investigated data association algorithm with corresponding inferred pose correction T_{M^*}, we compute the pose difference ΔT between both transforms:

$$\Delta T \;:=\; T_{M^{\text{GT}}}^{-1} \cdot T_{M^*}$$

Intuitively, this transform ΔT describes how much the pose correction inferred by the data association matches M^* of the investigated algorithm differs from the pose correction inferred by the GT data association matches M^{GT}. The rotation parameter $\Delta\theta$ of ΔT measures the difference in the heading of the inferred pose correction and the translational parameter Δt of ΔT measures the difference in translation of the inferred pose correction. For our evaluation we therefore use the parameters $\Delta\theta$ and Δt of the difference transform ΔT to measure the heading and translational accuracy of M^* with respect to M^{GT}.

Since the actual amount of pose correction applied to the pose estimate might be different depending on the underlying localization or mapping technique, the accuracy measures obtained with this technique can only provide an estimate for the attainable accuracy of a localization or mapping algorithm. Nevertheless, this technique provides a method to compute an accuracy measure for data association algorithms based on their expected effect on the pose estimate. We will use this measure in this thesis to evaluate the accuracy of data association algorithms.

Chapter 3

Robust Crop Row Detection

Robust perception and detection of objects in the environment is an integral part of any autonomous navigation framework. This is especially true for autonomous navigation applications in agriculture. When an agricultural vehicle traverses a crop field, there is usually only a few centimeters clearance between the wheels of the vehicle and the value crop of the field. Therefore, an accurate heading and lateral position estimate of the vehicle is crucial to guide this vehicle along the crop rows of the field without damaging the crops. This is usually facilitated by extracting crop rows from data captured by sensors that are mounted in front of the vehicle. For such a crop row detection technique to be useful in real-world agricultural autonomous navigation applications, a wide application range, i. e., detecting crop rows on different crop types and at different growth stages and adaptability to various hardware specifications, is crucial. Furthermore, robust detections throughout the entire field, i. e., not only while driving in-row but also during transition maneuvers at the headlands, are essential to enable full autonomous navigation. In this chapter, we present two novel crop row detection approaches that achieve these goals. Our key idea towards more robust crop row detection, especially during transition maneuvers, is to consider all available data to detect all visible crop rows jointly. To facilitate this joint detection, we propose to model the crop rows as a set of parallel and equidistant lines, that we call a Pattern. In extensive experiments on real-world data captured from five different crop fields we confirm that both our approaches are well suited to provide robust and accurate crop row detections for autonomous navigation throughout entire crop fields with a wide application range across different crop types and sizes as well as sensor modalities and sensor mounting positions. Additionally, we employed one of our approaches successfully in our autonomous navigation framework on the BoniRob, achieving fully autonomous traversal of entire crop fields.

3.1 Introduction

A successful autonomous navigation framework is usually based on a reliable perception module that robustly detects objects of interest in the environment. For autonomous navigation in agriculture this means that a reliable detection of the value crops is crucial to guide the vehicle while it traverses the crop field without damaging the crops. More precisely, an accurate pose estimate of the vehicle, and especially its wheels, relative to the value crop is mandatory. Even small errors in the heading or lateral position estimate of the vehicle can cause damage to the crops, since there is not much free space between the wheels of the vehicle and adjacent crops as also shown in Figure 1.3. This is usually facilitated by extracting the crop row structure from data perceived by sensors that are mounted in front of the vehicle. This row information is then used to accurately guide the vehicle along the crop rows of the field.

For fully autonomous navigation the vehicle should traverse the entire crop field autonomously including driving *in-row*, i. e., while it traverses the field aligned with the crop rows, as well as *transition* maneuvers at the headlands such as entering the field, turning at the end of the field and re-entering the field after turning or leaving the field. Therefore, the employed crop row detection algorithm needs to be reliable not only while the vehicle is driving *in-row*, but it also needs to be reliable during *transition* maneuvers, where detecting the crop row structure from the perceived sensor data is usually more challenging. We show example images captured during transition maneuvers (bottom) and while the vehicle is in-row (top) in Figure 3.1. These example images illustrate that detecting crop rows during these transition maneuvers is more challenging than in-row for three reasons:

First, during transition crop rows are only partially visible in the sensor data, since the vehicle is leaving or entering the field as can be seen in the image on the bottom left of Figure 3.1. Therefore, less information about the perceived row structure is available to detect the crop rows. Second, the heading of the vehicle is not necessarily aligned with the crop rows since it might be turning towards or away from the crop rows as shown in the image on the bottom mid of Figure 3.1. A crop row detection algorithm for fully autonomous navigation therefore needs to be able to extract crop rows that are perceived at an arbitrary angle in the sensor data. Third, since the vehicle is leaving or entering the field, less crop row structure is visible and more area of the headlands surrounding the traversed field is perceived instead. It is not uncommon for these headlands to contain wild growing vegetation, such as grass, weed or even bushes and trees as shown in the image on the bottom right of Figure 3.1. This additional vegetation causes a large amount of non-crop vegetation feature detections in the perceived sensor data so that the crop row structure is harder to detect. A reliable crop row detection therefore also needs to be robust against large amounts of non-crop vegetation feature detections in the input data.

Another important requirement for a crop row detection algorithm intended for an autonomous navigation framework is that it should not be tailored to specific scenarios, but have a wide application range instead to be useful in real-world applications. This is especially true in agriculture, where vehicles – autonomously or manually steered – are quite expensive and not invested in lightly. A wide application range of the employed crop row

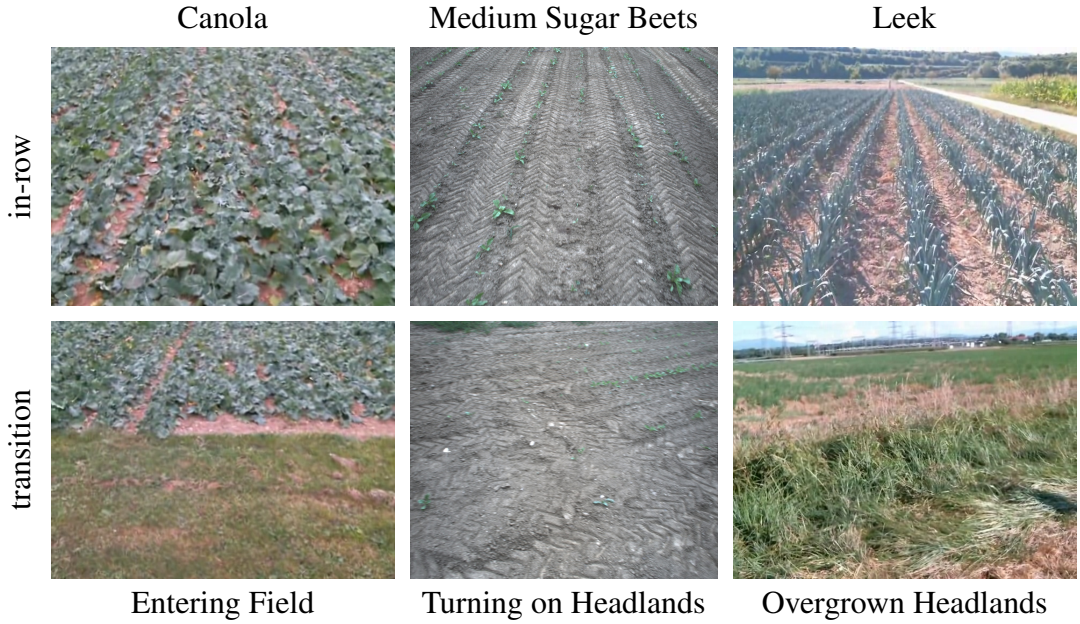

Figure 3.1: This figure shows examples for image data perceived while traversing the field in-row (top) and during transition (bottom). When traversing the field in-row, the crop rows are visible throughout the entire image, the rows are aligned with the heading of the vehicle and only little vegetation that disturbs the crop row structure, such as weed, is present. During transition, crop rows are only partially visible (bottom left), not necessarily aligned with the heading of the vehicle (bottom mid) and a lot of wild growing vegetation that disturbs the crop row structure is visible on the headlands (bottom right).

detection algorithm is therefore a key requirement. More specifically, this includes detecting crop rows on various crop types, at different growth stages, different sensor modalities and sensor mounting positions to facilitate autonomous navigation on a multitude of crop fields. We show examples for different crop types at different growth stages as well as data obtained from an RGB camera and a LIDAR sensor in Figure 3.2.

In this chapter we present a novel approach for crop row detection on agricultural fields that is reliable and accurate during the entire traversal of the field, explicitly including data perceived during transition maneuvers. While providing robust crop row detections throughout the entire field, our novel crop row detection technique is also aimed at a wide application range for a flexible autonomous navigation framework with a multitude of real-world applications. These goals impose the following challenges on our crop row detection approach:

1. *Partial Row Structure*: During transition less row information is available, since the field and its crop rows are only partially observed.

2. *Arbitrary Crop Row Orientation*: The heading of the vehicle relative to the crop rows changes during turning maneuvers. Thus, the crop row detection method can not assume that the crop rows are always oriented at a fixed angle in the sensor data.

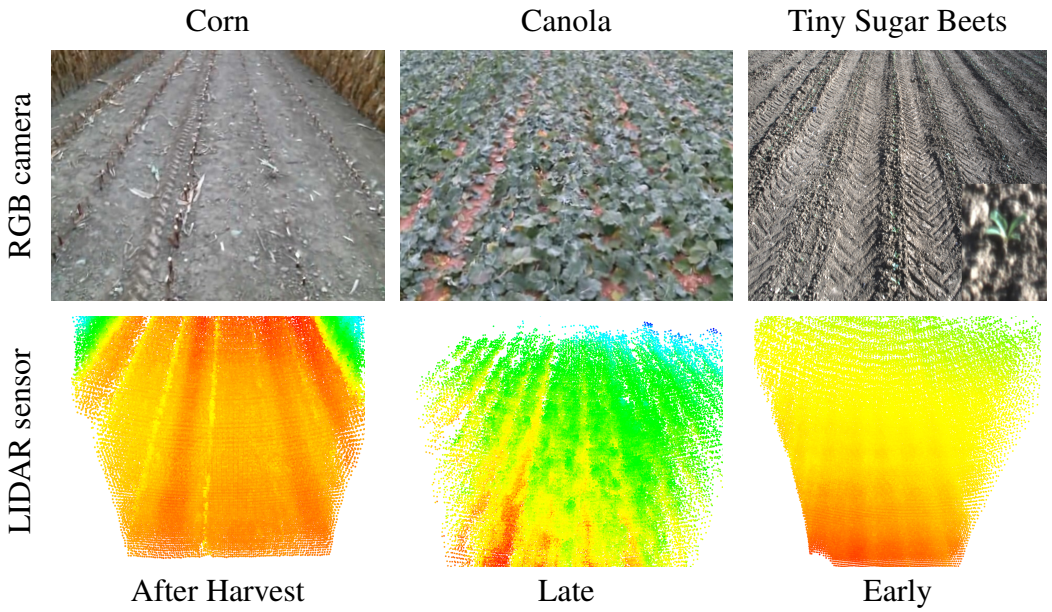

Figure 3.2: This figure shows examples of image (top) and LIDAR (bottom) data on different crop types at different growth stages. The point cloud is colored according to the height of the points above ground, i. e., their z-coordinate, where red indicates lower and green up to blue higher values. The overgrown canola plants (mid) cover the soil between the crop rows and therefore occlude the row structure in the image. The crop rows show more clearly in the LIDAR data (green clusters on yellow background). In contrast, with a high enough resolution, the much smaller sugar beet plants (right) are easy to detect using greenness information. Due to their small size of around 1 cm these tiny plants are hard to distinguish from ground in the LIDAR data.

3. *Large Amounts of Non-Crop Vegetation*: During transition the sensor data also captures the headlands, which possibly contains wild growing vegetation that can cause large amounts of noise.

4. *Wide Application Range*: For a wide application range, crop rows should be detected on a large variety of crop types at different growth stages. Additionally, the detection method should be adaptable to changes in hardware specifications and different sensor modalities.

For our approach to overcome these challenges our key idea is to develop a crop row detection technique that considers all available information to jointly extract all crop rows visible in the sensor data. A joint detection that considers all available data is by design robust to large amounts of noise as well as more accurate when only partial crop rows are perceived. To facilitate such a joint crop row detection on all available data, we propose to model the crop rows of the field as set of parallel and equidistant lines, called a *Pattern*. Using this representation of the crop rows, we introduce two novel approaches: Our first approach is based on the well known *Hough Transform*, that is often used to extract lines from image data [Hough, 1962]. Our second approach leverages the *random*

sample consensus (RANSAC) strategy to detect a crop row Pattern [Fischler and Bolles, 1981]. Based on these concepts, we design both our approaches so that they detect a Pattern of crop rows on all available data by estimating all required parameters, including the orientation of the crop rows, jointly. This ensures that our approaches can detect crop rows independent of their orientation in the sensor data. For a wide application range, our second key idea is to design our crop row detection approaches independent from the raw data perceived by the sensor. We achieve this independence through a segmentation step, in which we compute a vegetation feature map as described by Winterhalter et al. [2018]. This segmentation step is a powerful tool that can be easily adapted to different sensor modalities as well as other hardware specifications such as sensor mounting position and orientation. The resulting vegetation feature map is a two-dimensional grid map defined on the ground plane in front of the vehicle. Each cell in the grid map contains a weight, where values larger than 0 indicate the presence of vegetation at the location of the cell. An important property of these vegetation feature maps is that they recover the actual geometric relations between the crop rows independent of the sensor modality that was used to capture the raw data. For example, although crop rows do not appear parallel in image data, their Euclidean geometric relations are recovered during this segmentation step. The vegetation features therefore form parallel and equidistant lines in the resulting vegetation feature map. This requirement is crucial for our approach, since it is based on the assumption that a set of parallel and equidistant lines, i. e., a Pattern, can be extracted from the input data, the vegetation feature maps. However, while these vegetation feature maps ensure the adaptability of our approach to changing hardware specifications, crops of different types and different sizes still result in a wide variety of vegetation feature maps that display different vegetation feature distributions. We therefore perform our experimental evaluation on a large variety of different crops types and sizes, to confirm that our novel crop row detection can handle different crop types and sizes. Our key ideas and contributions towards a robust crop row detection approach can be summarized as follows:

1. *Detecting Crop Rows Jointly*: For a more robust crop row detection, especially during transition maneuvers, consider all available data and detect the crop rows jointly.

2. *Model Row Structure as Pattern*: For a joint detection of crop rows, we propose to model the crop row structure as a set of parallel and equidistant lines, which we call a *Pattern*.

3. *Two Novel Pattern Detection Approaches*: We present two novel approaches that use the definition of our crop row structure, the Pattern, to detect all crop rows visible in the sensor data jointly.

4. *Independence from Segmentation*: Our crop row detection approaches use the vegetation feature maps presented by Winterhalter et al. [2018] to segment the input data. This makes our approaches independent from the segmentation step allowing for a wide application range.

5. *Applicability for Autonomous Navigation*: We perform extensive experimental evaluation on real-world data featuring different types of crops at different growth stages using two different sensor modalities. The results confirm the suitability of our crop row detection approaches for a wide range of autonomous navigation applications on entire agricultural fields.

In the following sections, we first give an overview off state-of-the-art crop row detection approaches and discuss how they relate to our approach. In the third section, we present our two novel crop row detection approaches, including the definition of a Pattern as well as similarity measures between Patterns. In the fourth section, we evaluate our crop row detection approaches in-depth on five different real-world data sets. The results of this evaluation confirm that both our approaches are well suited for crop row detection in an autonomous navigation framework, since they yield reliable detection results on varying crop types and sizes not only in-row, but also during transition.

3.2 Related Work

In this section, we give an overview of state-of-the-art crop row detection approaches and discuss their relation to our crop row detection approach as well as how our approach contributes to the state of the art.

Crop row detection for navigation purposes has received much attention in past decades. Most crop row detection approaches rely on image data captured by a front facing camera to detect row structure of the crops on agricultural fields. Usually a vegetation segmentation step, that produces a segmented image similar to the feature maps that we use for our approach, is performed. The crop row structure is then detected on this preprocessed data. Some approaches as, for example, the method presented by Kise et al. [2005] incorporate additional depth information computed from a stereo camera setup for more robust results. In the work by Kise et al. [2005] this additional depth information is used to compute an elevation map. Given the spacing between crop rows and assuming that the crop rows are roughly aligned with the heading of the sensor four crop rows are extracted from the elevation map. Other approaches, such as the methods described by Søgaard and Olsen [2003] and Tillett and Hague [1999], purely rely on vision data and divide the image into a low number of horizontal strips, determine the center points of crop rows in each strip and connect these points to lines. The authors also present another method that computes a band-pass filter given the crop row spacing to create crop row templates. These templates are then matched against the image to detect the crop rows [Tillett et al., 2002]. These approaches not only assume a given row spacing, but also expect the crop rows to be roughly aligned with the vertical axis of the image. However, while template matching can also be used to estimate the row spacing and offset – even of curved – crop rows as the approach presented by Vidović et al. [2016] demonstrates, the approximately vertical orientation of the crop rows in the image data is still required. The crop row detection presented by Montalvo et al. [2012] has even stronger assumptions. Given the number of crop rows and their location in the image, regions of interest are determined

and a linear regression on these regions of interest is performed to obtain the individual line parameters of each crop row.

One of the most recent approaches by English et al. [2014] is based on an idea similar to the idea for our feature map: Before detecting the crop rows, the image is converted into a top-down view. On this image, the green values of each column in the image are summed and the variance over these sums is computed. After performing a skew operation on the image, this procedure is repeated. The skew that resulted in the highest variance then corresponds to the orientation of the crop rows. Peaks in the sums of each column denote centers of crop rows. This technique stands in contrast to the before mentioned approaches, as it not only detects the position of the crop rows in the image, but also estimates the orientation. English et al. extend their approach by employing a learning-based technique and also fusing additional depth information for more robust results [English et al., 2015]. The major drawback of this learning-based approach is that it is dependent on the data of the field it was trained on. As the authors state, on a new field an image with manually annotated crop rows as well as a training run over 20 seconds is required.

There are also more closely related crop row detection techniques that, similar to one of our approaches, are based on the Hough Transform [Hough, 1962]. Given the crop row spacing, these approaches either detect a single line or multiple lines [Åstrand and Baerveldt, 2005, Bakker et al., 2008, Leemans and Destain, 2006, Marchant, 1996]. The difference between these approaches is how they apply the Hough Transform. For example, the work by Åstrand and Baerveldt [2005] adds an additional parameter, the width of a line, so that neighboring pixels are also considered during line detection. This increases the robustness of the crop row detection on cluttered input data. The approach presented by Bakker et al. [2008] assumes that the number of crop rows perceived in the image data is a fixed number, namely three. Given also the crop row spacing in pixels, the image is split into three parts, where the width of these parts is equal to the provided spacing. Assuming parallel lines, these image parts are overlaid and corresponding pixel values accumulated. The Hough Transform is then applied on the resulting image. Again assuming parallel lines and a given spacing, the approaches presented by Leemans and Destain [2006] and Marchant [1996] compute the sum of the weights in the Hough Space accumulator over all parallel lines to detect parallel and equidistant crop rows. These approaches have similarities in terms of assumptions with our approach, since they require neither precise knowledge of the number of perceived crop rows nor the orientation of the crop rows to be vertically aligned. In contrast to our approach, they still require exact knowledge of the spacing between crop rows.

Overall, the main difference between these Hough-based approaches and our Hough-based crop row detection is that they all employ the classical Hough Transform by Hough [1962], which is based on the model of one single line to detect crop rows. In contrast, our Hough-based approach is inspired by the generalized Hough Transform as presented by Ballard [1981], since we adapt the model estimated by the Hough Transform to directly fit the desired crop row structure of parallel and equidistant lines, i. e., to fit our Pattern. Our assumptions for this model, i. e., that crop rows can be represented as a set of parallel and equidistant lines, are often also found in the literature. Since many approaches, especially the more closely related Hough-based approaches, impose similar assumptions on the

geometric structure of crop rows, our decision to model the crop rows as a Pattern is well supported.

So far, research in this area has been mainly focused on detecting crop rows from image data, using depth information only rarely [English et al., 2015, Kise et al., 2005]. In contrast, our approach is explicitly designed to be adaptable to different sensor modalities, since it uses the feature maps as presented by Winterhalter et al. [2018] to segment the perceived raw data. Independence of the segmentation step also enables our approach to detect crop rows on a large variety of different crop types and sizes. This allows for a flexible detection of crop rows on different crop fields without the need of additional labeling and retraining for every crop field [English et al., 2015]. We showcase this adaptability in our experimental evaluation, where we detect crop rows from data captured with an RGB camera as well as data perceived by a LIDAR sensor on different crop types in different growth stages. Our approach therefore goes beyond state-of-the-art approaches as it is adaptable to different sensor modalities and not tailored to one specific modality, thus allowing for a wide application range.

Our main contribution to the state of the art is that we design our approach to robustly detect crop rows not only while driving in-row, but also during transition maneuvers. We achieve this flexibility by making only two assumptions: First, we assume that the crops are distributed in parallel and equidistant rows. Second, we assume that a rough range of the crop row spacing is known. In contrast other approaches are based on assumptions that only hold while the vehicle is driving in-row. By design these approaches cannot detect crop rows during transition maneuvers.

For example, a common assumption is that the crop rows are always roughly found along the vertical axis of the image [Bakker et al., 2008, Kise et al., 2005, Søgaard and Olsen, 2003, Tillett and Hague, 1999, Tillett et al., 2002, Vidović et al., 2016], which constrains the heading of the vehicle to be always roughly aligned with the crop rows. As discussed in the previous section, this is not an issue while driving only in-row. However, during transition, the heading of the vehicle is not aligned with the orientation of the crop rows. These approaches are therefore not suited to detect crop rows during transition maneuvers.

Another common assumption is that the number of crop rows visible in the sensor data is always the same [Bakker et al., 2008, Kise et al., 2005, Montalvo et al., 2012]. These techniques are therefore – by design – constrained to always detect a fixed number of crop rows from the image data. However, during transition maneuvers the number of perceived crop rows is not always constant, which makes approaches based on this assumption unsuited for crop row detection during transition maneuvers. Additionally, this assumption also limits the transferability of the technique to other agricultural vehicles and crop fields with a different number of crop rows in the field of view due to a different spacing between the crop rows. These limitations can only be overcome by either adapting the technique – if possible – to detect a varying number of crop rows depending on the vehicle and crop field it should be used on, or changing the mounting position of the sensor on the vehicle to match the expected number of crop rows in the field of view of the sensor.

Another assumption is that the exact spacing between crop rows is known a priori [Leemans and Destain, 2006, Marchant, 1996, Tillett et al., 2002]. In contrast to the previous

assumptions, having a fixed crop row spacing is a feasible assumption since it is independent of the hardware configuration and the orientation of the vehicle. Furthermore, in practice this parameter can be easily determined in advance as it directly correlates to the type of crop growing on the field. However, the actual spacing between individual crop rows on the field can still vary since the crops are not always sown with perfectly constant spacing. Therefore, estimating the crop row spacing based on a rough range of the expected crop row spacing should yield more robust results than depending on an a priori fixed spacing value.

Overall, due to fewer assumptions and feasible choice of parameters, our crop row detection approaches contribute to state-of-the-art research towards robust and adaptable crop row detection for navigation across entire fields.

3.3 Detecting Crop Rows as a Pattern

For autonomous navigation on agricultural fields, an accurate pose estimate of the vehicle is required to prevent the vehicle from accidentally crushing value crops. More precisely, the localization algorithm of the navigation framework needs to provide precise heading and lateral pose information. This is usually realized by detecting the crop row structure of the field to correct the pose estimate relative to the crop rows. In this section, we present our approach for reliable and accurate detection of the crop row structure, which is crucial for precise navigation along agricultural fields.

For our approach to be independent of the sensor modality we assume that the sensor data has been segmented into a two-dimensional vegetation feature grid map as presented by Winterhalter et al. [2018]. This vegetation feature grid map encodes information about the presence of vegetation in each cell, where values larger than 0 imply that vegetation was perceived at the location indicated by the corresponding cell. We show examples for feature maps created from different sensor modalities from real-world data in Figure 3.3. A key requirement for this feature map is that it describes the location of vegetation features in Euclidean, i. e., metrical, space and therefore displays the undistorted, actual, geometric distribution of vegetation features. For vegetation features obtained from image data, this means that the raw data obtained from the sensor has to be transformed and projected accordingly. As our examples in Figure 3.3 show, this ensures that the vegetation features in the grid map appear true to their actual geometric distribution as parallel and equidistant rows. For crop row detection, we are only interested in the cells c of the feature map that contain a vegetation feature, i. e., cells c with a weight larger than 0. In the following, we therefore only consider the set of feature cells $\mathcal{C}$ that contains cells c with a weight larger than 0.

For reliable and accurate detection of crop rows, our key idea is to extract all rows visible in the feature map jointly. To implement this idea, we derive a mathematical formulation for the set of crop rows visible in the feature map based on the observation that crop rows are usually sown in parallel and equidistant lines. Leveraging that these crop rows also appear as a set of parallel and equidistant distributed lines in the feature map, we define a mathematical model of this set of lines, called a *Pattern*. We also

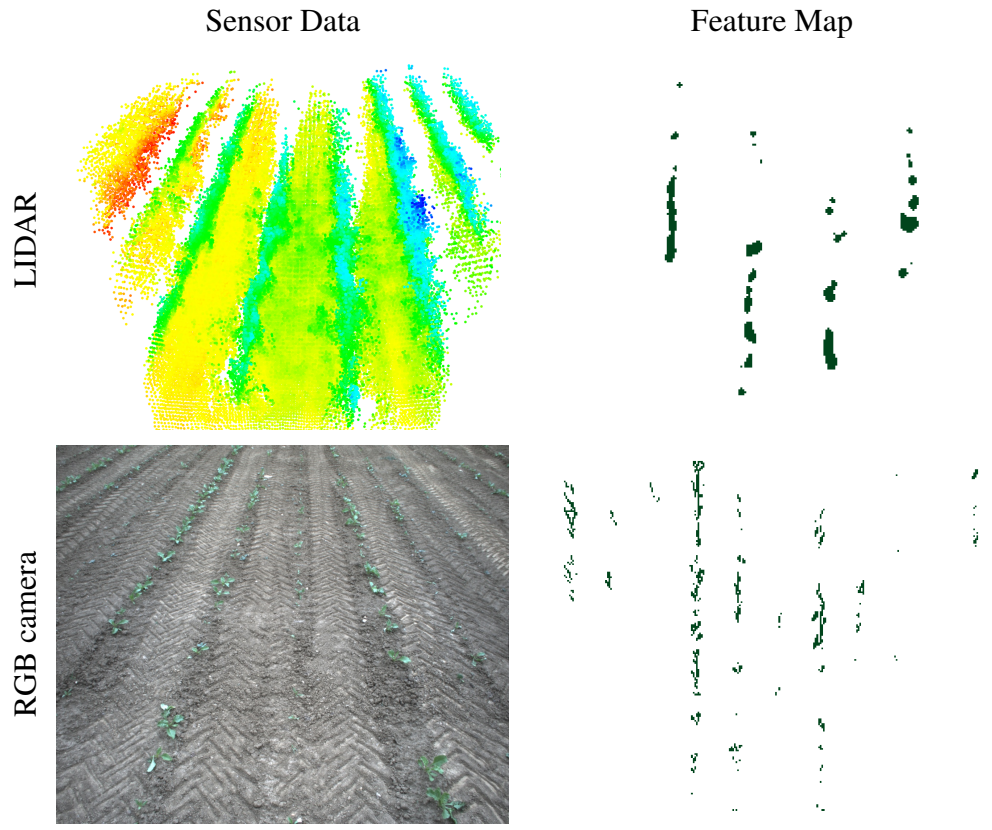

Figure 3.3: This figure shows the information obtained by different sensor modalities (left) and the corresponding vegetation feature maps (right). The point cloud is colored according to the height of the points above ground, i. e., their z-coordinate, where red indicates lower and green up to blue higher values. The cells of the feature map that contain vegetation are highlighted in green. Note that the vegetation features form parallel lines in both feature maps.

discuss how measures of similarity can be computed between two Patterns and use the presented similarity measures to determine the robustness and accuracy of the results of our approach in our experimental evaluation. Based on this representation of crop rows as a Pattern, we then present two approaches that extract a Pattern of crop rows from feature maps.

3.3.1 Pattern - a Mathematical Model for Crop Rows

In this section, we present our mathematical model for a set of parallel and equidistant lines. To this end, we first define a mathematical model of an individual line. Based on the model for an individual line, we then present our definition of a *Pattern* and explain how we measure similarity between two Patterns.

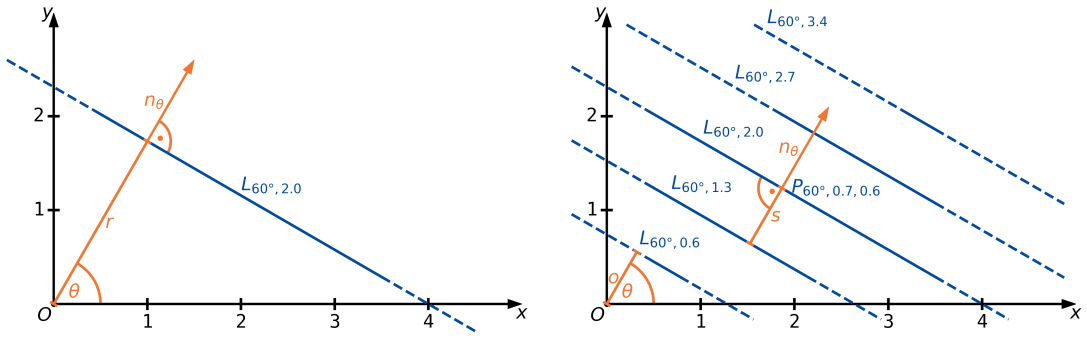

Figure 3.4: This figure shows illustrations for the Line (left) and Pattern models (right). The line $L_{\theta,r}$ (blue) has an angle parameter of $\theta = 60°$ and an offset parameter $r = 2.0\,\text{m}$. The Pattern $P_{\theta,s,o}$ has the same angle parameter $\theta = 60°$, a spacing of $s = 0.7\,\text{m}$ and an offset of $o = 0.6\,\text{m}$. The parameters that define the models are annotated in orange.

Line In the two-dimensional Euclidean space $\mathbb{R}^2$, we define a line $L_{\theta,r}$ using two parameters: the *angle* θ of the normal vector, i. e., the vector that points in the direction that is orthogonal to the direction of the line, and the *offset* r of the line to the origin, i. e., the Euclidean distance between the origin and the point on the line that is closest to the origin. This line is then mathematically described using its Hesse normal form:

$$L_{\theta,r} \quad := \quad \left\{(x,y) \in \mathbb{R}^2 \mid r = x \cdot \cos(\theta) + y \cdot \sin(\theta)\right\}. \tag{3.1}$$

We show an example illustration on how the angle θ and offset r define a line $L_{\theta,r}$ on the left of Figure 3.4. The Hesse normal form of a line $L_{\theta,r}$ can also be used to compute the signed distance $d(L_{\theta,r}, p)$ of a point $p = (p_x, p_y) \in \mathbb{R}^2$ to the line $L_{\theta,r}$:

$$d(L_{\theta,r}, p) \quad := \quad [p_x \cdot \cos(\theta) + p_y \cdot \sin(\theta)] - r. \tag{3.2}$$

Pattern We use this line representation to define the Pattern P as a set of parallel and equidistant lines in the two-dimensional Euclidean space $\mathbb{R}^2$. Such a Pattern is defined by three parameters: the *angle* θ of the pattern, the *spacing* s of the pattern and the offset o of the pattern:

$$P_{\theta,s,o} \quad := \quad \{L_{\theta,r} \mid \exists\, n \in \mathbb{Z} \colon r = n \cdot s + o\} \tag{3.3}$$

Since all lines of the pattern are parallel, they all share the same angle parameter θ. Therefore, the angle of the pattern θ describes the direction that is orthogonal to the direction of the set of parallel lines. Since all lines of the Pattern are equidistant, the distance between adjacent lines, i. e., the spacing s, is constant and can therefore be used to describe the Pattern. The third parameter, the offset o, is analogue to the offset parameter r of an individual line and describes the offset of the Pattern to the origin. An illustration of a Pattern is shown in Figure 3.4.

Similarity between Patterns Since a Pattern is described by three parameters, we can define a similarity based on these three parameters. Therefore, two Patterns are similar, if their corresponding parameter values are similar. This yields three measures that should be considered when comparing two Patterns: the *angular difference* Δ_θ, the *spacing difference* Δ_s and the *lateral difference* Δ_{lat}. Given two Patterns P_1 and P_2 with corresponding parameters, the definition of the first two measures is straightforward:

$$\Delta_\theta(P_1, P_2) := |\theta_1 - \theta_2| \tag{3.4}$$
$$\Delta_s(P_1, P_2) := |s_1 - s_2| \tag{3.5}$$

We normalize the angular difference Δ_θ to the range $[0, \pi)$. However, the straightforward definition for the lateral difference $\Delta_{\text{lat}}(P_1, P_2) = |o_1 - o_2|$, is a priori not well defined since the offset parameter o of the pattern is not unique: The offset of the pattern o can describe the offset of any line in the Pattern. For example, for a Pattern $P_{\theta,s,o}$ using $o' = s + o$ yields the same set of lines, i. e., $P_{\theta,s,o} = P_{\theta,s,o'}$. To compute the lateral difference, we therefore use the offset parameter r^* of the line closest to the origin. This yields the following lateral difference measure:

$$\Delta_{\text{lat}}(P_1, P_2) := |r_1^* - r_2^*| \tag{3.6}$$
$$L_{\theta,r^*} := \underset{L_{\theta,r} \in P_{\theta,s,o}}{\operatorname{argmin}} |r| \tag{3.7}$$

This lateral similarity measure computes the lateral distance relative to the origin since it uses the line offset parameters r that are the signed distance of the line from the origin. However, when using line features for pose correction, the lateral distance between the lines relative to a *reference point* p_{ref} in front of the vehicle is more relevant. We illustrate this idea in Figure 3.5, where we show an example situation of a vehicle and two Patterns. We highlight the lateral similarity measure relative to the origin Δ_{lat} (bottom left of the image) and the similarity measure $\Delta_{\text{lat}}^{p_{\text{ref}}}$ relative to a reference point p_{ref} in front of the vehicle. This example shows how different reference points p_{ref} affect the resulting lateral difference value. In order to evaluate the lateral difference between Patterns relative to a reference point p_{ref}, we define a more general lateral difference measure $\Delta_{\text{lat}}^{p_{\text{ref}}}$ that depends on the reference point p_{ref}:

$$\Delta_{\text{lat}}^{p_{\text{ref}}}(P_1, P_2) := |d(P_1, p_{\text{ref}}) - d(P_2, p_{\text{ref}})| \tag{3.8}$$
$$d(P_{\theta,s,o}, p_{\text{ref}}) := d\left(L_{\theta,r^*}^{p_{\text{ref}}}, p_{\text{ref}}\right) \tag{3.9}$$
$$L_{\theta,r^*}^{p_{\text{ref}}} := \underset{L_{\theta,r} \in P_{\theta,s,o}}{\operatorname{argmin}} |d(L_{\theta,r}, p_{\text{ref}})| \tag{3.10}$$

Here, the lateral difference of two patterns to a reference point p_{ref} is defined using the signed distances $d\left(L_{\theta_1,r_1^*}, p_{\text{ref}}\right)$ and $d\left(L_{\theta_2,r_2^*}, p_{\text{ref}}\right)$ of the lines L_{θ_1,r_1^*} and L_{θ_2,r_2^*} of each pattern, that are closest to the reference point. In the following, we show that this is indeed a generalization of Δ_{lat}, since we obtain Δ_{lat} from $\Delta_{\text{lat}}^{p_{\text{ref}}}$ by using the origin O as reference point p_{ref}, i. e., we need to proof that $\Delta_{\text{lat}} = \Delta_{\text{lat}}^{O}$.

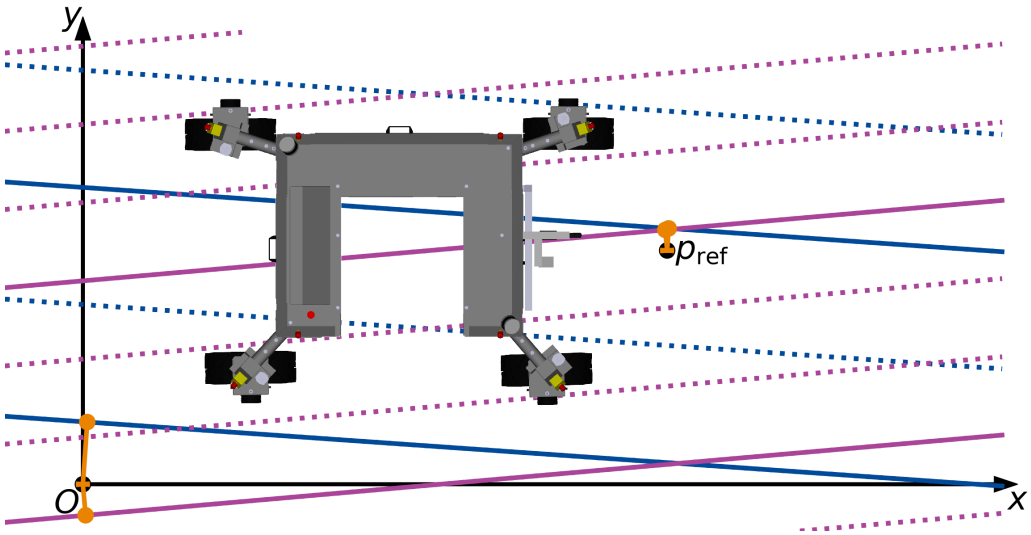

Figure 3.5: This figure illustrates the lateral difference measure between two Patterns (purple and blue dashed lines). The Pattern lines that are used to compute the lateral difference measure are drawn as solid lines. We annotate the lateral distance using orange lines and points. The simplified difference Δ_{lat}, where the reference point is equal to the origin O is located at the bottom left of the image. We also show the generalized difference $\Delta_{\text{lat}}^{p_{\text{ref}}}$ to a reference point p_{ref} located in front of the vehicle.

Proof. The key observation is that using the origin O as reference point means that the signed distance between a line $L_{\theta,r}$ and the reference point $p_{\text{ref}} = O$ is equal to the negative line offset $-r$ per definition, and consequently, the absolute distance is equal to the absolute value of the line offset r:

$$d\left(L_{\theta,r}, O\right) \overset{Eq.\ (3.2)}{=} \left[0 \cdot \cos\left(\theta\right) + 0 \cdot \sin\left(\theta\right)\right] - r = -r \tag{3.11}$$

$$\left|d\left(L_{\theta,r}, O\right)\right| = \left|r\right| \tag{3.12}$$

Therefore L_{θ,r^*} from Eq. (3.7) and $L_{\theta,r^*}^{p_{\text{ref}}}$ Eq. (3.10) are the same for $p_{\text{ref}} = O$ and it follows that:

$$\begin{aligned}
\Delta_{\text{lat}}^{O}\left(P_1, P_2\right) &\overset{Eq.\ (3.8)}{=} \left|d\left(P_1, O\right) - d\left(P_2, O\right)\right| \\
&\overset{Eq.\ (3.9)}{=} \left|d\left(L_{\theta_1,r_1^*}, O\right) - d\left(L_{\theta_2,r_2^*}, O\right)\right| \\
&\overset{Eq.\ (3.11)}{=} \left|-r_1^* - \left(-r_2^*\right)\right| \\
&= \left|r_1^* - r_2^*\right| \\
&\overset{Eq.\ (3.6)}{=} \Delta_{\text{lat}}\left(P_1, P_2\right)
\end{aligned}$$

□

In summary, we introduced our mathematical model for sets of parallel and equidistant lines, the Pattern, and explained how we measure similarity between Patterns with respect to all three parameters of the model. In the next section, we present our two crop row detections approaches that are based on our Pattern representation and therefore are able to extract all visible crop rows jointly.

3.3.2 Extracting Crop Row Patterns

In this section, we present two different approaches to extract a Pattern of crop rows from a feature map. The first approach is based on the well known Hough Transform. The key advantage of this approach is that it determines the model that best fits the data of the feature map over all available Patterns. In our case, this means that the globally best Pattern for the given feature map is found. However, depending on the feature density in the provided feature map, finding the globally best Pattern can become computationally expensive. We therefore also present a second approach based on a *random sample consensus* (RANSAC) scheme, that can be stopped at any time if necessary to provide Pattern detections at a high frequency. For a better intuition on how our *Pattern Hough Transform* differs from the well known *Line Hough Transform*, we first explain the Line Hough Transform and its key ideas. We then present our variation of the Hough Transform that operates on Patterns instead of lines. Finally, we introduce a sampling-based RANSAC method to extract Patterns from feature maps at an adjustable rate.

Line Hough Transform Traditionally, the Hough Transform is well known in image processing applications, where it is used to detect lines in image data. This Line Hough Transform is based on the fact that lines in the two-dimensional Euclidean space can be represented by a two-dimensional parametrization over the angle θ and the offset r of the line using the Hesse normal form (see Eq. (3.1)). In order to obtain a unique parameterized representation for each line $L_{\theta,r}$, we define the Line Hough Space $\mathcal{H}_L$ as follows:

$$\mathcal{H}_L \;:=\; \{L_{\theta,r} \mid \theta \in [0,\pi) \,\wedge\, r \in \mathbb{R}\}\,. \tag{3.13}$$

This definition of the Line Hough Space $\mathcal{H}_L$ still contains all lines of the two-dimensional Euclidean space, but each line can only be represented by one unique pair of parameters. Thus, a unique mapping between the Line Hough Space to the parameter space is ensured. The power set of the Line Hough Space $\mathcal{P}(\mathcal{H}_L)$ then contains all possible sets of two-dimensional lines in the Euclidean space. The idea of the Line Hough Transform is that this parameterized representation of a line can be used to determine the set of lines $h_L(c) \in \mathcal{P}(\mathcal{H}_L)$ that pass through the same cell $c \in \mathcal{C}$ in a grid map, or the same pixel in an image. This mapping from a cell $c \in \mathcal{C}$ to a set of lines $h_L(c) \in \mathcal{P}(\mathcal{H}_L)$ is called the Line Hough Transform h_L and defined as follows:

$$\begin{aligned} h_L\colon \quad & \mathcal{C} \to \mathcal{P}(\mathcal{H}_L) \\ & (c_x, c_y) \mapsto \{L_{\theta,r} \in \mathcal{H}_L \mid r = c_x \cdot \cos(\theta) + c_y \cdot \sin(\theta)\} \end{aligned} \tag{3.14}$$

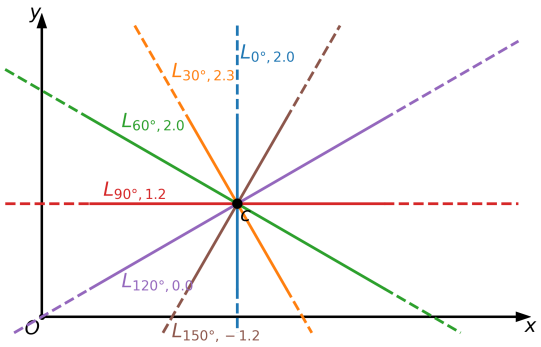

Figure 3.6: This figure shows an example for the Line Hough Transform. The Line Hough Transform h_L maps the cell c (black point) to the set of lines passing through this cell c. In this example, we show all lines in $h_L(c)$ with an angular resolution of $30°$.

For a better intuition of the Line Hough Transform h_L, we show an example in Figure 3.6. Given a feature map with non-empty feature cells $c \in \mathcal{C}$, the Line Hough Transform computes the line L_{θ^*,r^*} that is best supported by the features cells $c \in \mathcal{C}$, i. e., the line that passes through the most cells $c \in \mathcal{C}$:

$$L_{\theta^*,r^*} := \underset{L_{\theta,r} \in \mathcal{H}_L}{\operatorname{argmax}} \left| \{ c \in \mathcal{C} \mid L_{\theta,r} \in h_L(c) \} \right|$$

This is usually computed by discretizing the two-dimensional parameter space $[0, \pi) \times \mathbb{R}$ as a two-dimensional histogram with a bin size of $\theta_{\text{res}} \times r_{\text{res}}$, where θ_{res} is the angular resolution of the first dimension and r_{res} is the offset resolution of the second dimension in the histogram. Each bin in the histogram then corresponds to a specific parameter set (θ, r) and represents a unique line $L_{\theta,r}$. The histogram bins are then filled by iterating over all cells that contain a feature $c \in \mathcal{C}$, computing its Hough Transform $h_L(c)$ and increasing the bin of each line $L_{\theta,r} \in h_L(c)$ by 1. Each bin then contains the number of cells that the corresponding line passes through, i. e., it contains the value of $|\{c \in \mathcal{C} \mid L_{\theta,r} \in h_L(c)\}|$. The bin of the histogram with the highest value then yields the best line L_{θ^*,r^*}.

Computing a finite set of lines $h_L(c)$ for a cell $c \in \mathcal{C}$ is possible, since we are only interested in the lines that are represented in the histogram. Since the angle parameter θ is bounded between $[0, \pi)$ by definition (see Eq. (3.13)), one usually iterates over all possible values for θ at the resolution θ_{res} of the histogram. Since a line is already defined given a point c that it passes through and an angle θ, the corresponding line offset r is then computed according to its definition in Eq. (3.1) as shown in Eq. (3.14).

Pattern Hough Transform For our Pattern Hough Transform approach, we transfer the ideas of the Line Hough Transform to our mathematical model of parallel and equidistant crop rows, the Pattern. Similar to the Line Hough Transform, we constrain the parameter space to ensure a unique parameterized representation of our Patterns. We therefore define the *Pattern Hough Space* $\mathcal{H}_P$ as follows:

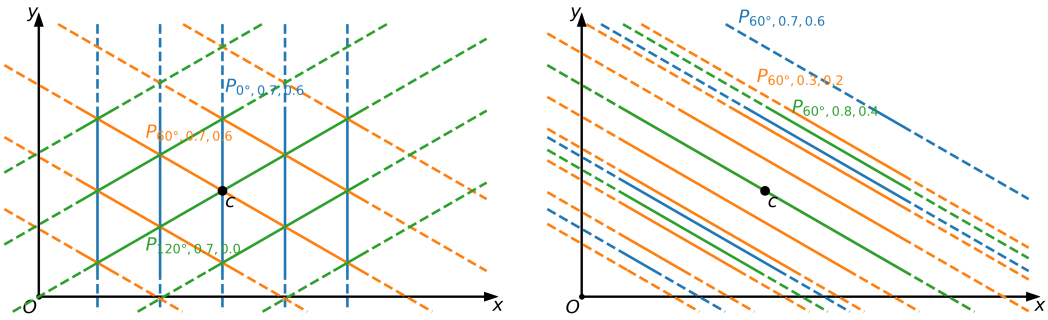

Figure 3.7: This figure shows an example for the Pattern Hough Transform. The Pattern Hough Transform h_P maps a cell c (black point) onto all Patterns that pass through c. On the left, we visualize all Patterns in $h_P(c)$ with an angular resolution of $60°$ and spacing $0.7\,\text{m}$. On the right, we show the Patterns in $h_P(c)$ with an angular parameter of $60°$ and varying spacing parameter values of $0.3\,\text{m}$, $0.7\,\text{m}$ and $0.8\,\text{m}$.

$$\mathcal{H}_P \;:=\; \{P_{\theta,s,o} \mid \theta \in [0,\pi) \;\wedge\; s \in \mathbb{R}_+ \;\wedge\; o \in [0,s)\}\,. \tag{3.15}$$

All possible Patterns are still represented in $\mathcal{H}_P$ but with a unique set of parameters. The power set $\mathcal{P}(\mathcal{H}_P)$ of the Pattern Hough Space then contains all possible sets of Patterns. Analogue to the Line Hough Transform algorithm, we define a *Pattern Hough Transform* h_P that maps a cell $c \in \mathcal{C}$ of a grid map to all Patterns that *pass through* this cell. Since a Pattern is a set of lines, we say that a Pattern P *passes through* a cell c, if there is a line $L \in P$ that passes through c. We therefore define the Pattern Hough Transform $h_P(c)$ of a cell $c \in \mathcal{C}$ as follows:

$$\begin{aligned} h_P\colon \quad & \mathcal{C} \to \mathcal{P}(\mathcal{H}_P) \\ & c \mapsto \{P_{\theta,s,o} \in \mathcal{H}_P \mid \exists\, L_{\theta,r} \in P_{\theta,s,o}\colon L_{\theta,r} \in h_L(c)\} \end{aligned} \tag{3.16}$$

For a better intuition of the Pattern Hough Transform h_P, we show an example in Figure 3.7. Given a feature map with non-empty feature cells $c \in \mathcal{C}$, the Pattern Hough Transform can then be used to determine the pattern P_{θ^*,s^*,o^*} that is best supported by the feature cells $c \in \mathcal{C}$, i. e., the pattern that passed through most cells:

$$P_{\theta^*,s^*,o^*} \;:=\; \underset{P_{\theta,s,o} \in \mathcal{H}_P}{\operatorname{argmax}} \left|\{c \in \mathcal{C} \mid P_{\theta,s,o} \in h_P(c)\}\right|$$

Again, analogue to the Line Hough Transform algorithm, we compute this *best Pattern* P_{θ^*,s^*,o^*} by discretizing the three-dimensional parameterized Pattern Hough Space $\mathcal{H}_P$ with bin size $\theta_\text{res} \times s_\text{res} \times o_\text{res}$, where θ_res is the angular resolution of the first dimension, s_res is the spacing resolution of the second dimension and o_res is the offset resolution of

the third dimension. We then count the number of cells c that pass through each Pattern $P_{\theta,s,o}$ represented by a bin in the histogram.

We compute the Pattern Hough Transform $h_P(c)$ for a cell $c = (c_x, c_y) \in \mathcal{C}$ over all Patterns represented in the histogram according to the following equation:

$$\begin{aligned} h_P(c) &\overset{Eq.\ (3.16)}{=} \{P_{\theta,s,o} \mid \exists\, L_{\theta,r} \in P_{\theta,s,o} \colon L_{\theta,r} \in h_L(c)\} \\ &\overset{Eq.\ (3.14)}{=} \{P_{\theta,s,o} \mid \exists\, L_{\theta,r} \in P_{\theta,s,o} \colon r = c_x \cdot \cos(\theta) + c_y \cdot \sin(\theta)\} \\ &\overset{Eq.\ (3.3)}{=} \{P_{\theta,s,o} \mid \exists\, n \in \mathbb{Z} \colon r = n \cdot s + o \;\wedge\; r = c_x \cdot \cos(\theta) + c_y \cdot \sin(\theta)\} \\ &= \{P_{\theta,s,o} \mid \exists\, n \in \mathbb{Z} \colon n \cdot s + o = c_x \cdot \cos(\theta) + c_y \cdot \sin(\theta)\} \\ &= \{P_{\theta,s,o} \mid o = [c_x \cdot \cos(\theta) + c_y \cdot \sin(\theta)] \mod s\} \end{aligned} \tag{3.17}$$

We can therefore compute the set of all Patterns that pass through a cell $c \in \mathcal{C}$ by iterating over all values of θ and s represented as bins in the histogram and computing the offset parameter o using the modulo operation as shown in Eq. (3.17). In order to obtain a finite set of Patterns $h_P(c)$, we constrain the spacing parameter values s to the range of $s \in [s^-, s^+]$, where s^- is the lowest and s^+ is the highest considered spacing value. This yields a finite set of Patterns $h_P(c)$ that are associated with a bin in the histogram. Using this procedure, we iterate over all feature cells $c \in \mathcal{C}$ and accumulate the number of cells a Pattern passes through in the histogram as described for the Line Hough Transform algorithm. The best Pattern P_{θ^*,s^*,o^*} is then represented by the bin with the highest value in the histogram.

Pattern RANSAC This Pattern detection approach is based on the *random sample consensus* technique. The idea of this technique is that given a set of data points the model that best fits the data points can be found by generating multiple candidate models from randomly sampled data points and keeping the model that is best supported by all data points of the set. For a Pattern RANSAC algorithm, we therefore need to define how we generate a candidate Pattern from randomly sampled data points, as well as how we measure the support of this candidate Pattern on the data points.

To generate a candidate Pattern, we first sample three cells c^1, c^2 and c^3 from the set of vegetation feature cells $\mathcal{C}$. The first two cells c^1 and c^2 define a line L_{θ_c,r_c}. Using the distance of the third cell c^3 to this line as spacing s_c the parameters of the Pattern P_{θ_c,s_c,o_c} are computed as follows:

$$
\begin{aligned}
(d_x, d_y) &:= c^1 - c^2 \\
(n_x, n_y) &:= \begin{cases} (-d_y, d_x), & \text{if } d_x > 0 \\ (|d_y|, 0), & \text{if } d_x = 0 \\ (d_y, -d_x), & \text{if } d_x < 0 \end{cases} \\
\theta_c &:= \tan^{-1}(n_y, n_x) \in [0, \pi) \\
r_c &\overset{Eq.\ (3.14)}{:=} c_x^1 \cdot \cos(\theta_c) + c_y^1 \cdot \sin(\theta_c) \\
s_c &:= \left| d\left(L_{\theta_c, r_c}, c^3\right) \right| \\
o_c &\overset{Eq.\ (3.17)}{:=} r_c \bmod s_c
\end{aligned}
$$

If the resulting spacing parameter s_c is not within reasonable range, i. e., the third sampled cell c^3 is too close to or too far from the line L_{θ_c, r_c}, we reject this candidate Pattern and sample a new set of cells. Analog to the Hough-based approach, we determine the support of the candidate Pattern P_{θ_c, s_c, o_c} by counting the number of all cells $c \in \mathcal{C}$ that lie on the candidate Pattern P_{θ_c, s_c, o_c}. Our Pattern RANSAC approach always samples a fixed amount of candidate Patterns and retains the candidate Pattern with the best support out of all candidate Patterns.

Both our presented approaches aim at finding the best supported Pattern, i. e., the Pattern that passes through the most feature cells. The key difference between both approaches is that the Pattern Hough algorithm searches the full (discretized) space of possible Patterns, and therefore always finds the best supported Pattern within this space. In contrast, the Pattern RANSAC is based on sampling and therefore does not necessarily find the best supported pattern. The advantage of the RANSAC-based approach is that it is not constrained by a discrete parameter set and therefore can potentially find a better fitting Pattern than the Pattern Hough approach.

In this section, we presented the Pattern, our mathematical model of parallel and equidistant lines, that enables detecting all visible crop rows jointly, without prior assumptions on the orientation or location of the crop rows within the sensor data. Based on this Pattern, we introduced two techniques, one based on the Hough Transform, the other based on RANSAC, that extract all visible crop rows jointly from a given feature map. In the next section, we perform an extensive evaluation on several real-world data sets, featuring different crop types in different growth stages to confirm that our crop row detection approach is well suited for guiding a vehicle along the crop rows of an agricultural field.

3.4 Experimental Evaluation

The goal of our Pattern-based crop row detection approach is enabling localization relative to the crops of the field in an autonomous navigation framework. Thus, we design our experimental evaluation towards confirming the suitedness of our approach for this application. The most crucial requirement for any perception algorithm that should provide

information to a localization algorithm is that the information extracted from the sensor data is reliable, i. e., that the provided information is sufficiently accurate throughout all parts of the environment. Therefore, we extensively evaluate the robustness of our approach on a large variety of crop fields – and throughout the entire crop field. The latter means, that we not only consider data recorded while the vehicle traverses the field *in-row* and is thus aligned with the crop rows, but we also explicitly evaluate the more challenging situations during turning maneuvers or when the vehicle leaves or enters the field, i. e., during *transition*. We provide in-depth results for the detection accuracy of our approach on especially challenging data sets. Another important requirement for application in a localization algorithm of an autonomous navigation framework is that the crop rows are detected fast enough to ensure that it runs online. We therefore present the computation times for all investigated crop row detection algorithms. Furthermore, we reference autonomous navigation sessions with the BoniRob that illustrate the applicability of our crop row detection approach in a localization algorithm. To provide a use case beyond localization, we also show an example mapping application for our crop row detection approach.

3.4.1 Methodology

In this section, we first give an overview over the different data sets on which we perform our evaluation. Then, we explain how we obtained ground truth information to compute the accuracy of a detected crop row Pattern. For comparison, we introduce two algorithms that do not leverage our key idea of extracting crop rows jointly, but instead detect individual lines, which is a common approach in the literature. Finally, we discuss how we set the parameter values for all investigated algorithms during our evaluation.

Data Set Overview For our experimental evaluation, we recorded data on a large variety of crop fields, featuring different crop types as well as crops at different growth stages. This allows an extensive evaluation of the robustness of the investigated crop row detection algorithms with respect to different crop types and sizes. We recorded all data sets using our agricultural robot BoniRob as explained in Section 2.1.2 with a maximum speed of $1.0\,\text{m/s}$. The only exception from that is the data recorded on a field of sugar beets in Ancona, Italy. This data was recorded during autonomous navigation of the BoniRob, where we set a maximum speed of $0.2\,\text{m/s}$. Since these sugar beets have a medium size of around $5\,\text{cm}$ we call this data set the *Medium Sugar Beets* data set. On a neighboring field, we also recorded data manually on much smaller sugar beets that just emerged with a size of around $1\,\text{cm}$, resulting in the *Tiny Sugar Beets* data set. Since the crops on these two fields are at a comparatively early growth stage and therefore still quite small, they are hard to detect in LIDAR data. On the other hand, they can be easily detected in high-resolution image data. Therefore, we use the vision information obtained with the PointGrey Blackfly to compute the feature maps as described by Winterhalter et al. [2018]. We provide example images as well as corresponding feature maps for each data set in Figure 3.8. As can be seen in the image for the Tiny Sugar Beets data set, the crops

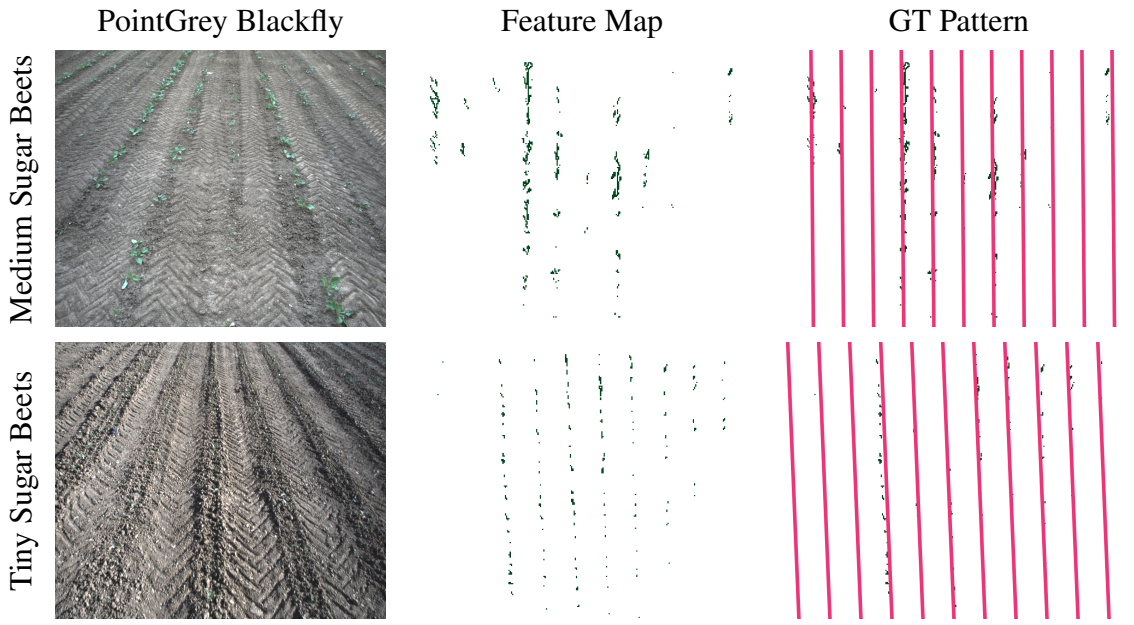

Figure 3.8: This figure shows vision data (left) and corresponding feature maps (mid) with ground truth (GT) Pattern (right) for the Medium Sugar Beets and Tiny Sugar Beets data sets. The image data was recorded with the PointGrey Blackfly mounted in front of the BoniRob at about $1\,\mathrm{m}$ above the ground and tilted downwards at about $25°$. The feature map shows a top-down view of the extracted vegetation features and is located on the ground plane in front of the vehicle. The manually labeled GT Pattern is shown in magenta.

are barely visible to the human eye.

We also recorded data of crops in later growth stages. Since larger crops tend to grow into the space between crop rows and even start overlapping with crops from adjacent rows, they occlude the ground otherwise visible between adjacent crop rows. Detecting crop row structure based on vision information is therefore quite challenging, since the row structure might be occluded by overlapping plants. In contrast, these oversized crops are easily detected in LIDAR data. Thus, we use the information recorded with the Nippon-Signal FX-8 LIDAR sensor to extract feature maps as described by Winterhalter et al. [2018] from the data recorded on the following fields: a production crop field in Eichstetten with leek, the *Leek* data set, and on a research field in Eschikon with canola, the *Canola* data set. In Eschikon, we also recorded data on a fifth crop field, where only corn stubbles remained. Detecting corn stubbles to compute a feature map is hard on both vision and LIDAR data. In image data it is challenging, since the corn stubbles have a similar color to the ground and in LIDAR data the corn stubbles are hard to detect due to their small size. Nevertheless, we chose to use LIDAR data to obtain feature maps for our evaluation, yielding the fifth data set, the *Corn* data set. We provide examples showing a point cloud rendered from the LIDAR information as well as the corresponding feature map for each data set in Figure 3.9. We provide more details on each crop field in

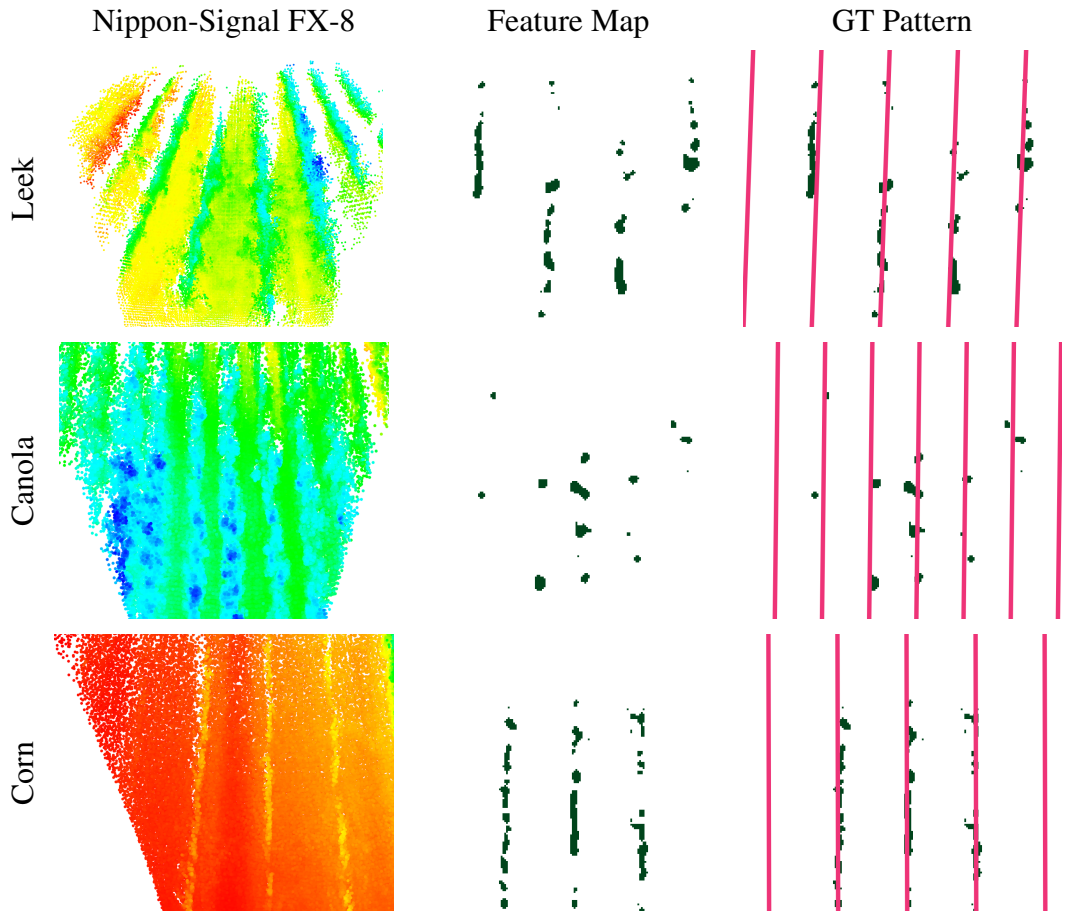

Figure 3.9: This figure shows LIDAR data (left) and corresponding feature maps (mid) with GT Pattern (right) for the Leek, Canola and Corn data sets. The LIDAR data was recorded with the Nippon-Signal FX-8 mounted in front of the BoniRob at about $1\,\mathrm{m}$ above the ground and tilted downwards at about $25°$. The feature map shows a top-down view of the extracted vegetation features and is located on the ground plane in front of the vehicle. The manually labeled GT Pattern is shown in magenta.

Section 2.1.3.

A perception algorithm for autonomous navigation applications such as localization needs to be reliable throughout the entire environment. In our setting, a crop row detection algorithm is therefore reliable, if it not only produces robust results while the vehicle traverses the field *in-row*, but also in more challenging situations when the vehicle approaches the end of the field or re-enters the field after a turning maneuver. In our evaluation, we explicitly include these more challenging situations, where the vehicle is not in-row, but rather in transition, i. e., leaving, entering or re-entering the field after turning. To distinguish between both situations during our evaluation, we split all data sets into an *In-Row* subset, that contains all data recorded while the vehicle was traversing the field in-row and a *Transition* subset that contains the remaining data, where the vehicle was in transition. We give an overview over the most important properties of each

		In-Row			Transition		
	#R	#FM	Δd [m]	VD [%]	#FM	Δa [°]	VD [%]
Canola	3	47	43.94	0.7	59	1.98	0.6
Corn	2	24	35.69	1.6	61	8.00	0.6
Leek	2	79	383.36	0.5	87	45.84	0.7
Medium Sugar Beets	3	70	70.81	0.6	55	49.46	0.3
Tiny Sugar Beets	3	63	109.00	0.5	88	62.72	0.3

Table 3.1: This table shows the properties of all data sets. Depending on the crop type, a different number of crop rows (#R) is sown between the wheel tracks. We also display the number of feature maps (#FM) evaluated and the mean vegetation density (VD) as percentage of cells in a feature map that contain vegetation features. When driving in-row, the heading of the vehicle stays almost constant. Therefore, we only show the translational distance covered (Δd) for the In-Row data sets. During transition, the vehicle does not move far but usually performs a turning maneuver. Therefore we give the angular distance covered (Δa), i. e., the change in heading of the vehicle, for the Transition data sets.

data set in Table 3.1. The comparatively small changes in the heading of the vehicle (Δa) during transition in the Canola and Corn data sets indicate that the vehicle was entering or leaving the field instead of performing a turning maneuver. This is in contrast to the larger angular distance covered during transition in the Leek, Medium Sugar Beets and Tiny Sugar Beets data set, where the vehicle was performing a turning maneuver to transition between crop rows. Since we only evaluate on transition data with visible crop row structure, the table shows the change in heading of the vehicle while the crop rows are still visible in the sensor data during the turning maneuver. Therefore, the angular distance Δa is smaller than the full amount of a turning maneuver, which is usually around $180°$ or more. During transition, where the vehicle might change its heading by large amounts, the part of the field perceived by the sensors of the vehicle also changes rapidly. We therefore extract a feature map from sensor data every 0.1 sec for the Transition data sets. Since the heading stays mostly constant while the vehicle is in-row and the vehicle was moving at a maximum speed of 1.0 m/s, we extract a feature map every 5 sec for the In-Row data sets. This yields different numbers of feature maps (#FM) for each data set as shown in Table 3.1.

Ground Truth Labels and Accuracy Measures For evaluation of the accuracy of a detected Pattern, we need to determine the correct Pattern for each feature map in our data set. To this end, we manually labeled the crop row structure to obtain a *ground truth* (GT) Pattern P^{GT} for each feature map. To produce labels that are independent from the feature maps – and the underlying vegetation feature extraction pipeline – we labeled the GT labels directly on the (integrated) raw sensor data as shown on the left of Figure 3.8 for vision data and Figure 3.9 for LIDAR data. To illustrate that the labeled GT Patterns P^{GT}, on which we base our quantitative evaluation, overlay well with the features extracted from the raw data, we show an example for each data set on the right of Figure 3.8 and

Figure 3.9.

In our quantitative evaluation, we are interested in the *Angular Error*, *Spacing Error* and *Lateral Error* of the detected Pattern $P_{\theta,s,o}$. Therefore, we use our similarity measures Δ_θ,Δ_s and $\Delta_{\text{lat}}^{p_{\text{ref}}}$ from Section 3.3.1 defined in Eq. (3.4), Eq. (3.5) and Eq. (3.8) to Eq. (3.10) respectively. For each detected Pattern $P_{\theta,s,o}$, we compute the error measure for all three Pattern parameters as the corresponding difference between the detected Pattern $P_{\theta,s,o}$ and the GT Pattern P^{GT}. A detected Pattern then has a high accuracy if the corresponding error measure is close to 0. For the Lateral Error, we choose the reference point $p_{\text{ref}} = (1.0, 0.0)$, which corresponds to a position that is directly 1.0 m in front of the vehicle. This is a good choice for the reference point, since the accuracy of the detections directly in front of the vehicle are most important during autonomous navigation.

Comparison Algorithms To confirm that our key idea to extract all visible crop rows jointly improves the robustness of a crop row detection algorithm, we also evaluate on two approaches that do not extract crop rows jointly. We base both comparison approaches on the traditional Hough Transform by [Hough, 1962], since this algorithm is often used to detect crop rows. Both comparison algorithms detect an individual line L_{θ^*,r^*} over all feature cells $c \in \mathcal{C}$ in the given feature map using the Line Hough Transform presented in Section 3.3.2. Both approaches then need to determine the detected crop row pattern P_{θ^*,s^*,o^*} according to the detected line L_{θ^*,r^*} and additional information.

Analogue to many approaches from the literature that assume that the exact crop row spacing s^{prior} is known a priori [Åstrand and Baerveldt, 2005, Bakker et al., 2008, Leemans and Destain, 2006, Marchant, 1996], we also incorporate this information in our first comparison algorithm: Using the detected line L_{θ^*,r^*} and the spacing parameter s^{prior}, we define the detected Pattern P_{θ^*,s^*,o^*} as follows: Each line of the Pattern P_{θ^*,s^*,o^*} is parallel to the detected line L_{θ^*,r^*} and has a distance to the detected line that is a multiple of the given spacing s^{prior}. The angular parameter of the detected Pattern is therefore equal to the angular parameter of the detected line and the spacing parameter s^* is equal to s^{prior}. The offset parameter of the Pattern is then computed according to the definition of the Pattern as $o^* := r^* \bmod s^{\text{prior}}$. Since this algorithm detects only a single line and uses a given spacing s^{prior} to extend it into a Pattern, we call it the *Line Hough* (LH) algorithm.

Extracting additional lines using only the histogram computed by the traditional Hough Transform is not straightforward: For example, extracting the second best line $L_{\theta^{**},r^{**}}$ according to the Hough Transform histogram means determining the bin with the second highest value. However, this bin is usually close to the best bin and therefore, the second best line is usually quite similar to the best line with only slightly different θ^{**} and r^{**} parameters, i. e., the second best line in the histogram usually represents the same crop row as the best line. However, the goal of extracting multiple lines is usually to obtain lines that represent different crop rows.

While extracting a second line representing a different crop row is not straightforward, it is not impossible: To extract a second line $L_{\theta^{**},r^{**}}$ that represents a different crop row, we rely on a rough estimate of the spacing between crop rows in form of a range of possible spacing values. This is in contrast to the Line Hough algorithm that requires the

exact spacing s^{prior}. We therefore assume that a rough estimate of the spacing between crop rows in form of a minimum spacing s^- and a maximum spacing s^+ is known beforehand. Given the first detected line L_{θ^*,r^*} and the range of possible values for the spacing between crop rows $[s^-, s^+]$, we define a subset $\mathcal{H}'_L$ of the Line Hough Space $\mathcal{H}_L$:

$$\mathcal{H}'_L \;:=\; \left\{ L_{\theta^*,r} \in \mathcal{H}_L \mid |r - r^*| \in \left[s^-, s^+\right] \right\}$$

By design, this subset $\mathcal{H}'_L$ contains only lines $L_{\theta,r}$ that are parallel to the best line L_{θ^*,r^*}, i. e., lines where the angular parameter θ is equal to θ^*. It also only contains lines $L_{\theta,r}$ with a distance $d := |r - r^*|$ to the best line L_{θ^*,r^*} within the required boundaries $[s^-, s^+]$. Since the definition of $\mathcal{H}'_L$ only restricts the available value range of the line parameters, the histogram over $\mathcal{H}'_L$ is also a subset of the histogram over $\mathcal{H}_L$. Therefore, computing the second best line $L_{\theta^{**},r^{**}}$ only requires the additional computation of finding the bin with the highest value in the histogram over $\mathcal{H}'_L$.

With this technique for extracting a second crop row represented by the second best line $L_{\theta^{**},r^{**}}$, we introduce our second comparison algorithm, called *Dual Line Hough* (DLH). This Dual Line Hough algorithm uses the Hough Transform to extract a second line $L_{\theta^{**},r^{**}}$ that is parallel to the first detected line L_{θ^*,r^*}, i. e., the angular parameters θ^* and θ^{**} of both lines are the same, but with sufficient distance to the first line L_{θ^*,r^*} to represent a different crop row as described above. This second line then defines the spacing parameter s^* of the detected Pattern P_{θ^*,s^*,o^*} as the distance $|r^* - r^{**}|$ between the first and the second line. The angular parameter of the detected Pattern is then again equal to the angular parameters of both lines θ^*. The offset parameter o^* is computed analogue to the Line Hough algorithm as $o^* = r^* \bmod s^*$, which is the same as $o^* = r^{**} \bmod s^*$, since $s^* = |r^* - r^{**}|$.

Both comparison approaches, the Line Hough and the Dual Line Hough, do not detect a Pattern by considering all visible crop rows jointly. Instead they only use parts of the available information (Dual Line Hough) or depend on prior knowledge of the exact crop row spacing s^{prior} (Line Hough) to extract a Pattern. They therefore provide a good comparison to our Pattern Hough and Pattern RANSAC approaches, that detect the crop row Pattern on all available data.

Parameter Values Before we can evaluate the investigated algorithms on our data sets, we need to clarify how we choose the required parameter values for each approach. Almost all parameters, except for the spacing parameters, are independent of the crop type. Therefore, all parameters, except for the spacing parameters, remain the same throughout all data sets as well as all investigated algorithms. These parameters are the histogram resolutions and value ranges for the Hough-based algorithms, as well as the number of iterations performed for the Pattern RANSAC algorithm.

For the Hough-based algorithms all resolutions of the histogram bins stay the same with an angular resolution of $\theta_{\text{res}} = 0.57°$ and the offset and spacing resolution set to $s_{\text{res}} = o_{\text{res}} = 0.01\,\text{m}$. The value range for the angular dimension of the histogram is $\theta \in [0, \pi)$ per definition of $\mathcal{H}_L$ (see Eq. (3.13)). The same holds for the value range of the Pattern offset parameter $o \in [0, s)$ according to the definition of $\mathcal{H}_P$ (see Eq. (3.15)).

Regarding the value range for the spacing parameter, i. e., s^- and s^+ as well as the required exact value of the crop row spacing s^{prior}, we choose these values depending on the crop type of the data set. As discussed in Section 2.1.1, on agricultural fields, there is usually a distance of $1.5\,\text{m}$ between the tracks created by the wheels of the vehicles. The crops are therefore sown in rows with an even spacing so that the rows have as much clearance from the wheel tracks as possible. Since different crops types have different requirements on the space between adjacent crop rows, this usually results in either 2 or 3 crop rows being sown between adjacent wheel tracks. We show the number of crops (#R) between adjacent wheel tracks for each crop type in Table 3.1. This prior knowledge allows the direct computation of s^{prior} depending on the crop type: For crops with 2 crop rows, we use $s^{\text{prior}} = 1.5\,\text{m}/2 = 0.75\,\text{m}$ and for 3 crop rows, we have $s^{\text{prior}} = 1.5\,\text{m}/3 = 0.5\,\text{m}$. This exact spacing parameter s^{prior} is only required for the Line Hough algorithm. All other algorithms, i. e., the Dual Line Hough and the Pattern Hough as well as the Pattern RANSAC-based algorithms only require a rough expected range $[s^-, s^+]$ of spacing values, which we set to be within $15\,\text{cm}$ around the expected spacing s^{prior}.

Due to the probabilistic nature of the Pattern RANSAC approach, we repeat each evaluation five times. We also decided to evaluate the Pattern RANSAC approach for three different amounts of candidate Patterns: 2500, 5000 and 25000 and call the resulting Pattern RANSAC algorithms *Pattern RANSAC 2500* (R2500), *Pattern RANSAC 5000* (R5000) and *Pattern RANSAC 25000* (R25000). We chose the number of sampled candidate Patterns, so that the Pattern RANSAC 2500 algorithm has a lower computation time than the Pattern Hough algorithm, the Pattern RANSAC 5000 algorithm has a computation time comparable to the Pattern Hough algorithm and the Pattern RANSAC 25000 algorithm gets close to an optimal solution with the highest amount of sampled candidate Patterns.

3.4.2 Robust Row Detection On Challenging Crops

In this experiment, we evaluate the robustness of all investigated crop row detection algorithms. A crop row detection algorithm is *robust*, if it has a high success rate, i. e., if most Pattern detections are successful. Therefore, we first explain how we determine, whether a Pattern detection was successful. We then present the success rates for each algorithm on each data set, split into the In-Row data set and the Transition data set and discuss the results of our evaluation. The success rates also enable us to draw conclusions on the overall reliability of the algorithms when applied in an autonomous navigation framework.

We determine whether a Pattern detection was successful based on the conditions for successful autonomous navigation. In the navigation framework, the detected Pattern is used to correct the heading and sideways tracking of the vehicle relative to the crop rows. We therefore call a Pattern detection *successful*, if the detected Pattern is accurate enough to ensure that the vehicle stays on track and does not accidentally crush value crops. We measure this Pattern accuracy by computing the Angular Error and Lateral Error values of the detected Pattern as explained in Section 3.4.1, where the Angular Error correlates with the heading error of the vehicle and the Lateral Error with the sideways tracking of the

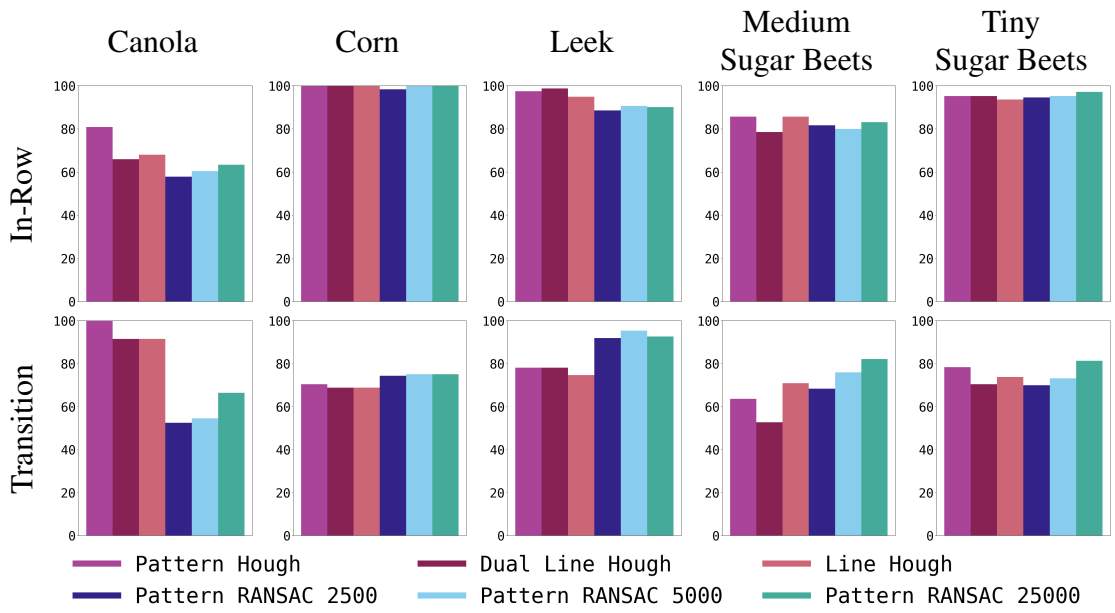

Figure 3.10: This figure shows the success rates for all algorithms on all data sets. The success rate is shown as percentage of the number of successful Pattern detections. Each bar shows the result of a different crop row detection algorithm on the In-Row data sets on the top and on the Transition data sets on the bottom. In each plot, from left to right the algorithms are: Pattern Hough, Dual Line Hough, Line Hough, Pattern RANSAC 2500, Pattern RANSAC 5000 and Pattern RANSAC 25000.

vehicle. Based on real-world experiments with the BoniRob, we determined that an Angular Error below 10° and a Lateral Error below $0.10\,\text{m}$ suffices for successful autonomous navigation (see Section 2.1.1). We thus say that a Pattern was detected *successfully*, if the Angular Error is below 10° and the Lateral Error does not exceed $0.10\,\text{m}$. Furthermore, we call a crop row detection algorithm *robust* if it has a high *success rate*, where the success rate is the percentage of successfully detected Patterns. The success rates for each algorithm on each data set, split into the In-Row and Transition subsets, are shown in Figure 3.10. We also show the detected Patterns on an example feature map of each data set in Figure 3.11 for the In-Row data sets and in Figure 3.12 for the Transition data sets.

Our first observation is that the success rate of each algorithm is dependent on the data set. This is not surprising, since we already explained in Section 3.4.1, that some data sets are more challenging than others, depending on the size and type of crop as well as whether the vehicle was in-row or in transition. Considering the data sets that contain smaller crops, i. e., the vision data sets Medium Sugar Beets and Tiny Sugar Beets, the results show a high success rate on the In-Row data sets for all algorithms. These results are expected since smaller plants allow for an accurate extraction of feature points and thus provide feature maps with clearly visible crop row structure. Therefore, all algorithms perform well on crops in earlier growth stages, when perceived in-row (see also Figure 3.11, first and second row). On the Medium Sugar Beets – In-Row data set,

the Pattern Hough algorithm performs best with a success rate of $85\,\%$, while all other algorithms show a similar performance. This is the same for most other In-Row data sets including the LIDAR data sets Leek and Corn. This is also not surprising since the corn stubbles in the Corn data set also allow for an accurate feature extraction and clearly visible crop row patterns, given good segmentation parameters to distinguish them from the ground despite their small height (see Figure 3.11, third row). Since the leek plant grows quite tall above the ground, it is easily detected in the LIDAR data. Furthermore, due to its large stalks, it also produces accurate feature maps with clearly visible row structure (see Figure 3.11, fourth row). The only exception is the Canola data set. Here, the success rate drops notably for all detection algorithms. This confirms that the Canola data set is more challenging than the other data sets – even in-row. This is again caused by the properties of the crop type and the late growth stage of the crop: As discussed in Section 3.4.1, the canola crop in a late growth stage is hard to segment in both vision and LIDAR data, since the plant grows into the space between crop rows, covering the soil and it also does not rise high above the ground. This causes less accurate feature maps with a row structure that is not always clearly visible as can be seen on the bottom row in Figure 3.11. Here, the Pattern Hough outperforms all others.

Looking at the results for the Transition data sets, we see that the overall success rates of all algorithms drop compared to the In-Row counterparts except for the Canoladata set. This again confirms our expectation that detecting the crop row Pattern successfully is in general more challenging during transition than while traversing the field in-row. This is explained by the fact that the sensors perceive different information, i. e., the headlands, that is not part of the field during transition. Since the crop row structure is only present on the field, there is less row structure visible in the sensor data during transition. Furthermore, additional vegetation features from plants that grow on the headlands cause an increased amount of noise in the feature maps. A good example for this fact is the Corn data set: Since the stubbles produce accurate feature maps – if segmented correctly, all algorithms have an almost perfect success rate of around $100\,\%$ on the In-Row data set. However, segmenting the corn stubbles during transition in the LIDAR data is more challenging and noise is introduced into the feature maps. Additionally, less crop row structure is visible while the vehicle is entering or leaving the field. Therefore, the success rates drop to around $75\,\%$ throughout. However, crop rows are still successfully detected as shown in the example in Figure 3.12 in the third row. On larger crops such as in the Canola, the Leek or the Medium Sugar Beets data sets, more vegetation feature points are distributed across a larger surface of the crops. In general, this causes feature maps with larger clusters of vegetation features compared to the sparser feature maps obtained from smaller sized crops, such as the Tiny Sugar Beets. The example feature maps shown in Figure 3.8 and Figure 3.9 illustrate this difference in vegetation feature distribution on all data sets. These denser feature maps already cause slight differences in performance on the In-Row data sets of Canola, Leek and Medium Sugar Beets, while all algorithms perform almost the same on the sparser feature maps of Corn and Tiny Sugar Beets. On the Transition data set, these differences in performance become more evident, since the Transition is in general already more challenging. In these challenging situations, i. e., during transition on larger sized crops with denser feature maps, on the

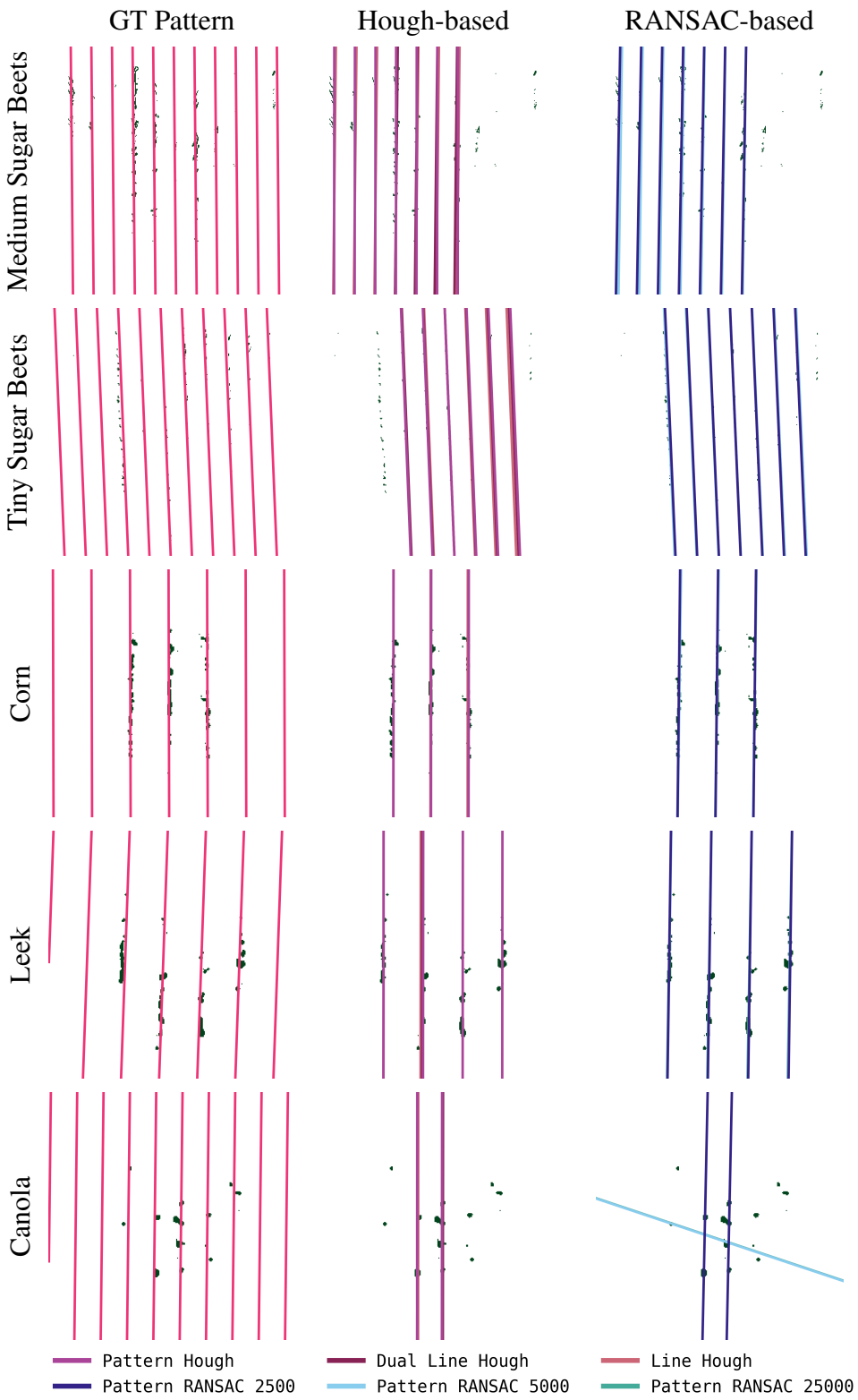

Figure 3.11: This shows **In-Row** detection results for all evaluated algorithms as well as the manually labeled GT Patterns for comparison. These are from left to right: the GT Pattern, all Hough-based detections, all Pattern RANSAC-based detections.

Transition data set of the Canola, Leek and Medium Sugar Beets, our approaches that use all available information to jointly detect the crop row Pattern outperform the comparison algorithms, that can not leverage all available information. While our Pattern RANSAC method performs overall better on the Leek and Medium Sugar Beets data sets during transition, our Pattern Hough algorithm performs especially well on the most challenging Canola data set. In comparison to the Line Hough or Dual Line Hough the Pattern Hough is always similar or better in performance than the Dual Line Hough. Note that the Line Hough does not estimate the spacing and gets the ground truth spacing s^{prior} as an input.

Overall our evaluation of the robustness confirms that all investigated algorithms show reliable results while the vehicle is in-row on most data sets, with the exception being the challenging Canola data set, where the Pattern Hough is more robust. Regarding the reliability during transition, our evaluation confirms that our Pattern-based approaches detect an accurate Pattern more often than the comparison algorithms, especially on the more challenging data sets with larger sized crops. Interestingly, the results also show that there is a large difference of performance between our Pattern-based approaches during transition: While the Pattern Hough is more robust on the Canola data set, the Pattern RANSAC-based approaches perform better on the Leek and Medium Sugar Beets data sets. We therefore investigate the results on these data sets in more detail in our next experimental evaluation.

3.4.3 Row Detection during Transition

The previous evaluation confirms that the overall robustness of our Pattern-based approaches is reliable enough both in-row and during transition to accurately guide a vehicle on an agricultural crop field. However, the results on the Transition data sets are not as clear as the results on the In-Row data sets. Therefore, we investigate the individual errors, i. e., the Angular Error, the Spacing Error and the Lateral Error, in more detail in this evaluation. In particular, we are interested in those data sets, where the performance of our Pattern-based approaches notably deviates, i. e., the Canola, the Leek and Medium Sugar Beets data sets during transition as seen in Figure 3.10.

For the Medium Sugar Beets – Transition data set, we take a closer look at the Spacing Error and Lateral Error values shown in Figure 3.13. The first observation is that the Dual Line Hough has a larger Spacing Error than all other algorithms. By design, the Dual Line Hough only considers the second best parallel line to estimate the spacing parameter s^* of the detected Pattern P_{θ^*,s^*,o^*}. These results show, that on real-world data, where crops are not sown in perfectly equidistant crop rows, considering all visible crops jointly is preferable and even crucial for robust results. While the Pattern Hough only finds a Pattern with suitable Lateral Error about $60\,\%$ of the time, the Lateral Error still remains low with most values only $2 - 3\,\text{cm}$ above the success threshold of $10\,\text{cm}$. By definition of our success rate, the Pattern Hough does have a lower success rate than the Pattern RANSAC algorithms. However, most of the unsuccessful Pattern extractions still have a low Lateral Error close to the threshold. We can therefore conclude, that the performance of the Pattern Hough is still reasonable on the Medium Sugar Beets data set, while only slightly exceeding the success threshold for the Lateral Error.

Figure 3.12: This shows **Transition** detection results for all evaluated algorithms as well as the manually labeled GT Patterns for comparison. These are from left to right: the GT Pattern, all Hough-based detections, all Pattern RANSAC-based detections.

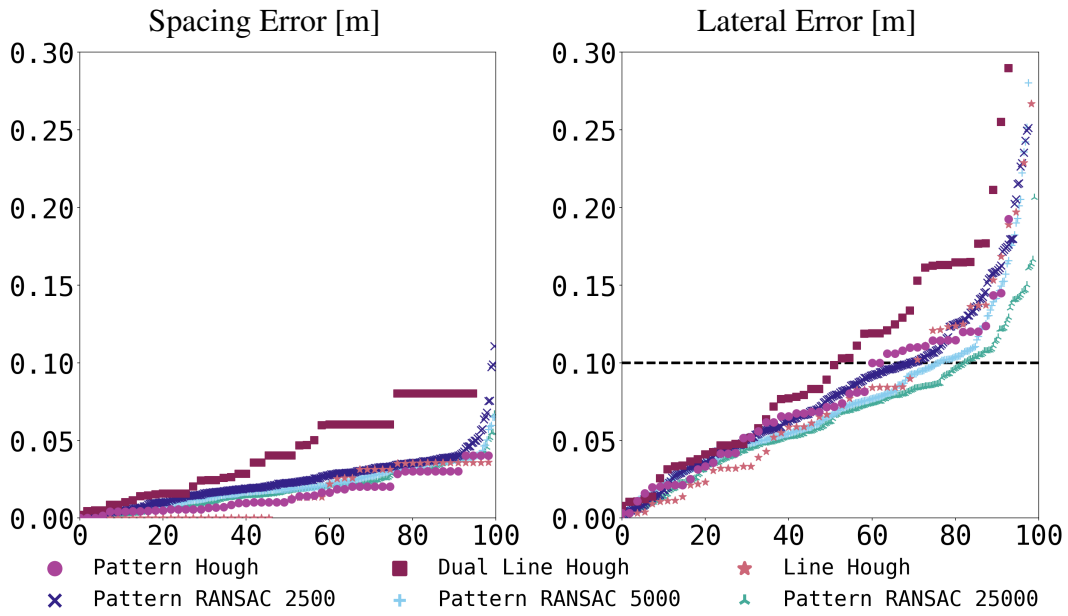

Figure 3.13: This figure shows the **Spacing Error** (left) and the **Lateral Error** (right) on the **Medium Sugar Beets – Transition** data set. The individual error values for each measurement are shown in ascending order. The horizontal axis shows the position of the measurement in the sorted list in percent ([%]). The error value is plotted along the vertical axis. Since we repeated the RANSAC algorithms five times, they have five times more measurements than the Hough-based algorithms.

Regarding the Angular Error and Lateral Error on the Leek – Transition data set shown in Figure 3.14, we see that the Pattern Hough has a larger Angular Error than the Pattern RANSAC-based algorithms between the $75\,\%$ and $95\,\%$ mark. This is caused by several feature maps in the Leek data set that only contain quite sparse amounts of vegetation features in comparison to the other feature maps of this data set. Due to the sparse distribution of vegetation features, detecting a Pattern-based on samples as done in the Pattern RANSAC algorithms is beneficial since it increases the chance of finding an accurate Pattern. This is in stark contrast to the Hough-based approaches: They are less suited for sparse feature distribution since the accuracy of a detected Pattern is always limited by the fixed resolution for each Pattern parameter in the histogram. Therefore, on very sparse feature maps, the Hough-based approaches might not be able to detect the crop row Pattern with sufficient accuracy. While the Angular Error of the Hough-based approaches is about $4°$ larger than the Angular Error of the Pattern RANSAC-based methods, with $7°$ error in total it is still accurate enough according to our success threshold of $10°$. However, the Lateral Error depends on an accurate detection of the angular parameter θ^* of the Pattern P_{θ^*,s^*,o^*}. Therefore, the increase in the Angular Error also causes an increase in the Lateral Error for the Hough-based approaches, leading to a Lateral Error that exceeds the lateral success threshold of $0.10\,\mathrm{m}$. We therefore conclude that on sparse data, using a sampling-based method such as the Pattern RANSAC can lead to more robust results than employing a full search on a discretized search space as done by the Hough-based

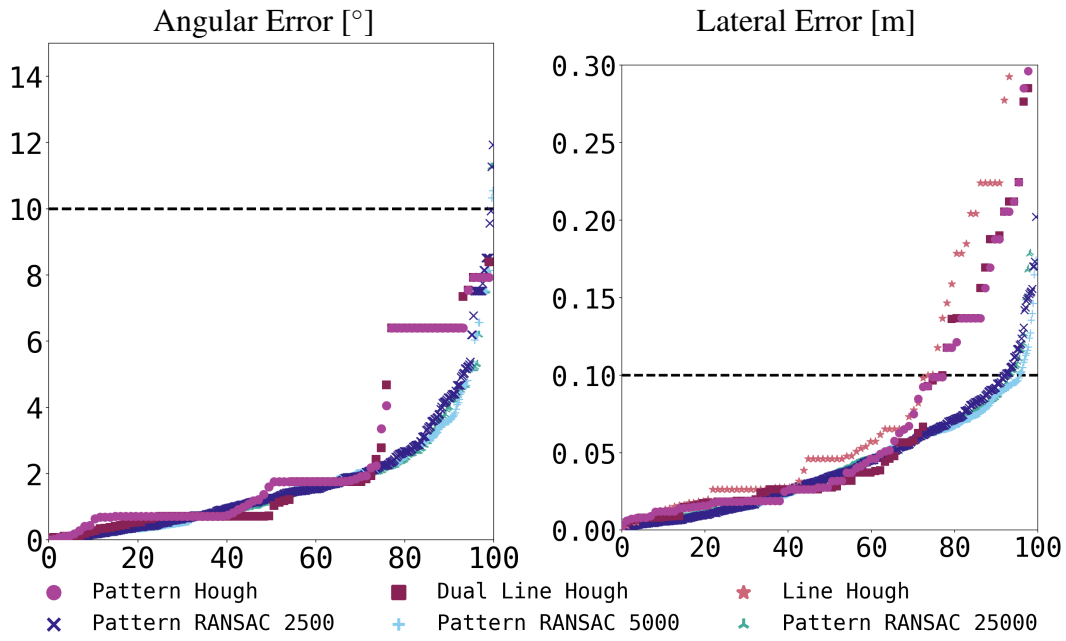

Figure 3.14: This figure shows the **Angular Error** (left) and the **Lateral Error** (right) on the **Leek – Transition** data set. The individual error values for each measurement are shown in ascending order. The horizontal axis shows the position of the measurement in the sorted list in percent ([%]). The error value is plotted along the vertical axis. Since we repeated the RANSAC algorithms five times, they have five times more measurements than the Hough-based algorithms.

approaches.

Transferring the findings for sparse vegetation feature distributions encountered on the Leek data set to the dense vegetation feature distributions of the Canola – Transition data set, it follows that the success rate results are inverted: the Pattern RANSAC-based methods show much lower success rates than the Hough-based approaches (see Figure 3.10). The Angular Error and Lateral Error on the Canola – Transition data set shown in Figure 3.15 confirm this conclusion, since here also the results are inverted when compared to the Angular Error and Lateral Error on the Leek – Transition data set in Figure 3.14. In the Canola data set, the Angular Error values of the Pattern RANSAC-based approaches diverge starting at different percentages depending on the amount of candidate Patterns sampled. This makes sense, since sampling more candidates has a higher chance of finding a set of samples with an accurate angular parameter θ^*. Therefore, the Angular Error of the Pattern RANSAC 2500 starts to diverge first at about $20\,\%$, followed by Pattern RANSAC 5000 and Pattern RANSAC 25000 at about $25\,\%$. Analogue to our observations for the Hough-based algorithms on the Leek data set, this also causes a notable increase of the Lateral Error, resulting in a small amount of successful Pattern detections of around $50\,\%$ for Pattern RANSAC 2500 and up to $65\,\%$ for Pattern RANSAC 25000. This is in contrast to the Leek – Transition data set, where the Hough-based approaches still show sufficient angular accuracy and reasonable lateral accuracy resulting in a man-

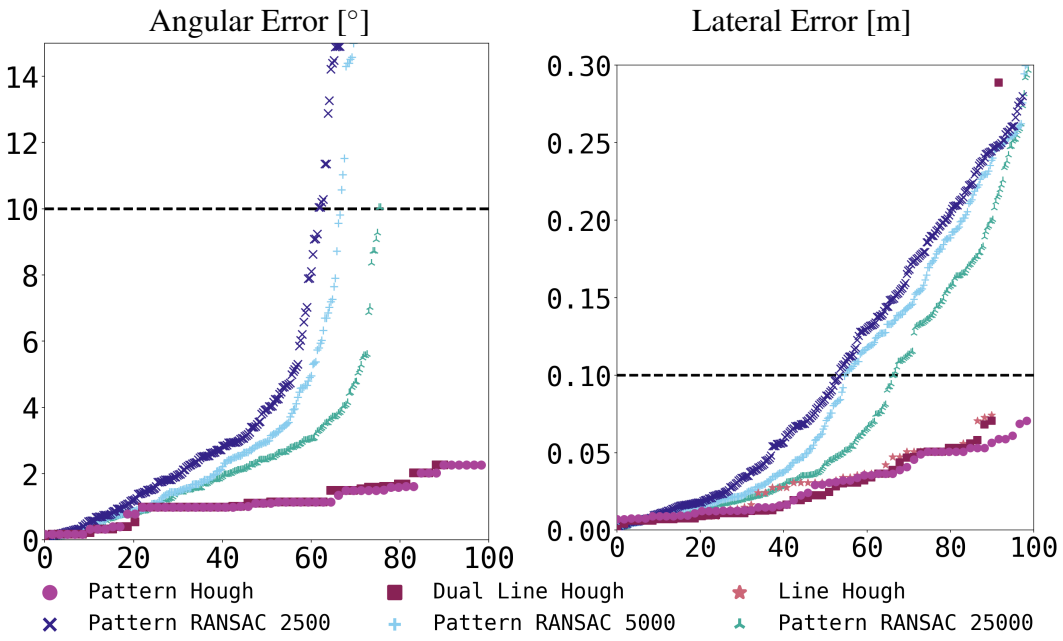

Figure 3.15: This figure shows the **Angular Error** (left) and the **Lateral Error** (right) on the **Canola – Transition** data set. The individual error values for each measurement are shown in ascending order. The horizontal axis shows the position of the measurement in the sorted list in percent ([%]). The error value is plotted along the vertical axis. Since we repeated the RANSAC algorithms five times, they have five times more measurements than the Hough-based algorithms.

ageable success rate of almost 80 % on this data set. Another interesting observation is that although the Pattern Hough algorithm shows an impressive 100 % success rate on the Canola – Transition data set, the Lateral Error still approaches the success threshold with values up to 7 cm.

In summary, our detailed evaluation on the challenging Transition data sets shows, that both the Pattern RANSAC methods as well as the Hough-based approaches, have advantages and disadvantages depending on the feature distribution of the provided data. Due to the sampling-based nature of the Pattern RANSAC methods, these algorithms are better suited for sparsely distributed vegetation features, e. g., on crops in early growth stages or crops that grow slim and tall, such as the Leek crop. In contrast, the Hough-based approaches are more robust on denser vegetation feature distributions that are usually observed on crops in later growth stages, and especially on crops that grow wide and stay close to the ground such as the Canola crop. While the robustness of the Pattern Hough algorithm on the Leek – Transition and Medium Sugar Beets – Transition data sets is lower than the robustness of the Pattern RANSAC approaches, these results are inverted on the Canola – Transition data set. However, the detailed analysis of the individual errors on these three data sets shows, that the accuracy of the Pattern Hough over all three data sets combined is higher than the accuracy of the Pattern RANSAC-based methods.

[msec]	LH	DLH	PH	R2500	R5000	R25000
Canola	8	8	89	34	69	340
Corn	13	12	119	40	79	391
Leek	8	8	71	33	64	319
Medium Sugar Beets	9	9	53	27	53	259
Tiny Sugar Beets	8	8	41	27	52	255

Table 3.2: This table shows the computation times for all algorithms in milliseconds.

3.4.4 Application in Autonomous Navigation

In order to apply the perception algorithm in an autonomous navigation framework not only robust results are important, but also computational efficiency. Using the detected crop rows as the input for pose correction in, for example, a localization algorithm requires that large amounts of data, i. e., images from cameras or point clouds from LIDAR sensors are processed in time. This is crucial to facilitate a high update rate for the pose correction step of the localization algorithm, and thus ensure an accurate pose estimate relative to the value crops and in turn guiding the vehicle across the crop field without damaging the crops.

We therefore also report the computation times for all investigated algorithms in Table 3.2. These timings include the time required for preprocessing the raw data into a feature map for a better intuition about the applicability of each algorithm in an online navigation framework. We evaluated all data sets on an Intel Core i7-4770 CPU with 16 GB RAM, which is comparable to the Pokini i2 of the BoniRob that is equipped with an Intel Core i7-4600U CPU and 16 GB RAM.

The Line Hough and Dual Line Hough are an order of magnitude faster than the Pattern Hough. This is to be expected since the dimension of $\mathcal{H}_L$ and $\mathcal{H}'_L$ is one dimension lower than the dimension of $\mathcal{H}_P$, i. e., the pattern spacing parameter is explicitly considered in the Pattern Hough algorithm and therefore adds an additional dimension to the histogram. By design (see Section 3.4.1), the Pattern RANSAC 2500 is faster than the Pattern Hough, the Pattern RANSAC 5000 has comparable computation times and the Pattern RANSAC 25000 is five times slower than the Pattern Hough. The larger timings for all algorithms on the Corn data set are caused by the higher vegetation density (see Table 3.1) while driving in-row. A higher number of vegetation features notably increases the computation times for all algorithms, since they all need to iterate over all vegetation features: The Pattern RANSAC methods compute the support over all vegetation features for each candidate Pattern. The Hough-based approaches compute the Hough transform for each vegetation feature. Overall, all computation times are feasible for online application.

In this section we also provide qualitative information on the applicability of our crop row detection approach for autonomous navigation. We used the Pattern Hough algorithm for autonomous navigation with our BoniRob to correct the pose estimate in a localization algorithm relative to the detected crop rows on many different occasions. For example, instead of manually steering the BoniRob, we used this navigation system to autonomously collected data from entire crop fields. One of the data sets collected autonomously is

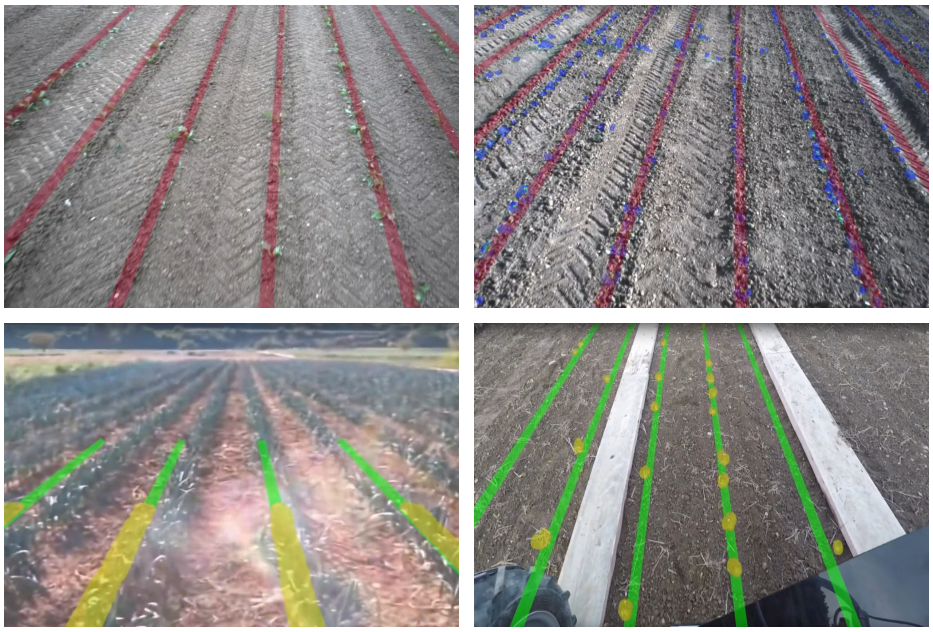

Figure 3.16: This figure shows screenshots from two different videos referenced in this section. The pictures on the top are captured from the first video showing crop row detection on the Medium Sugar Beets data set on the left and on the Tiny Sugar Beets on the right. The pictures on the bottom show detections from the second video on the Leek data set on the left and on tiny sugar beet plants on a different crop field on the right.

the Medium Sugar Beets data set of our experimental evaluation as already mentioned in Section 3.4.1. Furthermore, we have recorded several data sets and videos where the BoniRob navigates autonomously on different crop fields using our Pattern Hough crop row detection. The most relevant video for this chapter that also features autonomous navigation is the video *Crop Row Detection on Tiny Sugar Beets*[1](see Figure 3.16 on the top). In this video, we show data from the Medium Sugar Beets, the Leek and the Tiny Sugar Beets data sets. The sections of the video where the BoniRob is driving autonomously, show medium sized sugar beets from the Medium Sugar Beets data set. Towards the end of the video, we also show data from the Tiny Sugar Beets data set (see Figure 3.16 on the top). Another relevant video that also illustrates the performance of our Pattern Hough algorithm is *Crop Row Detection*[2](see Figure 3.16 on the bottom). Here, data from the Leek data set and a field located in Bonn featuring tiny sugar beets are shown.

There are also other possible use cases for our crop row detection method, such as mapping. In Figure 3.17 we show an example, where we used our Pattern Hough algorithm to extract crop rows from a GPS-referenced overhead image acquired with a UAV[3]. This GPS-referenced image with crop row annotations acquired from our Pattern Hough method can then be used to create a GPS-referenced map of crop rows. As we will see

[1]https://youtu.be/Bsa1o6vwof0
[2]https://youtu.be/0VIwuCaTHPM
[3]Thanks to Raghav Khanna from ETH Zurich for providing the image.

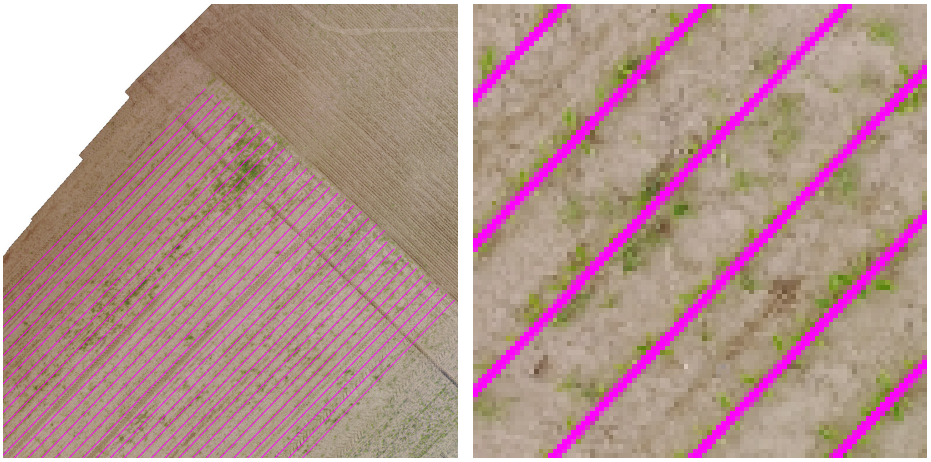

Figure 3.17: This figure shows an example for a mapping application of our crop row detection approach. We use our Pattern Hough algorithm to detect the crop rows of an entire crop field on a GPS-referenced overhead image (left). The close-up on the right shows that the detected Pattern nicely overlays with the crop rows.

in the following chapter, such a GPS-referenced map of crop rows is necessary to enable localization beyond crop row following.

In summary, our experimental evaluation confirms that extracting a Pattern, i. e., all visible crop rows jointly yields robust results especially during transition, which is crucial for application in an autonomous navigation framework. The in-depth evaluation of individual errors shows that our Pattern Hough algorithm is overall more reliable than the other investigated approaches and produces reasonable results on all data sets. Our extensive experience on applying the Pattern Hough algorithm in our autonomous navigation framework on the BoniRob confirms these results since we performed several successful autonomous runs over entire crop fields with the BoniRob.

3.5 Conclusion

In this chapter, we presented two reliable crop row detection approaches for autonomous navigation, in-row and during transition, with a wide application range. Our extensive experimental evaluation confirms, that our key idea of considering all available data to extract visible crop rows jointly improves the robustness of crop row detections, especially during transition. We also evaluated our crop row detection approaches on a variety of crop types at different growth stages, as well as on data from different sensor modalities, i. e., from an RGB camera and a LIDAR sensor, which where mounted at different positions in front of the vehicle. This confirms the wide application range of both our approaches. Overall, our Pattern Hough algorithm produces more reliable results than the Pattern RANSAC-based algorithms in most situations. We therefore employ the Pattern Hough algorithm in our navigation framework on the BoniRob, which resulted in many successful autonomous traversals of entire crop fields.

An interesting general direction for future work is to investigate additional sensor modalities and develop techniques to extract vegetation feature maps from these modalities as this would also extend the application range and adaptability of our crop row detection approaches. For example, cameras that capture images of different wavelengths in the near infrared spectrum are interesting since plant material, i. e., vegetation features, can be easily segmented on these images. For a similar reason, plant material also has higher reflectance in LIDAR data. Therefore, if the LIDAR sensor also provides intensity information and not only the depth, this additional intensity information could be used to improve vegetation segmentation on LIDAR data. Additionally, the use of state-of-the-art machine learning techniques can be investigated to achieve more robust segmentation results.

Another interesting research direction is to extend our key idea of joint crop row detection to different crop row patterns. For example, in areas with large amount of flat terrain, e. g., in the United States, Australia or Brazil, large center-pivot irrigation systems are often used to water the crops. Since these systems can efficiently distribute water on a circular shaped area, the crops on these fields are also sown in a circular pattern. Our crop row detection approach can be easily adapted to the structure of these types of field by modifying the definition of the Pattern. Similarly, during our data recording sessions on production fields, we encountered situations, where the crop rows were not sown in equidistant lines throughout. Instead, a slightly larger spacing $s^{>}$ was used between crop rows that are adjacent to the wheel tracks. This makes sense as it increases the clearance between the wheels of a vehicle and the crops thereby decreasing the chance of damaging the crops while traversing the field. However, this irregular spacing results in a slightly different crop row pattern, where two spacing parameters, i. e., the original Pattern spacing parameter s describing the spacing between crops and the slightly larger spacing parameter $s^{>}$, are required to model the crop row structure.

These ideas for extending the application range of our crop row approach are another indicator of its flexibility and usefulness in real-world applications.

Another idea for increasing the overall robustness of the entire navigation framework, is to extend the amount of information returned by our crop row detection approach. More precisely, instead of only returning the detected Pattern our approach could be extended so that it also provides an estimate of the reliability of this detected Pattern. Such an estimate for the reliability of a detection is always valuable in a navigation framework. For example, if the detected Pattern is intended as an input to a localization algorithm for pose correction, this localization algorithm can use the reliability estimate to decide whether to perform pose correction based on the detected Pattern: If the detection has a high reliability, the detection is used to correct the pose estimate. If the detection has a low reliability, it is discarded and the localization algorithm waits for a more reliable detection. Such a reliability measure, called the *quality* of a Pattern was introduced by Winterhalter et al. [2021].

In the next chapter we present our localization approach that employs the presented Pattern Hough algorithm for pose correction.

Chapter 4

Beyond Crop Row Following

For autonomous navigation on entire crop fields an accurate pose estimate is required to guide the vehicle along the crop rows without damaging the crops. A common approach is to use the heading and lateral offset information of locally detected crop rows to steer the vehicle along crop rows with high precision. However, these approaches cannot provide information about the position of the vehicle in the direction of the crop rows, usually limiting autonomous navigation to pure crop row following without the ability to perform transition maneuvers at the headlands. In contrast, GPS-based localization approaches, while providing accurate pose estimates to enable autonomous traversal of entire crop fields, do not contain any information of the pose of the vehicle relative to the crops. However, for precision agriculture applications, where individual treatment on a per plant basis might be required, a pose estimate relative to the crops is crucial. We therefore propose to leverage the advantages of crop row following and GPS-based localization in a fused localization approach that uses the heading and lateral offset information from local crop row detections as well as the global information from GPS measurements to estimate the longitudinal position of the vehicle. Since fusing sensor modalities that are defined in different frames is challenging, we contribute to the state of the art as follows: We fuse both modalities in a common frame defined by a map of GPS-referenced crop rows. For associating the local crop row detections with the mapped crop rows, we present a novel *Crop Row data association*. We fuse the measurements of both sensor modalities by defining the fused sensor measurement using a heading, lateral and longitudinal component. Additionally, we also develop an *End of the Field detection* to improve the longitudinal position estimate of our fused localization approach. In-depth experiments on two real-world data sets confirm that our localization approach consistently fuses the information contained in both sensor modalities, and that additionally detecting the end of the field notably improves the accuracy of the longitudinal position estimate. Our fused localization approach estimates the pose of the vehicle relative to the crops while also enabling autonomous transition maneuvers at the headlands – thereby going beyond the limitations of crop row following.

4.1 Introduction

For autonomous navigation on agricultural fields accurate knowledge of the pose of the vehicle relative to the crops is crucial to allow precise navigation without damaging the value crops as well as enable precision agriculture applications. While traversing the field in-row, a highly accurate estimate of the heading and lateral offset of the vehicle relative to the crop rows is required to guide the vehicle along the crop rows of the field without damaging crops. When approaching the end of the field, additionally to the heading and lateral offset of the vehicle, an accurate estimate of the longitudinal location of the vehicle in the direction of the crop rows is crucial to enable autonomous transition maneuvers, such as turning at the headlands. For example, while driving in-row, even a slightly inaccurate lateral offset estimate can cause the autonomous vehicle to drive over the valuable crops as illustrated in the top row of Figure 4.1. Here, the estimated position of the vehicle is too far above the tracked crop rows, so that the wheels of the vehicle are located on the crop row above. Therefore, the navigation system will correct the position of the vehicle and steer to the right to continue tracking the crop rows without driving over crops. However, considering the true position shown on the top right of Figure 4.1, steering to the right actually causes the vehicle to drive over the crop rows below. This example illustrates the importance of precise heading and lateral offset estimates of the vehicle relative to the crop rows while driving in-row. We also show two examples on the mid and bottom row of Figure 4.1 that explain why an accurate estimate of the position of the vehicle in the direction of the crop rows, i. e., in the longitudinal direction, is required to enable transition maneuvers at the headlands. The first situation in the middle of Figure 4.1 illustrates an incorrect longitudinal position estimate, where the vehicle has already left the crop field according to the localization module. Therefore, the navigation system initiates a turning maneuver to transition to the next set of crop rows. However, considering the true position of the vehicle on the right, the vehicle is actually still inside the field. Due to the incorrect longitudinal position estimate, the autonomous vehicle is initiating the turning maneuver too early and therefore causes damage to the crops located at the end of the field. The other example shows the opposite situation, where the vehicle is still inside the field according to the longitudinal position estimate as shown on the bottom left of Figure 4.1. Assuming that the vehicle is still inside the field, the navigation system does not stop but continues driving forward. Since the true position of the vehicle as shown on the right of Figure 4.1 is already outside of the field, a continued forward motion might cause the vehicle to leave the headlands and drive into possibly harmful terrain such as, e. g., bushes or even ditches. These examples demonstrate that not only a precise estimate of the heading and lateral offset of an autonomous vehicle but also an accurate longitudinal position estimate are required to traverse an entire crop field without driving over the crops.

In the literature two separate approaches exist for localizing a vehicle in a crop field for autonomous navigation. The first approach is based on detecting crop rows using a sensor mounted in front of the vehicle. These crop row detections contain information about the heading and lateral offset of the vehicle relative to the crop rows as shown on the right of Figure 4.2. Since these heading and lateral offset measurements are only defined

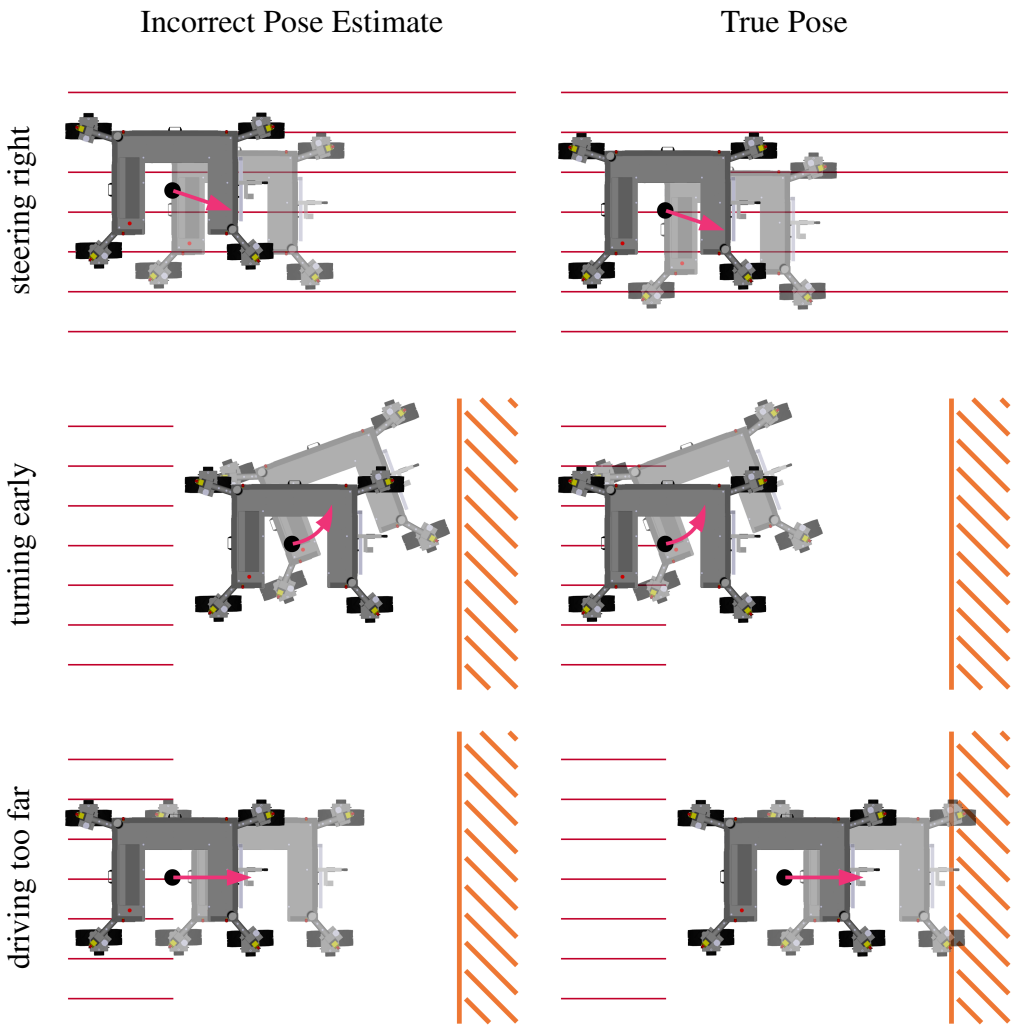

Figure 4.1: This figure shows different examples how an incorrect pose estimate (left) can cause critical navigation failure (magenta arrows). For comparison, we also show the correct pose of the vehicle on the right. The crop rows are visualized as red lines and possibly harmful terrain such as ditches or bushes at the far end of the headlands is highlighted in orange. We also show the pose of the vehicle after executing the navigation command (magenta arrows) as semi-transparent projection. On the top row the incorrect pose estimate is too far to the left of the crop rows, so that the vehicle is steering to the right to correct its position, which in reality then causes the vehicle to drive over crops as shown on the right. In the middle row the vehicle is turning too early and again damaging the crops, since the estimated position of the vehicle incorrectly suggests that the vehicle has already left the field. On the bottom row, we show the opposite situation, where the pose estimate incorrectly locates the vehicle inside the field, when it in fact has already left the field, causing the vehicle to drive into potentially harmful terrain at the edges of the headlands.

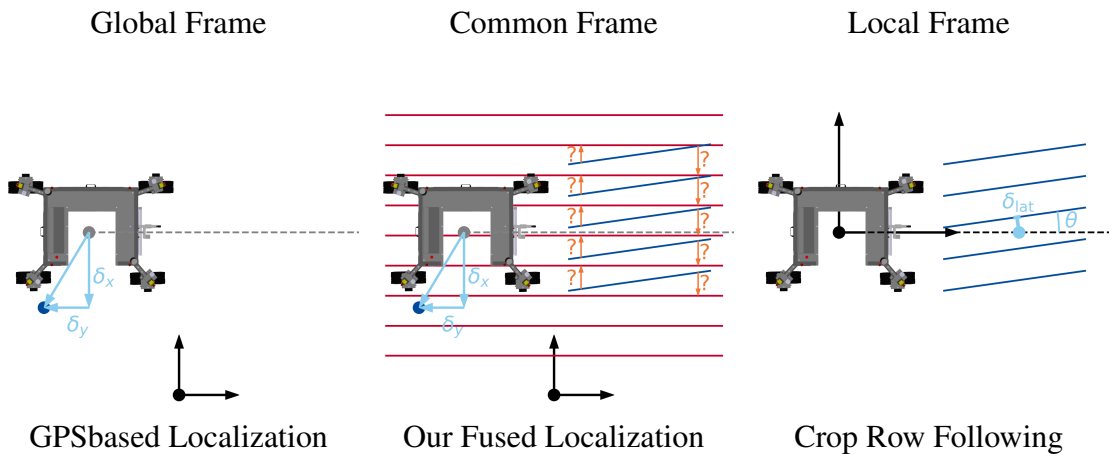

Figure 4.2: This figure shows GPS-based localization (left), our fused localization (mid) and the crop row following (right) approaches for pose estimation on crop fields. Sensor measurements such as the GPS position as well as detected crop rows are shown in blue. The resulting correction measurements for the pose estimate are highlighted in cyan. The origin and axes of each frame are shown in black. Our GPS-referenced map of crop rows is visualized as red lines. We also highlight the data association problem in orange.

within the *local frame* of the vehicle, we say that the detected crop rows contain *local* information. The vehicle then uses these local heading and lateral offset measurements to follow the detected crop rows without damaging the value crops. We therefore call such approaches *crop row following* approaches. While detecting crop rows provides precise heading and lateral offset measurements, it cannot provide any information about the position of the vehicle in the direction of the crop rows, i. e., about the longitudinal position of the vehicle. Thus, a crop row following approach can guide a vehicle along the crop rows with high precision, but it cannot provide the information required for transition maneuvers, such as turning at the headlands.

The second localization approach found in the literature is based on precise GPS information that can reach up to sub-centimeter accuracy. These approaches determine a highly accurate position of the vehicle within a global reference frame as shown on the left of Figure 4.2. If a differential GPS is used, the heading of the vehicle can also be estimated. Since these approaches purely rely on global GPS measurements, they do not have any information about the pose of the vehicle relative to the crops rows. However, as explained in more detail in Section 2.1, a key requirement for autonomous navigation in precision agriculture is an accurate pose estimate relative to the crops to enable reliable and accurate autonomous navigation in agricultural fields without damaging the crops.

While the crop row following approach provides accurate heading and lateral offset estimates relative to the crop rows but no longitudinal measurements, the GPS-based approach can only localize the vehicle in a global reference frame without any relation to the crops. Autonomous navigation for precision agriculture applications, however, requires an accurate pose estimate of the agricultural vehicle relative to the crops in all three dimensions, i. e., heading, lateral offset as well as longitudinal direction. In this chapter, we

therefore present our novel localization approach that *fully* localizes a vehicle relative to the crop rows in all three dimensions.

The key idea for our localization approach is to leverage the advantages of both localization approaches by fusing the local crop row detections used in a crop row following approach with the global GPS measurements used in the GPS-based approach. More precisely, we propose a localization approach that is based on crop row following and therefore utilizes the crop row detections to determine the heading and lateral offset of the vehicle relative to the crop rows. By correcting the longitudinal position of the vehicle according to the global position information obtained from a GPS sensor, our approach goes beyond simple crop row following.

While the idea of fusing crop row detections with GPS information for localization is straightforward, consistently fusing both sensor modalities in one localization is not straightforward. We already hinted at the first challenge in our introduction of the two existing localization approaches as well as in their illustration in Figure 4.2: While the crop row detections provide information relative to the local frame of the vehicle, the GPS measurements are independent of the crop rows and provide information within a global reference frame. For a fused localization that estimates a pose based on both local and global information, we need to be able to relate the local crop row detections with the global GPS measurements. Second, the information provided by both modalities, i. e., the detected crop rows and the GPS positions, provides measurements of the lateral position of the vehicle with a different accuracy in the corresponding reference frames. Integrating this information in a localization can result in inconsistent pose estimates. Third, if only a standard GPS sensor with an accuracy of around $3.0\,\mathrm{m}$ is used, we need to obtain an even more accurate longitudinal position estimate with an accuracy of around $1.0\,\mathrm{m}$ or below for autonomous transition maneuvers at the headlands. Fusing crop row detections with global position information from a standard GPS sensor to obtain consistent pose estimates therefore yields the following challenges:

1. *Relate Local and Global Information*: To localize the vehicle according to both, the global and local information, we need to be able to relate the local crop row detections with the global GPS measurements.

2. *Consistent Pose Estimates*: Both sensor modalities contain information about the lateral position of the vehicle with different accuracies in different coordinate frames. Therefore, our localization approach needs consider the crop row structure of the field when fusing both measurements to obtain consistent pose estimates.

3. *Improve Longitudinal Accuracy*: The accuracy of a standard GPS sensor of around $3.0\,\mathrm{m}$ is not sufficient for autonomous transition maneuvers at the headlands. Therefore, we need to increase the accuracy of the estimated longitudinal position.

In order to realize our idea of a fused localization approach that estimates the full pose of an agricultural vehicle relative to the crops, we propose the following solutions to these challenges: Our first idea is to use a GPS-referenced map of the crop field that contains the location of each crop row of the field relative to the global GPS coordinate system as

illustrated in the middle of Figure 4.2. The pose of the vehicle is then estimated in this *common map frame*. While the GPS information can be directly integrated since the map frame is in the GPS coordinate system, integrating the local information from the detected crop rows is not straightforward. The key idea of a GPS-referenced map of crop rows is that the pose estimate of the vehicle within the map frame can be used to project the local crop row detections into the common map frame (see middle of Figure 4.2). Comparing the orientation and lateral offset of the projected observed crop rows with the crop rows of the map then yields the required heading and lateral offset measurements in the common map frame.

For this comparison, we need to determine which of the observed crop rows corresponds to which crop row of the map. This is called the *data association problem*. As can be seen from our example, determining the right association is usually not easy, since individual lines cannot be distinguished from each other. To facilitate localization relative to the observed crop rows with respect to the GPS-referenced map, we present a novel *Crop Row data association* method that determines the correct association between observed and mapped crop rows. This data association is based on the idea of *geometric consistency*, i. e., the idea that relative distances and angles should be preserved when assigning observed features to mapped features. We give a more detailed introduction of geometric consistency in Section 2.2.2.

To obtain consistent pose estimates despite the difference in accuracy of the lateral offset measurements of both sensor modalities, our key idea is to split the measurements into a heading, a lateral and a longitudinal component, if available. This allows us to select the desired sensor modality for each individual component. More explicitly, we expect highly accurate heading and lateral offset measurements from the detected crop rows and therefore use the measurements of the detected crop rows to estimate the heading and the lateral offset of the vehicle. Since standard GPS measurements do not provide heading information and we expect a lower position accuracy from the GPS, we only use the GPS measurements to estimate the longitudinal position of the vehicle.

To facilitate autonomous transition maneuvers at the headlands, we improve the GPS-based longitudinal position estimate of our fused localization method by extracting a longitudinal measurement from the local sensor data. Our key observation is that if the vehicle approaches the end of the field, the row structure used to detect the crop rows does not span the whole field of view of the sensor anymore. This fact can be used to detect the end of the crop rows and therefore the *End of the Field* (EOF). This *End of the Field detection* then provides a longitudinal measurement, since it determines the distance of the vehicle to the end of the field in the direction of the crop rows. Analogous to the detected crop rows, the detected end of the field can then be projected into the map frame and compared with the expected end of the field to obtain an accurate longitudinal position estimate of the vehicle.

In summary, our novel fused localization approach goes beyond the limitations of state-of-the-art agricultural localization approaches, since it leverages the advantages of both crop row following as well as GPS-based approaches to estimate the full pose of an agricultural vehicle relative to the crops. We successfully fuse local crop row detections with global GPS measurements by providing the following contributions to localization

in agriculture:

1. *GPS-referenced Map of Crop Rows*: For fusing crop row following and GPS-based localization, we propose to use a GPS-referenced map of crop rows to combine global GPS measurements with local crop row detections in a *common map frame*.

2. *Crop Row Data Association*: In order to derive heading and lateral offset measurements from the observed crop rows in the common map frame, we present a novel *Crop Row data association* based on geometric consistency.

3. *Split Measurements*: For consistent pose estimates we propose to split the measurements of both sensor modalities into a heading, a lateral and a longitudinal component, where lateral means orthogonal to the crop rows and longitudinal means aligned with the crop rows.

4. *End of the Field Detection*: For an improved longitudinal position estimate that enables autonomous transition maneuvers at the headlands, we present our *End of the Field detection*. This detection uses the crop row structure perceived in local sensor data to determine the distance of the vehicle to the end of the field.

In the next section we present different crop row following and GPS-based localization approaches found in the literature and discuss how they compare to our approach. We then present our fused localization approach in detail. This includes our *Crop Row data association*, the *End of the Field detection* as well as the mathematical formulations for splitting the measurements of each sensor modality into heading, lateral and longitudinal measurements and integrating them into two different localization methods. We analyze the performance of our fused localization approach and compare it to crop row following and pure GPS-based localization in our experimental evaluation on two real-world data sets. In this evaluation we use our crop row detection presented in the previous chapter to detect crop rows in the recorded image data. Our evaluation investigates in how far our fused localization approach leverages the advantages of both the crop row following as well as the GPS-based approaches found in the literature. Furthermore, we investigate, if using the *End of the Field detection* notably increases the accuracy of the longitudinal position estimate to also enable autonomous transition maneuvers at the headlands. While our fused localization approach is based on crop row following, it estimates the full pose of the vehicle – thereby going beyond the limitations of crop row following and enabling autonomous transition maneuvers at the headlands.

4.2 Related Work

Estimating the pose of an agricultural vehicle on a crop field has received much attention in the last decades. Most research focuses on estimating the pose of the vehicle to create data rich maps of the field. These maps are then used to determine properties of interest in certain areas of the field such as, for example, the amount of fertilizer or water at different

locations of the field or the health of the crops including weed pressure or infection with pests. These mapping applications usually determine the pose of the vehicle in an offline post processing step of data recorded by manually steering the agricultural vehicle across the field. In contrast to our approach, these techniques are not explicitly designed to provide an online pose estimate to guide an autonomous vehicle along the crop rows without damaging the crops. However, they still provide insights about the challenges of estimating the pose of a vehicle in a crop field. Thus, we first give an overview of localization techniques that estimate the pose of a vehicle to create maps by post-processing recorded raw data offline. Then, we discuss the crop-based localization techniques that estimate the pose of the vehicle to achieve precise autonomous navigation on agricultural fields. Finally, we also present more closely related work that uses both sensor modalities, local crop row detections as well as GPS information for autonomous navigation.

For the purpose of creating data rich maps, the vehicles used to record these data sets are usually equipped with a multitude of sensors that perceive the crops of the field. Despite that fact, most of the offline mapping approaches usually rely on an additional high precision GPS receiver mounted on the vehicle to obtain a globally accurate position of the vehicle during data recording. During post-processing, these highly accurate positions are then used integrate the collected information into a map of the crop field. For example, the approach presented by Hague et al. [2006] uses GPS-referenced data to evaluate the crop and weed density to determine on which areas of the field weed treatment is required. Another approach by Dong et al. [2017] uses high precision GPS measurements together with a robust data association on RGB images to overlay maps over time. Maps that contain information across multiple weeks or even months store valuable information such as, e. g., how fast the crops grow. This information can be used by the farmer to determine whether the crops are healthy since they grow at the expected rate, or whether further investigation is required as to why the crops are not growing at the expected rate. Another example use case is the approach by Baia et al. [2016], where they use high precision GPS measurements to collect information about different phenotypes of soybean and wheat. In fact, phenotyping has received much interest as a use case of high precision GPS-based offline mapping approaches [Mueller-Sim et al., 2017, Ruckelshausen et al., 2009, Underwood et al., 2017]. However, the authors of these works also state the challenges they encountered by relying only on high precision GPS information for pose estimation. For example, Mueller-Sim et al. [2017] had problems with the accuracy of the received high precision GPS signal, when the antenna of the vehicle was covered by foliage, e. g., due to large growing crops such as corn or large growing vegetation at the headlands such as trees. As solution, the authors propose to use additional local sensor information such as LIDAR or vision data in future work to increase robustness in case of lower accuracy of the received high precision GPS signal. Having made similar observations, Ruckelshausen et al. [2009] also plan to investigate the use of LIDAR information in future work to increase the robustness of the pose estimate. Another – often not mentioned – requirement to obtain consistent high precision GPS paths over longer periods of time is a fixed reference point to align the previously recorded GPS path with the current GPS measurements. Such an alignment is necessary to overlay data recorded at different points in time for offline mapping applications. This fixed reference also facilitates

continued deployment of an autonomous vehicle as it enables the vehicle to navigate a crop field by following the previously recorded GPS path according to the current GPS measurements [Underwood et al., 2017, Watanabe, 2018]. According to the work by Watanabe [2018] this dependency on a fixed high precision reference infrastructure can be resolved by instead using a local marker as reference. However, this only shifts the dependency from a fixed high precision GPS reference to the accuracy of the measured high precision GPS position of the local reference marker. In summary, the related work shows that under optimal conditions, i. e., no signal outages and availability of a fixed high precision GPSreference infrastructure, autonomous navigation based solely on high precision GPS data is possible. However, many of these works also suggest that for robust long-term autonomous navigation considering additional information from local sensors, such as LIDAR or vision data, is advisable.

While the high precision GPS approach seems to be mostly used for offline mapping applications, using local sensor information for pose estimation is more commonly used to guide an autonomous vehicle across a crop field without driving over crops. This makes sense, since a highly accurate heading and lateral offset estimate relative to the crops are required to steer a vehicle along the crop rows with high precision. Despite its apparent advantages for precise autonomous navigation on a crop field, using local sensor data for localization relative to the crops has received less interest than the high precision GPS localization approach. Nevertheless, research on localization relative to crops using local sensor information has made interesting advances and also uncovered some challenges. While high precision GPS signals are often used in mapping applications, crop-based localization approaches usually do not leverage any GPS information. Instead, they detect the local crops on the field, usually in form of crop rows, and estimate the pose of the vehicle relative to these detections. For example, Åstrand and Baerveldt [2005] detected the crop rows in image data and directly convert these line detections into steering commands for the autonomous vehicle. Another crop-based approach by Xaud et al. [2019] proposes the use of infrared cameras for more accurate segmentation of the vegetation features in the image data to obtain more robust localization results. However, pure crop-based approaches that directly convert the detected crop structure into steering commands can only operate the autonomous vehicle as long as crop rows are visible in the sensor data. As the vehicle approaches the end of the field less crops are visible in the sensor data until the vehicle needs to stop since no more crops are perceived. Consequently, these approaches can only guide the vehicle while it is in-row. For transition maneuvers additional information is required, such as, for example, wheel odometry as presented in the localization approach by Riggio et al. [2018]. Another idea was presented by Libby and Kantor [2011], where markers were placed at the end of each crop row. These markers can then be used as reference by the navigation system to determine when to initiate a turning maneuver as well as to realign the vehicle with the crop rows after turning. While placing markers might be feasible on small research fields, it is tedious and impractical on large production fields. In contrast, as shown by Riggio et al. [2018], turning at the headlands based on wheel odometry is possible. The results in the publication by Riggio et al. [2018] confirm that autonomous navigation on crop fields is possible without high precision GPS measurements. Another crop-based localization approach, that also

leverages wheel odometry to localize the vehicle during turning at the headlands is presented by Chebrolu et al. [2019]. The main difference to previous approaches is that the authors of this work propose to use individual detections of crops, weeds and gaps, i. e., locations where crops are missing in the row structure, instead of detections of whole crop rows. This is an interesting approach as it enables to also track the longitudinal position of the vehicle while in-row, thereby achieving a maximum global localization error of $0.17\,\mathrm{m}$. While these are interesting global localization results, there is no mention of autonomous navigation performed based on these pose estimates. Also, the authors do not provide results for the maximum error in the lateral offset of the pose estimate relative to the crops, as reported by other related work on localization for autonomous navigation [Bakker et al., 2011, Ball et al., 2017]. Thus, it remains unclear, whether this approach is suited to steer a large agricultural vehicle with only few centimeters clearance to adjacent crop rows without driving over the crops. Overall, the results in the literature show that crop-based localization is well suited for guiding a vehicle along a crop field with high precision, at least while driving in-row. The challenge for crop-based approaches is turning at the headlands when the crops are not in the field of view of the sensor and thus tracking is lost. However, as Riggio et al. [2018] and Chebrolu et al. [2019] show, estimating the pose of the vehicle based on wheel odometry during turning can be sufficient for successful localization.

In the literature we also find works that – similarly to our approach – are based on the idea of using GPS information as well as detections of crop rows to localize the autonomous vehicle. Analogue to the localization techniques used for offline mapping applications, most of these works depend on a high precision GPS receiver as primary sensor for localizing the vehicle.

This usually results in two separate localization modules, one GPS-based localization that is used for localization during transition maneuvers, and a crop row following-based localization that is used to guide the vehicle along the crop rows while traversing the field in-row [Bakker et al., 2011, Biber et al., 2012]. For example, Biber et al. [2012] use depth and color information to detect the crop rows and guide the vehicle in-row and switch to GPS-based localization for turning at the headlands. A similar approach is presented by Bakker et al. [2011], where the authors propose to use two separate localization modules. The crop row following module uses the crop row detection developed by the authors to localize the vehicle relative to the crop rows [Bakker et al., 2008]. The other localization module is based on high precision GPS information. They compare the accuracy of both approaches relative to the crops in their evaluation. As expected, the GPS-based localization reaches centimeter accuracy when using a high precision GPS. For their crop-row-following-based method, the authors report lateral errors of up to $0.1\,\mathrm{m}$ while driving in-row. They also experience an increase in the lateral error when approaching the end of the field, so that they conclude that the accuracy of the pose estimate provided by their crop row following approach is only well suited for autonomous navigation while driving in-row. These results are well aligned with the results of our evaluation of crop row detection accuracy in the previous chapter. There is also a localization approach that is more closely related to our approach than the other techniques discussed so far, since the authors also use both sensor modalities, i. e., global GPS information as well as local crop row de-

tections in one localization module [Ball et al., 2017, English et al., 2013, 2014, 2015]. In the first presentation of their approach, the authors use a particle-filter-based localization method to integrate high precision GPS signals as well as crop row tracking results to guide an autonomous vehicle along a crop field [English et al., 2013]. Later works focus on improving the accuracy and robustness of the presented localization approach [English et al., 2014, 2015]. However, in the most recent work by Ball et al. [2017], the authors state that they now use a high precision GPS sensor as primary source of information. The crop row information is only used as redundancy measurement in case of GPS outages. This shift of focus away from fusing both sensor measurements towards GPS-based localization is also evident in the experimental evaluation. Using a customer grade high precision GPS the authors report an average accuracy of around 0.2 m for the purely GPS-based localization. During GPS outages the error slowly increases up to 1.5 m after 2 min of GPS outage. This is in contrast to the lateral offset error when the approach of the authors is tracking the crop rows without GPS corrections, where the lateral offset error is around 0.1 m and stays below a maximum of 0.2 m. These results are again aligned with our results from the previous chapter for the attainable lateral accuracy when using crop row detections for pose estimation. These approaches that use high precision GPS-based localization as well as crop row following to autonomously navigate a crop field confirm the strong need of using both sensor modalities to enable precise autonomous traversal of entire crop fields. While both approaches, GPS-based localization as well as crop row following, have been investigated, evaluated and applied independently, only little research was targeted at fusing both sensor modalities into one consistent pose estimate. We therefore contribute to state-of-the-art localization for autonomous navigation on crop fields, by presenting a novel localization approach that fuses – not necessarily high precision – GPS information with crop row detections to go beyond the limitations of crop row following and achieve precise autonomous navigation of entire crop fields.

4.3 Crop Row Localization on a GPS-referenced Map

In this section, we present our novel approach for a fused localization method that fully estimates the pose of a vehicle relative to crops to enable autonomous navigation across entire fields. The goal of our localization approach is to fuse the local information from crop row detections with global position information in a common reference frame to obtain accurate pose estimates of the vehicle relative to the crops. This common reference frame is defined by a GPS-referenced map of the crop field that contains the GPS-referenced start and end points of each crop row on the field. However, fusing both sensor modalities is challenging. In this section we therefore present our novel techniques required for successfully fusing both modalities to obtain consistent pose estimates.

In order to transfer the local crop row detections into the common map frame, we need to associate each observed crop row with its corresponding crop row in our GPS-referenced map. Therefore, we first present our novel *Crop Row data association* based on geometrical consistency in the following. Then, we explain in detail how our *End of the Field detection* method leverages vegetation feature information as well as the detected

crop row structure to extract the end of the field. The detected location of the end of the field is then used to improve the longitudinal position estimate in our fused localization approach. Finally, we mathematically define our fused sensor model and explain, how we integrate information with different accuracy from different sensor modalities. We also derive the mathematical formulations required to implement our sensor model in two different localization techniques.

4.3.1 Data Association on Crop Rows

For a crop-row-based localization to be able to fuse the global information obtained from GPS data with the information of crop rows detected in the local reference frame of the vehicle, the detected crop rows need to be associated with the provided GPS-referenced set of mapped crop rows. We thus present a data association approach that determines a matching between the set of detected lines from the Pattern, i. e., the set of observations F_o, and the set of GPS-referenced mapped lines, i. e., the set of mapped features F_m. As explained in Section 2.2.1, the general solution to a data association problem can be formulated as follows:

$$\mathcal{M}^{\text{valid}} \subseteq \mathcal{M} \subset \mathcal{P}\left(F_o \times F_m\right) \qquad\qquad M^* = \underset{M \in \mathcal{M}^{\text{valid}}}{\operatorname{argmax}}\, t\left(M\right)$$

In this formulation, the subset of valid data association matches $\mathcal{M}^{\text{valid}}$ is determined by *hard requirements* for a set of data association matches $M \in \mathcal{M}$ to be valid. The Target function t assigns a value, or score, to a set of data association matches $M \in \mathcal{M}$ depending on how well this set of matches M explains the observed features given the mapped features. How this score $t\left(M\right)$ is computed depends on the *soft criteria* relevant to the given data association problem. We provide a more detailed definition of these terms and introduce different hard requirements and soft criteria based on data association techniques in the literature in Section 2.2.2.

The soft criteria for this data association problem are straightforward. Since our Pattern Hough crop row detection robustly extracts all crop row information from the given sensor data, we do not expect a large amount of false positive or false negative detections. Therefore, for this row-based data association, we only use the criterion of *Positive Information* to design our Target function, i. e., the better the observed lines overlap with their corresponding mapped lines in a set of matches $M \in \mathcal{M}$, the higher is the score $t\left(M\right)$.

For the hard requirements, we focus on preserving the geometry between features of the same set. More precisely, a set of data association matches $M \in \mathcal{M}$ is valid, only if the relative distances between the crop rows of each set are preserved. We call such a valid set of data association matches *geometrically consistent*. For a better intuition of geometrically consistent sets of data association matches, we illustrate examples for consistent and inconsistent matches in Figure 4.3. Since this data association is intended for a localization application, we can assume that the current pose estimate of the vehicle is correct up to a certain limit, i. e., we can assume that the vehicle is localized. We therefore also include the hard requirement of *locality*. This means, that we only want to

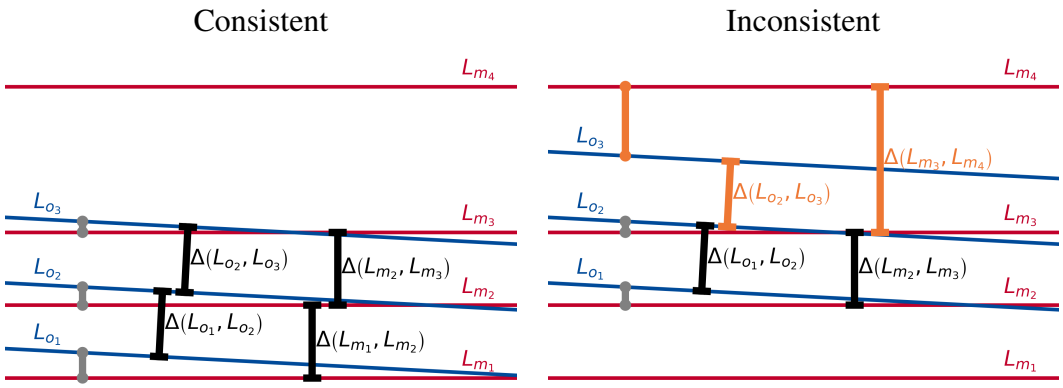

Figure 4.3: This figure shows an example for a geometrically consistent set of matches on the left and a set of geometrically inconsistent matches on the right. The set of observed lines $F_o = \{L_{o_1}, L_{o_2}, L_{o_3}\}$ is visualized as blue lines, the set of mapped lines $F_m = \{L_{m_1}, L_{m_2}, L_{m_3}, L_{m_4}\}$ are shown in red. The matches of each set are shown as gray and orange lines that connect matched line pairs. The relative distances between matched lines $\Delta(L_*, L_{**})$ of the same set are shown as black lines. The match as well as the corresponding relative distances that cause the set of matches on the right to be geometrically inconsistent are highlighted in orange.

consider data association matches, that infer a reasonable amount of pose correction onto the current pose estimate.

In the following, we first define the *Line to Line Distance* between two pairs of lines and discuss important properties of this distance measure. Based on the Line to Line distance measure and our definition of the Target function, we present our approach for computing the set of valid matches $\mathcal{M}^{\text{valid}}$. This set of valid matches $\mathcal{M}^{\text{valid}}$ only contains matches M that fulfill the *geometrical consistency* as well as the *locality* requirements. Finally, we also use this Line to Line distance measure to model our Target function t according to the *Positive Information* criterion.

Line to Line Distance Based on our definitions in the previous chapter, where we defined the lateral difference Δ_{lat} of a Pattern relative to a reference point p_{ref}, we define the *Line to Line Distance* on two lines L_{θ_1,r_1} and L_{θ_2,r_2} analogously to the definition in Eq. (3.8) to Eq. (3.10):

$$\delta^{p_{\text{ref}}}\left(L_{\theta_1,r_1}, L_{\theta_2,r_2}\right) := d\left(L_{\theta_1,r_1}, p_{\text{ref}}\right) - d\left(L_{\theta_2,r_2}, p_{\text{ref}}\right) \tag{4.1}$$

$$\Delta^{p_{\text{ref}}}\left(L_{\theta_1,r_1}, L_{\theta_2,r_2}\right) := \left|\delta^{p_{\text{ref}}}\left(L_{\theta_1,r_1}, L_{\theta_2,r_2}\right)\right|, \tag{4.2}$$

where $d\left(L, p_{\text{ref}}\right)$ is the signed point to line distance as defined in Eq. (3.2) in the previous chapter. Similar to the lateral difference measure between two Patterns, this distance measure is independent of the orientation of the lines (see Figure 4.4).

Another interesting observation is that shifting a line laterally by the amount of a *lateral shift* $s_{\text{lat}} \in \mathbb{R}$, also changes the distance of a point to the line that amount. Given a line $L_{\theta,r}$

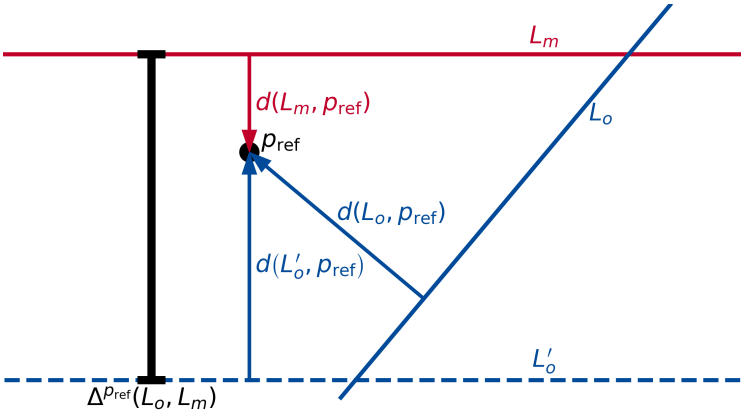

Figure 4.4: This illustrates the Line to Line Distance. The lines are drawn as solid blue and red lines. The rotated dashed blue line L'_o demonstrates that the Line to Line Distance measure is independent of the orientation of the lines relative to the reference point p_{ref}, since $d\left(L_o, p_{\text{ref}}\right) = d\left(L'_o, p_{\text{ref}}\right)$. The point to line distances of each line to the reference point p_{ref} are visualized using arrows to indicate the sign of the distance. The resulting Line to Line distance $\Delta\left(L_o, L_m\right)$ is highlighted as black solid line between L'_o and L_m.

and a lateral shift $s_{\text{lat}} \in \mathbb{R}$, we say that the line is *shifted laterally* and denote the shifted line as $L_{\theta,r}\left(s_{\text{lat}}\right) := L_{\theta,r+s_{\text{lat}}}$. According to the definition of the point to line distance in Eq. (3.2), we obtain for any point $p = (p_x, p_y)$:

$$\begin{aligned} d\left(L_{\theta,r}\left(s_{\text{lat}}\right), p\right) &:= \left[p_x \cdot \cos\left(\theta\right) + p_y \cdot \sin\left(\theta\right)\right] - \left(r + s_{\text{lat}}\right) \\ &= \left[p_x \cdot \cos\left(\theta\right) + p_y \cdot \sin\left(\theta\right)\right] - r - s_{\text{lat}} \\ &= d\left(L_{\theta,r}, p\right) - s_{\text{lat}} \end{aligned} \tag{4.3}$$

More importantly, this means that the Line to Line Distance remains constant, when two lines L_{θ_1,r_1} and L_{θ_2,r_2} are shifted using the same lateral shift $s_{\text{lat}} \in \mathbb{R}$:

$$\begin{aligned} \delta^{p_{\text{ref}}}\left(L_{\theta_1,r_1}\left(s_{\text{lat}}\right), L_{\theta_2,r_2}\left(s_{\text{lat}}\right)\right) &\overset{Eq.\ (4.1)}{=} d\left(L_{\theta_1,r_1}\left(s_{\text{lat}}\right), p_{\text{ref}}\right) - d\left(L_{\theta_2,r_2}\left(s_{\text{lat}}\right), p_{\text{ref}}\right) \\ &\overset{Eq.\ (4.3)}{=} d\left(L_{\theta_1,r_1}, p_{\text{ref}}\right) - s_{\text{lat}} - \left[d\left(L_{\theta_2,r_2}, p_{\text{ref}}\right) - s_{\text{lat}}\right] \\ &= d\left(L_{\theta_1,r_1}, p_{\text{ref}}\right) - d\left(L_{\theta_2,r_2}, p_{\text{ref}}\right) \\ &= \delta^{p_{\text{ref}}}\left(L_{\theta_1,r_1}, L_{\theta_2,r_2}\right) \end{aligned} \tag{4.4}$$

In the following, since we are computing the data association to correct the pose $\mathbf{x}_t$ of the vehicle, we use the current position p_t of the vehicle as reference point p_{ref} and therefore omit the p_{ref} parameter. Using the vehicle position as reference is desirable, as it ensures that two lines have a line to line distance of zero, if and only if they have the same signed distance from the vehicle, i. e., if they are located at the same distance from the vehicle, independent of their orientation.

Geometrically Consistent Data Association Matches Given a set of data association matches $M \in \mathcal{M}$, we can now define whether this set $M := \{(L_{o_i}, L_{m_i})\}_{i\in[1,\dots,n]}$ is *geometrically consistent* (gc) by enforcing that the distances between observed lines are similar to the distances between the matched mapped lines:

$$\mathrm{gc}\,(M) \;:=\; \forall i,j \in [1,n] :\; \left|\delta\left(L_{o_i}, L_{o_j}\right) - \delta\left(L_{m_i}, L_{m_j}\right)\right| < \varepsilon \tag{4.5}$$

Since the number of observed and mapped lines is usually not too large, the set of all geometrically consistent data association matches $\mathrm{gc}\,(\mathcal{M})$ can be explicitly computed by testing the geometric consistency of all possible sets of data association matches, i. e., by computing $\mathrm{gc}\,(\mathcal{M}) = \{M \in \mathcal{M} \mid \mathrm{gc}\,(M)\}$.

However, for a more efficient computation of the set of valid data association matches $\mathcal{M}^{\text{valid}}$, we can considerably reduce the number of sets of data association matches that need to be tested for geometric consistency based on our observation that the Line to Line Distance between two lines is preserved when shifting both lines laterally by the same amount of lateral shift $s_{\text{lat}} \in \mathbb{R}$. Based on this observation, we propose to efficiently generate sets of data association matches $M\,(s_{\text{lat}})$ that are geometrically consistent by laterally shifting all observed lines $L_o \in F_o$ over the set of mapped lines using this lateral shift parameter $s_{\text{lat}} \in \mathbb{R}$ and associating the shifted observed lines $L_o\,(s_{\text{lat}})$ with their closest mapped line, according to the Nearest Neighbor data association strategy. To ensure geometrically consistent matches, we only consider lateral shifts $s_{\text{lat}} \in S \subset \mathbb{R}$ that result in a set of matches $M\,(s_{\text{lat}})$, where the Line to Line Distance between the shifted observed line and its matched mapped line does not exceed the threshold of $\frac{1}{2}\varepsilon$. Mathematically, this parametrization $M\,(s_{\text{lat}})$ of the set of data association matches using the lateral shift $s_{\text{lat}} \in S$ is defined as follows:

$$\begin{aligned}
L_{m_*}\,(L_o, s_{\text{lat}}) \;&:=\; \underset{L_m \in F_m}{\operatorname{argmin}}\, \Delta\,(L_o\,(s_{\text{lat}}), L_m) \\
M\,(s_{\text{lat}}) \;&:=\; \{(L_o, L_{m_*}\,(L_o, s_{\text{lat}}))\} \in \mathcal{M} \\
s_{\text{lat}} \in S \;&\Leftrightarrow\; \forall\,(L_o, L_m) \in M\,(s_{\text{lat}}) : \Delta\,(L_o\,(s_{\text{lat}}), L_m) < \frac{1}{2}\varepsilon \\
s_{\text{lat}} \in S \;&\Rightarrow\; \mathrm{gc}\,(M\,(s_{\text{lat}}))
\end{aligned} \tag{4.6}$$

Since it is crucial for our approach that we only consider geometrically consistent matches, we proof that the last implication, Eq. (4.6), actually holds for all $s_{\text{lat}} \in S$.

Proof. Let $s_{\text{lat}} \in S$ be a lateral shift parameter so that $M\,(s_{\text{lat}}) = \{(L_{o_i}, L_{m_i})\}_{i\in[1,n]}$ is a set of data association matches where all matches $(L_{o_i}, L_{m_i}) \in M\,(s_{\text{lat}})$ satisfy the condition $\Delta\,(L_{o_i}\,(s_{\text{lat}}), L_{m_i}) < \frac{1}{2}\varepsilon$. An equivalent formulation of this condition is:

$$\begin{aligned}
\Delta\,(L_{o_i}\,(s_{\text{lat}}), L_{m_i}) < \frac{1}{2}\varepsilon &\Leftrightarrow \delta\,(L_{o_i}\,(s_{\text{lat}}), L_{m_i}) \in \left(-\frac{1}{2}\varepsilon, \frac{1}{2}\varepsilon\right) \\
&\Leftrightarrow d\,(L_{o_i}\,(s_{\text{lat}}), p_{\text{ref}}) - d\,(L_{m_i}, p_{\text{ref}}) \in \left(-\frac{1}{2}\varepsilon, \frac{1}{2}\varepsilon\right)
\end{aligned} \tag{4.7}$$

Given any two pairs of matches $(L_{o_i}, L_{m_i}), \left(L_{o_j}, L_{m_j}\right) \in M(s)$, we conclude that the difference between the Line to Line distance of the i-th and j-th matches is limited as follows:

$$
\begin{aligned}
& \left|\delta\left(L_{o_i}, L_{o_j}\right) - \delta\left(L_{m_i}, L_{m_j}\right)\right| \\
\overset{Eq.\ (4.4)}{=} & \left|\delta\left(L_{o_i}\left(s_{\text{lat}}\right), L_{o_j}\left(s_{\text{lat}}\right)\right) - \delta\left(L_{m_i}, L_{m_j}\right)\right| \\
\overset{Eq.\ (4.1)}{=} & \left|d\left(L_{o_i}\left(s_{\text{lat}}\right), p_{\text{ref}}\right) - d\left(L_{o_j}\left(s_{\text{lat}}\right), p_{\text{ref}}\right) - \left[d\left(L_{m_i}, p_{\text{ref}}\right) - d\left(L_{m_j}, p_{\text{ref}}\right)\right]\right| \\
\overset{Eq.\ (4.7)}{=} & \Big|\underbrace{d\left(L_{o_i}\left(s_{\text{lat}}\right), p_{\text{ref}}\right) - d\left(L_{m_i}, p_{\text{ref}}\right)}_{\in\left(-\frac{1}{2}\varepsilon, \frac{1}{2}\varepsilon\right)} - \underbrace{\left[d\left(L_{o_j}\left(s_{\text{lat}}\right), p_{\text{ref}}\right) - d\left(L_{m_j}, p_{\text{ref}}\right)\right]}_{\in\left(-\frac{1}{2}\varepsilon, \frac{1}{2}\varepsilon\right)}\Big| \\
< & \quad \varepsilon
\end{aligned}
$$

This means that

$$
\forall i, j \in [1, n]: \qquad \left|\delta\left(L_{o_i}, L_{o_j}\right) - \delta\left(L_{m_i}, L_{m_j}\right)\right| < \varepsilon.
$$

And therefore, the set $M(s_{\text{lat}})$ is geometrically consistent as defined in Eq. (4.5) for all $s_{\text{lat}} \in S$. □

For a better intuition for the lateral shift parametrization, we show four examples for different lateral shift parameters in Figure 4.5 in the bottom two rows. The shift parameters s_1 and s_2 shown in the second row result in matches with large distances (highlighted in orange) and therefore $s_1, s_2 \notin S$. These large distances indicate that the set of matches is not geometrically consistent. In contrast, the shifts s_3, s_4 in the bottom row result in matches that only have small distances and therefore $s_3, s_4 \in S$ yield geometrically consistent sets of matches.

Considering the fact that the set of observed and mapped lines are quite sparsely distributed, similar lateral shift parameters result in the same set of data association matches, since for a small lateral shift, the closest mapped line remains the same for all observed lines. It therefore makes sense to only consider a discrete set of lateral shifts that result in different sets of data association matches. Our key idea for determining these discrete lateral shift parameters $s_i \in S$ is that we can use a single observed line as *reference line* $L_{o_{\text{ref}}} \in F_o$ and match it to all mapped lines $F_m = \{L_{m_i}\}_{i \in [1, |F_m|]}$. The resulting signed Line to Line Distance between the i-th mapped line L_{m_i} and the reference line $L_{o_{\text{ref}}}$ then defines the lateral shift parameter $s_i := \delta\left(L_{m_i}, L_{o_{\text{ref}}}\right)$. If the i-th shift s_i is contained in S and therefore yields a geometrically consistent set of data association matches $M(s_i)$, we add it to the set of discrete and geometrically consistent shift parameters S^{gc}, so that $S^{\text{gc}} := \left\{s_1, \ldots, s_{|F_m|}\right\} \cap S$. This yields our set of geometrically consistent data association matches $\mathcal{M}^{\text{gc}} := \left\{M(s_{\text{lat}}) \mid s_{\text{lat}} \in S^{\text{gc}}\right\}$.

Since our crop row data association is designed for application in a localization technique, correcting the pose relative to the crop rows closer to the vehicle is more important than the crop rows at the edge of the field of view. Therefore, we define the reference line $L_{o_{\text{ref}}} \in F_o$ to be the observed line that is closest the current position of the vehicle p_t:

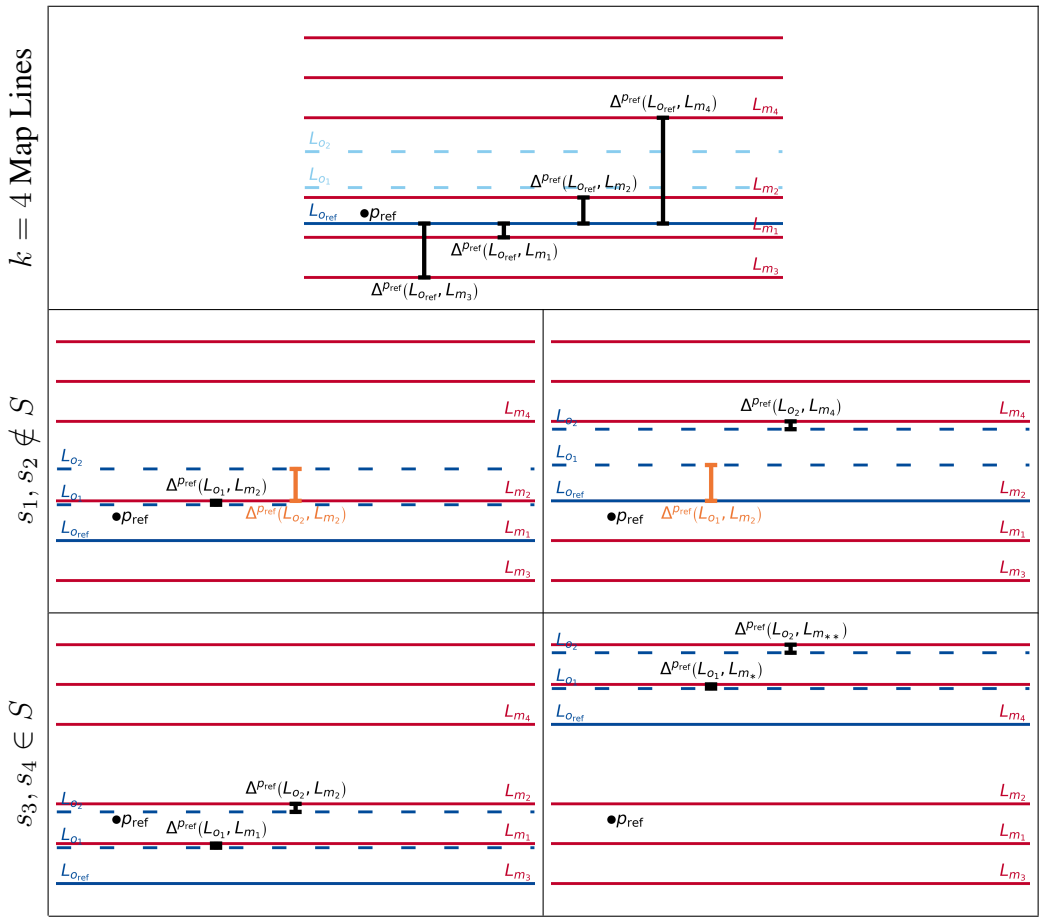

Figure 4.5: This figure shows an example for a set of valid data association matches $\mathcal{M}^{\text{valid}}$ for $k = 4$. The reference line $L_{o_{\text{ref}}}$ (solid blue) is the observed line closest to the position of the vehicle p_{ref} (black point). The $k = 4$ map lines L_{m_1}, L_{m_2}, L_{m_3} and L_{m_4} that are closest to the reference line $L_{o_{\text{ref}}}$ according to the Line to Line distance are shown in red. The other observed lines L_{o_1} and L_{o_2} are shown as blue dashed lines. Shifting the reference line $L_{o_{\text{ref}}}$ onto each of the 4 closest mapped lines yields a different shift parameter s_1, s_2, s_3 and s_4 respectively. The first two shifts s_1 and s_2 results in matches (L_{o_2}, L_{m_2}) for s_1 and (L_{o_1}, L_{m_2}) for s_2 that exceed the Line to Line distance threshold (orange). Therefore $s_1, s_2 \notin S$ and the corresponding set of matches $M(s_1), M(s_2) \notin \mathcal{M}^{\text{valid}}$. The sets of valid matches for this example is therefore $\mathcal{M}^{\text{valid}} = \{M(s_3), M(s_4)\}$.

$$L_{o_{\text{ref}}} := \underset{L_o \in F_o}{\operatorname{argmin}} d\left(L_o, p_t\right) \tag{4.8}$$

Towards satisfying our second requirement of only considering matches that infer a reasonable amount of pose correction, we constrain the number of considered lateral shifts $s_i \in S^{\text{gc}}$ – and therefore the number of considered sets of data associations by only using the k map lines $L_{m_1}, \ldots, L_{m_k}$ that are closest to the reference line $L_{o_{\text{ref}}}$. Restricting the number of considered data association matches to the k closest mapped line matches limits the amount of lateral shift considered to a maximum of $s_k = \delta\left(L_{m_k}, L_{o_{\text{ref}}}\right)$ which is directly correlated to the amount of inferred lateral pose correction. Combining the locality and the geometrical consistency requirements, we define the set of *valid lateral shifts* S^{valid} as the subset of shifts of $s_1, \ldots, s_k$ that are also contained in S^{gc}, so that $S^{\text{valid}} := \{s_1, \ldots, s_k\} \cap S^{\text{gc}}$. Consequently, the set of valid data association matches $\mathcal{M}^{\text{valid}}$, that are both geometrically consistent and local is then defined as $\mathcal{M}^{\text{valid}} := \left\{M\left(s_{\text{lat}}\right) \mid s_{\text{lat}} \in S^{\text{valid}}\right\}$. We give a detailed example for this definition of the set of valid data association matches in Figure 4.5.

Measuring the Overlap between Observed and Mapped Lines For designing our Target function according to the Positive Information criterion, we need to measure how well a given set of data association matches $M \in \mathcal{M}^{\text{valid}}$ overlaps the observed crop rows F_o onto the mapped crop rows F_m. Based on the parametrization of a two-dimensional line $L_{\theta,r}$ into its orientation θ and offset from the origin r, we know that two lines are perfectly overlapping, i. e., they are the same lines, if their corresponding parameters are equal. However, since crop rows are usually sown in approximately parallel lines, the angular difference between the matched crop rows will always be approximately constant, i. e., $|\theta_o, \theta_m| \approx c \; \forall \left(L_o, L_m\right) \in F_o \times F_m$. Since this yields the same angular difference value $|\theta_o - \theta m| \approx c$ independent of the set of data association matches M this measure does not yield any discriminative information and should therefore not be considered when designing the values of our Target function $t\left(M\right)$. Thus, we measure how well a set of data association matches $M \in \mathcal{M}^{\text{valid}}$ overlaps the observed lines onto the mapped lines according to the second line parameter, the offset parameter r. Here, we leverage again our definition of the Line to Line distance and define that an observed line overlaps well with its mapped line, if they have the same signed distance from the vehicle, i. e., if $\Delta\left(L_o, L_m\right)$ is small. These considerations lead to the following definition of our Target function:

$$t\left(M\right) := -\sum_{\left(L_o, L_m\right) \in M} \Delta\left(L_o, L_m\right) \in \mathbb{R}_{\leq 0} \tag{4.9}$$

Since $\Delta\left(L_o, L_m\right)$ is a distance measure, smaller values correspond to a better overlap between observed and mapped data. We therefore use the negative sum of the Line to Line Distance over all matches in M to define the score $t\left(M\right)$. This definition yields a higher

score, i. e., values closer to zero, for matches that have a small Line to Line Distance and a lower score for matches with a larger Line to Line Distance.

Using our definition of the valid set of data association matches $\mathcal{M}^{\text{valid}}$, as set of local and geometrically consistent data association matches as well as the definition of our Target function t, our data association approach can be summarized as follows:

$$S^{\text{valid}} := \{s_1, \ldots, s_k\} \cap S^{\text{gc}} \subset \mathbb{R} \qquad \mathcal{M}^{\text{valid}} := \left\{ M\left(s_{\text{lat}}\right) \mid s_{\text{lat}} \in S^{\text{valid}} \right\} \subset \mathcal{M}$$

$$t\left(M\right) := - \sum_{(L_o, L_m) \in M} \Delta\left(L_o, L_m\right) \qquad M^* := \underset{M \in \mathcal{M}^{\text{valid}}}{\operatorname{argmax}}\, t\left(M\right)$$

In this section, we introduced our *Crop Row data association* technique that is guaranteed to find geometrically consistent sets of matches. A consistent data association between the detected Pattern of crop rows and a GPS-referenced map of crop rows is a crucial requirement when fusing global GPS position information with the local information of a detected crop row Pattern. While the crop rows provide accurate heading and lateral information, they inherently do not contain longitudinal information. However, when the vehicle approaches the end of the field so that the end of the crop row structure is also visible in the sensor data, this information can be used to obtain longitudinal information about the position of the vehicle as well. In the next section, we therefore present our approach for detecting the end of the field in local sensor data.

4.3.2 Detecting the End of the Field

In this section, we focus on the longitudinal measurement δ_{long} required to correct the pose estimate in a localization algorithm in direction of the crop rows. To this end, we first need to detect the end of the field in the provided sensor data. Then, we explain how we use this information as well as the measurements from the GPS to obtain a longitudinal measurement δ_{long} for the localization algorithm.

While the detected crop row Pattern $P_{\theta,s,o}$ contains information about the orientation (θ) and lateral position (s, o) of the crop rows, it does not provide any information about the length of the crop rows. This is not critical when only driving in-row, where the crop rows usually span the whole field of view of the sensor. However, as the vehicle approaches the end of the field, the crop rows perceived in the sensor data do not necessarily span the whole field of view anymore. This means, that an end of the crop row structure is visible in the sensor data, which defines the end of the field. We show an example for the end of the field in sensor data on the left of Figure 4.6. Therefore, by explicitly determining the end of the crop row structure, which we call the *extent* of a Pattern, we also detect the end of the field. To determine the end of the crop row structure, the extent of a Pattern, we need to compare the detected Pattern with the perceived information about the crop row structure. This information is contained in form of the vegetation feature distribution in our Feature Map. When comparing the Feature Map with the Pattern as shown on the right of Figure 4.6, the extensions of the Pattern are clearly visible as the border between

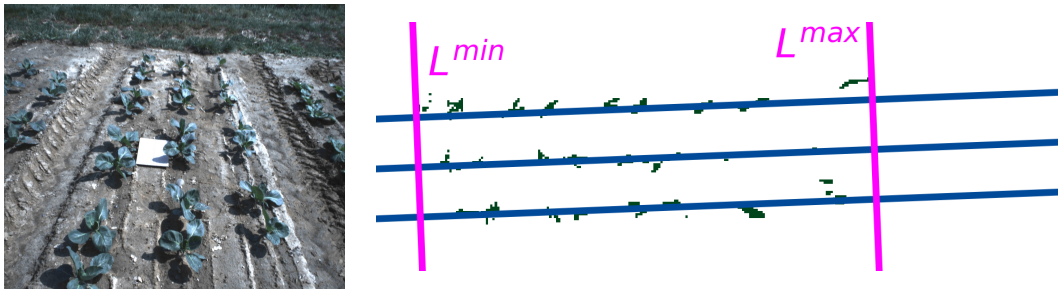

Figure 4.6: This figure shows the end of the field in the raw image data on the left and on the processed vegetation information on the right. We visualize the vegetation features in the Feature Map as dark green cells. The detected Pattern $P_{\theta,s,o}$ is shown using blue lines. The Pattern extensions $L^{\min}$ and $L^{\max}$, i. e., the lines orthogonal to the Pattern that include all parts of the Pattern that are supported by the vegetation information of the Feature Map, are illustrated as magenta lines. The line further away from the vehicle $L^{\max}$ then represents the end of the field.

the parts where the Pattern lines are *supported* by the vegetation feature distribution and where they are not *supported*.

Based on this observation, we therefore propose an *End of the Field detection* approach by distinguishing *supported* and unsupported parts of the detected Pattern, where we call the *supported* parts of a Pattern a *Pattern segment*. In the following, we properly define these *supported* parts of the Pattern, the Pattern segments, and then use this definition to determine the *extensions* of a Pattern $L^{\min}$ and $L^{\max}$, where the maximum extension line $L^{\max}$, i. e., the line further away from the vehicle, then represents the detected end of the field.

To determine the supported parts of a Pattern based on the information contained in the Feature Map, we transfer the continuous definition of the Pattern $P_{\theta,s,o}$ into the grid representation of the Feature Map by computing the set of *Pattern cells* $\mathcal{C}_P$, which contains all cells c_P of the Feature Map that are passed by a line of the Pattern. We illustrate this set of Pattern cells $\mathcal{C}_P$ as blue cells in Figure 4.7. To determine the supported parts of the Pattern, the Pattern segments, we first define the *support* on each individual Pattern cell $c_P \in \mathcal{C}_P$. We then use this definition of the support of individual Pattern cells to cluster well supported Pattern cells into Pattern segments.

Support of a Pattern Cell Our key idea is that we can measure the support $\sup(c_P)$ of any Pattern cell c_P by measuring if the vegetation features of the crop row at this position are distributed according to the detected crop row structure. Using the detected spacing between the crop rows, i. e., the pattern spacing parameter s of the detected Pattern $P_{\theta,s,o}$, all features that lie within a lateral distance of half the spacing belong to the corresponding crop row. We illustrate this in Figure 4.7 using a solid light blue line for the $\frac{1}{2}s$ threshold to separate the vegetation features of neighboring crop rows. We denote the set of vegetation feature cells that belong to the Pattern cell c_P as $\mathcal{C}_{\frac{1}{2}s}(c_P)$ (light green cells in Figure 4.7).

The support of a Pattern cell c_P is then computed by accumulating all vegetation feature

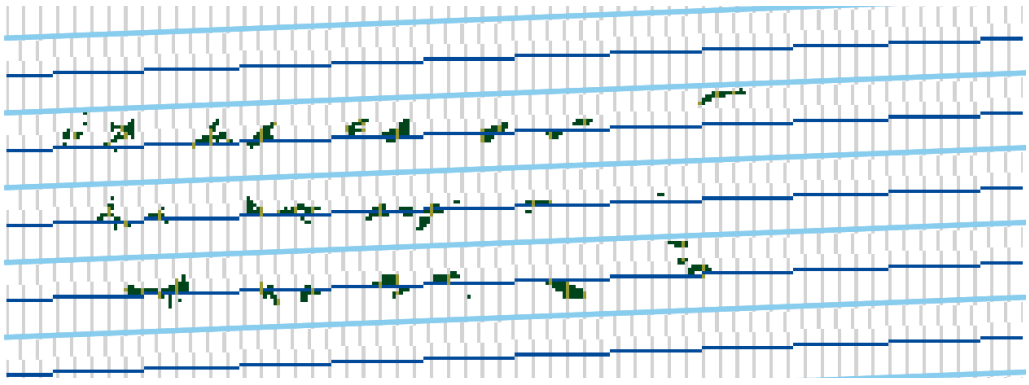

Figure 4.7: This figure shows the Pattern cells $\mathcal{C}_P$ of the detected Pattern $P_{\theta,s,o}$ as blue cells. We also visualize the relevant threshold of $\frac{1}{2}s$ that separates the vegetation features (dark green cells) of adjacent crop rows as solid light blue lines. We show the Feature Map cells within this threshold for every fifth pattern cell c_P in gray and highlight the cells that contain vegetation feature cells in light green. For each Pattern cell c_P, the highlighted light green vegetation feature cells are contained in the set $\mathcal{C}_{\frac{1}{2}s}(c_P)$

cells that belong to this Pattern cell c_P. Since vegetation features closer to the detected crop row, and therefore closer to the Pattern cell, support the detected Pattern better, we reward vegetation features closer to the Pattern cell c_P by using a larger weight and punish vegetation features that have a distance close to $\frac{1}{2}s$, since this means that they lie between two crop rows and therefore oppose the detected Pattern. Vegetation features that lie between those two extremes, i. e., at a distance of around one quarter of the spacing, do neither support nor oppose the detected crop row structure. Therefore, we assign them a weight of zero. For smooth transitions between those weights, we decide to use a sigmoid function $\mathcal{S}_s$ that is scaled and shifted according to the detected spacing parameter s and our weight value requirements as shown in Figure 4.8.

Additionally, we also need to take into account that a Pattern line might be slightly offset to individual vegetation feature clusters, i. e., crops, due to irregular plant growth, or crops that are not sown in perfectly straight lines. Thus, we also shift the Pattern cell c_P in lateral direction of the detected crop rows, i. e., in direction of θ, to find the *best supported cell* $c^*(c_P)$ within a window $\mathcal{C}_{\frac{1}{4}s}(c_P)$ of a quarter pattern spacing s around the Pattern cell c_P. Per definition, this best supported cell $c^*(c_P)$ either lies at the center of the vegetation cluster or at the edge of the window $\mathcal{C}_{\frac{1}{4}s}(c_P)$ if the vegetation cluster is outside of the window $\mathcal{C}_{\frac{1}{4}s}(c_P)$. We therefore call this best supported cell the *crop row center* of the Pattern cell c_P. The support computed at the crop row center $c^*(c_P)$ then defines the support $\sup(c_P)$ of the according Pattern cell c_P:

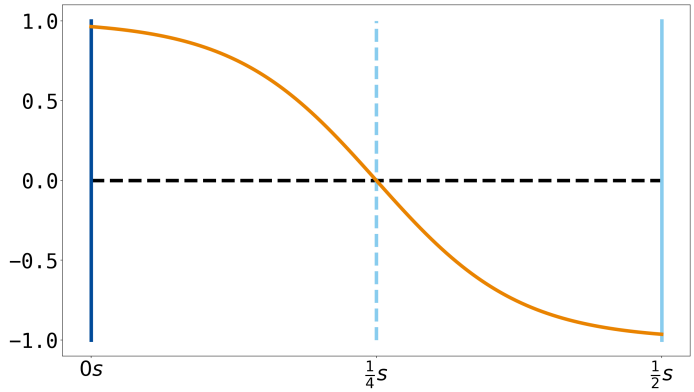

Figure 4.8: This show the sigmoid function $\mathcal{S}_s$ (orange) used for computing the support of a Pattern cell c_P. It is scaled, shifted and mirrored, so that the weights are computed according to our requirements: Vegetation features that coincide with the detected Pattern (blue line), i. e., vegetation feature cells with a distance of 0, support the detected Pattern best and therefore are assigned the highest weight of 1. In contrast, vegetation features close to the maximum distance of $\frac{1}{2}s$ (solid light blue line) are punished with a low negative weight down to -1. Vegetation features that lie in the middle at around $\frac{1}{4}s$ (dashed light blue line) do not contain much information and therefore only contribute by a small amount with weights close to 0.

$$
\begin{aligned}
\sup\left(c_P\right) &:= \sup\left(c^*\left(c_P\right)\right) \\
c^*\left(c_P\right) &:= \operatorname*{argmax}_{c_{\text{shift}} \in \mathcal{C}_{\frac{1}{4}s}(c_P)} \sup\left(c_{\text{shift}}\right) \\
\sup\left(c_{\text{shift}}\right) &:= \sum_{c \in \mathcal{C}_{\frac{1}{2}s}(c_P)} \mathcal{S}_s\left(\left|c - c_{\text{shift}}\right|\right)
\end{aligned}
$$

In Figure 4.9, we illustrate the shifted cells $\mathcal{C}_{\frac{1}{4}s}\left(c_P\right)$ considered when computing the support of a Pattern cell c_P as well as the resulting crop row center $c^*\left(c_P\right)$.

Supported Part of a Pattern – The Pattern Segment Using the support $\sup\left(c_P\right)$ of individual Pattern cells, we can now determine which parts of each line of the Pattern are supported by the crop row structure. Since the support $\sup\left(c_P\right)$ of a Pattern cell c_P is by design positive, if the vegetation features around the Pattern cell P show a crop row structure that supports the detected crop row Pattern, we say that a Pattern cell *is supported*, if $\sup\left(c_P\right)$ is larger than zero. All supported Pattern cells of our example are highlighted in Figure 4.10.

We use this definition of a supported Pattern cell to cluster all supported Pattern cells into Pattern segments, where each Pattern segment only consists of Pattern cells that belong to the same line. We also split each line into multiple Pattern segments, if there is a large gap between adjacent supported Pattern cells, i. e., if there is a large unsupported

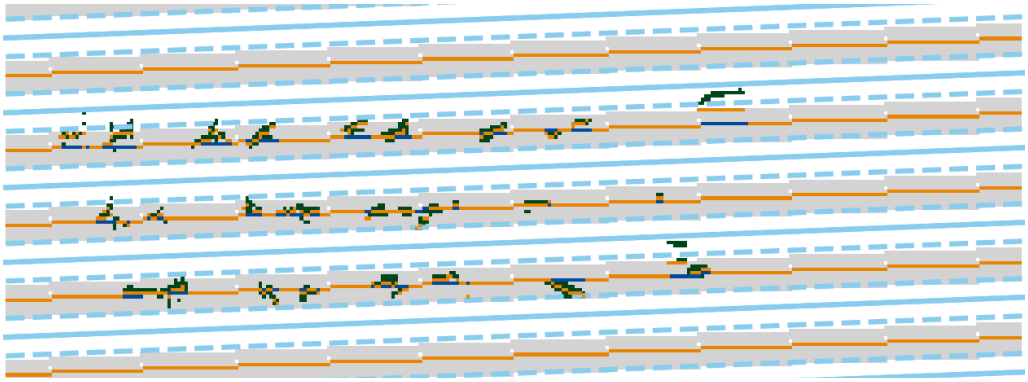

Figure 4.9: This figure shows the crop row centers for all Pattern cells $c_P \in \mathcal{C}_P$ (orange). We also visualize the shifted cells $\mathcal{C}_{\frac{1}{4}s}(c_P)$ considered during support computation in gray as well as the corresponding distance threshold $\frac{1}{4}s$ using a dashed light blue line. The threshold $\frac{1}{2}s$ that separates vegetation features of adjacent crop rows is again shown as solid light blue line. The vegetation feature cells of the Feature Map are shown in dark green and the Pattern cells in blue. Most crop row centers coincide with the Pattern cells due to the sparse distribution of vegetation features.

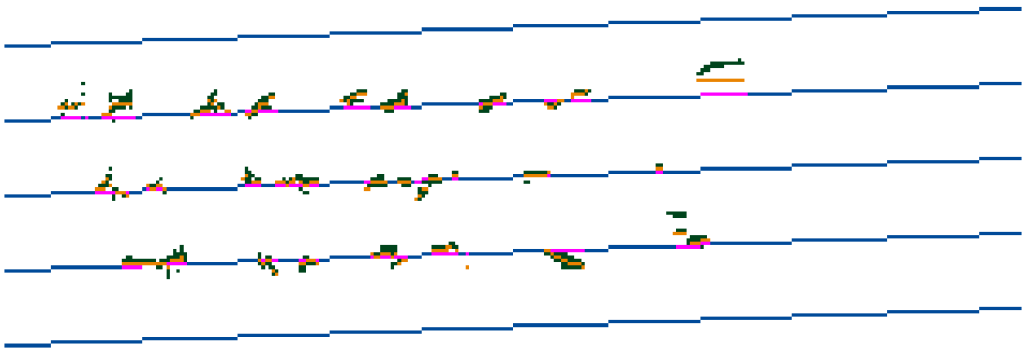

Figure 4.10: This figure shows the supported Pattern cells (magenta) as well as their corresponding crop row center cells (orange). Unsupported Pattern cells or cells without vegetation feature information and therefore a support value equal to 0 are shown in blue. The vegetation feature cells of the Feature Map are shown in dark green.

part along the line of more than $1.0\,\mathrm{m}$. A Pattern Segment S_P is therefore a cluster of mostly supported Pattern cells: $S_P \subset \mathcal{C}_P$.

Due to noisy vegetation feature distribution as well as wild vegetation growing at the headlands, this segmentation step can also lead to small fragmented Pattern segments. We therefore choose one Pattern segment as representative for each line L in the detected Pattern $P_{\theta,s,o}$ according to the following criteria: First, the Pattern segment should exceed a minimum *length* $l\left(S_P\right)$ of $1.5\,\mathrm{m}$ to ensure that it corresponds to a large part of a crop row. Second, it should also be well supported and therefore contain a large amount of supported cells depending on its length, i. e., it should have a high enough *density* $d\left(S_P\right)$ of at least 0.2 supported cells per meter. Third, to ensure that the Pattern Segment does not correspond to a large unstructured cluster of vegetation features at the headlands, it should also contain only few Pattern cells with a negative support value, i. e., the *invalid-valid ratio* $r\left(S_P\right)$ of the number of Pattern cells with negative support value divided by the number supported Pattern cells should also not be larger than $90\,\%$. These properties are computed on the Pattern cells c_P of the Pattern Segment S_P as follows:

$$
\begin{aligned}
l\left(S_P\right) &:= \max_{c'_P, c''_P \in S_P} \left\| c'_P - c''_P \right\| \\
d\left(S_P\right) &:= \frac{\left|\left\{c_P \in S_P \mid \sup\left(c_P\right) > 0\right\}\right|}{\left|S_P\right|} \\
r\left(S_P\right) &:= \frac{\left|\left\{c_P \in S_P \mid \sup\left(c_P\right) < 0\right\}\right|}{\left|\left\{c_P \in S_P \mid \sup\left(c_P\right) > 0\right\}\right|}
\end{aligned}
$$

If more than one Pattern Segment satisfies these conditions, we choose the Pattern Segment that is closer to the position of the vehicle, since the vehicle is still within the field and therefore the crop rows are located closer to the vehicle than the headlands. If such a representative Pattern Segment S_P for a Pattern line $L \in P_{\theta,s,o}$ is found, it defines the *support* of this Pattern line $\sup\left(L\right) = S_P$ as a set of mostly supported Pattern cells. If no such representative is found, the corresponding Pattern line L is not supported and therefore not considered in the following steps. This yields a finite set of supported Pattern lines $L_1, \ldots, L_n$ with corresponding Pattern Segments $\sup\left(L_1\right), \ldots, \sup\left(L_n\right)$ which represents the support of the whole Pattern $\sup\left(P_{\theta,s,o}\right) := \left\{\sup\left(L_1\right), \ldots, \sup\left(L_n\right)\right\}$. An illustration of these supported lines and their Pattern segments is shown in Figure 4.11.

The Pattern Extensions Using the support of the Pattern $\sup\left(P_{\theta,s,o}\right)$ extracting the extent of the Pattern is now straightforward: We shift a line $L_{\theta^\perp,r}$ that is orthogonal to the direction of the Pattern across the entire Feature Map and determine the two lines $L^{\min}$ and $L^{\max}$ that are closest to and furthest from the current position p_t of the vehicle while still passing through a cell in the Pattern support $\sup\left(P_{\theta,s,o}\right)$:

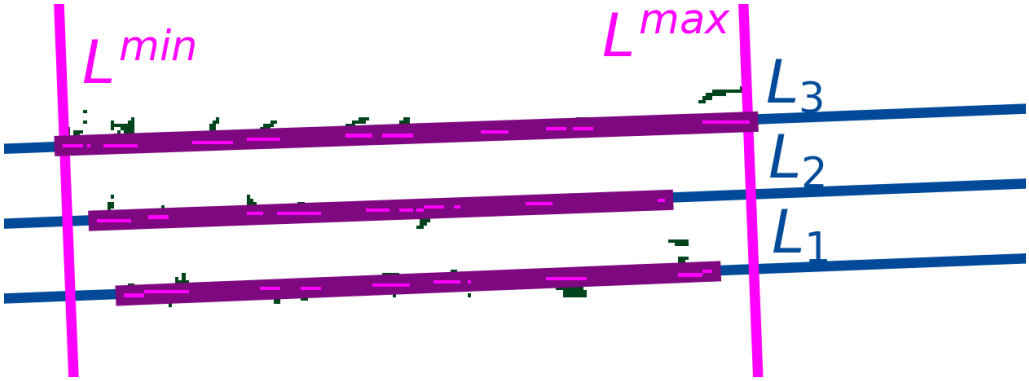

Figure 4.11: This figure shows the Pattern Segments (purple) based on the supported Pattern cells (magenta). The resulting supported Pattern lines L_1, L_2 and L_3 are shown in blue. The Pattern extensions computed from this information is shown as magenta lines L^{min} and L^{max}. The vegetation features cells of the Feature Map are shown in dark green.

$$
\begin{aligned}
R^{\text{valid}} &:= \left\{ r \in \mathbb{R} \mid \exists\, c_P \in \sup\left(P_{\theta,s,o}\right) \cap L_{\theta^{\perp},r} \right\} \subset \mathbb{R} \\
L^{\text{min}} &:= \underset{r \in R^{\text{valid}}}{\operatorname{argmin}}\, d\left(L_{\theta^{\perp},r}, p_t\right) \\
L^{\text{max}} &:= \underset{r \in R^{\text{valid}}}{\operatorname{argmax}}\, d\left(L_{\theta^{\perp},r}, p_t\right)
\end{aligned}
$$

We show the resulting minimum and maximum Pattern extensions L^{min} and L^{max} in Figure 4.11.

Since the Pattern extensions are well defined, we can always determine the maximum Pattern extension L^{max}. However, the maximum extension of the Pattern L^{max} is not always the same as the end of the field. In fact, while the vehicle is traversing the field in-row, the Pattern extensions usually coincide with the edges of the Feature Map since the crop rows span the whole field of view of the sensor. These extensions do not represent the end of the field, but rather the end of the Feature Map. Therefore, we only say that the end of the field was detected if the following two conditions hold: First, the current position p_t of the vehicle is close enough to the end of the field so that the vehicle could perceive the end of the field. Second, the Pattern extensions do not coincide with the edges of the Feature Map. If both conditions hold, the maximum extension L^{max} actually represents the end of the field and thus we say that the end of the field in form of the maximum Pattern extension L^{max} was detected.

In this section, we presented our *End of the Field detection* to provide additional longitudinal information obtained from the crop row structure perceived in the local frame of the vehicle. In the following section, we explain in-depth how our *End of the Field detection* as well as the *Crop Row data association* from the previous section can be used to consistently fuse the information from the GPS in the global reference frame with the detected crop row structure in the local frame of the vehicle.

4.3.3 Fusing GPS data and Crop Row Detections for Localization

The problem of localizing a vehicle in an environment based on relative motion measurements and sensor measurements is usually formulated as follows: Given a map m of the environment as well as time-synchronized relative motion measurements u_t and sensor measurements z_t for every time step t, a localization algorithm should estimate the pose $\hat{\mathbf{x}}_t$ of the vehicle inside the map m for every time step t. Most localization techniques intended for online localization compute this pose estimate $\hat{\mathbf{x}}_t$ iteratively. Given the previous pose estimate $\hat{\mathbf{x}}_{t-1}$ these iterative techniques use the current relative motion measurement u_t and the current sensor measurement z_t to derive an estimate for the current pose $\hat{\mathbf{x}}_t$ of the vehicle. First, the relative motion measurement u_t is used to predict the pose estimate $\bar{\mathbf{x}}_t$ depending on the previous pose estimate $\hat{\mathbf{x}}_{t-1}$. This step is therefore called the *Prediction Step*. Next, the sensor measurement z_t is used to correct the predicted pose estimate $\bar{\mathbf{x}}_t$, which yields the pose estimate $\hat{\mathbf{x}}_t$ for this time step t. This second step is called the *Correction Step*.

Since our approach is focused on localization of a ground vehicle, estimating the pose of the vehicle on the two-dimensional plane is sufficient. We therefore define the pose of the vehicle $\mathbf{x}$ as three-dimensional vector consisting of a heading parameter $\theta \in [-\pi, \pi]$ and a position parameter $p = (x, y)^T \in \mathbb{R}^2$. The heading parameter describes the orientation and the position parameter the translation of the vehicle relative to the coordinate system defined by the map m of the environment. We therefore call this coordinate system the *map frame*. The pose of the vehicle $\mathbf{x}$ also defines a local coordinate system relative to the map frame. This *local frame* has its origin at the position p of the vehicle and is rotated so that the x-axis is facing in the direction of the heading θ of the vehicle.

Relative motion measurements u_t that measure the difference between the previous pose $\mathbf{x}_{t-1}$ and the current pose $\mathbf{x}_t$, are usually obtained from wheel odometry and an IMU by directly formulating the raw data as relative transforms between both vehicle poses with parameters $\delta\theta_t$ for the measured change in the heading of the vehicle and δx_t and δy_t for the measured change in position of the vehicle. In the following we use $\oplus$ to denote the operation that transforms the previous pose $\mathbf{x}_{t-1}$ onto the current pose $\mathbf{x}_t$ using u_t, i. e., $\mathbf{x}_t = \mathbf{x}_{t-1} \oplus u_t$, so that the heading of the vehicle is first rotated by $\delta\theta_t$ and then translated according to $(\delta x_t, \delta y_t)^T$.

The more challenging part is to derive the sensor measurement z_t from the raw data of the sensors that are intended to be used to correct the pose of the vehicle. In our scenario, these are the GPS position data in form of a position p_t^{GPS} in the map frame as well as the detected Pattern P_t from the perceived image data I_t in the local frame. Along with the definition of the sensor measurement z_t, we also need to define a measurement prediction function $h(\mathbf{x}_t)$ that predicts a sensor measurement $\bar{z}_t$ based on a given vehicle pose $\mathbf{x}_t$ and the map m. This prediction function $h(\mathbf{x}_t)$ is required to compute the residual $y_t := z_t - h(\mathbf{x}_t)$ that describes how the measured information deviates from the expected information and therefore defines how the predicted pose estimate $\bar{\mathbf{x}}_t$ needs to be corrected.

In the following, we therefore first explain how we define the z_t and the measurement prediction function $h(\mathbf{x}_t)$ to fuse the information provided from the GPS position data

p_t^{GPS} as well as the detected Pattern P_t. Then, we demonstrate how this fused sensor measurement can be used in the Correction Step of a localization algorithm by deriving the mathematical formulations for two different localization techniques.

The Sensor Measurement Our key observation for fusing GPS position information p_t^{GPS} with the information of the detected Pattern $P_t = P_{\theta,s,o}$ is that the detected Pattern can only provide heading and lateral corrections, since it consists of lines that have an orientation θ and a lateral offset depending on the s and o parameters. On the other hand, while the GPS position information cannot provide heading information, it can provide lateral and longitudinal corrections. We therefore propose to define the sensor measurement z_t as a three-dimensional vector $z_t := (z_{\theta,t}, z_{\text{lat},t}, z_{\text{long},t})^T \in [-\pi, \pi] \times \mathbb{R}^2$. The first component $z_{\theta,t}$ describes the orientation of the crop rows relative to the heading θ_t of the vehicle. The second component $z_{\text{lat},t}$ defines the lateral offset, or distance, of the crop rows to the position p_t of the vehicle. The third component $z_{\text{long},t}$ defines the measured longitudinal offset, i. e., in direction of the crop rows, of the corresponding information.

The crucial advantage of this definition is that we can now explicitly control which type of sensor information determines which component of the sensor measurement. The detected Pattern $P_t = P_{\theta,s,o}$, for example, only contains information about the orientation and lateral offset of the crop rows. It should therefore influence the first and second component of the sensor measurement $z_{\theta,t}$ and $z_{\text{lat},t}$ but not the third component $z_{\text{long},t}$. In contrast, the GPS position p_t^{GPS} only contains position information and no information about the orientation of the vehicle. It should therefore only affect the $z_{\text{lat},t}$ and $z_{\text{long},t}$ sensor measurement components. If the end of the field is detected, the Pattern also contains the longitudinal information in form of the maximum extension line $L^{\max}$. In this case, the detected Pattern also influences the third component $z_{\text{long},t}$.

For a better intuition we illustrate all four components of the sensor measurement in the left column of Figure 4.12 in detail. We define the heading measurement $z_{\theta,t}$ as the difference between the normal of the crop rows θ and the heading of the vehicle θ_t. For the second component $z_{\text{lat},t}$ recall that $L_{o_{\text{ref}}}$ is defined in Eq. (4.8) in Section 4.3.1 as the observed line that is closest to the vehicle. Our lateral measurement $z_{\text{lat},t}$ is therefore the by amount smallest signed distance of any detected crop row to the position of the vehicle p_t. The longitudinal sensor measurement according to the end of the field information $z_{\text{long},t}^{\text{EOF}}$ is the signed distance of the maximum extension line $L^{\max}$ to the position of the vehicle p_t. In order to define the longitudinal GPS measurement $z_{\text{long},t}^{\text{GPS}}$, we project the GPS position p_t^{GPS} onto the unit vector $v\left(\theta_m^{\text{GPS}}\right)$ that points in the direction of the mapped line $L_m^{\text{GPS}} \in F_m$ that is closest to the vehicle position p_t. The vector $v(\alpha)$ denotes the directional vector of the angle α on the unit circle: $v(\alpha) := (\cos(\alpha), \sin(\alpha))^T$. This yields the following definition for the sensor measurement z_t given the GPS position p_t^{GPS} and the detected Pattern $P_t = P_{\theta,s,o}$ with the detected end of the field $L^{\max}$:

$$
\begin{aligned}
z_{\theta,t} &:= \theta - \theta_t \\
z_{\text{lat},t} &:= d\left(L_{o_{\text{ref}}}, p_t\right) \\
z^{\text{GPS}}_{\text{long},t} &:= v\left(\theta^{\text{GPS}}_m\right) \cdot p^{\text{GPS}}_t \\
z^{\text{EOF}}_{\text{long},t} &:= d\left(L^{\max}, p_t\right).
\end{aligned}
$$

Whenever different sources of information, i. e., the Pattern and the GPS, provide information for the same component, i. e., the lateral and longitudinal sensor measurements $z_{\text{lat},t}$ and $z_{\text{long},t}$, we choose the information that we expect to be more accurate. For the lateral component we use the information from the detected Pattern, since we expect it to be more accurate (below $0.1\,\text{m}$) than the GPS information (around $3.0\,\text{m}$). For the longitudinal component, we use the information provided by the GPS, unless the end of the field was detected.

The Expected Measurement Based on the definition of each component of the sensor measurement z_t, we also define the measurement prediction function $h\left(\mathbf{x}_t\right)$ that predicts the expected sensor measurement $\bar{z}_t$ from the pose of the vehicle $\mathbf{x}_t$ and the GPS-referenced map of crop rows m. Since we used the observed line $L_{o_{\text{ref}}}$ that is closest to the vehicle for the lateral component of the sensor measurement $z_{\text{lat},t}$, we also need to define the expected lateral sensor measurement relative to the associated mapped line $L_{m_{\text{ref}}} \in m$. We determine this associated mapped line using our *Crop Row data association* from Section 4.3.1. To compute the data association between the detected Pattern P_t and the map m, we first use our definition of the support of a Pattern from Section 4.3.2 to extract a finite set of supported lines $\{L_{o_1}, \ldots, L_{o_n}\}$ from the detected Pattern P_t and thus define the set of observed line features $F_o := \{L_{o_1}, \ldots, L_{o_n}\}$. Using $F_m := m$, our *Crop Row data association* then finds the best geometrically consistent set of matches M^*, where set match $(L_{o_{\text{ref}}}, L_{m_{\text{ref}}}) \in M^*$ defines the our associated mapped line $L_{m_{\text{ref}}}$. Using this mapped reference line $L_{m_{\text{ref}}} := L_{lineThetaMapRef,r_{m_{\text{ref}}}}$, we define the measurement prediction function $\bar{z}_t = h\left(\mathbf{x}_t\right)$ as follows:

$$
\begin{aligned}
h_\theta\left(\mathbf{x}_t\right) &:= \theta_{m_{\text{ref}}} - \theta_t \\
h_{\text{lat}}\left(\mathbf{x}_t\right) &:= d\left(L_{m_{\text{ref}}}, p_t\right) = v\left(\theta_{m_{\text{ref}}}\right) \cdot p_t - r_{m_{\text{ref}}} \\
h^{\text{GPS}}_{\text{long}}\left(\mathbf{x}_t\right) &:= v\left(\theta^{\text{GPS}}_m\right) \cdot p_t \\
h^{\text{EOF}}_{\text{long}}\left(\mathbf{x}_t\right) &:= d\left(L^{\max}_m, p_t\right) = v\left(\theta^{\max}_m\right) \cdot p_t - r^{\max}_m.
\end{aligned}
$$

For better intuition, we visualize these definition in the center column of Figure 4.12. Analogue to the observed measurement z_t, we define the angular and lateral offset using the associated mapped line $L_{m_{\text{ref}}}$ instead of the observed reference line $L_{o_{\text{ref}}}$. Thus, we compute the difference between the angular parameter $\theta_{m_{\text{ref}}}$ of the associated mapped line $L_{m_{\text{ref}}}$ and the heading of the vehicle θ_t, as well as the signed distance of the associated mapped line $L_{m_{\text{ref}}}$ to the position of the vehicle p_t. Similar to the longitudinal observed

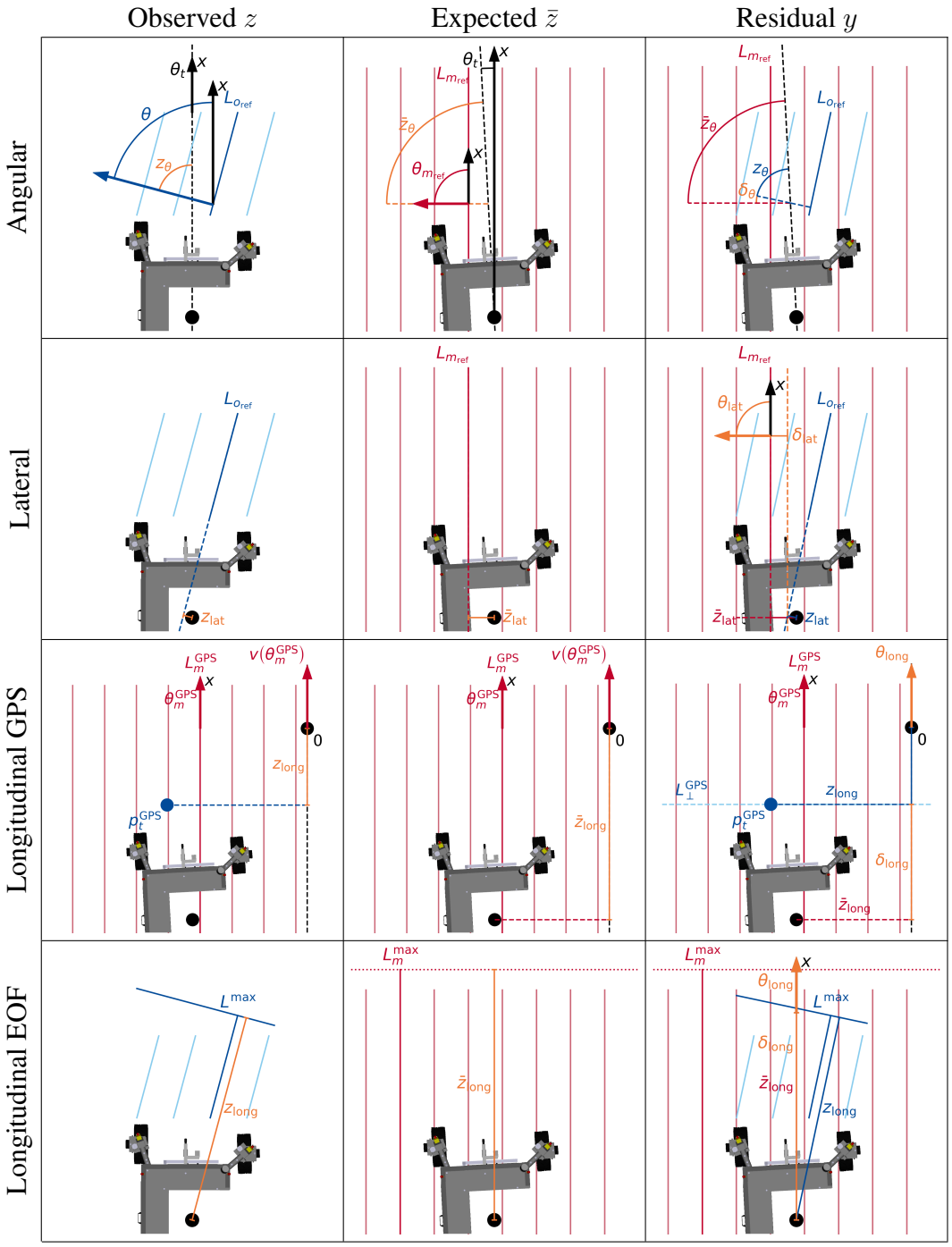

Figure 4.12: This figure shows an overview of the angular, lateral and longitudinal components (top to bottom) of the observed sensor measurement z (left), the expected sensor measurement $\bar{z}$ (mid) and the resulting residual y (right). The illustrated measurements are highlighted in orange. The sensor information contained in the GPS position p_t^{GPS} as well as the detected Pattern $P_t = P_{\theta,s,o}$ are shown in blue colors, while the information from the map is shown in red. The vehicle pose as well as the origin O and the coordinate system are shown in black.

measurement for the GPS information $z^{\text{GPS}}_{\text{long},t}$, we also project the position of the vehicle p_t onto the same unit vector $v\left(\theta^{\text{GPS}}_m\right)$, thereby using the longitudinal projection of the position of the vehicle p_t as expected measurement $\bar{z}^{\text{GPS}}_{\text{long},t}$.

For the expected longitudinal measurement according to the end of the field $\bar{z}^{\text{EOF}}_{\text{long},t}$, we to determine the end of the field $L^{\max}_m$ in the map m. To this end, we use the same procedure as described for determining the end of the field from the support of the Pattern, i. e., we compute the extensions of the field over the mapped lines $L_m \in F_m$, where we assume that the start and end point of each crop row is known, so that the support of these lines can be easily computed. Since the vehicle is inside the field, this yields two extension lines $L^{\min}_m$ and $L^{\max}_m$, where the minimum extension line $L^{\min}_m$ lies behind the vehicle and the maximum extension line $L^{\max}_m$ lies in front of the vehicle. The expected end of the field measurement $\bar{z}^{\text{EOF}}_{\text{long},t}$ is therefore the signed distance of the maximum map extension line $L^{\max}_m$ to the position of the vehicle p_t.

The Residual Using our definitions of the sensor measurement z_t and the expected sensor measurement $\bar{z}_t = h\left(\mathbf{x}_t\right)$, we obtain the following definition for the residual $y_t = z_t - h\left(\mathbf{x}_t\right)$:

$$
\begin{aligned}
y_{\theta,t} &= z_{\theta,t} - h_\theta\left(\mathbf{x}_t\right) = \theta - \theta_t - \left(\theta_{m_{\text{ref}}} - \theta_t\right) \\
&= \theta - \theta_{m_{\text{ref}}} =: \delta_\theta \\
y_{\text{lat},t} &= z_{\text{lat},t} - h_{\text{lat}}\left(\mathbf{x}_t\right) = d\left(L_{o_{\text{ref}}}, p_t\right) - d\left(L_{m_{\text{ref}}}, p_t\right) \\
&= \delta\left(L_{o_{\text{ref}}}, L_{m_{\text{ref}}}\right) =: \delta_{\text{lat}} \\
y^{\text{GPS}}_{\text{long},t} &= z^{\text{GPS}}_{\text{long},t} - h^{\text{GPS}}_{\text{long}}\left(\mathbf{x}_t\right) = v\left(\theta^{\text{GPS}}_m\right) \cdot p^{\text{GPS}}_t - v\left(\theta^{\text{GPS}}_m\right) \cdot p_t \\
&= -d\left(L^{\text{Gps}}_\perp, p_t\right) =: \delta^{\text{GPS}}_{\text{long}} \\
y^{\text{EOF}}_{\text{long},t} &= z^{\text{EOF}}_{\text{long},t} - h^{\text{EOF}}_{\text{long}}\left(\mathbf{x}_t\right) = d\left(L^{\max}, p_t\right) - d\left(L^{\max}_m, p_t\right) \\
&= \delta\left(L^{\max}, L^{\max}_m\right) =: \delta^{\text{EOF}}_{\text{long}}
\end{aligned}
$$

All components of the residual are also shown in the right column of Figure 4.12. Here, we call the component-wise residuals the *angular correction* δ_θ, the *lateral correction* δ_{lat} and the *longitudinal correction* δ_{long}, so that the residual is $y_t = \left(\delta_\theta, \delta_{\text{lat}}, \delta_{\text{long}}\right)^T$. Note that the lateral and longitudinal corrections relative to the detected Pattern δ_{lat} and $\delta^{\text{EOF}}_{\text{long}}$ are computed using our Line to Line Distance measure δ. Also note that the longitudinal correction for the GPS $\delta^{\text{GPS}}_{\text{long}}$ yields the negative point to line distance of the position of the vehicle p_t to the line $L^{\text{Gps}}_\perp$ that is perpendicular to the mapped line L^{GPS}_m and passes through the GPS position p^{GPS}_t.

For convenience, we call the angular parameter $\theta_{m_{\text{ref}}}$ of the mapped line $L_{m_{\text{ref}}}$ associated with the observed reference line $L_{o_{\text{ref}}}$, according to which the expected lateral measurement $\bar{z}_{\text{lat},t}$ is computed, the *lateral correction angle* $\theta_{\text{lat}} := \theta_{m_{\text{ref}}}$, since it also defines the orientation in which the lateral correction δ_{lat} should be applied. Analogue, we define the

longitudinal correction angles $\theta_{\text{long}}^{\text{GPS}} := \theta_m^{\text{GPS}}$ and $\theta_{\text{long}}^{\text{EOF}} := \theta_m^{\text{max}}$ as these are the angles used for computing the expected longitudinal sensor measurement $\bar{z}_{\text{long},t}$ and therefore also the angles under which the longitudinal correction δ_{long} should be applied.

The Gradient Descent (GD) The Gradient Descent (GD) localization tracks is an intuitive technique that directly tracks the pose of the vehicle. The state estimate s_t^{GD} at a time step t is therefore equal to the pose estimate $\hat{\mathbf{x}}_t$ at that time step. In the prediction step, the measured relative motion of the vehicle u_t is directly applied to the previous pose estimate $\hat{\mathbf{x}}_{t-1}$ to predict the pose estimate $\bar{\mathbf{x}}_t$ of the current time step. Similarly, the heading δ_θ, the lateral δ_{lat} and the longitudinal δ_{long} correction measurements are also applied to the predicted pose of the vehicle in a straightforward fashion. Here, we require the reference angles θ_{lat} and θ_{long} to compute the direction in which the position of the vehicle should be corrected. This results in the following mathematical definition of the GD localization method:

$$
\begin{aligned}
s_t^{\text{GD}} &:= \hat{\mathbf{x}}_t \\
\bar{\mathbf{x}}_t &:= \hat{\mathbf{x}}_{t-1} \oplus u_t \\
\hat{\mathbf{x}}_t &:= \bar{\mathbf{x}}_t \oplus \alpha_\theta \cdot \begin{pmatrix} \delta_\theta \\ 0 \\ 0 \end{pmatrix} \oplus \alpha_{\text{lat}} \cdot \begin{pmatrix} 0 \\ \delta_{\text{lat}} \cdot \cos\left(\theta_{\text{lat}}\right) \\ \delta_{\text{lat}} \cdot \sin\left(\theta_{\text{lat}}\right) \end{pmatrix} \oplus \alpha_{\text{long}} \cdot \begin{pmatrix} 0 \\ \delta_{\text{long}} \cdot \cos\left(\theta_{\text{long}}\right) \\ \delta_{\text{long}} \cdot \sin\left(\theta_{\text{long}}\right) \end{pmatrix}
\end{aligned}
$$

The scaling parameters $\alpha_\theta, \alpha_{\text{lat}}, \alpha_{\text{long}} \in \mathbb{R}_{>}0$ determine how strongly we correct the predicted pose estimate $\bar{\mathbf{x}}_t$ in the direction of the corresponding correction measurement.

The Extended Kalman Filter (EKF) The Extended Kalman Filter (EKF) is a probabilistic localization approach that estimates the current pose of the vehicle using a Gaussian distribution $\mathcal{N}_{\Sigma_t,\mu_t}$, where the mean μ_t is the current pose estimate $\hat{\mathbf{x}}_t$ and the covariance matrix Σ_t estimates the uncertainty of the current pose estimate $\hat{\mathbf{x}}_t$. The state estimate s_t^{EKF} is therefore comprised of both Gaussian parameters μ_t and Σ_t at every time step t.

$$
s_t^{\text{EKF}} := (\mu_t, \Sigma_t)
$$

The prediction step of the EKF then predicts the current state $\bar{s}_t$ by shifting the Gaussian distribution according to the measured relative motion u_t and adding uncertainty by widening the covariance matrix according to the expected accuracy of the measured relative motion u_t modeled by the matrix Q_{u_t}.

$$\begin{aligned} \bar{s}_t^{\text{EKF}} &:= \left(\bar{\mu}_t, \bar{\Sigma}_t\right) \\ \bar{\mu}_t &:= \mu_{t-1} \oplus u_t \\ \bar{\Sigma}_t &:= F_{u_t} \cdot \Sigma_{t-1} \cdot F_{u_t}^T + Q_{u_t} \end{aligned}$$

For our implementation, we define the motion model F_{u_t} and the uncertainty introduced by a motion Q_{u_t} for a relative motion $u_t = (\delta\theta_t, \delta x_t, \delta y_t)$ as follows:

$$\begin{aligned} F_{u_t} &:= \begin{pmatrix} 1 & 0 & 0 \\ f^x & 1 & 0 \\ f^x & 0 & 1 \end{pmatrix}, \text{with } \begin{pmatrix} f^x \\ f^y \end{pmatrix} = \begin{pmatrix} -\sin\left(\hat{\theta}_t\right) & -\cos\left(\hat{\theta}_t\right) \\ \cos\left(\hat{\theta}_t\right) & -\sin\left(\hat{\theta}_t\right) \end{pmatrix} \cdot \begin{pmatrix} \delta x_t \\ \delta y_t \end{pmatrix} \\ Q_{u_t} &:= \begin{pmatrix} q^\theta & 0 & 0 \\ 0 & q^x & 0 & 0 \\ 0 & 0 & q^y \end{pmatrix}, \text{with } \begin{pmatrix} q^\theta & q^x & q^y \end{pmatrix} = A \cdot \begin{pmatrix} |\delta\theta_t| \\ |\delta x_t| \\ |\delta y_t| \end{pmatrix} + a. \end{aligned}$$

The matrix $A \in \mathbb{R}^3 \times \mathbb{R}^3$ and the vector $a \in \mathbb{R}^3$ are parameters of the motion model that need to be determined depending on the accuracy of the measured relative motion.

Given a measurement prediction function $h(\mathbf{x}_t)$ and a measurement noise matrix R, the EKF corrects the predicted pose estimate using the residual y_t and the Jacobian matrix H of the measurement prediction function $h(\mathbf{x}_t)$ by computing the *Kalman Gain* K. The correction step of the EKF is then defined by the following equations:

$$\begin{aligned} \mu_t &:= \bar{\mu}_t + K \cdot y_t \\ \Sigma_t &:= (I - K \cdot H) \cdot \bar{\Sigma}_t \\ y_t &:= z_t - h(\mathbf{x}_t) \\ K &:= \bar{\Sigma}_t \cdot H^T \cdot \left[H \cdot \bar{\Sigma}_t \cdot H^T + R\right] \end{aligned}$$

The Jacobian matrix H contains the partial derivatives of each component of our measurement prediction function $h(\mathbf{x}_t)$ and is therefore defined as follows:

$$H := \begin{pmatrix} -1 & 0 & 0 \\ 0 & \cos(\theta_{\text{lat}}) & \sin(\theta_{\text{lat}}) \\ 0 & \cos(\theta_{\text{long}}) & \sin(\theta_{\text{long}}) \end{pmatrix},$$

We model the measurement noise R using component-wise parameters r_θ, r_{lat}, $r_{\text{long}}^{\text{GPS}}$ and $r_{\text{long}}^{\text{EOF}}$:

$$R := \begin{pmatrix} r_\theta & 0 & 0 \\ 0 & r_{\text{lat}} & 0 \\ 0 & 0 & r_{\text{long}} \end{pmatrix},$$

where r_{long} is set to either $r_{\text{long}}^{\text{GPS}}$, if the longitudinal measurement residual was computed using the GPS position measurement p_t^{GPS} and to the $r_{\text{long}}^{\text{EOF}}$ otherwise.

In this section, we presented our novel approach for fusing local information from crop row detections with global GPS information to enable accurate pose estimation in all three parameters of the pose of a ground vehicle. This includes our *Crop Row data association* that finds a geometrically consistent set of matches between the GPS-referenced set of mapped lines and the detected crop rows, as well as an *End of the Field detection* that provides additional longitudinal information to the heading and lateral information usually obtained from the crop row structure of the field. Finally, we explain in-depth how the information obtained from global GPS information and local crop row detections should be fused, by splitting the provided information into a heading, lateral and longitudinal correction measurement for each modality. We also derive the mathematical models for two different localization techniques to demonstrate how to apply these heading, lateral and longitudinal corrections in a localization algorithm. In the next section, we perform an extensive evaluation of both localization algorithms on data collected from a production vegetable field. The results will show, that our approach of fusing GPS information with crop row detections is well suited for autonomous navigation applications, since it accurately tracks not only the heading and lateral position of the vehicle relative to the crops, but also provides an accurate longitudinal pose estimate within a global GPS reference frame.

4.4 Experimental Evaluation

In this section, we evaluate the performance of our fused localization approach on two real-world data sets recorded on a production vegetable field in Eichstetten. For an unbiased evaluation of the performance of our localization algorithms, we need to ensure that the Patterns P_t are reliably and accurately detected on the three new crop types encountered in the two data sets of this evaluation. To confirm our findings from the previous chapter that the detected Patterns P_t are well suited as input to a localization algorithm, we perform the evaluation from the previous chapter on both data sets of this evaluation in our first experimental evaluation. After determining the accuracy and robustness of our crop row detection on the two data sets of this evaluation, we evaluate the performance of our fused localization approach in the following experiments. In our second evaluation, we first compare the performance of our fused localization approach with the performance of a pure crop row following approach as well as a pure GPS-based localization. The goal of this comparison is to confirm that our proposed fused localization can indeed leverage the advantages of both localization approaches while also maintaining the accuracy of each individual approach. In the third experimental evaluation, we then investigate how additionally integrating the information from detecting the end of the field improves the longitudinal position estimate, especially compared to our vanilla fused localization, where the longitudinal position estimate is based on GPS information. Finally, we also show qualitative results of the investigated localization algorithms and discuss relevant facts and findings for applying our proposed localization method in an autonomous nav-

igation system. In the following section, we first give an overview of the recorded data sets as well as the experimental setup required for our experimental evaluation.

4.4.1 Methodology

In this section, we give an overview over the data on which we perform the experimental evaluation in the following sections. This also includes a brief explanation of the required preprocessing steps to obtain the information required as input for the investigated localization methods. We also explain how we obtained the necessary ground truth poses to evaluate the Heading Error, the Lateral Error as well as the Longitudinal Error of each localization algorithm. Finally, we present seven different localization algorithms, where two are representatives for pure crop-row-following-based localization, one represents a pure GPS-based localization, and the remaining four are based on our fused localization approach. During our evaluation we use the first three of these localization algorithms for comparison.

Data Set Overview and Preprocessing For our experimental evaluation, we recorded two data sets on the production vegetable field in Eichstetten presented in Section 2.1.3. The field features three different crop types, Kohlrabi, Chinese Cabbage and Sweetheart Cabbage, that even change mid row. We recorded both data sets with our robotic platform the BoniRob while driving at different speeds of up to $4\,\text{m/s}$. We call the first data set that was recorded in the morning the *Run 1* data set and the second data set that was recorded in the afternoon of the same day the *Run 2* data set. This yields two sets of time synchronized data streams, where we have the following measurements for each time step t of the data set: The wheel odometry information o_t, the IMU measurements i_t, the GPS data g_t as well as images I_t. A detailed overview of the sensors mounted on the BoniRob, that were used to record this data is given in Section 2.1.2. In the following, we explain how we preprocess the recorded raw data into the required input information for our localization algorithms. This includes computing the measured relative motion u_t according to the odometry and IMU readings o_t and i_t, defining the map frame and transforming the GPS data g_t into this map frame, generating a map m of GPS-referenced crop rows as well as detecting crop rows as Pattern P_t from the recorded image data I_t.

Given the sensor readings from odometry o_t and IMU i_t, we directly integrate both into a unified relative motion measurement $u_t = (\delta\theta_t, \delta x_t, \delta y_t)$ for every time step t. We use the relative translational motion measured by the wheel odometry o_t as the translational part $(\delta x_t, \delta y_t)$ of the relative motion u_t. The change in the yaw orientation according to the IMU i_t measurement determines the heading component $\delta\theta_t$ of the relative motion. Since the IMU readings start to notably drift when the vehicle is not moving, we only integrate IMU readings i_t, if the vehicle is moving according to previously integrated wheel odometry measurements $o_{t'}, \ldots, o_{t-1}$. Note that these relative motion measurements are defined in the local frame of the vehicle.

The raw GPS data g_t is in the format of latitude and longitude that describes a global position on the earth in angular units. We convert these global GPS positions into metrical

units relative to a manually defined fixed *GPS reference* position g^{ref} using the UTM conventions. This defines our map frame in metrical units, where the GPS reference g^{ref} is the origin, the x axis points towards east and the y axis towards north. In the following, we convert all GPS measurements g_t into the metrical *GPS-referenced* format p_t^{GPS} of our map frame.

To generate a GPS-referenced map m of crop rows, we need to determine the position of each crop row of the field in our map frame. To this end, we performed a separate *Mapping Run*, where we drove the BoniRob around the edges of the field and stopped at the start and end of each crop row. We then use the average over all GPS position data while the BoniRob was standing at the corresponding start or end of a crop row to determine a GPS-referenced position of the start and end point of each crop row of the field. Using this information, we obtain a GPS-referenced map m of crop rows, i. e., a set of mapped lines $m := \{L_{m_1}, \ldots, L_{m_m}\}$, where each line is defined using the GPS-referenced start and end points of each row. In our experimental evaluation, we use this map m of GPS-referenced crop rows as input to our localization algorithms for both data sets Run 1 and Run 2. For a qualitative overview of the size of the field as well as the distribution of crop rows, we show a visualization of our map m in Figure 4.13. While obtaining the map m as described above is reasonable for our evaluation, it is quite tedious in practice. However, such a map m could also be obtained more efficiently as described in the previous chapter in Section 3.4.4.

For an overview of the fused odometry and IMU measurements u_t as well as the transformed GPS data p_t^{GPS} passed to our localization algorithms, we visualize both relative to our map m in Figure 4.13 for both data sets. Note that the fused odometry and IMU measurements u_t are defined in the local frame of the vehicle. To display these local measurements u_t in comparison with the global GPS position data p_t^{GPS}, we need to transform the local measurements into the map frame. To this end, we manually labeled the *initial pose* $\mathbf{x}_0^{\text{man}}$ of the vehicle, i. e., the heading and position of the vehicle at time step $t = 0$ for each data set. We then iteratively apply the relative motion measurements u_t on this initial pose $\mathbf{x}_0^{\text{man}}$ to obtain the trajectory shown in Figure 4.13.

For detecting crop rows from a given image I_t, we use our crop row detection approach presented in the previous chapter in Section 3.3. For an overview of the input data provided to our crop row detection in this evaluation, we list the most important properties of the vegetable field for Run 1 and Run 2 in Table 4.1. We also show an example image with corresponding Feature Map for each of the three different crop types that need to be detected by our crop row detection in Figure 4.14. While the Kohlrabi and the Sweetheart Cabbage are still at an early growth stage and therefore produce sparser Feature Maps with a mean vegetation density of $2.78\,\%$, the Chinese Cabbage is at a later growth stage causing a denser vegetation distribution in the corresponding Feature Maps of around $5\,\%$ during transition and up to $9\,\%$ in-row (see Table 4.1). Considering our findings from the previous evaluation, we expect that detecting crop rows on the denser Feature Maps of the Chinese Cabbage is more challenging. Therefore the amount of successful Pattern detections will most likely be lower on this crop type than on the other two crops.

In contrast to the crop fields on which we evaluated our crop row detection approach in the previous chapter, this vegetable field has an irregular crop row spacing, which can also

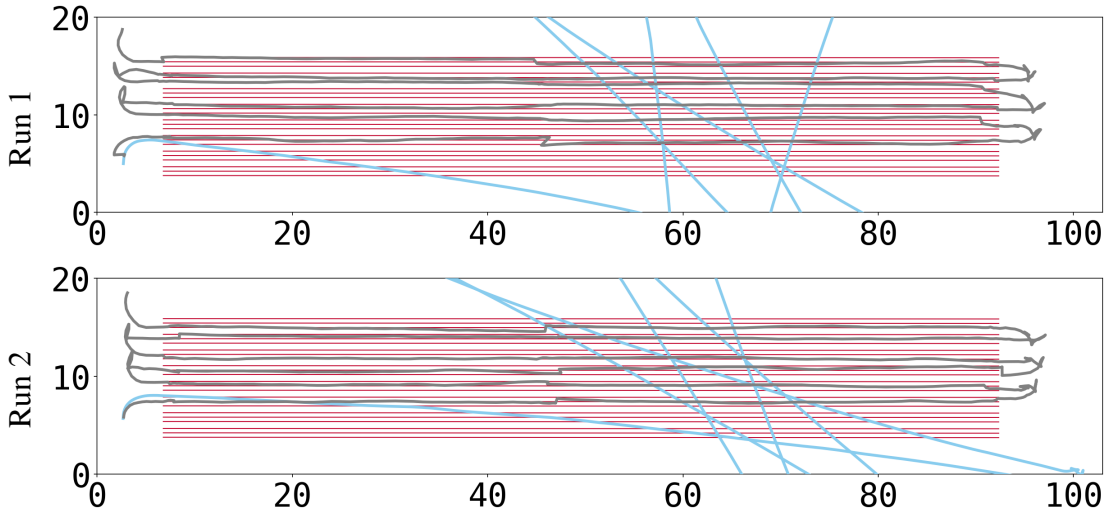

Figure 4.13: This figure shows the GPS positions p_t^{GPS} (gray) as well as the fused trajectory of the wheel odometry o_t and IMU i_t measurements (cyan) relative to our map of GPS-referenced crop rows (red) for both data sets Run 1 (top) and Run 2 (bottom). The x- and y-axis of the map coordinate frame are labeled in meters. The fused trajectory shows a drift in orientation typical for IMU measurements. While the vehicle is standing at the marker positions for measuring the ground truth poses, the GPS measurements also drift, which explains the jumps in the GPS positions at the start, the end and in the middle of each track.

be seen in our GPS-referenced map of the crop rows of the field in Figure 4.13. As mentioned in Section 2.1.3, some fields might have a larger spacing between the crop rows that are adjacent to the wheel tracks to increase clearance between the wheels of agricultural vehicles and the crop. Since our crop row detection is based on the assumption of a constant spacing between crop rows, we restrict the size of the Feature Map in this evaluation so that only the inner crop rows, i. e., the crop rows between the wheel tracks, are visible in the Feature Map (see Figure 4.14). While reducing the size of the Feature Map is reasonable to facilitate the evaluation of our localization algorithms in this chapter, such a restriction is not desirable in practice. We already mentioned a possible solution to this problem in Section 3.5, where our formulation of a crop row Pattern could be extended to also account for the irregular spacing of crop rows that are adjacent to wheel tracks. Since all three crop types are sown in pairs of three crop rows between adjacent wheel tracks, we set our expected minimum and maximum spacing parameters s^- and s^+ accordingly. Except for these adjustments, we do not need to change any other parameters to detect the crop rows on this field. We give a detailed overview of the quality of the detected Patterns P_t in our first experimental evaluation in Section 4.4.2.

Ground Truth Information and Measuring Accuracy For the evaluation of the crop rows, we manually labeled ground truth (GT) Patterns P^{GT} in Feature Maps and compute the Angular Error, the Spacing Error as well as the Lateral Error with respect to these

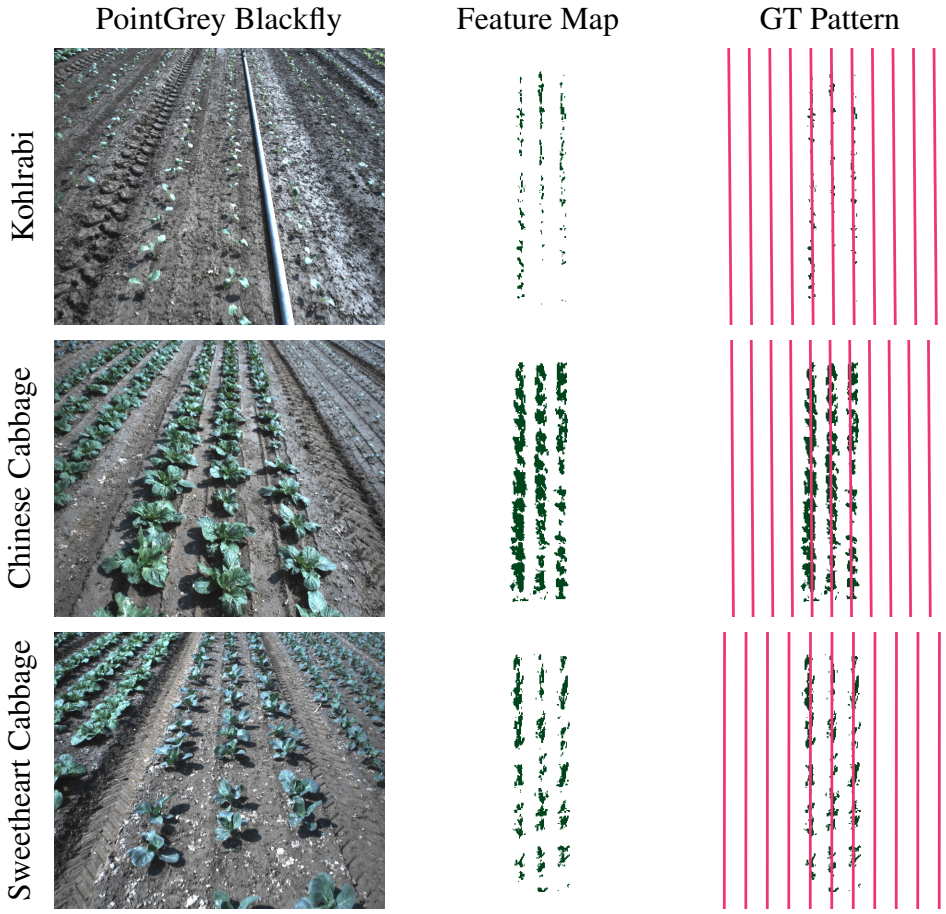

Figure 4.14: This figure shows vision data (left) and corresponding feature maps (mid) with GT Pattern (right) on all three crop types. The image data was recorded with the PointGrey Blackfly mounted in front of the BoniRob at about 1 m above the ground and tilted downwards at about 25°. The feature map shows a top-down view of the extracted vegetation features and is located on the ground plane in front of the vehicle. The manually labeled GT Pattern is shown in magenta.

		In-Row			Transition		
	#R	#FM	Δd [m]	VD [%]	#FM	Δa [°]	VD [%]
Kohlrabi	3	86/83	153	1.54/1.57	91/101	63/65	1.34/1.61
Chinese Cabbage	3	126/130	246	9.14/9.20	151/249	68/85	5.87/4.74
Sweetheart Cabbage	3	55/60	110	2.78/1.97	95/112	19/29	1.04/0.51

Table 4.1: This table shows the properties of the Run 1 / Run 2 data sets for all three crop types split into In-Row and Transition data. On this vegetable field the same number of crop rows (#R) is sown between the wheel tracks for all three crop types. We also display the number of feature maps (#FM) evaluated in our first experimental evaluation and the mean vegetation density (VD) as percentage of cells in a feature map that contain vegetation features. When driving in-row, the heading of the vehicle stays almost constant. Therefore, we only show the translational distance covered (Δd) for the In-Row data. During transition, the vehicle does not move far but usually performs a turning maneuver. Therefore we give the angular distance covered (Δa), i. e., the change in heading of the vehicle, for the Transition data.

GT Patterns P^{GT} as described in Section 3.4.1. The number of manually labeled Feature Maps and therefore evaluated Pattern detections is shown in Table 4.1. An example for the labeled GT Pattern for each crop type is shown on the right in Figure 4.14.

For our evaluation of the accuracy of the localization algorithms, we need to determine the ground truth (GT) pose $\mathbf{x}^{\mathrm{GT}}$ of the vehicle in our map frame. To this end, we manually measured the heading as well as the lateral and longitudinal position of the vehicle at critical points, i. e., the start and end of each row, relative to these crop rows. We also measured the heading and lateral offset of the vehicle in the middle of each row to also have information about the tracking accuracy while driving in-row. To obtain these measurements, we placed *ground truth markers* (M) into the field at the start, the end and the middle of each set of traversed crop rows as shown on the left of Figure 4.15. We aligned these markers with the direction of the crop rows and noted their relative position to the adjacent crop rows. Using this information, we determined the location of the ground truth markers M in our map frame. During data recording, we stopped the BoniRob at these marker locations to measure the pose of the vehicle relative to the ground truth marker using a fixed laser pointer that projects a cross onto the coordinate system printed on the marker as shown on the right in Figure 4.15. Given the pose of the marker in our map frame as well as the relative pose of the BoniRob to each marker during data recording enables us to compute a ground truth pose $\mathbf{x}_i^{\mathrm{GT}}$ for the i-th marker M i. We visualize the resulting ground truth poses $\mathbf{x}^{\mathrm{GT}}$ for each data set in Figure 4.16. We estimate that this method of measuring the ground truth poses $\mathbf{x}^{\mathrm{GT}}$ has an accuracy of $3°$ in the heading of the vehicle and $5\,\mathrm{cm}$ in the lateral and longitudinal position of the vehicle relative to the crop rows. Since these ground truth measurements are defined relative to naturally grown plants, the relative position cannot be measured more precisely. However, a higher measurement accuracy is not required, since a localization error within this magnitude is well suited for navigation applications. Analog to the thresholds from the evaluation of

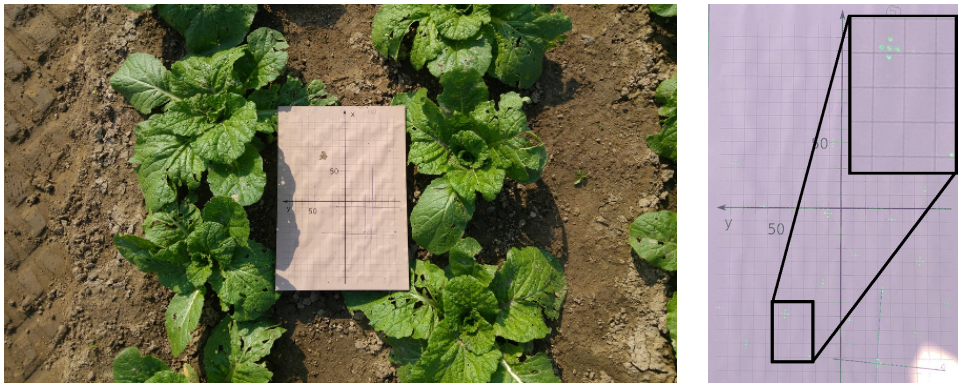

Figure 4.15: This figure shows one of the markers next to the crop rows (left) and a marker with the projected cross of the laser pointer (right). We marked the pose of the vehicle according to the projected position and orientation of the four dots of the laser pointer that form a cross on each marker in the field. We also noted the position of each marker relative to the crop rows. We then used this information to obtain the ground truth pose of the BoniRob during data recording at each marker position.

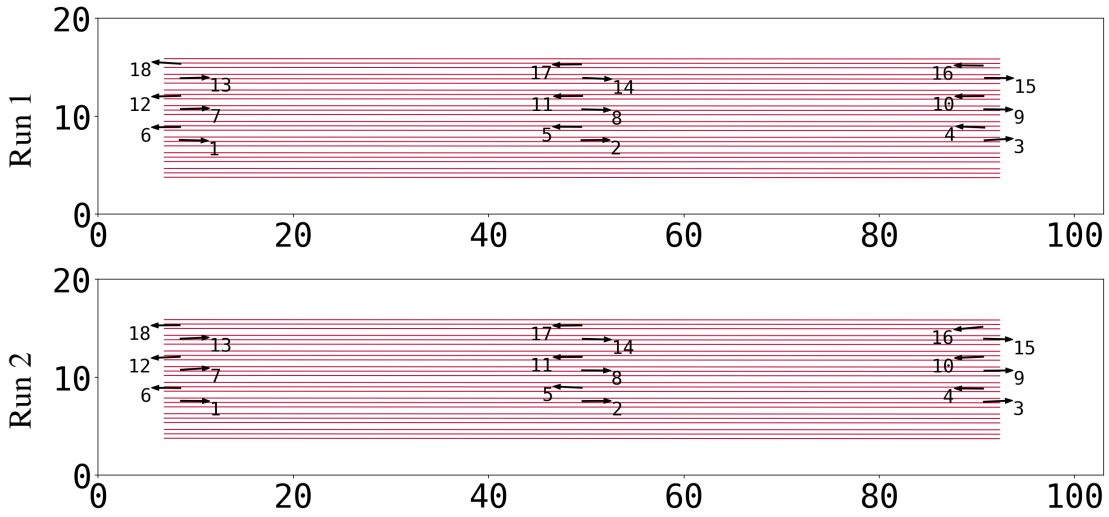

Figure 4.16: This figure shows the manually measured ground truth poses $\mathbf{x}^{GT}$ (black arrows) at each marker position for Run 1 on the top and Run 2 on the bottom. The ground truth poses are defined in the map frame relative to the GPS-referenced crop rows (red).

the previous chapter, we define the localization as *successful*, if the heading error does not exceed $10°$ and the lateral error stays below $10\,\text{cm}$ (see also Section 2.1.1).

Due to the different requirements for the accuracy of the position estimate with a high lateral accuracy of below $10\,\text{cm}$ and much lower requirements for the longitudinal accuracy in the magnitude of meters, we evaluate the localization error of the investigated algorithms by splitting the position error into a Lateral Error and a Longitudinal Error component. Analog to our definitions for the sensor measurements, we define the lateral

and longitudinal direction at each marker position using the angular parameter $\theta_{m_{\text{ref}}}(i)$ of the mapped crop row that is closest to the ground truth position $\mathbf{x}_i^{\text{GT}}$ at the i-th marker M i. We then use the i-th ground truth pose $\mathbf{x}_i^{\text{GT}}$ to compute the Heading Error $\Delta\theta(i)$, the Lateral Error $\Delta \operatorname{lat}(i)$ and the Longitudinal Error $\Delta \operatorname{long}(i)$ of the pose estimate $\mathbf{x}_{t_i}$, where the vehicle was standing at the corresponding location of the i-th marker as follows:

$$
\begin{aligned}
\Delta\theta(i) &:= \left|\theta_i^{\text{GT}} - \theta_{t_i}\right| \\
\Delta \operatorname{lat}(i) &:= \left|\begin{pmatrix} -\sin\left(\theta_{m_{\text{ref}}}(i)\right) \\ \cos\left(\theta_{m_{\text{ref}}}(i)\right) \end{pmatrix} \cdot \left[\begin{pmatrix} x_i^{\text{GT}} \\ y^{\text{GT}}i \end{pmatrix} - \begin{pmatrix} x_{t_i} \\ y_{t_i} \end{pmatrix}\right]\right| \\
\Delta \operatorname{long}(i) &:= \left|\begin{pmatrix} \cos\left(\theta_{m_{\text{ref}}}(i)\right) \\ \sin\left(\theta_{m_{\text{ref}}}(i)\right) \end{pmatrix} \cdot \left[\begin{pmatrix} x_i^{\text{GT}} \\ y^{\text{GT}}i \end{pmatrix} - \begin{pmatrix} x_{t_i} \\ y_{t_i} \end{pmatrix}\right]\right|
\end{aligned}
$$

Note that we always normalize the angular orientation appropriately to obtain $\Delta\theta(i)$ error values between 0° and 360°.

Localization Algorithms The goal of our experimental evaluation is to demonstrate how fusing GPS information with the detected crop row Pattern enables accurate localization of a vehicle not only in the heading and lateral component but also along the crop rows. In literature, localization in agriculture either purely depends on the local information obtained by detecting crop rows, called crop row following, or on purely GPSbased approaches that only leverage the global position information for pose estimation. We therefore compare our approach to both a crop row following localization as well as a GPS Localization. Additionally, to demonstrate that detecting the end of the field further improves the longitudinal position estimate compared to only fusing GPS information with the detected crop row Pattern, we evaluate our localization approach with and without the additional information from end of the field detections. Since we introduced mathematical formulations for two different localization algorithms, the EKFand the GDlocalization method, we also evaluate all localization variants on both localization techniques. This yields the following representatives for each localization variant:

- *GPS Localization*: A purely GPS-based localization algorithm that only uses p_t^{GPS} to correct the predicted pose estimate. We implemented this algorithm using the EKF localization method.
- *Pattern – GD Localization* & *Pattern – EKF Localization*: Two crop row following localization algorithms that only use the detected Pattern P_t to correct the predicted pose estimate. The first uses the GD method, the second is based on the EKF algorithm.
- *Pattern GPS – GD Localization* & *Pattern GPS – EKF Localization*: Both algorithms fuse the GPS information p_t^{GPS} with the detected Patterns P_t to correct the predicted pose estimate. The first uses the GD method, the second is based on the EKF algorithm.

- *Pattern GPS EOF – GD Localization & Pattern GPS EOF – EKF Localization*: Both algorithms fuse the GPS information p_t^{GPS} with the detected Patterns P_t to correct the predicted pose estimate. If the end of the field was detected, the longitudinal information from the detected end of the field is used instead of the GPS information. The first localization is based on the GD method, the second uses the EKF algorithm.

Since the GPS data only contains information of the position of the vehicle, we use the direction of the difference between the previous p_{t-1}^{GPS} and the current p_t^{GPS} position measurement to obtain a correction measurement for the heading of the vehicle. Note that this assumes that the vehicle only performs translational motion in the direction of its heading, i. e., moving sideways or rotating on the spot cannot be modeled by this GPS Localization. However, this is not an issue in this evaluation, since the BoniRob always moved in the direction of its heading in the data sets for this evaluation. In practice, a correction measurement for the heading of the vehicle for purely GPS-based localization can be obtained by mounting two highly accurate GPS receivers at opposite positions on the vehicle. Instead of using the previous measurement the heading is then computed as the difference between the measured position of both receivers.

For all other algorithms, that use the detected Pattern P_t to correct the pose estimate, we also use our definition of the support of a Pattern to compute a measure of *quality* for the detected Pattern as described by Winterhalter et al. [2021]. All Pattern-based localization algorithms use this quality measure to determine whether a Pattern was successfully detected or not according to a given threshold. If the quality is high enough, the Pattern is used to correct the pose estimate and a Correction Step is performed. If the quality of the Pattern is too low, it is rejected and no Correction Step is performed.

Another problem when leaving the traversed field is that the vehicle might perceive misleading Pattern detections from unmapped neighboring fields. To prevent these localization errors, we automatically disable Pattern integration when the vehicle is leaving the traversed field and the sensors do not perceive the traversed field anymore. We enable Pattern integration as soon as the vehicle faces the traversed field again so that the sensors perceive the crops of the traversed field. The localization algorithm then automatically re-localizes the vehicle relative to the detected crop row Pattern.

During the Correction Step, the Pattern-based localization algorithms use our *Crop Row data association* to determine how the set of detected crop rows $F_o = \{L_{o_1}, \ldots, L_{o_n}\}$, which are the supported lines of the Pattern P_t should be associated with the mapped crop rows $m = F_m$. However, if the spacing of the detected Pattern is incorrect, the detected crop rows cannot be aligned with the mapped crop rows, so that the geometrical consistency criterion can not be satisfied. Therefore, the set of valid data association matches $\mathcal{M}^{\text{valid}}$ is empty and no data association M^* can be found. Since this is caused by an incorrect spacing of the detected Pattern, an unsuccessful data association implies that the detected Pattern is incorrect. In these cases, the localization also rejects the detected Pattern and does not perform a Correction Step.

For the Pattern GPS EOF – GD Localization and Pattern GPS EOF – EKF Localization algorithms that leverage the additional information from the end of the field detections,

we already mentioned in Section 4.3.3 that by default the longitudinal measurement according to the GPS data is integrated unless the end of the field was detected.

In the next section, we evaluate the accuracy and robustness of our crop row detection on the three crop types, the Kohlrabi, the Chinese Cabbage and the Sweetheart Cabbage. This evaluation enables an unbiased evaluation of the performance of crop-row-following-based approaches, since the accuracy of the detected Patterns strongly relates to the expected accuracy of the heading and lateral offset estimates of a crop-row-following-based localization method.

4.4.2 Robustness and Accuracy of Crop Row Detections

In our first experimental evaluation, we demonstrate that our crop row detection approach from the previous chapter can reliably detect the crop row Pattern on the three crop types of the data sets Run 1 and Run 2 that we use to evaluate our localization algorithms in this chapter. To this end, we perform the evaluation of the previous chapter as described in Section 3.4.1 on the image data of Run 1 and Run 2. In this evaluation, our focus lies on the Angular Errorand the Lateral Errorof the detected Patterns as well as the overall success rate of the crop row detection algorithms described in the previous chapter. Recall that the comparison algorithms Line Hough and Dual Line Hough do not detect the Pattern on all visible data jointly, but instead rely on detecting individual lines. We therefore call the two comparison algorithms the *line-based* crop row detection methods. This is in contrast to our *Pattern-based* approaches the Pattern Hough and the Pattern RANSAC-based techniques that detect the crop row Pattern on all available information jointly. For our evaluation, we show the success rates of the line-based as well as the Pattern-based crop row detection algorithms in Figure 4.17.

Overall, the success rates do not differ notably between the Run 1 and the Run 2 data sets except for the results of the line-based approaches, the Line Hough and the Dual Line Hough, on the Transition data of Kohlrabi. Recall that Run 1 was recorded in the morning and Run 2 was recorded in the afternoon of the same day. The comparable performance of our Pattern-based algorithms on both data sets suggests that detecting all visible crop rows jointly also improves robustness against different lighting conditions.

Looking at the results for the In-Row data of both data sets, our Pattern-based approaches have a high success rate of at least $94\,\%$ throughout all three crop types. This is in contrast to the considerably lower success rates of the line-based comparison algorithms on the Chinese Cabbage for both data sets with the lowest success rate of only $72\,\%$ of the Line Hough on the Run 2 data set. This confirms our findings from the evaluation of the previous chapter, that our Pattern-based approaches are more robust especially on crops at later growth stages with a denser vegetation feature distribution such as Chinese Cabbage. In summary, all our Pattern-based crop row detection approaches are well suited for in-row navigation with a success rate of at least $94\,\%$ on the In-Row data of both data sets.

For a more detailed discussion of the overall lower success rate of all investigated algorithms on the Chinese Cabbage while traversing the field in-row compared to the other two crop types Kohlrabi and Sweetheart Cabbage, we show the Lateral Error of

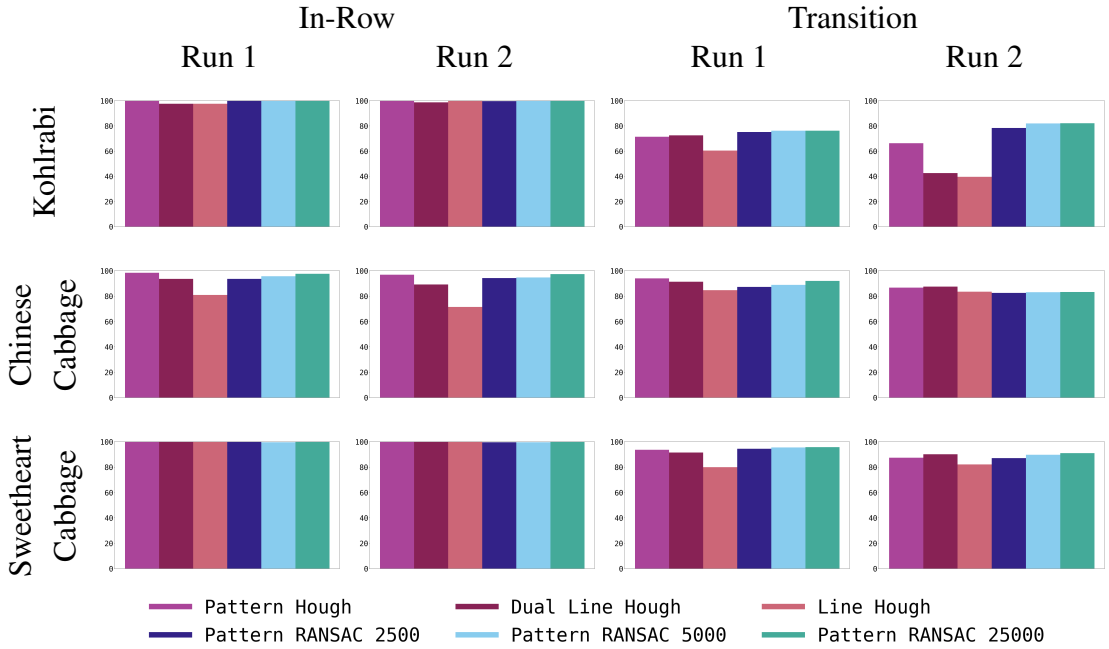

Figure 4.17: This figure shows the success rates for all algorithms on all data sets. The success rate is shown as percentage of the number of successful Pattern detections. Each bar shows the result of a different crop row detection algorithm on the In-Row data sets on the top and on the Transition data sets on the bottom. In each plot, from left to right the algorithms are: Pattern Hough, Dual Line Hough, Line Hough, Pattern RANSAC 2500, Pattern RANSAC 5000 and Pattern RANSAC 25000.

the In-Row data for all crop types in Figure 4.19. On the Kohlrabi and the Sweetheart Cabbage the Lateral Error stays below $5\,\text{cm}$ for most of the detected Patterns. This is in contrast to the Chinese Cabbage, where the Lateral Error exceeds $5\,\text{cm}$ for about $25\,\%$ of the evaluated data for our Pattern-based approaches and even more for the line-based algorithms with up to $50\,\%$ for the Line Hough. As already mentioned in Section 4.4.1, the main difference between the Chinese Cabbage and the other two crop types is that the Chinese Cabbage crops are larger than the other two crops, which causes a denser vegetation feature distribution in the Feature Map (see Table 4.1). On such clusters of vegetation features, the Pattern can be shifted in the lateral direction without decreasing the support of the Pattern on the Feature Map (see also Figure 4.14). Therefore, the best Pattern is not clearly defined which causes an overall larger Lateral Error in comparison to the sparser Feature Maps of the other two crop types.

The more challenging Chinese Cabbage data also reveals the advantage of estimating the spacing parameter instead of relying on a predefined, fixed spacing. Recall that the Line Hough uses such a predefined, fixed spacing value to obtain a crop row Pattern by extending an individual detected line with this spacing parameter into a set of lines. Therefore, the Line Hough can not adapt to the varying crop row spacings present on this vegetable field, which in turn causes an increased Lateral Error for the Patterns detected by the Line Hough compared to the other algorithms (see Figure 4.19). In contrast, the

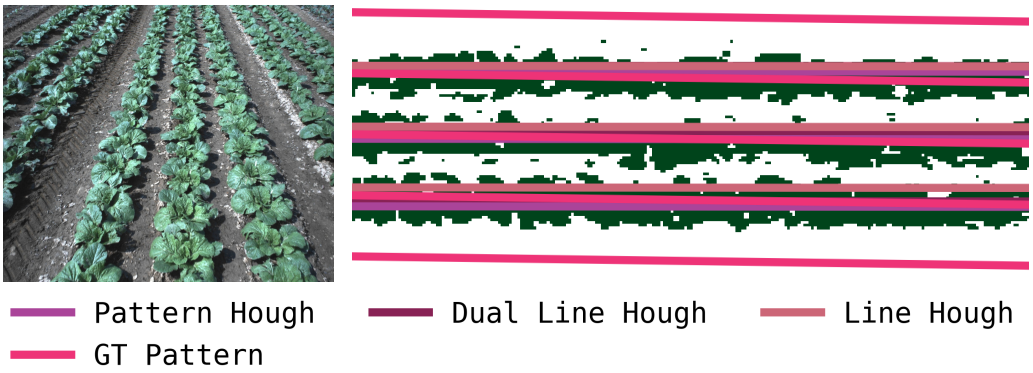

Figure 4.18: This figure shows Pattern detection results for the Line Hough (light green), the Dual Line Hough (green) and the Pattern Hough (magenta) on the Chinese Cabbage In-Row data. The Line Hough cannot adjust the spacing of the detected Pattern, therefore the Pattern lines are offset to the crop rows. In contrast, the Pattern Hough and even the other line-based approach, the Dual Line Hough, can estimate the correct spacing parameter and correctly detect the Pattern.

other line-based approach, the Dual Line Hough, is more flexible than the Line Hough as it uses a second detected line to estimate the spacing of the Pattern. Therefore, the Lateral Errors of the Patterns detected by the Dual Line Hough are much closer (Run 2) or even comparable (Run 1) to our Pattern-based approaches that all estimate the spacing between crop rows jointly. We also show an example for such a situation in the In-Rowdata on the Chinese Cabbagein Figure 4.18. Here, the Pattern detected by the Line Hough is offset to the correct Pattern due to an incorrect fixed spacing parameter. In contrast, our Pattern Hough and even the line-based Dual Line Hough estimate the correct spacing and therefore find the correct Pattern.

Considering the results for the Transition data of both data sets, we again see our reasoning from the previous chapter confirmed that – in general – the data perceived during transition maneuvers is more challenging, since the lower bound of the success rate drops notably for all algorithms compared to the lower bound on the In-Row data. These results are expected since less crop row structure is perceived during transition and additional vegetation that grows on the headlands adds noise to the Feature Maps. For Chinese Cabbage and Sweetheart Cabbage the success rates are reasonable for all investigated algorithms with at least $80\,\%$ for the line-based approaches and a bit higher with at least $83\,\%$ for our Pattern-based approaches. While $80\,\%$ of successful Pattern detections should still be feasible in a localization application, this result clearly shows that a crop-row-based localization needs to be robust against unsuccessful Pattern detections. For our localization approach, we employ different measures including the quality of the Pattern as well as an empty set of valid data association matches, to determine whether a Pattern was detected successfully and should be used to correct the pose estimate or whether the detection was unsuccessful and therefore the Pattern should be rejected.

While the success rates on the Chinese Cabbage and the Sweetheart Cabbage are suf-

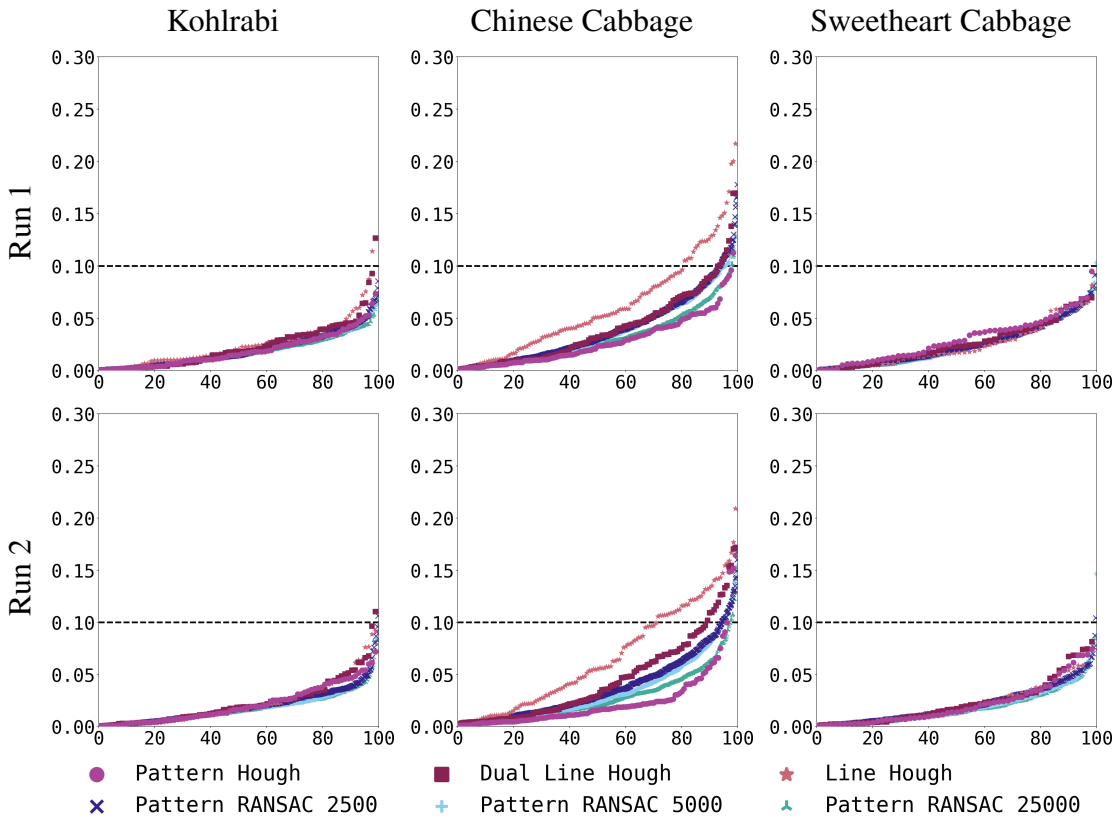

Figure 4.19: This figure shows the **Lateral Error[m]** on the **In-Row** data sets for all three crop types over both Runs. The individual error values for each measurement are shown in ascending order. The horizontal axis shows the position of the measurement in the sorted list in percent ([%]). The error value is plotted along the vertical axis. Since we repeated the RANSAC algorithms five times, they have five times more measurements than the Hough-based algorithms.

ficient, they drop considerably on the Kohlrabiwith as low as $40\,\%$ for the line-based methods and $66\,\%$ for our Pattern-based approach. This result is counterintuitive, since we would expect the larger – and therefore more challenging – Chinese Cabbage to also have the lowest success rate on the Transition data to be consistent with our findings on the In-Row data. An even more surprising result is that the success rates of the Chinese Cabbage on the Transition data are comparable to the results on the In-Row or even increase for the line-based algorithms. This is unexpected, since the Transition data set is supposed to be more challenging than the In-Row. For a better understanding of these results, we show the individual Angular Error and Lateral Error on the Transition data on Kohlrabi shown in Figure 4.20. The most prominent observation is that the Angular Error of all algorithms exceeds the success threshold and completely diverges on the Run 1 data set for around $20\,\%$ of the detected Patterns. In correspondence this also causes the Lateral Error to diverge and exceed the success threshold after around $80\,\%$ of successfully detected Patterns for all algorithms, except the Line Hough. The Lateral Error of the Line

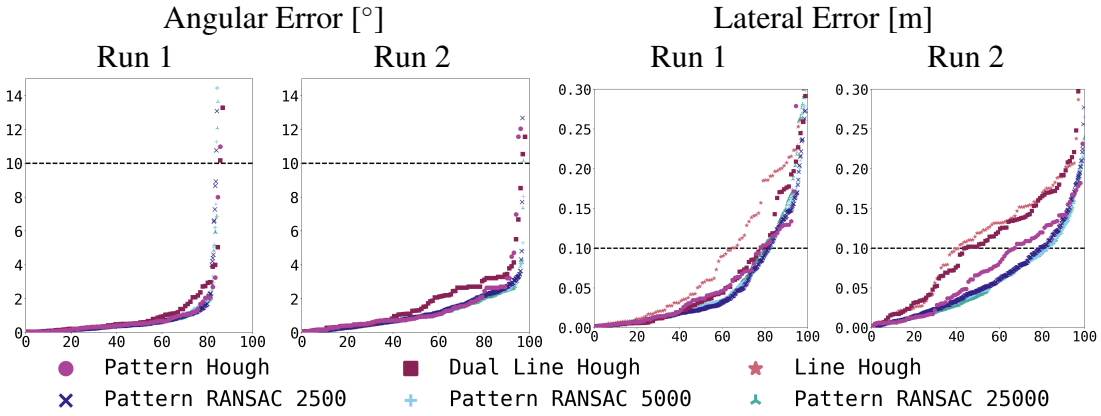

Figure 4.20: This figure shows the **Angular Error** (left) and the **Lateral Error** (right) for **Transition** data on the **Kohlrabi** crops. The individual error values for each measurement are shown in ascending order. The horizontal axis shows the position of the measurement in the sorted list in percent ([%]). The error value is plotted along the vertical axis. Since we repeated the RANSAC algorithms five times, they have five times more measurements than the Hough-based algorithms.

Hough diverges much earlier and therefore exceeds the success threshold after only $60\,\%$ of successfully detected Patterns. We show an example for such an unsuccessful Pattern detection on the Kohlrabi during transition on the top row of Figure 4.21 as well as an example for successful Pattern detection on the Chinese Cabbage during transition on the bottom.

The vegetation distribution of the Feature Maps in the example as well as the corresponding images explain both counterintuitive results for the Kohlrabi and the Chinese Cabbage on the Transition data: During the transition maneuver between crop rows of the Kohlrabi, the camera also perceives the neighboring crop rows that contain the larger Chinese Cabbage (see top left of Figure 4.21). This results in a dense vegetation feature distribution in the corresponding Feature Map with barely visible crop row structure (see top right of Figure 4.21). Even for a human, without prior knowledge of the true heading of the vehicle relative to the crop rows, it is hard to determine the correct orientation of the Pattern. Thus, in the example for the Kohlrabi, none of the investigated algorithms are able to detected the Pattern successfully. This observation can be transferred to the Chinese Cabbage as well, where during transition parts of the neighboring crop rows containing the Sweetheart Cabbage are also visible as shown in the example on the bottom row of Figure 4.21. Since the Sweetheart Cabbage results in vegetation Feature Maps with a clearly visible row structure, it makes sense that the success rate might increase on the Chinese Cabbage Transition data as it also contains Sweetheart Cabbage.

Since we traversed the crop rows in the same order for both data sets. The same explanation also holds for Run 2. While the Angular Error is accurate for most of the detected Patterns on the Kohlrabi Transition data, the Lateral Error still diverges and exceeds the success threshold for a large amount of detected Patterns which causes the low success rates seen in Figure 4.17. Analogue to Run 1, these diverging Lateral Error values are

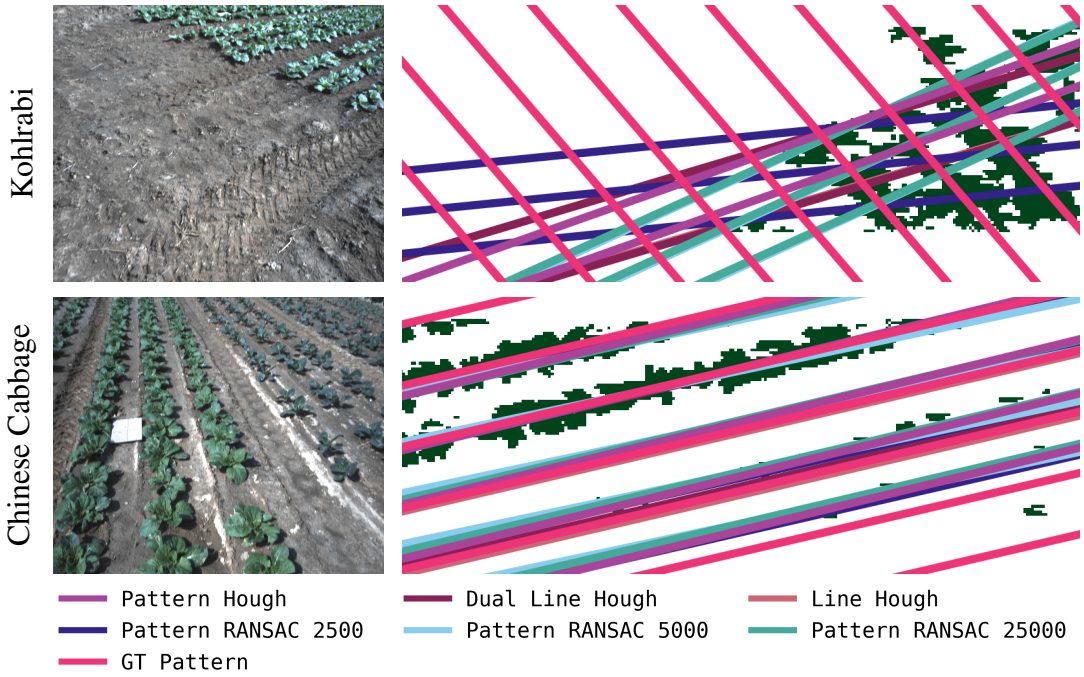

Figure 4.21: This figure shows Pattern detection results for all algorithms on the Transitiondata for the Kohlrabi (top) and the Chinese Cabbage (bottom). During transition between the Kohlrabi crop rows, the Chinese Cabbage are perceived in the sensor data causing dense Feature Maps with barely visible crop row structure (top, right). Therefore, the GT Pattern is not detected by any of the algorithms. During transition between the Chinese Cabbage crop rows, the Sweetheart Cabbage are perceived resulting in more clearly visible crop row structure (bottom left). Therefore, all algorithms can find the correct Pattern (bottom, right).

caused by the fact that the neighboring Chinese Cabbage is visible in the sensor data during turning. This results in a Lateral Error distribution similar to the Lateral Error distribution observed for the Chinese Cabbage on the In-Row data set.

Overall, the results of this evaluation are consistent with our findings from the crop row detection evaluation from our previous chapter. Our Pattern-based crop row detection is more robust than the line-based methods, especially on denser vegetation feature distributions as observed on larger crops in later growth stages such as the Chinese Cabbage, as well as during more challenging transition maneuvers. Therefore, we use our Pattern Hough approach to detect the Patterns P_t passed to the localization algorithms to determine the trajectory of the BoniRob on the Run 1 and Run 2 data sets. Our following evaluation of the localization algorithms will show that even with a success rate of only $66\,\%$ on the Kohlrabi Transition data our localization approaches can still accurately localize the vehicle on the vegetable field.

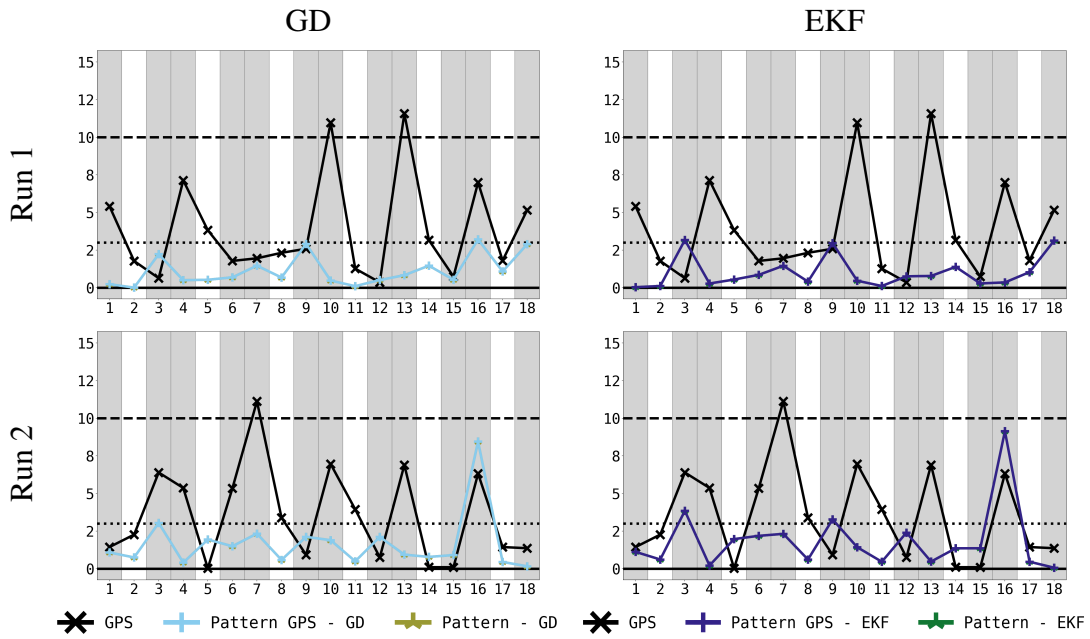

Figure 4.22: This figure shows the **Angular Error** for the GD (left) and the EKF (right) algorithms for Run 1 (top) and Run 2 (bottom). The error values are plotted along the y-axis in degrees. The x-axis shows the error value at each marker M. We highlight the markers located at the start and end of each row in gray. Note that the Angular Error for Pattern – GD and Pattern – EKF are identical to Pattern GPS – GD and Pattern GPS – EKF respectively. Thus they are occluded by the Pattern GPS variants.

4.4.3 Performance of Fused Localization

One important challenge when fusing information from different sensor modalities is that the information from different modalities is not necessarily uncorrelated. For example, both the GPS data as well as the detected Pattern can provide information about the position of the vehicle. In the first evaluation of our localization approach, we therefore compare the performance of our fused localization variant, the Pattern GPS, with the performance of the purely GPS-based localization, the GPS Localization, as well as the purely crop-row-following-based localization, the Pattern localization. Our goal is to confirm that our fused localization approach, the Pattern GPS localization, leverages the advantages of both sensor modalities and therefore maintains at least the same accuracy as the vanilla comparison algorithms. We therefore compare the Heading Error, the Lateral Error as well as the Longitudinal Errorof the GPS Localization, the Pattern as well as the Pattern GPS localization algorithms. We show the individual errors of each of the investigated algorithms at each marker position in Figure 4.22 for the Angular Error, in Figure 4.23 for the Lateral Error and in Figure 4.24 for the Longitudinal Error.

The most prominent observation is that the results for the Pattern localization algorithms (green) are not visible in the Heading Error and Lateral Errorfor both the GD as well as the EKF-based localization algorithms, since they perfectly coincide with the cor-

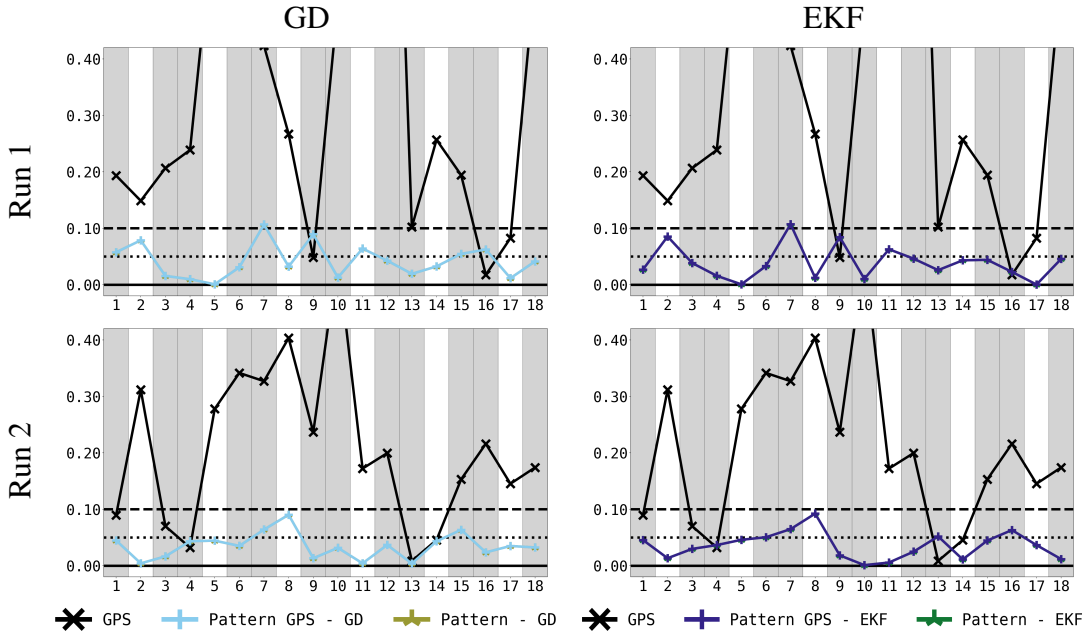

Figure 4.23: This figure shows the **Lateral Error** for the GD (left) and the EKF (right) algorithms for Run 1 (top) and Run 2 (bottom). The error values are plotted along the y-axis in meters. The x-axis shows the error value at each marker M. We highlight the markers located at the start and end of each row in gray. Note that the Angular Error for Pattern – GD and Pattern – EKF are identical to Pattern GPS – GD and Pattern GPS – EKF respectively. Thus they are occluded by the Pattern GPS variants.

responding error values of the fused Pattern GPS localization variants (see Figure 4.22 and Figure 4.23). This confirms that the fused Pattern GPS can maintain the accuracy of pure crop row following localization regarding the heading and lateral offset of the vehicle relative to the crop rows. These results confirm our design of the fused Pattern GPS localization in Section 4.3.3, where we use the heading and lateral offset information contained in the detected Pattern to correct the heading and lateral offset of the vehicle relative to the crop rows.

Another choice was to not use the heading and lateral offset information that can be obtained from the GPS information in our fused Pattern GPS localization. Looking at the Heading Error and Lateral Errorof the GPS Localization, we see our choice confirmed: While the GPS Localization still shows decent Heading Error results that only slightly exceed the success threshold of $10°$ at few marker positions it is outperformed by the crop-row-based localization approaches that mostly have an Heading Error below or around our measurement accuracy of $3°$, except for for Run 2 at marker M 16. The results for the Lateral Error are even more extreme: While the Lateral Error of the GPS Localization frequently exceeds the success threshold of 0.1 m with an Lateral Errorof up to 1.34 m, the Pattern localization is successfully localized at all measurement positions, except for Run 1 at marker M 7 where it slightly exceeds the threshold with an Lateral Errorof 0.11 m. Therefore, using the heading and lateral offset corrections obtained from the detected

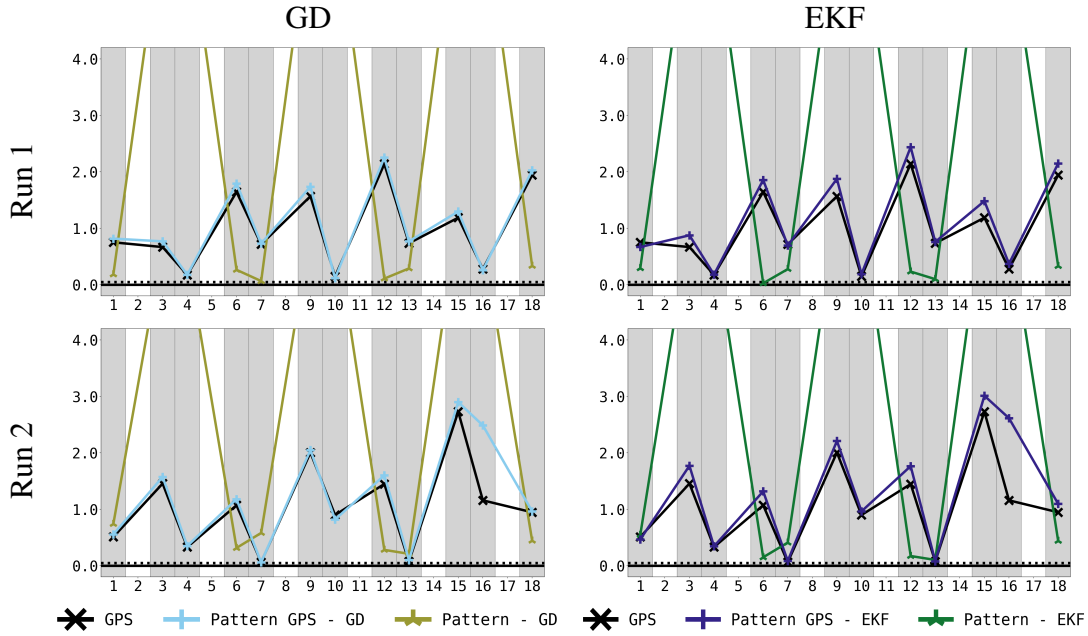

Figure 4.24: This figure shows the **Longitudinal Error** for the GD (left) and the EKF (right) algorithms for Run 1 (top) and Run 2 (bottom). The error values are plotted along the y-axis in meters. The x-axis shows the error value at each marker M. We highlight the markers located at the start and end of each row in gray. We only measured the Longitudinal Error at the start and end of each row. Therefore, there is no measurement for the markers located in the middle of the field (white background).

Pattern is always preferable over the heading and lateral offset information obtained from GPS sensor.

As expected, the results are inverted for the Longitudinal Error(see Figure 4.24). Here, the GPS Localization shows better localization results as it stays below a maximum of $3.0\,\mathrm{m}$ while the error values for the purely crop-row-based Pattern localization even exceed $4.0\,\mathrm{m}$. This again confirms our choice of using GPS information to correct the longitudinal position of the vehicle in our fused localization approach. The results also confirm that we succeeded in leveraging the longitudinal information contained in the GPS data, since the Pattern GPS localization algorithms show Longitudinal Error values close to the GPS Localization, except for Run 2 at marker M 16. However, the larger Longitudinal Errorof our Pattern GPS compared to the purely GPS-based localization is explained by the also larger Heading Error of the Pattern GPS at this marker, since the heading estimate of the vehicle is correlated to the longitudinal position estimate.

Our evaluation of the individual Heading Error, the Lateral Error and the Longitudinal Error shows, that our fused localization approach indeed estimates the full pose of the vehicle, i. e., it corrects the pose of the vehicle in all three dimensions. This is facilitated by integrating the heading and lateral offset information obtained from the detected Pattern to correct the heading and sideways tracking of the vehicle relative to the crop rows, as well as correcting the position of the vehicle along the crop rows using the GPS data. We also

confirm, that our fused localization approach maintains the accuracy in the heading and lateral component of a crop row following approach as well as the longitudinal position accuracy of a GPS Localization. By fusing GPS information with crop row detections, our localization approach shows a localization accuracy that is well suited to guide an agricultural vehicle along the crop rows of a field with high heading and sufficient lateral accuracy. In our next evaluation, we investigate in how far the longitudinal pose estimate of our fused localization approach can be improved by also integrating information from detecting the end of the field, thereby enabling localization beyond crop row following.

4.4.4 Improving the Longitudinal Position Estimate

In the previous evaluation, we confirmed that our fused localization approach can accurately track crop rows like a crop-row-following-based localization, while at the same time also maintaining an accurate longitudinal position estimate according to GPS information. However, to enable autonomous turning at the headlands, the accuracy of 3.0 m of a standard GPS sensor is not sufficient. If the vehicle overestimates the distance traveled along the crop rows, it might initiate the turning maneuver while still inside the crop field, potentially harming the crops. On the other hand, some headlands are quite narrow, so that underestimating the traveled distance could cause the vehicle to accidentally leave the headlands which in turn could not only result in damage to the vehicle but also to surrounding objects or even animals or people. Therefore, an accurate longitudinal pose estimate of the vehicle as it approaches the end of the field is crucial for autonomous traversal of an entire field, including turning maneuvers at the headlands. In this evaluation, we therefore analyze how much detecting the end of the field improves the longitudinal pose estimate of our fused localization approach.

For our analysis of the longitudinal position accuracy attainable by detecting the end of the field, we show the Longitudinal Error of all investigated algorithms in Figure 4.25. As we have already discussed in the previous evaluation, the Pattern localization has the worst longitudinal pose estimate, that periodically exceeds 4.0 m at marker positions M 3 and M 4, M 9 and M 10 as well as M 15 and M 16. This directly correlates to the vehicle reaching the far end of the field, i. e., the opposite side of where we started the data recording, which can also be seen in our overview of the marker positions in Figure 4.16. This is explained by the fact that the detected Pattern contains no information of the longitudinal position of the vehicle relative to the crop rows. Therefore, a pure crop row following approach can not correct the longitudinal position estimate, so that the longitudinal position estimate of the vehicle is based purely on the relative motion measurements obtained from the wheel odometry of the vehicle. The fact that the longitudinal position estimate is constantly overshooting at the far end of the field and returning to the correct longitudinal position estimate at the near end of the field is therefore simply explained by the wheel odometry overestimating the traversed distance.

In contrast to a pure row following localization, our Pattern GPS localization can leverage the GPS position information to correct the longitudinal position estimate similar to a GPS Localization. Therefore, the Longitudinal Error of our Pattern GPS localization algorithms is within the GPS measurement accuracy of around 3.0 m.

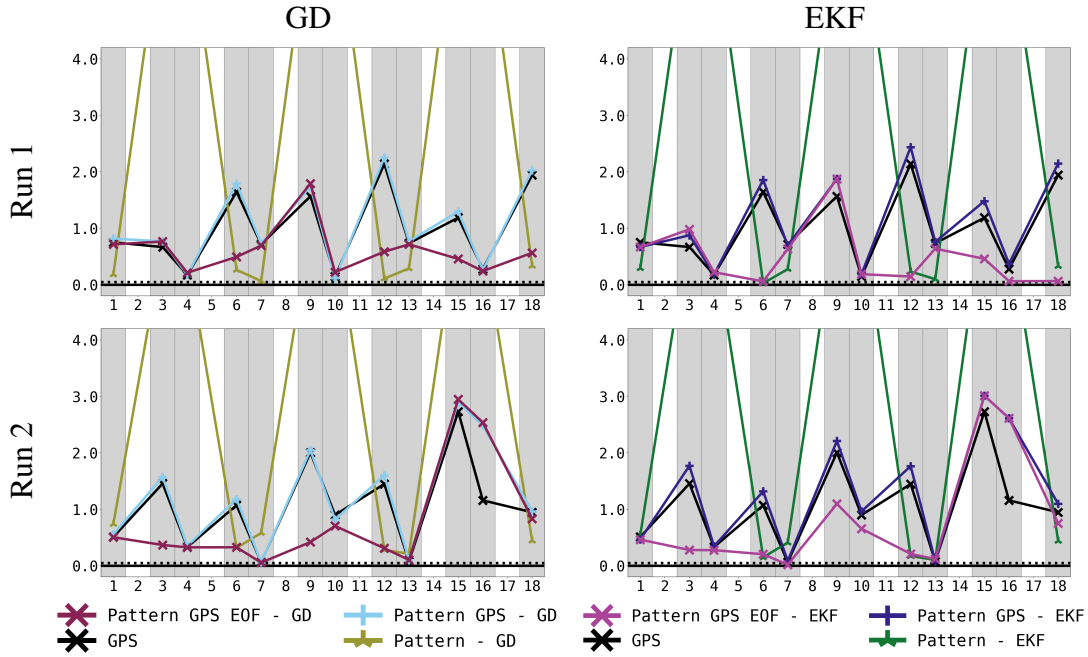

Figure 4.25: This figure shows the **Longitudinal Error** for the GD (left) and the EKF (right) algorithms for Run 1 (top) and Run 2 (bottom). The error values are plotted along the y-axis in meters. The x-axis shows the error value at each marker M. We highlight the markers located at the start and end of each row in gray. We only measured the Longitudinal Error at the start and end of each row. Therefore, there is no measurement for the markers located in the middle of the field (white background).

Looking at the results of our Pattern GPS EOF localization variant that received additional longitudinal information if the end of the field was detected, we can see an increased accuracy at the corresponding measurement locations. At first glance, it only seems like the end of the field was only detected on few occasions. However, recall that the end of the field can only be detected, if the vehicle was traversing the crop field and is approaching the end of the field. This means that the end of the field can only be detected at every third marker position, i. e., at markers M 3,M 6,M 9,M 12,M 15 and M 18. At measurement locations, where the longitudinal position estimate of the Pattern GPS EOF localization did not improve compared to the Pattern GPS localization, i. e., at markers M 3 and M 9 for Run 1 and markers M 15 and M 18 for Run 2, either the end of the field was not detected or the whole Pattern was rejected because of a low Pattern quality value. However, whenever the end of the field was detected, the longitudinal position estimate is considerably improved even down to 0.21 m with a maximum error of 1.1 m. This confirms that additionally including end of the field detections considerably increases the accuracy of the longitudinal position estimate to 1.1 m and below. Such an accurate longitudinal position estimate is crucial to enable autonomous turning on the headlands.

After this extensive quantitative evaluation, we also show some qualitative localization results in our last evaluation, where we not only give a qualitative overview over the

localization accuracy but also investigate interesting localization results at certain marker positions in detail. Finally, we also discuss facts and findings that are relevant for applying our localization approach in an autonomous navigation system.

4.4.5 Autonomous Navigation on Entire Crop Fields

In our last evaluation, we show qualitative results of some localization algorithms. This includes an overview over the position estimates for a better intuition of the performance of our fused localization approach compared to the pure crop row following and GPS-based localization approaches. We also discuss interesting localization results at certain maker positions by providing detailed illustrations of the available data overlaid with the resulting pose estimate at these marker locations. Finally, we also discuss facts and our findings from multiple autonomous navigation runs, where we employed our fused localization approach to provide an accurate pose of the vehicle to our navigation system.

For our qualitative evaluation, we first show an overview of the performance of different localization algorithms by overlaying the trajectory of the vehicle according to the Pattern GPS EOF – EKF Localization, the Pattern – GD Localization and the GPS Localization onto our map of the crop field in Figure 4.26. The results of our quantitative evaluation are visible in the different trajectories of each localization algorithm: For the crop row following localization, the Pattern – GD Localization, we can see that it nicely tracks the position of the vehicle along the crop rows, since there are no jumps in the trajectory as can be observed for the trajectory of the GPS Localization. On the other hand, the crop row following localization clearly overshoots at the far end of the field as discussed in the previous evaluation, since it cannot correct the pose estimate in the direction of the crop rows. While the GPS Localization fails to track the position of the vehicle relative to the crop rows, visible in the jumps of the trajectory, it can correct the overshooting caused by the wheel odometry. This overview also illustrates how our approach, the Pattern GPS EOF – EKF Localization, can leverage the advantages of the other two localization techniques: Our fused localization does not overshoot at the far end of the field while at the same time it also tracks the position of the vehicle along the crop rows without any jumps. Additionally, it also improves the longitudinal pose estimate compared to the GPS Localizationsince the trajectory of the Pattern GPS EOF – EKF Localization stays closer to where we actually turned on the headlands relative to the crop rows.

This overview, as well as the evaluation of the Lateral Errorin Figure 4.23 also contain an implicit result regarding our *Crop Row data association*: Recall that localizing the vehicle relative to the mapped crop rows is only possible due to our *Crop Row data association* that determines a geometrically consistent data association between the observed crop rows and the mapped crop rows. If this data association would have found an incorrect solution while the vehicle is tracking the crop rows, there would have been a notable jump in the trajectory as well as a high Lateral Error measurement, similar to the results observed for the GPS Localization. However, since there are no such jumps or high Lateral Error, we can conclude that our crop row detection always finds the correct data association on the evaluated data sets. This means that only considering geometrically consistent sets of line matches yields a robust data association technique. This

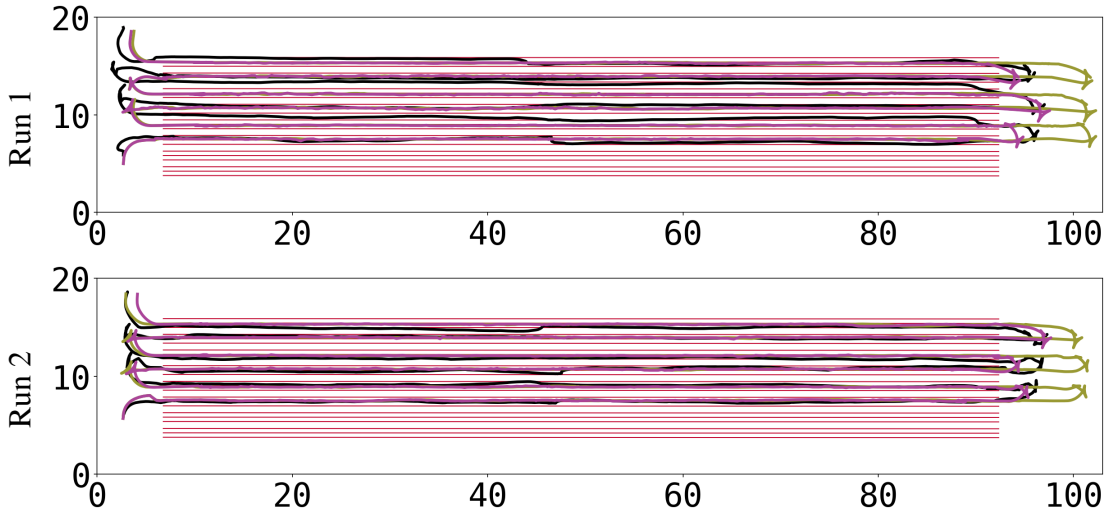

Figure 4.26: This figure shows the localization results of the GPS Localization (black), the Pattern – GD Localization (olive) and the Pattern GPS EOF – EKF Localization (purple). The pose estimates of each localization algorithm are visualized as trajectory (solid lines) by connecting the position of consecutive estimates. For reference, the trajectories are overlaid on the mapped crop rows (red lines).

robust *Crop Row data association* in turn is crucial for our fused localization since it facilitates fusing the local Pattern detections with the global information obtained from a GPS sensor.

Additionally to the qualitative overview, we also provide more detailed illustrations of the localization results at marker positions with interesting results. Recall our first evaluation, where the Heading Error of our fused localization as well as the crop-row-following-based approach was higher than the Heading Error of the GPS Localization in Run 2 at marker M 16 (see Figure 4.22). Another interesting result from our first evaluation is the Lateral Error in Run 1 at marker M 7. For a more detailed discussion of these two results, we show the estimated pose of the GPS Localization in comparison to the estimated pose of our Pattern GPS EOF – EKF Localization in Figure 4.27.

While the Heading Error in Run 2 at M 16 is larger than the Heading Error of the GPS Localizationat this measurement location, it is still within our success threshold. From the visualization of the pose of the vehicle relative to the crop rows on the right of Figure 4.27, we see that the detected crop rows (blue) do not overlap with the mapped crop rows (red) when projecting them using the pose estimated by our Pattern GPS EOF – EKF Localization. This means that the pose of the vehicle was not corrected according to the detected Pattern, due to an insufficient Pattern quality. Thus, the heading of the vehicle after turning was not yet corrected when we took the measurement at marker M 16. However, just after measuring the pose at marker M 16, the quality of the Pattern increases and the heading is corrected, resulting in a successful localization across the entire field for Run 2 for our Pattern GPS EOF – EKF Localization. This is in contrast to the pose of the GPS Localization at this location shown on the top right of Figure 4.27.

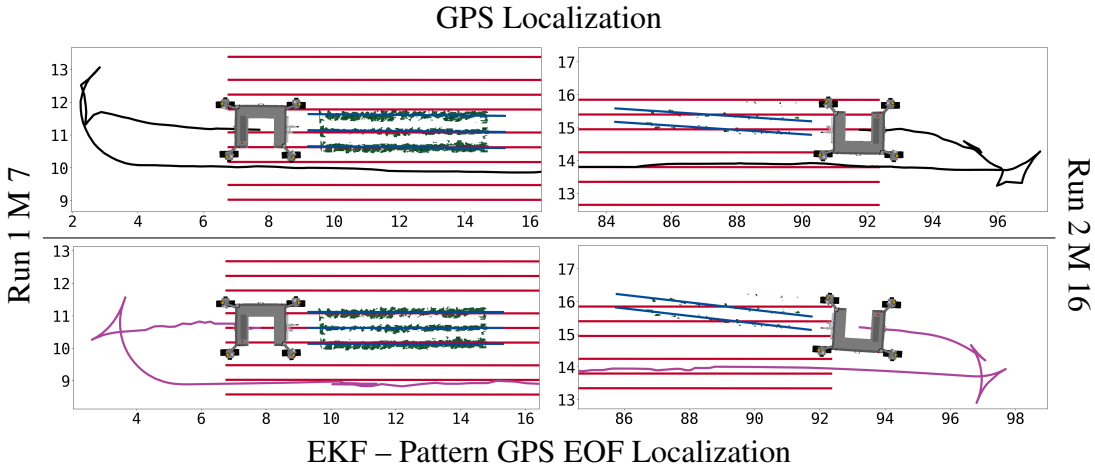

Figure 4.27: This figure shows detailed results for the GPS Localization (top) and the Pattern GPS EOF – EKF Localization (bottom) for interesting marker positions. In Run 1 at marker M 7 (left), our Pattern GPS EOF – EKF Localization shows a high Lateral Error and in Run 2 at marker M 16 (right) a high Heading Error. We show the pose estimates of each localization algorithm as trajectory (solid line) relative to our mapped lines of crop rows (red lines). We also visualize the Feature Map (green cells) as well as the detected Pattern (blue lines) perceived at the marker location.

Despite the fact that the Heading Error is lower compared to the crop-row-following-based approaches, the Lateral Error of the GPS Localization at this marker considerably exceeds our success threshold of $0.1\,\text{m}$ with a value of over $0.2\,\text{m}$. Thus, the estimated position of the vehicle is offset to the true position of the vehicle so that the wheels of the vehicle are directly located on top of a crop row. During autonomous navigation such an incorrect localization result has catastrophic consequences as it causes the navigation system to correct its allegedly undesirable position by steering to the right, when in fact the true position of the vehicle was correct. This in turn causes the vehicle to actually drive over the crop row located on the right side of the wheels. To prevent these situations, it is crucial that the localization algorithm provides an accurate lateral position of the vehicle relative to the crop rows.

Therefore, we also take a closer look at the Lateral Error results in Run 1 at marker M 7 on the left of Figure 4.27, where our Pattern GPS EOF – EKF Localization also exceeds the Lateral Error success threshold. While our localization only slightly exceeds the lateral success threshold of $0.1\,\text{m}$ with an Lateral Error of $0.11\,\text{m}$, the GPS Localization considerably exceeds the success threshold with a Lateral Error of above $0.4\,\text{m}$. This difference in the measured Lateral Error is also clearly visible when overlaying the estimated pose of the vehicle onto our map of crop rows. While our localization slightly exceeds the success threshold, the wheels of the vehicle are still located between the correct pairs of crop rows and also not located on the crop rows. Therefore, the navigation system will continue to track the correct crop rows without driving over the valuable crop rows. This is in contrast to the pose estimate of the GPS Localization, which shows another good

example for a too large Lateral Error. This situation can have two different outcomes, depending on the constrains of the navigation system. First, if there is no constraint that prevents the navigation system from intentionally driving over crop rows, it will attempt to correct the position of the vehicle to track the correct crop row and therefore steer towards the right, again resulting in crushed crops. If there is a constraint that prevents the navigation system from driving over crop rows, the vehicle will continue driving straight thereby following the crop rows, but potentially track an incorrect set of crop rows. Since the second option usually results in driving behavior that does not damage crops despite an incorrect lateral position of the vehicle, enforcing a constraint on the navigation system that prevents it from driving over crops is advisable. Therefore, a large Lateral Error does not necessarily lead to catastrophic failure of the whole navigation system, where crops are crushed by the wheels of the vehicle, but can instead be lessened into a simply inefficient traversal of possibly already traversed neighboring crop rows instead of the target crop rows.

The detailed analysis of these interesting localization results highlights the requirements for the accuracy of the heading and lateral offset pose estimate for successful and precise autonomous navigation on crop fields. Although the Lateral Error of the crop-row-following-based approaches slightly exceeds our success threshold for Run 1 at M 7, it does not cause navigation failure or divergence of the pose estimate in this situation. Therefore, this detailed analysis reaffirms that the heading and lateral accuracy of a crop-row-following-based approach are sufficient to guide a vehicle along crop rows, if the crop rows are detected with an accuracy similar to the accuracy of our crop row detection approach.

Overall, the performance of the two different localization methods, i. e., the GD and the EKF localization, is comparable, where the Pattern GPS EOF – EKF Localization shows slightly lower Longitudinal Error values with a mean Longitudinal Error of 0.39 m compared to the Pattern GPS EOF – GD Localization with a mean of 0.45 m. However, since we did not perform any parameter fine tuning for either of the algorithms, there might be a better parameter configuration for the GD method. On the other side, this also shows that both methods are not too sensitive regarding the exact choice of parameters. For our autonomous navigation system, we used the Pattern GPS EOF – EKF Localization to obtain a GPS-referencedpose estimate of the vehicle relative to the crops. During these runs, our crop row detection method took the major share of the computation time with 0.2 sec per image. Therefore, our Pattern GPS EOF – EKF Localization performs a Correction Step at a frequency of around 5 Hz, which is still sufficient for successful autonomous navigation on crop fields. We used our localization to guide the BoniRob along crop rows (for example shown in *Autonomous Navigation on a Field*[1]), and also to successfully navigate entire crop fields fully autonomously (for example shown in *Autonomous Navigation on Sugar Beets*[2]).

In summary, our evaluation shows that our fused localization approach can indeed leverage the advantages of both the crop row following as well as the GPS-based lo-

[1] https://youtu.be/1alnCUsQc1Y
[2] https://youtu.be/jy_VZIA7Rp0

calization on agricultural fields. We also confirm, that including the end of the field as additional longitudinal measurement for pose correction notably improves the accuracy of the longitudinal pose estimate compared to only integrating GPS information. This is especially valuable when aiming for an autonomous traversal of an entire crop field, including turning at the headlands.

4.5 Conclusion

In this chapter, we presented a fused localization approach that estimates the full pose of the vehicle relative to crops in a global GPS-referenced frame. Our fused localization therefore enables autonomous navigation of entire crop fields, while also providing a pose estimate relative to crops, which is crucial for precision agriculture applications. Our in-depth experimental evaluation on two real-world data sets shows that our fused localization successfully estimates the pose of the vehicle on both data sets. These results therefore confirm our idea of fusing local crop row detections with GPS information using a GPS-referenced map of crop rows to provide a common reference frame. Implicitly, these results also demonstrate that our *Crop Row data association* finds the correct association between detected and mapped crop rows using geometrical consistency constraints. Furthermore, since our fused localization approach is able to maintain the accuracy of a crop row following approach in the heading and lateral estimate, while at the same time having the longitudinal accuracy of at least a GPS-based localization, we show that we consistently fuse both sensor modalities with our fused sensor measurements. Our evaluation of the longitudinal accuracy when additionally also integrating longitudinal position information from detecting the end of the field reveals an increased accuracy of at least $1.1\,\mathrm{m}$ and down to $0.21\,\mathrm{m}$ compared to standard GPS accuracy of around $3.0\,\mathrm{m}$. The overall Heading Error and Lateral Error of the crop-row-following-based localization approaches also indicate that the Angular Error and the Lateral Error of a crop row detection algorithm are directly transferable to the expected accuracy of a localization algorithm that uses these crop row detections for pose correction. In summary, the quantitative results show that our fused localization provides a full pose estimate relative to crops with sufficient accuracy for autonomous traversal of entire crop fields including transition maneuvers at headlands. These results are also validated by our qualitative evaluation as well as several successful autonomous runs on different crop fields, where we employed our fused localization to provide an accurate pose estimate to the navigation system.

During these autonomous navigation runs, we realized that the critical part of traversing the entire crop field is turning at the headlands. This is caused by the fact that the vehicle looses track of the crop rows while leaving the crop field. It then performs the turning maneuver based only on the relative motion measurements without correcting its pose until the crop rows appear again in the field of view of the sensor. Therefore, uncertainty in the pose estimate accumulates during the turning maneuver and the vehicle needs to be re-localized when it perceives the crop rows again after turning. Finding the correct data association between the mapped and observed crop rows after turning "blindly" at the headlands is a critical point for efficient traversal of an entire crop field as it determines

the success of re-localizing the vehicle after turning.

As we briefly discussed in our qualitative evaluation, we implicitly know that our *Crop Row data association* always found the correct data association after turning on both data sets. Otherwise, we would have observed large Lateral Error values of a multiple of the spacing between crop rows, i. e., Lateral Error values of $0.45\,\mathrm{m}$ or above, if the a neighboring crop row would have been incorrectly associated. This confirms that our *Crop Row data association* based on geometric consistency is well suited for crop fields with a crop row structure similar to this vegetable field with irregular spacing of crop rows adjacent to wheel tracks (see also Figure 4.16). However, as we have discussed in the previous chapter, crop fields not necessarily have this irregular spacing. In fact, most of the crop fields that we have seen, have a more equidistant spacing between crop rows, so that it is sometimes not even clear where the tracks for the wheels are located. On these crop fields, with equidistant crop row spacing, data association is highly ambiguous, if it can only rely on the geometry of the crop rows as line features. These ambiguities can cause an incorrect data association that matches the wrong crop rows, which in turn localizes the vehicle relative to the wrong set of crop rows. While this is an incorrect localization result, it does not necessarily result in driving behavior that damages crops: Since the vehicle is localized offset by a multiple of the row spacing, it still traverses the crop rows with high precision by following the tracked set of rows. In practice, such a localization error thus causes the vehicle to skip a few crop rows or to traverse the same part of the field twice. Therefore, incorrect data association does not necessarily result in damaged crops, but definitely reduces the efficiency of the autonomous system. Our motivation for the next chapter is therefore to investigate how we can improve the capabilities of data association, especially in the critical situation of re-localizing the vehicle after turning on the headlands.

Another point for improvement is the longitudinal position estimate. As mentioned in our evaluation, the end of the field was not always detected, especially when the vehicle was approaching headlands that were overgrown with vegetation. One possible direction of further research is therefore to improve the robustness of the *End of the Field detection*.

However, since the *End of the Field detection* only provides a longitudinal position estimate when the vehicle approaches the end of the field so that it is visible in the sensor data, it can not improve the overall longitudinal position estimate throughout the field. In the next chapter we therefore investigate a different approach that can by design yield accurate heading, lateral and longitudinal estimates relative to the crops throughout the entire field.

Chapter 5

Data Association on Individual Plants

In the previous chapter we presented techniques to obtain a full pose estimate across the entire field including transition maneuvers at the headlands to enable precise and reliable autonomous navigation on crop fields. However, the presented techniques might not always re-localize the vehicle after turning resulting in a pose estimate with a large lateral offset to the correct crop rows. This results in inefficient driving behavior, since the vehicle might skip crop rows or traverse the same crop row multiple times. We therefore propose to use individual plant positions as additional features to associate the correct crop rows after turning at the headlands. Finding the correct data association based on individual plant positions is challenging, since plants of the same type are indistinguishable and almost uniformly and densely distributed along the crop rows resulting in many highly ambiguous feature distributions. Inspired by our *Crop Row data association* of the previous chapter, we present a novel data association technique that leverages slight irregularities in the plant distribution along each crop row to find the correct data association. This technique is based on an efficient representation of data association matches that preserves the row structure as well as a novel method that counts the number of matches in a continuous fashion. We evaluate our approach in extensive experiments on real-world data, including a comparison to other non row-based data association techniques. We demonstrate the suitedness of our approach for localization and mapping applications as well as a detailed analysis of the behavior of our approach depending on the amount and quality of the provided input data. We also provide a proof of concept showing qualitative results in form of maps of plant positions of entire agricultural fields obtained by employing our data association approach in a real-world simultaneous localization and mapping (SLAM) application. Overall, we present a powerful row-based data association approach that robustly and efficiently associates indistinguishable, almost uniformly and densely distributed point features to enable re-localization after turning at the headlands.

5.1 Introduction

In the previous chapter we obtained a full pose estimate to enable reliable and precise autonomous traversal of entire crop fields based on detecting the crop row structure in local sensor data. Since a front mounted sensor only captures the crop row structure while driving in-row or after turning when the vehicle faces the crop field again, tracking is lost during the turning maneuver at the headlands. While an autonomous vehicle can perform a turning maneuver solely based on relative motion measurements and GPS information, errors in the pose estimate can accumulate and result in an inaccurate pose estimate after turning. Therefore, the vehicle might continue autonomous traversal of the crop field offset to the desired set of crop rows and thus skip some of the crop rows or travel along the same crop rows multiple times, which is inefficient navigation behavior. In this chapter, we therefore investigate, how the vehicle can be re-localized after turning "blindly" at the headlands. The key requirement to this re-localization is associating the correct crop rows after turning despite the comparably larger error in the pose estimate of the vehicle. Our *Crop Row data association* from the previous chapter successfully re-localized the vehicle in our experimental evaluation leveraging the irregular spacing between crop rows. Since crop fields usually have more equally distributed crop row spacing, our *Crop Row data association* will most likely not perform well on these fields. We therefore propose a novel data association approach that determines the correct data association based on individual plant positions instead of entire crop rows. Inspired by our *Crop Row data association*, this novel data association leverages slight irregularities in the plant distribution along each crop row to find the correct data association. While using individual plant positions as features enables re-localization after turning at the headlands, it also provides more accurate longitudinal position information while the vehicle drives in-row thereby further improving the localization accuracy in the longitudinal direction. Such a precise pose estimate of the vehicle relative to the crops throughout the field does not only enable accurate localization for autonomous navigation, but also opens the door for creating accurate and detailed maps of a crop field from ground vehicle data without having to rely on highly accurate GPS information. In order to correct the pose estimate of the vehicle with respect to a set of mapped plant position features F_m the observed plant position features F_o need to be correctly associated with their correspondence in the map. This means a data association algorithm needs to find a set of matches between the observed and mapped features $M^* \in \mathcal{M} \subset \mathcal{P}\left(F_o \times F_m\right)$ that correctly associates observed features with their corresponding mapped features. We give a more detailed definition of $\mathcal{M}$ in Section 2.2.1.

In computer vision, the correct association between observed and mapped features is often established using the *appearance* of the detected objects, e. g., the pixel information of the local area around the point of interest in the image. The information contained in this local area is usually encoded in a feature descriptor and used to associate features if their descriptors have similar values. These appearance-based data association approaches rely on the assumption that the objects used as features can be reliably identified and therefore distinguished from one another. However, in our setting, with the crops as features, this cannot easily be assumed, since crops of the same type look fairly similar

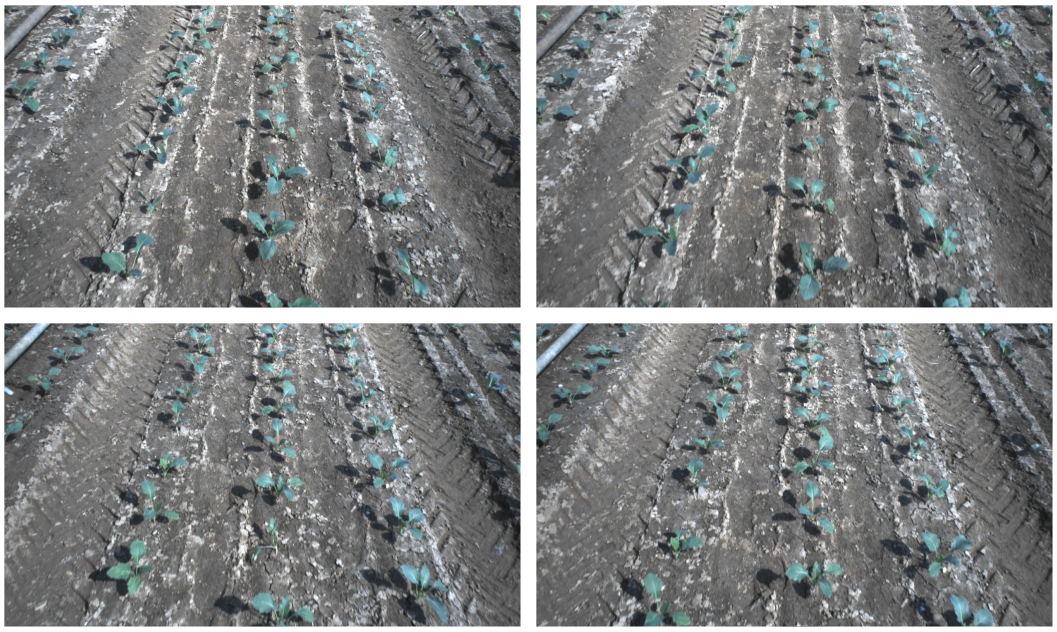

Figure 5.1: This figure shows two different locations around 14 m apart on the same crop row while the vehicle traverses the crop field. The images on the top show the first location. The bottom images show the second location. Images of the same location are only one second apart.

and are hard to distinguish by their appearance. To emphasize this we show four example images from a crop field featuring two different locations on the same field in Figure 5.1. Images of the same location differ only slightly in their point of view since they were captured only one second apart. Without knowing which image shows which location, it is hard to determine whether the plants shown in these images are the same or not – they all look alike. Since crops of the same type are fairly indistinguishable from one another, a data association on individual plants cannot rely on additional information such as appearance to determine the correct association between features. Therefore, we use the information provided by the position of the plant features in both, the observed and mapped set of features.

In literature this is usually realized by leveraging the implicit information contained in the *geometry* of the features of each set. More explicitly, to associate indistinguishable features, the geometry of the observed features F_o is compared with the geometry of the mapped features F_m to find similar patterns. In our setting where the geometry of a feature is a single point, the geometry of a set of features is described by relative distances and angles between these features, or short by the distribution of features. The main idea of a geometry-based data association approach is that a similar pattern, i. e., a similar distribution in the observed and mapped features indicates that the corresponding features forming this pattern are the same and therefore should be associated. Intuitively, a geometry-based data association approach assumes that the correct data association can be found by *maximizing the overlap* between the observed features F_o and mapped fea-

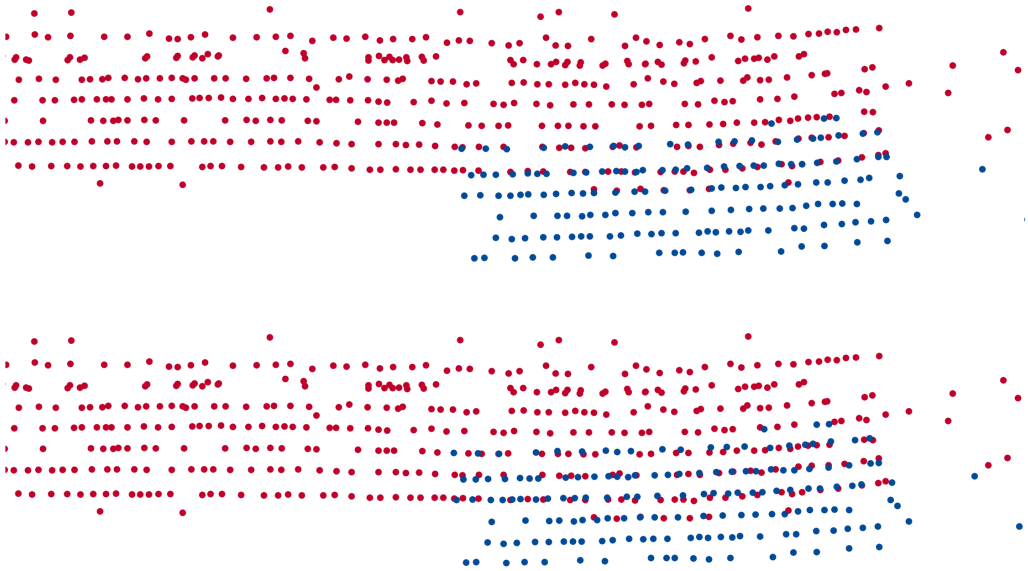

Figure 5.2: This figure shows an example for the distribution of plants positions obtained from real-world data. The observed plant position features F_o are shown as blue points and the mapped plant position features F_m as red points. The top shows the correct overlay of observed and mapped features. The bottom shows an incorrect overlay. Since this shows data taken from a mapping application, a large amount of observed data (blue points in the bottom crop rows) is not yet part of the map. Therefore, the correct data association shown on the top only overlaps two out of six observed crop rows. This in turn causes a large amount of around two thirds of all observed features to remain unmatched.

tures F_m. This works well, if the distribution of features is not uniform so that unique patterns exist to determine the correct data association – or more intuitively, if the best overlap between observed and mapped features is clearly visible. However, in our environment where the crops are sown in equidistant and parallel crop rows at approximately the same distance within a crop row the feature distribution is quite the opposite of non-uniform. Instead, the feature distribution obtained from crop positions on agricultural fields can rather be described as *almost uniform* and *dense*, where dense refers to a large amount of features within a small area. We show examples for the feature distribution on crop fields extracted from real-world data in Figure 5.2. Due to the dense and almost uniform distribution of crops it seems close to impossible to determine the best overlap and therefore the correct data association. Using the examples shown in Figure 5.2, we can determine the challenges that arise for a geometry-based data association approach on indistinguishable, densely and almost uniformly distributed point features more precisely:

First, due to the dense and almost uniform feature distribution it is not clear how to best overlay the observed features over the mapped features. This is due to the multitude of *ambiguous situations* caused by the similar distribution of features throughout the field.

Therefore, for a geometry-based data association approach to succeed, it needs to detect subtle differences in the feature distribution, i. e., it needs to have a *high discriminative ability*, to resolve ambiguities and find the correct data association.

Second, due to the dense distribution of features, the data association approach also has to be able to handle a *large amount of features* efficiently. This is crucial since the search space of all possible data association matches $\mathcal{M} \subset \mathcal{P}(F_o \times F_m)$ grows exponentially in the number of features. Handling large amounts of features is especially challenging for a geometry-based data association approach that requires high discriminative ability, since the commonly applied independence assumption, i. e., associating each feature independently from the other features of the set to facilitate data association on large feature sets cannot be applied.

To differentiate even slight differences in the feature distribution, the *relative geometry* between features needs to be considered. However, considering relative geometry such as angles and distances between points directly contradicts associating features independently from each other. Therefore, the third challenge for a relative geometry-based data association approach for our scenario is that it *cannot assume independence* between features.

While the data association needs to be highly sensitive to detect subtle differences in the feature distribution to find the correct data association, it still needs to be robust against inaccurate information such as noisy feature positions or false positive or missing feature detections caused by the feature detection pipeline. This is a crucial requirement for the data association to be applicable in real-world applications and therefore poses the fourth challenge since being *robust against noisy input data* is directly opposite to being highly sensitive to slight differences in the feature distribution.

For mapping applications we also need to consider a fifth challenge: During mapping, only parts of the environment are represented in the map. Therefore, the data association algorithm needs to be able to handle situations, where parts of the observation are not represented in the map. This means that some of the observed features might not yet be represented in the map and thus cannot be associated and should remain unmatched. Usually, on sparse and non-uniformly distributed features only few observed features fall into this category, so that there is no need to explicitly handle these unmapped features during data association. In our environment though, due to traversing the field aligned with the crop rows, usually a large subset of the observed features, i. e., entire crop rows, fall into the category of these unmapped features. In combination with the large amount of ambiguous situations caused by the dense and almost uniform feature distribution, correctly handling situations with *unmapped observed features* becomes a challenge. In summary, a relative geometry-based data association on individual plants needs to satisfy the following requirements:

1. *Highly Discriminative*: Detect subtle difference in feature distribution to resolve ambiguous situations caused by almost uniformly distributed features.

2. *Efficient*: Efficiently handle large amounts of features due to densely distributed features.

3. *No Independence*: Independence between features cannot be assumed since the geometric relations between them are key.

4. *Robustness*: Robustness against noisy input data is crucial for real-world application.

5. *Unmapped Features*: For mapping applications, unmapped observed features need to remain unmatched.

In this chapter, we present a relative geometry-based data association approach on individual plant positions that tackles all these challenges and therefore efficiently and robustly associates large amounts of indistinguishable, densely and almost uniformly distributed point features. The key idea of our approach is to leverage the row structure of crop fields to obtain an efficient and discriminative data association that can resolve the multitude of ambiguous situations caused by the dense and almost uniform feature distribution. Therefore, we mathematically define this row structure in form of a set of *Plant-Rows*.

We then use this set of *Plant-Rows* to define a two-dimensional parametrization that maps a parameter $s \in S \subset \mathbb{N} \times \mathbb{R}$ to a corresponding set of data association matches $M(s) \in \mathcal{M}$, so that only data association matches $M \in \mathcal{M}$ that preserve the crop row structure are represented. This dramatically reduces the number of possible data association matches, since only sets of matches $M \in M(S)$ that preserve the crop row structure need to be considered during data association and therefore enables our data association to efficiently process large amounts of features.

Based on our definition of the row structure, the Plant-Rows, and our parametrization of the data association matches, we present a novel method of measuring how well observed data F_o is overlaid with the mapped data F_m given a set of data association matches $M \in M(S)$. Our key idea for this novel method is that we use our parameterized representation of data association matches to define a continuous measure for the overlap between observed and mapped features over all features jointly. Traditional approaches usually count the overlap as the discrete number of matched observations. Since our method leverages the parameterized representation of the data association matches, it does not require an explicit definition of pairs of matched features and thus can define a continuous measure of the overlap between observed and mapped features. Our assumption behind measuring the overlap in this way is that without an explicit decision on whether two features are matched or not, our method is more robust against noisy input data, while still maintaining its discriminative ability. Computing the overlap based on our parameterized representation has additional advantages: First, since it represents a whole set of data association matches, all features are considered jointly by design as required. Second, since the parametrization only represents matches that preserve the row structure, it also ensures that geometric relations between features are preserved and therefore allows for an efficient computation of the overlap between observed and mapped features.

To enable our data association approach to properly handle unmapped crop rows, we introduce a regularization technique. Since the vanilla version of our proposed method is

biased towards matching as many crop rows as possible, we introduce this regularization technique to mitigate this bias. We also present an improved variant of our method that can measure the overlap between observed and mapped features without bias and therefore does not require any regularization.

We also provide a proof of concept of our row-based data association approach by employing it in a simultaneous localization and mapping (SLAM) application on real-world data. To this end, we present our data preprocessing pipeline as well as a pose graph-based SLAM framework that utilizes the results of our data association to build maps of individual plant positions over entire crop fields relying only on data collected from a ground vehicle. We summarize our contributions towards data association on indistinguishable, densely and almost uniformly distributed features as follows:

1. *Plant-Rows*: We mathematically define the row structure of crops as sets of Plant-Rows.

2. *Parametrization*: We define a efficient two-dimensional representation of data association matches that preserves the row structure and therefore the relative geometry between features. This representation enables efficient computations on large amounts of features without assuming independence between features.

3. *Continuous Measure of Overlap*: We define a novel method for measuring the overlap between observed and mapped features. Based on our parameterized representation of sets of data association matches we compute the overlap as a continuous measure to increase the robustness of our data association algorithm against noisy input data.

4. *Regularization Technique*: We introduce a regularization technique to counter the bias of our vanilla method towards matching as many observed crop rows as possible. Mitigating this bias is crucial for mapping applications.

5. *Unbiased Variant for the Continuous Measure of Overlap*: We also present an improved variant of our novel method that is unbiased and therefore does not require any regularization.

6. *Proof of Concept*: We provide a proof of concept for our data association approach by utilizing it in a SLAM application on real-world data.

In the following, we first give an overview on related data association algorithms. In the next section, we present our row-based data association approach in detail, including our definition of Plant-Rows, the parametrization of data association matches and our novel method that defines a continuous measure of the overlap between observed and mapped features. We also present the preprocessing pipeline as well as the SLAM framework that we utilize to provide the proof of concept for our data association approach in a separate section. In our experimental evaluation, we first compare the performance of our data association to other data association algorithms to confirm that considering the row structure is crucial for efficient and robust data association. In the second experiment,

we determine the success rate of our data association as well as the accuracy of our data association to confirm that it is well suited to enable localization and mapping applications for autonomous navigation on crop fields. The third experiment is focused on the performance of our data association approach along entire crop rows to demonstrate that it can indeed improve the longitudinal component of the pose estimate while the vehicle traverses the field. Finally, we provide a qualitative evaluation of our data association employed in a SLAM framework on real-world data to provide a proof of concept for our approach.

5.2 Related Work

The problem of data association can be described as finding the correct association between a set of observed and a set of mapped features. A solution to the data association problem therefore defines a set of associations that match a feature of the set of observations to an element of the set of mapped features. The problem of data association is well studied in the literature. For a better overview, we divide these data association techniques into different categories and sub-categories and discuss their relevance for data association on indistinguishable, densely and almost uniform point feature distributions as well as how they relate to our data association approach.

Many state-of-the-art techniques for associating points of interest detected in image data are based on the idea that these points can be distinguished by their appearance in the image, i. e., by the pixel information contained in a local patch around the detected point. Examples for such *appearance*-based techniques are SIFT, BRIEF or ORB [Calonder et al., 2010, Lowe, 1999, Rublee et al., 2011]. While how the appearance of an object, i. e., the information contained in this patch, is described varies for each method, they are all based on extracting additional information about the appearance of an object from the image data and can therefore be categorized as *appearance*-based data association techniques. This additional information is usually stored in a descriptor for each detected feature. The data association then uses this descriptor to determine the correct association between features. The key assumption is that features with similar descriptors represent the same object and should therefore be associated. However, since crops of the same type look quite similar, such appearance-based data association approaches usually struggle to find the correct data association on images of agricultural fields as confirmed by Chebrolu et al. [2018].

Since individual crops on an agricultural field are indistinguishable, appearance-based data association approaches are not well suited for this setting. Without additional information such as appearance, the information contained in the geometry of the detected point features needs to suffice to determine the correct data association. There are many data association techniques in the literature that consider only the geometric information of the provided features, which we therefore categorize as *geometry*-based data association techniques. The key assumption of geometry-based algorithms is that if the geometric properties of an observed feature are similar to the geometric properties of a mapped feature, these features correspond to the same object in the environment.

A well-known geometry-based technique is the *Nearest Neighbor* data association. The idea of this approach is that an observed feature corresponds to a mapped feature if it is the nearest neighbor, i. e., closest to the mapped feature. In practice this works well if the pose estimate required to project the observed features onto the mapped features is accurate. If the pose estimate is inaccurate, the projected observed features might not necessarily lie close to their corresponding mapped feature and therefore, the Nearest Neighbor data association fails to find the correct data association in these cases. This means that in applications, where pose correction occurs at high frequency so that the error accumulated between consecutive pose correction steps remains small, such as pose tracking or localization applications, the Nearest Neighbor data association is a good choice [Bar-Shalom and Fortmann, 1988, Castellanos et al., 1999, Feder et al., 1999]. Another advantage of the Nearest Neighbor data association is that it treats features independently, i. e., observed features are independent from each other and mapped features are independent from each other. This assumption allows associating each observed feature independently and therefore enables efficient computation of a solution on a large number of feature points, e. g., measurements from a three-dimensional LIDAR sensor.

However, since the Nearest Neighbor data association relies on this independence assumption, it struggles to find the correct data association in ambiguous situations, i. e., in situations where multiple mapped features lie close to an observed feature. As discussed in the introduction, we expect to encounter many ambiguous situations in our scenario where the features are densely and almost uniformly distributed. For a geometry-based data association to disambiguate these situations, additional information contained in the relative geometry between features needs to be considered during data association. We summarize these kind of approaches as the sub-category of *relative-geometry*-based data association techniques.

A common approach to enforce geometric relations while partially maintaining the efficiency of the Nearest Neighbor data association is employing the Nearest Neighbor data association in a *random sample consensus* (RANSAC) algorithm. The key idea of a RANSAC-based data association is that certain geometric relations are preserved if the matches can be described by a geometric transform. Therefore, the model of the RANSAC data association is a geometric transform that can be efficiently sampled by a small set of pairwise point matches. Using the sampled transform, the observed feature points are overlaid with the mapped feature points and the matches are determined using the Nearest Neighbor data association. The number of matched features is used as the score for this candidate model of the RANSAC iteration. The RANSAC loop continues sampling transforms based on point matches until a good model is found. The best transform and its corresponding matches are the output of this RANSAC data association. In computer vision, where matching sets of points detected in images is also an interesting problem, approaches with a similar idea to the RANSAC data association approach have been presented. The idea these approaches have in common with the RANSAC data association is that the matches should be estimated in conjunction with a geometric transform to ensure that geometric relations are preserved. This includes the approach presented by Gold et al. [1998], where an affine transform is estimated jointly with the point set matches, as well as the approach by Cross and Hancock [1998], where the relative ge-

ometry is encoded on a graph structure. Cross and Hancock use this graph structure to formulate a dual-step expectation maximization algorithm, where the transform and the matches between point sets are alternately optimized. The algorithm presented by Carcassoni and Hancock [2003] is based on the expectation maximization algorithm by Cross and Hancock [1998], but uses a spectral graph representation to be more robust against noisy point positions and missing points. The idea of representing geometric relations in a graph structure has also been employed by Bailey et al. [2000] for a robotics application: Bailey et al. represent geometric relations between point and line features in a graph structure and obtain the data association by computing the maximum clique of this graph. In contrast to the graph-based approaches from computer vision, the solution is directly computed based on the graph structure without a dual optimization including a geometric transform.

Another idea from the field of computer vision by Wolfson and Rigoutsos [1997] is to represent the geometric structure of point sets in a hash table. This hash table is based on the idea that each two points of the point set define their own reference coordinate system. A more recent approach in computer vision by Yang et al. [2015], inspired by Wolfson and Rigoutsos [1997], leverages the idea of representing relative geometry using reference coordinate frames. In the approach presented by Yang et al. [2015], they use this idea to encode local geometric relations in a descriptor, similar to the appearance-based data association approaches. The main difference to the appearance-based descriptors is that is does not encode pixel information of a local patch around the point, but instead encodes relative geometric information of the k nearest neighbors around the point. As stated by Yang et al. the key assumption of this technique is that "points are randomly distributed" so that "the relative positions between points in a local point set is very characteristic". The almost uniform distribution of plant positions in our application scenario directly contradicts this assumption and thus the geometric descriptor presented by Yang et al. is not well suited for our scenario. Nevertheless, the work by Chebrolu et al. [2018] successfully applies the idea of a geometric descriptor to register UAV images of crop fields. The authors present an approach based on the geometric descriptor presented by Yang et al. [2015] but instead of plant positions, as in our approach, the authors use *Gaps* within the crop row structure of the field as feature points. Since Gaps, i. e., locations within a crop row where crops are missing, are more non-uniformly distributed compared to the crop positions, they encode characteristic relative positions and enable successful data association as by Chebrolu et al. [2018]. Another beneficial effect of considering Gaps instead of crops is that there are inherently less Gaps than crops to be found on a crop field, which in turn significantly reduces the amount of features that need to be considered during data association.

A more traditional data association technique from robotics that is also based on geometric relations is the *Geometric Constraints Branch and Bound* algorithm presented by Neira et al. [2003]. This technique is based on the more commonly known probabilistic *Joint Compatibility Branch and Bound* data association introduced by Neira and Tardos [2001]. In the JCBB algorithm, the best data association is computed by finding the largest set of *jointly compatible* pairings. The key idea of this approach is that the joint compatibility test can be used to perform an efficient branch and bound tree search

on the space of all possible pairings. The GCBB algorithm is a more efficient variant of the JCBB algorithm since it employs the same branch and bound technique presented by Neira and Tardos [2001] but instead of the *joint compatibility* test uses a geometric constraints test during tree search. In the GCBB algorithm the joint compatibility test is only applied at the end of the search, i. e., when a leaf of the tree is found, to validate the set of data association matches.

Since the work by Chebrolu et al. [2018] already shows that appearance-based data association techniques are not well suited in agricultural environments, we compare our data association to three geometry-based data association techniques in our experimental evaluation inspired by the following techniques:

1. Nearest Neighbor: This data association is a widely used technique for localization and mapping applications that can efficiently handle large amounts of features in contrast to most other geometry-based data association approaches.

2. Geometric Constraints Branch and Bound by Neira et al. [2003]: This algorithm can find the optimal data association due to its efficient tree search over the whole space of possible data associations.

3. Gaps Descriptor by Chebrolu et al. [2018]: This is a closely related data association technique used to register UAV images of crop fields.

At first glance, the tracking algorithm for moving indoor systems equipped with a LIDAR sensor presented by Weiss et al. [1994] does not seem related to our data association problem. However, when considering LIDAR scan lines as a large set of points arranged along a row, or in this case, scan line, the relevance for our setting with large amounts of points aligned in crop rows becomes apparent. The key idea of this tracking approach is to leverage the cross-correlation between two LIDAR scans to find the correct orientation of the vehicle. Inspired by Weiss et al., we also use the cross-correlation over points distributed along a line to maximize the overlap between both sets of points in our data association approach.

5.3 Row-Based Data Association on Individual Plants

In this section, we present our row-based data association approach for localization and mapping applications on individual plant positions. In order to enable a localization or mapping technique to correct its pose estimate with respect to the information obtained from individual plant positions, a robust data association between the set of observed plant features F_o and the set of mapped plant features F_m is crucial. In our scenario, where the features are densely and almost uniformly distributed plant positions, determining the correct data association is challenging. For a better intuition, we also provide an example of a set of observed and mapped features in Figure 5.2 obtained from real-world data. For a human it is hard to determine, whether the top or the bottom data association is correct due to the high density of the features as well as the high ambiguity caused by the almost

uniform distribution. However, the key assumption for our data association approach on individual plants is that we can leverage the *row structure* present in crop fields to resolve enough ambiguity, which then facilitates robust data association. In order to create such a *row-based* data association technique we first need to specify the requirements and criteria that are crucial for successful data association in our scenario. As discussed in detail in Section 2.2, this entails defining *hard requirements* on the set of data association matches that determine whether a set of data association matches is *valid* and should be considered as a possible solution to the data association problem. These hard requirements then define a subset of the set of all possible data association matches $\mathcal{M}$, called the *set of valid data association matches* $\mathcal{M}^{\text{valid}}$. Additionally to these hard requirements, designing a data association technique also includes defining *soft criteria* to quantify whether a set of data association matches is *close to the correct data association*. These soft criteria are then used to define a Target Function $t : \mathcal{M} \to \mathbb{R}$, that assigns a value to each set of data association matches $M \in \mathcal{M}$, where higher values correspond to a set of data association matches that is closer to the correct data association. The best data association M^* is then found by searching the space of valid data association matches $\mathcal{M}^{\text{valid}}$ to find the set of data association matches with the highest value $t\,(M)$. Recalling Section 2.2.1 using these hard requirements and soft criteria to define the set of valid matches $\mathcal{M}^{\text{valid}}$ and the Target Function t the solution to the data association problem for a set of observed features F_o and a set of mapped features F_m can be defined as follows:

$$\mathcal{M}^{\text{valid}} \subseteq \mathcal{M} \subset \mathcal{P}\left(F_o \times F_m\right) \qquad\qquad M^* = \underset{M \in \mathcal{M}^{\text{valid}}}{\operatorname{argmax}}\, t\,(M)$$

We first state the hard requirements. Our key idea is to leverage the information contained in the row structure of crop fields to define, whether a set of data association matches $M \in \mathcal{M}$ is valid or not. In Figure 5.3, we show examples for a valid set of matches based on crop rows on the top left and based on feature matches on the top right. Below the valid set of matches we show invalid sets of matches, that for different reasons do not preserve the row structure. We use those four examples of invalid matches to formulate our hard requirements. The most obvious requirement, which follows from the example of inconsistent feature matches (second row on the right), is to enforce *row-wise matches*, i. e., observed plant features of the same crop row should also be matched to the same corresponding mapped crop row. Additionally, looking at the bottom row of invalid examples, we have matches that either change the relative order between crop rows (left) or between features within a row (right). This leads to two hard requirements: The order between crop rows should be preserved, i. e., *preserve inter row order*. The order between features within the same crop row should be preserved, i. e., *preserve intra row order*. Finally, considering what we learned from the data association in the previous chapter, we also include the requirement, that matches between crop rows should be *consistent*. An example of an inconsistent row match is shown in Figure 5.3 on the left of the second row. Here, the relative distances between adjacent crop rows are not similar, i. e., the distance between the observed crop rows ΔR_o is much smaller than the distance between the corresponding mapped crop rows ΔR_m. We give a more detailed explanation

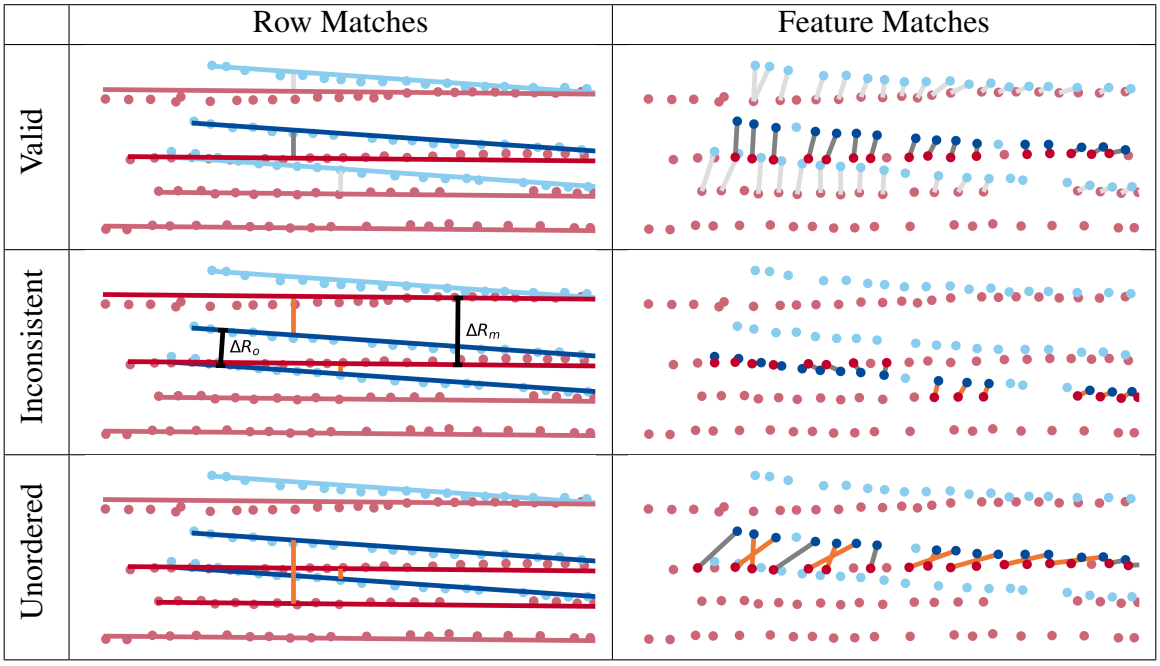

Figure 5.3: This figure shows different data association matches on an example set of observed (blue points) and mapped features (red points) as well as their corresponding crop rows (blue and red lines respectively). We indicate matched observed and mapped features by connecting them with lines on a per row basis on the left and per plant position on the right. Valid matches are shown in gray, matches that cause a set of matches to be invalid are highlighted in orange. The top row shows an example for a valid set of matches, the lower two rows show examples for invalid matches. These four examples of invalid matches illustrate the ideas behind the first four hard requirements of our row-based data association approach.

about the definition of consistent crop row matches in Section 4.3.1. Additionally to the hard requirements that are specific to the row structure of our data, we also include the hard requirement of *locality*. This means, that we only consider data association matches, that infer a reasonable amount of pose correction onto the current pose estimate. This constraint is usually used to reduce the number of valid data association matches for computational feasibility. However, in our case, we also apply the locality requirement to reduce the amount of ambiguous data association matches to increase the robustness of our data association approach. In summary, a set of data association matches is valid for our data association approach, if it satisfies the following five hard requirements:

1. *Row-Wise Matches*: Features from the same observed row should be matched to the same mapped row.

2. *Preserve Inter Row Order*: Adjacent observed rows should be matched to corresponding adjacent mapped rows.

3. *Preserve Intra Row Order*: Adjacent observed features within the same row should

be matched to corresponding adjacent mapped features.

4. *Consistent Row Matches*: The relative distance between adjacent observed rows and the corresponding adjacent mapped rows should be similar.

5. *Locality*: Data association matches should only infer a bounded amount of correction on the current pose estimate.

The soft criteria define, how close a given set of data association matches is to the correct data association. Since the correct data association is, of course, not known, the distance between a set of data association matches and the correct data association cannot be measured directly. Therefore, it is usually assumed that a set of data association matches is close to the correct data association, if it explains the observed data well given the information from mapped data. A Target Function thus needs to quantify, how well a set of matches $M \in \mathcal{M}^{\text{valid}}$ *overlays* the observed data F_o onto the mapped data F_m. As discussed in Section 2.2.2, the most common strategy to quantify this overlap is counting the number of matched observations based on the *Positive Information* criterion. This works well on data association problems with a distinctive feature distribution, since only the correct data association can match all observed features (see Figure 5.4, top). With an almost uniform feature distribution as in our setting, many sets of data association matches maximize the number of matched observed features, although they are not necessarily close to the correct data association (see Figure 5.4, bottom). For a higher discriminative ability, our Target function should therefore also consider *Negative Information*. As explained in Section 2.2.2, the *Negative Information* criterion is based on the idea that the absence of an observation also contains valuable information. If a mapped feature lies within the *detection range*, but no observation is associated with the mapped feature, the observed data does not overlay well with the mapped features. Therefore, the Target function should penalize sets of data association matches, if mapped features within the *detection range* remain unmatched. In Figure 5.4 we illustrate how additionally considering *Negative Information* increases the discriminative ability of a Target Function and thus improves the robustness of the data association strategy on ambiguous feature distributions.

Strategies that consider both, *Positive* and *Negative Information* are usually realized by counting the number of matched observations and subtracting the number of unmatched mapped features to define the values of the Target Function. However, counting matches would require a hard decision on whether two features match or not, which is difficult in ambiguous situations. Since we expect to encounter many ambiguous situations in our setting with almost uniform and densely distributed features, we aim at designing a Target Function that does not require a discrete decision on whether two features match. Without this discrete decision, the number of matches cannot be counted as a discrete number. Instead, the Target Function needs to define a continuous measure on how well the observed features overlap the mapped features. Therefore, we call this the *Continuity* criterion.

Finally, we need to add a fourth criterion, which is important for mapping applications, where the map is only partially known. In these applications, the vehicle is exploring the

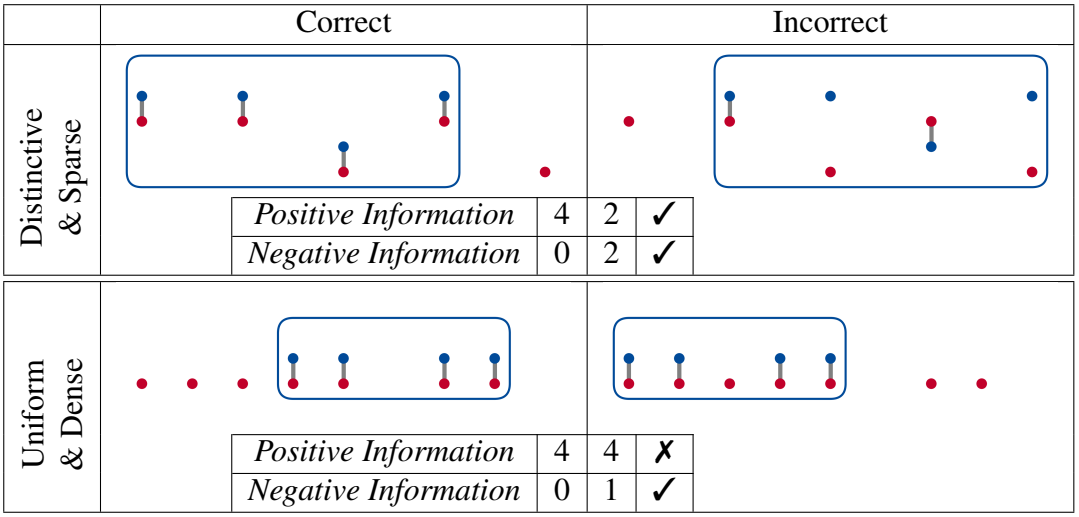

Figure 5.4: This figure showcases the *Positive* and *Negative Information* criteria for discriminative and sparsely distributed features on the top and uniform and densely distributed features on the bottom. The observed and mapped features are shown as blue and red points respectively. The *detection range*, which determines the mapped features relevant for the *Negative Information* criterion, is illustrated as blue outline. The correct set of matches, i. e., the correct data association, is shown on the left. An example for an incorrect set of matches is shown on the right. The sets of matches are visualized by connecting the observed feature with its corresponding mapped feature using a gray line. In the top setting with discriminative and sparsely distributed features, both, *Positive* and *Negative Information* can distinguish the correct from the incorrect set of matches. In the bottom example, with uniform and densely distributed features, the *Positive Information* criterion cannot discriminate the incorrect set of matches from the correct set of matches, since all observations are matched in both cases. In contrast, the *Negative Information* criterion can distinguish between both, since the incorrect set of matches has an unmatched map feature within the detection range.

environment to create a map by incorporating previously unobserved data into the current incomplete map of the environment. Thus a certain amount of observed data is usually not part of the map. In our case, the unmapped portion of observed data corresponds to crop rows at the edge of the field of view. In these cases the correct data association includes not matching the observed crop rows that are not part of the current state of the map. This directly contradicts the *Positive Information* criterion of maximizing the number of matched observed features. This means that a Target Function strictly following the *Positive Information* criterion will always be biased towards matching as many observed features and thus as many crop rows as possible. Since this is usually not correct for mapping applications we introduce a fourth criterion: The Target Function should not be *biased* towards matching as many crop rows as possible. In summary, we design a Target Function that quantifies data association matches according to the following four soft criteria:

1. *Positive Information*: Maximize the number of matched observations.
2. *Negative Information*: Minimize the number of unmatched map features within the detection range.
3. *Continuity*: Measure overlap as a continuous value without deciding whether two features match.
4. *No Bias*: No bias towards matching as many rows as possible.

To define the set of valid data association matches $\mathcal{M}^{\text{valid}}$ according to our hard requirements we first need to establish a mathematical definition of a row structure on a set of plant features, called a set of *Plant-Rows*. In addition to the set of Plant-Rows, we also define properties and operations on these sets of Plant-Rows that are required to properly define the set of valid data association matches $\mathcal{M}^{\text{valid}}$. We then use these sets of Plant-Rows to define a two-dimensional parametrization of the set of data association matches and define the set of valid matches $\mathcal{M}^{\text{valid}}$ with respect to this parametrization. Finally, we use the soft criteria to design variations for the Target Function t and discuss the properties of each variation in detail.

5.3.1 Definition and Properties of Plant-Rows

The main idea of our data association approach is to leverage the distribution of plant features in crop rows to determine whether a set of data association matches is valid. Therefore, we need to mathematically define this row structure of plant features, called *Plant-Rows*, as well as the properties and operations on these Plant-Rows that are relevant to define the set of valid data associations.

In the following, we assume that the knowledge about which plant feature belongs to which crop row is given a priori. This means that for a given set of plant features $F_p := \left\{p_1, \dots, p_{n_{F_p}}\right\} \subset \mathbb{R}^2$ and k crop rows, we know the mapping $\mathrm{row}\colon F_p \to \{1, \dots, k\}$, $p \mapsto j$, that assigns each feature p to its corresponding j-th crop row. In Section 5.4, we explain in detail how we extract this information from raw data to obtain this mapping row for a set of features F_p. For better intuition, we give an example in Figure 5.5, where each crop row and its associated plant features are highlighted in a different color.

Using this mapping row, we now define our set of Plant-Rows based on a two-dimensional *reference coordinate system* that is aligned with the direction of the crop rows. The first coordinate describes the position of a feature along the crop rows, i. e., a *longitudinal* coordinate, and the second coordinate describes the position of a feature in the direction that is orthogonal to the crop rows, called *lateral* coordinate. We then use this coordinate system to define an *order on the plant features* according to the longitudinal component as well as an *order on the crop rows* according to the lateral component. These orders then enable us to validate the hard requirements of preserving inter and intra row order for a given set of data association matches. We also utilize this reference coordinate

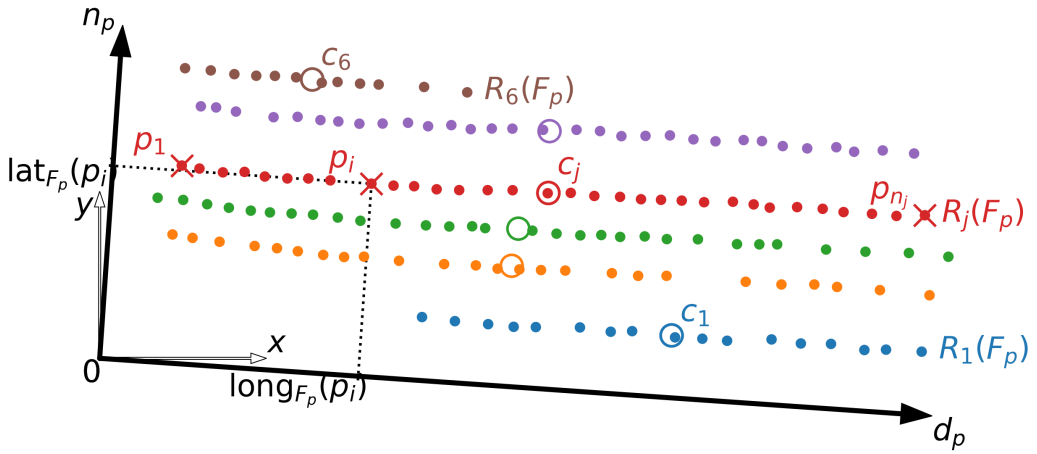

Figure 5.5: This figure shows an example for a set of Plant-Rows $\mathcal{R}(F_p)$ with features of different colors for each crop row $R_1(F_p), \ldots, R_6(F_p) \in \mathcal{R}(F_p)$. The position of the centroid $c_1, \ldots, c_6$ of each crop row is marked using a circle. The reference coordinate system defined by this set of Plant-Rows according to $\{d_p, n_p\}$ is illustrated with black arrows. The longitudinal and lateral coordinates are annotated as black dotted lines for a point $p_i \in F_p(j)$ in the j-th crop row $R_j(F_p)$. The indexed annotations of the crop rows $R_1(F_p), \ldots, R_6(F_p)$ as well as the indexed annotations of the plant features of the j-th crop row (red crosses) demonstrate how the crop rows and plant features are ordered according to the corresponding lateral and longitudinal coordinates.

system to define a *distance measure between crop rows* based on the lateral coordinate. This distance measure then allows us to validate the fourth requirement of consistent row matches, where relative distances between pairs of matched crop rows need to be preserved. In order to determine whether the order between plant features is preserved, we define a *projection* that transfers an observed plant feature from its crop row to its position in the corresponding matched row.

Plant-Rows Given a set of plant position features $F_p := \left\{p_1, \ldots, p_{n_{F_p}}\right\} \subset \mathbb{R}^2$ and k crop rows with a mapping $\text{row}\colon F_p \to \{1, \ldots, k\}$, $p \mapsto j$, that assigns each feature p to its corresponding j-th crop, we define a set of Plant-Rows $\mathcal{R}(F_p)$ as follows:

$$\begin{aligned} F_p(j) &:= \{p \in F_p \mid \text{row}(p) = j\} \\ n_j &:= |F_p(j)| \\ R_j(F_p) &:= (c_j, F_p(j)) \\ \mathcal{R}(F_p) &:= (d_p, n_p, \{R_1(F_p), \ldots, R_j(F_p), \ldots, R_k(F_p)\}) \end{aligned}$$

The set of Plant-Rows $\mathcal{R}(F_p)$ contains plant features in F_p assigned to their corresponding crop rows $R_1(F_p), \ldots, R_k(F_p)$. We also compute the centroid c_j for each crop row $R_j(F_p)$ as the average over all plant features of that row. Assuming approximately parallel crop rows, we define a direction $d_p \in \mathbb{R}^2$ and its orthogonal component $n_p \in \mathbb{R}^2$

on the set of Plant-Rows $\mathcal{R}(F_p)$, where d_p and n_p are the direction and the normal direction of the longest row of the set. We show an example for a set of Plant-Rows in Figure 5.5.

Order on Plant-Rows The direction d_p and normal direction n_p define a coordinate system in $\mathbb{R}^2$ on F_p, i. e., we can split each point $p \in F_p$ into a longitudinal component $\text{long}_{F_p}(p)$ and a lateral component $\text{lat}_{F_p}(p)$ using the linear combination:

$$p = \text{long}_{F_p}(p) \cdot d_p + \text{lat}_{F_p}(p) \cdot n_p$$

The longitudinal coordinate $\text{long}_{F_p}(p)$ describes, where along a crop row the point p is located relative to the origin. The lateral coordinate $\text{lat}_{F_p}(p)$ describes the displacement of p from the origin along the normal direction n_p (see Figure 5.5). We also define the longitudinal and lateral distance between two points $q_1, q_2 \in F_p$ as the component-wise absolute distance:

$$\begin{aligned} \Delta \text{long}_{F_p}(q_1, q_2) &:= \left| \text{long}_{F_p}(q_1) - \text{long}_{F_p}(q_2) \right| \\ \Delta \text{lat}_{F_p}(q_1, q_2) &:= \left| \text{lat}_{F_p}(q_1) - \text{lat}_{F_p}(q_2) \right| \end{aligned}$$

The lateral and longitudinal coordinates then define an order on the set of Plant-Rows as well as each set of plant features within each row $R_j(F_p)$ respectively (see Figure 5.5). Using the lateral coordinate of the centroids c_j of the j-th crop row $R_j(F_p)$, we sort the set of Plant-Rows in ascending order, such that $R_1(F_p)$ has the centroid with the smallest lateral coordinate and $R_k(F_p)$ has the centroid with the largest lateral coordinate. Similarly we arrange the plant features in each crop row $p_i \in R_j(F_p)$ in ascending order according to their longitudinal coordinate. This results in the sorted set of Plant-Rows $\mathcal{R}(F_p)$:

$$\begin{aligned} \mathcal{R}(F_p) &:= (d_p, n_p, \{R_1(F_p), \ldots, R_k(F_p)\}) \\ R_j(F_p) &:= (c_j, F_p(j)) \\ F_p(j) &:= \{p_1, \ldots, p_{n_j}\}, \text{ where} \\ \forall j \in [1, k-1]: &\quad \text{lat}_{F_p}(c_j) \leq \text{lat}_{F_p}(c_{j+1}) \\ \forall i \in [1, n_j - 1]: &\quad \text{long}_{F_p}(p_i) \leq \text{long}_{F_p}(p_{i+1}) \end{aligned}$$

In the following we always assume that the set of Plant-Rows is sorted. During data association, we usually have two sets of plant features F_o and F_m, where the first corresponds to the set of observed features and the second to the set of mapped features. For the following definitions, let $\mathcal{R}(F_o)$ be the set of Plant-Rows for F_o and $\mathcal{R}(F_m)$ the corresponding set of Plant-Rows for F_m.

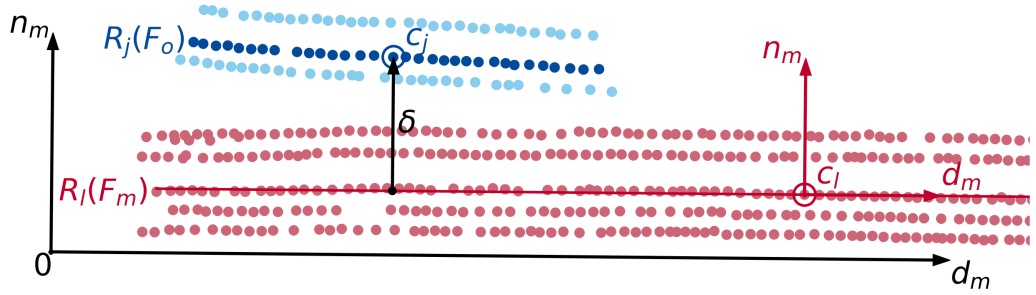

Figure 5.6: This figure illustrates our definition of the line to line distance between two matched crop rows $R_j(F_o) \in \mathcal{R}(F_o)$ (blue) and $R_l(F_m) \in \mathcal{R}(F_m)$ (red). The corresponding centroids c_j and c_l are annotated using a blue and red circle respectively. The signed line to line distance δ between the rows is shown as a black arrow.

Line to Line Distance When computing the valid set of data association matches, the distance between matched crop rows is an important criterion. Therefore, we define the signed line to line distance δ, and the absolute line to line distance Δ between two crop rows $R_j(F_o)$ and $R_l(F_m)$ of not necessarily the same set of Plant-Rows as follows:

$$\begin{aligned} \delta(R_j(F_o), R_l(F_m)) &:= \operatorname{lat}_{F_m}(c_j) - \operatorname{lat}_{F_m}(c_l) \\ \Delta(R_j(F_o), R_l(F_m)) &:= |\delta(R_j(F_o), R_l(F_m))| \\ &= |\operatorname{lat}_{F_m}(c_j) - \operatorname{lat}_{F_m}(c_l)| \\ &= \Delta \operatorname{lat}_{F_m}(c_j, c_l) \end{aligned}$$

For this line to line distance measure, we use the lateral distance according to $\mathcal{R}(F_m)$ between the centroids c_j and c_l of the lines $R_j(F_o)$ and $R_l(F_m)$. We give an intuition of this line to line distance in Figure 5.6.

Projection During data association, when we consider matches between crop rows, we project observed plant features $q \in F_o$ from their crop row into the matched crop row $R_l(F_m) \in \mathcal{R}(F_m)$ (see Figure 5.7). This operation translates the original point q to the same position of q along its corresponding row, but projected as if it were in the target row $R_l(F_m)$, i. e., the longitudinal coordinates of the original point q and its projection q' are equal. Since we want to project the point q onto $R_l(F_m)$, we also need to define a lateral coordinate that puts the projected point onto the crop row $R_l(F_m)$. To this end, we compute a reference point $y_q \in R_l(F_m)$ that is close to the longitudinal position of q and use its lateral coordinate for our projected point. If no point $y \in R_l(F_m)$ is close to q, we use the centroid c_l of $R_l(F_m)$ as a reference. Therefore, the projected plant feature $q' = \operatorname{prj}_{R_l(F_m)}(q)$ is defined as follows:

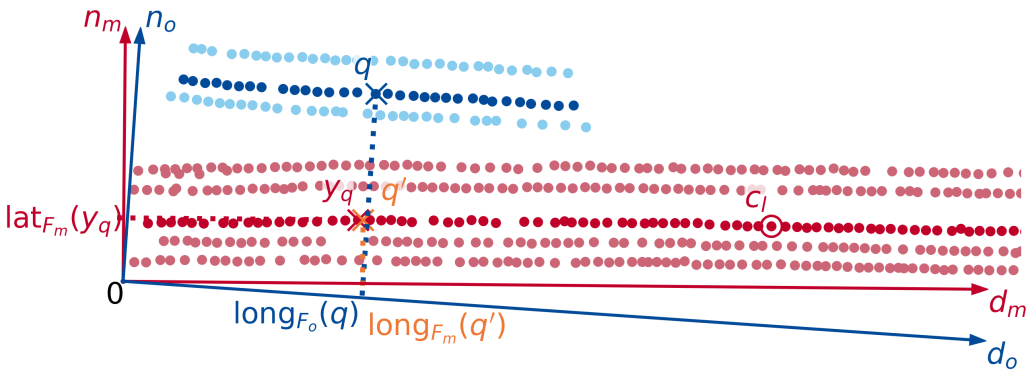

Figure 5.7: This figure demonstrates how a point $q \in F_o$ is projected onto the crop row $R_l(F_m) \in \mathcal{R}(F_m)$. The point q and the corresponding coordinate system are shown in blue. Its projection q' and corresponding longitudinal coordinate in the target coordinate system $\mathcal{R}(F_m)$ are highlighted in orange. The lateral components considered during projection, i. e., the points on $R_l(F_m)$, including y_q, and the centroid c_l are visualized in red.

$$
\begin{aligned}
\text{prj}_{R_l(F_m)}(q) &:= \text{long}_{F_o}(q) \cdot d_m + \text{lat}_{F_m}(y_q) \cdot n_m \\
y_q &:= \begin{cases} y^*(q), & \text{if } \left|\text{long}_{F_o}(q) - \text{long}_{F_m}(y^*(q))\right| < 0.3 \\ c_l, & \text{otherwise} \end{cases} \\
y^*(q) &:= \underset{y \in R_l(F_m)}{\text{argmin}} \left|\text{long}_{F_o}(q) - \text{long}_{F_m}(y)\right|
\end{aligned}
$$

In this section, we defined the row structure called Plant-Rows as well as a reference coordinate system that is split into a longitudinal coordinate along the crop row direction and a lateral coordinate that is orthogonal to the direction of the crop rows. We use these coordinates to define an order on crop rows and plant features as well as a distance measure between crop rows. We also present a technique to project a feature from an observed crop row onto the corresponding mapped crop row. In the following, we use these definitions on Plant-Rows to define the set of valid data association matches $\mathcal{M}^{\text{valid}}$ according to our hard requirements.

5.3.2 Parametrization of Data Association Matches

In this section, we use the definitions of the set of Plant-Rows to define the set of valid data association matches $\mathcal{M}^{\text{valid}}$, i. e., the subset of all possible sets of data association matches $\mathcal{M}$, that satisfy the five hard requirements: *Row-Wise Matches*, *Preserve Inter Row Order*, *Preserve Intra Row Order*, *Consistent Row Matches* and *Locality*. To this end, we first define a two-dimensional parametrization of the set of data association matches. The sets of matches that can be expressed by this parametrization are guaranteed to satisfy

the first three hard requirements. We then restrict the value range of the parametrization to enforce the last two requirements and define our set of valid data association matches $\mathcal{M}^{\text{valid}}$ using this restricted two-dimensional parametrization.

Parameterized Search Space Our definition for the two-dimensional parametrization of sets of plant feature matches is inspired by our one-dimensional parametrization of sets of crop row matches in the *Crop Row data association* presented in Section 4.3.1. For our one-dimensional parametrization we shift the observed crop rows in the lateral direction according to the lateral shift parameter $s_{\text{lat}} \in \mathbb{R}$ to overlay them with the mapped crop rows and efficiently generate geometrically consistent sets of crop row matches. We transfer this idea to sets of plant features as follows: First, analogously to the lateral shift parameter s_{lat} we use a *row-shift* parameter $r \in \mathbb{Z}$, to align the observed crop rows with the mapped crop rows. In contrast to the lateral shift s_{lat} that describes the amount of lateral shift, the row-shift r uses the fact that the crop rows of the set of Plant-Rows are ordered and therefore assigns crop row pairs based on their indices, i. e., the i-th observed crop row is assigned to the $(i + r)$-th mapped crop row. In order to align the plant features of the matched crop row pairs, we introduce a second shift parameter, called *long-shift* parameter $l \in \mathbb{R}$ that shifts the observed plant features in the longitudinal direction. Both parameters together then define the two-dimensional *shift* parameter $s = (r, l) \in \mathbb{Z} \times \mathbb{R}$ of our two-dimensional parametrization of sets of plant feature matches. We illustrate how a shift parameter s overlays a set of observed Plant-Rows onto a set of mapped Plant-Rows in Figure 5.8. We call the operation that shifts an observation $o \in F_o$ according to a given shift parameter $s \in \mathbb{Z} \times \mathbb{R}$ the *projection* $\text{prj}_s(o)$. Analogously to the one-dimensional parametrization, we determine a set of feature matches $M(s) \in \mathcal{M}$ for a shift parameter s by projecting all observed features $o \in F_o$ according to the shift s and computing the Nearest Neighbor data association between the projected observations and the mapped features F_m. We illustrate how the shift projects the observed features onto the mapped features as well as the set of matches represented by that shift in Figure 5.8.

Given the set of observed plant features F_o and the set of mapped plant features F_m, we arrange both sets into ordered sets of Plant-Rows, $\mathcal{R}(F_o)$ with k_o rows and $\mathcal{R}(F_m)$ with k_m rows respectively. Our two-dimensional parametrization is then defined as follows:

$$
\begin{aligned}
s &:= (r, l) \in \mathbb{Z} \times \mathbb{R} && (5.1)\\
\text{prj}_s(o) &:= \text{prj}_{R_{j+r}(F_m)}\left(\left(\text{long}_{F_o}(o) + l, \text{lat}_{F_o}(o)\right)\right) \forall\, o \in F_o(j), \forall\, j \in [1, k_o] && (5.2)\\
m_s(o) &:= \underset{m \in F_m(j+r)}{\text{argmin}} \left\|\text{prj}_s(o) - m\right\| && (5.3)\\
M(s) &:= \{(o, m_s(o)) \mid \forall\, o \in F_o \colon \left\|\text{prj}_s(o) - m_s(o)\right\| < \varepsilon\} && (5.4)
\end{aligned}
$$

Since adjacent crops are usually planted around $0.3\,\text{m}$ apart within the crop rows, we choose half the distance of $\varepsilon := 0.15\,\text{m}$ as the distance threshold for the Nearest Neighbor data association. Of course, this parameter can be adapted for crop fields with different intra-row spacing if necessary.

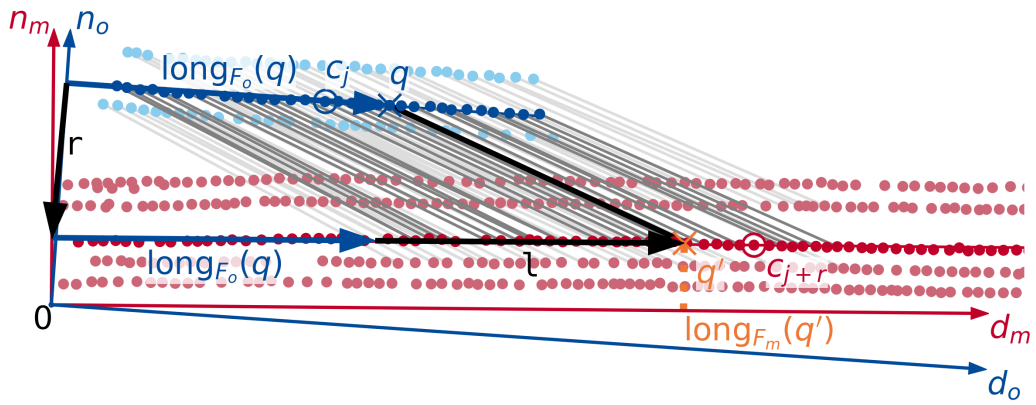

Figure 5.8: This figure illustrates how a shift $s = (r, l)$ (black arrows) defines a matching between the set of observed Plant-Rows $\mathcal{R}(F_o)$ (blue points) and the mapped Plant-Rows $\mathcal{R}(F_m)$ (red points). The set of data association matches $M(s)$ recovered from the shift s is shown by connecting paired plant positions with gray lines. As an example, we highlight how the observed crop row $R_j(F_o)$ is shifted onto the corresponding mapped crop row $R_{j+r}(F_m)$ with darker colors. We illustrate how a point $q \in F_p(j)$ is projected according to the shift s by first shifting q onto the mapped crop row according to the row-shift r and then shifting along the crop row by its longitudinal coordinate $\text{long}_{F_o}(q)$ as well as the long-shift l. The resulting projected point q' and its longitudinal coordinate are visualized in orange.

When computing the matching mapped feature $m_s(o)$ for an observed feature, we only consider $m \in F_m(j + r)$, i. e., features o in the j-th crop row are only mapped on features m that are in the associated $(j + r)$-th crop row. This ensures that the set of matches $M(s)$ only contains *Row-Wise Matches* and thus requirement 1 is satisfied. The second requirement to *preserve inter row order* is satisfied, since the crop rows of both sets are ordered in ascending order and we use the same row-shift r to match observed crop rows with mapped crop rows. Therefore, any neighboring observed row $R_{j+1}(F_o)$ will be matched onto the mapped crop row $R_{(j+1)+r}(F_m)$ that is adjacent to the corresponding mapped crop row $R_{j+r}(F_m)$ by definition. Regarding the third requirement of *preserving intra row order*, we need to confirm that the order of the plant features within corresponding rows is preserved, while it is allowed to match different observed plant features onto the same map feature. During projection, the feature order is preserved since the longitudinal position of each feature is always shifted by the same amount l. Therefore, for any crop row $R_j(F_o)$ a plant feature $o_{i+1} \in F_o(j)$ adjacent to $o_i \in F_o(j)$ within the same crop row remains a direct neighbor after projecting, since the new longitudinal coordinates $\text{long}_{F_o}(o_i) + l$ and $\text{long}_{F_o}(o_{i+1}) + l$ still have the same relative distance $\left|\text{long}_{F_o}(o_i) + l - \left(\text{long}_{F_o}(o_{i+1}) + l\right)\right| = \left|\text{long}_{F_o}(o_i) - \text{long}_{F_o}(o_{i+1})\right|$. Using a Nearest Neighbor data association on the projected observations also maintains the order of plant features. For a better intuition, we give an example in Figure 5.9. Confirming that the order of plant features is preserved for all matches is equivalent to showing that no match can be found that breaks the order as shown in this example. Recalling that we always

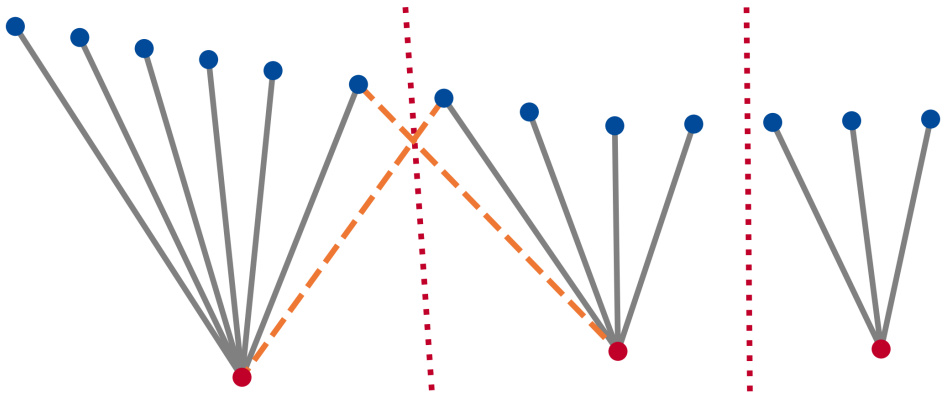

Figure 5.9: This figure illustrates a valid set of data association matches (gray lines) of projected observations (blue points) along a mapped crop row (red points) according to the Nearest Neighbor data association. An example for feature matches that break the order of plant features and would therefore cause the set of matches to be invalid are highlighted with dashed orange lines. The neighborhood around each mapped feature defined by the Nearest Neighbor data association is visualized by dashed red lines that separate adjacent neighborhoods. All observed features inside a neighborhood are closest to the corresponding mapped feature and therefore matched with this feature by the Nearest Neighbor data association (gray lines). Thus, the invalid feature matches (orange lines) cannot occur in the Nearest Neighbor data association, since the observed features would not be matched to the closest mapped feature, i. e., the observed features lie not inside the neighborhood of the mapped feature they would be matched to.

match an observation o to the map feature $m_s(o)$ closest to its projected position $\mathrm{prj}_s(o)$, we can immediately conclude that the situation shown in the example can not occur in our algorithm. In summary, our technique to represent matches using shifts $s \in S = \mathbb{Z} \times \mathbb{R}$ always yields sets of data association matches $M(s)$ that satisfy the first three requirements. Thus, we use this parametrization to define our search space during data association as a subset of the set of all possible matches $M(S) \subset \mathcal{M}$. In the following, we also call the parametrization S search space, since it implicitly defines the data association matches $M(S)$.

Valid Search Space Using the set of shifts S as our search space during data association, we know that the resulting sets of data association matches $M(S)$ satisfy the first three requirements. However, there can be instances, where a shift $s \in S$ results in an empty set of data association matches $M(s) = \emptyset$, or cases, where multiple shifts $\{s_1, \ldots, s_n\} \in S$ result in the same set of data association matches $M(s_1) = \cdots = M(s_n)$. Therefore, we constrain the row-shift and long-shift parameters of our search space S, so that no duplicate sets of matches and no empty matches are considered during data association. We further constrain both parameters to ensure that the last two requirements of *Consistent Row Matches* and *Locality* are satisfied.

The same set of data association matches is only represented by shifts $s_i, s_j \in S$ that

have the same row-shift $r_i = r_j$ and similar long-shifts $|l_i - l_j| < \varepsilon$. In these cases, the observed crop rows are matched onto the same mapped crop rows and the difference in the long-shift is not large enough, so that the closest mapped feature $m_s(o)$ is still the same for all $o \in F_o$. To avoid shifts that are too similar, we discretize the space of considered long-shifts L^{valid} by a given resolution l^{res}. This ensures, that the distance between consecutive long-shifts is always equal to l^{res}. If the resolution l^{res} is too large, some sets of data association matches might get excluded from the set of valid matches. Therefore, we choose a sufficiently small long-shift resolution $l^{\text{res}} = 0.01\,\text{m}$ for our data association approach. The set of discretized valid long-shifts L^{valid} is then defined as:

$$L^{\text{valid}} := \{l_i \in \mathbb{R}, i \in \mathbb{Z} \mid l_0 = 0, l_{i+1} = l_i + l^{\text{res}}\}$$

The *Locality* requirement, as explained in Section 2.2.2, is based on the idea that the corrected pose estimate should remain reasonably close to the current pose estimate. Therefore, only a local set of valid data association matches $M \in \mathcal{M}^{\text{local}}$ that infer a small amount of pose correction T_M onto the current pose estimate should be considered during data association. Since the observed and mapped crop rows are approximately parallel, the inferred amount of angular pose correction θ is similar for all data association matches $M \in \mathcal{M}$ that preserve the crop row structure. It thus suffices to constrain the translational component t of the inferred pose correction T_M. Since our parametrization of sets of data association matches as shifts $s \in S$ by design models the inferred translational pose correction t, we define the set of local data association matches $\mathcal{M}^{\text{local}}$ in our parameterized space S as follows: In order to determine the shifts $s \in S$ that only infer a small amount of translational pose correction, we first determine the shift s_{current} that infers the smallest possible amount of pose correction relative to the current pose estimate. We then use this shift s_{current} as a reference to define the set of *local shifts* $S^{\text{local}} := R^{\text{local}} \times L^{\text{local}}$ that represent the local set of data association matches $\mathcal{M}^{\text{local}} := M\left(S^{\text{local}}\right)$.

Given the current pose estimate of the vehicle, we determine the corresponding reference shift $s_{\text{current}} = (r_{\text{current}}, l_{\text{current}}) \in S$ as follows: The long-shift l of the shift s determines how far the plant positions should be shifted in the longitudinal direction relative to the current pose estimate. Therefore, minimizing the longitudinal component of the inferred pose correction is equal to $l_{\text{current}} := 0$. Finding the value that minimizes the inferred lateral pose correction, i. e., determining the row-shift component r_{current} is more complex. In contrast to the long-shift, the row-shift is not defined relative to the pose estimate. Instead, it is a crop row index that depends on the observed and mapped crop rows. To find the row-shift r_{current} that minimizes the introduced lateral shift, we find the pair of observed and mapped crop rows $(R_{j^*}(F_o), R_{l^*}(F_m))$, where the matched rows are closest to each other according to the crop row distance Δ. This then defines the row-shift $r_{\text{current}} := l^* - j^*$ with the smallest inferred lateral translation. In summary, the current shift s_{current} that infers minimal lateral and longitudinal correction and therefore best represents the current pose estimate is defined by $s_{\text{current}} := (l^* - j^*, 0)$.

To apply the locality principle, we need to determine the expected pose estimate accuracy. From our previous experiments we know that during navigation of an entire crop

field, the largest pose errors occur after the turning maneuver at the headlands before traversing the next set of crop rows. Here, the vehicle usually looses track of the features as it leaves the field. Therefore the pose cannot be corrected and errors accumulate in the estimate. The error in the pose estimate at this point usually spans around two crop rows to the left and right and around $3\,\mathrm{m}$ longitudinally – assuming the accuracy of a standard GPS without corrections from the *End of the Field detection*. Therefore, we define the set of local shifts $S^{\text{local}} = R^{\text{local}} \times L^{\text{local}}$ as follows:

$$\begin{aligned} R^{\text{local}} &:= [r_{\text{current}} - 2, r_{\text{current}} + 2] \\ L^{\text{local}} &:= [l_{\text{current}} - 3, l_{\text{current}} + 3] \end{aligned}$$

Note, that we choose these thresholds conservatively, i. e., we will most likely not exclude the correct shift parameter. To ensure that no empty matches are represented in our search space, we constrain the row-shift parameter, so that at least one observed crop row is matched with a mapped crop row. We therefore compute the smallest and largest possible row-shifts $r^{\min}$ and $r^{\max}$ so that at least one observed crop row is matched. These are exactly the row-shifts, where the last observed row k_o is matched with the first mapped row and vice versa. This determines the range of valid row-shift values $r^{\min} := 1 - k_o$ and $r^{\max} := k_m - 1$. According to the *Consistent Row Matches* requirement, only row-shifts that define crop row pairs with similar relative distances are valid.

Therefore, we check for every row-shift $r \in \left[r^{\min}, r^{\max}\right]$, whether the crop row pairs defined by that row-shift are *geometrically consistent* as defined in Section 4.3.1 in Eq. (4.5). Recall, that we used a continuous lateral shift parameter $s_{\text{lat}} \in \mathbb{R}$ to shift the observed crop rows in the lateral direction and determine a set of crop row data association matches $M_R(s_{\text{lat}}) \in \mathcal{P}\left(\{R_j(F_o)\}_{j\in[1,k_o]} \times \{R_l(F_m)\}_{l\in[1,k_m]}\right)$ according to the Nearest Neighbor strategy. We also proofed that the implication of Eq. (4.6) holds: If shifting the observed crop rows by s_{lat} results in a set of crop row matches $M_R(s_{\text{lat}})$ with a pairwise line to line distance below $\frac{1}{2}\varepsilon$ for a given threshold ε, we can conclude that the set of crop row matches is geometrically consistent and $\mathrm{gc}(M_R(s_{\text{lat}}))$ holds.

In this chapter, we use the discrete row-shift parameter r to determine the matched crop row pairs $M_R(r) \in \mathcal{P}\left(\{R_j(F_o)\}_{j\in[1,k_o]} \times \{R_l(F_m)\}_{l\in[1,k_m]}\right)$. To apply our findings from the previous chapter, we therefore determine the continuous lateral shift parameter $s_{\text{lat}}(r)$ corresponding to each row-shift parameter $r \in \left[r^{\min}, r^{\max}\right]$. We then use Eq. (4.6) to determine whether the set of matched crop rows $M_R(r)$ according to our row-shift parameter r is geometrically consistent, i. e., whether $\mathrm{gc}(M_R(r))$ holds, by shifting the observed crop rows by $s_{\text{lat}}(r)$ and checking that the shifted observed crop rows have a lateral distance of below $\frac{1}{2}\varepsilon$ to their corresponding matched crop row. By defining the lateral shift $s_{\text{lat}}(r)$ inferred by a given row-shift parameter r as the smallest signed distance of all matched crop rows in $M_R(r)$, we determine geometric consistency of $M_R(r)$ as follows:

$$
\begin{aligned}
M_R(r) &:= \{(R_j(F_o), R_l(F_m)) \mid l = j + r\} \\
(R_{j^*}(F_o), R_{j^*+r}(F_m)) &:= \underset{(R_j(F_o), R_{j+r}(F_m)) \in M_R(r)}{\operatorname{argmin}} \Delta(R_j(F_o), R_{j+r}(F_m)) \\
o_{\text{ref}} &:= c_{j^*} \\
s_{\text{lat}}(r) &:= \delta(R_{j^*}(F_o), R_{j^*+r}(F_m)) \\
\operatorname{gc}(M_R(r)) &\Leftarrow \forall (R_j(F_o), R_{j+r}(F_m)) \in M_R(r) : \\
&\quad \Delta^{o_{\text{ref}}}(R_j(F_o), R_{j+r}(F_m)(s_{\text{lat}}(r))) < \frac{1}{2}\varepsilon
\end{aligned} \tag{5.5}
$$

Note that Δ and δ are the line to line distances defined in this chapter on our Plant Row structure. This distance measure is used to determine the reference point o_{ref}, which is the centroid c_{j^*} of the reference observed crop row $R_{j^*}(F_o)$ as well as the lateral shift parameter $s_{\text{lat}}(r)$. For determining whether $M_R(r)$ is geometrically consistent, we use the line to line distance measure $\Delta^{o_{\text{ref}}}$ as defined in Eq. (4.2) with reference point o_{ref}. Note that we shift the mapped lines $R_{j+r}(F_m)$ by s_{lat} resulting in $R_{j+r}(F_m)(s_{\text{lat}}(r))$ instead of the observed lines $R_j(F_o)$ for convenience. We only keep row-shifts r that yield crop row matches $M_R(r)$ that are geometrically consistent according to the condition on the right of Eq. (5.5). Using this testing procedure, we define the set of *valid row-shifts* R^{valid} as follows:

$$
R^{\text{valid}} := (r \in [1 - k_o, k_m - 1] \mid \operatorname{gc}(M_R(r)))
$$

Combining all search space constraints, we define the set of valid shifts S^{valid} as well as the resulting set of valid matches $\mathcal{M}^{\text{valid}}$ and state our data association approach more precisely as follows:

$$
\begin{aligned}
S^{\text{valid}} &:= S^{\text{local}} \cap \left(R^{\text{valid}} \times L^{\text{valid}}\right) \\
\mathcal{M}^{\text{valid}} &:= M\left(S^{\text{valid}}\right) \\
M^* &= \underset{M \in \mathcal{M}^{\text{valid}}}{\operatorname{argmax}} t(M) = \underset{\{M(s) \mid s \in S^{\text{valid}}\}}{\operatorname{argmax}} t(M(s)) = M(s^*) \\
s^* &:= \underset{s \in S^{\text{valid}}}{\operatorname{argmax}} t(M(s))
\end{aligned}
$$

The set of valid shifts S^{valid} contains all shifts $s = (r, l)$ that have valid row- and long-shift values $r \in R^{\text{valid}}$ and $l \in L^{\text{valid}}$ and only introduce a reasonable amount of pose correction $s \in S^{\text{local}}$. The resulting set of valid data association matches $\mathcal{M}^{\text{valid}} = M\left(S^{\text{valid}}\right)$ then only contains sets of data association matches $M \in \mathcal{M}^{\text{valid}}$ that fulfill all five hard requirements. We leverage our parametrization of valid data association matches using shifts $s \in S^{\text{valid}}$ to find the best data association M^* according to the Target function t. This is equivalent to finding the best shift s^* according to the Target function t with

$M^* = M(s^*)$. Next, we explain how we design the Target Function t for our row-based data association approach to achieve a robust and accurate data association on dense and almost uniformly distributed features.

5.3.3 Counting Matches on Challenging Feature Distributions

In this section, we explain how we design our Target Function to measure the overlap between observed and mapped features for a set of data association matches. In our setting with dense and almost uniformly distributed features, the main challenge is to derive a Target Function t that can efficiently quantify a large amount of possible feature matches as well as robustly discriminate highly ambiguous situations. To this end, we introduced four soft criteria to incorporate in our Target Function: *Positive Information*, *Negative Information*, *Continuity* and *No Bias*. Recalling that we define the set of valid matches $\mathcal{M}^{\text{valid}}$ using our parametrization $\mathcal{M}^{\text{valid}} = M\left(S^{\text{valid}}\right)$, the search space during data association is the set of valid shifts S^{valid}. Therefore, we define our Target Function t as a function that operates on shifts $s \in S^{\text{valid}}$ and considers the corresponding set of data association matches $M(s) \in \mathcal{M}$ to determine a value $t(M(s)) \in \mathbb{R}$. We call this value $t(M(s)) \in \mathbb{R}$ a *score* for the corresponding shift s and our corresponding Target Function that is designed to operate on shifts a *Score Function* $\mathrm{sf}\colon S \to \mathbb{R}$, $\mathrm{sf}(s) := t(M(s))$. Accordingly, we then define the solution to our data association problem as finding the best shift defined by our Score Function sf:

$$s^* = \underset{s \in S^{\text{valid}}}{\operatorname{argmax}}\, t(M(s)) = \underset{s \in S^{\text{valid}}}{\operatorname{argmax}}\, \mathrm{sf}(s)$$

In the following, we first present our vanilla Score Function, that uses a continuous measurement to quantify the overlap between observed and mapped features, thereby satisfying criterion 1 and 3. However, this vanilla Score Function is biased towards matching as many crop rows as possible. Therefore, we also introduce a regularization technique to counter the bias of our vanilla Score Function. Since the vanilla Score Function does not consider *Negative Information*, we then present three variations of the vanilla Score Function that consecutively improve on criteria 2 and 4. We also discuss different properties of the presented Score Functions to give an intuition about the expected behavior for each Score Function and discuss properties of a Score Function that is well suited for a localization or mapping application on challenging feature distributions.

Cross-Correlation to Continuously Count Matches In contrast to many other data association approaches that count the number of matched observed features, our Score Function should compute a continuous measurement. The main idea for our Score Function, inspired by Weiss et al. [1994], is that we use the cross-correlation to design a continuous equivalent to the discrete match counting strategy and therefore satisfy the *Positive Information* and *Continuity* requirements. The continuous cross-correlation cc and the

discrete cross-correlation $\mathrm{cc}^{\text{disc}}$ measure the similarity of two continuous or discrete functions over a displacement τ or z respectively:

$$\begin{aligned} \mathrm{cc}\colon \mathbb{R} &\to \mathbb{R} \\ \tau &\mapsto \int_{\mathbb{R}} f(t) \cdot g(t+\tau)\, dt && \forall f, g\colon \mathbb{R} \to \mathbb{R} \\ \mathrm{cc}^{\text{disc}}\colon \mathbb{Z} &\to \mathbb{R} \\ z &\mapsto \sum_{i \in \mathbb{Z}} f(i) \cdot g(i+z) && \forall f, g\colon \mathbb{Z} \to \mathbb{R} \end{aligned}$$

Our key idea is to transfer the properties of the cross-correlation to our Score Function. Given a shift $s = (r, l) \in S$ the row-shift r determines which rows to match and the long-shift l defines the longitudinal displacement. We therefore compute the continuous cross-correlation between each pair of crop rows and use the long-shift l as the displacement τ.

For a better intuition, we first show how the discrete version of the cross-correlation can be used to count the number of data association matches for a given displacement z and a distance threshold ε that determines whether two features match. Afterwards, we explain how this approach can be transferred to the continuous case to count the number of matches in a continuous matter without defining a hard distance threshold ε.

We obtain a discrete match counting strategy with distance threshold ε using the discrete cross-correlation $\mathrm{cc}^{\text{disc}}$ as follows: Given a set of observed Plant-Rows $\mathcal{R}(F_o)$ and mapped Plant-Rows $\mathcal{R}(F_m)$, we compute the cross-correlation between pairs of crop rows $R_j(F_o) \in \mathcal{R}(F_o)$ and $R_{j+r}(F_m) \in \mathcal{R}(F_m)$, where $j + r$ is the index of the mapped crop row that is paired with the j-th observed crop row according to the row-shift parameter r. The overall number of matched observations N_o for a given displacement $z \in \mathbb{Z}$ is then given by the sum over the pairwise crop rows:

$$N_o(z) \;:=\; \sum_{j \in [1,\ldots,k_o]} \mathrm{cc}^{\text{disc}}_{j,j+r}(z)$$

To compute the cross-correlation between a pair of crop rows, we represent the feature distribution along each row $R_j(F_o)$ and $R_{j+r}(F_m)$ as a discrete function $f^{\text{disc}}_j, g^{\text{disc}}_{j+r}\colon \mathbb{Z} \to \mathbb{R}$. Our key idea is to leverage the coordinate system defined by the corresponding set of Plant-Rows. The longitudinal direction d_o and d_m respectively define the domain of the discrete functions $f^{\text{disc}}_j, g^{\text{disc}}_{j+r}$. We discretize this domain using a threshold ε to obtain a discrete set of bins $\{t_i\}_{i \in \mathbb{Z}} = T_\varepsilon$. All observed features $o \in F_o(j)$ that fall into the same bin t_i as a mapped feature $m \in F_m(j + r)$ are per definition within the distance threshold ε to any mapped feature in t_i and therefore count as matched observations. We thus define the values of the discrete observed function f^{disc}_j for each bin t_i as the number of observed features $o \in F_o(j)$ that fall into t_i. For the values of the discrete mapped function g^{disc}_{j+r} it

suffices to check for each bin t_i whether at least one mapped feature falls into each bin t_i. If a mapped feature $m \in F_m\,(j+r)$ falls into t_i, we set the value of $g^{\text{disc}}_{j+r}\,(t_i)$ to 1 and to 0 otherwise. The product over both functions at each bin t_i is then equal to the number of matched observed features at each bin t_i. We give an example for the discrete functions of a pair of crop rows as well as the resulting product over both functions on the left of Figure 5.10. Mathematically, the observed and mapped discrete functions are defined as follows:

$$
\begin{aligned}
t_i \in T_\varepsilon &:= \{t_i \in \mathbb{R},\, i \in \mathbb{Z} \mid t_0 = 0, t_{i+1} = t_i + \varepsilon\} \\
f^{\text{disc}}_j\,(t_i) &:= \left|\left\{q \in F_o\,(j) \mid t_i \leq \log_{F_o}(q) < t_{i+1}\right\}\right| \\
g^{\text{disc}}_{j+r}\,(t_i) &:= \begin{cases} 1, & \text{if } \exists q \in F_m\,(j+r) : t_i \leq \log_{F_m}(q) < t_{i+1} \\ 0, & \text{otherwise} \end{cases}
\end{aligned}
$$

Given a displacement $z \in \mathbb{Z}$ the corresponding discretized cross-correlation on a pair of crop rows $R_j\,(F_o)$, $R_{j+r}\,(F_m)$ is then defined as:

$$
\text{cc}^{\text{disc}}_{j,j+r}\,(z) := \sum_{i \in \mathbb{Z}} f^{\text{disc}}_j\,(t_i) \cdot g^{\text{disc}}_{j+r}\,(t_{i+z})
$$

By definition of f^{disc}_j and g^{disc}_{j+r}, the product over both function values is equal to the number of observed features at bin t_i that are matched to a mapped feature at bin t_{i+z} using a displacement of $z \in \mathbb{Z}$. Since the discretized cross-correlation $\text{cc}^{\text{disc}}_{j,j+r}\,(z)$ computes the sum over this product for a given displacement $z \in \mathbb{Z}$ over all bins t_i, the resulting value $\text{cc}^{\text{disc}}_{j,j+r}$ is equal to the number of matched observations for a pair of crop rows using displacement z (see bottom left of Figure 5.10).

To apply the idea of continuously counting the number of matched features using the continuous cross-correlation, we only need to replace our discrete function representation of crop rows with a continuous function representation (see Figure 5.10, right). While we retain the idea from the discrete case to use the longitudinal coordinates of the feature positions, we adapt how we encode the feature information along the crop rows to obtain a continuous function. For a set of Plant-Rows $\mathcal{R}\,(F_p)$ of a set of plant features F_p with k crop rows, we define a continuous function f^j_p for the j-th crop row $R_j\,(F_p)$ with $j \in [1, k]$ as follows:

$$
\begin{aligned}
f^j_p\colon\ & \mathbb{R} \to \mathbb{R}_{\geq 0} \\
& t \mapsto (\mathbb{1}_j * \mathcal{N}\,(0, \sigma))\,(t) \\
\mathbb{1}_j\,(t) := & \begin{cases} 1, & \text{if } \exists\, q \in F_p\,(j) :\ \log_{F_p}(q) = t \\ 0, & \text{otherwise} \end{cases}
\end{aligned}
$$

Analogue to the discrete case, we encode the presence of a feature at position t with a non-zero value. Initially, this value is equal to one as defined by the $\mathbb{1}$-function. To

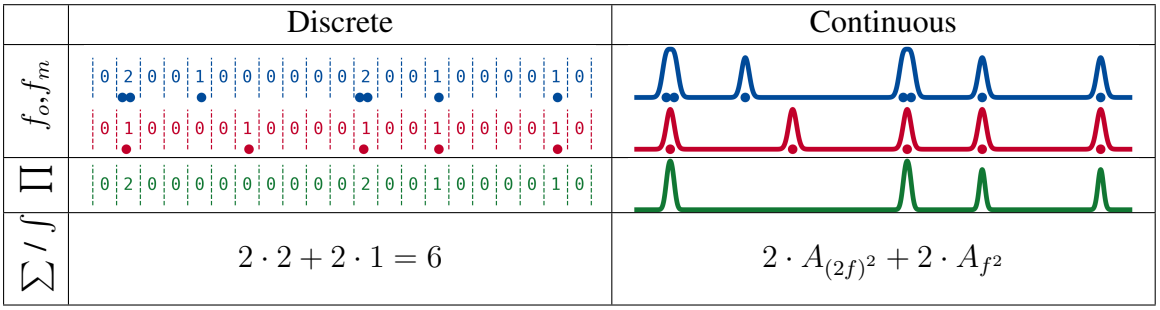

Figure 5.10: This figure shows an example for the discrete and continuous crop row functions and cross-correlation. Observed and mapped features with corresponding functions f_o, f_m are visualized using blue and red respectively. The bins of the discretized function domain along the crop row are illustrated with dashed lines and the corresponding function values are printed in each bin between the dashed lines. The observed and mapped continuous functions are visualized as blue and red lines. The product function of f_o, f_m is shown in green in the second row. In the bottom row, the discrete and continuous cross-correlation are computed.

account for inaccuracies in the feature position, we apply a Gaussian kernel $\mathcal{N}(0, \sigma)$ with a predefined sigma σ on the values of the $\mathbb{1}$-function to obtain a continuous function f_p^j. This σ parameter is the analogon of the hard distance threshold ε from the discrete counting strategy as it serves a similar purpose: It enables the algorithm to robustly handle inaccurate information about the feature position, i. e., sensor noise or slightly inaccurate detection results. We choose a value of $\sigma = 0.1$, which corresponds to the accuracy of the employed plant detection method.

Let $\mathcal{R}(F_o)$ be the set of k_o observed Plant-Rows and $\mathcal{R}(F_m)$ be the set of k_m mapped Plant-Rows, where $j + r \in [1, k_m]$ is the index of the mapped crop row $R_{j+r}(F_m)$ that is matched with the j-th observed crop row $R_j(F_o)$ for a $j \in [1, k_o]$. Using the continuous function f_o^j for the j-th observed crop row and the continuous function f_m^{j+r} for the $(j + r)$-th mapped crop row, we define the continuous cross-correlation between both crop rows for a displacement $\tau \in \mathbb{R}$ as follows:

$$\mathrm{cc}_{j,j+r}(\tau) \quad := \quad \int_{\mathbb{R}} f_o^j(t) \cdot f_m^{j+r}(t + \tau)\, dt$$

By design of the continuous functions this continuous cross-correlation measures the number of matched observations as a continuous value for the same reason as in the discrete case: When integrating over the product of the function values $f_o^j(t)$ and $f_m^{j+r}(t + \tau)$, the integral value only increases if both function values are non-zero. Again, this is only true for locations, where an observed and mapped feature are close to each other (see Figure 5.10, right).

By computing the sum of the continuous cross-correlation for a displacement $\tau \in \mathbb{R}$ over all crop row pairs we obtain our continuous match counting method and define our vanilla Score Function sf as follows:

$$\begin{aligned} \mathrm{sf}: \quad & S \to \mathbb{R} \\ & (r, l) \mapsto \sum_{j=1}^{k_o} \mathrm{sf}_{j,j+r}\,(l) \end{aligned}$$

$$\mathrm{sf}_{j,j+r}\,(l) := \begin{cases} \mathrm{cc}_{j,j+r}\,(l)\,, & \text{if } j + r \in [1, k_m] \\ 0, & \text{otherwise} \end{cases}$$

The score $\mathrm{sf}_{j,j+r}$ for the j-th observed crop row is equal to the value of the cross-correlation between the corresponding continuous function pair, if the mapped crop row with index $j + r$ exists. If a mapped crop row with index $j + r$ does not exist, the j-th observed crop row is unmatched. This means that there are no matched observed features for this crop row and we therefore assign a score of zero.

This works well for a localization application, where the full map is already known, and therefore all observed crop rows should be paired with a mapped crop row. However, in a mapping application, where the map is only partially known, row-shifts that do not match all observed rows also need to be considered during data association. This is crucial since parts of the observed information might not yet be part of the mapped features and therefore some observed crop rows have to remain unmatched to find the correct data association. However, since we do not know the part of the map, that we have not yet observed, we cannot easily compute a score for the unmatched observed crop row. Since our Score Function counts the number of matched observations, we assign a value of zero in these cases, because no observed features are matched for this crop row. We give an example for the behavior of this vanilla Score Function on real-world data in the top, left corner of Figure 5.11. This figure shows the score distribution of our vanilla Score Function over different shifts $s \in S$ for $k_o = 5$ observed crop rows and $k_m = 10$ mapped crop rows. For a better overview of the behavior of the Score Function we show the complete results for all row-shifts, where at least one observed row is matched with a mapped row, ignoring the *Locality* and *Consistent Row Matches* requirement for the row-shifts. Therefore, the smallest row-shift $r^{\min} = -4$ corresponds to pairing the last observed row with index $k_o = 5$ with the mapped crop row at index 1 and the largest row-shift $r^{\max} = 9$ corresponds to matching the first observed row with the last mapped crop row with index $k_m = 10$. Row-shifts towards the center, i. e., row-shifts -3, -2, -1 and 8, 7, 6 consecutively match one more observed crop row until the row-shifts at the center, i. e., row-shifts 0, 1, 2, 3, 4 and 5 match all observed crop rows. This distribution clearly shows, that our vanilla Score Function is biased towards matching as many observed crop rows as possible, since row-shifts in the center, where more observed crop rows are matched, receive larger values. This behavior is caused by our choice of assigning a score of zero to unmatched crop rows, which is the worst possible score a crop row can receive from our vanilla Score Function. As stated before, in form of the *No Bias* criterion, mitigating such a bias is crucial to achieve robust data association results, especially for the mapping use case. Therefore, in the next paragraph, we introduce a regularization technique to counter the bias introduced by unmatched crop rows.

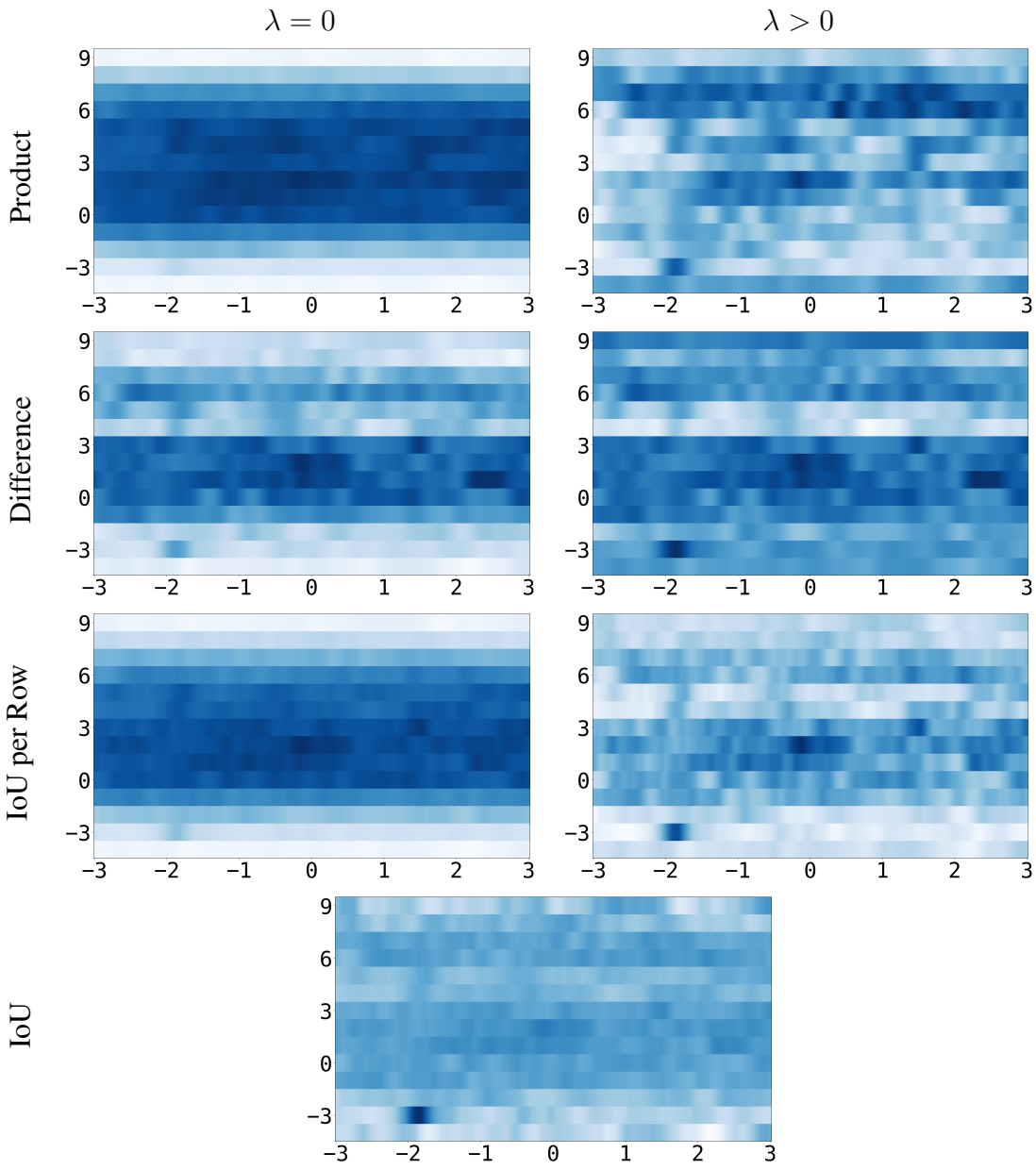

Figure 5.11: This figure shows the score distribution for all Score Function variants for $k_o = 5$ observed crop rows and $k_m = 10$ mapped crop rows. For each Score Function, the score distribution over all shifts $s = (r, l) \in \left[r^{\min}, r^{\max}\right] \times \left(L^{\text{valid}} \cap L^{\text{local}}\right)$ is arranged in a table, where the rows of the table define the row-shift values r, and the columns the long-shift values l. Therefore, each cell of the table visualizes the score for a certain shift s according to a blue color scale. Darker blues correspond to higher scores and lighter colors to lower scores.

Regularization Technique to Handle Unmatched Crop Rows To counter the bias of our vanilla Score Function towards matching as many observed crop rows as possible, we need to define a better score for unmatched crop rows. However, defining such a score without knowing the true feature distribution of the observed crop row is difficult. Intuitively, this unmatched crop row score depends on how much we expect a crop row to remain unmatched. For example, in the localization use case we would not expect to observe any crop row that is not already part of the map. Here, assigning an unmatched score of zero would be correct, i. e., an unmatched crop row should receive the worst possible score. In contrast, during a mapping application we expect that a certain amount of observed data is not yet part of the map since we do not have the full map and are exploring the environment to obtain more information that can then be added to the partial map. The score for an unmatched crop row therefore needs to incorporate this prior information about how many crop rows are expected to remain unmatched. For this purpose, we introduce a *trade-off* parameter $\lambda \in [0, 1] \subset \mathbb{R}$ with the goal to define an unmatched score $\mathrm{unm}_\lambda(j)$ for the j-th observed crop row, where a value of $\lambda = 0$ corresponds to a score, where no or few unmatched crop rows are expected and $\lambda = 1$ corresponds to all, or at least many, crop rows remaining unmatched. The main idea of our regularization technique is that a representative unmatched score $\mathrm{unm}_\lambda(j)$ for an observed crop row $R_j(F_o) \in \mathcal{R}(F_o)$ can be found by approximating the best and worst possible score for this crop row and interpolating between these two extreme values using the trade-off parameter λ. Since a Score Function is a measure of how well the observed data is explained by the matched mapped data, we can approximate the worst and best possible score by creating two extreme situations. For the worst possible case, the observed data is not at all explained by the matched mapped data. Therefore, we can approximate such a situation by not matching any of the observed features, i. e., matching the observed crop row with an entirely empty row. Regarding the best possible case, which means that the observed data is fully explained by the matched mapped data, we simply match the observed crop row with itself, i. e., all observed features are perfectly matched. The best and worst possible scores are then defined by the score of the j-th crop row when matched with an empty crop row and with itself:

$$\begin{aligned} \mathrm{sf}_{j,0} &:= \mathrm{cc}(0), \text{ for } f = f_o^j \text{ and } g = \mathbf{0} \\ \mathrm{sf}_{j,j} &:= \mathrm{cc}(0), \text{ for } f = g = f_o^j \end{aligned}$$

The worst possible score $\mathrm{sf}_{j,0}$ matches the j-th crop row with an empty row. Therefore the corresponding functions used in the cross-correlation are $f = f_o^j$ representing the observed features of the j-th crop row and $g = \mathbf{0}$ the function that is equal to zero everywhere for the empty crop row. Using these approximations of the best and worst possible score, we can now define our unmatched crop row score $\mathrm{unm}_\lambda(j)$ for an observed crop row $R_j(F_o)$ and a trade-off parameter λ as the linear interpolation between both extreme values as follows:

$$\mathrm{unm}_\lambda(j) := \lambda \cdot \mathrm{sf}_{j,j} + (1 - \lambda) \cdot \mathrm{sf}_{j,0}$$

Incorporating this unmatched crop row score into our Score Function then defines a regularized version of the vanilla Score Function:

$$
\begin{aligned}
\mathrm{sf}\colon \quad & S \to \mathbb{R} \\
& (r,l) \mapsto \sum_{j=1}^{k_o} \mathrm{sf}_{j,j+r}\left(l\right) \\
\mathrm{sf}_{j,j+r}\left(l\right) := \quad & \begin{cases} \mathrm{cc}_{j,j+r}\left(l\right), & \text{if } j+r \in [1,k_m] \\ \mathrm{unm}_{\lambda}\left(j\right), & \text{otherwise} \end{cases} \\
= \quad & \begin{cases} \mathrm{cc}_{j,j+r}\left(l\right), & \text{if } j+r \in [1,k_m] \\ \lambda \cdot \mathrm{sf}_{j,j} + (1-\lambda) \cdot \mathrm{sf}_{j,0}, & \text{otherwise} \end{cases} \\
\mathrm{sf}_{j,0} := \quad & \mathrm{cc}\left(0\right), \text{ for } f = f_o^j \text{ and } g = \mathbf{0} \\
\mathrm{sf}_{j,j} := \quad & \mathrm{cc}\left(0\right), \text{ for } f = g = f_o^j
\end{aligned}
$$

We obtain the vanilla Score Function from the regularized Score Function by applying $\lambda = 0$ as the trade-off parameter. For $\lambda = 0$ the unmatched score is equal to $\mathrm{unm}_{\lambda}\left(j\right) = \mathrm{sf}_{j,0} = 0$ for all crop rows $R_j\left(F_o\right) \in \mathcal{R}\left(F_o\right)$, since the product over all function values of f_o^j with $\mathbf{0}$ is equal to zero. To demonstrate how our regularization technique influences the Score Function distribution, we also show the behavior of the regularized Score Function with a trade-off parameter of $\lambda = 0.75$ on the top right of Figure 5.11. The positive effect on the score distribution is evident, since the scores are more evenly distributed over all row-shifts. A bias towards matching all crop rows, i. e., towards the center row-shifts, as observed for the vanilla Score Function without regularization on the left, is not visible anymore.

So far we created a regularized Score Function that satisfies the *Positive Information* and *Continuity* criteria as well as the *No Bias* criterion. However, the *Negative Information* criterion, i. e., that unmatched mapped features should be minimized, is not satisfied. We explain, why this is the case using the examples given in the upper rows of Figure 5.12 until the row labeled prod. In this figure we show two different situations for each criterion to explain if and why the corresponding criterion is satisfied. The first situation for the *Positive Information* criterion, labeled Matched, shows two matched observed features. The second situation, labeled Unmatched, has one matched and one unmatched observation. According to the *Positive Information* criterion the Matched situation should therefore receive a higher score than the Unmatched situation, since more observed features are matched. As visualized by the product function in the row labeled prod, our vanilla Score Function correctly assigns a higher score to the Matched situation and therefore satisfies the *Positive Information* criterion. Regarding the *Negative Information* criterion, we again have a Matched and an Unmatched example situation. Here, the Matched situation shows one matched observation and no unmatched mapped feature. The Unmatched situation illustrates the same matched observed feature but adds an unmatched mapped feature. According to the *Negative Information* criterion, the Matched

situation should receive a higher score than the Unmatched situation, since the Matched situation has no unmatched mapped feature while the Unmatched situation has one unmatched feature. However, since the vanilla Score Function uses the product over the continuous functions to compute the score, it assigns the same value to both situations. This is caused by the fact that the product over two values is zero as soon as one of the values is zero. Therefore, the vanilla Score Function cannot differentiate between situations with and without unmatched mapped features, since the absence of an observed feature causes the product to be zero, irrespective of the presence of a mapped feature. In the following, we therefore present different variations of our vanilla Score Function that – in addition to the other three criteria – also satisfy the *Negative Information* constraint.

Variations of the Score Function From our analysis we learned that the product as the *inner* function of the cross-correlation prevents our Score Function from recognizing unmatched mapped features. We thus propose to replace this inner function of the cross-correlation with different, better suited, functions $\gamma\colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$. Using this inner function γ, we define a new cross-correlation inspired measurement cc^{γ} as follows:

$$\begin{aligned} &\mathrm{cc}^{\gamma}\colon \mathbb{R} \to \mathbb{R} \\ &\qquad \tau \mapsto \int_{\mathbb{R}} \gamma\left(f\left(t\right), g\left(t+\tau\right)\right) dt \qquad\qquad \forall\, f, g\colon \mathbb{R} \to \mathbb{R} \end{aligned}$$

For $\gamma = \mathrm{prod}$, i. e., $\gamma\left(x, y\right) = \mathrm{prod}\left(x, y\right) := x \cdot y$, $\forall\, x, y \in \mathbb{R}$, we obtain the cross-correlation $\mathrm{cc}^{\gamma} = \mathrm{cc}^{\mathrm{prod}} = \mathrm{cc}$. Therefore, to distinguish between different variations of our Score Function, we call the vanilla Score Function based on the cross-correlation as defined above the *Product Score Function* $\mathrm{sf}^{\mathrm{prod}}$.

To obtain a Score Function that also considers unmatched features and therefore incorporates *Negative Information*, we define a second Score Function that uses the negative squared difference as the inner function, called *Difference Score Function* $\mathrm{sf}^{\mathrm{diff}}$ as follows:

$$\begin{aligned} &\mathrm{cc}^{\mathrm{diff}}\colon \mathbb{R} \to \mathbb{R} \\ &\qquad \tau \mapsto \int_{\mathbb{R}} -\left|f\left(t\right) - g\left(t+\tau\right)\right|^2 dt \qquad\qquad \forall\, f, g\colon \mathbb{R} \to \mathbb{R} \end{aligned}$$

The corresponding Difference Score Function $\mathrm{sf}^{\mathrm{diff}}$ is then defined analogue to the Product Score Function, where $\mathrm{cc} = \mathrm{cc}^{\mathrm{prod}}$ is replaced with $\mathrm{cc}^{\mathrm{diff}}$ as follows:

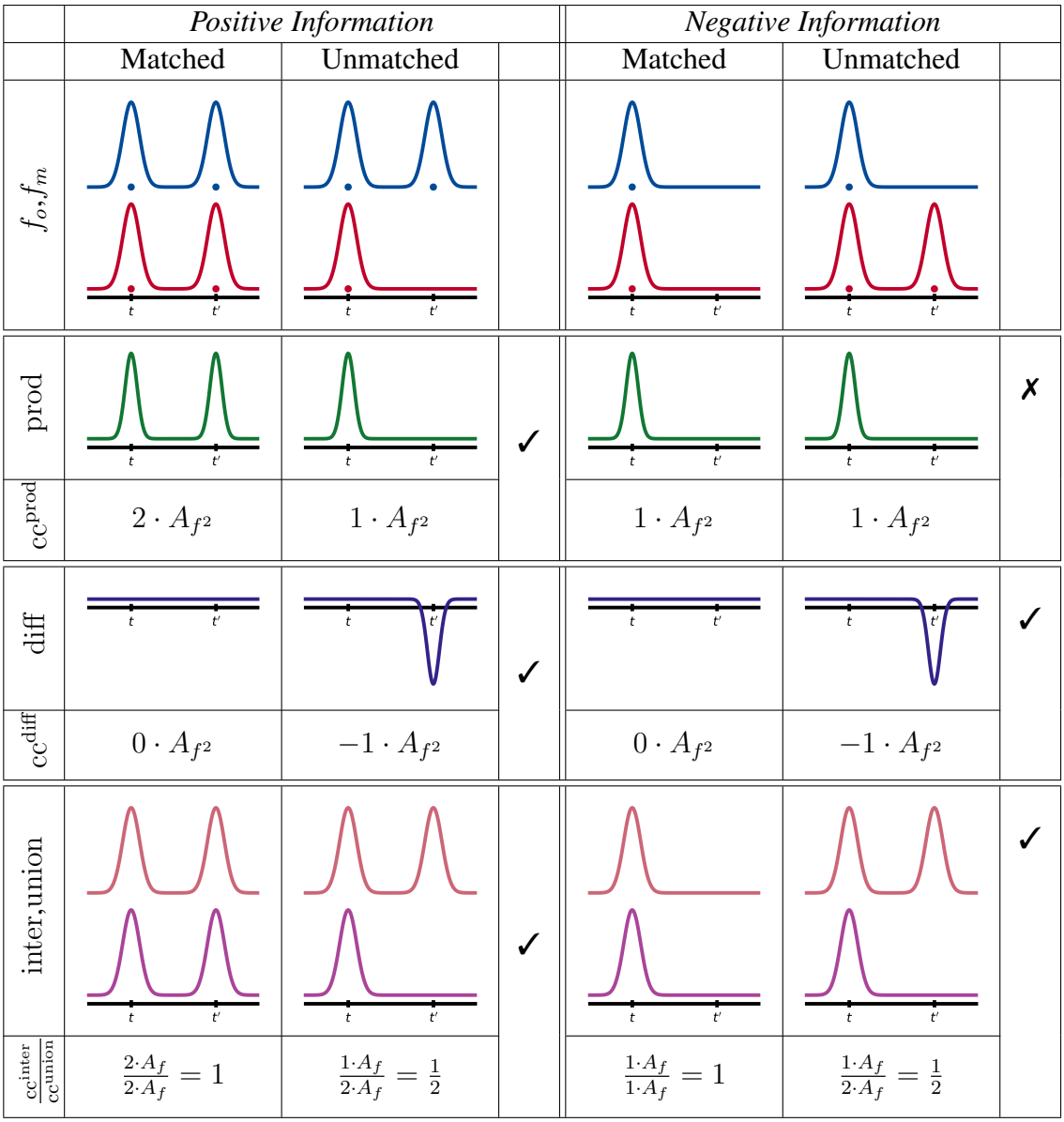

Figure 5.12: This illustrates the behavior of Score Functions for different inner functions γ regarding the *Positive* and *Negative Information* criteria. The Matched and Unmatched example situations at the top visualize the observed and mapped features as blue and red points as well as the corresponding continuous functions f_o, f_m as blue and red lines. For each inner function $\gamma \in \{\mathrm{prod}, \mathrm{diff}, \mathrm{inter}, \mathrm{union}\}$, we plot the resulting function for each situation and compute the corresponding score cc^{γ} in the row below. For the *Positive Information* on the left, the number of matched observations should be maximized. For the *Negative Information* on the right, the number of unmatched mapped features should be minimized. Therefore, the *Positive* or *Negative Information* criterion is satisfied (✓), if the score for the corresponding Matched situation is strictly greater than the score for the corresponding Unmatched situation and not satisfied otherwise (✗).

$$
\begin{aligned}
\mathrm{sf}^{\mathrm{diff}}: \quad & S \to \mathbb{R} \\
& (r,l) \mapsto \sum_{j=1}^{k_o} \mathrm{sf}^{\mathrm{diff}}_{j,j+r}(l) \\
\mathrm{sf}^{\mathrm{diff}}_{j,j+r}(l) := \quad & \begin{cases} \mathrm{cc}^{\mathrm{diff}}_{j,j+r}(l), & \text{if } j+r \in [1,k_m] \\ \lambda \cdot \mathrm{sf}^{\mathrm{diff}}_{j,j} + (1-\lambda) \cdot \mathrm{sf}^{\mathrm{diff}}_{j,0}, & \text{otherwise} \end{cases} \\
\mathrm{sf}^{\mathrm{diff}}_{j,0} := \quad & \mathrm{cc}^{\mathrm{diff}}(0), \text{ for } f = f_o^j \text{ and } g = \mathbf{0} \\
\mathrm{sf}^{\mathrm{diff}}_{j,j} := \quad & \mathrm{cc}^{\mathrm{diff}}(0), \text{ for } f = g = f_o^j
\end{aligned}
$$

Using the squared negative distance between function values as the inner function enables the Difference Score Function to penalize unmatched mapped features, while at the same time rewarding locations where observed features are matched with mapped features. This becomes evident, when considering the same example situations as for the Product Score Function, but this time with $\gamma = \mathrm{diff}$ as the inner function as shown in the row labeled diff in Figure 5.12. For both criteria, the Unmatched situations contain locations with different function values. Therefore, the integral over the diff function accumulates negative values and the Unmatched situations receive smaller scores than the Matched situations. This confirms that the Difference Score Function $\mathrm{sf}^{\mathrm{diff}}$ satisfies both, the *Positive* as well as the *Negative Information* criterion.

The Difference Score Function is the first Score Function that satisfies all four requirements. However, as we can see in the score distribution in Figure 5.11 in the second row, we still need our regularization technique to mitigate the bias introduced by unmatched crop rows. Here, a trade-off value of $\lambda = 0.3$ seems well suited since the value distribution of the regularized Difference Score Function on the right does not show a bias towards matching as many crop rows as possible. In our regularization technique, we approximate the best and worst possible score for the Difference and Product Score Function to compute a score for unmatched crop rows. This is due to the fact that the functions we used so far, i. e., the product- and negative-squared-distance-based functions $\mathrm{cc}^{\mathrm{prod}}$ and $\mathrm{cc}^{\mathrm{diff}}$ are in general not bounded. Because of the integral term that accumulates all inner function values along the crop rows, they can produce arbitrarily large, or small, scores for different crop rows depending on the length of the individual crop rows. Thus, we need to approximate the best and worst possible score to compute the unmatched crop row score for each crop row individually. Thus, our next goal is to find a Score Function that considers *Positive* as well as *Negative Information* and additionally also stays within a bounded value range for any crop row score. For such a bounded Score Function, the best and worst possible score are explicitly defined as the bounds of this function, which eliminates the approximation step for the best and worst possible crop row score in our regularization technique. With a bounded Score Function, we get more accurate unmatched crop row scores, which in turn improves the data association results, especially in a mapping application, where a good estimate of the unmatched crop row score is crucial.

To obtain a Score Function with similar discriminative abilities as the Difference Score Function, i. e., satisfying *Positive* and *Negative Information*, but bounded value range for individual crop rows, we propose a second variation of the Product Score Function based on the *Intersection over Union (IoU)* measure. The IoU is a robust measure with bounded value range between zero and one that determines how well two areas overlap by computing the ratio of the intersection over the union of both areas. This IoU measure is commonly used in computer vision to, for example, measure the accuracy of an object detection algorithm by computing the IoU of the detected and ground truth detection bounding boxes. In order to find the best overlap between two pairs of crop rows $R_j\,(F_o)$ and $R_{j+r}\,(F_m)$, we are interested in measuring how well the corresponding continuous functions f_o^j and f_m^{j+r} overlap. Our key idea is that we obtain a robust and bounded measure for this overlap by computing the IoU of the areas of both continuous functions. To this end, we first introduce the two functions $\mathrm{cc}^{\mathrm{union}}$ and $\mathrm{cc}^{\mathrm{inter}}$, where $\mathrm{cc}^{\mathrm{inter}}$ computes the intersection of the area of two functions $f, g\colon \mathbb{R} \to \mathbb{R}_{\geq 0}$ and $\mathrm{cc}^{\mathrm{union}}$ the union of the area respectively:

$$
\begin{aligned}
& & \mathbb{R} &\to \mathbb{R} & & \\
\mathrm{cc}^{\mathrm{inter}} &\colon \tau \mapsto & \int_{\mathbb{R}} & \min\left(f\left(t\right), g\left(t+\tau\right)\right) dt & & \forall\, f, g\colon \mathbb{R} \to \mathbb{R}_{\geq 0} \\
\mathrm{cc}^{\mathrm{union}} &\colon \tau \mapsto & \int_{\mathbb{R}} & \max\left(f\left(t\right), g\left(t+\tau\right)\right) dt & & \forall\, f, g\colon \mathbb{R} \to \mathbb{R}_{\geq 0}
\end{aligned}
$$

The area $A\left(f\right)$ under a function $f\colon \mathbb{R} \to \mathbb{R}_{\geq 0}$ is defined as the integral over the function, i. e., $A\left(f\right) := \int_{\mathbb{R}} f\left(t\right) dt$. We can omit taking the absolute value of the integral, since the integral is always non-negative for positive valued functions. The intersection area $A\left(f\right) \cap A\left(g\right)$ for two non-negative functions $f, g\colon \mathbb{R} \to \mathbb{R}_{\geq 0}$ is then given as the integral over the minimum function values. Similarly, the union $A\left(f\right) \cup A\left(g\right)$ is the integral over the maximum value of both functions. Since the IoU is the value of the intersection area over the union area, it is defined by the fraction $\frac{\mathrm{cc}^{\mathrm{inter}}}{\mathrm{cc}^{\mathrm{union}}}$. The values of this IoU ratio are always bounded between zero and one, since the intersection of two areas is by definition always smaller or equal to the union of the same areas.

Consulting Figure 5.12 again, we see the behavior for IoU-based Score Functions in the row labeled $\mathrm{inter}, \mathrm{union}$ and the resulting score in the row below labeled $\frac{\mathrm{cc}^{\mathrm{inter}}}{\mathrm{cc}^{\mathrm{union}}}$. Here, the intersection and union area are the same in the Matched situations, which yields the best possible score of 1 in these scenarios. This is in contrast to the Unmatched examples, where the intersection area is half the size of the union area, which yields a smaller intersection over union score of $\frac{1}{2}$. Since the score for the Matched situations is always larger than the score for the Unmatched situations, IoU-based Score Functions satisfy both, the *Positive* and the *Negative Information* criteria. Therefore, we can now define a bounded Score Function that also satisfies all four criteria, the *IoU per Row Score Function*:

$$
\begin{aligned}
\mathrm{sf}^{\text{iou-per-row}}: \quad & S \to \mathbb{R} \\
& (r,l) \mapsto \sum_{j=1}^{k_o} \mathrm{sf}_{j,j+r}^{\text{iou-per-row}}(l) \\
\mathrm{sf}_{j,j+r}^{\text{iou-per-row}}(l) := \quad & \begin{cases} \frac{\mathrm{cc}_{j,j+r}^{\text{inter}}(l)}{\mathrm{cc}_{j,j+r}^{\text{union}}(l)}, & \text{if } j+r \in [1,k_m] \\ \lambda \cdot \mathrm{sf}_{j,j}^{\text{iou-per-row}} + (1-\lambda)\,\mathrm{sf}_{j,0}^{\text{iou-per-row}}, & \text{otherwise} \end{cases} \\
\mathrm{sf}_{j,0}^{\text{iou-per-row}} := \quad & \frac{\mathrm{cc}^{\text{inter}}(0)}{\mathrm{cc}^{\text{union}}(0)}, \text{ for } f = f_o^j \text{ and } g = \mathbf{0} \\
\mathrm{sf}_{j,j}^{\text{iou-per-row}} := \quad & \frac{\mathrm{cc}^{\text{inter}}(0)}{\mathrm{cc}^{\text{union}}(0)}, \text{ for } f = g = f_o^j
\end{aligned}
$$

Looking at our regularization technique, we can now explicitly compute the unmatched row score $\mathrm{unm}_\lambda(j)$ for the j-th observed crop row as follows:

$$
\begin{aligned}
\mathrm{sf}_{j,0}^{\text{iou-per-row}} &= \frac{\mathrm{cc}^{\text{inter}}(0)}{\mathrm{cc}^{\text{union}}(0)}, \text{ for } f = f_o^j \text{ and } g = \mathbf{0} \\
&= \frac{\int_{\mathbb{R}} \min\left(f_o^j(t), 0\right) dt}{\int_{\mathbb{R}} \max\left(f_o^j(t), 0\right) dt} = \frac{\int_{\mathbb{R}} 0 dt}{\int_{\mathbb{R}} f_o^j(t)\, dt} = \frac{0}{A\left(f_o^j\right)} = 0 \\
\mathrm{sf}_{j,j}^{\text{iou-per-row}} &= \frac{\mathrm{cc}^{\text{inter}}(0)}{\mathrm{cc}^{\text{union}}(0)}, \text{ for } f = g = f_o^j \\
&= \frac{\int_{\mathbb{R}} \min\left(f_o^j(t), f_o^j(t)\right) dt}{\int_{\mathbb{R}} \max\left(f_o^j(t), f_o^j(t)\right) dt} = \frac{\int_{\mathbb{R}} f_o^j(t)\, dt}{\int_{\mathbb{R}} f_o^j(t)\, dt} = \frac{A\left(f_o^j\right)}{A\left(f_o^j\right)} = 1 \\
\mathrm{unm}_\lambda(j) &= \lambda \cdot \mathrm{sf}_{j,j}^{\text{iou-per-row}} + (1-\lambda) \cdot \mathrm{sf}_{j,0}^{\text{iou-per-row}} \\
&= \lambda \cdot 1 + (1-\lambda) \cdot 0 = \lambda
\end{aligned}
$$

This means that in the IoU per Row Score Function the unmatched crop row score is always equal to the trade-off parameter λ for all crop rows. As desired, we now have a Score Function that yields bounded scores over any crop row. As discussed before, a Score Function based on the IoU ratio also considers the *Positive* and *Negative Information*. We show the score distribution for the IoU Per Row Score Function in the third row for the unregularized version on the left with $\lambda = 0$ and a regularized version with $\lambda = 0.4$ on the right of Figure 5.11. The distribution of the IoU per Row Score Function is similar to the distribution of the Difference Score Function. The main advantage of the IoU per Row Score Function over the Difference Score Function is that the unmatched crop row score does not depend on the crop row anymore.

Inspired by our idea of using the IoU to find a better unmatched row score, we go one step further and use the IoU to define a Score Function without any bias towards matching as many crop rows as possible – and therefore a Score Function that does not require any regularization or unmatched row score. Our key idea is that this can be achieved by

computing the IoU jointly over all observed crop rows. We thus propose a third variation of the Score Function, called *IoU Score Function*:

$$\begin{aligned} \mathrm{sf}^{\mathrm{iou}}\colon \quad & S \to \mathbb{R} \\ & (r,l) \mapsto \frac{\sum_{j \in J(r)} \mathrm{cc}^{\mathrm{inter}}_{j,j+r}(l)}{\sum_{j \in J(r)} \mathrm{cc}^{\mathrm{union}}_{j,j+r}(l)} \in [0,1] \\ J(r) := \quad & \{j \in [1,\dots,k_o] \mid j+r \in [1,\dots,k_m]\} \end{aligned}$$

The difference between the IoU per Row and the IoU Score Function is that for the IoU function we first accumulate the intersection and union areas over all matched crop rows and take the fraction of the overall intersection and union to obtain one joint IoU score. This is in contrast to the IoU per Row Score Function, where we compute an individual IoU score for each crop row. Also, this IoU Score Function only considers the matched crop rows $j \in J(r)$ when computing the score for a shift s. This means that it does not require the unmatched row score unm_λ. Recall that we introduced this unmatched score for our regularization technique in order to mitigate the bias of the other three Score Functions towards row-shifts that have more matched crop rows. Taking a look at the score distribution of the IoU function shown in the bottom row of Figure 5.11, we confirm that the IoU Score Function is not biased towards matching as many crop rows as possible, although we are not using any regularization. This is explained by the fact that for the IoU Score Function matching more crop rows does not necessarily result in a higher score: For additional pairs of crop rows the intersection as well as the union area increase. While a larger intersection area results in a higher score, a larger union area decreases the score. Therefore, the IoU Score Function is by design not biased towards matching as many crop rows as possible, i. e., neither a regularization technique nor considering unmatched crop rows is necessary. Another advantage of the IoU Score Function is that not the individual crop row scores as in the IoU per Row Score Function, but the overall score for a shift is bounded between zero and one. Since the IoU Score Function, like the IoU per Row Score Function, is based on the intersection over union between crop row functions, the same arguments regarding the *Positive* and *Negative Information* criteria also hold for this Score Function (see Figure 5.11). Therefore, the IoU Score Function satisfies all four criteria, like the other two variations, i. e., the Difference and IoU per Row Score Function, required for a robust data association algorithm. The main advantage of the IoU Score Function over the other two variants is that it does not require our regularization technique to be unbiased towards the number of matched crop rows.

In this section we defined the Search Space S for our data association approach by introducing a parametrization $M(s)$ of a set of data association matches into a tuple of a row- and long-shift, called shift $s \in S$. This parameterized representation of sets of data association matches ensures that only matches that preserve the crop row structure are considered for data association. We further constrained our Search Space to a set of valid shifts S^{valid} according to the *Consistent Row Matches* and *Locality* requirements.

Finally, we presented four different Score Functions that can be used as the Target Function t for our data association approach. These Score Functions operate on shifts $s \in S$ and measure how well a shift s overlays the observed data onto the mapped data. The presented Score Functions are specifically designed for data association on densely and almost uniformly distributed features for both localization and mapping applications. This includes the ability to quantify how well the observed data is overlaid with the mapped data in a continuous matter to enable robust and efficient score computation, as well as the introduction of a regularization technique crucial for mapping applications. Our data association approach can therefore be summarized using the following equations:

$$
\begin{aligned}
S^{\text{valid}} &= S^{\text{local}} \cap \left(R^{\text{valid}} \times L^{\text{valid}}\right) & \mathcal{M}^{\text{valid}} &= M\left(S^{\text{valid}}\right) \\
s^* &= \underset{s \in S^{\text{valid}}}{\operatorname{argmax}} \operatorname{sf}(s) & M^* &= M\left(s^*\right) \\
\operatorname{sf} &\in \left\{\operatorname{sf}^{\text{prod}}, \operatorname{sf}^{\text{diff}}, \operatorname{sf}^{\text{iou-per-row}}, \operatorname{sf}^{\text{iou}}\right\}
\end{aligned}
$$

Before we present the experimental evaluation of our row-based data association approach, we explain how we obtain the input for our approach, i. e., the Plant-Rows, from raw data. Extracting Plant-Rows from raw data requires many different techniques ranging from detecting plant positions on the field from image data, over integrating those plant positions to obtain a set of plant features of reasonable size to then extract crop rows, as well as tracking those detected crop rows to assign the plant features to their corresponding crop rows. In the next section, we explain the techniques that we used to extract the Plant-Rows for our experimental evaluation in detail.

5.4 Extracting a Set of Plant-Rows

Our data association approach on individual plants is based on the row structure found on crop fields. In the previous Section 5.3.1, we defined this row structure as sets of Plant-Rows $\mathcal{R}(F_p)$ given a set of plant position features F_p and a mapping $\operatorname{row}(p)$ for each feature $p \in F_p$ to the index of the crop row it belongs to. However, several preprocessing steps of the recorded raw data are required to obtain this set of features F_p as well as the corresponding assignment to crop rows. In this section, we give an overview of these steps that we used to extract the required information from our real-world data.

As explained in Section 2.1.2, our real-world data contains timestamped high-resolution RGB image data I_t as well as odometry o_t, IMU i_t and GPS g_t information. Using the timestamps, we arrange our data set $D = \{d_t\}$ as chronological sequence of time synchronized tuples $d_t = (I_t, o_t, i_t, g_t)$ containing one element of each data type at each time step t. When creating this sequence, we only add a tuple d_t to the sequence, if the motion of the vehicle between time step $t-1$ and t is sufficiently large, i. e., either the heading of the vehicle changed by more than 4° or the vehicle traversed a distance of more than $0.1\,\text{m}$. We extract the Plant-Rows from this raw information in three steps: First, we detect the plant positions $z \in Z_t$ in each image I_t. Second, we accumulate the detected plant positions Z_t while the vehicle traverses the field in the direction of the crop rows,

i. e., until it reaches the end of the field at time step t_{eof}, to obtain a larger set of plant positions F_p. To this end, we iteratively build and optimize a pose graph $\mathcal{G}_t$ from data set elements $d_1, \ldots, d_t$ that jointly estimates the poses of the vehicle $\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_t$ as well as the plant feature positions $\hat{z} \in \hat{Z}_t$ for all plants detected in $I_1, \ldots, I_t$ in a common reference frame. Third, we apply our Pattern Hough Transform on the accumulated set of plant position estimates $\hat{Z}_t$ to first detect and then track the crop row structure of the field during the construction of the graph. In each iteration, we use this crop row information to determine a crop row index $\text{row}(\hat{z})$ for the plant positions $\hat{z} \in \hat{Z}_t$ that are not yet assigned to a crop row. By repeating these three steps until the vehicle reaches the end of the field at time step t_{eof}, we obtain a pose graph $\mathcal{G}_t$ for each time step $1, \ldots, t_{\text{eof}}$. After reaching the end of the field, we create a new pose graph to integrate plant positions over the next set of traversed crop rows. Therefore, each pose graph always represents one traversal of the field along the same subset of crop rows. The set of plant features F_p^t at each time step t is then equal to the feature nodes $\hat{Z}_t$ of $\mathcal{G}_t$. Together with the corresponding mapping $\text{row}(\hat{z})$ for each feature $\hat{z} \in F_p^t$ we can construct the set of Plant-Rows $\mathcal{R}\left(F_p^t\right)$ at each time step t as defined in the previous section.

Additionally to the preprocessing required to obtain the row structure of the field, we also need to determine the *detection range*, which is the area where the vehicle could have detected plant features. This information is crucial to correctly apply the Positive and Negative Information criteria when computing scores with our Score Functions: Both criteria are based on rewarding matched and punishing unmatched features. However, there are situations where a feature remains unmatched because it is not within the detection range, i. e., the vehicle did not gather any information about this area. Due to the limited sensor range, the vehicle could not have detected any feature close to the unmatched feature. Therefore, we cannot and should not reason about features outside of the detection range of the vehicle, i. e., any data association algorithm based on Positive or Negative Information should ignore these features. In the following, we present the techniques used in each step of our raw data preprocessing.

5.4.1 Stem Emerging Point Detection

For localization and mapping applications on individual plants, we first need to detect plants in the images I_t of our data set. Common object detection approaches return bounding boxes that surround the outline of the target object in the image. If a position of the detected object is required, this position is usually defined as the center of this bounding box $c = (c_i, c_j)$ in image coordinates. However, since the size of the bounding box depends on the outline of the object in the image, which in turn depends on the point of view from which the object is observed, the center of the bounding box also depends on the point of view (see Figure 5.13).

However, in our application scenario, where the detected plant positions are used to correct the pose estimate of a localization or mapping algorithm, detecting plants with a position that does not depend on the point of view is crucial. Therefore, instead of detecting the whole plant, we decide to detect a unique point on the plant so that this

Figure 5.13: This figure shows a close up of the same plant in two images captured about 2.5 sec apart. Even slight changes in the view point cause bounding boxes of different sizes when detecting the whole plant (orange outline). Using the center of the bounding box to define the position of the detection (orange point) therefore results in different detected locations for the same plant. In contrast, when only detecting the SEP of the plant with a fixed-sized bounding box (blue outline), so that the SEP is located at the center of the bounding box (blue point), yields more accurate detections.

point is always located in the center of the bounding box. Detecting this unique point on a plant instead of the whole plant then yields stable plant positions independent of the point of view (see Figure 5.13). This unique point, that exists on every plant, and therefore can be detected on all crop types, is the *stem emerging point* (SEP), i. e., the location where the stem of the plant emerges from the ground. It is unique since a stem of a plant has exactly one location where it can emerge from the ground. To detect these SEPs in our RGB image data I_t, we decided to use an off-the-shelf neural network for object detection. This network was then retrained on a subset of our image data, where the SEP position of each plant was labeled in each image. A constant-sized bounding box centered around the labeled position was added to facilitate training. We define the SEP position of a detection by computing the center of the detected bounding boxes c in image coordinates[1].

Processing an image I_t of our data set with the retrained network then yields a set of plant feature positions C_t in image coordinates. However, to incorporate this information into our pose graph in the second preprocessing step, we need to project the SEP position from the image coordinate frame into the coordinate frame of the vehicle, i. e., into metrical measurements z that describe the position of the detected SEP relative to the pose of the vehicle. The projection prj of the detected SEP position c in the image is defined by the intrinsic and extrinsic calibration of the camera sensor (see Section 2.1.2 for details on the calibration procedure). Geometrically, this projection is described by a ray r_c that emerges from the origin of the camera sensor mounted on the robot p_{camera} and

[1]The SEP detection network was developed by Nina Pant for her Bachelor Thesis.

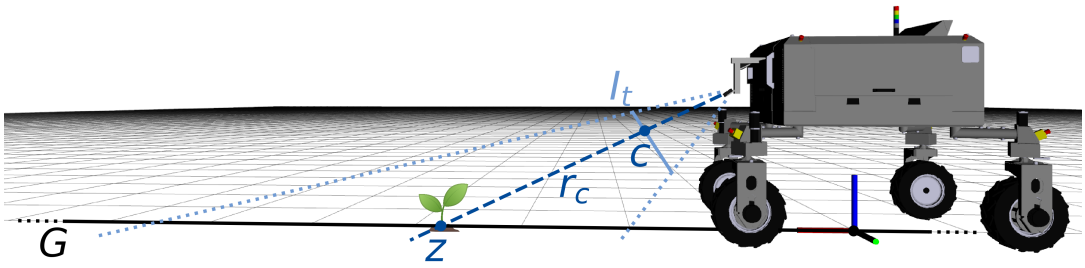

Figure 5.14: This illustrates the projection prj for the detection of a plant in front of the vehicle. The image plane I_t (solid line) and its projection onto the ground (dotted lines) are shown in light blue. The detection of the SEP in the image plane c and its projection onto the ground $z = \mathrm{prj}\,(c)$ are shown as blue points. The ray that projects c to the ground is visualized as blue dashed line.

passes through the detected SEP position c on the image plane in the direction d_c (see Figure 5.14). To determine the position of the SEP in the coordinate frame of the vehicle, we leverage the fact, that the SEP, i. e., the stem *emerging* point, is always located where the stem emerges from the ground. Therefore, the SEP is always located where the projected ray r_c intersects the ground G in the vehicle coordinate frame. Assuming that the ground around the vehicle is locally flat, we model the ground plane G in the frame of the vehicle as the plane where the z-coordinate is equal to 0. The metrical position z of the SEP in the vehicle coordinate frame is then defined as the intersection of the projected ray r_c with the ground plane G, i. e., the point on the ray with a z-coordinate equal to 0. Applying this projection prj to all detections c, we obtain the set of metrical SEP positions Z_t for each image I_t:

$$r_{c_n} := p_{\text{camera}} + s \cdot d_{c_n} \qquad C_t := \{c_1, \ldots, c_{n_c}\} \tag{5.6}$$

$$\mathrm{prj}\,(c_n) := r_{c_n} \cap G \qquad G := \left\{(x, y, z) \in \mathbb{R}^3 \mid z = 0\right\} \tag{5.7}$$

$$z_n := \mathrm{prj}\,(c_n) \qquad Z_t := \{z_1, \ldots, z_{n_c}\} \tag{5.8}$$

To confirm that the detections of this algorithm are accurate enough to be used for pose correction in localization and mapping applications, we evaluate the SEP detection algorithm on the Eichstetten and Eschikon data sets recorded in 2018 (see Section 2.1.3). A qualitative example for the detection results on each of the four crop types Kohlrabi, Chinese Cabbage, Sweetheart Cabbage and Sugar Beet is shown in Figure 5.15.

We list the quantitative results sorted by crop type in Table 5.1. Overall, the *mean detection error* stays below $5\,\mathrm{cm}$ on all crop types while missing at most $2\,\%$ on Sugar Beet of the plants. These results confirm that the SEP detection algorithm produces accurate detections with a low *false negative rate* (FNR) and therefore should be well suited for our application scenario. Notably, the results also show slight variation in performance on different crop types: On Chinese Cabbage we see the highest mean error of over $4\,\mathrm{cm}$ compared to an error of around $3\,\mathrm{cm}$ on the other crop types. Similarly on Sugar Beet,

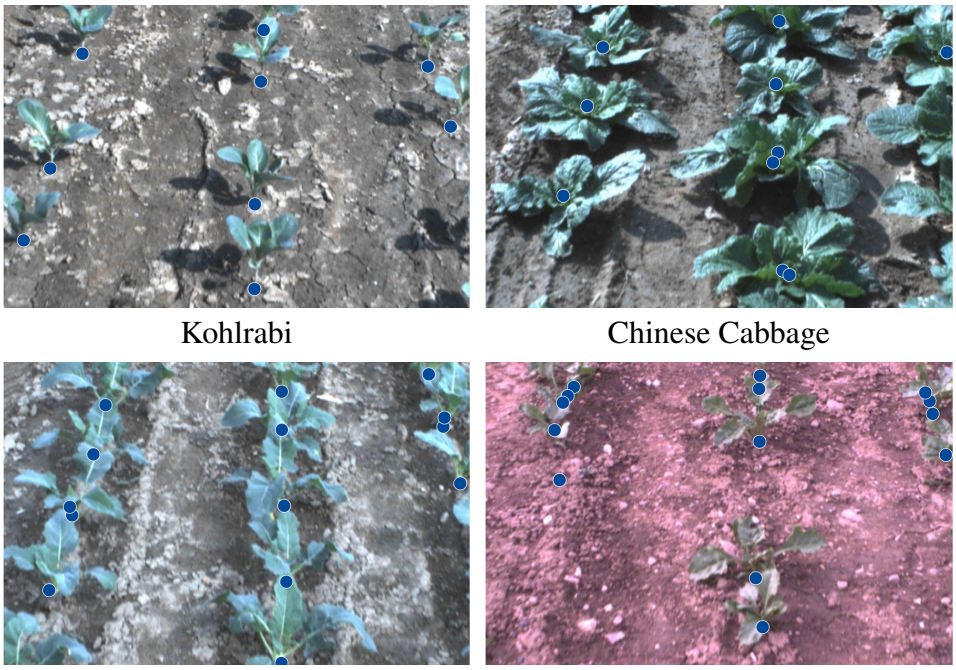

Figure 5.15: This figure shows the plant positions found by the SEP detection algorithm (blue) for each crop type.

we observe the highest false negative rate of 2 % compared to almost negligible false negative rates on the other crop types. This is an interesting fact to keep in mind for the evaluation of our row-based data association, since less accurate detections and a higher false positive rate cause more ambiguous situations during data association. Therefore, data recorded on Chinese Cabbage and Sugar Beet poses a greater challenge for our data association approach. We discuss how the difference in performance of the detection algorithm affects the performance of our data association approach in our experimental evaluation. In the next section, we explain how we integrate the detected SEP positions Z_t over time to obtain a set of plant position features F_p.

5.4.2 Building the Graph Structure

We obtain a larger set of plant position features F_p by integrating the position measurements $z \in Z_t$ of detected plants along the crop rows. To this end, we use a pose graph optimization that jointly estimates the poses $\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_t$ of the vehicle as well as the positions $\hat{z} \in \hat{Z}_t$ of the plants detected in $I_1, \ldots, I_t$ in a common reference frame. The set of estimated plant positions $\hat{Z}_t$ after optimization of the pose graph $\mathcal{G}_t$ then yields a set of plant position features F_p^t for every time step t.

To estimate the vehicle pose $\hat{\mathbf{x}}$ as well as the position of the plant features $\hat{z}$ in a pose graph, we need to represent both estimates as different types of nodes. Since our approach is focused on data from a ground vehicle, it suffices to estimate the pose of the vehicle

Crop Type	Mean	TP	FN	FNR	GT
Kohlrabi	0.027	2196	3	0.1	2036
Chinese Cabbage	0.043	843	0	0.0	724
Sweetheart Cabbage	0.033	1073	0	0.0	838
Sugar Beet	0.029	843	17	2.0	673

Table 5.1: This table shows the detection performance of the SEP detection algorithm used in our preprocessing pipeline on each crop type. We compute the error of a detection as metrical distance to the closest labeled plant position by projecting both onto the ground plane G using prj. The mean column shows the mean error over all detections of a crop type. The detected plant positions are counted as *true positive* detections (TP). The missed plants, i. e., labeled plant positions that remained unmatched, are counted as *false negative* detections (FN). Since multiple detections of the same plant are allowed, we can have more true positive (TP) detections than labeled plant positions (GT). The *false negative rate* (FNR $= \frac{\mathrm{FN}}{\mathrm{FN+TP}}$) is shown in percent.

$\hat{\mathbf{x}}$ and the plant feature position $\hat{z}$ on the ground plane, i. e., in a two-dimensional space. The pose of the vehicle is therefore represented as

$$\hat{\mathbf{x}}_t = \left(\hat{p}_t, \hat{\theta}_t\right)$$

at a time step t with a two-dimensional position $\hat{p}_t = (\hat{p}_{x,t}, \hat{p}_{y,t})$ and a heading $\hat{\theta}_t$. The feature estimate $\hat{z}$ is a two-dimensional position

$$\hat{z} = (\hat{z}_x, \hat{z}_y)$$

of the detected plants. The pose graph $\mathcal{G}_t$ for a time step t therefore consists of

$$\mathcal{G}_t = \left(\hat{X}_t, E_{\hat{X}_t}, \hat{Z}_t, E_{\hat{Z}_t}\right),$$

where $\hat{X}_t := \{\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_t\}$ is the set of pose nodes, $E_{\hat{X}_t}$ is the set of *pose edges* that only constrain pose nodes, $\hat{Z}_t$ is the set of plant feature position nodes and $E_{\hat{Z}_t}$ is the set of *feature edges* that constrain a feature node relative to a pose node.

To create our pose graph $\mathcal{G}_t$ from the time synchronized data $d_1, \ldots d_t$, we extend the previously constructed graph $\mathcal{G}_{t-1}$ that already contains the information from data elements $d_1, \ldots, d_{t-1}$ with the information contained in d_t. To incorporate the information of time step t with $d_t = (I_t, o_t, i_t, g_t)$, we first add a new pose node $\hat{\mathbf{x}}_t$ to $\hat{X}_{t-1}$ to obtain the set of pose nodes $\hat{X}_t$ at time step t. For the information about the vehicle pose from incremental measurements, i. e., the odometry o_t and IMU i_t measurements, we add two binary edges between consecutive pose nodes $\hat{\mathbf{x}}_t$ and $\hat{\mathbf{x}}_{t-1}$. The IMU information is used to constrain the heading estimate $\hat{\theta}_t$ of the pose node $\hat{\mathbf{x}}_t$ relative to the previous pose estimate $\hat{\mathbf{x}}_{t-1}$ and the odometry information constrains the position estimate $(\hat{p}_{x,t}, \hat{p}_{y,t})$ relative to $\hat{\mathbf{x}}_{t-1}$. To obtain globally referenced pose and feature positions, we also add a unary edge that constrains the position estimate of $\hat{\mathbf{x}}_t$ according to the measured GPS

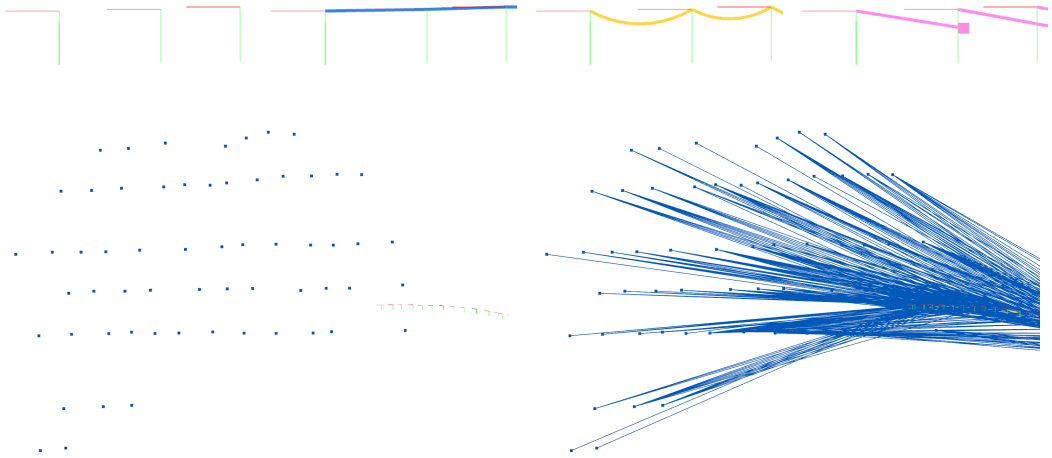

Figure 5.16: This figure shows the nodes and edges of the pose graph $\mathcal{G}_t$ for a time step t. The pose estimates $\hat{\mathbf{x}} \in \hat{X}_t$ are visualized as RGB colored axes objects (top, first). The set of pose edges $E_{\hat{X}_t}$ is shown as straight, light blue lines for the odometry (top, second), curved yellow lines for the IMU (top, third) and as pink lines with a pink square indicating the measured GPS location of the vehicle (top, fourth). In the bottom row, we show an example for the set of feature position estimates Z_t (blue squares) on the left and the feature nodes with the corresponding set of feature edges $E_{\hat{Z}_t}$ (blue lines) on the right.

coordinates g_t. Since these edges purely operate on the pose nodes, we add them to the set of pose edges $E_{\hat{X}_{t-1}}$ to obtain $E_{\hat{X}_t}$. An example for the set of pose nodes $\hat{X}_t$ and corresponding pose edges $E_{\hat{X}_t}$ of a pose graph $\mathcal{G}_t$ created from real-world data is shown on the top row of Figure 5.16.

For the image I_t, we first apply the SEP detection algorithm on I_t to obtain a set of SEP detections in image coordinates. We then project these detections onto the ground plane with the pose of the vehicle at its center as described in the previous section. This yields metrical measurements of the position of each plant feature $z \in Z_t$ relative to the pose estimate $\hat{\mathbf{x}}_t$ at time step t. Next, we need to determine whether a plant detected in image I_t was already observed in a previous time step and therefore, whether a feature position estimate $\hat{z} \in \hat{Z}_{t-1}$ corresponding to the measurement z already exists in our graph structure. Since the vehicle is traversing the field aligned with the crop rows, we know that the change in heading and position remains small between two consecutive timestamps $t-1$ and t. Therefore, the amount of accumulated error between two consecutive pose estimates $\hat{\mathbf{x}}_{t-1}$ and $\hat{\mathbf{x}}_t$ should also remain small. Thus, we can assume that a plant detected in I_t and projected using $\hat{\mathbf{x}}_t$ is close to a feature position estimate $\hat{z}^* \in \hat{Z}_{t-1}$, if the detected plant is represented by this feature node $\hat{z}$. This means that we can use the Nearest Neighbor algorithm to associate measurements of detected plant positions $z \in Z_t$ with the corresponding feature position estimate $\hat{z} \in \hat{Z}_{t-1}$ of previous time steps: For each measurement $z \in Z_t$, we compute the estimated position of this feature z' by projecting the measurement z into the global reference frame of our pose graph using the corresponding pose estimate $\hat{\mathbf{x}}_t$. If the plant position feature $\hat{z}^* \in \hat{Z}_{t-1}$

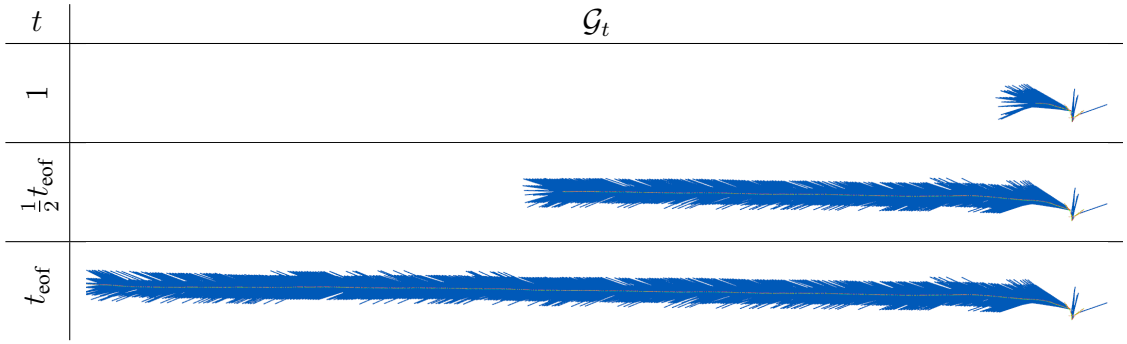

Figure 5.17: This figure shows the pose graph $\mathcal{G}_t$ for different time steps: At the beginning $t = 1$ (top), in the middle of the field at $t = \frac{1}{2}t_{\text{eof}}$ (mid) and at the end $t = t_{\text{eof}}$ when the vehicle reaches the end of the field (bottom).

closest to the projected measurement z', i. e., its nearest neighbor, is closer than $0.15\,\text{m}$, we associate the measurement z with the feature position estimate $\hat{z}^*$. If no such estimate exists, we insert a new feature position estimate $\hat{z}^* = z'$ into the graph. The measurement $z \in Z_t$ is then modeled as binary edge between the associated feature position estimate $\hat{z}^*$ and the pose estimate $\hat{\mathbf{x}}_t$ of the corresponding timestamp t. Since these binary edges formulate constraints between feature and pose nodes of the graph, they are added to the set of feature edges $E_{\hat{Z}_{t-1}}$ to obtain $E_{\hat{Z}_t}$. An example for the full pose graph $\mathcal{G}_t$ including feature nodes and edges is shown on the bottom of Figure 5.16.

This procedure iteratively constructs a pose graph $\mathcal{G}_t$ for each time step t of the data set, where the data $d_1, \ldots, d_t$ from time steps $1, \ldots, t$ is represented in the graph $\mathcal{G}_t$. Before repeating this procedure for the next data set element d_{t+1} and thereby constructing $\mathcal{G}_{t+1}$, we optimize the pose graph structure using the *g2o* framework by Kümmerle et al. [2011]. Optimizing the graph structure in each iteration ensures that the pose graph converges to the correct solution.

We repeat this procedure for all data set elements d_t until the vehicle reaches the end of the field t_{eof}. This creates an optimized pose graph $\mathcal{G}_{t_{\text{eof}}}$ that contains the pose of the vehicle as well as the position of all observed plant features while traversing a set of crop rows (see Figure 5.17). The set of plant feature nodes $\hat{Z}_t$ of the graph $\mathcal{G}_t$ then defines the set of plant features F_p^t for a time step t. In the next section, we explain how we assign each plant position $\hat{z} \in \hat{Z}_t$ in the graph $\mathcal{G}_t$ to its crop row during the construction of the graph, i. e., how we obtain the mapping $\text{row}\,(\hat{z})$ required to determine the set of Plant-Rows $\mathcal{R}\,(F_p)$ on a set of plant position features F_p.

5.4.3 Plant-Row Extraction

In this section, we explain how we extract the crop row structure from a set of plant position features F_p^t using the information accumulated until time step t in the pose graph $\mathcal{G}_t$. We then use this row structure to define a mapping $\text{row}\,(\hat{z})$ for all features $\hat{z} \in F_p^t$ as required to define the set of Plant-Rows $\mathcal{R}\left(F_p^t\right)$. To this end, we first detect the crop row structure on a local subset of plant features $\hat{z} \in F_p^t\,(\hat{\mathbf{x}}_t) \subset F_p^t$ around the latest pose

node $\hat{\mathbf{x}}_t$. Second, we associate the detected crop rows with the crop row structure of $\mathcal{G}_{t-1}$ to track the crop rows while the vehicle traverses the field. Finally, we use these indexed crop rows to assign the crop row index to each plant position feature which defines the mapping row.

Given a set of plant position features F_p^t, we first extract the local crop row structure around the current pose estimate of the vehicle

$$\hat{\mathbf{x}}_t = \left(\hat{p}_t, \hat{\theta}_t\right).$$

To this end, we only consider plant position features in F_p^t that are within $4\,\mathrm{m}$ to the current position estimate $\hat{p}_t$, i. e., we compute a subset of local plant position features

$$F_p^t\left(\hat{\mathbf{x}}_t\right) \;:=\; \left\{\hat{z} \in F_p^t \mid \|\hat{z} - \hat{p}_t\| < 4.0\right\}.$$

On these local plant position features, we extract the crop row structure using our crop row detection algorithm, the Pattern Hough Transform, presented in Chapter 3. We directly use the plant position features $\hat{z} \in F_p^t\left(\hat{\mathbf{x}}_t\right)$ as vegetation feature input to create the Feature Map required for the Pattern Hough Transform algorithm. This yields a set of sorted, parallel and equidistant local crop rows $P_{\theta,s,o}$ with offset o, normal direction θ and spacing s:

$$\begin{aligned}
P_{\theta,s,o} &= \left\{R_j\left(F_p^t\left(\hat{\mathbf{x}}_t\right)\right)\right\}_{j\in\mathbb{Z}} \\
R_j\left(F_p^t\left(\hat{\mathbf{x}}_t\right)\right) &:= \left\{p \in \mathbb{R}^2 \mid d_j\left(p\right) = 0\right\} \\
d_j\left(p\right) &:= \left|p \cdot n_\theta - \left(o + j \cdot s\right)\right| \\
n_\theta &:= \left(\cos\left(\theta\right), \sin\left(\theta\right)\right)^T
\end{aligned}$$

We show an example for the set of local plant position features and the detected crop rows as well as the following definitions in Figure 5.18.

According to the definition of the pattern $P_{\theta,s,o}$, we obtain an infinite set of indexed crop rows $\mathcal{R}\left(F_p^t\left(\hat{\mathbf{x}}_t\right)\right)_{\mathbb{Z}} := \left\{R_j\left(F_p^t\left(\hat{\mathbf{x}}_t\right)\right)\right\}_{j\in\mathbb{Z}}$ with equidistant spacing s between neighboring crop rows $R_{j-1}\left(F_p^t\left(\hat{\mathbf{x}}_t\right)\right)$, $R_j\left(F_p^t\left(\hat{\mathbf{x}}_t\right)\right)$ and $R_{j+1}\left(F_p^t\left(\hat{\mathbf{x}}_t\right)\right)$. We use this equidistant spacing to assign a feature $\hat{z} \in F_p^t\left(\hat{\mathbf{x}}_t\right)$ to the j-th crop row, if it is considerably closer to the j-th crop row than to its neighbors. A point $p \in \mathbb{R}^2$ is closer to the j-th crop row than its neighbors, if its distance to the crop row is smaller than half the spacing. We therefore assign a plant feature $\hat{z} \in F_p^t\left(\hat{\mathbf{x}}_t\right)$ to the j-th crop row, if its distance is smaller than $75\,\%$ of half the spacing:

$$\mathrm{row}_{F_p^t\left(\hat{\mathbf{x}}_t\right)}\left(\hat{z}\right) \;:=\; j, \text{ such that } d_j\left(\hat{z}\right) < 0.75 \cdot \frac{s}{2}.$$

If no such crop row can be found, i. e., the plant feature lies in the middle between two crop rows, it does not belong to either crop row and the feature $\hat{z}$ remains unassigned (see Figure 5.18). Features that are not assigned to a crop row are outliers and therefore

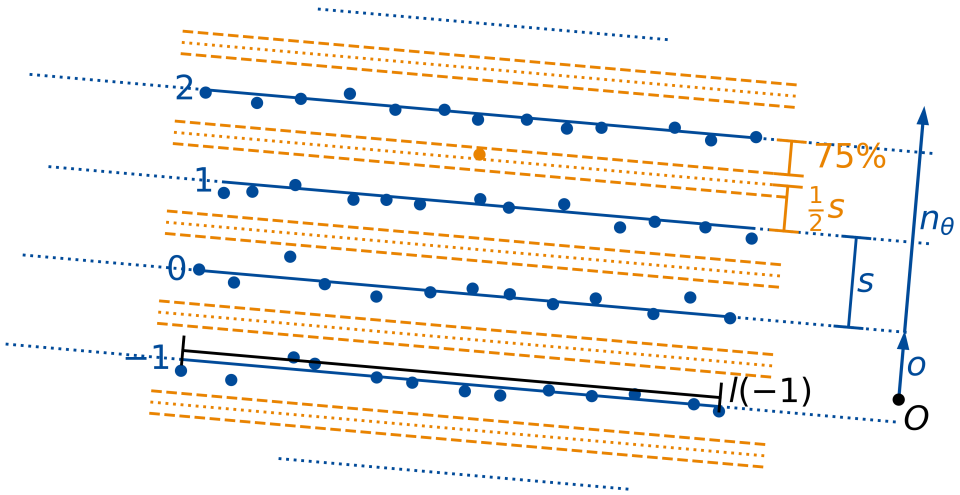

Figure 5.18: This figure shows an example of the $\mathrm{row}_{F_p^t(\hat{\mathbf{x}}_t)}(\hat{z})$ assignment for the local features $\hat{z} \in F_p^t(\hat{\mathbf{x}}_t)$ (blue and orange points). The infinite set of crop rows defined by the detected pattern $P_{\theta,s,o}$ is visualized as blue lines (solid and dotted). We illustrate the distance equal to half the spacing $\frac{1}{2}s$ as dotted orange line. The distance threshold used in $\mathrm{row}_{F_p^t(\hat{\mathbf{x}}_t)}(\hat{z})$, i. e., 75 % of $\frac{1}{2}s$ is shown as dashed orange line around each crop row. Blue features lie within the threshold and are therefore assigned to their corresponding crop rows, while the orange feature lies between crop row 1 and crop row 2 and therefore remains unassigned. The length $l(-1)$ of crop row -1 is illustrated in black. The finite set of detected crop rows F_o is comprised of the solid blue lines.

not considered in the following steps. The mapping $\mathrm{row}_{F_p^t(\hat{\mathbf{x}}_t)}$ then also defines the set of features $F_p^t(\hat{\mathbf{x}}_t, j)$ for each crop row $R_j\left(F_p^t(\hat{\mathbf{x}}_t)\right)$:

$$F_p^t(\hat{\mathbf{x}}_t, j) \;:=\; \left\{\hat{z} \in F_p^t(\hat{\mathbf{x}}_t) \mid \mathrm{row}_{F_p^t(\hat{\mathbf{x}}_t)}(\hat{z}) = j\right\}.$$

For more robust results, we remove outlier crop rows $R_j\left(F_p^t(\hat{\mathbf{x}}_t)\right)$ that are not well supported by their set of plant features $F_p^t(\hat{\mathbf{x}}_t, j)$. To this end, we define two criteria, the crop row *length* $l(j)$ and *density* $d(j)$, that describe the distribution of plant features $F_p^t(\hat{\mathbf{x}}_t, j)$ along the j-th crop row $R_j\left(F_p^t(\hat{\mathbf{x}}_t)\right)$. Given the set of features $F_p^t(\hat{\mathbf{x}}_t, j)$, we define the length $l(j)$ and density $d(j)$ of the j-th crop row as follows:

$$\begin{aligned} n(j) &:= \left|F_p^t(\hat{\mathbf{x}}_t, j)\right| \\ l(j) &:= \max_{\hat{z}_1, \hat{z}_2 \in F_p^t(\hat{\mathbf{x}}_t, j)} \|\hat{z}_1 - \hat{z}_2\| \\ d(j) &:= \frac{n(j)}{l(j)} \end{aligned}$$

The length of a detected crop row is therefore equal to the longest distance between two of its associated features in meters (see Figure 5.18). The density, i. e., the number of plant

features per meter, is then equal to the number of plant features $n(j)$ divided by the length $l(j)$ of the crop row. A crop row is regarded as an outlier, if it is considerably shorter than the other detected crop rows or if its overall plant density is smaller than expected. We therefore compute the average length $\bar{l}$ over all non-empty crop rows:

$$\begin{aligned} \bar{l} &:= \frac{1}{|J_{>0}|} \sum_{j \in J_{>0}} l(j) \\ J_{>0} &:= \{j' \in \mathbb{Z} \mid n(j') > 0\}. \end{aligned}$$

We then disregard a crop row, if either its length $l(j)$ is smaller than $50\,\%$ of the average crop row length $\bar{l}$ or its overall density $d(j)$ is smaller than 1.5 plants per meter. After filtering the outliers, we obtain the finite set of detected crop rows (see Figure 5.18)

$$F_o := \left\{ R_j\left(F_p^t(\hat{\mathbf{x}}_t)\right) \mid l(j) \geq 0.5 \cdot \bar{l} \wedge d(j) \geq 1.5 \right\}.$$

Since we detect the crop rows locally around the current pose estimate $\hat{\mathbf{x}}_t$, the indices of the crop rows are not consistent with the indices of the crop rows from previous iterations, i. e., until time step $t-1$. We therefore need to associate the detected crop rows F_o with the already tracked crop rows $F_m = \left\{ R_{l_1}\left(F_p^{t-1}\right), \ldots, R_{l_{k_m}}\left(F_p^{t-1}\right) \right\}$.

For this crop row tracking, we present a row-based data association algorithm, designed for tracking the crop row structure while the vehicle traverses the field along the crop rows. Again, we first explain how we model the space of sets of data association matches $\mathcal{M}$ and define the valid subset of data association matches $\mathcal{M}^{\text{valid}}$ considered for this application as well as an appropriate Target function t. The row-based data association algorithm for crop row tracking is then formulated as:

$$M^* := \underset{M \in \mathcal{M}^{\text{valid}}}{\operatorname{argmax}}\, t(M) \qquad \mathcal{M}^{\text{valid}} \subset \mathcal{M} \subset \mathcal{P}(F_o \times F_m)$$

Analogue to our previously presented row-based data association algorithms, we again use a row-shift parameter r to represent a set of data association matches between the set of detected (F_o) and tracked (F_m) crop rows. This row-shift parameter maps a detected row onto a tracked crop row by shifting the index j of a detected crop row onto its mapped crop row $j + r$. Let F_o be the set of detected crop rows with smallest crop row index j_{min} and largest crop row index j_{max} and let F_m be the set of tracked crop rows. We then define the valid range of row-shifts R^{valid} as all shifts that represent non-empty sets of crop row matches, i. e., all shifts that match at least one pair of crop rows:

$$\begin{aligned} r^{\text{min}} &:= \min(r_{\text{left}}, r_{\text{right}}) & r_{\text{left}} &:= l_1 - j_{\text{max}} \\ r^{\text{max}} &:= \max(r_{\text{left}}, r_{\text{right}}) & r_{\text{right}} &:= l_{k_m} - j_{\text{min}} \\ R^{\text{valid}} &= \left[r^{\text{min}}, r^{\text{max}}\right] \subset \mathbb{Z} \end{aligned}$$

Similarly to previous definitions, we define the set of valid data association matches as $\mathcal{M}^{\text{valid}} := M\left(R^{\text{valid}}\right)$, where

$$M(r) := \left\{ \left(R_j\left(F_p^t(\hat{\mathbf{x}}_t)\right), R_{j+r}\left(F_p^{t-1}\right)\right), j \in [j_{\text{min}}, j_{\text{max}}] \mid j + r \in [l_1, \ldots, l_{k_m}] \right\}$$

is the parametrization of a set of matched crop rows defined by the row-shift r.

Since we represent the set of valid data association matches $\mathcal{M}^{\text{valid}}$ using row-shifts $r \in R^{\text{valid}}$, we need to define a Target Function t that operates on row-shifts r. For our crop row tracking data association, we can leverage that $F_p^t(\hat{\mathbf{x}}_t) \cap F_p^{t-1}$ is not empty, i. e., we have features $\hat{z} \in F_p^t(\hat{\mathbf{x}}_t) \cap F_p^{t-1}$ that are assigned to both a detected crop row in F_o as well as a tracked crop row in F_m. We use these features to find the correct row-shift and therefore the correct assignment of detected to tracked crop rows. Our main idea is that each feature $\hat{z} \in F_p^t(\hat{\mathbf{x}}_t) \cap F_p^{t-1}$ defines its own row-shift $r_{\hat{z}}$ as the difference between the indices of its assigned tracked and detected crop rows:

$$r_{\hat{z}} := \text{row}(\hat{z}) - \text{row}_{F_p^t(\hat{\mathbf{x}}_t)}(\hat{z}).$$

We then count for each valid row-shift $r \in R^{\text{valid}}$ how many features $\hat{z} \in F_p^t(\hat{\mathbf{x}}_t) \cap F_p^{t-1}$ support this row-shift r to obtain our Target Function t:

$$\begin{aligned} t : R^{\text{valid}} &\to \mathbb{R} \\ r &\mapsto \left| \left\{ \hat{z} \in F_p^t(\hat{\mathbf{x}}_t) \cap F_p^{t-1} \mid r_{\hat{z}} = r \right\} \right| \end{aligned}$$

The best data association M^* between the detected and tracked crop rows, is therefore defined by the best row-shift r^*, that is best supported by the already tracked features:

$$R^{\text{valid}} = \left[r^{\min}, r^{\max} \right] \qquad \mathcal{M}^{\text{valid}} = M\left(R^{\text{valid}}\right)$$
$$r^* = \underset{r \in R^{\text{valid}}}{\text{argmax}}\, t(r) \qquad M^* = M(r^*)$$

An example for this row tracking data association is shown in Figure 5.19. We use this row-shift r^* to assign the crop row index $\text{row}(\hat{z}) := j + r^*$ to each feature $\hat{z} \in F_p^t(\hat{\mathbf{x}}_t)$ that is associated with a detected crop row $R_j\left(F_p^t(\hat{\mathbf{x}}_t)\right)$ but does not yet have a $\text{row}(\hat{z})$ index. This implicitly creates new tracked crop rows, since the index $j + r^*$ might not yet be represented in the set of tracked crop rows. For the first iteration, where the set of tracked crop rows F_m is empty, we skip the data association step and directly assign the index of the detected crop rows. This preprocessing step robustly tracks the detected crop row structure along the field as shown in Figure 5.20.

As a result, we obtain sets of plant feature positions F_p^t, that are assigned to their respective crop row by the mapping row. With this information, we can compute the set of Plant-Rows $\mathcal{R}\left(F_p^t\right)$ at any time step t in our data set as required for our row-based data association. We use these preprocessing steps to obtain sets of Plant-Rows for our experimental evaluation.

5.4.4 Partial Crop Rows

As explained in Section 5.3.3, we use the criteria of Positive and Negative Information in our Score Functions. To correctly apply both criteria in our row-based data association,

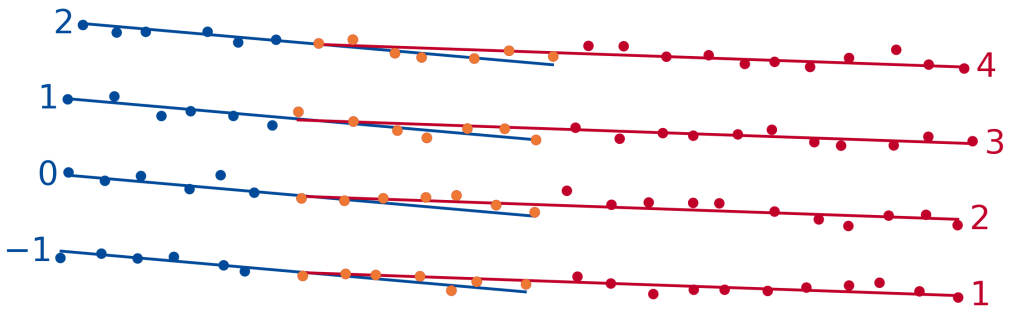

Figure 5.19: This illustrates our key idea for the row tracking data association. The set of detected crop rows F_o with corresponding features is shown as blue points and lines. The set of tracked crop rows F_m and features is visualized in red. The features assigned to both, the detected as well as the tracked crop rows are highlighted in orange. For example, a feature $\hat{z}$ assigned to the detected row 2 is also assigned to the tracked crop row 4, therefore the row-shift $r_{\hat{z}}$ is equal to $4 - 2 = 2$. In this example, all row features $\hat{z} \in F_p^t(\hat{\mathbf{x}}_t) \cap F_p^{t-1}$ define the same row-shift $r_{\hat{z}} = 2$ and therefore r^* returned by our row tracking data association is equal to 2.

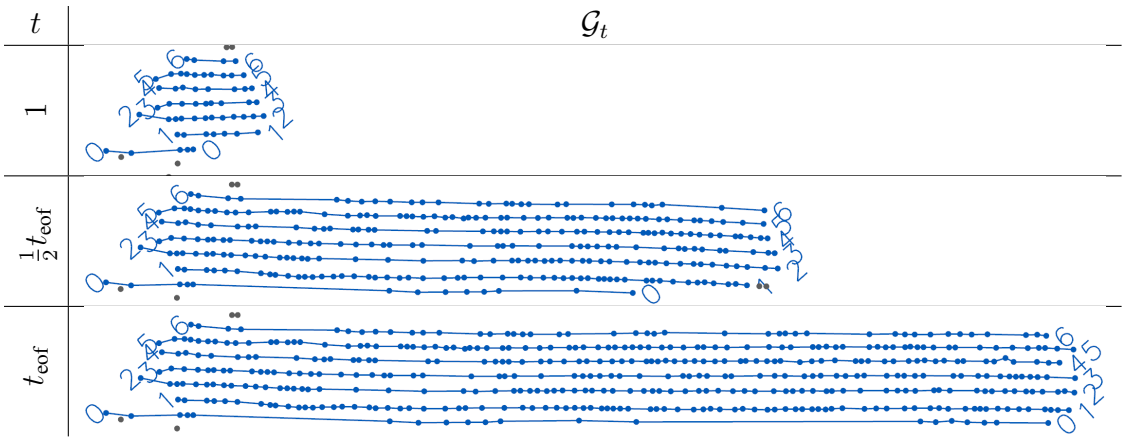

Figure 5.20: This figure shows the extracted row structure during construction of the pose graph for different time steps. The detected plant features are shown as points. Blue features are assigned to a crop row, gray features are unassigned. The crop row structure is visualized by connecting neighboring plant features of the same crop row with lines. The index of each row is printed at the start and end of each row.

we should only consider features as unmatched, if they lie within the detection range of the sensor. We call the detection range *Observed Area* for the set of observed features F_o and *Mapped Area* for the set of mapped features F_m. For the Positive Information criterion, this means that an observed plant feature is only unmatched, if it lies within the Mapped Area. Analogue for the Negative Information, a mapped plant feature only counts as unmatched, if it lies within the Observed Area. In general, this is already ensured by the design of our Score Functions: Given a shift s, we only integrate over the part of the mapped crop rows, that are overlaid with the observed crop rows. Thereby, we ensure that only unmatched mapped features within the Observed Area are considered. The unmatched observed features that are not within the Mapped Area are explicitly handled by our regularization technique (*No Bias* criterion). However, since some of the crop rows are close to the edge of the *field of view* (FoV) of the sensor, they are not always visible in the sensor data. This can lead to crop rows, where only a small amount of the plants is detected. We therefore call such a crop row with many unobserved plant features a *partial crop row*. A good example of a partial crop row is crop row 0 shown in Figure 5.20. During graph construction, this row is only occasionally visible in the sensor data. Therefore, a large part of the plants of this crop row remain undetected causing a sparse feature distribution along this row. These partial crop rows cause problems for a data association algorithm, since the plant feature distribution of such a partial crop row is far from the true feature distribution. Correctly matching this partial crop row with its corresponding crop row therefore yields many unmatched features. This in turn would cause a Score Function based on Positive and Negative Information to diverge from the correct data association. To prevent wrong data associations caused by these partial crop rows, a fourth preprocessing step is required, where partial crop rows are identified and removed from both sets of Plant-Rows $\mathcal{R}(F_o)$ and $\mathcal{R}(F_m)$. The filtered input $\mathcal{R}(F_o)$ and $\mathcal{R}(F_m)$ to our row-based data association algorithm then only contains fully observed and fully mapped crop rows respectively. In the following, we explain how we determine the partial crop rows for a set of Plant-Rows F_p^t with corresponding pose graph $\mathcal{G}_t$ for a time step t. For our experimental evaluation, we apply this procedure to both, the observed as well as the mapped set of Plant-Rows, before passing them as input to our row-based data association.

Given a set of Plant-Rows $\mathcal{R}\left(F_p^t\right)$ for a time step t with corresponding pose graph $\mathcal{G}_t$, we first determine the detection range D_t, which is the area of the FoV of the vehicle, as well as the area $A\left(R_j\left(F_p^t\right)\right)$ of each crop row $R_j\left(F_p^t\right)$ within the set of Plant-Rows $\mathcal{R}\left(F_p^t\right)$. We consider a crop row $R_j\left(F_p^t\right)$ partial, if less than $50\,\%$ of its area $A\left(R_j\left(F_p^t\right)\right)$ lies within the detection range D_t, and therefore if more than $50\,\%$ of its plant features might be undetected.

The detection range D_t of the vehicle until time step t, is the union of the FoV of the sensor at time step t' over all time steps $t' \leq t$. Let $\hat{\mathbf{x}}_{t'} \in \hat{X}_t$ of $\mathcal{G}_t$ be the estimated pose of the vehicle at time step t'. We compute the FoV for this time step t' by projecting the four corners of the detection area A_I of our SEP detection algorithm from the image plane onto the ground plane centered around the robot pose $\hat{\mathbf{x}}_{t'}$. We use the same projection prj as defined in Section 5.4.1 with Equations 5.6–5.8 for the SEP detections. This yields

a polygon defined by the four projected corner points that describes the detection range A_{G} on the ground plane centered around the pose $\hat{\mathbf{x}}_{t'}$. By transforming this polygon with the pose estimate $\hat{\mathbf{x}}_{t'}$ we obtain the detection area $A_{t'}$ for time step t'. Using the CGAL library[2], we compute the union over all these polygons to obtain a polygon, possibly with holes, that describes the detection range D_t of the vehicle at time step t:

$$\begin{aligned} A_I &:= \{c_{\mathrm{lb}}, c_{\mathrm{rb}}, c_{\mathrm{lt}}, c_{\mathrm{rt}}\} \\ A_{\mathrm{G}} &:= \{\mathrm{prj}\left(c_{\mathrm{lb}}\right), \mathrm{prj}\left(c_{\mathrm{rb}}\right), \mathrm{prj}\left(c_{\mathrm{lt}}\right), \mathrm{prj}\left(c_{\mathrm{rt}}\right)\} \\ A_{t'} &:= \{\hat{\mathbf{x}}_{t'} \cdot \mathrm{prj}\left(c_{\mathrm{lb}}\right), \hat{\mathbf{x}}_{t'} \cdot \mathrm{prj}\left(c_{\mathrm{rb}}\right), \hat{\mathbf{x}}_{t'} \cdot \mathrm{prj}\left(c_{\mathrm{lt}}\right), \hat{\mathbf{x}}_{t'} \cdot \mathrm{prj}\left(c_{\mathrm{rt}}\right)\} \\ D_t &:= \bigcup_{t' \leq t} A_{t'} \end{aligned}$$

Regarding the area of a crop row $R_j\left(F_p^t\right) \in \mathcal{R}\left(F_p^t\right)$, we first represent the crop row as a set of connected linear segments with the plant features $\hat{z} \in F_p\left(j\right) \subset F_p^t$ of the crop row as anchors. The area of the row $A\left(R_j\left(F_p^t\right)\right)$ is then defined as a polygon that envelopes these linear segments using a width of $0.3\,\mathrm{m}$. Again, we use CGAL to compute the polygon $A\left(R_j\left(F_p^t\right)\right)$. An example for the envelopes around the crop rows and the detection range is shown in Figure 5.21. Finally, we use both the detection range D_t and the area of the crop row $A\left(R_j\left(F_p^t\right)\right)$ to determine, whether a crop row $R_j\left(F_p^t\right)$ is only partially observed $R_j\left(F_p^t\right) \in \mathcal{R}_{\text{partial}}\left(F_p^t\right)$ or fully observed $R_j\left(F_p^t\right) \in \mathcal{R}_{\text{full}}\left(F_p^t\right)$:

$$\begin{aligned} \mathcal{R}_{\text{partial}}\left(F_p^t\right) &:= \left\{ R_j\left(F_p^t\right) \in \mathcal{R}\left(F_p^t\right) \mid \frac{\left|A\left(R_j\left(F_p^t\right)\right) \cap D_t\right|}{\left|A\left(R_j\left(F_p^t\right)\right)\right|} < 0.5 \right\} \\ \mathcal{R}_{\text{full}}\left(F_p^t\right) &:= \mathcal{R}\left(F_p^t\right) \setminus \mathcal{R}_{\text{partial}}\left(F_p^t\right) \end{aligned}$$

Using this procedure, we determine the subset of fully observed crop rows $\mathcal{R}_{\text{full}}\left(F_p^t\right)$ for any time step t of our data set. During our experimental evaluation, we always pass the subset of fully observed crop rows to our row-based data association algorithm.

In this section, we explained all four steps of our preprocessing pipeline. We use this procedure to extract the row structure of a crop field from our real-world raw data. This entails detecting plant position features by detecting the SEP of different types of crops on raw image data, integrating these detections over time using a pose graph, as well as extracting and tracking the crop row structure on accumulated plant features using our Pattern Hough Detection algorithm from Chapter 3 and a data association technique specifically designed to track the crop rows. Additionally, we also explain how we compute the detection range from integrated raw data to filter crop rows that are not fully observed, i. e., rows that only partially lie within the detection range. This last preprocessing step is crucial to obtain robust results for data association techniques, that are based on the Positive and Negative Information criteria such as our data association approach. For our extensive experimental evaluation in the following section, we use the

[2] `https://www.cgal.org/index.html`, Accessed: 2022-10-31

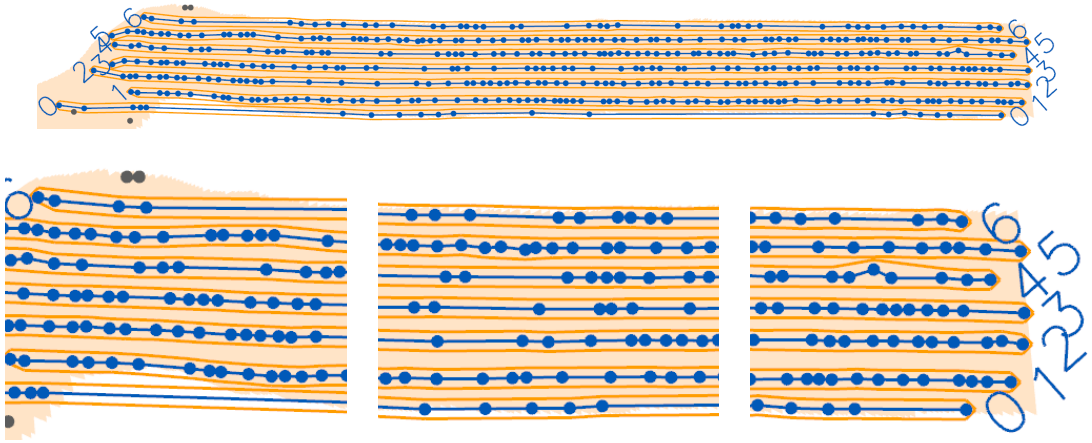

Figure 5.21: This visualizes the detection range (orange) and the envelopes around a set of crop rows (orange outline) for an example from real-world data on the top. Close ups of the same example are shown on the bottom. As already stated in the previous example, crop row 0 is only partially observed: The envelope around crop row 0 is not fully inside the detection range. Since more than $50\,\%$ of the envelope are outside of the detection range, this crop row is considered partial and therefore filtered and not considered in our row-based data association algorithm.

presented preprocessing pipeline to obtain sets of Plant-Rows and pass them as input to our data association algorithm.

5.5 Experimental Evaluation

We develop our row-based data association approach to enable accurate localization and mapping techniques relative to individual plant positions. To show the relevance and applicability of our approach for localization and mapping scenarios on agricultural fields, we design our experimental evaluation towards the following aspects: First, our experimental setup evaluates the investigated data association algorithms within the framework of a localization or mapping application. Second, the performance of the data association is measured with respect to properties that are relevant in navigation applications. This not only includes the robustness in form of a success rate, but also the accuracy of the data association method split into an angular, lateral and longitudinal component. In Section 2.2.3, we explain in more detail how we define the accuracy of a data association method. Third, to highlight the potential of our approach as well as its limitations, we perform our evaluation on challenging real-world input data.

In Section 5.5.1 we present our experimental setup in detail, the GT data association used for the quantitative evaluation, as well as the properties and challenges of our real-world data sets for data association algorithms. In our first experiment we determine the impact of our approach by comparing the performance of our row-based data association with other promising data association techniques inspired from the literature. Here, we

evaluate the success rate as well as angular and translational accuracy. The results show that a row-based approach is indeed required to address individual plant position data association problems. Regarding the expected accuracy during localization or mapping applications, the angular accuracy of our approach is well suited for these kind of problems. However, the overall translational accuracy seems low. Therefore, in the second evaluation, we investigate the translational accuracy of our approach in more detail by splitting it into a lateral and longitudinal component. The results of the second evaluation show, that our approach has sufficient lateral and longitudinal accuracy for localization and mapping applications. In the third evaluation we discuss interesting properties as well as limitations of our approach. We investigate in detail how the performance of our data association approach depends on the amount and quality of input data provided. This information is highly relevant for any real-world localization and mapping application as it influences the decision on when to perform data association. Finally, we show qualitative results of our real-world application that we used as the experimental setup. These give an intuition on the quality that can be obtained when using our approach in real-world localization and mapping applications on crop fields.

5.5.1 Methodology

In this section we first present our experimental setup in the form of a SLAM application. We then illustrate how we obtain a set of ground truth data association matches that we use to quantify the performance of a data association algorithm. Finally, we give a qualitative and quantitative overview of the data sets used for evaluation and discuss multiple interesting and challenging data set properties.

Experimental Setup For our experiments we recorded data on two different fields containing different crop types. The first field is a production field located in Eichstetten near Freiburg. The other field is a sugar beet field of ETH Zurich located at their crop science research station in Eschikon (see Section 2.1.3). We give a detailed overview of both fields and discuss their properties at the end of this section. Using our agricultural robot BoniRob, we collect odometry, IMU, GPS and image data as explained in Section 2.1.2 on both fields. We call the data collected on the field in Eichstetten the *Eichstetten* data and the data collected on the field in Eschikon *Eschikon* data respectively.

For a meaningful evaluation, our first goal is to design a experimental setup that evaluates the investigated data association algorithms on real-world data within a localization or mapping framework. In general, a SLAM application, where the map is built simultaneously to localizing the vehicle, is more challenging than a localization task, where the map is prior knowledge. More precisely SLAM is more challenging for a data association algorithm since some of the observed features might not yet be part of the map. Therefore, not matching an observed feature might be correct in a SLAM application. In contrast, during localization, where all map features are already known, maximizing the number of matches is usually the best strategy. For a SLAM application however, the data association needs to trade-off maximizing the number of matched observations against not

matching observations that do not have a correspondence in the map. Therefore, we evaluate the investigated data association algorithms in the more challenging scenario of a SLAM application.

To this end we process the recorded raw data to iteratively construct a SLAM pose graph

$$\mathcal{G}_t = \left(\hat{X}_t, E_{\hat{X}_t}, \hat{Z}_t, E_{\hat{Z}_t}\right)$$

for every time step t as described in Section 5.4.2. The resulting pose graph contains a set of pose nodes $\hat{X}_t$, a set of unary and binary edges between the pose nodes corresponding to the GPS, odometry and IMU measurements $E_{\hat{X}_t}$, a set of plant position nodes $\hat{Z}_t$ and the corresponding set of binary edges $E_{\hat{Z}_t}$ between the plant position nodes and the pose nodes according to the detections from the image data. A realistic application scenario also requires us to consider the influence of the employed perception method. Thus, we do not rely on manually labeled plant positions during graph construction. Instead, we extract the plant positions from the raw image data using the SEP detection algorithm described in Section 5.4.1.

We start building the pose graph and continue integrating data until the vehicle reaches the end of the first set of crop rows. The graph built so far is defined as the *global graph*

$$\mathcal{G}^{\text{global}} := \left(\hat{X}^{\text{global}}, E_{\hat{X}}^{\text{global}}, \hat{Z}^{\text{global}}, E_{\hat{Z}}^{\text{global}}\right)$$

and contains the already mapped part of the field during our evaluation. We then build a separate *local graph*

$$\mathcal{G}^{\text{local}} := \left(\hat{X}^{\text{local}}, E_{\hat{X}}^{\text{local}}, \hat{Z}^{\text{local}}, E_{\hat{Z}}^{\text{local}}\right)$$

starting from where we stopped constructing the global graph until the vehicle reaches the end of the field for the second time (see Figure 5.22, top). This local graph contains the observed part of the field during our data association evaluation. Whenever the vehicle reaches the end of the field, we merge the local graph into the global graph using ground truth data association information, reset the local graph and restart the procedure until we have processed the whole data set (see Figure 5.22, mid and bottom). We use ground truth information for this merge to ensure an objective evaluation. Using the investigated data association algorithm to perform the merge would make the input data dependent on the investigated algorithm. In the next paragraph we explain, how we obtain the ground truth data association information to merge both graphs.

Given a set of data association matches

$$M \in \mathcal{M} \subset \mathcal{P}\left(\hat{Z}^{\text{local}} \times \hat{Z}^{\text{global}}\right)$$

between the observed plant position features $\hat{Z}^{\text{local}}$ and the mapped plant position features $\hat{Z}^{\text{global}}$, we merge the local graph into the global graph as follows:

$$
\begin{aligned}
\mathcal{G}^{\text{global}} \cup \mathcal{G}^{\text{local}} &:= \left(\hat{X}^{\text{global}} \cup \hat{X}^{\text{local}}, E^{\text{global}}_{\hat{X}} \cup E^{\text{local}}_{\hat{X}}, \hat{Z}^{\text{global}} \cup \hat{Z}^{\text{new}}, E^{\text{global}}_{\hat{Z}} \cup E^{\text{merged}}_{\hat{Z}} \right) \\
\hat{Z}^{\text{new}} &:= \left\{ \hat{z} \in \hat{Z}^{\text{local}} \mid \hat{z} \neq o_i \, \forall \, (o_i, m_i) \in M \right\} \\
E^{\text{merged}}_{\hat{Z}} &:= \left\{ e^{\text{merged}} \left(e^{\text{local}} \right) \mid e^{\text{local}} \in E^{\text{local}}_{\hat{Z}} \right\} \\
e^{\text{local}} &:= (\hat{\mathbf{x}}, \hat{z}, z) \in \hat{X}^{\text{local}} \times \hat{Z}^{\text{local}} \times \mathbb{R}^2 \\
e^{\text{merged}} \left(e^{\text{local}} \right) &:= \begin{cases} (\hat{\mathbf{x}}, m_i, z), & \text{if } \exists \, (o_i, m_i) \in M \colon o_i = \hat{z} \\ (\hat{\mathbf{x}}, \hat{z}, z), & \text{otherwise} \end{cases}
\end{aligned}
$$

First, we copy all pose nodes $\hat{X}^{\text{local}}$ and edges connected only to pose nodes $E^{\text{local}}_{\hat{X}}$ into the global graph. Then, we iterate over all plant position nodes $\hat{z} \in \hat{Z}^{\text{local}}$ of the local graph. If $\hat{z}$ is not matched to any global plant position node, i. e., there is no match $(o_i, m_i) \in M$ so that $\hat{z} = o_i$, we add $\hat{z}$ to the set of plant position nodes $\hat{Z}^{\text{new}}$ and therefore to the plant position nodes of the global graph. We also transfer all local graph edges, that contain this unmatched feature $\hat{z}$ to the global graph. If the plant position node $\hat{z}$ is matched with a mapped plant position node, i. e., there is a match $(o_i, m_i) \in M$ so that $\hat{z} = o_i$, we do not transfer the local plant position node $\hat{z}$ as the feature is already represented in the global graph by feature m_i. We transfer all local graph edges, that contain the matched feature $\hat{z}$ into the global graph, maintaining the pose node $\hat{\mathbf{x}}$ as well as the measurement z for all edges, but exchange the plant position feature node $\hat{z}$ with the corresponding feature node from the map m_i.

Since the local graph is merged into the global graph before each turning maneuver, we count the number of merges by counting the number of turns performed. This procedure iteratively constructs a full SLAM graph over the whole data set. We also use this technique in the last experiment of our evaluation to provide a proof of concept for the applicability of our data association approach in a SLAM use case on real-world data. In that experiment we merge the local graph into the global graph using the set of matches M produced by the data association algorithm instead of ground truth information.

To evaluate our row-based data association approach, where the global graph $\mathcal{G}^{\text{global}}$ provides the mapped features $F_m := \hat{Z}^{\text{global}}$ and the local graph $\mathcal{G}^{\text{local}}$ the observed features $F_o := \hat{Z}^{\text{local}}$, we extract and track the row structure of the plant features during graph construction as described in Section 5.4.3. We also filter out all partial crop rows, i. e., crop rows that are only partially observed, using the procedure described in Section 5.4.4. This results in an observed and mapped set of Plant-Rows, $\mathcal{R}(F_o)$ and $\mathcal{R}(F_m)$ that only contain fully observed crop rows.

While processing the recorded raw data, we create several *data points* that form the *Eichstetten* data set and the *Eschikon* data set respectively as follows: During construction of the local graph we save the state of both, the local and the global graph every time 20 new pose nodes have been added to the local graph. Each pair of local and global graph and their corresponding Set of Plant-Rows then defines a data point of the corresponding data set. Since we add a new pose node after a distance of $0.1\,\text{m}$ has been traversed, this corresponds to creating a data point every $2\,\text{m}$ along each traversed set of crop rows.

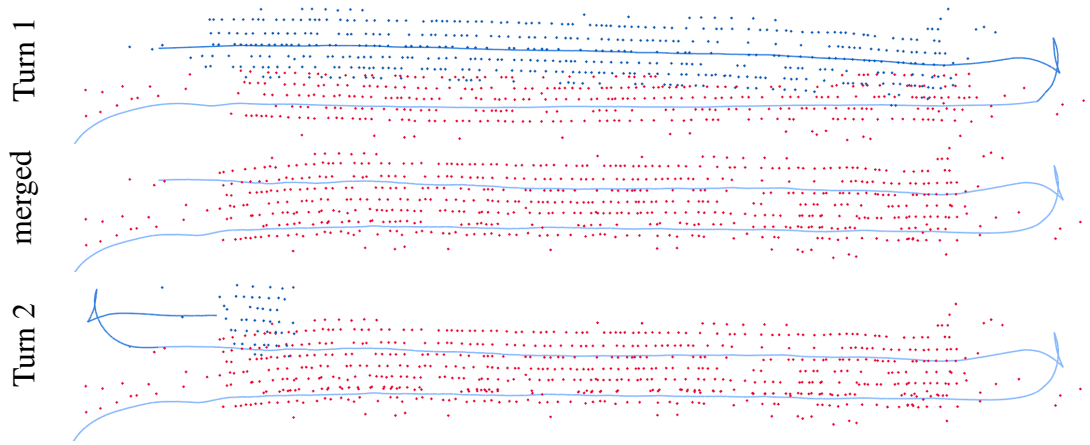

Figure 5.22: This figure shows different steps during local and global graph construction. The feature nodes $\hat{Z}^{\text{global}}$ of the global graph are shown as red dots, the trajectory connecting the pose nodes $\hat{X}^{\text{global}}$ of the global graph is shown as a light blue line. The feature nodes $\hat{Z}^{\text{local}}$ of the local graph are marked as blue dots and the corresponding trajectory is visualized as blue line. The top image shows the local and global graph at the end of turn 1, just before merging. The middle image shows both graphs directly after merging. The local graph is empty and therefore not visible. The bottom image shows both graphs at the beginning of turn 2.

During evaluation the feature nodes of the global graph $\hat{Z}^{\text{global}}$ are used as mapped features F_m with Plant-Rows $\mathcal{R}(F_m)$ and the feature nodes of the local graph $\hat{Z}^{\text{local}}$ are used as observed features F_o with Plant-Rows $\mathcal{R}(F_o)$. The number of data points created for each data set are shown in Table 5.2. Creating multiple data points along each traversed set of crop rows not only enables a more detailed evaluation, but also allows us to analyze data association results with respect to the amount of distance traversed along crop rows.

Ground Truth Data Association For the evaluation of data association algorithms we need to know, whether a certain match between two features, in our case plant positions, is correct, i. e., whether both features describe the same plant. To this end, we manually assign a unique number, called *id*, to each plant on the field. During evaluation, we use these unique numbers to compute a set of GT data association matches M^{GT} for each data point, where a match between two features is correct and part of the GT data association, if the ids of both features are the same. In the following, we first describe how we obtain a unique number for each plant in each image of the data set. Then, we explain how we use those plant ids to compute a set of GT data association matches M^{GT} at each data point. Finally, we explain how we use this set of GT data association matches to quantify the performance of a data association algorithm.

To assign a unique number to each plant, we placed markers into the field as shown in Figure 5.23 in the top left image. With these markers we can identify and assign an id to each plant by its relative position to the marker. We then record one *Marker* data set with these markers distributed in the field and visible in the image data and one *Empty* data set

without any markers for each of the fields. The second Empty data set is necessary since we need marker free images for a realistic evaluation. After recording this data, we first manually label plant position (i. e., the SEP) and id for plants in the lower part of each image in the Marker data set (see Figure 5.23, top right). We only label the lower part of the image, since plants in the upper part appear smaller and the SEP is therefore hard to determine precisely for those plants. This ensures manual labels with high accuracy. Next, we use the labels from the Marker data set to label plant position and id in the Empty data set by manually transferring the id for the plant from the Marker data set onto the same plant in the Empty data set. Here, we use the fact, that we traversed the rows of the field in the same order and at roughly the same velocity for both data sets. Therefore, we can track the plants by replaying both data sets simultaneously image by image. This yields annotated images for the Empty data set where each plant is described by a position in image coordinates (pixel) and a unique plant id (see Figure 5.23, bottom left). Since we evaluate on detections from a plant detection algorithm, i. e., we use the detections of the SEP detection algorithm during graph construction, the last step is to transfer the ids from the manually labeled plant positions to the detections of the algorithm. In this step, it is crucial that we only generate high quality id labels for the detections, as the ids on the detections determine, whether a match is correct during the evaluation. Therefore, we assign ids to detections conservatively, i. e., only if the detection is reasonably close to a manually labeled position. We choose a straight line distance smaller than 60 pixels. This has two effects: First, false positive detections, which are not close to a labeled plant position, do not receive an id and are not considered in our evaluation. Second, plants in the upper part of the image that are detected by the algorithm, but have not been labeled, are also not considered in our evaluation. Both effects, i. e., removing false positives and considering only high accuracy plant positions, ensure that we create only high quality id labels for the evaluation. All detections that are not assigned to a plant receive the id -1. After transferring the ids, we obtain a set of plant detections for each image in the Empty data set, where each detection is annotated with an id, that either is the id of the plant this detection originates from or -1 if no labeled plant is close enough (see Figure 5.23, bottom right). During evaluation we use these ids to determine, whether a match within a set of data association matches M is correct. Given a set of observed and mapped plant features F_o^{id} and F_m^{id}, where features $o \in F_o^{\text{id}}$ and $m \in F_m^{\text{id}}$ are assigned a unique plant $\text{id}\,(o)$ and $\text{id}\,(m)$, we define a ground truth data association algorithm based on these unique plant ids id and call it the *ID data association* (ID). This data association computes a set of data association matches M^{ID} as follows: For each observation $o \in F_o$, we determine the map feature $m \in F_m$, that has the same id as the observed feature o. If such a map feature is found, it is added to the set of matches M^{ID}. If no such map feature can be found, the observation remains unmatched.

$$M^{\text{ID}} := \left\{ \left(o, m^{\text{id}}\right) \in F_o \times F_m \mid \text{id}\,(o) \neq -1 \wedge \text{id}\,(o) = \text{id}\,(m) \right\} \tag{5.9}$$

Since a match between observed and mapped features is only added to M^{ID} if both features have the same plant id and since having the same id implies that the features

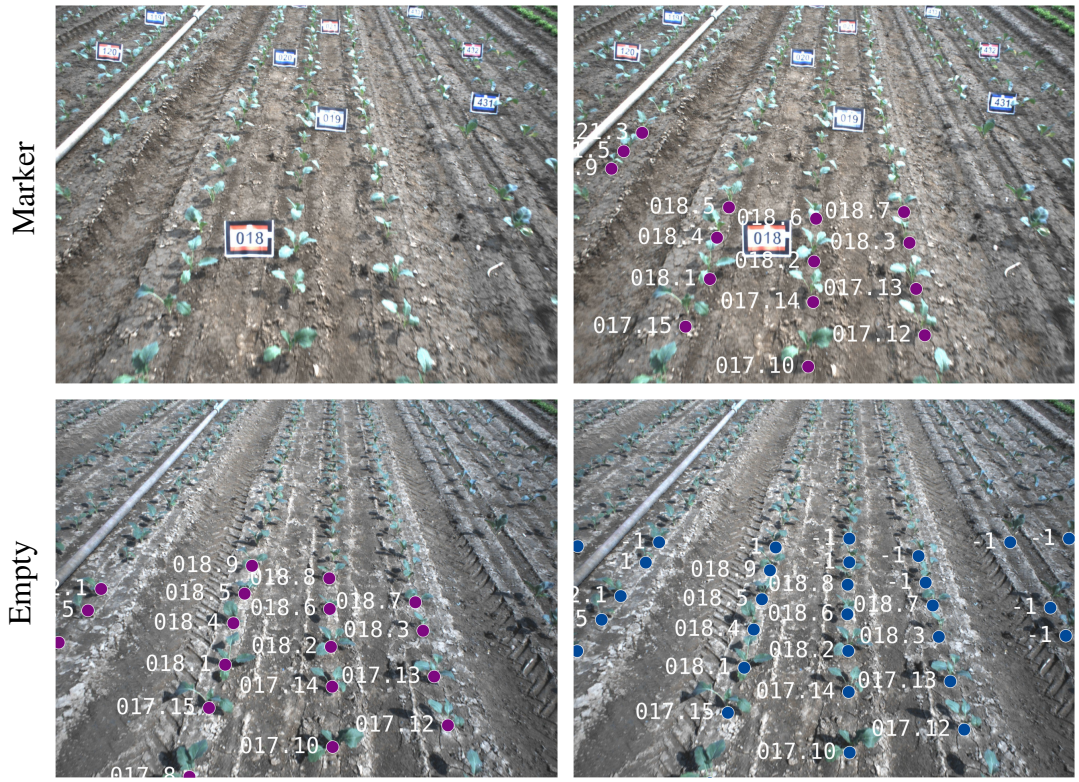

Figure 5.23: This figure shows images from the Marker (top) and Empty (bottom) data set. The markers are used to assign a unique id (white number) to each plant and its labeled position (purple). The ids from the images in the Marker data set are transferred to the plants and their labeled position (purple) in the Empty data set. These are then used to assign ids to the detections (blue) of the plant detection algorithm.

represent the same plant on the field, this ID data association computes the correct data association according to our ground truth labels. We therefore use this ID data association to compute sets of ground truth data association matches $M^{\text{GT}} = M^{\text{ID}}$. However, the ID data association can only match observations, that have an id different from -1. This situation only occurs if either the observation is a false positive detection or labeling the plant position with high accuracy was not possible. In the case of a false positive detection, it is in fact correct to leave the observation unmatched. The case of an unlabeled observation remaining unmatched does not impede the validity of using M^{ID} as ground truth in our evaluation either, since this means that we only evaluate on a smaller subset of GT matches instead of the maximal possible set of GT matches. It rather ensures evaluation results of high quality, since we only evaluate on GT matches that can be inferred with a high accuracy and therefore certainty.

With an id for each plant feature, we can compute the ID data association on the feature nodes of the local and global graph to obtain a set of GT data association matches M^{GT} for each data point. During our evaluation, we use these GT matches to quantify, how well any data association algorithm approximates the GT data association and therefore

the correct data association. Given the set of GT matches M^{GT} with n matched observations and a set of matches M of the investigated data association algorithm we quantify the performance of this data association algorithm by counting the *number of correctly associated observations* $c(M)$ as follows:

$$c(M) := \left|M \cap M^{\text{GT}}\right| \qquad \in [0, n] \tag{5.10}$$

$$c_{\text{ratio}}^{\#>0}(M) := \frac{1}{n} \cdot c(M) \qquad \in [0, 1] \tag{5.11}$$

In the following experiments, we use the number of correctly associated observations c to quantify the performance of the investigated data association algorithms. Since the number n of GT data associations matches in M^{GT} is different for each data point, we also use its ratio $c_{\text{ratio}}^{\#>0}$, where we normalize with the number of GT matches in M^{GT}. This ensures comparability of the performance throughout all data points.

Data Set Overview and Properties We give an overview of the structure of both fields as well as an intuition about the quality of the input and ground truth data for both data sets. For a ground truth visualization of both fields we need to obtain ground truth positions of the crops as well as a ground truth trajectory of the vehicle. To this end, we process the recorded data using a traditional SLAM technique, where we build one pose graph by continuously integrating data into the graph. The input data is processed as described in Section 5.4.2 but instead of the Nearest Neighbor data association we use the ID data association as defined by Eq. (5.9) to associate the plant position measurements $z \in Z_t$ from an image at time step I_t with the already existing feature nodes of the graph $\hat{Z}_{t-1}$. Here, we use the feature nodes $\hat{Z}_{t-1}$ of the graph built so far as mapped features F_m, where we obtain ids for each features using the feature edges as described above. The observed features F_o are the plant positions measurements Z_t and their ids are the corresponding id of the plant. For the same reasons as mentioned above, we obtain ground truth data association matches $M^{\text{ID}} = M^{\text{GT}}$ using the ID data association. The resulting pose graph therefore computes the ground truth map of plant positions and the ground truth vehicle trajectory that can be obtained using a traditional pose-graph-based SLAM technique. Thus, we call such a pose graph obtained using the ID data association the *GT pose graph* $\mathcal{G}^{\text{GT}}$. For an overview over the quality of our *manual* plant position and id labels that we use to define the GT data association during our evaluation, we compute a *manual* GT pose graph $\mathcal{G}^{\text{GT, man}}$, where we use the manually labeled plant positions and ids as input during graph construction. For a qualitative comparison between the manually labeled plant positions and the plant positions detected by the SEP detection algorithm, we also compute a *detections* GT pose graph $\mathcal{G}^{\text{GT, det}}$, where we use the detected SEP positions and the transferred id labels. The detections GT pose graph $\mathcal{G}^{\text{GT, det}}$ shows all detections of the SEP detection algorithm that are used as an input during construction of the local graph construction. In other words, all plant position features visible in this graph have to be associated by the data association algorithms during our evaluation. Of course, this includes the detections with an id equal to -1, i. e., detections that either are

	width	row length	# rows	# Turns	# data points
Eichstetten	13 m	90 m	22	6	272
Eschikon	15 m	35 m	30	9	164

Table 5.2: Data set statistics for both fields.

false positive detections or correspond to plants located in an unlabeled area of the image. However, the set of GT matches M^{GT} only contains observed features with an id not equal to -1, i. e., true positive, labeled detections. To also give an intuition about the subset of observations on which we perform our quantitative evaluation, i. e., the observations contained in M^{GT}, we show a second visualization of the detections GT pose graph $\mathcal{G}^{\text{GT, det}}$, where we only draw the plant position feature nodes that have an id other than -1. We visualize all three types of pose graphs in Figure 5.24 for the Eichstetten data set and in Figure 5.25 for the Eschikon data set. Using these GT pose graphs, we compute the dimensions of both fields and also give a summary of other interesting data set dimensions in Table 5.2.

The Eichstetten data set has two interesting properties: First, the spacing between crop rows is not always the same. Every three crop rows the spacing between adjacent rows is slightly larger. This irregular crop row spacing decreases the amount of consistent crop row matches considered during data association and therefore makes finding the correct crop row matches easier on the Eichstetten data set (see Section 5.3.2). Second, as highlighted by the different colors overlaid on the field in Figure 5.24, the Eichstetten data set contains three different kinds of crop: Kohlrabi, Chinese Cabbage and Sweetheart Cabbage. The crop types even change mid row. As we discuss in Section 5.4.1, the detection algorithm used to create the input data for our evaluation has varying accuracy and false positive rate depending on the type of crop. Therefore, during evaluation, we can directly infer from the results, if and how the difference in perception performance affects the results of the data association algorithm. Both properties make Eichstetten an interesting data set.

Another noteworthy observation is that the last half of the last turn does not have any labeled plant positions. We did not put markers in that part of the field, and therefore ids could not be assigned to these plants. However, we still apply the detection algorithm to evaluate on the full data set. Since there are no labeled plant positions for this part of the field, the detections of the algorithm are assigned the id -1. By definition of the set of GT matches M^{GT} in Eq. (5.9), detections with an id of -1 are not included in M^{GT}. Therefore, these additional detections of the last half of the last turn are not considered during our qualitative evaluation. As we discussed earlier, this does not invalidate our evaluation, as we are only evaluating on a smaller subset of the full set of GT matches.

Looking at the manual GT graph $\mathcal{G}^{\text{GT, man}}$, we see very clear individual plant features throughout the map. This is caused by the fact that all manually labeled plant positions have an id, and therefore all plant position nodes are properly associated in the graph. In contrast to the manual GT graph $\mathcal{G}^{\text{GT, man}}$ that has no duplicate plant position nodes, the detections GT graph $\mathcal{G}^{\text{GT, det}}$ has many duplicate plant position nodes. This is to be

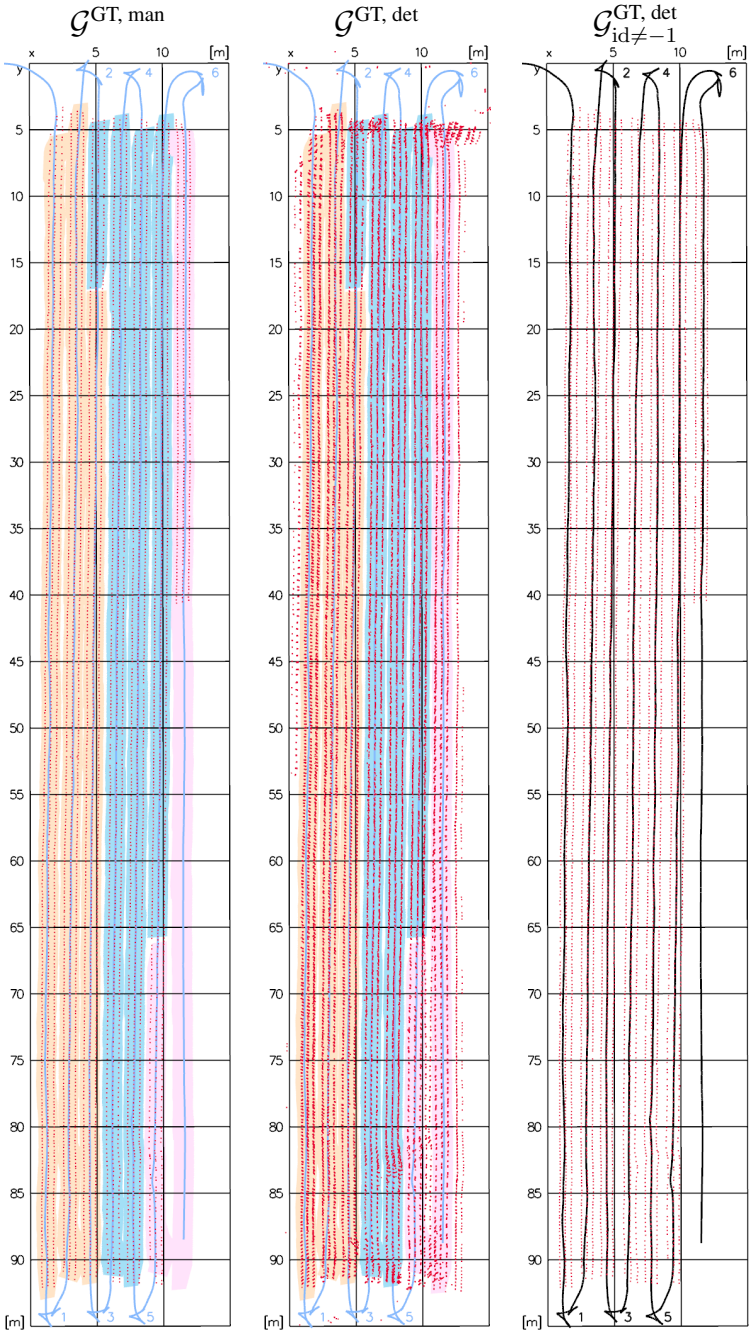

Figure 5.24: This figure shows the manual $\mathcal{G}^{\text{GT, man}}$ and detections $\mathcal{G}^{\text{GT, det}}$ GT pose graphs on the **Eichstetten** data set. The estimated plant positions $\hat{Z}$ are shown as red dots. The estimated trajectory of the vehicle is shown as blue or black line connecting consecutive pose nodes ($\hat{X}$). The number of each turn is shown next to the turn. We use different colors to illustrate the location of different crop types throughout the field: Kohlrabi in orange, Chinese Cabbage in blue and Sweetheart Cabbage in rose. On the right we omit the colors that indicate the crop type, so that the plant positions are more clearly visible.

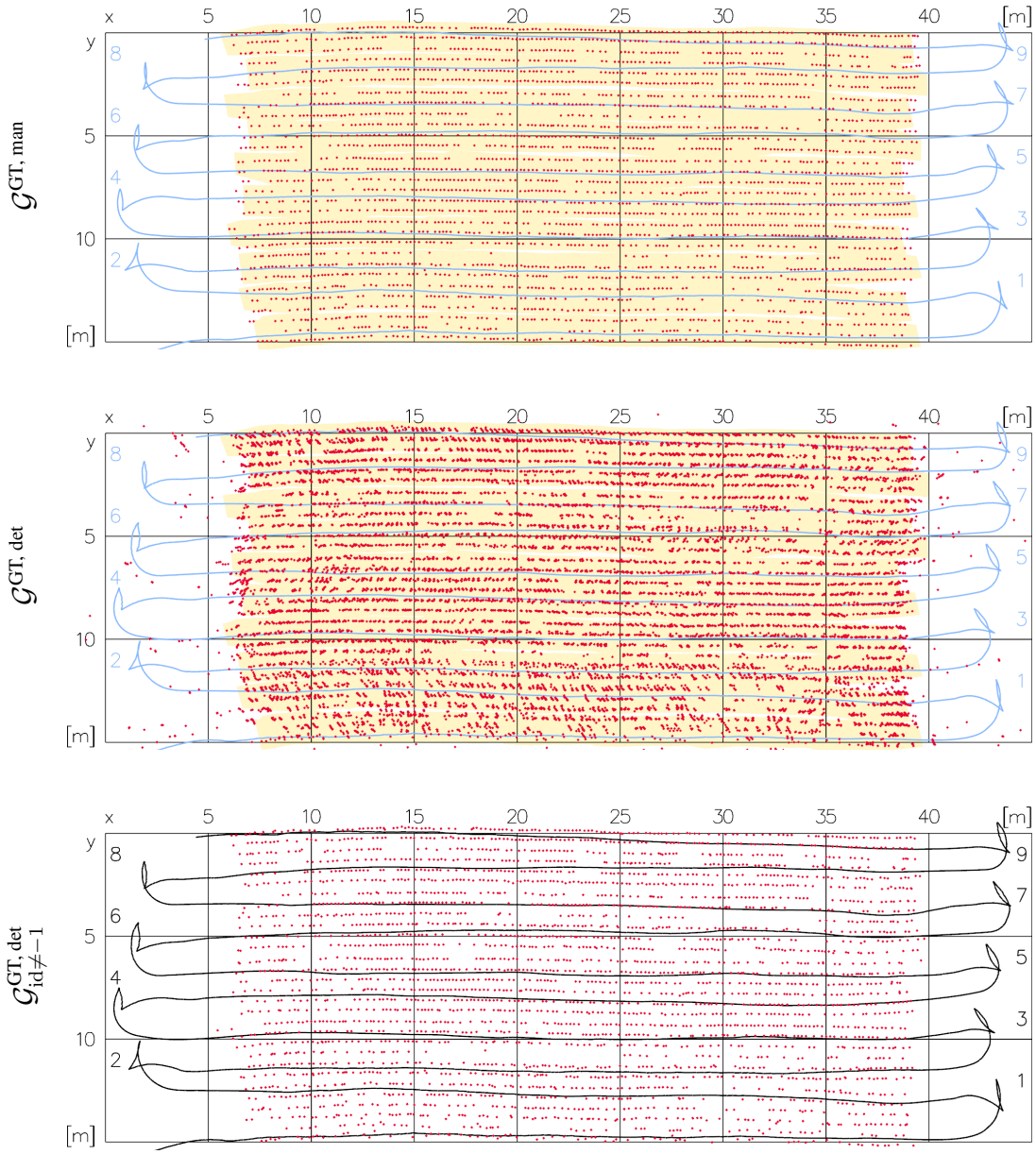

Figure 5.25: This figure shows the manual $\mathcal{G}^{\text{GT, man}}$ and detections $\mathcal{G}^{\text{GT, det}}$ GT pose graphs on the **Eschikon** data set. The estimated plant positions $\hat{Z}$ are shown as red dots. The estimated trajectory of the vehicle is shown as blue or black line connecting consecutive pose nodes ($\hat{X}$). The number of each turn is shown next to the turn. This field only contains Sugar Beet highlighted in yellow. On the bottom we omit the colors that indicate the crop type, so that the plant positions are more clearly visible.

expected, since not all detections received a plant id and therefore some detections remain unassociated. This graph gives a good overview of the quality of the input data, including the false positive and unlabeled detections that need to be associated by the investigated algorithms during our evaluation. To give an intuition about the subset of detections on which we perform our quantitative evaluation, i. e., the set of GT matches M^{GT}, we also show the detections GT graph containing only plant position nodes that are part of the set of detections GT matches on the right of Figure 5.24. We see that the resulting graph closely resembles the manual GT graph $\mathcal{G}^{\text{GT, man}}$. Therefore, we can conclude, that we indeed created a high quality set of GT matches M^{GT} for the detections by transferring the manual labels.

The graphs of the Eschikon data set in Figure 5.25 show that this field has contrasting properties to the Eichstetten crop field: The spacing between the crop rows is uniform, so that resolving ambiguity and associating the correct crop rows is more challenging on this field. Also, the Eschikon data set only features one crop type, the Sugar Beet. However, as explained in Section 5.4.1, the perception quality on this kind of crop is lower compared to the other three crop types. Therefore, the Eschikon data set is overall more challenging than the Eichstetten data set.

Similar to the manual GT graph $\mathcal{G}^{\text{GT, man}}$ of Eichstetten, the manual GT graph $\mathcal{G}^{\text{GT, man}}$ of Eschikon shows clear individual plant features and crop row structure. This means, that the ids were transferred correctly from the Marker data set to the Empty data set. Comparing the detections GT graph $\mathcal{G}^{\text{GT, det}}$ of Eschikon with the one from Eichstetten, the detections appear more spread out in the Eschikon data set, making it harder to distinguish adjacent crop rows in some parts of the field. This confirms the notably lower detection accuracy of the detection algorithm on Sugar Beet compared to the other three crop types. However, comparing the detections GT graph $\mathcal{G}^{\text{GT, det}}$ to the graph $\mathcal{G}^{\text{GT, det}}_{\text{id}\neq -1}$, that only shows the detections with ids, i. e., the features included in the set of GT matches M^{GT}, we again see a clear row structure with individual plant features. Furthermore, the graph closely resembles the manual GT graph $\mathcal{G}^{\text{GT, man}}$, confirming also for the Eschikon data set, that we have a high quality set of GT matches M^{GT} on which we perform our quantitative evaluation.

In the following evaluation we use the data points from the Eichstetten and the Eschikon data set to evaluate the performance and properties of our data association algorithm as well as comparison algorithms. We always perform data association on the detections of the perception algorithm. We evaluate the results of all data association algorithms with respect to the set of GT matches M^{GT}. The Eichstetten data set poses an interesting challenge with its three different crop types. The Eschikon data set is overall more challenging due to its uniform crop row spacing and the low performance of the perception algorithm on its crops. In the next section we perform our first evaluation, where we compare the performance of our row-based data association approach with the results of other data association algorithms to confirm that our row-based data association approach is indeed required in this kind of scenario.

5.5.2 Impact of Row-Based Data Association

In this experimental evaluation we investigate, whether leveraging information about the row structure of the crop field facilitates robust data association on individual plant positions. To this end, we compare the performance of our row-based data association with other data association approaches that do not use information about the row structure of the field. Therefore, we implemented the following data association approaches that do not utilize the row structure:

- *Nearest Neighbor (NN)* data association
- *Geometric Consistency Branch and Bound (GCBB)* data association
- *Gap Descriptor (Gaps)* data association

Since these data association techniques do not require row structure information, the input to these is the set of plant positions $\hat{Z}^{\text{global}}$ of the global graph for the mapped features F_m and the set of plant positions $\hat{Z}^{\text{local}}$ of the local graph for the observed features F_o. The output is a set of data association matches $M \in \mathcal{M} \subset \mathcal{P}\left(F_o \times F_m\right)$ that contains pairwise matches between observed and mapped features.

Nearest Neighbor The *Nearest Neighbor (NN)* data association is a widely used technique. As the name suggests, each observed feature is associated with its closest, and therefore nearest, feature in the mapped set of features. This strategy strongly relies on the assumption that the pose estimate is accurate, because only then are the observed features close to their corresponding mapped features. To increase the robustness of the NN data association, a match between mapped and observed feature is rejected and the observation remains unmatched, if the distance to the closest mapped feature is larger than a given threshold. Another key assumption of the NN data association technique is independence between the features in each set. This independence then allows for an efficient computation of each match individually. The NN data association is introduced in more detail in Section 2.2.2. For our implementation, we use the straight line, i. e., Euclidean, distance between observed and mapped features and reject matches with a distance larger than $0.15\,\text{m}$. The set of NN data association matches M^{NN} is therefore computed as:

$$\begin{aligned} M^{\text{NN}} &:= \left\{ \left(o, m^{\text{NN}}\left(o\right)\right) \in F_o \times F_m \mid \left\| o - m^{\text{NN}}\left(o\right) \right\| < 0.15 \right\} \\ m^{\text{NN}}\left(o\right) &:= \underset{m \in F_m}{\operatorname{argmin}} \left\| o - m \right\| \end{aligned}$$

We compute the set of NN data association matches M^{NN} for all data points on both data sets for our evaluation.

Geometric Consistency Branch and Bound This data association approach is inspired by the *Joint Compatibility Branch and Bound (JCBB)* algorithm presented by Neira and

Tardos [2001]. The JCBB algorithm is a probabilistic approach that explicitly uses information about the dependency between features. This approach defines the Target function as the number of matches, i. e., observations matched to map features. Therefore this technique solely relies on Positive Information when comparing different sets of data association matches $M \in \mathcal{M}$, called *data association hypotheses* $\mathcal{H} = \mathcal{M} \subset \mathcal{P}\left(F_o \times F_m\right)$ in the JCBB approach. The best data association hypothesis $H^* \in \mathcal{H}$ according to the JCBB technique is thus the one that maximizes the number of matches. However, the key idea of this approach does not lie in the design of the Target function. Instead, the authors of the JCBB algorithm focus on efficiently traversing the space of possible data association hypotheses $\mathcal{H}$ with a branch and bound technique. During this tree search, an entire branch might be skipped for one of the following reasons: First, adding a match to a data association hypothesis would make it fail the *joint compatibility test*. Second, not matching an observation will always lead to a data association hypothesis with less number of matches than the currently best hypothesis. For a more in-depth introduction of the JCBB algorithm refer to Section 2.2.2.

Since we do not have the required probabilistic information to compute the *joint compatibility test* in our setting, we apply a *geometric consistency test* instead, similar to the idea of the *Geometric Consistency Branch and Bound* algorithm presented by Neira et al. [2003]. We call a set of data association matches $M = \{(o_i, m_i)\}_{i \in [1,n]} \in \mathcal{M}$ *geometrically consistent*, if it passes our *geometric consistency test* gc, i. e., if it belongs to the subset of geometrically consistent data association matches $\mathrm{gc}\left(\mathcal{M}\right) \subset \mathcal{M}$:

$$
\begin{aligned}
\mathrm{gc}\left(\mathcal{M}\right) &:= \left\{M \in \mathcal{M} \mid \mathrm{gc}\left(\left(o_i, m_i\right), M_i\right) \forall i \in [1,n]\right\} \\
M_i &:= M \setminus \{(o_i, m_i)\} \\
\mathrm{gc}\left(\left(o_i, m_i\right), M_i\right) &:= \begin{cases} \|o_i - m_i\| < 3.0\,\mathrm{m} \wedge \\ \left|\|o_i - o_j\| - \|m_i - m_j\|\right| < 0.15\,\mathrm{m} \quad \forall j \neq i \in [1,n] \end{cases}
\end{aligned}
$$

Our geometric consistency test gc consists of two parts. First, due to the large amount of features, we use the locality requirement and constrain the absolute distance between matched point features. Analogously to the locality threshold for the long-shift parameter l used for our row-based data association we use a threshold of $3.0\,\mathrm{m}$ to that the matched features are close enough to each other. Second, we test for geometrically consistent matches using the binary constraint that the relative distance between matched feature pairs should be similar. Since this binary constraint has to be true for all pairwise matches, this implicitly ensures that the matched point features form similar geometric patterns. For a better intuition consider the following example: Given two sets of three points both sets form two congruent triangles and can therefore be overlaid, only if their relative distances are the same. As the similarity threshold we use $0.15\,\mathrm{m}$ analogue to the NN data association threshold. Using the relative distance between point features to test for geometric consistency between sets of point features is suggested by Neira et al. [2003].

Due to the iterative definition of the geometric consistency test gc, we can iteratively determine, whether adding a match (o_i, m_i) to a set of geometrically consistent data association matches $M = \{(o_j, m_j)\}_{j \in [1,i-1]} \in \mathrm{gc}\left(\mathcal{M}\right)$ preserves the geometrical consistency,

i. e., whether $M \cup \{(o_i, m_i)\} \in \mathrm{gc}(\mathcal{M})$. This iterative definition is required to efficiently search the set of all possible data association matches $\mathcal{M}$. During the tree search as presented in the JCBB technique, only branches yielding geometrically consistent data association matches are explored. Therefore, the set of valid data association matches $\mathcal{M}^{\text{valid}}$ is defined by the geometrically consistent data association matches $\mathcal{M}^{\text{valid}} := \mathrm{gc}(\mathcal{M})$ in this data association approach.

We use the Target function and branch and bound search as described in the publication by Neira et al. [2003] for our implementation of the *Geometric Consistency Branch and Bound (GCBB)* algorithm. Due to the large number of features in our setting, and the computational complexity of the approach (exponential in number of features), we stop the search after 300 sec and use the best set of data association matches found so far in our evaluation. The matches of the GCBB algorithm M^{GCBB} are therefore defined as:

$$M^{\text{GCBB}} := \underset{M \in \mathrm{gc}(\mathcal{M})}{\operatorname{argmax}} |M|$$

We apply the GCBB algorithm on both data sets to obtain a set of M^{GCBB} data association matches for each data point. We use these matches M^{GCBB} to evaluate the performance of the GCBB algorithm.

Gap Descriptor The *Gap Descriptor (Gaps)* data association is inspired by Chebrolu et al. [2018], which presents a descriptor-based data association approach for registering UAV images of crop fields. The main idea is to use gaps, i. e., locations within the crop rows where plants are missing, as features instead of plant positions. Therefore, this Gaps data association computes data association matches $M^{\text{Gaps on Gaps}}$ between a set of observed gap features F_o^{Gaps} and a set of mapped gap features F_m^{Gaps}. Given the sets of gap features the authors first compute a gap descriptor d_g for each gap feature g based on geometric relations to its k closest gap features. For each gap feature g the values of the descriptor d_g consist of $k-1$ relative distances δ_g and $k-1$ angles α_g between the gap feature g and its k neighbors. These descriptor values are designed to be rotation, translation and scale invariant to be robust against different UAV poses (orientation and position including height) during data recording as follows: Given a gap g its *local k-neighborhood of gaps* is defined as the k closest gaps $\{g_1, \ldots, g_k\}$. Let g_k be the gap farthest away from g and the other elements $\{g_i\}_{i \in [1,k-1]}$ be sorted by their angle $\angle g_k, g, g_i$ in ascending order. Then, the descriptor values $d_g = (\alpha_g, \delta_g) \in \mathbb{R}^{2(k-1)}$ are defined on the k-neighborhood as follows:

$$\begin{aligned} (\alpha_g)_i &= \frac{\angle g_k, g, g_i}{2 \cdot \pi}, \forall i \in [1, k-1] \\ (\delta_g)_i &= \frac{\|g - g_i\|}{\|g - g_k\|}, \forall i \in [1, k-1] \end{aligned}$$

The authors define the distance between two descriptors as the Euclidean L_2-Norm over the descriptor values. This distance is used in conjunction with a threshold to restrict

the number of considered data association hypotheses as well as the Target function to determine the best data association hypothesis.

The data association approach presented by Chebrolu et al. [2018] is divided into three steps. First, a set of *valid* matches

$$M_{\text{valid}}^{\text{Gaps on Gaps}} \in \mathcal{M}^{\text{Gaps on Gaps}} \subset \mathcal{P}\left(F_o^{\text{Gaps}} \times F_m^{\text{Gaps}}\right)$$

between observed gap features F_o^{Gaps} and mapped gap features F_m^{Gaps} is computed according to the following criteria:

$$\begin{aligned}
M_{\text{valid}}^{\text{Gaps on Gaps}} &:= \left\{(o, m^*) \in F_o^{\text{Gaps}} \times F_m^{\text{Gaps}} \mid \text{valid}\,(o, m^*)\right\} \\
m^* &:= \underset{m \in F_m^{\text{Gaps}}}{\text{argmin}} \|d_o - d_m\| \\
m^{**} &:= \underset{m \in F_m^{\text{Gaps}} \setminus \{m^*\}}{\text{argmin}} \|d_o - d_m\| \\
\text{valid}\,(o, m^*) &:= \begin{cases} \|d_o - d_{m^*}\| < \varepsilon \wedge \\ \frac{\|d_o - d_{m^*}\|}{\|d_o - d_{m^{**}}\|} \leq 0.8 \end{cases}
\end{aligned}$$

Only matches where the descriptor distance is smaller than a given threshold ε are considered, which ensures that only gaps with similar descriptor values, and therefore similar local geometric structure, are matched. Also, to avoid ambiguous matches the authors compute the distance ratio between the two best matches m^* and m^{**} that have the smallest and second smallest descriptor distance. Only if the ratio is smaller or equal to 0.8, i. e., the second smallest distance value is considerably larger than the smallest distance value, the best match is considered unambiguous and therefore added to the set of valid matches $M_{\text{valid}}^{\text{Gaps on Gaps}}$. This first step yields a set of high quality gap matches $M_{\text{valid}}^{\text{Gaps on Gaps}}$. Second, to reject outliers, the authors compute similarity transforms in a RANSAC loop on the set of candidate matches $M_{\text{valid}}^{\text{Gaps on Gaps}}$ resulting in a set of data association matches $M^{\text{Gaps on Gaps}} \subset M_{\text{valid}}^{\text{Gaps on Gaps}}$ on gap features with corresponding two-dimensional similarity transform $T_{M^{\text{Gaps on Gaps}}}$. In the last step, the authors refine the solution $M^{\text{Gaps on Gaps}}$ from the first two steps using the Hungarian Method to recover more matches from the matches discarded in the previous steps.

For our implementation, we first extract two sets of gaps F_o^{Gaps} and F_m^{Gaps} from our Plant-Row data structure as follows: We iterate over the plant positions that are sorted in ascending order along the row. If the distance between consecutive plant positions is larger than a given threshold of $0.5\,\text{m}$, we add a gap in between these two positions. We perform this procedure on each Plant-Row on both, the mapped and observed data, and get two sets of gaps F_o^{Gaps} and F_m^{Gaps} respectively. As threshold ε for the descriptor distance, we empirically determined 0.1. In contrast to the paper, in the second step, we do not sample in a RANSAC loop, but rather test each pairwise subset from the set of high quality matches $M_{\text{valid}}^{\text{Gaps on Gaps}}$, since two matches suffice to compute a similarity transform to test on. We compute the similarity transform $T_{M^{\text{Gaps on Gaps}}}$ using the procedure described in Section 2.2.3 for the computation of the *inferred pose correction* transforms. We do

not perform the third step to refine the result by recovering more matches. Since we are interested in matches between plant positions M^{Gaps}, we use the similarity transform $T_{M^{\text{Gaps on Gaps}}}$ obtained at the end of the second step to instead recover matches between plant positions $M^{\text{Gaps}} \in \mathcal{M} \subset \mathcal{P}\left(F_o \times F_m\right)$ from the matches between gaps $M^{\text{Gaps on Gaps}} \in \mathcal{M}^{\text{Gaps on Gaps}} \subset \mathcal{P}\left(F_o^{\text{Gaps}} \times F_m^{\text{Gaps}}\right)$. Here, we apply a similar technique as described in Section 5.3 to recover plant position matches from our row-based data association: We correct the position of the observed plant features using transform $T_{M^{\text{Gaps on Gaps}}}$. Then, we perform the NN data association as described above on the transformed observations to compute a set of data association matches M^{Gaps} on plant positions. We use this set of matches M^{Gaps} as the result of the Gap Descriptor data association algorithm in our evaluation.

In this first experimental evaluation, our goal is to compare the performance of different non row-based data association algorithms with our row-based data association approach. Since we expect the IoU Score Function to perform better than the other three Score Functions, we equip our row-based data association with the IoU Score Function for this evaluation and call the matches returned by our approach M^{our}. Although each data association determines the best set of data association matches M^* according to its respective Target function, they are all designed to find the correct data association. As explained in Section 5.5.1, our GT data association matches M^{GT} obtained from the manually labeled ground truth plant ids is a subset of the correct data association. We can therefore use this set of GT data association matches M^{GT} to objectively compare the performance of the investigated data association algorithms. To this end, we compute the number of correctly associated observations $c\left(M\right)$ for the matches M of the different data association algorithms over all data points for the Eichstetten and Eschikon data set (see Eq. (5.10)). Since the number $n = \left|M^{\text{GT}}\right|$ of GT matches changes for each data point, we compute the percentage of correctly associated observations using $c_{\text{ratio}}^{\#>0}$ for comparable results over all data points (see Eq. (5.11)). In Table 5.3, we present the number of data points, where at least one correct match was found (first column, $c > 0$ [#]) and the average amount of correct matches found (second column, $c_{\text{ratio}}^{\#>0}$ [%]). Since most algorithms only rarely find any correct matches, we compute the average amount only over the data points, where at least one correct match was found.

We also evaluate the angular and translational accuracy of a data association algorithm that can be expected during a localization or mapping application as defined in Section 2.2.3: Both applications, localization and mapping, use the data association matches M to correct the pose estimate of the vehicle. Therefore, we can compute the *inferred pose correction* transform T_M for the evaluated data association algorithm as well as a transform $T_{M^{\text{GT}}}$ based on the GT matches M^{GT}. The angular $\Delta\theta$ and translational Δt accuracy of a data association algorithm are then defined as the transform parameters of the relative transform ΔT between both inferred pose correction transforms T_M and $T_{M^{\text{GT}}}$:

$$\begin{aligned} \Delta T &= (\Delta\theta, \Delta t) \\ \Delta T &:= T_M \cdot T_{M^{\text{GT}}}^{-1} \end{aligned}$$

We evaluate the accuracy on all data points for both data sets and show the average

	Eichstetten				Eschikon			
	$c > 0$	$c_{\text{ratio}}^{\#>0}$	$\Delta\theta$	Δt	$c > 0$	$c_{\text{ratio}}^{\#>0}$	$\Delta\theta$	Δt
	[#]	[%]	[°]	[m]	[#]	[%]	[°]	[m]
NN	16/272	1.06	0.73	1.14	2/164	14.58	1.98	1.93
GCBB	1/272	1.49	9.43	7.18	0/164	0.00	5.84	3.51
Gaps	1/272	0.41	46.64	46.79	0/164	0.00	90.05	32.24
our	204/272	63.57	0.11	0.70	113/164	61.91	0.60	0.70

Table 5.3: This table shows the results of our evaluation. The first column ($c > 0$ [#]) lists the number of times at least one correct match was found out of the whole number of data points (272, 164). The second column presents the average number of correct matches in percent ($c_{\text{ratio}}^{\#>0}$ [%]). The average is computed over all data points where the corresponding data association found at least one correct match. The third column ($\Delta\theta$ [°]) and fourth column (Δt [m]) show the average angular and translational error compared to the transform inferred from the GT matches over all data points.

angular ($\Delta\theta$) and translational (Δt) accuracy in the third and fourth columns of Table 5.3 respectively.

The results show that our data association approach is well suited to find good data associations as it often finds GT matches, i. e., 204 of 272 times on the Eichstetten data set and 113 of 164 times on the Eschikon data set. Additionally, out of all investigated data association algorithms, it finds the most GT matches with around $60\,\%$ on average. However, the other data association approaches that do not use the row structure information, almost never find any GT data association matches. The reason for this is that these approaches are not designed for the specific challenges of this environment, where the features are densely and almost uniformly distributed.

The Gaps data association, which was designed for agricultural environments, cannot find the correct data association in our evaluation. This can be explained by the different target application in the work by Chebrolu et al. [2018] of registering images captured by a UAV flying over the field. Here, the focus is on designing a rotation, translation and scale invariant feature descriptor to robustly match gap features in UAV images. This invariance is important since the UAV is flying freely above the field and thus perceives the same part of the field from many different orientations and positions at different heights. However, in our scenario, where the trajectory of the ground vehicle is highly restricted by the crop row structure of the field and the height of the ground vehicle also does not change, these invariances are less important in our scenario. Additionally, the portion of the field observed in one image is magnitudes smaller in ground vehicle images than in the images recorded by a UAV. Therefore, a much smaller amount of gaps can be observed with a ground vehicle, which additionally decreases the performance of a gap-based descriptor. The design towards invariance also explains the high angular and lateral errors of this approach. Since, by design, large changes in rotation between images are ignored, the Gaps data association can find matches as shown in *Situation 1* in Figure 5.26, where the data association infers a large rotation of the observed features to match them onto the

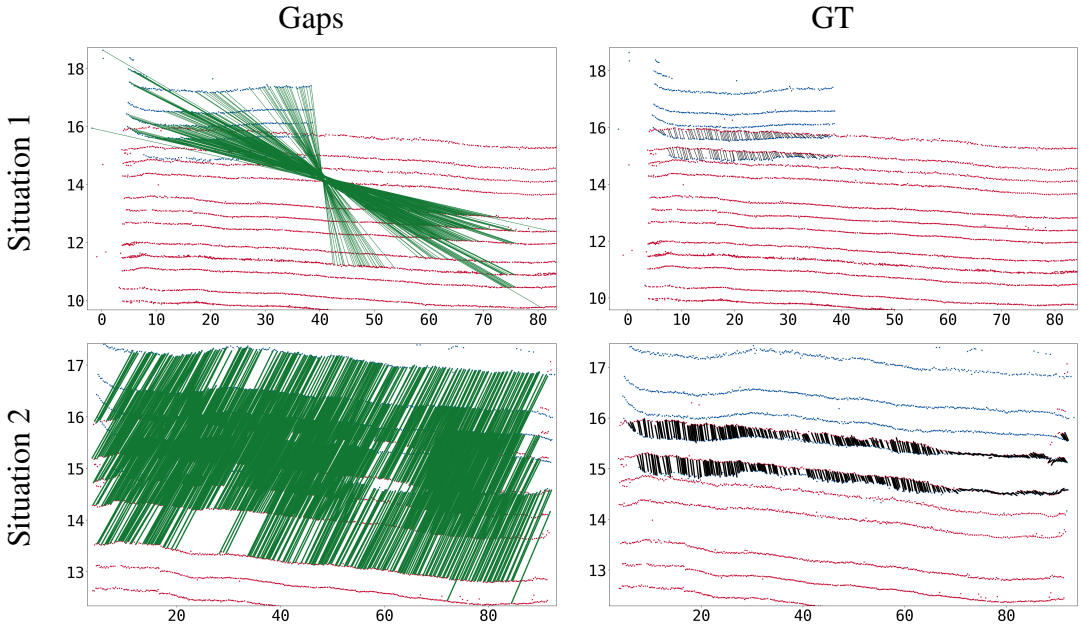

Figure 5.26: These images illustrate two example situations for the Gaps data association. Both situations visualize results taken from two data points of the Eichstetten data set. The mapped and observed features F_m and F_o are shown as red and blue dots respectively. Matches between features are highlighted using a solid line connecting the paired features. On the left, the matches M^{Gaps} found by the Gaps data association are shown in green. For comparison, the GT matches M^{GT} for this data point are shown in black on the right.

mapped features. This results in an angular error of $179.97°$ for this data point. However, since the Gap descriptor uses information about relative geometry in form of relative angles and distances, it can sometimes find the correct angular orientation as shown in *Situation 2* in Figure 5.26. Here, the angular error of $0.03°$ is almost perfect. A small angular error also results in row-wise matches, i. e., plants of the same observed crop row are only associated with plants in the corresponding mapped crop row. This can also be seen in Situation 2. However, due to the translation invariance and only few observed gaps, the translational error is still high with 7.09 m and therefore no correct matches are found. Our results show, that – though designed for data association on crop fields – the Gaps data association is not well suited for our scenario, i. e., data association on ground vehicle images, where only a small part of the field is observed and descriptor invariance is not a strong requirement.

The GCBB approach shows similar results to the Gaps approach with almost no GT matches found. In contrast, the angular error of $9.43°$ and $5.84°$ as well as the translational errors with 7.18 m and 3.51 m are much lower. In comparison to our approach, the angular and translational errors are still high, though. Since GCBB does not specifically aim at rotational and translational invariance, it produces matches with smaller angular and translational displacement on average. Similar to the Gaps algorithm, due to the use of relative distances in the GCBB test, GCBB can find row-wise matches with low an-

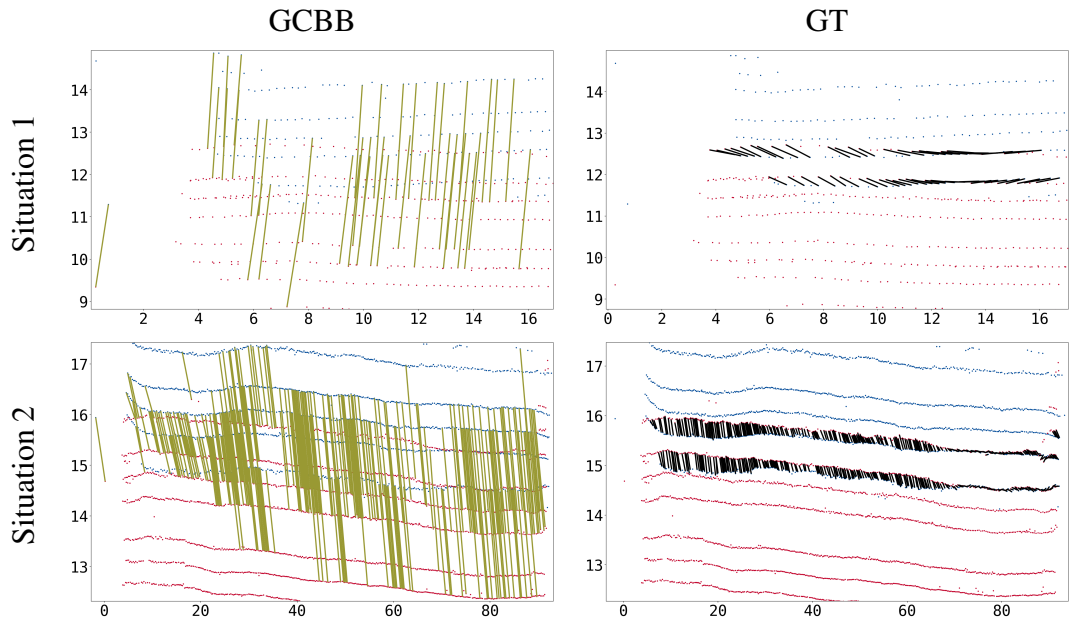

Figure 5.27: These images illustrate two example situations for the GCBB data association. Both situations visualize results taken from two data points of the Eichstetten data set. The mapped and observed features F_m and F_o are shown as red and blue dots respectively. Matches between features are highlighted using a solid line connecting the paired features. On the left, the matches M^{GCBB} found by the GCBB data association are shown in olive. For comparison, the GT matches M^{GT} for this data point are shown in black on the right.

gular errors of $2.35°$ and $0.66°$ respectively as shown in *Situation 1* and *Situation 2* in Figure 5.27. However, since the Target function of the GCBB algorithm is to maximize the number of matched observations, i. e., only Positive Information, and no Negative Information is considered, the GCBB data association tends to pull the observed features into the map, producing as much overlap and therefore as many matches as possible. Since this is usually not correct in our scenario as large parts of the observation are not yet mapped (No-Bias criterion), this results in high translational errors of $2.27\,\text{m}$ and $2.61\,\text{m}$ respectively and no GT matches for our two example situations.

The NN algorithm performs slightly better than the GCBB data association approach as it finds around 15% of the GT matches in 2 data points of the Eschikon data set and at least one GT match in 16 data points of the Eichstetten data set. It also shows the best angular and translational errors of all three non row-based data association approaches. This is surprising as it is the least sophisticated approach with no notion of geometric relations and instead assuming independence between features. This is caused by the fact, that for few data points in each data set the pose estimate before data association is already quite accurate. In these cases, the observations lie close to their correct correspondence in the map. Therefore, the NN strategy, which associates the features closest to each other, finds correct matches in these situations. An example with an accurate pose estimate is

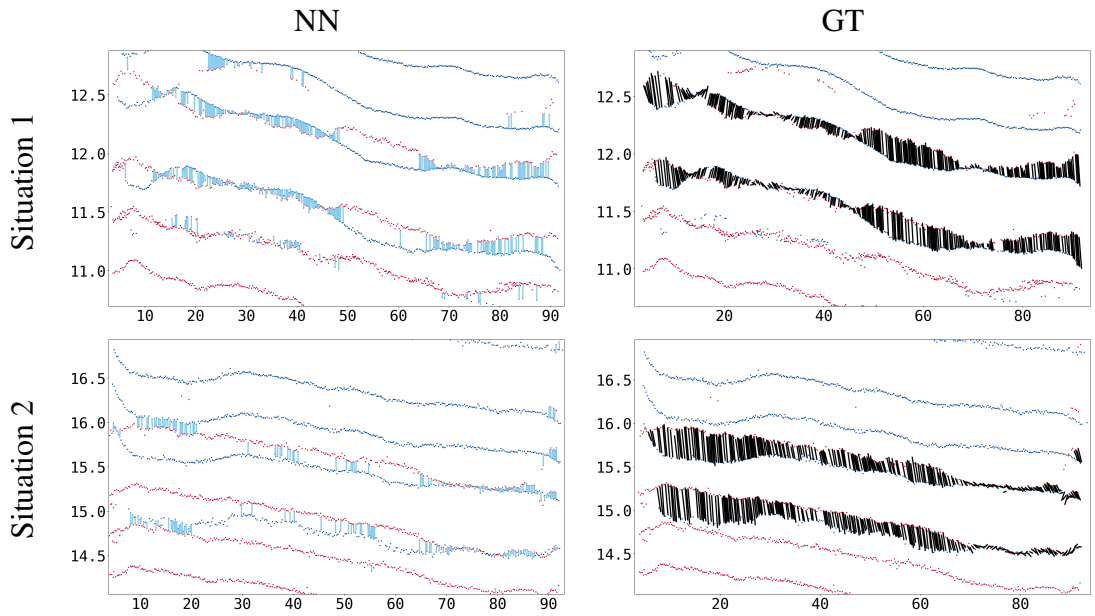

Figure 5.28: These images illustrate two example situations for the NN data association. Both situations visualize results taken from two data points of the Eichstetten data set. The mapped and observed features F_m and F_o are shown as red and blue dots respectively. Matches between features are highlighted using a solid line connecting the paired features. On the left, the matches M^{NN} found by the NN data association are shown in light blue. For comparison, the GT matches M^{GT} for this data point are shown in black on the right.

shown in *Situation 1* in Figure 5.28. Here, the NN algorithm finds 8 of 490 GT matches resulting in a low angular error of $0.02°$ and also comparably low translational error of $0.38\,\text{m}$. In most other situations, where the pose estimate is not sufficiently accurate, the NN algorithm cannot find correct matches. In contrast to the Gaps and GCBB approach that use geometric relations between features in their data association, the NN algorithm assumes independence between features. Therefore, unlike the other two non-row-based data association approaches, it usually cannot preserve the row structure of the field, i. e., , it cannot produce row-wise matches. Thus, observed features belonging to the same crop row are often associated with map features from different rows as shown in *Situation 2* in Figure 5.28. In this situation, although the angular error with $0.30°$ and the translational error of $0.75\,\text{m}$ are reasonably low, the NN data association does not find any correct matches.

In contrast to non row-based data association approaches, our approach shows much better results, with GT matches found 204 out of 272 times on the Eichstetten data set and 113 out of 164 times on the Eschikon data set. The average amount of GT matches found in these situations is also much higher with around $60\,\%$ for both data sets. This confirms that leveraging row structure improves data association results in this kind of setting. Considering the row structure is especially beneficial to the angular accuracy, where our approach has an average error below $1°$ on both data sets. Combined with a comparatively

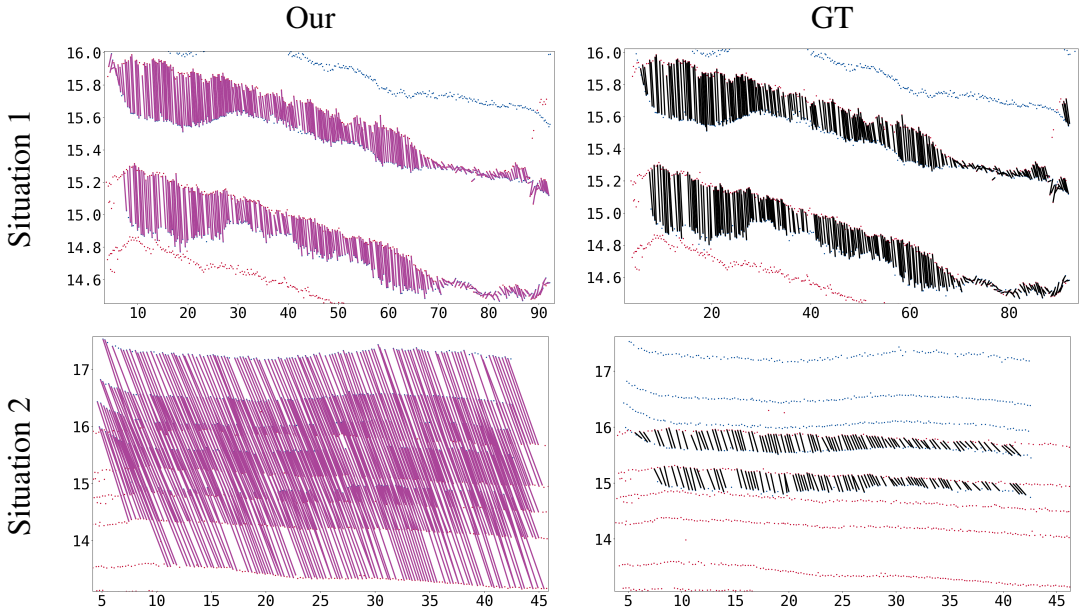

Figure 5.29: These images illustrate two example situations for our data association. Both situations visualize results taken from two data points of the Eichstetten data set. The mapped and observed features F_m and F_o are shown as red and blue dots respectively. Matches between features are highlighted using a solid line connecting the paired features. On the left, the matches M^{our} found by our data association are shown in purple. For comparison, the GT matches M^{GT} for this data point are shown in black on the right.

low translational error of $0.70\,\text{m}$, this often yields a successful data association as shown in *Situation 1* in Figure 5.29. Here, our approach finds a large amount of 379 out of 446 GT matches resulting in an angular error smaller than $0.01°$ and a translational error of $0.01\,\text{m}$. At first glance, an average translational error of $0.70\,\text{m}$ seems too high for precise navigation on a crop field. However, this high average translational error stems from few situations where the data association misaligns the crop rows and therefore causes a high lateral error as shown in *Situation 2* in Figure 5.29. Here, our approach fails to associate the correct rows and shifts the observed rows too far into the map. Since our approach – by design – preserves the crop row structure, e. g., plants of the same row are matched into the same row, the angular error is still low with $0.01°$ in this situation. However, associating the wrong crop rows causes a large lateral and longitudinal misalignment, resulting in a large translational error of $3.74\,\text{m}$ and thus no GT matches are found.

The results of this evaluation show that leveraging the row structure of crop fields is crucial for a successful data association. More generally applicable data associations that do not use the row structure of crop fields struggle to find correct data association matches. In contrast, our row-based data association can leverage the row structure of the crop field to find correct matches. Therefore, we will focus on our row-based data association approach in the following experiments. In the next section, we present a more detailed evaluation of the translational error of our approach to confirm its suitability for

precise navigation in agriculture.

5.5.3 Suitability for Precise Navigation on Crop Fields

The evaluation of the previous experiment confirmed that considering row structure is crucial to finding a good data association on individual plant positions. Therefore, our focus now lies on our row-based data association approach and showing its suitability for precise navigation on agricultural fields. To this end, we evaluate the robustness and translational accuracy of our approach in more detail as this is crucial for autonomous navigation on crop fields. We also investigate and compare the performance of our approach when equipped with the different Score Functions presented in Section 5.3.3.

Precise navigation on crop fields requires robust results and high angular and lateral accuracy to navigate the narrow space between crop rows without driving over the crops. For autonomous turning at the headlands a lower longitudinal accuracy suffices. The evaluation of the first experiment shows that the average angular accuracy of our approach with less than 1° is well suited for navigation on an agricultural field. However, evaluating the average translational error as in the previous experiment does not take into account the different requirements on lateral and longitudinal accuracy when traversing the rows of a crop field. In this experiment, we therefore investigate the translational accuracy in more detail by evaluating the lateral and longitudinal accuracy separately. Using the fact that our row-based data association already returns a result that is split into a lateral and longitudinal component, i. e., the shift s^*, we perform our evaluation on the shift returned by the row-based data association.

To evaluate the performance of our approach with respect to the GT data association based on shifts, we compute the best possible shift according to the GT data association matches M^{GT}, called the *ground truth shift* (GT shift) s^{GT}. Recalling that each shift $s \in S$ is a parametrization of a set of matches $M(s)$ (see Eq. (5.1) to Eq. (5.4) in Section 5.3.2), we define the GT shift s^{GT} as the shift $s \in S$ that represents a set of matches $M(s)$, such that $M(s)$ contains as many GT matches $m \in M^{\mathrm{GT}}$ as possible. In other words, to determine the shift $s^{\mathrm{GT}} \in S$, so that the corresponding set of matches $M(s)$ maximizes the number of correctly associated observations $c(M(s))$ as defined in Eq. (5.10):

$$\begin{aligned} s^{\mathrm{GT}} &:= \operatorname*{argmax}_{s \in S} \left| M(s) \cap M^{\mathrm{GT}} \right| \\ &= \operatorname*{argmax}_{s \in S} c(M(s)) \end{aligned}$$

Since our data association approach returns the shift s^* with the highest score for a given Score Function, we can compute the GT shift s^{GT} by equipping our row-based data association approach with a *GT Score Function* $\mathrm{sf}^{\mathrm{GT}}$, that returns the number of correctly associated observations c for any shift $s \in S$:

$$\begin{aligned} \mathrm{sf}^{\mathrm{GT}} \colon S &\to \mathbb{N} \\ s &\mapsto c(M(s)) \end{aligned}$$

We call the row-based data association equipped with the GT Score Function *GT data association algorithm*, or short *GT algorithm* to distinguish it from our data association approach equipped with the Score Functions presented in Section 5.3.3.

For each data point in our data sets, we split the translational error into a lateral component, called *row-shift* error Δr, and a longitudinal component, called *long-shift* error Δl, where the row- and long-shift error are the component-wise, absolute difference between the shift $s^* = (r^*, l^*)$ of our data association approach and the GT shift $s^{\mathrm{GT}} = \left(r^{\mathrm{GT}}, l^{\mathrm{GT}}\right)$ of the GT algorithm:

$$\begin{aligned} \Delta r\left(s^*, s^{\mathrm{GT}}\right) &:= \left|r^* - r^{\mathrm{GT}}\right| \in \mathbb{N} \\ \Delta l\left(s^*, s^{\mathrm{GT}}\right) &:= \left|l^* - l^{\mathrm{GT}}\right| \in \mathbb{R}_{\geq 0} \end{aligned}$$

Since the long-shift quantifies how far the observed features are shifted along the crop rows in metrical units, the long-shift error Δl is the longitudinal component of the translational error. The row-shift error Δr is a natural number, where 0 means, that the correct row-shift was found and a number larger than 0 implies that the data association is off by that many rows. Any row-shift error other than 0 infers a high lateral error of the magnitude of at least one spacing between crop rows. Since a high lateral accuracy is required, any row-shift other than 0 is not sufficient. Therefore, we do not evaluate the lateral error metrically, but instead use it to define a data association as *successful* if the row-shift error $\Delta r\left(s^*, s^{\mathrm{GT}}\right)$ is equal to 0 and thus the correct rows are associated.

We evaluate our data association approach on the four different Score Functions introduced in Section 5.3.3: *Product*, *Difference*, *IoU per Row* and *IoU*. To mitigate the bias towards matching as many observed crop rows as possible of the first three Score Functions, we introduced a regularization technique with a trade-off parameter $\lambda \in [0, 1]$. Higher values of this parameter λ correspond to a stronger regularization of the natural behavior of the Score Function and a value of $\lambda = 0$ disables the regularization. We therefore use a parameter of $\lambda = 0$ to disable the regularization technique and evaluate the natural behavior of the first three Score Functions. Additionally, we also evaluate a regularized version of the first three Score Functions. To this end, we tested the performance of different parameter values for each of the three Score Functions on a small test set of data points. Based on these tests we empirically determined the following regularization parameter values λ for the first three Score Functions given in round brackets: *IoU per Row* (0.4), *Difference* (0.3) and *Product* (0.75).

In Table 5.4 we present the robustness of our data association approach for the investigated Score Functions. We quantify the robustness of our data association approach as the number of successful data associations over all data points in each data set. For the Eichstetten data set, we get a success rate of more than $90\,\%$ for all Score Functions. This is due to the uneven spacing between the crop rows on this field as it generates many inconsistent row associations. We detect these inconsistent row associations using the data association presented in the previous chapter as explained in Section 5.3.2. Inconsistent row associations are labeled as invalid and therefore not considered in our row-based

%(#)	Eichstetten (# 272)	Eschikon (# 164)
IoU	90.81 (247)	70.73 (116)
IoU per Row	91.91 (250)	9.76 (16)
IoU per Row (0.4)	91.54 (249)	40.85 (67)
Difference	92.28 (251)	15.85 (26)
Difference (0.3)	91.91 (250)	51.83 (85)
Product	91.91 (250)	9.76 (16)
Product (0.75)	91.54 (249)	44.51 (73)

Table 5.4: This table shows the robustness for the different Score Functions and both data sets. We quantify the robustness by the number of times (#) the row-shift error is 0 and therefore the data association successful. For better comparison, we show the success rate as the percentage of successful data associations over the whole data set.

data association approach. Thus, only a small number of consistent row-shifts remains and therefore the chance of finding the correct row-shift is quite high on this data set. In the more challenging Eschikon data set, the Score Functions have different success rates. Here, the crop row spacing is more uniform and thus more row-shifts are consistent and therefore considered for data association. The IoU Score Function clearly outperforms all other Score Functions with a 70 % success rate compared to the second best Score Function Difference (0.3), which performs considerably worse with a success rate of only 52 %. Another clear result is the positive influence of the regularization technique for all three Score Functions, since the success rate is more than quadrupled on the Eschikon data set for each of them compared to the respective vanilla Score Function without regularization.

In Figure 5.30 we visualize the distribution of the longitudinal error as box plots. We show the overall longitudinal error over all data points on the top row. On the bottom row, we evaluate the longitudinal error only the data points, where the corresponding data association was successful, i. e., where the correct row-shift was found. This properly determines the longitudinal accuracy of our approach, since we only consider those values, where the rows are associated correctly and therefore the correct longitudinal shift can be found. The results of the overall longitudinal error (top row) confirm the findings from our robustness evaluation: The IoU performs best on both data sets with a median of 0.07 m on Eichstetten and 0.09 m on Eschikon. Furthermore, the upper quartiles of 0.69 m on Eichstetten and 0.77 m on Eschikon infer that 75 % of all data points have an error below 1.0 m respectively. This is comparable to the overall longitudinal error of 1.1 m we achieved in the previous chapter using the end-of-field detection. The important difference is, that here we achieve this overall longitudinal error over the whole field and not only towards the end of a crop row, where we approach and detect the end of the field. Again in line with the results of the robustness evaluation, the regularization technique improves the longitudinal error on the Eschikon data set for all three Score Functions but is not required on the Eichstetten data set due to its non-uniform row spacing. For the longitudinal accuracy over the data points with successful row association (bottom row),

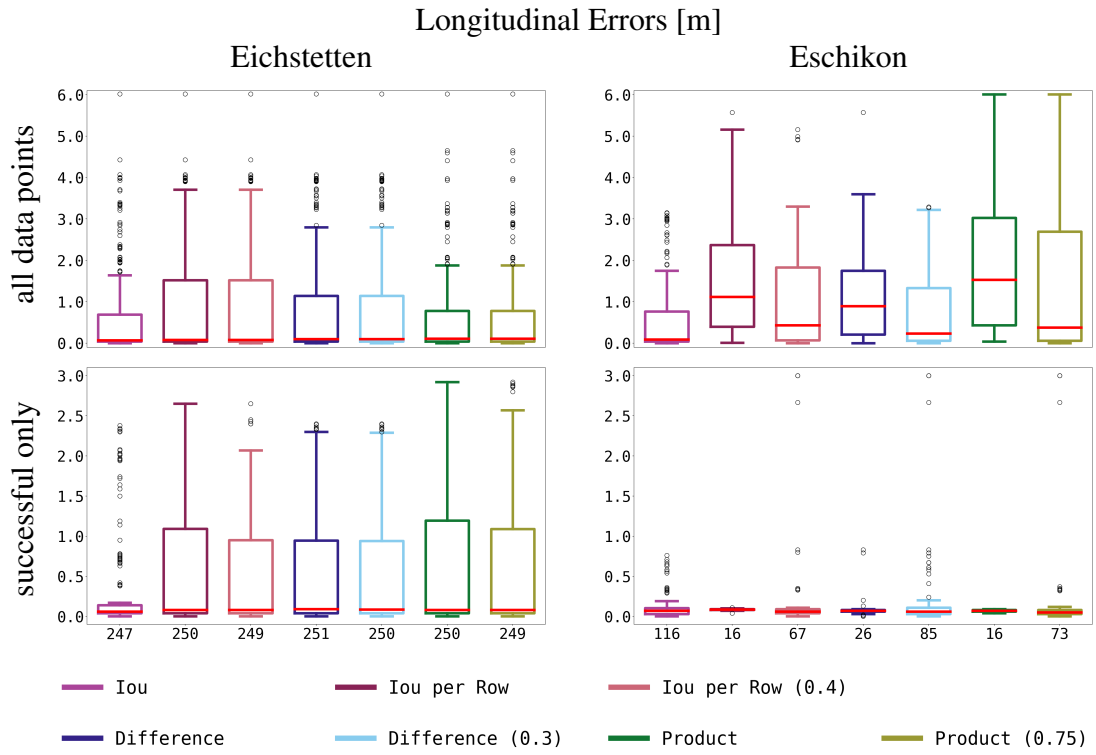

Figure 5.30: This figure shows the distribution of long-shift errors for the Eichstetten (left) and Eschikon (right) data set for all Score Functions in box plots. In the top row the longitudinal errors over all data points are shown. In the bottom row only the longitudinal errors at data points, where the data association is successful, i. e., where the row-shift error is equal to 0, are considered. The box plots are in the order of the legend from left to right.

we get an exceptionally high median accuracy of $0.09\,\mathrm{m}$ or higher throughout. This is an order of magnitude higher than the requirements for autonomous turning at the end of a field and therefore shows that our row-based data association approach is indeed well suited for precise navigation on agricultural fields. Furthermore, this high longitudinal accuracy enables the creation of maps featuring individual crop positions from ground vehicle data as we will show in the last experiment of our evaluation. At first glance, the IoU does not perform much better on Eschikon than the other three Score Functions, since the median error values of all Score Functions are all similarly small and additionally the vanilla variants have smaller boxes. However, the smaller boxes of the vanilla Score Functions originate from a much smaller number of considered values of 16 and 26 compared to 116 of the IoU Score Function, i. e., a much lower success rate compared to the IoU Score Function. The same holds for the median values of the Score Functions. Considering both, robustness and longitudinal accuracy jointly, the IoU Score Function clearly performs best, as it is more robust than the other Score Functions while maintaining a similar longitudinal accuracy.

The in-depth evaluation of the translational accuracy – split into lateral and longitudi-

nal accuracy – shows that our row-based data association approach is indeed well suited for precise navigation on crop fields. Our approach performs best when combined with the IoU Score Function, resulting in a robustness of $70\,\%$ up to $90\,\%$ and a longitudinal accuracy of below $0.1\,\mathrm{m}$. This longitudinal error is an order of magnitude smaller than the requirements for autonomous turning at the end of the field and enables longitudinal corrections not only when approaching the end of the field but also throughout the whole field. These results encourage further investigation into how our approach can be applied in localization and mapping techniques. Therefore, we discuss relevant properties as well as limitations of our approach in the following experiment of our evaluation, where we focus on how the performance of our approach depends on the amount and the quality of the provided input data.

5.5.4 Longitudinal Accuracy Along Crop Rows

The previous experiment confirms that the overall robustness and accuracy of our approach are well suited for localization and mapping applications based on individual plant positions. When implementing a localization or mapping algorithm, there is the critical question of when to perform data association. In our environment, this question translates to when within the crop field, i. e., how far along the crop row, should data association be performed. Depending on the application, different strategies are preferable: A localization algorithm needs to frequently correct the pose estimate. Therefore, the observed information has to be frequently associated with the information provided by the map. Here, being able to perform data association as early as possible, i. e., early within the crop row, is important. On the other hand, during a SLAM application, associating the observed data later might be beneficial as there is more time to collect information and therefore to improve the chance of correct data association. In this evaluation we investigate, whether our approach can be used for an early data association in a localization application as well as a later data association preferable in the SLAM use case. To this end we evaluate, how the results of our approach depend on where along the crop row the data association is performed.

Another important aspect for real-world application is the behavior of our data association depending on data of different input quality. As explained in Section 5.5.1 we therefore chose to evaluate on two real-world data sets posing different challenges to our data association approach: The Eichstetten data set has on the one hand non-uniform row spacing, which makes correct row association easier. On the other hand, it contains three different crop types, where the Chinese Cabbage is detected with lower accuracy, which makes finding the correct data association more challenging. The Eschikon data set is especially challenging as it features uniformly spaced crop rows as well as sugar beet plants that are detected with a comparatively higher false positive rate. Investigating the performance of our approach in detail in this evaluation also allows us to discuss how these challenges in the input data affect the performance of our approach.

We investigate the robustness and longitudinal accuracy of our approach depending on where along the crop row the data association is performed using the row-shift and long-shift errors from the previous evaluation. Here, we show the individual results for each

data point in both data sets for the Score Functions with the best performance, i. e., the IoU Score Function and the Difference (0.3) Score Function (see Figure 5.31). We visualize the row- and long-shift error in a table, where each row corresponds to one turn and the columns are the corresponding data points in chronological order. This means that cells on the left contain results for short local graphs, where the vehicle has traveled only a short distance and is still close to the beginning of the crop row. In turn, cells further to the right show results for longer local graphs as the vehicle moves through the field along the crop rows. As explained in Section 5.5.1, the distance traveled between one data point to the next within the same turn is approximately $2\,\mathrm{m}$. Therefore, this table describes in detail how the performance of our data association approach depends on the distance traveled along the crop rows.

To discuss interesting results at certain data points, we also show the distribution of Score Function values over the different row- and long-shifts considered in our data association approach at these data points. Additionally to the distribution of the IoU Score Function and the Difference (0.3) Score Function, we also present the distribution of the GT Score Function. This distribution is interesting, as it shows for which shifts GT matches M^{GT} can be found. Ideally, the GT Score Function should show a clear global maximum at the GT shift s^{GT} as for example in Figure 5.32. However, in some cases, the GT Score Function distribution also shows a more ambiguous distribution, with multiple local maxima along the same row shift as for example in Figure 5.34 in the middle and bottom image. Errors and inaccuracies in the detection algorithm and the NN data association during local graph construction cause some plant positions to be shifted along the crop rows. Therefore, GT matches M^{GT} can be found for different long-shift values along the same row-shift. This also means that the pattern of plant positions passed as input to the data association approach is less discriminative. Therefore, a less ideal GT Score Function distribution hints at more ambiguous and thus more challenging input data for our data association algorithm at the corresponding data point. For the IoU and Difference (0.3) Score Functions $\mathrm{sf}^{\mathrm{iou}}$ and $\mathrm{sf}^{\mathrm{diff}}$, we also highlight the shifts with values close to the maximum value at the best shift s^*. Given the Score Function distribution we highlight all values $v \in V_{0.95}$ that are larger than the $95\,\%$-quantile of the Score Function value range. These highlighted values $v \in V_{0.95}$ show alternative solutions for the data association as they also receive a high score. This gives an intuition about the uniqueness of the solution found by the corresponding Score Function. In other words, the more shifts receive a comparatively high score, the more ambiguous is the result of the data association.

The results for the Eichstetten data set on the left of Figure 5.31 show that performing data association after approximately half of the crop row was traversed results in a successful and accurate data association. This confirms the general intuition that integrating more data and therefore traversing a longer distance along the crop rows improves the success rate and accuracy of our data association approach. We also observe that we get good results already at the beginning of turn 1, 5 and 6. This directly correlates with the type of crop observed at the beginning of these turns versus the other turns. Recalling Figure 5.24, turn 1 starts with Kohlrabi and turn 5 and 6 start with Sweetheart Cabbage, where the detection algorithm has high accuracy and a low false positive rate. The other turns start with Chinese Cabbage, where the detections are less accurate. This directly

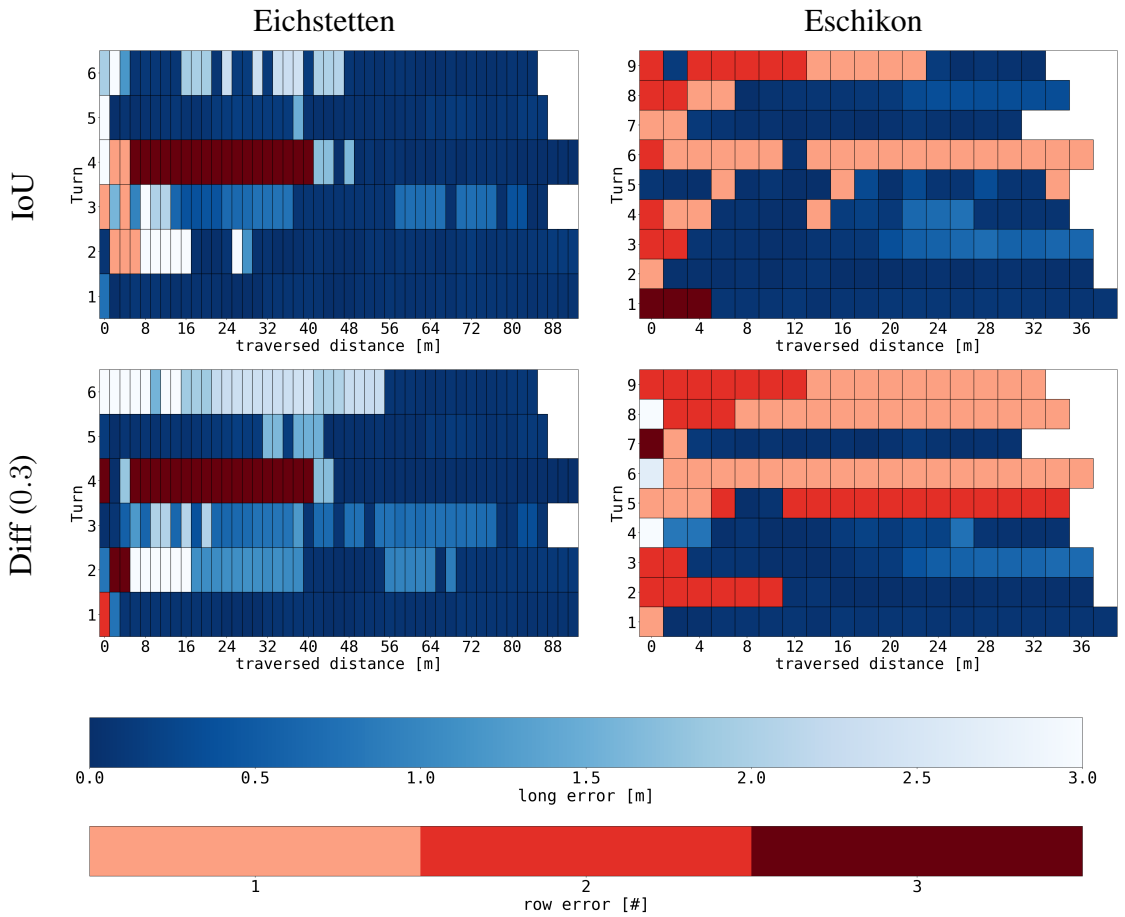

Figure 5.31: This figure shows the row- and long-shift errors for each turn on the Eichstetten and Eschikon data set for the IoU and Difference (0.3) Score Functions. The row-shift error is shown in shades of red. For cells with a row-shift error of 0, we show the long-shift error in shades of blue instead.

correlates the performance of our approach to the performance of the detection algorithm. We therefore conclude that our approach is able to find the correct data association early in the crop row, under the condition that the quality of the input data is high enough as is the case for turns 1, 5 and 6. We also conclude that our approach can handle input data of lower quality, if enough information is provided, i. e., a longer distance along the crop rows is traversed.

This is especially evident in turn 4, where initially both Score Functions find a data association that is 3 rows off from the correct row association until around 40 m into the crop rows. However, both Score Functions can recover and continuously find the correct data association after about half of the crop row length was traversed. The distribution of the Score Functions in Figure 5.32 show, how this is possible: The first observation is, that the distribution of the GT Score Function is ideal, meaning that the local graph does not have any grave errors. Still, both Score Functions cannot find the correct shift after 40 m into the crop rows. The Difference Score Function gives a very low overall score to

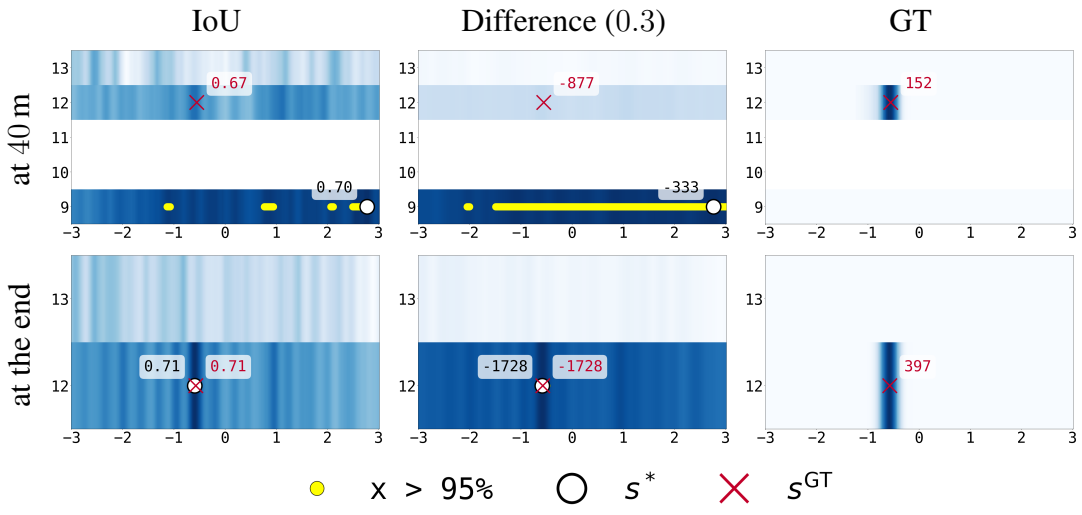

Figure 5.32: This figure shows the distribution of the IoU, Difference (0.3) and GT Score Functions at **turn 4** of the **Eichstetten** data set. Each row visualizes the Score Function distribution for a different location along the crop rows. We visualize the score distribution as image, where each pixel corresponds to the score $\mathrm{sf}\,(s)$ of a shift $s = (r, l) \in S^{\mathrm{valid}}$. The row-shift r is plotted along the vertical axis and the long-shift l along the horizontal axis. Following a blue color scale, pixels in darker blues indicate higher scores while pixels in lighter colors indicate lower scores. We annotate the GT shift s^{GT} with a red cross and the best shift s^* of the corresponding Score Function with a white circle. The corresponding scores $\mathrm{sf}\left(s^{\mathrm{GT}}\right)$ and $\mathrm{sf}\,(s^*)$ are shown in red and black respectively. Scores $v \in V_{0.95}$ that are close to the best score $\mathrm{sf}\,(s^*)$ are highlighted using yellow dots.

the correct row-shift of 12. On the other hand, it also gives very uniformly distributed, high scores to many shifts with row-shift 9 as indicated by the yellow markers along this row-shift. This score distribution leads to the conclusion that a discriminative match between the pattern of plant positions in the map and in the observed features has not been found and therefore many shifts yield a similarly good score. In contrast, although also not finding the correct shift, the IoU Score Function yields more discriminative results. The correct row-shift, although not yet highlighted, is already visible as a local maximum. There are also more distinguishable local maxima at row-shift 9 compared to the Difference (0.3) Score Function. This again confirms the good discriminative ability of the IoU Score Function and its suitedness for this kind of scenario. At the end of turn 4, both distributions have converged towards the correct shift. The ambiguity present at $40\,\mathrm{m}$ has been resolved and the distribution shows only one global maximum.

Against the general intuition that more information yields better data association results, turn 3 is an example of the Eichstetten data set, where the longitudinal accuracy of our approach decreases as the vehicle approaches the end of the row. The distribution of the GT Score Function shows, that this is not due to large errors in the local graph (see Figure 5.33). From the distributions of the IoU and Difference (0.3) we also observe, that they both find the correct row-shift with high certainty, as all high values are located

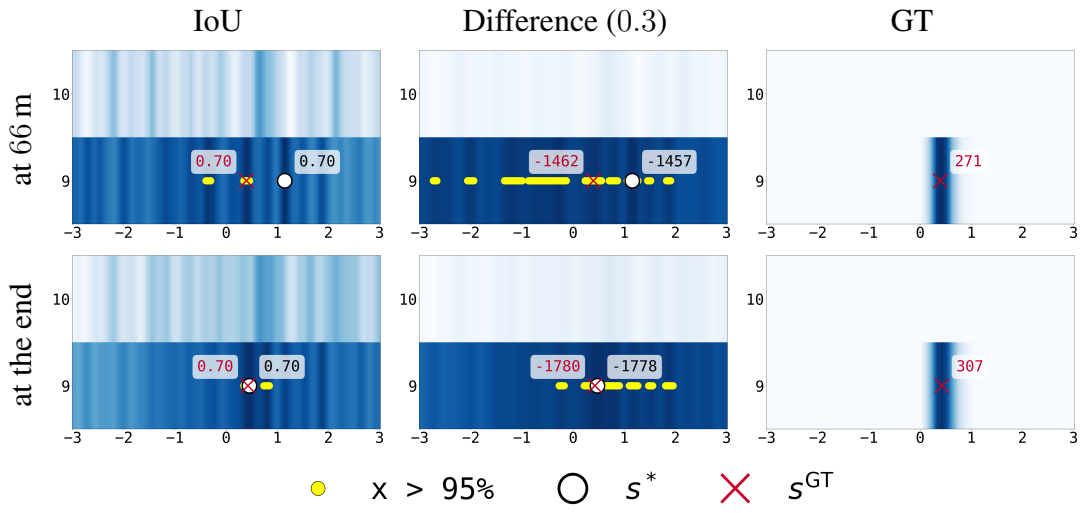

Figure 5.33: This figure shows the distribution of the IoU, Difference (0.3) and GT Score Functions at **turn 3** of the **Eichstetten** data set. Each row visualizes the Score Function distribution for a different location along the crop rows. We visualize the score distribution as image, where each pixel corresponds to the score $\mathrm{sf}\,(s)$ of a shift $s = (r, l) \in S^{\text{valid}}$. The row-shift r is plotted along the vertical axis and the long-shift l along the horizontal axis. Following a blue color scale, pixels in darker blues indicate higher scores while pixels in lighter colors indicate lower scores. We annotate the GT shift s^{GT} with a red cross and the best shift s^* of the corresponding Score Function with a white circle. The corresponding scores $\mathrm{sf}\left(s^{\text{GT}}\right)$ and $\mathrm{sf}\,(s^*)$ are shown in red and black respectively. Scores $v \in V_{0.95}$ that are close to the best score $\mathrm{sf}\,(s^*)$ are highlighted using yellow dots.

along row-shift 9. However, the pattern of plant positions is not discriminative enough to find the correct long-shift with high certainty. We can see multiple local maxima in both distributions at both data points. Again, we can observe, that the IoU Score Function shows more discriminative results with fewer local maxima. Although the correct shift is found at the end of the turn, the ambiguity remains for both Score Functions.

The results on the Eichstetten data set show that our data association approach can find the correct data association, even on less accurate input data, after traversing about half of the crop row length. Our approach can also find good data association solutions early along the crop row, if the quality of the input data is high enough. We also see, how the distribution of the Score Functions allows reasoning about the ambiguity of the given data association problem.

Confirming the results from the previous experiment, our approach shows lower performance on the Eschikon data set (see Figure 5.31). This is not surprising, as we already discussed, that the Eschikon data set is more challenging than the Eichstetten data set. Recalling Section 5.5.1, the crop rows in the Eschikon data set are more uniformly distributed than in the Eichstetten data set. Also, the detection algorithm has a higher false positive rate on the sugar beets planted throughout the Eschikon crop field. Therefore, the Eschikon data set is well suited to showcase some limitations, but also the potential

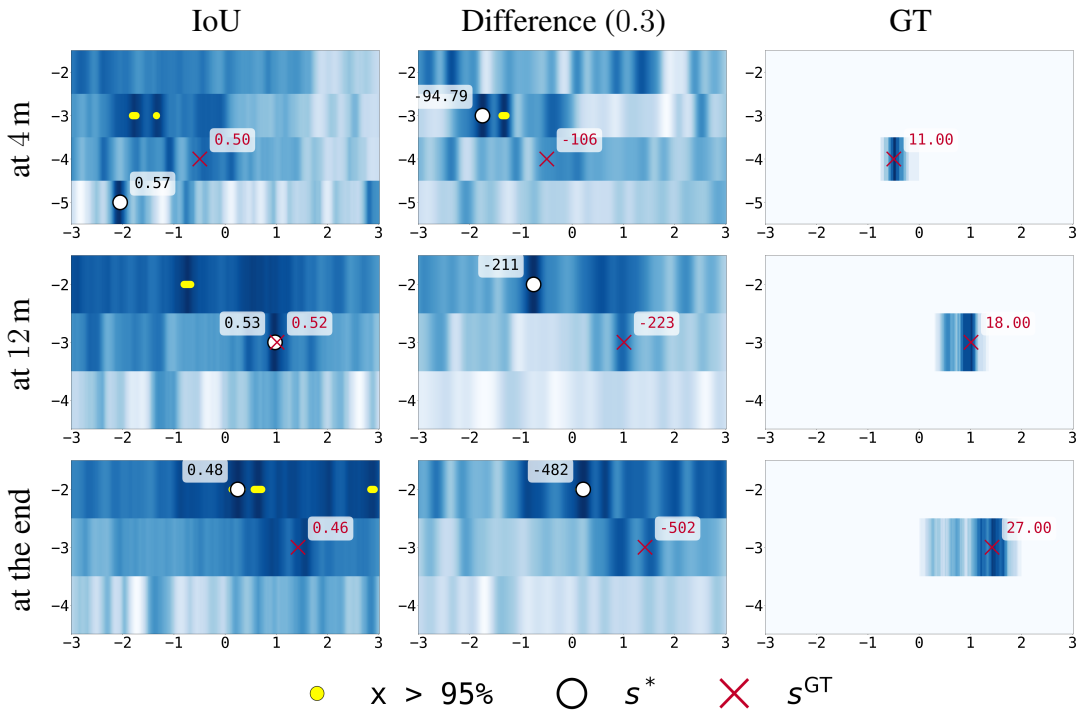

Figure 5.34: This figure shows the distribution of the IoU, Difference (0.3) and GT Score Functions at **turn 6** of the **Eschikon** data set. Each row visualizes the Score Function distribution for a different location along the crop rows. We visualize the score distribution as image, where each pixel corresponds to the score $\mathrm{sf}(s)$ of a shift $s = (r, l) \in S^{\text{valid}}$. The row-shift r is plotted along the vertical axis and the long-shift l along the horizontal axis. Following a blue color scale, pixels in darker blues indicate higher scores while pixels in lighter colors indicate lower scores. We annotate the GT shift s^{GT} with a red cross and the best shift s^* of the corresponding Score Function with a white circle. The corresponding scores $\mathrm{sf}\left(s^{\text{GT}}\right)$ and $\mathrm{sf}(s^*)$ are shown in red and black respectively. Scores $v \in V_{0.95}$ that are close to the best score $\mathrm{sf}(s^*)$ are highlighted using yellow dots.

of our approach. The most prominent observation is that the Difference (0.3) Score Function shows overall less robust results compared to the IoU Score Function, especially in turns 5 and 8 but also in turns 2 and 9. In contrast to the Eichstetten data set, where both performed quite similarly, the Eschikon data set with lower quality input data pushes towards the limits of our approach but also showcases the high discriminative ability of the IoU Score Function. Inline with the findings on the Eichstetten data set, waiting until at least half of the field was traversed before performing data association is a good strategy. However, earlier data association can also yield good results especially during the first half of the field for turns 1 to 4.

Interesting exceptions to these findings are turn 6 and 9, where the correct shift cannot be found for the most part of the crop row length. In Figure 5.34 we show the distribution of the Score Functions for turn 6, where the correct shift is only found once by the IoU Score Function throughout the whole turn. The distribution of the GT Score Function

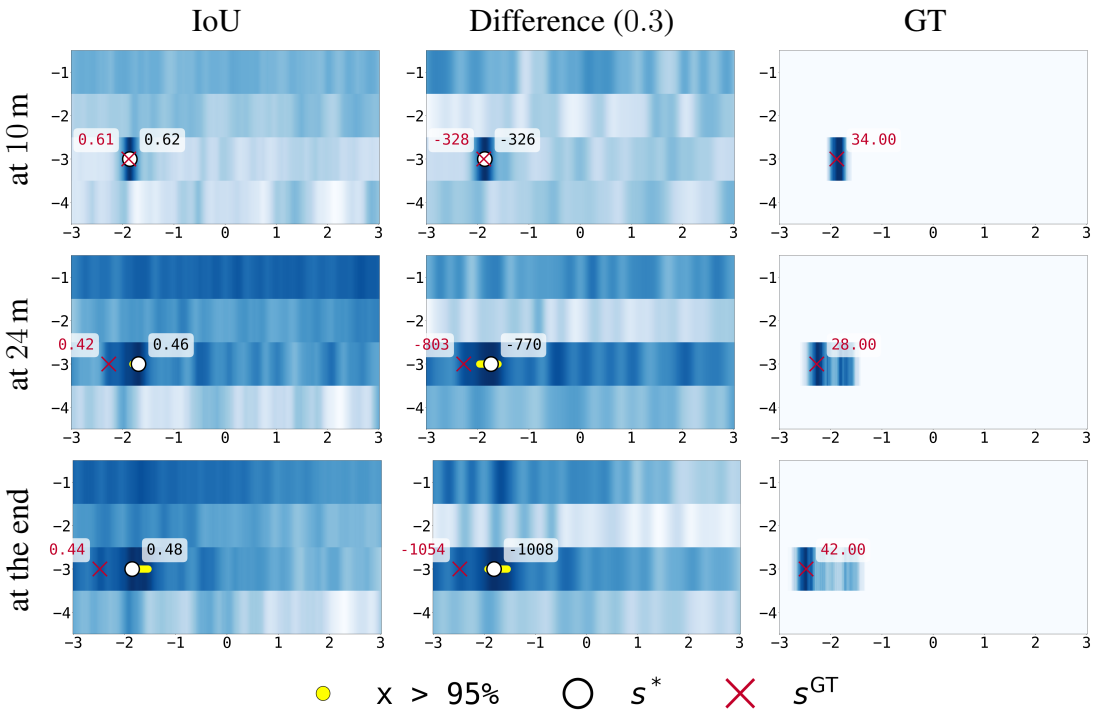

Figure 5.35: This figure shows the distribution of the IoU, Difference (0.3) and GT Score Functions at **turn 3** of the **Eschikon** data set. Each row visualizes the Score Function distribution for a different location along the crop rows. We visualize the score distribution as image, where each pixel corresponds to the score $\mathrm{sf}(s)$ of a shift $s = (r, l) \in S^{\text{valid}}$. The row-shift r is plotted along the vertical axis and the long-shift l along the horizontal axis. Following a blue color scale, pixels in darker blues indicate higher scores while pixels in lighter colors indicate lower scores. We annotate the GT shift s^{GT} with a red cross and the best shift s^* of the corresponding Score Function with a white circle. The corresponding scores $\mathrm{sf}\left(s^{\text{GT}}\right)$ and $\mathrm{sf}(s^*)$ are shown in red and black respectively. Scores $v \in V_{0.95}$ that are close to the best score $\mathrm{sf}(s^*)$ are highlighted using yellow dots.

shows multiple local maxima at $12\,\mathrm{m}$ and at the end of the turn instead of one the local maximum at the beginning (at $4\,\mathrm{m}$). Multiple maxima indicate that there are multiple good solutions to the data association problem according to the Score Function. This showcases that with more detections accumulated along the crop rows, errors also accumulate and therefore the pattern of the observed plant positions can become less accurate and less distinguishable. This in turn makes it more challenging for our data association algorithm to find the correct shift, since the distribution of the observed plant positions does not match the distribution of the mapped plant positions. In this example more information does not improve the results of our approach, so that our data association cannot find the correct shift almost throughout the whole turn. Considering the highlighted shifts, we see that there are multiple local maxima. This again implies, that our approach did not find a unique solution and that therefore the data association results are ambiguous.

Another interesting set of data points is found in turn 3. Here, we observe results

similar to turn 3 of Eschikon, where the longitudinal accuracy decreases when traversing further along the crop rows. The distribution of the Score Functions for this turn is shown in Figure 5.35. We see that the GT Score Function distribution is similar to the distribution in turn 6, except for the distribution at the beginning of turn 3 at 10 m. Here, the distribution is close to ideal with only one global maximum. However, with more information accumulated, as the vehicle traverses along the crop rows, the quality of the local graph decreases and the GT Score Function distribution fans out along the correct row-shift producing several local maxima. In this turn, we see a good example of how the initial local graph has already enough information for our approach to find the correct data association: Both Score Functions find the correct shift as a unique solution, since no other shifts are highlighted. As more data is accumulated, the plant pattern becomes less clear. Similar to the Eichstetten example the correct row-shift is still found. However, both Score Functions find the correct shift further to the right and are quite certain about it, since no other shifts are highlighted. This causes the long-shift error to increase although more information is available as the vehicle progresses along the crop rows.

The results of the more challenging Eschikon data set showcase the potential and limitations of our approach well. The results on Eschikon confirm that our approach can still find the correct data association, even with lower quality input data throughout the entire field and almost uniformly spaced crop rows. It confirms the findings on the Eichstetten data set, that performing data association after about half of the crop rows have been observed usually yields good data association results. However, if the quality of the input data is low and too much error is accumulated, performing data association earlier might be beneficial.

This evaluation shows that the performance of our approach not only depends on the amount of information gathered along the crop rows. It also depends on the quality of the input data and how much error is accumulated, while traversing the crop rows. Therefore, we conclude that with good quality input data, it is preferable to delay data association as much as possible to reduce ambiguity and ensure a successful data association. On the other hand, with lower quality input data, where errors accumulate faster, it might be beneficial to perform data association earlier to prevent introducing ambiguity caused by the accumulated error. Our results show that our data association approach can be applied towards the end of the crop rows as might be preferable in a SLAM use case. Especially, when the input data is accurate enough, an earlier and more frequent data association is also possible as required for localization applications. The results also show that it might be helpful to consider the whole distribution of the Score Function during data association to obtain a measure of the ambiguity of the input data and therefore about the quality of the computed data association. Such a measure is valuable in both application scenarios as it allows us to determine, whether the result of the data association can be trusted or whether data association should be delayed until ambiguities are resolved. In future work, further research towards obtaining such a measure of ambiguity using the Score Function distribution is quite interesting. Our extensive evaluation of the robustness as well as angular and translational accuracy confirms the suitedness of our approach for localization and mapping applications on agricultural fields. To go one step further, we provide a proof of concept of our approach by applying our row-based data association in

a SLAM application on real-world data in the last experiment of our evaluation.

5.5.5 Use Case SLAM - Qualitative Evaluation

In this evaluation we show qualitative results of our data association approach in a SLAM application. This experiment therefore provides a proof of concept for the applicability of our data association approach in a SLAM framework. To this end, we use a graph-based SLAM algorithm to build maps with individual plant positions from our Eichstetten and Eschikon data sets. Within this SLAM algorithm, we use our approach for data association in the most challenging situation, i. e., after the vehicle performed a turning maneuver at the end of the field. These situations are especially challenging, since the vehicle looses track of the crop rows during the turning maneuver and therefore needs to re-localize its pose relative to the crop field after turning. We show the maps created by our data association algorithm and qualitatively compare them to the maps created using the GT algorithm and the GT data association matches M^{GT}.

To create maps featuring individual plant positions of the whole crop field, we iteratively build a SLAM graph using the global and local graph as described in Section 5.5.1. To accumulate as much information as possible, we merge the local graph into the global graph whenever the vehicle reaches the end of the field. In contrast to the merging strategy described in Section 5.5.1, we use the set of data association matches $M\,(s^*)$ computed by our data association to merge the local graph into the global graph instead of ground truth matches. After merging, we optimize the global graph. Since we perform a merge whenever the vehicle reaches the end of the field, we track the number of merges performed by counting the number of turns the vehicle executed so far. Using the same procedure, we build a pose graph using the GT algorithm and its GT shifts s^{GT} inferred from the GT data association matches M^{GT}. Recall that the GT algorithm is our row-based data association equipped with the GT Score Function, which by design is as close as possible to the GT data association, since it counts the number of GT matches for each shift. In other words, the GT algorithm computes the best possible data association, when modeling the space of data associations as a set of shifts (Eq. (5.12)). Therefore, the results of the GT algorithm demonstrate how our choice of parameterizing the space of data association matches $\mathcal{M}$ using shifts $s \in S$ affects the quality of the maps. Additionally, we show the detections GT pose graph $\mathcal{G}^{\text{GT, det}}$ as defined in Section 5.5.1. Recall that this graph is built relying only on the GT data association matches M^{GT} to associate plant features detected by the SEP detection algorithm and does not use the crop row structure. For an easier comparison, we show only the plant features that are associated with a plant, i. e., that have an id not equal to 0. This GT graph $\mathcal{G}^{\text{GT, det}}$ serves as the ground truth comparison in our qualitative evaluation as it is built using a traditional SLAM technique with GT data association matches M^{GT}. The feature nodes $\hat{Z}$ in each of the (global) pose graphs then define the map of plant positions. For a better comparison, we also show the resulting trajectory of the vehicle by connecting consecutive pose nodes $\hat{\mathbf{x}}_{t-1}, \hat{\mathbf{x}}_t \in \hat{X}$ of each graph. For an easier comparison, using the fact that the pose nodes of all graphs are GPS-referenced, we also overlay the GT vehicle trajectory obtained from the GT graph $\mathcal{G}^{\text{GT, det}}$. We show the final maps of the IoU and Difference (0.3) Score Functions next to the maps of the

GT algorithm and the GT graph $\mathcal{G}^{\text{GT, det}}$ in Figure 5.36 for the Eichstetten data set and in Figure 5.37 for the Eschikon data set.

Looking at the map quality as well as the trajectory of the GT algorithm, we see that both are very similar to the results obtained using the GT matches for the GT graph $\mathcal{G}^{\text{GT, det}}$. This means that given a Score Function that models the GT data association well, i. e., the GT Score Function, our approach is able to closely match the results of the GT data association. This confirms, that the quality of the map is not diminished by representing sets of data association matches M as shifts s and therefore supports our idea to parameterize data association matches using shifts.

The results of the IoU and Difference (0.3) Score Functions on the Eichstetten data set show that both Score Functions are able to robustly and accurately associate the plant features after each turn. Both trajectories are close to the trajectory defined by the GT graph, the map features can be clearly distinguished and the row structure is visible. Regarding the longitudinal accuracy, we do not see errors such as the trajectory being too long or too short compared to the trajectory defined by the GT matches. We also do not see longitudinally misaligned crop rows, which we would expect to see if there were large errors in longitudinal alignment. This shows, that given enough information to distinguish adjacent crop rows, our data association approach can be used to create accurate maps of individual plants in crop fields. These results are also supported by our evaluation of the row- and long-shift errors in the previous evaluation (see Figure 5.31), where the correct data association is found for each turn by both Score Functions as the vehicle approaches the end of the crop rows (for traversed distance larger than $50\,\text{m}$).

On the Eschikon data set the longitudinal alignment is also good for both investigated Score Functions as we do not see any longitudinally misaligned crop rows or too long or too short trajectories. Still, the Eschikon data set is clearly more challenging, as the IoU and Difference (0.3) Score Functions do not always find the correct data association and crop rows are misaligned. This leads to shifted trajectories, that run in parallel to the GT trajectory, more noisy and duplicate plant features, as well as distorted and – in some parts of the map – unrecognizable crop row structure. In an extreme case of crop row distortion, the Difference (0.3) Score Function shows a crop row gap between turn 5 and turn 6 (at $y \approx 6\,\text{m}$). In contrast, the IoU Score Function does not show such extreme errors. Again, these results coincide with our evaluation of the row- and long-shift errors in the previous evaluation (see Figure 5.31), where the Difference (0.3) Score Function cannot find the correct data association for both turns 5 and 6. In contrast, the IoU Score Function only fails to associate the crop rows correctly in turn 6, which explains the more accurate mapping results of the IoU Score Function. Overall, these qualitative results further confirm the higher robustness of the IoU Score Function compared to the Difference (0.3) Score Function as evaluated in the previous experiments. In most parts of the map, however, both Score Functions show good mapping results, with individual plant features and clear crop row structure.

For a better understanding on how an incorrect row association affects the map quality, we show two intermediate steps during graph creation in Figure 5.38 for the IoU Score Function. Until turn 4 the results of the IoU Score Function are quite similar to the GT graph. After merging the local graph into the global graph at the end of turn 5, we can see

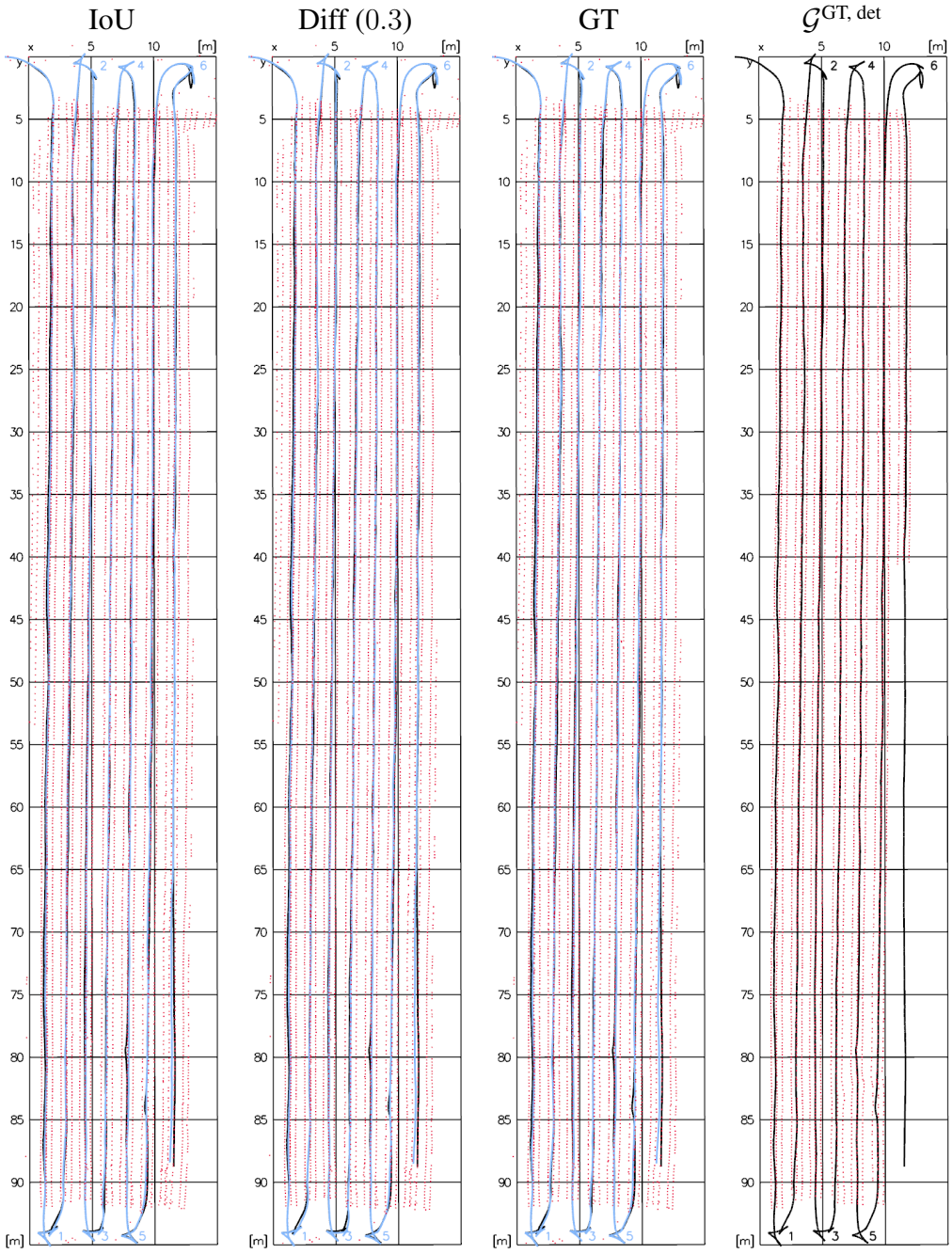

Figure 5.36: This figure shows the SLAM results on the **Eichstetten** data set. The map features, i. e., plant position nodes $\hat{Z}$ of the graph, are shown as red dots. The trajectory of the vehicle is shown as a blue line by connecting the pose nodes $\hat{X}$ of the graph. The number of each turn is shown next to the turn. The GT vehicle trajectory defined by the GT graph $\mathcal{G}^{\text{GT, det}}$ is shown as a black line.

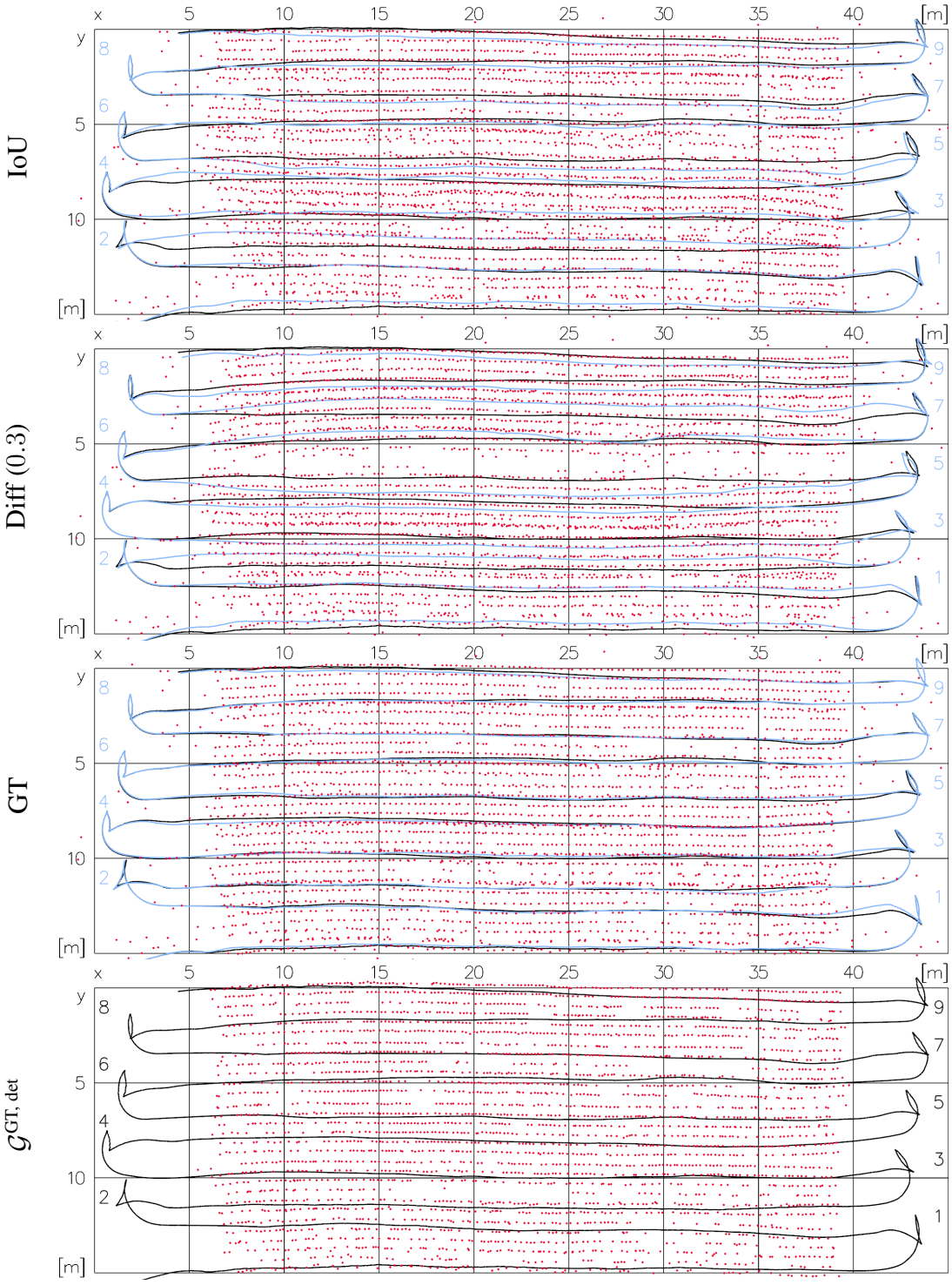

Figure 5.37: This figure shows the SLAM results on the **Eschikon** data set. The map features, i. e., plant position nodes $\hat{Z}$ of the graph, are shown as red dots. The trajectory of the vehicle is shown as a blue line by connecting the pose nodes $\hat{X}$ of the graph. The number of each turn is shown next to the turn. The GT vehicle trajectory defined by the GT graph $\mathcal{G}^{\text{GT, det}}$ is shown as a black line.

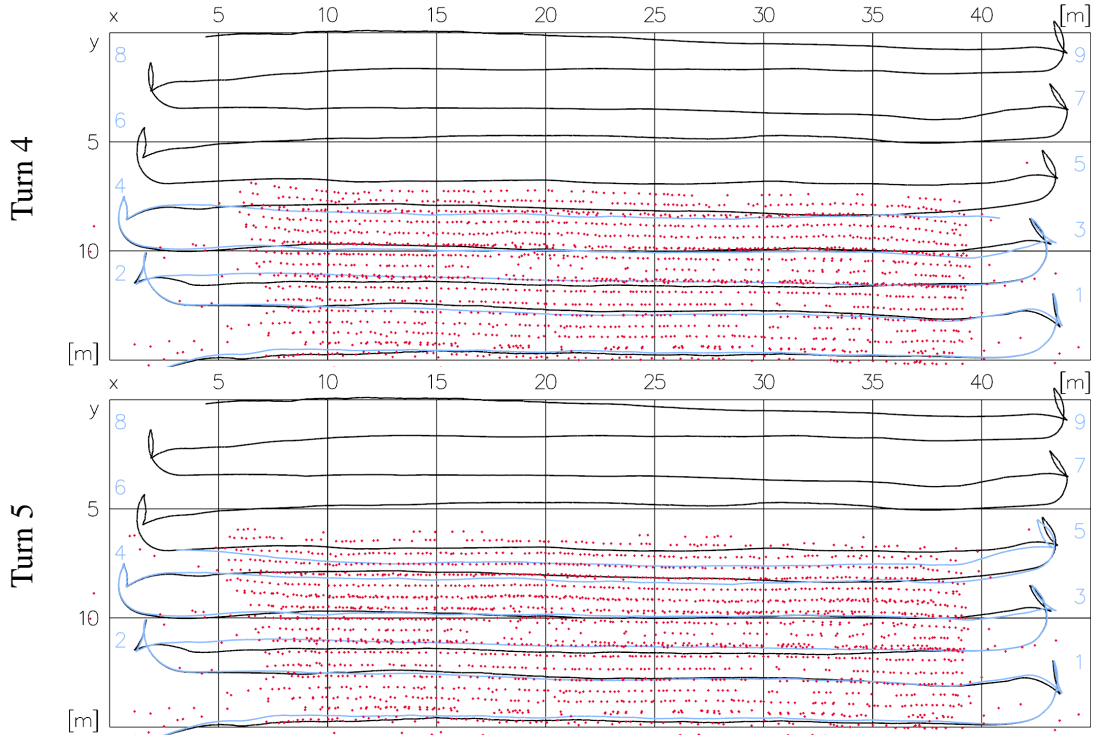

Figure 5.38: This figure shows intermediate SLAM results on the **Eschikon** data set for the **IoU** Score Function. The map features, i. e., plant position nodes $\hat{Z}$ of the graph, are shown as red dots. The trajectory of the vehicle is shown as a blue line by connecting the pose nodes $\hat{X}$ of the graph. The number of each turn is shown next to the turn. The GT vehicle trajectory defined by the GT graph $\mathcal{G}^{\text{GT, det}}$ is shown as a black line.

that the data association is one row-shift off, since the trajectory is pulled one row down and runs in parallel to the trajectory of the GT graph. Here, we also see that the crop rows below are pulled up and get distorted during the optimization. This makes them harder to recognize, especially in the area at the beginning of turn 3 at $x \geq 30\,\text{m}$ and $y \approx 11\,\text{m}$. As the graph construction and optimization continues, the crop rows in this area keep being pulled up, which explains the result that we see in Figure 5.37, where the crop rows in this area are not recognizable anymore. However, for the rest of the data set the IoU Score Function finds the correct data association. Therefore, the graph built using the IoU Score Function recovers from this wrong data association and converges towards the GT graph, with the trajectory before turn 5 slightly pulled up and after turn 5 slightly pulled down respectively. This detailed analysis shows that if crop row misalignments occur during the graph construction, they cause a shift of the trajectory parallel to the crop row direction and distortion of the crop row structure. Since these data association errors only affect local areas of the graph, during optimization the graph can partially correct these errors and still converge to a good solution, with most of the crop row structure clearly visible.

Overall, the evaluation shows that the quality of the maps mostly depends on the correct row alignment. Furthermore, the longitudinal accuracy is more than sufficient for

a mapping application. Using our row-based data association approach, we can create accurate maps with individual plant features.

Our extensive experimental evaluation confirms that row-based data association enables data association on individual plant positions. In-depth evaluation of the robustness as well as angular, lateral and longitudinal accuracy of our approach proves that it is well suited for localization and mapping techniques in real-world applications. Furthermore, we demonstrate that – depending on the quality of input data – our approach can be applied to successfully associate data already shortly after turning as well as further along the crop rows. Therefore, it is suited to correct the longitudinal pose estimate in a localization application throughout the entire field – not only at the end of the field as in the previous chapter. It is also suited for SLAM applications, where data association might be performed only once at the end of each traversed set of crop rows, as demonstrated in the last experiment.

5.6 Conclusion

When performing a turning maneuver at the headlands, errors in the pose estimate usually accumulate since the crop row structure of the field is not perceived in this situation. Relocalizing the vehicle after turning is therefore crucial for efficient autonomous navigation behavior, where the vehicle does not skip crop rows or traverses the same crop rows multiple times. This requires associating the observed crop rows with the mapped crop rows, which is hard due to the uniform distribution of the crop rows and the larger error in the pose estimate. In order to resolve these ambiguous situations, our key idea is to consider individual plant positions, i. e., point features, instead of the crop rows, i. e., line features. The slight irregularities of the distribution of plant features along the crop rows then enable us to distinguish different crop rows and find the correct data association. Data association on plant features poses its own challenges: First, plants of the same crop type look similar and therefore are hard to distinguish. Second, the plants are uniformly distributed along parallel and equidistant rows. This results in a dense and almost uniform feature distribution across the whole field, which is highly ambiguous.

In this chapter, we presented our data association algorithm that tackles these challenges by leveraging slight differences in the distribution of individual plants along the crop rows to find the correct data association after turning. This includes our parametrization of data association matches that constrains the search space of all possible data association matches to a feasible subset of data association matches that preserve the crop row structure. We also introduced a novel continuous match counting strategy based on the cross-correlation to obtain a highly discriminative data association method that can detect slight irregularities in the plant feature distribution to find the correct data association.

We performed an extensive experimental evaluation on real-world data captured on two different crop fields. In this evaluation, we first compare our approach to other data association approaches that are not designed for our specific scenario. The results of this evaluation confirm that considering the crop row structure of the field is required to find the correct data association on individual plant positions as features. In our second

experiment, we investigate the translational accuracy of our approach in more detail by evaluating the longitudinal and lateral component separately. Our IoU Score Function shows the best performance with $70\,\%$ and up to $90\,\%$ of correct crop row alignment and a longitudinal accuracy below $0.1\,\mathrm{m}$, which is well suited for precise autonomous navigation on agricultural fields. In our third experiment, we analyze the behavior of our data association approach depending on where along the crop rows the data association is executed. Our evaluation shows that an early execution while the vehicle is still at the beginning of the crop rows is possible if the quality, i. e., false detection rate and accuracy, of the input data is sufficient. In general, a later execution, when the vehicle has collected more data while traversing along the crop rows, usually results in more robust data association results. In our last experiment we showcase our data association approach in a SLAM application on real-world data.

Interesting further application scenarios of our data association algorithm in the field of autonomous navigation in agriculture include: deploying our data association in an online localization algorithm to autonomously navigate entire crop fields leveraging the highly accurate pose estimate including the position along crop rows while traversing the field; exploring different agricultural use cases of the maps containing individual GPS-referenced plant positions produced by our data association algorithm to, e. g., create detailed reports on the status of each individual plant of the field, which is especially interesting for high value crops or crop research such as phenotyping.

Overall, our experimental evaluation confirms that leveraging the row structure is crucial for successful data association on this kind of feature distribution. It thus confirms that our key ideas of defining *Plant-Rows* and our *Parametrization* based on Plant-Rows as well as *counting the measure of overlap continuously* enables *robust* computations on large amounts of almost uniformly distributed features while at the same time maintaining *high discriminative* ability since *all features are considered jointly*. The experiments also confirm that our regularization technique can mitigate the bias of the first three Score Function variants, the Product, Difference and IoU per Row Score Functions, which enables these variants to handle *unmapped features*. Additionally, our extensive evaluation highlights the performance of the IoU Score Function, which is an improved version of the other three Score Functions as it is by design an *unbiased variant for the continuous measure of overlap*. All our results confirm that this unbiased variant is best suited for robust and accurate data association in mapping applications. Our last experiment, which is a SLAM application on real-world data that utilizes our data association approach, extends our quantitative evaluation, since it also confirms qualitatively that our approach is well suited for localization and mapping applications in agricultural fields.

In this chapter, we presented a row-based data association technique that can handle indistinguishable, densely and almost uniformly distributed point features. We show that it enables estimating the full pose of the vehicle for re-localization after turning, which is important for precise, reliable and efficient autonomous navigation on agricultural fields.

Chapter 6

Discussion

In this thesis we presented novel techniques that enable estimating the full pose of an agricultural vehicle relative to the plants of a crop field. In order to obtain an accurate and robust estimate of the heading, the lateral offset as well as the longitudinal position of an agricultural vehicle relative to the crops we had to overcome the following challenges:

First, the crops need to be detected reliably across many different crop types at different growth stages. Therefore, a crop detection method should detect crops of a large variety of shapes and sizes, while at the same time being able to distinguish them from wild growing vegetation such as weeds. Second, the local detections need to be fused with information from other sensor modalities such as, for example, GPS information to obtain a consistent pose estimate in all three dimensions, and enable transition maneuvers at the headlands, where the crops are not visible in the local sensor data. This is especially challenging since the crops are hard to distinguish and therefore data association is not straightforward. Third, finding the correct data association after performing turning maneuvers at the headlands is even more challenging, since tracking of the crops in the field is lost during turning. While re-localizing the vehicle after turning is not required to prevent driving over crops, it is important for efficient traversal of crop fields. Fourth, the pose estimate needs to be highly accurate in the heading and lateral offset component to enable the autonomous navigation system to guide the agricultural vehicle along the crop rows without damaging crops.

Our first technique focuses on robust detection of crop rows on a large variety of crop types at different growth stages. We contribute to state-of-the-art techniques by leveraging the crop row structure of the field and extracting all crop rows jointly to achieve more robust crop row detections. Our experimental evaluation on real-world data of crop fields featuring many crop types at different growth stages confirms that our crop row detection approach can robustly detect the crop row structure of the field even in challenging situations during transition maneuvers, where the crop row structure is only partially visible. Overall, the heading and lateral accuracy of the detected crop rows is also well suited to guide a vehicle along the crop rows without damaging crops.

We also presented several techniques that enable integration of the local crop row detections with other sensor modalities such as GPS information to obtain a consistent pose estimate in all three components, the heading, the lateral and the longitudinal position of the vehicle. These include our novel *Crop Row data association* approach based on geometric consistency between the detected and the mapped crop rows, as well as our

End of the Field detection to obtain more accurate longitudinal position estimates as the vehicle approaches the headlands. Furthermore, we also proposed to split the sensor measurement into a heading, lateral and longitudinal component to enable fusing measurements from different sensor modalities with different accuracy into a consistent pose estimate. Our evaluation of two different localization methods equipped with our novel techniques on real-world data confirms the relevance of our techniques: Using local crop row detections and GPS information in separate localization algorithms clearly shows the advantages and disadvantages of each sensor modality. In contrast, employing our techniques to obtain one fused localization method based on both sensor modalities enables the localization to fully leverage the advantages of both sensor modalities without being affected by their respective disadvantages. In conclusion, our techniques for a fused localization method enable full pose estimation with sufficient accuracy to enable precise and reliable autonomous traversal of entire crop fields – beyond crop row following.

For more efficient autonomous traversal of a crop field, we also investigated re-localizing the autonomous vehicle after performing transition maneuvers at the headlands. To this end, we presented a novel data association that uses individual plant positions along each crop row to find the correct data association required for re-localization. Finding the correct data association using the distribution of individual plants along the crops is not straightforward, since they are usually sown with regular spacing between individual plants along the each crop row. This results in highly ambiguous data due to the dense and almost uniform distribution of plant features. Inspired by our *Crop Row data association*, our data association on individual plants leverages the crop row structure of the field to represent possible data association matches using an efficient two-dimensional parametrization. The correct data association is then determined based on slight differences in the distribution of the plant features using our novel continuous match counting strategy. This allows our data association to pick up on slight irregularities in the feature distribution, while at the same time being robust to missing, duplicate or inaccurate plant feature detections. Our experiments on real-world data confirm that our novel data association approach can successfully resolve highly ambiguous situations to determine the correct data association. To provide a proof of concept, we applied our novel data association in a mapping framework to re-localize the vehicle after turning and present the resulting maps in a qualitative evaluation. Furthermore, the evaluation results show that our data association technique on individual plants not only enables re-localizing the vehicle after turning, but it also has the potential to provide highly accurate pose estimates throughout the entire field in all three components, especially improving the attainable accuracy of the longitudinal position estimate.

The methods presented in this thesis were used in the autonomous navigation system of the BoniRob to achieve multiple successful fully autonomous runs on entire crop fields. Since our techniques are easily transferable to different platforms, as long as they provide similar sensor modalities, interesting future work would be to apply our techniques on machines that are already widely used in agriculture, such as tractors. This showcases how our methods can help to bridge the gap between research and real-world applications as well as uncover new and unforeseen challenges for fully autonomous vehicles in agriculture.

In this thesis we presented techniques that contribute towards more accurate and robust pose estimation which is an integral part of any localization or mapping method and crucial for precise and reliable autonomous navigation across entire crop fields. Developing techniques, such as ours, towards precise and more reliable autonomous navigation is essential for fully autonomous execution of agricultural tasks on crop fields. Performing agricultural tasks autonomously has not only large potential to increase the efficiency of conventional agriculture, but it can also open the door to many precision farming applications that are otherwise not feasible at all or at least economically not feasible – thereby contributing towards more sustainable precision agriculture.

Bibliography

B. Åstrand and A.-J. Baerveldt. A vision based row-following system for agricultural field machinery. *Mechatronics*, 15(2):251–269, 2005.

G. Baia, Y. Ge, W. Hussain, P. S. Baenziger, and G. Graef. A multi-sensor system for high throughput field phenotyping in soybean and wheat breeding. *Computers and Electronics in Agriculture*, 128:181–192, 2016.

T. Bailey, E. Nebot, J. Rosenblatt, and H. Durrant-Whyte. Data association for mobile robot navigation: a graph theoretic approach. In *International Conference on Robotics and Automation (ICRA)*, 2000.

T. Bakker, H. Wouters, K. van Asselt, J. Bontsema, L. Tang, J. Müller, and G. van Straten. A vision based row detection system for sugar beet. *Computers and Electronics in Agriculture*, 60(1):87–95, 2008.

T. Bakker, K. van Asselt, J. Bontsema, J. M. ller, and G. van Straten. Autonomous navigation using a robot platform in a sugar beet field. *Biosystems Engineering*, 109(4): 357–368, 2011.

D. Ball, P. Ross, A. English, P. Milani, D. Richards, A. Bate, B. Upcroft, G. Wyeth, and P. Corke. Farm workers of the future: vision-based robotics for broad-acre agriculture. *IEEE Robotics & Automation Magazine (RAM)*, 24(3):97–107, 2017.

D. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

Y. Bar-Shalom and T. E. Fortmann. *Tracking and data association*. Academic Press, 1988.

P. Biber, U. Weiss, M. Dorna, and A. Albert. Navigation system of the autonomous agricultural robot "BoniRob". In *International Conference on Intelligent Robots and Systems (IROS)*, 2012.

C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: toward the robust-perception age. *IEEE Transactions on Robotics (TRO)*, 32(6):1309–1332, 2016.

M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: binary robust independent elementary features. In *European Conference on Computer Vision (ECCV)*, 2010.

M. Carcassoni and E. R. Hancock. Spectral correspondence for point pattern matching. *Pattern Recognition*, 36(1):193–204, 2003.

J. Castellanos, J. Montiel, J. Neira, and J. Tardos. The SPmap: a probabilistic framework for simultaneous localization and map building. *IEEE Transactions on Robotics and Automation*, 15(5):948–952, 1999.

N. Chebrolu, P. Lottes, A. Schaefer, W. Winterhalter, W. Burgard, and C. Stachniss. Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields. *International Journal of Robotics Research (IJRR)*, 36(10):1045–1052, 2017.

N. Chebrolu, T. Läbe, and C. Stachniss. Robust long-term registration of UAV images of crop fields for precision agriculture. *Robotics and Automation Letters (RA-L)*, 3(4): 3097–3104, 2018.

N. Chebrolu, P. Lottes, T. Läbe, and C. Stachniss. Robot localization based on aerial images for precision agriculture tasks in crop fields. In *International Conference on Robotics and Automation (ICRA)*, 2019.

A. Cross and E. Hancock. Graph matching with a dual-step EM algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(11):1236–1253, 1998.

J. Dong, J. G. Burnham, B. Boots, G. C. Rains, and F. Dellaert. 4d crop monitoring: spatio-temporal reconstruction for agriculture. In *International Conference on Robotics and Automation (ICRA)*, 2017.

A. English, D. Ball, P. Ross, B. Upcroft, G. Wyeth, and P. Corke. Low cost localisation for agricultural robotics. In *Australasian Conference on Robotics and Automation*, 2013.

A. English, P. Ross, D. Ball, and P. Corke. Vision based guidance for robot navigation in agriculture. In *International Conference on Robotics and Automation (ICRA)*, 2014.

A. English, P. Ross, D. Ball, B. Upcroft, and P. Corke. Learning crop models for vision-based guidance of agricultural robots. In *International Conference on Intelligent Robots and Systems (IROS)*, 2015.

H. J. S. Feder, J. J. Leonard, and C. M. Smith. Adaptive mobile robot navigation and mapping. *International Journal of Robotics Research (IJRR)*, 18(7):650–668, 1999.

M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Association for Computing Machinery*, 24(6):381–395, 1981.

F. Fleckenstein, C. Dornhege, and W. Burgard. Efficient path planning for mobile robots with adjustable wheel positions. In *International Conference on Robotics and Automation (ICRA)*, 2017.

F. Fleckenstein, W. Winterhalter, C. Dornhege, C. Pradalier, and W. Burgard. Smooth local planning incorporating steering constraints. In *International Conference on Field and Service Robotics (FSR)*, 2019.

S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu, and E. Mjolsness. New algorithms for 2d and 3d point matching: pose estimation and correspondence. *Pattern Recognition*, 31(8): 1019–1031, 1998.

A. S. Gomez, E. Aptoula, S. Parsons, and P. Bosilj. Deep regression versus detection for counting in robotic phenotyping. *Robotics and Automation Letters (RA-L)*, 6(2): 2902–2907, 2021.

T. Hague, N. D. Tillett, and H. Wheeler. Automated crop and weed monitoring in widely spaced cereals. *Precision Agriculture*, 7(1):21–32, 2006.

P. Hough. Method and means for recognizing complex patterns. U.S. Patent 3.069.654, 1962.

M. Kise, Q. Zhang, and F. R. Más. A stereovision-based crop row detection method for tractor-automated guidance. *Biosystems Engineering*, 90(4):357–367, 2005.

R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: a general framework for graph optimization. In *International Conference on Robotics and Automation (ICRA)*, 2011.

R. Kümmerle, M. Ruhnke, B. Steder, C. Stachniss, and W. Burgard. Autonomous robot navigation in highly populated pedestrian zones. *Journal of Field Robotics (JFR)*, 2014.

V. Leemans and M.-F. Destain. Line cluster detection using a variant of the hough transform for culture row localisation. *Image and Vision Computing*, 24(5):541–550, 2006.

J. Levinson and S. Thrun. Robust vehicle localization in urban environments using probabilistic maps. In *International Conference on Robotics and Automation (ICRA)*, 2010.

J. Libby and G. Kantor. Deployment of a point and line feature localization system for an outdoor agriculture vehicle. In *International Conference on Robotics and Automation (ICRA)*, 2011.

D. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision (ICCV)*, 1999.

F. Magistri, N. Chebrolu, J. Behley, and C. Stachniss. Towards in-field phenotyping exploiting differentiable rendering with self-consistency loss. In *International Conference on Robotics and Automation (ICRA)*, 2021.

J. Marchant. Tracking of row structure in three crops using image analysis. *Computers and Electronics in Agriculture*, 15(2):161–179, 1996.

M. Montalvo, G. Pajares, J. Guerrero, J. Romeo, M. Guijarro, A. Ribeiro, J. Ruz, and J. Cruz. Automatic detection of crop rows in maize fields with high weeds pressure. *Expert Systems with Applications*, 39(15):11889–11897, 2012.

M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2003.

T. Mueller-Sim, M. Jenkins, J. Abel, and G. Kantor. The robotanist: a ground-based agricultural robot for high-throughput crop phenotyping. In *International Conference on Robotics and Automation (ICRA)*, 2017.

J. Neira and J. Tardos. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897, 2001.

J. Neira, J. Tardos, and J. Castellanos. Linear time vehicle relocation in SLAM. In *International Conference on Robotics and Automation (ICRA)*, 2003.

C. Pradalier and S. Sekhavat. Concurrent matching, localization and map building using invariant features. In *International Conference on Intelligent Robots and Systems (IROS)*, 2002.

A. Pretto, S. Aravecchia, W. Burgard, N. Chebrolu, C. Dornhege, T. Falck, F. Fleckenstein, A. Fontenla, M. Imperoli, R. Khanna, F. Liebisch, P. Lottes, A. Milioto, D. Nardi, S. Nardi, J. Pfeifer, M. Popović, C. Potena, C. Pradalier, E. Rothacker-Feder, I. Sa, A. Schaefer, R. Siegwart, C. Stachniss, A. Walter, W. Winterhalter, X. Wu, and J. Nieto. Building an aerial-ground robotics system for precision farming: an adaptable solution. *IEEE Robotics & Automation Magazine (RAM)*, 28(3):29–49, 2021. IEEE Robotics & Automation Magazine Best Paper Award.

G. Riggio, C. Fantuzzi, and C. Secchi. A low-cost navigation strategy for yield estimation in vineyards. In *International Conference on Robotics and Automation (ICRA)*, 2018.

E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: an efficient alternative to SIFT or SURF. In *International Conference on Computer Vision (ICCV)*, 2011.

A. Ruckelshausen, P. Biber, M. Dorna, H. Gremmes, R. Klose, A. Linz, R. Rahe, R. Resch, M. Thiel, D. Trautz, and U. Weiss. BoniRob: an autonomous field robot platform for individual plant phenotyping. *Precision Agriculture*, 9:841–847, 2009.

C. Smitt, M. Halstead, T. Zaenker, M. Bennewitz, and C. McCool. Pathobot: A robot for glasshouse crop phenotyping and intervention. In *International Conference on Robotics and Automation (ICRA)*, 2021.

H. Søgaard and H. Olsen. Determination of crop rows by image analysis without segmentation. *Computers and Electronics in Agriculture*, 38(2):141–158, 2003.

S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

N. Tillett and T. Hague. Computer-vision-based hoe guidance for cereals—an initial trial. *Journal of Agricultural Engineering Research*, 74(3):225–236, 1999.

N. Tillett, T. Hague, and S. Miles. Inter-row vision guidance for mechanical weed control in sugar beet. *Computers and Electronics in Agriculture*, 33(3):163–177, 2002.

J. Underwood, A. Wendel, B. Schofield, L. McMurray, and R. Kimber. Efficient in-field plant phenomics for row-crops with an autonomous ground vehicle. *Journal of Field Robotics (JFR)*, 34(6):1061–1083, 2017.

I. Vidović, R. Cupec, and Ž. Hocenski. Crop row detection by global energy minimization. *Pattern Recognition*, 55:68–86, 2016.

A. Walter, R. Finger, R. Huber, and N. Buchmann. Opinion: Smart farming is key to developing sustainable agriculture. *Proceedings of the National Academy of Sciences*, 114(24):6148–6150, 2017.

K. Watanabe. Ariadne with ambiguity resolution: visual marker based rapid initialization of ppp-ar. In *International Conference on Intelligent Robots and Systems (IROS)*, 2018.

G. Weiss, C. Wetzler, and E. von Puttkamer. Keeping track of position and orientation of moving indoor systems by correlation of range-finder scans. In *International Conference on Intelligent Robots and Systems (IROS)*, 1994.

W. Winterhalter, F. Fleckenstein, B. Steder, L. Spinello, and W. Burgard. Accurate indoor localization for RGB-D smartphones and tablets given 2D floor plans. In *International Conference on Intelligent Robots and Systems (IROS)*, 2015.

W. Winterhalter, F. Fleckenstein, C. Dornhege, and W. Burgard. Crop row detection on tiny plants with the pattern hough transform. *Robotics and Automation Letters (RA-L)*, 3(4):3394–3401, 2018.

W. Winterhalter, F. Fleckenstein, C. Dornhege, and W. Burgard. Localization for precision navigation in agricultural fields—beyond crop row following. *Journal of Field Robotics (JFR)*, 38(3):429–451, 2021.

H. Wolfson and I. Rigoutsos. Geometric hashing: an overview. *IEEE Computational Science and Engineering*, 4(4):10–21, 1997.

M. F. S. Xaud, A. C. Leite, and P. J. From. Thermal image based navigation system for skid-steering mobile robots in sugarcane crops. In *International Conference on Robotics and Automation (ICRA)*, 2019.

Y. Xiong, Y. Ge, L. Grimstad, and P. J. From. An autonomous strawberry-harvesting robot: Design, development, integration, and field evaluation. *Journal of Field Robotics (JFR)*, 37(2):202–224, 2020.

L. Yang, J.-M. Normand, and G. Moreau. Local geometric consensus: a general purpose point pattern-based tracking algorithm. *IEEE Transactions on Visualization and Computer Graphics*, 21(11):1299–1308, 2015.