UNIVERSITÄT BONN

Inaugural-Dissertation
zur Erlangung des Grades
Doktor der Ingenieurwissenschaften (Dr.-Ing.)
der Landwirtschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn
Institut für Geodäsie und Geoinformation

Spatio-Temporal Registration Techniques for Agricultural Robots

von

Nived Chebrolu

aus
Guntur, Indien

Bonn 2021

Referent:
Prof. Dr. Cyrill Stachniss, University of Bonn, Germany

Korreferent:
Prof. Dr. Roland Siegwart, ETH Zurich

Tag der mündlichen Prüfung:
27. Mai 2021

Angefertigt mit Genehmigung der Landwirtschaftlichen Fakultät der Universität Bonn

Erklärung der Urheberschaft

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenenen Hilfsmittel angefertigt habe; die aus fremdem Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form in keiner anderen Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Ort, Datum	(Unterschrift)

Zusammenfassung

Die stark wachsende Weltbevölkerung erfordert aufgrund des kontinuierlich steigenden Bedarfs an Lebensmitteln, Futtermitteln, und Biokraftstoffen neuartige Ansätze in der landwirtschaftlichen Produktion. Die begrenzten Anbauflächen und verschärften Anforderungen an den Umweltschutz stellen eine große Herausforderung dar. So führt der derzeit übliche Einsatz von Agrochemikalien zu Umweltschäden und einem beschleunigtem Artensterben. Die notwendige Intensivierung der landwirtschaftlichen Produktion kann vermutlich ökonomisch und ökologisch nur durch eine nachhaltigere Nutzung der vorhandenen Ressourcen in Verbindung mit neuen Technologien erreicht werden.

Agrarroboter sind ein mögliches Hilfsmittel um dieses Ziel zu erreichen. Diese Systeme können die Produktivität durch eine automatisierte, kontinuierliche und selektive Behandlung einzelner Pflanzen steigern und gleichzeitig den Einsatz von Agrochemikalien deutlich reduzieren. Die Entwicklung solcher automatisierten Roboter wird in der Zukunft der landwirtschaftlichen Produktion eine wesentliche Rolle spielen. Solche Roboter sind mit Sensoren, verschiedenen Aktuatoren und mechanischen Werkzeugen ausgestattet. Dadurch können Pflanzen individuell mit einer hohen räumlichen und zeitlichen Auflösung überwacht und bedarfsgerecht, selektiv behandelt werden.

Der Fokus der vorliegenden Arbeit liegt in der Registrierung von Sensordaten, die wesentlicher Bestandteil dieser Robotersysteme ist. Ziel der Registrierung ist die Verknüpfung von räumlich und zeitlich getrennt erfassten Daten durch eine Transformation in ein einheitliches Koordinatensystem. Sie ist ein zentraler Bestandteil der Zustandsschätzung in der Robotik, Geodäsie und Photogrammetrie. Aus diesem Grund wurde die Registrierung von Sensordaten in der Literatur von verschiedenen Disziplinen ausgiebig untersucht. Die bestehenden Verfahren sind jedoch aufgrund einer Reihe einzigartiger Herausforderungen im landwirtschaftlichen Bereich nicht zuverlässig einsetzbar. Starke, saisonal bedingte, visuelle und strukturelle Veränderungen einzelner Pflanzen sowie der gesamten Ackerfläche in den verschiedenen Wachstumsphasen erschweren beispielsweise die Aufgabe der Registrierung. Wenn Daten von unterschiedlichen luft- und bodengestützten

Systemen erfasst werden, führt dies zu Abweichungen durch unterschiedliche Perspektiven. In dieser Arbeit präsentieren wir daher verschiedene, neuartige Techniken zur Registrierung, welche die oben genannten Herausforderungen explizit berücksichtigen. Wir zeigen, dass unsere Verfahren zur Registrierung auch unter komplexen Bedingungen zuverlässig funktionieren sowie ihre Vorteile gegenüber herkömmlichen Methoden. Dabei stellen wir dar, wie diese Registrierungsverfahren für die Langzeitüberwachung von Nutzpflanzen, die Lokalisierung und der Navigation von bodengestützten Robotern im Feld einsetzbar sind. Des Weiteren führen wir eine automatisierte Phänotypisierung durch, um das Wachstum einzelner Pflanzenteile aus Punktwolkendaten zu analysieren. Wir untersuchen auch die Auswirkungen von Ausreißern in Daten für Registrierungs-und Zustandsschätzungsprobleme und schlagen eine allgemeine Lösung für eine robuste Zustandsschätzung in Anwesenheit von verschiedenen Ausreißern vor, die bei diesen Aufgaben auftreten. Die in dieser Arbeit entwickelten Registrierungsverfahren tragen zum robusten Betrieb von autonomen Robotern, auch über lange Zeiträume hinweg, bei. Sie bilden das Rückgrat von Anwendungen, die räumlich-zeitliche Merkmalen von Pflanzen untersuchen.

Zusammengefasst leistet diese Arbeit mehrere Beiträge im Kontext der räumlichen und zeitlichen Registrierung im pflanzenbaulichen Bereich. Im Vergleich herkömmlichen Methoden ermöglichen unsere Ansätze eine robustere und längerfristige Registrierung der erfassten Daten und gehen effektiv mit den Herausforderungen um, die sich aus dem natürlichen Pflanzenwachstum ergeben. Alle in dieser Arbeit beschriebenen Ansätze sind in begutachteten Konferenzbeiträgen und Zeitschriftenartikeln veröffentlicht worden. Darüber hinaus haben wir die meisten der in dieser Arbeit entwickelten Techniken als Open-Source-Software der wissenschaftlichen Community zur Verfügung gestellt und auch drei anspruchsvolle Datensätze für langfristige räumlich-zeitliche Registrierungsaufgaben veröffentlicht.

Abstract

A critical challenge that we face today is to meet the rising demand for food, feed, fiber, and fuel from an ever-growing world population. We must meet this demand within the limited arable land available to us and do so in the aggravated situation caused by climate change. Moreover, present-day levels of agro-chemical usage is unsustainable. They lead to large scale environmental pollution and adverse effects on the biodiversity of our planet. A promising way to meet this challenge is through intensifying production sustainably using existing resources and novel technology in combination. Robotic systems deployed in agricultural fields are seen as a potential solution to achieve this goal. These systems can increase productivity by providing high-quality site-specific treatment at the level of an individual plant through continuous monitoring and timely intervention in the field, while drastically reducing or eliminating the use of agro-chemicals. The development of such automated robotic systems is envisioned to play an essential role in the future of agricultural plant production.

Agricultural robots are ideal platforms to monitor the plants in the field with a high spatial and temporal frequency and provide intervention capability whenever an action is required. In this thesis, we focus on the fundamental task of registration, which would form the core of such robotic systems. The goal of registration is to bring two sets of measurements into a common coordinate frame, which forms the basis for associating data separated in space and time. It is a core building block for solving several state estimation problems in robotics, geodesy, and photogrammetry. As a result, registration of sensor data has been extensively studied in the literature from multiple disciplines. However, existing techniques fail to perform reliably in the agricultural domain due to a unique set of challenges. These challenges vary from the large change in the visual appearance of the field over time to the structural change of individual plants as they grow over the crop season and to the vastly differing viewpoints where data is captured from multiple platforms in an aerial-ground robotics system.

Our main contribution in the thesis is a set of novel registration techniques that explicitly considers the challenges brought forward by the spatio-temporal nature of the task in agricultural application. We show that our registration techniques perform reliably in challenging conditions and demonstrate their advantages over state-of-the-art registration approaches. We use these registration techniques to demonstrate their application for long-term monitoring of crops in the field, for accurate localization of ground robots for navigation in crop fields, and for performing automated phenotyping to analyze the growth of individual plant parts from high-fidelity point cloud data. We also study the effect of outliers in data for registration and state estimation problems and propose a general solution for robust state estimation in the presence of different outlier distributions that occur in these tasks. The registration techniques developed in this thesis contribute to the robust operation of autonomous robots in crop fields over long periods of time and form the backbone of applications interested in tracking spatio-temporal traits of plants.

In sum, this thesis makes several contributions in the context of spatio-temporal registration in the agricultural domain of plant production. Compared to the current state-of-the-art, the approaches presented in this thesis allow for a more robust and longer-term registration of data captured by robots in the fields and effectively handle the challenges resulting from plant growth. All approaches described in this thesis have been published in peer-reviewed conference papers and journal articles. In addition to that, we have released most of the techniques developed in this thesis as open-source software and also published three challenging datasets for long-term spatio-temporal registration tasks.

Acknowledgements

The journey culminating in this thesis has been one that I would cherish for the rest of my life. I would like to take this opportunity to thank all the people who made it possible.

Firstly, I would like to thank Cyrill Stachniss for being the best mentor I could ever ask for. I want to thank him for the constant encouragement and support throughout the last many years. I am grateful for the innumerable things that I learned from him, including writing papers, making presentations, teaching in the classroom, and so many more things. I am amazed at the incredible patience he has shown, particularly when things didn't always go as planned, and always giving me the time and space to explore and experiment, and the confidence needed to overcome the challenges. I thank him for creating an environment where it has been an absolute pleasure to work and making all of us here at the group feel like being part of a larger adventure. I would always cherish the memories of the fun-filled conference trips and the many conversations over amazing cups of cappuccino that he made. The example he has set both at work and outside, is one that I would always aspire to live up to.

I would like to thank my friends and colleagues who have made this journey an exciting and colorful one. I would like to thank Olga Vysotska for being a friend and an ally I could always go to, for all the wonderful and edifying conversations we have had, and for making this place feel like a home away from home. I would like to thank Lorenzo Nardi for making me feel welcome from the very first day I arrived at work, and always agreeing to join every single project I asked for his help, no matter however tedious or impractical they were. I would like to thank Philipp Lottes for being the coolest office mate I could have, for the infectious energy that he brought with him every single day at work. I will remember the sweet and bitter experiences of collecting data with him in the fields in all kinds of weather, under blazing sun and freezing cold. I am grateful to Thomas Läbe for the numerous brainstorming sessions we have had and for helping me with the implementations. I would like to thank Igor Bogoslavskyi, Johannes Schneider, Sussane Wenzel, Kaihong Huang, Jens Behley, Emanuele Palazzolo, Andres Milioto, Xieyuanli Chen, Ignacio Vizzo, Jan Weyler, Louis Wiesmann, Federico Magistri, Benedikt Mersch and Rodrigo Marcuzzi. I could not have asked for a more supportive team to have worked with, and each one of them has helped me become better. I would like to give special thanks to Birgit Klein, who has not only helped me navigate the administrative maze but has amazed me with her kindness and willingness to help in every situation.

I am grateful to have the opportunity to participate and contribute to the EU project Flourish. I would like to thank all the colleagues from the various groups participating in the project. I will remember the numerous integration meets running late into the nights, and fixing issues seconds before the demonstrations, all of which was a truly joyful and rewarding experience, and something I will remember with great fondness.

Most of all, I would like to thank my family, who have always supported and encouraged me to pursue my dreams.

Contents

Chapter 1

Introduction

Demand for food, feed, fiber, and fuel is on the rise due to an ever-growing world population. It is estimated that we need to nearly double the global crop yield by 2050 to meet the projected demands [43]. This goal needs to be achieved within the limited arable land available while drastically reducing the environmental footprint of agricultural production. Furthermore, present-day levels of agro-chemical usage is leading to large-scale environmental pollution and detrimental effects on bio-diversity. The situation is aggravated by climate change putting further stress on the limited resources available [10]. Therefore, achieving sufficient crop production to meet the growing demands in a sustainable manner has become an issue of critical importance.

One of the ways to meet this demand is to intensify production but do that sustainably. This means using technologies that increase production per hectare without negative environmental consequences. The next generation of agriculture techniques realized through automated robotic systems are key technologies to approach this goal. These systems can allow continuous monitoring of crops and their environment. Novel forms of agriculture envision providing necessary care at the individual plant level throughout the whole crop season through robots. The ability to continuously monitor the status of the crop and its environment ensures that timely action is taken to maximize the yield and reduce any potential loss of produce. Another dimension of precision agriculture is providing targeted mechanisms for various field management activities. For example, instead of applying chemicals uniformly over the entire field to manage the weeds, precision agriculture systems are able to identify individual weeds and selectively spray required chemicals on the targeted weeds. This can lead to a drastic reduction in the overall usage of agro-chemicals and limits the negative environmental impact.

Robotic systems have the capacity to accomplish precision agricultural tasks effectively. For example, an integrated robotic system consisting of multiple platforms working collaboratively can provide flexible solutions to accomplish

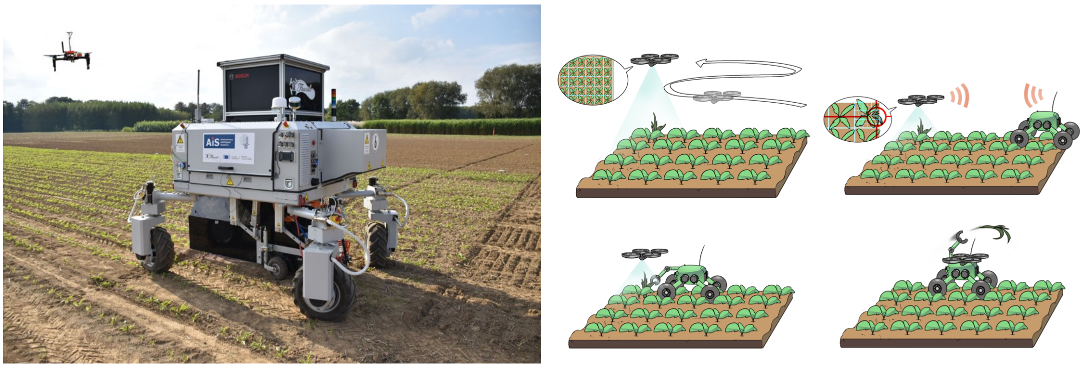


Figure 1.1: Example of a multi-robot system for precision agriculture. Left: A team of UAV and UGV equipped with sensors and actuators to perceive and execute actions in the field. These robots collaborate with each other to carry out precision agriculture tasks jointly in the field. Right: Concept of a robotic system for performing continuous crop monitoring and weed management used in the EU project Flourish [94], courtesy of ASL, ETH Zurich.

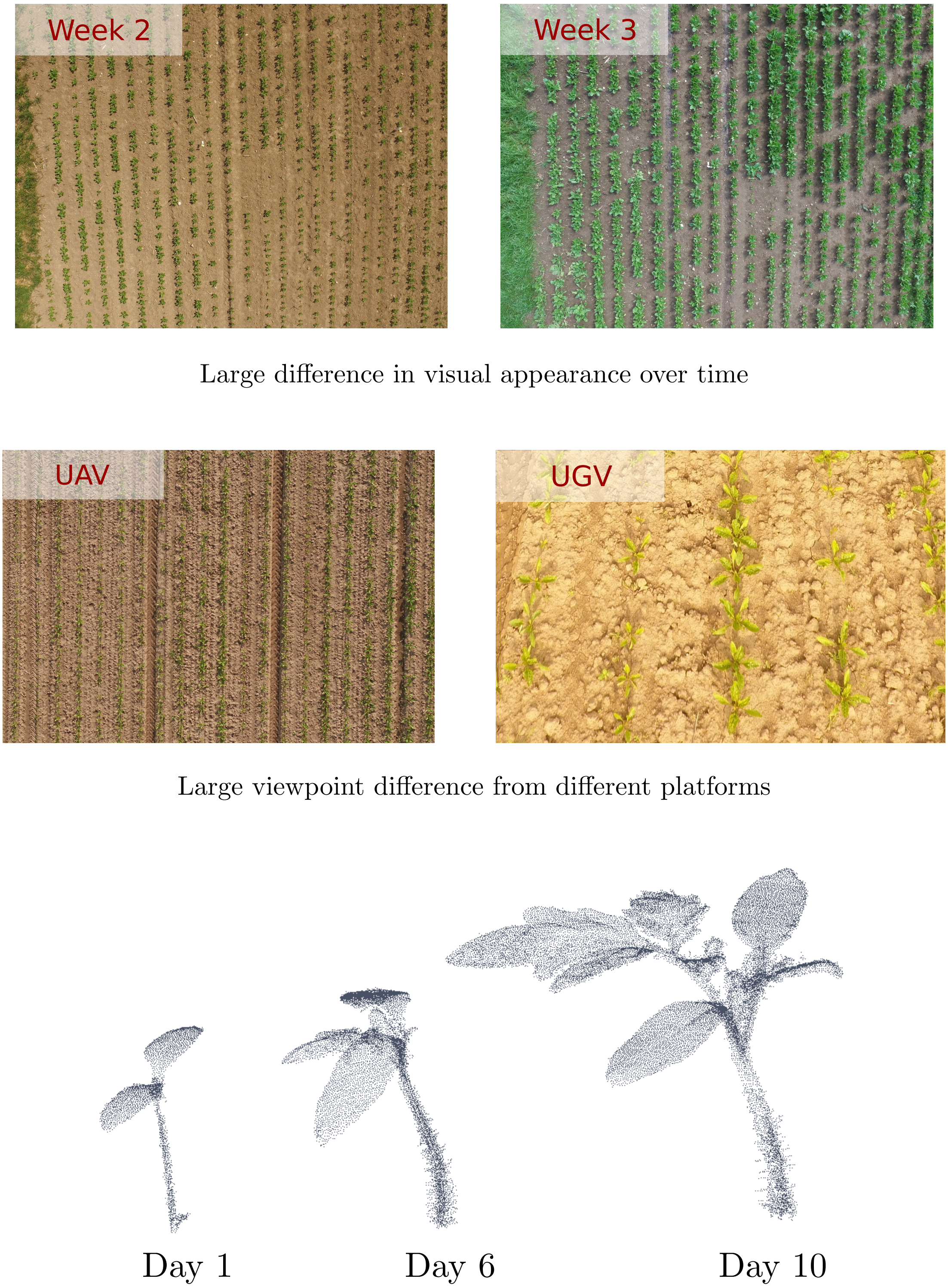
different precision agricultural tasks. Such a system could consist of unmanned aerial vehicles (UAVs) equipped with sensors to analyze the crop field and unmanned ground robots (UGVs), which can carry out farm management tasks on the field through its onboard implements. These platforms are often equipped with sensors that complement each other and help perform tasks effectively on the field. An example of such a system used in the EU project Flourish is shown in Figure 1.1 (left), where a team of a UAV and a UGV operate jointly, carrying continuous monitoring and weed management for crop fields. This system's typical operation is illustrated in Figure 1.1 (right). During this operation, a UAV first performs a survey flight over the field and identifies regions where intervention actions such as removing the weeds are required. The UAV then communicates this information to the UGV, which then proceeds to the identified regions and uses its onboard implements to execute the intervention action. These robotic systems can bring to practice the ideals envisioned by precision agriculture of providing care at the individual plant level and eventually lead to an increase in productivity in a sustainable manner.

One of the fundamental capabilities for developing such robotic systems is to register the data captured by the onboard sensors of the robots. The task of registration is essentially of bringing two sets of measurements into a common coordinate frame. These sets of measurements can be separated over time, i.e., measurements of the surroundings are acquired at different points of time by the same robot. Similarly, the measurements can be separated over space, i.e., measurements are acquired simultaneously but from different viewpoints, possibly from different robots. Or a combination of both, i.e., the measurements are separated both in time and space, which leads us to the spatio-temporal registration task.

Registration is of critical importance as it forms the core of several state estimation problems in robotics as well as applications to analyze the surroundings. For example, by registering measurements captured by a robot against a map of the environment, we can figure out where the robot is at each instance of time, which is also referred to as the localization task. Registration is also critical for understanding the environment in which the robots operate. In the agricultural domain, this includes registering plant data, which is acquired at different points of time and from different robots to analyze its growth. In this thesis, we will focus on developing novel spatio-temporal registration techniques, which would contribute towards the operation of autonomous robots in the field over long periods of time and provide a basis to analyze the growth at an individual plant level.

The task of registering spatio-temporal data is riddled with several challenges, some of which are rather unique to the agricultural domain. The registration methods must perform robustly in the presence of large spatial and temporal changes in the field environment. We highlight some of these challenges that we tackle in this thesis in Figure 1.2. For example, the appearance of the field changes dramatically between two images captured a week apart by a UAV, making the task of registering these images difficult, as illustrated in Figure 1.2 (top). Another critical challenge that arises is to register data observed from different viewpoints, such as images taken from UAVs to those from UGVs as seen in Figure 1.2 (middle). Finally, a further challenge in registering data from the agricultural domain is introduced due to the plant growth dynamics. The change in the size, shape, and topology of the plant over time, as shown in Figure 1.2 (bottom), requires a sophisticated approach to register them against each other. All these challenges put together render the spatio-temporal registration task in agricultural applications a difficult one.

In this thesis, we present novel spatio-temporal registration approaches, which can perform robustly facing the various challenges that arise in the agricultural robotics domain. This includes techniques for registering images acquired by UAVs over a crop season, and registering data from UAVs and UGVs to enable collaborative operations, and registering high fidelity plant pointclouds over time for phenotyping applications. Each of these registration tasks typically involves a data association step, which is often affected by outliers. We address this by developing a robust estimation framework that adapts to the prevailing outlier distribution dynamically and does not rely on a specific, previously chosen robust kernel. This framework to deal with outliers is also applicable outside of the agricultural domain, including several registration and state estimation problems in robotics.



Large difference in visual appearance over time

Large viewpoint difference from different platforms

Large change in shape of the plant due to growth

Figure 1.2: Challenges of spatio-temporal registration in the agricultural domain. Top: Large difference in the visual appearance of the field over time due to growth of the plants, changing lighting conditions, state of the soil etc. Middle: Strong viewpoint difference between images from different robotic platforms, here shown for an UAV camera looking down from a height of 15 m and a UGV image in a perspective view from about 3 m above ground. Bottom: Time series of tomato plant point clouds capturing the change in the size, shape and structure as new leaves emerge over time.

1.1 Main Contributions

The main contribution of this thesis is a set of novel spatio-temporal registration techniques designed to meet relevant challenges faced in agricultural robotics. The techniques we propose in this thesis provide the basis for long-term operation of robots in the field environment and form the backbone for phenotypic applications such as analyzing the plant growth over time.

The first contribution, which is presented in Chapter 2, provides a solution for monitoring crop fields using UAVs over long periods of time. We develop a registration technique for associating images captured over an entire season by exploiting the inherent geometric structure of a crop field. Our approach provides robust correspondences between images despite the dramatic change in the visual appearance as illustrated in Figure 1.2 (top). We achieve this by proposing a novel feature descriptor that exploits crop and gap location information along the crop rows, which is mostly invariant within the same field over time. This spatial information about crops and gaps provides critical cues to our approach, allowing us to successfully register images even in situations where matching based on state-of-the-art visual descriptors fail completely. Based on the registration results, we show that it is possible to analyze an individual plant's growth in the field, which in-turn is invaluable data for the farmers and crop scientists to make informed decisions.

The second contribution of this thesis is to enable collaboration between UAVs and UGVs and exploit the flexibility of using multiple robots to navigate in crop fields. In Chapter 3, we explore the advantages of collaboration with a UAV to improve the localization capabilities of a UGV. We develop a localization framework for the UGV, which exploits aerial images of the field captured using a UAV. We propose a novel data association scheme to register the data observed from UAV and UGV with large viewpoint differences as seen in Figure 1.2 (middle). This leads to a novel pointwise feature descriptor targeted to crop fields. We are able to solve this data association challenge by exploiting the geometry as well as the semantics of the crop field, and additionally, integrate it within a Monte-Carlo localization framework to estimate the pose of the ground robot in the UAV map in an online fashion. We also show that our approach provides reliable localization with crop-row level accuracy over several sessions despite the large changes in the field and provides better pose estimates than by a single-phase GPS-based localization.

As a third contribution, we explore techniques suited for registering high-fidelity point cloud data for phenotyping tasks. In Chapter 4, we present a novel approach for spatio-temporal registration of 3D point clouds of individual plants. The proposed approach works for raw sensor data stemming from a range sensor such as a 3D LiDAR or a depth camera. We perform the registration in a fully automated fashion without any manual intervention. Our approach is designed to perform robustly given the challenges visualized in Figure 1.2 (bottom) and register the temporal data from growing plants. We capture the non-rigid deformation as well as the change in shape that occurs during plant growth by exploiting the skeletal structure and the semantics of the plant. This technique gives us the capability to analyze individual parts of the plant and analyze their growth over time. This level of precise and high-resolution analysis of plant growth is required for phenotyping and is used by scientists involved in crop breeding and other related applications. We evaluate our approach on datasets from two plant species and demonstrate an automated phenotyping application of tracking plant traits over time.

As the final contribution of the thesis, we propose a solution to deal with challenge caused by outliers and its effects on the robustness of registration and other state estimation techniques. The outliers arise typically from the data association process. These outliers could be due to different reasons such as incorrect matches between features computed in UAV images due to lack of descriptor specificity (Chapter 2), or ambiguous correspondences made between features computed in UGV images to the features in the reference maps generated from UAV images due to aliasing (Chapter 3), or wrong correspondences made between skeleton nodes of temporally separated plant point cloud data (Chapter 4). In Chapter 5, we thus propose a general solution for robust state estimation in the presence of different outlier distributions that occur in these tasks. The technique we propose looks into general robust estimation and applies to several other estimation tasks common to robotics and computer vision applications. It avoids specifying a particular robust kernel such as Huber or Cauchy beforehand and allows to adapt the shape of the kernel during the optimization process.

In sum, we propose novel methods that form building blocks of the perception and navigation system of future agricultural robots.

1.2 Publications

Parts of this thesis have been published in the following peer-reviewed conference and journal articles, or are currently under review:

- N. Chebrolu, P. Lottes, A. Schäfer, W. Winterhalter, W. Burgard, and C. Stachniss. Agricultural Robot Dataset for Plant Classification, Localization and Mapping on Sugar Beet Fields. *Intl. 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. *IEEE 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 *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019

- N. Chebrolu, T. Läbe, and C. Stachniss. Spatio-Temporal Non-Rigid Registration of 3D Point Clouds of Plants. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2020

- N. Chebrolu, F. Magistri, T. Läbe, and C. Stachniss. Registration of Spatio-Temporal Point Clouds of Plants for Phenotyping. *PLOS ONE*, 16(2):1–25, 2021

- N. Chebrolu, T. Läbe, and C. Stachniss. Adaptive Robust Kernels for Non-Linear Least Squares Problems. *IEEE Robotics and Automation Letters (RA-L)*, 6(2):2240–2247, 2021

- D. Schunck, F. Magistri, R. A. Rosu, A. Cornelißen, N. Chebrolu, S. Paulus, J. Léon, S. Behnke, C. Stachniss, H. Kuhlmann, and L. Klingbeil. Pheno4D: A Spatio-Temporal Dataset of Maize and Tomato Plant Point Clouds for Phenotyping and Advanced Plant Analysis. *PLOS ONE*, 2021. Revised version submitted, under review.

The following are publications I was involved in during my doctorate, but which are not covered in the chapters of this thesis:

- P. Lottes, J. Behley, N. Chebrolu, A. Milioto, and C. Stachniss. Joint Stem Detection and Crop-Weed Classification for Plant-specific Treatment in Precision Farming. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2018

- P. Lottes, N. Chebrolu, F. Liebisch, and C. Stachniss. UAV-based Field Monitoring for Precision Farming. In *25. Workshop Computer-Bildanalyse in der Landwirtschaft*, 2019

- F. Magistri, N. Chebrolu, and C. Stachniss. Segmentation-Based 4D Registration of Plants Point Clouds for Phenotyping. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020

- A. Ahmadi, L. Nardi, N. Chebrolu, and C. Stachniss. Visual Servoing-based Navigation for Monitoring Row-Crop Fields. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2020

- P. Lottes, J. Behley, N. Chebrolu, A. Milioto, and C. Stachniss. Robust Joint Stem Detection and Crop-Weed Classification using Image Sequences for Plant-Specific Treatment in Precision Farming. *Journal of Field Robotics (JFR)*, 37:20–34, 2020

- 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. *IEEE Robotics and Automation Magazine (RAM)*, 2020

- I. Vizzo, X. Chen, N. Chebrolu, J. Behley, and C. Stachniss. Poisson Surface Reconstruction for LiDAR Odometry and Mapping. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2021

- F. Magistri, N. Chebrolu, J. Behley, and C. Stachniss. Towards In-Field Phenotyping Exploiting Differentiable Rendering with Self-Consistency Loss. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2021

To facilitate further research in spatio-temporal registration techniques for precision agriculture applications, we have open-sourced our code for the community:

- Temporal UAV image matching using similarity invariant geometric feature: `https://github.com/PRBonn/sigf`

- Python toolkit for working with datasets captured with agricultural robot: `https://github.com/PRBonn/pybonirob`

- Spatio-temporal registration of plant point clouds: `https://github.com/PRBonn/4d_plant_registration`

We have further published the following challenging datasets for long-term spatio-temporal registration tasks:

- Long-term UAV image registration dataset from sugarbeet fields together with the Autonomous Sustems Lab and the Crop Science Group at ETH Zürich:
  `https://www.ipb.uni-bonn.de/data/uav-sugarbeets-2015-16/`
- Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields together with Autonomous Intelligent Systems Lab, University of Freiburg:
  `https://www.ipb.uni-bonn.de/data/sugarbeets2016/`
- 4D plant point cloud registration dataset developed together with Institute for Geodesy and Geo-information, University of Bonn:
  `https://www.ipb.uni-bonn.de/data/4d-plant-registration/`

Chapter 2

Spatio-Temporal Registration of UAV Images of Crop Fields

Continuous crop monitoring is an important aspect of modern farming, and a necessary step towards providing care at an individual plant level throughout the crop season. It allows the farmers to make informed decisions regarding when, where, and how much fertilizer or pesticide to apply in the field as well as to improve yield estimation. It also plays an important role in understanding plant growth and provides critical information to scientists involved in the crop breeding process. In this chapter, we investigate how UAV images can enable monitoring crop fields over long periods of time.

With the wide availability of commercial UAVs, it has become fairly easy to repeatedly acquire image data of the fields without any expert assistance. However, in order to exploit this data captured by UAVs for monitoring applications, it is necessary to register the data captured at different locations and time instances into a common coordinate frame. This kind of spatio-temporal registration forms the backbone of any application that analyzes plant growth over time. For typical scenes urban scenes, such as buildings and other permanent structures, state-of-the-art image registration methods are able to register them and compute 3D models of the environment [40]. Typically, these methods rely on a visual descriptor such as SIFT, ORB, BRIEF or similar to perform the data association amongst the images. A critical assumption that these methods make is that the appearance of the scene remains the same over time. In crop farming, however, fields and crops are affected by strong visual changes due to the weather, growing crops, and farm equipment such as tractors affecting the soil as shown in Figure 2.2. As a result, most registration methods are not able to cope well with these changes in appearance.

In this chapter, we address the problem of registering UAV images of a field recorded over the crop season in the presence of large visual changes caused by

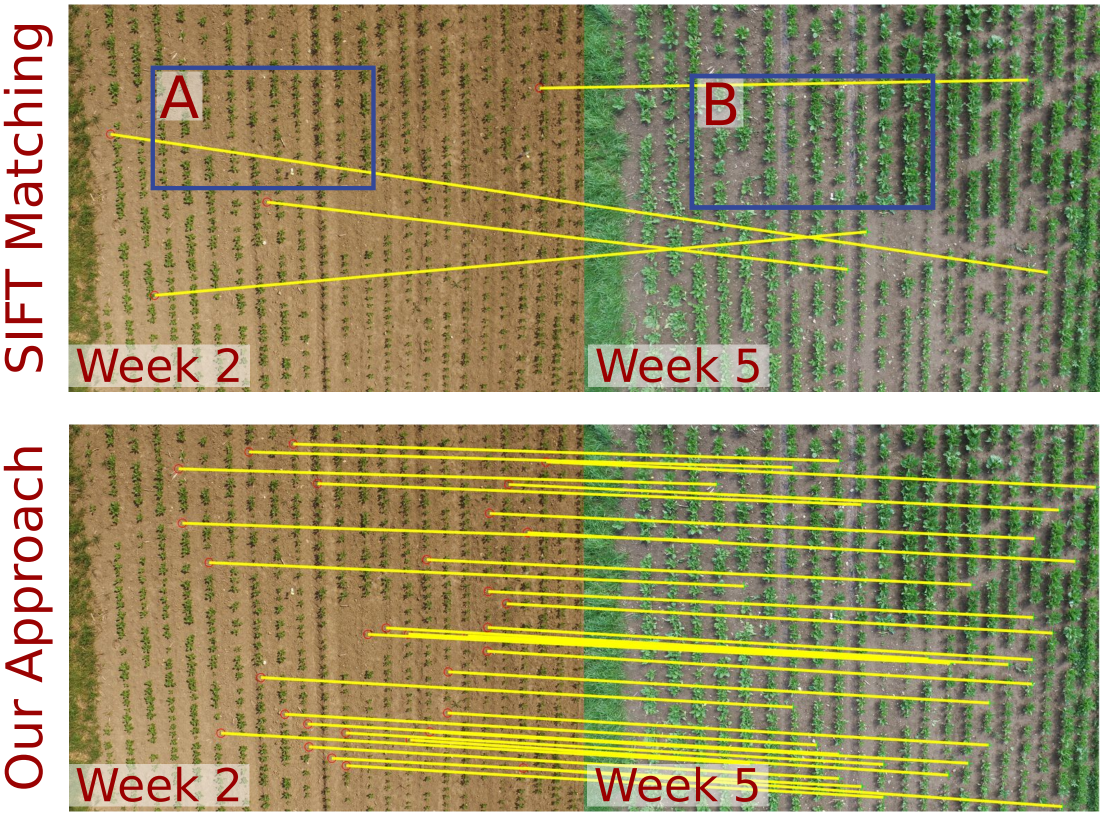


Figure 2.1: Matching UAV images taken three weeks apart. Our method uses geometric cues to perform matching successfully whereas matching using SIFT fails in challenging conditions with large visual changes. Figure 2.2 shows a zoomed-in view of the field shown in the blue boxes above to highlight the changes.

crop growth and field management. The main idea of our approach is to take advantage of the fact that the position of crops as well as gaps between crops remains roughly the same over time, even if the visual appearance of the plants themselves changes dramatically. A two-image matching example is depicted in Figure 2.1. The first row shows SIFT-based correspondences. As we can see from the lines connecting the identified corresponding points, SIFT based association is rather poor. Our approach, however, finds better correspondences, as seen in the second row.

The main contribution of this chapter is a novel method for registering images of a crop field taken using a UAV across the crop season. Our approach provides robust correspondences between images under changing conditions caused by crop growth, weather, and field management. It also copes with the visual aliasing problem in crop fields. We achieve this by presenting a descriptor that exploits crop and gap location information along the crop rows, which is mostly invariant within the same field over time. We exploit this spatial information about crops and gaps for matching images when the visual appearance is drastically changing.

Finally, using our approach for matching images taken from a UAV during

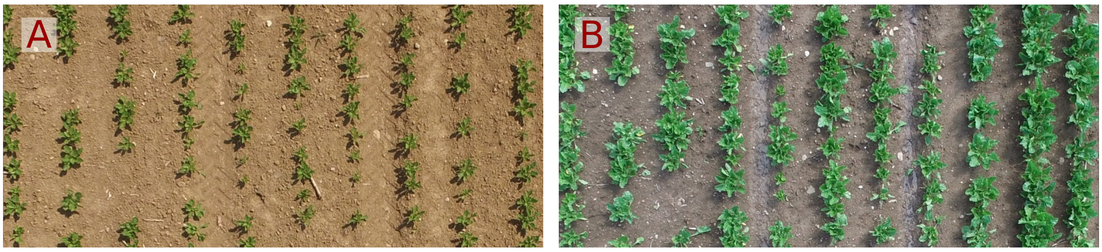


Figure 2.2: Zoomed-in view of the same area on the field three weeks apart. In addition to the vegetation growth, the texture of soil also changes dramatically over time. Texture rich regions such as the tire marks from the tractor in the left image are washed away in the rain while revealing other new objects like the stones embedded in the ground. Such strong changes make it very challenging for visual matching methods to work reliably.

multiple sessions, we can compute a 3D model of the field with a temporal dimension capturing the evolution of growing plants in the field. This model, in turn, allows us to monitor crop growth parameters such as leaf area over time. Through this application, we demonstrate the possibility of analyzing the growth of an individual plant in the field, which provides important data for the farmers and crop scientists to make informed decisions.

2.1 Our Approach to Long-Term UAV Image Registration

In this chapter, we present a technique for registering images of agricultural fields taken by a UAV over the crop season and present a complete pipeline for computing temporally aligned 3D point clouds of the field. Our approach exploits the inherent geometry of the crop arrangement in the field, which remains mostly static over time. This allows us to register the images even in the presence of strong visual changes. To this end, we propose a scale-invariant, geometric feature descriptor that encodes the local plant arrangement geometry.

To register images over multiple sessions, i.e., different UAV flights over the crop season, into a common reference frame, we perform the registration based on four consecutive steps. First, we compute a point-based geometric representation for the images exploiting the crop arrangement on the field as described in Section 2.1.2. This leads to a detection of points, which remains mostly static over different sessions. Second, we exploit this information to encode the local geometry around each detected point in the image using a scale-invariant descriptor as proposed in Section 2.1.3. We then compute point correspondences between overlapping images in a data association step as explained in Section 2.1.4. Finally, through bundle adjustment followed by a dense matcher, we compute the optimized camera poses and spatially aligned 3D point clouds of different sessions in a common reference frame as detailed in Section 2.1.5. The comparison of the point clouds allows us, on the one hand, to qualitatively check the registration accuracy, and on the other hand, to derive crop growth parameters, which serve as an application example.

2.1.1 Assumptions

Our approach is able to match UAV images of the field taken over multiple data acquisition sessions separated over time and makes the following assumptions regarding the setup:

- the UAV camera is mounted in a near-nadir view, and there is sufficient overlap between consecutive images;
- the field is roughly planar in a local region (i.e., our approach may not work in wine yards);
- the images have a ground sampling distance so that plants span over several pixels, but this ground sampling distance does not need to be known nor to be constant;
- the crops are planted in rows, the row positions and plant spacing, however, is unknown (c.f. Figure 2.1).

In the following sections, we discuss the steps outlined above that form the registration process.

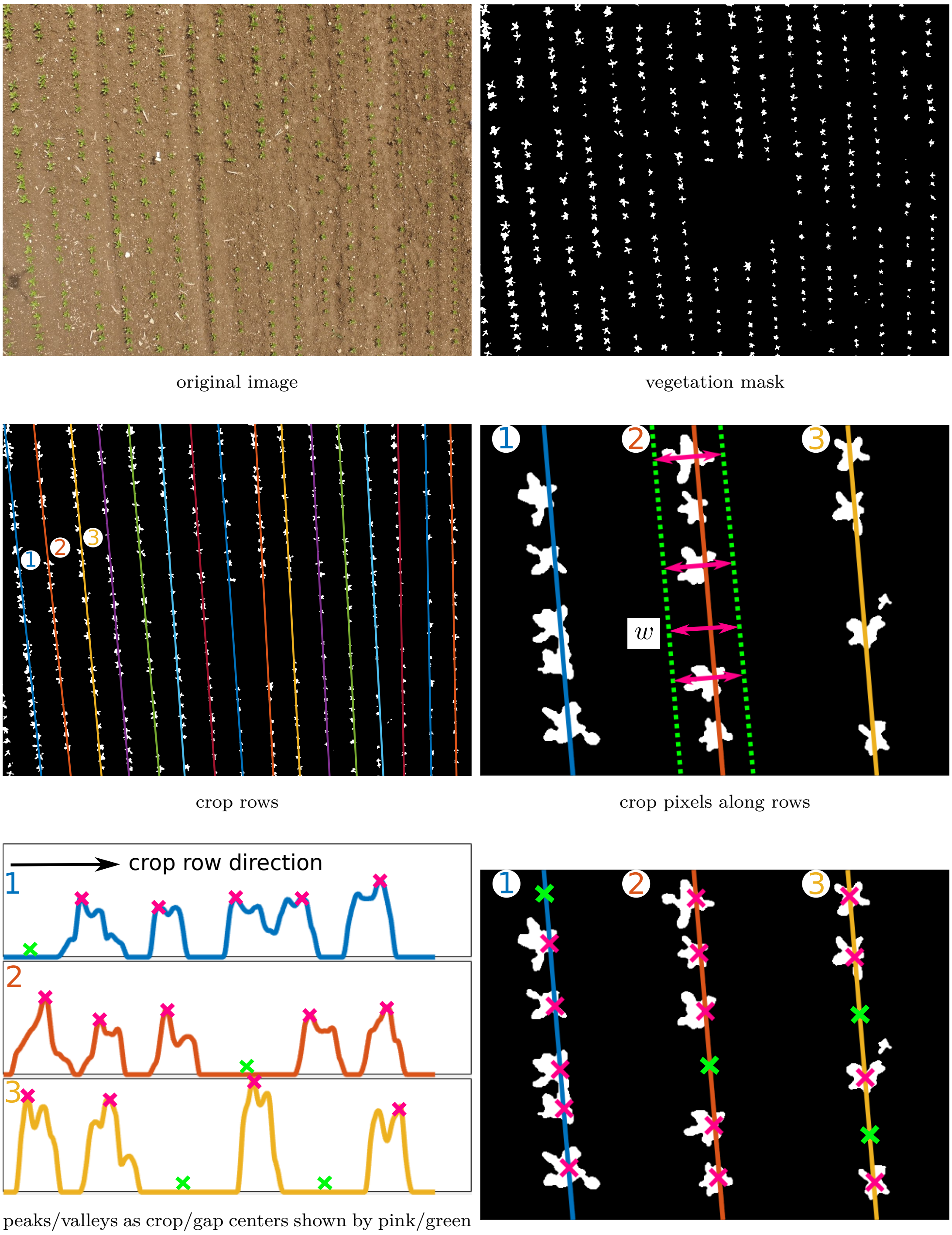
2.1.2 Extract Geometry Information from UAV Images

To capture the structure of the crop field that remains invariant over time, we need to identify the static aspects given the images. Once the crops are planted, they do not move, and the stems/centers of the crops remain rather fixed over time. Therefore, the locations of the crop centers can be used as a fairly static description of the field. The local constellations formed by these points can be seen as a geometric signature of a particular local region covered by an image. Our current implementation assumes that crops are planted in rows as this simplifies the computation of features. The row arrangement is not assumed to be known beforehand, but the existence of crop rows is assumed.

We can compute the crop centers using the following procedure which is also illustrated in Figure 2.3:

1. Compute the vegetation mask exploiting the excess green index (ExG) given by

$$I_{ExG} = 2\,I_G - I_R - I_B \tag{2.1}$$



original image

vegetation mask

crop rows

crop pixels along rows

peaks/valleys as crop/gap centers shown by pink/green crosses

crop/gap center locations shown by pink/green crosses

Figure 2.3: Steps for computing crop and gap centers from the image.

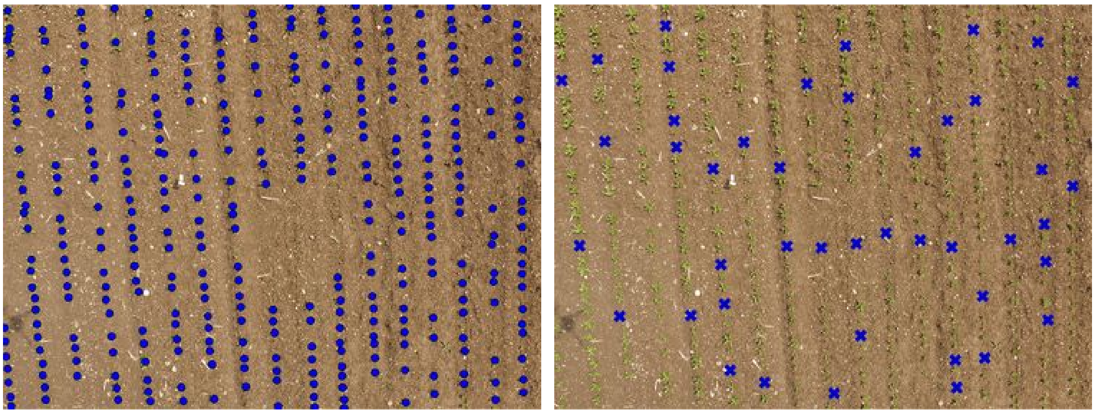


Figure 2.4: Extracted points (left for crops, right for gaps) for the same image.

where I_R, I_G and I_B correspond to intensities of the red, blue, and green channels of the original image. We then apply a threshold θ given by the Otsu's method [86] on I_{ExG} to get a binarized image (Figure 2.3, top right).

2. Fit or compute the lines through the vegetation pixels using the Hough transform for finding crop rows (Figure 2.3, middle left).

3. Compute a histogram of vegetation pixels perpendicular to the direction of the detected rows. The width w is taken to be half the inter crop row distance (Figure 2.3, middle right).

4. Find the peaks of this histogram to identify the potential centers of the crops (Figure 2.3, bottom row).

We observed in multiple experiments that instead of crop centers, the missing crops, i.e., the gaps within the rows, provide an even more distinctive representation than the crop centers itself. This is particularly the case for later growth stages, in which nearby crops often overlap. Therefore, we use the gaps instead of the crop centers as the points representing the geometry in the field based on the images.

To exploit the gaps instead of the crop centers, we follow the same procedure as for the crop centers, but with the difference that the gaps correspond to the valleys in the histogram computed in Step 3, which are marked with green crosses in Figure 2.3 (bottom row, right). Multiple missing crops occurring consecutively are represented by a single gap point at the center of the valley. Further steps of the method are agnostic to the choice of points or how these points are calculated. Figure 2.4 illustrates an example with the extracted crop centers and gaps overlaid on the original image.

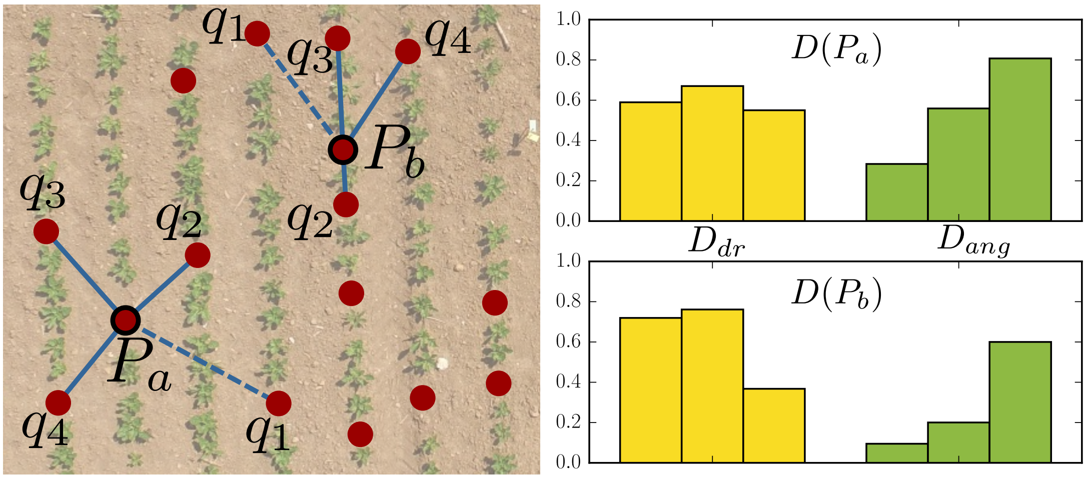


Figure 2.5: Computing a scale-invariant descriptor for a point using local geometry. Left: descriptor computation for two gap points P_a and P_b with k = 4. Right: visualizes the corresponding descriptors.

2.1.3 Scale-Invariant Local Geometry Descriptor

Given the points identified in Section 2.1.2, we aim to encode the local geometry around each point as a descriptor vector to facilitate image matching. We exploit the nadir-view assumption of the UAV and thus can assume that images taken during different flights may only differ in scale, translational offset, and rotation in the image domain. No affine transformation needs to be considered because of the nadir view. To estimate these transformation parameters, we need a descriptor, which is scale-invariant in addition to being invariant to translation and rotation. We construct an own descriptor for each point P using the ratios of distances and relative angles between the k nearest neighboring points of P in the image to meet these criteria. The number of neighbors to consider is a user-defined parameter. The smaller the k, the less expressive/unique is the descriptor of P and the larger k, the more sensitive is the description with respect to outlier points. In our implementation and all experiments, we use $k = 4$, i.e., we consider the four nearest points to P for the computation of the descriptor. In this work, we chose the value of k empirically by evaluating the number of matches obtained for different values of $k \in [3, 8]$ and found that $k = 4$ gave the best results for our datasets. Consider Figure 2.5 for an illustration of how to compute the descriptor D for a given point P. More formally, this is defined by the following computations:

1. Given the k nearest points $Q_k = \{q_1, \ldots, q_k\}$ to P, we compute the so-called reference point R as the point in Q_k with the largest distance to P in the image:

$$R = \underset{q \in Q_k}{\operatorname{argmax}} \|P - q\|. \tag{2.2}$$

Without loss of generality, we assume that q_1 is the reference point R in Q_k and that Q_k is ordered according to the anti-clockwise angle between the line $\overline{Pq_1}$ (dashed line in Figure 2.5) and the lines $\overline{Pq_i}$ with $i = 2, \ldots, k$. Computing all the elements of the descriptor in this order makes the descriptor rotation invariant.

2. The descriptor D will be $2(k-1)$-dimensional and consists of two parts of equal size $D = (D_{dr}, D_{ang})$.

3. The first half D_{dr} of the descriptor vector D consists of distance ratios from P to the individual point, normalized by $\|P - q_1\|$:

$$D_{dr} = \left[\frac{\|P - q_2\|}{\|P - q_1\|}, \frac{\|P - q_3\|}{\|P - q_1\|}, \ldots, \frac{\|P - q_k\|}{\|P - q_1\|} \right]. \tag{2.3}$$

We chose distance ratios in the descriptor because they remain invariant to scale as opposed to individual distances between keypoints in the image.

4. The second half D_{ang} of the descriptor vector D consists of the angles that each point in Q_k has with respect to $\overline{Pq_1}$, normalized by 2π:

$$D_{ang} = \left[\frac{\angle(q_1, P, q_2)}{2\pi}, \frac{\angle(q_1, P, q_3)}{2\pi}, \ldots, \frac{\angle(q_1, P, q_k)}{2\pi} \right]. \tag{2.4}$$

The term $\angle(q_1, P, q_i)$ refers to the angle between the lines $\overline{Pq_1}$ and $\overline{Pq_i}$. An example illustrating the descriptor vector computation for two points P_a and P_b is shown in Figure 2.5.

2.1.4 Data Association amongst Images

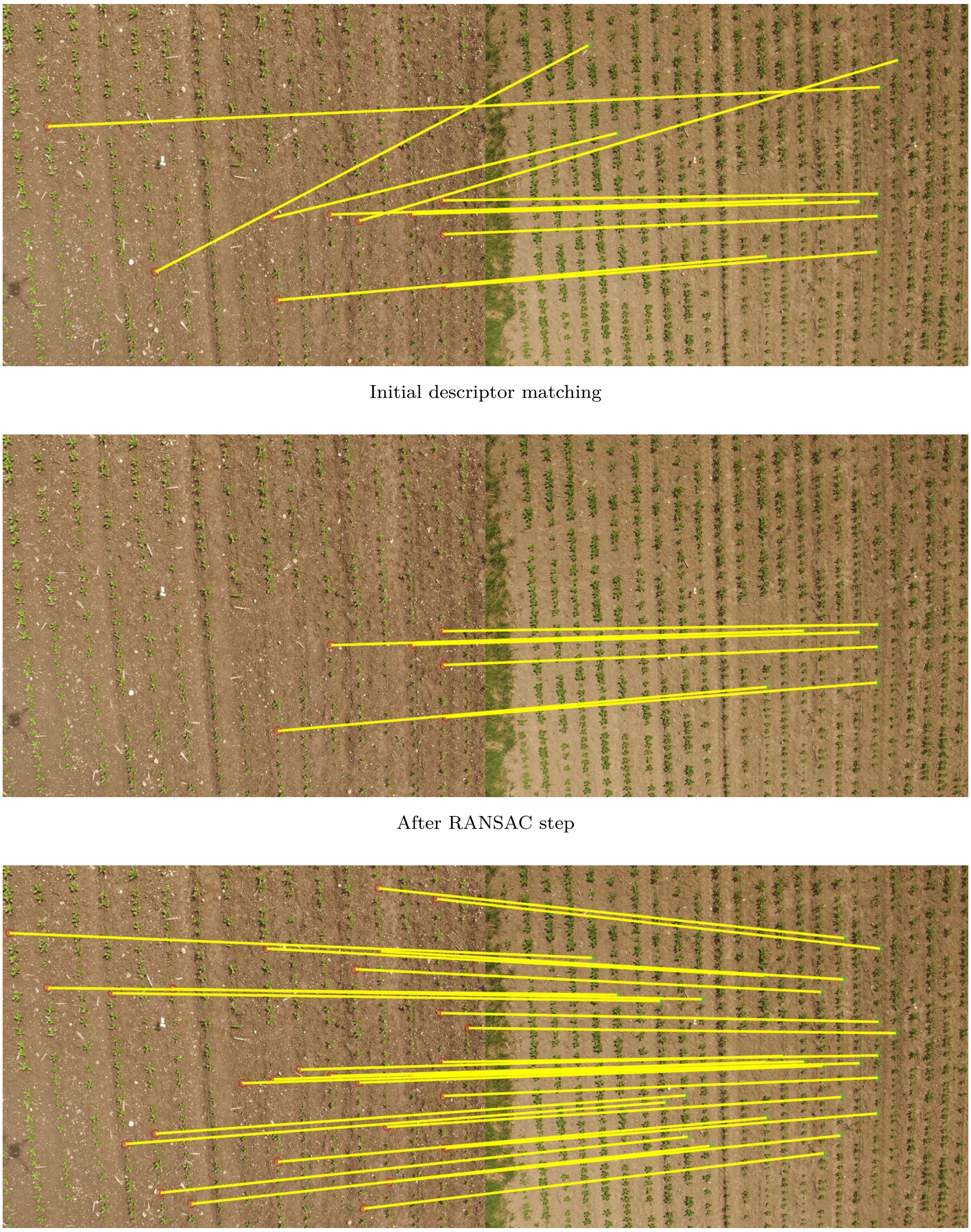
We compute the set of feature descriptors for each image, one descriptor per detected point in the image. Our data association consists of three steps; the first two steps of the data association are rather standard. First, we compute a pair-wise matching of the descriptors of I_1 and I_2 and compare them using the L_2 norm. In the same spirit as done by Lowe [74] for SIFT matching, we reject those matches that have a high distance under the L_2 norm as well as those where the $\frac{L_{\text{best}}}{L_{\text{second}}} > 0.8$, where L_{best} and L_{second} are the scores for best and the second-best match for a descriptor respectively. Second, we compute similarity transformations in a RANSAC loop to identify and remove outliers from the set of corresponding points.

The third step is a correspondence recovery step that deviates from standard data association approaches. Given that the crop arrangement on the field is highly repetitive, i.e., has high visual aliasing, a comparably large number of correspondences get eliminated by Lowe's ratio test. In this step, we consider to re-add those correspondences that were eliminated in the second step in case they are compatible with the transformation found by RANSAC. Thus, the first two steps provide the initial alignment from a potentially small set of correspondences, which is typically free of gross errors. Then, we refine the alignment estimate by re-adding those correspondences, which are consistent with the initial guess. These are locally distinct but potentially ambiguous with respect to the descriptor globally and thus were eliminated before. To ensure high-quality, one-to-one correspondences, we use the Hungarian method [83] for data association in this recovery step. This step allows us to recover more correspondences that were not obtained directly by descriptor matching by using the transformation estimated in the RANSAC step. The Hungarian method computes the theoretically optimal assignment but has a complexity of $\mathcal{O}(n^3)$ and thus is computationally expensive. However, given that the number of possible associations with low distances compatible with the transformation is typically not too large, this does not turn out to be a computational bottleneck in practice, and we can apply this optimal method. Figure 2.6 depicts the correspondence between two images after each step of the matching.

2.1.5 Point Cloud Computation using Bundle Adjustment

As a final step, we perform a pairwise matching between all overlapping images, both spatially and in time across different sessions. Here, we have two options. If we have (a low quality) GPS information available, we can generate a candidate set of overlapping images using the location as prior information. This allows for increasing the speed as only a subset of the images must be tested for correspondences. If no GPS or other position information is available, all image pair combinations are tested.

We compute the possible matches between all the potentially overlapping images and feed them into a bundle adjustment procedure. This algorithm combines the pairwise matches to object points with multiple observations and generates approximate values for the camera poses and 3D object points, which serves as an initial guess for the subsequent optimization. After the adjustment, we obtain a set of optimized camera poses in a common reference frame. For each session separately, we can then compute a dense point cloud using these poses. Any dense matcher can be used here, and we applied the patch-based multi-view stereo reconstruction technique (PMVS) by Furukawa and Ponce [40]. The individual point clouds from each session are already aligned to a common reference frame since the used poses result from a common adjustment in the previous step. The complete pipeline is illustrated in Figure 2.7.



Initial descriptor matching

After RANSAC step

After recovery step using Hungarian assignment

Figure 2.6: Stages of data association procedure. The details regarding each stage is discussed in Section 2.1.4.

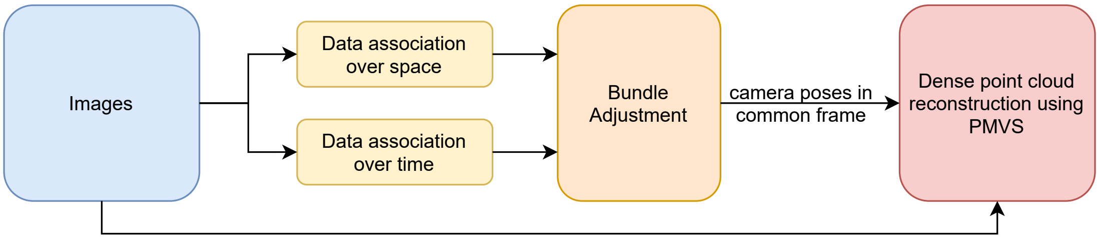

Images
Data association over space
Data association over time
Bundle Adjustment
camera poses in common frame
Dense point cloud reconstruction using PMVS

Figure 2.7: 3D reconstruction pipeline showing steps for computing temporally aligned point clouds from different sessions.

Table 2.1: Overview of the datasets

Field	Session	Date	# of images	crop size	weather
A	1	May 20	45	7 cm	cloudy
	2	May 27	175	10 cm	sunny
	3	June 17	121	15 cm	overcast
	4	June 22	140	20 cm	cloudy
B	1	May 8	99	5 cm	sunny
	2	June 5	95	15 cm	cloudy

2.2 Experimental Evaluation

The experiments in this section are designed to illustrate the capability of our image registration approach for field monitoring tasks in agriculture and to support the claims made in the introduction of the chapter.

2.2.1 Dataset Description

We recorded several datasets [1] of sugar beet crops spanning over multiple weeks for two different fields, referred as A and B here. For the field A, we recorded the datasets across four sessions using a DJI MATRICE 100 UAV. The flight altitude for each session is between 8 m to 12 m above the ground. We recorded the images using the Zenmuse X3 camera with an image resolution of 4000 × 2250 pixels having a ground sampling distance of 4 mm per pixel at a height of 10 m. For the field B, we used a DJI PHANTOM 4 UAV across two sessions recorded almost one month apart. The UAV was equipped with a GoPro camera set up to take an image every second at a resolution of 3840 × 2880. The flight altitude for the two sessions varied between 10 m and 18 m above the ground having a ground sampling distance of 9 mm per pixel at 15 m height. As the GoPro uses a wide

[1] The datasets used in the chapter can be downloaded from here: `http://www.ipb.uni-bonn.de/data/uav-sugarbeets-2015-16/`

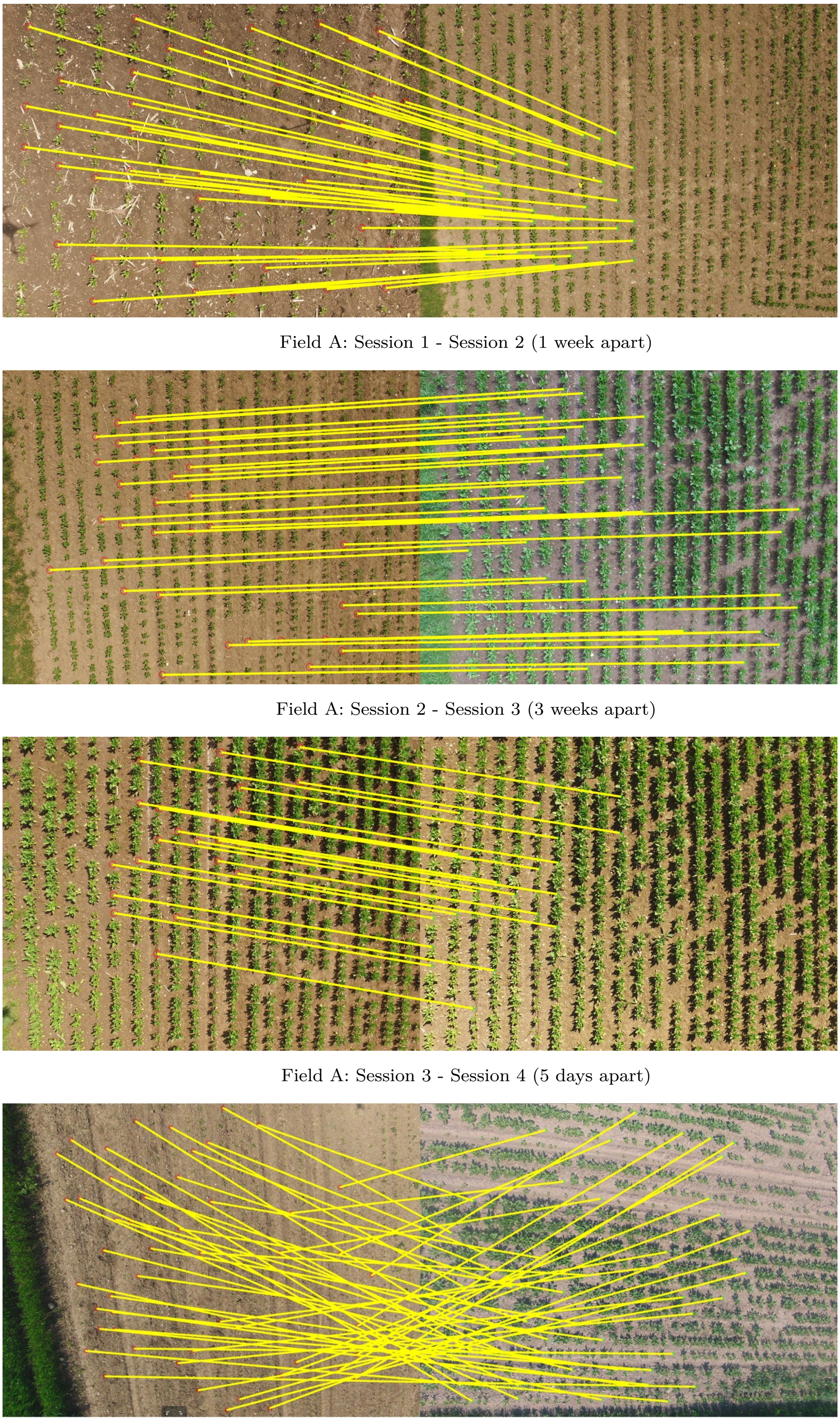
Table 2.2: Matching statistics across the crop season

Field	Session	points per image pair	# of matches			residual error (pixels)
			Lowe test	RANSAC	Recovery	
A	1-2	58	27	11	42	4.21
	2-3	55	20	7	38	4.38
	3-4	57	24	9	40	4.35
B	1-2	74	39	15	56	4.91

angle lens, we first undistort the images before applying the registration pipeline. The average plant sizes in the fields range from 5 cm to 20 cm in diameter across the crop season. Furthermore, the images were taken under different weather and soil conditions. Table 2.1 provides an overview. The most challenging datasets are Session 2-3 (A) and Session 1-2 (B) due to the large time gap of 3-4 weeks between them whereas Session 3-4 (A) is the easiest being only 5 days apart.

2.2.2 Matching Images across the Crop Season

The first experiment is designed to show that our approach is able to match images across the crop season having large difference in visual appearance. We perform matching between overlapping images then across sessions. As described in Section 2.1, we compute the gap points and construct our geometric descriptor for each of the images. We compute descriptors with $k = 4$ neighboring points to encode the local geometry. Table 2.2 summarizes the overall statistics for matching images across the sessions. It lists the average number of common gap points per image pair, the number of correspondences after the Lowe-ratio test, RANSAC, and recovery steps as well as the average residual error. We observe that around 30% of the initially matched points survive the RANSAC step and correspondences for roughly 70% of the points are re-established in the recovery step. Overall for field A, we observe an average residual error of 4.3 pixels, which corresponds to a ground distance of less than 2 cm. We have similar residual errors for field B at 4.9 pixels. While this accuracy does not match up to the usual sub-pixel accuracy of visual matching methods such as SIFT applied in non-changing environments, it is still a very good performance given the fact that physical growth of the plants and their changing appearance limits the accuracy with which the crop centers or the gaps can be detected. Figure 2.8 shows example results from consecutive sessions for both fields. In all examples, visual matching using SIFT fails to find any reasonable set of correspondences.



Field A: Session 1 - Session 2 (1 week apart)

Field A: Session 2 - Session 3 (3 weeks apart)

Field A: Session 3 - Session 4 (5 days apart)

Field B: Session 1 - Session 2 (4 weeks apart)

Figure 2.8: Matching between image pairs from consecutive sessions.

Table 2.3: Evaluation against visual descriptor matching

Field/	SIFT / SIFT-gaps / ORB / BRIEF / Our approach		
Session	% pairs matched	max matches	% inlier
A/1-2	26 / 15 / 10 / 0 / **89**	10 / 8 / 10 / 0 / **42**	19 / 29 / 15 / 0 / **41**
A/2-3	16 / 40 / 5 / 0 / **85**	4 / 12 / 4 / 0 / **38**	10 / 34 / 8 / 0 / **35**
A/3-4	84 / 75 / 80 / 65 / **86**	**103** / 22 / 75 / 70 / 40	**67** / 65 / 65 / 55 / 38
B/1-2	9 / 15 / 0 / 0 / **87**	4 / 9 / 0 / 0 / **56**	7 / 21 / 0 / 0 / **39**

2.2.3 Comparison against SIFT, ORB, and BRIEF

This experiment is designed to compare the matching performance of our approach against visual matching procedures using different descriptors. We perform the comparison between overlapping image pairs between each consecutive session for both the fields. In addition to the standard SIFT matching procedure using the default detector, we also compute the SIFT descriptor at the gap points computed by the detector in our approach. The intuition for doing this is that the gap regions are the least affected regions due to the movement of the tractor etc. on the field. Therefore, it provides the possibility of matching the texture of the soil in these regions across different sessions. Table 2.3 provides a comparison of the matching performance using standard SIFT, SIFT at gap points, ORB, BRIEF, and our approach. The table lists the percentage of image pairs matched successfully, maximum matches found for an image pair, and the inlier percentages for the matches computed by the three approaches. We consider image pairs having at least 4 matches resulting in a correct transformation as a successful match. For both fields A and B, we see that for most challenging datasets, i.e, Session 2-3 (A) and Session 1-2 (B), visual matching using the SIFT descriptor only matches between 9% to 16% of the image pairs successfully. Even for the successfully matched image pairs, the number of matches is very few, and the percentage of inlier matches is only around 10%, indicating that the matches are not reliable. The percentage of successful matches obtained with ORB and BRIEF is even worse. For example, they cannot match any pairs from the dataset Session 1-2 (B). The SIFT descriptor computed at the gap points slightly improves the percentage of successful matches for Session 2-3 (A) while providing no improvement for other cases. However, for the relatively simpler dataset, i.e., Session 3-4 (A), the visual approaches perform well as these images were captured only five days apart and are visually similar. For this dataset, we observe that the SIFT based matching gives the best performance with the highest number of matches and the best inlier percentage. This occurs mainly because the appearance of the field during the two sessions remains static. As a

Table 2.4: Evaluation against ground truth

Field	Session	% of estimated matches	residual error (cm) (est/ref)	registration error (trans/rot/scale)
A	1-2	91.67	1.47/0.98	3.19 px /0.38° /0.31%
	2-3	84.86	1.75/0.77	4.54 px /0.60° /0.42%
	3-4	85.19	1.74/0.86	4.07 px /0.42° /0.32%
B	1-2	87.22	3.93/2.16	3.94 px /0.47° /0.35%

result, the SIFT based matching is able to exploit the texture information to find the correspondences, which was not possible with the other datasets. Overall, our approach consistently matched around 85% of the image pairs with higher inlier percentages for each of the sessions, including the challenging datasets of Session 2-3 (A) and Session 1-2 (B). This is because our approach exploits the geometry rather than relying on the visual appearance of the field.

2.2.4 Ground Truth Accuracy Evaluation

This experiment evaluates the accuracy of our matching results against the ground truth. We compare our results with ground truth parameters for 10 image pairs between each session to perform this analysis. All the evaluation parameters are summarized in Table 2.4. The ground truth parameters are computed based on control points, which have been provided manually. Using these control points, we compute the reference ground truth registration parameters under a similarity transform. We further manually establish unique correspondences between the image pair points under these registration parameters and consider them as the ground truth correspondences. We provide a measure of the quality of matching in terms of the percentage of correspondences estimated by our method compared to the ground truth correspondences. On average, our method is able to recover up to 90% of all possible correspondences. We also compute the residual error based on the estimated correspondences and compare it to the residual error of the manually generated ground truth. For field A, we obtain an average residual error of around 1.6 cm, indicating that the estimated parameters are correct. The residual error for field B is in the same range as that of field A. The absolute value of the error is higher only due to the lower ground resolution of 9 mm per pixel for this flight. Further, we evaluate the accuracy of the registration parameters by computing the average errors (translation, rotation, and scale) with respect to the ground truth parameters. We observe an average translation error of close to 4 pixels. We also obtain an average rotational error of 0.5°, and a scale error of less than 0.5% with respect to the ground truth parameters.

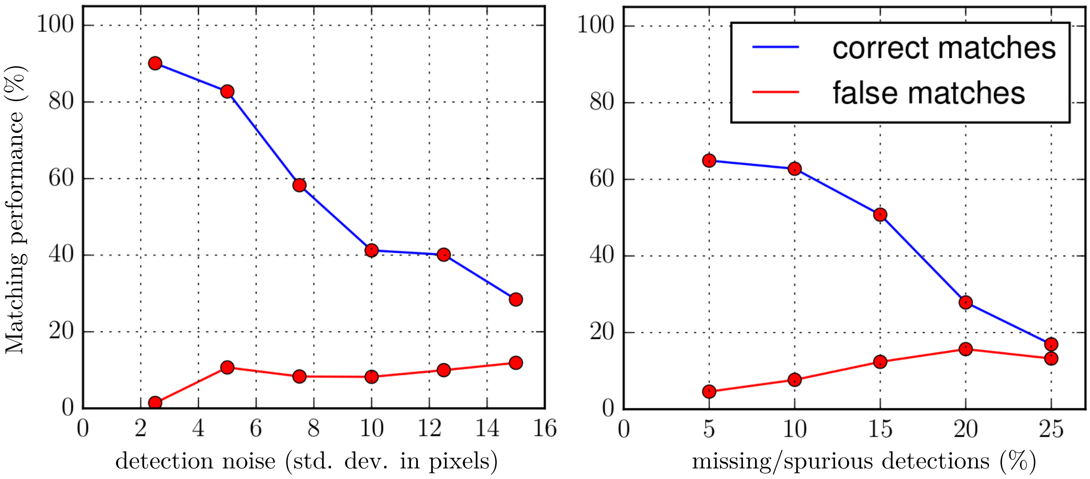


Figure 2.9: Descriptor robustness under varying noise levels assessed in terms of percentage of correct matches (true positives) and false matches (false positives + false negatives).

2.2.5 Descriptor Performance Under Noisy Detections

This experiment shows the robustness of the descriptor under noisy detection conditions. We perform this analysis by simulating two kinds of noise, (i) a Gaussian noise affecting the location of the keypoints, and (ii) missing/spurious detection of keypoints. The missing points refer to the keypoints that exist in the image but have not been detected, whereas the spurious points refer to the keypoints which have been detected but do not correspond to a true keypoint in the image. We perform the analysis considering the gap centers as the keypoints. We assess the performance of descriptors by computing the percentages of correct matches (true positives) and the false matches under varying levels of noise. The false matches include both false positives, i.e., the points that are incorrectly matched, and false negatives, i.e., the matches that were missed. For the noise of type (i), we vary the standard deviation of the Gaussian noise on the location of gap centers up to 15 pixels. The typical noise level for the gap detection procedure for our images is around 5 pixels. In Figure 2.9 (left), we observe that even for high noise levels (15 pixels), about 30% of the correspondences are identified correctly, whereas the false matches are below 20% after performing the Lowe's test. We observe a similar trend under missing/spurious points noise in Figure 2.9 (right). We can identify up to 20% of the matches even when one-fourth of the points are wrongly detected. These correspondences provide sufficient information for our data association procedure to match the images successfully. Furthermore, the RANSAC step eliminates the wrong correspondences resulting from incorrect descriptor matching. We finally recover only the consistent but initially ambiguous correspondences during the recovery step. This further supports the claim that we can perform matching robustly under substantial noise.

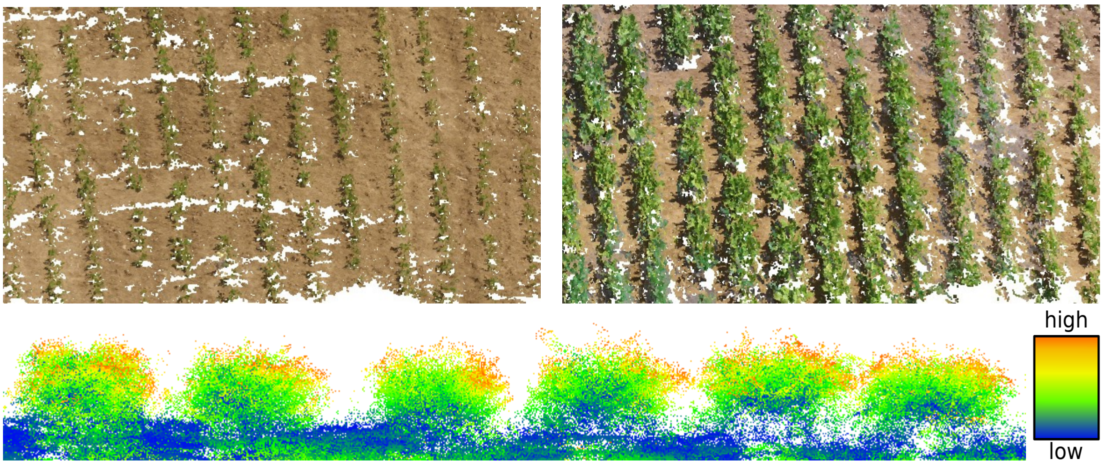


Figure 2.10: Temporally aligned 3D point clouds. Top: 3D reconstruction for a portion of the field from the same viewpoint for Session 2 (left) and Session 3 (right). Bottom: cross section of a part of the point cloud from Session 3. The color of the point cloud represents the difference between the point clouds from the two sessions, i.e. Session 2 and Session 3. The portion close to ground does not change much between the sessions and therefore has a small difference indicated with blue color. In contrast, the top parts of the crops are colored green/yellow/red indicating bigger differences between the point clouds from the two sessions. This is due to the physical growth of the plant between the two sessions.

2.2.6 Time-Aligned 3D Point Clouds

This experiment is designed to show that our reconstruction pipeline allows us to compute temporally aligned 3D point clouds of the field (Section 2.1.5) and thus support our second claim. We first compute matches within the same session and matches between images in consecutive sessions. We then feed them all into a single bundle adjustment to obtain the 3D camera poses of all sessions in a common reference frame. Using the aligned poses, we now can compute dense 3D point clouds for each session separately with PMVS [40]. Due to the common bundle adjustment, the point clouds for different sessions are already registered to each other and can be compared directly. The top two point clouds in Figure 2.10 illustrate the result by rendering a portion of the field from the exact same camera position both for Session 2 and Session 3 respectively. This allows us to monitor the evolution and the changes on the field over time. Note that the point clouds from two sessions are aligned based on the respective camera poses obtained from the bundle adjustment and not a point-to-point correspondence of the 3D point clouds over time. To assess the quality of the alignment, we visualize the difference between the two aligned point clouds. The bottom part of Figure 2.10 shows a cross-section view of the point cloud from Session 3, where the color signifies the difference between the point cloud from Session 2 and Session 3. The difference increases as the color changes from blue to red. We see that the alignment of the point clouds looks qualitatively correct as the space in-between the crop rows

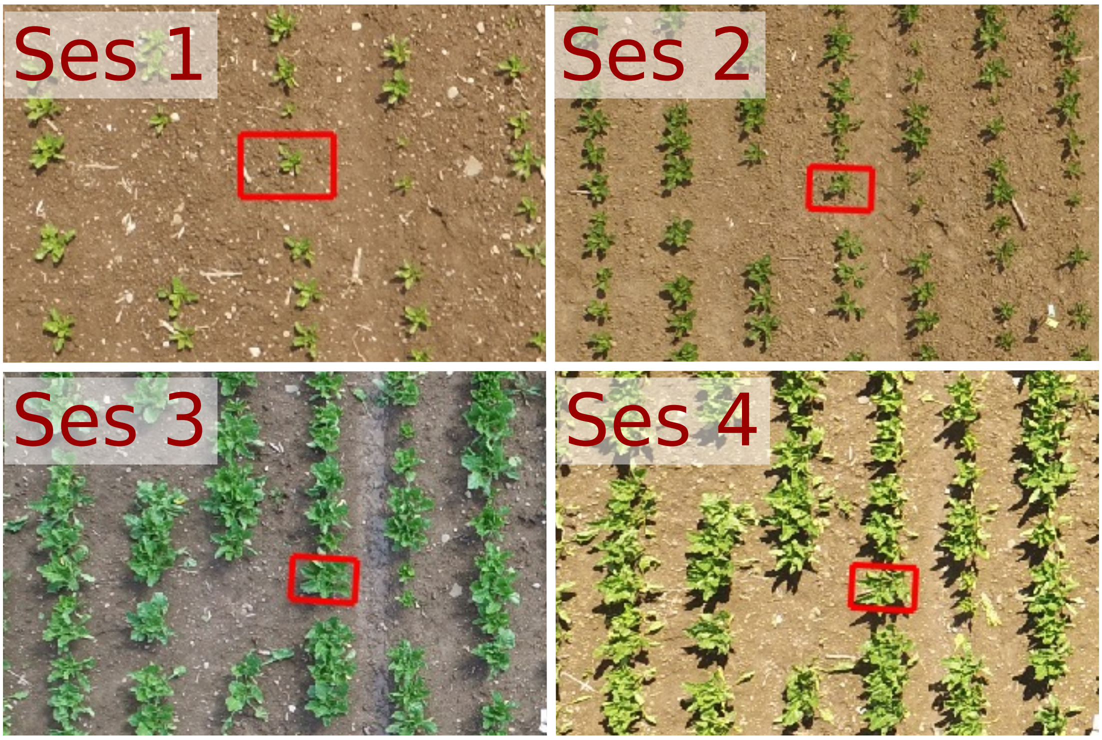


Figure 2.11: Monitoring crop growth parameters. Same crop identified in the bounding box over different sessions using our registration results.

has a small difference indicated by the blue color. We also observe that the lower portions of the crops have a smaller difference as this portion overlaps with the crops from Session 2, whereas portions at the top have a larger difference reflecting the crop growth between the two sessions.

2.2.7 Monitoring Crop Growth Parameters

To support our final claim, we show in the following experiment that our registration results allow us to monitor growth parameters at a per plant level. We manually provide bounding boxes around crops in the first session and compute the locations of the new bounding boxes in the corresponding images from different sessions using our registration results. Figure 2.11 shows an example where the same plant is identified through different sessions. To monitor the plant's growth, we compute the total leaf area (from the top view) for the plant in each of the sessions. We compute this area by first extracting a vegetation mask inside the bounding box using the excess green index (ExG) and compute the area under it. Figure 2.11 shows the plot of the total leaf area for individual plants at five different sites on the field over all the sessions. Here each site refers to a distinct 30 cm x 30 cm region on the field. As expected, we see a general trend of increasing leaf area with time. For the plant shown in our example , the leaf area increases from about 150 cm^2 in Session 1 to 430 cm^2 in Session 4. The growth

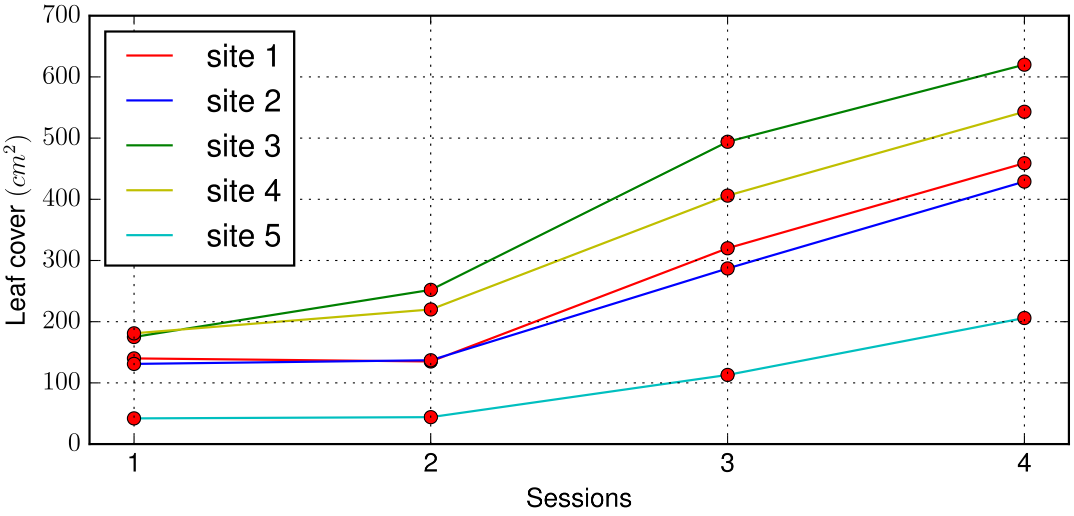


Figure 2.12: Monitoring crop growth parameters. Plot of leaf cover over time at five different sites in the field. Each site refers to a distinct 30 cm x 30 cm region on the field.

on the sites is consistent with the BCCH growth scale index for sugar beets. This experiment illustrates that our registration results are accurate enough for monitoring growth parameters at a per plant level. However, it should be noted our main goal here is not to analyze crop growth but to facilitate such analysis by registering images taken over time to a common coordinate frame. Other works such as [69], [92] address the issue of analyzing crop growth in more detail. Later in Chapter 4, we investigate techniques for registering high-fidelity point cloud data, which allows us to monitor plant growth at a much higher resolution.

2.3 Related Work

With the wide availability of commercial UAVs, it has become fairly easy to acquire image data repeatedly without any expert assistance. This has also resulted in the gathering of large amounts of field data and led to the development of several new applications in the agricultural robotics community. Das *et al.* [29] and Bryson *et al.* [16] propose various sensor setups and software components for some typical field monitoring tasks using ground and aerial vehicles. Some other techniques have been developed with the goal of performing intervention tasks in the field. For example, Lottes *et al.* [73, 71] focuses on distinguishing crops and weeds for targeted weeding application while Kusumam *et al.* [62] detect and localize broccoli heads for selective harvesting. Other works such as [69, 92] have investigated towards analyzing plant growth from multi-spectral images and point clouds. However, most of these works focus on the interpretation of data at a given point of time. They do not consider the temporal changes in the field that present challenging situations in terms of data association. In this chapter, we have aimed at bridging this gap and presented a method for temporal registration of field images.

In related literature from long-term localization and place recognition applications, several methods address the problem of finding data associations amongst images having large differences in visual appearance. Visual localization and place recognition for long-term applications require robust image matching in the presence of strong illumination and seasonal changes [82, 125]. A comprehensive survey of the visual place recognition techniques can be found in [75]. Most of these techniques are designed for autonomous driving applications and do not lend themselves to be used for finding matches in field images having a large baseline. Griffith *et al.* [45] go further by incorporating 3D structure information in the scene to address the problem of aligning images of natural scenes across seasons. Martin-Brualla *et al.* [80] have created impressive time-lapse videos of dynamic scenes such as construction sites or glacier movements over the years. These methods utilize a large number of photos from the Internet to warp them into a single viewpoint. However, they still require temporally close images to have a substantial portion of the scene to be similar for the data association to be performed using standard visual descriptors such as SIFT [74].

Some other works [121, 88] are directed towards detecting the changes in the scene and updating the model accordingly. Ulusoy *et al.* [121] propose an approach that updates an initial model of the environment by analyzing the geometric discrepancy between current measurements and the previously built model. Similarly, Palazzolo *et al.* [88] develop a method to quickly find the rough location of structural changes between the current state of the world and a reference 3D model using only a few images of the scene. Sakurada *et al.* [103] and Taneja *et al.* [118] address the issue of detecting the changes in urban environments using images from a camera mounted on a vehicle. Qin *et al.* [97] highlight the challenges of detecting changes in measurements acquired at different resolutions and provides an overview of techniques aimed at addressing these challenges. However, these change detection techniques are restricted to a local region of the scene and do not apply to our situation where the whole field is continuously changing over time. Instead of computing local changes, we match the whole images that are acquired at different times, and the changes in terms of the growth of the individual plants are then computed based on the registration results.

Descriptors that exploit geometrical patterns have been used in various applications, and a large corpus of literature exists for matching point patterns in images and other synthetic data. Gold *et al.* [44] and Hancock *et al.* [17] propose different formulations for estimating correspondences from noisy point sets enabling them to deal with deformable objects in the image. Wolfson [130] proposes a hashing based method using invariant properties of transformations to retrieve the correct object from a large database of objects. Our use of geometric descriptor is similar to Moreau *et al.* [133] where they use affine invariant properties to construct a descriptor for tracking planar objects. While these works are not directly applicable to our scenario, we borrowed ideas from them and designed a new descriptor as well as a robust matching procedure suitable for matching point patterns detected in nadir view UAV images.

A closely related work has been proposed by Dong *et al.* [34] that address the problem of matching images from a field across time for the purposes of crop monitoring. They use a SLAM system to fuse the measurements from different sensors such as camera, GPS, IMU, etc., to obtain a high-quality estimate of the camera poses and the field structure. This information is used to reject outliers during the data association step and improve overall robustness. As the matching still relies on visual information, it is still bound to fail when visual appearance changes dramatically, such as in situations like rain as well as any large change in the appearance of the field. In contrast, our method can deal with such situations since it uses geometrical information that remains mostly static even if the appearance of the field changes dramatically.

2.4 Conclusion

In this chapter, we investigated the challenging temporal data association problem that arises while matching images of crop fields taken over the course of a crop season. Our main insight was to exploit the local geometrical patterns that remain relatively static despite the large change in the appearance of the plants and the field itself. Building upon this idea, we presented a novel approach capable of registering UAV images of agricultural fields despite the large variation in the visual appearance over the crop season. We proposed a descriptor that captures the inherent geometry of the crop arrangement in the field by exploiting the negative information about missing crops, i.e., gaps in the crop rows, and use these descriptors for matching images from different times. This approach allowed us to successfully register images even when matching based on common visual descriptors such as SIFT, ORB, or BRIEF fail. Finally, in the experiments, we demonstrated our approach provides a robust and efficient method for registering crop field images. The registration results, in turn, allowed us to compare individual plants in temporally separated images and capture their growth. We also showed that these registration results could also be exploited by a bundle adjustment procedure to obtain temporally aligned 3D point cloud and monitor changes in plant properties such as the canopy height. We believe that this work would form an important step for UAV-supported monitoring applications such as in-field phenotyping, continuous yield forecasting, which require temporally aligned models of whole fields up to an individual plant level.

Chapter 3

Collaborative Localization in Fields Exploiting UAV Imagery

The ability to localize is a key capability for robots to navigate autonomously in the environment. To localize means to means to answer the question “where is the robot?”. Localization is often done with respect to a map of the environment that is built beforehand. An accurate localization system is critical for robots as it provides the knowledge of their current location, which is necessary for planning and navigation in the environment. This is also the case in agricultural fields, where the UGV needs to localize itself accurately to navigate through the crop rows, perform monitoring and precise intervention actions in the field.

Although several localization approaches exist, the crop field environment poses several unique challenges, which are difficult to cope with. For example, the repetitive structure of the crops in the field gives rise to aliasing. This easily results in multi-modal distributions about the robot’s pose that is difficult to resolve. In addition to this, the appearance of the vegetation in the field changes continuously over time, even on the same day. This makes it challenging to localize over multiple sessions, which is a requirement in most agriculture applications requiring monitoring and precise intervention. Collaboration between UAVs and UGVs has the potential to provide flexible solutions to accomplish different precision agricultural tasks effectively. In this chapter, we explore how can we improve the localization capabilities of a UGV by collaborating with a UAV.

Presently, the solution for localization in field environments is through the use of high-precision real-time kinematic (RTK) GPS. Although these sensors can provide the desired accuracy most of the time, they are rather expensive and are still vulnerable to signal outages resulting in degraded estimates. Several other localization approaches based on visual features such as SIFT [74] or similar features often fail due to the large difference in the appearance of the field over the crop season, as illustrated in the previous chapter.

In this chapter, we present a solution to the localization problem for ground robots operating in crop fields over long periods in a collaborative manner using images captured from a UAV. This localization system can also be used independently to provide redundancy to other localization systems such as those based on GPS. Our system only requires the ground robot to be equipped with a monocular camera, an odometer and uses an aerial map of the field as a map. Such a simple image-map can be obtained easily from unmanned aerial vehicles (UAVs) flying over the field. The key idea of our approach is to take advantage of the salient features in the field that are easily identifiable from different viewpoints and remain invariant over the crop season, even when the visual appearance changes dramatically. To meet these criteria, we propose to use the locations of the plants and the gaps in the field as features capturing the inherent geometry of the field and exploit the plant semantics to further tackle visual ambiguities. The idea of using crop and gap locations as features is inspired from the previous chapter, and we modify them to perform data association between different platforms, i.e., a UAV and a UGV, with varied viewpoints. Furthermore, we also capture the changes in the field by explicitly modeling the existence of plants as a probabilistic belief and using this information to curate the map after each session.

In the previous chapter, we developed a technique for registering UAV images of crop fields captured at different time instances by exploiting local geometric patterns in the fields. Similarly, in this chapter, we exploit the semantics as well as the locations of crops, weeds, and gaps for localizing the ground robot. Here, we use these feature detections within an estimation framework to obtain an online estimation for the ground robot pose. Different from the previous chapter, the features are observed from different viewpoints from the two platforms and thus violate some of the previously made assumptions. The UAV observes the field in a near-nadir view with a large field of view. In contrast, the ground robot observes a small region of the field from a close distance, often with a tilted camera resulting in perspective distortions. As a result, we need to employ more sophisticated detection algorithms to extract the features and estimation frameworks capable of dealing with more ambiguous data associations.

The main contribution of this chapter is a novel localization system for robots operating in crop fields over an extended period of time. Initially, we construct a reference map as a set of sparse features encoding the geometry and semantics of the field using the images taken from a UAV. For localization, we use the feature detections from the ground robot within a Monte-Carlo localization algorithm to estimate the robot's pose using an observation model targeted to the crop field domain. In addition to that, we update the map of the field at the end of each session based on the belief of the existence of each feature. The map update allows us to reduce the potential for wrong feature associations and thus

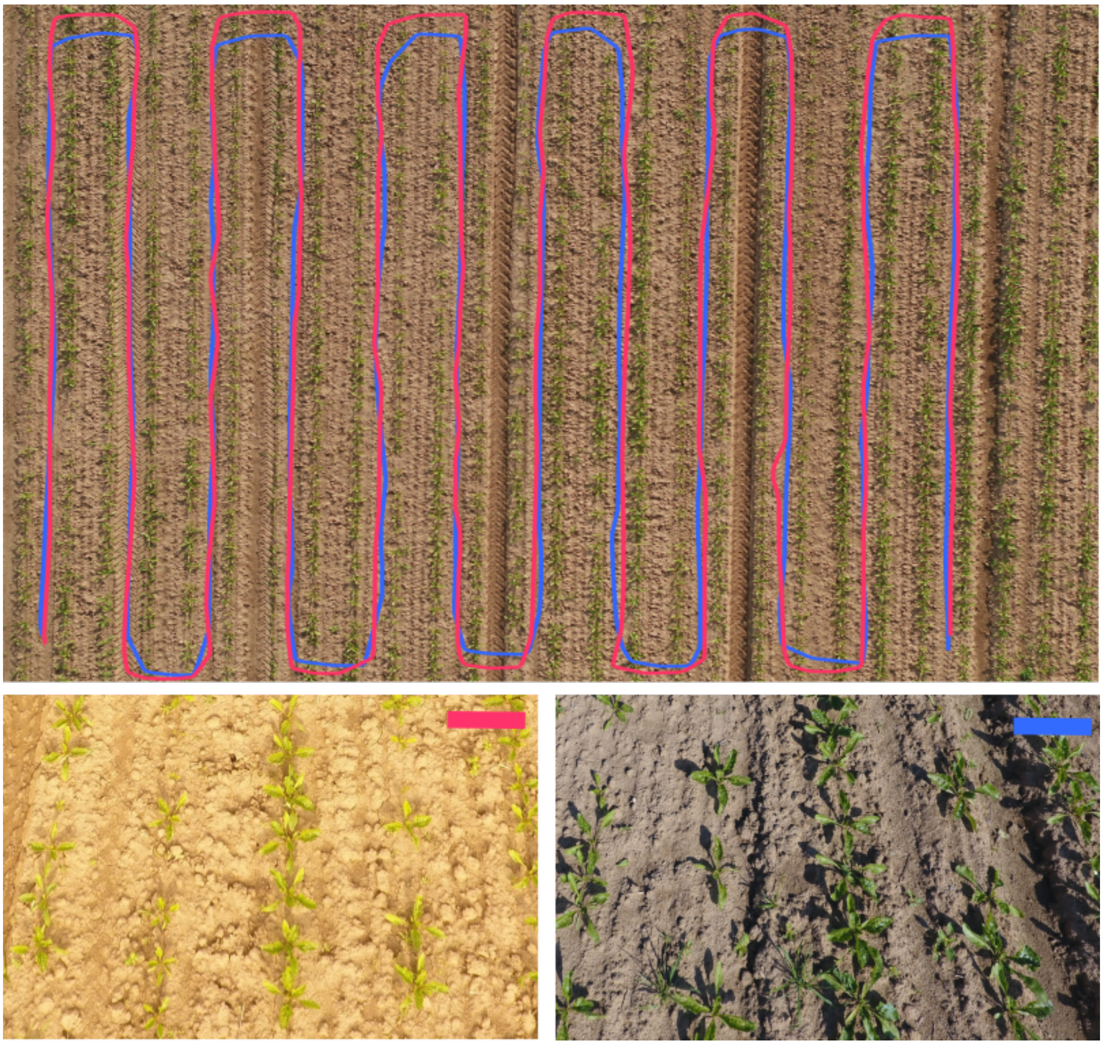


Figure 3.1: Top: Robot trajectory estimated by our approach recorded at two different points in time (sessions) visualized on top of the aerial map used as a reference map. Bottom: Images from the ground robot recorded at same location but at the different times of data acquisition.

improve the performance of the feature-based localization over time in changing environments such as agricultural fields.

Our approach proposed in this chapter enables a UGV to (i) localize with sufficient accuracy allowing the robot to navigate within the gap between adjacent crop rows, (ii) provide better performance than standard GPS approaches and is more robust to environmental changes over the season as compared to methods relying on purely visual features, (iii) perform localization over multiple sessions without needing to remap the field each time, and (iv) maintain an updated map of the environment by integrating the current measurements allowing the system to function smoothly over long time periods. This kind of localization capability for a UGV is a pre-requisite to perform precision agriculture tasks and operate reliably in challenging field environments over longer periods of time.

3.1 Ground Robot Localization using Aerial Images of Crop Fields

Here, we present our localization system for a ground robot operating in the crop fields by using aerial maps that have been acquired from UAV survey flights. We achieve an accurate localization estimate for the ground robot through several steps. First, we compute an aerial orthomosaic of the field using the images captured by the UAV during a survey flight. We then compute distinctive features such as stem locations of the crops and weeds on the aerial orthomosaic by employing an end-to-end fully convolutional network (FCN) as explained in Section 3.2. When deploying the ground robot in the field, we compute the features on the live frames from the camera stream, also using an FCN and match them against the aerial map features to estimate the robot's pose using a particle filter. This procedure is described in Section 3.3. Finally, we update the aerial reference map periodically as the field conditions are constantly changing due to plant growth as well as other management activities such as weeding operations. The aerial map is updated each time the ground robot enters the field using its current measurements as described in Section 3.4. By keeping the map updated, the ground robot is able to localize in the fields for longer periods of time.

3.2 Features for Localization in Crop Fields

In order to localize the robot using aerial images of fields, we need to find data associations between the UAV and the ground robot images. As these images are taken from two very different viewpoints, we need to extract features that are visible in both images. This section describes how to compute these features and use them to estimate the robot's pose.

3.2.1 Generating Aerial Landmark Map

To generate the aerial landmark map, we capture images of the field from a UAV with a downward-looking camera over the whole field. We align these UAV images with a standard bundle adjustment procedure [120] and estimate the camera poses and the digital elevation model of the field. Using these estimates, we stitch individual images to generate the aerial orthomosaic. This is done using the commercially available software Agisoft PhotoScan.

From this orthomosaic, we compute a landmark-based representation, which consists of stem locations of the plants. The main idea behind using the plant stem locations as landmarks is that they provide a representation of the field that is comparably static. In addition to the stem location, we further use the camera images to classify each plant as a crop or a weed and use this information to avoid inter-class associations of the features during localization. Thereby, our approach takes advantage of the natural semantics of the field.

Both, the stem locations of individual plants and the semantic label, are estimated using an end-to-end trainable fully convolutional network (FCN), which is described by Lottes *et al.* [70]. In addition to that, we compute the gap locations between crops in the field, i.e., positions of missing crops on the field surface. We are able to do this because we expect the crops to grow in a row. The gap locations provide a more distinctive pattern, which allows us to tackle the problem of visual aliasing in row crops as described in the previous chapter.

From these features, we construct a map $\mathcal{M}$ of the environment as collection of landmark tuples

$$\mathcal{M} \;=\; \left\{(l^{(1)}, s^{(1)}), \ldots, (l^{(L)}, s^{(L)})\right\}, \tag{3.1}$$

where $l = (l_x, l_y)^\top$ is the location of the landmark in the global coordinate frame derived from the aerial reference map and $s \in \{\mathrm{crop}, \mathrm{weed}, \mathrm{gap}\}$ denotes its corresponding semantic label. Figure 3.2 (left) show an example with landmarks computed for an UAV image with crop (green), weed (red), and gap (blue) features.

3.2.2 Extracting Features from Ground Robot Images

For the ground robot, we extract features for every incoming image to find a data association between the current observation and the aerial landmark map. We extract the same features as in the aerial image, i.e., the locations of crop and weed stems as well as gaps between the crops. However, extracting precise locations of plants from the ground robot images is more challenging as the camera

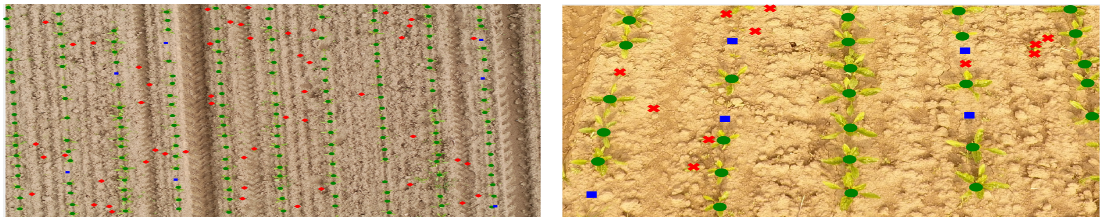


Figure 3.2: Feature detections from an UAV (left) and ground robot image (right). We visualize crop (green), weed (red) and gap (blue) features detected from both views.

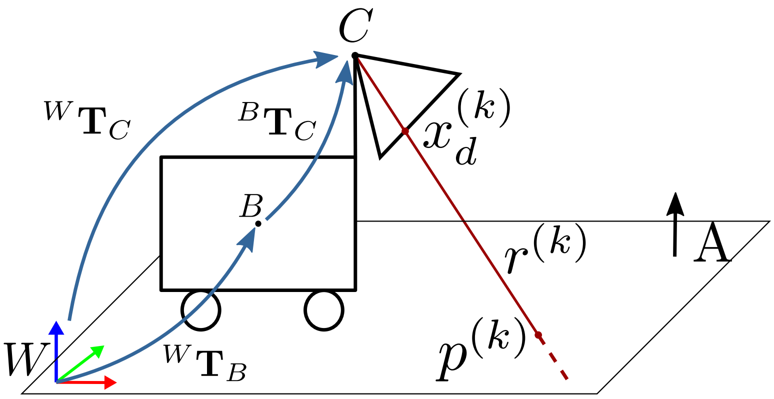


Figure 3.3: Projecting a feature $x_d^{(k)}$ from the image plane to the coordinate frame of the map $\mathcal{M}$.

orientation is tilted with respect to the nadir view. This camera setup results in images with varying ground sampling distance, where the farther parts of the field have a lower resolution. Furthermore, during the later stages, when the crops are bigger, this view suffers from occlusions induced by large crops in the front of the scene. To deal with these challenges, we deploy a re-trained version of the same FCN [70], which we had used for the aerial data. To allow the FCN to detect stems from a perspective view, we fine-tune it by additional training on a small portion of labeled image data captured from the ground robot. At inference time, the FCN yields in a set of feature detections for an image

$$\mathcal{F} \;=\; \left\{(x_d^{(1)}, s^{(1)}), \ldots, (x_d^{(K)}, s^{(K)})\right\}, \tag{3.2}$$

where x_d is the pixel location of a feature in the image coordinate frame and s denotes the semantic label of the detection. Figure 3.2 (right) shows the feature detections for a ground robot image extracted using this procedure.

3.2.3 Camera Projection

To match the features detected in the ground robot image, we project each detection $x_d^{(k)} \in \mathcal{F}$ onto the aerial map $\mathcal{M}$. For making this projection, we need to know the pose of the camera ${}^{W}\mathbf{T}_C$ and the parameters of the ground plane A in the world frame (illustrated in Figure 3.5). We assume to have a calibrated camera and that the relative transformation from the robot base to the camera ${}^{C}\mathbf{T}_{B}$ is known. The pose of the robot base ${}^{W}\mathbf{T}_{B}$ is estimated by the localization algorithm explained in the next section.

To obtain the projected point $p^{(k)}$ in the map $\mathcal{M}$ corresponding to $x_d^{(k)}$, we first compute the direction of the ray $r^{(k)}$ in world coordinates using the camera calibration matrix $\mathbf{K}$ and the rotation matrix $\mathbf{R}$ from the ${}^{W}\mathbf{T}_{C}$ as

$$r^{(k)} \;=\; \mathbf{R}^{\mathsf{T}}\mathbf{K}^{-1}x_d^{(k)}. \tag{3.3}$$

Then, we compute the location $p^{(k)}$ of the feature observation on the plane A as the intersection of the ray $r^{(k)}$ and the plane A. This can be obtained efficiently by expressing $r^{(k)}$ in Plücker coordinates. We express $r^{(k)}$ as a line $L^{(k)}$ joining the camera projection center C and a point $q = C + r^{(k)}$ along the ray as

$$L^{(k)} = \begin{bmatrix} L_h \\ L_0 \end{bmatrix} = \begin{bmatrix} C - q^{(k)} \\ C \times q^{(k)} \end{bmatrix}. \tag{3.4}$$

From $L^{(k)}$, we compute the transposed Plücker matrix

$$\Gamma^{\mathsf{T}}(L^{(k)}) \;=\; \begin{bmatrix} \mathsf{S}(L_0) & L_h \\ -L_h^{\mathsf{T}} & 0 \end{bmatrix}, \tag{3.5}$$

where $\mathsf{S}(L_0)$ is the skew symmetric matrix computed through the vector L_0. Finally, we obtain $p^{(k)}$ as

$$p^{(k)} \;=\; \Gamma^{\mathsf{T}}(L^{(k)})A. \tag{3.6}$$

Due to the limited field of view of the camera, the number of features detected in a single image frame is often small (around 30). Typically, such data is not distinct enough to cope with the visual aliasing in the environment. Therefore, we aggregate features from consecutive images into a small sub-map until it covers an area of $15\,\mathrm{m}^2$. The accumulated data represents an observation for the particle filter described in the subsequent section.

This allows the accumulated observations to have sufficient features in order to be able to match against the aerial map effectively. These accumulated observations, i.e., the set of all points $p^{(k)}$ and their semantics $s^{(k)}$ in the sub-map, form the measurement $\mathcal{Z}$ for our system

$$\mathcal{Z} \;=\; \{(p^{(1)}, s^{(1)}), \ldots, (p^{(N)}, s^{(N)})\}, \tag{3.7}$$

where p is location of feature projected in the global co-ordinate frame and s denotes corresponding the semantic label.

3.3 Global Localization in Crop Fields

Due to the repetitive structure of the environment, data association between the ground robot observations $\mathcal{Z}$ and the aerial landmark map $\mathcal{M}$ is potentially ambiguous. Additionally, detecting features from the ground robot images is noisy and can result in false detections that have no correct associations in the map. This makes the application of EKF-based systems challenging. Therefore, we use the Monte-Carlo localization or MCL [30] to estimate the pose of the ground robot as it provides a natural way to better deal with multiple hypotheses.

MCL estimates a belief over the robot pose using a set of weighted particles where each particle represents a possible pose of the robot. For our implementation, we consider pose as the position and its orientation of the robot on the field surface. The MCL filter performs three main steps to maintain the belief over the pose. It first propagates the particles based on the odometry estimated by the wheel encoder measurements from the robot. We use the odometry motion model based on the wheel encoder readings as described in [119] to implement this step. Whenever a new measurement $\mathcal{Z}$ is available, it updates the weight of each particle based on an observation model. This model provides a measure of how well the observation agrees with the map given the current pose. Finally, a new set of particles is re-sampled from the old ones, where the chance of survival for each particle is proportional to its weight in the old particle set.

We design an observation model that takes into account the semantics of the features in addition to their locations. By considering the semantics, we are able to reduce the number of wrong data associations, which helps us deal with the aliasing in the field. We define an error $\xi^s(z_i)$ for each point of the semantic type s. The error $\xi^s(z_i)$ is computed as the distance to the nearest landmark l of the same semantic type s in the map $\mathcal{M}$. This means that we only associate crop features in the observations to crops in the map. Similarly, weeds and gaps in the observations are associated against their counterparts in the map.

We can compute $\xi^s(z_i)$ efficiently using a distance transform map $\mathcal{D}^s$ [81], which is pre-computed separately for each feature type, i.e., crop, weed, and gap. The distance transform map $\mathcal{D}^s$ is essentially a look-up table providing the distance to the nearest landmark for each location in the map. Therefore, the error for the measurement z_i is obtained simply by looking-up the value in $\mathcal{D}^s$ at $p^{(i)}$, which is the projection of the feature on the map $\mathcal{M}$. We then compute the average error from all points belonging to a semantic type $Q_s = \{(p, \zeta) \in \mathcal{Z} \mid \zeta = s\}$:

$$\xi^s_{avg}(z, m) \quad = \quad \frac{1}{|Q_s|} \sum_{Q_s} \mathcal{D}^s(p^{(i)}), \tag{3.8}$$

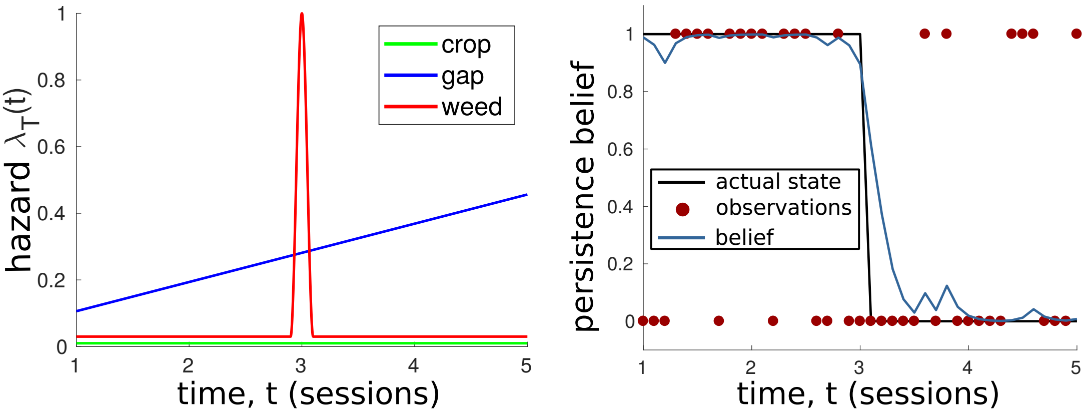


Figure 3.4: Left: Hazard function $\lambda_T(t)$ for crop, weed, and gap features specifying the prior information regarding their survival. Right: existence belief for a gap feature computed by the persistent filter.

and update the particle weight under our observation model as

$$w \propto \prod_s \exp\left(-\frac{\xi^s_{avg}}{2\,\sigma_d}\right)^2, \tag{3.9}$$

where σ_d is the expected measurement noise in the feature detection. The average error ξ^s_{avg} is truncated to a maximum value for robustness against outliers, which is equivalent to using a truncated Gaussian. This turns our observation model into a likelihood field, which can be evaluated efficiently and is used the correction step of the particle filter.

3.4 Map Update

Typically, the map $\mathcal{M}$ used for localization is constructed only once at the beginning of the crop season. However, over time as the field appearance changes, new plants will appear and some of the existing ones may no longer be present. For example, new weeds appear in the field, while some of the gaps are no longer present when the crops increase in size. Therefore, to account for these changes in the field, we integrate the ground robot observations into our map $\mathcal{M}$ and curate it over time. If our observations were perfect, we would only need to remove a feature from the map if that feature's location is re-observed and the feature is not detected, and equivalently for the situation of adding a new feature to the map. In reality, however, the detector outputs are noisy which compounded by the error in estimated pose of the robot itself. This makes it difficult to determine unambiguously if a feature is present in the map or not. Therefore, the best we can achieve is to estimate a belief over the presence of the feature given the observations.

To compute this belief, we realize the so-called persistence filter described by Rosen *et al.* [99]. In addition to integrating the observations $\mathcal{Z}$, the persistence filter also provides an elegant way to incorporate prior information about the feature in terms of its expected survival time in the environment. One of the ways to integrate this survival prior is through a hazard function $\lambda_T(t)$, which encodes the information of how the feature disappearance rate varies over time. A hazard function $\lambda_T(t)$ allows us to describe the various changes occurring in the field in an intuitive manner. In our implementation, we design three different hazard functions to model the survival priors for crops, weeds and gaps. These three hazard function are visualized in the left image of Figure 3.4 (left). For example, the hazard function $\lambda_T(t)$ for crops (green) is very small and constant through out as we expect the crops to survive till the end of the season. Instead, $\lambda_T(t)$ for gaps (blue) increases over time as some of the gaps get covered by the nearby crops and are not detectable anymore. Finally, for weeds (red) we see a sharp rise at $t = 3$, this is to account for a weeding treatment performed at $t = 3$ on the field after which we expect the majority of weeds to be removed. Similarly, this hazard function can be adapted to reflect the different weed management activities performed in the field.

Once the survival priors are defined via the hazard functions $\lambda_T(t)$ for all the features, the filter fuses the observation at time t to update the feature existence belief. Thus, every feature maintains an existence probability that can be used to add/remove features. As an example, we show the belief computed by the filter for a weed feature in the field (Figure 3.4, right). We observe that despite the false detections, the filter maintains a belief close to the ground truth by exploiting the prior information and using successive observations. At the end of each session, we update the map $\mathcal{M}$ by removing the features whose existence belief is less than a fixed threshold. We also add the newly discovered features from the current session and initialize them with an existence probability of one.

3.5 Experimental Evaluation

In the experimental section, we evaluate the performance of our localization system for the ground robot exploiting the aerial maps as described in the chapter. The experiments are designed to evaluate the localization accuracy of our system, and its applicability for navigation in crop-row fields. We compare the accuracy of our localization estimates against that of a standard single-phase GPS as well as localization using state-of-the-art visual feature detectors. We also compare the estimated trajectory for the robot against high-fidelity ground truth trajectory obtained using an external tracking system. Finally, we show the impact of updating the map over time using live measurements from the ground robot on

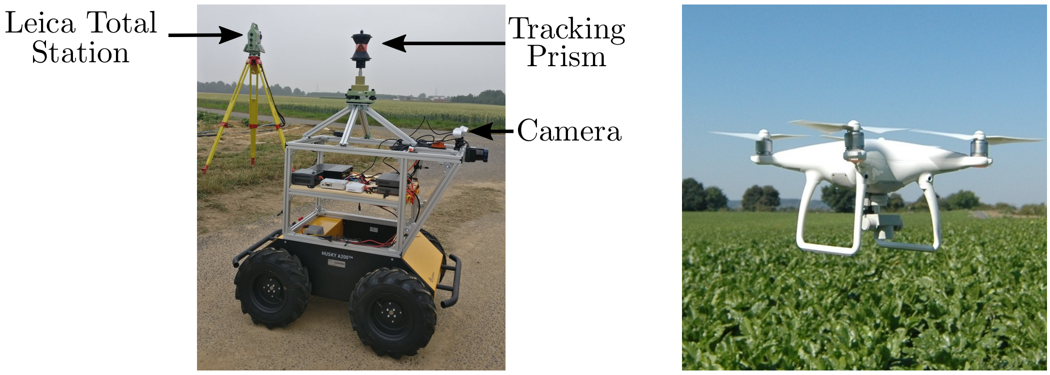


Figure 3.5: Platforms used for the experimental evaluations. Left: Clearpath Husky robot used as the ground robot with a Leica the tracking setup used to record ground truth trajectory in the crop fields. Right: A DJI Phantom 4 used to collect the images for the aerial reference map.

our overall system.

3.5.1 Experimental Setup

The experiments were performed on a real sugarbeet field, where we recorded data over several weeks. The images for generating the aerial map were taken at the beginning of the season using a DJI Phantom 4 UAV. The images were captured from a height of 10 m covering the whole field. The orthomosaic map generated from the images has a ground resolution of 5 mm per pixel. We used a Clearpath HuskyA200 equipped with wheel encoders, Ublox EVK-7 GPS, and a ZED stereo camera for recording the ground robot data. We only use the RGB images from the left camera for our experiments. The camera was mounted at the height of 1.2 m from the base, tilted at an angle of 45 ° towards the ground, see Figure 3.5. We operated the ground robot by manually joysticking it with an average speed of 0.6 m/s. We collected the data over five different sessions, each roughly separated by a week. During this period, the crop size ranged between 5 cm to 20 cm in diameter. Additionally, the farmers performed a weeding treatment just before the third session, by which most of the weeds in the field were removed. We also recorded the robot's ground truth trajectory by tracking a prism target placed on top of the robot using a Leica Total Station TS50 with an accuracy of substantially below 1 cm.

3.5.2 Evaluation of Localization Accuracy

The first experiment is designed to show that we are able to localize with sufficient accuracy required to carry out precision agriculture tasks in crop fields. This essentially requires that the robot both localizes in the correct crop row and is accurate enough to navigate within that row. This means a global accuracy of

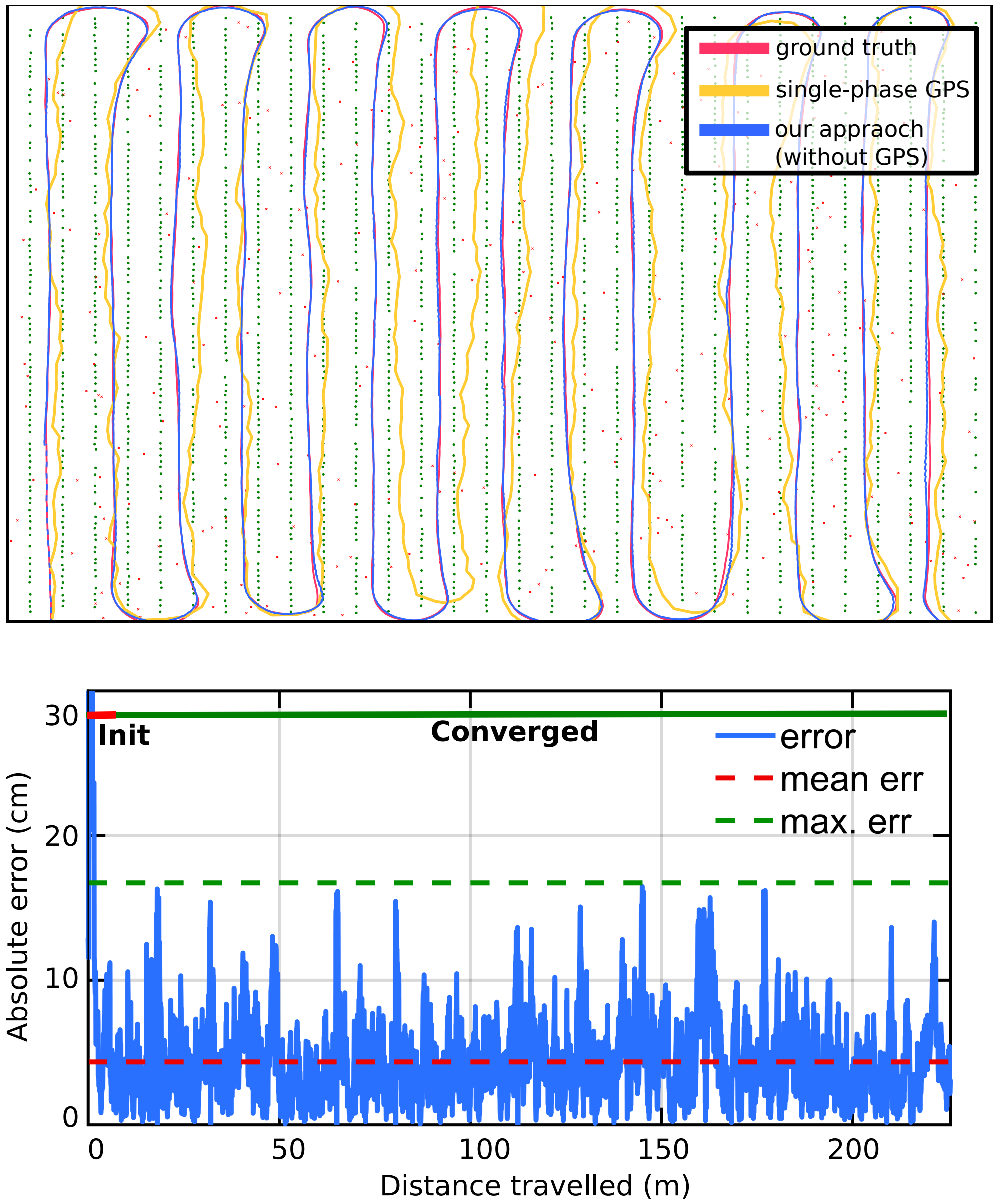


Figure 3.6: Top: Comparison of the trajectory estimated by our localization approach against GPS and ground truth. Bottom: Absolute trajectory error over the whole trajectory.

under 25 cm is required, which corresponds to the inter-crop row distance in our field. This accuracy has to be achieved under changing appearance and strong visual aliasing.

We begin the experiment by initializing our MCL filter with 5,000 particles with an initial variance of 5 m around the estimate provided by the GPS. In Figure 3.6, we show the localization results for the first session using the aerial map created on the same day. To visualize the performance of the filter, we plot the estimated mean pose of the robot (blue), GPS (yellow), and ground truth (red) measurements overlaid on the map. The filter converges after the robot travels a distance of about 6 m (dashed-blue). In Figure 3.6 (top), we see that our approach (blue) provides a smooth estimate of the robot's path along the crop rows, whereas the GPS measurements often "jump" between the crop rows. We evaluate the accuracy of our trajectory estimate in terms of the absolute difference between our solution and the ground truth. We also note that our estimated trajectory is close to the ground truth, indicated by the blue (ours) follows the red (ground truth) trajectory. Once the filter has converged, we obtain an average error of 4.3 cm, with the maximum error being around 17 cm. This error is less than the inter-crop row distance of 25 cm, such that the robot can navigate safely without going over the crops. The localization error over the entire trajectory is plotted in Figure 3.6 (bottom). These results also suggest that our localization system is suitable for crop-row fields with a smaller inter-crop row distance.

Further, we perform an ablation study highlighting the effect of the different feature types and semantics on the localization performance. The results are summarized in Table 3.1. We observe that using detections from all feature types, i.e., crops, weeds, and gaps, provides the best performance. Also, we see that by additionally using semantics, localization the performance is better than using the same features but without the semantic information. In particular, the maximum error is lower while using the semantics. In the last row of the Table 3.1, we observe that when using just the crop features (and not gaps), the filter estimate converges to the wrong row indicated by its mean error of around 50 cm (shifted by two rows). This is caused by the high visual aliasing in the crop fields and indicates that crop locations alone are not sufficient to address this aliasing challenge.

3.5.3 Localization Performance over Multiple Sessions

In this experiment, we demonstrate that our system can localize the robot successfully over multiple sessions spanning several weeks. In contrast, state-of-the-art methods relying on visual features are unable to work properly. For this experiment, we update the map at the end of each session and use it as the reference for

Table 3.1: Ablation study on the localization performance with different features

	Feature type	μ, σ (cm)	max (cm)
With Semantics	crops + weeds + gaps	(4.3, 2.8)	16.7
	crops + weeds	(5.8, 3.9)	18.3
Without Semantics	crops + weeds + gaps	(5.1, 3.1)	22.7
	crops + weeds	(6.6, 3.5)	28.4
	crops	(54.5, 3.5)	79.3

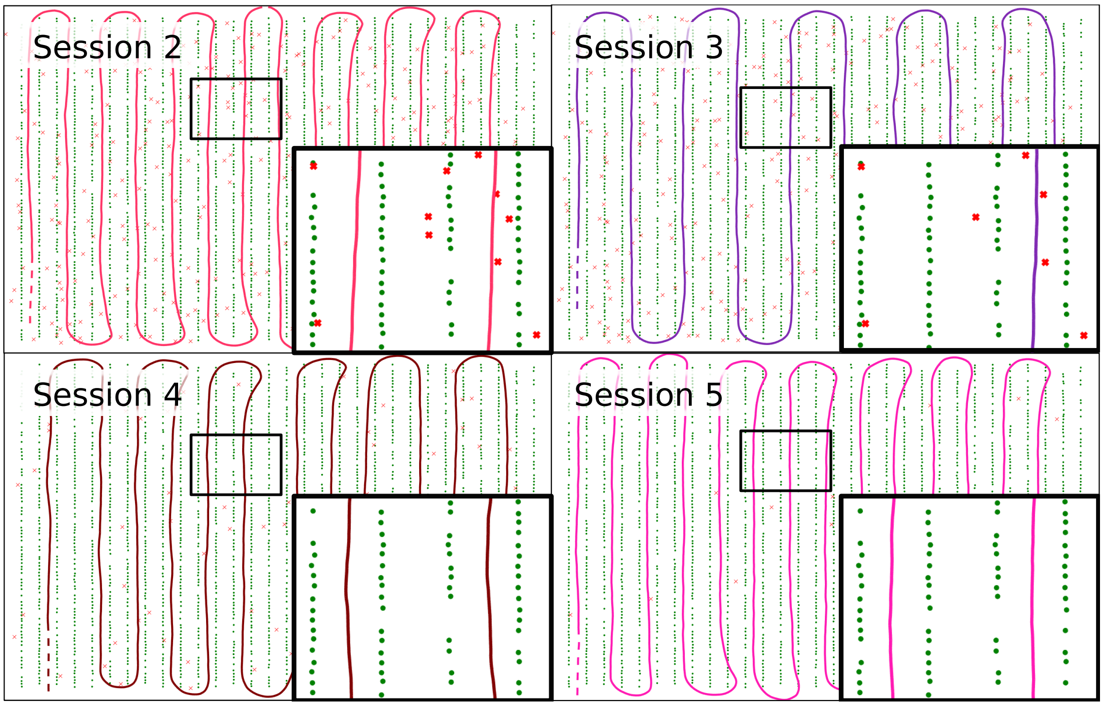


Figure 3.7: Robot localizing over multiple sessions overlaid on updated reference map from the previous session. Dashed trajectory corresponds to the initialization phases. Zoomed-in view visualizes the changes in the map due to the update step reflecting the actual changes in the field.

Table 3.2: Localization performance over multiple sessions

	Initial Map		Updated Map	
Session	μ (σ)[cm]	max [cm]	μ(σ) [cm]	max [cm]
1	4.3 (2.8)	16.7	–	–
2	5.3 (3.6)	18.6	4.8 (2.9)	16.2
3	6.1 (3.8)	21.2	6.9 (3.5)	16.7
4	7.4 (6.2)	29.2	5.1 (3.3)	12.2
5	∞	∞	4.2 (3.8)	14.9

localizing the robot in the next session. Figure 3.7 visualizes the estimated trajectory for sessions 2-5. We were able to localize successfully over all the sessions with an average error of about 5 cm and a maximum error of about 17 cm. Note that in Figure 3.7, the trajectory from different sessions sometimes visit different crop rows. This is because the robot was actually joysticked through these rows and is not an error in the estimated trajectory. In the same figure, we see the zoomed-in view for a particular location in the field, where the landmarks in the map have been updated based on the observations from the previous sessions. This allows the robot to localize accurately despite the changes in the field.

We also analyze the advantage of the map update step by comparing the performance against the setup where the initial map was used as the reference for all the sessions. The results are summarized in Table 3.2. In general, we see that using the updated map results in better performance, both in terms of a lower mean and maximum error. In particular, we see that for session 5, the filter fails to localize using the initial map while it is successful while using the updated map. This is due to the fact that the field changed substantially since the initial map was acquired.

As a qualitative evaluation of the map update step, we report the number of features in the updated map after each session in Table 3.3. We note that the number of crops remain roughly the same over the whole season, and the number of gaps reduce gradually over time as crops grow and gaps are closed by the canopy cover. In particular, we can see that after the map update for session 3, the number of weeds drops from 303 in session 2 to 76 in session 3, reflecting the actual state of the field due the execution of weed control by the farmer. The estimated number further goes down in session 4 when more robot measurements are integrated by the persistence filter described in Section 3.4.

Our approach was able to localize over multiple sessions due to the combination of features that can be detected effectively in a crop field and a map that is curated after each session using robot observations. In contrast, we were unable to localize over multiple sessions using visual features such as SIFT, ORB,

Table 3.3: Number of features after each map update

Feature type	Session 1	Session 2	Session 3	Session 4	Session 5
crops	2472	2422	2406	2384	2353
gaps	417	341	221	211	206
weeds	306	303	76	14	22

Table 3.4: Performance of visual matching across sessions

Desc.	% pairs matched successfully				
Type	Session	Session	Session	Session	Session
	1 vs. 1	1 vs. 2	1 vs. 3	1 vs. 4	1 vs. 5
SIFT [74]	93.8	25.0	12.5	8.3	4.2
ORB [100]	91.7	22.9	18.7	6.2	0
BRISK [68]	89.6	16.7	0	0	0

BRISK or similar. This is because we are not able to find data associations reliably between different sessions. Table 3.4 reflects this situation where matched images from each session and the corresponding images UAV images taken from the first session. Here, we observe that while matching images from session 2 to session 1, about 75% percent of the images fail to match against corresponding images from session 1 when using SIFT descriptor for matching. The situation gets worse when matching images from session 5, where 96% of the pairs do not match. These results are consistent with the results obtained in Chapter 2.

3.5.4 Limitations

Our approach assumes the field to be locally planar while projecting the feature detections on the map. The MCL filter used to estimate the robot's pose also uses positions and orientations on the plane. In principle, to deal with fields with slopes, we can estimate the height by augmenting it to each particle in MCL and defining an appropriate motion update model. However, in our experiments, we did not take it into consideration. Also, as our approach relies on the location of crops, weeds, and gaps, it is suited for crop fields such as sugarbeet, carrot, maize, strawberry, etc., but would not work for example, in wheat/rice fields. Also, note that based on the type of weed treatment, for example, chemical treatment instead of mechanical removal, the prior introduced in the persistent filter needs to be changed, reflecting the actual physical process of the weeds dying slowly rather than instantly.

3.6 Related Work

The combination of aerial survey capabilities of UAVs with targeted intervention abilities of agricultural UGVs can provide effective robotic systems for carrying out precision agriculture tasks. In this context, building and updating a common map of the field is an essential but challenging task. However, the maps built using robots of different types have differences in terms of scale, resolution, and perspective. In addition, the repetitive nature of the crop row structure found within agricultural context renders standard visual feature based registration techniques ineffective. In order to meet these challenges, Potena *et al.* [93] develop a registration pipeline that uses a multi-modal environment representation including a vegetation index map and a digital surface model to associate the data captured from UAV and UGV. Similar to this work, we exploit the semantics of the field in terms of the crop, weed and gaps to obtain data association in this chapter. In addition to the semantics, our localization system also explicitly accounts for the changes in the field over time resulting from plant growth as well as other farm management activities. Therefore, our work presented in chapter differs from the existing approaches by addressing both the viewpoint difference in data acquired from different platforms, and the changes in the field environment over time.

In the past, several works have used aerial imagery as a reference map for localization. This is motivated by the fact that such prior information about the environment can be exploited to improve the localization quality. For example, Kümmerle *et al.* [64] show that by incorporating aerial imagery into a SLAM system, they are able to both localize better and acquire maps with increased global consistency. The main challenge in exploiting information from aerial images is to find the data associations to the ground robot sensor data. This is often challenging due to large viewpoint difference between the two sources. To find these associations, several approaches [122], [79], [67], [63] propose new features that can be detected and matched against aerial images. These approaches typically employ a robust outlier rejection mechanism to deal with the large number of false correspondences. Other approaches by Ding *et al.* [33] and Wang *et al.* [128] obtain correspondences between ground and aerial views by finding vertical structures and other keypoints such as corners and planes which are seen from both views. However, most of the methods that we have discussed are tailored for urban environments having planar surfaces and vertical edges which is not the case for crop fields. But in the spirit of identifying features which can be commonly observed, we exploit plant and gap locations as features which are more suitable for agricultural fields as these locations do not change.

Some other works have looked at developing systems that can localize under appearance changes and different weather conditions. Churchill *et al.* [27] present a localization framework that maintains an environment model using multiple images of the same place taken under different conditions, which is then used for localization in future runs. Milford *et al.* [82] develop a system for matching image sequences under strong seasonal changes by computing a similarity score between the images in a query and database sequence and determine the best-matched image through a graph search procedure. Vysotska *et al.* [125, 126] extended this idea towards an online approach with lazy data association and built up a data association graph online on-demand. These approaches typically exploit cues from image sequences and maintain multiple hypotheses to obtain robust localization performance. They also often employ features that can be detected in images taken in different seasons and viewpoints. Building on ideas from these works, we proposed using features suited for crop fields and using them effectively in an MCL filter to tackle the data association challenge.

Several other approaches exploit semantic information from the environment to find better and robust features for the localization task. For example, Ruchti *et al.* [101] and Christie *et al.* [26] use range information from the laser scanner along with semantics of the environment for matching ground level images against aerial images or OpenStreetMap data. In the agricultural domain, some recent works by Dong *et al.* [34], Winterhalter *et al.* [129], and Kraemer *et al.* [60] have aimed at exploiting situations specific to crop fields for performing data association. Taking inspiration from these works, we incorporated semantics of the plants in terms of crops and weeds which help both in finding correspondences and tackling the visual ambiguity problem.

With a similar goal of operating in the field environments over time, Dong *et al.* [34] propose a localization and mapping system which fuses measurements from different sensors such as camera, GPS, IMU. However, in this approach as matching data from different sessions still relies visual features, it remains vulnerable when visual appearance changes substantially. However, these changes are typical in agricultural fields where the vegetation grows continuously, soil appearance changes due to rain or tire tracks left by tractors operating on the field. On the other hand, our method is able to deal with such situations since it uses the geometrical information which remains mostly static even if the appearance changes dramatically.

3.7 Conclusion

The ability to localize is a pre-requisite for a robot carrying out monitoring and other precision agriculture tasks in a crop field environment. The field environment presents unique challenges such as the highly repetitive structure of the crops leading to visual aliasing as well as the continuously changing appearance of the field, making it difficult to localize over time. In this chapter, we explored the advantages of collaboration with a UAV to improve the localization capabilities of a ground robot. To realize this, we presented a localization system, which uses an aerial map of the field generated from UAV images. We exploited the semantic information of the crops, weeds, and their stem positions as well as gaps in the field to resolve the visual aliasing problem by finding reliable data associations. The choice of our features is motivated by the fact that they must be matched in the presence of large viewpoint differences. These features are observed in a near nadir-view from the UAV, whereas the UGV images are much closer to the ground with a perspective view due to the camera tilt. An additional factor in choosing these features, i.e., the location of the crops, weeds, and gaps, is that they are relatively static over time, and making them suitable for localization over extended time periods.

Using these features, we proposed a Monte-Carlo localization framework to estimate the pose of the ground robot. MCL provides an elegant way to consider multiple hypotheses arising from the similar-looking crop rows and avoids committing to a single crop-row until enough evidence is gathered. In the experiments, we showed that our approach provides crop-row accurate localization. It is better than the accuracy of a typical single-phase GPS that often provides an estimate in the wrong crop row. This is a critical requirement as the UGV must be in the correct crop-row to carry out the various tasks using its implements.

Finally, we keep the reference map of the environment updated by integrating current observations from the ground robot during its operation. This step captures the changes occurring in the environment and allows us to localize for multiple sessions over the crop season successfully. We update the probabilities for each feature's existence via a persistence filter that takes into account the flaws in the detection pipeline as well as any prior information about the feature coming from the field setting. We conducted experiments spanning several weeks on a real sugarbeet field and evaluated our approach over multiple sessions. The experiments demonstrated that our approach provides reliable crop-row level accuracy over several sessions despite the large changes in the field. We also showed that using the UAV reference maps in combination with well-designed features, the ground robot could robustly localize over long periods, whereas current localization techniques relying on typical visual features fail.

Chapter 4

Spatio-Temporal Registration of 3D Point Clouds of Plants

In plant sciences and modern agriculture, high-resolution monitoring of plants plays a vital role [127, 39]. It forms the basis for analyzing crop performance and provides an indicator of the plant stresses. Measuring how individual plants develop and grow over time is often a manual and laborious task. It often even requires invasive methods that harm the crop. For example, the standard approach to measuring the leaf area is to cut off the leaves and scan them with a flatbed scanner. New measurement technologies for measuring and tracking phenotypic traits employing robots and robotic sensors open up possibilities to automate the process of measuring plant performance [37, 38]. In Chapter 2, we registered UAV images over time, which allows us to analyze the growth of the crops in the field. Similarly, in this chapter, we develop a registration technique that operates on data acquired with much higher spatial resolution enabling us to analyze the growth of the parts on an individual plant.

Recent studies by Paulus *et al.* [91] and Klose *et al.* [57] showcase the use of 3D laser data for computing geometric plant traits with high fidelity. This has the potential to be scaled up by equipping agricultural robots equipped with such laser scanners that can acquire 3D plant data from fields and facilitate high-resolution phenotyping. This would be a step forward for scaling up phenotyping from plants grown in a greenhouse to the level of plots and maybe even entire fields. To facilitate such a scale-up in phenotyping using 3D sensing, one of the fundamental requirements is the ability to register scans taken at different times in an automated manner.

Registering plant scans recorded at different points in time is a comparably challenging task due to the anisotropic growth of different organs of the plant, the change in its topology, and the non-rigid deformations caused due to external stimuli such as wind or sunlight. Also, the measurement process using laser

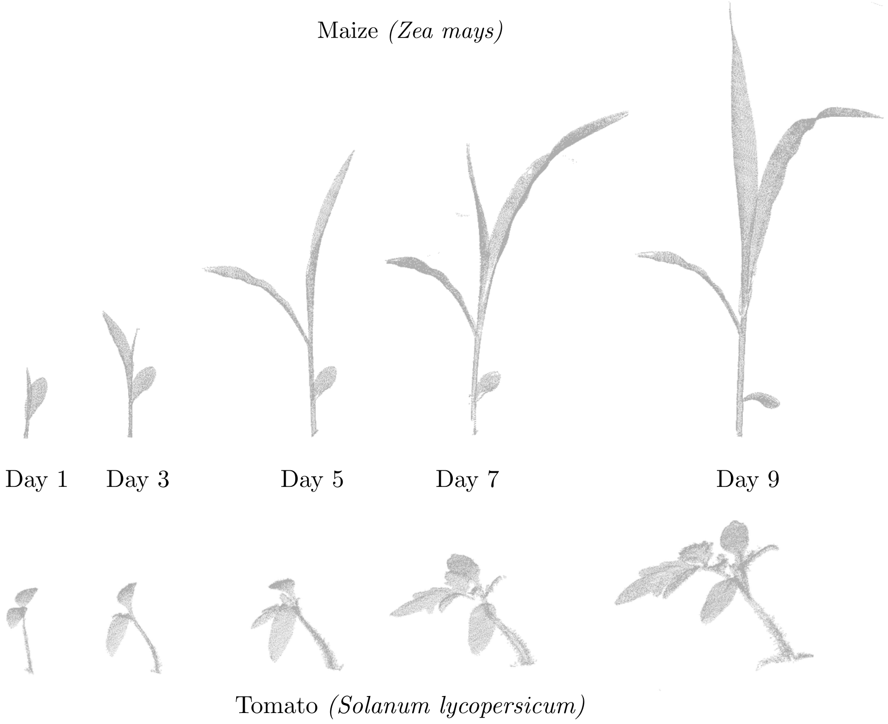


Figure 4.1: A time-series of 3D point clouds of Maize (top) and Tomato (bottom) captured during its growth. Our goal is to develop techniques for automatically registering such 3D scans captured under challenging conditions of changing topology and anisotropic growth of the plant.

scanners usually results in incomplete scans of the plant due to the numerous self-occlusion amongst leaves. Given these challenges, we aim at developing techniques that facilitate the automatic computation of phenotypic traits from 3D time-series point clouds of plants.

Typically, point cloud registration is performed using ICP-based approaches. However, these approaches are often unable to capture the dynamic deformations in the object and are prone to divergence due to new or missing plant organs. This chapter investigates an approach to account for the growth and non-rigid deformations that the plant undergoes. This bears quite some resemblance to the SLAM problem [112], loop closing [28, 25], and ICP-based scan matching [12], often used in the context of laser-based SLAM. Similar to graph-based SLAM in non-static environments, we need to make data associations in changing scenes, align point clouds, and perform optimizations and iterative refinement procedures.

The main contribution of this chapter is a fully automatic registration technique for plant point clouds that have been acquired over time. We propose using the plant's skeleton structure to drive the registration process as it provides a compact representation of the plant by capturing its overall shape and topology. We propose a method for extracting the skeletal structure along with semantic information to perform the data association step. We classify each point of the plant as a leaf or stem point, and a further clustering allows us to compute individual leaf instances. We then determine correspondences between plant skeletons using a hidden Markov model. These correspondences allow us to estimate parameters, which can capture the deformation and the growth of the plant skeleton. We then transfer the deformations estimated on the plant skeletons to the whole point cloud to register the temporally separated point clouds. Using these registration parameters, we are also able to interpolate over the registration parameters to obtain an estimated point cloud at a time instant in-between the actual acquisition times.

The approach that we propose in this chapter enables us to (i) register temporally separated plant point clouds by explicitly accounting for the growth, deformations, and changes in the plant topology, (ii) exploit the skeletal structure as well as semantic information computed from the data, (iii) find correspondences between the different organs of the plant and track them over time. Using our approach, we show robust registration results on long-term datasets of two plant species captured using a 3D laser scanner mounted on a robotic arm. We also demonstrate how our registration procedure forms the basis of an automated phenotyping application where we estimate some basic phenotypic parameters such as leaf area, leaf length, as well as stem length and diameter, and track their development over time in an automated fashion.

4.1 Our Approach to Plant Point Cloud Registration

Our approach to registration operates on a time-series of 3D point clouds of plants. The registration procedure starts with extracting a skeleton along with the organ level semantics for each point cloud. The skeletons are undirected acyclic graphs, which represent the topology or the inner structure of the plant. Each node contains the x, y, z coordinates of its position, a 4x4 affine transformation matrix T to describe the local transformation, and a semantic class label as attributes. We use affine transformations to capture the non-rigid deformations caused due to plant growth. The skeletons extracted from the point cloud data are often imperfect, and we consider this aspect explicitly during the registration procedure. We operate directly on unordered point clouds and do not require a mesh structure or other cues such as the normals providing the inside-outside information of the surface. The skeleton structures allow us to compute data associations between temporally separated 3D scans and use these correspondences to perform an iterative procedure that registers the plant scans obtained at different times. Finally, the registered skeletons are used to deform the complete point cloud, e.g., the point cloud from time step t_1 deformed to time step t_2.

This approach for registering a point cloud pair $(\mathcal{P}_1, \mathcal{P}_2)$ in an iterative manner is similar to the iterations of the popular ICP approach [12]. We alternate between correspondence estimation steps and registration steps given the correspondences. In contrast to the nearest neighbor, point-to-plane, or normal-shooting correspondences used in typical ICP procedures, we use the skeletal structure of the plant to establish correspondences $\mathcal{C}_{12}$. The procedure for the extraction of the skeleton structure from the plant point cloud is described in Section 4.1.1. The correspondence estimation is done via a hidden Markov model (HMM) formulation as detailed in Section 4.1.2. Also, in deviation from a standard ICP procedure, which assumes a rigid transformation between $\mathcal{P}_1$ and $\mathcal{P}_2$, we explicitly model the deformation through different 3D affine transformations defined for each node of the skeleton $\mathcal{S}_1$. We estimate this set of affine registration parameters $\mathcal{T}_{12}$ using a non-linear least-squares procedure as described in Section 4.1.3. We exit the iterative scheme when there is no change in the estimated correspondence set $\mathcal{C}_{12}^t$. After computing the registration parameters between the nodes of the skeletons $\mathcal{S}_1$ and $\mathcal{S}_2$, we apply these parameters to the entire point cloud to obtain the final registered point clouds as described in Section 4.1.4.

4.1.1 Extracting the Skeletal Structure and Semantics

The first step of our approach is to extract the skeleton structure $\mathcal{S}$ of the plant from the input point cloud $\mathcal{P}$. Following the idea proposed by Magistri *et al.* [78], we exploit the semantics of the plant to drive the skeleton extraction process from the point cloud. We first perform a segmentation step aimed at grouping points that belong to the same plant organ, namely a leaf instance or the stem. To do this, we start by classifying each point of the point cloud $\mathcal{P}$ as a point belonging to either the stem or leaf category. We use a standard support vector machine classifier with the x, y, z coordinates along with with the fast point feature histograms (FPFH) [102] as a feature vector. The FPFH technique computes a histogram of directions around a point using the neighborhood information, thereby capturing the local surface properties in a compact form. With these feature vectors as inputs, the support vector machine classifies each point of a plant point cloud into stem and leaf points. We train the support vector machine model in a supervised manner during the training phase by providing labels for a subset of point clouds from the temporal sequence.

After the model is trained, we use it to predict the semantics for all the remaining point clouds of the sequence. Once the classification step is complete, we perform a clustering step to find the individual leaves or the stem as unique instances. We perform the clustering using the density-based spatial clustering

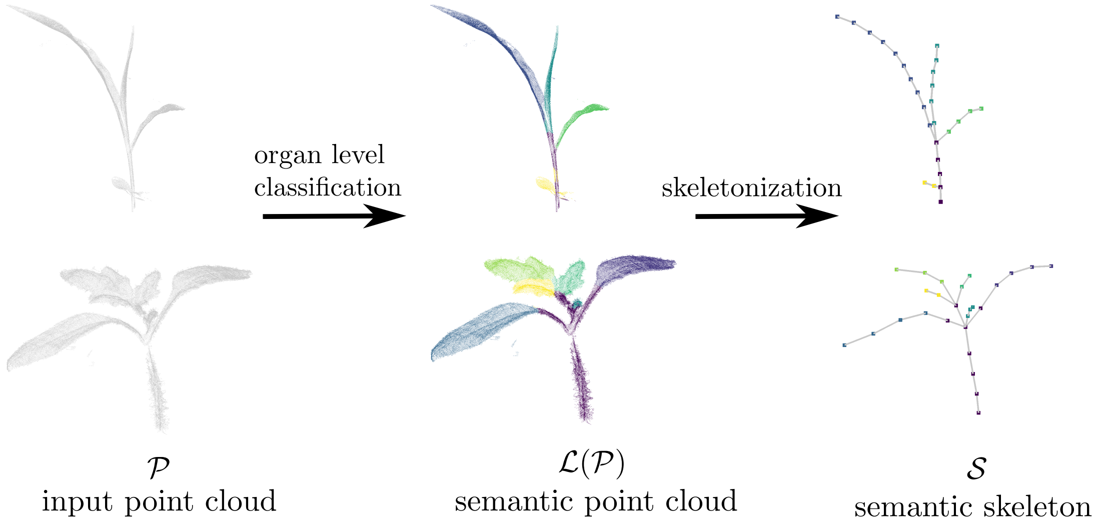


Figure 4.2: Extracting skeletal structure using semantics of the plant. The figure illustrates the skeletonization pipeline for a maize (top) and tomato (bottom) plant scan. Note that for the tomato plant, we classify individual leaflets (green + yellow + light-blue) as separate instances rather than as an individual leaf. The leaflets can be combined into a single leaf in case this distinction is not desired/required for the application.

algorithm [36]. It uses the x, y, z coordinates of the points to obtain an initial segmentation, which is then refined by discarding small clusters and assigning each discarded point to one of the remaining clusters based on a k-nearest neighbor approach.

At this stage, each point in the point cloud $\mathcal{P}$ is assigned to an organ, namely to the stem or to a particular leaf instance. We then learn a set of keypoints for each organ using self-organizing maps [59], which helps us determine the skeleton structure. These keypoints form the nodes of the skeleton structure. Self-organizing maps are unsupervised neural networks that use the concept of competitive learning for clustering the data. They take as input a grid of keypoints that organizes itself to capture the topology of the input data. Given an input grid $\mathcal{G}$ and the input set of points $\mathcal{P}$, the self organizing map defines a fully-connected layer between $\mathcal{G}$ and $\mathcal{P}$. This network learns a transformation for the grid $\mathcal{G}$ points in manner to cluster the data $\mathcal{P}$ effectively. The learning process is composed of two alternating steps until convergence. First, the winning unit is computed as:

$$x = \underset{p \in \mathcal{P}}{\operatorname{argmin}} \, ||p - w_i||, \tag{4.1}$$

where p is a randomly chosen sample from $\mathcal{P}$ and w_i is the weight vector most similar to x, also called the best matching unit. The second step consists of updating the weights of each unit according to:

$$w_n = w_n + \eta \, \beta(i) \, (x - w_i), \tag{4.2}$$

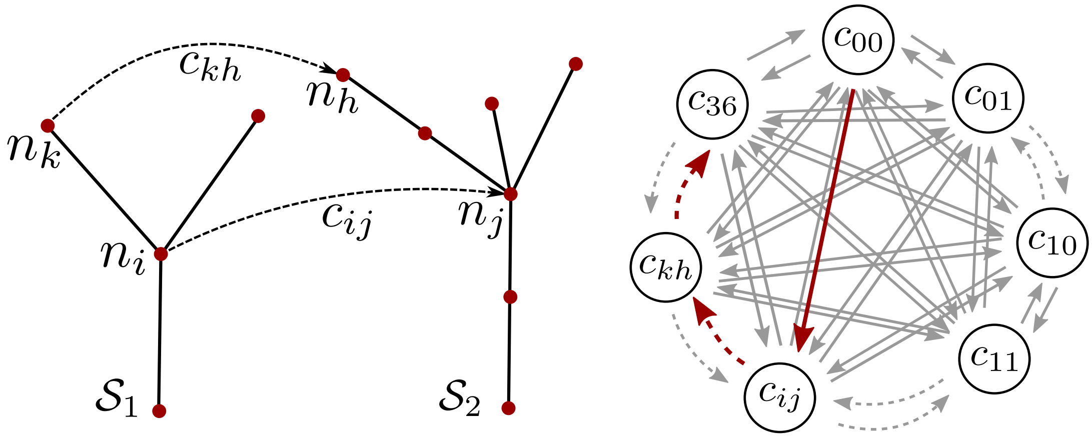


Figure 4.3: Left: Example of skeletal matching with all the variables involved. Right: Hidden Markov model used for correspondence estimation. The red line depicts the sequence of best correspondence estimated by the Viterbi algorithm.

where η is the learning rate and $\beta(i)$ a function, which weights the distance between unit n and the best matching unit.

In our case, we exploit the knowledge that individual components of the skeleton can be well explained using curved lines in the local neighborhood. As a result, we define the grid $\mathcal{G}$ for each organ as an $n \times 1$ chain of 3D points that will form the nodes along the skeleton for that organ. The length of the chain n is proportional to the size of the organ, such that the keypoints are expected to have a minimum distance between 1 cm between them. In this way, it is possible to obtain a skeleton-like structure for each plant of the temporal sequence of plant point clouds that is consistent in terms of the number of nodes depending on the plant's size. Figure 4.2 visualizes the organ segmentation as well as the skeleton structures extracted from the input point cloud $\mathcal{P}$ for two sample scans of our dataset.

Although we use the skeletal structure extracted using the procedure described above, our registration procedure is independent of how the skeletons are computed. There are several techniques for extracting the skeleton structures from point clouds. Huang *et al.* [52] and Tagliasacchi *et al.* [117] propose approaches to extract curve skeletons from unorganized point clouds, which can be used as an input to our approach. A detailed state-of-the-art review for extracting skeletons of 3D objects is given by Tagliasacchi *et al.* [116]. The skeletons obtained from any of these techniques can be used for the next steps of the registration procedure.

4.1.2 Estimating Skeletal Correspondences

Before data from any 3D objects can be aligned, we need to establish the data associations between the sensor readings, i.e., estimating which part of the source point cloud $\mathcal{P}_1$ corresponds to which parts of the target point cloud $\mathcal{P}_2$. Establishing this data association is especially hard for objects that change their appearance, and automatic processes are likely to contain data association outliers, which affects the subsequent alignment. For registering temporally separated plant scans, we propose to perform data association by matching the corresponding skeleton structures and not work directly on the raw point clouds.

In this data association step, we estimate correspondences between two skeletons by exploiting their geometric structure and semantics, which are computed using the approach described in the previous section. As the skeleton structure and the semantics are estimated from sensor measurements, it might suffer from several imperfections. To cope with these imperfections in the individual skeletons and inconsistencies between them, we use a probabilistic approach to associate the skeleton parts instead of graph matching approaches, which typically do not tolerate such errors well. We, therefore, formulate the problem of finding correspondences between the skeleton pair, i.e., the source skeleton $\mathcal{S}_1$ and target skeleton $\mathcal{S}_2$ using a hidden Markov model formulation [98] as illustrated in Figure 4.3. The HMM model provides the flexibility to encode different cues, define constraints for the correspondences, as well as include prior information about the skeleton structure. This allows us to track several correspondence candidates and choose the best correspondences between the skeleton pair.

The unknowns or the hidden states of the HMM model represent all the potential correspondences between the nodes of the two skeletons. In addition, we also add a so-called “not matched state” for each node in the HMM to account for the situations in which the node may have no correspondences at all. For example, this happens when nodes belong to new organs that were not present before or when new nodes emerge on the curve skeleton due to the plant growth. As required in a HMM formulation, we need to define the emission cost Z and the transition cost Γ. The emission cost Z describes the cost for a given hidden state (here the correspondence information) to produce a certain observation. In our case, the observations are the sequence of nodes of the first skeleton $\mathcal{S}_1$ arranged in depth first manner starting from the node at the base of the stem. We define this cost for a correspondence $c_{ij} \in \mathcal{C}_{12}$ between node n_i of $\mathcal{S}_1$ and node n_j of $\mathcal{S}_2$ as:

$$\begin{aligned} Z(c_{ij}) &= w_d |deg(n_i) - deg(n_j)| + \\ &\quad w_e \left\| x_i - x_j \right\| + \\ &\quad w_{sem} \rho_{sem}(\mathcal{L}(n_i), \mathcal{L}(n_j)), \end{aligned} \tag{4.3}$$

where the first term yields the absolute difference, denoted by $|\cdot|$, between the degrees of the corresponding nodes, where $deg(n)$ is the number of edges incident to a node. The second term refers to the Euclidean distance, denoted by $\|\cdot\|$, between them with x_i, x_j being the 3D locations of the nodes n_i, n_j respectively. The final term ρ_{sem} is set to one in case the semantics for the corresponding nodes $\mathcal{L}(n_i), \mathcal{L}(n_j)$ the are not the same; otherwise, it is set to zero. The idea behind combining these three terms is to capture the geometric aspects, i.e., the topology difference, the spatial distance between the nodes, and the semantics of the skeleton nodes being matched. This combined cost will be smaller for correspondences between nodes that have similar topology, are located close to each other, and have the same semantic label. We weigh all the three terms using w_d, w_e, and w_{sem} to properly scale the measures.

The transition cost Γ describes the cost involved in transitioning from one hidden state c_{ij} to another c_{kh}. This can be treated as the cost involved in having c_{kh} as a valid match given that c_{ij} is a valid match as well. We define this cost as:

$$\begin{aligned}\Gamma(c_{ij}, c_{kh}) \quad = \quad & |d_g(n_i, n_k) - d_g(n_j, n_h)| + \\ & w_{nbr}|n_{br}(n_i, n_k) - n_{br}(n_j, n_h)| + \\ & \rho_{dir}((x_i - x_j), (x_k - x_h)), \end{aligned} \tag{4.4}$$

where the first term computes the difference of the geodetic distances d_g between the nodes involved in the two correspondence pairs along their respective skeletons. This means that a pair of correspondences (c_{ij}, c_{kh}) having equal geodetic lengths $d_g(n_i, n_k)$ along $\mathcal{S}_1$ and $d_g(n_j, n_h)$ along $\mathcal{S}_2$ will have a lower cost than the ones which have much different lengths along the skeleton. The second term captures the difference in the number of branches n_{br}, i.e., nodes with degree greater than 2, along the way on the skeleton. The weight w_{nbr} is automatically set as the maximum geodetic distance between all node pairs of the first skeleton. The final term ρ_{dir} is a function that penalizes the correspondence pairs (c_{ij}, c_{kh}) with a large cost if the directions determined by $(x_i - x_j)$ and $(x_k - x_h)$ are opposite, i.e., the angle between them are greater than $\frac{\pi}{2}$.

Once the emission and transition costs are defined, we compute the correspondences between the skeletons by performing an inference on the HMM. The result is the most likely sequence of hidden variables, i.e., the set of correspondences between $\mathcal{S}_1$ and $\mathcal{S}_2$. We perform this inference using the Viterbi algorithm [123]. In case a node has more than one correspondence, we choose the correspondence with the smaller Euclidean distance to ensure a one-to-one correspondence. Figure 4.3 (left) shows an example skeleton pair for which we want to estimate the correspondences $\mathcal{C}_{12}$. Figure 4.3 (right) depicts the HMM for the example pair where the red path indicates the set of correspondences estimated by the Viterbi algorithm. The HMM model only shows a subset of the connections between the hidden states, where in practice each state is connected to every other state.

4.1.3 Computing Skeletal Deformation Parameters

In this step, we compute the registration parameters between $\mathcal{S}_1$ and $\mathcal{S}_2$ given the set of correspondences $\mathcal{C}_{12}$. While registering temporally separated plant scans, the shape and the topology of the plant changes. Therefore, to capture these changes, we need to forego the usual assumption of rigidity often used in point cloud registration. Our goal is to capture the non-rigid changes by computing sets of deformation parameters between skeleton parts of the respective plant scans. We estimate these deformation parameters through a non-linear least-squares optimization procedure based on the correspondences obtained from the procedure described in the previous section.

To model the deformations between the plant scans, we attach an affine transformation T_i to each node n_i of the skeleton $\mathcal{S}_1$ Figure 4.4 (left). The intuition behind such a model is that the skeleton may be deformed differently at different locations along the skeleton. By modeling the deformations through a 3D affine transformation with 12 unknown parameters per node, we are able to capture the growth as well as bending of the plant via the scaling, shearing, and rotation parameters.

We define the objective function of the optimization problem as a combination of the three energy terms. The first term $E_{corresp}$ is defined as:

$$E_{corresp} \quad = \quad \sum_{c_{ij} \in \mathcal{C}_{12}} \|T_i x_i - y_j\|, \tag{4.5}$$

where x_i and y_j are the node positions given by the correspondence pair c_{ij} estimated in Section 4.1.2. This energy term captures the distance between corresponding nodes in $\mathcal{S}_1$ and $\mathcal{S}_2$ and strives to make this error as small as possible during optimization.

The second energy term E_{rot} captures how close the estimated affine transformation is to a pure rotation and it determines the smoothness of the deformation. We define E_{rot} as:

$$E_{rot} \quad = \quad \sum_{\substack{i=1 \\ j=\mathrm{mod(i+1,3)}}}^{3} (R_{c_i}^{\top} R_{c_j})^2 + \sum_{i=1}^{3} (R_{c_i}^{\top} R_{c_i} - 1)^2, \tag{4.6}$$

where R_{c_i} represents the columns of the rotation part of affine transformation (i.e. the first three rows and columns of T_i). The first term in E_{rot} in Equation (4.6) measures the deviation for a pair of columns to be orthogonal with each other, whereas the second term measures the deviation of each column from being unit length. The term E_{rot} forces the estimated affine parameters T_i to be as close to a true rotation as possible.

We also define a regularization term E_{reg} as:

$$E_{reg} \;=\; \sum_{j \in N(i)} \operatorname{norm}_F(T_i^{-1} T_j - I), \tag{4.7}$$

where T_i, T_j are transformations corresponding to nodes n_i, n_j such that j is the neighbor $N(i)$ along $\mathcal{S}_1$, and norm_F is the Frobenius norm after performing the homogeneous normalization of the involved matrices. The term E_{reg} is a regularizer, which forces the transformation parameters of neighboring nodes to be similar. This results in a smooth deformation along the skeleton and achieves similar results as the as-rigid-as-possible constraint described by Sorkine *et al.* [111]. The regularization term is also necessary to constrain the nodes that do not have any correspondences. Finally, the combined energy E_{total} is obtained as a weighted combination of all the three energies as:

$$E_{total} \;=\; w_{corresp} E_{corresp} + w_{rot} E_{rot} + w_{reg} E_{reg} \tag{4.8}$$

We use the weights $w_{corresp} = 100$, $w_{rot} = 10$, and $w_{reg} = 1$ for all the example in our datasets. The weights have been chosen such that the cost due to each component of the loss is in the same order of magnitude. We employ the standard Gauss-Newton algorithm to solve the unconstrained non-linear least squares problem [49]. We also use Cauchy's robust kernel [76] for the error residuals belonging to $E_{corresp}$ as this prevents incorrect correspondences from having an adverse effect during the optimization process. The robust kernel down-weighs potentially wrong correspondences, which have large residuals. Later, in chapter Chapter 5, we propose to use an adaptive robust kernel to deal with a larger number of wrong correspondences.

Overall, our procedure in this section is related to the formulation by Sumner *et al.* [114] for estimating deformation parameters for surfaces parametrized as triangular meshes. In our case, we adapt the energy terms to reflect the constraints valid for deformation between curve skeletons as opposed to surfaces. Furthermore, the approach by Sumner *et al.* [114] cannot fully constrain the nodes with a degree smaller than 3, but is essential for the registration of curve skeletons.

4.1.4 Point Cloud Deformation

Traditional approaches to point cloud registration assume rigid objects. In this case, the alignment results in the execution of a 6 degree of freedom transformation consisting of rotations and translations. This, however, is substantially different in our case. To obtain the final registered point cloud $\hat{\mathcal{P}}_1$ of a growing plant, we need to apply the deformation parameters estimated for the skeleton nodes to all the 3D points of the scan. This means that the individual data points will be affected by individual affine transformation to obtain the aligned cloud.

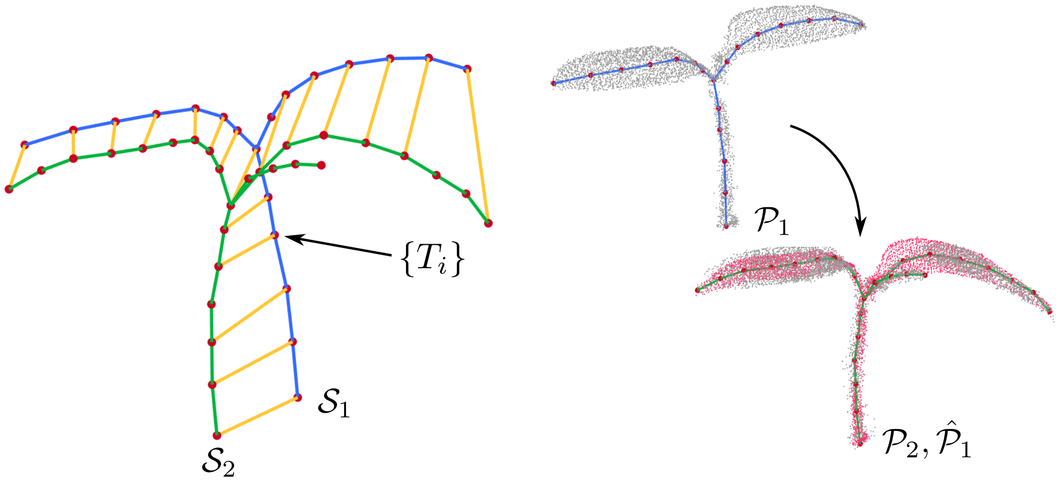


Figure 4.4: Left: Registering the skeleton pair involves estimating the deformation parameters attached to the nodes of the source skeleton $\mathcal{S}_1$. Right: Transferring the deformation results to the entire point cloud.

For each point $p \in \mathcal{P}_1$, we obtain the deformed point $\hat{p}$ as a weighted sum of affine transformations corresponding to the two nearest nodes to the point p as

$$\hat{p} \quad = \sum_{k \in N(p)} \alpha_k T_k p, \tag{4.9}$$

where k is the index of the nearest node $N(p)$ and α_k is computed according to the projection of the point p on the edge of the skeleton determined by the nearest nodes. Let p_e be the projection of point p on edge e. Then the weight is given by:

$$\alpha_k \quad = \quad 1 - \frac{\|p - e\|}{\|e\|}. \tag{4.10}$$

An example of the resulting deformed source point cloud $\hat{\mathcal{P}}_1$ overlaid on the target point cloud $\mathcal{P}_2$ is visualized in Figure 4.4 (right).

4.1.5 Iterative Non-Rigid Registration Procedure

We use the steps from the previous sections to formulate an iterative approach to register the point cloud pair $(\mathcal{P}_1, \mathcal{P}_2)$ as summarized in Algorithm 1. Similar to the popular ICP approach [12], we alternate between correspondence estimation steps and registration steps given the correspondences. We start out by computing the organ level instance segmentation and skeleton structure with semantic information (lines 3-6 of Algorithm 1). We then start the iterative procedure, which alternates between estimating the correspondences $\mathcal{C}_{12}$ (line 9) and the registration parameters, i.e., the 3D affine transformations attached to each node

Algorithm 1 Skeleton-driven iterative non-rigid registration procedure

```
1: $\mathcal{P}_1, \mathcal{P}_2$    ▷ Input point clouds
2: $\mathcal{C}_{12}^{t-1}, \mathcal{C}_{12}^{t} = \emptyset$    ▷ Initialization
3: $\mathcal{O}_1 \leftarrow$ performInstanceSegmentation($\mathcal{P}_1$)    ▷ Segment $\mathcal{P}_1$
4: $\mathcal{O}_2 \leftarrow$ performInstanceSegmentation($\mathcal{P}_2$)    ▷ Segment $\mathcal{P}_2$
5: $\mathcal{S}_1 \leftarrow$ computeSemanticSkeleton($\mathcal{P}_1, \mathcal{O}_1$)    ▷ Compute skeleton $\mathcal{S}_1$
6: $\mathcal{S}_2 \leftarrow$ computeSemanticSkeleton($\mathcal{P}_2, \mathcal{O}_2$)    ▷ Compute skeleton $\mathcal{S}_2$
7: while $(\mathcal{C}_{12}^{t} \setminus \mathcal{C}_{12}^{t-1}) \cup (\mathcal{C}_{12}^{t-1} \setminus \mathcal{C}_{12}^{t}) = \emptyset$ do    ▷ Repeat until matches are same
8:     $\mathcal{C}_{12}^{t-1} = \mathcal{C}_{12}^{t}$
9:     $\mathcal{C}_{12}^{t} \leftarrow$ findSkeletalCorrespondences($\mathcal{S}_1, \mathcal{S}_2$)    ▷ Compute matches
10:    $\mathcal{T}_{12} \leftarrow$ compSkeletalDeformation($\mathcal{S}_1, \mathcal{S}_2, \mathcal{C}_{12}^{t}$)    ▷ Compute deformation
11: $\hat{\mathcal{P}}_1 \leftarrow$ applyDeformation($\mathcal{P}_1, \mathcal{T}_{12}$)    ▷ Apply deformation to $\mathcal{P}_1$
```

(line 10). By iterating through these steps multiple times, we can obtain new correspondences, which might not have been captured before due to the large distance between the skeletons given their initial configuration. Finally, we exit the iterative scheme when there is no change in the estimated correspondence set $\mathcal{C}_{12}^{t}$. After computing the registration parameters $\mathcal{T}_{12}$ between the nodes of the skeletons $\mathcal{S}_1$ and $\mathcal{S}_2$, we apply these parameters to the entire point cloud $\mathcal{P}_1$, which results in the registered point cloud $\hat{\mathcal{P}}_1$ (line 11).

4.1.6 Interpolating Point Clouds

In addition to registering the plant scans recorded at different times, we would also like to interpolate how the plant may be deformed at an intermediate time in-between the actual acquisition times. We compute the deformed point cloud by interpolating the deformation parameters T estimated between the two registered scans in Section 4.1.3. To obtain a smooth interpolation, we first decompose the estimated affine transformation T into scale/shear transformation T_s, pure rotation T_R, and a pure translation T_t using the polar decomposition approach described by Shoemake [109].

$$T = T_s T_R T_t \tag{4.11}$$

We then linearly interpolate T_s and T_t to obtain the transformation at time t. For interpolating T_R, we use the spherical linear interpolation described by Shoemake [108]. We show an example of point cloud interpolation at an intermediate time interval in Figure 4.5.

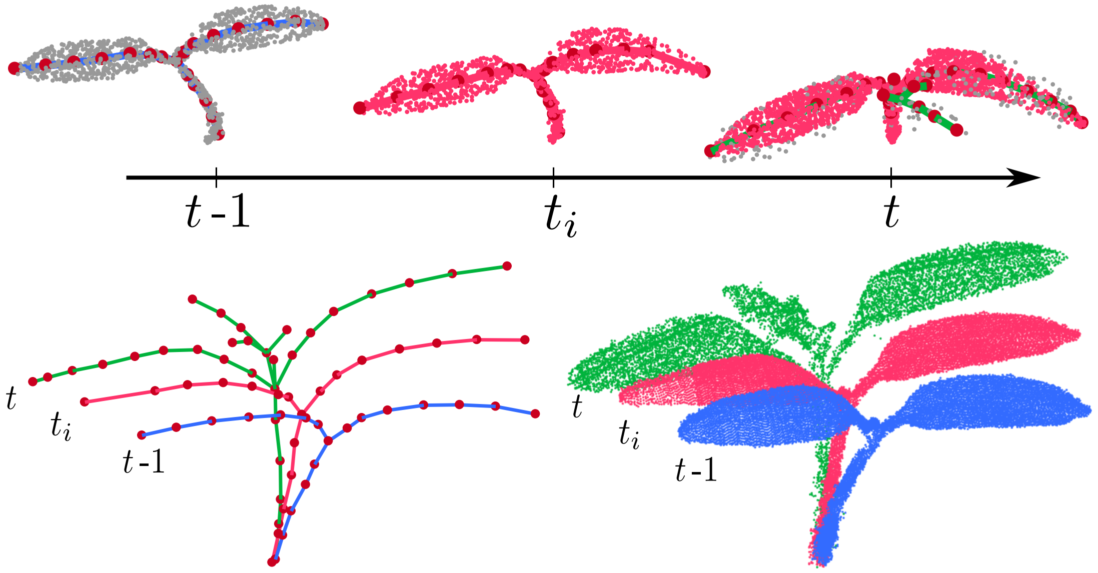


Figure 4.5: Top: interpolation of point clouds at an intermediate time interval. Point clouds (gray) at time $t-1$ and t come from actual scan measurements whereas the points cloud (pink) at an in-between instant t_i is the interpolated scan. Bottom left: shows the skeletons at time $t-1$ (blue), t_i (pink) and t (green). Bottom right: shows the corresponding point clouds in the same color as the skeletons. We see that the interpolated skeleton and the point cloud (in pink) captures the growth well between $t-1$ and t.

4.2 Experimental Evaluation

In this section, we evaluate the performance of our skeleton-driven non-rigid registration technique. We show that our approach is able to (i) compute a skeleton from the plant point cloud by exploiting the semantics, (ii) register temporally separated plant point clouds by explicitly accounting for the growth, deformations and changes in the plant topology, (iii) find correspondences between the different organs of the plant, which allows for tracking plant growth parameters over time, (iv) demonstrate robust registration results on challenging datasets of two different plant types.

4.2.1 Dataset Description

We evaluate our approach on time-series 3D point cloud data of three sample plants of maize (*Zea mays*) and tomato plants (*Solanum lycopersicum*). The scans were recorded using a robotic arm (Romer Absolute Arm) equipped with a high precision laser scanner. The dataset was recorded daily over a period of 10 days. This results in a total of 60 point clouds, which are used for the experimental evaluation. Some sample point clouds from the datasets are shown in Figure 4.6. The plants have been scanned in a manner so as to minimize self occlusions whenever possible. The point cloud data undergoes a pre-processing step where all the points that do not belong to the plant are removed, such as the points belonging to the soil and the pot. The datasets cover substantial growth of the plants, starting from the two cotyledons (i.e., seed leaves) at the start to around eight leaflets (2 cotyledons + 2 leaves) for the tomato plants and 1 to 4 leaves for the maize plants. The plants undergo substantial leaf and stem growth and includes several branching events till the end of the dataset acquisition period as illustrated in Figure 5.1.

The datasets used in experiments have been captured by researchers at the Institute of Geodesy and Geo-information [105]. We have post-processed the data as described above and also labeled a portion of the data used to train the algorithm for the semantic segmentation task. The data used in the evaluation is available at `https://www.ipb.uni-bonn.de/data/4d-plant-registration/`.

4.2.2 Evaluation of Semantic Classification of Plant Point Clouds

In the first experiment, we evaluate the performance of our approach for organ level semantic classification. The classification system has been trained on two randomly selected point clouds from each dataset. All the remaining point clouds in the sequence are used as test datasets. The ground truth information for both the training and test sets have been generated manually by human users. We show the qualitative results computed by our approach by visualizing the semantic instances for some point clouds from the two datasets in Figure 4.6. Each stem and leaf instance is visualized with a different color. We can visually inspect the stem and the leave instances throughout the temporal sequence and see that the classification is successful for instances despite their size and shape changing over time. The colors of the same leaf instances do not persist over the temporal sequence since the data associations between them have not been computed at this stage.

We also perform a quantitative evaluation of the classification performance of our classification approach by computing standard metrics such as precision

$$p = \frac{tp}{tp + fp}, \tag{4.12}$$

recall,

$$r = \frac{tp}{tp + fn}, \tag{4.13}$$

and intersection over union (IoU)

$$IoU = \frac{tp}{tp + fp + fn}, \tag{4.14}$$

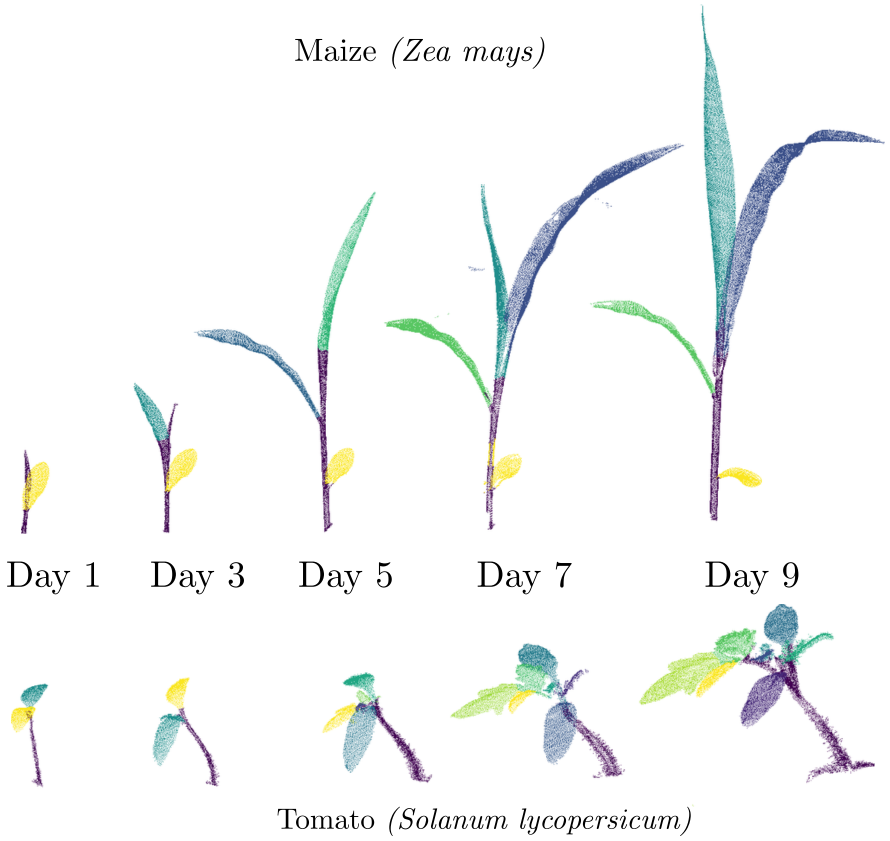


Figure 4.6: Semantic classification of Maize (top) and Tomato (bottom) point clouds. Each stem and leaf (or leaflet) instance is visualized with a different color. Note that the colors of same leaf instances do not correspond over time, as data associations have not been computed at this stage.

for the maize and tomato plant datasets. The results are summarized in Table 4.6, Table 4.7, and Table 4.8 respectively. For each metric, we show the mean, minimum, and maximum values over the dataset as well as the standard deviation. In the definitions above, *tp* stands for true positive, *fp* for false positive, and *fn* for false negative. In all tables, the SVM is responsible for the stem and the leaf class, while instance refers to the unsupervised clustering of individual leaves. We obtain over 90% precision and recall for leaf point in both the datasets, whereas they are around 85% for the stem points. Regarding the leaves instances, all the three metrics are around 90%. The results are summarized in Table 4.5.

Despite these accurate results, it is worth noticing that the ability of the SVM to classify stem points is lower on the maize dataset than on the tomato dataset. This is due to a smoother transition between stem and leaves in the maize plants. In contrast, the performance of the clustering is higher on the maize dataset. This behavior can be explained by looking at how leaves develop in the two species. While for the maize plants, there is a clear separation between individual leaves, this separation is not as clear for the leaves in the tomato plants.

Table 4.1: Semantic classification results for datasets

Dataset	Precision			Recall			*IoU*		
	Leaf	Stem	Instance	Leaf	Stem	Instance	Leaf	Stem	Instance
Maize	95.5	86.3	94.4	92.9	85.7	94.7	94.75	80.01	94.02
Tomato	97.9	89.6	83.4	96.5	92.2	78.2	93.42	82.58	69.14

Table 4.2: Precision values for class-wise and instance segmentation on our datasets

Dataset	Stem				Leaf				Instances			
	mean	min	max	std	mean	min	max	std	mean	min	max	std
Maize	86.3	74.5	99.6	8.6	95.5	93.6	99.4	2.2	94.4	91.9	99.6	2.7
Tomato	89.6	68.6	99.2	9.6	97.9	96.9	99.0	0.9	83.4	59.8	99.7	15.4

Table 4.3: Recall values for class-wise and instance segmentation on our datasets

Dataset	Stem				Leaf				Instances			
	mean	min	max	std	mean	min	max	std	mean	min	max	std
Maize	85.7	48.6	99.1	16.3	92.9	89.1	99.8	3.3	94.7	91.3	99.6	3.1
Tomato	92.2	60.6	99.4	11.4	96.5	74.0	99.6	8.3	78.2	66.6	99.4	11.2

Table 4.4: Intersection over Union (IoU) score for our datasets

Dataset	Stem				Leaf				Instances			
	mean	min	max	std	mean	min	max	std	mean	min	max	std
Maize	80.0	47.8	94.6	12.7	94.6	88.9	97.4	2.5	94.0	91.0	97.6	2.3
Tomato	82.6	60.6	92.2	10.4	93.4	73.9	99.0	7.8	69.1	51.4	98.7	15.7

Table 4.5: Semantic classification results for datasets

Dataset	Precision			Recall			IoU		
	Leaf	Stem	Instance	Leaf	Stem	Instance	Leaf	Stem	Instance
Maize	95.5	86.3	94.4	92.9	85.7	94.7	94.75	80.01	94.02
Tomato	97.9	89.6	83.4	96.5	92.2	78.2	93.42	82.58	69.14

Table 4.6: Precision values for class-wise and instance segmentation on our datasets

Dataset	Stem				Leaf				Instances			
	mean	min	max	std	mean	min	max	std	mean	min	max	std
Maize	86.3	74.5	99.6	8.6	95.5	93.6	99.4	2.2	94.4	91.9	99.6	2.7
Tomato	89.6	68.6	99.2	9.6	97.9	96.9	99.0	0.9	83.4	59.8	99.7	15.4

Based on these organ level semantic classification results, we extract the skeletons for each plant point cloud as described in Section 4.1.1. We obtain a suitable skeleton representation for individual plant scans and ensure that it is connected, which is an assumption that the deformation estimation step described in Section 4.1.3 relies upon. As the skeletal representation in itself is not uniquely defined, we do not provide any quantitative evaluation for resulting skeletons. They are, however, suitable for performing registration between different scans, as shown in further experiments. We see some examples of the plant skeletons extracted using this approach in Figure 4.7 and Figure 4.8.

Based on these organ level semantic classification results, we extract the skeletons for each plant point cloud as described in Section 4.1.1. We obtain a suitable skeleton representation for individual plant scans, and ensure that it is connected, which is an assumption that the deformation estimation step described in Section 4.1.3 relies upon. As the skeletal representation in itself is not uniquely defined, we do not provide any quantitative evaluation for resulting skeletons. They are, however, suitable for performing registration between different scans as shown in further experiments. We see some examples of the plant skeletons extracted using this approach in Figure 4.7 and Figure 4.8.

Table 4.7: Recall values for class-wise and instance segmentation on our datasets

Dataset	Stem				Leaf				Instances			
	mean	min	max	std	mean	min	max	std	mean	min	max	std
Maize	85.7	48.6	99.1	16.3	92.9	89.1	99.8	3.3	94.7	91.3	99.6	3.1
Tomato	92.2	60.6	99.4	11.4	96.5	74.0	99.6	8.3	78.2	66.6	99.4	11.2

Table 4.8: Intersection over Union (IoU) score for our datasets

Dataset	Stem				Leaf				Instances			
	mean	min	max	std	mean	min	max	std	mean	min	max	std
Maize	80.0	47.8	94.6	12.7	94.6	88.9	97.4	2.5	94.0	91.0	97.6	2.3
Tomato	82.6	60.6	92.2	10.4	93.4	73.9	99.0	7.8	69.1	51.4	98.7	15.7

4.2.3 4D Registration of Plant Point Clouds

The second experiment is designed to illustrate the results of our plant registration pipeline for time-series point cloud data of the plants and to quantitatively evaluate the accuracy of the registration pipeline. Figure 4.7and Figure 4.8 illustrate the results of the registration procedure for two example scan pairs. The first example (Figure 4.7) visualizes the registration results for scans from consecutive days, whereas the second example (Figure 4.8) shows registration between scans which are farther apart (4 days here). For both examples, we show the input point clouds ($\mathcal{P}_1, \mathcal{P}_2$) along with their corresponding skeletons. The correspondences estimated during the registration procedure are depicted by the yellow-lines joining nodes of the skeleton pair. Our approach was able to find the correspondences reliably despite the growth and the change in topology, which is especially prominent in the second example (Figure 4.8). We visualize the final registered point cloud $\hat{\mathcal{P}}_1$ (in pink) by deforming the point cloud $\mathcal{P}_1$ using the deformation parameters estimated by our approach and overlay it on the target point cloud $\mathcal{P}_2$ (in gray) and observe that it overlaps well indicating that the registration results are reasonable.

Further, we quantitatively evaluate the accuracy of our registration pipeline by registering all consecutive scans of the two datasets. First, we compute the accuracy of our skeleton matching procedure by computing the percentage of correspondences estimated correctly. We define the correct correspondences as those which belong to the same organ (i.e., the same leaf or the stem) in the skeleton pair as there is no unique way to define a correct correspondence due to the growth in the plant. We manually label the different organs of the plant with a unique identifier to provide the ground truth to compute this metric. For our tomato datasets, we obtain an average of 95% correct correspondences between consecutive skeleton pairs with most pairs having all the correspondences estimated correctly. For the maize dataset, we obtain 100% of the correspondences between consecutive days correctly. Similarly, we also evaluated the accuracy of the correspondence estimation between skeleton pairs with 2 and 3 days apart from each other. For the tomato dataset, we obtain again an average of 95% correspondences with scans taken 2 days apart, whereas this falls down to 88% with scans taken 3 days apart. Again for the maize dataset, we obtain all the

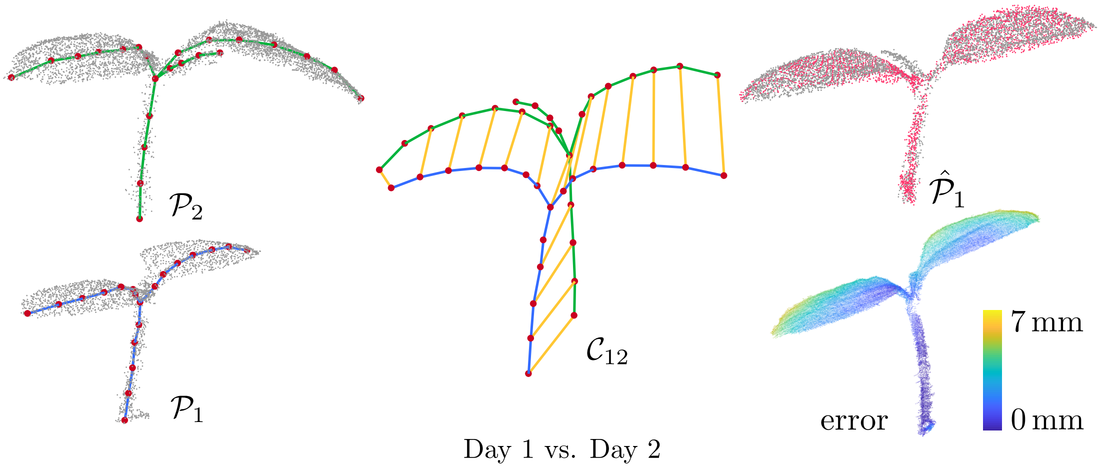


Figure 4.7: 4D registration of point clouds. Shows registration results between scans from consecutive days (Day 1 and Day 2). The left column shows the two input point clouds ($\mathcal{P}_1, \mathcal{P}_2$) along with their skeletons, middle column shows the estimated correspondences (yellow lines) between the skeletons, and the right column shows the deformed point cloud $\hat{\mathcal{P}}_1$ (in pink) overlaid on $\mathcal{P}_2$ along with registration error visualized as a heat map.

correspondences correctly both with skeletons taken 2 and 3 days apart. The higher accuracy for the maize plants is likely due to the simpler shape of the plant as compared to the tomato plants.

Secondly, we evaluate the accuracy of the estimated registration parameters by computing the error between the deformed source point cloud $\hat{\mathcal{P}}_1$ and the target point cloud $\mathcal{P}_2$. We define this registration error e_{reg} as:

$$e_{reg} \quad = \quad \frac{1}{|\hat{\mathcal{P}}_1|} \sum_{\substack{i=1 \\ j \in N(i)}}^{|\hat{\mathcal{P}}_1|} \left\| \hat{p}_1^i - p_2^j \right\|, \tag{4.15}$$

where p_2^j is the nearest point to p_1^i and $|\hat{\mathcal{P}}_1|$ is the number of points in $\hat{\mathcal{P}}_1$.

For our datasets, we obtain a mean error of 3 mm and a maximum error of 13 mm for consecutive scans, which indicates that the registration results are accurate. As a baseline comparison, we computed the average overlap error by assuming a rigid transformation between the scans and obtain an average error e_{reg} of 35 mm and maximum error of 166 mm. The large errors using a rigid transformation assumption are both due to the plant growth, and in some cases, the ICP procedure diverging completely. This indicates that a rigid transformation assumption is inadequate, and a non-rigid registration procedure is required to capture the growth and movement of the plant.

We visualize the registration error as a heat map for the two example point cloud pairs in Figure 4.7 and Figure 4.8 (bottom right of each example). The heat map is projected on $\hat{\mathcal{P}}_1$ to show how well different portions of the plant are registered. The blue regions in the heat map represent a smaller registration

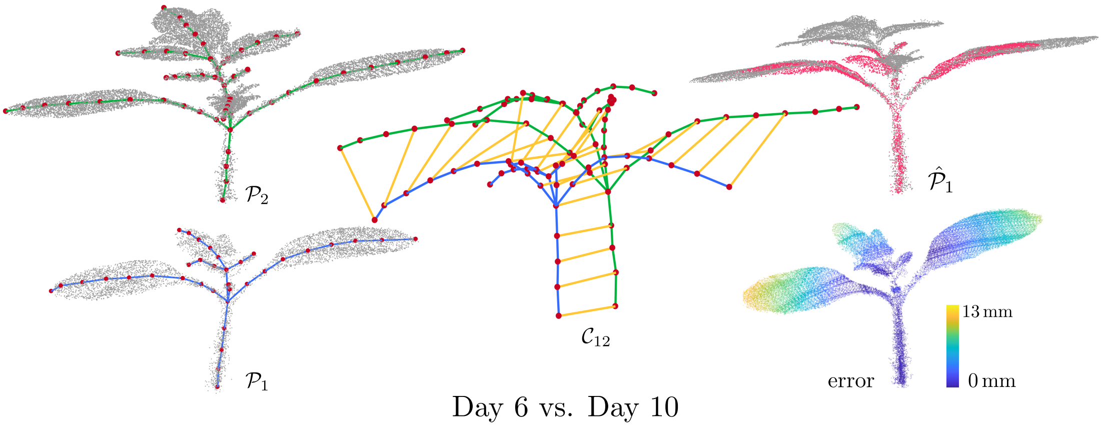


Figure 4.8: 4D registration of point clouds. Registration results between scans which are 4 days apart (Day 6 and and Day 10). The left column shows the two input point clouds ($\mathcal{P}_1, \mathcal{P}_2$) along with their skeletons, middle column shows the estimated correspondences (yellow lines) between the skeletons, and the right column shows the deformed point cloud $\hat{\mathcal{P}}_1$ (in pink) overlaid on $\mathcal{P}_2$ along with registration error visualized as a heat map.

error, whereas the yellow regions indicate large errors. Most of the regions are blue, indicating successful registration. However, we notice that the errors are usually high towards the outer sections of the leaves, which are farther away from the skeleton curve. This effect is to be expected as the skeleton curves do not capture this area well.

4.2.4 Temporal Tracking of Phenotypic Traits

In this experiment, we show that the spatial-temporal registration results computed by our approach allow us to compute several phenotypic traits and track them temporally. We compute the area l_a and length l_l for leave instances and the diameter s_d and length s_l of the stem for each point cloud in the temporal sequence and associate it over time using the data associations estimated by our approach during the registration process. The tracking results for the three sample plants from the maize and tomato datasets are visualized in Figure 4.9. The first two columns in Figure 4.9, track the leaf area and leaf length over time. Different shades of blue and green in these plots represent individual leaf instances. Besides, we can also detect specific events, which mark a topological change in the structure of the plant, such as the appearance of a new leaf. These events can be recognized from the leaf area or leaf length plots in Figure 4.9 whenever a new line rises from the zero level. In the rightmost column in Figure 4.9, we see that stem length and diameter for both the datasets increase considerably over the data acquisition period. Such phenotypic information can also be used to compute the BBCH scale [51] of the plant, which is a growth stage scale and provides valuable information to the agronomists.

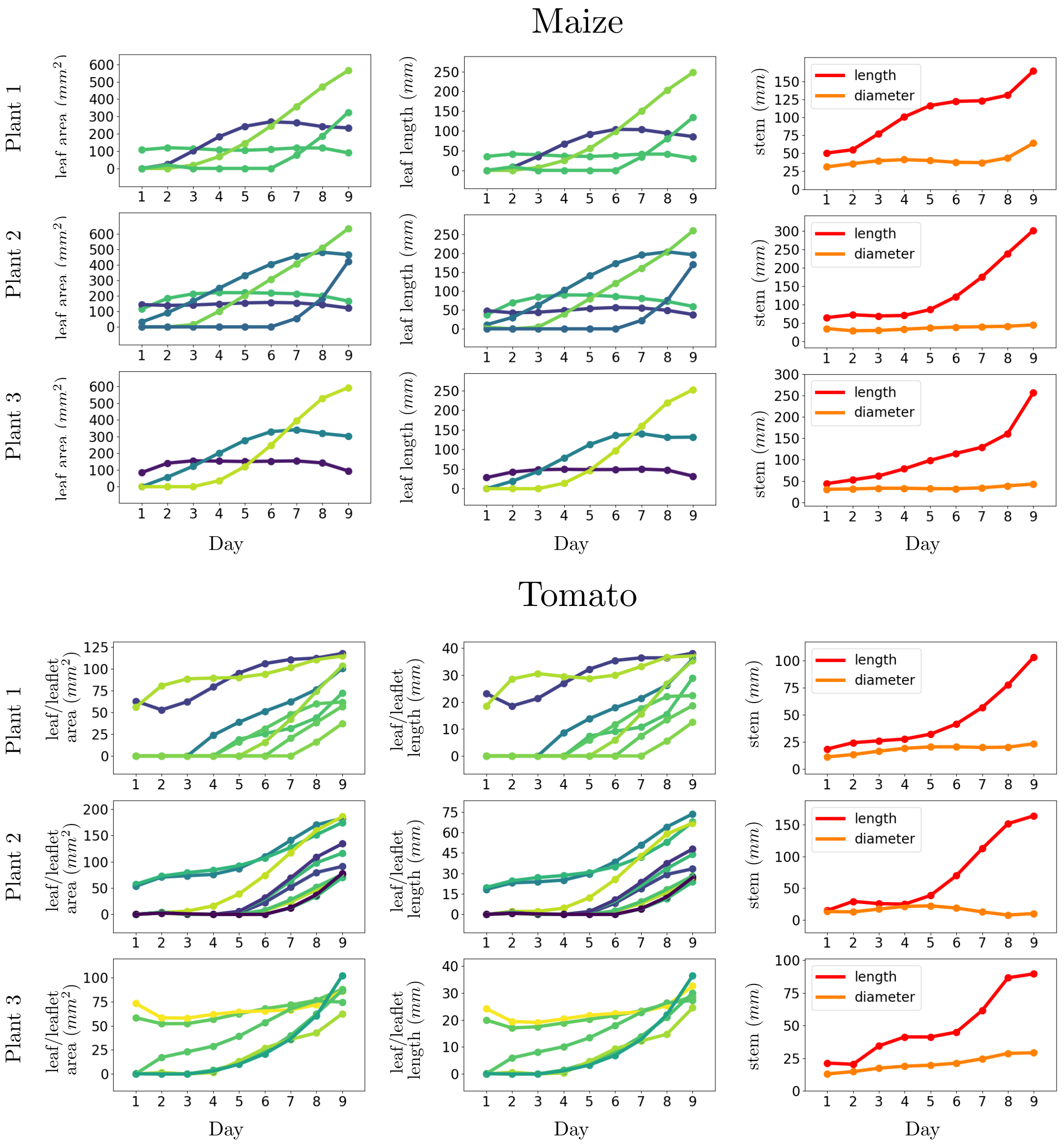


Figure 4.9: Tracking phenotypic traits for individual organs of the plant. Our registration procedures allows us to track the growth of the stem and different leave lengths over time and detect topological events such as the emergence of new leaves.

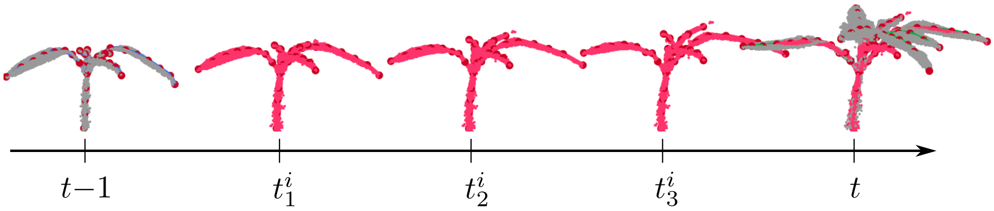


Figure 4.10: Interpolation of point clouds at intermediate time intervals. Point clouds (gray) at time $t-1$ and t come from actual scan measurements, whereas the points clouds (pink) at time instants t_1^i, t_2^i, t_3^i visualize the three interpolated scans. We see that at time instants $t-1$ and t, the interpolated point clouds in gray overlap entirely with the actual point clouds in pink as they correspond to the original measurements.

4.2.5 Temporal Interpolation of Point Clouds

In the last experiment, we illustrate that the registration results can be used to interpolate the point clouds at intermediate points in time, i.e., in between the actual acquisition times of the scans. The ability to interpolate is useful for analyzing the properties of the plants even when actual measurements are not available. It allows us to predict both the motion and growth at intermediate time intervals. We visualize the interpolated point cloud at three time instances t_1^i, t_2^i, t_3^i between the two scans in top of Figure 4.10. This allows us to animate a time-lapse view of the plants. The pink point clouds represent the interpolated scans and overlap well with the point cloud (gray) at time t, indicating that the interpolation is reasonable. As the interpolation procedure does not actually model the movement or the plant's growth, the result of the interpolation may differ from the actual plant at those instances. In order to evaluate the interpolation step, we take the scans on day $t-1$ and day $t+1$, then interpolate the point cloud at day t and compare against the actual point cloud on day t. We compute the registration error as described in Equation (4.15) and obtain a mean e_{reg} of 4 mm, suggesting that our interpolation is a reasonable approximation of the real plant growth.

4.3 Related Work

Over the last decade, automated phenotyping has emerged an important topic within the precision agriculture community [39]. It also has attracted the attention of the robotics as well as the agricultural research community. Various techniques for automated phenotyping have been developed, analyzing the sensor data captured at different spatial and temporal resolutions. The choices of the techniques often depend on the sensor setup and the frequency of measurements that the particular application allows for.

At a coarse spatial resolution level, approaches such as those by Carlone *et* *al.* [18], Dong *et al.* [34], and Lottes [73] *et al.* aim at obtaining plant traits over the entire field using image data captured from UAVs or ground robots. This is similar to the scenario that we addressed in Chapter 2. Such approaches achieve a typical spatial resolution in the range of $10\,\mathrm{cm}^2$ to $1\,\mathrm{m}^2$, which is already sufficient for a coarse phenotypic analysis. This performance can further be improved by equipping the UAVs with higher resolution cameras that are often only feasible with larger UAVs. Another drawback with such approaches is that they are limited to top-down views, making it challenging to capture traits other than those visible on the canopy of the plants. On the other hand, the advantage of these approaches is that they also allow for high-frequency measurements providing a dense temporal resolution, as it is a reasonably small effort to fly UAVs over the field to capture new data regularly.

Instead, in this chapter, we aim to exploit the high-resolution plant point clouds acquired from close range using a LiDAR scanner, which has become more prevalent in the precision agriculture community. Over the last decade, several works such as those by Klose *et al.* [57], Alenya *et al.* [5], and Paulus *et al.* [91] have looked at using this high-resolution 3D data with the goal of computing phenotypic traits with high fidelity. Li *et al.* [69] and Paproki *et al.* [89] extend the analysis over a time-series of point cloud data to detect topological events such as branching, decay and track the growth of different organs. While these works emphasize obtaining phenotypic traits at an organ level, our main objective in this chapter has been to develop basic techniques for matching and registering temporally separated scans of individual plants using the whole point cloud data. Our work essentially brings the temporal plant data into a common coordinate frame and allowing for other phenotypic applications that rely on temporal data association to be developed on top of it.

As raw point clouds are often cumbersome to use and lack any inherent structure, a common approach is to extract a skeleton structure that captures the topology of the object in a compact manner. Several techniques in the past have attempted to leverage the topological structure of the object to drive the registration process, primarily in the field of human motion tracking [41, 50, 106]. A large corpus of literature exists for extracting skeletons from 3D models, which are then used for different applications such as animation, surface reconstruction, etc. Huang *et al.* [52] and Tagliasacchi *et al.* [117] propose approaches to extract curve skeletons from unorganized point clouds, which can be used as an input to our approach. A detailed state-of-the-art review for extracting skeletons of 3D objects is given by Tagliasacchi *et al.* [116]. In contrast to these approaches, in our work, we exploited both supervised and unsupervised machine learning techniques to compute the skeleton curve for the plant point clouds. In this process, we classify the plant into stem and leaf points, cluster them together as individual organs and use this semantic information for computing the skeleton structure effectively.

Exploiting semantic information of the plant for extracting the skeleton structure is quite helpful. While a large corpus of literature exists for classification in 2D images, the number of approaches that operate on 3D point clouds is relatively small. Paulus *et al.* [90] propose an SVM-based classifier that relies on a surface histogram to classify each point in a 3D point cloud as leaf or stem. The recent approach by Zermas *et al.* [136] uses an iterative algorithm called randomly intercepted node to tackle the same problem. Sodhi *et al.* [110] use 2D images to extract 3D phytomers, namely fragments of the stem attached to a leaf for leaves detection. Shi *et al.* [107] propose a multi-view deep learning approach inspired by Su *et al.* [113] to address the organ segmentation problem, while Zermas *et al.* [135] uses a skeletonization approach to segment leaf instances. More recently, deep neural-networks such as PointNet [95] and PointNet++ [96] have been proposed operating directly on 3D raw point clouds to produce the semantic segmentation. These methods have shown impressive results. However, they typically require large datasets with labels for the training process, which is often difficult to get in the plant domain. In our work, we go for a more straightforward approach as our primary goal of segmenting the point cloud is to use this information for extracting the skeleton structure. We build upon the work of Paulus *et al.* [90] and additionally group the leaf points with an unsupervised clustering algorithm to extract leaf instances. In this way, we achieve an organ segmentation exploiting labeled data for the leaves. We use this in turn as the basis for our registration approach across plant point clouds.

Registering point clouds is a common problem in a lot of disciplines, and multiple techniques have been proposed for laser-based or RGB-D-based mapping systems [12, 84, 137]. These techniques typically work under the assumption the objects being registered only undergo rigid motion. They have also been extended by relaxing the rigidity assumption, and several non-rigid registration techniques such as those by Bouaziz *et al.* [15], Sorkine *et al.* [111], and Sumner *et al.* [114] aim at capturing the deformation in the object. Other approaches such as [47, 55, 87] aim at reconstructing scenes in an online fashion either in the presence of dynamic objects or deformations. Such approaches typically operate on scans captured at a high frame rate (10-30 Hz) and thereby deal with rather small deformations in-between consecutive scans. This assumption is violated for the point cloud data considered here. In our application, the plants are usually scanned at a comparably low frequency (once per day), thereby showing larger growth and deformations between consecutive scans. In addition, the problem becomes even more complicated if the object changes its appearance and topology over time. Zheng *et al.* [139] proposes an interesting approach to register 3D temporal point clouds of objects by exploiting skeleton structures of different objects. In their applications, the objects have roughly the same size and retain their topology over the entire sequence. Our work in this chapter builds upon their ideas and extends them to deal with the plant's growth as well as its changing topology. We achieve this by introducing an improved data association approach, which accounts for the topology and the semantics of the plant. These data associations are then used to register temporally separated 3D plant point clouds in an iterative non-rigid registration scheme.

The problem at hand, as well as the technique used in our approach to address plant registration, shows some relation to techniques from the SLAM community [7, 8, 112]. This similarity starts with the iterative scan alignment procedures such as ICP [12, 47], although we build upon a skeleton representation and not the point cloud itself. Furthermore, we build up data associations over time between skeleton nodes using a hidden Markov model formulation. For the optimization, least-squares approaches related to graph-based SLAM [46] are used, including robust kernel functions. Unlike typical SLAM systems, however, we allow for multiple affine transformations between plant parts and extend the rigid-body transformations often found in the SLAM literature.

4.4 Conclusion

This chapter presented a novel approach for spatio-temporal registration of 3D point clouds of individual plants. The proposed method works for raw sensor data stemming from a range sensor such as a 3D LiDAR or a depth camera, which is processed fully automated fashion without any manual intervention. We implemented and evaluated our approach on datasets of tomato and maize plants presenting challenging situations. The experiments in the chapter show that our registration approach can be used as a basis for tracking plant traits temporally and contribute towards automated phenotyping. In sum, our approach works by first estimating the skeletal structure of the plant, also exploiting a point-wise classification approach to compute a skeleton representing the plant. This skeleton structure, along with the semantic information, is used to find reliable correspondences between parts of the plant recorded at different points in time using a novel data association approach that relies on a hidden Markov model. As our results showcase, this approach can deal with changing appearance and topology of the 3D structure of the plant. This is an essential capability to form a robust alignment of the 4D data, i.e., of 3D geometry and time. Given the data associations, we explicitly model the deformation and growth of the plant over time using multiple affine transformations associated with the individual nodes of the skeleton structure. In this way, individual parts of the plant are trans-formed using different affine transformation modeling the different growth along the plant. The parameters for these transformations are estimated using a robust least-squares approach, including regularizations. Given the resulting parameters, we can align 3D scans taken at different points in time and transform them according to the growth. This, in turn, allows us to estimate basic phenotypic parameters such as leaf area, leaf length, as well as stem diameter or length and track their development over time in an automated fashion.

Chapter 5

Adaptive Robust Kernels for Registration and State Estimation

In the previous chapters, we have developed different spatio-temporal registration techniques for applications in agricultural robotics. A common challenge that each of this approach faces is the presence of outliers, which arise typically from the measurements and the data association process. Outliers may result from incorrect matches between features computed in UAV images due to lack of descriptor specificity (Chapter 2), or ambiguous correspondences made between features computed in UGV images to the features in the reference maps generated from UAV images due to aliasing (Chapter 3), or wrong correspondences made between skeleton nodes of temporally separated plant point cloud data (Chapter 4). Such types of errors happen in a large range of perception problems, including the ones just mentioned above. Thus, the ability to deal with outliers is of high relevance for a large set of state estimation process.

In order to address these challenges, we employed different strategies in our registration pipelines to make them robust to outliers. For example, in Chapter 2, we use a RANSAC approach and other heuristics to identify and remove the outliers from the set of all candidate matches between corresponding images. Instead, in Chapter 3, we deal with the ambiguous correspondences between the UGV detections and the aerial reference map by both exploiting extra information about the features in terms of semantics as well as choosing a particle filter-based framework to deal with the multiple correspondence hypotheses. For the 4D registration approach in Chapter 4, we used a hidden Markov model framework, which tends to choose a set of correspondences that are mutually consistent with each other. In addition, we use a robust loss function to minimize the effect of any outlier in the registration process. In this chapter, we build upon this idea of using robust kernels and discuss an automated technique for robustifying general registration and state estimation tasks.

Registration and state estimation form important building blocks not just in applications related we have seen in the previous chapters, but are also used in a variety of different components in robotics, including simultaneous localization and mapping. A large number of state estimation solvers perform some form of non-linear least squares minimization. Prominent examples are the optimization of SLAM graphs consisting of landmarks and poses, the ICP algorithm, robot localization, visual odometry, or bundle adjustment, which all seek to find the minimum of some error function. As soon as real-world data is involved, outliers will occur in the data. A common source of such outliers stems from data association mistakes, for example, when matching features. Robust kernel functions are used to down-weight the effect of gross errors and avoid that just a few such outliers have strong effects on the final solution. Several robust kernels have been developed to deal with outliers arising in different situations. Prominent examples include the Huber, Cauchy, Geman-McClure, or Welsch functions, which can be used to obtain a robustified estimator [138].

The optimal choice of the best kernel for a given problem is not straightforward. As the robust kernels define the distribution from which the outliers are generated, their choice is problem-specific. In practice, the choice of the kernel is often made in a trial and error manner, as we have only limited prior knowledge about the outlier process in most situations. For some approaches such as bundle adjustment, today's implementations even vary the kernel between iterations or pair them with outlier rejection heuristics. Moreover, for several robotics applications such as SLAM, the outlier distribution itself changes continuously depending on the structure of the environment, dynamic objects in the scene, and other environmental factors like lighting. This often means that a fixed robust kernel chosen a-priori cannot deal effectively with all situations.

In this chapter, we aim at circumventing the trial and error process for choosing a kernel and at exploring the automatic adaptation of kernels to the outliers online. To achieve this, we use a family of robust loss functions proposed by Barron [9], which generalizes several popular robust kernels such as Huber, Cauchy, Geman-McClure, Welsch, etc. The key idea is to dynamically tune this generalized loss function automatically based on the current residual distribution so that one can blend between such robust kernels and make the choice a part of the optimization problems.

The main contribution of this chapter is an easy-to-implement approach for dynamically adapting the robust kernels in non-linear least squares (NLS) solvers, which builds on top of the generalized formulation of Barron [9]. We achieve this by estimating a hyper-parameter for a generalized loss function, which controls

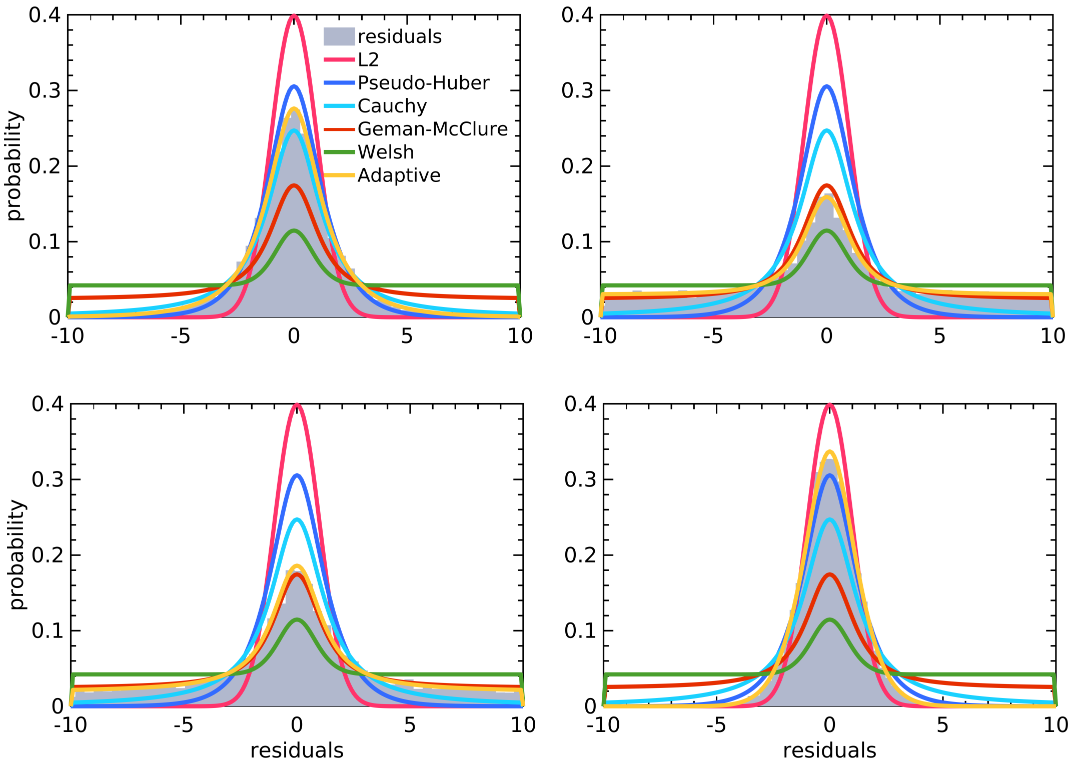


Figure 5.1: Probability densities of different robust kernels. The four plots correspond to different residual distributions (in gray) occurring during the state estimation process. Different fixed kernels and the adaptive kernel distributions are overlaid on top. The closer the kernel distribution is to the actual residual distribution, the better that particular kernel is for dealing with the outliers in that situation. We see that the adaptive robust kernel (in yellow) is able to describe the actual residual distribution in different situations better than a fixed robust kernel for all cases. As a result, it provides better robustness to different types of outliers during the state estimation process.

the shape of the robust kernel. This parameter becomes part of the estimation process, and we determine it along with the unknown parameters of the model. We extend the usable range of this parameter compared to the formulation of Barron [9]. This allows us to better deal with a larger set of outlier distributions compared to fixed kernels and to the Barron formulation. See Figure 5.1 for a visualization.

In sum, in this chapter, we develop an approach that performs robust estimation without committing to a fixed kernel beforehand or requiring any manual tuning and automatically adapts the shape of the kernel to the actual outlier distribution. We also illustrate the advantage of our approach for the plant point cloud registration task from Chapter 4, as well as two typical problems from the robotics and photogrammetry applications, namely ICP and bundle adjustment.

5.1 Least Squares with an Adaptive Robust Kernel

We present an approach that dynamically adapts robust kernels when solving NLS problems. This is done by estimating a hyper-parameter that controls the shape of the robust kernel. This parameter becomes part of the estimation process in an alternating error minimization procedure. Before explaining our approach, we first explain robust NLS estimation and generalized kernels to give a complete view in Section 5.1.1. We then present the generalized robust kernel proposed by Barron [9], which is the foundation of our work in Section 5.1.2. Our main contribution in this chapter is to extend Barron's robust kernel to deal with strong outliers typically encountered in robotics applications in Section 5.1.3. We propose a novel algorithm for using the adaptive robust kernel for solving typical non-linear least squares problems with minimal changes to existing optimization frameworks in Section 5.1.4.

5.1.1 Robust Least Squares Estimation

Several state estimation problems in robotics involve estimating unknown parameters θ of a model given noisy observations z_i with $i = 1, \ldots, N$. These problems are often framed as non-linear least squares optimization, which aims to minimize the squared loss:

$$\theta^* = \operatorname*{argmin}_{\theta} \frac{1}{2} \sum_{i=1}^{N} w_i \left\| r_i(\theta) \right\|^2 , \tag{5.1}$$

where $r_i(\theta) = f_i(\theta) - z_i$ is the residual and w_i is the weight for the i^{th} observation. The estimate θ^* is statistically optimal if the error on the observations z_i is Gaussian. In case of non-Gaussian noise, however, the estimate θ^* can be poor [53]. To reduce this impact of outliers, sub-quadratic losses are typically applied. The main idea of a robust loss is to downweight large residuals that are assumed to be caused from outliers such that their influence on the solution is reduced. This is achieved by optimizing:

$$\theta^* = \operatorname*{argmin}_{\theta} \sum_{i=1}^{N} \rho(r_i(\theta)), \tag{5.2}$$

where $\rho(r)$ is a sub-quadratic loss also called the robust loss or kernel. Several robust kernels such as Huber, Cauchy, and others have been proposed to deal with different outlier distributions [138]. A summary of the popular robust kernels can be found in the work by MacTavish *et al.* [76].

The optimization problem in Equation (5.2) can be solved using the iteratively reweighted least squares (IRLS) approach [138], which solves a sequence of weighted least squares problems. We can see the relation between the least squares optimization in Equation (5.1) and robust loss optimization in Equation (5.2) by comparing the respective gradients, which go to zero at the optimum:

$$\frac{1}{2}\frac{\partial(w_i r_i^2(\theta))}{\partial\theta} = w_i r_i(\theta)\frac{\partial r_i(\theta)}{\partial\theta} \tag{5.3}$$

$$\frac{\partial(\rho(r_i(\theta)))}{\partial\theta} = \rho'(r_i(\theta))\frac{\partial r_i(\theta)}{\partial\theta}. \tag{5.4}$$

By setting the weight,

$$w_i = \frac{1}{r_i(\theta)}\rho'(r_i(\theta)), \tag{5.5}$$

we can solve the robust loss optimization problem by using the existing techniques for weighted least-squares. This scheme allows standard solvers using Gauss-Newton and Levenberg-Marquardt algorithms to optimize for robust losses and is implemented in popular optimization frameworks such as Ceres [3], g2o [61], GTSAM [31], and iSAM [56].

5.1.2 Adaptive Robust Kernel

We build on the work by Barron [9] who proposes a single robust kernel that generalizes for several popular kernels such as Huber/L1-L2, Cauchy, Geman-McClure, Welsch. The generalized kernel ρ by Barron is given by:

$$\rho(r, \alpha, c) = \frac{|\alpha - 2|}{\alpha}\left(\left(\frac{(r/c)^2}{|\alpha - 2|} + 1\right)^{\alpha/2} - 1\right), \tag{5.6}$$

where α is a real-valued parameter that controls the shape of the kernel and $c > 0$ is the scale parameter that determines the size of quadratic loss region around $r = 0$. Adjusting the parameter α essentially allows us to realize different robust kernels. Some special cases are squared/L2 loss ($\alpha = 2$), Huber/L1-L2 ($\alpha = 1$), Cauchy ($\alpha = 0$), Geman-McClure ($\alpha = -2$), and Welsch ($\alpha = -\infty$).

The general loss function $\rho(r, \alpha, c)$ and the corresponding weights curve $w(r, \alpha, c)$ are illustrated in Figure 5.2 for several values of α. The shape of the weights curve provides an insight into the influence that a residual has on the solution while minimizing the robust loss function in Equation (5.2). For example, for $\alpha = 2$, the weights for all residuals are one, meaning that all residuals are treated the same as done for non-robust least squares. On the other extreme, for $\alpha = -\infty$, the

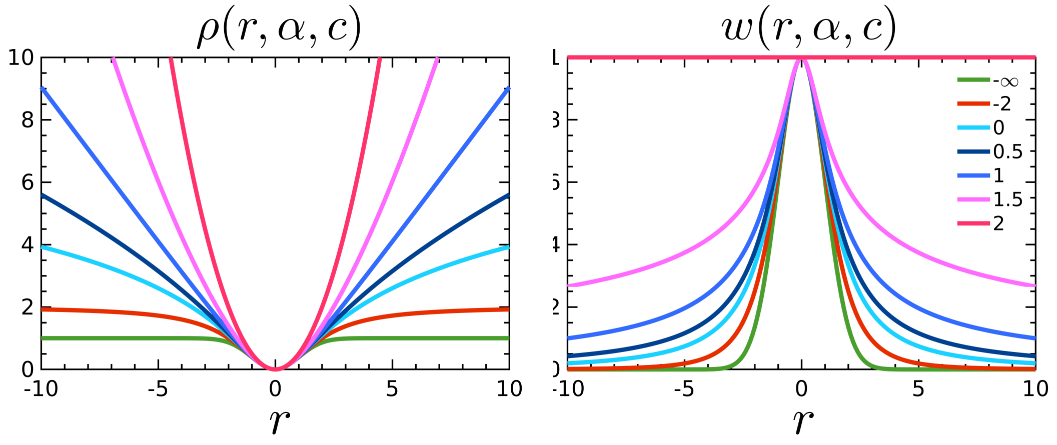


Figure 5.2: Left: General robust loss $\rho(r, \alpha, c)$ takes different shapes depending on the value of α. Right: Corresponding weights for kernels with different α values. A smaller α corresponds to a larger down-weighting of the residuals.

weights for all the all residuals greater than $3c$ are close to zero, basically resulting in these large residuals that are potential outliers not affecting the solution θ^*.

With this generalized robust loss, we can interpolate between a range of robust kernels simply by tuning α. To automatically determine the best kernel shape through the parameter α, we treat α as an additional unknown parameter while minimizing the generalized loss:

$$(\theta^*, \alpha^*) \quad = \quad \underset{(\theta,\alpha)}{\operatorname{argmin}} \sum_{i=1}^{N} \rho(r_i(\theta), \alpha). \tag{5.7}$$

However, the optimization problem in Equation (5.7) can be trivially minimized by choosing an α that weighs down all residuals to near-zero values without affecting the model parameters θ, essentially treating all data points as outliers. Barron [9] avoids this by constructing a probability distribution based on the generalized loss function $\rho(r, \alpha, c)$ as

$$P(r, \alpha, c) \quad = \quad \frac{1}{cZ(\alpha)} e^{-\rho(r,\alpha,c)}, \tag{5.8}$$

$$Z(\alpha) \quad = \quad \int_{-\infty}^{\infty} e^{-\rho(r,\alpha,1)} \, dr, \tag{5.9}$$

where $Z(\alpha)$ is a normalization term, also called partition function, which defines an adaptive general loss as the negative log-likelihood of Equation (5.8),

$$\rho_a(r, \alpha, c) \quad = \quad -\log P(r, \alpha, c) \tag{5.10}$$

$$\quad = \quad \rho(r, \alpha, c) + \log cZ(\alpha). \tag{5.11}$$

The adaptive loss $\rho_a(\cdot)$ is simply the general loss $\rho(\cdot)$ shifted by the log partition. This shift introduces an interesting trade-off. A lower cost for increasing the

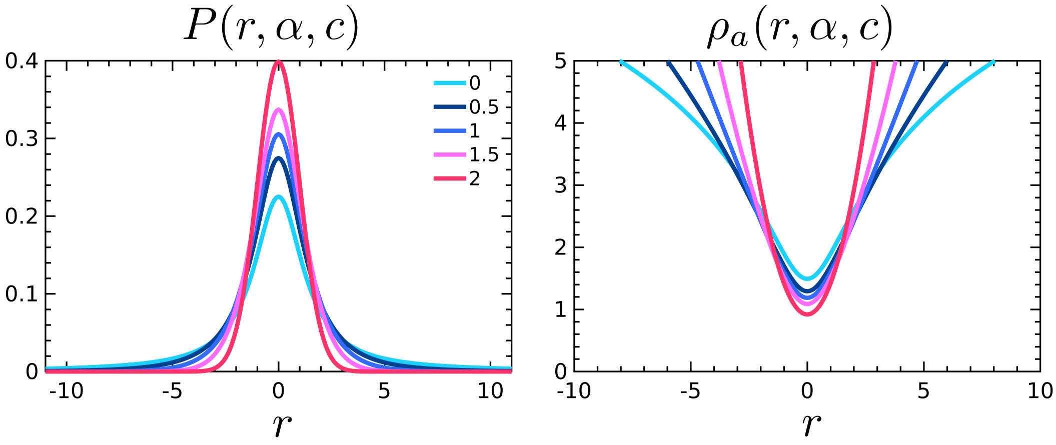


Figure 5.3: Left: Probability distribution $P(r, \alpha, c)$ of generalized robust loss function for different values of α. Right: Adaptive robust loss $\rho_a(r, \alpha, c)$ obtained as the negative log-likelihood of $P(r, \alpha, c)$. This adaptive loss enables automatic tuning of $\alpha \in [0, 2]$.

set of outliers comes with a penalty for the inliers and vice versa. This trade-off forces the optimization in Equation (5.7) to choose a suitable value for α instead of trivially ignoring all residuals by turning every data point into an outlier. The probability distribution $P(r, \alpha, c)$ and the adaptive loss function are plotted in Figure 5.3 for visualization.

5.1.3 Truncated Robust Kernel

The probability distribution $P(r, \alpha, c)$ is only defined for $\alpha \geq 0$, as the integral within the partition function $Z(\alpha)$ is unbounded for $\alpha < 0$. This means that values for $\alpha < 0$ cannot be achieved while minimizing the adaptive loss $\rho_a(\cdot)$ in Equation (5.10). This limits the range of kernels that can be dynamically adapted. As we can see in Figure 5.2, the smaller the parameter α is, the stronger is the down-weighting of outliers. Such a behavior is often desired in situations where a large number of outliers are present in the data.

In this chapter, we propose an extension to the adaptive loss in Equation (5.10), which allows the parameter α to be dynamically adapted for a larger range of values, especially those below zero. We achieve this by limiting the value of the partition function $Z(\alpha)$ to bounded values. To re-gain the kernels corresponding to the negative range of α with the adaptive loss function, we compute an approximate partition function $\tilde{Z}(\alpha)$ as

$$\tilde{Z}(\alpha) \quad = \quad \int_{-\tau}^{\tau} e^{-\rho(r,\alpha,1)} dr, \tag{5.12}$$

where τ is the truncation limit for approximating the integral. This results in a finite partition $\tilde{Z}(\alpha)$ for all α as the integral is computed within the limits $[-\tau, \tau]$.

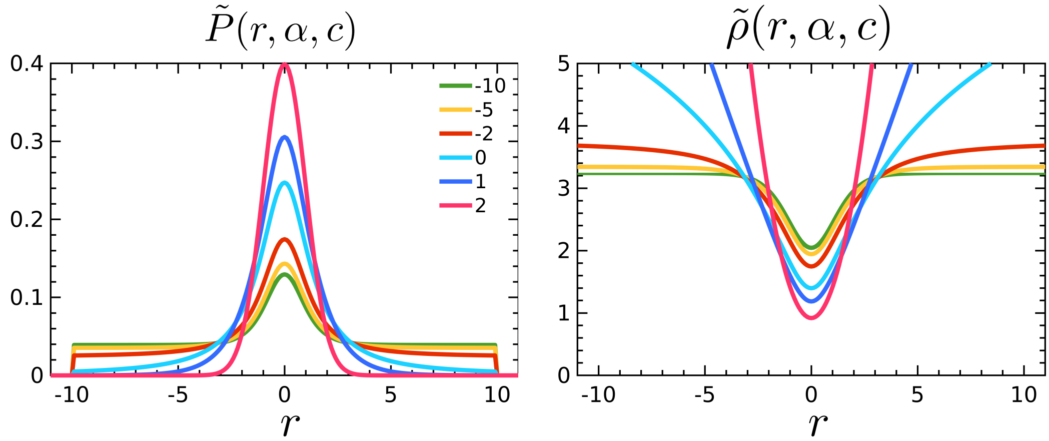


Figure 5.4: Left: Modified probability distribution $\tilde{P}(r, \alpha, c)$ obtained by truncating $P(r, \alpha, c)$ at $|r| < \tau$. Right: The truncated robust loss $\tilde{\rho}_a(r, \alpha, c)$ allows the automatic tuning of α in its complete range, including $\alpha < 0$.

We use this to define our truncated loss function $\tilde{\rho}_a$ as

$$\tilde{\rho}_a(r, \alpha, c) = \rho(r, \alpha, c) + \log c\tilde{Z}(\alpha). \tag{5.13}$$

The truncated probability distribution $\tilde{P}(r, \alpha, c)$ and the corresponding truncated loss $\tilde{\rho}_a(r, \alpha, c)$ are shown in Figure 5.4. Since the truncated loss is defined for all values of α including $\alpha < 0$, we can adapt α in its entire range during the optimization procedure. We discuss the effect of this truncation of the loss function below in Section 5.1.5.

5.1.4 Optimization of α via Alternating Minimization

We propose to solve the joint optimization problem over θ and α in an iterative manner using an alternating minimization procedure. The joint loss is defined as

$$(\theta^*, \alpha^*) \quad = \quad \underset{(\theta,\alpha)}{\operatorname{argmin}} \sum_{i=1}^{N} \tilde{\rho}_a(r_i(\theta), \alpha). \tag{5.14}$$

The procedure alternates between two steps. In the first step, the maximum likelihood value for α is computed, and in the second step, the optimal parameters for the model given the α from the previous step is computed. This approach can be seen as a variation of a coordinate descent approach. By solving the joint optimization in this manner, we decouple the estimation of the robust kernel parameter α from the original optimization problem. This allows to us solve for the model parameters θ in the same way as before α was introduced, and leverage existing solvers such as Ceres [3], g2o [61], GTSAM [31], and iSAM [56] to perform the optimization. This means that we can reuse our existing code for optimization and only need to extend it by the alternating minimization.

Algorithm 2 Optimization with adaptive robust kernel

```
Initialize $\theta^0 = \theta, \alpha^0 = 2, c$
while !converged do
    Step 1: Minimize for $\alpha$
    $\alpha^t = \text{argmin}_\alpha - \sum_{i=1}^{N} \log P(r_i(\theta^{t-1}), \alpha^{t-1}, c)$
    Step 2: Minimize robust loss using IRLS
    $\theta^t = \text{argmin}_\theta \sum_{i=1}^{N} \rho(r_i(\theta), \alpha^t, c),$
```

We estimate the parameters α in the first step by minimizing the negative log-likelihood of observing the current residuals,

$$
\begin{aligned}
L(\alpha) &= -\sum_{i=1}^{N} \log P(r_i(\theta), \alpha, c) && (5.15)\\
&= \sum_{i=1}^{N} \log c\tilde{Z}(\alpha) + \rho_a(r_i(\theta), \alpha, c), && (5.16)
\end{aligned}
$$

i.e.,

$$
\alpha^* = \underset{\alpha}{\text{argmin}}\, L(\alpha). \qquad (5.17)
$$

A solution to Equation (5.17) can be obtained by setting its first derivative $\frac{dL(\alpha)}{d\alpha} = 0$. Since its not possible to derive the partition function $\tilde{Z}(\alpha)$ analytically, we settle for a numerical solution. As α is a scalar value, $L(\alpha)$ can be minimized comparably easily by performing a 1-D grid search for $\alpha \in [\alpha_{min}, 2]$.

In terms of a practical implementation, we chose lower bound $\alpha_{min} = -10$ as its difference to the corresponding weights for $\alpha = -\infty$ for large residuals ($|r| > \tau$) is small and in most practical problems negligible. The difference in weights for $\alpha = -10$ and $\alpha = -\infty$ is less than 10^{-5} for a residual $|r| = \tau$. The maximum value for α is set to 2 as this corresponds to the standard least squares problem. The scale c of the robust loss is fixed beforehand and not adapted during the optimization. This value for c is usually fixed based on the measurement noise for an inlier observation z and the accuracy of the initial solution. To be computationally efficient, we pre-compute $\tilde{Z}(\alpha)$ as a lookup table for values $\alpha \in [\alpha_{min}, 2]$ with a resolution of 0.1 and use the lookup table during optimization. This leads to the overall minimization approach shown in Algorithm 2.

5.1.5 Effect of using the Truncated Loss

The truncated loss approximation affects the first step of the optimization procedure in Algorithm 2, i.e., determining the parameter α. By using our truncated loss, we are implicitly assuming that no outliers have a residual $|r| > \tau$ during the first step. If we choose a large enough value for τ, the error that we introduce affects situations with large outliers only and therefore results in small values of α. The effect of small α values such as $\alpha = -10$ vs. $\alpha = -\infty$ on the optimization, however, is negligible as the outliers will be down-weighted to basically zero. We observe in our experiments that by setting choosing $\tau = 10c$, we are able to deal with all of the outlier distributions in practice for ICP, SLAM, BA applications.

5.2 Experimental Evaluation

In the experimental section, we evaluate the performance of our adaptive robust kernel approach on the spatio-temporal point cloud registration application from the previous chapter and two typical applications in robotics, namely the Iterative Closest Point (ICP) algorithm and bundle adjustment (BA). The experiments are designed to evaluate the effectiveness of our approach in the presence of strong outliers and showcase its applicability for common NLS problems. We compare the performance of our approach against hand-crafted outlier rejection mechanisms using fixed robust kernels on multiple datasets.

5.2.1 Application to Registration of Plant Point Clouds

In the first experiment, we show the advantages of using the adaptive robust kernel for the spatio-temporal registration of plant point clouds proposed in the previous chapter. As described in Chapter 4, we compute the registration parameters between the temporally separated plant skeleton nodes using a robust kernel to deal with the wrong data associations computed during the correspondence estimation step. In the previous chapter, we used a fixed Cauchy's robust kernel to perform this step. We replace the fixed robust kernel used in the registration procedure with our truncated adaptive robust kernel proposed in this chapter. We then repeat the registration experiment performed in Section 4.2.3, where we register all consecutive scans from the maize and the tomato time-series datasets. We then evaluate the accuracy of the registration process as defined in Equation (4.15) in Chapter 4, which computes the error between the deformed source cloud and the target cloud. Using the fixed Cauchy's robust kernel, we obtain a mean error of 3 mm and a maximum error of 13 mm for consecutive scans. The accuracy results improve using our truncated adaptive robust kernel with an average mean error of 2.5 mm and a maximum error of 7.2 mm. We see a considerable improvement in the worst-case scenario from 13 mm to 7.2 mm. These are precisely the cases where outliers have a large effect on the registration, and our approach deals with it better than a fixed robust kernel. We obtained similar results by using the original adaptive robust kernel formulation by Barron [9] giving a mean registration error of 2.8 mm and a maximum error of 8.6 mm.

To further investigate the advantages of using our adaptive robust kernel, we simulated addition correspondence errors while matching the plant point clouds. We then repeat the previous experiment of registering plant point clouds from consecutive days with the additional correspondence errors that have been artificially added. We varied the percentage of additional correspondence errors from 10% to 50% with an increment of 5% each time. We categorize the registration between the point clouds to be successful if the registration error $e_{reg} < 20\,\mathrm{mm}$. With this experimental setup, we observe that registration with fixed Cauchy's kernel was successful up to 15% additional outliers for the tomato dataset and 25% for the maize dataset. Using the original Barron's adaptive kernel [9], the registration succeeded with 25% additional outliers for the tomato dataset and 40% for the maize datasets. Finally, using our approach with the modified truncated loss, the registration was successful up to 45% additional outliers for the tomato dataset and 50% for the maize datasets. This is a considerable gain in terms of the number of outliers that our approach can handle. Overall, we obtain better registration results both for maize and tomato datasets, especially in challenging outlier situations. These results suggest that our approach provides better registration performance when the percentage of outliers is high in the data, which is necessary in case of errors during the data association step.

5.2.2 Application to Iterative Closest Point

In this experiment, we show the advantages of our approach for LiDAR-based registration in the form of ICP. We integrated our truncated adaptive robust kernel into an existing SLAM system, called surfel-based mapping (SuMa) [11], which performs point-to-plane projective ICP for 3D LiDAR scans. The ICP registration is performed in a frame-to-frame fashion on consecutive scans. We compare the performance of our approach against two fixed robust kernels, i.e., Huber and Geman-McClure, as well as to a hand-crafted outlier rejection scheme as used in the original implementation of SuMa [11]. This hand-crafted scheme combines a Huber kernel with an additional outlier rejection step that removes all correspondences, which have a distance of more than $2\,m$ or an angular difference greater than 30° between the estimated normals of observations and the corresponding normals of the surfels. Finally, we compare it to the original adaptive kernel as proposed by Barron [9].

We evaluate all these approaches on the odometry datasets of the KITTI vision benchmark [42] and summarize the results in Table 5.1. The best performance in terms of relative translation error for each sequence is highlighted in bold. We observe that our proposed approach, which does *not require an outlier rejection step* at all, performs better or is on-par with fixed kernel plus outlier re-

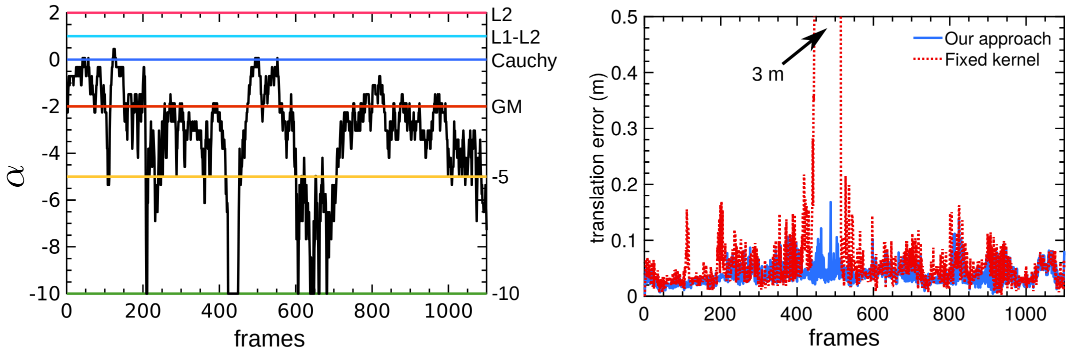


Figure 5.5: Adaptive kernel performance on KITTI-01 sequence. Left: plot showing α values estimated at each frame based on the residual distribution for KITTI 01 sequence. Right: translation error (in meters) for our approach and fixed kernel ICP. The lower α values (stronger outlier down-weighting) are observed while matching scans consisting of outliers arising from dynamic objects in the scene. The blue and dotted red lines show the translational error using our approach and fixed kernel, respectively. We observe that around frame 400, the fixed kernel ICP shows a large translation error of 3 m, indicating that the ICP process diverged.

jection scheme for many of the sequences. At the same time, using only the fixed kernel without the outlier rejection step fails for some of the sequences. In particular, for Sequence 04, both the fixed kernel using Huber and the hand-crafted outlier rejection scheme fail, whereas our approach performs the best on this sequence. Our approach performs slightly better with respect to the adaptive kernel by Barron [9], with the biggest gain in Sequence 01, which requires negative values of α to deal with the outliers coming from dynamic objects in the scene. On average over all the sequences, our approach provides the best accuracy in terms of a relative translation error. Here, we note that our approach is not the best in terms of relative rotational error but is around the 1% mark, which is on-par with other approaches. These results are promising as by using our adaptive robust kernel, we do not need any hand-crafted outlier rejection mechanism, which in practice requires manual tuning for new data, different sensor configurations, or different tasks.

As a qualitative evaluation, we illustrate the advantage of using the adaptive robust kernel for a challenging dataset (Sequence 01), which contains several moving cars moving with the vehicle itself along the highway with little additional geometric structures. In Figure 5.5 (left), we plot the values of α for each iteration while mapping the sequence. We observe that α adapts to smaller and more negative values whenever there are more outliers, which arise mainly from moving vehicles in the scan. This effect can be seen in Figure 5.5 (right) where the translation error for the fixed kernel increases as it cannot handle the outlier situation well. At the same time, the error remains small for our adaptive kernel.

The two 3D scenes show the registrations at the same point in time, once

Table 5.1: Results on KITTI odometry datasets [Relative rot. error in degrees per 100 m / relative trans. error in %]

	Sequence											
Approach	00	01	02	03	04	05	06	07	08	09	10	Average
Our Approach	1.5/2.8	1.3/**3.8**	0.91/**1.8**	1.5/1.9	0.81/**0.95**	0.97/1.7	0.51/1.1	2.1/2.6	1.3/2.7	0.80/**1.4**	1.3/**1.7**	1.18/**2.03**
Adaptive Kernel (Barron [9])	1.6/3.0	1.2/6.7	0.93/1.9	1.4/1.8	0.82/1.0	0.97/1.8	0.51/1.1	2.2/2.7	1.3/2.8	0.88/**1.4**	1.2/**1.7**	1.19/2.35
Fixed Kernel (Huber)	0.93/**2.1**	1.2/4.5	0.79/2.3	0.7/**1.4**	1.1/49	0.79/1.5	0.64/**0.95**	1.2/**1.8**	0.96/**2.5**	0.78/1.9	0.97/1.8	0.92/6.34
Fixed Kernel (Geman-McClure)	1.8/3.4	1.3/**3.8**	1.0/1.9	1.5/2.0	0.88/1.2	0.98/1.7	0.62/1.3	2.6/3.0	1.5/3.0	1.0/1.6	1.3/1.9	1.32/2.27
Hand-Crafted Outlier Rejection [11]	0.9/**2.1**	1.2/4.0	0.8/2.3	0.7/**1.4**	1.1/11.9	0.8/1.5	0.6/1.0	1.2/**1.8**	1.0/**2.5**	0.8/1.9	1.0/1.8	0.9/2.90

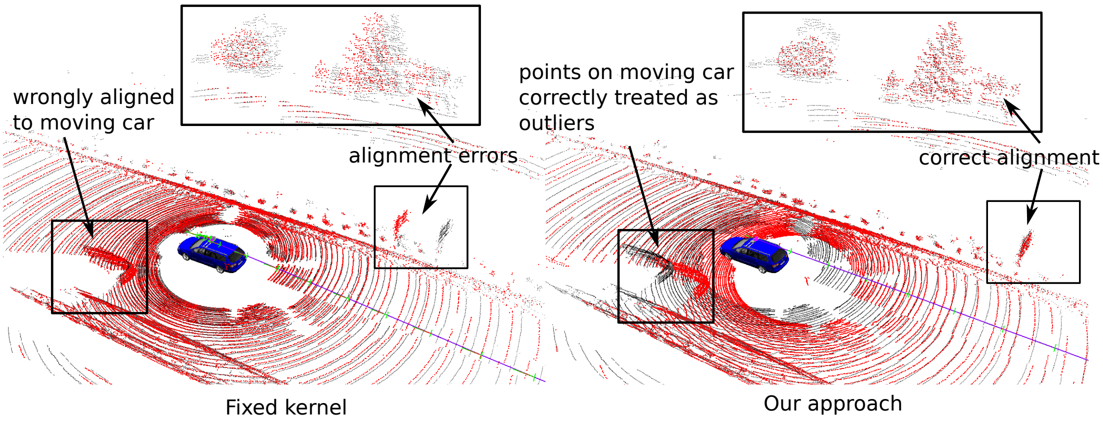


Figure 5.6: Registration by ICP and our approach on a challenging dataset. An example ICP result for a pair of consecutive frames where ICP converges to the correct transformation using our adaptive kernel (right) whereas it diverges for a fixed kernel (left). In the left image, the alignment error can be noticed with the sign-board along the road as well as the trees in the background.

computed with the adaptive kernel and once with a fixed one. The adaptive kernel results in a successful alignment while the fixed kernel fails to find the correct solution due to the outlier in the data association, see Figure 5.6. The adaptive kernel-based ICP can correctly treat the observations belonging to the moving car as outliers and nullify their effect during optimization automatically. For this sequence, we note that for large portions of the scans, α is negative and even reaches down to $\alpha_{\min} = -10$ in some instances. This suggests that our truncated adaptive loss proposed in Equation (5.13) is critical for the successful application of ICP as it enables using values $\alpha < 0$, whereas the original formulation of the adaptive loss is limited to $\alpha \in [0, 2]$. Thus, our approach greatly supports ICP-based registration as it avoids hand-crafted outlier strategies and simultaneously adapts to the outlier challenges present in each pair of scans automatically, and will adapt that for every pair of scans in a dataset.

5.2.3 Application to Bundle Adjustment

The second experiment is designed to illustrate the performance of our approach and its advantages for a state estimation problem. We choose the bundle adjustment problem using a monocular camera as an example as it is a commonly solved optimization problem in photogrammetry and computer vision, and entails estimating a large number unknowns of the model, namely the 6 DoF camera pose corresponding to each image and the 3D coordinates of features in the environment. We integrated the adaptive robust kernel to an existing bundle adjustment framework proposed by Schneider *et al.* [104]. The initial estimate for camera poses and 3D points is obtained by three commonly used steps. First, extract

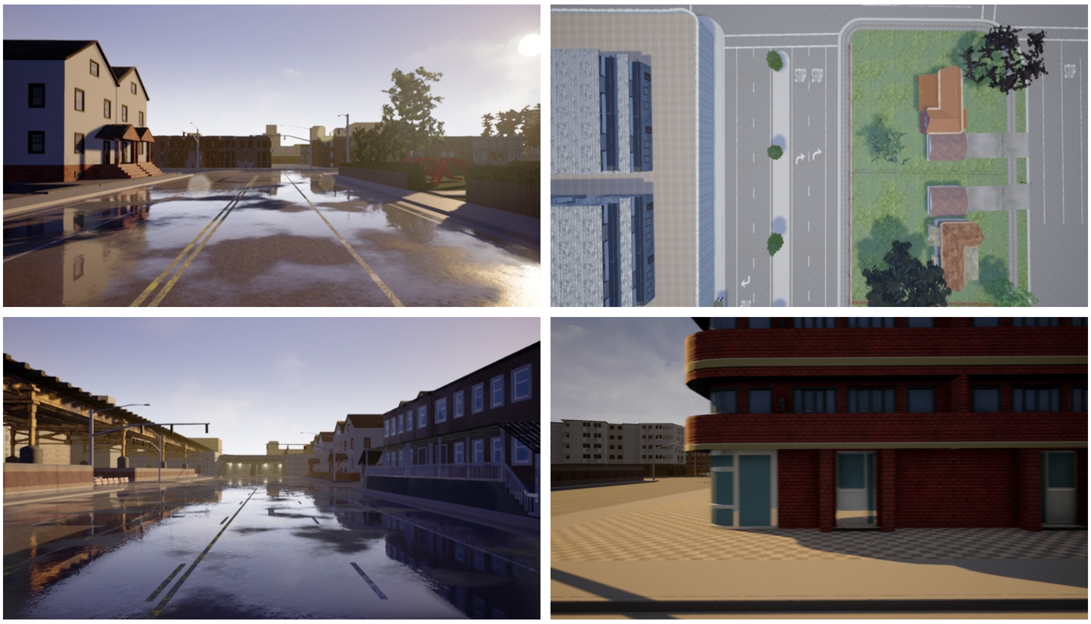


Figure 5.7: Examples images from CARLA simulated datasets. Top left: front-looking image mounted on a car from the first dataset. Top right: UAV image in a nadir view looking downwards from the second dataset. Bottom left: front-looking image with strong shadows and reflections from the third dataset. Bottom right: side facing image from the fourth dataset where portions of the image have significant motion blur.

SIFT features and compute possible matches between all image pairs. Second, compute the relative orientation using Nister's 5-point algorithm [85] together with RANSAC for outlier rejection and chaining the subsequent images to obtain the initial camera trajectory. Third, compute the 3D points as described by Läbe *et al.* [65] given the camera trajectory of the second step.

To test the performance of our approach in solving the bundle adjustment problem, we created four datasets covering different scenarios using the CARLA simulator [35] generating near-realistic images. The advantage of the simulator is that ground truth information for the camera poses is available. The first dataset contains images from a front looking camera mounted on a car, the second dataset simulates downward-looking aerial images from a UAV, the third dataset contains images where around half of each image shows strong shadows, and the fourth dataset simulates a side-ward looking camera where close-by objects suffer from significant motion blur. We have generated these datasets to cover situations where feature matching is challenging, thereby resulting in a large number of outliers. Example images from the datasets are depicted in Figure 5.7.

For each of these datasets, we evaluate the bundle adjustment results by comparing the performance of our approach against squared error loss as well as the standard Huber loss as a fixed kernel. We compute the accuracy of the camera pose estimates by comparing against the ground truth poses from the simulator

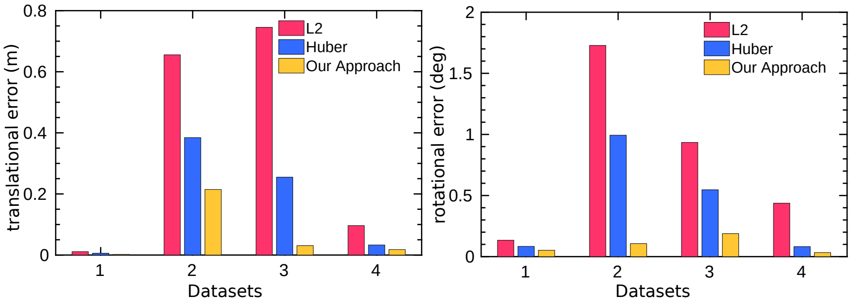
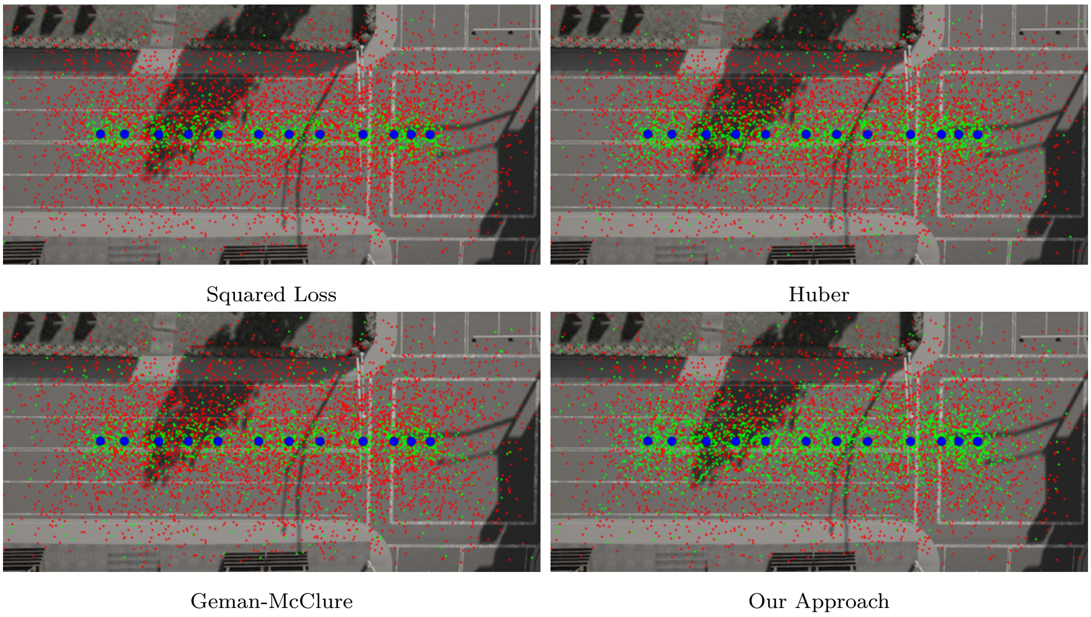


Figure 5.8: Translation and rotational errors for BA on different datasets.



Squared Loss

Huber

Geman-McClure

Our Approach

Figure 5.9: Convergence analysis for BA. Green points indicate poses for which BA converged, whereas red points indicate divergence. The blue circles represent the ground truth camera poses. The larger spread of green points reflects that our approach obtains a larger convergence basin than other fixed kernels. This indicates that our approach is more robust to error in initial solutions.

as described by Dickscheid *et al.* [32]. This difference is computed by estimating the optimal transformation between the bundle adjustment result and the ground truth using all 6 DoF pose parameters with the approach by Dickscheid *et al.* [32]. Figure 5.8 illustrates the results for all the four datasets where our approach has a lower translation and rotational error than using squared error or the fixed Huber kernel. We obtain a translation and rotational error, which is between 2 to 5 times better as compared to using Huber. We perform the ground truth comparison based only on the camera poses and do not consider the 3D point as they have been extracted using the SIFT descriptor from the simulated images. Thus, no ground truth 3D information is available.

The last experiment in this section is designed to analyze the influence of

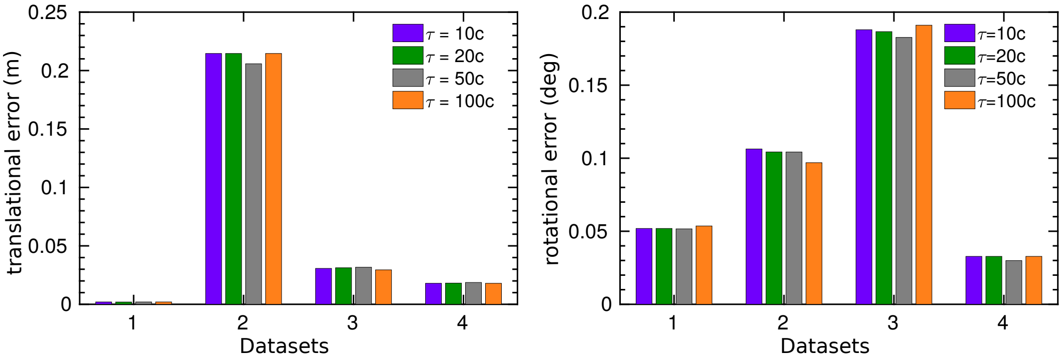


Figure 5.10: Effect of the truncation parameter τ on the accuracy on the bundle adjustment task.

our approach on the convergence properties of BA. A large basin of convergence is important for robust operation, especially for BA, due to the missing range information with the image data. We initialized the bundle adjustment procedure by adding significant noise to the initial camera poses, i.e., $\sigma \in [0.1\,\mathrm{m}, 5\,\mathrm{m}]$ to the ground truth poses of the camera. The noise in the camera poses is propagated to the 3D points during the forward intersection step. We sample 20 instances of each noise level (500 instances in total) and run the bundle adjustment for our approach, using squared loss, the Geman-McClure as well as the Huber kernel. We consider the adjustment to have converged if the final RMS error of the camera center is less than 1 cm from the true position. We visualize the results in Figure 5.9 where the poses from which the BA has converged are shown in green and the ones that caused divergence in red. We can clearly see that our approach has a larger convergence radius as the green points are spread over a larger area compared to the squared loss or fixed Huber or Geman-McClure kernel. We obtain a successful convergence rate for 45% of all instances for our approach against 24.8% for squared loss, 33% for Huber, and 28.2% for Geman-McClure. Overall, the experiments suggest that by using our approach, we can obtain a more accurate estimate and have a larger convergence area than a fixed kernel. Thus, our approach is an effective and useful approach for optimization in bundle adjustment problems.

5.2.4 Effect of the Truncation Parameter

In this experiment, our goal is to analyze the effect of the truncation parameter τ on the state estimation task. The parameter τ is used to define the integral limits for the partition function in Equation (5.12). This approximate partition function is used to define our truncated loss function proposed in this chapter. By choosing a value of τ from the set $\{10c, 20c, 50c, 100c\}$, we define a truncated partition function $\tilde{Z}(\cdot)$ for each value of τ. This results in multiple robust kernels

$\tilde{\rho}_a(\cdot)$ which are then used for solving the bundle adjustment problem on the four datasets as described in Section 5.2.3. The results for these experiments are shown in Figure 5.10. We observe that the accuracy results using different values of τ is similar for all the datasets. The maximum difference in the translation error is about 5%, and the difference in rotational error is 8% using different values for τ. We also note that during the experiments, the α estimated in Step 1 of Algorithm 2 of the optimization process belongs to a similar range of α values for each of the τ used. These results in this experiment suggest that the approach is not critically sensitive to the value of the truncation parameter used and can be used effectively for multiple state estimation tasks without any need for tuning it.

5.3 Limitations and Potential Future Work

In this chapter, we have mainly focused on extending the generalized robust kernel formulation by Barron [9] for its use in common state estimation problems in photogrammetry and robotics. It shows a notable performance gain and performs better than existing techniques. However, there are several interesting directions in which this work could be extended. We see the following aspects that offer space for further investigations:

Adapting the scale parameter c: In our current implementation, we use a fixed scale parameter c, which is set based on the measurement noise for an inlier observation. In principle, both the shape parameter α and the scale parameter c can be adapted simultaneously. In practice, however, learning both these parameters jointly becomes tricky as each of them can individually find *some explanation* of the residual distribution. Typically, a large change in the residuals can be captured by a relatively small change in c without affecting α at all. This results in a situation where the shape parameter α cannot be estimated properly. At least the straight forward integration of c as an additional variable in the optimization problem did not lead to an effective method. This suggests that a different and more advanced optimization scheme needs to be developed to adapt the parameter α and the scale parameter c jointly to respond to a change in the outlier distribution.

Use of multiple α parameters: We use a single α value to capture the outlier distribution for the individual optimization problem in our example applications. For example, in the ICP case, we estimate only one α value for all the points for a scan pair. This means the α value is adapted between different scan pairs, however, it is the same for all the points in the scan pair at one timestep t. By estimating a different α value for groups of points belonging to different objects (e.g., vegetation, road, vehicles, etc.) in the scan, we can model an inlier/outlier weight for each group of points in addition to time. The effect may be even more extreme for the BA task, as all images are considered in a single optimization problem. The current approach can only adapt α at each iteration of the optimization procedure but not estimate different α values for individual points. Here, one could estimate different α values for each sub-block of the optimization problem, where each of the sub-block could consist of the camera poses and measurements of a particular location in the environment.

Use of an alternative regularization term for the truncated loss: The truncated partition function $\tilde{Z}(\alpha)$ can be seen as playing the role of a regularizer for α in Equation (5.13). An alternative approach to regularizing α would be to replace the truncated integral term $\tilde{Z}(\alpha)$ with a suitable regularizer that is defined for any α in the range $[-\infty, 2]$. This is possible as there is no strict requirement that a robust loss $\rho(\cdot)$ must correspond to the negative log-likelihood of a probability distribution function as we have defined in this paper. This opens up the possibility of designing regularization terms with potentially better outlier rejection properties and provides an interesting direction for future work.

5.4 Related Work

Robust kernels are the de-facto solution to perform state estimation using least-squares minimization in the presence of outliers. The idea of using robust kernels for least-squares problems emerged from the seminal works by Huber [54], Hampel [48], and Koch [58]. Several robust kernels such as Huber, Cauchy, Geman-McClure, or Welsch have been proposed in the literature to deal with different outlier distributions. Zhang [138] and Bosse [14] apply these kernels to different kind of estimation problems in vision and robotics. Black and Rangarajan [13] investigate equivalence between robust loss minimization and outlier processes and apply this idea to several vision problems such as surface reconstruction, segmentation, optical flow, etc. Babin *et al.* [6] analyzed several popular robust kernels for registration problems and provided advice for using different kernels depending on the scenario. Similar analysis and recommendations are provided by MacTavish [76] and Zach [134] for visual odometry and BA. In this work, instead of choosing a specific robust kernel for a particular scenario, we dynamically adapt a robust kernel to the actual outlier distribution during the optimization process. To do this, we build upon the generalized kernel formulation recently proposed by Barron [9] for training neural networks. It generalizes over popular robust kernels, and we formulate an approximation of it for use in NLS estimation.

Several approaches have been explored in robotics literature to deal with the outliers dynamically, particularly for SLAM and bundle adjustment problems. Sünderhauf and Protzel [115] propose introducing additional switch variables to the original optimization problem, which determines whether an observation should be used or discarded during optimization. This essentially results in turning a particular constraint on or off during the optimization based on the residual distribution. Agarwal *et al.* [2] propose a robust kernel that dynamically weighs the observations without requiring to estimate any additional variables. Lajoie *et al.* [66] and Yang *et al.* [132] further this idea as a truncated least squares problem that can be solved efficiently as a semi-definite program. They also provide solutions with certain robustness guarantees for the registration and SLAM problem. Recently, Yang *et al.* [131] propose a robust estimation framework based on graduated non-convexity methods, which solves a sequence of minimization problems that are initially convex and converge eventually to the original non-convex robust loss. In a similar spirit to these approaches, we dynamically adapted the shape of a generalized robust kernel formulation based on the residual distribution resulting from the observations. We proposed a mechanism to use it within existing NLS optimization frameworks for state estimation tasks.

Taking a probabilistic view, several robust kernels are understood to arise from a probability distribution, which can be used to determine the best kernel type based on the actual observations. This means that if we know the outlier distributions for the set of observations, then we can choose a suitable robust kernel for that problem based on this knowledge. However, in practice, the outlier distribution is either unknown or changing over space and time. Agamennoni *et al.* [1] propose to use an elliptical distribution to represent several popular robust kernels. They estimate hyper-parameters for each kernel type based on the residual distribution and perform a model comparison to determine the best kernel for the situation at hand. In this chapter, we take a different approach and adapt the robust kernel shape by using the probability distribution of a generalized loss function [9]. We also do not require an explicit model comparison to choose the best kernel and estimate the kernel shape through an alternating minimization procedure.

5.5 Conclusion

State estimation is a key ingredient in photogrammetric, geodetic, and robotic systems and is often performed using some form of least squares minimization. Almost all error minimization procedures that work on real-world data use robust kernels as the standard way for dealing with outliers in the data. These kernels, however, are often hand-picked, sometimes in different combinations, and their parameters need to be tuned manually for a particular problem. In this chapter, we explored using a generalized robust kernel family, which is automatically tuned based on the distribution of the residuals and includes the common m-estimators.

Our robust optimization approach avoids the need to commit to a fixed robust kernel and potentially has a broad application area for state estimation. We use a generalized robust kernel that can adapt its shape with an additional parameter that has recently been proposed by Barron [9]. We extended the original formulation, which enabled us to use the adaptive kernel also in situations with a large proportion of outliers. We integrated our adaptive kernel into and tested it for the spatio-temporal point cloud registration and for two further popular state estimation problems in robotics, namely ICP and bundle adjustment. The experiments showcase that we achieve better performance as compared to using fixed kernels such as Huber or Geman-McClure, and at the same time, do not require any hand-crafted outlier rejection schemes. We evaluate our approach on the KITTI dataset for the registration task using ICP, and show that our approach on average gives the best accuracy when compared against other state-of-the-art robust kernels and avoids failures in strong outlier situations where other approaches tend to fail. We also show that our approach increases the radius of convergence for the bundle adjustment task suggesting that it can deal with larger errors in the initial estimates of the unknowns in the problem. We believe that several other problems in robotics, which rely on robust least-squares estimation, can benefit from our proposed approach.

Chapter 6

Conclusion

To meet the increasing demand for agricultural produce for an ever-growing world population is one of the critical challenges we face today. We need to achieve this goal within the limited arable land available to us sustainably, all the while dealing with the challenges put forth by climate change. One way to meet this challenge is by intensifying production in a sustainable manner, and we look towards the potential of robotics systems deployed in agricultural fields to achieve this goal. These robotics systems have the ability to increase productivity by providing localized high-quality care for each individual plant through continuous monitoring and timely intervention in the field while drastically reducing the use of agro-chemicals, which have taken a severe toll on our environment and biodiversity. The development of automated robotic systems has the potential to play an important role in the future of agricultural production.

In this thesis, we have developed techniques that are relevant for monitoring and phenotyping tasks for robotics systems in the agricultural domain. We have focused on the core task of registration, which is a fundamental requirement for any mobile system that perceives its surroundings. Existing techniques for registering sensor data from agricultural settings fail to perform reliably due to a unique set of challenges inherent to this domain. These challenges vary from the large change in the field's visual appearance, to the structural change of individual plants as they grow over the crop season, and to the vastly differing viewpoints where data is captured from multiple platforms in an aerial-ground robotics system. Throughout the thesis, we have developed registration techniques that are designed to handle these challenges and explicitly take into account the spatio-temporal nature of the task. We build on these registration techniques to demonstrate their application for long-term monitoring of crops in the field, developing an accurate localization system for navigation in crop fields, and performing automated phenotyping tools to analyze the growth of individual plant parts from high fidelity point cloud data.

We believe that the registration techniques developed here would contribute to the robust operation of autonomous robots in the field over long periods of time and provide a basis for analyzing plants on a large scale. The key contributions of this thesis are novel techniques for spatio-temporal registration, which enable the operation of agricultural robots over long time periods and perform robustly in the face of various challenges that arise in this domain.

The first contribution of this thesis is a registration technique that enabled monitoring crop fields using UAVs over long periods of time. We designed a novel descriptor and a data association scheme designed for crop fields, which allowed for registering UAV images over an entire crop season. We exploited the fact the plants change their appearance but cannot move. Based on this knowledge, we took into account the geometric patterns in the crop field to develop an effective solution in challenging situations where state-of-the-art visual descriptor based association failed. Using our registration results, we showed that it is possible to analyze the plant growth in fields over long time periods at a spatial resolution few square centimeters.

The second contribution of this thesis is a localization system for UGVs operating in crop fields by exploiting aerial maps generated by UAVs. We register the data captured from the two platforms with a large viewpoint difference reliably based on a combination of features that captures the geometry and the semantics of the crop field. The semantic information of the crops, weeds, and their stem positions allowed us to resolve the visual aliasing problem caused due to the similarly looking crop rows. This feature combination allowed us to match the images captured from the UGV to the aerial map of the field generated from a prior UAV survey flight. We also built upon domain knowledge of crop fields and other field management operations to capture the changes and integrate them properly into our state belief. All this information enabled us to register the data robustly, captured both from differing viewpoints and with extended time intervals between the sessions. Our robust registration procedure along coupled with a particle filter based estimation framework, helped us realize an accurate localization system and provided better pose estimates than by a typical single phase GPS-based localization approach.

The third contribution is a novel approach for spatio-temporal registration of high fidelity 3D point clouds of plants. This is relevant for tracking plant growth at a detailed level, which is used in crop sciences for phenotyping tasks. We addressed the task of registering high-resolution plant scans, which enabled us to deal with the non-rigid motion as well as plant growth. We took advantage of the skeletal structure and the semantic information of the plant to find reliable matches between corresponding parts of the plant over time, which was then used to drive the registration process through an iterative procedure. The

Figure 6.1: Photos from various integration and review meetings of the Flourish project. Left: Integration of the precision spraying system on the robot for weed treatement. Middle: Demonstration of precision weeding capability in autonomous mode to local farmers and other stakeholders in Ancona, Italy. Right: The UAV and UGV team entering the field for final project demonstration in Eschikon, Switzerland.

experimental results showed that our approach effectively dealt with changing appearance and topology of the plant, whereas typical state-of-the-art registration techniques that consider only rigid motions are not suitable for plant data. Based on the registration technique developed in this thesis, we analyzed plant growth at an organ level. We also detected interesting events such as the emergence of new leaves, all of which play a key role in characterizing phenotypic traits of a plant. We evaluated our approach on challenging datasets from two plant species and demonstrated how automated phenotyping could be performed using our registration approach.

As the final contribution of the thesis, we propose a technique for dealing with outliers, which arise during the data association process. These outliers result in registration failures and lead to poor solutions during the state estimation process. To deal with this, we studied a generalized robust kernel approach, which adapts automatically based on the current error residual. We extended the capability of this technique to deal with strong outlier distributions. We applied it successfully in the context of least-squares optimization problems, which are frequently required for registration tasks, as well as other state estimation procedures in robotics and photogrammetry.

6.1 Contributions to the Projects Flourish and PhenoRob

The spatio-temporal registration techniques for crop fields presented in this thesis have contributed to the development of an autonomous robotics system for precision agriculture as part of a successful EC-funded project Flourish, H2020-ICT-644227-FLOURISH and later on to the cluster of excellence EXC-2070 PhenoRob. The goal of the project was to show the feasibility of using an autonomous robotic system for precision agriculture tasks such as continuous plant monitoring and target intervention for field management using a team of an aerial and a ground robot.

On September 18, 2018, the final review of the 42-month project was successful and Flourish project has been evaluated excellently in all the review meetings. The techniques presented in this thesis provided core building blocks for registering data acquired by sensors on-board the robots, and enabled the spatio-temporal analysis of the plant growth. Figure 6.1 shows photos from various integration and project review meetings as well demonstrations given in an effort to reach the public and other stakeholders.

Our work in Chapter 4 on spatio-temporal registration of plant point clouds happened in the context of PhenoRob for deriving plant traits and analyzing growth from high-resolution scans, and is linked to core project 1 on 4D crop reconstruction. Thus, we would also like to acknowledge that parts of this thesis are contributions to the Excellence cluster PhenoRob supported by the German Research Foundation under Germany's Excellence Strategy, EXC-2070-390732324.

In addition to the contributions to the projects, we have released most of the techniques developed in thesis as open source software, and published three challenging datasets for long-term spatio-temporal registration tasks.

6.2 Future Work

The techniques we developed as well as the promising results presented in this thesis open different directions for future research. In Chapter 2, we proposed a descriptor for registering UAV images of crop fields that exploited the geometric arrangement of crops and gaps, i.e., the missing crops along the crop-row. We developed this idea in the classical expert-driven way, where we arrived at this descriptor by carefully observing the data and exploiting this pattern to design the descriptor. However, such patterns may not be available for other fields, or their distribution could be completely different, in which case our approach might not be adequate. An alternative approach is not to specify the pattern directly but to use the data itself to figure out the pattern. Given the recent advancements in deep learning techniques over the last few years, new descriptors learned from data suited for field environments could offer potential solutions. These approaches would have several challenges in terms of generalizability and re-training effort required to achieve robust performance. However, exploiting some prior knowledge of the fields might tackle some of these problems and offer an interesting research avenue.

Later in Chapter 2, we proposed a co-operative localization system where the information exchange between the UAV and UGV is only via the maps that the two robots share. There is further scope for having much closer and higher frequency collaboration between different robots operating on the field. In the field scenario, there are potentially several situations in which the robots can anticipate each others' behavior and thereby take actions that increase throughput and productivity for the desired task. Developing techniques that further tighter collaborations between the robots would be a critical requirement for realizing several field applications.

Our registration approach for the plant point clouds in Chapter 4 assumes to have high-quality scans available. Although this is a reasonable assumption to make for data acquired in laboratory settings or at a small scale in green houses, data captured in the fields from moving platforms will often be much lower resolution, with larger noise levels, with parts of the plant unobserved. Extending our approach to deal with the noisier point cloud data in these challenging situations would be an important milestone.

Finally, in Chapter 5, we had mainly focused on applying a generalized robust kernel formulation to common state estimation problems in robotics. We identified several aspects such as adapting the scale parameter c of the kernel dynamically, using multiple α parameters for large state estimation problems and using alternative regularization terms for the loss function. We think that further investigation into these issues opens up the possibility of achieving better outlier rejection properties and provides an interesting direction for future work.

Overall, we believe that this thesis provides a strong baseline for long-term registration techniques in the agricultural domain and provides solutions to employ in real systems. As always, there are several dimensions that one could build upon this work, including extending our techniques to a larger variety of plants, improving the computational complexity of the registration techniques, and dealing with even more challenging data.

Bibliography

[1] G. Agamennoni, P.T. Furgale, and R. Siegwart. Self-Tuning M-Estimators. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2015.

[2] P. Agarwal, G.D. Tipaldi, L. Spinello, C. Stachniss, and W. Burgard. Robust Map Optimization using Dynamic Covariance Scaling. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, Karlsruhe, Germany, 2013.

[3] S. Agarwal, K. Mierle, and Others. Ceres Solver. http://ceres-solver.org, 2010.

[4] A. Ahmadi, L. Nardi, N. Chebrolu, and C. Stachniss. Visual Servoing-based Navigation for Monitoring Row-Crop Fields. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2020.

[5] G. Alenya, B. Dellen, and C. Torras. 3d modelling of leaves from color and tof data for robotized plant measuring. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 3408 – 3414, 06 2011.

[6] P. Babin, P. Giguere, and F. Pomerleau. Analysis of robust functions for registration algorithms. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019.

[7] T. Bailey and H.F. Durrant-Whyte. Simultaneous localisation and mapping (SLAM): Part I. *IEEE Robotics and Automation Magazine (RAM)*, 13(2):99–110, 2006.

[8] T. Bailey and H.F. Durrant-Whyte. Simultaneous localisation and mapping (SLAM): Part II. *IEEE Robotics and Automation Magazine (RAM)*, 13(3):108 –117, 2006.

[9] J. T. Barron. A General and Adaptive Robust Loss Function. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[10] J. Beddington, M. Asaduzzaman, A. Fernandez, M. Clark, M. Guillou, M. Jahn, L. Erda, Mamo T, N.V. Bo, C.A. Nobre, R. Scholes, R. Sharma, and J. Wakhungu. Achieving food security in the face of climate change: Final report from the commission on sustainable agriculture and climate change, 2012.

[11] J. Behley and C. Stachniss. Efficient Surfel-Based SLAM using 3D Laser Range Data in Urban Environments. In *Proc. of Robotics: Science and Systems (RSS)*, 2018.

[12] P.J. Besl and N.D. McKay. A Method for Registration of 3D Shapes. *IEEE Trans. on Pattern Analalysis and Machine Intelligence (TPAMI)*, 14(2):239–256, 1992.

[13] M. J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *Intl. Journal of Computer Vision (IJCV)*, 19(1):57–91, 1996.

[14] M. Bosse, G. Agamennoni, and I. Gilitschenski. Robust Estimation and Applications in Robotics. *Foundations and Trends in Robotics*, 4(4):225–269, 2016.

[15] S. Bouaziz, A. Tagliasacchi, H. Li, and M. Pauly. Modern techniques and applications for real-time non-rigid registration. In *SIGGRAPH ASIA 2016 Courses*, SA '16, pages 11:1–11:25, New York, NY, USA, 2016. ACM.

[16] M. Bryson, A. Reid, F.T. Ramos, and S. Sukkarieh. Airborne vision-based mapping and classification of large farmland environments. *Journal of Field Robotics (JFR)*, 27(5):632–655, 2010.

[17] M. Carcassoni and E.R. Hancock. Spectral correspondence for point pattern matching. *Pattern Recognition*, 36(1):193–204, 2003.

[18] L. Carlone, J. Dong, S. Fenu, G.G. Rains, and F. Dellaert. Towards 4d crop analysis in precision agriculture: Estimating plant height and crown radius over time via expectation-maximization. In *ICRA Workshop on Robotics in Agriculture*, 2015.

[19] N. Chebrolu, T. Läbe, and C. Stachniss. Robust Long-Term Registration of UAV Images of Crop Fields for Precision Agriculture. *IEEE Robotics and Automation Letters (RA-L)*, 3(4):3097–3104, 2018.

[20] N. Chebrolu, T. Läbe, and C. Stachniss. Spatio-Temporal Non-Rigid Registration of 3D Point Clouds of Plants. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2020.

[21] N. Chebrolu, T. Läbe, and C. Stachniss. Adaptive Robust Kernels for Non-Linear Least Squares Problems. *IEEE Robotics and Automation Letters (RA-L)*, 6(2):2240–2247, 2021.

[22] N. Chebrolu, P. Lottes, T. Läbe, and C. Stachniss. Robot Localization Based on Aerial Images for Precision Agriculture Tasks in Crop Fields. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019.

[23] N. Chebrolu, P. Lottes, A. Schäfer, W. Winterhalter, W. Burgard, and C. Stachniss. Agricultural Robot Dataset for Plant Classification, Localization and Mapping on Sugar Beet Fields. *Intl. Journal of Robotics Research (IJRR)*, 36(10):1045–1052, 2017.

[24] N. Chebrolu, F. Magistri, T. Läbe, and C. Stachniss. Registration of Spatio-Temporal Point Clouds of Plants for Phenotyping. *PLOS ONE*, 16(2):1–25, 2021.

[25] X. Chen, T. Läbe, A. Milioto, T. Röhling, O. Vysotska, A. Haag, J. Behley, and C. Stachniss. OverlapNet: Loop Closing for LiDAR-based SLAM. In *Proc. of Robotics: Science and Systems (RSS)*, 2020.

[26] G. Christie, G. Warnell, and K. Kochersberger. Semantics for UGV Registration in GPS-denied Environments. *arXiv preprint*, 1609.04794v2 [cs.RO], 2016.

[27] W. Churchill and P. Newman. Experience-Based Navigation for Long-Term Localisation. *Intl. Journal of Robotics Research (IJRR)*, 2013.

[28] M. Cummins and P. Newman. FAB-MAP: Probabilistic localization and mapping in the space of appearance. *Intl. Journal of Robotics Research (IJRR)*, 27(6):647–665, 2008.

[29] J. Das, G. Cross, C. Qu, A. Makineni, P. Tokekar, Y. Mulgaonkar, and V. Kumar. Devices, systems, and methods for automated monitoring enabling precision agriculture. In *Proc. of the IEEE on Automation Science and Engineering (CASE)*, pages 462–469, 2015.

[30] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA)*, 1999.

[31] F. Dellaert and M. Kaess. Square root sam: Simultaneous localization and mapping via square root information smoothing. *Intl. Journal of Robotics Research (IJRR)*, 25(12):1181–1203, 2006.

[32] T. Dickscheid, T. Läbe, and W. Förstner. Benchmarking automatic bundle adjustment results. In *Cong. of the Intl. Society for Photogrammetry and Remote Sensing (ISPRS)*, 2008.

[33] M. Ding, K. Lyngbaek, and A. Zakhor. Automatic registration of aerial imagery with untextured 3d lidar models. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

[34] J. Dong, J.G. Burnham, B. Boots, G. Rains, and F. Dellaert. 4D Crop Monitoring: Spatio-Temporal Reconstruction for Agriculture. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2017.

[35] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. CARLA: An open urban driving simulator. In *Proc. of the Conf. on Robot Learning (CORL)*, 2017.

[36] M. Ester, H. Kriegel, J. Sander, and X.Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of the Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 226–231, 1996.

[37] F. Fiorani, U. Rascher, S. Jahnke, and U. Schurr. Imaging plants dynamics in heterogenic environments. *Current Opinion in Biotechnology*, 23(2):227–235, 2012.

[38] F. Fiorani and U. Schurr. Future scenarios for plant phenotyping. *Annual Review of Plant Biology*, 64:267–291, 2013.

[39] R. T. Furbank and M. Tester. Phenomics – technologies to relieve the phenotyping bottleneck. *Trends in Plant Science*, 16(12):635 – 644, 2011.

[40] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE Trans. on Pattern Analalysis and Machine Intelligence (TPAMI)*, 32(8):1362–1376, 2010.

[41] J. Gall, C. Stoll, E. De Aguiar, C. Theobalt, B. Rosenhahn, and H.P. Seidel. Motion capture using joint skeleton tracking and surface estimation. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1746–1753, 2009.

[42] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3354–3361, 2012.

[43] H.C.J. Godfray, J.R. Beddington, R.R. Crute, L. Haddad, D. Lawrence, J.F. Muir, J.N. Pretty, S. Robinson, S.M. Thomas, and C. Toulmin. Food security: the challenge of feeding 9 billion people. *Science*, 327 5967:812–8, 2010.

[44] 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.

[45] S. Griffith and C. Pradalier. Reprojection flow for image registration across seasons. In *Proc. of British Machine Vision Conference (BMVC)*, 2016.

[46] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard. A tutorial on graph-based SLAM. *IEEE Trans. on Intelligent Transportation Systems Magazine (T-ITS Mag)*, 2:31–43, 2010.

[47] D. Haehnel, S. Thrun, and W. Burgard. An extension of the icp algorithm for modeling nonrigid objects with mobile robots. In *Proc. of the Intl. Conf. on Artificial Intelligence (IJCAI)*, volume 3, pages 915–920, 2003.

[48] F.R. Hampel, E.M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics: The Approach Based on Influence Functions.* John Wiley and Sons, 1986.

[49] M. T. Heath. *Scientific Computing: An Introductory Survey.* McGraw-Hill Higher Education, 2nd edition, 1996.

[50] L. Herda, P. Fua, R. Plankers, R. Boulic, and D. Thalmann. Skeleton-based motion capture for robust reconstruction of human motion. In *Proc. on Computer Animation*, pages 77–83, May 2000.

[51] M Hess, G Barralis, H Bleiholder, L Buhr, TH Eggers, H Hack, and R Stauss. Use of the extended bbch scale—general for the descriptions of the growth stages of mono; and dicotyledonous weed species. *Weed Research*, 37(6):433–441, 1997.

[52] H. Huang, S. Wu, D. Cohen-Or, M. Gong, H. Zhang, G. Li, and B. Chen. L1-medial skeleton of point cloud. *ACM Trans. on Graphics (TOG)*, 32(4):65–1, 2013.

[53] P. J. Huber. Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35(1):73–101, 1964.

[54] P. J. Huber. *Robust Statistics.* Wiley, 1981.

[55] M. Innmann, M. Zollhöfer, M. Nießner, C. Theobalt, and M. Stamminger. VolumeDeform: Real-time Volumetric Non-rigid Reconstruction. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, pages 362–379, 2016.

[56] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Trans. on Robotics (TRO)*, 24(6), 2008.

[57] R. Klose, J. Penlington, and A. Ruckelshausen. Usability study of 3d time-of-flight cameras for automatic plant phenotyping. *Bornimer Agrartechnische Berichte*, 69(93-105):12, 2009.

[58] K.R. Koch. *Parameter Estimation and Hypothesis Testing in Linear Models.* Springer-Verlag, 1988.

[59] T. Kohonen. The self-organizing map. In *Proc. of the IEEE*, pages 1464–1480, 1990.

[60] F. Kraemer, A. Schaefer, A. Eitel, J. Vertens, and W. Burgard. From Plants to Landmarks: Time-invariant Plant Localization that uses Deep Pose Regression in Agricultural Fields. In *IROS Workshop on Agri-Food Robotics*, 2017.

[61] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A general framework for graph optimization. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 3607–3613, 2011.

[62] K. Kusumam, T. Krajník, S. Pearson, G. Cielniak, and T. Duckett. Can you pick a broccoli? 3d-vision based detection and localisation of broccoli heads in the field. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 646–651, 2016.

[63] T.B. Kwon and J.B. Song. A new feature commonly observed from air and ground for outdoor localization with elevation map built by aerial mapping system. *Journal of Field Robotics*, 28(2):227–240, 2010.

[64] R. Kümmerle, B. Steder, C. Dornhege, A. Kleiner, G. Grisetti, and W. Burgard. Large scale graph-based slam using aerial images as prior information. *Autonomous Robots*, 30(1):25–39, 2011.

[65] T. Läbe and W. Förstner. Automatic relative orientation of images. In *Proceedings of the 5th Turkish-German Joint Geodetic Days*, Berlin, 2006.

[66] P. Lajoie, S. Hu, G. Beltrame, and L. Carlone. Modeling perceptual aliasing in SLAM via discrete-continuous graphical models. *IEEE Robotics and Automation Letters (RA-L)*, 4(2):1232–1239, 2019.

[67] K. Y. K. Leung, C. M. Clark, and J. P. Huissoon. Localization in urban environments by matching ground level video images with an aerial image. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 551–556, 2008.

[68] S. Leutenegger, M. Chli, and R. Siegwart. BRISK: Binary robust invariant scalable keypoints. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, pages 2548–2555, 2011.

[69] Y. Li, X. Fan, N.J. Mitra, D. Chamovitz, D. Cohen-Or, and B. Chen. Analyzing growing plants from 4d point cloud data. *ACM Transactions on Graphics*, 32(6):157, 2013.

[70] P. Lottes, J. Behley, N. Chebrolu, A. Milioto, and C. Stachniss. Joint Stem Detection and Crop-Weed Classification for Plant-specific Treatment in Precision Farming. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2018.

[71] P. Lottes, J. Behley, N. Chebrolu, A. Milioto, and C. Stachniss. Robust Joint Stem Detection and Crop-Weed Classification using Image Sequences for Plant-Specific Treatment in Precision Farming. *Journal of Field Robotics (JFR)*, 37:20–34, 2020.

[72] P. Lottes, N. Chebrolu, F. Liebisch, and C. Stachniss. UAV-based Field Monitoring for Precision Farming. In *25. Workshop Computer-Bildanalyse in der Landwirtschaft*, 2019.

[73] P. Lottes, R. Khanna, J. Pfeifer, R. Siegwart, and C. Stachniss. UAV-Based Crop and Weed Classification for Smart Farming. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2017.

[74] D.G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *Intl. Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004.

[75] S. Lowry, N. Sunderhauf, P. Newman, J.J Leonard, D. Cox, P. Corke, and M.J. Milford. Visual place recognition: A survey. *IEEE Trans. on Robotics (TRO)*, 32(1):1–19, 2016.

[76] K. MacTavish and T. D. Barfoot. At all costs: A comparison of robust cost functions for camera correspondence outliers. In *Proc. of the IEEE Conf. on Computer and Robot Vision*, pages 62–69, 2015.

[77] F. Magistri, N. Chebrolu, J. Behley, and C. Stachniss. Towards In-Field Phenotyping Exploiting Differentiable Rendering with Self-Consistency

Loss. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2021.

[78] F. Magistri, N. Chebrolu, and C. Stachniss. Segmentation-Based 4D Registration of Plants Point Clouds for Phenotyping. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[79] A. Majdik, D. Verda, Y. Albers-Schoenberg, and D. Scaramuzza. Air-ground matching: Appearance-based gps-denied urban localization of micro aerial vehicles. *Journal of Field Robotics*, 32(7):1015–1039, 2015.

[80] R. Martin-Brualla, D. Gallup, and S.M. Seitz. 3d time-lapse reconstruction from internet photos. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, pages 1332–1340, 2015.

[81] C. R. Maurer, Rensheng Qi, and V. Raghavan. A linear time algorithm for computing exact euclidean distance transforms of binary images in arbitrary dimensions. *IEEE Trans. on Pattern Analalysis and Machine Intelligence (TPAMI)*, 25(2):265–270, 2003.

[82] M. Milford and G.F. Wyeth. SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2012.

[83] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society of Industrial and Applied Mathematics*, 5(1):32–38, 1957.

[84] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-Time Dense Surface Mapping and Tracking. In *Proc. of the Intl. Symposium on Mixed and Augmented Reality (ISMAR)*, pages 127–136, 2011.

[85] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. on Pattern Analalysis and Machine Intelligence (TPAMI)*, 26(6):756–770, 2004.

[86] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[87] E. Palazzolo, J. Behley, P. Lottes, P. Giguere, and C. Stachniss. ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[88] E. Palazzolo and C. Stachniss. Fast Image-Based Geometric Change Detection Given a 3D Model. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2018.

[89] A. Paproki, X. Sirault, S. Berry, R. Furbank, and J. Fripp. A novel mesh processing based technique for 3d plant analysis. *BMC Plant Biology*, 12(1):63, 2012.

[90] S. Paulus, J. Dupuisand A. Mahlein, and H. Kuhlmann. Surface feature based classification of plant organs from 3d laserscanned point clouds for plant phenotyping. *BMC Bioinformatics*, 14(1):238, 2013.

[91] S. Paulus, H. Schumann, H. Kuhlmann, and J. Léon. High-precision laser scanning system for capturing 3D plant architecture and analysing growth of cereal plants. *Biosystems Engineering*, 121:1–11, 2014.

[92] J. Pfeifer, R. Khanna, C. Dragos, M. Popovic, E. Galceran, N. Kirchgessner, A. Walter, R. Siegwart, and F. Liebisch. Towards automatic uav data interpretation for precision farming. In *Proc. of the Conf. of Agricultural Engineering (CIGR)*, 2016.

[93] C. Potena, R. Khanna, J. Nieto, R. Siegwart, D. Nardi, and A. Pretto. Agricolmap: Aerial-ground collaborative 3d mapping for precision farming. *IEEE Robotics and Automation Letters (RA-L)*, 4(2):1085–1092, 2019.

[94] 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. *IEEE Robotics and Automation Magazine (RAM)*, 2020.

[95] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[96] C.R. Qi, K. Yi, H. Su, and L. J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Proc. of the Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[97] R. Qin, J. Tian, and P. Reinartz. 3D Change Detection – Approaches and Applications. *ISPRS Journal of Photogrammetry and Remote Sensing (JPRS)*, 122:41–56, 2016.

[98] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2):257–286, 1989.

[99] D. Rosen, J. Mason, and J. Leonard. Towards Lifelong Feature-Based Mapping in Semi-Static Environments. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2016.

[100] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: an efficient alternative to sift or surf. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2011.

[101] P. Ruchti, B. Steder, M. Ruhnke, and W. Burgard. Localization on OpenStreetMap Data Using a 3D Laser Scanner. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2015.

[102] R.B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, pages 3212–3217, 2009.

[103] K. Sakurada, T. Okatani, and K. Deguchi. Detecting Changes in 3D Structure of a Scene from Multi-View Images Captured by a Vehicle-Mounted Camera. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 137–144, 2013.

[104] J. Schneider, F. Schindler, T. Läbe, and W. Förstner. Bundle adjustment for multi-camera systems with points at infinity. In *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume I-3, 2012.

[105] D. Schunck, F. Magistri, R. A. Rosu, A. Cornelißen, N. Chebrolu, S. Paulus, J. Léon, S. Behnke, C. Stachniss, H. Kuhlmann, and L. Klingbeil. Pheno4D: A Spatio-Temporal Dataset of Maize and Tomato Plant Point Clouds for Phenotyping and Advanced Plant Analysis. *PLOS ONE*, 2021. Revised version submitted, under review.

[106] L.A. Schwarz, A. Mkhitaryan, D. Mateus, and N. Navab. Human skeleton tracking from depth data using geodesic distances and optical flow. *Image and Vision Computing*, 30(3):217 – 226, 2012.

[107] W. Shi, R. van de Zedde, H. Jiang, and G. Kootstra. Plant-part segmentation using deep learning and multi-view vision. *Biosystems Engineering*, 187:81–95, 2019.

[108] K. Shoemake. Animating Rotation with Quaternion Curves. *Proc. of the Intl. Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 245–254, 1985.

[109] K. Shoemake and T. Duff. Matrix animation and polar decomposition. In *Proc. of the Conf. on Graphics Interface*, volume 92, pages 258–264, 1992.

[110] P. Sodhi, S. Vijayarangan, and D. Wettergreen. In-field segmentation and identification of plant structures using 3d imaging. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 5180–5187, 2017.

[111] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry Processing*, volume 4, pages 109–116, 2007.

[112] C. Stachniss, J. Leonard, and S. Thrun. *Springer Handbook of Robotics, 2nd edition*, chapter Chapt. 46: Simultaneous Localization and Mapping. Springer Verlag, 2016.

[113] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, pages 945–953, 2015.

[114] R. W. Sumner, J. Schmid, and M. Pauly. Embedded deformation for shape manipulation. *ACM Trans. on Graphics (TOG)*, 26(3):80, 2007.

[115] N. Sünderhauf and P. Protzel. Switchable constraints for robust pose graph slam. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1879–1884, 2012.

[116] A. Tagliasacchi, T. Delame, M. Spagnuolo, N. Amenta, and A. Telea. 3d skeletons: A state-of-the-art report. In *Computer Graphics Forum*, volume 35, pages 573–597. Wiley Online Library, 2016.

[117] A. Tagliasacchi, H. Zhang, and D. Cohen-Or. Curve skeleton extraction from incomplete point cloud. In *ACM Trans. on Graphics (TOG)*, volume 28, page 71, 2009.

[118] A. Taneja, L. Ballan, and M. Pollefeys. Geometric Change Detection in Urban Environments Using Images. *IEEE Trans. on Pattern Analalysis and Machine Intelligence (TPAMI)*, 37(11):2193–2206, 2015.

[119] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics.* MIT Press, 2005.

[120] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*, ICCV '99, page 298–372, Berlin, Heidelberg, 1999. Springer-Verlag.

[121] A.O. Ulusoy and J.L. Mundy. Image-Based 4D Reconstruction Using 3D Change Detection. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, pages 31–45, 2014.

[122] T. A. Vidal-Calleja, C. Berger, J. Solà, and S. Lacroix. Large scale multiple robot visual mapping with heterogeneous landmarks in semi-structured terrain. *Journal on Robotics and Autonomous Systems (RAS)*, 59(9):654 – 674, 2011.

[123] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. on Information Theory*, 13(2):260–269, 1967.

[124] I. Vizzo, X. Chen, N. Chebrolu, J. Behley, and C. Stachniss. Poisson Surface Reconstruction for LiDAR Odometry and Mapping. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2021.

[125] O. Vysotska and C. Stachniss. Lazy Data Association For Image Sequences Matching Under Substantial Appearance Changes. *IEEE Robotics and Automation Letters (RA-L)*, 1(1):213–220, 2016.

[126] O. Vysotska and C. Stachniss. Effective Visual Place Recognition Using Multi-Sequence Maps. *IEEE Robotics and Automation Letters (RA-L)*, 4:1730–1736, 2019.

[127] A Walter, F. Liebisch, and A. Hund. Plant phenotyping: from bean weighing to image analysis. *Plant Methods*, 11(1), 2015.

[128] X. Wang, S. Vozar, and E. Olson. FLAG: Feature-based Localization between Air and Ground. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2017.

[129] W. Winterhalter, F. V. Fleckenstein, C. Dornhege, and W. Burgard. Crop row detection on tiny plants with the pattern hough transform. *IEEE Robotics and Automation Letters*, 3:3394–3401, 2018.

[130] H.J. Wolfson and I. Rigoutsos. Geometric hashing: an overview. *IEEE Computational Science and Engineering*, 4(4):10–21, 1997.

[131] H. Yang, P. Antonante, V. Tzoumas, and L. Carlone. Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection. *IEEE Robotics and Automation Letters (RA-L)*, 5(2):1127–1134, 2020.

[132] H. Yang, J. Shi, and L. Carlone. TEASER: Fast and Certifiable Point Cloud Registration. *IEEE Trans. on Robotics (TRO)*, 37(2):314–333, 2020.

[133] L. Yang, J.M. Normand, and G. Moreau. Local geometric consensus: A general purpose point pattern-based tracking algorithm. *IEEE Trans. on Visualization and Computer Graphics*, 21(11):1299–1308, 2015.

[134] C. Zach. Robust bundle adjustment revisited. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, pages 772–787, 2014.

[135] D. Zermas, V. Morellas, D. Mulla, and N. Papanikolopoulos. Estimating the leaf area index of crops through the evaluation of 3d models. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 6155–6162, 2017.

[136] D. Zermas, V. Morellas, D. Mulla, and N. Papanikolopoulos. Extracting phenotypic characteristics of corn crops through the use of reconstructed 3d models. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 8247–8254, 2018.

[137] J. Zhang and S. Singh. LOAM: Lidar Odometry and Mapping in Real-time. In *Proc. of Robotics: Science and Systems (RSS)*, 2014.

[138] Z. Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. *Image and Vision Computing*, 15:59–76, 1997.

[139] Q. Zheng, A. Sharf, A. Tagliasacchi, B. Chen, H. Zhang, A. Sheffer, and D. Cohen-Or. Consensus skeleton for non-rigid space-time registration. In *Computer Graphics Forum*, volume 29, pages 635–644, 2010.

List of Figures

List of Tables

List of Algorithms