Navigation control of a robotic vehicle for field-based phenotyping

Jingyao Gai

Iowa State University
Navigation control of a robotic vehicle for field-based phenotyping

by

Jingyao Gai

A dissertation submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Agricultural and Biosystems Engineering

Program of Study Committee:
Lie Tang, Major Professor
Stuart Birrell
Kathleen Delate
Liang Dong
Brian L. Steward

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this dissertation. The Graduate College will ensure this dissertation is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University
Ames, Iowa
2020

Copyright © Jingyao Gai, 2020. All rights reserved.
### TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>LIST OF FIGURES</th>
<th>iii</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIST OF TABLES</td>
<td>vi</td>
</tr>
<tr>
<td>ACKNOWLEDGMENT</td>
<td>vii</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>viii</td>
</tr>
<tr>
<td>CHAPTER 1. GENERAL INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>Background</td>
<td>1</td>
</tr>
<tr>
<td>Review of the State of the Art of Field-based HTPP</td>
<td>2</td>
</tr>
<tr>
<td>Research Objectives</td>
<td>7</td>
</tr>
<tr>
<td>Dissertation Outline</td>
<td>9</td>
</tr>
<tr>
<td>References</td>
<td>10</td>
</tr>
<tr>
<td>CHAPTER 2. GPS-BASED NAVIGATION CONTROL FOR A CENTRAL-ARTICULATED ROBOTIC VEHICLE</td>
<td>14</td>
</tr>
<tr>
<td>Abstract</td>
<td>14</td>
</tr>
<tr>
<td>Introduction</td>
<td>15</td>
</tr>
<tr>
<td>Navigation System Development</td>
<td>19</td>
</tr>
<tr>
<td>Rover Design and Kinematic Modeling</td>
<td>20</td>
</tr>
<tr>
<td>Robot Localization</td>
<td>23</td>
</tr>
<tr>
<td>Path Tracking (Steering Control)</td>
<td>27</td>
</tr>
<tr>
<td>System Implementation and Simulation</td>
<td>30</td>
</tr>
<tr>
<td>Experiments</td>
<td>34</td>
</tr>
<tr>
<td>Path Tracking Algorithms Evaluation in Simulation</td>
<td>34</td>
</tr>
<tr>
<td>Navigation System Evaluation in Field Tests</td>
<td>36</td>
</tr>
<tr>
<td>Results and Discussions</td>
<td>37</td>
</tr>
<tr>
<td>Tracking Algorithms Evaluation in Simulation</td>
<td>37</td>
</tr>
<tr>
<td>Navigation System Evaluation in Field Tests:</td>
<td>39</td>
</tr>
<tr>
<td>Conclusions</td>
<td>40</td>
</tr>
<tr>
<td>Acknowledgment:</td>
<td>41</td>
</tr>
<tr>
<td>References</td>
<td>41</td>
</tr>
<tr>
<td>CHAPTER 3. USING A DEPTH CAMERA FOR VISUAL ODOMETRY AND FIELD MAPPING IN UNDER-CANOPY AGRICULTURAL ROBOTIC NAVIGATION</td>
<td>45</td>
</tr>
<tr>
<td>Abstract</td>
<td>45</td>
</tr>
<tr>
<td>Introduction</td>
<td>46</td>
</tr>
<tr>
<td>Related works</td>
<td>48</td>
</tr>
<tr>
<td>Material and Method</td>
<td>51</td>
</tr>
<tr>
<td>1. PhenoBot 3.0</td>
<td>51</td>
</tr>
<tr>
<td>Vision-Based Navigation System Development</td>
<td>55</td>
</tr>
<tr>
<td>1. Row Detection</td>
<td>55</td>
</tr>
<tr>
<td>2. Mapping</td>
<td>61</td>
</tr>
<tr>
<td>3. Visual odometry</td>
<td>63</td>
</tr>
</tbody>
</table>
Results and Discussions ........................................................................................................63
  1. Crop Row Detection Performance Evaluation ..............................................................63
  2. Mapping and Visual Odometry Performance Evaluation ..............................................71
Conclusions ..........................................................................................................................75
Acknowledgment ................................................................................................................76
References .............................................................................................................................77

CHAPTER 4. ROBOTIC IN-FIELD NAVIGATION BASED ON CROP ROW
DETECTION USING DISCRETE FOURIER TRANSFORM (DFT) ...............................81
 Abstract .............................................................................................................................81
Introduction ..........................................................................................................................82
Robot Platform .....................................................................................................................85
  PhenoBot 3.0 .......................................................................................................................85
  Navigation System ............................................................................................................86
Crop Row Detection and Navigation ..................................................................................89
  Image Preprocessing .........................................................................................................90
  DFT-based Crop Row Detection ......................................................................................92
  Linear Quadratic Gaussian (LQG) Steering Control ..........................................................94
Experimental Evaluation ......................................................................................................96
  Row Detection Algorithm Evaluation ..............................................................................96
  Navigation System Evaluation .........................................................................................103
Conclusion ...........................................................................................................................106
Acknowledgment: ..............................................................................................................107
Appendix: ............................................................................................................................107
  Solving Homography Matrix ............................................................................................107
  2D Fourier Transformation and the Limitations: ..............................................................108
References ...........................................................................................................................110

CHAPTER 5. GENERAL CONCLUSION .................................................................113
 Conclusions .........................................................................................................................113
Future Works .......................................................................................................................115
References ...........................................................................................................................116
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-1</td>
<td>Phenobot 3.0</td>
<td>7</td>
</tr>
<tr>
<td>2-1</td>
<td>Phenobot 3.0</td>
<td>18</td>
</tr>
<tr>
<td>2-2</td>
<td>The diagram of the navigation control system of Phenobot 3.0</td>
<td>20</td>
</tr>
<tr>
<td>2-3</td>
<td>Kinematic modeling of the Phenobot 3.0 rover</td>
<td>23</td>
</tr>
<tr>
<td>2-4</td>
<td>The linear regression scheme for orientation estimation</td>
<td>25</td>
</tr>
<tr>
<td>2-5</td>
<td>The sensor fusion diagram for the state estimation of Phenobot 3.0</td>
<td>26</td>
</tr>
<tr>
<td>2-6</td>
<td>The ROS node network of the Phenobot 3.0 control system</td>
<td>32</td>
</tr>
<tr>
<td>2-7</td>
<td>The simulation environment in Gazebo for Phenobot development</td>
<td>34</td>
</tr>
<tr>
<td>2-8</td>
<td>The definition of the path and the robot initial pose for the evaluation of</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>the proposed navigation system</td>
<td></td>
</tr>
<tr>
<td>2-9</td>
<td>Example robot trajectories from different path tracking algorithms (left)</td>
<td>38</td>
</tr>
<tr>
<td></td>
<td>and the corresponding path tracking lateral position error for different</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sections (right)</td>
<td></td>
</tr>
<tr>
<td>2-10</td>
<td>The path tracking results of the example field tests using the proposed</td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>navigation system with LQR (left) and PPC (right) path tracking controllers.</td>
<td></td>
</tr>
<tr>
<td>3-1</td>
<td>Images for field-based navigation</td>
<td>48</td>
</tr>
<tr>
<td>3-2</td>
<td>Phenobot 3.0</td>
<td>52</td>
</tr>
<tr>
<td>3-3</td>
<td>The architecture of the Phenobot 3.0 navigation system</td>
<td>54</td>
</tr>
<tr>
<td>3-4</td>
<td>Navigation sensor placement on Phenobot 3.0</td>
<td>55</td>
</tr>
<tr>
<td>3-5</td>
<td>Example raw IR intensity image (left) and depth image (right) from the ToF</td>
<td>57</td>
</tr>
<tr>
<td></td>
<td>camera</td>
<td></td>
</tr>
<tr>
<td>3-6</td>
<td>The slicing-based clustering algorithm to enhance the structural features of</td>
<td>60</td>
</tr>
<tr>
<td></td>
<td>corn plants</td>
<td></td>
</tr>
<tr>
<td>3-7</td>
<td>The map and the mapping area</td>
<td>62</td>
</tr>
<tr>
<td>3-8</td>
<td>The illustration of the evaluation metrics definition</td>
<td>65</td>
</tr>
<tr>
<td>3-9</td>
<td>Sample images of representative step results</td>
<td>66</td>
</tr>
<tr>
<td>3-10</td>
<td>The ground detection performance on the 30 DAT and 80 DAT datasets</td>
<td>69</td>
</tr>
</tbody>
</table>
Figure 3-11. Row detection performance on the 30 DAT and 80 DAT datasets.................70

Figure 3-12. The mapping results of the crop rows and the ground truth positions of the example field tests of corn (top) and sorghum (bottom).........................72

Figure 3-13. The distribution of the mapping position error for the corn (left) and sorghum (right) field tests. Unit: m..........................................................73

Figure 3-14. The visual odometry results and the robot position ground truth of the example field tests of corn (top) and sorghum (bottom)........................73

Figure 3-15. The distribution of the visual odometry position error for the corn (left) and sorghum (right) field tests. Unit: m..................................................74

Figure 3-16. The distribution of the visual odometry direction error for the corn (left) and sorghum (right) field tests. Unit: rad.......................................................74

Figure 4-1. Typical camera setup for in-field robot navigation and an example image from a top navigation camera .................................................................84

Figure 4-2. Phenobot 3.0........................................................................................................86

Figure 4-3. Illustration of the sensor placement and the field of view of the camera relative to the robot navigation reference position Probot.......................87

Figure 4-4. Software architecture of the vision-based navigation control system........89

Figure 4-5. The illustration of the Homography transform between the image and the map, the ROI extraction, and the tracking deviation......................................91

Figure 4-6. Frequency spectrum of a rectangular window and a Hanning window........93

Figure 4-7. Passes used for row detection algorithm evaluation and navigation system evaluation.................................................................97

Figure 4-8. Step results of the proposed DFT-based row detection algorithm.................100

Figure 4-9. Tracking deviation estimation of example passes.........................................102

Figure 4-10. Trajectories of the robot using the proposed navigation system...............104

Figure 4-11. The distribution of the tracking error (lateral position), unit: m.................105

Figure 4-12. Demo of spectral leakage effect..................................................................110
LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1.</td>
<td>Parameters of the kinematic model of PhenoBot 3.0</td>
<td>23</td>
</tr>
<tr>
<td>2-2.</td>
<td>The mean absolute tracking error (MATE) of different algorithms following paths sections with different shapes. Unit: cm.</td>
<td>38</td>
</tr>
<tr>
<td>3-1.</td>
<td>Crop row detection performance.</td>
<td>66</td>
</tr>
<tr>
<td>4-1.</td>
<td>The correlation index and rooted mean square error (RMSE) of the lateral and angular deviation estimation using the proposed row detection algorithm.</td>
<td>101</td>
</tr>
<tr>
<td>4-2.</td>
<td>Equal mean tests and results of the RMSE’s of the lateral position and heading deviation estimation by different factors.</td>
<td>103</td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENT

First of all, I should give my sincere appreciation to Dr. Lie Tang, my major professor, for his constant support, instruction, encouragement, and inspiration. He has provided sound advice guiding through my PhD study, including but not limited to literature reviewing, research experimenting, paper writing, and research presenting. Without his illuminating guidance, this thesis could not have reached its present form. Dr. Tang also has encouraged me to believe in myself and build confidence. His astounding knowledge, passion, and diligence inspire me to pursue research as my lifetime career. And he has set an excellent model as an advisor for me to follow in my future. I truly cherish the opportunity to work with Dr. Tang and to be his student.

Secondly, my hearty thanks go to my dissertation committee members: Dr. Stuart Birrell, Dr. Kathleen Delate, Dr. Liang Dong, and Dr. Brian L. Steward, for their encouragement, advice, and comments to widen my research from various perspectives.

I would also like to thank my labmates, Dr. Yin Bao, Hang Lu, Dylan Shah, Taylor Tuel, Yang Zhao and Lirong Xiang, for their collaborative work, helpful discussions, and supportive company in the lab. In addition, I would also like to thank my friends, colleagues, the department faculties and staffs for making my time at Iowa State University a wonderful experience.

Special thanks go to my beloved family. I would thank my parents for their unconditional love and support all through these years. Thanks to my dear wife Yuanfen Chen and cherished son Eric Gai, who are always here to support me, and bring myriad of joy to my PhD journey.
ABSTRACT

Field-based phenotyping heavily relies on infield manual measurement, which is labor-intensive, repetitive, and time-consuming. With the rapid advancements of robotic technology, automated in-field phenotyping technologies can significantly increase data throughput and reduce labor demand. A robotic mobile platform PhenoBot 3.0 was designed by our research group to traverse between crop rows and acquire phenotypic data automatically. However, the field-based navigation control is a critical and challenging task due to the complex and unstructured/semi-structured environment in the field. This dissertation documents our investigation of a field-based navigation control system for an agricultural field robotic vehicle. Different functional modules were developed and implemented for the system, including the motion control module based on robot kinetic model, the robot localization module using a single RTK-GPS receiver, the path tracking module running different tracking algorithms, and the computer vision-based row mapping and in-field localization module using different sensor setups.

Path tracking based on GPS localization is the most common navigation strategy for agricultural robotic vehicles. Three specific path tracking algorithms including Linear-Quadratic Regulator (LQR), Pure Pursuit control (PPC) and Timed Elastic Band (TEB) were implemented. The performance of the proposed navigation control systems were assessed on our PhenoBot 3.0 platform under both simulated and real field conditions. Satisfactory accuracies in terms of the mean absolute tracking error (MATE) were achieved while running the LQR controller on our proposed navigation control system in both simulation and field tests. The results showed the proposed navigation control system is capable of guiding the PhenoBot 3.0 robot to follow predefined paths to traverse between crop rows on uneven terrain.
For situations where global localization is denied or a pre-defined path is not available, computer vision was applied to detect the crop rows in order to locate the robot, create field maps, and navigate the robot through row-guidance. A vision-based system using a Time-of-Flight (ToF) camera was developed for under-canopy navigation, specifically for crop row mapping and robot localization under canopies of the crop rows. The potential and limitations of using ToF cameras for under-canopy navigation were investigated through field tests.

Since the agronomically-spaced crop rows are well-constructed in parallel and lend to unique features in the frequency domain, Discrete Fourier Transform (DFT) can be potentially used to solve crop row detection problems of robot navigation in agriculture. A novel image processing pipeline was developed to detect crop rows from top-view color images using frequency domain analysis. A Linear Quadratic Gaussian (LQG) controller was used with the proposed algorithm for robot navigation between crop rows. The field tests showed that the proposed crop row detection algorithm was capable of detecting crop rows with plants at different growth stages and under variable illumination conditions; and the algorithm is feasible for navigation control using a row-tracking strategy.
CHAPTER 1. GENERAL INTRODUCTION

Background

The world population was estimated to exceed 9 billion by the year of 2050 (United Nations Department of Economics and Social Affairs, 2019), and crop production needs to double to feed the rising global population (Tilman, Balzer, Hill, & Befort, 2011). Besides, food security and climate change are becoming a crucial challenge for food production. This growing agricultural crisis must be tackled with from many different strategies for boosting crop yield in a sustainable way. Improvement of crop plant genetics is one of the most effective ways to increase crop yield potential (Duvick, 2005; Vermerris et al., 2007). Genetic improvement explores crosses between different varieties of the plants, under different environments, and selects the best progeny based on the plant phenotypes. High-throughput genotyping technologies which analyze genetic information are rapidly advancing and have greatly improved the efficiency of molecular breeding in the last few decades (Appleby, Edwards, & Batley, 2009; Arif et al., 2010). However, field-based plant phenotyping, which helps to identify important genes and evaluate new crop genotypes, still heavily relies on manual measurement and scouting in crop fields. It is still a labor-intensive and time-consuming work and prone to human errors. As the current phenotypic data collected lack spatial and temporal resolutions as well as precision, and the massive genomic data acquired from high-throughput genotyping data has not been fully utilized for crop improvement.

High-throughput phenotyping (HTPP) is highly demanded in measuring the response of crops for plant breeding. Compared with phenotyping with controlled environments (chambers and greenhouses), the field-based phenotyping usually has a larger scale and is more challenging due to the uncontrolled environment conditions.
Review of the State of the Art of Field-based HTPP

During the last decade, various field-based HTPP platforms were developed, which can be classified into two categories: aerial-based platforms and ground-based platforms. Each platform is capable of collecting different types of phenotypic data, which depends on the sensors, view ports, and field conditions. All platforms have their own advantages and limitations.

Aerial-based phenotyping platforms use satellites, aircrafts and unmanned aerial vehicles as sensor carriers. Most of them use remote sensing technologies which use sensors such as RGB color cameras (Bendig et al., 2014), multi-spectral cameras (Potgieter et al., 2017), hyper-spectral cameras (Xie, Yang, & He, 2017) and thermal cameras (Santesteban et al., 2017) for field-level phenotyping (Xie & Yang, 2020). These platforms are capable of covering a large field area in a short time. But due to the limited spatial resolution of the sensors, and the limitations imposed by the top-view images, this approach is unable to resolve individual plant level and plant organ level phenotyping details.

Ground-based phenotyping platforms offer more possibilities of sensor viewing angles and sensor selections compared with aerial-based platforms. For short crop plants, the majority of the ground-based platforms still employ a top-view imaging strategy, which is efficient and effective in collecting plot-level phenotypic data, such as the normalized difference vegetation index (NDVI), leaf area index (LAI) and the temperature of leaves. However, for some plant-level and organ-level traits, top-view imaging is not capable of capturing the needed information due to occlusion. Thus, side-view proximal imaging is required for solving the occlusion problem when using top-view imaging. That is especially true for tall-growing plants such as corn and sorghum. Side-view imaging was used in revealing traits of tall growing plants such as plant height (Busemeyer et al., 2013), biomass
Various ground-based platforms have been developed for the high-throughput phenotyping of tall growing plants. These ground-based platforms can be further categorized into stationary platforms and mobile platforms. The stationary platforms are essentially infrastructures in the field which allow certain degrees of sensor mobility. Some large scale HTPP platforms were built as gantry systems and cable-suspended systems. For instance, the Field Scanalyzer is based on a railed gantry with a lifting arm, which can move the sensors along three axis (XYZ) (Virlet, Sabermanesh, Sadeghi-Tehran, & Hawkesford, 2017). The NU-Spidercam and FIP are two cable-suspend systems which control the sensor position using cables from four poles at the corners of the imaging area (Bai et al., 2019; Kirchgessner et al., 2017). There are also platforms with sensors fixed on sensor poles (Reynolds et al., 2019). The main advantages of stationary platforms are the ability to carry heavy sensors, and the robustness against different weathers due to the stationary infrastructure. However, the stationary systems are unlikely to be relocated, and are not suitable in large-scale research programs due to the extremely high cost.

Mobile platforms are able to travel in the field and move the sensors to capture the regions of interest. In general, the developed mobile HTPP platforms can be classified into three categories: pushcart-based, agricultural vehicle-based, and robot-based. Pushcart-based platform is the simplest form of a mobile HTPP platform, which have simple structure and are manually propelled. Some platforms were designed as wheeled gantries and straddle single or multiple rows. Theses platforms were usually used for the proximal sensing of short crop plants such as cotton (Thompson et al., 2018), peanut (Yuan, Bennett, Wang, & Chamberlin, 2019), wheat (Bai, Ge, Hussain, Baenziger, & Graef, 2016) and early stage corn (Nakarmi & Tang, 2014). Some platforms have narrower body for operating in narrow alleys.
(Tang & Tian, 2008; Thompson et al., 2018), but they are usually hard to maneuver without collision in crop rows.

Agricultural vehicle with high ground clearance (e.g., high clearance tractors) are able to carry heavy payload and sustain long operation hours, and usually serve as convenient HTPP platforms. Top-viewing sensors were commonly used which are mounted on a boom or a frame attached to the agricultural vehicle or an implement for the phenotypic data acquisition of short plants such as wheat (Barker et al., 2016), cotton (Jiang et al., 2018), early stage corn (Peshlov, Nakarmi, Baldwin, Essner, & French, 2017) and sorghum (Wang, Singh, Marla, Morris, & Poland, 2018).

Since agricultural vehicles with higher clearance requires larger footprint considering the stability of the vehicle, for tall-growing crop plants such as grown-up corn and sorghum, smaller vehicles such as utility tractors are suitable to operate between crop rows. Cameras were mounted on a frame for side-view imaging at multiple heights to extract traits such as the plant height, stem diameter and LAI (Bao et al., 2019). Most agricultural vehicle-based phenotyping platforms were semi-automated, which typically requires a driver to operate the vehicle, and some studies integrated auto-steering modules to enable fully automated HTPP (Bao et al., 2019). However, one limitation is the need of vehicle passes in the field due to the large footprint of the vehicle, and another limitation is the frequent traffic may cause soil compaction and soil erosion due to the heavy weight of the vehicle (Capello, Biddoccu, Ferraris, & Cavallo, 2019).

Mobile robot-based HTPP platforms are designed for fully automatic data acquisition without human interventions, which could dramatically reduce the labor cost in highly repetitive tasks of phenotypic data acquisition. Due to the superior autonomy and portability compared to stationary and large agricultural vehicle-based phenotyping platforms, mobile ground robots have greater potential to become a widely adopted tool for field-based HTPP.
Some robots straddle one or multiple rows while operating. BoniRob (Klose, Möller, Vielstädte, & Ruckelshausen, 2010; Ruckelshausen et al., 2009) was designed as a hydraulically powered high-clearance four-wheel-steering rover for corn and wheat plant phenotyping. Ladybird (Underwood et al., 2015) and Thorvald II (Grimstad & From, 2017) also employs a four-wheel steering design and were electrically powered. These large robotic platforms are like the powerful high-clearance agricultural vehicles, are able to operate on rough terrains and carry heavy payload. However, it is still challenging to use large high-clearance robots for side-view proximal sensing of tall growing plants such as corn and sorghum.

Smaller robots which feature narrow body designs and are able to traverse between crop rows are more feasible for side-view proximal sensing of tall growing plants. Sensors were placed on vertical sensor rigs or lifting mechanism for collecting phenotypic data at higher heights. Some small robotic HTPP platforms were developed based on commercial off-the shelf all-terrain UGVs as convenient bases. Vinobot (Shafiekhani, Kadam, Fritschi, & DeSouza, 2017) was developed based on a Huskey UGV (Clearpath Robotics, Ontario, Canada) for corn plant phenotyping, which installs a robot arm to carry a stereo camera for 3D reconstruction of the plants. However, since the width of the Huskey UGV was 0.67 m, the conventional corn row spacing of 0.76 m for corn field does not allow the robot to traverse through the corn rows. Hence, many research teams design their customized robotic platforms for their specific crop species and field conditions. The general challenge for robotic platforms operating between crop rows is the roll stabilization due to the narrow robot body, high sensor placement, and the uneven terrain.

Most of the customized robotic platforms operating between rows were designed as four-wheeled or tracked robots. Two steering schemes were generally employed: skid steering for four-wheel-drive or tracked robots, and Ackermann steering for car-like robots.
Young et al. (2019) developed a tracked robot with a sensor mast for bioenergy sorghum phenotyping. It has a stereo camera and a TOF camera on the mast for measuring the height and width of sorghum plants. Robotanist (Mueller-Sim, Jenkins, Abel, & Kantor, 2017) is a four-wheel-drive robot for bioenergy sorghum phenotyping, which also uses the skid steering schema. These differential drive design can reduce the complexity of control and the cost of construction, and enables zero-radius turning. However, skid steering is not as efficient as the Ackermann steering and the accuracy of tuning motion control is affected by soil parameters (Yamauchi, Nagatani, Hashimoto, & Fujino, 2017).

Robots with car-like steering mechanisms such as Ackermann steering and articulated steering are more efficient and enable narrow space operations such as between crop rows or in corridors. Rowbot (Rowbot Systems LLC, MN, USA) is a four-wheel-drive vehicle with articulated-steering mechanism. It is capable of operating between corn rows with 0.76 m spacing. However, it was designed for field operations such as nitrogen fertilization and cover crop seeding, which needs to carry heavier payloads and requires a higher power output than phenotyping applications. The robot was designed as a heavy and stable vehicle, with a gasoline engine as the power source. Therefore the robot is not suitable for HTPP due to the power efficiency concern and the vibrations from the gasoline engine.

The PhenoBot 3.0 (Figure 1-1) developed at Iowa State University was designed to traverse between corn rows and carry a sensor package mounted on a sensor mast, to acquire phenotypic corn and sorghum plant field data. It features a narrow body design, a central articulated steering mechanism, and driven wheels with differential gears. The telescoping sensor mast made of carbon fiber has an adjustable height between 2.1 m and 3.7 m. Additionally, the roll angle of the sensor mast was actively controlled to maintain a vertical orientation in the presence of uneven ground surface. Multiple PhenoStereo cameras were mounted on the sensor mast to acquire close-range, side-view stereo images of the two rows
of plants. The motorized telescoping mast and the multi-sensor configuration enable the robot to simultaneously image plant sections at different heights. Various organ-level traits such as brace roots, stalks, ears, leaf angles, and tassels/panicles can be imaged for corn and sorghum plants across different growth stages. An RTK-GPS receiver was mounted on the top of the sensor mast to avoid potential occlusion of satellite signal by the tall plants. During the mechanical design process, different design requirements including structural strength, parts machinability, and Ingress Protection (IP) rating were carefully evaluated so that the manufacturing process can be easily scaled up to produce multiple units, and meet the robustness requirement to operate in the field.

![Figure 1](image1.png)

Figure 1-1. Phenobot 3.0 is articulated with a front and back section, and an actively controlled vertical sensor mast for carrying sensors between corn plants.

**Research Objectives**

Robotic ground vehicles like PhenoBot 3.0 offer a promising solution for high-throughput field-based phenotyping, but the challenges for the navigation control system imposed by the variable working environments, tall-growing plants, and narrow row spacing must be addressed. Many commercialized field robots employ Global Navigation Satellite Systems (GNSS) data as a global positioning source, and operate with a field map surveyed through aerial imaging or seed maps generated by planters (Bonadies & Gadsden, 2019). But the performance of these navigation systems highly depends on the GNSS signal quality, and
the local information of the environment is not considered. Especially for light-weight autonomous robots operating in corn and sorghum fields, the GNSS signal might even be occluded by the canopies. One solution is to use computer vision-based techniques to detect crop rows and avoid potential collisions on-the-go.

Computer vision-based techniques have been explored in recent years for navigation in crop fields or orchards. Sensors such as cameras and LIDAR provide a rich source of the depth, color and texture of the local environment (Shalal, Low, McCarthy, & Hancock, 2013). The crop/tree rows present straight, parallel, and regularly spaced line patterns; and have been identified as the navigation cues. Most of reported field-based visual navigation algorithms use images acquired from cameras placed above row-crop canopies, or between trees in orchards (Bonadies & Gadsden, 2019). For navigation using top-view images, the potential of frequency domain analysis is worth exploring since the well-constructed parallel crop rows lead to salient features in the frequency domain. For small robots operating under crop canopies, the row detection is even more challenging due to the limited crop plant visibility and the noisy and cluttered sensing environment caused by weeds and leaves.

The overall goal of this research was to develop a navigation control system for robotic vehicles to accurately navigate in the crop field. The research project reported in this dissertation has two phases. The first phase was the development of a GPS-based path tracking navigation pipeline. The specific objectives were to (1) assess and compare the path tracking performance of the proposed system using different path tracking algorithms in simulated field environments and (2) verify the performance of the proposed system in field tests.

The second phase was the study of the feasibility of computer vision techniques for robot localization, row mapping, and row guidance in GPS-denied environment or when the pre-defined path is not available. Two different imaging configurations were investigated,
including a front-view Time-of-Flight (ToF) camera under crop canopies, and a top-view color camera above crop canopies. For the first study, the feasibility of using a front facing ToF camera to address the localization and mapping problem under the canopy of leafy crops with narrow row spacing, such as corn and sorghum was investigated. The specific objectives were to (1) assess the developed under-canopy ToF-camera-based crop row detection algorithm; and (2) evaluate the mapping and visual odometry performance using the proposed row-detection algorithm. For the second study, the potential of DFT and frequency domain analysis was studied to solve the problem of crop row detection through top-view imaging. The specific objectives of this study are to (1) assess the developed row detection algorithm under different field and illumination conditions, and (2) evaluate the performance of the visual guidance system which integrated the proposed row detection algorithm and an LQG controller.

**Dissertation Outline**

This dissertation comprises a compilation of three journal articles that document the studies in two strands of research in field-based navigation control system development for small robot-based phenotyping platforms: (1) the development of a GPS-based row tracking pipeline and (2) the studies of computer vision-based localization, mapping and row guidance. Chapter 1 provides a general introduction followed by the research objectives. The first article (Chapter 2) presents the development of GPS-based row tracking pipeline for robotic vehicles with central-articulated steering mechanism. The second article (Chapter 3) investigated the feasibility of using a front facing ToF camera to addresses the localization and mapping problem under the canopy. The third article (Chapter 4) details the study of frequency domain-base crop row detection for automatically row following. General conclusions are drawn in Chapter 5 along with a list of recommendations for future research.
References


CHAPTER 2. GPS-BASED NAVIGATION CONTROL FOR A CENTRAL-ARTICULATED ROBOTIC VEHICLE

A manuscript prepared for submission to Computers and Electronics in Agriculture

Jingyao Gai\textsuperscript{a}, Taylor Tuel\textsuperscript{a}, Yin Bao\textsuperscript{b}, Lirong Xiang\textsuperscript{a}, Lie Tang\textsuperscript{**}

\textsuperscript{a} Department of Agricultural and Biosystems Engineering, Iowa State University, Ames, Iowa 50011-3270, USA

\textsuperscript{b} Department of Biosystems Engineering, Auburn University, Auburn, Alabama 36849, USA

\textsuperscript{*Corresponding author. E-mail address: lietang@iastate.edu (L. Tang)}

Abstract

During the last decade, many robotic vehicles have been developed for automated plant phenotyping in agriculture. Field-based navigation control is critical and challenging due to the complex and unstructured environment. In this paper, a navigation control system was proposed for a central-articulated robotic vehicle for field-based corn plant phenotyping. Different functional modules were developed and implemented for the system, which include the motion control module based on robot kinetic model, robot localization module using single RTK-GPS signal and the path tracking module running different tracking algorithms. Three specific path tracking algorithms including Linear-Quadratic Regulator (LQR), Pure Pursuit control (PPC) and Timed Elastic Band (TEB) were implemented. The performance of the proposed navigation control system was assessed on the PhenoBot 3.0 platform under both simulated and real field conditions. Different path tracking algorithms on the PhenoBot 3.0 platform were assessed and compared. Satisfactory accuracies in terms of the mean absolute tracking error (MATE) were achieve while running the LQR controller on our proposed navigation control system in both simulation (MATE: 5.72 cm for the entire path, and 4.48 cm for the straight line sections, specifically) and field tests (MATE: 7.01 cm for the
entire path, and 5.67 cm for straight line sections). The results showed the proposed navigation control system is capable of guiding PhenoBot 3.0 to follow predefined paths to traverse between crop rows on uneven terrain.

**Introduction**

There is a strong demand for field-based high-throughput phenotyping (HTPP) technologies to measure the response of crops for plant sciences research and plant breeding. Compared with phenotyping under controlled environments (chambers and greenhouses), the field-based phenotyping is usually carried out in a larger field and more plants were involved. And it is more challenging due to the uncontrolled environments. During the last decade, robotic technologies have been evolved and were applied to various field-based HTPP applications. Mobile robot-based HTPP platforms designed for fully automated data acquisition without human interventions could dramatically reduce the labor cost in highly repetitive tasks of phenotypic data acquisition. Due to their autonomy and portability, mobile ground robots have a great potential to become a widely adopted tool for field-based HTPP.

Many robotic vehicles have been developed for field-based HTPP. Some larger robots straddle one or multiple rows while operating. For example, BoniRob (Klose, Möller, Vielstädté, & Ruckelshausen, 2010; Ruckelshausen et al., 2009) was designed as a hydraulically powered high-clearance four-wheel-steering rover for corn and wheat plant phenotyping. Ladybird (Underwood et al., 2015) and Thorvald II (Grimstad & From, 2017) also employ four-wheel steering design and were electrically powered. These large robotic platforms are able to operate on rough terrains and carry a large payload, but there are challenges to use large high-clearance robots for side-view proximal sensing of tall growing plants such as corn and sorghum.

Smaller robots which feature narrow body designs and are able to traverse between crop rows and are more feasible for side-view proximal sensing of tall growing plants.
Sensors were placed on vertical sensor rigs or lifting mechanism for collecting phenotypic data at higher height. Some small robotic HTPP platforms were developed based on commercial off-the-shelf all-terrain UGVs as convenient bases. Vinobot (Shafiekhani, Kadam, Fritschi, & DeSouza, 2017) was developed based on a Huskey UGV (Clearpath Robotics, Ontario, Canada) for corn plant phenotyping, where a robot arm was installed to carry a stereo camera head for 3D reconstruction of plants. However, due to the width of the Huskey UGV, the agronomic row spacing of 0.76 m for corn plants is not sufficient for the robot to traverse. Hence, many research teams design their customized robotic platforms for their specific crop species and field conditions. The general challenge for robotic platforms operating between crop rows is the roll stabilization due to the narrow robot body, high sensor placement and uneven terrains.

Most of the customized robotic platforms operating between rows were designed as four-wheeled or tracked robots. Two steering schemes were generally employed: skid steering for four-wheel-drive or tracked robots, and Ackermann steering for car-like robots. Young et al. (2019) developed a tracked robot with a sensor mast for bioenergy sorghum phenotyping. It has a stereo camera and a TOF camera on the mast for measuring the height and width of sorghum plants. Robotanist (Mueller-Sim, Jenkins, Abel, & Kantor, 2017) is a four-wheel-drive robot for bioenergy sorghum phenotyping, which also uses the skid steering schema. These differential drive design can reduce the complexity of control and the cost of construction, and enables zero-radius turning. However, skid steering is not efficient as the Ackermann steering and the accuracy of tuning motion control is affected by soil parameters (Yamauchi, Nagatani, Hashimoto, & Fujino, 2017).

Robots with car-like steering mechanisms such as Ackermann steering and articulated steering are more efficient and space conservative to operate in narrow space, such as between crop rows or in corridors. Rowbot (Rowbot Systems LLC, MN, USA) is a four-
wheel-drive vehicle with articulated-steering mechanism. It is capable of operating between corn rows with 0.76 m spacing. However, it is not designed for HTPP, but for nitrogen fertilization and cover crop seeding.

The ISU PhenoBot 3.0 (Figure 2-1) was designed to traverse between corn rows and carry a sensor package mounted on a sensor mast so that phenotypic data of corn plants can be acquired across fields. It features a narrow body design, a central articulated steering mechanism, and driven wheels with differential gears. The telescoping sensor mast made of carbon fiber has an adjustable height between 2.1 m and 3.7 m. Additionally, the roll angle of the sensor mast was actively controlled to maintain vertical in the presence of uneven ground surface. Multiple PhenoStereo cameras were mounted on the sensor mast to acquire close-range, side-view stereo images of the two rows of plants. The motorized telescoping mast and the multi-sensor configuration enable the robot to simultaneously image plant sections at different heights. Various organ-level traits such as brace roots, stalks, ears, leaf angles, and tassels/panicles can be imaged for corn and sorghum plants across different growth stages.

An RTK-GPS receiver was mounted on the top of the sensor mast to avoid potential occlusion of satellite signal by the tall plants. During the mechanical design process, different design requirements including structural strength, parts machinability and Ingress Protection rating were carefully evaluated so that the manufacturing process can be easily scaled up to produce multiple units, and the robot is reliable to operate in the field (Tuel, 2019).
Figure 2-1. Phenobot 3.0 is articulated with a front and back section, and an actively controlled vertical sensor mast for carrying sensors between corn plants.

Accurate navigation is a critical feature for robotic vehicle to perform agricultural field operations such as phenotyping automatically. For small robotic vehicles like PhenoBot 3.0 which operates between crop rows, high navigation accuracy is needed to avoid colliding with crop plants. While there have been considerable studies reported about the navigation control of skid-steering vehicles in the field, relatively little has focused on articulated steering vehicles. Most articulated guidance studies modeled the vehicles from a truck and trailer perspective (Karkee & Steward, 2010; Pradalier & Usher, 2008), but they are still different from articulated steering since the steering joint was not actively controlled. The navigation control of industrial articulated vehicles such as wheel loaders for underground mining was studied (Alshaer, Darabseh, & Alhanouti, 2013; Dekker, Marshall, & Larsson, 2019). But the vehicles were modeled into separate bodies, which are not compatible with most off-the-shelf navigation algorithms.

A navigation system was developed for PhenoBot 3.0 to accurately navigate the robot in the crop field. In the developed system, the robot was modeled that was compatible with most widely used path tracking and path planning algorithms. A localization scheme was proposed using a single RTK-GPS receiver. Multiple path tracking algorithms were implemented. A virtual testing environment was developed for validating the navigation
algorithms. The primary objective of this study was to evaluate the performance of the proposed navigation control system in tracking pre-defined paths under field conditions. The specific objectives were to (1) assess and compare the path tracking performance of proposed system while using different path tracking algorithms in simulated field environments and (2) verify the performance of the proposed system in field tests.

The rest of the paper is organized as follows. In section 3, the navigation system and all the functional modules are described in detail. Section 4 covers the design of both the simulation-based tests and the field-based tests. Section 5 provides the results of the path tracking performance assessment. The capability and limitations of the proposed system are discussed. Finally, the study is concluded in section 6.

**Navigation System Development**

The navigation system aims to guide the PhenoBot robot to traverse between the corn plant rows and avoid colliding with crop rows by keeping the robot at the center of two crop rows. As for the dataflow of the navigation system (Figure 2-2), data from multiple navigation sensors were fused for localizing the robot in the field map. The robot was driven by a path tracking algorithm to correct the position and heading errors relative to the central line of the crop rows. Finally, the motions of individual motors were controlled based on the kinematic model of the robotic rover. Such a localization-and-tracking strategy was employed by many robotic navigation systems (Li, Imou, Wakabayashi, & Yokoyama, 2009; Mueller-Sim et al., 2017; Young et al., 2019). This paper focuses on the navigation control without computer vision, and the vision-based navigation will be introduced in future papers.
Rover Design and Kinematic Modeling

Constrained by the narrow working space, the robotic rover was designed as an articulated vehicle. The steering was controlled by actuating the steering joint connecting the front half and the rear half of the vehicle. The rover features a narrow body design, which has a width of 20 inch (50.8 cm). Compared with common Ackermann steering, the features of articulated steering are: first, it enables the rover to perform sharper turns. Second, there is less inner- or outer-wheel off tracking, meaning the rear wheels follow the same tracks left by the front wheels if the front and rear parts have an equal length (Ishimoto, Tsubouchi, Sarata, & Yuta, 1998). In addition, compact differential gears were employed to further increase the energy efficiency and reduce the wheel slip angle, since the power flow from the drive motors were passively distributed to individual wheels (Tuel, 2019).

The development of a navigation control system demands the use of a well-defined model for robot motion control. Therefore, the vehicle modeling was carried out as the first
step. Most path tracking and planning algorithms for car-like robots requires the vehicle being modeled as a whole body instead of multiple parts, and the turning curvature was limited (Corke, 2011). A generalized kinematic model for articulated vehicles was proposed in this study (Figure 2-3), which is compatible with most control algorithms for car-like robots. The articulated vehicle body has two separate parts, namely a front section and a rear section, linked with an articulated joint for steering. Due to the differential gears for the front and rear wheels and the low travel speed, the rover model was simplified into a bicycle model and wheel slip angle was ignored.

The pose of the vehicle is represented by a reference coordinate frame \{V\}. The x-axis which represents the heading of the rover was defined by linking the centers of the front and rear axles. The y-axis passes through the instantaneous center of rotation (ICR). Therefore, origin \(V_0\) of the reference frame has an instantaneous velocity whose direction is consistent with the vehicle heading. Since the frame origin is not a fixed point on the vehicle body, the position of origin can be determined by:

\[
\frac{l_A}{l_B} = \frac{L_B}{L_A} \frac{R_A}{R_B} = \frac{L_B}{L_A} \frac{L_A \cos \gamma + L_B}{L_A + L_B \cos \gamma} = p \cdot \frac{\cos \gamma + p}{1 + p \cos \gamma}, \quad p = \frac{L_B}{L_A}
\]

(1)

where \(l_A\) and \(l_B\) are the distances from the reference frame origin to the centers of the front and rear axles, respectively. And \(L_A\) and \(L_B\) are the distances from the steering joint to the centers of the front and rear axles, respectively. While assuming the steering angle \(\gamma\) is small, equation 1 can be approximated as:

\[
\frac{l_A}{l_B} \approx p
\]

(2)

And the turning radius of the front \((R_A)\), rear \((R_B)\), and the reference frame \((R)\) can be calculated from their geometric relationship:
While assuming the steering angle $\gamma$ is small, equation 3 can be approximated as:

$$R_A \approx R_b \approx R \approx \frac{L_A + L_b}{\gamma}$$ (4)

The configuration of the vehicle is represented by the 2D generalized coordinates $q = (x, y, \theta)$, in which $x$ and $y$ are the position of the robot, and $\theta$ is the orientation in the world coordinate system. The velocity of the vehicle $v$ is by definition along the x-axis of the reference frame. While assuming the steering angle $\gamma$ is small, the speeds of the front and rear wheels are approximately equal to the vehicle velocity $v$. And the angular velocity $\dot{\theta}$ is determined by the velocity $v$ and the turning radius $R$ or the turning curvature $\kappa$ as the inverse of the turning radius.

$$v \approx v_A \approx v_B$$ (5)

$$\dot{\theta} = \frac{v}{R} = v \cdot \kappa \approx \frac{\gamma}{L_A + L_b} \cdot v$$ (6)

When the motion control module received the speed command $(v, \dot{\theta})$ from the path tracking module, it calculated the command for each individual joint using equations 5 and 6. For PhenoBot 3.0, some parameters for kinematic modeling are included in Table 2-1. Programmable motor controllers (RoboteQ FBL2360, RoboteQ, AZ, USA) were used to control the speed of the two drive motors and the angular position of the steering motor by using PID control.
Figure 2-3. Kinematic modeling of an articulated steering vehicle with unequal front and rear length.

Table 2-1. Parameters of the kinematic model of PhenoBot 3.0

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L_A$</td>
<td>Distances from the steering joint to the front axle center</td>
<td>0.276 m</td>
</tr>
<tr>
<td>$L_B$</td>
<td>Distances from the steering joint to the rear axle center</td>
<td>0.673 m</td>
</tr>
<tr>
<td>$p$</td>
<td>$L_B/L_A$</td>
<td>2.44</td>
</tr>
<tr>
<td>$\gamma_{\text{max}}$</td>
<td>The maximum steering angle</td>
<td>30°</td>
</tr>
<tr>
<td>$R_{\text{min}}$</td>
<td>The minimum turning radius of the robot</td>
<td>1.81 m</td>
</tr>
<tr>
<td>$v_{\text{rated}}$</td>
<td>Desired travel speed</td>
<td>1 m/s</td>
</tr>
</tbody>
</table>

Robot Localization

Accurate estimations of global pose (i.e., position and orientation) and motion (i.e., speed and acceleration) of the robot are critical to avoid crop damage during navigation between narrow crop rows. Many studies estimate the global position using RTK-GPS signals, and the global orientation are commonly estimated by compass sensors or differential GPS (DGPS) (Bonadies & Gadsden, 2019). However, compass sensors are not compatible with our application due to the magnetic-conducted material on the robot affects the accuracy of compass sensors. Additional GPS receivers for DGPS will increase the moment of inertia of the sensor mast, which will reduce the stability of the robot.
In this study, a localization method using a single real-time kinematic (RTK) global positioning system (GPS) receiver (Reach RS2, Emlid, Russia) was developed for both position and orientation estimation. The GPS receiver delivers position data at sub-centimeter accuracy at 10 Hz, and the receiver was mounted on the top of the sensor mast to avoid signal obstruction by the tall plants. The strategy of the estimation of global orientation \( \theta \) is by differentiating the robot global position over a short sample period (~1.5s) while the robot is moving straight (Figure 2-4), which is to fit a linear model:

\[
y(x; a, b) = a + bx
\]

(7)

in which \((x, y)\) is the global position of the robot. The robot position \( P \) was calculated by subtracting the GPS antenna position \( G \) by the position bias \( T_{PG} \) which is a function of the robot steering angle \( \gamma \), mast tilt angle \( \varphi \) and the robot orientation \( \theta \):

\[
P_i = G_i - T_{PG}(\theta, \gamma, \varphi)
\]

(8)

The mast tilt angle \( \varphi \) was tracked by an IMU (Adafruit BNO055, Adafruit, NY, USA) on the sensor mast. However, there are two challenges for this strategy. The first challenge is that the GPS-based robot position estimation is not sufficiently reliable for heading estimation due to the measurement error and synchronization error from the GPS and other sensors for posture measurement. Therefore, a robust linear regression algorithm was applied for robust orientation estimation. The absolute deviation instead of the squared deviation was minimized in this robust linear regression, which is proven to be more robust to outliers (Press, Teukolsky, Vetterling, & Flannery, 1988). The data points were rotated to coarsely align with the \( x \)-axis based on the coarse orientation estimation \( \hat{\theta} \) determined by the first and the last point \((x_1, y_1), (x_n, y_n)\) in the sample period. Then the robust linear regression was applied, which minimized the cost function:

\[
\sum_{i=1}^{n} |\hat{y}_i - a - bx_i|
\]

(9)
in which \((\hat{x}, \hat{y})\) is the robot position after aligning with x-axis, and \(n\) is the sample size within the sample period. The solution of parameter \(a\), \(b\) was calculated from the following equations:

\[
a = \text{median}\{\hat{y}_i - b\hat{x}_i\} \tag{10}
\]

\[
0 = \sum_{i=1}^{n} \hat{x}_i \text{sgn}(\hat{y}_i - a - b\hat{x}_i) \tag{11}
\]

By replacing \(a\) in equation 10 with equation 11, the \(b\) was solved by iteratively bracketing and bisection, which are numerical root-finding methods and safe for discontinue equations. Then the orientation estimation \(\tilde{\theta}\) of the robot was determined by:

\[
\tilde{\theta} = \hat{\theta} + \arctan b \tag{12}
\]

The sign of the orientation angle was determined by drive wheel rotation direction.

Figure 2-4. The linear regression scheme for orientation estimation

The second challenge is the robot global position estimation requires the orientation estimation result, since the GPS receiver position is not identical with the navigation reference frame. Thus, the pose estimation using a single GPS receiver is a recursive process and requires an initial orientation. Therefore, an initialization step was designed for initial orientation estimation. During initialization step, the robot was controlled to travel straight along an arbitrary direction. The GPS antenna position was used to calculate the initial
orientation using the algorithm above (Equations 9-12). Although the accuracy of the initial orientation is limited due to not considering the antenna position bias, the error will still converge in later robot position and orientation estimation loops.

Although the GPS is a global localization source which provides the global pose estimation, the update rate is insufficient for robot navigation control. As a local motion sensing method, an inertial measurement unit (IMU) (3DM-GX5-15, LORD Sensing MicroStrain, Williston, USA) was used to track the changes in position, speed, and heading over time in a local frame. The data from rotary encoders on the drive and steering motors along with the robot’s kinematic model were used to calculate the motion of the robot. Data from above local localization sources had high sensitivity and high update frequency. However, pose estimation through the integration of robot motion drifted over time. Thus, an Extended Kalman Filters (EKF) (Hoshiya & Saito, 1984) was applied to fuse both global and local localization sources for both high update rate and low drift over time. The redundancy of different sources also improved the accuracy and reliability of robot localization (Figure 2-5).

![Sensor fusion diagram for the state estimation of PhenoBot 3.0](image)

Figure 2-5. The sensor fusion diagram for the state estimation of PhenoBot 3.0. The visual odometry is acting as a redundant source to improve the system robustness and will be introduced in later chapters.
Another strategy for robot localization within a crop field is to use visual odometry, which employs computer vision to detect and locate crop rows in images or 3D point clouds in real-time, and inversely estimate robot position relative to crop rows. This technique applies to applications where global localization is denied or a field map is not available, and will be reported in later chapters.

Path Tracking (Steering Control)

A path tracking algorithm is responsible for determining appropriate actuation, especially the steering of the robot to follow a pre-defined path, which is to minimize the tracking error, i.e. the position and heading deviations. Many algorithms have been reported for robotic vehicles operating in the field, and most steering control algorithms can be classified into two categories: feedback control and feedforward control. Most feedback control methods use the tracking error at current and previous timestamp to control the steering motion, such as the widely used proportional-integral-derivative (PID) control (Malavazi, Guyonneau, Fasquel, Lagrange, & Mercier, 2018; Underwood, Wendel, Schofield, McMurray, & Kimber, 2017), linear-quadratic-regulator (LQR) control (Karkee & Steward, 2010) and Fuzzy logic control (Xue, Zhang, & Grift, 2012). These algorithms are generally more robust to inaccurate modeling and unpredicted disturbances compared with feedforward control methods. However, the steering motion is carried out only after the tracking error has been detected. Therefore, lag error and sometime undesired fluctuations and/or instability may occur (Mokhatab & Poe, 2012).

Feedforward control relies on the prior knowledge of the system. The control system gives the corresponding adjustments based on the predication of the dynamic response of a system. Pure pursuit control (PPC) is a simple and effective feedforward control algorithm, mainly for nonholonomic vehicles. The steering motion was controlled based on the tracking error at a certain distance ahead of the robot. The algorithm was successfully implemented
and validated in many applications (Mueller-Sim et al., 2017; Rains, Faircloth, Thai, & Raper, 2014; W. Zhang et al., 2019).

Model Predictive Control (MPC) refers to a set of optimization-based feedforward control algorithms, which are more computationally demanding compared to the algorithms mentioned above. The basic concept of MPC is to use a system model to forecast system behavior and optimize the forecast to find the best control decision at the current time (Rawlings, Mayne, Diehl, & Barbara, 2009). MPC including its varieties were implemented on various autonomous agricultural vehicles and achieved good performance during tracking both straight paths and complex paths (Backman, Oksanen, & Visala, 2012; Kayacan, Young, Peschel, & Chowdhary, 2018; Utstumo, Berge, & Gravdahl, 2015; Z. Zhang, Kayacan, Thompson, & Chowdhary, 2020). Timed Elastic Band (TEB) (Rösmann, Hoffmann, & Bertram, 2017) is a variety of MPC which is capable of planning collision-free or space-optimal paths online by including a collision or space term in the optimization cost function. These optimization-based algorithms are suitable for critical scenarios during navigation (e.g., head-landing turning) to avoid collisions and be adaptive to limited space.

The navigation system of PhenoBot 3.0 adopted both the feedback and feedforward control algorithms for path following. The algorithms include the LQR (Corke, 2011), PPC (Coulter, 1992), and the TEB (Rosmann, Hoffmann, & Bertram, 2017).

**LQR control**

The LQR controller was designed based on the general path tracking error model for car-like vehicles. The discrete system model is defined as:

\[
\begin{align*}
x_{n+1} &= Ax_n + Bu_n \\
y_n &= Cx_n
\end{align*}
\]

where the states \(x\) of the system were defined as the lateral position error \(e_y\), the heading error \(e_{\theta}\) and their rates. The input \(u\) is the turning curvature \(\kappa\) of the robot, controlled by
actuating the steering joint angle. The measurement $\mathbf{y}$ are the measured lateral position error $\hat{e}_y$, and the heading error $\hat{e}_\theta$ using the localization results.

$$\mathbf{x} = (e_y, e_y', e_\theta, e_\theta')$$
$$\mathbf{u} = (\kappa)$$
$$\mathbf{y} = (\hat{e}_y, \hat{e}_\theta)$$

Based on the kinematic model of a car-like vehicle, the system model (Equation 13) becomes:

$$\begin{bmatrix} e_y \\ e_y' \\ e_\theta \\ e_\theta' \end{bmatrix}_{n+1} = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 0 & v & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_y \\ e_y' \\ e_\theta \\ e_\theta' \end{bmatrix}_n + \begin{bmatrix} 0 \\ 0 \\ \kappa \\ v \end{bmatrix}$$

in which $v$ is the separately controlled forward speed, and $\Delta t$ is the control cycle time. In this study, the controller operated at 10 hz, thus $\Delta t$ is 100 ms.

**Pure pursuit control**

The design of the pure pursuit controller follows the study of Zhang et al. (2019). The controller calculates the future position errors $\hat{e}_y$ by comparing with a set point in the path at a certain distance $l$ ahead of the robot. And then the algorithm calculates an arc path for the robot to join the path at the set point. The curvature $\kappa$ was calculated by:

$$\kappa = \frac{2e_y}{l^2}$$

The maximum curvature of the path was constrained by the maximum steering angle $\gamma_{max}$. In this study, the controller operated at 10 hz.
Timed-elastic-band (TEB) control

The TEB controller (Rösmann et al., 2017) is a variety of MPC, and the path tracking problem was formulated as finding an optimal sequence of robot poses $q_k = (x_k, y_k, \theta_k)$ in the future and their corresponding time intervals $\Delta T_k$ by solving an optimization problem:

$$b^* = \arg \min \mathcal{V}(b)$$

$$b = [q_1, \Delta T_1, q_2, \Delta T_2, ..., q_{n-1}, \Delta T_{n-1}, q_n]^T$$

$$\mathcal{V}(b) = \sum_{k=1}^{n-1} \left[ \Delta T_k^2 + \phi(h_k, \sigma_h) + \phi(r_k, \sigma_r) + \phi(s_k, \sigma_s) + \chi(a_k, \sigma_a) + ... + \chi(v_k, \sigma_v) + \chi(o_k, \sigma_o) \right] + \chi(p_a, \sigma_p)$$

in which $n$ is the prediction horizon, which represents the number of future poses to predict by the algorithm. The cost function (Equation 19) includes the penalty terms relative to the total transition time $T$, the kinematic constraints $h$, the minimum turning radius $r$, the steering joint angular velocity $s$, the velocity $v$, acceleration limits of the robot $a$, the obstacles $o$, and the deviation to the followed path $p$. Different penalty terms were weighted by separate weight factors. Since the navigation task is path following in this study, the obstacle term was set to low values, and the weight of the path tracking deviation was set to a high value. The algorithm outputs both the speed and the angular speed command. In this study, the prediction horizon $n$ was set to 5, and the controller operated at 5 hz.

System Implementation and Simulation

ROS-based control system architecture:

For the navigation control development process, ROS (Robot Operating System, Kinetics version) was used as a middleware to couple different functional modules in the control system. ROS is an open-source middleware that provides a framework for connecting many different software components (i.e., ROS nodes) of a complex robotic application. ROS manages a graph-like peer-to-peer network of processes that can be distributed across multiple machines. The processes are loosely coupled using the ROS communication...
protocols. Different communication protocols were provided for different requirements, including synchronous request/response communication over “service”, asynchronous data streaming over “topics”, and shared parameter storage named “parameter server” (Koubâa, 2019).

The navigation control system of PhenoBot 3.0 was implemented using ROS. The functional modules were implemented as ROS nodes, including the motion control, robot localization and path tracking nodes. With the aforementioned modules and other ROS-provided supportive modules such as TF (Foote, 2013) for managing the transformation between different coordinate frames, a “graph” of ROS nodes for robot control was established (Figure 2-6).

**Robot simulation**

Simulation technology can provide frameworks to test and evaluate the functionality and performance of the developed systems in dynamic scenarios, which accelerates the development process. Various types of simulation software tools such as Gazebo (“Gazebo,” n.d.), V-REP (Rohmer, Singh, & Freese, 2013) and ARGos (Pinciroli et al., 2012) are available and were proven capable of simulating field environments for field robotic system development.

The Gazebo simulator is widely used for robot simulation as it easily interfaces with ROS, which enables the development of software in the context of testing the developed code with a simulated robot and a virtual environment. Gazebo features a time-efficient physics engine to simulate the interaction between the robot and the environment. The descriptions of the robot and the environment are input through an XML file which contains the physical properties and geometry parameters that describe the bodies of the robot and the objects in the environment.
Figure 2-6. The ROS node network of the PhenoBot 3.0 control system (in the large rectangle named “robot”). The ellipses represent ROS nodes and the rectangles the ROS message topics for data interchanges between nodes.
When using Gazebo with ROS, Gazebo loads a series of ROS plugins, which turns Gazebo into a ROS node. The plugins provide message and service publishers for interfacing Gazebo with ROS. The ROS plugins implement functions to simulate force, motion and perception sensors, actuate motors, and dynamically re-configure parameters in the simulation using the Gazebo API. Once the rest of the ROS nodes for robot control were developed and tuned in the simulation, they can be migrated into the real-world robot directly or with minor modifications.

In this application, Gazebo was used to simulate PhenoBot 3.0 in a virtual corn field for debugging and validating the functionality of the control algorithms. To reduce the complexity of the model, the rover with differential gears and the articulated body design was simplified to a bicycle model, which composed of a front wheel, a rear wheel, and an articulated steering joint between the wheels. The navigation sensors introduced above were included in the navigation system and simulated in the Gazebo software. In the virtual environment, a realistic corn plant model was duplicated into a plot, a crop row and the entire corn field. The virtual ground was made up of an uneven surface, which aimed to simulate the uneven terrain in real-world soil conditions (Figure 2-7).
Figure 2-7. The simulation environment in Gazebo for PhenoBot development. The robot pose, the map, the planned trajectory and the sensor output are displayed on the Rviz-based control panel on the bottom right.

Experiments

Experiments were conducted to demonstrate the capabilities of the proposed navigation pipeline and individual path tracking algorithms. The developed navigation system was tested in the simulated environments and under real field conditions.

Path Tracking Algorithms Evaluation in Simulation

The experiments in simulation were designed to evaluate the performance of different path tracking algorithms within the proposed navigation control system, specifically (1) compare the response characteristics of different algorithms, including the LQR, the Pure pursuit, and the TEB control, and (2) evaluate and compare performance of different path tracking algorithms in tracking paths with different shapes. The PhenoBot 3.0 virtual model was used as a testing platform, and the experiment was conducted on simulated uneven terrain without corn plants in Gazebo. The path to follow during the experiment was defined as a series of cubic splines (Figure 2-8). The path is composed of straight path sections (1, 3, 5, 7, 9, 11) as the centerlines
of the crop rows, c-shape turn path section (2, 6, 8, 12) and bulb turn path sections (4, 10) for the transition between crop rows. These path sections are the common elements in paths for field-based operations. Adjacent straight paths have an equal spacing of 0.76m, which is the standard corn crop row spacing in North America. And the curvatures of the turning paths were defined according to the limit of the steering joint of PhenoBot 3.0. The total length of the path was about 150 m.

To compare the response characteristics of different path tracking algorithms, the starting pose of the robot was set parallel to the start of the path section 1 with a lateral position bias of 2.15m as a step input (Figure 2-8). The rise time $T_r$ and the settling time $T_s$ are two metrics for evaluating path tracking algorithms. However, since the velocity control was different for each algorithm, and the robot trajectory is more important than the time response, the response was evaluated using metrics with respect to the traveled distance instead of time. The metrics include the rise distance $D_r$ which is the traveled distance to reduce the lateral position error $e_y$ to 10% of the original, and the settling distance $D_s$ which is the traveled distance for $e_y$ to stay within 5%.

For performance evaluation of different path tracking algorithms in tracking paths with different shapes, the distributions of the lateral position error $e_y$ for three type of sections (excluding path section 1) were measured and analyzed. The means of the absolute of the lateral tracking position error (MATE) were calculated. The experiments were repeated three times so the metrics were averaged and compared among different algorithms.
Figure 2-8. The definition of the path and the robot initial pose for the evaluation of the proposed navigation system. Different straight lines and wide/sharp turning sections are labeled with numbers indicating the sequence of the followed path sections.

Navigation System Evaluation in Field Tests

Field tests were conducted to evaluate the functionality of the proposed navigation system on the developed PhenoBot 3.0 robotic platform under field conditions. The experiment site was at Iowa State University Agricultural Engineering and Agronomy Research Farm located in Boone, Iowa, and the experiments were conducted in 2019 and 2020. The robot was operated on a patch of grass field with compacted and uneven terrain, and the path to follow was the same path defined in the simulation test (Figure 2-8). The path tracking lateral position error $e_y$ was measured for different tracking algorithms by comparing the GPS-based localization with the referenced path. The tracking errors were compared with the simulation results. Only the LQR and the PPC controller were tested.
Results and Discussions

Tracking Algorithms Evaluation in Simulation

One example path tracking trajectory and the tracking error for each control algorithm was displayed in Figure 2-9. As the response characteristics, the rise distance $D_r$ was 5.0 m, 4.4 m, and 6.2 m for LQR, PPC and TEB control algorithms, respectively. And the settling distance $D_s$ was 7.4 m, 9.3 m, and >12 m (failed to correct the steady state error within the first section of the path) for LQR, PPC and TEB control algorithms, respectively. The MATEs of different algorithms for following paths with different shapes were listed in Table 2-2.

According to the response characteristics of the tracking algorithms, with the proposed navigation system and the simulated environment, the TEB control algorithm has the largest $D_r$ and $D_s$. That was because there is no penalty term for the steady-state error in the cost function (Equation 19); therefore, the robot was actuated to stay close to the reference path, but the tracking error was not fully corrected. In addition, the steering joint angular velocity term regulates the steering motion, which leads to a long rise distance. Compared with LQR, the PPC controller has a smaller $D_r$ and a larger $D_s$. That was because the control rule of PPC (Equation 16) has a similarity in form to a proportional controller, which generally has shorter rise time, but longer settling time, and higher overshoot when compared with PID and LQR controllers.
Figure 2.9. Example robot trajectories from different path tracking algorithms (left) and the corresponding path tracking lateral position error for different sections (right).

Table 2-2. The mean absolute tracking error (MATE) of different algorithms following paths sections with different shapes. Unit: cm.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Straight (3, 5, 7, 9, 11)</th>
<th>C-shape turn (2, 6, 8, 12)</th>
<th>Bulb turn (4, 10)</th>
<th>Average (2-12)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LQR</td>
<td>4.84</td>
<td>8.59</td>
<td>4.78</td>
<td>5.72</td>
</tr>
<tr>
<td>PPC</td>
<td>5.46</td>
<td>13.96</td>
<td>14.35</td>
<td>9.31</td>
</tr>
<tr>
<td>TEB</td>
<td>11.21</td>
<td>18.31</td>
<td>12.82</td>
<td>14.22</td>
</tr>
</tbody>
</table>

According to the tracking analysis of the tracking algorithms, the LQG controller had the minimum tracking error among all tested algorithms in following path sections of all shapes, and
the average tracking error was 5.72 cm. The PPC controller had an average tracking error of 9.31 cm. PPC had a poorer performance during tracking c-shape and bulb turn paths. The tracking error was low during tracking straight path, but it suffered from the overshoot problem during the transition from tracking curved paths. The TEB has an average tracking error of 14.22 cm. The algorithm was sensitive to localization noise and minor tracking deviation, which led to low accuracy in tracking straight and c-shape turn paths. However, the performance of TEB was still improvable by including the penalty of steady state error in the cost function and tuning the weight parameters, or including feedback control (Bai et al., 2019).

In general, the comparison of different algorithms in terms of response characteristics and the tracking error indicates the capability of the navigation system for path tracking, and the LQR controller has the best performance among the tests control algorithms.

**Navigation System Evaluation in Field Tests:**

The example path tracking of the PhenoBot 3.0 with LQR and PPC path tracking controllers were shown in Figure 2-10. The robot control system ran on an industrial computer with an Intel i5-6500TE processor. During the tests, and the MATEs for tracking straight paths were 5.67 cm and 8.20 cm using LQR and PPC control, respectively; and the average MATEs for tracking the entire path were 7.01 cm and 10.65 cm using LQR and PPC control, respectively.

The testing results indicate that PhenoBot 3.0 robot was able to follow predefined paths with the proposed navigation control system. And the LQR controller has better overall performance than the PPC controller in terms of the tracking accuracy. The tracking error level during field tests is slightly higher than the simulation, which is mainly caused by the localization error due to the combined error sources from the GPS and the posture sensors, as
well as the imperfect synchronization of the sensors. The LQR is more suitable for field-based path tracking to prevent crop damage. The PPC controller still encountered the overshooting problem and may damage crop rows while the robot is entering the crop rows.

![Figure 2-10. The path tracking results of the example field tests using the proposed navigation system with LQR (left) and PPC (right) path tracking controllers.](image)

**Conclusions**

In this paper, a navigation control system was proposed for a central-articulated robotic vehicle for field-based corn phenotyping. Different functional modules were developed and implemented for the system, which include the motion control module based on robot kinetic model, robot localization module using single RTK-GPS signal and the path tracking module with multiple tracking algorithms implemented. A Gazebo-simulated environment was constructed to aid the system development and test the performance of different path tracking algorithms. Three specific path tracking algorithms including Linear-Quadratic Regulator (LQR), Pure Pursuit control (PPC) and Timed Elastic Band (TEB) were implemented for the path tracking module. The performance of different path tracking algorithms on our PhenoBot 3.0 platform was assessed and compared in both simulated environments and in field conditions.

Both the simulation-based and field-based experiments demonstrated the capability of the proposed navigation control system in guiding the PhenoBot 3.0 robot to follow predefined paths
on uneven terrain in the field. The system achieved the best path tracking performance with the LQR steering control algorithm in both simulation and field environments. The mean absolute tracking errors (MATE) of tracking a testing path composed of straight lines and curves were 5.72 cm and 7.01 cm in simulation and field tests, respectively. Specifically, the MATEs for tracking the straight-line sections were 4.48 cm and 5.67 cm in simulation and field tests, respectively. The LQR steering control algorithm with the proposed system was able to guide the robot to traverse between the crop rows without colliding with crop rows. The proposed navigation control system can also be used in navigation applications with central-articulated vehicles with GPS signals available.

Acknowledgment:

This project was funded by the National Science Foundation Major Research Instrumentation Program (Grant No. DBI-1625364) and the Plant Sciences Institute at Iowa State University. We thank Taylor Tuel, Benjamin Steward, and Duan Liu for their assistance in hardware design.

References


https://doi.org/10.1109/ICRA.2017.7989418


https://doi.org/10.1109/IROS.2017.8206458


CHAPTER 3. USING A DEPTH CAMERA FOR VISUAL ODOMETRY AND FIELD MAPPING IN UNDER-CANOPY AGRICULTURAL ROBOTIC NAVIGATION

A manuscript under review at Computers and Electronics in Agriculture

Jingyao Gai\textsuperscript{a}, Lirong Xiang\textsuperscript{a}, Lie Tang\textsuperscript{a}*

\textsuperscript{a} Department of Agricultural and Biosystems Engineering, Iowa State University, Ames, Iowa 50011-3270, USA

*Corresponding author. E-mail address: lietang@iastate.edu (L. Tang)

Abstract

Computer vision provides local environmental information for robotic navigation in crop fields. It is particularly useful for robots operating under canopies of tall plants such as corns (\textit{Zea Mays}) and sorghums (\textit{Sorghum bicolor}), where GPS signal is not always receivable. However, the development of under-canopy navigation systems is still an open research area. The key contribution of our work is the development of a vision-based system for under-canopy navigation using a Time-of-Flight (ToF) camera. In the system, a novel algorithm was used to detect parallel crop rows from depth images taken under crop canopies. Two critical tasks in navigation were accomplished based on the detection results: 1) generating crop field maps as occupancy grids when reliable robot localization is available (from other sources such as GPS and IMU), and 2) providing visual odometry (VO) data when the field map is available and the localization is not reliable. The proposed system was evaluated in field tests. The test results showed that the proposed system was able to map the crop rows with mean absolute errors (MAE) of 3.4 cm and 3.6 cm in corn and sorghum fields, respectively. It provides visual odometry data with MAE of 8.9 cm and 8.2 cm for positioning in corn and sorghum crop rows, respectively. The potential and limitations of using ToF cameras for under-canopy navigation were discussed.
Introduction

Robotic techniques were incorporated into agricultural production and research to automate various field operations (Bergerman, Billingsley, Reid, & Van Henten, 2016). The robotic techniques increase the operational efficiency and reduce the labor cost when compared with traditional methods. Many agricultural field operations were already automated through automating heavy-duty field equipment, such as seeding (Pedersen, Fountas, Sørensen, Van Evert, & Blackmore, 2017), weeding (Steward, Gai, & Tang, 2019) and harvesting (Birrell, Hughes, Cai, & Iida, 2019). However, lighter-duty and complicated field operations such as plant inspection and phenotyping usually require reduced human interaction while maintaining higher precision and robustness, imposing challenges to the intelligence and autonomy levels of the robot (Mueller-Sim, Jenkins, Abel, & Kantor, 2017). An important technology which determines the level of autonomy is autonomous navigation. Navigation system design remains as a challenge for a field robot, due to the complex and unstructured/semi-structured outdoor environment compared with the indoor conditions (Bergerman et al., 2016). It is particularly true for a small robot working under canopies of crop plants.

Many commercialized field robots employ Global Navigation Satellite Systems (GNSS) data as a global positioning source, and operate with a field map surveyed through aerial imaging or seed maps generated by planters (Bonadies & Gadsden, 2019). But the performance of these navigation systems highly depends on the GNSS signal quality, and the “dynamics” of the environment are not considered. Especially for light-weight autonomous robots operating in corn and sorghum fields, the GNSS signal might be blocked by the canopies and the robots need to detect crop rows and avoid potential collisions on-the-go. Therefore, a large body of studies has
been devoted to increasing the localization robustness by fusion more sensors. For example, Inertial navigation system (INS) is able to provide additional position and heading data of the robot with high accuracy in short time frames, but suffers from long-term sensor drift (Eaton, Katupitiya, Siew, & Howarth, 2008).

Computer vision-based techniques have been explored in recent years for navigation in crop fields or orchards. Sensors such as camera and LIDAR provide a rich source of the depth, color and texture of the local environment (Shalal, Low, McCarthy, & Hancock, 2013). The crop/tree rows present straight, parallel, and regularly spaced line patterns were identified as the navigation cues. Computer vision enables both robot localization and field mapping: with pre-generated field maps, visual odometry (VO) can be calculated and act as an additional localization source for row guidance. Inversely, with reliable localization results, the crop row detection results can be used to generate or update the field map for future navigation. In addition, these two operations can happen concurrently, which is referred to as Simultaneous localization and mapping (SLAM) (Shalal et al., 2013). Most of the reported field-based visual navigation algorithms use images acquired from cameras placed above row-crop canopies, or between trees in orchards (Bonadies & Gadsden, 2019). But such algorithms are not suitable for robots navigating under tall plant canopies in narrow rows. With the limited crop plant visibility from the two adjacent rows only and the noisy and cluttered sensing environment caused by weeds and leaves, under canopy row detection represents a challenging sensing task. (Figure 3-1)

This reported research was set out to investigate the feasibility of using a front-facing Time-of-Flight (ToF) camera to addresses the problem of computer vision-based navigation under the canopy of leafy crops with narrow row spacing, such as corn and sorghum. The specific objectives were to (1) assess the developed under-canopy ToF-camera-based crop row
detection algorithm; (2) evaluate the mapping and visual odometry performance using the proposed row-detection algorithm.

![Images for field-based navigation. Left: images of crop rows acquired above-canopy in field; Middle: images of tree rows acquired between rows in orchard; Right: images of crop rows acquired under tall plant canopies in field, where the crop rows are more difficult to detect.]

This paper is organized as follows: In section 2, the state-of-the-art row detection methods for field-based navigation are reviewed. In section 3, our developed robot PhenoBot 3.0 as the experimental platform is introduced, and the navigation system is described. In section 4, the pipeline of crop row detection, VO calculation and map generation is described. Section 5 provides the results of quantitative analyses of the visual odometry and mapping. The capability and limitations of the proposed method and the ToF-based system are discussed in depth. Finally, the study is concluded in section 6.

**Related works**

The computer vision-based navigation is widely used on agricultural field robots to compensate the unreliability of GPS signals. Crop row detection is the main strategy for in-field navigation. Generally, the row detection algorithms can be divided into categorizes based on the sensor placement: above-canopy or under-canopy. Two types of sensors have been used mainly: One type of sensors includes cameras which measure reflectance intensity of light, such as color cameras and infrared cameras; Another type of sensors is called depth sensor that can measure the distance from sensor to object, such as light-detection-and-ranging (LiDAR) sensors and ToF cameras.
Most reported studies use above-canopy sensors for crop row detection where multiple rows are observed simultaneously. Algorithms using reflectance images mainly focus on a series of features with parallel patterns in color or in textures (English, Ross, Ball, & Corke, 2014; Gottschalk, Burgos-Artizzu, Ribeiro, Pajares, & Sánchez-Miralles, 2008; Li, Zhang, Du, & He, 2020; Morio, Teramoto, & Murakami, 2017; Takagaki, Masuda, Iida, & Suguri, 2013). However, these algorithms rely on the reflectance difference between crop plants and the background. Challenges of these algorithms lie in the presence of weeds, and the connected inter-row plant canopies in above-canopy images, which usually happen at late growth stages. Range sensors such as stereo cameras were utilized to extract height information of the plants to improve the robustness of row detection against weeds and connected canopy (Kise, Zhang, & Rovira Más, 2005; Kneip, Fleischmann, & Berns, 2020). But they are only feasible when there is a height difference between crop plants and weeds. In addition, the algorithms using above-canopy images were mainly designed for large ground vehicles such as tractors, or aerial platforms such as UAV’s. Since this series of algorithms has restrictions on camera height, field condition, and plant growth stages, it is not suitable for light-weighted under-canopy robots.

Another series of algorithms used data acquired from sensors placed under-canopy. These algorithms were mainly designed for small robots operating under canopies of tall row crops such as corn and sorghum, or robots operating between tree rows in orchards (Shalal et al., 2013). When a robot travels under canopies, only two adjacent rows are observable at a time. In reflectance-based approaches, the pixels of plants, soil, and sky were segmented in the images, and the boundaries were extracted for directrix estimation. Yang et al. (2018) proposed an algorithm to detect the centerlines of maize rows under canopy by extracting the bottom section of plants. In the study of Radcliffe et al. (2018), the sky pixels were extracted from color images
to calculate the tree row direction in orchard, since both the sky and the ground outline a similar path. Zhang et al. (2012) tracks tree trunk pixels to calculate the position of the trees for directrix estimation. However, the nature of inconsistent illumination and shadow in the crop field would pose a challenge to these image-based algorithms.

Since 3D features are less prone to the illumination inconsistency, sensors which are capable of directly supplying distance measurements, such as LiDAR and ToF camera, have advantages compared to other cameras which measure reflectance intensity. LiDAR was broadly used in various of navigational applications including SLAM (J. Zhang & Singh, 2014) and autonomous driving (Wang, Wang, Liu, Meng, & Yang, 2017). Interests of applying LiDAR in agriculture was sparked due to the reduced sensor price in recent years. In case of row-following tasks, the use of single-layer LiDAR sensors was reported in orchard navigation (Bergerman et al., 2015; Jones et al., 2019) and in less structured crop fields (Higuti, Velasquez, Magalhaes, Becker, & Chowdhary, 2019; Mueller-Sim et al., 2017). In these applications, the general strategy was to place a single-layer LiDAR sensor levelly and close to the ground, aiming at the tree trunks or the crop stalks. The directrix was therefore measured by applying line fitting algorithms such as least square, RANSAC and PEARL to the measured data points (Isack & Boykov, 2012).

According to Higuti et al. (2019), there are several challenges posed to single-layer LiDAR-based navigation in the field. One is the cluttered rows where weeds and leaves may block the tree trunks or the crop stalks in the sensor output. Another is the loss of detailed 3D data since only a 2D set of distances measurement is available. In addition, frequent sensor obstruction can be caused by weeds and leaves, especially when navigating in crop rows. These challenges can be alleviated by using a sensor with a wider vertical view angel. A rotating 2D
LiDAR system was developed to output 3D data for tree trunk detection in orchards, and the performance was better than using a single 2D LiDAR sensor (J. Zhang, Chambers, Maeta, Bergerman, & Singh, 2013). Similarly, a 3D LiDAR-based navigation system was reported to outperform single-layer LiDAR-based systems during row following in pergola structured orchards (Bell, MacDonald, & Ahn, 2016).

Time-of-flight camera is another range sensor which measures distance based on the time of flight of modulated IR light. It produces depth images of a scene with low computational overheads at high acquisition rates. It has a wider vertical view angle compared to LiDAR sensors. ToF cameras were widely used in indoor navigation and mapping (Behrje, Himstedt, & Maehle, 2018; Weingarten, Gruener, & Siegwart, 2004; Yuan et al., 2009). But to our knowledge, there is no published research or commercial product on using ToF cameras as the primary sensor source for in-field, under-canopy navigation and mapping. That was because most commercially available ToF sensors come with weak light sources, therefore have poor measurement accuracy under directly sunlight, as a result of spectrum confliction between the sensor’s active light source and the sunlight. Since some recently available ToF sensors have strong light sources, which enable the sensor to function in outdoor environments, the potential of using ToF cameras for under-canopy navigation becomes worthwhile to investigate.

Material and Method

1. PhenoBot 3.0

This section describes an autonomous field-based ground robot, PhenoBot 3.0, as part of the project. PhenoBot 3.0 is a robotic ground vehicle designed for high-quality organ-level phenotypic data acquisition for corn and sorghum plants within an agronomic row spacing of 30” (76 cm) (Figure 3-2, left). The robot was equipped with a telescoping (7’ - 12’, 2 m - 3.5 m) and auto-balancing sensor mast. A series of stereo cameras were carried to acquire close-range side-
view images of the plants (Figure 3-2, middle). The telescoping mast and the motorized camera placement enable the robot to image the plants at different heights simultaneously. Various organ-level traits can be acquired from imaging including brace roots, stalks, ears, leaf angles, and tassels or panicles of maize and sorghum plants at different growth stages. The auto-balancing feature of the mast prevents the mast from colliding with the crop plants while the robot is running on uneven field surfaces, and keeps the cameras at a relatively constant imaging distances to the plants to ensure image quality. Our team has also developed a stereo camera module with an active illumination, which further increases the image quality under variable outdoor lighting conditions.

The robot rover features a narrow body design with 20” (51 cm) width to fit the required 30” (76 cm) row spacing. A central articulated steering mechanism and a differential gear based power transmission enable the robot to swiftly traverse between narrow crop rows with high energy efficiency (Figure 3-2, right). A set of high-capacity batteries were placed inside the robot, which can support up to 8 hours driving in the field – enough time to survey about 4.85 acres (1.8 hectares) at 2 mph (3.2 Kmh) travel speed.

Figure 3-2. PhenoBot 3.0, an autonomous robotic ground vehicle designed for field-based corn and sorghum plant phenotyping within a 30-inch row spacing.

The navigation system of PhenoBot 3.0 features a centralized control scheme (Figure 3-3). It comprises a main computer running the control software, a series of sensors and motion actuators. The central CPU running Robotic Operation System (ROS) (Stanford Artificial Intelligence Laboratory et al, 2018) middleware handles the sensor data and executes control algorithms, then sends commands to low-level motor controllers to control the motion of the robot. The navigation software consists of a computer vision module for sensing the crop rows and calculating the visual odometry (VO) for robot pose estimation and field mapping. The rest are non-vision modules for robot pose estimation, mapping, path planning and tracking. The computer vision module is the focus of this reported research.

In PhenoBot 3.0, various sensors were used to accomplish the navigation task, including a ToF camera, an RTK-GPS module, an Inertia measurement unit (IMU), and a set of encoders. The main sensor for vision-based navigation functions is a StarForm Swift-G ToF camera (Odos Imaging, Scotland). The camera equips with seven strong infrared LED light sources, which enables the camera to operate in outdoor conditions. It has an angle of view of 43° x 33°, and outputs depth and infrared intensity images simultaneously with a resolution of 640 x 480. The precision of depth measurement is in centimeter-level when operating at the range of 0.3 m-2.5 m, which is sufficient for crop row detection. The camera was mounted 8” above the ground on the front part of the rover with a forward facing pitch angle of 8° downward from the horizontal direction (Figure 3-4).

The other sensors act as additional localization sources for the robot pose estimation. An RTK GPS (Reach RS2, Emlid, Russia) antenna was mounted on the top of the sensor mast, which provides centimeter level global positioning of the robot. An IMU sensor (3DM-GX5-15, LORD Sensing MicroStrain, USA) was placed in the front part of the rover, which measures the
heading change rate and acceleration of the robot. Encoders were placed at the output shafts of the drive motors and the steering joint, to provide the speed and the steering angle of the robot. These signals were fused by an Extended Kalman Filter (Moore & Stouch, 2016) to calculate the position and heading of the robot in the world coordinate.

Figure 3-3. The architecture of the PhenoBot 3.0 navigation system.
Vision-Based Navigation System Development

There are three major functions in our developed vision-based navigation module. The first is a novel image processing algorithm for the detection of parallel crop rows from ToF depth and intensity images. The second is field mapping, which uses the row detection results and a robot pose estimation algorithm to generate crop row maps as occupancy grids. The last function is the calculation of visual odometry at the presence of pre-surveyed crop row maps. In our application, the desired level of error of field mapping and visual odometry is less than 2” (5 cm) to avoid collision with the crop rows.

1. Row Detection

The basic strategy of the row detection is to fit two parallel planes in the point cloud with a prior distance estimation using vegetation pixels representing the crop rows. In the row detection function, the depth image and the infrared intensity image from the depth sensor were taken as inputs. The framework of the crop row detection and localization algorithm is outlined as the following steps:
Step 1: Preprocessing: This procedure filtered the depth image using a modified bilateral filter and removed invalid pixels in the depth image.

Step 2: Ground detection: The filtered depth image was transformed into a 3D point cloud. The soil surface was detected (assumed to be a plane) in the 3D point cloud. Vegetation points above the soil surface were extracted.

Step 3: Crop row detection: The vegetation points were sliced and clustered into supervoxels to increase the percentage of representative points of crop rows (i.e. the corn stalk pixels). A linear programming method was applied to fit the parallel crop rows.

The algorithm was tuned and tested for corn and sorghum plants at different growth stages with different weed infestation levels. Each of these steps will be further described in the following sub-sections.

**Preprocessing**

The raw depth images acquired outdoor by the ToF sensor contained a substantial amount of noise and flying pixels that need to be filtered. However, the infrared intensity image was observed less noisy (Figure 3-5). Therefore, a modified bilateral filter which uses both depth image and infrared intensity image was developed to reduce the noise level in depth image. Then a pass-through filter was applied to remove the invalid points.
Figure 3-5. Example raw IR intensity image (left) and depth image (right) from the ToF camera. The color represents the raw intensity/depth value of each pixels. The depth measurement contains a substantial level of noise.

Bilateral filter is an edge-preserving smoothing algorithm (Paris, Kornprobst, Tumblin, & Durand, 2007), and it is widely used for improving the quality of the depth image. The original algorithm acts as a weighted averaging of pixels in a search window, during which both the difference of depth value of the pixels and the pixel distance on the image plane were taken into consideration. The depth value \( D \) of each pixel \( p \) after filtering can be denoted as \( BF[D]_p \) by:

\[
BF[D]_p = \frac{1}{W_p} \sum_{q \in S} G_{\sigma_s}(\|p - q\|) \cdot G_{\sigma_r}(D_p - D_q) \cdot D_q
\]

where \( q \) is pixel in the search window \( S \) centered at \( p \). \( G_{\sigma_s} \) is a spatial Gaussian term that decreases the influence of pixels further away from \( p \), and \( G_{\sigma_r} \) is a range Gaussian term that decreases the influence of pixels with more different depth values than that of \( p \). \( W_p \) is a normalization factor, which is the sum of weights in the search window. Both the spatial Gaussian and the range Gaussian term are calculated from the same image.

In this application, in order to take advantage from the intensity image, the bilateral filtering was modified to calculate the range Gaussian term \( G_{\sigma_s} \) from the infrared intensity image \( I \). Therefore, the depth value \( D \) of each pixel \( p \) in the depth image after filtering becomes:
Additionally, a passthrough filter was applied to reject pixels with lower measurement accuracy, which are the pixels either too far or too close to the camera. In this application, the pixels with depth less than 0.3 m or greater than 2.5 m were removed, based on the specifications of the camera.

Point cloud generation and ground detection

After filtering the depth image, a 3D point cloud was back-projected from the depth image using the camera projection matrix:

\[
\begin{bmatrix}
X \\
Y \\
Z
\end{bmatrix} = \begin{bmatrix}
f_x & 0 & c_x \\
0 & f_y & c_y \\
0 & 0 & 1
\end{bmatrix}^{-1} \begin{bmatrix}
x \\
y \\
1
\end{bmatrix} \cdot d,
\]

where \((x, y)\) are the pixel coordinates in the depth image, \(d\) is the depth value of the pixel, \((f_x, f_y)\) are the focal length along the camera X and Y axis, respectively, \((c_x, c_y)\) are the principle points along the camera X and Y axis, respectively, \((X, Y, Z)\) are the 3D coordinates of the depth pixel. The parameters were obtained through camera calibration (Z. Zhang, 2000).

The ground plane was detected in the point cloud by fitting a 3D plane with Random Sample Consensus (RANSAC) (Fischler & Bolles, 1981). Due to the presence of weeds, the detection result had a chance of returning incorrect results. Thus, the detection result was verified with the prior estimation of the camera pose to improve the detection robustness. If the detection result is consistent with the estimation, the detection result was kept. Otherwise, it was discarded, and the prior estimation was used for future steps.

The points above the ground were extracted into a new point cloud, which was composed of vegetation pixels from the crop rows and potentially weeds. The detected ground was used to
serve as a reference plane for row detection since the crop plants were assumed to be perpendicular to the ground.

**Row detection**

As observed in the raw images, the pixels of corn/sorghum stalks served as a structural feature in crop row detection. But the stalk pixels were difficult to extract directly due to the presence of substantial amount of leaf pixels in the extracted point cloud and the noisy data even after filtering. In this application, the point cloud was clustered into supervoxels using an clustering algorithm based on horizontally slicing (Bao, Tang, Srinivasan, & Schnable, 2019; Xiang, Bao, Tang, Ortiz, & Salas-Fernandez, 2019) to reduce the proportion of leaf pixels in the point cloud. The clustering algorithm was originally developed to skeletonize the side-view point cloud of maize plants, and was proved to be effective in emphasizing the stalk features of tall growing plants such as maize and sorghum. In the clustering algorithm, the point cloud was first evenly sliced into thin layers along the growth direction of the plants. Afterwards, a Euclidean Clustering algorithm (Rusu & Cousins, 2011) was applied to cluster each layer into supervoxels based on the spatial connectivity. And the 3D centroid point of each supervoxel was extracted into a new point cloud (Figure 3-6).
By assuming the crop rows are straight and parallel, the row detection problem was abstracted as a mathematical problem: fitting a pair of parallel planes \((\mathbf{n}_1, \mathbf{n}_2)\) with prior estimated distance \(s_0\) in 3D point clouds. A dynamic programming-based algorithm was applied to solve the problem. A pair of parallel planes in 3D space can be defined by four parameters: \((\theta, \varphi, s, \rho)\). The polar angle \(\theta\) and the azimuthal angle \(\varphi\) indicate the direction of the normal vector of the plane in spherical coordinates. \(s\) is the distance between the parallel planes, and \(\rho\) is the distance from the origin to the central plane of the parallel planes toward the normal direction. Therefore, the two planes have the equations:

\[
\mathbf{n}_1: (\sin \theta \cos \varphi, \sin \theta \sin \varphi, \cos \theta, 1) \cdot (x, y, z, s - \frac{\rho}{2}) = 0
\]

\[
\mathbf{n}_2: (\sin \theta \cos \varphi, \sin \theta \sin \varphi, \cos \theta, 1) \cdot (x, y, z, s + \frac{\rho}{2}) = 0
\]

A cost function to minimize was defined as:

\[
\text{cost}(\theta, \varphi, s, \rho) = w_1 \sum_{p \in N_1} \frac{\text{dis}^2(p, \mathbf{n}_1)}{\text{size}(N_1)} + w_2 \sum_{p \in N_2} \frac{\text{dis}^2(p, \mathbf{n}_2)}{\text{size}(N_2)} + \gamma (s - s_0)^2
\]

in which \(N_1\) and \(N_2\) are the disjoint unions of the points in the point cloud which are closer to \(n_1\) and \(n_2\), respectively. \(s_0\) is the prior estimation of the distance between the planes. The cost function is composed of three terms. The first two terms are the weighted average squared
distances to the nearest plane. The weight factors $w_1$ and $w_2$ are defined based on the size of $N1$ and $N2$ as:

$$w_1 = \frac{\sqrt{\text{size}(N_1)}}{\sqrt{\text{size}(N_1)} + \sqrt{\text{size}(N_2)}}$$

$$w_2 = \frac{\sqrt{\text{size}(N_2)}}{\sqrt{\text{size}(N_1)} + \sqrt{\text{size}(N_2)}}$$

which sets a higher weight to the plane consists of more inliers, since more inliers usually means higher fidelity. The third term is the violation of the distance prior estimation, and its weight is controlled by a weight factor $\gamma$. The cost function is non-linear due to the conditional operation to dynamically determine the plane inliers.

The Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm (Chambers & Fletcher, 2001), which is an iterative method for solving unconstrained nonlinear optimization problems, was applied to minimize the cost function. In the algorithm, the approximated the Hessian matrix of the cost function was calculated approximately using local gradients during each iteration, and the parameters gradually approach the optimal value to minimize the cost function. Therefore, the optimal parameters ($\theta, \varphi, s, \rho$) representing the estimated position and direction of the crop rows can be calculated. In addition, if the crop rows are assumed vertical to the ground, the problem can be reduced by fixing parameter $\theta$ to 0.

2. Mapping

In navigation applications, grid maps are used to represent the occupancy status of different locations in the world. In grid maps, the space is discretized into cells to approximate the obstacles in the real world, and each grid stores the probability that the cell is occupied by an obstacle. Given the crop row detection results, a local 2D grid map representing the crop row locations was constructed in real-time. The grid map was pie-shaped, with a radius of 2.5 m, which is consistent with the sensor’s field of view (Figure 3-7a). The cell size was set to 2.5 cm
based on the concerns of navigation accuracy requirement, the sensor accuracy, and the computational cost. Based on the beam model of range finders theory (Thrun, Burgard, & Fox, 2005), the cells in the local map were divided into three categories: free space, occupied space, and unknown space, with occupancy probability of 0.3, 0.7 and 0.5, respectively. The categorization is performed according to the detected crop row position.

Based on the available global pose of the robot, the local map was used to update the global occupancy map (Figure 3-7b). Using the Bayesian Mapping strategy (Thrun et al., 2005) the probability of occupancy of each cell was updated based on the Bayesian rule. This strategy is popularly applied in various mapping applications with LiDAR sensors. After the construction of the probability map, a global map of collision-free space was extracted by thresholding the occupancy probability, which provided references for collision-free navigation to the robot.

Figure 3-7. The map and the mapping area. (a) The circular sector shaped local occupancy map with three occupancy states determined by the detected crop rows. (b) The updated area (in yellow color) in the global map, which is determined by the camera pose. The grid size is for representation and does not match the actual size.
3. Visual odometry

Visual odometry serves as localization sources to localized the robot in a mapped environment in many applications. It requires existing maps and perceiving sensors and is beneficial while the GNSS signal is not reliable. In this application, the direction and the lateral position of the robot relative to the crop row were characterized by the detected row direction and the distance to the camera origin in the point cloud. They were used to compare with the existing map to locate the robot in the map. However, due to the limited reliable features for robot positioning along the crop row, the visual odometry is beneficial in lateral positioning only. And the positioning along the crop row still relies on other localization sources such as the IMU and encoders on the robot.

Results and Discussions

1. Crop Row Detection Performance Evaluation

The row detection algorithm was evaluated in terms of the accuracy of the estimated ground plane and row planes parameters. Images for evaluating the developed crop row detection algorithm were collected in the summer of 2018 using our developed PhenoBot 3.0 robot as described in the material section. The images were taken at two different times at two different locations in order to test the performance of the algorithm under different conditions such as different plant growth stages, weed infestation levels, and soil conditions:

The first dataset was collected in a breeding farm located in Omaha, Nebraska. The plants were at about the V5 growth stage, 30 days after planting (30 DAP). The soil was dry. The second dataset was collected at Iowa State University Agricultural Engineering and Agronomy Research Farm located in Boone, Iowa. The plants were at about the R1 growth stage, 80 DAP. The soil was wet. The fields all featured a standard 76 cm (30”) row spacing, and about 15 cm (6”) inter-plant spacing. About 3000 images were collected each time. The weather condition
was sunny during data collection, and the sunlight illuminance ranged from 60,000 to 80,000 lux, and the camera exposure parameters were adjusted to fit the illumination conditions.

In this study, 120 images from the 30 DAP dataset and 200 images from the 80 DAP dataset were randomly sampled and manually labeled. More images were sampled from the 80 DAP dataset because the detection task is more challenging due to its higher weed infestation level, more complicated crop plant structure, and more frequent sensor blockage by the leaves. To quantify the performance of the algorithm, the accuracy of two critical steps: ground detection and crop row detection, were evaluated.

Several parameters were calculated from the detected ground and crop row plants for performance evaluation. In the ground detection step, two parameters were calculated from the detected ground plane, including the normal direction and the distance to the centroid of the point cloud (D2C) (Figure 3-8a). In the crop row detection step, three parameters were calculated for each plane, including the normal direction, the distance from the crop rows to the camera origin position (D2O), and to the cloud centroid (D2C) (Figure 3-8b). Therefore, accuracies of the ground detection and crop row detection were evaluated by comparing the above parameters of the detected planes with the ground truth. The ground truth was generated by manually labeling the ground plane and the crop row planes in the point cloud based on human observation, and calculating the corresponding parameters using the labeled planes. The cloud centroid of each frame was calculated from the point cloud after pre-processing, which remained consistent in experiments and in the ground truth extraction. The accuracy of the ground plane and crop row detection steps were evaluated by the mean absolute error (MAE) of these parameters. The accuracies of crop rows normal directions and D2Os of crop rows were critical for the visual odometry task, and the accuracy of D2Cs was critical for the field mapping task.
Figure 3-8. The illustration of the plane parameters definition. Left: The illustration of the ground plane normal direction and its distance to the cloud centroid from side view. Right: The illustration of the crop row planes normal direction and their distances to the cloud centroid (D2C) and to the camera origin (D2O) from top view.

Figure 3-9 shows two example sets of step results of the detection algorithm for both the 30 DAP and 80 DAP datasets. The statistical analysis is presented in Table 3-1. For the 30 DAP dataset and 80 DAP dataset, 114 out of 120 images (95%) and 144 out of 200 images (72%) were valid for processing, respectively. The invalid images were those with either the ground or the crop rows fully or mostly occluded by leaves of crop plants and weeds. For the valid images, the accuracies of both ground plane estimation and crop row plane estimation were evaluated (Figure 3-10 - 11). For the 30 DAP dataset, the algorithm was able to detect the ground plane with mean absolute error (MAE) of 0.013 rad (0.74 deg) in normal direction, and 0.16 cm in distance to the cloud centroid. And the crop rows were detected with MAE of 0.057 rad (3.2 deg) in normal direction, and 2.2 cm and 6.9 cm in D2C and D2O, respectively.

For the 80 DAP dataset, the algorithm was able to detect the ground plane with MAE of 0.79 rad (4.5 deg) in normal direction, and 2.1 cm in distance to the cloud centroid. The crop rows were detected with MAE of 0.054 rad (3.1 deg) in normal direction, and 3.6 cm and 8.0 cm in D2C and D2O, respectively. The processing time for each frame was about 0.3 s on average, using a computer with a i5-6500TE processor.
Table 3-1. Crop row detection performance. MAE: mean absolute error compared with the ground truth. DAP: days after planting.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>30 DAP</th>
<th>80 DAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valid images/total images</td>
<td>114/120 (95%)</td>
<td>144/200 (72%)</td>
</tr>
<tr>
<td>Ground detection accuracy (MAE)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Direction error</td>
<td>0.013 rad (0.74 deg)</td>
<td>0.079 rad (4.5 deg)</td>
</tr>
<tr>
<td>Distance to cloud centroid (D2C) error</td>
<td>1.6x10^{-3} m</td>
<td>0.021 m</td>
</tr>
<tr>
<td>Row detection accuracy (MAE)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Direction error</td>
<td>0.057 rad (3.2 deg)</td>
<td>0.054 rad (3.1 deg)</td>
</tr>
<tr>
<td>Distance to cloud centroid (D2C) error</td>
<td>0.022 m</td>
<td>0.036 m</td>
</tr>
<tr>
<td>Distance to camera origin (D2O) error</td>
<td>0.069 m</td>
<td>0.080 m</td>
</tr>
</tbody>
</table>

Figure 3-9. Sample images of representative step results (data collected at 30 DAT for left images, 80 DAT for right images) including a) the raw 3D point clouds without pre-processing; b) the 3D point clouds after pre-processing; c) the detected ground planes; d) the extracted vegetation pixels above the ground planes and the clustering results; e) supervoxels projected to top-view, and f) the detected parallel planes representing the crop rows. The points in a), b), c) and f) were colored by the infrared intensity value from the raw images.
Figure 3-9. Continued.
Most of the invalid cases observed were caused by the sensor obstruction by the leaves, in which the structural features of crop rows were not perceivable. And the rest of the failures were caused by the tall weeds which affected both the ground detection and row detection.

As observed, performance differences exist in both ground detection and row detection steps on the two different datasets. For the ground detection step, lower accuracy was observed on the 80 DAP dataset, which was mainly caused by the higher weed density and the higher soil moisture level. The weeds affected the RANSAC based ground detection as outliers in the algorithm, and the wet soil absorbed more IR light for ToF measurement, which reduced the measurement range and accuracy of the ToF camera.

For the row detection step, lower accuracy was also observed on the 80 DAP dataset. Cases with obvious errors were caused by two main reasons:

The first is the sensor measurement noise of the soil pixels. The proposed algorithm relies on the stalk pixels, particularly the section close to the ground. The inaccurate ground plane led to difficulties in extracting these stalk pixels.

The second reason is weed infestation and more complex structure, especially the dropping leaves of the crop plants at late growth stages. They both introduced outliers in the crop row fitting step.

In general, the moderate row detection accuracy in terms of row direction and distance to the cloud centroid (D2C) indicates its high fidelity in field mapping applications. However, the error level in estimating the distance to the camera (D2O) is high, which means lower fidelity in visual odometry. That was due to the sensor's limitation of a narrow field of view and relatively short measurement range. In this study, the cloud centroid is about 1.5 m away from the camera, which means minor error in row direction would lead to high D2O error.
Comparing to the LiDAR-based algorithm developed by Vitor et al. (2019), the advantage of our proposed algorithm is the ability to detect parallel crop rows with a certain row spacing. However, one limitation is that the row detection results were produced by using only one image, which is prone to outliers such as weeds and leaves, due to the least-square based strategy and relying on no historical measurements. Additionally, the proposed algorithm is computationally more costly because of the higher data density of the ToF camera compared to LiDAR sensors.

Figure 3-10. The ground detection performance on the 30 DAT and 80 DAT datasets. (a, b): The linear regression results of the measured distance to the cloud centroid and the ground truth in both datasets. Strong correlations exist in both datasets. (c): The distribution of the absolute angular error for ground plane normal measurement.
Figure 3-11. Row detection performance on the 30 DAT and 80 DAT datasets. (a, b): The linear regression results of the measured distance to the cloud centroid (D2C) and the ground truth in both datasets. (c, d): The linear regression results for measured distance to the camera position (D2O) and the ground truth in both datasets. Strong correlations exist in both D2C and D2O correlation in both datasets. (e): The distribution of the absolute angular error for row direction measurement.
2. **Mapping and Visual Odometry Performance Evaluation**

The performance of field mapping and visual odometry based on the proposed row detection algorithm were evaluated. Field tests in both corn and sorghum fields were conducted at Iowa State University Agricultural Engineering and Agronomy Research Farm located in Boone, Iowa in October of 2019. The plants were planted in late season (August), with a standard 76 cm (30”) row spacing, and about 15 cm (6”) inter-plant spacing. During field tests, the plants were at about the V5 growth stage (about 60 cm tall, 40 DAP). The soil was dry, and the weather was sunny (60k lux sunlight) with a strong southwest wind (24 mph). The developed PhenoBot 3.0 was manually driven between the crop rows with a speed of about 1 m/s (~2 mph). The mapping performance was evaluated by comparing the generated map with the manually surveyed crop rows, and the visual odometry performance was evaluated by comparing the system-derived results with the GPS-IMU fusion localization results.

Figure 3-12 shows example run trails for testing field mapping in corn and sorghum fields. Both example run trials are 35 m in length. Among the ToF camera frames during the run trails, 325 out of 504 and 359 out of 401 frames were valid and processed for the corn field trail and the sorghum field trail. As observed, the constructed maps using the proposed algorithm matches the manually surveyed ground truth. The means of the absolute position errors (MAE) were 3.4 cm (1.3”) and 3.6 cm (1.4”) in corn and sorghum field tests, respectively (Figure 3-13). The results are acceptable considering the stalk size of the crop plants, and the requirement of our navigation system. And the results are generally consistent with the algorithm performance evaluation results in the previous section.

Figure 3-14 shows example run trials for testing visual odometry in corn field and sorghum field. The visual odometry was calculated using images acquired in the mapping tests. As observed, the visual odometry results have a moderate level of error in heading estimation,
but a higher level of error in lateral position estimation compared with the field mapping. The means of the absolute heading errors were 0.054 rad (3.0 deg) and 0.055 rad (3.1 deg) in corn and sorghum field tests, respectively (Figure 3-15). And the means of the absolute positioning errors were 8.9 cm and 8.2 cm in corn and sorghum field tests, respectively (Figure 3-16). The positioning error levels are slightly higher than what presented in the algorithm performance evaluation in the previous section, which was due to the strong south wind during data collection. The leaves were blown to the north, which made the detected row on the south of the robot closer than the ground truth.

Figure 3-12. The mapping results of the crop rows and the ground truth positions of the example field tests of corn (top) and sorghum (bottom).
Figure 3-13. The distribution of the mapping position error for the corn (left) and sorghum (right) field tests. Unit: m.

Figure 3-14. The visual odometry results and the robot position ground truth of the example field tests of corn (top) and sorghum (bottom).
Figure 3-15. The distribution of the visual odometry position error for the corn (left) and sorghum (right) field tests. Unit: m.

Figure 3-16. The distribution of the visual odometry direction error for the corn (left) and sorghum (right) field tests. Unit: rad.

As a summary, the results of this study showed that the proposed single ToF-camera-based system was capable of detecting crop rows and contributed to both field mapping and the visual odometry for navigation in the field. Compared to the LiDAR-based systems, there are some benefits of using ToF-camera-based systems:

- The critical information was more likely to be perceived as ToF sensors offer a wider vertical field of view compared with the single-layer measurement of LiDAR sensors.
- ToF camera provides 3D point cloud data, in which the soil surface could be extracted as a reference plane for crop row detection.
Although some limitations of LiDAR sensors were overcome, clear limitations were found in using ToF sensors for field-based navigation between crop rows:

- The narrow horizontal field of view and short measurement range of the ToF camera led to higher uncertainties in visual odometry, since a minor error in row direction estimation would produce a large error in lateral position estimation of the robot.

- The data quality of the ToF sensors is affected by the field conditions. For example, strong sunlight and high soil moisture would increase the measurement noise and reduce the reliable measurement range, therefore reduce the row detection accuracy.

In order to improve the reliability of ToF-camera-based navigation systems, some methods are suggested: a) expand the field of view by introducing other sensors, e.g. another back facing ToF camera on the robot, or laser range sensors on the side of the robot; b) improve the data reliability by using an auxiliary sensor such as a color camera, which would help extracting critical structural features in row detection; c) use a Kalman Filter based on the robot kinematic model to refine the visual odometry. The error level from the above experiments can be used as a guidance for parameter tuning; d) fuse the historical data to increase the data density while adding the temporal dimension to the data, which can potentially improve crop row detection, as what proposed in Vitor et al. (2019)’s LiDAR-based system.

**Conclusions**

In this paper, we investigated the feasibility of using a front-facing ToF camera for vision-based navigation and mapping in corn/sorghum fields under crop canopies. In the system, a novel algorithm was developed to detect parallel crop rows from under-canopy depth images using linear programming techniques. Two critical tasks in navigation were accomplished based on the detection results: 1) generating crop field maps as occupancy grids; and 2) providing
visual odometry (VO) data for positioning the robot in the crop rows. The results of algorithm evaluation showed that our proposed row detection algorithm was able to provide reliable crop row detection results as long as three requirements were met: known crop row spacing, straight crop rows, and flat and level terrain. The proposed system was evaluated in field tests. The test results showed that the proposed system was able to map the crop rows with mean absolute errors (MAE) of 3.4 cm and 3.6 cm in corn and sorghum fields, respectively. And it provides visual odometry data with MAE of 8.9 cm and 8.2 cm for positioning in corn and sorghum crop rows, respectively.

Using a front-facing ToF camera for navigation demonstrated certain feasibilities. The depth information of a ToF camera enables the crop row detection under crop canopies, therefore contribute to field mapping and visual odometry for robot navigation. The ToF-camera-based system encounters less sensor occlusion and is less prone to weeds or leaves compared with LiDAR-based systems. However, due to the limited sensing range, narrow horizontal field of view and unstable data quality of the ToF camera, the performance of the visual odometry is less promising when using a single sensor data frame each time. Actions that will potentially improve the performance are to fuse the ToF sensor with other sensors, as well as to fuse past historical frames to improve the row detection accuracy.

In summary, the presented algorithm and field-testing results demonstrate the feasibility of using a ToF camera for robotic navigation between the crop rows while revealed its limitations which can lead to further investigations. The framework of the proposed algorithm can also be extended to different types of mobile robotic operations under crop canopies.

Acknowledgment

This project was funded by the National Science Foundation Major Research Instrumentation Program (Grant No. DBI-1625364) and the Plant Sciences Institute at Iowa State
University. We thank Taylor Tuel, Benjamin Steward, and Duan Liu for their assistance in hardware design; and Yang Zhao, Juan Panelo and Le Wang for their contributions to the data collection.

References


International Conference on Robotics and Automation (pp. 3634–3639). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/ICRA.2017.7989418


CHAPTER 4. ROBOTIC IN-FIELD NAVIGATION BASED ON CROP ROW DETECTION USING DISCRETE FOURIER TRANSFORM (DFT)

A manuscript prepared for submission to Computers and Electronics in Agriculture

Jingyao Gai¹, Lirong Xiang¹, Lie Tang*¹

¹Department of Agricultural and Biosystems Engineering, Iowa State University, Ames, Iowa 50011-3270, USA

*Corresponding author. E-mail address: lietang@iastate.edu (L. Tang)

Abstract

Two-dimensional Discrete Fourier Transform (DFT) is widely used in image processing applications such as image filtering and pattern detection in frequency domain. The agronomically-spaced crop rows are well-constructed in a parallel spatial pattern that is mathematically linked to a spectral signature in the frequency domain. Therefore, the DFT can be potentially used to solve the crop row detection problems of robot navigation in agriculture, where the spacing and direction of rows are represented by the 2D frequency distribution and the position is represented by the phase in frequency domain. In this study, a novel image processing pipeline was developed to detect crop rows for agricultural robot navigation applications using frequency domain analysis. A series of techniques including interpolation, Hanning window, and Discrete Time Fourier Transform (DTFT) were applied to improve the detection accuracy of peak and phase in frequency domain. A Linear Quadratic Gaussian (LQG) controller was used with the proposed algorithm for robot navigation between crop rows. The field tests showed that the proposed crop row detection algorithm was able to estimate lateral position and heading deviations relative to the row centerlines accurately for plants with different growth stages and under variable illumination conditions. With the LQG controller, the robot was able to follow the centerline of crop rows with a mean absolute tracking error of 3.74 cm.
Introduction

Agricultural robots have been widely used in various field operations such as cultivation, spraying, harvesting, and weeding (Åstrand & Baerveldt, 2005). A robust navigation system is critical for robots to traverse crop rows and avoid crop damages. RTK-GNSS-based path tracking and visual guidance are two widely adopted strategies for robot navigation in the field (Kise, Zhang, & Rovira Más, 2005). Some systems employ both of them to improve the robustness of the navigation system (Young, Kayacan, & Peschel, 2019).

The development of RTK-GNSS enables the accurate measurements at centimeter-level of the global position of a robot in the Universal Transverse Mercator (UTM) coordinate. By fusing RTK-GNSS with inertial measurement units (IMU) and other sensors such as compass, the heading of the robot can be measured. The integrated system was often referred as a GPS-IMU system. With the localization results, path tracking algorithms were usually applied, with which the robot was driven to correct the position and heading deviation relative to the pre-defined path. The GPS-based navigation was reported in many applications to achieve high tracking accuracies at centimeter level when following desired paths (Nagasaka et al., 2009).

With its lower cost when compared with high-accuracy RTK-GPS modules, vision-based navigation system can be an economical alternative for in-field robotic navigation. Different sensors including stereo cameras (Kise et al., 2005), laser sensors (Winterhalter, Fleckenstein, Dornhege, & Burgard, 2018), and other odometry sensors (Vidović, Cupec, & Hocenski, 2016) were investigated in visual-based navigation systems. Visual guidance is usually applied to keep the robot at a central position when the traversable space is constrained by parallel obstacles such as the corridors-like environments (Hiremath, van Evert, Braak, Stein, & van der Heijden, 2014; J. Zhang, Chambers, Maeta, Bergerman, & Singh, 2013), where the obstacles were detected and located from images from color or depth cameras. The parallel obstacles are the crop rows for in-
field navigation in agriculture. Astrand (2005) listed several requirements for row guidance systems:

- Track rows with centimeters position accuracy, and degree-level angle accuracy if necessary
- Sufficiently high row tracking rate
- Able to work on sown crops, which could be smaller than weeds at the same time.
- Able to work with high weed pressure
- Able to deal with missing plants in the row

The most widely used setup for visual guidance in agriculture is using a top-view or front-view above-canopy camera (Figure 4-1) (Pajares et al., 2016). One of the most commonly used methods for identifying crop rows in the images is the Hough transform (Slaughter, Giles, & Downey, 2008). The Hough transform is a computationally efficient procedure for detecting discontinuous lines or parametrical curves in images, thus it meets the aforementioned requirements of row-following system. Hough transform and its applications for crop row detection were reported in a number of studies (Abdulsalam et al., 2016; Bossu, Gée, Guillemin, & Truchetet, 2006; Choi et al., 2015; Winterhalter et al., 2018). Other algorithms including linear regression (Benson, Reid, & Zhang, 2003) and green pixels accumulation (García-Santillán, Guerrero, Montalvo, & Pajares, 2018; Li, Zhang, Du, & He, 2020) were used to detect crop rows inside images. Furthermore, some applications also fuse local odometry information for row detection using particle filters (Blok, van Boheemen, van Evert, IJsselmuiden, & Kim, 2019; Grisetti, Stachniss, & Burgard, 2007).
Most of the studies detect crop rows based on the assumption of straight crop rows. Since agronomical crop rows are also parallel and equally spaced with a known spacing, this global spatial pattern of the image produces some inherent properties in the frequency domain. Specifically, the row patterns in images contribute to a certain frequency domain response, in which the period represents the spacing, and the corresponding frequency location represents the direction of the crop rows, and the associated phase represents the position of the crop rows. However, little study has been done in investigating how to apply the frequency domain analysis, especially the Discrete Fourier Transform (DFT), to detect crop rows for in-field navigation.

In this study, in order to solve the problem of crop row detection for in-field navigation, the potential of DFT and frequency domain analysis was studied. An image processing algorithm was developed to effectively detect crop rows in frequency domain, and a Linear Quadratic Gaussian (LQG)-based visual guidance system was implemented to navigate the robot using the detection results. The specific objectives of this study were to (1) assess the developed row
detection algorithm under different field and illumination conditions, and (2) evaluate the performance of the visual guidance system with the proposed row detection algorithm.

The rest of the paper is organized as follows. In section 3, our developed robot PhenoBot 3.0 as the experimental platform is introduced, and the navigation system is described. Section 4 covers the pipeline of crop row detection and the controller design. Section 5 provides the results of quantitative analyses of the tracking deviation estimation and robot navigation experiments. The capability and limitations of the proposed method is discussed in depth. Finally, the study is concluded in section 6.

**Robot Platform**

**PhenoBot 3.0**

This section describes an autonomous field-based ground robot, PhenoBot 3.0 from Iowa State University (ISU), as part of the project. The ISU PhenoBot was a series of robots designed to autonomously traverse between corn rows and carry a sensor package mounted on the robot, with which the phenotypic data of corn plants can be acquired across fields. The most recent design, PhenoBot 3.0, is capable of self-navigating between corn rows with conventional row spacing of 0.76 m by featuring a narrow body design and a central articulated steering mechanism (Figure 4-2). The telescoping sensor mast has an adjustable height between 2.1 m and 3.7 m. Additionally, the roll angle of the sensor mast was actively controlled to maintain vertical position in the presence of uneven ground surface. Multiple PhenoStereo cameras (Xiang, Tang, Gai, & Wang, 2020) were mounted on the sensor mast to acquire close-range and side-view stereoscopic images of the two rows of plants. The telescoping mast and the multi-sensor configuration enable the robot to simultaneously image plant sections at different heights.
Various organ-level traits such as brace roots shape, stalks size, ear height, leaf angle, and tassel/panicle shape can be imaged for corn and sorghum plants across different growth stages.

Figure 4-2. Phenobot 3.0 is articulated with a front and back section, and an actively-controlled vertical sensor mast for carrying sensors between corn plants.

Navigation System

PhenoBot 3.0 relies on vision-based navigation when there is no pre-defined path to follow. For the vision-based navigation system in this study, an RGB color camera (Phoenix 3.2MP, Lucid Vision labs, Canada) equipped with a lens of 4.0 mm focal length was used as the main navigation sensor for crop rows detection. The camera has a view angle of 85.8° in horizontal direction and 63.6° in vertical direction. It was placed on the top of the sensor mast of the PhenoBot, which is 2.1 m above the ground when the mast is vertical. The camera was placed with a down pitch angle of 37°. Its image vertical field of view covers the crop rows from 0.8 m to 23 m in front of the robot position reference, and the width of the covered area varies from 4 m (@0.8 m) to 40 m (@23 m). Part of the robot body was included in the image for artificially verifying the camera pose estimation result. The raw images have a resolution of 2048
x 1536 pixels. The spatial resolution is 2.0 mm/pix at 0.8 m ahead of the robot, and 21 mm/pixel at 23 m ahead of the robot (Figure 4-3).

![Illustration of the sensor placement and the field of view of the camera relative to the robot navigation reference position $P_{\text{robot}}$.](image)

In addition to the color camera, the robot was equipped with a series of sensors for estimating the pose and tracking the motion of the joints of the robot. The steering motor encoder measures the steering angle to calculate the robot posture change, and an IMU (Adafruit BNO055, Adafruit, NY, USA) was placed on the base of the sensor mast to measure the incline angle of the mast. The RTK-GPS antenna on the top of the sensor mast was used to evaluate the performance of the proposed algorithm, which provides centimeter-level global positioning accuracy of the robot. Motor encoders and a separate IMU on the rover were used for vehicle dead-reckoning and heading tracking. The signals from these sensors were fused by an Extended Kalman Filter (EKF) (Moore & Stouch, 2016) to calculate the position and heading of the robot.
in the world coordinate. The global positioning results is only used for validating the visual guidance system in this study.

The navigation system of PhenoBot 3.0 comprises a central computer running the control algorithms, a series of sensors, and the motion actuators (Figure 4-4). The central CPU running Robotic Operation System (ROS) (Stanford Artificial Intelligence Laboratory et al, 2018) middleware to firstly handle the sensor data and execute control algorithms, and then send commands to low-level motor controllers to control the motion of the robot. The navigation software used in this study consists of a computer vision module for detecting the crop rows and estimating the row tracking deviation.

Since the measurement from vision-based row detection contains a certain amount of noise, a controller compatible with noisy input signal is necessary. In our previous study, linear-quadratic-regulator (LQR) was found promising in tracking pre-defined path. LQG is suitable for this application, since it is a controller which combines the LQR with a linear quadratic state estimator (LQE), i.e. a Kalman filter. Therefore, an LQG controller was used in this study to steer the robot and keep it at the center of the adjacent rows. And the performance analysis of the row detection algorithm can give an insight to parameter tuning of the Kalman filter.
Crop Row Detection and Navigation

The vision-based row detection problem can be interpreted as finding the parallel pattern in an image with perspective distortion. In this study, a processing pipeline was developed to extract the direction and position of the crop rows from the frequency domain. The image process pipeline includes the following:

1. Preprocess the raw image to correct the distortion, wrap the image into a simulated bird’s eye view, and convert the image into gray scale.
2. Transform the image into frequency domain using Fast Fourier Transform (FFT) and Hanning window function.
3. Extract the crop row position and orientation by detecting the peak of magnitude with Taylor expansion-based interpolation and calculating the phase with Discrete Time Fourier Transform (DTFT).
4. Estimate robot tracking position and heading deviation.
With the row detection results, a Linear Quadratic Gaussian (LQG) controller was used to steer the robot and keep the robot on the center line between the crop rows.

**Image Preprocessing**

The raw image is deformed by both lens distortion and perspective distortion, with which the crop rows do not appear in parallel patterns. Therefore, these distortions need to be corrected to reveal the parallel pattern in the images. The lens distortion was corrected through the intrinsic parameters obtained by camera calibration (Z. Zhang, 2000). And the perspective distortion was rectified by generating a simulated bird’s eye view by projecting the image onto the ground plane (or the map plane). The problem can be interpreted as to find a 3x3 homography matrix $H$ that transforms the image coordinate $(x_i, y_i)$ to the map coordinate $(x_m, y_m)$ in homogenous coordinates. The transformation can be expressed as:

$$
\begin{bmatrix}
  x_i \\
  y_i \\
  1
\end{bmatrix} =
\begin{bmatrix}
  x_m \\
  y_m \\
  1
\end{bmatrix}
= H
\begin{bmatrix}
  x_i \\
  y_i \\
  1
\end{bmatrix}
$$

(20)

The transformation between the camera coordinate and the projected map coordinate is necessary to calculate the matrix $H$. The map coordinate was defined by the robot local coordinate frame (Figure 4-5). The origin of the map coordinate, $O_{\text{map}}$, was defined by the robot’s navigation reference point. The x-y plane is aligned with the ground level plane, where the x axis is aligned with the robot heading, thus the projected point has the form $(x_m, y_m, 0)$ in 3D (Figure 4-5). To get the real-time camera pose $P_{\text{cam}}$ in the local coordinate frame, the IMU sensor on PhenoBot’s sensor mast was used for tracking the camera orientation change in roll and pitch angle, and the steering motor encoder was used to track the robot posture change. The homography matrix can be solved by the camera projective transformation. The details are explained in Appendix A.
After generating the simulated bird’s eye view using the homography transform, the maximum axis-aligned inscribed rectangle of the projected area was extracted as an region of interest (ROI) image (Figure 4-5). This ensures the pixels in the ROI are all valid (i.e. has corresponding pixels in the raw image) and involves most information for crop row detection. The ROI can be refined by limiting the map area according to the previous knowledge of the field boundary and accuracy concern. The maximum looking ahead distance is limited to 16 m for heading estimation and 8 m for position deviation estimation in this study. The spatial resolution of the ROI image was set to 0.01 m/pixel. The ROI image was converted into grayscale using the Excess Green Index (ExG) (Yang, Wang, Zhao, Zhang, & Feng, 2015) to highlight the vegetation pixels without further denoising.

Figure 4-5. The illustration of the Homography transform between the image and the map, the ROI extraction, and the tracking deviation.
DFT-based Crop Row Detection

The 2D DFT was applied to decompose the image into periodical sinusoidal wave components, which represent the image in frequency domain, or Fourier domain. The crop row pattern in the image corresponds to a peak in the spectrum. The period of the peak represents the spacing, the angular location of that peak represents the direction of the crop rows, and the corresponding phase represents the position of the crop rows. However, it is inaccurate to extract the crop row pattern directly by finding the bin with maximum magnitude in the spectrum, due to limitation of insufficient frequency resolution and the spectrum leakage of the 2D DFT (more details can be found in Appendix B). To solve these problems, Taylor expansion-based interpolation was applied to increase the frequency resolution during peak detection, and Hanning window was applied to the image during DFT to reduce the spectrum leakage.

Hanning Window for Fourier Transform

Hanning window is a widely used window function which cause less spectrum leakage and increase the accuracy of peak detection. The Hanning window function in 1D is defined as:

\[
w(n) = 0.5 - 0.5 \cos \left( \frac{2\pi n}{M-1} \right), \quad 0 \leq n \leq M - 1
\]

where \(M\) is the sample size and \(n\) is the sample position (i.e. image pixel coordinate along one axis). Compared with the rectangular window, the frequency domain of Hanning has wider main lobe and lower side lobes (Figure 4-6). Due to the convolution theorem of Fourier Transform, windowing is equivalent to convolute the DFT spectrum with window spectrum. So, the Hanning window has less impact on the DFT result compared with rectangular window, thus reduced the spectral leakage. After applying the Hanning window to the image, FFT algorithm was applied to compute the DFT results.
Row pattern detection

A bandpass filter was first applied onto the DFT results by only keeping the contents with frequency closed to the crop row distancing. This operation was accomplished by masking the DFT results. The frequency \((u, v)\) of the row pattern was detected by locating the peak of the magnitude in the spectrum \(f(x, y)\). After finding the maximum magnitude value at \((x_0, y_0)\), Taylor expansion-based interpolation was applied to refine the peak detection result. The position correction \((\Delta x, \Delta y)\) was solved by applying the Taylor expansion to the gradient at position \((x_0 + \Delta x, y_0 + \Delta y)\). Since the gradient of a peak is zero, the correction can be computed by:

\[
\begin{bmatrix}
\Delta x \\
\Delta y
\end{bmatrix}
= -
\begin{bmatrix}
f_{xx} & f_{xy} \\
f_{xy} & f_{yy}
\end{bmatrix}^{-1}
\begin{bmatrix}
f_x \\
f_y
\end{bmatrix}
\]

(22)

In order to reveal the phase \(\varphi\), DTFT is applied on the windowed ROI image. DTFT applies Fourier transform on a specific frequency and outputs accurate result of phase estimation (while DFT is only on the sampled frequencies). The definition of 2D DTFT is:

\[
\text{DTFT}\{f(x, y)\}(u, v) = F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux + vy)} dx dy
\]
\[ F(u, v) = \sum_{m=0}^{M} \sum_{n=0}^{N} f(m, n)e^{-j2\pi (um/M + vn/N)} \]  \hspace{2cm} (23)

in which \((m, n)\) is the pixel coordinates in the source image, \((u, v)\) is the real-value target frequency, and \((M, N)\) is the image size. The mathematical definition is the same as DFT, but with no restriction of frequency sample resolution. Since DTFT is computationally costly, this method was only used for phase estimation after finding the peak frequency in this study.

The position of the crop rows in the image are the wavefronts of the sinusoid waves corresponding to the detected peak frequency. The points on the wavefront lines have their phases equal to 0. With detected peak frequency \((u, v)\), and the image size \((M, N)\), the period along \(x\) and \(y\) directions are:

\[ T_x = \frac{M}{u} \]
\[ T_y = \frac{N}{v} \]  \hspace{2cm} (24)

The units are in pixels. The line direction is:

\[ l = (-T_x, T_y) \]  \hspace{2cm} (25)

And a series of intersection on \(x\)-axis:

\[ P = [T_x * (2k\pi - \varphi), 0], k \in Z \]  \hspace{2cm} (26)

With the intersections and the line direction determined, the position and direction of the crop rows were determined on the map. And the heading and the lateral position deviation of the robot were calculated by referencing the center of the adjacent crop rows in the local map.

**Linear Quadratic Gaussian (LQG) Steering Control**

Since the position and heading deviation measurement using the DFT-based method is not perfect, it cannot be used directly for robot steering control. In this application, an LQG controller was applied to steer the robot to correct the lateral position deviation and heading
deviation for in-row navigation. The LQG controller combines a Linear Quadratic Regulator (LQR) with a Kalman filter (a linear-quadratic state estimator), which takes the noise in the linear system and in measurement process into consideration. According to the separation principle, the state estimator and the LQR feedback can be designed independently. Therefore, it is safe to use Kalman filter to reduce the error of the proposed row detection algorithm.

In this study, the general path tracking error model for car-like vehicles was used to model the robot (Corke, 2011). The discrete system model with process and measurement noise is defined as:

\[
\begin{align*}
    x_{n+1} &= Ax_n + Bu_n + w \\
    y_n &= Cx_n + v
\end{align*}
\]

(27)

in which \(w\) is the process noise vector with covariance matrix \(Q\), and \(v\) is the measurement noise vector with covariance matrix \(R\). The states \(x\) of the system were defined as the lateral position error \(e_y\), the heading error \(e_\theta\) and their rates. The input \(u\) is the turning curvature \(\kappa\) of the robot, controlled by actuating the steering joint. The measurement \(y\) are the lateral position error \(\hat{e}_y\), and the heading error \(\hat{e}_\theta\) acquired by the proposed row detection algorithm.

\[
\begin{align*}
    x &= (e_y, e_y', e_\theta, e_\theta') \\
    u &= (\kappa) \\
    y &= (\hat{e}_y, \hat{e}_\theta)
\end{align*}
\]

(28)

Based on the kinematic model of a car-like vehicle, the system model (Equation 8) becomes:
\[
\begin{bmatrix}
  e_y \\
  e_y' \\
  e_\theta \\
  e_\theta'
\end{bmatrix}_{n+1} = 
\begin{bmatrix}
  1 & \Delta t & 0 & 0 \\
  0 & 0 & s & 0 \\
  0 & 0 & 1 & \Delta t \\
  0 & 0 & 0 & 0
\end{bmatrix} 
\begin{bmatrix}
  e_y \\
  e_y' \\
  e_\theta \\
  e_\theta'
\end{bmatrix}_n + 
\begin{bmatrix}
  0 \\
  0 \\
  0 \\
  v
\end{bmatrix} + w
\]

\[
\begin{bmatrix}
  e_y \\
  e_\theta
\end{bmatrix}_n = 
\begin{bmatrix}
  1 & 0 & 0 & 0 \\
  0 & 0 & 0 & 1
\end{bmatrix} 
\begin{bmatrix}
  e_y \\
  e_y' \\
  e_\theta \\
  e_\theta'
\end{bmatrix}_n + v
\]

in which \(s\) is the separately controlled forward speed, and \(\Delta t\) is the control cycle time. The covariance matrix \(Q\) of process noise \(w\) was determined by the confidence level of the robot kinematic model. The covariance matrix \(R\) of measurement noise \(v\) was determined by evaluating the proposed row detection results.

**Experimental Evaluation**

The experiments were designed to evaluate the performance of our proposed navigation control system for in-field navigation, specifically with respect to (1) evaluate the performance of the DFT-based row detection algorithm for measuring the tracking deviation, and (2) evaluate the performance of the LQG controller for visual guidance.

**Row Detection Algorithm Evaluation**

The row detection algorithm was evaluated in terms of the accuracy of the tracking deviation estimation, including the lateral position deviation and heading deviation. The effect of different crop growth stages, sunlight directions, and robot positions in the field were tested and analyzed. Images for evaluating the developed crop row detection algorithm were collected in the fall of 2020 using our PhenoBot 3.0. In order to test the performance of the algorithm under various conditions, the images were taken at two different plant growth stages at different times during the day.
The images were collected at Iowa State University Agricultural Engineering and Agronomy Research Farm located in Boone, Iowa, when the corn plants were at the V3 and V7 growth stages. The field contained 12 crop rows and the length of each row is about 50 m, and featured a 76 cm (30”) row spacing and roughly 15 cm (6”) inter-plant spacing (Figure 4-7). The robot traveled at a speed of 0.8 m/s during data collection, and images were collected at 4 frames per second. The images were taken at noon time (~55° solar altitude angle) and evening time (~20° solar altitude angle). The weather condition was sunny during data collection, and the sunlight illuminance ranged from 20,000 to 80,000 lux. The camera exposure parameters were adjusted according to the illumination conditions. The ground truth of tracking deviation was generated according to the GPS-based localization, which fuse the information from an RTK-GPS unit, an IMU, and the encoders with an EKF. The ground truth of row centerlines was measured manually by surveying the GPS position of the start and end points.

*Figure 4-7. Passes used for row detection algorithm evaluation and navigation system evaluation. The testing field was surrounded by grasslands and grown corn plants. The numbered black path segments indicate that the proposed row detection algorithm was active while following the paths. For the rest of the path in yellow color, the proposed algorithm was not active, and the robot was manually driven.*

In order to evaluate the robustness of the algorithm with respect to image collected positions in the field and sunlight angles in the images, images were collected from four passes,
including two passes (2, 3) inside the field, and two passes (1, 4) close to the field edges in order to assess the row detection performance when only one side of the images is cover by crop rows. Images from passes 1 and 3 were collected facing the sunlight direction, thus has glares in the images. And images from passes 2 and 4 were collected along the sunlight direction, thus has the shadow of the robot in the images. Since the algorithm was not functioning when the robot approaching the end of a row due to lack of crop row patterns in the images, images taken less than about 8 m to the end were excluded from evaluation (Figure 4-7). In order to show the robustness of the algorithm to glares and shadows, two example sets of intermediate results of the detection algorithm are presented (Figure ). The two examples are from two different datasets taken at evening time, in which the plants were at V3 and V7 growth stages and the robot traversed in pass 3 and 4, respectively. In total, 6264 out of 6755 images (93%) were successfully processed and the crop rows were accurately detected. Most of the invalid images were caused by hard bumps in the field, which resulted in rapid camera pose changes that cannot be tracked by the IMU. For the valid images, the Pearson correlation coefficient (r) and root mean square error (RMSE) of the tracking deviation estimation were calculated and evaluated (Table 4-1). The average RMSE for lateral position and heading deviation estimation are 6.43 cm and 1.48°, respectively. The tracking deviation estimation results were found to be precise when compared with the ground truth of the two example passes (Figure 4-9). The algorithm was implemented in Python using OpenCV library. The average processing time for a frame was 252 ms with a standard deviation of 33 ms, using a computer with an i5-6500TE processor.
Figure 4-8. Step results of the proposed DFT-based row detection algorithm: (a) The raw images. (b) The images projected onto robot local coordinate. (c) The ROI images for applying the DFT-based row detection. (d) The gray-scale ROI image generated by ExG index. (e) The row detection results with crop row center lines superimposed in red color. The left images are taken at V3 growth stage, evening time, pass 3. And the right images are taken at V7 growth stage, evening time, pass 4.
Figure 4-8. Continued.
Table 4-1. The correlation index and rooted mean square error (RMSE) of the lateral and angular deviation estimation using the proposed row detection algorithm.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Growth stage / time of the day</th>
<th>V3 / Noon</th>
<th>V3 / Evening</th>
<th>V7 / Noon</th>
<th>V7 / Evening</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pass</td>
<td>1 2 3 4</td>
<td>1 2 3 4</td>
<td>1 2 3 4</td>
<td>1 2 3 4</td>
</tr>
<tr>
<td>Position deviation estimation</td>
<td>Correlation index (r value)</td>
<td>0.54 0.68 0.48 0.45 0.67 0.40 0.44 0.33 0.45 0.67 0.46 0.46 0.59 0.44 0.61 0.60</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSE (cm)</td>
<td>5.50 6.28 4.50 4.26 7.53 7.43 5.97 7.09 7.29 5.25 7.00 5.96 7.46 7.64 6.48 7.22</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heading deviation estimation</td>
<td>Correlation index (r value)</td>
<td>0.67 0.85 0.48 0.59 0.86 0.67 0.66 0.64 0.50 0.79 0.49 0.76 0.84 0.75 0.72 0.84</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RMSE (deg)</td>
<td>1.24 1.40 1.30 1.31 2.22 1.68 1.04 1.07 1.78 1.59 1.86 1.65 1.34 1.45 1.50 1.32</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Figure 4-9. Tracking deviation estimation of example passes, including (top) the estimated robot trajectory calculated from position deviation, and (bottom) the measured heading deviation compared with ground truth from GPS-IMU localization. (a) V3 growth stage, evening time, pass 3; (b) V7 growth stage, evening time, pass 4.

In order to investigate the factors that affect the accuracy of lateral position and heading deviation estimation, multiple statistical tests were conducted using data listed in Table 4-1, and the results were listed in Table 4-2. The null hypothesis of the tests are equal means. Only test 2 was found significant, which means the performance of the lateral deviation estimation is affected by the operation time of the day, i.e. the glares and shadows caused by sunlight with inclined solar angle will reduce the accuracy of lateral position estimation. But the accuracy is still acceptable. The difference of the accuracy with plants at different growth stages was not significant. Although the crop row patterns from images of later growth stages were much less obvious than that of earlier growth stages, and the row detection algorithm cannot locate all crop rows accurately, the heading and position
deviations were still estimated correctly. That was because the row direction was accurately estimated by referring the global pattern in the frequency domain, and the crop rows in the center of the images were accurately located since they have higher weights in the phase estimation step due to the Hanning window function.

In general, the algorithm was able to detect the crop rows and estimate the robot tracking deviation accurately. And the proposed method is robust to plant growth stages, operation position in the field, and illumination variations including glares and shadows in the images.

However, the proposed algorithm was found less promising when the camera orientation was poorly calibrated, in which the crop rows were not parallel and equally spaced in the simulated bird’s eye view. And the algorithm is not functioning at the end of the crop rows, since the crop rows has limited contribution to the target frequency in the images.

Table 4-2. Equal mean tests and results of the RMSE’s of the lateral position and heading deviation estimation by different factors.

<table>
<thead>
<tr>
<th>No.</th>
<th>Response</th>
<th>Factor</th>
<th>Test method</th>
<th>p-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Lateral pos RMSE</td>
<td>Growth stage (V3/V7)</td>
<td>t test</td>
<td>0.132</td>
</tr>
<tr>
<td>2</td>
<td>Lateral pos RMSE</td>
<td>Time of the day (Noon/Evening)</td>
<td>t test</td>
<td>0.011*</td>
</tr>
<tr>
<td>3</td>
<td>Lateral pos RMSE</td>
<td>Passes (Pass 1,2,3,4)</td>
<td>Oneway ANOVA</td>
<td>0.620</td>
</tr>
<tr>
<td>4</td>
<td>Heading RMSE</td>
<td>Growth stage (V3/V7)</td>
<td>t test</td>
<td>0.345</td>
</tr>
<tr>
<td>5</td>
<td>Heading RMSE</td>
<td>Time of the day (Noon/Evening)</td>
<td>t test</td>
<td>0.681</td>
</tr>
<tr>
<td>6</td>
<td>Heading RMSE</td>
<td>Passes (Pass 1,2,3,4)</td>
<td>Oneway ANOVA</td>
<td>0.551</td>
</tr>
</tbody>
</table>

Navigation System Evaluation

The second experiment is to evaluate the performance of the proposed navigation system in crop row tracking. The navigation system uses an LQG controller along with the visual-derived tracking deviation for robot steering control. The field tests were conducted in the same field used in the row detection experiment with the same passes (Figure 4-7). Using
the PhenoBot 3.0 as the testing platform, the experiment was conducted with corn plants at V3 growth stage at evening time. The actual robot trajectory was measured by GPS-based localization, and the tracking error was analyzed. The system was only active before reaching the row ends. The forward speed was controlled at 0.8 m/s. The sampling rate of the camera in the row detection algorithm was 3 frames per second, and the update frequency of LQG controller was 8 Hz.

From our experimental results, it can be concluded that the trajectory along which the robot navigated matched well with the reference central path (Figure 4-10) with reasonable distribution of the tracking error (Figure 4-11). The proposed algorithm with an LQG controller was capable to steer the robot to stay in the center between the crop rows with a mean absolute tracking error of 3.74 cm, and the position error was less than 8.5 cm for 90% of the cases.

![Figure 4-10. Trajectories of the robot using the proposed navigation system](image-url)
Figure 4-11. The distribution of the tracking error (lateral position), unit: m.

In summary, the results of this study showed that the proposed DFT-based row detection algorithm was capable of detecting crop rows accurately and contributing to the agricultural robot navigation in the field. Compared with other vision-based row detection algorithms, there are some benefits of using the proposed DFT-based algorithm:

1. The parallel property and the prior knowledge of the row spacing were properly used in the algorithm.
2. The algorithm focuses on the global frequency features in the images, therefore, the approach is robust to a variety of local noise such as weeds, glares, and shadows.
3. The computational load for processing each image is low and relatively consistent. And only few configuration parameters are involved in the algorithm.

The limitations of the algorithm are:

1. The algorithm relies on accurate camera pose estimation for field-based navigation, which requires accurate robot-camera calibration and real-time sensing of robot pose and posture.
2. A row-end-detection method is required since dominant frequency component from the crop rows was diluted by the non-crop area at the end of the rows.
3. The RMSE is about 6 cm for lateral position deviation estimation, therefore, a controller compatible with noisy input signal is necessary for robot navigation control, such as the LQG controller used in this study.

**Conclusion**

In this paper, a frequency domain-based crop row detection algorithm was proposed. Based on the assumptions of straight, parallel, and equally-spaced crop rows, a novel image processing pipeline was developed to detect crop rows for agricultural robot navigation. A series of techniques including interpolation, windowing, and DTFT were applied to improve the detection accuracy of the peak and phase in frequency domain. A LQG controller was used with the proposed algorithm for robot navigation between crop rows. Field tests were conducted to evaluate the row detection performance with corn plants at different growth stages, under different illumination conditions, and at different positions in the field. The results showed that the proposed crop row detection algorithm was able to estimate lateral position and heading deviations relative to the row center accurately with plants at V3 to V7 growth stages. The average RMSE for lateral position and heading deviation estimation are 6.43 cm and 1.48°, respectively. With the LQG controller, the robot was able to follow the centerline of crop rows with a mean absolute tracking error of 3.74 cm. The tracking error was less than 8.5 cm for 90% of the cases.

Our study demonstrated that DFT and frequency domain methods are feasible for crop row detection in agricultural robot navigation. The parallel property and the prior knowledge of the row spacing was properly used in the proposed algorithm. Focusing on the global frequency feature, the algorithm is robust to local noise such as weeds, glares, and shadows. However, the algorithm relies on accurate robot-camera calibration and real-time robot pose and posture sensing.
In summary, the presented algorithm and field-testing results demonstrated the feasibility of using frequency domain analysis for robotic navigation between the crop rows while revealed its limitations which can lead to further investigations. The framework of the proposed algorithm can also be extended to different types of mobile robotic operations in the field with above-canopy cameras. It can also be used in platforms capable of generating accurate top-view images, such as unmanned aerial vehicles (UAVs).

Acknowledgment:

This project was funded by the National Science Foundation Major Research Instrumentation Program (Grant No. DBI-1625364) and the Plant Sciences Institute at Iowa State University. We thank Taylor Tuel, Benjamin Steward, and Duan Liu for their assistance in hardware design.

Appendix:

Solving Homography Matrix

The homography matrix \( H \), which transforms the image coordinate \((x_i, y_i)\) to the map coordinate \((x_m, y_m)\) in homogenous coordinates can be calculated through the camera projection. Given the camera pose \( P_{cam} \) relative to the map, we have:

\[
\begin{align*}
\begin{bmatrix}
    x_i \\
    y_i \\
    1
\end{bmatrix} &= K[I \mid 0] P_{cam}^{-1} \begin{bmatrix}
    x_m \\
    y_m \\
    0
\end{bmatrix} \\
\begin{bmatrix}
    x_i \\
    y_i \\
    1
\end{bmatrix} &= K[I \mid 0] \begin{bmatrix}
    e_1 & e_2 & e_3 & t \\
    \mid & \mid & \mid & \mid \\
    0 & 0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
    x_m \\
    y_m \\
    0
\end{bmatrix} \\
\begin{bmatrix}
    x_i \\
    y_i \\
    1
\end{bmatrix} &= K \begin{bmatrix}
    e_1 & e_2 & t \\
    \mid & \mid & \mid \\
    0 & 1
\end{bmatrix} \begin{bmatrix}
    x_m \\
    y_m \\
    1
\end{bmatrix}
\end{align*}
\]
in which $K$ is the camera projection matrix and acquired through the camera intrinsic parameter calibration.

**2D Fourier Transformation and the Limitations:**

2-D Discrete Fourier Transform (DFT) was used to compute the Fourier Transform of an image. The DFT applies with discrete data, which is not continuous in space domain. The 2-D DFT is given by:

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(u/M + vy/N)}$$

where $f(x,y)$ is an M by N image, and the $F(u,v)$ is the M by N 2D Fourier Transform of that image. And the inverse 2-D DFT (IDFT) is given by:

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v)e^{j2\pi(u/M + vy/N)}$$

Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) are two algorithms that compute the DFT and IDFT of a sequence in a fast way. They greatly reduced the complexity of computing DFT and IDFT from $O(n^2)$ to $O(n \log n)$.

However, DFT is the sampled Fourier Transform. Therefore, it does not contain all frequencies forming an image, and only a set of sample frequencies large enough to fully describe the spatial domain image were included. Thus, in the results of FFT, it is almost impossible to accurately find the crop rows directly, due to the following limitations:

**Limitation 1: low frequency resolution**

The DFT and FFT’s frequency resolution depends on the length of data used. From equation 11, the frequency bins are \{u/M, u = 0, 1, ..., M/2 (Nyquist limit)\} in horizontal, and \{v/N, v = 0, 1, ..., N/2\} in vertical. Unit is (1/pixel). And the bins for period are \{M,
M/2, M/3, …, 2} for horizontal, \{N, N/2, N/3, …, 2\} for vertical. Unit in pixels. It is obvious that the resolution for period bins are extremely low at low frequencies.

For crop row detection application, if the extracted bird’s eye view image is 512 x 512 in resolution, and 0.01m/pixel in spatial resolution, the theoretical period should be about 75 pixels for 0.75 m row spacing. And the field of view covers about 6 vertical crop rows. The bins around 75 pixels are \{512/4 = 128, 512/5 = 102, 512/6 = 85, 512/7=73\}. If the period was directly extracted from the frequency bins, it will potentially result to a period estimation error within [-12, +17] pixels, which is equivalent to about 15 cm. And the error will become more significant due to the inaccurate angle estimation, which is too high for row-guidance applications.

**Limitation 2: spectral leakage**

The boundary of image act as a rectangular window to the signal. When the signal in the rectangular window has an integer cycles, the DFT result is correct (Figure 4-12, blue signal). When the signal in the rectangular window has a fraction of cycles, the energy will “leak” to other frequency bins (Figure 4-12, red signal). The symptom is called “spectral leakage”.

In the row detection application, the number of cycles is usually not an integer, which leads to spectral leakage if using rectangular windows. It is inaccurate to extract the peak directly from the DFT result.
Spectral leakage caused by “windowing”

Figure 4-12. Demo of spectral leakage effect. The blue signal has 13 cycles in the rectangle window, and the DFT results (the dots) are precise. The red signal has 13.5 cycles in the window, and the DFT result has Spectral Leakage. Image retrieved from Wikipedia website: https://en.wikipedia.org/wiki/Spectral_leakage

References


CHAPTER 5. GENERAL CONCLUSION

Conclusions

In this study, a navigation control system was developed for a central-articulated robotic vehicle designed for field-based high-throughput phenotyping. Firstly, a navigation control pipeline compatible with central-articulated robotic vehicle was designed. Secondly, the capability and feasibility of different navigation strategies including GPS-based path tracking, and computer vision-based localization, row mapping and row tracking were studied.

Within the developed navigation control system, different functional modules were developed and implemented, which include the motion control module based on robot kinetic model, the robot localization module using single RTK-GPS signal, the path tracking module with multiple tracking algorithms implemented, and the computer vision-based row mapping and in-field localization module using different sensor setups. A Gazebo-simulated environment was constructed to aid the system development and test the performance of different path tracking algorithms.

Path tracking based on GPS localization is the most common navigation strategy for agricultural robotic vehicles. Three specific path tracking algorithms including Linear-Quadratic Regulator (LQR), Pure Pursuit control (PPC) and Timed Elastic Band (TEB) were implemented for the path tracking module. Both the simulation-based and field-based experiments demonstrated the capability of the proposed navigation control system in guiding the PhenoBot 3.0 robot to follow predefined paths on uneven terrain in the field. The system achieved the best path tracking performance with the LQR steering control algorithm in both simulation and field environments. The mean absolute tracking errors (MATE) of tracking a testing path composed of straight lines and curves were 5.72 cm and 7.01 cm in simulation
and field tests, respectively. Specifically, the MATEs for tracking the straight-line sections were 4.48 cm and 5.67 cm in simulation and field tests, respectively.

Computer vision-based modules using different sensor setups were developed for situations where global localization is denied or a pre-defined path is not available. Crop rows were detected in order to locate the robot, create field maps, and navigate the robot through row-guidance. The feasibility of using a front-facing under-canopy ToF camera was investigated for vision-based robot localization and mapping in corn/sorghum fields. The results of algorithm evaluation showed that our proposed row detection algorithm was able to provide reliable crop row detection results as long as three requirements were met: known crop row spacing, straight crop rows, and flat and level terrain. In the field tests, the proposed system was able to map the crop rows with mean absolute errors (MAE) of 3.4 cm and 3.6 cm in corn and sorghum fields, respectively. And it provides visual odometry data with MAE of 8.9 cm and 8.2 cm for robot positioning in corn and sorghum crop rows, respectively. The test results showed that using a front-facing ToF camera for navigation has certain feasibilities. The ToF-camera-based system encounters less sensor occlusion and is less prone to soil-unevenness compared with LiDAR-based systems. However, due to the limited sensing range, narrow horizontal field of view and unstable data quality of the ToF camera, the performance of the visual odometry is less promising when using a single sensor data frame each time.

A frequency domain-based crop row detection algorithm using a top-view color camera was developed. A series of techniques including interpolation, windowing, and DTFT were applied to improve the detection accuracy of the peak and phase in frequency domain. An LQG controller was used with the proposed algorithm for robot navigation between crop rows. Field tests were conducted to evaluate the row detection performance with corn plants at different growth stages, under different illumination conditions, and at different positions.
in the field. The results showed that the proposed crop row detection algorithm was able to estimate lateral position and heading deviations relative to the row center accurately with plants at V3 to V7 growth stages. The average RMSE for lateral position and heading deviation estimation are 6.43 cm and 1.48°, respectively. With the LQG controller, the robot was able to follow the centerline of crop rows with a MATE of 3.74 cm. The tracking error was less than 8.5 cm for 90% of the cases. The results demonstrated that the frequency domain method is feasible for crop row detection in agricultural robot navigation. Focusing on the global frequency feature, the algorithm is robust to local noise such as weeds, glares, and shadows. However, the algorithm relies on accurate robot-camera calibration and real-time robot pose and posture sensing.

**Future Works**

The performance of the navigation control system is challenged by the semi-structured environment in the field, including the uneven terrain, variable plant shapes and the weed infection. The reliability and performance of field-based navigation control can be improved in several aspects:

1. The motion control module was designed based on the 2D bicycle kinematic model under the assumption that the field is a planar surface. The accuracy of motion control was affected by the terrain unevenness and the soil type. One solution to improve the robustness of motion control is to include the 3D dynamics of the robotic vehicle and the soil mechanics in robot modeling.

2. The robot localization also assumes the robot is operating on a planar field surface. However, while operating on uneven terrains or rolling terrains, the data from motion sensing sources such as the wheel odometry and the IMU is inconsistent with the robot motion in the 2D map plane. One solution to improve the localization
performance is to expand the current localization capacity to track the 3D spatial pose and motion of the robot.

3. For the path tracking module, the performance of TEB is still improvable by including the penalty of steady state error in the cost function and tuning the weight parameters, or including feedback control (Bai et al., 2019). In addition, many algorithms such as fuzzy logic control (Xue, Zhang, & Grift, 2012), H∞ control (Velasquez et al., 2020) were also reported effective in field-based navigation and worth trying in our application.

4. In order to improve the reliability of ToF-camera-based row detection, some suggested solutions are: a) expanding the field of view by introducing other sensors, e.g. another back facing ToF camera on the robot, or laser range sensors on the side of the robot; b) improving the data reliability by using an auxiliary sensor such as a color camera, which would help extracting critical structural features in row detection; c) using a Kalman Filter based on the robot kinematic model to refine the visual odometry. The error level from the field tests can be used as a guidance for parameter tuning; d) fusing the historical data to increase the data density while adding the temporal dimension to the data, which can potentially improve crop row detection, as what proposed in Vitor et al. (2019)’s LiDAR-based system.

5. For row detection using a top-view color camera, the algorithm was found less promising when the camera orientation was poorly calibrated. The suggestion to improve the reliability is to improve the accuracy of posture estimation and camera-robot calibration. And an algorithm for detecting the end of the rows is necessary.

References
