Under canopy LiDAR-based autonomous navigation

Vitor A. H. Higuti [*]
Mechanical Engineering Department
University of Sao Paulo
Sao Carlos, Sao Paulo, Brazil
vitor.higuti@usp.br

Andres E. B. Velasquez
Mechanical Engineering Department
University of Sao Paulo
Sao Carlos, Sao Paulo, Brazil
andresbaquero@sc.usp.br

Daniel Varela Magalhaes
Mechanical Engineering Department
University of Sao Paulo
Sao Carlos, Sao Paulo, Brazil
daniel@sc.usp.br

Marcelo Becker
Mechanical Engineering Department
University of Sao Paulo
Sao Carlos, Sao Paulo, Brazil
becker@sc.usp.br

Girish Chowdhary
Agricultural and Biological Engineering
Coordinated Science Lab (CSL)
University of Illinois at Urbana-Champaign
1304 W Pennsylvania Ave
Urbana, IL, 61801
girishc@illinois.edu

Abstract

This paper describes a LiDAR-based autonomous navigation system for an ultra-lightweight ground robot in agricultural fields. The system is designed for reliable navigation under cluttered canopies using only a 2-D Hokuyo UTM-30LX LiDAR sensor as the single source for perception. Its purpose is to ensure that the robot can navigate through rows of crops without damaging the plants in narrow row-based and high-leaf-cover semi-structured crop

[*] This research was carried out during the time Vitor Higuti visited Chowdhary's group at the University of Illinois at Urbana-Champaign, supported by grant no. 2017/00033-7, Sao Paulo Research Foundation (FAPESP). The information, data, or work presented herein was funded in part by the Advanced Research Projects Agency-Energy (ARPA- E), U.S. Department of Energy, under Award Number DE- AR0000598

plantations, such as corn (*Zea mays*) and sorghum (*Sorghum bicolor*). The key contribution of our work is a LiDAR based navigation algorithm capable of rejecting outlying measurements in the point-cloud due to plants in adjacent rows, low-hanging leaf cover or weeds. The algorithm addresses this challenge using a set of heuristics that are designed to filter out outlying measurements in a computationally efficient manner, and linear least squares is applied to estimate within-row distance using the filtered data. Moreover, a crucial step is the estimate validation, which is achieved through a heuristic that grades and validates the fitted row-lines based on current and previous information. The proposed LiDAR-based perception subsystem has been extensively tested in production/breeding corn and sorghum fields. In such variety of highly cluttered real field environments, the robot logged more than 6 km of autonomous run in straight rows. These results demonstrate highly promising advances to LiDAR based navigation in realistic field environments for small under-canopy robots.

1 Introduction

Reliable autonomous navigation in cluttered crop-canopies where Global Navigation Satellite System (GNSS) signals are unreliable is the key to enable low-cost, low-weight, and compact agricultural robots. This is particularly true for small robots that are designed to go under the canopy to obtain useful measurements or perform management activities. GNSS, and especially Real-Time Kinematic (RTK or Differential) GNSS, is a reliable source for navigation for large equipment and robots that navigate above the crop canopy. However, GNSS signal can be unreliable under the crop canopy due to multi-path errors and signal attenuation. In particular, our test runs (see Section 5) indicated over 0.4 m error in GNSS signal with a high-end (>7K USD) RTK GNSS antenna from Trimble.

The main contribution of this paper is in demonstrating the feasibility of reliable navigation under crop-canopy using *low-cost* (under USD 5K off-the-shelf) 2-D LiDAR Hokuyo UTM30-LX, without using GNSS, or any other sensor. The contribution is significant beyond existing work in autonomous agricultural robots (surveyed in Section 2) because of its emphasis on using only a low-cost 2-D LiDAR, the ability to handle cluttered and leafy under-canopy environments, and extensive field testing in real production and breeding fields. Our results are remarkable due to two main reasons: 1. They establish that row-keeping navigation of small robots under cluttered canopies without GNSS is feasible. To do so, field trials in later growth stages in corn, sorghum and soybean were performed. 2. They pave the way for follow-on work where other sensors (such as vision or inertial) may be fused with LiDAR to create robust under-canopy perception systems that will enable a new breed of low-cost, compact, and highly autonomous field robots.

The current phenotyping practices are highly labor intensive and prone to human measurement errors, which has led to the *phenotyping bottleneck*. To overcome such issue preventing rapid advances in plant breeding (Furbank and Tester, 2011; Rahaman et al., 2015), low-cost robots could be the key technology as cost has been the primary barrier in adoption of robotic technologies in agriculture (Pedersen et al., 2006). Indeed, overcoming the *phenotyping bottleneck* is the single most important challenge in ensuring food security for an increasingly populous globe. Especially since according to United Nations projections, with 95% confidence, global human population will be between 9.2 and 10.1 billion in 2050 from the 7.3 billion in mid-2015 (Department of Economic and Social Affairs of the United Nations, 2015). To meet this growing population, the Food and Agriculture Organization of the United Nations has recommended that global production of food needs to increase by 70% by 2050, relative to 2009 levels (FAO, 2011).

There has been significant work on automating the analytics of phenotyping through computer vision and other sensors (Fahlgren et al., 2015; Mahlein, 2016). However, the process of obtaining data still remains largely manual. Furthermore, heavier phenotyping robots such as the very recent Mueller-Sim et al. (2017) among others, seems to be cost-prohibitive and may damage plants when they run over them. On the other hand, the LiDAR based navigation system presented here is demonstrated on low-cost ultra-lightweight and compact robots, and will enable further development of such robots. Furthermore, looking beyond phenotyping, low-cost robots could also be the key to enabling a new class of equipment that is better suited for intricate tasks in a team-based approach rather than single large fields equipment pulled behind an expensive tractor. Nevertheless, the study presented here can also be utilized for creating navigation systems for heavier robots that operate in cluttered environments.

This paper addresses the problem of autonomously navigating in narrow row-based and high-leaf-cover semi-structured crop plantations, such as corn (*Zea mays*) and sorghum (*Sorghum bicolor*). Section 2 presents the related work for autonomous ground robots in agriculture environments. Besides mentioned crop and field characteristics, the use of an ultra-lightweight ground robot (presented in Section 3) plays a significant role in the method suitability for the row-following task. Hence, we propose a novel LiDAR-based perception subsystem in Section 4, and show and discuss its experimental results in Section 5, concluding in Section 6.

2 Related Work

Lightweight ground robots are a class of equipment that address the critical niche between heavy agricultural equipment and human utilized tools. They enable versatile applications in croplands, and enable location-specific (high-precision) management or remote-sensing of crops. As such there have been many ground robotic platforms investigated for agricultural applications. Robotanist (Mueller-Sim et al., 2017) is used for automate phenotyping, Hall et al. (2017) introduces weed scouting robot AgBotII, Thorvald platform (Grimstad et al., 2015) is designed for multi-purposes e.g. seeding, weeding and harvesting, Haibo et al. (2015) presents a wheat precision seeding robot, and Torgersen (2014) provides in-depth details for the design of the NMBU Mobile Robot. A key difference between the one used in this work and mentioned robots is that the former is an order of magnitude lighter (just below 10 kg) compared to the latter ones (over 100 kg). Additionally, Bechar and Vigneault (2016) reviews agricultural robots and their components for field operation. Attention is given to devices used for sensing and self-localization, mainly those considered external sensors. They obtain current state of the environment relative to the position of the robot and its components. Examples of such sensors are: Global Navigation Satellite System (GNSS), cameras and Light Detection And Ranging (LiDAR).

A GNSS system, typically augmented with Real-Time Kinematic (RTK) differential correction, can provide highly accurate latitude and longitude information when it has unobstructed view of GNSS satellites. Indeed, GNSS-based navigation systems have been guiding automatic agricultural tractors (Bell, 2000; Nørremark et al., 2008; Zhang et al., 2016; Wang and Noguchi, 2016) and even smaller over-the-canopy autonomous robots (Bak and Jakobsen, 2004; Bakker et al., 2011). However, GNSS receivers suffer from occlusion, attenuation, and multi-path errors under dense crop canopies. Furthermore, GNSS cannot detect unexpected obstacles such as animals, humans, stubble, or machines. Therefore, GNSS alone is not able to provide safe navigation of autonomous mobile robots in agricultural environments (Reina et al., 2016). Moreover, Rovira-Más et al. (2015) highlights that long-term navigational solution stability in agricultural field cannot be granted as multi-path reflections and signal-blockage are unpredictable, requiring redundancy and continuous fail-safe checking. Although cited works may not badly suffer from such GNSS issues as the platforms are big enough to have the receiver well above the canopy, the same cannot be expected when the vehicle drives under the canopy, such as when operating in orchards (Bergerman et al., 2015), vineyards (dos Santos et al., 2015) and late season corn, soybean, and sorghum as in this paper.

As such, there has been much work in GNSS-free navigation systems capable of autonomously navigating robots through crop lanes. Used devices are usually those capable of capturing actual state of the robot's surroundings: LiDAR and cameras. The first is able to accurately measure a relative vectorial position (distance and direction) of nearby objects. And the second is a highly versatile sensor, usually operating in the infrared, near-infrared or visible spectrum to extract both color and depth information (Bechar and Vigneault, 2016).

Cameras have been actively used for the crop-row identification task (Montalvo et al., 2012; Jiang et al., 2015; Zhao and Zhang, 2016; Zhai et al., 2016). Although the methods vary, they have a common characteristic of needing an image from above canopy. Additionally, such vision-based systems are usually devised for tractors, which possess three features in advantage to small (under canopy) vehicles: 1. A higher camera position leading to a broader view and thus information of several meters ahead can be acquired at once; 2. GNSS systems may be used between processing loops to maintain the vehicle on track; 3. Larger vehicles are less prone to soil unevenness. In maize fields, Guerrero et al. (2017); García-Santillán et al. (2017) detect straight crop rows even in the presence of weeds. Although the second also works for curved rows, both are limited to initial growth stages. As they are proposed for tractors, the reached processing time of 0.6 s for both methods is reasonable. But this time may signify a collision considering an under canopy vehicle within a nominal 0.8 m width crop lane subject to: 1. Unreliable GNSS information; 2. Constant sensor occlusions due hanging leaves and weeds; 3. Greater effect from soil unevenness; 4. Forward movement about 0.2 m/s. Also in maize field, Hiremath et al. (2014a) demonstrates an image-based particle filtering for navigation with a robot that is slightly bigger than ours. But it required a downward-looking camera at a height of 1.65 m, which limits the use of such method in later crop stages. Moreover, the system was tested for plant sizes in the earlier stages (ranging from 0.1 m to 0.5 m).

English et al. (2014) addresses the common issues of sensibility to lightning changes and of requiring knowledge about crop-specific features such as color, spacing and periodicity. They propose a novel method that extracts and tracks the direction and lateral offset of the dominant parallel texture in simulated overhead view of the scene. But again the method requires a view from above canopy. Moreover, the algorithm assumes a leveled camera attitude with respect to the ground. Although an external stabilization is not performed, this camera attitude condition is satisfactorily achieved for larger vehicles, which tend to sit relatively flat to local terrain. These two conditions, easily checked when the platform is a tractor, are hardly observed on smaller robotic platforms, especially those going sub-canopy.

LiDAR has the advantage of directly supplying distance measurements, being less sensible to environment conditions and possessing an extended range compared to other sensors (Bechar and Vigneault, 2016). Although traditionally used in indoor applications (e.g. industry and warehouses), the past decade reducing costs sparked interest in using such technology in new areas such as agriculture. In the case of row following tasks, most research with LiDAR is focused on orchards (Barawid et al., 2007; Zhang et al., 2013; Bergerman et al., 2015; Bell et al., 2016). For well-structured tree rows, 2-D LiDAR suffices to provide enough information for row following (Bergerman et al., 2015). But when ground vegetation occludes tree trunks, Zhang et al. (2013) uses a rotating 2-D LiDAR to retrieve a 3-D point cloud and then accurately obtain robot's position. Similarly, Bell et al. (2016) employs a Velodyne VLP16 (3-D LiDAR) to handle sparseness in pergola structured orchards.

Corn and sorghum (CS) crops greatly differ from orchards. Instead of well-defined tree trunks in orchards, weeds and hanging leaves can be easily mistaken with or even occlude the stalks in the mentioned crops. Moreover, normal spacing between tree rows is few meters while it is around 0.8 m in CS crops. Hence, a dense and cluttered environment is expected for CSS crops, especially in later growth stages. In corn crops, some works with 2-D LiDAR-based row-following systems can be found (Hiremath et al., 2014b; Troyer et al., 2016; Velasquez et al., 2016).

Hiremath et al. (2014b) proposes a LiDAR measurement model that is directly used by the particle filter (PF) algorithm needed for the autonomous navigation. The PF is able to handle different kinds of uncertainties, e.g. noise due uneven terrain and the varying colors, shapes and sizes of the plants, and accurately estimates the robot-environment state of the system such as robot heading and lateral deviation. Nonetheless, this method is limited to plants smaller than 0.6 m, which corresponds to the situation where leaves are not extending to the middle of the rows.

Troyer et al. (2016) demonstrates a row-following platform using two single-reading LiDAR sensors. It presented reasonable results for indoor case, but the system did not perform well even on simulated corn rows (flat terrain and PVC as corn plants). As the authors pointed out, the varying gaps between stalks provided navigation reference inconsistencies leading to larger errors. Moreover, as the sensors are able to provide just a single reading at a time, only one distance measurement for each side is available and it may not even correspond to the stalk.

Velasquez et al. (2016) presents a LiDAR-based row-following system tested on a simulated corn crop (corn plants in pots arranged in straight rows). To generate the distances to rows, it considered a large set of the readings inside a reasonable big region of interest (ROI) without checking if such points pertained to the row. Thus it was prone to errors due hanging leaves, weeds or even unexpected objects such as a person walking in the neighboring lane. Moreover, the definition of ROI depended upon heading measurements with Inertial Measurement Unit device. The issue rises from the fact that the reference value for a heading angle parallel to rows is obtained only once in the beginning of the run. Therefore, on-the-run changes in the row direction would lead to system failure.

In summary, navigating rows in the modern row-crop agriculture environments considered in this paper remains a highly challenging problem due to the following open issues:

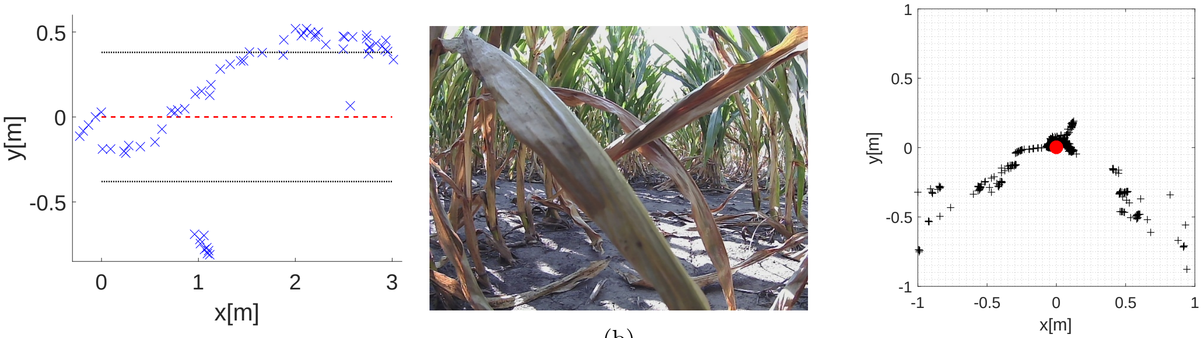
- Lack of reliable GNSS under canopy: The coverage greatly affects the signal reception by onboard receiver due to issues of multi-path and signal attenuation. For navigation purposes in crops with standard lane width of 0.8 m, a positioning system with error greater than a few centimeters may not guarantee crop safety against collisions. Figure 1a provides information recovered from Trimble receiver operating in RTK mode along 3 m of a sorghum plot. The data was collected keeping the antenna in the middle of a 0.76 m lane width;

- Highly cluttered rows: Presence of weeds, leaves and even fallen stalks brings a dynamic scenario that may be erroneously perceived as part of the crop rows. Moreover, such elements may impose movement restriction;

- Varying plant spacing and mildly varying lane width: Although the seeding process may be centimeter-level accurate and the seeds form equally spaced straight lines, some of them may not germinate leaving gaps and the stalks may not grow straight up, for instance having a natural angle of growth, being logged or bent by wind. Thus, the lane width is different depending on the part of the plant that is used for the measurement.

- Frequent occlusion of the sensor: An exteroceptive sensor retrieves information from the environment. Thus, its occlusion hinders the robot's capability of knowing its surroundings and, consequently, avoiding collision. For instance, hanging leaves that cover the sensor (Fig. 1b) provoke sudden changes in the readings, greatly reducing available information (Fig. 1c). From field tests, it has been determined that about fifty frames (∼2 s) would be needed to disentangle a forward moving robot from such condition;

- 2-D point cloud: Single layer LiDAR sensors are limited to a 2-D set of distance measurements. Hence, there is a huge loss of detail, which for instance could be used to detect unwanted features such as weeds and leaves.



(a)

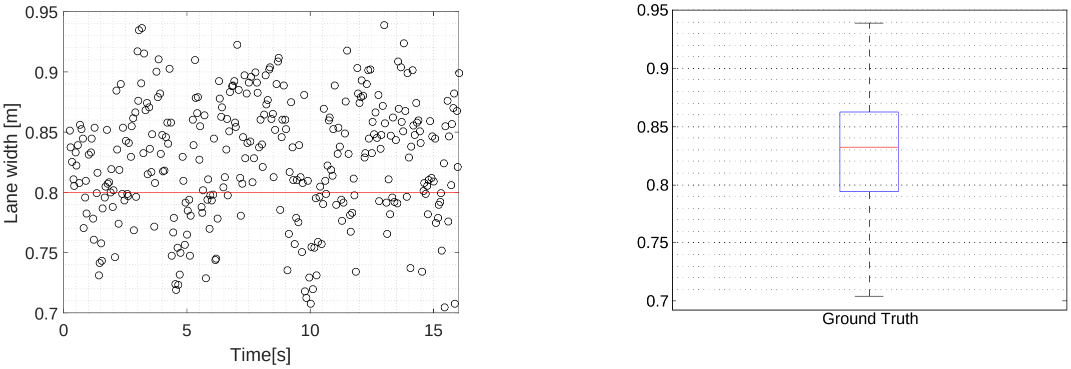
(b)

(c)

Figure 1: (a) A small robot (completely under canopy) was manually driven in the middle of 3 m sorghum lanes with 0.76 m nominal width and height over 2 m. In such conditions, plot (a) illustrates a typical absolute positioning (crosses) obtained with a RTK-GNSS receiver attached to the robot at a 0.5 m height. Note that under canopy, RTK-GNSS cannot reliably provide positioning within its usual precision (around 50 mm). For reference, the dashed line marks the expected positioning in the middle of the lane and the dotted lines represent the lateral rows. (b) The same robot is equipped with a camera and a LiDAR, both positioned at the center of the robot's frontal part. This picture shows the robot's surroundings in a corn crop. As the robot moves in the direction of the picture, the leaves block both camera and LiDAR field of view. (c) A single LiDAR scan few instants after (b). Crosses are the distance readings and the circle at origin represents the sensor, whose field of view is blocked by the leaves in the middle of path.

Even though the sensor presents a millimeter-level variance, the practical measurement of the lane width along the lane presents several issues. Each plant grows differently, leading to a nonlinear shaped sequence of plants. Moreover, after fitting the best line to each row, the expected parallelism does not exist. We confirmed such considerations performing manual ground truth for each frame from a LiDAR measurement set in cornfield (Fig. 2). The procedure for manual ground truth is explained in Sec. 5.1. As it can be seen in Fig. 2a, there is a significant difference between actual and nominal lane width. Although a continuous sequence of values could be expected, two consecutive measurements may change up to 0.17 m, having an average absolute difference of 0.04 m for the studied set. Finally, Fig. 2b allows the analysis of lane width values: Despite of being manually done, there is a huge variance and noise in the lane width estimation.

It should be noted that in addition to the field navigation literature discussed above, there is a significant amount of work in simultaneous localization and mapping using LiDAR and other sensors (Montemerlo et al., 2002; Wang et al., 2003; Durrant-Whyte and Bailey, 2006; Thrun and Leonard, 2008; Sobers et al., 2010; Cheein et al., 2011; Lepej and Rakun, 2016). A large body of that work focuses on GNSS denied environments, including indoor environments, riverine environments, operation near infrastructure, and orchards/vineyards. Although the researches above mentioned were quite inspiring, their methods are not applicable to under canopy navigation without the contributions presented in this paper. This is so because the environments that under canopy robots have to navigate are quite different from urban, indoor, or riverine environments.



(a)

(b)

Figure 2: A small robot is manually driven through a corn crop while recording single layer LiDAR data. Such measurements were processed to obtain the lane width, a value that encloses both left and right lateral distances, thus alleviating unexpected changes in the robot's pose due to the uneven terrain. Compared to building walls, even though seeds could be precisely planted, the stalks grow differently in size and especially in direction. Thus, the definition of the lateral row is highly dependent on which stalks are visible. To illustrate the lane width variance down a crop lane, plot (a) provides manual ground truth using a single layer LiDAR measurements. The lane width obtained from each scan is marked with circles. The full line marks the expected value. Moreover, plot (b) summarizes the ground truth with a boxplot (0.827 m mean value and 0.05 m standard deviation).

In particular, under canopy row navigating robots face a large amount of clutter due to leaves, debris, brushing leaves, and very tight row-spacing.

3 TerraSentia Robot

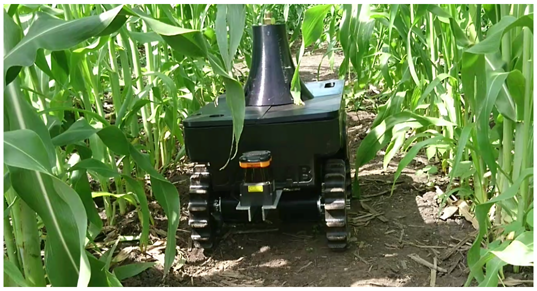
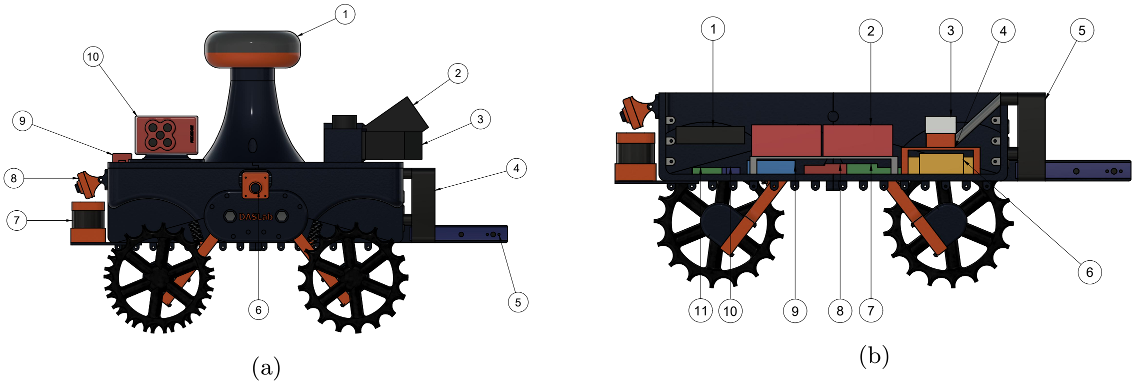


Figure 3: TerraSentia robot (0.31 m wide) in sorghum field. This crop has a nominal lane width varying between 0.76 m and 0.91 m. A LiDAR Hokuyo UTM-30LX is positioned in the front part.

This section describes the low-cost, ultra-compact (0.31 m wide) and ultra-light (6.6 kg) 3D printed field robot TerraSentia (Kayacan et al., 2018). It is a novel robot conceived to automate the measurement of plant traits for efficient phenotyping. Because of its design, it does not permanently damage plants even if it drives over them. Nonetheless, driving over plants is not a desired situation. Hence, such platform requires mechanisms that are capable of keeping it in the clear paths. This way, the perception subsystem proposed in this work allows the navigation within row-crops without hitting plants, and subsequently, it is tested using the mentioned robot. Figure 3 shows TerraSentia in sorghum plot, Fig. 4a presents a CAD drawing of the robot with a selection of available sensors and Fig. 4b depicts the internal components.



(a)

(b)

Figure 4: (a) TerraSentia CAD drawings with a suite of sensors. 1. GNNS antenna, 2. Bayspec hyperspectral sensor, 3. Bayspec hyperspectral sensor (sideward facing), 4. Radiator for the liquid cooling system, 5. Mount for 3D Sensor Intel RealSense, 6. Embedded visual sensor, 7. LiDAR sensor, 8. Embedded visual sensor, 9. GNNS receiver for RedEdge multispectral sensor, 10. RedEdge multispectral sensor. Reprinted from Kayacan et al. (2018). (b) CAD drawing showing TerraSentia interior. 1. Raspberry Pi 3, 2. Lithium Ion Batteries, 3. Tegra, 4. Heat sink, 5. Cooling Fan, 6. Sabertooth Board, 7. Kangaroo Board, 8. 3-axis gyroscope, 9. Voltage regulator, 10. Raspberry Pi Cobbler, 11. Printed Circuit Board (PCB). Reprinted from Kayacan et al. (2018).

A brushed 12V DC motor with a 131.24:1 metal gearbox (Pololu Corporation, USA) drives each of its four wheels - a two-motor set for left side and another for right one. Each two-motor set is driven by one of the Sabertooth Motor Controller channels, whose nominal supply current is 12A per channel. Furthermore, a two-channel Hall-effect encoder (Pololu Corporation, USA) provides 64 counts per revolution for each motor, enabling wheel velocity measurement. The self-tuning PID controller Kangaroo x2 Motion Controller (Dimension Engineering, USA) uses information from encoders to maintain desired angular and linear velocity commands.

In contrast to common practice in building agricultural robots (Bak and Jakobsen, 2004; Ruckelshausen et al., 2009; Tabile, 2012; Haibo et al., 2015; Grimstad et al., 2015; Mueller-Sim et al., 2017) , most of the mechanical parts, including wheels, are built through additive manufacturing with polylactic acid (PLA). The capability of controlling infill percentage enabled optimal stress distribution throughout structure, resulting in overall ultralight robot while preserving strength and durability.

A LiDAR sensor needs to be chosen based on the angular resolution, range and update rate needed. Consider a stalk with nominal width of 0.01 m. This is a reasonable minimum width for corn and sorghum. For a robot in the middle of a 0.8 m lane width, sensor measurements of stalks are in the range 0.2 m to 0.6 m for the immediate sideward readings. In this scenario, a 0.47° angular resolution provides at least two readings from the nearest stalks. In cluttered environments, a sensor would be able to get distinguishable information for only the next few meters. Nevertheless, an extended distance range would be useful for row end maneuvers. Finally, the robot would traverse the crop with speeds not exceeding 1 m/s. At such speed and for a robot in the middle of the lane with a heading error of 30° with respect to the rows, it takes around 0.6 s to hit a row. Therefore, it is reasonable to expect that sensor updates its readings at least in the order of 0.1 s. Needless to say, such sensor should be suitable for outdoor applications. A sensor that complies with discussed requirements is Hokuyo UTM-30LX, a 2-D LiDAR sensor which covers a 270° range with 0.25° angular resolution, maximum distance reading of 30 m and 40 Hz update rate. Anecdotal experiments show that the under USD 2K off-the-shelf Hokuyo UST-10LX 2-D LiDAR provides similar results with the proposed perception subsystem. Although a 3-D LiDAR sensor would provide a lot more information, its price (at least USD 8K) and computational toll required for processing incoming data make 3-D LiDAR sensor less suitable for a small and low-cost robot as TerraSentia.

Given the single layer (2-D) nature of LiDAR Hokuyo UTM-30LX's point-cloud, LiDAR's vertical position must be such that the reading layer intersects the crop rows. The sensor and other devices should be placed carefully to ensure the 270° angular field-of-view of the LiDAR is not obstructed. These are the only two constraints in positioning the LiDAR. Other than that, the position of the LiDAR in front of the robot does not affect the algorithms presented here. For this research, the sensor is placed in the front part of TerraSentia: In the center of the robot's traverse axis for symmetry, 0.15 m to the front of the robot to avoid reading obstruction and 0.13 m above soil.

An embedded 64bit minicomputer (1.2GHz quad-core Raspberry Pi 3 Model B) handles measurement acquisition from sensors and generation of output signals to other embedded devices. In particular, Raspberry Pi 3 acquires distance readings from Hokuyo UTM-30LX through USB2.0 using SCIP2.0 protocol, runs the proposed perception subsystem and generates desired control signals (i.e. desired angular and linear velocities), sent to Kangaroo x2 Motion Controller as two Pulse-Width Modulation (PWM) signals.

4 LiDAR-based perception subsystem

This paper proposes a novel LiDAR based perception subsystem for a ground mobile robot, allowing its navigation in narrow and cluttered lanes of agricultural fields, e.g. corn (*Zea mays*) and sorghum (*Sorghum bicolor*), with high leaf coverage, frequent occlusion of sensor, and visible neighboring rows. Although LiDAR based and LiDAR fused with vision based agricultural navigation has been demonstrated before in orchards and to limited extent in row-crops (see Section 2), to our knowledge it is the first extensive series of successful autonomous navigation demonstration of small robots using a single layer LiDAR in multiple heterogeneous and highly cluttered row-crop environments. For practical reasons, we make the following assumptions about the conditions that the robot can encounter in the field. These assumptions are motivated by modern precision agriculture practices prevalent in South America and North America. Targeting row-crops, such assumptions are:

- Known lane width: Accepted practice in modern agriculture as this is pretty much guaranteed with GNSS guided tractors. Nonetheless, our algorithm has been demonstrated to be capable of accommodating mild variations in the row width in field testing;
- Parallel rows: Same as above, but note that we do not assume that the rows need to be straight, the presented method can accommodate a wide variety of curved rows, although accommodating sharp turns in rows is left for future work;
- Obstacle-free lanes: We assume that there is no rigid obstacle obstructing the motion of the robot, such as pieces of machinery, walls or workers. Although the information from the LiDAR may be used to detect these obstacles, this is left for future work.

Figure 5 depicts the main logic sequence of the robot's embedded code, composed of ***Initialize***, ***Update***, ***Estimate***, ***Validate*** and ***Control*** steps. The ***Initialize*** stage handles: 1. Import of configuration files with physical characteristics of the robot, threshold values and function limits; 2. Setting of row-crop characteristics and other initial conditions; 3. Initialization of the robot's embedded devices, such as LiDAR and motor driver, and respective communication protocols. Then, inside a loop, ***Update*** refreshes the sensor readings, ***Estimate*** produces lateral distance estimations, ***Validate*** evaluates the outputs, and ***Control*** generates and sends suitable control values for the powertrain module.

The following sections present details of the perception subsystem. First, the frame coordinates are specified in Section 4.1. Then, Section 4.2 explains the lateral distance estimation process and Section 4.3 proceeds explaining how the estimates are validated. Section 4.4 presents how the robot control uses such output to generate suitable signals for the motors. Finally, Section 4.5 comments on initial settings.

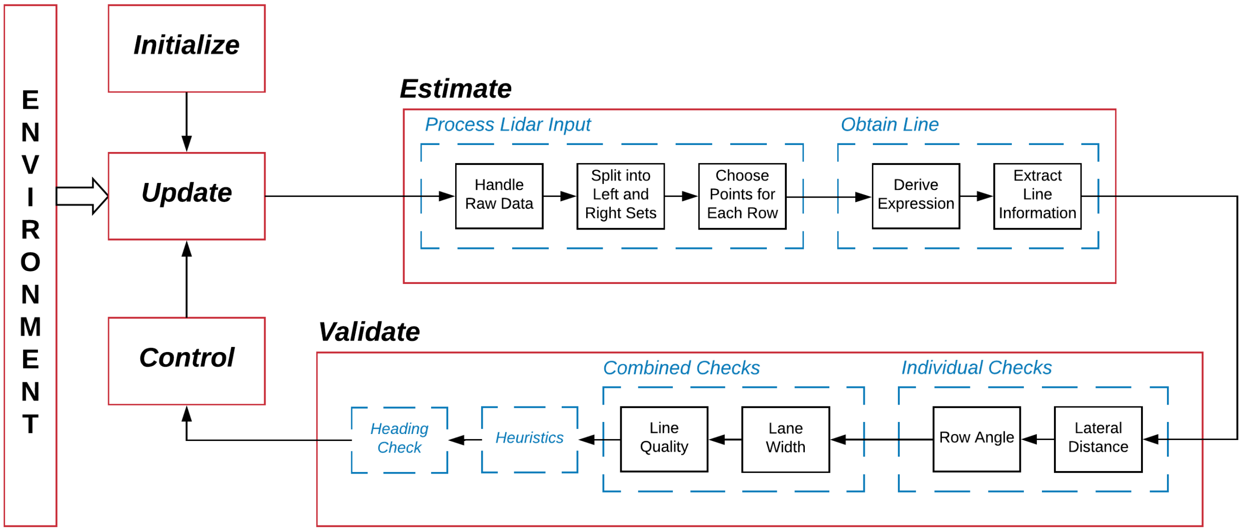


Figure 5: Diagram of embedded robot code. Configuration files are loaded, devices are started and initial values set in the ***Initialize*** step. Sensor readings are refreshed in ***Update***. The ***Estimate*** and the ***Validate*** stages compose the proposed perception subsystem. First, ***Estimate*** processes LiDAR input and obtain the lines corresponding to lateral rows. Then, ***Validate*** ensures correct estimates were obtained. Finally, using the perception outputs, ***Control*** uses a PID controller to generate steering commands.

4.1 Frame coordinates

Figure 6a represents the expected situation encountered by the robot: Between two rows of row-crop plants; a 2-D LiDAR sensor, marked with orange square, provides a point-cloud of distance readings of the surroundings. Using this point-cloud, the perception subsystem estimates the robot's lateral distances d (on both sides) and heading θ with regards to the adjacent rows of corn or sorghum crop, while ignoring outlying measurements from other rows visible to the robot and accounting for occasional sensor occlusion. The lateral distance d is defined as the orthogonal distance between the center of LiDAR and an adjacent row, while the heading θ refers to the angle between y-axis of robot and crop coordinate frames, respectively (x_R,y_R) and (x_{CR},y_{CR}) in Fig. 6a. Both frames have origin attached in the center of the LiDAR sensor, but while the first rotates with the robot, the latter has its y-axis (y_{CR}) always parallel to one of the rows.

4.2 Estimation of lateral distances

The ***Estimate*** stage receives LiDAR data and previous values to estimate lateral distances. First, it processes the LiDAR vector to obtain suitable left and right sets. Then, it obtains and characterizes the lines representing each side row.

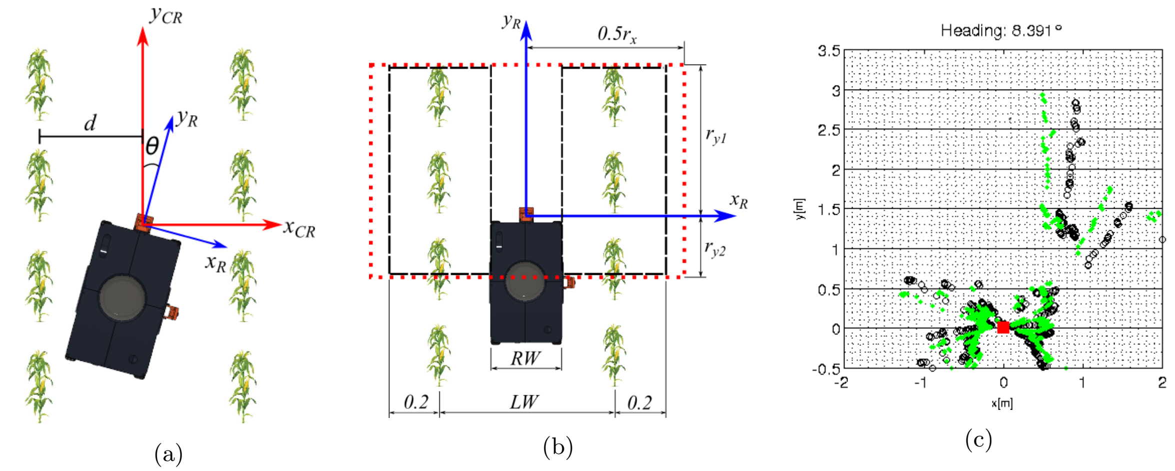


Figure 6: (a) Crop (red) and robot (blue) reference frames. Their origins are fixed at LiDAR, but while robot y-axis (y_R) is aligned with robot's longitudinal axis, crop y-axis (y_{CR}) is parallel to rows. The angle between these two y-axis is defined as heading θ and the lateral distance d is the orthogonal distance between the center of sensor and an adjacent row. (b) Limits for LiDAR readings and robot reference frame. In **Process LiDAR input**, readings outside red dotted rectangle (defined by r_x, r_{y1} and r_{y2}) are filtered out. we define r_x to read only the immediate lateral row. r_{y1} and r_{y2} weights the importance between the readings ahead with those behind the sensor. Later, only readings inside black dashed rectangles are considered as possible distance measurements for each lateral row. LW is the lane width and RW is the robot width (c) Black circles are original LiDAR readings and green dots are rotated ones using estimated robot's heading θ. Red square marks the sensor position.

4.2.1 Process LiDAR input

The first task is to handle the raw LiDAR data. The readings are converted from polar to Cartesian coordinates - robot coordinate frame (x_R, y_R) in Fig. 6a - using Eq. 1 where l_i is the i-th element of the LiDAR vector and σ_i is the respective angle of reading. The center of the sensor is considered the origin of such reference frame that has positive y-axis dividing angular range in two symmetrical halves and pointing in the direction of robot's forward movement, and x-axis following the right-hand rule. For convenience, as rows should follow the same direction as y-axis, σ_i with i $\in [0, 1080]$ covers the angular range between -45° and 225° with angular resolution of 0.25°. Additionally, θ_{k-1} of Eq. 1 is the last frame heading for rotation around origin. Besides preventing the elimination of readings that pertain to rows, this rotation intends to obtain at least one of the rows parallel to y-axis (see Fig. 6c).

$$\begin{cases} x_i = & l_i cos(\sigma_i + \theta_{k-1}) \\ y_i = & l_i sin(\sigma_i + \theta_{k-1}) \end{cases} \tag{1}$$

As the robot needs to locate itself, in local coordinates, inside a lane whose width does not exceed 1m, using all the measurements pointlessly increases the complexity and computational cost. Thus, a rectangle of sides r_x in x-axis and r_y in y-axis limits used readings and original vector is reduced to *FilteredPoints*. As it is expected that both sides provide equally useful information regarding crop rows, side r_x has a midpoint in $y = 0$. On the other hand, midpoint of side r_y weights the importance of readings ahead of the robot (r_{y1}) with regards to readings of the row that just passed (r_{y2}). This rectangle is pictured in Fig. 6b with red dotted lines.

Subsequently, *FilteredPoints* is split into two *SidePoints* sets (one for left and another for right row), leaving only the readings inside the black dashed rectangles shown in Fig. 6b. For the task of segmenting crop rows, the space between two rows and in front of the robot do not provide useful information. Thus, with the assumption that the heading θ sufficiently rotates LiDAR readings to obtain rows parallel to robot's longitudinal axis for the next L meters, the measurements in this region may be filtered out. Note that these measurements can still be used to detect obstacles in the path, but this is beyond the scope of the present work. Such inner limits are suitably chosen in the end of ***Validate*** stage based on the last valid values of lateral distances.

Algorithm 1 Pseudocode for choosing readings based on Histogram

```
ChooseBasedOnHistogram():
    Histogram Bins = fcnHistogram(SidePoints, Histogram Intervals);
    Histogram Index = indexMaxDouble(Histogram Bins);
    validateHistogramIndex(Histogram Index, Previous Histogram Index);
    ChosenPoints = chooseReadingsToBeFitted(Histogram index, Histogram Bins, SidePoints)
```

After splitting, there is a left *SidePoints* set as well as a right one, and now we want to choose which points will be used to fit the lines. To do that, a histogram-based algorithm is employed (**ChooseBasedOnHistogram** in Alg. 1). First, **fcnHistogram** sorts *SidePoints* x-values into bins with the bin edges specified by *Histogram Intervals*. The number of bins is user-defined and the size of the histogram bins is set according to Eq. 2. The *Histogram Intervals* is then constructed to be centered in the origin having half of the bins for the negative x-values and the other half for positive x-values.

$$BINS\_SIZE = \frac{\lceil 2 \cdot LW \rceil}{n\_bins} \tag{2}$$

The function **indexMaxDouble** searches for *Histogram Index*, the index of the bin with most elements in it. This value indicates the bin that will be used as reference to choose readings for line fitting. Then, **validateHistogramIndex** validates it: *Histogram Index* must be inside the *Histogram Intervals* vector size, excluding the extremities, due to the need of using adjacent bins to fit the line. Additionally, it must not represent a change from the bin used in the previous frame bigger than a threshold value *max_diff_bin*. If *Histogram Index* is not valid, it is replaced by its previous value.

Finally, the function **chooseReadingsToBeFitted** proceeds storing in *ChosenPoints* the readings inside the bin indicated by *Histogram Index*. As heading rotation generally allows only one of the rows to be parallel to y-axis, the other row possibly has a considerable inclination, being spread over more than one histogram bin. To avoid disposal of valuable information, *ChosenPoints* also contains the readings pertaining to the half bin size to each side of that marked by *Histogram Index*.

4.2.2 Obtain line

Now having the points that are most likely linked to the stalks, a first-order line given by $x = my + b$ is fitted to *ChosenPoints* using least squares linear regression. The slope m is given by Eq. 3 and the intercept b is given by Eq. 4. Moreover, if the calculated slope is a finite number, then a *FittedPoints* set is generated and it possesses the fitted x-values for *ChosenPoints* y-values applied to line equation $x = my + b$.

$$m = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2} \tag{3}$$

$$b = \frac{\sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i y_i}{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2} \tag{4}$$

After fitting lines, five characteristics are extracted from them: length, lateral distance, angle, standard deviation of its difference with regards to original x-values, and difference between estimated current and previous lateral distance. Length is simply the difference between minimum and maximum y-values of the respective line. Lateral distance $d[\mathcal{S}]$, where $\mathcal{S}$ is L for left or R for right side, is the orthogonal distance from the origin to the line and it is calculated with Eq. 5. Row angle $\alpha[\mathcal{S}]$ is given by Eq. 6 and refers to the angle between line and robot's longitudinal axis.

$$d[\mathcal{S}] = \left\| \frac{b}{m} \sin(\arctan(m)) \right\| \tag{5}$$

$$\alpha[\mathcal{S}] = \arctan(m) \tag{6}$$

4.3 Validity of Estimated Values

Some estimations do not reflect the reality because of frequent occlusions by hanging leaves, which causes sudden changes to the LiDAR readings and therefore to the estimated values. For such reason, the validity of certain variables is assessed through ***Estimate*** step and also in ***Validate*** stage. While the former checks the values individually, the latter checks them as a whole. After all validation process, there are two booleans *valid[S]* stating whether the side is valid or not, and two *LineGrade[S]* quantifying the quality of each side estimation.

In the ***Estimate*** step, there are two checks: 1. Estimated distance d is valid if $d \in (0.5RW, 1.1(LW - 0.5RW))$ where RW is robot width and LW is lane width; 2. Row angle α is valid if its difference to previous value is less than a threshold value *max_diff_alpha*.

The ***Validate*** step proceeds with three phases. First, it calculates current lane width lw as the sum of distances to right and to left. It is considered valid if $lw \in (1.1RW, 1.2LW)$. Otherwise, the side whose distance value changed most is considered invalid. Then, left and right lines are compared in four aspects: 1. Number np of points used to fit the line; 2. Length ly in y-axis; 3. Standard deviation sd of the difference between fitted and original y-values; and 4. Difference dd between current and previous value of orthogonal distance. A greater np indicates there are more points to rely on and fit a line. A greater length shows that a given side is visible for more time. A lower sd confirms that less outliers such as leaves were used to fit the line. And a lower dd is key to observe continuity in the estimations, as occlusions produce sudden changes that should be filtered out. These four conditions grants more points to the line possessing them. For example, Eq. 7 shows how these conditions are quantified to generate a *LineGrade* for left side. The numerator of the fractions changes to right side values to calculate *LineGrade[R]*. Additionally, *LineGrade* of the side that was used in the last frame receives a 0.1 bonus to avoid bouncing the valid status between left and right sides when they have similar grades. Finally, the third phase is the Heuristics, discussed in the next section.

$$LineGrade[L] = \frac{np[L]}{np[L] + np[R]} + \frac{ly[L]}{ly[L] + ly[R]} + \left(1 - \frac{sd[L]}{sd[L] + sd[R]}\right) + \left(1 - \frac{dd[L]}{dd[L] + dd[R]}\right) \tag{7}$$

4.3.1 Heuristics

Perception Outputs (heading, lateral distances and next frame row reading limits) require special attention to avoid sharp discontinuities and cumulative errors. Based on *valid[S]* and *LineGrade[S]*, the proposed

Heuristics decides how to update them and *Analytics Outputs* (previous values of d, θ, α, *Histogram Index* and *LineGrade*). Four possible cases are addressed:

1. Both sides have true *valid[$\mathcal{S}$]*;
2. Only left side is valid and *LineGrade[L]* $>$ *LineGrade[R]*;
3. Only right side is valid and *LineGrade[R]* $>$ *LineGrade[L]*;
4. None of above.

In the first case, row angle of the side with greater *LineGrade* is added to the heading. This happens because row angle is estimated after rotation when heading is performed. Therefore, the actual angle with robot's y-axis is $\theta + \alpha$. When dealing with real world crops, side rows are not parallel and it is even expected significant divergence or convergence for the next meter. For this reason, as the heading updates with only one of the angle estimations, there is a better chance that at least one of rows will be projected as parallel to robot's longitudinal axis in the first step of **Process LiDAR input** (Section 4.2.1).

In the second and third cases, heading is updated using the respective row angle. Opposed to first case, one of the lateral distance estimations is invalid and needs to be replaced. If the sum of distance estimations is within 20% difference of the expected lane width, then the invalid estimation is replaced by previous value minus the difference between estimations of the valid side. Otherwise, it is replaced by the difference between lane width and lateral distance of the valid side.

In these three cases, lateral distances of *Perception Outputs* are updated with the local variables, while previous values of distance d, heading θ, *Histogram Index*, *LineGrade* and row angle α are stored as *Analytics Outputs*, and row reading limits are updated to use the regions between $(-d[L] - 0.2, -d[L] + 0.1)$ and $(d[R] - 0.1, d[R] + 0.2)$. The regions are not centered in $d[\mathcal{S}]$ because, although undesired, some of the leaves falling inside the lane contribute to the distance estimation. This way, they pull the estimation inwards hence an additional space is given to outer region defined by $d[\mathcal{S}]$ as it is more likely to contain the stalks. Finally, no variable is updated in the fourth case.

4.3.2 Final check

The heading value is constantly being summed up with other quantities. For such reason, as a safety measurement, there are two cases that indicate loss of reliability on estimations. The first is when heading is greater than *zero_rotation_angle_lim*. And the other happens when the sum of the lateral distance estimations compared to lane width has a greater difference than *zero_rotation_diff_row_width*. If any of these two cases occur, all calculations of ***Estimate*** stage are repeated but the LiDAR readings are projected to Cartesian coordinates with a null heading. If the achieved *LineGrade* for the side used to update last heading value is greater than the one obtained in the first execution of ***Estimate***, then all variables will be updated and heading will be set to zero.

4.4 Control robot

The ***Control*** step implements a classic PID controller whose input is distance error e_d and output is the desired steering. The signal e_d is the difference between the desired and actual distance to center of the lane. This value is calculated by the difference between the right lateral distance and the left lateral distance.

4.5 Comments on initial settings

The ***Initialize*** stage starts the robot embedded devices, loads configuration files and sets *Perception Outputs* to initial values. The latter occurs in accordance to the assumptions considered in the beginning of the section. For instance, with the condition that the robot starts roughly in the middle of the row with known width LW and also parallel to rows, lateral distances initialize as half of the lane width and angular displacement between robot longitudinal axis and lane (θ in Fig. 6a) is zero. Furthermore, the initial row limits are set: Inner to half of the robot's width and outer to half of expected lane width plus 0.2 m as shown by black dashed lines in Fig. 6b. The desired cruise speed is set to a fixed value accordingly to user's need within the limits of the interval $[0.1; 0.7]$ m/s. For the maximum speed and also due to robot's size, the threshold values max_diff_alpha and max_diff_bin are set to 5° and number of bins that represent ∼0.2 m, respectively.

5 Results and Discussion

To test the proposed perception system, multiple runs in corn and sorghum fields took place in the state of Illinois, logging more than 6 km of autonomous navigation. As a common characteristic, all fields had rows on both sides and they were at least 3 m long. The lane width varied from 0.7 m to 0.9 m, also depending on the orientation that the plants grew. Furthermore, it was tested in a specific soybean field, whose lane width was around 0.8 m and had low presence of leaves/branches. Table 1 summarizes the performance of the autonomous runs. It also provides a glimpse of the robot's surroundings. Details for each crop type are discussed in the next subsections. First, a discussion about ground truth is provided.

Table 1: Summary of field experiments. The robot ran autonomously in two corn and one sorghum crops while cameras installed on the robot recorded the surroundings for other research. Each field had a different purpose for the visual data, but a common requirement was that collected info should be from a complete row. Therefore, a successful autonomous run is defined as one that did not stop from beginning to the end of the row, regardless of length or time.

Crop Type	Total Number of Runs	Successful Runs	Success Rate	Total Distance Traveled [m]
Corn 1	30	27	90%	1115
Corn 2	35	33	94.29%	1022
Sorghum	246	220	89.43%	660

In addition to the 311 runs shown in Table 1, another 70 autonomous attempts (3450 m) were conducted by external people in diverse corn fields in Champaign, IL using several replicates of TerraSentia. Eight (11.42%) of these last runs did not complete the full travel. For the subsequent sections analysis, only the experiments shown in Table 1 are used.

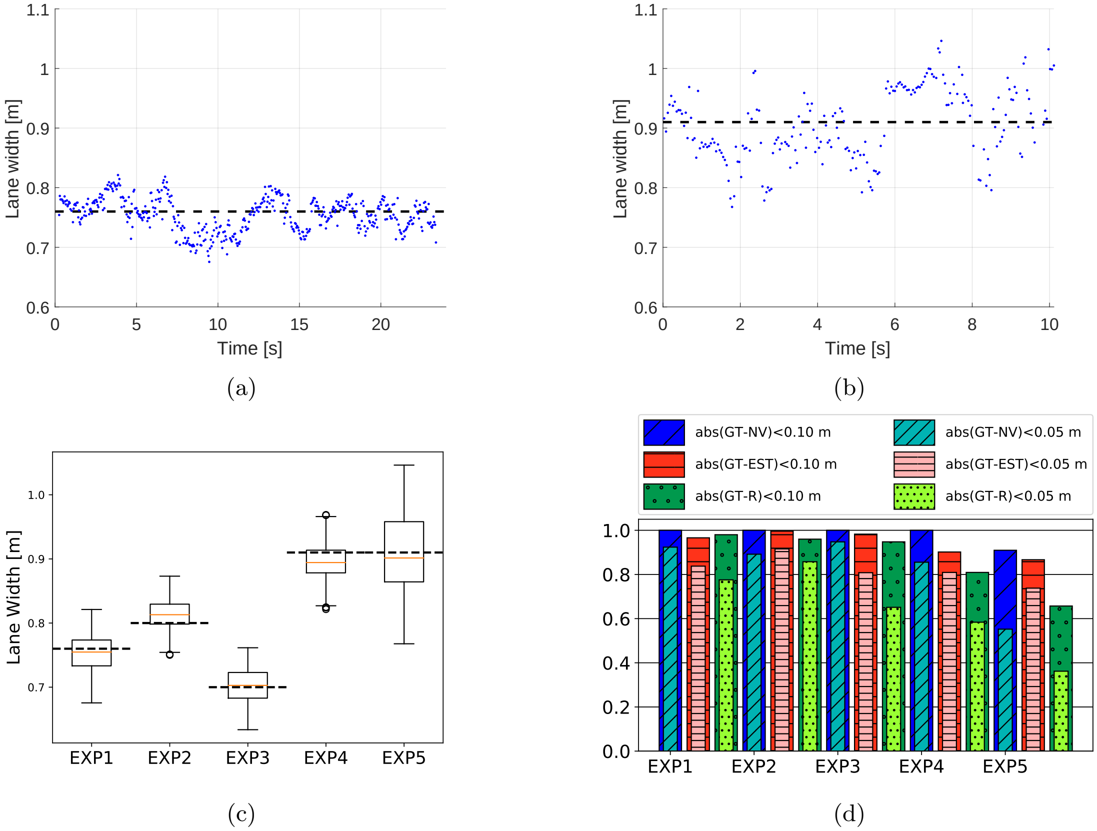
5.1 Ground Truth

The main contribution of this study is the perception subsystem. Therefore, an appropriate ground truth provides insight about that subsystem performance with minor interference from other sources. RTK-GNSS devices typically provides ground truth to evaluate outdoor performance of mobile robots. However, their measurements under canopy are not reliable because of multi-path and signal attenuation. Although a possible solution would be to put the receiver on top of a high mast, it is not practical due to two main reasons: 1. The robot height is only 0.3 m but the mast would need to be at least 2 m above ground for corn and 3 m for sorghum. Profiled aluminum bars could be used to achieve a light and rigid mast, but because of robot's relative small width (0.31 m), any roll inclination due to soil unevenness is heavily amplified at the top of the most. 2. The receiver weights 0.5 kg, which is significant compared to robot's 6.6 kg. The swinging at the top generates a tipping torque that affects the robot's performance.

Another option is manual ground truth from LiDAR data by choosing the best line for each lateral row. The process requires laborious frame by frame analysis to determine four points per frame (two for each line). Subsequently, Eq. 5 provides the lateral distances. From their difference, it is possible to know the distance between the LiDAR (subsequently the robot) and lane centers (DC). Such measure is a great indicator of the control performance, i.e. the level of capability to keep the robot in the desired path, usually middle of the lane. Nonetheless, without an external measurement for comparison, it hardly can be used to evaluate the perception system because inconsistencies originate from multiple sources other than the perception itself. An intrinsic one is the high dependency on visible stalks, which do not form a straight line due to different growth directions and widths. Another is the ground unevenness with unexpected slips, bumps, weeds and/or fallen stalks. Both lead to sudden changes to lateral distances, which generate a great difference between current and previous DC. To alleviate such issues, a viable measure is the lane width, sum of lateral distances. The lane width (LW) still reflects the intrinsic problem, but an appropriate perception subsystem should obtain the correct lane width regardless of the robot's position, which has major influence from ground unevenness and the lane-keeping controller. Therefore, LW is selected to assess perception performance for row-following purpose.

Ground truth is obtained for five experiments: Corn 1 (Fig. 7a), Corn 2, two Sorghum and one highly cluttered Sorghum (Fig. 7b). The nominal LW are 0.76 m, 0.8 m, 0.7 m, 0.91 m and 0.91 m, respectively. The last experiment presented some degree of sensor occlusion due to leaves in 80% of the experiment. As it can be seen in Figs. 7a, 7b and 7c, the measured lane width gravitated around nominal value, marked with a dashed line. Even as output of manual measurement, the LW did not present continuity. In the worst case, there was a 0.157 m difference between consecutive frames. That happened because after several frames with leaves completely occluding the left row, a sudden clearance showed that the left row was farther than previously interpreted. Figure 7d shows the agreement between ground truth, nominal value and estimated LW with proposed perception subsystem. Most of the ground truth are under 0.05 m difference from nominal value (cyan bars). The exception is fifth set with only 55% below threshold. This arose from row occlusion by leaves. In such case, the leaves themselves formed a wall that were used for reference as no other was available. Most of the estimated values were within 0.1 m error from ground truth (scarlet bars). Bigger errors usually happen after LiDAR occlusion stops or in the initial/final part of the row, where there is fewer information to gather. The lowest percentage of estimated values under 0.05 m difference is 73% (sundown bars), which occurred for the experiment with constant occlusion by leaves.

Additionally, RANdom SAmple Consensus (Fischler and Bolles, 1981), a widely used iterative method to estimate parameters of a mathematical model in the presence of outliers, was evaluated with regards to the extraction of the lateral rows. The algorithm imported from Point Cloud Library (PCL) replaces the *Choose Points for Each Row*, *Derive Expression* and *Extract Line Information* steps in ***Estimate*** (Fig. 5), thus its inputs are left and right *SidePoints* and its outputs are the line parameters. Analogously, Eq. 5 calculates the lateral distances, which are further used to estimate the lane width. Figure 7d also displays the percentage of scans regarding absolute difference between GT and RANdom SAmple Consensus (RANSAC). As it can be seen, RANSAC performs relatively well in Corn, but the same cannot be said for Sorghum, which is generally more cluttered. Therefore, we do not consider standard RANSAC for ground truth generation or as part of the proposed perception subsystem for this study.



(a)

(b)

(c)

(d)

Figure 7: Ground truth was manually obtained for five experiments (Corn 1, Corn 2, two Sorghum, and one highly cluttered Sorghum): A line for left row and another for right row were picked for each LiDAR scan. The lines provide lateral distances, which can be used to estimate the lane width (LW) and the distance between the LiDAR and lane centers (DC). The ground truth LW for each scan can be seen in (a) for Corn 1 and (b) for cluttered Sorghum. Plot (c) displays the ground truth LW distribution for all five experiments. The dashed lines are the nominal lane widths. Plot (d) presents the percentage of scans where the absolute difference between ground truth (LW) and nominal value (NV)/proposed perception subsystem (EST)/RANSAC (R) is less than 0.1 m (darker bars) or less than 0.05 m (lighter bars).

5.2 Perception Subsystem Configuration and Sample Execution

This subsection presents the operation of the Perception Subsystem for two situations: 1. Few seconds after its start; 2. Sensor occlusion by leaves. TerraSentia robot was employed with a LiDAR Hokuyo UTM-30LX placed in the center of the front part of the robot, having its plane of view parallel to ground. As the field of view was unobstructed, all angular range is considered, but bounding rectangle was set with $r_x = 2$ m and $r_y = r_{y1} + r_{y2}$ where $r_{y1} = 2.5$ m and $r_{y2} = 1.5$ m. TerraSentia's width is 0.36 m. Although the lane width inside the tested crop varies, the nominal values were set to 0.76 m and 0.91 m, respectively. Desired distance from center that the robot should keep was zero (robot follows centered in the lane). Cruise speed was fixed and it had an average value of 0.24 m/s. For the histogram analysis, there were 20 bins whose 0.1 m width was given by Eq. 2. Finally, *zero_rotation_angle_lim* and *zero_rotation_diff_row_width* were respectively set to $\pi/18$ rad and 0.2 m.

5.2.1 Example 1: Few seconds after Perception Subsystem start

The ***Estimate*** step starts by processing LiDAR input. Figure 8a shows raw LiDAR readings (black circles) of a single scan in the cornfield. Even though the sensor range is up to 30 m, useful readings are already limited due to cluttering of leaves, weeds or corn/sorghum stalks themselves. Nonetheless, the 4 m x 2 m bounding box used to limit the Cartesian projections still provokes a significant reduction of elements. The projection uses estimated robot heading $\theta_{k-1} = -0.1062$ rad to suitably rotate current readings. As shown in Fig. 8b, heading rotation mitigates the original inclination to the left. The result of such processes is *FilteredPoints* and it is marked with red crosses in all Fig. 8 plots. To split into left and right rows, the method uses the last valid lateral distances (red full lines) to determine boundaries for the readings (red dashed lines). Then, such dashed lines delimit two sets in Fig. 8c: black diamonds for left *SidePoints* and blue squares for right *SidePoints*.

Not all elements are used to fit the line. In the *Choose points for each row* stage, histogram is applied to each *SidePoints* x-values to determine where the readings are concentrated. Fig. 9a shows current frame histograms with blue full bars and last frame ones with dashed red bars. In both cases, there are two peaks: one centered at $x = -0.35$ m for the left side and another centered at $x = 0.35$ m for the right side. Therefore, there is no change between *Histogram Index* values and thus, current ones are valid and they indicate the bins that contain readings that will be part of the line fitting. In Fig. 9b, blue full lines delimit the bin given by *Histogram Index* and whose number of elements is stored in np_k. Black dashed lines show the additional half bin size also considered in *ChosenPoints*, marked with red dots. Subsequently, the process

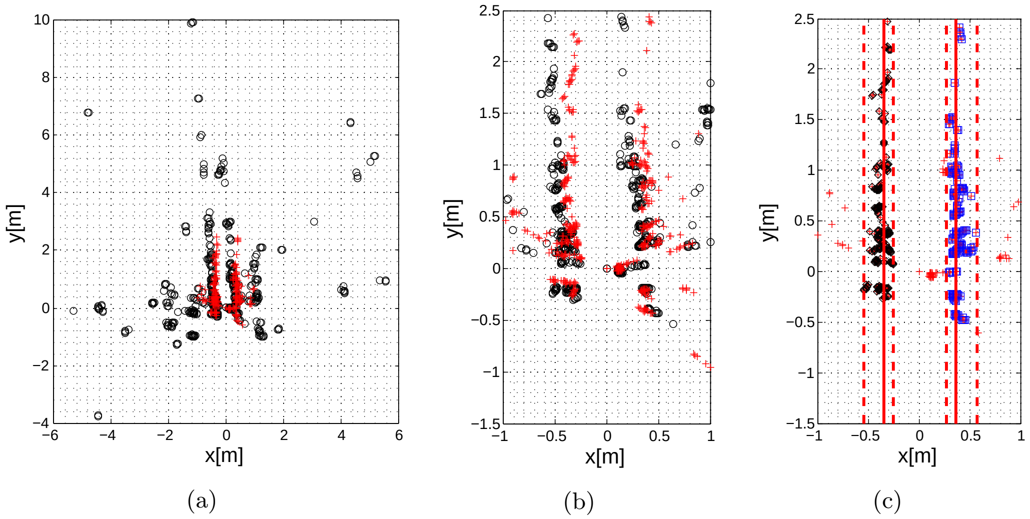


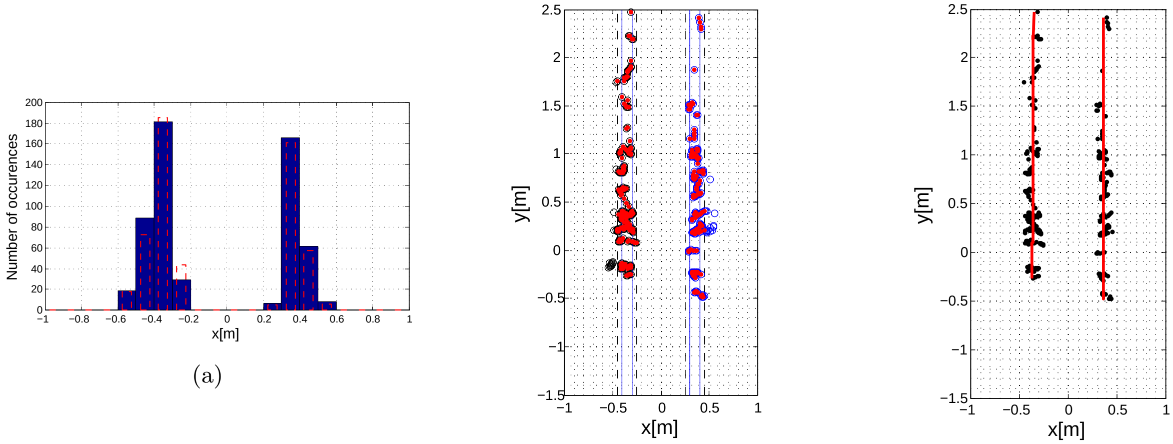
Figure 8: Perception subsystem steps for Example 1. (a) Raw data is projected to Cartesian coordinates (black circles). Additionally, rotation using last heading estimate is performed on elements inside a 4 m x 2 m to obtain *FilteredPoints* rows more closely parallel to y-axis (red crosses). (b) A close-up of *FilteredPoints*. (c) *FilteredPoints* is split into left (black diamonds) and right (blue squares) *SidePoints* sets.

calculates the slope and the intercept that best fit each *ChosenPoints* set. The obtained lines are represented by the red full lines shown in Fig. 9c. Using Eq. 5 and Eq. 6, lateral distances are $d[L] = 0.368$ m and $d[R] = 0.366$ m, and row angles are $\alpha[L] = 0.0056$ rad and $\alpha[R] = -0.0007$ rad. As shown in Fig. 9c, minimum and maximum y-values for the left line are respectively -0.27 m and 2.47 m, resulting in a length $ly[L] = 2.47 - (-0.27) \Rightarrow ly[L] = 2.74$ m. Analogously, $ly[R] = 2.41 - (-0.48) \Rightarrow ly[R] = 2.89$ m.

Table 2: Values obtained from perception subsystem for last and current LiDAR scan (Example 1)

	Previous			Current					
	np_{k-1}	d_{k-1} [m]	α_{k-1} [rad]	np_k	d_k [m]	α_k [rad]	ly_k [m]	sd_k [m]	$LineGrade_k$
LEFT	180	0.352	-0.0026	182	0.368	0.0056	2.74	0.048	1.4597
RIGHT	164	0.367	0.0013	169	0.036	-0.0007	2.89	0.366	2.6403

The method proceeds with the validation step. First, the individual checks: 1. Lateral distances are valid as $d[\mathcal{S}] \in (0.5RW, 1.1(LW - 0.5RW)) \Rightarrow d[\mathcal{S}] \in (0.18, 0.64)$; 2. Row angles are also valid because their difference with respect to previous values are less than $max\_diff\_alpha = 5° = 0.087$ rad. Then, the combined ones: 1. Estimated lane width is $lw = d[L] + d[R] \Rightarrow lw = 0.734 \Rightarrow lw = 0.96LW$, thus valid; 2. With Eq. 7 and values from Table 2, $LineGrade[L] = 1.4597$ and $LineGrade[R] = 2.5403$. As right values were used in the last frame to update heading, $LineGrade[R]$ receives 0.1 as bonus. Both sides are valid, thus *Heuristics* determines that heading θ_k is updated with row angle α_k of the side with greater *LineGrade*. Thus, $\theta_k = \theta_{k-1} + \alpha_k[R] \Rightarrow \theta_k = -0.1062 + (-0.0007) \Rightarrow \theta_k = -0.1069$ rad and the other *Perception Outputs* are replaced with the calculated values. A final check is done: heading $\theta = -0.1069 < \pi/18$ rad and



(a)

(b)

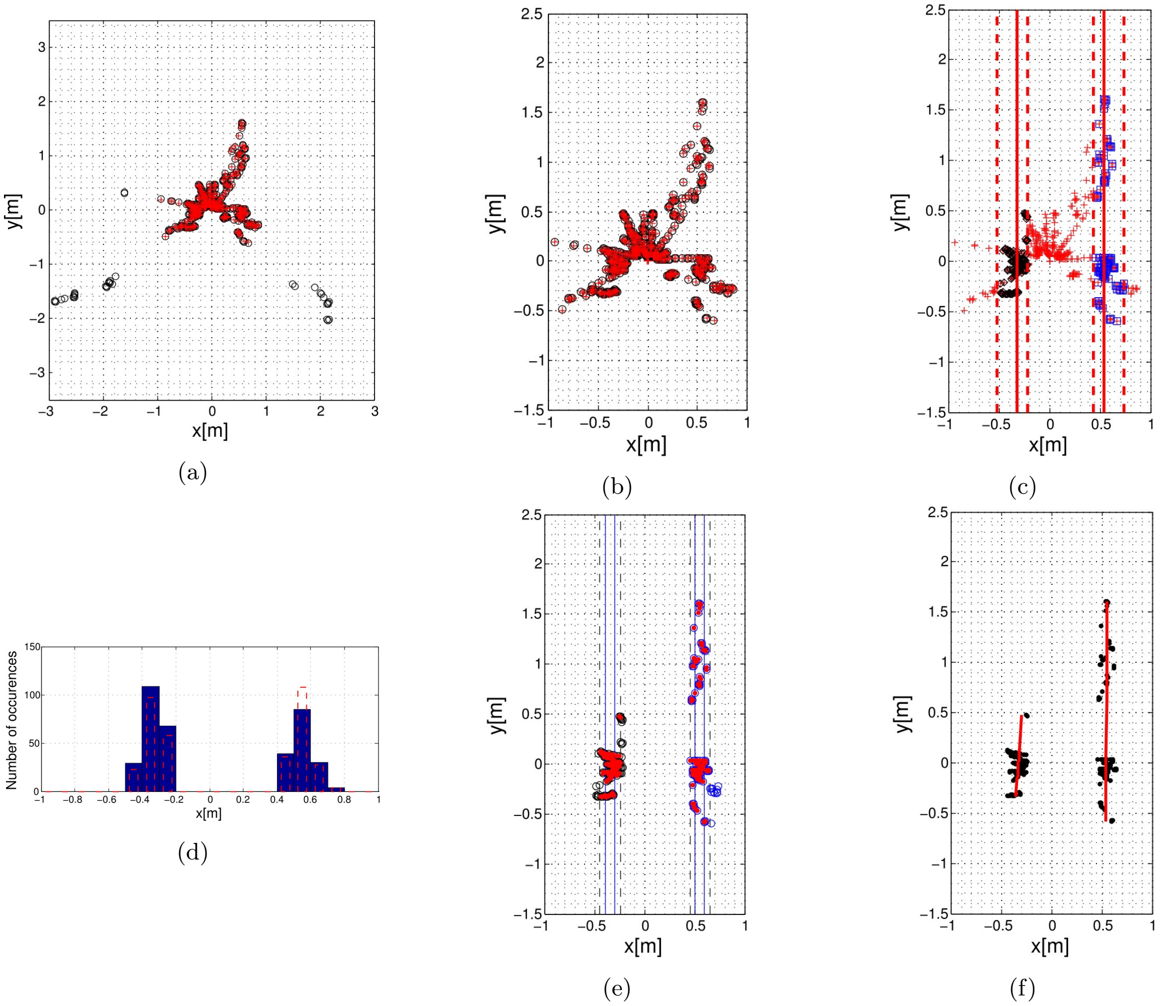
(c)

Figure 9: Subsequent perception subsystem steps. (a) Histogram performed on *SidePoints* to discover *Histogram Index*, which is correspondent to an approximate lateral distance. Full rectangles refer to current scan while dashed rectangles show previous one; (b) The readings in the bin marked by *Histogram Index* and in its half-bins adjacent ones are chosen to form the lines; (c) Red full lines represent the obtained line for each row.

$|LW - lw| = |0.76 - 0.734| \Rightarrow |LW - lw| = 0.026 < 0.2$ m, then ***Estimate*** step does not need to be repeated with null heading.

5.2.2 Example 2: Sensor occlusion by leaves

In this example, the effect of leaves covering the sensor can be seen from the start, as shown in Fig. 10a. While the maximum distance reading is 3.3 m, there are some that reach 10 m in Fig. 8a. Indeed, 94.17% of the readings are below 2 m thus almost no filtering is performed by the 4 m x 2 m bounding box. As the rows are already parallel to robot's y-axis, the previous estimated robot heading $\theta_{k-1} = -0.1062$ rad does not bring great changes to *FilteredPoints* (red crosses in all Fig. 10 plots). It proceeds to separation into two *SidePoints* showed in Fig. 10c by black diamonds for left and blue squares for right. Figure 10d shows that readings are concentrated at $x = -0.35$ m for the left side and at $x = 0.55$ m for the right side. They are valid as the previous histogram has the same *Histogram Index*. Subsequently, Fig. 10e indicates the boundaries for getting the *ChosenPoints*, marked with red dots. Finally, Fig. 10f presents the obtained lines. Using Eq. 5 and Eq. 6, lateral distances are $d[L] = 0.333$ m and $d[R] = 0.54$ m, and row angles are $\alpha[L] = 0.0072$ rad and $\alpha[R] = 0.006$ rad. Minimum and maximum y-values for the left line are respectively -0.31 m and 0.48 m, resulting in a length $ly[L] = 0.48 - (-0.33) \Rightarrow ly[L] = 0.81$ m. Analogously, $ly[R] = 1.61 - (-0.58) \Rightarrow ly[R] = 2.19$ m.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 10: Perception subsystem steps for Example 2. (a) Raw data is projected to Cartesian coordinates (black circles). Note that rotation by previous estimated heading does not bring perceptible change to *FilteredPoints* as the rows are already parallel to y-axis. (b) A close-up of *FilteredPoints*. (c) *FilteredPoints* is split into left (black diamonds) and right (blue squares) *SidePoints* sets. (d) Histogram performed on *SidePoints* to discover *Histogram Index*, which is correspondent to an approximate lateral distance. (e) The readings in the bin marked by *Histogram Index* and in its half-bins adjacent ones are chosen to form the lines. (f) Red full lines represent the obtained line for each row.

Table 3: Values obtained from perception subsystem for last and current LiDAR scan (Example 2)

	Previous			Current					
	np_{k-1}	d_{k-1} [m]	α_{k-1} [rad]	np_k	d_k [m]	α_k [rad]	ly_k [m]	sd_k [m]	$LineGrade_k$
LEFT	98	0.324	0.039	109	0.333	0.070	0.81	0.047	1.5783
RIGHT	108	0.537	-0.027	85	0.539	0.006	2.19	0.046	2.5217

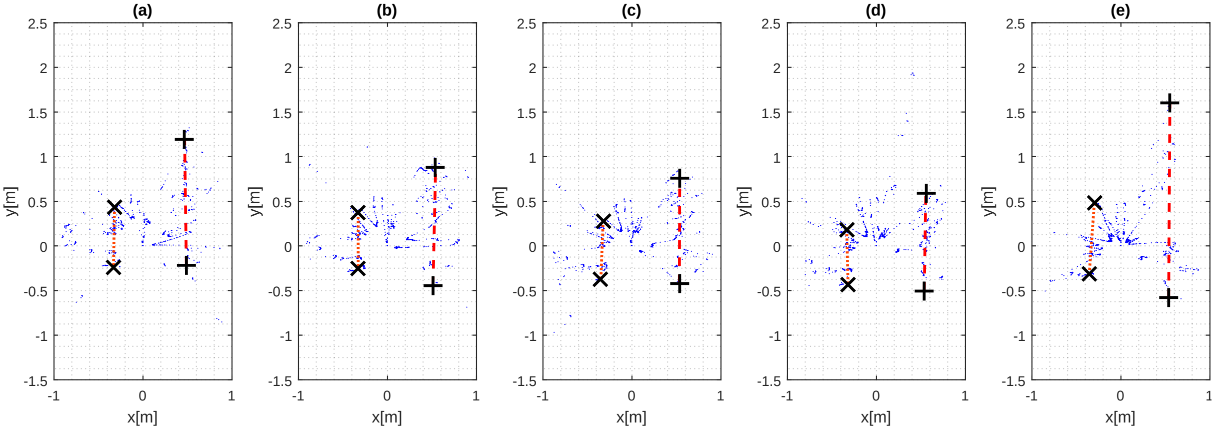


Figure 11: Five frames that happened in a period of 746 milliseconds, considering only one every four frames. They were obtained in a Sorghum run where the sensor suffered occlusion for more than 80% of the experiment. The green dotted and red dashed lines depict the estimated left and right row, respectively. As it can be seen, although all of the frames occurred below 1 second, there was significant change between them. Nonetheless, due to correct identification of rows in (a), the following estimates did not diverge even though sensor occlusion increased. Additionally, there are no visible stalk in (e), situation where the perception subsystem utilizes a wall formed by leaves to generate the left lateral estimate.

The method proceeds with the individual checks from ***Validate***: 1. Lateral distances are valid as $d[\mathcal{S}] \in (0.5RW, 1.1(LW - 0.5RW)) \Rightarrow d[\mathcal{S}] \in (0.18, 0.80)$; 2. Row angles are also valid because their difference with respect to previous values are less than $max\_diff\_alpha = 5° = 0.087$ rad. Then, the combined ones: 1. Estimated lane width is $lw = d[L] + d[R] \Rightarrow lw = 0.87 \Rightarrow lw = 0.96LW$, thus valid; 2. With Eq. 7 and values from Table 3, $LineGrade[L] = 1.5783$ and $LineGrade[R] = 2.5217$. As right values were used in the last frame to update heading, $LineGrade[R]$ receives 0.1 as bonus. Both sides are valid, thus *Heuristics* determines that heading θ_k is updated with row angle α_k of the side with greater *LineGrade*. Thus, $\theta_k = \theta_{k-1} + \alpha_k[R] \Rightarrow \theta_k = 0.01 + (0.007) \Rightarrow \theta_k = 0.017$ rad and the other *Perception Outputs* are replaced with the calculated values. A final check is done: heading $\theta = 0.017 < \pi/18$ rad and $|LW - lw| = |0.91 - 0.87| < 0.2$ m, then ***Estimate*** step does not need to be repeated with null heading. As it could be seen, the steps are the same for more cluttered situations. The reader should note that the success to estimate lateral distance rises from the use of previous estimates to suitably discard irrelevant measurements.

Additionally, Fig. 11 presents five frames from a period of 746 milliseconds, considering only one every four frames. As it can be seen, there is significant change between scans and the LiDAR at origin is constantly being hit by leaves, which block expressive sensor's field of view. The green and red lines depict for each case the estimated left and right row, respectively. In Fig. 11a the lateral rows can be clearly identified. Because of this estimate, subsequent ones do not diverge due to: readings scattered because of leaves (Fig. 11b,c), significant amount of readings concentrated on left leaves (Fig. 11d) and leaves blocking most of the stalks readings in front of the sensor (Fig. 11). Using previous estimates, there is already an approximate distance where the row should be. Therefore, the leaves that are hanging towards the lane center can be filtered out. As it can be seen in Fig. 11e, there are no visible stalks in the left row, situation which forces the algorithm to use the leaves themselves to form the row. Although it hinders the precise positioning of the robot inside the lane, a wall formed by leaves is a conservative estimation, which guarantees no-collision with plants.

5.3 Field experiments

The field experiments described in Table 1 happened in three locations and each one has a different plantation, plot size or lane width. Some of the runs finished earlier than nominal length due to lane blockage. The first cornfield, further referred as Corn 1, was located in Champaign, IL. It had lanes with 0.76 m nominal width and except for some lodged areas, these lanes were continuous for 50 m. The second cornfield - Corn 2 - was located in Ivesdale, IL. Although it had similar nominal lane width to Corn 1, it was divided into 3.3 m plots. Another set of experiments was conducted in Sorghum fields in Champaign, IL. The fields consisted of several varieties of sorghum distributed in 3 m x 3 m plots. The diversity in height, stem-width, and inter-plant spacing between individual plants provided a variety of different scenarios for testing. The lane had width varying between 0.76 m and 0.91 m, and it was divided into 25 plots of 3 m each. In this particular field, additional data was retrieved from right side facing camera for other research. To improve camera's field of view, desired distance from center is set to -0.2 m. Each run has a nominal length of 3 m (size of the plot).

Figure 12 shows one of the Corn 2 trials reconstructed with LiDAR readings. The green dots are the plants (stalks and leaves), the red dots mark the estimated lateral distances, the black dots display the estimated robot position while traversing the crop and the blue full lines delimits a [-0.1,0.1] m region around lane center. Between 7 m and 13 m, the presence of leaves hanging towards lane center can be easily spotted, which did not lead to visible changes in the perception outputs. But the combined effect of field of view blocked by leaves and the shortage of information due to gap between 3.3 m plots around 7 m led to a spike in the estimated position (0.18 m to the left side).

In the sorghum field, the trials were much shorter but perception subsystem presented similar behavior.

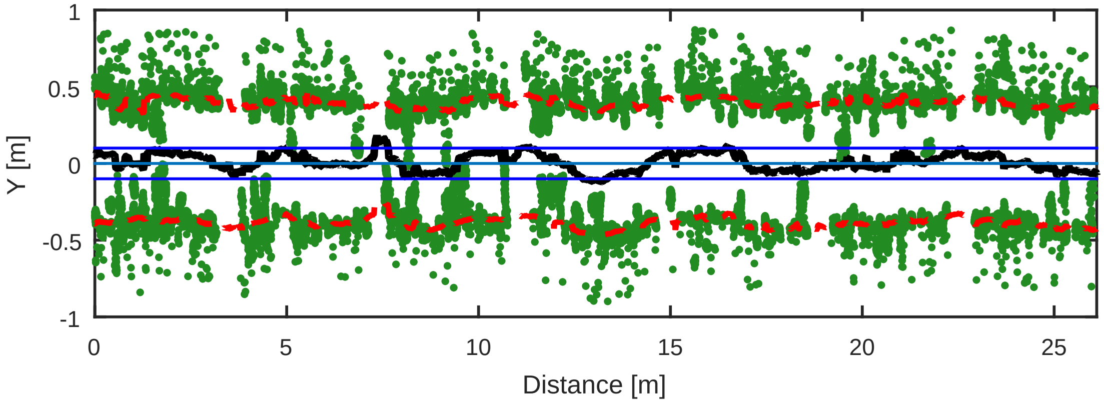


Figure 12: A Corn 2 run reconstructed with LiDAR data. Green points for crop rows, red dashed lines for estimated lateral distance and black full line for estimated robot position. It can be seen that most of the estimations are bounded inside a [-0.1, 0.1] m around lane center.

Figure 13 provides five video frames showing the robot going through a plot. It is important to note that unlike the example presented in Section 5.2, TerraSentia keeps an offset of 0.2 m to from the center. This improves the side-facing camera field of view. Moreover, Fig. 14 shows the difference between local positioning using LiDAR perception subsystem and RTK-GNSS. While the first kept bounded inside a 0.2 m zone around the reference value, the latter shows some values that even go beyond the lane boundaries (marked as green dashed lines).

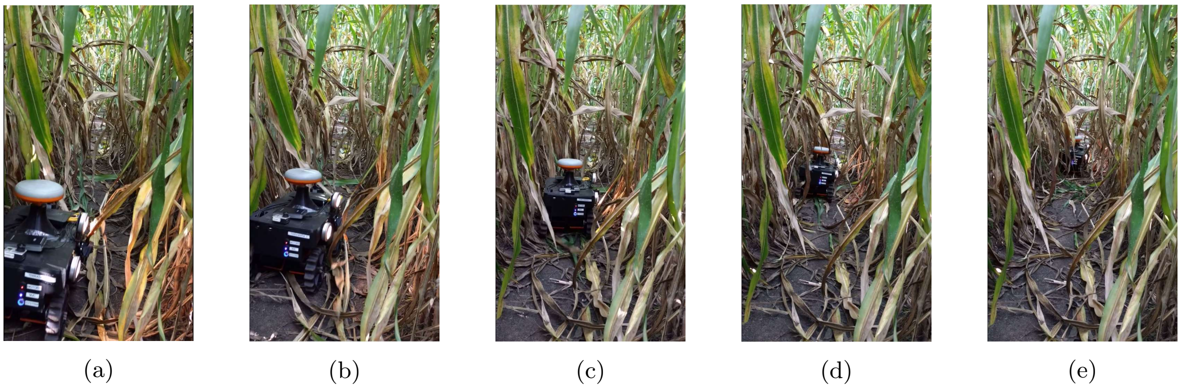
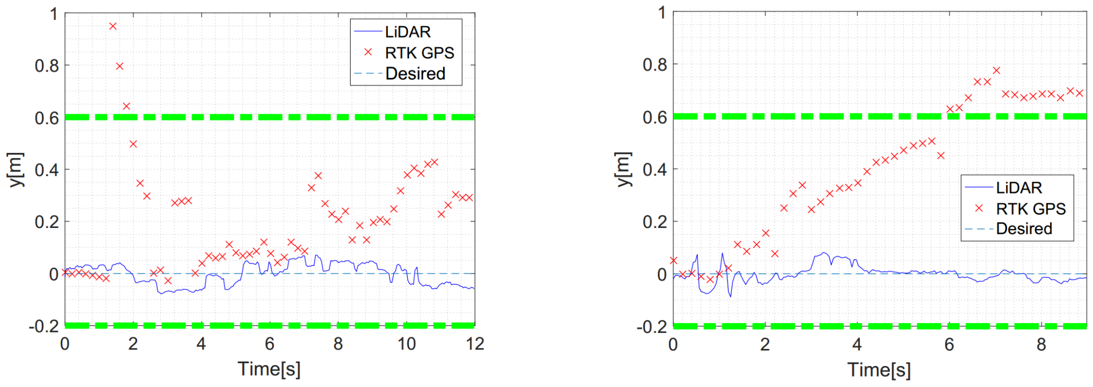

(a) (b) (c) (d) (e)

Figure 13: Video frames of an autonomous run in sorghum field. To improve right-side facing camera field of view, the reference to center was set with a 0.2 m offset.



(a)

(b)

Figure 14: Comparison between LiDAR output for distance from reference and absolute position given by RTK-GNSS. Regarding RTK-GNSS, occasional drops may be seen in accuracy (a) and drift (b). Thick green dashed dot lines show the position of the sorghum rows and blue dashed line mark the desired path for the robot.

5.3.1 Failures

From discussed in Section 5.1, ground truth for every run is not achievable because it would require either a manual analysis of each LiDAR frame or positioning with RTK-GNSS, which does not provide reliable measurements under canopy. Hence, the criterion of success is defined as the capability of going from the beginning to the end of the lane without hitting the rows, as long as both rows are continuous. This criterion also meets the robot's main goal: visual data collection for complete rows, regardless of length or time.

In retrospective of the carried experiments, the failures may be divided in five categories:

- Mechanical: damage to robot's powertrain that requires immediate maintenance e.g. broken motors and large quantity of leaves wrapped around motor shaft;
- Control: robot presented response different than desired e.g. robot turning left when commanded to turn right;
- Perception: the perception subsystem is unable to or incorrectly estimates distance to rows. This happens due unexpected clearances in the one of the rows, and highly cluttered lanes that also compromises robot's movement;
- Start: when the reference distance to lane center is set to a value other than zero, the manual positioning of the robot in the starting position may lead to a wrong first estimation given the difficulty of placing the robot at a certain distance;

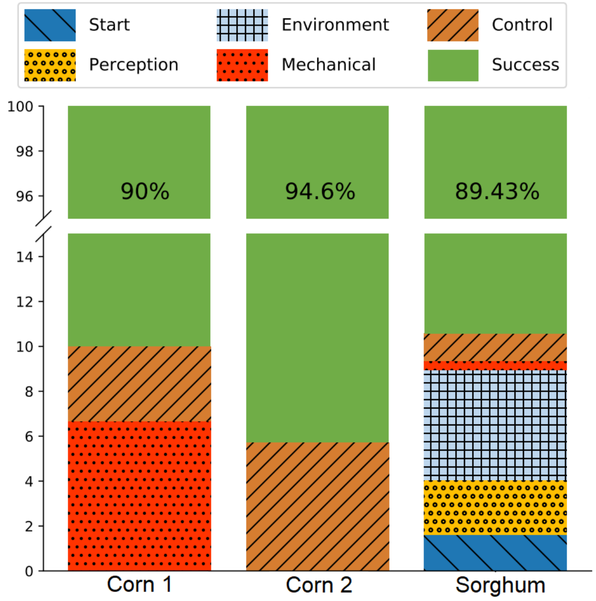


Figure 15: The 31 failures happened due to: 1. Error when starting the robot. 2. Error in the perception subsystem. 3. Environment obstacles that stopped robot's movement. 4. Mechanical problems in the robot. 5. Error in the signal generated by ***Control***.

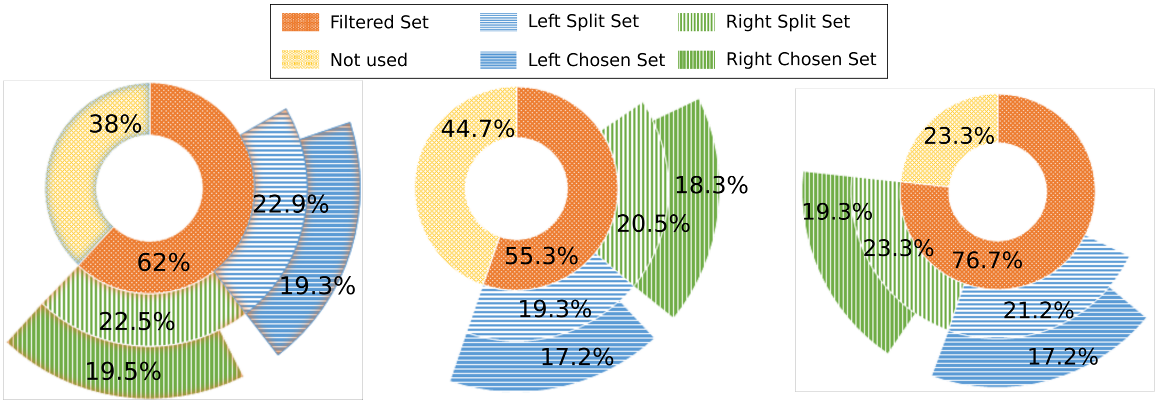
- Environment: lodging, fallen plants or presence of grown weeds may configure as obstacles in the middle of the lane, which are not considered in this work. Therefore, the robot gets stuck in the presence of such elements.

Figure 15 summarizes the success rate and the failure conditions for each field. In Corn 1, thirty runs (one ten meter, five twenty meters and twenty-four runs of 48 m) were conducted, achieving 90% success rate (27 successful runs) and covering 1115 m. The three failed due to mechanical (2 runs) and control (1 run) issues. In Corn 2, seventeen runs with length varying between 2 m and 15 m, two twenty meters, twelve forty meters, one seventy-five meter and three ninety-five meters compose the 35 runs resulting in 1022 m. The two failures are related to control issues. Finally, in sorghum, two hundred forty six runs were conducted, achieving 89.43% (220 successful ones) and covering 660 m. The other twenty six failed due to failure in start (4), highly cluttered lane (6), other environment interference that restricted the robot's movement such as fallen stalks (6), mechanical issues (1), failure in control (3) and failure in perception (6).

5.3.2 Set sizes

The raw LiDAR data has 1081 readings. After each step of ***Estimate***, the number of elements decreases, eliminating unnecessary information. Figure 16 shows the average size of *FilteredPoints*, *SidePoints* and

ChosenPoints sets for each location. For Corn 1, the original 1081 measurements are reduced to an average set of 670 points, a significant reduction of 38%. Then, such set was further filtered and split into two: left with 248 points and right with 243 points. Based on histogram applied on x-values to each sets, the final *ChosenPoints* sets have an average size of 211 (left) and 209 (right). For the less cluttered Corn 2 field, the initial reduction is even greater (45%) and in the end, only 35% of the measured points were used. On the other hand, in the high cluttered Sorghum field, there is a modest initial reduction of 23.3%. But, later stages (Split and Choose) had greater action, reducing final chosen sets to 36.5% of the measured points.



(a)

(b)

(c)

Figure 16: Average set sizes for ***Estimate*** stage for (a) Corn 1; (b) Corn 2; (c) Sorghum. The initial 1081 elements for a single scan reduces to *FilteredPoints* (orange) and yellow represents the readings out of the bounding box in *Handle Raw Data* stage. *FilteredPoints* is separated into two *SidePoints* groups: blue for left and yellow for right one. Note that a third group contains the *FilteredPoints* that does not belong to any *SidePoints*. A relatively large third group indicates great presence of leaves in the region in front of the LiDAR, as it is expected in the Sorghum (c). Finally, *ChosenPoints*, the readings that are effectively used to obtain the line expressions, are represented by green vertical lines for left and blue horizontal lines for right. As it can be seen, an average of 38.8% (19.5% from left and 19.8% from right *ChosenPoints*) of the readings are used in Corn 1, 35.5% in Corn 2 and 36.5% in Sorghum.

5.3.3 Calculation Time

The ability to supply relevant information in sufficient rate is an important characteristic for any system that interacts with the surroundings. In our case, the perception subsystem provides lateral distance estimations that are used to keep the robot in the middle of the lane and a delay in generating such estimates could lead to damages to the crop. Therefore, investigation of time required for calculations validates the use of the system for real-world situations and it also exposes time-consuming steps that could be trimmed. The elapsed times were measured with the perception subsystem embedded in a Raspberry Pi 3, the main computational unit as described in Section 3.

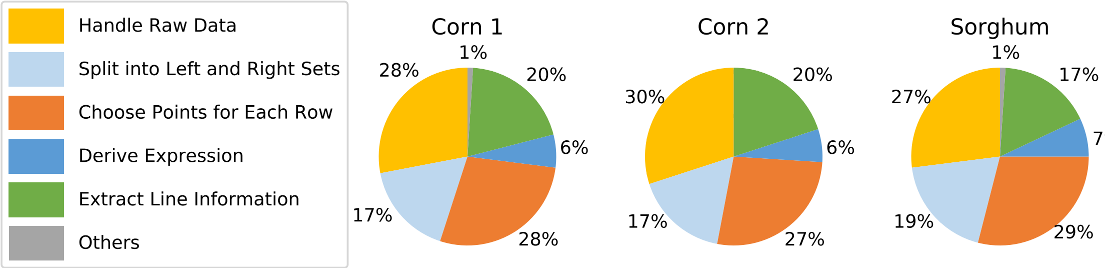


Figure 17: Average calculation time for ***Estimate*** internal stages. The distribution of time is similar regardless of the considered field, having *Handle raw Data* as the most time-consuming step. The average total time are 4.711 ms (Corn 1), 4.341 ms (Corn 2) and 5.409 ms (Sorghum).

Figure 17 shows that the distribution of time is similar, and the most time-consuming steps are *Handle Raw Data* and *Choose Points for Each Row*. The first has little room for improvement as it is due to the quantity of LiDAR measurements. The same is not true for the other step. Nonetheless, all calculations occurred in about 5 milliseconds for all cases and the difference happened due to how much the initial filtering reduced the original LiDAR set. The more it reduces, there are fewer points to be further processed, reducing execution time, which is the case of Corn 2. And the opposite happens for Sorghum. Other functions compose the remaining time, with especial consideration to *Heuristics* that does not take more than 2 us.

5.3.4 Soybean Field

Only initial experiments were conducted in the soybean field. For the current perception subsystem, it is required at least a gap of the robot's width (0.36 m) between left and right rows. This gap is necessary for the Handle Raw Data step presented in Section 4 as it relies on the fact that the space in front of the robot does not have relevant information. Unlike corn and sorghum, characterized by a single stem, soybean develops into branches that significantly close the navigable space between rows. Nonetheless, the experiments showed potential use of this perception subsystem under certain conditions for soybean fields, especially earlier stages with reduced number of branches.

5.4 Analysis and discussion

A positive outcome from sorghum experiments was a standardized data set collected with the side-facing camera for vision-based stalk width estimation. It was possible as a result of the proposed perception subsystem keeping the robot in the desired path. For instance, as it happened in the sorghum field, the robot can be set to maintain a non-zero distance from one of the rows. This task would demand additional scenario set up, such marks on the ground, to be executed by a human driver. Additionally, as it frees the operator of driving the robot through the crop, the operator may give greater attention to data monitoring, which is important to ensure reliable post-processing.

From the ground truth analysis, we could learn the particularities of measuring lane width using LiDAR readings. Although the mean value is close to the nominal value, Fig. 7 clearly shows that sudden changes happen even for manual ground truth. This is greatly influenced by leaves. When they are in the middle of the lane, they occlude the sensor, diminishing field of view. And when they are close to stalks, the leaves can completely hide the stems and there is no option other than using the leaves themselves to estimate a lateral distance.

Three hundred eighty one autonomous runs have been achieved with the proposed perception subsystem in corn and sorghum crops. Seventy of them were executed by external people who reported the quantities of failures (eight in total). But the 31 failures from the 311 runs shown in Table 1 were descriptively logged and they provide an insight of the robot's performance on field. 38.7% (12) of the failures is related to environment and 9.7% (3) to mechanical issues, thus the robot (2017 version used in this research) itself needed improvements to deal with a wider range of situations. First, a stronger powertrain may be able to overcome entangling cases such as weed infestation, high leaf coverage and ground covered with debris. Additionally, implementation of a surroundings assessment system would help taking decisions such as going through obstacles or going back. Six failures (19.3%) are due to control issues. For each field, the robot presented different behavior due differences in ground unevenness and visible wheels slip. These aspects could not be completely coped by the implemented classic PID whose gains are fixed. Start issues correspond to 12.9% (4) of the failures. Additional work must be done to enable an automatic awareness of initial conditions (expected lane width and initial distance to center). While such values need to be given by the user, the system will be prone to failures when robot's initial condition does not meet inserted ones.

Perception issues correspond to 19.4% (6) of the failures. They happened in plots whose lane width were visually different from the nominal value. Although the lane width at ground level is pretty much guaranteed due to plantation techniques, the same cannot be affirmed for the lane width at LiDAR's height e.g. because stalks grew bending towards the center of the lane. As the growth issues can affect only part of the plot, a single nominal lane width is not representative. Currently, the algorithm relies on a nominal lane width as a baseline to validate the estimated lane width and also to generate a fictitious lateral distance to replace an invalid estimate one. Therefore, correctly updating the nominal value is key to not discard estimates or to not generate wrong ones. Additionally, feature detection would provide at least anticipated information of incoming obstacles and leaf clusters. It would support the definition of regions with useful LiDAR readings instead of solely relying on past distance estimations. Such feature detection system could be implemented with a forward-facing camera or using a 3D LiDAR. In the second case, an option is adding a roll degree of freedom to the single layer LiDAR, as demonstrated by Zhang et al. (2013).

Figure 16 summarizes the evolution of the number of measurements that are used by the perception subsystem. In either cases (highly cluttered or not), a significant number of points is filtered out, and 35.5% to 38.5%of the initial set is used to obtain the lines representing the rows. While it is not desirable to reduce angular resolution, a reduction in the distance range would greatly reduce total cost of the system without performance decay.

Although a device with much greater computational power is also present, the time of execution obtained by using Raspberry Pi 3 is meaningful. It shows that it is possible to execute the perception subsystem in the same device that handles the robot's low-level, resulting in increased operation safety. Moreover, as it does not take more than 6 ms, there is plenty of time for additional features, including addition of other sensors, until it reaches the LiDAR update rate of 25 ms. As shown by Fig. 17, the most time-consuming steps that can be improved are *Choose Points for Each Row* and *Extract Line Information*. Although *Handle Raw Data* uses up to 30% of the calculation time, it happens due to raw data containing the highest number of elements to be processed.

In summary, given our results of 90.03% of success rate in the 311 assessed runs and overall 89.76% in the 381 reported runs, this work clearly demonstrates the feasibility of navigating cluttered agricultural environments autonomously using a 2-D LiDAR sensor.

Limitations and Further Work It should be noted that most of our test runs were in straight rows. This is not excessively limiting, since large parts of modern farms consist of long straight rows. The approach should be able to handle moderately curved rows with minor modification, as long as the portion of the point-cloud over which the straight-line templates are fit are locally "straight". Indeed we have anecdotal evidence for that. However, handling large curves remains open. End of rows, transition involving turns from one row to another, and transition between two concurrent plots separated by a large horizontal row (as is common practice in breeding fields) have not been considered in this present work. Transition between plots could be handled using our approach by adding the ability to detect point-clouds representing transitions or end of rows. However, turning at the end of the row from one row to another requires significant further work.

The most promising further work direction might be fusing vision information with the LiDAR based navigation to create a robust perception system. The main challenge with 2-D (and even 3-D) LiDAR in repetitive environments like crops, is that it may be difficult to distinguish between different features with only point-cloud information. Visual information plays a significant role, for example, in finding ends of rows or obstacles that should be avoided (instead of filtered). Hence, efficient multi-sensor integration is an open direction of research that should be pursued in future work. Integrating INS with LiDAR or simple visual features with inertial and LiDAR is an immediate extension, and has been accomplished in non-agricultural environments in the past by various authors (Chowdhary et al., 2012, 2013; Goldberg et al., 2002; Tang et al., 2015; Chen, 2012). However, key advances could be made if semantic information extracted from a machine vision system is fused with LiDAR data.

6 Conclusions

A LiDAR-based perception subsystem for autonomous navigation between rows of cluttered and high leaf covered agricultural fields has been presented. The main contribution consists of demonstrating reliable within-row navigation using only a low-cost 2-D LiDAR. This was established through extensive testing in several corn and sorghum production/breeding fields. The success in four different fields demonstrated the subsystem's capability of handling multiple environments as long three requirements are met: known lane width; continuous rows without gaps or sharp turns; and relatively obstacle-free lanes.

The significance of our results lies in the fact that it clearly establishes that one single-layer (i.e. 2-D) LiDAR sensor is sufficient for within row navigation in realistic and challenging fields. Our algorithm enables the design of robots that can work reliably and autonomously under cluttered crop canopies where GNSS is untrustworthy due to multi-path error and signal attenuation. As such, the presented system will ensure that once a robot enters the row from one end, it will emerge from the other end even in leafy fields with a lot of weeds, as long as the requirements mentioned above are satisfied. Nevertheless, inclusion of additional sensors will be necessary to ensure full autonomy throughout the season. Particularly, fusion with inertial sensors may lead to more reliable yaw-angle estimation, and fusion with machine vision will enable the robot to handle end of rows, row-to-row transition, and unforeseen-within-row obstacles. In addition, any LiDAR based approach will be fundamentally limited by the fact that the plants must be sufficiently tall to that LiDAR based techniques might be applied. This issue may be tackled to some extent by carefully positioning the sensor. However, chances are that the navigation will be unreliable if only LiDAR is used, considering very young plants. When plants are very small, this is not a limiting factor. Indeed, GNSS is usable since there is little multi-path error due to cluttered canopies.

In summary, the presented algorithm and field-testing results clearly demonstrate the feasibility of within row-navigation of small robots using a relatively low-cost LiDAR sensor. The good news is that the presented algorithm completes within less than 27.6% of the LiDAR update time on the Raspberry-Pi computer. LiDAR update time is 25 ms, while the algorithm only used 6.9 ms at most. This low-computational cost nature of the algorithm leaves much processing time for further improvements to the algorithm.

Acknowledgments

Vitor A. H. Higuti was supported by Sao Paulo Research Foundation (FAPESP) grant number 2017/00033-7. Vitor was a visiting researcher at UIUC in Chowdhary's group during large portions of this research. Andres E. B. Velasquez was supported by Coordination of Superior Level Staff Improvement - Brazil (CAPES) - Finance Code 001. The data, equipment and facilities utilized in this research were partially supported by the Advanced Research Projects Agency-Energy Award Number DE-AR0000598, NSF SBIR #1820332, and an award from the USDA/NIFA-NSF joint CPS program. The authors would like to thank DASLab (Distributed Autonomous Systems Laboratory directed by Chowdhary) members for their assistance in the development and testing of the robotic platform: Sri Vuppala Theja, Nolan Thomas Replogle, Benjamin C. Thompson, Beau Barber, Hunter Young and Dr. Erkan Kayacan. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

References

Bak, T. and Jakobsen, H. (2004). Agricultural Robotic Platform with Four Wheel Steering for Weed Detection. *Biosystems Engineering*, 87(2):125–136.

Bakker, T., van Asselt, K., Bontsema, J., Müller, J., and van Straten, G. (2011). Autonomous navigation using a robot platform in a sugar beet field. *Biosystems Engineering*, 109(4):357–368.

Barawid, O. C., Mizushima, A., Ishii, K., and Noguchi, N. (2007). Development of an Autonomous Navigation System using a Two-dimensional Laser Scanner in an Orchard Application. *Biosystems Engineering*, 96(2):139–149.

Bechar, A. and Vigneault, C. (2016). Agricultural robots for field operations: Concepts and components. *Biosystems Engineering*, 149:94–111.

Bell, J., MacDonald, B. A., and Ahn, H. S. (2016). Row following in pergola structured orchards. In

2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 640–645, Daejeon, South Korea. IEEE.

Bell, T. (2000). Automatic tractor guidance using carrier-phase differential GPS. *Computers and Electronics in Agriculture*, 25(1-2):53–66.

Bergerman, M., Maeta, S. M., Zhang, J., Freitas, G. M., Hamner, B., Singh, S., and Kantor, G. (2015). Robot Farmers: Autonomous Orchard Vehicles Help Tree Fruit Production. *Robotics & Automation Magazine*, 22(1):54–63.

Cheein, F. A., Steiner, G., Paina, G. P., and Carelli, R. (2011). Optimized EIF-SLAM algorithm for precision agriculture mapping based on stems detection. *Computers and Electronics in Agriculture*, 78(2):195–207.

Chen, S. Y. (2012). Kalman filter for robot vision: a survey. *IEEE Transactions on Industrial Electronics*, 59(11):4409–4420.

Chowdhary, G., Johnson, E. N., Magree, D., Wu, A., and Shein, A. (2013). GPS-denied Indoor and Outdoor Monocular Vision Aided Navigation and Control of Unmanned Aircraft. *Journal of Field Robotics*, 30(3):415–438.

Chowdhary, G., Sobers, D. M., Pravitra, C., Christmann, C., Wu, A., Hashimoto, H., Ong, C., Kalghatgi, R., and Johnson, E. N. (2012). Self-contained autonomous indoor flight with ranging sensor navigation. *Journal of Guidance, Control, and Dynamics*, 35(6):1843–1854.

Department of Economic and Social Affairs of the United Nations (2015). World population prospects: The 2015 Revision, Key Findings and Advance Tables.

dos Santos, F. B. N., Sobreira, H. M. P., Campos, D. F. B., dos Santos, R. M. P. M., Moreira, A. P. G. M., and Contente, O. M. S. (2015). Towards a Reliable Monitoring Robot for Mountain Vineyards. In *2015 IEEE International Conference on Autonomous Robot Systems and Competitions*, pages 37–43, Vila Real, Portugal. IEEE.

Durrant-Whyte, H. and Bailey, T. (2006). Simultaneous localization and mapping: part I. *IEEE robotics & automation magazine*, 13(2):99–110.

English, A., Ross, P., Ball, D., and Corke, P. (2014). Vision based guidance for robot navigation in agriculture. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1693–1698, Hong Kong, China. IEEE.

Fahlgren, N., Gehan, M. A., and Baxter, I. (2015). Lights, camera, action: High-throughput plant phenotyping is ready for a close-up. *Current Opinion in Plant Biology*, 24:93–99.

FAO (2011). *The State of the World's land and water resources for Food and Agriculture (SOLAW) - Managing systems at risk*. FAO and Earthscan, London, England.

Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

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

García-Santillán, I. D., Montalvo, M., Guerrero, J. M., and Pajares, G. (2017). Automatic detection of curved and straight crop rows from images in maize fields. *Biosystems Engineering*, 156:61–79.

Goldberg, S. B., Maimone, M. W., and Matthies, L. (2002). Stereo vision and rover navigation software for planetary exploration. In *Proceedings, IEEE Aerospace Conference*, volume 5, pages 5–5, Big Sky, USA. IEEE, IEEE.

Grimstad, L., Pham, C. D., Phan, H. T., and From, P. J. (2015). On the design of a low-cost, light-weight, and highly versatile agricultural robot. In *2015 IEEE International Workshop on Advanced Robotics and its Social Impacts (ARSO)*, pages 1–6, Lyon, France. IEEE.

Guerrero, J. M., Ruz, J. J., and Pajares, G. (2017). Crop rows and weeds detection in maize fields applying a computer vision system based on geometry. *Computers and Electronics in Agriculture*, 142(PART A):461–472.

Haibo, L., Shuliang, D., Zunmin, L., and Chuijie, Y. (2015). Study and Experiment on a Wheat Precision Seeding Robot. *Journal of Robotics*, 2015:9.

Hall, D., Dayoub, F., Kulk, J., and McCool, C. (2017). Towards unsupervised weed scouting for agricultural robotics. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5223–5230, Singapore, Singapore. IEEE.

Hiremath, S., van Evert, F. K., ter Braak, C., Stein, A., and van der Heijden, G. (2014a). Image-based particle filtering for navigation in a semi-structured agricultural environment. *Biosystems Engineering*, 121:85–95.

Hiremath, S. A., van der Heijden, G. W. A. M., van Evert, F. K., Stein, A., and Ter Braak, C. J. F. (2014b). Laser range finder model for autonomous navigation of a robot in a maize field using a particle filter. *Computers and Electronics in Agriculture*, 100:41–50.

Jiang, G., Wang, Z., and Liu, H. (2015). Automatic detection of crop rows based on multi-ROIs. *Expert Systems with Applications*, 42(5):2429–2441.

Kayacan, E., Thompson, B., Young, H., and Chowdhary, G. (2018). High precision of an Ultra-Compact 3D Printed Field Robot in the Presence of Slip. In *International Conference on Robotics and Automation (ICRA)*, Brisbane, Australia. IEEE. Submitted.

Lepej, P. and Rakun, J. (2016). Simultaneous localisation and mapping in a complex field environment. *Biosystems Engineering*, 150:160–169.

Mahlein, A.-K. (2016). Plant Disease Detection by Imaging Sensors – Parallels and Specific Demands for Precision Agriculture and Plant Phenotyping. *Plant Disease*, 100(2):241–251.

Montalvo, M., Pajares, G., Guerrero, J. M., Romeo, J., Guijarro, M., Ribeiro, A., Ruz, J. J., and Cruz, J. M. (2012). Automatic detection of crop rows in maize fields with high weeds pressure. *Expert Systems with Applications*, 39(15):11889–11897.

Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. (2002). FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI)*, pages 593–598, Alberta, Canada. AAAI.

Mueller-Sim, T., Jenkins, M., Abel, J., and Kantor, G. (2017). The Robotanist: A ground-based agricultural robot for high-throughput crop phenotyping. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3634–3639, Singapore, Singapore. IEEE.

Nørremark, M., Griepentrog, H. W., Nielsen, J., and Søgaard, H. T. (2008). The development and assessment of the accuracy of an autonomous GPS-based system for intra-row mechanical weed control in row crops. *Biosystems Engineering*, 101(4):396–410.

Pedersen, S. M., Fountas, S., Have, H., and Blackmore, B. S. (2006). Agricultural robots—system analysis and economic feasibility. *Precision agriculture*, 7(4):295–308.

Rahaman, M. M., Chen, D., Gillani, Z., Klukas, C., and Chen, M. (2015). Advanced phenotyping and phenotype data analysis for the study of plant growth and development. *Frontiers in Plant Science*, 6:1–15.

Reina, G., Milella, A., Rouveure, R., Nielsen, M., Worst, R., and Blas, M. R. (2016). Ambient awareness for agricultural robotic vehicles. *Biosystems Engineering*, 146:114–132.

Rovira-Más, F., Chatterjee, I., and Sáiz-Rubio, V. (2015). The role of GNSS in the navigation strategies of cost-effective agricultural robots. *Computers and Electronics in Agriculture*, 112:172–183.

Ruckelshausen, A., Biber, P., Dorna, M., Gremmes, H., Klose, R., Linz, A., Rahe, R., Resch, R., Thiel, M., Trautz, D., and Weiss, U. (2009). BoniRob: an autonomous field robot platform for individual plant phenotyping. In van Henten, E. J., Goense, D., and Lokhorst, C., editors, *Precision Agriculture '09, 7th European Conference on Precision Agriculture (ECPA)*, pages 841–847, Wageningen, Netherlands. Wageningen Agricultural Publishers.

Sobers, D. M., Yamaura, S., and Johnson, E. N. (2010). Laser-aided inertial navigation for self-contained autonomous indoor flight. In *AIAA Guidance, Navigation, and Control Conference (GNC'10)*, pages 1–58, Toronto, Canada.

Tabile, R. A. (2012). *Development of a modular and multifunction robotic platform for data acquisition in precision agriculture system*. P.h.d. thesis, University of Sao Paulo, São Carlos, SP, Brazil.

Tang, J., Chen, Y., Niu, X., Wang, L., Chen, L., Liu, J., Shi, C., and Hyyppä, J. (2015). LiDAR scan matching aided inertial navigation system in GNSS-denied environments. *Sensors*, 15(7):16710–16728.

Thrun, S. and Leonard, J. J. (2008). Simultaneous localization and mapping. In Siciliano, B. and Khatib, O., editors, *Springer handbook of robotics*, pages 871–889. Springer-Verlag Berlin Heidelberg, Berlin, Germany.

Torgersen, J. (2014). *Mobile Agricultural Robot Independent Four Wheel Ackerman Steering*. mathesis, Norwegian University of Life Sciences, Ås, Norway.

Troyer, T. A., Pitla, S., and Nutter, E. (2016). Inter-row Robot Navigation using 1D Ranging Sensors. *IFAC-PapersOnLine*, 49(16):463–468.

Velasquez, A. E. B., Higuti, V. A. H., Guerrero, H. B., Milori, D. M. B. P., Magalhaes, D. V., and Becker, M. (2016). Helvis - a Small-Scale Agricultural Mobile Robot Prototype for Precision Agriculture. In *13th International Conference of Precision Agriculture*, page 17, St. Louis, Missouri, USA. International Society of Precision Agriculture.

Wang, C.-C., Thorpe, C., and Thrun, S. (2003). Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. In

2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422), volume 1, pages 842–849, Taipei, Taiwan, Taiwan. IEEE, IEEE.

Wang, H. and Noguchi, N. (2016). Autonomous maneuvers of a robotic tractor for farming. In *2016 IEEE/SICE International Symposium on System Integration (SII)*, pages 592–597, Sapporo, Japan. IEEE.

Zhai, Z., Zhu, Z., Du, Y., Song, Z., and Mao, E. (2016). Multi-crop-row detection algorithm based on binocular vision. *Biosystems Engineering*, 150:89–103.

Zhang, C., Noguchi, N., and Yang, L. (2016). Leader-follower system using two robot tractors to improve work efficiency. *Computers and Electronics in Agriculture*, 121:269–281.

Zhang, J., Chambers, A., Maeta, S., Bergerman, M., and Singh, S. (2013). 3D perception for accurate row following: Methodology and results. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5306–5313, Tokyo, Japan. IEEE.

Zhao, S. and Zhang, Z. (2016). A new recognition of crop row based on its structural parameter model. *IFAC-PapersOnLine*, 49(16):431–438.