LIDAR-based Detection of Furrows for Agricultural Robot Autonomous Navigation

Javier Luna-Santamaria, Jose Ramiro Martinez-de Dios, Anibal Ollero
GRVC Robotics Lab, Univ. de Sevilla, Spain. Emails: javierluna3@gmail.com, {jdedios, aollero}@us.es

Abstract

Robust and accurate autonomous navigation is a main challenge in agricultural robotics. This paper presents a LIDAR-based processing system for autonomous robot navigation in crops with high vegetation density. The method detects and locates the crop furrows and provides them to the robot control system, which guides the robot such that its caterpillar tracks move along the furrows preventing damages in the crop. The proposed LIDAR-based processing pipeline includes various inconsistencies removal and template matching steps to deal with the high noise level of LIDAR scans. It has been implemented in C++ using ROS Noetic and validated in two different plantations with different crop growth status.

Keywords: Agricultural robots, LIDAR processing, robot applications.

1 Introduction

The need for improving crop production to fulfill the growing food demand and, at the same time, reducing the use of water and fertilizers and improving product quality has motivated intense R&D activities in autonomous robots for agriculture. In the last years a number of agricultural robots have been developed. Some of them have been designed as general purpose robots, such as Swagbot [1] or Agrobot [2]. The complexity of the robot autonomous operation in unstructured environments has focused the development of robots for specific agricultural tasks such as disease identification, mechanical and chemical weeding, pollination, pruning, or harvesting, among many others [3]. Some few robots have been designed for a variety of crops. Some examples are: Cäsar [4], designed for orchard or vineyard applications, or Greenbot [5], which is devised for horticulture, fruit, and arable farming. However, the most common approach is to specify and design a robot for a particular crop. That is the case, for instance, of AgBot [6] for corn, Lumai-5 [7] for wheat, eA-GROBOT [8] for cotton and groundnut, Ted [9] for grape, or RIPPA [10] for lettuce and cauliflower.

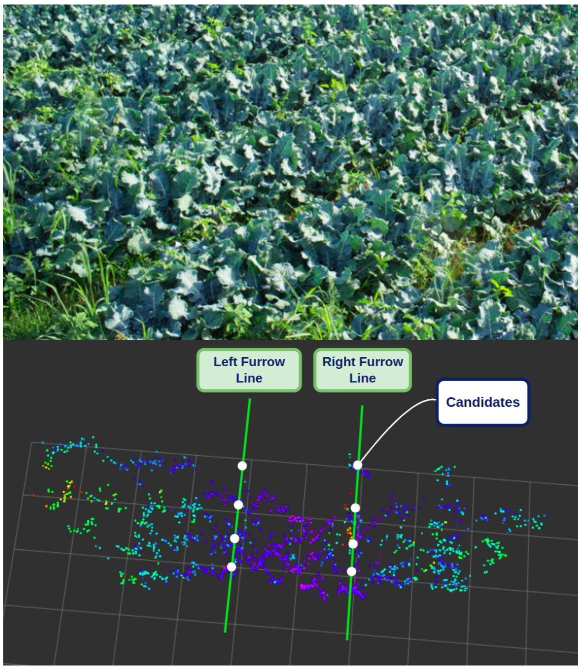

Figure 1: Top) visual image of a crop. Bottom) 3D LIDAR scan vegetation.

Autonomous navigation in complex unstructured environments is one of the main challenges of agricultural robots. A good number of autonomous navigation systems for these robots rely on the use of GNSS, see e.g. [5], [9]. However, GNSS does not always provide the accuracy and robustness required for complex tasks such as harvesting. Some few robots, such as K-Weedbot [11], use vision for navigation. However, agricultural environments have drastic lighting changes that induce difficulties in the use of vision systems for autonomous navigation. GNSS is often combined with other external sensors to increase the accuracy of navigation and task execution. In robots such as Agrirobot [12] and in [13], GNSS is combined with camera and LIDAR. LIDAR is an interesting sensor for these tasks since it provides rich geometrical information, it is robust to changes in lighting conditions, and can operate at night.

This work deals with the autonomous navigation of agricultural robots operating in crops with high vegetation density only using LIDAR, see Figure

https://doi.org/10.17979/spudc.9788497498418.0728

1. Although LIDAR has been widely used in agricultural robot navigation, it has been generally combined with GNSS. The number of existing agricultural robots or perception systems using only LIDAR is very scarce. Also, existing robots that use LIDAR operate in crops with low or moderate vegetation densities, whereas our method is devised for crops with high vegetation densities.

This paper presents a LIDAR-based method to detect and locate furrows for autonomous guidance of ground robots in agricultural tasks. The method detects the furrows and provides them to the robot control system, which guides the robot equipped with caterpillar tracks to move along the furrows preventing damages in the crop. The method is composed of four LIDAR processing modules that: 1) filter the initial LIDAR scan; 2) extract candidate points likely to be caused by furrows; 3) perform an spatio-temporal analysis grouping the candidates to select the central furrows along which the robot caterpillar tracks move; and 4) extract the furrows lines, which are used for robot control. The LIDAR processing scheme includes several filtering and inconsistencies removal mechanisms to increase accuracy and robustness, and deal with the high noise level present in the incoming LIDAR point clouds. It has been implemented in C++ and validated in a wide range of conditions, see Figure 1, including crop growth status and ground conditions.

The rest of the paper is structured as follows. The general diagram of the adopted autonomous navigation system is presented in Section 2. The proposed LIDAR processing methods are summarized in Section 3. Section 4 presents the experimental validation. Section 5 closes the paper and highlights the main future steps.

2 General Description

Our objective is to devise a robust and low-cost solution, with moderate accuracy sensors and low computational requirements in order to enhance its applicability in real applications. The robot is an Unmanned Ground Vehicle equipped with a GNSS receiver, which accuracy cannot enable the required robot navigation. It is also equipped with a moderate-cost 3D LIDAR, consisting of four 2D scan channels with different elevation angles. The LIDAR is mounted at the front of the robot with a negative pitch angle of $-30°$ approximately.

The crop environment is assumed unstructured, without any markers and any adaptations that would constrain its applicability in real conditions. We assume that a map of the crop area is available. If the crop area has a polygonal shape, the coordinates of the polygon vertices are assumed known. The crop contains a number of parallel and approximately rectilinear furrows covering all the area. The furrows start and end at the boundaries of the crop. The separation distance between furrows is assumed known and approximately constant. The crop is assumed free of obstacles for navigation. Conversely to many existing robot navigation systems, our scheme does not require a detailed map nor an explicit estimation of the robot pose: they would require accurate sensors and navigation methods, increasing cost and onboard computational requirements. In our system the operator provides the boundaries of the crop. The robot detects the furrows and uses their location to guide the robot. When the robot has reached the crop boundary, it moves to the adjacent furrow.

Figure 2 shows a scheme of the developed architecture. *Trajectory Following* analyzes the LIDAR scans, detects the crop furrows, and provides them to the robot control system to enable navigation along the furrows, adapting to the potential furrows curves and irregularities. *Trajectory Planner* implements a state machine that triggers the execution of *Trajectory Following* for navigating in the plantation using the crop boundaries. Finally, the *Navigation Manager* synchronises all modules to enable the expected operation. This paper presents *Trajectory Following*, which is the most complex module. For brevity, the descriptions of modules *Trajectory Planner*, *Robot Control*, and *Navigation Manager* are omitted.

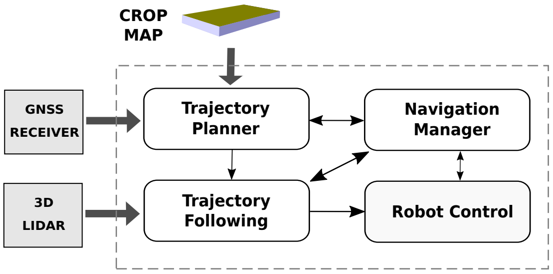

Figure 2: Scheme of the adopted LIDAR-based robot autonomous navigation system.

3 Methods

Our objective is to extract the furrows using solely the points resulting from each channel of the robot LIDAR scans. Each LiDAR point $p = (x, y, z, r, t)$ contains information of the point in the environment that reflected the laser beam: (x, y, z), 3D position on the LiDAR coordinate system; r, reflectivity of the impact point; and t, timestamp.

The proposed LIDAR processing scheme, see Fig-

https://doi.org/10.17979/spudc.9788497498418.0728

ure 3, is composed of four modules. First, *Pre-filtering* removes LIDAR points with no interest for furrow extraction, filtering out points that hamper furrow extraction, reducing the number of LIDAR points to be processed, hence decreasing the computation cost. *Candidate Extraction* selects the points with higher likelihood to be originated by furrows. Next, *Spatio-Temporal Analysis* examines the spatio-temporal consistency of the detected furrow candidates using a time window grouping them in different clusters. Finally, *Furrows Extraction* extracts the furrow using the spatio-temporal information. The output furrows are characterised as lines, from which the robot control references can be determined.

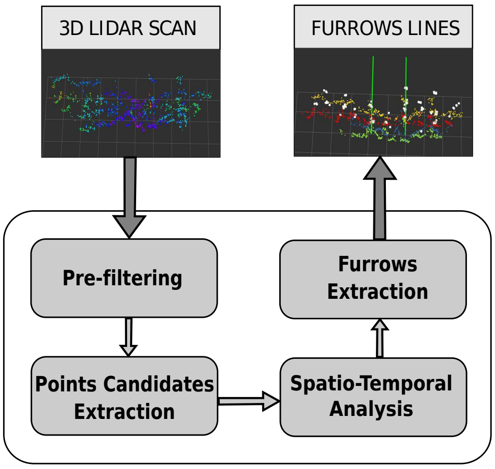

Figure 3: Pipeline of the proposed LIDAR-based processing scheme.

3.1 Pre-filtering

LIDAR scans in crops have high noise level, see Figure 1, due to the LIDAR reflections on different surfaces such as plant leaves, stems, or ridges. This module process each LIDAR channel in each scan and provides as output a point cloud with the points of interest for furrows extraction. It is divided into two main stages. The first one estimates a linear model that contains most of the points caused by LIDAR reflections on leaves. The next stage uses this model to select the points of interest from the point cloud and remove the remaining points.

First, for each LIDAR channel in each scan it estimates a linear model that contains the majority of the points originated by reflections on leaves. Leaf points are a large proportion of the point cloud. RANSAC [14] is adopted due to its robustness against outliers. In our problem RANSAC treats leaf points as inliers as they are scattered around a straight line. The remaining points are treated as outliers. In the second stage, the estimated linear model, see Figure 4, is used to decide which points should be selected for the furrows candidates extraction. An efficient linear classifier that uses discriminant function $f_D = \tilde{w}^\top \tilde{x}$ is adopted. $\tilde{w}^\top$ and $\tilde{x}^\top$ are defined as $\tilde{w}^\top = \begin{bmatrix} m & -1 & b \end{bmatrix}$, $\tilde{x}^\top = \begin{bmatrix} x & y & 1 \end{bmatrix}$, where m and b are the slope and intercept values obtained from the model estimation, and (x, y), the coordinates of the point being filtered. f_D distinguishes the points into two groups: those below f_D, and those above f_D, see Figure 4. The latter points are kept: they are of interest for extracting points candidates. The former, are removed.

For clarity, the operation each stage in the proposed scheme is illustrated using simulated data. The real LIDAR scan is too noisy and complex to make evident the operation of the modules. Results with real LIDAR data in crops are shown in Section 4. The operation of *Pre-filtering* in one channel of the LIDAR scan is illustrated in Figure 4. The figure shows the estimated model (black line), the points classified as of interest (in green), and the points removed (in grey). The LIDAR reference coordinate system is also shown.

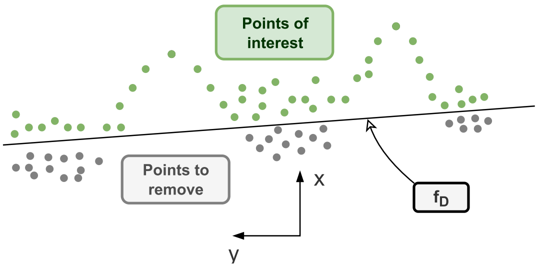

Figure 4: Illustration of the operation of *Pre-filtering*.

3.2 Candidate Points Extraction

This module extracts points that are candidate to be caused by furrows. It has two steps. LIDAR points caused by reflections on furrows have the highest range values. First, local maxima points for each LIDAR channel are detected. In the second step, each local maxima is submitted to a verification step in which it is checked whether the point is a candidate to belong to a furrow or unwanted points result of noisy point distributions.

First, the local maxima of each channel in each LIDAR scan are obtained by comparing each point against their neighbors and analyzing their distance to the estimated model. Point p_0 will be detected as local maximum if $d_{p_0} = \sqrt{{x_{p_0}}^2 + {y_{p_0}}^2}$, its distance to the model on the ground X-Y plane, is higher than d_{p_i}, the X-Y distance to the model

https://doi.org/10.17979/spudc.9788497498418.0728

of all the points p_i in the neighborhood of p_0:

$$d_{p_0} > d_{p_i} \quad \forall i \in S_{p_0}, \tag{1}$$

where S_{p_0} the neighborhood of p_0 defined as:

$$S_{p_0} := [p_0 - SPAN, p_0 + SPAN], \tag{2}$$

where $2SPAN$ is the size of the neighborhood. Too low values of $SPAN$ originate extra undesired local maxima as a consequence of noise in small sets of points. Too high values prevent detecting actual furrows points as candidates. The value used in the experiments is $SPAN = 24$ points.

Figure 5-top illustrates the operation of the first step. The rectilinear model and points resulting from *Pre-filtering* are shown respectively as a black line and green dots. The neighborhood used in local maxima detection is shown in dashed rectangles, and the detected local maxima, in black.

The second step verifies if the detected local maxima points are likely to be originated by a furrow or not. Even after filtering, the pattern of a furrow in a point cloud keeps being very noisy. In this step each local maximum point is verified by matching its neighbourhood using templates with triangular shapes consisting of two straight lines that intersect at the maximum point with angle α. Figure 5-bottom illustrates the operation and shows the template (red lines), angle α between the straight lines, and d, the distance from the local maximum to the model computed in *Pre-filtering*.

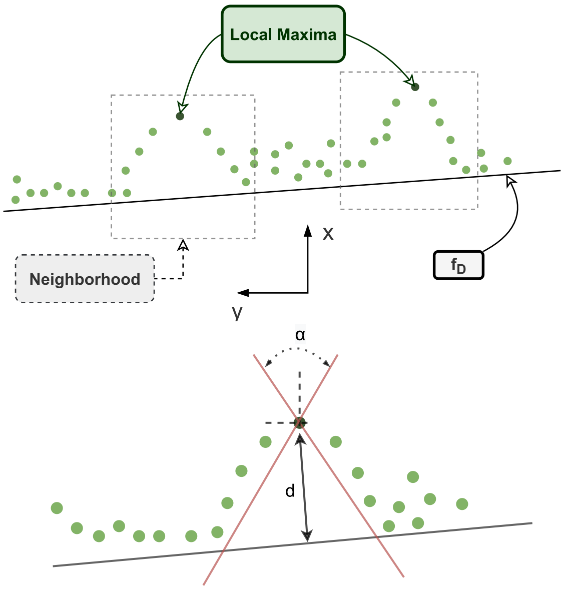

Figure 5: Illustration of: top) local maximum detection; and bottom) local maximum point verification through template matching.

3.3 Spatio-Temporal Analysis

This module analyses the spatio-temporal consistency of the furrow candidate points to improve robustness in furrow extraction. The robot motion is low when compared to LIDAR rate. Hence, the environment does not change significantly between consecutive scans. In order to improve accuracy and robustness, furrows are characterized with more spatial and temporal information.

The spatio-temporal analysis is performed by collecting the candidate points extracted from all the channels for N consecutive scans and then applying a clustering algorithm to group together those belonging to the same furrows. N is set taking into account the LIDAR frame rate, the output rate desired for furrows extraction, and the robot motion (to make sure that the environment perceived by the sensor within the time window has not changed significantly). The clustering algorithm adopted has been designed for the characteristics of the problem. First, furrow candidate points are grouped according to their Y-axis coordinate. Second, the separation in the Y-axis between the furrows is assumed known as described in Section 2. This enables using intra-cluster distance for clustering. Additionally, the clustering method can be adapted for a given number of clusters. In our problem since the inter-furrow distance is known, considering the LIDAR setting, the number of expected furrows in the scans can be foreseen. Clustering is as follows. If two furrow candidate points taken in scans within the temporal window have a distance in the Y-axis lower than the intra-cluster distance, these points are assigned as belonging to the same cluster.

Figure 6 illustrates operation of the spatio-temporal analysis. The candidate points of all channels of the LIDAR scans within the time window are grouped into a total of 4 furrow clusters, shown at the bottom of the figure.

3.4 Furrows Extraction

This module analyzes the furrow clusters obtained in *Spatio-Temporal Analysis* to extract the furrows lines. First, the two central furrows, along which the robot caterpillar tracks traverse, are extracted by matching the furrow clusters with a template based on the known inter-furrow distance in the Y-axis. The template is slided along the Y-axis in the central range of coordinates. The clusters' positions that minimizes the total squared error are taken as the position of the central furrows. Next, the furrow lines corresponding to each of the extracted central furrows are computed by fitting the candidate points. This data has a low level

https://doi.org/10.17979/spudc.9788497498418.0728

of outliers. A linear regression using least squares is used to obtain the straight lines in the ground X-Y plane that best fit the candidate points corresponding to the central furrows. The adopted model is described by the slope m, and the intercept b. Both central furrows are approximately parallel, and have the same slope. The candidate points for all channels from both furrows are mixed to obtain the slope of the model. This enables increasing accuracy, which is particularly interesting for the clusters that do not have candidate points from all the scan channels. Afterwards, the intercept b of each furrow line is determined separately using only the candidate points from each central furrow.

Figure 6 illustrates the operation of furrows extraction. It computes the coordinates on the ground X-Y plane of the central furrows. The furrows lines are provided to the robot controller.

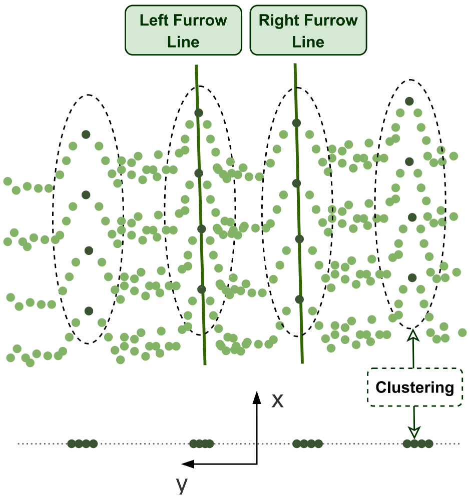

Figure 6: Illustration of furrows extraction. The resulting furrow lines are shown.

4 Experiments

The proposed method has been experimentally validated in various broccoli plantations in the Spanish province of Sevilla in a wide variety of conditions including crop growth status and ground. The 3D LIDAR used was a Pepperl+Fuchs R2300, which provides a scanning rate of 25 Hz, 4 channels per scan, an angular resolution of 0.1°, 100° horizontal, and 9° vertical field of view, and maximum range of 10 m. The LIDAR was placed at the front of the robot at 1.6 m from the ground, and with a pitch angle of $-30°$. The robot moved approximately at a speed of 30 cm/s. The scheme has been implemented in C++ using ROS Noetic. The number of scans to define the time window for candidate points accumulation in *Temporal analysis* was set to $N = 5$. The inter-furrow distance was approximately 1.5 m in all the scenarios.

Figure 7 summarizes the results when applying each module of the processing scheme to a LIDAR scan in one experiment. The point cloud from the input scan is shown in Figure 7-a. There are three furrows in the field of view of the LIDAR, the two central ones and one more at the left side of these. Figure 7-b shows the resulting points after applying module *Pre-filtering* to the points in one channel. The estimated model is represented with a green straight line. The points below the straight line are filtered out as not interesting for furrow extraction. The point cloud from the filtered LIDAR scan (considering all channels) is shown in Figure 7-c. Next, the furrow candidate points are extracted from the filtered point cloud as described in Section 3.2. The candidates are shown in white color in Figure 7-d. Next, *Spatio-Temporal Analysis* computes the clusters from the candidate points accumulated from all channels during $N = 5$ scans. The resulting clusters represented in different colors are shown in Figure 7-e below the point cloud. The points of the same cluster are represented by the same colour. Finally, *Furrows Extraction* detected the lines corresponding to the central furrows. The resulting extracted furrow lines are shown in Figure 7-f. They are provided as input to the robot control system.

Figure 8 summarizes the performance of the method in crops with different sizes of plants: large (a), medium (b), and small (c). For each case, the figure shows the initial point cloud from a LIDAR scan (left), resulting clusters after *Spatio-Temporal Analysis* (center), and resulting central furrow lines (right). In the three cases the scheme performed as expected.

The scheme was validated in sets of experiments in a wide variety of conditions including crop growth status and ground conditions. In average the scheme successfully detected and located the furrows in 90% of the LIDAR scans. The success rate was higher for crops with smaller plants than with larger plants, due to the vegetation occluding the LIDAR. Even with large plants the success rate was above 85%. Moreover, the robustness has been enhanced to avoid providing false references to the robot controller. A time consistency filter was implemented to detect and discard if the resulting furrow lines are found inconsistent with the furrow lines obtained in previous scans. Thus, two seconds is the maximum time that no valid furrow line has been provided in all the performed experiments, specifically in the crop with the largest plants. Being that the worse performance, the method performance is acceptable, as

https://doi.org/10.17979/spudc.9788497498418.0728

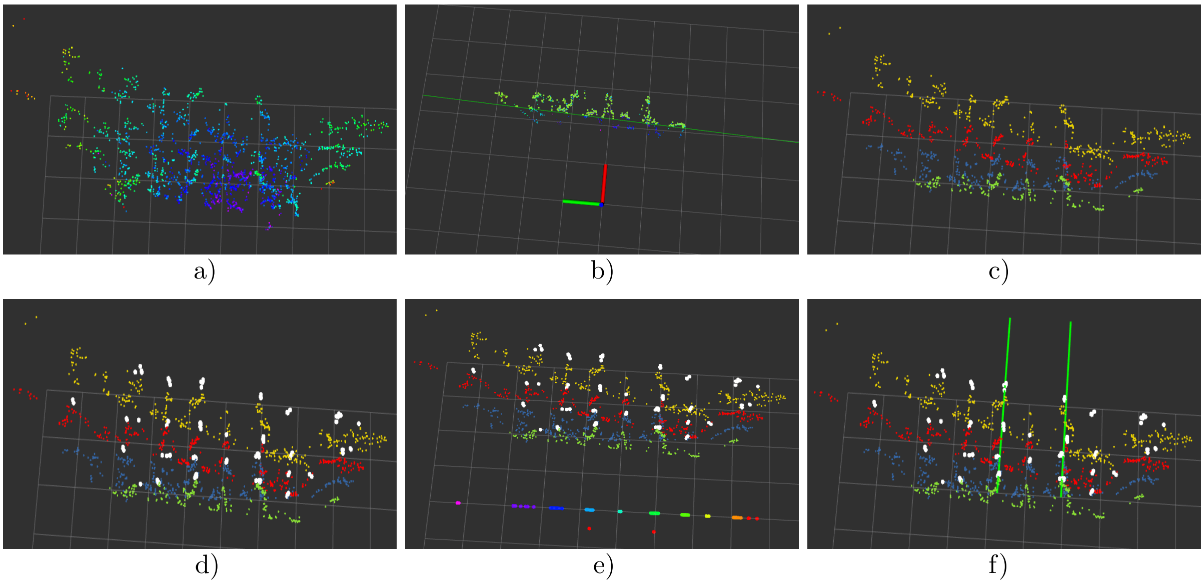
a) b) c) d) e) f)

Figure 7: Results after applying each module of the processing scheme to a LIDAR scan in one experiment: a) initial point cloud from a LIDAR scan; b) points resulting from applying *Pre-filtering* to a single LIDAR channel; c) point cloud (with all channels) resulting after *Pre-filtering*; d) candidate points (in white colour) resulting from *Point Candidate Extraction*; e) resulting clusters after *Spatio-Temporal Analysis*, the points in each cluster are in a different colour; and f) central furrow lines resulting from *Furrows Extraction*.

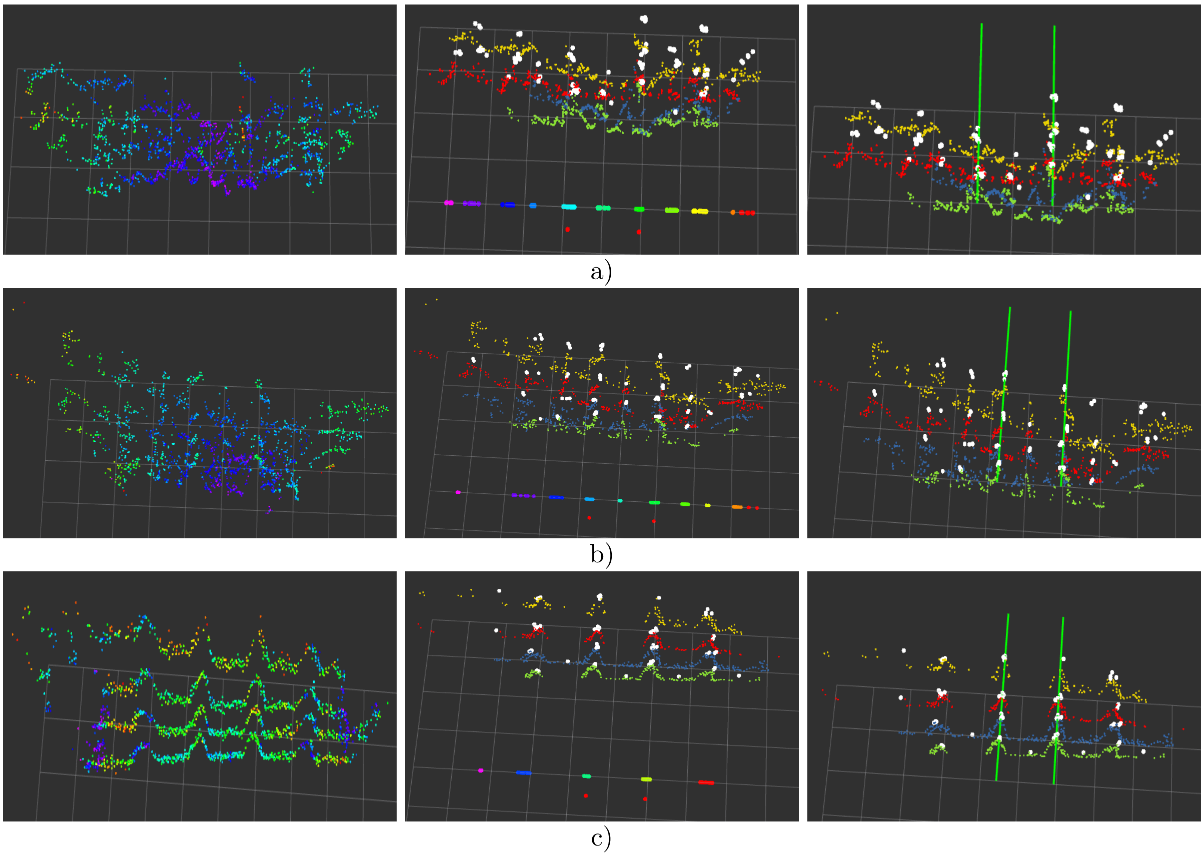
a) b) c)

Figure 8: Performance of the proposed scheme with a) large , b) medium, and c) small plants. For each case the figure shows: left) initial point cloud from a LIDAR scan; center) resulting clusters after *Spatio-Temporal Analysis*; and right) resulting central furrow lines.

due to the the low motion of the robot, it can keep navigating with the previous reference. Besides robust, the proposed method is computationally efficient. It has been tested in real time in a Intel Core i5-7200U (7th Gen) processor being the output rate of furrows extracted approximately 5 Hz, sufficient for the desired robot navigation speed.

5 Conclusions

Autonomous navigation is one of the main difficulties of agricultural robots. This paper presented a LIDAR-based processing scheme for detection and localization of furrows in crops with high vegetation density. The detected furrows are input of the robot control system such that the robot tracks can accurately move along the furrows. The processing pipeline 1) filters each LIDAR scan; 2) extracts candidate points likely to be caused by furrows; 3) selects furrows using spatio-temporal information of the candidates; and 4) extracts the furrows lines. The scheme was validated in crops in a wide range of conditions.

Future works includes more extensive validation and the development the LIDAR processing methods to guide the robot in manoeuvres of moving between adjacent crop furrows.

References

[1] Wallace, N.D., Kong, H., Hill, A.J., Sukkarieh, S. L., (2019) “Energy aware mission planning for WMRs on uneven terrains”, *IFAC-PapersOnLine*, vol. 52, no 30, pp 149–154.

[2] Avrora Robotics. Agrobot Project-Automation of Agriculture. 2020. Available online: https://avrora-robotics.com/en/projects/agrobot/ (accessed on 27 June 2022).

[3] Oliveira, L.F.P., Moreira, A.P., Silva, M. (2021) “Advances in Agriculture Robotics: A State-of-the-Art Review”, *Robotics*, vol. 10, no 2, pp 52.

[4] Raussendorf. Fruit Robot. Available online: https://www.raussendorf.de/en/fruit-robot.html (accessed on 27 June 2022).

[5] Precision Makers. GREENBOT. Available online: https://precisionmakers.com/ greenbot (accessed on 27 June 2022).

[6] Khan, N., Medlock, G., Graves, S., Anwar, S. (2018) “GPS Guided Autonomous Navigation of a Small Agricultural Robot with Automated Fertilizing System”, *SAE Technical Paper*, vol. 1, pp 1.

[7] Haibo, L., Dong, S., Zunmin, L., Chuijie, Y. (2015) “Study and Experiment on a Wheat Precision Seeding Robot”, *Journal of Robotics*, vol. 1, pp 1–9.

[8] Pilli, S.K., Nallathambi, B., George, S.J., Diwanji, V. (2015) “eAGROBOT-A robot for early crop disease detection using image processing”, *In Proc. 2nd Intl. Conf. on Electronics and Communication Systems*, pp 1684–1689.

[9] Naio Tecnologies. Ted-Multifunctional Straddling Vineyard Robot. 2020. Available online: https://www.naio-technologies.com/wp-content/uploads/2019/04/brochure-TED-ENGLISH-3.pdf (accessed on 27 June 2022).

[10] Bogue, R. (2016) “Robots poised to revolutionise agriculture”, *Industrial Robot*, vol. 43, pp 450–456.

[11] Choi, K.H., Han, S.K., Han, S.H., Park, K.H., Kim, K.S., Kim, S. (2015) “Morphology-based guidance line extraction for an autonomous weeding robot in paddy fields”, *Computers and Electronics in Agriculture*, vol. 113, pp 266–274.

[12] Adamides, G., Katsanos, C., Constantinou, I., Christou, G., Xenos, M., Hadzilacos, T., Edan, Y. (2017) “Design and development of a semi-autonomous agricultural vineyard sprayer: Human-robot interaction aspects”, *Journal of Field Robotics*, vol 34, pp 1407–1426.

[13] Berenstein, R., Edan, Y. (2018) “Automatic Adjustable Spraying Device for Site-Specific Agricultural Application”, *IEEE Transactions on Automation Science and Engineering*, vol 15, pp 641–650.

[14] 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*, vol. 24, no 6, pp 381–395.

https://doi.org/10.17979/spudc.9788497498418.0728