A GNSS-free navigation strategy for orchards

A. Durand-Petiteville[1], V. Cadenat[2,3], A. F. R. Araújo[4], A. Herbulot[2,3]
A. Villemazet[3], A. F. L. A. da Silva[4], C. Riou[3]

[1]Departamento de Engenharia Mecânica, UFPE, Recife, PE, Brasil

[2]Université de Toulouse, Toulouse, France

[3]CNRS-LAAS, Toulouse, France

[4]Centro de Informática, UFPE, Recife, PE, Brasil

`adrien.durandpetiteville@ufpe.br`

`{cadenat,avillemaze,aherbulo,riou}@laas.fr`

`{aluizioa,aflas}@cin.ufpe.br`

Abstract. *This paper proposes a GNSS-free strategy allowing a robot to navigate through orchards autonomously. It relies on a perception system made of four cameras arranged to enlarge the robot's field of view. Thanks to this choice, it becomes possible to perform complete navigation (both alley crossing and headland maneuvers) using only visual data, thus increasing the task's robustness. Results validate the proposed strategy.*

1. Introduction

Agriculture in the twenty-first century has to face two main challenges: (i) the increase of the world population [Foley et al. 2011] and its necessary production augmentation under both economic and environmental constraints [Lenain et al. 2019]; and (ii) the farm labor shortages due to the insufficient attractiveness of agricultural work. Agricultural robotics technologies can tackle these two challenges by providing systems able to allow precision farming and help workers to fulfill their tasks. It may then be possible to maximize production while taking care of the environment [Vougioukas 2019].

The ARPON (Autonomous Robotic Platform for Orchard Navigation) project belongs to the movement aiming at using robotics as a tool to develop sustainable agriculture where production increase matches environmental and societal concerns. It is a joint international project between CNRS-LAAS and CIn, UFPE funded by both French and Brazilian research agencies (ANR and FACEPE). The goal is to design a framework allowing a mobile robot to autonomously navigate in commercial orchards. Indeed, guaranteeing a safe motion in these environments is a prior condition to the realization of any treatment or operation such as fruit transportation, harvesting, thinning, etc. Thus building a safe navigation strategy is a mandatory step towards efficient precision horticulture.

Autonomous navigation has been widely studied in robotics [Siegwart et al. 2011]. However, it appears that its application to orchards is challenging for several reasons. First, the GNSS signal is not always available to localize the robot because of the tree canopy, whereas this signal is widely used in the agricultural robotics domain especially in open fields [Li et al. 2009, Verbiest et al. 2020]. Second, as the orchard is a natural environment, it is subject to large visual variations. Its appearance significantly varies depending, not only on the seasons and the treatments applied to the trees but also on the weather and the daytime. It is then necessary to develop robust perception and control systems and to regularly update the environment map which is required to realize the large displacements. Moreover, the climatic conditions modify the ground, making it more or less slippery or accidental and inducing vibrations or undesired slippage. Finally, the orchard is an environment shared with human operators, other vehicles, or animals, leading to potential safety issues. All these aspects significantly challenge the design of a proper control strategy which, in addition, cannot rely on absolute data but only on local information, this latter being sometimes significantly reduced in some areas of the orchard, such as the headland[1] where fruit trees are not present.

To create such an efficient navigation system, we have designed an approach relying on a vision-based control strategy coupled with a topological map. The latter is less sensitive to environmental variations, while the former offers the necessary reactivity to unexpected events. Before describing the different methods, we first present the robotic system and its instrumentation which has been thought to be adapted to orchards.

2. The robotic system

The considered robot is the Hunter 2.0 robot by Agilex (see Fig. 1). It has been chosen for the following reasons. First, it is a car-like robot that has two control inputs, the steering angle, and the linear velocity. Its minimum turning distance is 1.6m, while its maximum velocity is fixed at 6 km/h. This mechanical structure appears to be well-suitable for the orchard because it eases the headland maneuvering and avoids damaging soils. Second, it can carry up to a 150kg payload, which makes it evolutive in terms of equipment. Third, it is also able to climb small slopes (less than 10 degrees) and small obstacles (less than 5cm), which is an interesting feature in an orchard.

As shown by Fig. 1, the robot has been equipped with several exteroceptive low-cost sensors: a laser-rangefinder (Slamtech RPLIDAR S1), and four RGB-D Intel Real Sense cameras (two D455 and two D435). The range of the D455 cameras is larger than the 435 one (0.6 – 6m versus 0.3 – 3m). It has thus been decided to fix the former at the front of the robot and the latter on its left and right sides. This particular positioning allows benefiting from an enlarged field of view which makes possible the detection of the trees both in the headlands and in the alleys. In this way, it will be possible to control the robot using exteroceptive information only and thus improve the execution robustness. The proposed sensor-based strategy mainly relies on the vision system and the corresponding perception algorithm.

[1]The free space beyond the alleys.

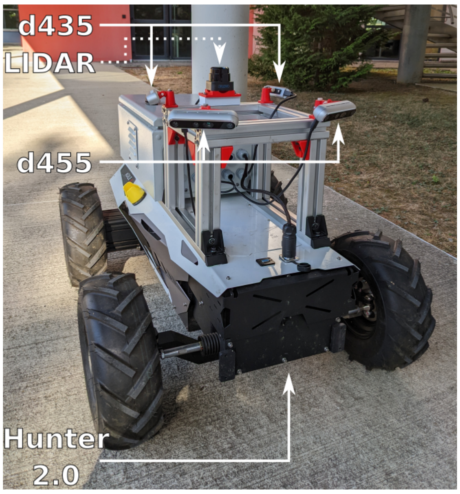

Figure 1. Robotic platform.

3. Perception

The perception system provides the required information for the row following and the U-turn. It relies on the algorithm [Durand-Petiteville et al. 2018] which computes the positions of the tree trunks by processing in real-time the range component of the point cloud data. It first computes a top view of the point cloud in which the trees are materialized by concavities corresponding to the empty spaces they leave in the point cloud (see orange triangles in Fig. 2). Then, it determines the origins of these concavities using an original coin-dropping technique [Durand-Petiteville et al. 2018].

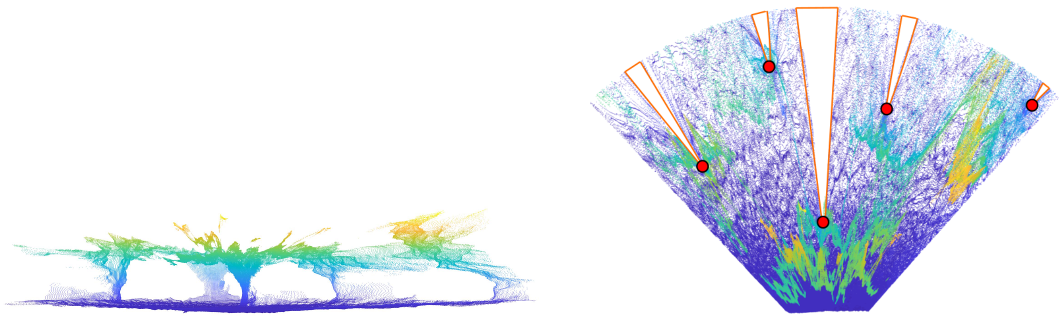

Figure 2. The orchard point cloud (left) and its top view (right)

This algorithm is separately applied to each of the four onboard RGB-D cameras, providing the tree trunk positions in the corresponding camera frame. These latter are then expressed into a unique frame (the laser frame in our case). A calibration process based on [Li et al. 2011] is then run prior to the navigation to determine the pose of each camera relative to the laser. An example of a result of this calibration is shown in Figure 3 where a vertical bar (in blue) was positioned between the two front cameras. After re-projecting the four point clouds in the laser frame, it can be seen that the structure of the bar is kept, thus showing the success of the calibration process.

The proposed algorithm has been implemented on an NVIDIA Jetson Xavier NX GPU which has been installed on our robot. It runs at a 15Hz rate, which is suitable for

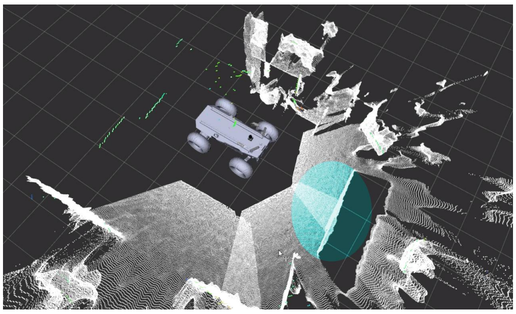

Figure 3. Calibration results: re-projection of the point clouds in the laser frame.

the planning and control process which are described hereafter.

4. Planning and control

The orchard navigation task mainly consists of sequencing row traversing with U-turns in the headland. To do so, we first compute a reference path before designing a control law to follow it. To define the reference path, we use the 3-D trunk coordinates provided by the perception system. From these data, we compute at each iteration a set of suitable points. Then an adequate curve is fitted to create the path.

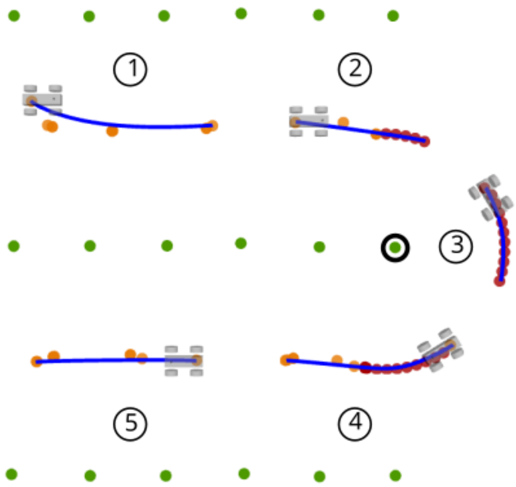

Figure 4. Examples of path generation. Green circle: tree - Black circle: pivot point - Orange circle: Voronoi vertex - Dark red circle: Spiral point - Blue curve: NURBS - Step 1/5: alley crossing - Step 2: path connecting the alley crossing to the headland maneuver - Step 3: headland maneuver - Step 4: path connecting the headland maneuver to the alley crossing..

To obtain the points, we consider separately the alley crossing and the headland U-turn. In the first case, a Voronoï diagram computed using 3-D trunk coordinates provides points roughly in the middle of the row (see Fig. 4, steps ① and ⑤). In the second case, the spiral model proposed in [Boyadzhiev 1999] is used. It is centered on the last detected tree[2] whose position is updated at each instant (*cf.* Fig. 4, step ③). The spiral

[2]In particular cases, the spiral can be centered on an object fixed at the end of the alley, such as a tensioner or a bin.

parameters are adapted at the beginning and the end of the U-turn to connect the (first (respectively, last)) spiral point to the (last (respectively, first)) Voronoï diagram vertices (*cf.* Fig. 4, steps ② and ④). Once the points are available, a NURBS (Non-Uniform Rational B-Spline [Piegl and Tiller 1996]) curve is fitted on these points to provide the reference path.

Now, it remains to follow this reference. Two control inputs are available on the robot: the linear velocity and the steering angle (see section 2). The first one has been fixed to a constant nonzero value to fulfill classical path following techniques [Cadenat et al. 2006]. The second one has been computed thanks to a Nonlinear Model Predictive Control (NMPC) [Grüne and Pannek 2017]. This optimal control technique allows computing a sequence of steering angles which minimizes the position and orientation errors between the robot and the path over a predefined prediction horizon under the constraints that the inputs are limited. The proposed control thus enables a geometrical convergence towards the path to be followed.

5. Orchard mapping

The presented reactive control strategy does not require localizing the robot. However, at a higher level of abstraction, the completion of certain tasks might require the localization of the robot, *i.e.,* only certain rows have to be harvested or the robot has to report a leak at a specific area. To do so, it is not necessary to metrically localize the robot with high accuracy, such as it is done in [Mur-Artal et al. 2015] or in [Pire et al. 2019] for an agricultural environment. Although this approach eases the control of the robots, it is limited to environments that are mainly static, structured, and of limited size. It might then be sufficient to rely on topometric or topological localization methods, giving the robot the ability to localize itself within sectors of the orchard. Indeed, topological maps offer an abstract and compact representation of the environment relying on graphs, where nodes represent distinctive places in the environment and arcs define the relations between them [Burgard et al. 2016]. When relying on a camera as the primary sensor, the topological mapping/localization problem can be seen as the visual place recognition problem [Lowry et al. 2016]. First, a set of images at several places in the environment are stored to create the graph. Next, the localization system has to match the current image with the image of the data set taken at the closest location to the robot's current pose.

In this project, we use a visual place recognition system relying on a set of Self-organizing Maps (SOM) [da Silva Júnior and Araújo 2022] coupled with the VGG-16 feature extractor [Simonyan and Zisserman 2014]. Thus, the image-matching process relies on visual features, instead of the raw image, reducing the sensibility of viewpoint variation and decreasing the amount of memory required to store the data. Moreover, the VGG-16 descriptor allows robustly extracting visual features in changing environments and does not require any environment-specific training. Regarding the visual map, it is divided into a given number of sub-sets used to train several SOMs. Thus, each node of the SOMs represents a sub-cluster of the original data set. This subspace clustering method to handle high-dimensional input data aims to perform the match process only inside a set of clustered images, instead of the entire data set. This approach significantly reduces the localization processing time, a necessary condition for using localization processes online.

6. Results

In this section, we present the preliminary results obtained during two experimental tests[3]. For the first test, the robot drives in an alley (Fig. 5(a)), starts a left-side U-turn at the end of the row (Fig. 5(b)), performs the headland maneuvers (Fig. 5(c)), and finally reaches the next alley (Fig. 5(d)) where it re-starts a row crossing. A similar sequence is presented in Fig. 6 where the robot switches from one alley to the next one by performing a right-side U-turn. In both experiments, the robot was able to cross the alley and perform the U-turn maneuvers despite the distribution of trees and branches, as well as variations in light and shade. This shows that our approach allows the robot to navigate in an orchard, based solely on onboard sensors and not requiring the use of GNSS.

(a) Alley crossing

(b) Beginning of the U-turn

(c) U-turn

(d) End of the U-turn

Figure 5. Key steps of the first experiment (left-side U-turn)

7. Conclusion

This paper has proposed a fully reactive GNSS-free navigation strategy for orchards. Its main strength relies on using data provided by four cameras adequately fixed on the robot to provide a wide field of view of the surroundings. In this way, it is possible to control the robot using exteroceptive data only, whether it is navigating in the row or in the headland. In addition, the use of such information makes the strategy less sensitive to orchard natural variations. Finally, it has been thought to be easily extendable to take into account new constraints, thus improving its adaptation skills to the evolution of the environment. The proposed methods have been validated on our robotic platform and the results have shown their interest in orchard navigation.

For future works, we plan to improve the perception algorithm by adding a filtering process to detect and track the trees and considering deep learning algorithms to

[3]Video: link

(a) Alley crossing (b) Beginning of the U-turn

(c) U-turn (d) End of the U-turn

Figure 6. Key steps of the second experiment (right-side U-turn)

use also 2D information. The control strategy will also be enhanced by introducing constraints to deal with obstacle avoidance. Finally, the topological mapping and localization algorithm will be modified to deal with repetitive environments.

References

Boyadzhiev, K. N. (1999). Spirals and conchospirals in the flight of insects. *The college mathematics Journal*, 30(1):23.

Burgard, W., Hebert, M., and Bennewitz, M. (2016). World modeling. *Springer handbook of robotics*, pages 1135–1152.

Cadenat, V., Souéres, P., and Hamel, T. (2006). A reactive path-following controller to guarantee obstacle avoidance during the transient phase. *International Journal of Robotics and Automation*, 21(4):256–265.

da Silva Júnior, M. R. and Araújo, A. F. R. (2022). Subspace clustering multi-module self-organizing maps with two-stage learning. In Pimenidis, E., Angelov, P., Jayne, C., Papaleonidas, A., and Aydin, M., editors, *Artificial Neural Networks and Machine Learning – ICANN 2022*, pages 285–296, Cham. Springer Nature Switzerland.

Durand-Petiteville, A., Le Flecher, E., Cadenat, V., Sentenac, T., and Vougioukas, S. (2018). Tree detection with low-cost three-dimensional sensors for autonomous navigation in orchards. *IEEE Robotics and Automation Letters*, 3(4):3876–3883.

Foley, J. A., Ramankutty, N., Brauman, K. A., Cassidy, E. S., Gerber, J. S., Johnston, M., Mueller, N. D., O'Connell, C., Ray, D. K., West, P. C., et al. (2011). Solutions for a cultivated planet. *Nature*, 478(7369):337.

Grüne, L. and Pannek, J. (2017). Nonlinear model predictive control. In *Nonlinear Model Predictive Control*, pages 45–69. Springer.

Lenain, R., Tricot, N., and Berducat, M. (2019). La robotique agricole: l'essor de nouveaux outils pour l'agro-écologie. *Sciences Eaux et Territoires*, 29:64–67.

Li, M., Imou, K., Wakabayashi, K., and Yokoyama, S. (2009). Review of research on agricultural vehicle autonomous guidance. *International Journal of Agricultural and Biological Engineering*, 2(3):1–16.

Li, Y., Ruichek, Y., and Cappelle, C. (2011). 3d triangulation based extrinsic calibration between a stereo vision system and a lidar. *Conference Record - IEEE Conference on Intelligent Transportation Systems*, pages 797–802.

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

Mur-Artal, R., Montiel, J. M. M., and Tardos, J. D. (2015). Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163.

Piegl, L. and Tiller, W. (1996). *The NURBS Book*. Springer-Verlag, New York, NY, USA, second edition.

Pire, T., Mujica, M., Civera, J., and Kofman, E. (2019). The rosario dataset: Multisensor data for localization and mapping in agricultural environments. *The International Journal of Robotics Research*, 38(6):633–641.

Siegwart, R., Nourbakhsh, I., and Scaramuzza, D. (2011). *Introduction to autonomous mobile robots*. A bradford book, Intelligent robotics and autonomous agents series. The MIT Press, second edition.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Verbiest, R. and Ruysen, K., Vanwalleghem, T., Demeester, E., and Kellens, K. (2020). Automation and robotics in the cultivation of pome fruit: Where do we stand today ? *Journal of Field Robotics*, 38(4):513–531.

Vougioukas, S. G. (2019). Agricultural robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 2:365–392.