

Computer-Vision Based Real Time Waypoint Generation for Autonomous Vineyard Navigation with Quadruped Robots

Lee Milburn

RIVeR Lab
IIT & NEU

Boston, Massachusetts

Milburn.l@northeastern.edu

Juan Gamba

Dynamic Legged Systems Lab
Istituto Italiano di Tecnologia

Genova, Italy

Juan.Gamba@iit.it

Miguel Fernandes

Advanced Robotics Lab
IIT & UniGe

Genova, Italy

Miguel.Ferreira@iit.it

Claudio Semini

Dynamic Legged Systems Lab
Istituto Italiano di Tecnologia

Genova, Italy

Claudio.Semini@iit.it

Abstract—The VINUM project seeks to address the shortage of skilled labor in modern vineyards by introducing a cutting-edge mobile robotic solution. Leveraging the capabilities of the quadruped robot, HyQReal, this system, equipped with arm and vision sensors, offers autonomous navigation and winter pruning of grapevines reducing the need for human intervention. At the heart of this approach lies an architecture that empowers the robot to easily navigate vineyards, identify grapevines with unparalleled accuracy, and approach them for pruning with precision. A state machine drives the process, deftly switching between various stages to ensure seamless and efficient task completion. The system's performance was assessed through experimentation, focusing on waypoint precision and optimizing the robot's workspace for single-plant operations. Results indicate that the architecture is highly reliable, with a mean error of 21.5cm and a standard deviation of 17.6cm for HyQReal. However, improvements in grapevine detection accuracy are necessary for optimal performance. This work is based on a computer-vision-based navigation method for quadruped robots in vineyards, opening up new possibilities for selective task automation. The system's architecture works well in ideal weather conditions, generating and arriving at precise waypoints that maximize the attached robotic arm's workspace. This work is an extension of our short paper presented at the Italian Conference on Robotics and Intelligent Machines (I-RIM), 2022 [1].

Index Terms—Agricultural Robotics, Computer-Vision, Autonomous Vineyard Navigation, Quadruped Control

I. INTRODUCTION



Fig. 1: IIT's HyQReal quadruped robot in a Vineyard close to Piacenza, Italy.

The wine industry has been experiencing a significant labor shortage in vineyards worldwide¹. Vineyards heavily rely on seasonal workers, including international workforces, for their labor needs. Consequently, they have been exploring the potential of robotic automation of seasonal work to address the labor shortage.

The VINUM² project uses IIT's HyQReal quadruped robot to do the winter pruning of grapevines autonomously with the Kivoa Gen3 robotic arm [2][3]; see Fig. 1, [4]. Quadruped robots can navigate obstacles and adjust their movements accordingly, making them well-suited for specific vineyards representing a complex and dynamic environment. This improved maneuverability in conjunction with a robotic pruning arm leads to a more efficient pruning process and reduces the need for manual labor. Ultimately, these benefits can contribute to improved growth and yield of grapevines. For example, HyQReal can move laterally as well as forward and backward. This extra movement allows for more mobility when searching for grapevines and more freedom of approach when moving to detected grapevines [5].

To autonomously winter-prune grapevines, the VINUM robot has to reliably and autonomously navigate vineyards, arriving at each grapevine that needs winter pruning. To reach each grapevine for pruning, the navigation stack has to autonomously generate waypoints relative to the grapevine which are ideal for the robotic workspace of the Kinova Gen3 arm attached to HyQReal, see Fig. 2. This approach allows the robot to perform selective, plant-by-plant task automation within the vineyard. The navigation pipeline uses an RGB-D camera instead of a more expensive, intensive sensor like LiDAR to make the system more cost-effective. However, an RGB-D camera produces erroneous camera detections that the architecture must overcome to be effective in a broader array of conditions. This paper proposes a reliable navigation architecture that generates precise waypoints based on RGB-D computer vision and higher-level control algorithms for

¹<https://www.winemag.com/2021/12/07/wine-industry-labor-supply/>

²<https://vinum-robot.eu/>

quadruped robots.

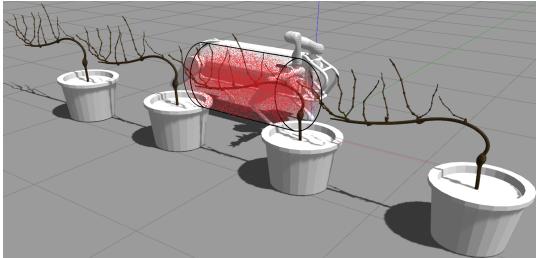


Fig. 2: Kinematic workspace (in red) of the Kinova Gen3 manipulator arm mounted on HyQReal.

The proposed navigation stack moves autonomously through vineyards, moving to precise waypoints autonomously generated from camera input. The navigation stack uses computer vision to detect the grapevine trunks and a filter using a rolling average of the detections to eliminate noisy input. It then generates waypoints to approach based on the detected grapevine trunks. We implemented instance segmentation for grapevine trunks using an RGB-D sensor. Detections of the grapevine trunks are made using a Mask-RCNN trained off a manually annotated dataset with over 500 images. Combining the higher-level control with the grapevine detections makes the basis for the VINUM navigation stack. We performed a series of experiments with HyQReal, and our approach achieved a mean of 21.5cm and a standard deviation of 17.6cm of distance from the desired position. Since the kinematic workspace of the Kinova Gen3 arm is larger than the distance between the first and last spur of a single grapevine plant (typically 70cm for a spur-pruned vine), the achieved results are acceptable for this automated task.

The contributions of this proposed autonomous navigation architecture are:

- Higher-level control algorithms for precise robot placement, ideal for a robotic workspace to perform selective, plant-by-plant task automation in the vineyard row.
- Algorithms for accurately generating navigation waypoints from RGB-D sensors during runtime, while navigating down the vineyard row.
- A hand-annotated instance segmentation dataset of grapevine trunks, used for the training of our Mask-RCNN.

II. STATE OF THE ART

The development of autonomous robots and vehicles that can navigate vineyards has significantly advanced in the field [6]. While many of these robots are wheeled and move continuously throughout the vineyard row, some non-continuous models require manually added waypoints. Currently, systems can generate waypoints for autonomous navigation, but these typically rely on initial aerial maps of the vineyard, which limits their effectiveness in adapting to changing conditions at runtime. The EU Project BACCHUS robot is a wheeled vehicle under development to harvest grapes and care for

vineyards. The BACCHUS robot uses semantic segmentation of vineyard trunks for accurate robot motion estimation and consistent metric maps [7]. Our proposed navigation architecture takes the same segmentation approach, but it is used to identify discrete positions for the robot to walk to instead. Previous vineyard navigation has described moving down each row, using a laser sensor, until there are no more grapevines in a row [8]. Other navigation stacks also move autonomously down vineyard rows, using laser scanners for perception [9][10]. The EU Project CANOPIES aims to develop a human-robot collaborative paradigm for harvesting and pruning in vineyards¹. It is a wheeled robot directed by human workers to perform precision agriculture tasks. A similar autonomous over-the-vineyard row robot is the ViTiBOT Bakus which is used to improve vineyard help by removing herbicides and using precision spraying. This solution does not include stopping at each grapevine. YANMAR's autonomous over-the-row robot, YV01, does a similar task that autonomously sprays vineyard rows without stopping at a specific grapevine². Another wheeled robot for precision agriculture is the Agri.q02, which is meant to work in unstructured environments in collaboration with a UAV [11]. A navigation stack was created for the wheeled Ackerman Vehicles in precision farming, path planning from pose to pose [12]. There was autonomous navigation outlined in the Echord++ GRAPE experiment, which maps a vineyard that uses a wheeled robot and moves to locations on the map to perform tasks [13]. Other proposed navigation architectures use deep learning to understand a camera's depth sensors. However, their movement continues down the vineyard row [14]. Other work has been done with deep learning for navigation, where the robot learns to navigate to the end of a vineyard row, but it is also in continuous motion [15]. In the case where a robot would need to stop at specific locations to prune, the navigation of the robot had the locations inserted manually into its path [16]. Other robots are built for the winter pruning of grapevines but do not have autonomous navigation to reach the grapevines themselves [17]. These autonomous robots are all-wheeled; most do not have to stop at precise locations in the vineyard, and those that stop at exact locations have manually inserted those locations ahead of time. Work has been done in generating waypoints for vineyards for navigation purposes. However, these works take an initial aerial overview and derive a list of poses from the initial map [18][19][20].

Our approach generates waypoints in real time as the robot moves throughout the row without requiring initial drone mapping. This paper's proposed quadruped navigation architecture introduces a way to automate waypoint generation at runtime for navigating to every plant in the vineyard for precision agricultural tasks.

III. NAVIGATION ARCHITECTURE

The navigation architecture is a combination of higher-level control and grapevine detection. The higher level control will

¹www.canopies-project.eu

²<https://www.yanmar.com/eu/campaign/2021/10/vineyard/>

decide on the robot's movement path through a vineyard row based on the waypoints generated from the grapevine trunks detected. The grapevine detection is an instance segmentation of grapevine trunks for HyQReal to approach for pruning. The instance segmentation was done by training a Mask-RCNN from Detectron2 [21].

A. Higher Level Control

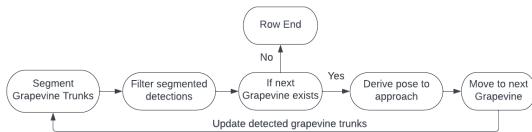


Fig. 3: Navigation Flow.

The higher level control is a state machine for the robot to move throughout a vineyard row, as illustrated in Fig. 3. The process begins with an initial search to find the starting lines for both sides of the row. The initial detections get sent through a filter to find each detection's rolling averages. From the filtered detection points, the control will see the lines on which the vineyard rows begin. After determining the vector of the row, the waypoints are generated to approach.

During the initial search, the Mask-RCNN model classifies random background objects as grapevines and gets multiple detections of the same grapevine. Each new grapevine detection is filtered to sort through the noisy detections. Every time a grapevine trunk was detected, a ROS subscriber would listen to the detection and determine if the identified grapevine was newly detected or detection of a grapevine had already been found. These detections are processed in the Algorithm 1.

The filter performs clustering on a list of "grapevine poses". The clustering is done based on the x and y positions of the poses. If any new grapevine trunk detection was within a set radius of a previously detected trunk, the new trunk detection was combined with the established trunk by taking a rolling average of the detections. The calculations are seen in the (1), where $p_{1x/y}$ is the existing grapevine detection's respective x and y coordinate, $p_{2x/y}$ is the new grapevine detection's x and y coordinates and a is the number of times p_1 has been averaged to that point. The more detections a given point has, the more reliable the point's location.

$$p_{1x} = \frac{p_{1x}a + p_{2x}}{a+1} \quad p_{1y} = \frac{p_{1y}a + p_{2y}}{a+1} \quad (1)$$

After determining the grapevines, the robot has to approach in parallel to the grapevines to prune correctly. To find the correct destination point, initially, the robot determines the orientation of the approach by calculating the vector of the vineyards in a row. This is derived from a list of points found in the initial search. Each possible row line could be found using the determinant of the grapevine points detected from the list of points, as shown in (2). If the determinant is 0, the list of points is a line. In the real-world trials, however, the determinant of the lines found was never 0. ϵ was used to set

Algorithm 1 Cluster Detections

```

Input: Grapevine Pose  $gp$ 
Output: Poses of clustered grapevine trunks  $cgt$ 
averaged Grapevine poses  $agp$  tupled with times averaged
 $ta$   $(agp, ta) \leftarrow empty$ 
if  $agp = 0$  then
     $agp \leftarrow gp$ 
     $ta \leftarrow 1$ 
else
    for every pose tuple  $pt$  in  $agp$  do
        get x and y bottom and top around the grapevine's
        location
         $xb, xt, yb, yt$ 
        if  $xb \leq tpx \leq xt \& yx \leq tpy \leq yt$  then
            pose tuple x  $ptx = (ptx * ta + gpx) / (ta + 1.0)$ 
            pose tuple y  $pty = (pty * ta + gpy) / (ta + 1.0)$ 
             $ta + 1$ 
        end if
    end for
    if  $gp$  is not in  $atp$  then
         $agp \leftarrow gp$ 
    end if
     $cgt \leftarrow$  all grapevine poses  $gp$  in  $agp$  where  $gp_i$ 's  $ga_i >$ 
    2 times
end if
  
```

the determinant size of a line to allow for more leniency in finding lines.

$$\det \begin{bmatrix} X_i & X_j \\ Y_i & Y_j \end{bmatrix} \leq \epsilon \quad (2)$$

The initial vector is determined by taking the closest grapevine pose found gp , taking the subset of the found lines which only have the ones with the closest pose A , then determining the subset of A where gp is the first pose B in the found line. Then take the line with the shortest distance from B . The assumption is that after the initial search, a vineyard row found would be a shorter line than the line found between two rows. See Fig. 4.

Let G denote the set of all grapevine trunk poses. The subset of lines found with the closest trunk pose to gp is denoted by $A = L_i | L_i \subseteq G, L_i$ has the closest pose to gp . The subset of lines in A , where gp is the first pose, is denoted by $B = L_j | L_j \in A, gp =$ first pose in L_j . The line with the shortest distance from B is denoted by $C = \min_{L_j \in B}$ distance between L_j and B . C determines the initial vector.

After the initial row vector for the robot is determined, the robot must approach at a parallel offset at a desired distance d to maximize the pruning arm's robotic workspace. A new transformation is created with the grapevine trunk at the center to ensure a parallel approach. This transformation is calculated using the angle of the row's vector to the world frame. The Euler angles are converted into a quaternion, and the grapevine



Fig. 4: Segmented Grapevine trunk image showing three detected grapevine trunks which are part of the initial line.

trunk's x, y world positions are set to the center of the trunk frame. Finally, the destination pose is determined by adding an offset of d in the lateral direction of the grapevine.

The higher level control updates the vector for possible deviances of grapevines as the robot moves along the row. The robot then approaches the grapevines in parallel at a desired distance that depends on the robot's size and the arm's workspace.

After the robot has reached the determined location in the vineyard, the control stack removes that grapevine from the list of vines to approach. Next, the control will choose the closest grapevine to the robot as its next target. This method is continued until no more grapevines are identified in a row.

B. Grapevine Identification

Instance segmentation is a computer vision task that involves identifying and segmenting individual objects in an image or video. In the case mentioned, a Mask R-CNN is used, for instance, segmentation, to detect grapevine trunks in a vineyard. The Mask R-CNN is a two-stage deep learning model that generates region proposals and classifies and segments the regions of interest.



Fig. 5: Result of the image segmentation to detect grapevine trunks. (4 examples).

The training of the Mask R-CNN was done using Detectron2, an open-source platform for object detection research. 500+ hand-annotated images of potted grapevines were created and used for this training¹. These annotated images helped the

¹https://universe.roboflow.com/vinum/potted_grapevines/dataset/2

neural network to learn the characteristics of grapevine trunks and improved its ability to detect them in new, unseen images.

An aligned depth image was used to determine the depth of the detection; the depth image provided the necessary information to calculate the distance between the grapevine trunks and HyQReal. This information was used to find the locations of the grapevine trunks relative to the quadruped. However, depth readings were not always reliable, so to account for depth errors from the sensor, we created an algorithm to derive the closest possible valid depth value, seen in Algorithm 2.

Algorithm 2 takes in two inputs: x and y , and computes the corresponding 3D coordinates ($realX, realY, realZ$) of a point in a given depth image $depth$. The algorithm first sets a variable $nPixels$ to 2 and computes the depth image's height h and width w . It then enters a loop, where it gradually expands a square around the point of interest, defined by the input x and y , until the mean depth value in that square is found, stored in the variable $realZ$. The loop continues until $realZ$ is no longer NaN or the square has reached the boundaries of the depth image.

Once $realZ$ is found, the function computes the corresponding 3D coordinates using the intrinsic parameters of the camera, stored in the $camInfoMsg.K$ matrix and returns the 3D coordinates as $realX, realY, realZ$.

Algorithm 2 Reliable Depth Value

Input: 2D image coordinates (x, y) , depth image $depth$, camera information message $camInfoMsg$
Output: 3D real world coordinates $(realX, realY, realZ)$

```

nPixels ← 2
h, w ← depth.shape
realZ ← NaN
i ← 1
minY, maxY, minX, maxX ← y, y, x, x
while realZ is NaN and not ((minY == 0) and (minX == 0) and (maxY == h - 1) and (maxX == w - 1)) do
    minY ← clip(y - i * nPixels, 0, h - 1.0)
    maxY ← clip(y + i * nPixels, 0, h - 1.0)
    minX ← clip(x - i * nPixels, 0, w - 1.0)
    maxX ← clip(x + i * nPixels, 0, w - 1.0)
    subImg ← depth[minY : maxY + 1.0, minX : maxX + 1.0]
    realZ ← nanmean(asarray(subImg))
    i ← i + 1
end while
cx, cy, fx, fy ← camInfoMsg.K
realX ← (x - cx) * realZ / fx
realY ← (y - cy) * realZ / fy
return (realX, realY, realZ)

```

A Mask-RCNN previously trained in Detectron2 framework was used [22]. The results of the trained network can be seen in Fig. 5.

IV. EXPERIMENTS

These experiments aim to determine the precision of the waypoints generated for the quadruped robot to move to. They aim to align the robot's geometric center with the grapevine trunk; this way, an arm mounted on the front of HyQReal is in the center of the grapevine's cordon and thus optimizes the workspace of the arm for single-plant operations. Two



Fig. 6: Experimental setup in the lab with the quadruped robot Aliengo.

rounds of testing were conducted; initially, the higher level control was tested on Aliengo, then the architecture was moved onto HyQReal. The higher level control was tested in a lab using Unitree's Aliengo robot. Aliengo was used to simplify experiments since it is 21kg and 61cm long, as compared to the 140kg and 1.3m length of HyQReal. Aliengo is equipped with Intel's Realsense D435 RGB-D camera. Red balls for segmenting were used to test in the lab instead of the grapevine trunks; see Fig. 7. The red balls are spaced out at about 80cm from each other, the approximate distance that grapevines are from each other. The setup of the experiment can be seen in Fig. 6. How the precision of the robot approaching a position was measured shown in Fig. 7. The tests were

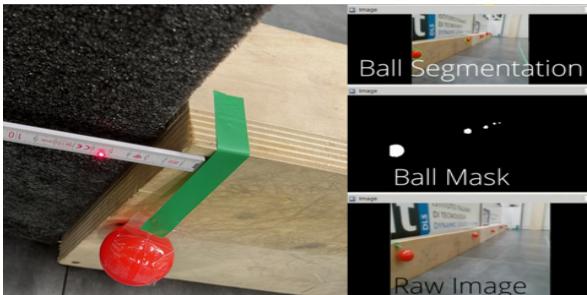


Fig. 7: Measurement of Aliengo's arrival at a position using a laser pointer mounted onto the robot and the Image Segmentation of Red Balls.

then repeated with HyQReal in the lab, equipped with the ZED depth camera. HyQReal was also tested in the vineyard at the University Cattolica of Piacenza with potted vines. During the tests, the robot initially searches the area using its RGB-D camera to segment the red balls. After the robot finds the row of red balls, it approaches the first position in the row. After the robot arrives at the initial pose, it pauses for an automated task and updates its detections. This process repeats until the row is finished and then stops. With Aliengo, ten trials were conducted with five balls. A laser pointer mounted on the robot's torso was used to show the point that Aliengo's center of mass reached to measure the error between the achieved destination point and the red ball.



Fig. 8: Piacenza experiment setup with HyQReal.

With HyQReal, five trials were conducted with three red balls; see Fig. 8. The error was measured from the base of the robotic arm mount on HyQReal to the center of the grapevine's cordon; see Fig. 9. Aliengo's error of reaching the destination point has a mean of 3.4cm and a standard deviation of 2.2cm. HyQReal's error of reaching the destination point has a mean of 21.5cm and a standard deviation of 17.6cm. The

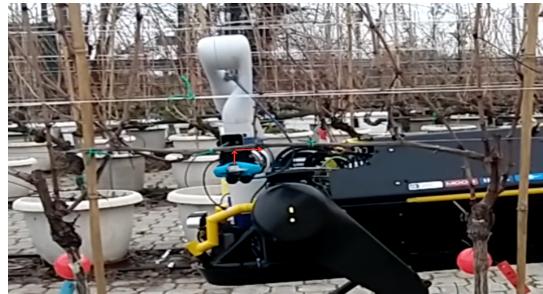


Fig. 9: Measurement of the shear to the center of the cordon.

tests showed a reliable higher-level control with repeatable navigation. However, the results are significantly predicated on an accurate initial search. The grapevine segmentation was initially tested in Piacenza. However, the model trained had difficulty generalizing different points of view needed for identifying the grapevines, so we used red balls in the interim to test the navigation architecture. In cases where HyQReal did not accurately identify a grapevine, it would skip over. In tests in Piacenza, there was fog one morning, and the ZED camera could not make reliable detections, resulting in skipped grapevines. However, when the grapevine search went well, the robot reliably approached each grapevine.

V. CONCLUSION

This paper presented a method of computer-vision-based navigation in vineyards for quadruped robots. This method will allow for precise waypoint generation to perform selective task automation. Our approach with HyQReal achieved a mean of 21.5cm and a standard deviation of 17.6cm of distance from the desired position, which is sufficient for an automated task in a vineyard.

The system's architecture works reliably, generating and arriving at precise waypoints that maximize the attached robotic

arm's workspace. However, there could be more accuracy in grapevine detection as the RGB-D camera's initial search needs to be trained off more data for more robust detections.

The future steps for this architecture are to improve the dataset created to train a more robust Mask-RCNN and then the full integration of the navigation combined with the pruning arm for VINUM autonomous winter pruning of grapevines.

VI. ACKNOWLEDGMENTS

Thanks to the contributions of Professors Matteo Gatti and Stefano Poni from the Universita Cattolica del Sacro Cuore for their important inputs regarding the experiments. Also, thanks to Chundri Boelens for helping set up HyQReal and Lorenzo Amatucci for the configuration of the robot's controller.

REFERENCES

- [1] L. Milburn, J. Gamba, and C. Semini. "Towards Computer-Vision Based Vineyard Navigation for Quadruped Robots". In: *Istituto di Robotica e Macchine Intelligenti (I-RIM)* (2022). DOI: [10.48550/ARXIV.2301.00887](https://doi.org/10.48550/ARXIV.2301.00887).
- [2] M. Fernandes, A. Scaldaferri, G. Fiameni, T. Teng, M. Gatti, S. Poni, C. Semini, D. Caldwell, and F. Chen. "Grapevine Winter Pruning Automation: On Potential Pruning Points Detection through 2D Plant Modeling using Grapevine Segmentation". In: *2021 IEEE 11th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*. 2021, pp. 13–18. DOI: [10.1109/CYBER53097.2021.9588303](https://doi.org/10.1109/CYBER53097.2021.9588303).
- [3] P. Guadagna, T. Frioni, F. Chen, A. I. Delmonte, T. Teng, M. Fernandes, A. Scaldaferri, C. Semini, S. Poni, and M. Gatti. "16. Fine-tuning and testing of a deep learning algorithm for pruning regions detection in spur-pruned grapevines". In: *Precision agriculture '21*. Wageningen Academic Publishers, June 2021. DOI: [10.3920/978-90-8686-916-9_16](https://doi.org/10.3920/978-90-8686-916-9_16).
- [4] C. Semini, V. Barasuol, M. Focchi, C. Boelens, M. Emara, S. Casella, O. Villarreal, R. Orsolino, G. Fink, S. Fahmi, et al. "Brief introduction to the quadruped robot HyQReal". In: *Istituto di Robotica e Macchine Intelligenti (I-RIM)* (2019).
- [5] P. Biswal and P. K. Mohanty. "Development of quadruped walking robots: A review". In: *Ain Shams Engineering Journal* 12.2 (2021), pp. 2017–2031. ISSN: 2090-4479. DOI: <https://doi.org/10.1016/j.asej.2020.11.005>.
- [6] I. Hroob, R. Polvara, S. M. Mellado, G. Cielniak, and M. Hanheide. "Benchmark of visual and 3D lidar SLAM systems in simulation environment for vineyards". In: *CoRR* abs/2107.05283 (2021). arXiv: [2107.05283](https://arxiv.org/abs/2107.05283).
- [7] A. Papadimitriou, I. Kleitsiotis, I. Kostavelis, I. Mariolis, D. Giakoumis, S. Likothanassis, and D. Tzovaras. "Loop Closure Detection and SLAM in Vineyards with Deep Semantic Cues". In: *2022 International Conference on Robotics and Automation (ICRA)*. 2022, pp. 2251–2258. DOI: [10.1109/ICRA46639.2022.9812419](https://doi.org/10.1109/ICRA46639.2022.9812419).
- [8] G. Riggio, C. Fantuzzi, and C. Secchi. "A Low-Cost Navigation Strategy for Yield Estimation in Vineyards". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 2200–2205. DOI: [10.1109/ICRA.2018.8462839](https://doi.org/10.1109/ICRA.2018.8462839).
- [9] M. Bergerman, S. M. Maeta, J. Zhang, G. M. Freitas, B. Hammer, S. Singh, and G. Kantor. "Robot Farmers: Autonomous Orchard Vehicles Help Tree Fruit Production". In: *IEEE Robotics & Automation Magazine* 22.1 (2015), pp. 54–63. DOI: [10.1109/MRA.2014.2369292](https://doi.org/10.1109/MRA.2014.2369292).
- [10] H. Nehme, C. Aubry, T. Solatges, X. Savatier, R. Rossi, and R. Boutteau. "LiDAR-based Structure Tracking for Agricultural Robots: Application to Autonomous Navigation in Vineyards". In: *Journal of Intelligent Robotic Systems* 103.4 (Nov. 2021). DOI: [10.1007/s10846-021-01519-7](https://doi.org/10.1007/s10846-021-01519-7).
- [11] G. Quaglia, C. Visconte, L. S. Scimmi, M. Melchiorre, P. Cavallone, and S. Pastorelli. "Design of a UGV Powered by Solar Energy for Precision Agriculture". In: *Robotics* 9.1 (2020). ISSN: 2218-6581. DOI: [10.3390/robotics9010013](https://doi.org/10.3390/robotics9010013).
- [12] R. F. Carpio, C. Potena, J. Maiolini, G. Ulivi, N. B. Rosselló, E. Garone, and A. Gasparri. "A Navigation Architecture for Ackermann Vehicles in Precision Farming". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 1103–1110. DOI: [10.1109/LRA.2020.2967306](https://doi.org/10.1109/LRA.2020.2967306).
- [13] P. Astolfi, A. Gabrielli, L. Bascetta, and M. Matteucci. "Vineyard Autonomous Navigation in the Echord++ GRAPE Experiment (FP7-601116). <http://echord.eu/grape/>". In: *IFAC-PapersOnLine* 51.11 (2018). 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018, pp. 704–709. ISSN: 2405-8963. DOI: <https://doi.org/10.1016/j.ifacol.2018.08.401>.
- [14] D. Aghi, V. Mazzia, and M. Chiaberge. "Local Motion Planner for Autonomous Navigation in Vineyards with a RGB-D Camera-Based Algorithm and Deep Learning Synergy". In: *Machines* 8.2 (2020). ISSN: 2075-1702. DOI: [10.3390/machines8020027](https://doi.org/10.3390/machines8020027).
- [15] M. Martini, S. Cerrato, F. Salvetti, S. Angarano, and M. Chiaberge. "Position-Agnostic Autonomous Navigation in Vineyards with Deep Reinforcement Learning". In: *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*. IEEE, Aug. 2022. DOI: [10.1109/case49997.2022.9926582](https://doi.org/10.1109/case49997.2022.9926582).
- [16] A. Silwal, F. Yandun, A. Nellithimaru, T. Bates, and G. Kantor. "Bumblebee: A Path Towards Fully Autonomous Robotic Vine Pruning". In: *Field Robotics* 2.1 (Mar. 2022), pp. 1661–1696. DOI: [10.55417/fr.2022051](https://doi.org/10.55417/fr.2022051).
- [17] T. Botterill, S. Paulin, R. Green, S. Williams, J. Lin, V. Saxton, S. Mills, X. Chen, and S. Corbett-Davies. "A Robot System for Pruning Grape Vines". In: *Journal of Field Robotics* 34.6 (2017), pp. 1100–1122. DOI: <https://doi.org/10.1002/rob.21680>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21680>.
- [18] V. Mazzia, F. Salvetti, D. Aghi, and M. Chiaberge. "Deep-Way: A Deep Learning waypoint estimator for global path generation". In: *Computers and Electronics in Agriculture* 184 (2021), p. 106091. ISSN: 0168-1699. DOI: <https://doi.org/10.1016/j.compag.2021.106091>.
- [19] F. Salvetti, S. Angarano, M. Martini, S. Cerrato, and M. Chiaberge. *Waypoint Generation in Row-based Crops with Deep Learning and Contrastive Clustering*. 2022. DOI: [10.48550/ARXIV.2206.11623](https://doi.org/10.48550/ARXIV.2206.11623).
- [20] L. C. Santos, F. N. Santos, A. S. Aguiar, A. Valente, and P. Costa. "Path Planning with Hybrid Maps for processing and memory usage optimisation". In: *2022 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. 2022, pp. 27–33. DOI: [10.1109/ICARSC55462.2022.9784767](https://doi.org/10.1109/ICARSC55462.2022.9784767).
- [21] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. *Detectron2*. <https://github.com/facebookresearch/detectron2>, 2019.
- [22] M. Fernandes, A. Scaldaferri, P. Guadagna, G. Fiameni, T. Teng, M. Gatti, S. Poni, C. Semini, D. G. Caldwell, and F. Chen. "Towards Precise Pruning Points Detection using Semantic-Instance-Aware Plant Models for Grapevine Winter Pruning Automation". In: *CoRR* abs/2109.07247 (2021). arXiv: [2109.07247](https://arxiv.org/abs/2109.07247).