

An Autonomous Sucker Management Architecture for Large-scale Hazelnut Orchards

Renzo Fabrizio Carpio*
Roma Tre University
Rome, Italy

Martina Lippi*
Roma Tre University
Rome, Italy

Jacopo Maiolini*
Roma Tre University
Rome, Italy

Matteo Santilli*
Roma Tre University
Rome, Italy

Emanuele Garone
Université Libre de Bruxelles
Bruxelles, Belgium

Valerio Cristofori
Università degli Studi della Tuscia
Viterbo, Italy

Andrea Gasparri
Roma Tre University
Rome, Italy*

Abstract

In this work, inspired by the needs of the H2020 European project PANTHEON, we address the problem of managing hazelnut suckering plants on a per-plant basis in a large-scale orchard. Sucker management is an essential practice in agriculture given the fact that suckering plants, i.e., basal shoots that grow at the base of a tree, compete with the tree itself for nutrients and water. Generally, in large-scale orchards, suckering plants, or shortly, suckers, are treated with the application of herbicide through spraying tractors that continuously spray the crops while navigating the whole orchard. This approach however does not consider the individual needs of each plant and it is definitely not environmentally-friendly since a lot of unnecessary solution is being drained in the soil. For this reason, we propose an autonomous sucker management architecture that is able to detect the presence of the suckers for each plant and then individually treat them according to the real plants' needs, significantly reducing pollution and waste. Experimental results in a real-world (1:1 scale) hazelnut orchard located in Caprarola, a municipality in the province of Viterbo (Italy), are provided to corroborate the proposed architecture.

Keywords— Agricultural Robotics, Sucker Management, Precision Farming

1 Introduction

The recent paradigm of Precision Agriculture (PA) introduces a transformation in the way large-scale crops are managed: from a very coarse resolution, where only a few samples of plants in the field are analyzed and their findings are extended to the surrounding area, to a very detailed resolution, where a plant-by-plant analysis is carried out. In particular, PA envisions a continuous monitoring of the health state of each plant and a prompt response accordingly. This implies that an effective realization of the PA paradigm in large-scale fields is possible only automatizing most of the required monitoring operations and agronomic interventions.

Motivated by the above and driven by the needs of the European H2020 Project PANTHEON, aimed at developing a SCADA system for large-scale hazelnut orchards, we focus on developing an autonomous sucker management architecture for large-scale orchards. More in detail, as shown in Figure 1, a suckering plant (or sucker) is a basal

*Authors are listed in alphabetical order and all contributed equally. M. Lippi is corresponding author: martina.lippi@uniroma3.it



Figure 1: Overview of the proposed autonomous sucker management architecture: each sucker is detected and reconstructed, its surface area is computed and, based on this, the needed amount of herbicide is estimated and applied.

shoot that grows at the base of a tree and competes with the tree itself for nutrients and water. Furthermore, suckers represent an obstacle to mechanized operations, mainly during the harvesting of nuts. Hence, sucker management is an essential practice in agriculture, especially in Europe, where the European hazelnut, as the one involved in PANTHEON, is a species characterized by a high sucker emission aptitude. In small orchards, sucker management is generally performed by hand, using pruning shears. This enables the farmer to select a limited number of suckers to be left unpruned, to rejuvenate the plant. While this approach is environmentally friendly, it has the drawback of being labour intensive and expensive, as reported in (Tomasone et al., 2009; Serdar and Akyuz, 2018). In larger orchards, manual and mechanical control result in an extremely expensive and time-consuming procedure. For these reasons, the most common method for sucker management in large plantation management today remains the chemical control through the application of licensed herbicides. This is mainly carried out manually by field workers, who spray all plants in the orchard. For larger orchards, a tractor with a pump for herbicides is used. In both cases, a non-calibrated amount of herbicide is applied to all plants. In addition, to ensure the effectiveness of chemical applications, suckers must be in the herbaceous stage, in other words their length should be shorter than 30 cm and their stems must not be woody yet, which implies that the suckering control must be carried out frequently.

In this view, we aim to advance the current suckering control strategies by proposing an autonomous framework able to treat each tree according to its individual needs. Our suckering control solution is composed of two steps. In the first step, the characterization and estimation of sucker canopy dimensions of every plant is carried out. In the second step, tailor-made treatments are computed and applied, to allow the application of different rates of herbicide to each tree. This innovative solution is expected to reduce herbicide volumes and enhance plant health conditions.

This paper builds on (Potena et al., 2020) where we presented preliminary results on sucker detection and volume estimation. However, this work extensively advances (Potena et al., 2020) in the following aspects:

1. We improved the detection phase integrating the depth map information;
2. We replaced the sucker-specific volume estimation phase with a canopy area estimation phase, enhancing the generality of the approach and making it suitable for different varieties of plant stands beyond sucker types;
3. We designed and implemented the strategies for the herbicide computation, the sprayer time computation and the sprayer motion;
4. We fully validated the approach on a real hazelnut orchard using the ground robotic platform SHERPA-HL;

5. We reduced the acquisition sensors relying on a RGB-D camera only for sucker detection and reconstruction instead of also requiring a Lidar sensor;

Note that, to the best of our knowledge, this is the first unified solution in literature that provides fully autonomous management of suckers, from detection to possible treatment, through a mobile robotic platform.

The remainder of the paper is structured as follows. First, relevant works for autonomous solutions in PA settings are discussed in Section 2 and the contributions of the paper are stated. Second, the addressed problem is formalized in Section 3 and an overview of the proposed solution is provided. Next, the methodology to estimate the quantity of herbicide to distribute for each tree is presented in Section 4, while the procedure for the subsequent sucker treatment is described in Section 5. In Section 6 the experimental setup is presented while validation results with a SHERPA-HL on a real-world hazelnut orchard are reported in Section 7. Finally, in Section 8 conclusions are drawn and future works are discussed.

2 Related Work

The use of automatized solutions in agricultural settings has attracted great interest in the last decades to improve yield of high-quality fruit with minimal reliance on seasonal human labor as reported in (Zhang et al., 2019). Robotic platforms are a promising direction towards this automatization since they provide flexible in-field actuation capabilities and potentially allow to perform several agricultural operations such as pruning as in (Botterill et al., 2017), weeding as in (Zhang et al., 2017), spraying as in (Sammons et al., 2005), harvesting as in (De-An et al., 2011) and pest management as in (Lippi et al., 2021). In this work, we focus on a spraying agricultural operation for automatic sucker management.

As a first step for most automatized PA applications, a 3D reconstruction of the plant to treat is generally required. In this regard, the authors of (Pound et al., 2014) propose a fully automatic approach to carry out realistic 3D representations of plant shoots by using a single low-cost camera. The architecture of the leaf surfaces is obtained by combining a series of small planar sections and making a few assumptions about the nature of the plant. A depth camera for 3D plant phenotyping is considered instead in (Ch  n   et al., 2012) where depth images are segmented from a single top-view. High-resolution, simultaneous and multi-view digital color images are then used in (Nguyen et al., 2016) for full 3D reconstruction system. Cameras are mounted on a super arc-shaped structure and organized into 16 stereo pairs in four separate arcs. The comparison of structure-from-motion and stereo vision techniques reveals that both can yield satisfactory 3D reconstruction models and are suited for high-throughput phenotyping without destroying any leaves or stems of the plant.

In our application, it is not sufficient to 3D reconstruct the entire plant, but it is also necessary to isolate and 3D reconstruct the portion of interest, that is the respective sucker if present. In this regard, preliminary results exist in literature. In particular, the study in (Kang et al., 2012) proposes a sucker detection system for sucker control in grape vineyards where extra-green image segmentation is applied and width and height of the bounding boxes surrounding the suckers are defined as a rough measure for establishing the herbicide to release. However, a white calibration board is assumed to be placed behind the suckers for segmenting them with respect to the background which is not practical in real settings. Similarly, the presence of a white calibration board is also assumed in (Wang et al., 2017) to identify suckers of grapevine plants. In contrast, we advance the state of the art in this phase by first identifying the suckers using a real-time data driven approach and then we 3D reconstruct them using a real-time meshes registration method, leading to an accurate estimation of the sucker without the need of additional items for simplifying the segmentation.

Once the suckers are localized and reconstructed, the agronomic spraying operation can be carried out. In the following we thus review works focusing on this agronomic operation.

An adjustable device for accurate pesticide spraying is presented in (Berenstein and Edan, 2018). In particular, this device is equipped with a camera and distance sensors allowing to adapt the spraying diameter according to the shape and size of the target. However, the study mainly addresses the sprayer mechanism design and does not cover the target detection algorithms, which we tackle in our work. A smart pesticide spraying system based on machine vision is presented in (Xie et al., 2020) in which holes in the leaves caused by pests are identified and the plant is sprayed accordingly. As in (Berenstein and Edan, 2018), the focus of (Xie et al., 2020) is primarily on the sprayer

mechanical design, while we are interested in building a complete sucker management architecture covering all the necessary steps from detection to spraying.

In this regard, the authors of (Xiao et al., 2017) present a more comprehensive framework including an extraction algorithm as well as the spraying phase. More in detail, the algorithm is developed using a Microsoft’s Kinect to estimate the size and the density of the Leaf Wall Area (LWA), i.e., an indicator of the size of the orchard canopy in relation to the spacing of the rows and the altitude of the plants. Based on the estimated LWA density, the authors roughly suggest the application of a corresponding dose of pesticides, i.e., high dose when the density is high and low dose if the density is low. A custom-made mobile robot for spraying pesticides in greenhouses is then developed in (Hejazipoor et al., 2021). Specifically, the robot is equipped with a Kinect camera mounted on a vertical rail system which collects images at different altitudes for each plant. Based on a rough distance-based segmentation made by the Kinect camera, the authors design an automatic algorithm to estimate the volume of the plant. Next, the amount of pesticides is computed and sprayed with a nozzle located at the top of the robot. Contrarily to (Hejazipoor et al., 2021), in our approach we adopted a tailored-made segmentation system which enables a better estimation of the solution to be sprayed to the plants.

Finally, further research topics on spraying robots, that go beyond the scope of this work, involve, for instance, *i*) path planning aspects, such as in (Nakao et al., 2017), where graph theory is exploited to find shortest paths, and in (Danton et al., 2020), where LiDAR measures are used to detect edges, representing the rows of plantation, and a robot motion strategy is designed to follow them during spraying; *ii*) autonomous navigation and localization, such as in (Cantelli et al., 2019), where a classical pump spraying system is mounted on a robotic prototype and used to apply pesticides uniformly across the orchard, independently from the plants’ needs; *iii*) multi-robot coordination, such as in (Kim and Son, 2020), where a task allocation method based on a Voronoi diagram is devised; and *iv*) inverse kinematics algorithms, such as in (Wang et al., 2021) where an algorithm combining analytical and numerical methods for a redundant spraying robot is described.

In summary, we propose a *comprehensive* autonomous sucker management architecture for which the main contributions are:

1. Design of a real-time data-driven detection system based on RGB image and respective depth map which is proved to outperform the RGB-only detection system;
2. Development of an integrated segmentation and reconstruction system which generates 3D meshes of the suckers and provides an estimation of their canopy area;
3. Design of an herbicide estimation method which exploits the estimated canopy area and prior knowledge gained from agronomical experts to provide an accurate per-plant treatment;
4. Design of an autonomous spraying and aiming system for the application of the needed herbicide to the plants;
5. In-field collection of an RGB-D dataset for automatic sucker detection, released publicly at the link <https://tinyurl.com/sucker-dataset>, and datasets for the parameters fitting used in herbicide and spraying time estimations;
6. A full in-field validation of the approach in a hazelnut orchard using a ground robotic platform equipped with RGB-D camera.

We reiterate that, as stated in the Introduction, to the best of our knowledge this work represents the first fully autonomous solution for handling suckers in large scale fields.

3 Problem Setting

In this section we introduce the problem setting which defines our proposed autonomous sucker management architecture for large-scale hazelnut orchard.

Consider a $r \times c$ lattice graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with node set $\mathcal{V} = \{v_1, \dots, v_{rc}\}$ representing trees in an orchard and edge set $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ linking two nodes $i, j \in \mathcal{V}$ if the trees i, j are adjacent. In addition, consider a $(r + 1) \times (c + 1)$ square grid graph $\mathcal{G}^o = \{\mathcal{V}^o, \mathcal{E}^o\}$ with node set $\mathcal{V}^o = \{v_1^o, \dots, v_{(r+1)(c+1)}^o\}$ encoding the set of all possible locations on which

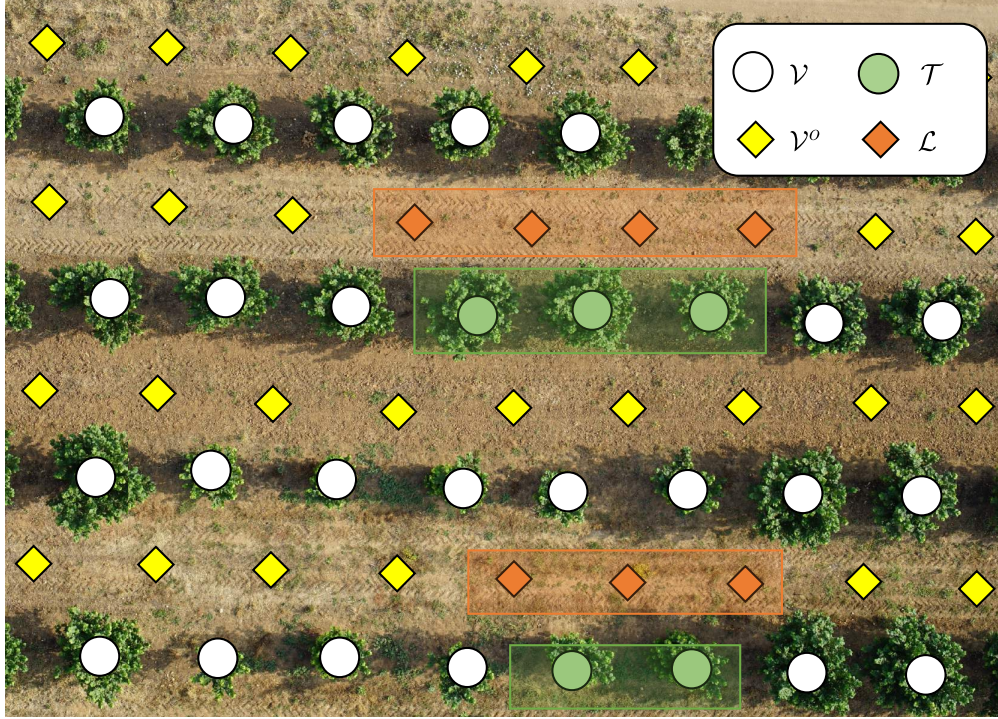


Figure 2: Depiction of the field graph nodes \mathcal{V} (white circles) and respective locations \mathcal{V}^o where operations can be performed (yellow diamonds) as well as the set \mathcal{T} of trees to be treated (green circles) and the respective locations \mathcal{L} (orange diamonds).

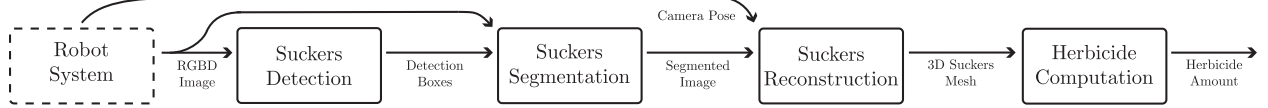


Figure 3: Overview of the proposed autonomous architecture for per-plant herbicide computation. Inputs and outputs of each block are reported.

an agronomic operation can be performed and edge set \mathcal{E}^o encoding the existence of a physical passage between a pair of locations v_i^o and v_j^o . Clearly, the nodes in the sets $\{\mathcal{V}, \mathcal{V}^o\}$ are associated with a spatial position corresponding to their position in the orchard. Furthermore, a spatial relationship exists between the two of them: an agronomical operation can be computed on the tree $v_i \in \mathcal{V}$ only in the closest four possible locations $v_j^o \in \mathcal{V}^o$. Figure 2 shows the square grid structure of the orchard and the relationship between the two grid graphs. The trees (\mathcal{V}) have been represented with white circles whereas location (\mathcal{V}^o) on which agronomical operations can be performed have been represented with yellow diamonds.

Consider also a ground robot \mathcal{R} provided with a navigation architecture able to guarantee safe autonomous motion inside the orchard and equipped with an RGB-D camera and tools for sucker treatment, i.e., a tank filled with a solution of water and herbicide and a sprayer mechanism.

Problem 1 Consider a set of target trees $\mathcal{T} \subseteq \mathcal{V}$ and a set of locations $\mathcal{L} \subseteq \mathcal{V}^o$ from which the robot \mathcal{R} can see and treat the suckers of the trees in \mathcal{T} . Our objective consists in designing an autonomous sucker management architecture which is able to detect the presence of the suckers, estimate the amount of herbicide solution, and apply it to the detected suckers.

Figure 2 depicts the target trees in green and respective locations with orange diamonds. Our approach to solve Problem 1 can be decomposed in three steps: *i*) detect the presence of suckers at the target trees \mathcal{T} from the locations \mathcal{L} using a real-time object detection system; *ii*) exploit the depth map of the RGB-D camera mounted on

the robot to virtually reconstruct the detected suckers and, based on the reconstruction, estimate the canopy surface of the suckers which can then be used to compute the amount of herbicide solution required to treat the suckers; *iii*) aim the suckers with the sprayer and activate the atomizer for a time window long enough to apply the computed amount of herbicide solution.

We point out that in this work we focus on the development of the management architecture and leave out the selection of the subset of locations \mathcal{L} from which the trees can be analyzed and treated. This is motivated by the fact that these two problems are de-facto decoupled and for this reason we leave the autonomous selection of the best location for future work. In this regard, a possible approach would be to follow the framework we proposed in (Furchi et al., 2022), where a Multi-Platform Steiner Traveling Salesman Problem (MP-STSP) is designed for selecting the best routes for a team of robots that need to perform agronomical activities in an orchard.

4 Herbicide Estimation

In this section we describe the processing pipeline that allows the definition of the amount of herbicide to provide to each sucker in the area of interest given visual information collected by the robot. Figure 3 depicts the block diagram of the software processing pipeline where each component is shown along with their inputs and outputs. The pipeline is supposed to be activated once that the robot is on the field near the area of interest. In particular, as soon as the robot reaches the first location of the set \mathcal{L} , the following operations are performed. First, the RGBD camera mounted on the robot is activated and its output, i.e., the RGB color image and the depth map, are fed into a neural network that detects the presence of the suckers. Next, this information is used to segment both the RGB color image and the depth map and is fed into another component which reconstructs a 3D mesh model of the segmented environment. When the robot has visited all the locations \mathcal{L} , a filtering procedure is applied to the reconstructed environment to properly extract and isolate the suckers from the 3D mesh. Multiple 3D meshes are generated (exactly one per sucker, i.e., per tree) and an estimation of the canopy area for each of them is computed. Finally, the estimated area is used to compute the amount of herbicide that is required for the sucker treatment.

We will now describe in detail the components of our pipeline. Note that this has been fully integrated within the ROS framework as detailed in Section 6.2. Moreover, note that in the following by herbicide we intend the solution consisting of water and herbicide as typical in agronomic applications and as reported in detail in Section 6.2.

4.1 Suckers Detection

The first element in charge of processing the RGB color and depth data received from the camera is a data-driven detection system. We resort to YOLO (You Only Look Once) framework which enables global regression in a single iteration, providing *real-time* processing capabilities that are crucial for in-field applications. In particular, given an input image, a YOLO-based model provides in a single step center, height and width of the bounding boxes enclosing the objects of interest along with the respective class probabilities, also referred to as confidence scores. An example is provided in Figure 4a, where the presence of two suckers is detected.

Among the available versions for YOLO architecture, we consider YOLOv4 defined in (Bochkovskiy et al., 2020) that outperforms the previous ones in terms of accuracy and computational burden on different benchmark datasets, e.g., YOLOv3 in (Redmon and Farhadi, 2018) and YOLOv2 in (Redmon and Farhadi, 2017). Briefly, YOLOv4 is made up of three main components: *i*) a backbone layer that produces feature maps encoding relevant features from image pixels, *ii*) a neck layer that collects feature maps at different scales, improving the capability of detecting different sizes of the objects of interest, *iii*) a head layer that predicts bounding boxes and respective class probabilities. We use a CNN-based Cross Stage Partial architecture comprising ResUnit components for the backbone, a spatial pyramid pooling layer, fusing features at different scales and amplifying the receptive field, followed by a Path Aggregation Network for the neck, and an anchor-based YOLO layer for the head. Further details on YOLOv4 can be found in (Bochkovskiy et al., 2020).

In contrast to the standard YOLOv4 architecture, besides the RGB components, we additionally provide the depth map as input for the backbone layer. This choice is motivated by the fact that the depth information may potentially allow to capture more effectively the sucker structure and improve the robustness of the system with respect to visually similar objects/vegetation in the scene. Moreover, we apply a normalization preprocessing step to the depth

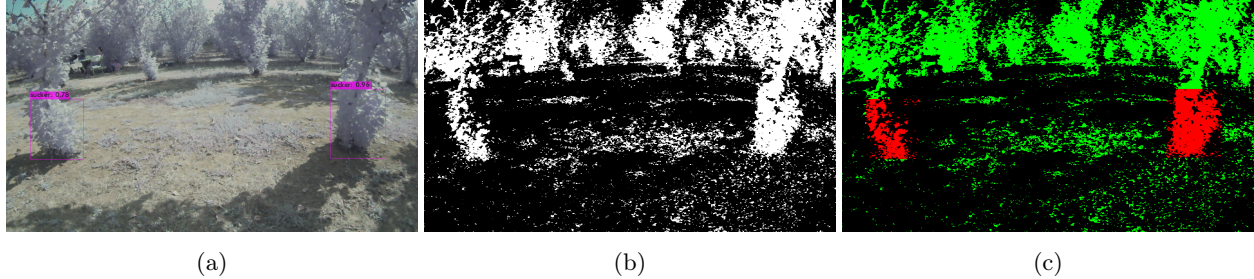


Figure 4: (a) Detection of two suckers in the orchard. (b) Application of the ExG index (1) thresholding operation to the input RGB image. (c) Segmentation of the input RGB image: the canopies of the trees are highlighted in green, whereas the suckers’ canopies are highlighted in red.

map to enhance its contrast, as shown in the experimental results in Section 7.1.

Therefore, to train a YOLO model, we provide RGB images and depth maps along with ground truth bounding boxes surrounding suckers in the field. Then, the model is trained to minimize the complete Intersection over Union (IoU) loss (Zheng et al., 2020), where the IoU represents the ratio of the overlapping area between ground truth and predicted bounding boxes to the union area between the same boxes. Note that we consider a single-class detection problem, i.e., only suckers are detected, however, YOLO framework can generally handle multiple classes, which we will investigate for future extensions. For instance, a two-class detection problem could be set up in order to distinguish between suckers to be treated and those already treated with herbicide. However in a SCADA architecture, as the one in PANTHEON project, it is reasonable to assume that the central database maintains the information of previously treated and untreated suckers.

As briefly introduced above, the collected RGB and depth data along with the information about the sucker detection are then used to create a segmented image of the suckers, whose process is the subject of the next section.

4.2 Suckers Segmentation

Image segmentation is a practice used in computer vision to partition the image in different segments representing different objects. In our case, image segmentation allows to determine which *pixels* are associated with suckers, i.e., to remove background and/or irrelevant objects present in the bounding boxes predicted by the YOLO system. To perform this operation, we exploit the fact that suckers are parts of the tree canopies and we split the segmentation in two phases: first, the portions of the canopies that are captured in the image are segmented and, then, the respective pixels are defined to belong to a sucker or not based on the predicted bounding boxes.

More in detail, to segment the canopy, we compute the following Excess Green (*ExG*) index defined in (Woebbecke et al., 1995), as common in agronomic image analysis as in (Guijarro et al., 2011),

$$ExG(I) = 2G - R - B \quad (1)$$

where R , G , B denote the red, green and blue channels of the input RGB image I . This image color transformation process leads to enhance the green color in the image, which is distinctive for vegetation. Next, we operate a thresholding operation based on the *ExG* index and obtain a binary mask which is 1 in correspondence of the canopy, 0 otherwise. This simple approach allows to efficiently and effectively identify the canopy and distinguish it from the soil, trunk and any other object in the scene.

The binary mask is then applied to the output of the YOLO detection system in order to isolate the canopies of the suckers in the predicted bounding boxes. An example of the resulting segmented image is shown in Figure 4c, where the canopies of the trees are highlighted in green, whereas their suckers are represented with red color. The figure also shows the ability of the *ExG*-based binary mask to filter out the parts of the image that are not related to the canopies of the trees.

The segmentation procedure detailed above is applied to both the RGB color image and the depth map. The data related to the suckers, i.e., the red area in the figure, is then fed to the next component which will virtually

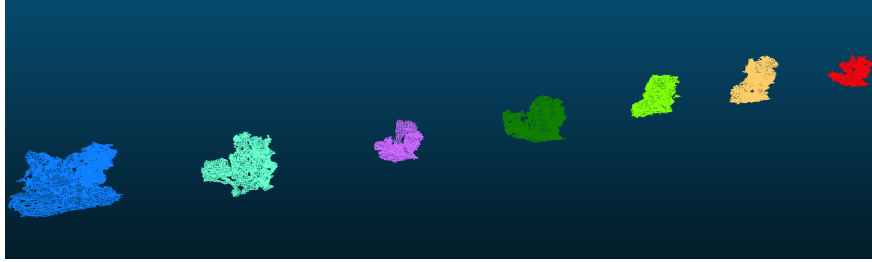


Figure 5: Example of suckers' canopies 3D mesh reconstruction for a set of 7 trees.

reconstruct them.

4.3 Suckers Reconstruction

The detection and image segmentation detailed above enable the 3D reconstruction of the suckers, which is then exploited to produce an estimation of the surface area of their canopies, which in turn will be used to compute the amount of herbicide needed for their treatment.

Our goal in this step is hence to generate a 3D mesh of the suckers and utilize the fact that meshes are made up of polygonal faces (generally triangles or squares) to produce an estimation of the surface area of the reconstructed object. To this end, many of the state-of-the-art 3D mesh reconstructor such as (Teixeira and Chli, 2016; Greene and Roy, 2017; Rosinol et al., 2020) could be used. In our case, we rely on Kimera proposed in (Rosinol et al., 2020), an open-source C++ library for real-time metric-semantic localization and mapping, mesh reconstruction, and semantic labeling in 3D. In particular, Kimera elaborates in real-time stereo video data to compute a 3D point cloud which is then used to create a 3D global mesh made up by many triangles through the marching cubes technique.

When the robot has visited all the assigned locations \mathcal{L} , a global single mesh is generated. At this point, a simple nearest neighbors clustering algorithm (e.g., (Muja. and G. Lowe., 2009; Arya et al., 1998)) is applied to the global 3D mesh to separate the suckers from the others and obtain the respective meshes as well as to recover the spatial information of the trees from which the suckers have growth. Figure 5 depicts an outcome of the reconstruction procedure where the clustering algorithm has been applied, highlighting the presence of 7 clusters of suckers. Since the generated meshes could generally present some noises, a Statistical Outlier Removal (SOR) filter is applied to each one of them (see Section 6.2 for details). From these procedures, a set $\mathcal{S} = \{1, \dots, n\}$ of suckers is generated and an association with the trees from which the suckers growth is made, i.e., each group of suckers $i \in \mathcal{S}$ is associated to a tree. The operations that will be detailed in the following are executed by the system only for the suckers \mathcal{S} that are associated with a tree that belongs to the original target tree set \mathcal{T} . We point out that, in the following, we are going to use the same indices set \mathcal{S} to refer the trees to which the groups of suckers are associated.

The next step, which will be the subject of the following section, involves the computation of the herbicide amount's estimation for the suckers $i \in \mathcal{S}$.

4.4 Herbicide Computation

Given a set of trees indexed by the set \mathcal{S} , the amount of herbicide $h_i \in \mathbb{R}_{\geq 0}$ required for the sucker of the tree $i \in \mathcal{S}$ can be computed according to the following equation:

$$h_i = \min \{f(A_i), h_i^{\max}\},$$

where $f(A_i) : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ is a function depending on the area $A_i \in \mathbb{R}_{\geq 0}$ of the canopy of the suckers of the i -th tree, and $h_i^{\max} \in \mathbb{R}_{\geq 0}$ is the maximum allowed amount of herbicide that could be given to that specific tree. The area A_i is computed from the 3D meshes obtained at the previous step. In particular, since the mesh is composed of polygonal faces, the canopy area of the suckers can be computed by evaluating the sum of the area of each face of the mesh. The maximum allowed amount h_i^{\max} is instead suggested by an agronomical expert which considers different tree characteristics such as age, geometrical structure, and plantation layout according to (Silvestri et al.,

2021a; Silvestri et al., 2021b). Regarding the definition of the function $f(\cdot)$, given that big suckers in size correspond to larger amounts of herbicide, it is reasonable to consider a law that is directly proportionate to the size of the canopies of the suckers, i.e.,

$$f(A_i) = k A_i + c, \quad (2)$$

where $k, c \in \mathbb{R}$ are parameters to be estimated. In our case, as detailed in 6.2, we followed a data-driven approach to estimate them. In particular, we first estimated the canopy area of a set of 35 trees and then collected for those trees the recommended amounts of herbicide from an agronomical expert, allowing us to estimate these parameters with a data-driven least square regression.

5 Sucker Treatment

In this section we detail the final phase of the autonomous sucker management architecture, in which the robot autonomously applies the computed herbicide solution to the suckers in need of treatment. For this purpose, we consider that the robot is equipped with a sprayer mechanism composed of two actuators providing pan-tilt motion capabilities and an atomizer nozzle, as in our experimental setup.

To apply the herbicide, first of all, the robot navigates the orchard towards the target trees \mathcal{T} . Once the robot has reached the area of interest, it travels from location to location of the list \mathcal{L} to evaluate the plants' needs and executes the processing pipeline described in Section 4 to compute the required amount of herbicide solution.

Next, based on this information, for each tree $i \in \mathcal{S}$, the time required for the sprayer to release the estimated amount h_i is computed. At this point, the robot is ready to run the application procedure: *i*) it navigates towards the location $j \in \mathcal{L}$ related to the tree $i \in \mathcal{S}$, possibly orienting itself to facilitate the application of the herbicide to the suckers with the sprayer, *ii*) it estimates the time necessary for the sprayer to release the needed amount of herbicide, *iii*) it computes the positions in which the sprayer actuators must be placed to point the sucker and moves them, and *iv*) it activates the atomizer for the estimated amount of time.

5.1 Spraying Time Computation

We now focus on the problem of computing the time in which the atomizer must be active in order to release the estimated amount of herbicide h_i for each sucker $i \in \mathcal{S}$. In particular, assuming that the atomizer sprays at a constant pressure, as validated for the atomizer mounted on the robot used for the experimental validation, the amount of sprayed solution $h_i \in \mathbb{R}$ for a group of suckers $i \in \mathcal{S}$ in a certain time t_i can be approximated as a linear function as

$$h_i = r_s t_i^\kappa, \quad (3)$$

where $r_s \in \mathbb{R}$ is the mL/s rate of spraying of the atomizer and $\kappa \in \mathbb{R}$ is a parameter to be estimated. Clearly, the time t_i in which the atomizer must be active can be computed as

$$t_i = r_s^{-\frac{1}{\kappa}} h_i^{\frac{1}{\kappa}}. \quad (4)$$

In our case, as it will be detailed later in Section 6.2, the parameters r_s and κ have been estimated in a data-driven manner.

5.2 Sprayer Motion

In order to appropriately move the sprayer mechanism, we derive its analytical inverse kinematic model, that provides the sprayer joint configurations given a target position to spray. We choose the target position to be coincident with the estimated center of the sucker to treat. The sprayer mechanism, depicted in Figure 7, is composed of two revolute joints $\mathbf{q} = [q_1, q_2]^T \in \mathbb{R}^2$ and two links with lengths $l_1, l_2 \in \mathbb{R}^+$, respectively. We denote the base positions of the first and second links by $\mathbf{p}_0 = [x_0, y_0, z_0]^T \in \mathbb{R}^3$ and $\mathbf{p}_1 = [x_1, y_1, z_1]^T \in \mathbb{R}^3$, respectively, where \mathbf{p}_0 is obtained by the odometry of the robot with z_0 fixed and equal to 0.625 m in our setup. Moreover, we denote the robot orientation by $\theta \in (-\pi, +\pi]$ and the target position by $\mathbf{p}_t = [x_t, y_t, z_t]^T \in \mathbb{R}^3$. Finally, we introduce the distance between the first link base, \mathbf{p}_0 , and the target, \mathbf{p}_t , given by $d_{0t} = \|\mathbf{p}_0 - \mathbf{p}_t\|$ as well as the same distance in the xy plane given by

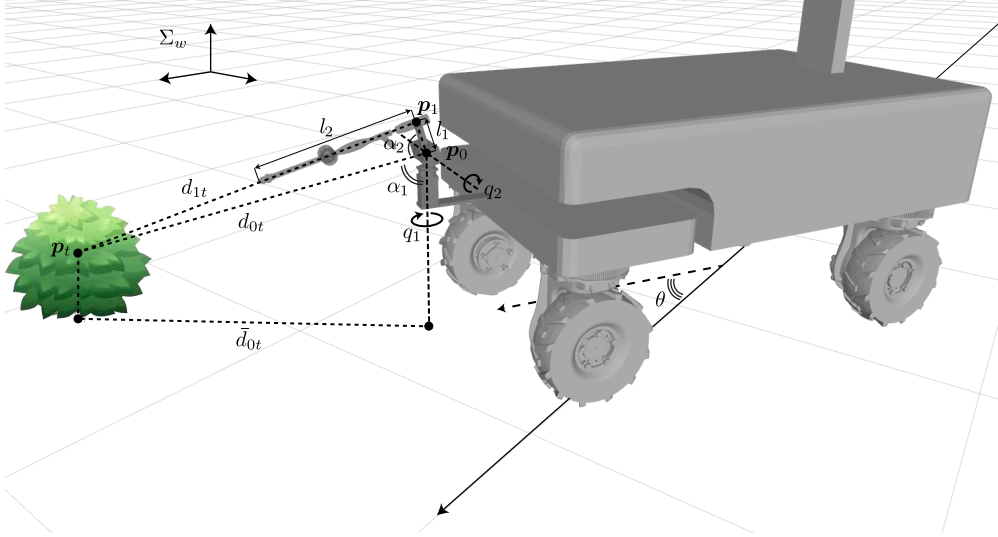


Figure 6: Model of the pan-tilt sprayer system with depiction of its main variables. The targeted sucker of a tree with position \mathbf{p}_t (on the left) and the world reference frame Σ_w (on the top) are also shown.

$\bar{d}_{0t} = \sqrt{(x_t - x_0)^2 + (y_t - y_0)^2}$. Similarly, we define the distance between the second link base, \mathbf{p}_1 , and the target, \mathbf{p}_t , given by $d_{1t} = \|\mathbf{p}_1 - \mathbf{p}_t\| = \sqrt{d_{0t}^2 - l_1^2}$.

The desired joint configurations $\mathbf{q}_d = [q_{d,1}, q_{d,2}]^T$ to point the target are given by:

$$\begin{cases} q_{d,1} = \text{atan2}(y_t - y_0, x_t - x_0) - \theta \\ q_{d,2} = \underbrace{\text{atan2}(\bar{d}_{0t}, z_0 - z_t)}_{\alpha_1} + \underbrace{\text{atan2}(d_{1t}, l_1)}_{\alpha_2} \end{cases} \quad (5)$$

where α_1 and α_2 are highlighted in Figure 6 to facilitate the understanding. At this point, given the current sprayer mechanism joint configuration \mathbf{q} , a trajectory is planned in the joint space to reach the desired configuration \mathbf{q}_d and is provided as input to the low-level controller of the sprayer mechanism. Finally, when the desired configuration is achieved, the sprayer is activated for the amount of time t_i calculated as in (4).

6 Experimental Setup

In this section we report the experimental setup and the implementation details for the validation of the proposed architecture. The experimental campaigns have been carried out within the hazelnut orchard available for the project PANTHEON, within the ‘‘Azienda Agricola Vignola’’, a farm located in the municipality of Caprarola, in the province of Viterbo, Italy. Figure 7 depicts the ground robotic platform used for the project, namely SHERPA-HL, equipped with a MYNT EYE D120 RGB-D camera for the detection of the suckers, a tank and sprayer system for the application of the herbicide, a NVIDIA Jetson AGX Xavier for the on-board computation, and an autonomous navigation architecture based on (Carpio et al., 2020).

6.1 Dataset collection

As first step for the development of the detection system, we collected images, and respective depth maps, containing examples of suckers in the hazelnut orchard involved in PANTHEON. In particular, the dataset was collected over different vegetative seasons between May 2019 and August 2020 in order to improve the system generalization capabilities. The dataset has been made public¹ to boost reproducibility and results comparison.

¹ <https://tinyurl.com/sucker-dataset>



Figure 7: The robotic ground platform used for the experimental validation.



Figure 8: Samples of the RGB images in the dataset collected in the hazelnut orchard.

Figure 8 shows examples of RGB images depicting suckers in the hazelnut orchard. Note that, differently from existing works involving laboratory setups only, the collected dataset is fully representative of real-world conditions as it was directly acquired on trees in need of sucker treatment in the hazelnut orchard. This led to the occurrence of a natural variability in the dataset in terms of dimension of the suckers, acquisition distances and orientations, backgrounds. A 3D camera MYNT EYE D120 with maximum resolution 1280×720 and depth map with nominal accuracy of 5 mm was used to collect the dataset. Images were recorded both by having the camera mounted on the robotic platform, as shown in Figure 1, and by manually holding it and navigating the field. A total of 1100 and 330 annotated images composes the training and validation datasets, respectively. Moreover, we used a test dataset comprising 150 images for the performance assessment of the detection system.

6.2 Implementation details and ROS integration

Regarding the detection system, we used a smaller architecture for YOLOv4 models compared to the one in (Bochkovskiy et al., 2020) in order to achieve real-time performance. In particular, the considered architecture comprises 21 convolutional layers for backbone and neck and two YOLO head layers. The input network resolution was $416 \times 416 \times 4$, where 4 means that RGB and depth channels, i.e., RGB-D images, are fed in. Moreover, we compared the results to the case of RGB input only, i.e., input resolution equal to $416 \times 416 \times 3$. Transfer learning was applied considering pre-trained weights on ImageNet dataset (Deng et al., 2009). A total of 10000 iterations were considered for each training process with scheduled learning rate (initialized to 0.0025 and reduced by 10% at 80% and 90% of the training process). Models were trained on a NVIDIA GeForce RTX 3090 GPU and deployed on the embedded system NVIDIA Jetson AGX Xavier mounted on the robot. The following data augmentation techniques were considered for the training: color transformations on hue, saturation and intensity channels randomly perturbed to up to 0.05%, 1.5%, and 1.5%, respectively, compared to the original values, and geometric transformations consisting in flipping operations and cropping and scaling operations in the range [0.8%, 1.2%].

Regarding the herbicide computation, to estimate the parameters k and c introduced in eq. (2) we resorted to a least square regression as mentioned in Section 4.4. In particular, we computed the area of a set of 35 suckers in the

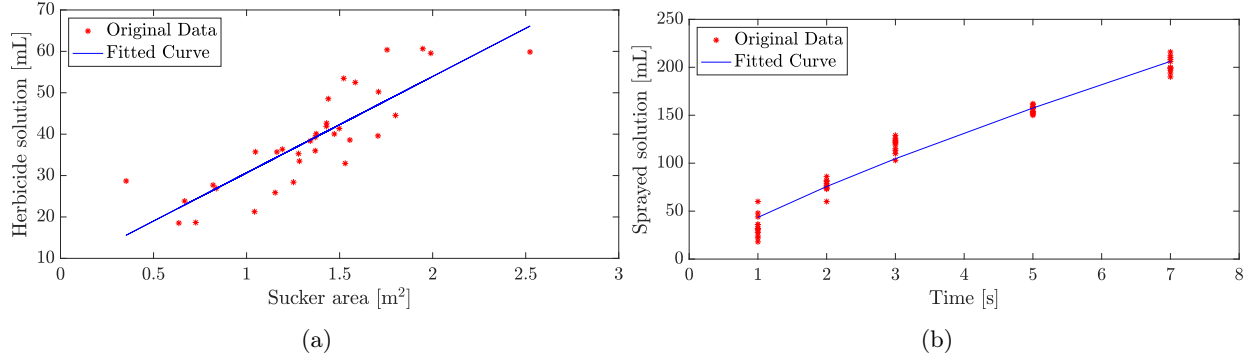


Figure 9: Results of the least square estimation procedure used to tune the parameters k and c in eq. (2) (a) and the parameter r_s in eq. (3) (b).

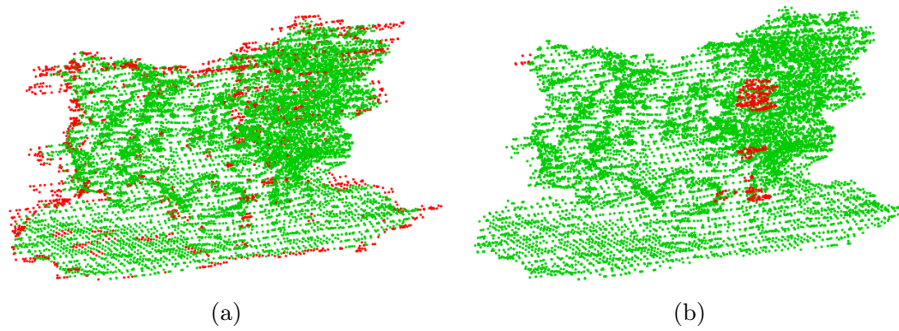


Figure 10: Representation of the application of steps S1) (a) and S2) (b) to a cluster of reconstructed suckers. The red points are points being filtered out by the ROS node.

orchard and asked an agronomical expert to evaluate the amount of herbicide recommended for that specific suckers, as shown in the first part of the accompanying video.

The results of the linear regression are shown in Figure 9a where on the horizontal axis it is reported the value of the area in m^2 and on the vertical axis it is reported the amount of mL. Note that the value in mL corresponds to a solution of water and herbicide. In particular, the herbicide used in our evaluation is the “Evolution” herbicide which needs to be the 0.00267% of the solution, i.e., for 1 L of solution of water and herbicide, only 2.67 mL is effectively herbicide. The resulting values are $k = 23.278$ and $c = 7.378$ with a normalized root-mean-square deviation equal to 0.1445, which we recall is computed as $NRMSD = \frac{1}{y_{\max} - y_{\min}} \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$, where y_i with $i \in 1, \dots, n$ are the correct amount of milliliters, \hat{y}_i are the milliliters values estimated with eq. (2), y_{\min} and y_{\max} are the minimum and maximum correct amounts of milliliters and n is the samples’ size.

Following the same approach used for the parameters k and c , we estimated the parameter $r_s^{-\frac{1}{\kappa}}$ used in the computation of the spraying time t_i introduced in eq. (4). Specifically, activating the sprayer for a fixed amount of seconds $t \in \{1, 2, 3, 5, 7\}$, the amount of sprayed solution has been collected and measured. Then, the collected data composed of 56 samples has been used to fit a generic function of the form $a f^\kappa$ where the parameter κ has been selected equal to 0.8 as the result of different numerical evaluations. The results of the least square estimation are summarized in Figure 9b. The estimated value resulted in $r_s = 43.4646$ mL/s with a normalized root-mean-square deviation equal to 0.0563.

When the reconstruction phase ends, a service call to another ROS node is made to filter the generated mesh file and extract the data about the individual plants. In particular the following steps are executed by this latter node:

- The mesh is converted to a point cloud using Point Cloud Library (PCL (Rusu and Cousins, 2011));
- The individual suckers are extracted using an Euclidean clustering algorithm that exploit the known distance



Figure 11: Example of RGB image (a) and respective raw (b) and normalized (c) depth map.

of the trees in the orchard;

- For each sucker:
 - S1) A sparse outlier removal (SOR) filter is applied to remove spurious points and outliers;
 - S2) A second Euclidean clustering algorithm is used to select the biggest cluster within the sucker’s point cloud (the first clustering operation and the SOR filter may have introduced some ill-defined points);
 - S3) The 3D mesh of the sucker is reconstructed recovering portions of the original mesh generated by Kimera.

Figure 10 shows the application of the different steps of the filter process. In particular, Figure 10a depicts the effect of the filter applied in step S1), i.e., the SOR filter, where the points represented in red color are points that are being removed, whereas Figure 10b shows the effect of step S2), i.e., the second clustering procedure, where the final cluster is represented in green color.

7 Experimental Validation

In this section we present the results obtained in the experimental validation of the proposed architecture. The accompanying video, as detailed in the following Section 7.3, shows the execution of the entire autonomous sucker management system in the hazelnut orchard located in the municipality of Caprarola.

7.1 Suckers detection performance

To assess the performance of the detection system, we considered: the average IoU ($\overline{\text{IoU}}$), which is equal to 1 in case ground truth and predicted bounding boxes are perfectly overlapping and 0 in case they are disjoint, the mean Average Precision (mAP), which is the the area under the precision-recall curve (Everingham et al., 2010), taking into account the confidence scores, and is 1 in case of maximum precision, 0 otherwise, and frames per second (fps), that is the number of frames processed per second and allows to evaluate the computational performance of the system.

As far as the ROS integration is concerned, we integrated the YOLOv4 framework in (Bjelonic, 2018), which originally provides a ROS package for the real-time use of the YOLOv3 framework. Our integration enabled the YOLOv4 framework on ROS. The output of the YOLO-ROS package are detection images, bounding boxes, and number of detected suckers. These outputs are then filtered by a ROS node to produce the segmented data as discussed in Section 4.2 and are then used to virtually reconstruct a 3-dimensional mesh of the suckers with the Kimera library as detailed in Section 4.3. In our case, since the SHERPA-HL is provided with an autonomous localization system, the camera’s pose data, required by Kimera to virtually reconstruct the environment, are provided by the robot itself, enabling a more precise localization. Alternatively, as we initially employed in our first validations, the localization of the camera can be provided by other SLAM architecture for RGB-D camera available at the state of the art, e.g., ORBSLAM (Mur-Artal and Tardós, 2017). Notably, this alternative choice allows our proposed architecture to be used even when no ground robot is available. Indeed, in the case that a human operator would like to analyze and evaluate the amount of herbicide only for a small set of suckers without deploying a ground robot, it would suffice a hand-operated RGB-D camera (connected to the ROS architecture) to obtain the desired information.

ID	Input	Data aug.	$\overline{\text{IoU}}$ [%]	mAP [%]
M1	RGB		63.49	75.36
M2	RGB	✓	64.23	78.14
M3	RGB-D		66.81	76.91
M4	RGB-D	✓	66.27	78.65
M5	RGB-D Norm.		69.14	78.12
M6	RGB-D Norm.	✓	70.39	79.43

Table 1: Evaluation results of the sucker detection system in terms of $\overline{\text{IoU}}$ and mAP. Best results in bold.











Suckers Image										
Expert [mL]	55.80	39.02	60.66	29.12	66.49	23.23	35.53	44.20	55.28	39.15
Auto. [mL]	63.03	44.73	64.94	30.71	59.80	19.78	33.26	42.86	60.00	37.66
Rel. Error [%]	12.96	14.63	7.06	5.46	10.06	14.85	6.39	3.03	8.54	3.81

Table 2: Comparative analysis conducted on a set of 10 suckers: for each plant, the estimated amounts of herbicide suggested by an agronomical expert and by the proposed automatic framework and the relative error are reported.

Regarding the accuracy, Table 1 summarizes the results of the sucker detection system on the test set. In particular, we compared the performance when changing the input, i.e., RGB image, shown in Figure 11a, RGB image with raw depth map, that is, the one from the RGB-D camera, shown in Figure 11b, and RGB image with normalized depth map, shown in Figure 11c, as well as when applying or not the data augmentation techniques described in the previous section. Starting from the model with RGB input and no data augmentation (model M1) achieving $\text{mAP} = 75.36\%$ and $\overline{\text{IoU}} = 63.49\%$, we can observe that a first improvement of the accuracy is recorded when introducing data augmentation (model M2), obtaining an improvement of both mAP and $\overline{\text{IoU}}$, i.e., $\text{mAP} = 78.14\%$ and $\overline{\text{IoU}} = 64.23\%$. A further improvement compared to model M1 is recorded when using the additional information of the depth map (model M3), which leads to $\text{mAP} = 76.91\%$ and more accurate bounding boxes with $\overline{\text{IoU}} = 66.81\%$, thus confirming the effectiveness of exploiting the depth information for suckers. This performance is increased when data augmentation is introduced (model M4), achieving $\text{mAP} = 78.65\%$ and $\overline{\text{IoU}} = 66.27\%$, which also outperforms model M2. Finally, best performance is obtained when also applying a preprocessing normalization step to the depth map reaching $\text{mAP} = 78.12\%$ ($\overline{\text{IoU}} = 69.14\%$) and $\text{mAP} = 79.43\%$ ($\overline{\text{IoU}} = 70.39\%$) without (model M5) and with (model M6) data augmentation, respectively. This can be motivated by the fact that, as shown in Figure 11, the normalization preprocessing step (Figure 11c) allows to enhance the contrasts of the depth map compared to the raw map (Figure 11b), thus facilitating the detection process.

Examples of detected suckers, using the latter model, along with their prediction confidence scores are provided in Figure 4a as well as in the accompanying video.

As far as the computational performance is concerned, the detection system reaches 49.5 fps and 49 fps when using the RGB input and when adding the depth information, also with normalization step, respectively, on the embedded system NVIDIA Jetson AGX Xavier. This confirms the real-time processing capabilities of the detection system.

7.2 Herbicide estimation validation

In order to evaluate the amount of estimated herbicide solution computed by the proposed framework, we conducted an experimental validation on a set of 10 suckers comparing the estimations of our framework with the suggestions of agronomical experts. Table 2 reports the results of the comparison where the first column contains images of the analyzed suckers, the second and third columns report the values suggested by the agronomical expert and by the proposed framework, respectively, and the last column shows the relative error. The mean error between the amount



Figure 12: Screenshots of the video showing in (a) the robot inspecting the trees, detecting suckers and reconstructing them, and in (b) the robot spraying a sucker.

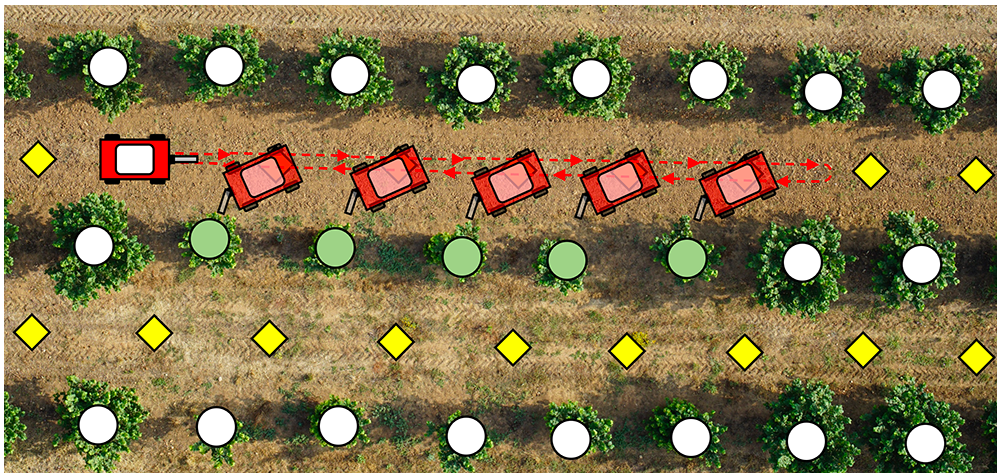


Figure 13: Depiction of the field graph and the target trees \mathcal{T} (in green). The path of the robot from the initial configuration (on the left) is highlighted with red arrows.

computed by the proposed framework and the one suggested by the agronomical expert is equal to 3.88 mL with a standard deviation of 2.20 mL, i.e., mean relative error is equal to 8.68% with a standard deviation of 4.32%. These results thus show that on average an estimation error below 9% is made which has been positively acknowledged by a set of agronomical experts. Note that these results allow to validate the entire pipeline from the sucker detection, to its reconstruction up to the herbicide computation based on expert knowledge.

7.3 Autonomous pipeline validation

The entire proposed autonomous sucker management architecture was validated in the PANTHEON hazelnut orchard as shown in the accompanying video and depicted in the screenshots in Figure 12.

In particular, five trees belonging to the same row were considered as target trees \mathcal{T} , as depicted in Figure 13. In the figure, the path followed by the robot during the experiment is also reported in red, showing the configurations where treatments are performed.

In the second part of the video, the robot is shown to autonomously navigate the locations \mathcal{L} associated with the selected trees and inspect the respective trees, as shown in Figure 12a, according to the pipeline in Section 4. In detail, despite the different sizes and shapes, all five suckers are *i*) detected in real-time using the RGB-D based YOLO framework (Section 4.1), *ii*) segmented using the *ExG* index (Section 4.2) and *iii*) reconstructed and localized using Kimera library (Section 4.3).

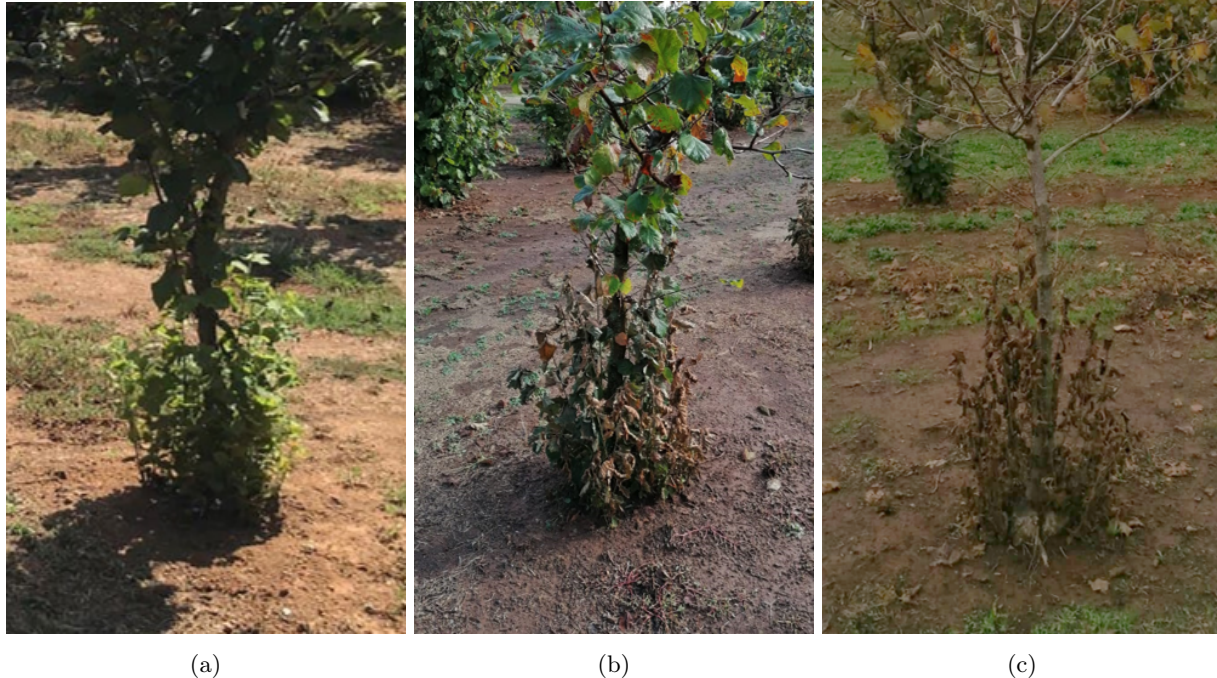


Figure 14: Depiction of sucker #2 before treatment (a), as well as two weeks (b) and one months (c) after treatment.

In the third part of the video, the actual treatment of the suckers is shown, as reported in Figure 12b. To this aim, the robot autonomously navigates the same row backwards and, for each detected sucker, computes the amount of herbicide according to its canopy area (Section 4.4). For example, suckers #5 and #4 require 59.8 mL and 30.7 mL of herbicide solution, respectively, coherently with their significant size difference, i.e., they have estimated areas equal to 2.252 m² and 1.002 m², respectively. Then, for each sucker, the sprayer is positioned to aim it (Section 5.2) and the herbicide solution is applied by activating the atomizer for the required amount of time (Section 5.1).

Figure 14 depicts sucker #2 before treatment (Figure 14a) as well as two weeks (Figure 14b) and one month (Figure 14c) after the herbicide application, respectively, showing its desiccation.

8 Conclusions

In this work, we proposed a modular architecture for autonomous sucker management which enables a per-plant treatment in large-scale orchards, reducing waste and environmental pollution. As a first step, suckers are detected through a data-driven detection system based on YOLOv4 framework, and are segmented resorting to the Excess Green index. Next, a 3D reconstruction is performed for each sucker in order to estimate the surface area of its canopy. Based on the latter, the amount of herbicide to apply is computed by exploiting prior knowledge provided by an expert agronomist. At this point, the actual sucker treatment can be carried out: the amount of required spraying time is estimated, the sprayer mechanism is positioned in the desired configuration to aim the suckers and the atomizer is activated for the estimated amount of time in order to release the herbicide. We validated the effectiveness of the individual components of the proposed architecture as well as the overall integrated system on a real-world hazelnut orchard using a ground robot autonomously navigating the field. Future works will focus on improving the detection system to discern between treated and non-treated suckers and on the development of an algorithm to autonomously select the best agronomic intervention location \mathcal{L} for the robot given a set of target trees \mathcal{T} .

Acknowledgments

This work was supported by the European Commission under the Grant agreement number 774571 – Project PAN-THEON.

References

- Arya, S., Mount, D. M., Netanyahu, N. S., Silverman, R., and Wu, A. Y. (1998). An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM*, 45(6):891–923.
- Berenstein, R. and Edan, Y. (2018). Automatic adjustable spraying device for site-specific agricultural application. *IEEE Transactions on Automation Science and Engineering*, 15(2):641–650.
- Bjelonic, M. (2016–2018). YOLO ROS: Real-time object detection for ROS. https://github.com/leggedrobotics/darknet_ros.
- Bochkovskiy, A., Wang, C.-Y., and Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.
- Botterill, T., Paulin, S., Green, R., Williams, S., Lin, J., Saxton, V., Mills, S., Chen, X., and Corbett-Davies, S. (2017). A robot system for pruning grape vines. *Journal of Field Robotics*, 34(6):1100–1122.
- Cantelli, L., Bonaccorso, F., Longo, D., Melita, C. D., Schillaci, G., and Muscato, G. (2019). A small versatile electrical robot for autonomous spraying in agriculture. *AgriEngineering*, 1(3):391–402.
- Carpio, R. F., Potena, C., Maiolini, J., Ulivi, G., Rosselló, N. B., Garone, E., and Gasparri, A. (2020). A navigation architecture for ackermann vehicles in precision farming. *IEEE Robotics and Automation Letters*, 5(2):1103–1110.
- Chéné, Y., Rousseau, D., Lucidarme, P., Bertheloot, J., Caffier, V., Morel, P., Étienne Belin, and Chapeau-Blondeau, F. (2012). On the use of depth camera for 3d phenotyping of entire plants. *Computers and Electronics in Agriculture*, 82:122–127.
- Danton, A., Roux, J.-C., Dance, B., Cariou, C., and Lenain, R. (2020). Development of a spraying robot for precision agriculture: An edge following approach. In *IEEE Conference on Control Technology and Applications*, pages 267–272.
- De-An, Z., Jidong, L., Wei, J., Ying, Z., and Yu, C. (2011). Design and control of an apple harvesting robot. *Biosystems engineering*, 110(2):112–122.
- Deng, J., Dong, W., Socher, R., Li, L., Kai Li, and Li Fei-Fei (2009). Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.
- Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. *Int. J. Computer Vision*, 88(2):303–338.
- Furchi, A., Lippi, M., Carpio, R. F., and Gasparri, A. (2022). Route optimization in precision agriculture settings: A multi-steiner TSP formulation. *IEEE Transactions on Automation Science and Engineering*, pages 1–18.
- Greene, W. N. and Roy, N. (2017). FLAME: Fast lightweight mesh estimation using variational smoothing on delaunay graphs. In *IEEE International Conference on Computer Vision*, pages 4696–4704.
- Guijarro, M., Pajares, G., Riomoros, I., Herrera, P., Burgos-Artizzu, X., and Ribeiro, A. (2011). Automatic segmentation of relevant textures in agricultural images. *Computers and Electronics in Agriculture*, 75(1):75–83.
- Hejazipoor, H., Massah, J., Soryani, M., Asefpour Vakilian, K., and Chegini, G. (2021). An intelligent spraying robot based on plant bulk volume. *Computers and Electronics in Agriculture*, 180:105859.
- Kang, F., Wang, H., Pierce, F., Zhang, Q., and Wang, S. (2012). Sucker detection of grapevines for targeted spray using optical sensors. *Transactions of the ASABE (American Society of Agricultural and Biological Engineers)*, 55:2007–2014.
- Kim, J. and Son, H. I. (2020). A voronoi diagram-based workspace partition for weak cooperation of multi-robot system in orchard. *IEEE Access*, 8:20676–20686.
- Lippi, M., Bonucci, N., Carpio, R. F., Contarini, M., Speranza, S., and Gasparri, A. (2021). A yolo-based pest detection system for precision agriculture. In *Mediterranean Conference on Control and Automation*, pages 342–347.
- Muja, M. and G. Lowe, D. (2009). Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Applications - Volume 1: VISAPP*, pages 331–340. INSTICC, SciTePress.

- Mur-Artal, R. and Tardós, J. D. (2017). ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262.
- Nakao, N., Suzuki, H., Kitajima, T., Kuwahara, A., and Yasuno, T. (2017). Path planning and traveling control for pesticide-spraying robot in greenhouse. *Journal of Signal Processing*, 21(4):175–178.
- Nguyen, T., Slaughter, D., Townsley, B., Carriedo, L., Maloof, J., and Sinha, N. (2016). Comparison of structure-from-motion and stereo vision techniques for full in-field 3d reconstruction and phenotyping of plants: An investigation in sunflower. In *ASABE Annual International Meeting*.
- Potena, C., Carpio, R. F., Pietroni, N., Maiolini, J., Ulivi, G., Garone, E., and Gasparri, A. (2020). Suckers emission detection and volume estimation for the precision farming of hazelnut orchards. In *IEEE Conference on Control Technology and Applications*, pages 285–290.
- Pound, M., French, A., Murchie, E., and Pridmore, T. (2014). Automated recovery of three-dimensional models of plant shoots from multiple color images. *Plant physiology*, 166.
- Redmon, J. and Farhadi, A. (2017). Yolo9000: better, faster, stronger. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7263–7271.
- Redmon, J. and Farhadi, A. (2018). YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
- Rosinol, A., Abate, M., Chang, Y., and Carlone, L. (2020). Kimera: an open-source library for real-time metric-semantic localization and mapping. In *IEEE International Conference on Robotics and Automation*, pages 1689–1696.
- Rusu, R. B. and Cousins, S. (2011). 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation*, pages 1–4.
- Sammons, P. J., Furukawa, T., and Bulgin, A. (2005). Autonomous pesticide spraying robot for use in a greenhouse. In *Australian Conference on Robotics and Automation*, volume 1.
- Serdar, U. and Akyuz, B. (2018). Sucker management methods in hazelnut cultivation. In *ActaHortic.*, number 1226, pages 309–314. International Society for Horticultural Science, Leuven, Belgium.
- Silvestri, C., Bacchetta, L., Bellincontro, A., and Cristofori, V. (2021a). Advances in cultivar choice, hazelnut orchard management, and nut storage to enhance product quality and safety: an overview. *Journal of the Science of Food and Agriculture*, 101(1):27–43.
- Silvestri, C., Santinelli, G., Pica, A. L., and Cristofori, V. (2021b). Mechanical pruning of european hazelnut: effects on yield and quality and potential to exploit its by-product. *European Journal of Horticultural Science*, 86(2):189–196.
- Teixeira, L. and Chli, M. (2016). Real-time mesh-based scene estimation for aerial inspection. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4863–4869.
- Tomasone, R., Colorio, G., Cedrola, C., and Pagano, M. (2009). Mechanical and physical control of hazelnut suckers. In *ActaHortic.*, number 845, pages 407–412. International Society for Horticultural Science, Leuven, Belgium.
- Wang, Y., Shasha, X., Wenbin, L., Kang, F., and Yongjun, Z. (2017). Identification and location of grapevine sucker based on information fusion of 2d laser scanner and machine vision. *International Journal of Agricultural and Biological Engineering*, 10(2):84–93.
- Wang, Y., Zhao, C., Wang, X., Zhang, P., Li, P., and Liu, H. (2021). Inverse kinematics of a 7-dof spraying robot with 4r 3-dof non-spherical wrist. *Journal of Intelligent & Robotic Systems*, 101(4):1–17.
- Woebbecke, D. M., Meyer, G. E., Bargaen, K. V., and Mortensen, D. A. (1995). Color indices for weed identification under various soil, residue, and lighting conditions. *Transactions of the ASABE (American Society of Agricultural and Biological Engineers)*, 38:259–269.
- Xiao, K., Ma, Y., and Gao, G. (2017). An intelligent precision orchard pesticide spray technique based on the depth-of-field extraction algorithm. *Computers and Electronics in Agriculture*, 133:30–36.
- Xie, B.-X., Wang, C.-H., Ke, J.-Y., and Chang, C.-L. (2020). Design and implementation of a machine visionbased spraying technique for smart farming. In *International Automatic Control Conference*, pages 1–5.
- Zhang, Q., Chen, M. S., and Li, B. (2017). A visual navigation algorithm for paddy field weeding robot based on image understanding. *Computers and Electronics in Agriculture*, 143:66–78.
- Zhang, Q., Karkee, M., and Tabb, A. (2019). The use of agricultural robots in orchard management. *Burleigh Dodds Series in Agricultural Science*, page 187–214.
- Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., and Ren, D. (2020). Distance-iou loss: Faster and better learning for bounding box regression. In *AAAI Conference on Artificial Intelligence*, volume 34, pages 12993–13000.