

Original papers

Grapevine winter pruning: Merging 2D segmentation and 3D point clouds for pruning point generation

Miguel Fernandes^{a,b}, Juan D. Gamba^a, Francesco Pelusi^c, Angelo Bratta^a,
Darwin Caldwell^b, Stefano Poni^c, Matteo Gatti^c, Claudio Semini^a

^a Dynamic Legged Systems Lab (DLS), Istituto Italiano di Tecnologia, Via S. Quirico 19D, Genova, 16163, Italy

^b Advanced Robotics (ADVR), Istituto Italiano di Tecnologia, Via S. Quirico 19D, Genova, 16163, Italy

^c Department of Sustainable Crop Production (DLPRO.VE.S.), Università Cattolica del Sacro Cuore, Via Emilia Parmense 84, Piacenza, 29122, Italy

ARTICLE INFO

Keywords:

Computer vision
Machine learning
Spur pruning
Plant organ segmentation
Viticulture

ABSTRACT

Grapevine winter pruning is a labor-intensive and repetitive process that significantly influences grape yield and quality at harvest and produced wine. Due to its complexity and repetitive nature, the task demands skilled labor that needs to be trained, as in many other agricultural sectors. This paper encompasses an approach that targets using a robotic system to perform autonomous grapevine winter pruning using a vision system and artificial intelligence. In our previous work, we presented a 2D neural network that segmented images of grapevines into 5 different classes of plant organs during their dormant season. In this paper, we expand into the third dimension, introducing point clouds into our algorithm. The 3D approach creates instance-segmented point clouds using depth images and segmentation masks obtained with our 2D neural network. After the 3D reconstruction, the system extracts thickness measurement and uses agronomic knowledge to place pruning points for balanced pruning. The study not only delineates the integration of 2D and 3D methods but also scrutinizes their efficacy in pruning point identification. The real-world performance of the created system was evaluated and statistically analyzed on data collected during field trials in the winter pruning season 2022/2023, where the system was used in a potted vineyard to prune a set of test vines, where the positive success rate is 54.2%. Moreover, as one of the main contributions, the paper underscores a unique facet of adaptability, presenting a customizable framework that empowers end-users to fine-tune parameters according to the expected balanced pruning. This adaptability extends to variables such as the number of nodes to retain on pruned spurs and the preferred cane thickness, encapsulating the versatility of the 3D approach.

1. Introduction

An important task to perform in a vineyard is winter pruning, a complex operation that needs to be completed during the dormant season (Poni et al., 2016). A balanced winter pruning allows a good compromise between remunerative yield and desired grape quality, maximizing the grower's income (Intrieri and Poni, 1995; Poni et al., 2018). This selective operation requires about 80 to 100 man-hours per hectare annually. Table 1 shows an overview of the man-hours needed for a list of annual operations in the vineyard. Due to increased skill shortages and limited labor availability, automating winter pruning is a crucial step to improve efficiency and reduce production costs. Fig. 1 shows an example of manual pruning and mechanical pre-pruning completed by manual follow-up.

The project VINUM, which is a collaborative effort between the Istituto Italiano di Tecnologia (IIT) and the Università Cattolica del

Sacro Cuore in Piacenza (UCSC), has been focussing on automating the process of spur winter pruning for vertically shoot positioned (VSP) grapevines. This ongoing collaboration has been dedicated to developing innovative solutions for automating and streamlining the winter pruning operation. Over the past few years, the VINUM project has progressed in advancing automation technologies specifically tailored for grapevine spur winter pruning.

1.1. Context of the study

This study targets the automation of the winter pruning operation by integrating a mobile platform and a robotic arm capable of working in the vineyard. To achieve the best pruning decisions, i.e., to detect the correct point to cut, it is important to understand the structure of the pruning regions, and the plant organs that compose it. AI algorithms

* Corresponding author.

E-mail address: miguel.ferreira@iit.it (M. Fernandes).



Fig. 1. Example of manual pruning (left) and mechanical pre-pruning (right). Manual follow-up is still needed after the mechanical pre-pruning.

Table 1

Labor demand in hours per hectare for the annual vineyard management. Guyot* refers to Guyot + Mechanical Harvesting, and Spur pruned cordon** corresponds to Spur pruned cordon + Mechanical harvesting + Pre-pruning. The targeted type of plant for this work, highlighted in bold, is the spur pruned cordon, and winter pruning is the second most expensive operation, with the first being harvesting.

Source: (Table adapted from Gatti et al. (2011a,b)).

	Guyot	Guyot*	Spur pruned cordon	Spur pruned cordon**
Fertilization	1	1	1	1
Weeding	2	2	2	2
Mowing	2–3	2–3	2–3	2–3
Plant protection (spray)	10–15	10–15	10–15	10–15
Desuckering	5	5	5	5
Shoot thinning	30	30	50	50
Shoot positioning	2–4	2–4	2–4	2–4
Trimming	2–4	2–4	2–4	2–4
Leaf removal	–	–	5	5
Harvest	140	2–4	150	2–4
Winter Pruning (Including cane stripping)	100	100	80	–
Pre-pruning followed by hand-finishing	–	–	–	25
Cane positioning	20	20	–	–
Total	324	188	319	118

such as deep learning and other machine learning techniques have successfully demonstrated how they can overcome the limitations of classical computer vision. As shown in recent works, machine learning algorithms can be successfully applied to grapevine structure estimation and pruning (Häring et al., 2024; Gentilhomme et al., 2023; Botterill et al., 2017; Silwal et al., 2022).

The structure of a pruning region of grapevines subjected to spur pruning can be seen in Fig. 2, along with the identified grapevine organs and the correct pruning points. Each class corresponds to a different visible organ during the winter season: the *cordon* is the oldest permanent structure of the grapevine canopy; the *arm* is an at least three-year-old wood coming off the cordon; the *spur* is a two-year-old wood, and the *canes* are dormant shoots that generally appear vertically in response to vertical shoot positioning from the previous season (Poni et al., 2023). The *cane* has multiple instances of *nodes*, the points where shoots may grow in the future. In the context of a pruning region, the *basal cane* is the dormant shoot located closest to the *cordon*.

The most desirable spur, regarding physiological and enological purposes, requires specific characteristics in terms of well-mature wood related color, hardness, wood-to-pith ratio, internode length, rounded internode cross-section, and an average diameter of about 7 mm (Tassie and Freeman., 1992). The selection of pruning points on grapevines is based on agronomic knowledge and may vary depending on the enological objective; it aims to regulate the number, length, and position of each *spur* to optimize canopy growth, grape quality and yield in the next season.

1.2. Related work

According to the literature, several computer vision and AI approaches have been tested and applied in agriculture. This section gives an overview of the most relevant works and their limitations.

The authors in Esser et al. (2023) present a robotic system equipped with laser and camera sensors for high-resolution in-field plant scanning. The main contributions stated by the authors are the mobile robot, the methods to deal with outdoor lighting and localization, and a comparison between laser scanners and cameras. Both systems have positive and negative points regarding the precision of the laser scanner system versus the photo camera capture system. The authors concluded that both systems complement each other.

The authors (You et al., 2023) introduce a framework for the reconstruction of apple tree topology that can be used in tree pruning, using only RGB information in real-time. The experimental validation demonstrates that the setup can generate primary branch models with an accuracy of 4–5 mm and secondary branch models with an orientation accuracy of 15 degrees relative to the ground truth model.

Shifting our focus onto grapevines now, authors in Santos et al. (2020) present a new public dataset with grape clusters annotated in 300 images and a new annotation with interactive image segmentation to generate object masks, identifying background and occluding foreground pixels using scribbles. It has images, bounding boxes, and masks and an evaluation of two state-of-the-art methods for object detection, object segmentation, and a fruit counting methodology. The presented dataset is not exploitable for winter pruning due to the presence of grapes and leaves, being more suitable for summer pruning.

The authors of Häring et al. (2024) target the creation of an AI-based augmented reality system that provides pruning suggestions for a “gentle pruning” strategy, presenting three main contributions: a dataset of dormant grapevines, an approach to generate the pruning suggestions using a smartphone video and a mobile augmented reality application to see the pruning suggestions. This approach does not consider an evaluation of the cane diameter or other metrics for balanced pruning.

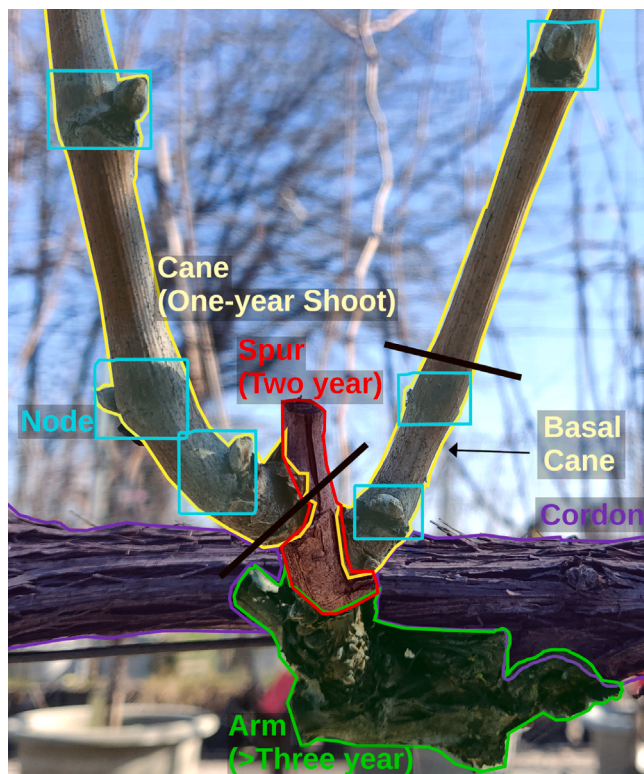


Fig. 2. Semi-schematic representation of a grapevine pruning region, illustrating the five grapevine organs: Cordon, Arm, Spur, Cane, and Node. The black lines indicate the desired pruning points. The basal cane is the cane closest to the cordon. The color overlay for each organ is created with data annotation created by agronomic experts. (Image captured in Università Cattolica del Sacro Cuore in Piacenza.).

In Gentilhomme et al. (2023), the authors introduce ViNet, a network designed to identify nodes within a grapevine plant and reconstruct the plant's structure. The study also introduces the 3D2Cut dataset, offering annotated data containing node information and synthetic backgrounds. The approach employs a stacked hourglass network to reconstruct grapevine structure, encompassing node locations, branch types, and interconnections. The drawback of a similar approach is that it requires a controlled environment with the use of a static background.

One of the main issues in recognizing grapevine organs is the occlusion of nodes at certain camera angles (Guadagna et al., 2023). With this in mind, a natural step is the emulation of a human that tilts and moves its head to obtain a different angle over the grapevine. One way to merge the visual information from multiple points of view is via 3D point clouds.

Regarding 3D point clouds applied to grapevine metric extraction, it is worth mentioning the relevant work of Williams et al. (2023). The authors have implemented a system that allows the 3D reconstruction of cane pruned grapevines for extracting several quantifiable metrics used to evaluate the quality of canes. These metrics are the diameter, orientation, length, position, visible health (if there is visible disease or damage), internode length, and node count. It uses a pair of UR5 robot arms on each side of the grapevine, each with a stereo camera system and a hardware trigger that synchronizes the images for accurate stereo matching. The work focused on 3D reconstruction without pruning or an underlying data structure that would allow the generation of pruning points.

Authors in Botterill et al. (2017) present a robotic system for automating grapevine winter pruning. Using a mobile platform with trinocular stereo cameras that envelop the grapevine, the system captures images of the vines while a computer vision system constructs

simplified 3D models. A support vector machine (SVM) based system decides which canes to prune, and a six-degree-of-freedom robot arm executes the cuts. The integrated system is tested in the vineyard, showing initial trials of unbalanced pruning. The use of AI to directly decide where to place the pruning points can be seen as a limitation since the pruning for a specific grapevine variety may not be the same for other varieties and/or different agronomic purposes.

The authors state that the system has reliability issues related to cable tangling, connection failures, and cameras losing calibration. A limitation can be seen in the controlled environment, where the system envelops the entire plant with a structure that keeps out natural light and illuminates it with synthetic uniform light.

Bumblebee (Silwal et al., 2022) is a prototype of an autonomous robotic system that performs winter pruning, introducing some novelties on several systems: camera, perception, manipulation, navigation, and robotic platform. The proposed system targets vigorous Concord (*Vitis labrusca*) grapevines trained to bilateral cordon with an average cordon height of 1.8 m. As stated by the authors, the complex vines structures were simplified by manually pre-pruning with a machine. Furthermore, the study adopted a simplified spur pruning rule to only retain 4 buds per cane. The authors reported an average pruning accuracy of 87% for a down-selected subset of canes. The pre-pruning resulted in 40% overpruned canes (less than the desired 4 buds left on the cane) and 25% already exactly pruned canes (4 buds). The subset of the remaining 35% resulted in *prunable canes* that were used in their study. The authors also mention lessons learned, such as the need for a secondary camera system to reduce occlusion, which brings such problems as the constant recalibration and manual tuning of parameters to achieve real-time point cloud processing. Lastly, one of the stated limitations is the need for “advanced sensing capabilities to assess cane health and vine size for balanced pruning”.

1.3. Contributions

In our previous work (Fernandes et al., 2021), we present a three-class neural network capable of detecting and segmenting images of grapevines during their dormant season, into cane, node, and cordon organs. It also introduced proof-of-concept pruning points. These points are placed between every node along the cane organs that are detected and connected to a cordon and are obtained via a plant graph model created based on the detected organs.

In Guadagna et al. (2023), we present an evolution to the neural network, which segments images of grapevines during their dormant season into five classes of plant organs: cordons, arms, spurs, canes and nodes.

In this work, we propose a system for autonomous winter pruning of vineyards with grapevines trained to spur-pruned cordons. A 3D reconstruction algorithm combined with a ruleset based on agronomic background detects the correct pruning points, while a robotic arm is controlled to execute the cut.

This work uses and improves upon our previous works by using the mentioned 5-class neural network, along with the plant graph model.

In particular, to address the limitation of the current state of the art introduced above, the contributions are:

- Spur-pruned grapevine 3D reconstruction via Point clouds with instance segmentation, leading to a 3D plant graph model;
- Implementation of customizable pipeline for balanced pruning point generation on 3D plant graph models;
- Experimental field trials for grapevine winter pruning of the methods introduced in this paper.

The remainder of the paper is organized as follows: Section 2 describes the algorithm that creates the grapevine's 3D model and how the pruning points are generated following agronomical criteria. Section 3 presents the material and methods. Section 4 presents the experimental setup and results. Section 5 discusses the achieved results and the limitations of our approach. Finally, Section 6 presents the conclusions and possible future works.

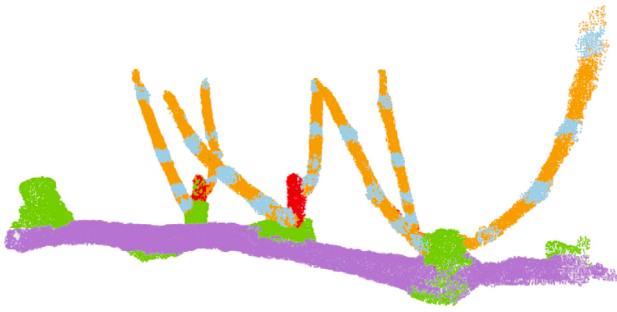


Fig. 3. Example of a final point cloud. The colors represent the individual classes, with purple representing the cordon, green the arm, red the spur, orange the cane, and blue the node.

2. Point cloud creation & pruning point generation

Perception systems face different difficulties in outdoor environments, such as shadow effects, varying light conditions, occlusions, and reflections, making it challenging to understand the entire plant structure from a single perspective. Therefore, collecting and fusing information from different perspectives is suitable for dealing with the mentioned difficulties. In this sense, we merge the segmented instances from the neural network at every collected view to obtain a 3D representation of the grapevine. Fig. 3 presents an example of a merged point cloud created by the system, with purple representing the cordon, green the arm, red the spur, orange the cane, and blue the node (see Section 2.5).

To create the point clouds, we employ a data structure that includes the following components: a timestamp, an RGB image, a depth image, and the transformation between the camera and a fixed point in the world. These components are synchronized in time. The process begins by applying a segmentation neural network to the first RGB image. The output from the network then undergoes further processing. First, within the segmentation masks domain, several tasks are executed. Initially, overlapping masks of the same class are merged to simplify the output, thereby reducing the number of resulting masks. Second, areas of the mask considered too small and indicative of errors are removed. Finally, the depth image is subject to processing. It is filtered using all the segmentation masks, with any value in the depth image falling outside a segmentation mask set to zero. Since we are filtering the depth image based on the segmentation information, we ensure that the point cloud created by the system only contains points that are also classified by the neural network. The camera position and orientation are then changed by moving the robotic arm, as explained in Section 3.3.1.

2.1. ICP & pose graph optimization

The segmentation merge is a process performed on the data, as mentioned above, in which the captured data instances are processed in a way that allows them to be merged into a single point cloud or discarded. Utilizing depth information, we project visual data into a point cloud. By incorporating the transformation information between the camera and the world, we determine the point cloud's spatial relationship to the world. Since we have a matching between the depth image and the color image, it is possible to create a relationship between the color image, the object segmentation, the depth image, and the camera's position in the world. With this related data, it is possible to build a point cloud that only includes the grapevine and each point the object class belongs to.

We first calibrated the intrinsic camera parameters using the “Dynamic Calibration Tool” from Intel RealSense. Then, we mounted the camera on the robotic arm and calibrated the extrinsic camera parameters using the “Hand-Eye Calibration” package from MoveIt (Coleman

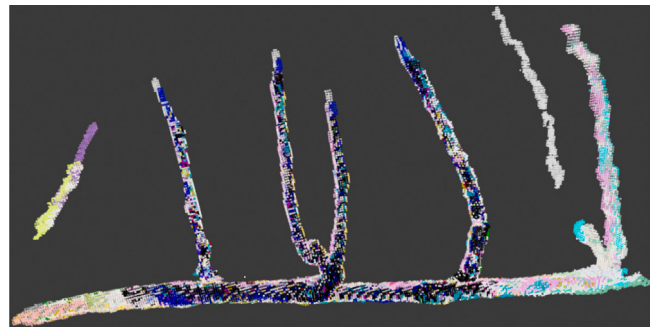


Fig. 4. Illustrative example of the point selection process, where each color represents a different view, the points in black belong to the first view, and every other random color represents subsequent iterations.

et al., 2014). In this sense, it is possible to project a colored point cloud referenced to the camera frame using the intrinsic camera parameters and depth information. Consequently, we can transform the colored point cloud to the arm base using the extrinsic camera parameters and the arm kinematics. This process is repeated for every captured frame to obtain a “global” point cloud of the plant. However, due to system imprecisions, whether on the camera or robot side or changes in the world, the alignment of point clouds may deviate, making it challenging to overlap multiple point clouds accurately.

To address this misalignment issue, we use a two-step strategy consisting of a local optimization to find an approximate homogeneous transformation between frames one and two and successively until closing the loop, and then a global optimization (pose graph optimization) is run on the closed loop to refined the locally optimized transformations from the previous step. To address this, we use a “pose graph” structure to achieve optimal point cloud overlap using the open-source library Open3D (Zhou et al., 2018). The pose graph consists of nodes representing individual point clouds and links indicating the transformations between these nodes. A circular graph needs to be formed to optimize the pose graph; in other words, a graph that consists of a closed chain connecting all the vertices. To achieve this, we employ the Point-to-Plane Iterative Closest Point (ICP) algorithm, Chen and Medioni (1992), to calculate the transformations between different point cloud nodes.

A pairwise calculation between all the point clouds is typically done to build the pose graph. The main issue is that the complexity of the pairwise registration is quadratic ($O(n^2)$). Due to the large quantity of data acquired over scanning, a simplified approach to building the pose graph was created.

We execute the following steps to add nodes to the pose graph: The first point cloud is added by default and serves as the “root” node. Starting from the second point cloud, we check the overlap between the current and last point clouds in the graph, using the ICP algorithm to determine the fitness metric. If the fitness exceeds 50%, indicating substantial overlap, the current point cloud is added to the pose graph. If not, the current point cloud is discarded, and we move on to the next point cloud and continue the process.

By only creating a closed loop, the complexity of the pairwise registration becomes linear ($O(n)$), with the drawback being that additional checks are needed to understand if all point clouds are fully aligned.

An optimization process is executed after adding all possible point clouds to the pose graph. This process brings all point clouds closer to the root point cloud, improving their overall fitness.

2.2. Point selection-based point cloud merging

After obtaining the optimized pose graph containing all the matching point clouds, the next step is to merge them based on segmentation

information. To accomplish this, we create several lists that store essential information for each point. These lists include point coordinates, class, class probability, and node probability, and they are collectively called the final point lists.

The selection of the points that make up the merged point cloud is essential to avoid duplicated information. This is caused by the fact that a single point can be seen from multiple points of view. By selecting the points where the neural network is more confident in the result, we can ensure that we have the best available information in a space-efficient manner.

The point selection is executed iteratively, using the vertices of the optimized pose graph. The first element, the root node of the pose graph, is used as the base of the point cloud. The information in this element, the data structure mentioned in the introduction to Section 2, is converted into several point clouds, creating one with color information, a second with class information, a third with probability information, and the last with node probability information. These point clouds, for performance reasons, are downsampled via voxelization to 1 mm.

By creating these four different point clouds, we have the various types of information mapped against the depth information. These point clouds are converted into five separate lists named *positions*, *colors*, *classes*, *probabilities*, and *node probabilities*. With the first element of the pose graph being used as a base for the final point cloud, the initial lists contain all the points in the element. Starting with the second iteration, it is first checked if the normal of the translation lies within a threshold (see Section 2.1). If beyond the threshold, it means that the instance is not correctly aligned with the rest of the vertices and is discarded.

If it is inside the threshold, a filter based on the point-to-point distance is used, where points are divided into two: points not present in the final point list and those already present. The points that are not present are added to the final point lists. For the present points, we check if the new probability the neural network inference provides is higher than the current one. If it is higher, we replace the existing point information with the new information; else, we discard the new information. This process ensures that the final point cloud contains the most probable information for each point. An example of the final point cloud can be seen in Fig. 4. While the system can see the adjacent pruning regions, it only considers the target pruning region in the pruning point generation.

2.3. Point clustering

Once we have obtained the final point lists through merging, the next step is to organize the points into distinct instances using a clustering method. This allows us to differentiate different objects of the same class in the final created point cloud.

Two methods were tested, DBSCAN (Density-Based Spatial Clustering of Applications with Noise) (Ester et al., 1996) from the Open3D library (Zhou et al., 2018) and HDBSCAN (Hierarchical Density-Based Spatial Clustering) (Malzer and Baum, 2020).

DBSCAN relies on two parameters: *eps* controls the distance to neighbors for forming a cluster, and *min_points* specifies the minimum number of points required to create a cluster. With the parameter *eps*, it is possible to define how close the points are for them to be considered a single cluster. However, this leads to information being deleted because specific points are not associated with a cluster, for example, in areas where the list of points is more sparse.

HDBSCAN is an enhanced version of DBSCAN that only requires the *min_points* parameter. It automatically adapts the *eps* value based on the data distribution, which allows the clustering to be more adapted to the created point cloud. The adaptability based on the data distribution led to the selection of HDBSCAN as a clustering method.

By clustering the points, we can identify different instances of organs and generate individual point clouds for each organ instance.

Table 2

Types of pruning cuts considered in the study.

Cut type	Description
<i>Base-bud cut</i>	The base-bud is kept while the remaining (upper) part is removed.
<i>Spur cut</i>	The first <i>N</i> nodes of the basal cane are left, while the rest of it is removed. The <i>spur parent</i> organ is cut above the basal cane if it is not a new pruning region.
<i>Replacement cut</i>	The first <i>N</i> nodes of the basal cane are left, while the rest of it is removed. The <i>arm parent</i> organ is cut above the basal cane.

Table 3

Relationships between the different types of organs.

Parent organ type	Children organs types
<i>Cordon</i>	<i>Arm, Spur, Cane</i>
<i>Arm</i>	<i>Spur, Cane</i>
<i>Spur</i>	<i>Cane</i>
<i>Cane</i>	<i>Node</i>

This step is crucial as it enables us to analyze each organ separately, facilitating estimations such as organ thickness, orientation, and position relative to the rest of the plant. These estimates provide valuable information for deciding which parts of the plant to retain or remove.

2.4. Plant graph creation

After the creation of the segmentation instances described above, the last step is creating the plant graph structure similar to the one introduced in our previous approach (Fernandes et al., 2021).

The graph structure is tree-shaped, considering the *cordon* instances as the root node and the *node* instances as the leaf nodes. This allows the skeletonization of the detected organs and keeps track of information used for the pruning point generation algorithm. The significant difference between our previous implementation and the current version is the adaptation from the 2D into the 3D domain.

The organs are correlated to each other as defined in Table 3, where the *cordon* connects to *arms*, *spurs* and *canes*, the *arm* connects to *spurs* and *canes*, the *spur* connects to *canes*, and the *cane* contains *nodes*.

To understand if the organs are connected, we check if there is significant proximity to the parent organ. This is achieved by calculating the percentage of points of the child organ that are closer than 1 mm to the parent organ.

2.5. Agronomic pruning point generation

In this paper, we will focus on the following three main types of cuts: base-bud cut, spur cut, and replacement cut. These operations normally involve making two physical cuts on the pruning region. The corresponding cuts are described in Table 2.

The base-bud cut is performed above the base bud and removes the old spur.

The replacement cut involves two physical cuts: first, the replacement cane is shortened to the desired count node. Second, the removal of the old spur and arm of the pruning region.

The most common pruning cut in grapevines trained to a VSP spur pruned cordon is the Spur cut, where the first physical cut is placed above the *N*th count node of the basal cane, and the second physical cut removes the old spur. The number of count nodes to be kept is a parameter that can be adjusted depending on grapevine variety, balance, and target yield (Tassie and Freeman, 1992; Poni et al., 2004).

With this knowledge, we can process the generated point clouds and plant graph structure into pruning points that the system is expected to prune. This follows a simplified ruleset, targeting a pruning region, as described next.

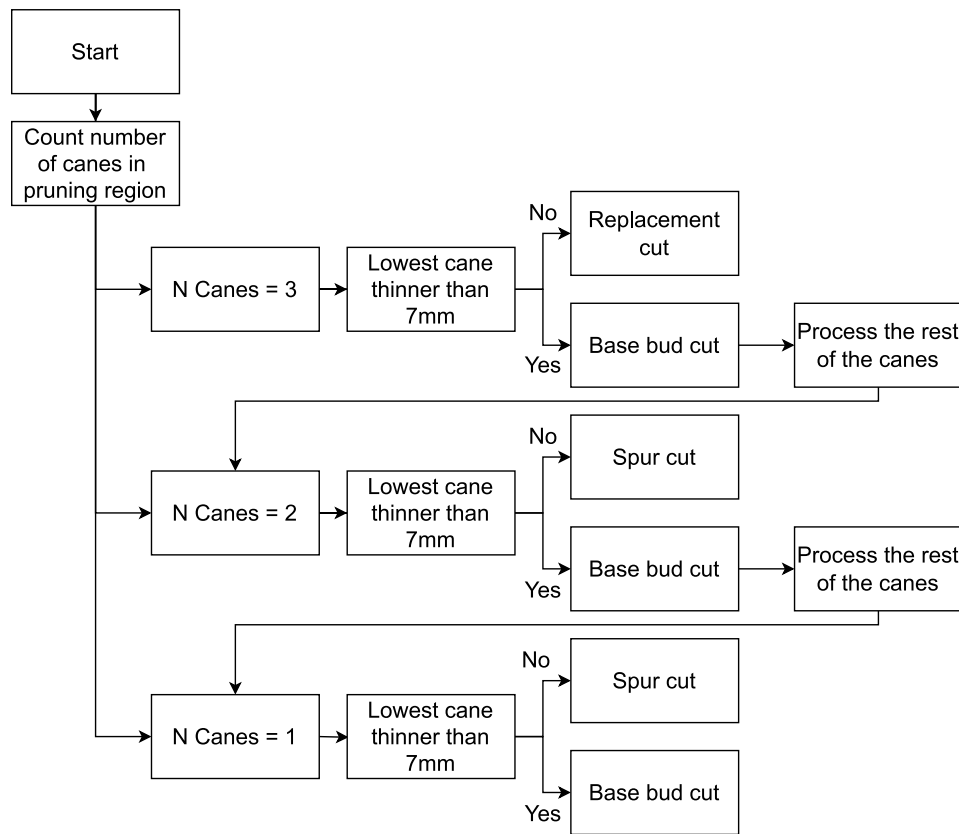


Fig. 5. Pruning decision flowchart used by the system.

Our algorithm detects pruning points using a threshold value for cane thickness (too-thin dormant shoots are removed) and the number of retained nodes on the next year's spur. Despite the development of a customizable system, we are currently setting the thickness threshold to 7 mm (Tassie and Freeman., 1992) and the number of nodes to two for *spur cut* and *replacement cut*, while only one node is retained in case of *base-bud cut*.

We begin by counting the number of canes in the pruning region, which can usually go from one up to three.

If there is only one cane, we measure its thickness. If it is thinner than 7 mm, we perform a *base-bud cut*, keeping only one node. If it is thicker than 7 mm, we perform a *spur cut*, retaining two nodes.

If there are two canes, we start evaluating the thickness of the lowest cane. If it is thinner than 7 mm, we perform a *base-bud cut* on that cane and move to the next cane, applying the rule already introduced for a spur region with one cane. If the lower cane is thicker than 7 mm, we perform a *spur cut*, retaining the first two nodes and removing the rest of the pruning region.

Finally, in the presence of three canes, if the lowest cane is thicker than 7 mm, we perform a *replacement cut*, keeping the first two nodes and removing the rest of the pruning region. If, instead, it is thinner than 7 mm, we perform a *base-bud cut* on that cane, shortening it to one node and processing the rest of the pruning region following the behavior presented above for the two-can case. Fig. 5 shows in a schematic way the decision process taken by the system.

2.6. Pruning point generation

As mentioned above, the presented algorithm requires extracting a few metrics among the possibilities, thanks to a 3D model of the plant:

the number of canes in a pruning region, cane thickness, and their location.

While the number of canes and their location are directly obtained from the 3D model, the cane thickness has to be calculated. For our currently implemented estimation, we divided the cane into 20 vertical segments, and for each of them, we compute the Euclidean distance between the lateral edges. Distances considered outliers were eliminated, and the remaining values averaged for the final cane thickness estimation. The left part of Fig. 6 shows an example of the estimation procedure.

One of the main contributions and major advantages of the proposed system is the flexibility to customize the pruning parameters. End users can therefore adjust various parameters to meet their specific requirements. For instance, parameters such as the number of retained nodes, desired cane thickness can easily be modified in response to varietal differences, vineyard vigor, yield targets, etc.

For now, the pruning points are placed at the middle point of the line created by the elements being considered for the pruning. For example, when cutting a cane, the elements being considered are the desired count node and the node above it, with the pruning point placed on the midpoint between the two nodes.

In another example, for a pruning point placed on a spur, the elements being considered are the two canes, where the pruning point is placed on the midpoint between the lower extremities of the two canes. A visual explanation can be seen in Fig. 6.

3. Material & methods

3.1. Robot prototype

The mobile manipulator robot used for this study was a Robotnik Summit XL four-wheeled robot with a Kinova Gen3 arm with seven

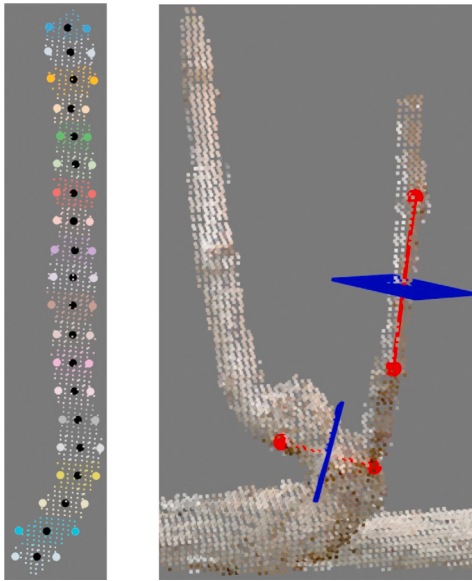


Fig. 6. The figure on the left demonstrates the thickness estimation of a cane. The multiple colors are the twenty subdivisions made of the cane, with the spheres corresponding to the extremities of the corresponding color segment. The black spheres correspond to the center of the segment. The thickness value of a segment is given by the Euclidean distance between the two extremities. The figure on the right shows the pruning point placement; the blue markers represent the pruning point, and the red spheres show the considered organ points. The red spheres are located in the nodes' center for the upper pruning point. For the lower pruning point, the red spheres are placed in the lower extremity of the segmented area of the two canes of the pruning region.

degrees of freedom (DoF), as can be seen in Fig. 7. The pruning end effector consists of electric shears and an off-the-shelf, low-cost RGB-D camera (Intel RealSense D405). This mobile manipulator is agile and suitable for outdoor environments. The current version of the pruning end effector was designed to simplify the depth camera calibration by placing the camera in line with the arm end effector mount. Moving the shears partially underneath and parallel to the last link of the arm allows the tip of the shears to be closer to the arm end effector mount, which benefits the overall dexterity of the robot arm.

3.2. Execution pipeline

The execution pipeline, shown in Fig. 8, starts by deploying the robotic manipulator and performing the block “Identify Pruning Regions” using the neural network (Guadagna et al., 2023). It selects the inferences that are spurs or arms since these make up the pruning regions (see Section 1.1 for more details on the plant organ classes).

The robotic manipulator then approaches the first detected pruning region and executes the block “Plan movement for pruning region scan”, which corresponds to the task described in Section 3.3.1, and while the manipulator is moving, the system captures data for the point cloud creation module, represented by the blocks related to the activation and deactivation of “Data Capture Worker”.

After finishing the movement, the captured data is processed in the green section of the pipeline, named “Segmented Point Cloud”, where the individual blocks are described in Section 2.

After creating the segmented point cloud, the system follows the logic presented in the yellow section of the pipeline, named “Pruning”. The system first plans an arm movement that brings the shears to the pruning point, then executes the motion, performs the cutting, and repeats these actions as long as pruning points are available. When no more pruning points are available, the task is considered finished, and the robot moves to the next available pruning region.

3.3. Arm planning and movement

This section explains the arm motion planning and movements. MoveIt Task Constructor (MTC) (Görner et al., 2019) is a framework that simplifies the process of planning, executing, and monitoring complex manipulation tasks for robots, by providing a high-level interface for task planning and execution. Our pruning pipeline uses three different tasks. The first task scans a pruning region to acquire the information needed to capture the data to build the point cloud, see Fig. 9. The second task, shown in Fig. 10, uses a pruning point and plans a movement that allows the engagement of the pruning shears in the desired point. The third task entails only disengaging from the plant and moving to a known arm configuration, such as the fold position, where the arm is folded on top of the mobile robot safely, allowing the mobile robot to move without damaging the robot arm.

Each task will be more thoroughly explained in the following subsections, but they share some parameters; for example, their cost function is not a metric of time but rather a metric of how much the end effector moved. This way, the MTC plan selector uses movements that are close to the plant, preventing undesired movements away from the plant. The selection process for the generated plan is also common to the tasks, where we select the plan with the lowest cost by sorting the valid plans by cost and discarding the invalid plans.

3.3.1. Task N°1 - Plan to scan

This task creates a plan that scans the selected pruning region. The scanning motion, which can be seen in Fig. 9 involves the following poses: The robotic arm initially orients itself toward the designated pruning area and executes a specific motion pattern centered around the target object within the image. This motion comprises a sequence of distinct poses involving rotational adjustments. In other words, the goal is to keep the camera always at the same distance from the plant while acquiring images from different points of view. It achieves this by performing rotations on the end effector, that keeps the target pruning region in the center of the image.

Using the positions presented in Fig. 9, the motion in position 1, in front of the spur. The motion commences with a diagonal right-up rotation to reach point 2 and continues to point 7 with a yaw rotation. The robot pitches downwards from point 7 to 8, followed by a yaw rotation from point 8 toward point 13. Then, the robot pitches upwards toward point 14 and finishes the movement with a left-down rotation, arriving at point 15.

Considering that Cartesian planning can sometimes fail in more complicated motions, MoveIt Task Constructor allows the usage of *Alternatives*, which enables the planning with multiple types of planners, where in this case, it plans using the Cartesian planner, joint interpolation planner, and sampling planner. By using these planners in parallel and selecting the one with the lowest cost, the final motion plan has a combined lower cost, where the cost metric corresponds to the smallest joint movement. In short, Cartesian planning is used with desired positions and orientations of the end effector using Cartesian coordinate space (X, Y, Z). Joint interpolation planning involves specifying the desired positions for each robot arm joint. The planner then computes a trajectory that smoothly interpolates between the initial and final joint configurations. Finally, the sampling planner randomly samples configurations and constructs a graph or tree connecting the sampled configurations to find feasible paths for a robot to move from its initial state to the goal state. It is important to note that while the robot is performing this scanning motion, it is capturing images for 3D point cloud reconstruction. This image capture occurs once per second.

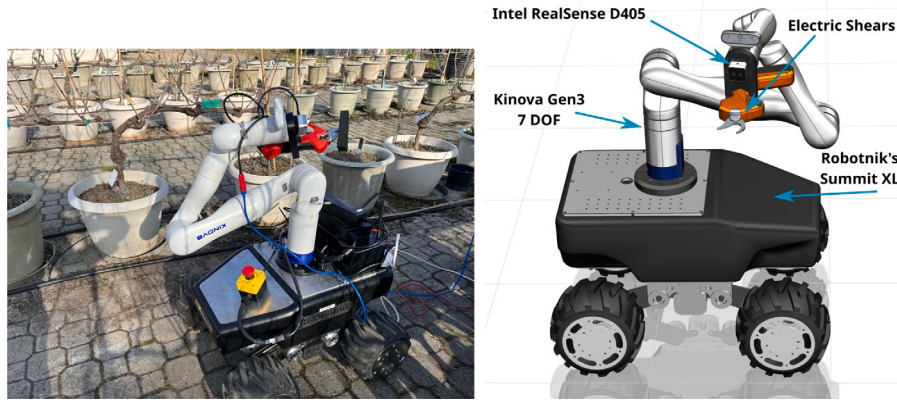


Fig. 7. Robotnik's Summit XL wheeled robot with Kinova Gen3 seven DoF manipulator arm with pruning shears as end effector. The left side of the figure shows the complete platform, and the right side of the figure presents a labeled CAD rendering of the robot.

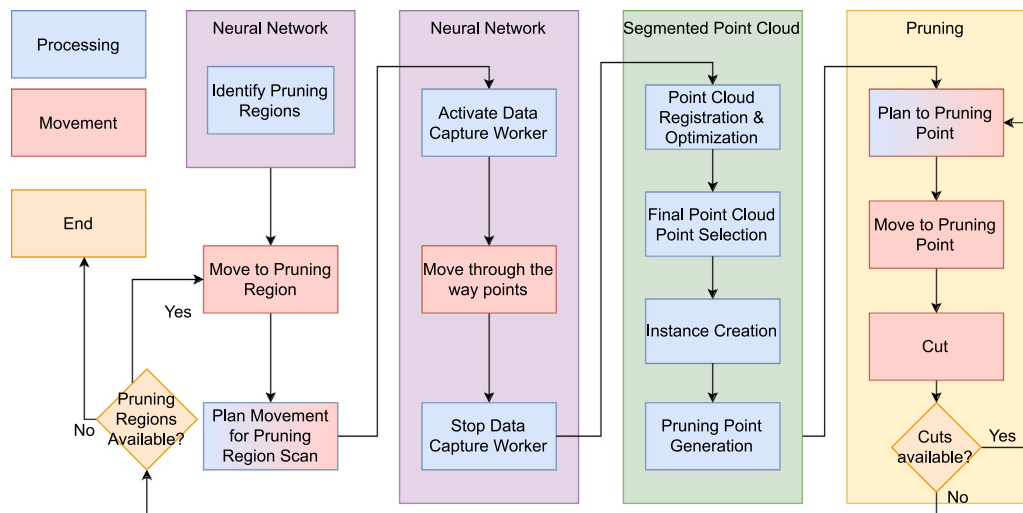


Fig. 8. Pipeline of the system execution loop, from the initial navigation step up to the cutting. The colored blocks on the top left named "Processing" and "Movement" act as a legend for the rest of the blocks of the figure.

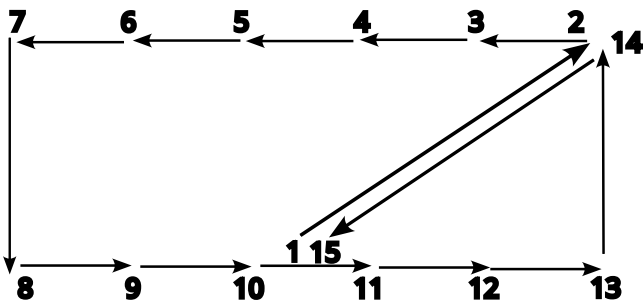


Fig. 9. The image shows in a schematical way the motion pattern taken by the scanning motion. It is important to notice that positions 1 and 15, along with positions 2 and 14, are the same positions and that positions 1 and 15 are located between positions 10 and 11.

3.3.2. Task N° 2 - Plan to cut

Plan to Cut is the task used to plan a movement to a pruning point, as seen in Fig. 10. It receives the pruning point we desire to prune, which is a point rather than a pose due to the fact that from the vision side of the system, there is an abstraction of the used arm and end effector.

The difference between a point and a pose in this situation is the fact that a point sets the end-effector position to given (X, Y, Z) coordinates,

and a pose additionally specifies an orientation, normally in quaternion format (X, Y, Z, W) .

This abstraction allows us to use the same concepts the grasping research community uses. This is because pruning can be considered as a grasping task, from the robotic point of view; instead of grasping, we cut the intended target. In this sense, we explore the availability of the plant's 3D model to not only find the correct pruning point but to constrain the manipulator's motion while pruning; in this regard, the algorithm finds the motion to engage the shears in the desired pruning location avoiding any collision with the plant. To add the 3D plant model to the planning scene, we provide the created point cloud to MoveIt's 3D perception module, which is responsible for updating an occupancy map based on Octomap (Hornung et al. 2013). The latter is used for collision avoidance. Due to the need to validate the feasibility of the end of the motion, the plan is calculated inversely from the execution. With this, the first thing this task does is to verify via the inverse kinematics if the end effector, or, more specifically, the center point of the shear blades, can be inserted into the provided pruning point. This discards the positions of the end effector that either touch the plant or try to approach the pruning point from the opposite side of the plant. These end effector poses are obtained by searching around the provided point, with an angular delta around the three coordinate frames, X , Y and Z , and by using code from MTC related to grasping tasks. After finding valid pruning poses, the planner is configured to drive the manipulator toward the pruning location and executes a final

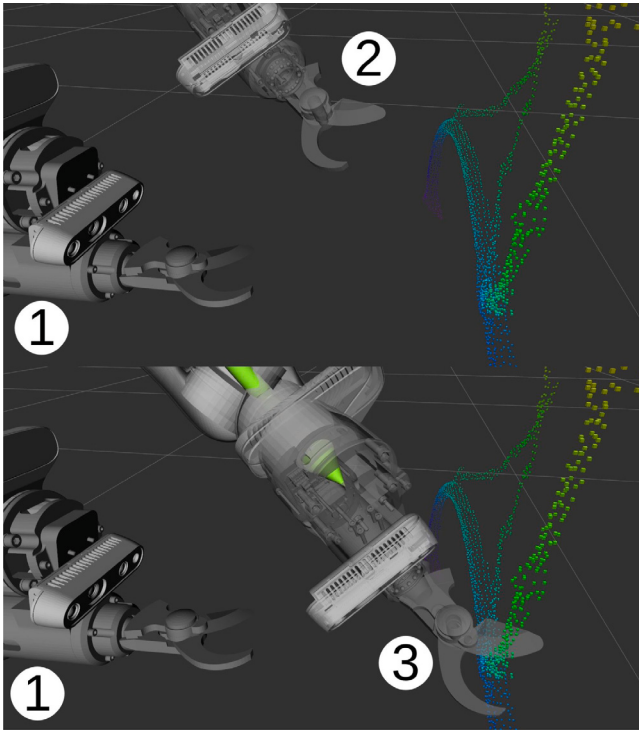


Fig. 10. Task plan to cut in parts, where the ghosts are positions where the robot will pass. The robot starts its motion in position 1, it then moves to position 2, which acts as the pre-engagement position. From position 2 the robot starts its linear approach to the pruning point, finishing the movement in position 3.

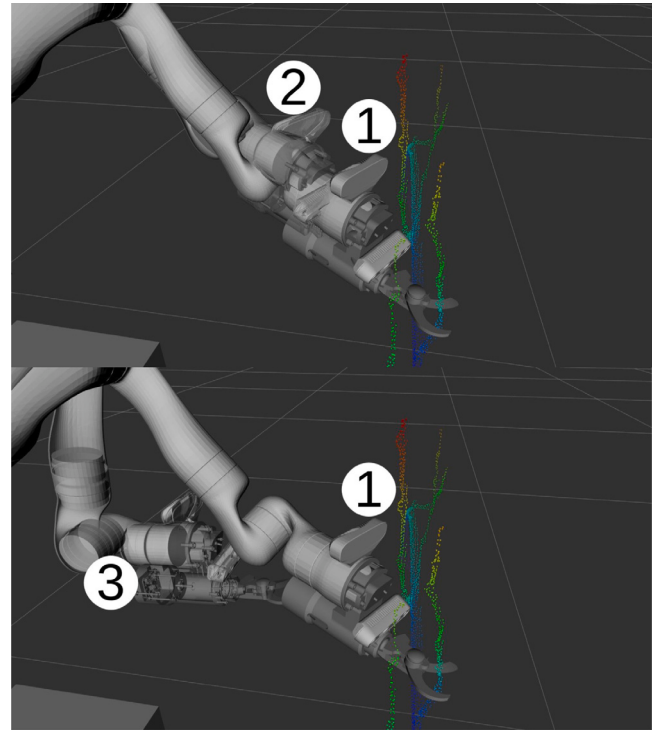


Fig. 11. Task plan back in parts, where the ghosts are positions where the robot will pass. The motion starts in position 1 where the robot disengages from the pruning point, moving linearly toward position 2. The robot then moves toward position 3, returning to the detection position.

translation to avoid any collision while the manipulator tries to engage the plant. This is first planned with Cartesian planning, which falls into joint interpolation if the former fails. The last part plans the motion from the current state of the arm to the position of the beginning of the relative engagement movement.

3.3.3. Task N°3 - Plan back

This task, shown in Fig. 11, uses as input a set of joint positions, which are generally the joint positions when the robotic arm is moved to the detection position or is folded on top of the robot, where the former is used for moving to the next pruning point or next pruning region and the latter to stow the arm during platform navigation. The task comprises two phases, where a relative movement is done backward, in a range of 5 to 15 cm, selected by the motion planner, followed by the motion to the provided joint positions.

4. Pruning experiments and results

The vision pipeline and pruning point selection explained in the previous sections were tested in the winter 2022/2023 season, using the robot platform presented in Section 3.1.

4.1. Experimental setup

The tests were executed on six mature potted grapevines at the Università Cattolica del Sacro Cuore in Piacenza (Italy) for a total of 24 available pruning regions. As reported by Guadagna et al. (2023), the vines were aligned in NE-SW oriented row, trained to a spur-pruned cordon since 2017 with five 2-node spurs and a vine spacing of 0.9 m. Some of the plants used in the experimental validation were pruned in the previous season by an earlier version of our pruning system without assessing the correctness of the cuts. Other plants were pruned by hand. Since some of the robot-cut plants have been incorrectly pruned, some

of the evaluated pruning regions are considered more complex than usual.

The execution process is the same as presented in Section 3.2. It is important to note that even though the system can also perceive the neighboring pruning regions, we only consider the pruning points that affect the scanned pruning region.

4.2. Experimental results overview

The experimental results for our system are split into three different components, the first addressing the split between used and discarded images for the point cloud 3D reconstruction (Section 4.3), the second component targeting the system's detection capabilities (Section 4.4), and the last component evaluating the calculated pruning points (Section 4.5).

4.3. Point cloud merging results

As presented in Section 2.1, the system captures a stream of images that are converted into point clouds that are used to create the 3D reconstructions. Table 4 presents the data usage related to the reconstruction process, with the mean and standard deviation for the number of images used to create the merged point clouds. These values are the number of images that were captured, the images that were not matched during the pose graph creation, the images that were discarded during the point cloud reconstruction due to their transformations, the number of used images for the reconstruction process. The last row contains the mean and standard deviation of the number of points of the merged point clouds.

Table 4

Point cloud reconstruction data usage, per pruning region, considering the mean and standard deviation of the number of images for each processing step. The merged point cloud presents the same metrics but for points.

	Mean	SD
Captured	61.6	7.18
Not matched	12.75	11.90
Discarded	11.2	7.18
Used	37.6	12.83
Merged point cloud size	40724.76	8860.66

Table 5

Point cloud detection results, cane detection evaluation, where the *incorrect clustering* entails canes incorrectly merged into one instance. In one case of incorrect clustering in this situation, the cane was merged into an adjacent pruning region, where this evaluation only targets the scanned pruning region, leading to the odd number of incorrect clusters. This situation can be seen in Fig. 15.

	Canes	Percentage
Correct detection	49	79.2%
Incorrect clustering	5	7.27%
Missed detection	1	1.82%
Total	55	100%

Table 6

Point cloud detection results, where the cane detection is evaluated per pruning region. By evaluating per pruning region, we can understand how the cane detection affects the resulting pruning point generation. The entire pruning region is classified as incorrect clustering or missed detection, regardless of which cane lead to that classification.

	Pruning regions	Percentage
Correct detection	19	79.2%
Incorrect clustering	4	16.7%
Missed detection	1	4.2%
Total	24	100%

4.4. Point cloud detection results

Taking the created point cloud reconstructions, we are evaluating the accuracy of the detection, targeting the relevant organs for winter pruning, i.e., the *canes* and the *nodes*. Tables 5, 6, 7 and 8 present the results for the detections related to the *canes*, *canes per pruning region*, *nodes*, and *nodes per cane* respectively.

The classification of the detections of these organs can be split into three possible conditions, *correct detection*, *incorrect clustering*, and *missed detection*.

- The *correct detection* categorizes the detection as correct and matching the ground truth.
- The *incorrect clustering* categorizes the result as positive detection of the organ, but an incorrect clustering, merging two separate instances into just one.
- The *missed detection* states that the organ was not seen or was classified as the wrong class.

Table 5 shows the results related to cane detection, and Table 6 contextualizes the cane detection results per pruning region, where each pruning region has two to three canes. Although our algorithm also considers the existence of pruning regions with just one cane, there are no pruning regions of that type in our data.

Table 7 presents the results related to the node detection in a global manner, with Table 8 contextualizing these results with their corresponding cane type.

We standardize the maximum number of nodes to three for this evaluation since the correct pruning points are placed between the second and third nodes, according to the metrics established in Section 2.5.

To better understand the previous tables, we calculate several evaluation metrics, *Precision*, *Recall*, and *F1-Score*, summarized in Table 9, using the following equations adapted to our evaluation categories:

$$precision = \frac{correct\ detections}{(correct\ detections + incorrect\ clustering)} \quad (1)$$

Table 7

Point cloud detection results, node detection evaluation. Incorrect clustering entails pairs of nodes that were merged into one instance. This evaluation considers the ground truth quantity of *canes* (55).

	Node	Percentage
Correct detection	123	79.2%
Incorrect clustering	26	7.27%
Missed detection	16	1.82%
Total	165	100%

Table 8

Point cloud detection results, *node* detection evaluation per cane type (Basal, Distal or Replacement). This evaluation considers the ground truth quantity of *canes* (55).

	Basal	Distal	Replacement	Total	Percentage
Correct detection	17	12	3	32	58.2%
Incorrect clustering	4	7	2	13	23.6%
Missed detection	3	5	2	10	18.2%
Total	24	24	7	55	100%

Table 9

Performance metrics (Precision, Recall, and F1-score) comparing different categories: Canes and nodes. The results are calculated using Eqs. (1), (2), and (3). It distinguishes between the general performance of cane and node classes, along with their specific evaluation, with canes being evaluated per pruning region (PR) and the nodes being evaluated per cane type, basal, distal, and replacement.

	Precision	Recall	F1-Score
Canes	90%	98%	94.2%
Canes per pruning region	82.6%	95%	88.3%
Nodes	82.6%	88.5%	85.4%
Nodes per basal canes	81%	85%	82.9%
Nodes per distal canes	63.2%	70.6%	66.7%
Nodes per replacement canes	60%	60%	60%

$$recall = \frac{correct\ detections}{total\ elements} \quad (2)$$

$$F1 - score = 2 * \frac{(precision * recall)}{(precision + recall)} \quad (3)$$

In Eq. (2), the total number of elements is relative to the specific class being evaluated, the canes or the nodes.

4.5. Pruning point generation results

The pruning points obtained by the method described in Section 2.5 are compared with the correct cut type (Table 2) that an expert pruner would apply. We evaluate if the selected cut type is correct and if the selected locations of the physical cuts are correct.

These results are classified into the following different classes: *correct cut*, *incorrect length*, *incorrect cane thickness estimation*, *incorrect cut*.

- The *correct cut*, as the name implies, is an agronomically correct pruning performed on the pruning region, see Fig. 12.
- The *incorrect length* is related to a correct cut, but with the error of improper shortening of the cane. In these cases, the system always kept three instead of two nodes, underestimating the quantity of *nodes* in the cane.
- The *incorrect cane thickness estimation* happens when the spur is set from a worse cane while the correct cane is removed.
- The *incorrect cut* is where the system misclassifies the cut type of a pruning region, leading to wrong pruning points.

Table 10 contains the quantitative results obtained by the system in the field tests, along with the information on whether the obtained case can be fixed by additional manual pruning. Note that the number of pruning regions does not correspond to the number of physical cuts since some of the cases, such as the *spur cut* or the *base-bud cut* imply performing two physical cuts, as mentioned in Section 2.5.



Fig. 12. Example of correctly generated pruning points, with the expert evaluation on the left side of the image, with the plant designation ID on the top left and the cut type designation on the bottom left, in this case, *Spur cut*. The right image shows the point cloud output, with the blue markers representing the expected cuts.

Table 10

Results obtained by the system during the 2022/2023 pruning season, along with information on whether additional manual pruning can fix the case.

	Pruning regions	Percentage	Can be fixed?
Correct cut	9	37.5%	No need
Incorrect length	4	16.7%	Yes
Incorrect cane thickness estimation	6	25.0%	No
Incorrect cut	5	20.8%	No
Total	25	100%	

Table 11

Statistical analysis pruning point evaluation results using a bootstrap Confidence Interval (CI) of 95%.

Class	Samples	Proportion	Lower CI	Upper CI	CI width
Correct cut	9	37.5%	20%	58%	38%
Incorrect length	4	16.67%	4%	33%	29%
Incorrect cane thickness estimation	6	25%	8%	41%	33%
Incorrect cut	5	20%	4%	38%	34%

Due to the low amount of pruning regions evaluated, an additional statistical evaluation is presented in Table 11, using a bootstrap confidence interval of 95%. From the presented table, we can see that the quantity of samples from the statistical analysis is low, leading to a high width between the bounds of the confidence intervals. The need to test the system in a higher quantity of pruning regions emerges. Due to the selected confidence interval, we are 95% confident that the true proportion of the correct cuts is located between 20% and 58%. The higher this proportion, the lower the other classes' proportions are since they are mutually exclusive.

Several conclusions can be drawn from the results that explain what caused those wrong results.

In the case of *incorrect length* (Table 10), all instances in this class are canes with three instead of two nodes. Too long canes can be corrected via additional manual pruning. Although this did not occur in this study, an incorrect length can also be the result of too severe pruning that leads to shorter spurs and, therefore, in a lower bud load than desired. In both cases, the incorrect length estimation of a spur will result in an array of undesired conditions, such as wrong bud load per vine and bud distribution per pruning region, leading to undesired plant responses the next season, including unbalanced bud-burst ratio, vegetative gradients, and variable shoot vigor. This

information is shown on Table 7, where the number of *basal canes* that had *incorrect clustering* match with the cases that have *incorrect length* (see Fig. 13).

Regarding the *incorrect cane thickness estimation*, these instances correspond to an incorrect estimation of cane thickness.

Lastly, the *incorrect cut*, are pruning points that are wrong, due to incorrect processing of the pruning region. An example can be seen in Fig. 15 where canes from adjacent pruning regions are merged into one.

5. Discussion

In this section we discuss the achieved results and we highlight the limitations of our approach.

Due to overly conservative collision constraints in the arm planning algorithm, the autonomous system could not perform any of the spur cuts. For these cuts, the operator was manually guiding the robot end effector to the pruning location proposed by the algorithm.

Considering the qualitative evaluation of the results, the positive success rate, including both entirely *correct cuts* and *incorrect length cuts* that can be fixed, was 54.2%.

To provide context to the related work in the field of automated winter pruning of grapevines, a comparison to the most relevant works is needed. Both (Botterill et al., 2017) and Silwal et al. (2022), and our work target pruning of grapevines, however using different imaging techniques, using different platforms, and most importantly, different grapevine varieties (Sauvignon Blanc, Concord, Sangiovese), training systems (cane pruning, bilateral cordon on high-wire sprawl canopies, spur pruning on hedge trained vines).

Botterill et al. (2017) uses AI to place the pruning points, selecting two long canes to keep and two canes that are shortened at 10 cm. While Silwal et al. (2022) achieves an 87% accuracy, it is worth highlighting that their approach uses a simple rule of keeping four nodes per cane and their study only treated a subset of canes considered prunable after a pre-pruning operation. In contrast, the results we present go a step further by providing an additional understanding of grapevine physiology, using measurements obtained from the plant to make decisions, and following customizable parameters that can be tuned according to enological objectives.

By analyzing the wrong cuts and the reason that led to the incorrect evaluation, the system can be improved, where we evaluate each erroneous class.

Starting with the *incorrect length* class, we can identify that the problem in these specific cases is either the wrong clustering (Section 2.3) of

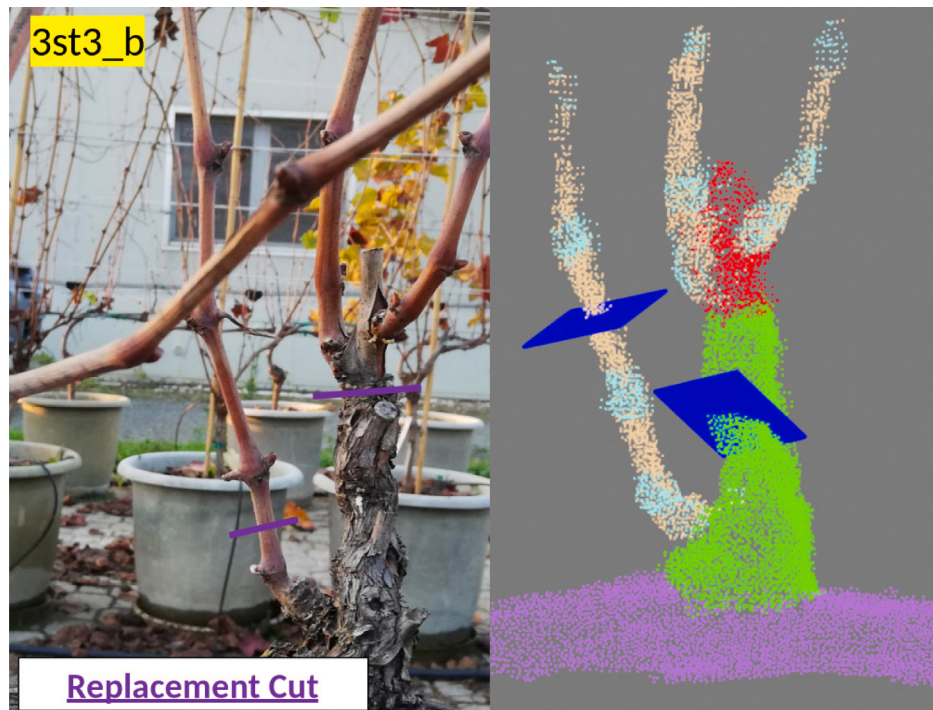


Fig. 13. Example of an *incorrect length cut*, with the expert evaluation on the left side of the image, with the plant designation on the top left, with the cut type designation on the bottom left, in this case, *Replacement Cut*. The right image shows the point cloud output, with the blue markers representing the expected cuts.



Fig. 14. Example of an *incorrect cane thickness estimation cut*, with the expert evaluation on the left side of the image, with the plant ID on the top left, with the cut type designation on the bottom left, in this case, *Base Bud Cut* and *Spur Cut*. In this example, the overestimation of the thickness of the replacement cane leads to the system (wrongly) performing a replacement cut.

nodes close to the base of the cane. This leads to the false identification of two nodes as just one instance or the nonidentification of one of the present nodes, making the system think that it is pruning between nodes 2 and 3, while it is pruning between nodes 3 and 4.

We consider the main issue that causes *incorrect cane thickness estimation* is the error of the camera itself, which has a depth estimation error of 2% at a distance of 50 cm (according to the datasheet). This causes the system to have issues with estimating the thickness of the canes, leading to several cases of performing the wrong cut altogether. Using the same error of 2% at the normal range of operation of 30 cm corresponds to an error of ± 6 mm. However, as part of a selective task the canes are one of the organs that need to be measured to decide if they are meant to be kept or removed from the plant, where we keep

the canes thicker than 7 mm. In the specific situation of the presented results, the six *incorrect cane thickness estimation* cases were caused by the inaccurate cane thickness estimation, which was done by keeping a cane thinner than seven millimeters and pruning the rest of the region. An example of this case can be seen in Fig. 14. Unfortunately, during the execution of these field trials, we did not capture ground truth data from the evaluated grapevine pruning regions that allow a qualitative evaluation of the thickness measurements. With this information, a future study on mitigating or removing this estimation error is needed, either by further calibration or changing how the thickness estimation is done.

For the *incorrect cut*, this case can be seen as the failure of one or multiple parts of the system, where the detection, clustering, or

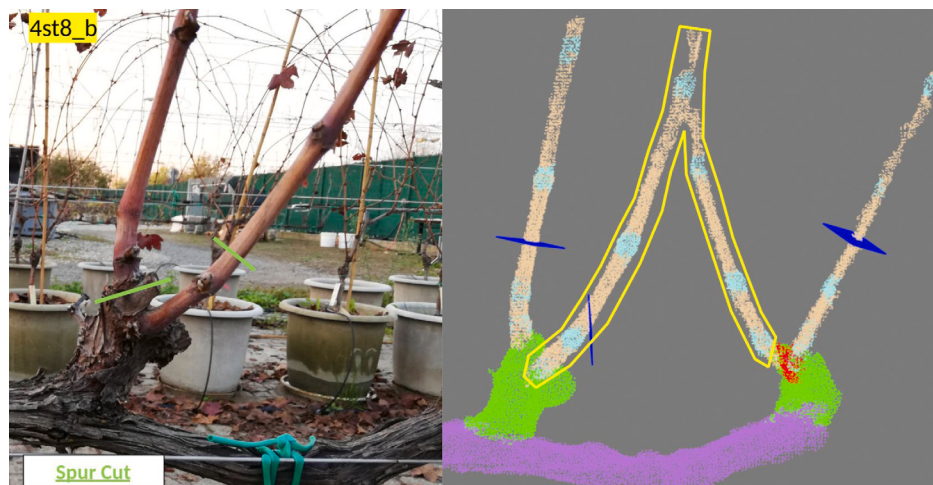


Fig. 15. Example of an *incorrect cut type*, with the expert evaluation on the left side of the image, with the plant ID on the top left, with the cut type designation on the bottom left, in this case, *Spur cut*. This cut, in particular, is also an example of an incorrect clustering of a cane, with the yellow bounding box representing the two incorrectly merged canes.

estimation components failed. This leads to incorrect detections, such as *cane* segments being detected as part of the *spur* or *arm* and *spur* segments having conflicting matches, or *canes* from adjacent pruning regions (Fig. 15) being merged together.

6. Conclusions

This paper presented an automated winter pruning system that uses a simplified robotic platform equipped with a robotic arm, pruning shears, and a depth camera. The presented system performed automated winter pruning in the winter season 2022/2023, using a parametrized approach that aims at the implementation of sophisticated pruning criteria, leading to a balanced pruning. The main strength of this system over the current state of the art in grapevine winter pruning is the step further we take by providing an additional understanding of grapevine physiology, using measurements obtained from the plant to make pruning decisions, and following customizable parameters that can be tuned according to enological objectives. In particular, in order to compensate for camera occlusions, we introduced a 3D reconstruction algorithm of the spur region, by merging 2D segmented images. We tested our approach on 24 pruning regions of six mature potted grapevines; we have evaluated the accuracy of the 3D reconstruction (number of identified canes and nodes) and the correct identification of the pruning points. We presented a statistical analysis to compensate for the low amount of evaluated pruning regions.

To achieve a fully automated pruning platform, several aspects of the systems need to be improved, such as autonomous navigation in the vineyard, arm manipulation, safe end-effector engagement, and improvements in grapevine reconstruction, recognition, and interpretation.

Future work will address the currently low sample quantity by incorporating more comprehensive testing, targeting a higher quantity of pruning regions. In addition, to improve the thickness estimation process, methods such as Brown et al. (2024) will be evaluated. We also consider that the data captured from each trial can be improved by having ground truth measurements of the pruning regions, which will allow the evaluation of our 3D reconstruction, including, for example, the cane thickness estimation. Additionally, improvements in the clustering method are required to address the issue of wrongly clustered organs. Finally, the over-conservative collision constraints in the robot arm planning algorithm need to be solved to allow the system to autonomously prune the entire grapevine.

CRediT authorship contribution statement

Miguel Fernandes: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Juan D. Gamba:** Writing – review & editing, Software, Methodology, Investigation. **Francesco Pelusi:** Writing – review & editing, Validation, Formal analysis. **Angelo Bratta:** Writing – review & editing, Supervision, Formal analysis. **Darwin Caldwell:** Project administration, Funding acquisition. **Stefano Poni:** Project administration, Funding acquisition. **Matteo Gatti:** Writing – review & editing, Validation, Supervision, Project administration, Methodology, Funding acquisition, Data curation, Conceptualization. **Claudio Semini:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We would like to thank João Carlos Virgolino Soares for his contributions during the writing of this paper.

The research was carried out within the UCSC-IIT Joint Laboratory (Vinum Project). The study was co-funded by the Italian Ministry of University and Research, PRIN 20172HHNK5 Project.

This work is also part of the “Technologies for Sustainability” Flagship program of the Italian Institute of Technology (IIT).

Additional funding by the European Union - NextGenerationEU and by the Ministry of University and Research (MUR), Italy, National Recovery and Resilience Plan (NRRP), Italy, Mission 4, Component 2, Investment 1.5, project “RAISE - Robotics and AI for Socio-economic Empowerment” (ECS00000035)

Data availability

The data that has been used is confidential.

References

- Botterill, T., Paulin, S., Green, R., Williams, S., Lin, J., Saxton, V., Mills, S., Chen, X., Corbett-Davies, S., 2017. A robot system for pruning grape vines. *J. Field Robot.* 34 (6), 1100–1122. <http://dx.doi.org/10.1002/rob.21680>, arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21680> URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21680>.
- Brown, J., Paudel, A., Biehler, D., Thompson, A., Karkee, M., Grimm, C., Davidson, J.R., 2024. Tree detection and in-row localization for autonomous precision orchard management. *Comput. Electron. Agric.* URL: <https://www.youtube.com/watch?v=VH3EroQQak&t=11s>.
- Chen, Y., Medioni, G., 1992. Object modelling by registration of multiple range images. *Image Vis. Comput.* 10 (3), 145–155. [http://dx.doi.org/10.1016/0262-8856\(92\)90066-C](http://dx.doi.org/10.1016/0262-8856(92)90066-C), URL: <https://www.sciencedirect.com/science/article/pii/026288569290066C>.
- Coleman, D., Şucan, I.A., Chitta, S., Correll, N., 2014. Reducing the barrier to entry of complex robotic software: a MoveIt! case study. *J. Softw. Eng. Robot.* 5 (1), 3–16, URL: <http://dx.doi.org/10.6092/JOSER.2014.05.01.p3>.
- Esser, F., Rosu, R.A., Cornelißen, A., Klingbeil, L., Kuhlmann, H., Behnke, S., 2023. Field robot for high-throughput and high-resolution 3D plant phenotyping: Towards efficient and sustainable crop production. *IEEE Robot. Autom. Mag.* 30 (4), 20–29. <http://dx.doi.org/10.1109/mra.2023.3321402>.
- Ester, M., Kriegel, H.-P., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. KDD '96*, AAAI Press, pp. 226–231, URL: <https://dl.acm.org/doi/10.5555/3001460.3001507>.
- Fernandes, M., Scaldaferrì, A., Fiameni, G., Teng, T., Gatti, M., Poni, S., Semini, C., Caldwell, D., Chen, F., 2021. Grapevine winter pruning automation: On potential pruning points detection through 2D plant modeling using grapevine segmentation. In: *11th IEEE International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (IEEE-CYBER 2021)*, pp. 13–18. <http://dx.doi.org/10.1109/CYBER53097.2021.9588303>, URL: <https://ieeexplore.ieee.org/document/9588303>.
- Gatti, M., Civardi, S., Bernizzoni, F., Poni, S., 2011a. Long-term effects of mechanical winter pruning on growth, yield, and grape composition of barbera grapevines. *Am. J. Enol. Vitic.* 62 (2), 199–206. <http://dx.doi.org/10.5344/ajev.2011.10101>, arXiv:<https://www.ajevonline.org/content/62/2/199.full.pdf> URL: <https://www.ajevonline.org/content/62/2/199>.
- Gatti, M., Civardi, S., Bernizzoni, F., Poni, S., 2011b. Potatura invernale. Tradizionale o meccanizzata? *Corriere Vinic.* (32), 17–20.
- Gentilhomme, T., Villamizar, M., Corre, J., Odobez, J.-M., 2023. Towards smart pruning: ViNet, a deep-learning approach for grapevine structure estimation. *Comput. Electron. Agric.* 207, 107736. <http://dx.doi.org/10.1016/j.compag.2023.107736>, URL: <https://www.sciencedirect.com/science/article/pii/S0168169923001242>.
- Görner, M., Haschke, R., Ritter, H., Zhang, J., 2019. MoveIt! Task constructor for task-level motion planning. In: *2019 International Conference on Robotics and Automation. ICRA*, pp. 190–196. <http://dx.doi.org/10.1109/ICRA.2019.8793898>.
- Guadagna, P., Fernandes, M., Chen, F., Santamaria, A., Teng, T., Frioni, T., Caldwell, D.G., Poni, S., Semini, C., Gatti, M., 2023. Using deep learning for pruning region detection and plant organ segmentation in dormant spur-pruned grapevines. *Precis. Agric.* 24 (4), 1547–1569. <http://dx.doi.org/10.1007/s11119-023-10006-y>.
- Häring, S., Folawiyo, S., Podguzova, M., Krauß, S., Stricker, D., 2024. Vid2Cuts: A framework for enabling AI-guided grapevine pruning. *IEEE Access* 12, 5814–5836. <http://dx.doi.org/10.1109/ACCESS.2024.3350432>.
- Hou, J., Dai, A., Nießner, M., 2019. 3D-SIS: 3D semantic instance segmentation of RGB-D scans. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. CVPR*, pp. 4416–4425. <http://dx.doi.org/10.1109/CVPR.2019.00455>.
- Intrieri, C., Poni, S., 1995. Integrated evolution of trellis training systems and machines to improve grape quality and vintage quality of mechanized Italian vineyards. *Am. J. Enol. Vitic.* 46 (1), 116–127, arXiv:<https://www.ajevonline.org/content/46/1/116.full.pdf> URL: <https://www.ajevonline.org/content/46/1/116>.
- Malzer, C., Baum, M., 2020. A hybrid approach to hierarchical density-based cluster selection. In: *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems. MFI*, pp. 223–228. <http://dx.doi.org/10.1109/MFI49285.2020.9235263>.
- Poni, S., Bernizzoni, F., Presutto, P., Rebucci, B., 2004. Performance of Croatia under short-cane mechanical hedging: A successful case of adaptation. *Am. J. Enol. Vitic.* 55 (4), 379–388. <http://dx.doi.org/10.5344/ajev.2004.55.4.379>, arXiv:<https://www.ajevonline.org/content/55/4/379.full.pdf> URL: <https://www.ajevonline.org/content/55/4/379>.
- Poni, S., Frioni, T., Gatti, M., 2023. Summer pruning in Mediterranean vineyards: is climate change affecting its perception, modalities, and effects? *Front. Plant Sci.* 14, <http://dx.doi.org/10.3389/fpls.2023.1227628>, URL: <https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2023.1227628>.
- Poni, S., Gatti, M., Palliotti, A., Dai, Z., Duchêne, E., Truong, T.-T., Ferrara, G., Matarrese, A.M.S., Gallotta, A., Bellincontro, A., Mencarelli, F., Tombesi, S., 2018. Grapevine quality: A multiple choice issue. *Sci. Hort.* 234, 445–462. <http://dx.doi.org/10.1016/j.scienta.2017.12.035>, URL: <http://www.sciencedirect.com/science/article/pii/S0304423817307586>.
- Poni, S., Tombesi, S., Palliotti, A., Ughini, V., Gatti, M., 2016. Mechanical winter pruning of grapevine: Physiological bases and applications. *Sci. Hort.* 204, 88–98. <http://dx.doi.org/10.1016/j.scienta.2016.03.046>, URL: <http://www.sciencedirect.com/science/article/pii/S0304423816301650>.
- Santos, T.T., de Souza, L.L., dos Santos, A.A., Avila, S., 2020. Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association. *Comput. Electron. Agric.* 170, <http://dx.doi.org/10.1016/j.compag.2020.105247>, arXiv:[1907.11819](https://arxiv.org/abs/1907.11819).
- Schneider, E., Jayanth, S., Silwal, A., Kantor, G., 2023. 3D skeletonization of complex grapevines for robotic pruning. In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS*, pp. 3278–3283. <http://dx.doi.org/10.1109/IROS55552.2023.10341828>.
- Silwal, A., Yandún, F., Nellithimaru, A.K., Bates, T., Kantor, G., 2022. Bumblebee: A path towards fully autonomous robotic vine pruning. *Field Robot.* 2 (1), 1661–1696. <http://dx.doi.org/10.55417/FR.2022051>.
- Tassie, E., Freeman, B., 1992. Pruning. In: *Coombe, E.B., Dry, P. (Eds.), Viticulture Volume 2- Practices. Winetitles*, pp. 66–84, URL: <https://winetitles.com.au/product/viticulture-volume-2-practices/>.
- Vu, T., Kim, K., Nguyen, T., Luu, T.M., Kim, J., Yoo, C.D., 2024. Scalable SoftGroup for 3D instance segmentation on point clouds. *IEEE Trans. Pattern Anal. Mach. Intell.* 46 (04), 1981–1995. <http://dx.doi.org/10.1109/TPAMI.2023.3326189>.
- Williams, H., Smith, D., Shahabi, J., Gee, T., Nejati, M., McGuinness, B., Black, K., Tobias, J., Jangali, R., Lim, H., Duke, M., Bachelor, O., McCulloch, J., Green, R., O'Connor, M., Gounder, S., Ndaka, A., Burch, K., Fourie, J., Hsiao, J., Werner, A., Agnew, R., Oliver, R., MacDonald, B.A., 2023. Modelling wine grapevines for autonomous robotic cane pruning. *Biosyst. Eng.* 235, 31–49. <http://dx.doi.org/10.1016/j.biosystemseng.2023.09.006>, URL: <https://www.sciencedirect.com/science/article/pii/S1537511023001897>.
- You, A., Mehta, A., Strohhenn, L., Hemming, J., Grimm, C., Davidson, J.R., 2023. A real-time, hardware agnostic framework for close-up branch reconstruction using RGB data. <http://dx.doi.org/10.48550/arXiv.2309.11580>, URL: <https://arxiv.org/abs/2309.11580>.
- Zhou, Q.-Y., Park, J., Koltun, V., 2018. Open3D: A modern library for 3D data processing. arXiv:[1801.09847](https://arxiv.org/abs/1801.09847) URL: <https://arxiv.org/abs/1801.09847>.