Regular Article

Bumblebee: A Path Towards Fully Autonomous Robotic Vine Pruning

Abhisesh Silwal[1], Francisco Yandun[1], Anjana Nellithimaru[1], Terry Bates[2] and George Kantor[1]

[1]Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15217
[2]CLEREL, Cornell University, Portland, NY 14769

Abstract: Dormant season grapevine pruning requires skilled seasonal workers, but they are becoming less available. As workers hasten to prune more vines in less time due to the short-term seasonal hiring culture and low wages, vines are often pruned inconsistently, leading to imbalanced grapevines. In addition, grapevines cannot be pruned selectively using currently existing mechanical methods, thus manual follow-up operations are often required, further increasing production cost. In this paper, we present the design and field evaluation of a rugged and fully autonomous robot for end-to-end pruning of dormant season grapevines. The proposed design incorporates novel camera systems, a kinematically redundant manipulator, a ground robot, and novel algorithms in the perception system. The presented research prototype robot system was able to spur-prune a row of vines from both sides completely in 213 s/vine with a total pruning accuracy of 87%. Initial field tests of the autonomous system in a commercial vineyard have shown significant variability reduction in dormant season pruning when compared to mechanical prepruning trials. The design approach, system components, lessons learned, future enhancements, as well as a brief economic analysis are described in the manuscript.

Keywords: Robotic pruning, agricultural robotics, vineyard automation, autonomous navigation, unstructured environment

1. Introduction

Pruning is a primary tool used by grape growers to manipulate vine size and shape, which helps to regulate crop-load and maintain vine balance. Dormant season grapevine pruning involves removal of plant tissues in the form of spurs and excess one-year-old canes from the previous year's growth. It is a highly labor-intensive task that requires skilled workers during winter season, who are becoming less available. As labor workers are paid per vine to prune, the short-term seasonal hiring culture often leads to workers rushing to prune more vines in less time. This leads to inconsistent pruning of vines that often results in over/undercropping and could take several years of careful mitigation

Received: 17 December 2021; revised: 5 May 2022; accepted: 7 June 2022; published: 4 August 2022.

Correspondence: Abhisesh Silwal, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15217,
Email: asilwal@andrew.cmu.edu

DOI: https://doi.org/10.55417/fr.2022051

to recover and remain profitable (Bates and Morris, 2009). Different pruning strategies have been extensively studied in various grape-growing regions and grape varieties to achieve sustainable vine vegetative and reproductive growth, often referred to as vine or vineyard balance (Howell, 2001). Some of the contemporary vineyard mechanization systems in vineyards during the dormant season include (in sequence) mechanical prepruning, manual pruning follow-up, and mechanical shoot or fruit thinning to maintain vineyard balance (Bates, 2014).

Grapes are the leading fruits based on the production volume in the United States (National Agricultural Statistics Service, 2019). The U.S. grape industry mainly consists of wine, table, juice, and raisin varieties that combined produced around 7.4 million tons of produce in 2017 (National Agricultural Statistics Service, 2019; Economic Research Service, 2016), and it is currently valued at U.S. $6.6 billion. Despite its impressive growth in the past decade, the grape industry continues to rely on hand labor for many operations. Among the most labor-intensive and costly tasks in grape production are harvesting, pruning, cluster thinning, and equipment operation. Pruning is often labeled as one of the top three most costly tasks, and it could make up a quarter of labor costs in the fruit production cycle (Johnson and CourtneyRoss, 2016). According to the University of California Cost and Return Studies in 2017 (Fidelibus et al., 2018; Alston et al., 2018), table grape growers can annually incur operating costs up to $18,000 per acre to generate income of about $30,000 per acre (Fidelibus et al., 2018). These account for approximately 45% of costs just for labor. Future projection of the labor issue is expected to become more critical both in terms of uncertainty in the availability and increasing costs (Fennimore and Doohan, 2008; Calvin and Martin, 2010). These concerns about labor supply have promoted renewed focus and enhanced interest in mechanization and the use of advanced technologies to secure long-term sustainability of the grape and fresh fruit industry, in general.

To reduce labor cost, vineyard mechanization research has played an important role in the grape processing industry in the U.S. The invention and adoption of the mechanical grape harvester in the early 1970s eliminated hand harvesting as a labor issue in the grape juice industry. Research and development of mechanical pruning has continued since the mid-1970s (Morris, 2007) and it alone has further reduced labor costs. However, the lack of specificity in retained nodes causes the vines to be overcropped (out of balance) with poor fruit quality (Bates, 2008; Bates and Morris, 2009). This lack of selective pruning capability only provides a partial solution as additional follow-up operations are often required to complete the task, which further increases production cost. Grapevines are perennial plants with indeterminant growth habits, so canopy structures are highly vigorous, and the entanglement of canes quickly leads to canopies that are too complex to analyze even for trained human eyes, let alone for computer vision algorithms. Thus, a robotic pruner as a follow-up operation after mechanical prepruning could be a pragmatic solution. This work presents a systematic approach to integrate robotic systems to fully automate hand follow-up operations after mechanical prepruning. Further, the profit margins for commercial vine production in general are low, the quantity and quality of manual labor are declining, while the cost for fuel and fertilizers is ever-increasing (Uzes and Skinkis, 2016). The development of automated robotic pruning as the mechanical pruning follow-up operation would further reduce labor costs and increase specificity in retained node quantity and quality. The growing region considered is located in western New York, one of the largest grape juice producers in the U.S.

Our long-term goal is to develop a commercially viable and fully autonomous pruning system to reduce dependency on seasonal semiskilled workers while improving productivity. The overall objective is to investigate robotic technology to significantly improve and stabilize the balance between vegetative and reproductive growth that would yield better fruit quality and predictable crop load. Our approach deviates significantly from the established paradigm in agricultural robotics in two major ways. First, it is recognized that a grapevine training system that facilitates robotic technology in vineyards is key to successful implementation of autonomous and selective pruning of vines. Thus, the commercial vineyard in our study was specifically designed and is constantly modified to facilitate automation. Second, the design of the proposed robot is multifunctional with the capability to perform other tasks such as autonomous multisensor data collection throughout

the growing seasons while remaining compatible with different varieties and canopy architectures of vines that add more novelty to existing systems and potential for commercialization.

The remainder of this paper is organized as follows: Section 2 discusses prior work in this field and how the outcomes of the previous research motivated some design selections. In Section 3, we describe the work environment modifications and basic viticultural terminologies for context. A key requirement for accurate perception for vine modeling and manipulation in complex environments in the outdoors was a robust illumination invariant camera system. The inclusion of such a camera system to measure thin vine structures in this systems integration work is based on (Silwal et al., 2021). Similarly, the three-dimensional (3D) computer vision pipelines to generate and process vine models are based on shortcomings of 2D methods previously reported by (Botterill et al., 2017a). The camera design consideration along with navigation and manipulation pipelines for robotic pruning are detailed in Section 4. Section 5 reports the results and lessons learned from field-testing of the pruner in a commercial vineyard. Finally, some concluding thoughts and discussion on further improving the robustness of the current design are in Section 6.

2. Relevant literature

In the past several decades, a great deal of research on the development and use of robotic systems for various agricultural tasks has been done by the scientific community. Automated solutions for sowing seeding, monitoring, or pest-detection are widely documented in the literature (Gollakota and Srinivas, 2011; Diago and Tardaguila, 2015; Ebrahimi et al., 2017; Li et al., 2009). These are complex systems designed to work in unstructured environments and changing lighting conditions (Bac et al., 2014; Gongal et al., 2015). One of the most targeted applications of robotics in agriculture is harvesting of fruits and vegetables. In a recent work, (Bac et al., 2014) reviewed 36 different robotic projects completed between the years 1985 and 2012. All reviewed projects were developed for fruit or vegetable harvesting. Historically, the limiting factor in perception has been to robustly detect fruit under occlusion and uncontrolled natural illumination (De-An et al., 2011; Li et al., 2011), while removing fruits without damaging and achieving a picking speed comparable to that of a human picker have been the major bottleneck in the manipulation side (Botterill et al., 2017b). Despite the obvious advantage of automated pruning and the underlying commercial benefit, automated pruning has not received much attention when compared to harvesting. The lack of research interest and progress could be attributed to the complexity of the task itself.

For harvesting applications, the target fruits are generally easy to reach, and simple point-to-point paths are enough without the need for collision avoidance (Botterill et al., 2017b). Pruning, on the other hand, presents significant challenges as the system not only has to detect the canopy structure but also measure topological parameters such as the location and orientation of the cutting points in the branches (He and Schupp, 2018; Tabb and Medeiros, 2017a). As vine structures become more vigorous, the entanglement of multiple canes could easily become too complex to solve. In the past, very limited attempts have been made to design and evaluate a full-scale robotic system for pruning vines. A robot system to spur-prune grape vine was designed by Vision Robotics Corporation in 2015 (Vision Robotics Corp., 2020). This commercial prototype used a pair of stereo cameras to identify and localize the cut-point in canes and an industrial robot arm to prune highly manicured vine structures. However, the performance characteristics, and the design details on perception and manipulation are unknown and publicly not shared. A recent full-scale vine-pruning prototype consisting of a robot arm, a multicamera system, and over-the-row supportive structure for controlled lighting was proposed by (Botterill et al., 2017b). This system generated models of vines for collision-free manipulation and autonomously pruned a row of a vineyard.

Robotic systems designed to interact with plants such as for the pruning of dormant vines require robust perception capabilities for motion planning and manipulation in unstructured environments. Before such interaction happens in robotic pruning, locating the pruning points is a necessary step, which itself is a challenging problem given that vines lack consistent structures in their natural form. To automate the process of pruning point detection in vines, (Corbett-Davies et al., 2012)

presented an AI-based expert system. It was based on rules defined by a viticulturist, and it used 3D topological features of the tree such as length, curvature, angle, etc. in deciding whether to keep or prune the branches. Similarly, (Katyara et al., 2021) used a combination of mean predictive histogram of gradients and statistical pattern recognition with a K-means algorithm to classify pruning locations. These resent efforts used some form of an optimization-based approach to identify pruning locations in vines. In pruning, the answer to where to make cuts is dictated by the pruning rules set by a viticulturist. However, regardless of the pruning rule, the number of buds retained plays an important role as the new parts of the vine (both vegetative and reproductive) emerge from the retained buds. To the best of our knowledge, only our work physically detects and associates buds individually to each cane for the pruning decision not only to closely resemble manual pruning but also to prevent accidental overpruning.

Furthermore, to identify pruning locations in complex vine structures accurately, an additional semantic understanding of the scene is required. For example, the segmentation of the canes from vine structures and the precise measurement of important topological parameters such as bud distribution and cane lengths must be known. Getting a detailed semantic map of plants in real time and consistently in the outdoors has always been a bottleneck in pruning and perception-based agricultural robotics, in general (Kazmi et al., 2014; Houle et al., 2010). One of the major factors affecting consistency in having such capability is the changing outdoor lighting conditions that affect image quality. Historically, to limit effects from changing outdoor illumination, researchers have relied on external structures with controlled lighting. Examples include works by (Marin et al., 2015; Botterill et al., 2013; Vision Robotics Corp., 2020), where they addressed the background and illumination challenges by employing a wheeled platform with controlled lighting that completely covered the vines during imaging. The large platform had to be pulled along the rows at low speeds, resulting in a complex and slow application for pruning. Similarly, (Kicherer et al., 2017) presented two different approaches to avoid uncontrolled lighting conditions and the presence of the vines from another row in the background: (i) manual segmentation of images using an artificial white background, and (ii) the use of a multicamera system for depth reconstruction. A robotic system to measure tree traits by 3D reconstruction of a fruit tree in field settings was also presented by (Tabb and Medeiros, 2017b). They measured parameters such as branching structure, branch diameter, length, and angle with low mean square error but required extensive computation time (more than 5 min per tree), making it not suitable for real-time in-field applications.

In a similar application, (Tabb and Medeiros, 2018) present a super-pixel-based image segmentation method for semantic segmentation in field environments for tree reconstruction and apple flower detection. It also involved using a mobile background unit and capturing hundreds of images per tree. Furthermore, image-based cane segmentation and applied Gibbs sampling was used to recover 2D structures of a dormant season grape plant from images by (Marin et al., 2015). They also presented a quantitative comparison of their method with previous work on 2D cane structure extraction (Botterill et al., 2013). Although their method performed well in detecting cane segments, it suffers from low precision due to its inability to detect branching points and hence ending up with disjoint cane segments. Their system also relied on a customized background screen in the field to perform foreground-background segmentation. In a similar study, (Millan et al., 2019) presented an image-based cane segmentation method to assess pruning weight in a vineyard. They overcame the background segmentation challenge in outdoor environment by using a white background to avoid the presence of the vines from another row in the scene and also by taking images at night without any background. Their research was more focused on background-cordon-trunk-cane segmentation for pruning weight assessment rather than pruning point identification. To achieve consistent image exposure in any lighting condition, (Pothen and Nuske, 2016) used a high-resolution stereo sensor with flash to predict yield in vineyards from image-based counting for grapes. The use of flash imagery in this study generated images with uniform white balance and had minimal effects from natural illumination. Our design of the camera system in this paper is motivated by this work. In a follow-up study, we show that the consistency in images not only facilitates a classical computer

vision algorithm but also tends to reduce the amount of data required to train deep-learning networks (Silwal et al., 2021).

Another aspect that makes a robotic system especially valuable in agricultural applications is its capability to navigate around the environment. Agricultural fields normally have off-road terrains where any vehicle has to drive in a safe, socially predictable, and in some cases energy-efficient manner. Challenges including noise in the sensors, loss of traction, and space constraints, among others, make this task especially complicated. Depending on the application and the type of crops, various strategies using perception, planning, and control have been studied to develop autonomous or semiautonomous systems to drive in these scenarios (Mousazadeh, 2013; Bechar and Vigneault, 2016). The perception subsystem normally uses data from cameras, laser range finders, inertial measurement units (IMUs), or GPS receivers to obtain information about the environment and localize the robot within a map (Chen et al., 2015), or it uses simultaneous localization and mapping (SLAM) algorithms (Abouzahir et al., 2018). With the continuous improvement in computation capabilities, machine-learning approaches have become popular for this task, mainly using visual sensors (Chen et al., 2020). Additionally, a diverse group of planners and controllers have been designed and used to guide and command robots to navigate in an Ag settings (Papadakis, 2013; Ding et al., 2018). For example, in (Chen et al., 2020) a local planner was combined with a custom control law for an Ackerman vehicle driving in a hazelnut orchard. In this case, both the planner and the controller were designed to account for the kinematic constraints of the vehicle as well as the space restrictions that limited the maneuverability. Other than custom control laws, the stability, accuracy, and smoothness in the navigation that predictive approaches provide made them especially suitable for agricultural applications (Ding et al., 2018). Furthermore, when the characteristics of the terrain strongly constrain the vehicle movement, predictive traction control strategies have arisen as a suitable solution (Sunusi et al., 2020). The mentioned perception, planning, and control approaches have been successfully implemented mainly for supervision and sensing tasks (Fountas et al., 2020). However, little work has been reported in the integration of an autonomous navigation system to work alongside specific complex agricultural activities such as harvesting and pruning. In fact, the design and evaluation of a methodology for an integrated autonomous system capable of performing these tasks remains as a gap in field applications.

In summary, because of very complex requirements in perception and actuation, extremely limited work and success has been seen in robotic pruning of grapevines and pruning in general. The existing prototypes rely on external physical structure (over the row platform) for acquiring images in the outdoors. This makes the robot's ability to turn, enter, and row-follow in an agricultural terrain extremely challenging and less pragmatic. Most importantly, the rigid frame designs further limit compatibility to different varieties and canopy architecture that could potentially limit commercial adoption. We believe that our rugged and modular robot equipped with an illumination invariant camera system and a novel approach to perception and manipulation will lead to a pragmatic and economical solution for automated pruning.

3. Field environment and workspace modifications

The vineyard used for this study was located at the Cornell Lake Erie Research and Extension Laboratory in Portland, New York. Concord (Vitis labruscana, Baily) grapevines were own-rooted and planted in Chenango gravel-loam soil in 2012 at 2.6 m row by 2.4 m vine spacing and trained in bilateral cordon architecture with an average cordon height of 1.8 m. This variety of grapevine has indeterminant growth habits that results in canopy structures that are vigorous with a high degree of cane entanglements, which creates a work environment where even manual pruning becomes a cumbersome task. A standard way in the industry eases labor intensity by mechanically prepruning the vines. The mechanical prepruners such as the VMech prepruning head comb grape canes up or down (white fingers in Figure 1, right) into reciprocating cutter bars (red vertical in Figure 1, right) that have an adjustable mechanism to retain longer or shorter canes. This mechanical preprocessing step, although a nonselective process, greatly minimized the complexity of the work environment.

Figure 1. Dormant Concord grapevines before and after mechanical prepruning (left top and bottom) with an OXBO V-Mech sprawl prepruner (right).

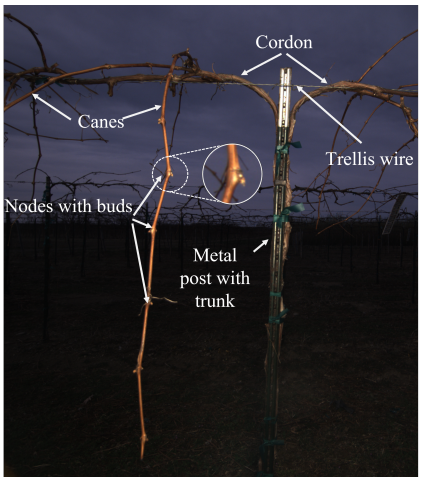


Figure 2. Dormant grapevine canopy showing canes, cordon, trellis, buds, and nodes.

Following this industrial standard, in our experiments we mechanically prepruned the vines with an OXBO VMech 1210 Tool Arm and Sprawl prepruner (Vmech LLC, Fresno, CA). The mechanical prepruner was attached to a tractor and manually driven along the rows and was calibrated to remove canes greater than five nodes long. The result of mechanical prepruning and the prepruning machine are shown in Figure 1.

Additionally, during our latest field trip (which was pushed towards the end of the pruning season because of the COVID-19 global pandemic), some of the vines that started to show vegetative growth were trimmed to retain their original dormant shape by removing the new shoots.

For context, the following brief definitions provide description of the vine canopy and commonly used terminologies in viticulture (see Figure 2).

- **Bud**: *A bud is a growing point that develops in the leaf axils and is often regarded as a compressed shoot.*
- **Shoot**: *New green growth developing from a bud.*
- **Cane**: *A matured long, woody shoot after leaf fall.*
- **Node**: *The bulged part of a cane where buds are attached.*
- **Cordon**: *The main lateral expansion of the trunk that supports shoots, canes, and fruits.*
- **Pruning rule**: *A set of rules that define a systematic way to remove older canes from grapevines.*

4. Methods

This section describes all components of Bumblebee. First, we describe the mechanical design of the robot that includes a prismatic base to increase the reachable workspace and then the design of the end-effector to prune vines. Secondly, we then detail the perception pipeline that describes the camera system, 3D reconstruction of vines from multiple views, and the novel 3D cane segmentation algorithm. The rest of the section elaborates onmotion planning, navigation, and systems integration components.

4.1. Mechanical design

4.1.1. Manipulator

An unrestrained rigid body in 3D space has 6 Degrees of Freedom (DoF) described by the three translations and rotational angles about the three independent axes (Donald, 1984). In theory, a robot arm with at least 6 DoF is required to achieve any pose in the workspace. In practice, this capability is severely limited by different factors such as singularities, self-collisions, collision models of the environment, etc., to name a few. However, in kinematically redundant mechanisms, the desired motion of the tool-end or the end-effector can be accomplished in an unlimited number of ways. In the design of the robotic manipulator for pruning dormant season vines, we extend the 6 DoF of a UR5 robot arm to 7 DoF by adding a prismatic joint to the base, as shown in Figure 3, left.

Since the DoF of the manipulator is greater than the Degrees of Constraints (DoC), our proposed design is kinematically redundant and offers several advantages. First, the kinematic redundancy physically allows the end-effector to achieve any combination of orientations required to reach pruning locations in a complex and unstructured work environment. Secondly, the motion of the robot arm is restricted by multiple constraints, such as joint limits and end-effector poses. These restrictions further narrow the convergence of the Inverse Kinematics (IK) and motion planning algorithms. Having redundancy greatly increases the odds of finding possible solutions and improves the convergence time and accuracy of these algorithms. And lastly, the span of the prismatic base drastically increases the reachable work envelope of the arm, as shown in Figure 3, right. This feature was particularly designed to provide the system with the ability to reach the entirety of the vine from a single stationary position without having to move the mobile base to new locations to work on the same vine. Thus, the possibilities of inducing errors in both manipulation and perception pipelines caused by the motion of the ground robot and the repetitive sensing of the same environment are highly reduced. The design of the redundant P6R open chain robot arm is shown in Figure 3, left.

Although posing joint restrictions limits the full range of motion of each joint, constraints are important factors in motion planning. To control unnatural, unachievable, or unnecessary motions, several joint limits and constraints were set, as described in Table 1. A virtual wall (behind the linear base in Figure 3, left) imposed restrictions on the motion planner as target locations were always

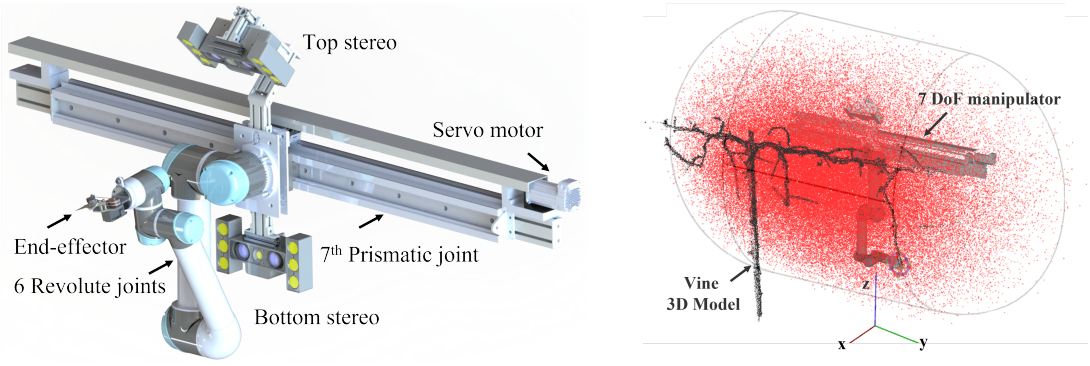


Figure 3. CAD rendering of the 7-DoF robot with its components (left). Reachable end-point positions in the work envelope (right). A wire diagram is added to aid visualization.

Table 1. Manipulator parameters and joint information.

Robot Parameters		Joint Information		
Robot Parameter	**Value**	**Joint Name**	**Lower Limits**	**Upper Limits**
Degrees of Freedom	7	Base	-0.675 m	0.675 m
Max canopy depth reach	0.9 m	Shoulder pan	-2.43 rad	2.01 rad
Prismatic base length	1.35 m	Shoulder tilt	-2.62 rad	0.05 rad
Max joint velocity	0.15 m/s & 1.1 rad/s*	Elbow	-2.62 rad	0.05 rad
Max joint acceleration	0.25 m/s & 0.25 rad/s*	Writs 1	-3.14 rad	3.14 rad
Workspace volume	3.95 m^3	Wrist 2	-3.14 rad	3.14 rad
End-effector weight	0.5 kg (1.2 lb)	Wrist 3	-3.14 rad	3.14 rad

*Values used during the experiment.

in front of the arm. This resulted in the "elongated hemispherical" shape of the work envelope for the 7-DoF arm.

4.1.2. End-effector

Pruning the grapevines require making precision cuts at a specific location in the canes. However, before such cuts could be made, it was important to understand the mechanical properties of the canes, especially the force required to cut dormant canes for the proper design of the pruning end-effector. Due to numerous environmental factors such as soil properties and access to nutrients and water, vines exhibit wide variation in the length and diameter of dormant canes (Bates, 2017). In this experiment, we selected 30 samples of Concord canes with a wide range in sizes and age to quantify the force required for a successful cut. On average, the cane diameter of the collected samples varied from 5 to 11 mm.

While the freshly cut live canes showed the presence of a hard outer shell with soft internal tissues, dead samples exhibited relatively harder, shrunk, and dry internals and required higher force to cut. An experimental setup for this quantitative experiment is shown in Figure 5 (left). The mechanism of using hand-held shears to cut canes (like scissors) operates under the principles of the first-class lever, and it involves applying normal force on both handles at the same time. The experimental setup in Figure 5 (left) simplifies this requirement to just one normal force by fixing one of the handles to a rigid surface. An incremental static load was then applied at the top of the movable handle to cut the cane samples, which were set at a fixed distance from the pivot and placed orthogonal to the cutting plane. Then, the total weight (load) along with the mechanical advantage of the lever at the abscission point provided the cutting force. This experiment was repeated on all sample canes. On average, 320 N force (at the abscission point) was required to cut a typical cane with 8 mm diameter. In Figure 5 (right), a fairly linear relationship ($R^2 = 0.74$) can be seen between the cane diameter and the normal force required for cutting.

A popular choice among professional pruners to prune grapevines is bypass pruning shears. This variety of pruning shears has blades that completely "bypass" each other for precise cuts and clean separations of the canes. Motivated by this pragmatic feature, the design of the end-effector includes a similar bypass mechanism (Figure 4). In this custom designed end-effector, one end of the scissors was fixed and bolted to the frame of the end-effector, whereas the other end was movable and actuated with a combination of a high torque (3 Nm) servo motor and floating pulleys. The floating pulley mechanism transferred power from the motor to the blades with a 200 lb (90.72 kg) fishing line. This simple machine system also added a mechanical advantage (M.A) of 6 and increased the overall factor of safety by nearly fourfold. In the design of the end-effector, the results of the experiment described above mostly assisted in the selection of the motor with adequate torque. Essentially, matching the torque required cutting the canes at a fixed distance from the pivot to the max-stall torque of the motor. To reduce cost and weight, a slightly undersized motor (MX-64AT from Dynamixel Inc.) was selected, which required us to use simple machines to increase the mechanical advantage of the system. The combination of lightweight materials, a simple machine,

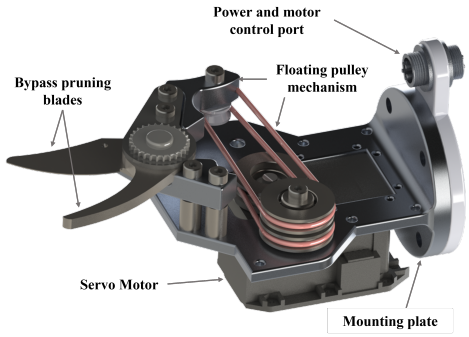


Figure 4. A CAD model showing the design and components of the end-effector. A compression spring (not shown here) keeps tension on the pulley and has a mechanical advantage of 3.

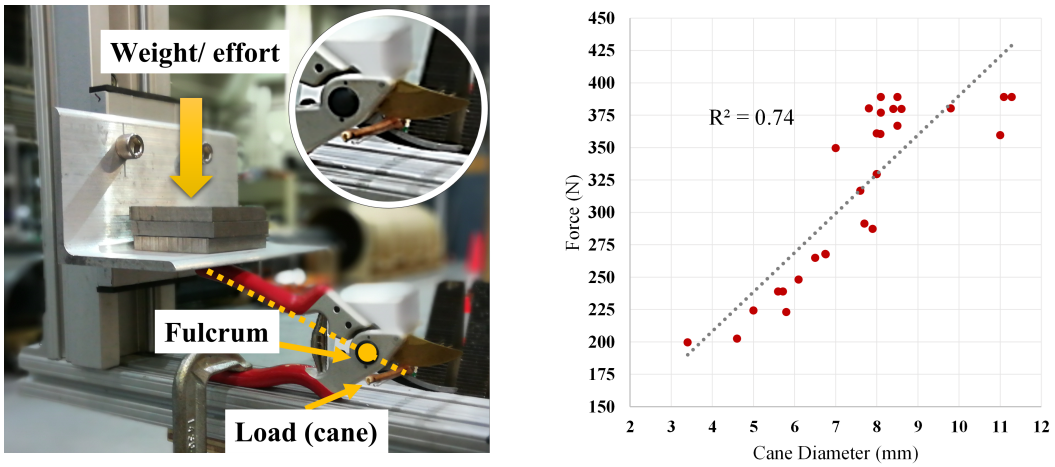


Figure 5. Experimental setup to measure force required to cut dormant canes (left) with a bypass pruner. Correlation plot between cane diameter and cutting force (right).

and a high torque servo motor ensured a small (0.45 kg) yet powerful end-effector that fell well within the payload capacity (5 kg) of the robot arm. The small footprint and weight of the end-effector were also critical to ensure not only wide ranges of accelerations while executing motion trajectories, but most importantly full horizontal extension of the manipulator into the canopy, which could have been unattainable with heavier and larger end-effectors.

4.2. Perception

The overall perception pipeline to perceive the vines and identify the pruning locations is as shown in Figure 6. The major steps in the pipeline involve acquiring static images from 14 viewpoints, point cloud registration using ICP algorithm, bud detection with Faster-RCNN, and cut-point detection with 3D region growing-based cane segmentation and a graph search algorithm.

4.2.1. Camera system

The performance of computer vision algorithms in the outdoors greatly depends on factors such as motion blur and changing illumination, to name a few. Among other things, abrupt changes in the lighting condition can alter image quality, which can lead to large data requirements in machine-learning-based computer vision algorithms to compensate for variance in images (Silwal et al., 2021). To minimize such an effect, the camera system in our design uses active lighting that

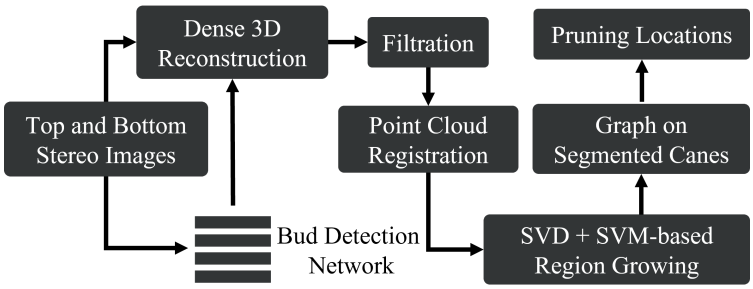


Figure 6. A block diagram with all major steps in the perception pipeline. Program flows from left to right.

greatly minimizes the effects of varying environmental lighting while maintaining consistent image exposure. The detailed description, functionality, and advantages of the active light camera system are described in our previous work (Silwal et al., 2021). The camera system in this paper uses two of these cameras (Figure 12) in a top-bottom stereo configuration, and they moved along the linear slide to image the vines from different viewpoints. The bottom stereo provided front views from a plane parallel to the vines, whereas the top stereo provided tilted views from higher elevation to include occluded parts deeper into the canopy not visible from just the front view. Altogether, the 3D information from multiview geometry enabled us to generate accurate and complete 3D models of complex vine structures. The design of the dual stereo camera with the linear base is shown in Figure 3 (left), and a sample image-set of a vine with and without active lighting is shown below in Figure 15.

4.2.2. Dense 3D reconstruction

One of the critical pieces of perceptual information required for autonomous pruning is precise and accurate 3D modeling of the vines. Once the 3D model is generated, the analysis of the plant geometry and topology can be automated, which also becomes essential for obstacle detection and ultimately in detecting the pruning locations. To generate a dense 3D point cloud of the vines, the dual stereo camera imaged vines at seven precisely set positions along the liner slider. First, the individual point clouds from the top and bottom cameras at the seven different linear positions were registered separately (PCL_{top} & PCL_{bottom}) using Eq. (1). To keep the point cloud registration time low, PCL_{top} & PCL_{bottom} were registered in parallel. As show in Eq. (2), the final 3D model (PCL_{final}) of the vine was then constructed by registering PCL_{top} and PCL_{bottom} using the top-bottom camera extrinsic parameters ($T_{\ top2bottom}$). Alternatively, each top and bottom point cloud pair could have been stitched in series to generate PCL_{final}. However, because of higher point cloud overlap and accurate initial transformation from the slider, this sequence of stitching individually registered PCL_{top} & PCL_{bottom} resulted in a better reconstructed model.

$$PCL_{\ top/bottom} = PCL_1 + \sum_{i=2:n} PCL_i * T_{i-1} \tag{1}$$

$$PCL_{final} = PCL_{bottom} + PCL_{top} * T_{top2bottom} \tag{2}$$

Given two point clouds, fixed (F) and moving (M), where $F = \{f_1, f_2, \ldots, f_n\}$; $f_i \in R^3$ and $M = \{m_1, m_2 \ldots, m_n\}$; $m_i \in R^3$, the Iterative Closest Point (ICP) algorithm finds a rotation matrix R and a translation vector t such that the error between transformed M and F point clouds is minimum. We take advantage of the precise initial transformation or correspondence between F and M provided by the linear sliding mechanism, and we calculate R and t in closed form. For this point cloud registration process, we experimented with the nonlinear version of the point-to-plane ICP by (Fitzgibbon, 2003). This variant of ICP provides more accurate estimation in cases where consecutive point clouds have different densities and exact correspondences are sparse (Fitzgibbon, 2003). Although the point-to-plane ICP takes more time per iteration due to the added cost of point cloud normal computation, it usually converges in fewer iterations compared to classical ICPs. The

point-to-plane ICP for our pipeline iteratively estimates R and t to minimize the distance between every point m_i and the tangent plane at its corresponding point f_i. A tangent plane is represented by its unit normal n_i computed around a small 3D neighborhood of 50 points in the vicinity of the point f_i. To reduce computation time, the size of each point cloud from the camera was reduced using voxel grid sampling, and outliers were removed using statistical outlier filtering prior to ICP registration. The governing nonlinear equation is shown in Eq. (3),

$$\min_{R,t} \sum_{i=1}^{k} [|(Rm_i + t - f_i)^T n_i|^2]. \tag{3}$$

4.2.3. Bud detection

A vine node usually consists of several buds (also known as a compound bud) and has primary, secondary, and tertiary backups. During the growing season, if the primary bud is damaged for reasons such as external injury, frostbite, or other environmental factors, vines sequentially release each remaining backup to replace the damaged/fallen buds. These buds are relatively small and are randomly positioned in the node, which makes it harder to detect in images and 3D models. Thus, despite the presence of multiple buds in a single node, detecting and counting buds as nodes is a reasonable approach. Moreover, this assumption is valid considering the fact that the secondary and tertiary buds generally bear an insignificant amount of fruit compared to the primary bud (Hellman, 2019). In this work, the task of detecting buds is accomplished by detecting nodes. A node as described in Section 3 is the bulged part of the cane that is more visible and has distinct features compared to the rest of the vine. From this point forward, we will use the terms bud and node interchangeably.

Detecting buds is a critical step in the pruning process of grapevines. Pruning rules such as cane and spur-pruning, which are popular in commercial settings, involve retaining a certain number of buds per cane (King, 2021). Thus, accurate counting of buds is extremely important for autonomous pruning of dormant grapevines. To count buds, we leverage the robustness of deep learning-based 2D object detection in the color images of the vines. We used the Faster-RCNN object detection network (Ren et al., 2015) to detect buds in one image from each stereo pair (top and bottom). For training, we used transfer learning and initialized the network weights with the pretrained *imagenet* model before fine-tuning the network to our custom dataset. It consisted of 120 hand labeled images of buds collected prior to the field experiments. Although the number of images in the dataset seems small, the number of instances of buds per image was significantly larger. On average, 45 bud instances were present per image.

The detected buds in the 2D images (top and bottom) were then projected into the 3D space using the camera intrinsic parameters that produced sparse point clouds of the buds. This operation occurred in parallel to the point cloud registration process discussed in the above section and utilized the optimized ICP transformations for final registration. The combination of registered vine and bud point cloud (hereafter referred to as input point cloud, PC_i) completed the 3D vine modeling process. The bud-detection network dataset details and training parameters are listed in Tables 2 and 3, respectively. Figure 7 show sample detections and 3D projection of the buds in the top camera image.

Table 2. Dataset details for the bud detection network.

Network Name	No. of images	Image size	No. of buds
Faster-RCNN VGG16	120	2448 x 2048	5420

Table 3. Training parameters for the bud detection network.

Training samples	Test images	Learning rate	Epochs	Augmentation
85	35	0.01	300	Horizontal flip, scale, random crop

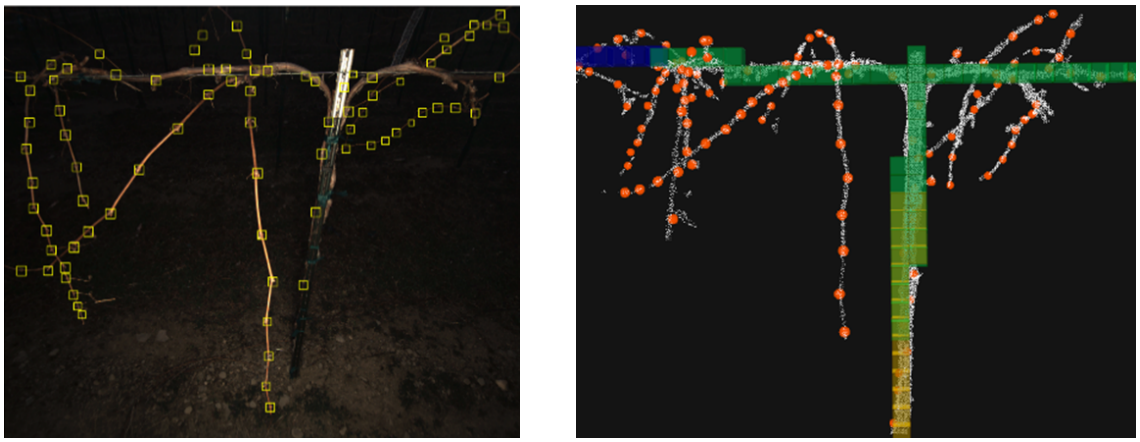


Figure 7. Bud detection with Faster-RCNN (left) marked by the yellow bounding boxes. Use of just cordon and trellis wire as hard obstacles for motion planning (right).

4.2.4. Obstacle detection

Using the dense 3D point cloud and nonlinear point to plane ICP registration, we were able to consistently generate precise and clean point clouds of vines with buds that could be directly used for manipulation tasks. To avoid damage and reduce contact between the robot arm with the vine and its rigid support structure, it was necessary to define obstacles. In this work, only the central trunk with the metal post and horizontal cordons were taken as obstacles, as contact with these rigid structures could potentially cause serious damage. However, as canes are relatively flexible and move when pushed, contacts between these soft objects and the robot arm were allowed to facilitate the motion planning (see Section 4.3.1 for more details). Occupancy grid maps are popular choices to define occupied versus free spaces in the robot's workspace. To define cordon and trellis wire as obstacles, a RANSAC algorithm (Derpanis, 2010) fitted two (vertical and horizontal) lines in the 3D model of the vines (Figure 7, right). As seen in Figure 7 (left), the new vine architecture has a vertical metal post to support the trunk and a horizontal trellis for the cordon to extend laterally. The existence of these features in the point cloud greatly helped the RANSAC algorithm to precisely and consistently fit 3D lines in all vines used in our experiment. These fitted 3D lines were then the only elements taken as occupied space in the Octomap occupancy grid mapping algorithm (Wurm et al., 2010) (Figure 7, right).

4.2.5. Region growing for cane segmentation

In general, the purpose of the region growing algorithms is to merge adjacent data points depending on a region membership criterion. In 3D point cloud space, this criterion could be smoothness constraints, resulting in points with similar smoothness profiles clustered together. In our case, we use a region growing algorithm for cane segmentation by clustering 3D points belonging to the canes. As our 3D structures of interest are thin canes, they did not have planar surfaces for normal computation to make use of a normal based smoothness constraint. Hence, we propose a novel region-growing-based segmentation method that utilizes local structural properties of the vines. We used Singular Value Decomposition (SVD) (Stewart, 1993) on a small neighborhood of points of the vine's point cloud to understand the local structure of the vine in that small area. If there was only one dominant vector after SVD, that indicated the local neighborhood has a linear shape. If the number of dominant vectors was two or three, that indicated the local shape is planarlike and spherelike, respectively. Using this local shape information, we were able to segment out the linear portions of a cane from cane-cordon or cane-cane intersection regions which have nonlinear 3D distribution in the local neighborhood.

We started by randomly selecting a 3D projected bud location as a seed point. A set of points around the seed location were then extracted using a radial neighborhood search operation. This

Algorithm 1. Point cloud region growing

Data: Point cloud P, Directions=positive, negative
Result: Point cloud R
$S \longleftarrow$ Bud Detections;
$A \longleftarrow \{1, ..., |P|\}$ - Cordon points;
while $S \neq \Phi$ **do**
  s element of S;
  if *s is visited* **then**
    remove s from S;
    continue
  end
  for $d \in D$ **do**
    N $\longleftarrow \Omega(s)$; /* Neighbor finding function */
    singular vecs $\longleftarrow$SVD(N);
    if *SVM(Dominant vecs) is true* **then**
      point $\longleftarrow s +$ singular vec ;
      Add *point* to S
    end
    else End points$\longleftarrow$ End points $\cup$ s;
    remove s from S
  end
  Mark points in N as visited.
end

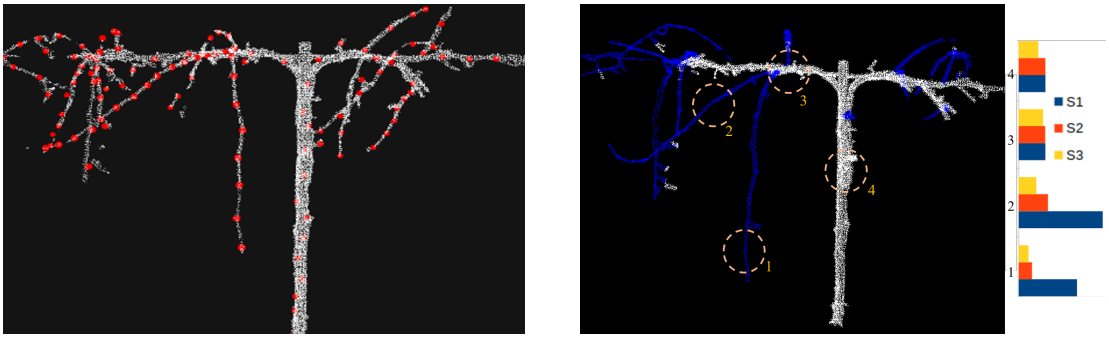


Figure 8. A 3D vine model with buds (red dots) as seeding location for segmentation (left). Segmented canes (center and blue colored). The circles 1 and 2 belong to canes, whereas circles 3 and 4 belong to cordon. S1, S2, and S3 are the singular values after SVD for each circled region (right) with unique patterns.

search radius (a hyperparameter) was predetermined empirically based on the average distance between two consecutive buds in canes. Then, SVD calculated the singular vectors and values from the set of neighborhood points. For the extracted set of input points $N = (Pi) \in R_{mx3}$, the output singular values (nonzero diagonal elements) were $M = (Si) \in R_{1x3}$. The normalized singular values were then passed through a Support Vector Machine (SVM) that learned to classify patterns in the singular values (Figure 8) as the cane or intersection region. Based on the inference from the SVM, the set of extracted points were then labeled as a part of cane or intersection region between cane and cordon. This operation was iterated for all bud locations, and the steps are listed in the algorithm box 1. The training data for SVM involved 120 manually picked samples of cane and intersection regions. About 70% (84 samples) were used for training and the remaining 30% for test and validation.

4.2.6. Pruning rule

Pruning rules define a systematic way to remove older canes to keep the vigor and vine balance in control. Cane pruning and spur pruning are two of the most practiced pruning strategies in the U.S.

Algorithm 2. Cut-point detection

Data: Graph with vertices V and edges E
Result: Cut-point pose $C(x, y, z, \alpha, \beta, \gamma)$
$C \longleftarrow$ Cut Points
$v \longleftarrow V + V_B; V_B \in \{Bud\ Vertices\}$
$e \longleftarrow E + E_B; E_B \in \{Bud\ Edges\}$
$n \longleftarrow$ No. of buds to keep ; /* Pruning rule */
$g \longleftarrow \{graph(v, e)\}$; /* Set of disjoint graphs */
$G \longleftarrow$ **min span tree**(g)
$R \longleftarrow r$; /* Root nodes */
for $i \in G$ **do**
  $L_i \longleftarrow$ Leaf nodes of G_i
  $all\_paths \longleftarrow$ **find_multiple_paths**(G_i, R_i, L_i, $thresh$)
  $unique\_paths \longleftarrow$ **sequence_match**(all_paths)
  for $i \in unique\_paths$ **do**
    $bud\_index \longleftarrow$ **depth_first_search**($unique\_paths_i$)
    $P \longleftarrow \{bud\_index[n], bud\_index[n+1]\}$
  end
  $C \longleftarrow$ { P.x, P.y, P.z, α, β, γ}
end

grape industry. One major difference between these two rules is the number of buds retained after pruning. Cane pruning usually retains longer cane segments with variable counts of buds per canes, while spur pruning leaves fixed but smaller numbers of buds per canes. Algorithmically, cane pruning requires the ability to track buds in longer sections of canes, which could be a very difficult task because of the high degree of entanglement between the canes. On the other hand, spur pruning has simpler requirements, and pruning locations are close to the cordons. For a proof-of-concept robotic pruning of vines, we adopted a simplified spur pruning rule to only retain four buds per cane. In addition to bud retention, pruning rules also necessitate qualitative parameters such as cane diameter and health of canes and buds. In this proof-of-concept design, we considered all canes for pruning and list the inclusion of qualitative parameters as future enhancements.

4.2.7. Cut-point detection

Once the points belonging to just canes are segmented from the rest of the vine structure, the next step in the pipeline is to identify pruning locations. One possible approach could be to further segment clusters of multiple canes into individual clusters and process each cane individually. However, an additional segmentation or clustering step has the potential to induce more uncertainties, as segmentation and clustering processes are not perfect in themselves. So our approach deviates from this logic and processes all segmented canes at once using a graph-based approach. The SVM model explained in the previous section not only labels cane regions in the 3D models, but intersection regions between canes with cordons as well. The algorithm described in this section essentially uses the cane-cordon intersection region to solve the graphs.

To identify pruning locations, the foremost requirement in our pipeline was to convert the segmented cane-bud point clouds into an undirected acyclic graph G. The first step in this process involved the use of Octree data structure to voxelize the cane clusters. The octree data structure recursively subdivides 3D point clouds into octants or voxels until a minimum voxel size is reached (Wurm et al., 2010). Here, an octree with a resolution of 5 cm was used to voxelize the extracted cane point cloud. Once the voxelization process was completed, the centroid of each voxel was extracted as a subsample of the canes. This was necessary to maintain the size of the graph and to keep computation time as low as possible. Subsequently, to generate the graph, a 3D kernel (Figure 9) traversed throughout the equally spaced octree centroids of the canes. With each step, the 16-neighbor kernel assigned vertices and edges to the voxelized cloud. The vertices and edges that

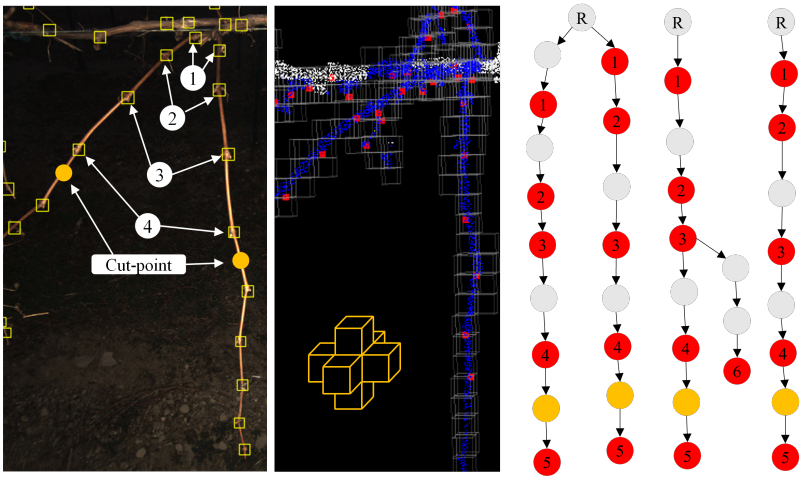


Figure 9. The method employed for counting buds manually (left). The octree voxel representation of the point cloud model (center). Various examples of a graph-based method to identify pruning locations (right). The red, gray, and yellow circles represent bud locations, generic nodes in a graph, and pruning locations, respectively. Similarly, "R" represents the root node.

contained buds were assigned to a special set of vertices V_B and edges E_B. The postprocessing of the graphs mainly involved the removal of loops using the minimum spanning tree (MST) algorithm. It is a greedy approach that removes cycles in weighted graphs while picking the smallest weighed edges. To preserve the edges belonging to bud positions, the set of special edges E_B were assigned smaller numerical weights, whereas the rest of the edges were initialized to larger weights. This simple step inherently preserved all bud vertices and edges, as the global cost of the graph was minimized. Figure 9 demonstrates this logic.

Pruning rules require the correct ordering and numbering of buds per cane, which in our case requires to properly assign each bud to its respective location in the canes. This process essentially involved converting the undirected graph to a treelike data structure that assigned directions in the graph G. By assigning the cane-cordon intersection region as root nodes in the graph, a depth first search-based (DFS) algorithm first computed all possible paths to the leaf nodes. As canes have random and complex 3D structures with or without branching, the paths from the root node to the leaf nodes could have multiple overlapping routes (Figure 9). To suppress this ambiguity, a similarity score was computed on all the paths generated by the DFS algorithm using the sequence matching algorithm from (Jeh and Widom, 2002). This similarity score essentially quantified path overlaps between the root nodes and the end points. A threshold of 0.9 (90% similarity) was used to discard redundant routes in case multiple routs were found between the root node and the leaf nodes on the same cane. On these unique paths, newly discovered buds with respect to the root node were sequentially ordered, and the pruning points were identified using the pruning rule described in the previous section. In our approach, the reduced canopy complexity with prepruning greatly facilitated this heuristic-based bud association algorithm. However, in complex vines with multiple crisscrossing canes, a more robust approach might be necessary.

Once the pruning locations were identified, the next step in the pipeline was to compute its pose (position and orientation). A full pose was required as the cutting tool needs to approach the bud with a certain orientation to successfully make the cut. To calculate the cut-point orientation, we projected cane segments as 3D vectors on all three perpendicular planes. Here a 3D vector is defined by the 3D coordinates of the Nth and $(N+1)$th bud section of each cane that is projected to the XY, YZ, and ZX plane. The angles made by the projected line to the respective planes then provide the roll, pitch, and yaw angles with respect to the reference frame. The midpoint of the vector was taken as the pruning location. This process is depicted in Figure 10 (left).

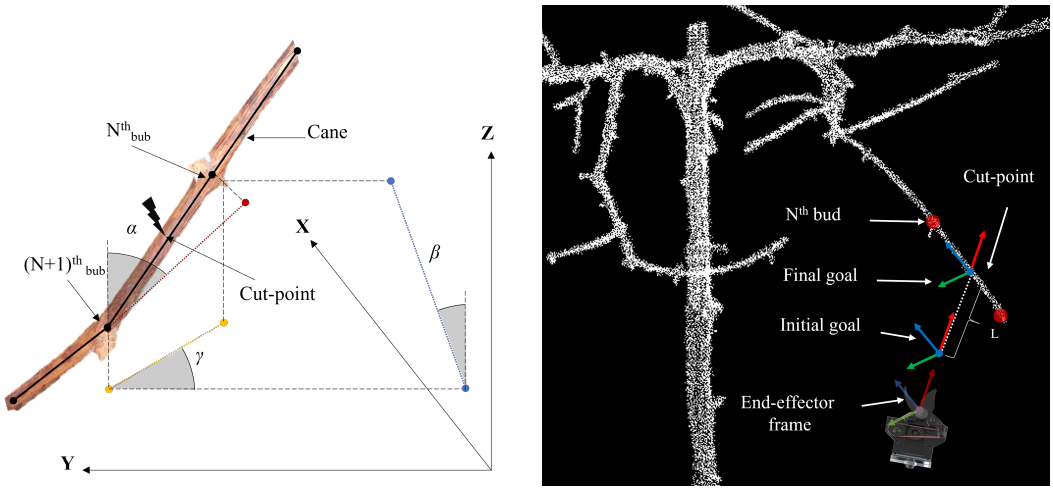


Figure 10. Cut-point detection (left). Three-dimensional vector projection of a cane section in the YZ [red line – roll angle (α)], ZX [blue line – pitch angle (β)], and XY axis [yellow line – yaw angle (γ)]. Pose of the angle made by the projected line made with respective axes. Initial and final approach to the pruning location (right).

4.3. Manipulation

4.3.1. Motion planning

The kinematic redundancy offered by the 7-DoF manipulator only reaches its full potential when the motion planning algorithm can incorporate all degrees of freedom in its planning context. The mechanical design of the robot (Figure 3, left) uses a 6 DoF off-the-shelf robot arm with a custom-built prismatic base. With few modifications, the ROS-MoveIt based planner (Chitta et al., 2012) can be redesigned to plan the joint trajectories for an integrated 7 DoF arm, but at hardware level custom software drivers split the trajectories in real time and control all joints synchronously. To plan motion between all cut-points, we choose RRT-Connect (Kuffner and LaValle, 2000) as an option in the OMPL integrated with ROS that utilizes the entire 7 DoF. The RRT-Connect motion planner provides collision-free motion planning features which greatly fit the requirements for this proof-of-concept prototype (e.g., real time operation, obstacle avoidance, or constrained movement). Additionally, in a comparative study of motion planners for robotic pruning by (Paulin et al., 2015), RRT and its variants were found to have overall better performance.

Once the poses of the pruning locations were computed, the path of the end-effector to the end goal positions was divided into two discrete sets of trajectories. The first set were computed using the RRT-connect solver and were used to make an initial approach to the cutting point, positioning the tool 15 cm ahead of that cut-point (Figure 10, right). Once the tool was in this location, the end-effector was commanded to orientate perpendicularly to the branch that contains the pruning point. Subsequently, to accurately position the cane in-between the cutting blades, the motion planning scheme then switched to a Cartesian path planner. This Cartesian path planner (also from OMPL library) was constrained to maintain the orientation of the end-effector while inching slowly towards the final pose in a straight line. Then the blades closed and opened to mark the end of a successful cut operation. This process proved to be effective in much of the experimental cases; however, it is open-loop as the robot does not have any real-time feedback during the final approach to the cutting point. In Section 6, we discuss some of the ways to close this loop to enhance the robustness of the system.

One way to define obstacles for motion planning could be to take the entire vine structure as obstacle and force planning algorithms to find solutions for all pruning locations. However, motion planning with collision detection and avoidance can be computationally expensive, especially in the unstructured and complex environment of dormant vines. The random arrangement of canes

in the robot's workspace as obstacles could result in the failure to converge to a solution or—as seen in practice—generate trajectories that result in erratic movements of the arm. To avoid such situations, collisions between the arm and the canes were allowed, whereas trunk, trellis, and cordon that are more structured and easier to identify were considered as obstacles. In addition to this, once a cutting action was executed, the arm always retracted backward to the initial pose (similar to the pose shown in Figure 3, right) before planning the path to the next pruning location. This process not only generated a natural-looking motion, but most importantly it provided more open space for the "connect" heuristics in the RRT-Connect planner to generate effective planning queries and solutions (Kuffner and LaValle, 2000).

Despite these systematic steps to supervise cautionary motion of the arm, contacts with canes would still be undesirable. To further minimize contact with canes, the task of sequencing the pruning locations for cutting operations was optimized using a traveling salesman problem (TSP). The nearest-neighbor heuristic-based TSP exhaustively calculated all possible cutting route combinations and prioritized closer pruning locations over the ones farther from the end-effector while also minimizing the total travel distance. Although TSP is an NP-harp problem, the small set of pruning locations per vine made the exhaustive optimization task possible in short duration.

4.4. Navigation

The RTK-GPS receiver mounted on the vehicle, wheel encoders, and the robot's onboard IMU were the main localization sensors in the navigation system. An extended Kalman filter (EKF) fused the IMU, wheel odometry, and RTK position to localize the robot in the real world with the RTK-base as the reference frame. All RTK-GPS waypoints for autonomous navigation, including the locations of the vines, were manually collected. Also, all vines selected for pruning in this work were located in the same vineyard row (Figure 11). However, the current architecture of the vine required us to address the same vine from both sides of the row. Thus, the main task for the navigation system was to drive the robot down the aisles, accurately turn, and enter the aisle on the other side of the same row while stopping at each vine selected for pruning. To navigate in-between vineyard rows, we used a Model Predictive Controller (MPC) (Allgöwer and Zheng, 2012) to follow GPS waypoints.

The MPC controller first connected all waypoints using a spline to produce a smooth global path along with curvatures and speed profiles for the complete route. Subsequently, the RTK-GPS points

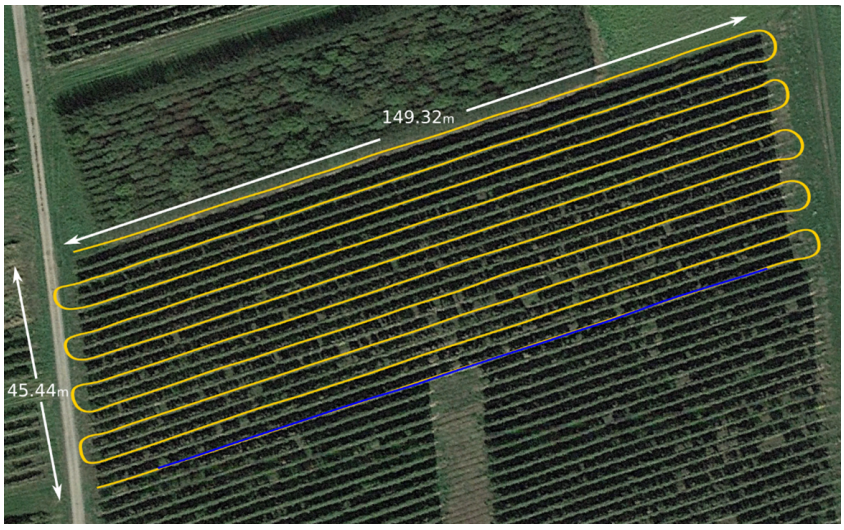


Figure 11. Top view of the test site with RTK waypoint and pruning locations overlayed. The autonomous ground robot was capable of driving around the entire block without manual intervention (yellow path). The blue path corresponds to the row selected for autonomous pruning.

corresponding to the pruning sites were read, and a local path from the current robot position to the next vine to be pruned was generated. Additionally, the planner also calculated local speed profiles (based on the global profile) along with the deceleration and acceleration required to smoothly and accurately stop in front of the tree trunk and start moving again. This approach allowed us to define parameters such as acceleration and deceleration ramps or cruise velocity, ensuring that a complete plan was provided that minimized jerky motion that could potentially damage the robot's components or the crops. For a system with state x and control input u at time t, the general discrete form of the MPC controller used is shown in Eq. (4),

$$\begin{aligned} \min_{x(\cdot),u(\cdot)} \quad & \mathbf{x}^T(t_0+N)Q_f\mathbf{x}(t_0+N) + \sum_{t=t_0}^{t_0+N} \mathbf{x}^T(t)Q\mathbf{x}(t) + \mathbf{u}^T(t)R\mathbf{u}(t) \\ \text{subject to:} \quad & \mathbf{x}(t_0) = \mathbf{x}_0 \\ \forall t \in [t_0, t_0+N]: \quad & \mathbf{x}(t+1) = f(t, \mathbf{x}(t), \mathbf{u}(t)) \\ \forall t \in [t_0, t_0+N]: \quad & 0 \geq s(t, \mathbf{x}(t), (t)) \end{aligned} \tag{4}$$

Here, N is the time horizon, $f(\cdot)$ is the robot motion model, s represents the path constraints, and Q, R, Q_f are the weighting symmetric and positive (semi)definite matrices. The span of the navigation system for autonomous pruning only required traveling a distance of two rows (approx. 0.3 km or 0.19 miles) and it included lane following, stopping between vines, turning, and reentering. To adequately evaluate the autonomous navigation system, we tested the self-driving capability in the entire block (1 mile/1.6 km). In this larger navigation experiment, the robot skipped a row, and entered a new row with every turn without stopping, as depicted by the yellow path in Figure 11. Overall, during this experiment the robot navigated 10 rows and made 9 U-turns autonomously.

4.5. Systems integration

4.5.1. Robot platform

The rugged ground robot (Warthog, Clearpath Robotics Inc.) fitted with a custom aluminum extrusion frame provided a base platform for the gantry system and the field server. The standalone integrated system with all perception, manipulation, and navigation components, and the hardware are shown in Figure 12. All electrical components, including the computers, RTK-GPS, and cameras, were powered by the ground robot's battery except for the arm that was powered with the portable 1000 W gas generator for the AC control box. The edge field server ran on an Intel Xeon E5-2687Wv4

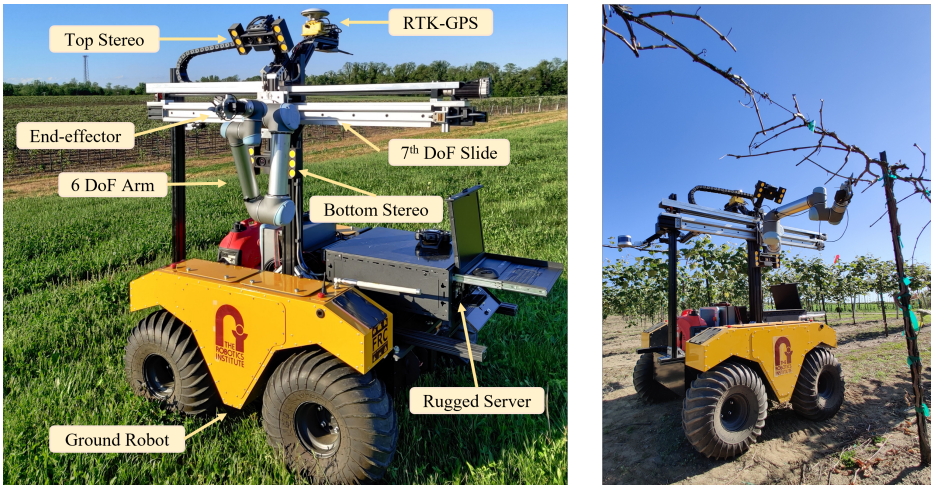


Figure 12. Integrated robotic system with 7 DoF robot arm, ground robot, cutting end-effector, dual stereo cameras, and onboard computers (left). An instance of autonomous pruning (right).

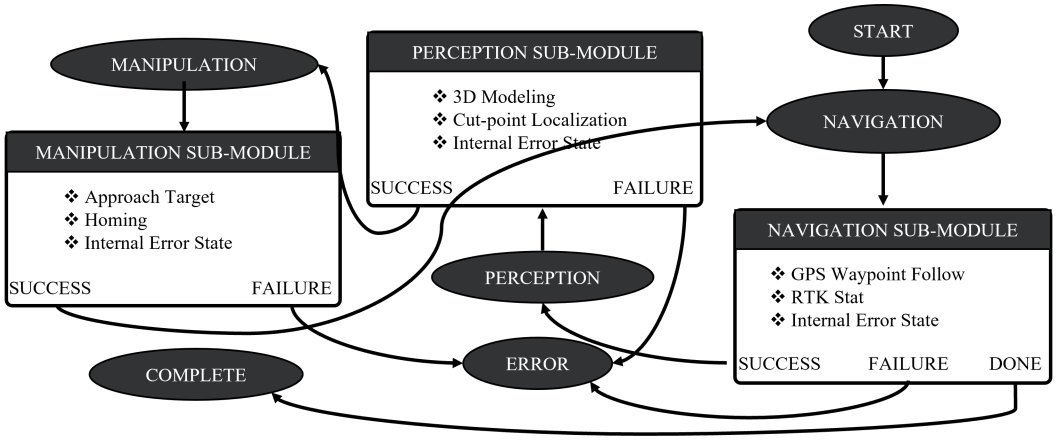


Figure 13. Finite state machine. Depending on the state of the submodules, the state machine transitions between navigation, manipulation, perception, and error states for autonomous high-level control of the robot.

processor with 32GB of RAM and an NVIDIA GeForce GTX1080 GPU for deep neural networks inferencing. All software was packaged for ROS Kinetic under an Ubuntu 16.04 LTS 64-bit Linux environment. A local NTP server was used to sync the clock for all sensors and computers for accurate temporal operations in ROS.

4.5.2. Full autonomy

One complete pruning cycle consists of several tasks that include navigating to the vine position, scanning and 3D modeling, identifying cut points, and executing motion plans to physically remove canes from vines. For efficient high-level coordination and execution of multiple tasks, we used a finite state machine (FSM) as shown in Figure 13. The states in the FSM were navigation, perception, manipulation, and error. Depending on the status of the submodules within each state, the SFM transitions between different states following a predefined sequence for autonomous high-level control of the robot until all vines were pruned. Additionally, for robustness, each of the subprocesses of the states was programmed with internal error substates to self-diagnose software level issues and pause all operations in the occurrence of events belonging to the higher level error state. At the higher level error state caused by events such as hardware failure or unknown issues, after human intervention and fix, the robot continues from its last known location to the next vine.

5. Results

To evaluate all the systems that Bumblebee comprises, four datasets were employed. The first one was used to train and evaluate the bud detector. From this dataset, we selected randomly five samples (vines) to evaluate the reconstruction completeness and the region-growing algorithm. The quality of the overall point cloud generated with the ICP approach was assessed using a single vine imaged in the field conditions. Finally, a total of 20 vines from a single row in a commercial vineyard were selected for pruning. These vines were prepruned with a mechanical prepruning machine to reduce the strength of the vines and to simplify the cluttered work environment. We also provide a brief analysis on how this nonselective process minimizes the complexity of the vines, enabling our system to perform precise pruning.

The methodology employed for all these tests, as well as the results obtained, are described in the following sections.

5.1. Prepruning

As detailed in Section 3, to simplify the canopy complexity, the vines were mechanically prepruned with an OXBO VMech 1210 Tool Arm and Sprawl prepruner (Figure 1, right). In the 20 field vines,

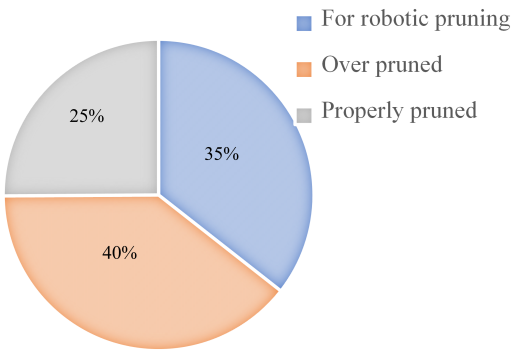


Figure 14. The post prepruning statistic of the nonselective prepruning process showing the percentage of properly pruned, overpruned, and underpruned canes in the experimental row.

we manually counted all canes as well as the number of buds per cane to evaluate the prepruning that set the stage for robotic operation. In total, 268 canes were present in these vines with an average of 13 canes per vine. After the prepruning operation, we observed that 25% of the canes had exactly 4 buds, 35% had more than 4, and 40% were over pruned with fewer than 4 buds per cane. Here, "4 buds" is used as a reference as the simplified spur-pruning rule adopted in this study only required to retain 4 buds per cane. Also, the distribution of buds ranged from 1 to 14 buds per cane as a result of nonselective manual prepruning operation. This large variation in the bud distribution is the variable we aim to minimize with our robotic pruning system, and the results are presented in the following subsections. As seen in Figures 14 and 20, the prepruning step not only greatly reduced the strength of the vines and the length of each cane, but it also reduced the total number of canes to be pruned (35%). Additional statistics include 1122 bud counts in total in all vines with a standard deviation of 2.08 bud counts about the mean of 4 buds per cane.

5.2. Perception and Reachability

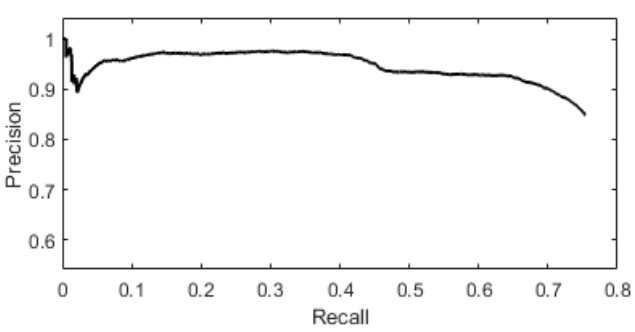
Perceiving the environment is usually one of the earliest tasks for autonomous robots. In our case, detection of dormant buds in 2D images and their projection into the 3D coordinates along with the generation of the point cloud of the entire vine were some of the initial steps in the perception pipeline. As any inconsistency in bud detection or significant error in 3D reconstruction could highly affect subsequent processes such as estimating pruning locations and ultimately pruning, ensuring robust perception capabilities was crucial.

5.2.1. Camera system

The active lighting camera described in Section 4.2.2 was adequate to efficiently suppress affects from natural illumination. As a result, the camera system was able to produce images with consistent exposure (i.e., quality) in all lighting conditions present during the experiments. Figure 15 shows images of the same vine taken at different times of the day with and without flash. The first image (Figure 15, left) was taken with the robot camera in typical broad daylight, whereas the remaining two were taken with the active light camera [Figure 15 (center) at the same time as Figure 15 (left), and Figure 15 (right) at nighttime]. Qualitatively, Figure 15 (center and right) look alike as the flash from the camera in conjunction with fast shutter speed overpowered daylight effects. Quantitatively, the Structural Similarity Index Metric (SSIM) of images taken at various times of the day with the proposed camera system showed an average similarity of 90%. The SSIM is a quality metric that embeds structural as well as contrast and luminance as quality parameters (Wang et al., 2004).

As mentioned in Section 4.2.3, the bud detection network was trained on 85 images and evaluated in 35. Consistency in the image quality is considered a major contributor for requiring such a small amount of data to train our deep object detector. For instance, in Figure 15 (center and right), both images appear similar in exposure, color consistency, and background subtraction regardless

Figure 15. Illumination invariant imaging. A sample image of a test vine taken without flash (left). An image of the same vine taken at the same instance with flash (center). The nighttime image of the same vine with flash (right).



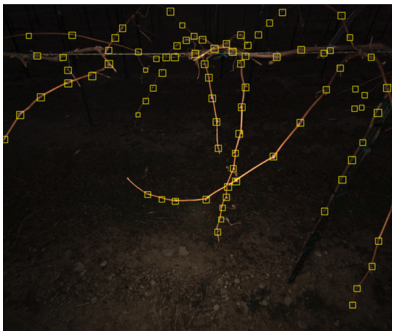


Figure 16. The P-R curve for bud detection network with Faster-RCNN network (left). Each yellow bounding box localizes a bud in the image (right).

of outdoor illumination. The P-R curve of the trained network is shown in Figure 16. Additionally, we obtained a mean precision average (mAP) of 0.93 in the test dataset. A thorough analysis of image quality from our camera system and reduction in training dataset size to fine-tune deep object detectors is explained in our previous publication (Silwal et al., 2021).

5.2.2. Vine reconstruction

To quantify the uncertainty in the measurement of depth information from cameras, depth measurements from stereo pairs were compared against highly precise and accurate laser measurements (±0.1 mm resolution). This process included comparing point stereo measurements from 30 different locations in a single vine at different depths against laser point measurements at the same location. The point measurements under comparison ranged from 0.2 up to 1 m, which also represented the reachable span of the robot arm. As displayed in Figure 12, the dual stereo rig images each vine from 7 different positions, producing 14 different views. The overall point cloud registration process was largely facilitated by the precise and accurate movement of the linear slider. As motion in all directions other than the slider was mechanically constrained, the initial estimates of the point cloud transformations prior to the ICP optimization were very accurate. With such initial estimates, the point to plane ICP optimally computed transformation between successive point clouds as well as the final point cloud registration described by Eqs. (1) and (2). Here, the ICP registration error is defined as the absolute difference between the ICP optimized translation against the actual distance travelled by the camera while imaging at different positions (measured with the encoder of the motor drive of the linear slide). The mean absolute error between these measurements averaged to ±2 mm. With uncertainties from individual stereo measurements and multiple ICP registration steps, the final accumulated registration error was estimated to be within ±6.8 mm. As explained in Section 4.1.2, the average diameter of the canes was 8 mm and the widest blade opening was 38 mm (Figure 4). Therefore, the accuracy achievable from the 3D reconstruction pipeline was well

within the tolerance of the end-effector. All accuracy analyses were done in a mock-up vine in a laboratory setup. This was necessary mainly to rule out effects from wind that could alter both ground truth and the stereo measurements. All laboratory tests used the same camera system as in the field prototype and a real vine collected from the test site.

5.2.3. Reconstruction completeness

In addition to measuring reconstruction accuracy, the completeness of the 3D model also plays a crucial role in the overall success of autonomous pruning. For instance, largely fragmented cane structures and missing buds in the model could highly impact the cut point detection algorithm (see Section 4.2.7), which ultimately affects the pruning efficiency. In the literature, the quality of point clouds is usually assessed using objective and subjective metrics (Karantanellis et al., 2020). Objective metrics usually compare point clouds to a reference or well-defined objects in the scene (Karantanellis et al., 2020; Moon et al., 2019; Zhang et al., 2018). In contrast, subjective evaluation is based on visual inspection and usually involves completeness, density, etc., as factors in point cloud assessment (Karantanellis et al., 2020). However, because of a lack of consistent structures in vines, reference or ground truth point clouds are difficult to generate and are not available for comprehensive comparison. To assess the quality of the point cloud from our dual stereo camera system, we present the following objective metrics:

- **Number of points**: *Total number of points in the registered point cloud.*
- **Number of neighbors**: *Average number of points within the search radius of a sphere with r = 0.05 m.*
- **Surface roughness**: *Average distance between each point in the point cloud to the best fitting plane using neighbors in the search radius of a sphere with r = 0.05 m.*
- **Surface density**: *Average number of points per square meter.*
- **Volume density**: *Average number of points per cubic meter.*

Furthermore, we compare the above objective metrics between three models: one reconstructed only using the bottom camera (BC), the other reconstructed only using the top camera (TC), and finally with the registered point cloud using both the top and bottom cameras (TBC) on 5 vines. The results are shown in Table 4. As expected, the results show that the TBC model has more points, neighbors, and a higher surface as well as a higher volume density when compared to BC and TC models. However, the surface roughness of the TBC mostly remained average between the TC and BC point clouds.

For subjective evaluation, here we define two quality metrics to quantify the subjective quality of the reconstructed vine structures. The first metric, connected components, attempts to quantify the completeness of the vine structure as a function of the octree graphs connectivity described in Section 4.2.7. This approach essentially exploits the connected component properties of graphical structures. For a 3D model without any significant gaps in the model, we would anticipate a single or very few connected components, whereas fragmented/incomplete reconstruction would result in large numbers of connected components. The second metric, bud counts, involves the number of buds in the reconstructed model. Similar to the objective metrics in the above paragraph, we compare the results from the TC, BC, and the TBC model on 5 test vines in the subjective evaluation as well. The

Table 4. Objective quality metrics for vine point clouds.

	No. of Points			No. of Neighbors			Roughness (mm)			Density (#/m^2)			Volume (#/m^3)		
Vine	TC	BC	TBC	TC	BC	TBC	TC	BC	TBC	TC	BC	TBC	TC	BC	TBC
1	32,134	23,197	43,479	381	359	600	6.22	5.50	5.89	48,524	45,735	76,415	727,863	685,956	1,146,260
2	32,522	32,996	51,553	461	441	743	6.85	6.64	6.36	58,757	56,147	94,568	881,357	842,255	1,418,513
3	35,969	37,344	57,036	404	382	654	6.51	6.34	6.42	51,488	48,609	83,290	772,314	729,102	1,249,345
4	29,256	25,642	42,764	418	396	669	6.35	5.64	5.85	53,239	50,481	85,221	798,623	757,188	1,278,310
5	27,912	23,457	39,925	377	387	630	5.95	5.70	5.72	47,958	49,245	80,185	719,353	738,648	1,202,803

Table 5. Statistics on ground truth bud counts and buds present on different point cloud models.

Vine	Ground Truth	TBC	TC	BC	Correlation
1	86	88	64	66	**Ground Truth vs TBC**
2	54	59	47	50	$R^2 = 0.98$
3	77	83	60	63	**Ground Truth vs TC**
4	81	80	56	56	$R^2 = 0.96$
5	121	115	79	76	**Ground Truth vs BC**
Total	419	425	306	311	$R^2 = 0.88$
(100-MAPE)%		94.89	74.76	76.62	

results show that the number of incompletely formed canes was significantly reduced in the TBC model when compared to the TC and the BC models (6 vs 17 vs 19, respectively). Similarly, the number of buds that were essential features for cane segmentation were also more present in the TBC model than in the TC and BC models (425 vs 306 vs 311, respectively). In total, the TBC model had a Mean Absolute Percentage Error (MAPE) of 5.11% in bud count, whereas the TP and BC has a significantly higher MAPE in bud counts of 25.23% and 23.37%, respectively, when compared to manually counted ground truth. Likewise, the R^2 correlation between the manual counts of the buds vs the final registered bud counts was highest in the TBC to TC and BC. Table 5 summarizes all these details on 5 vines for the TBC, BC, and TC models. The high level of completeness in the reconstruction of vines is attributed to the additional (elevated and slanted) views of the canopy provided by the top camera. Evidently, while point clouds from the bottom multiple views provided the majority of the vine structure, the top camera data filled gaps in the most occluded regions. Some of the fragments/disconnected canes in the TBC model were mainly stray canes from the adjacent vines, as no two consecutive vines had a well-defined separation.

5.2.4. Region growing

The validation of the region-growing-based point cloud segmentation algorithm has two parts: (i) accuracy of the SVM classifier, and (ii) the resulting overall accuracy of cane segmentation. For the first part, we measured the performance of the SVM as a binary classifier to classify SVD decomposed values into cane regions. To evaluate the first part, we manually selected 536 sample points from 5 different vines (273 canes vs 263 noncane regions). Out of the 536 random test samples, the SVM correctly classified cane/noncane regions with a F1 score of 0.97.

For the second part, we conducted point-to-point comparison between the hand-labeled cane point cloud to the region-growing segmented point cloud. As hand labeling of complex vine structures is resource-intensive, we are currently limiting the segmentation evaluation to 5 vines. The analysis shows that the cane segmentation pipeline achieved an overall F1 score of 0.91. The confusion matrix for the overall cane segmentation and the SVM-based individual region classification are shown in Figure 17.

5.2.5. Cut point localization

To validate the localization accuracy of the cut points (in between the 4th and 5th buds), we kept track of all the canes (in the 20 vines) with bud counts exceeding 5. In all canes with a significant number of buds, the algorithm estimates of the cut positions were compared against the manual labels. Although, the midpoint between the nth and $(n + 1)$th buds was used as the 3D location of the cut points, any location within the two buds was taken as a valid solution since it was more important to correctly associate the bud sequencing. This algorithm on average achieved an accuracy of 94% across all manually selected pruning locations.

5.2.6. Workspace Quantification

A Monte Carlo experiment was carried out to estimate the volume of the robot's workspace. The idea was to sample points in the joint position space to estimate the overall reaching capability of the 7 DoF robot for one vine. These tests were performed in a simulated environment using a point

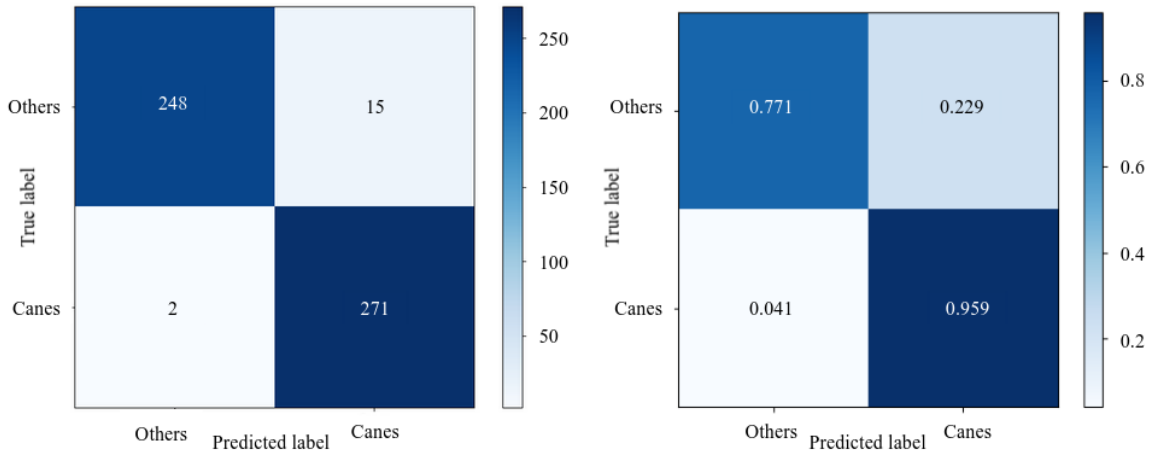


Figure 17. Confusion matrix for SVM-based cane/ noncane classification of SVD values (left). Normalized confusion matrix for hand-labeled cane regions to region-growing algorithm predictions (right). In both confusion matrices, the label "Others" refers to sample points not belonging to canes.

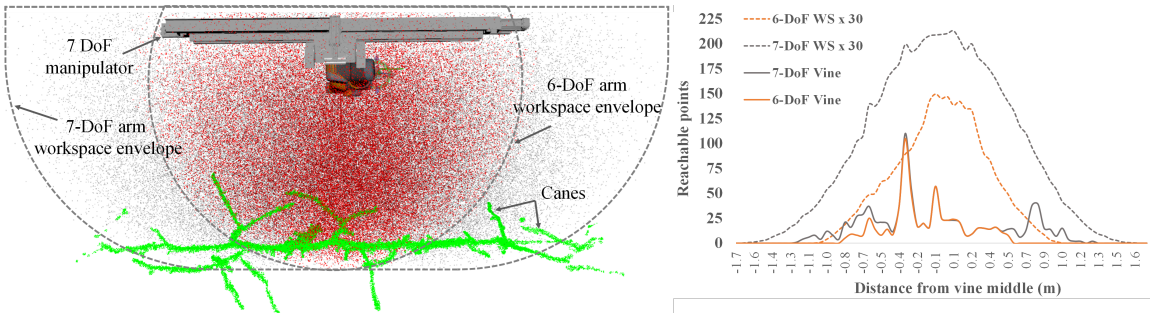


Figure 18. Top view of the work volume of the 6 and 7 DoF arms (left) separated by the dashed line. The red and black dots are reachable positions for the 6 and 7 DoF, respectively. A graphical representation of the reachable locations in the vine for both 6 and 7 DoF arms (right).

cloud model of the field vines. In total, nearly 200 000 end-effector positions were collected as samples of the reachable positions in the workspace. The volume enclosing all the positions reached by the end-effector was estimated fitting a convex hull model implemented in MATLAB. This experiment was repeated for two cases: with the 6 DoF arm fixed to the center position of the linear slide, and with the 7 DoF counterpart fully articulated allowing the prismatic base to move (Figure 18, left). As expected, the results showed that the 3D work volume of the 7-DoF arm (3.5 m^3) was more than 2 times higher compared to the lower 6-DoF design (1.6 m^3). Similarly, because of the current architecture of the vines, this experiment also showed that if the mobile base is close enough to the canopy, an average of 68% of the canes were within the reachable workspace of the manipulator, while the remaining 32% had to be addressed from the other side. A graphical representation of the number of points in the workspace and the number of reachable locations in the vine structure from the center of the vine is shown in Figure 18. In this figure (Figure 18, left), the dashed lines represent the total reachable points in the workspace, whereas the solid lines are the reachable points in the vine structure.

5.3. Pruning

To measure the overall effectiveness of the presented robotic pruner, we introduce several metrics to evaluate its performance. First, Total Pruning Accuracy (TPA) quantifies the robot's ability to

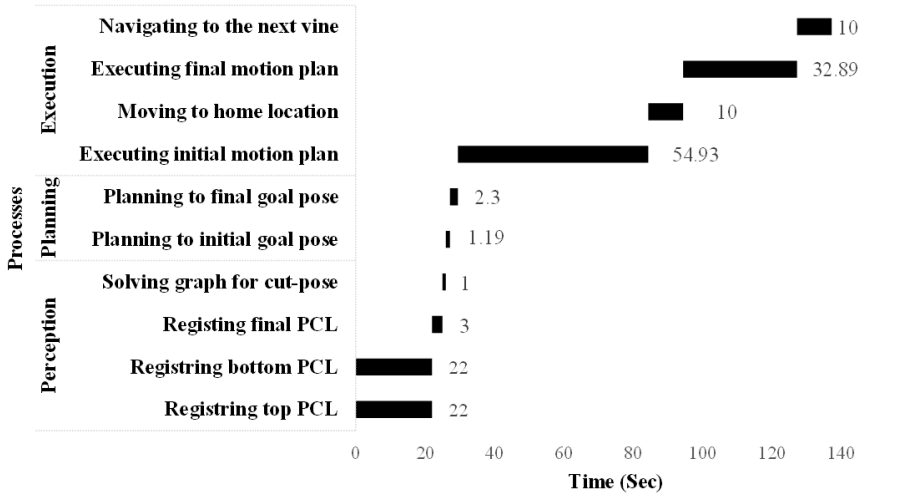


Figure 19. Total computation time breakdown for pruning a vine from a single side of the canopy.

prune successfully at the right pruning locations. Equation (5) defines TPA as:

$$TPA = \frac{Total\ valid\ cuts}{Total\ pruning\ locations} \tag{5}$$

Similarly, Total Pruning Cycle (TPC) is the average time required to prune each vine, as described in Eq. (6). This metric linearly combines the computation cost of all subprocesses in the perception, planing, manipulation, and navigation systems. The computation timing breakdown of all major suboperations in TPC is shown in Figure 19.

$$TPC = T_{perception} + T_{planning} + T_{execution} \tag{6}$$

Figure 19 shows the TPC from a single side pruning and the significant subprocesses under perception, planning, and execution stages. In total, it took 137 s to prune a vine from one side. The current vine training system allowed canes to be randomly distributed on both sides of the canopy, and nearly 32% of the canes were on the opposite side and outside the reachable workspace of the robot. For this reason, the robot had to repeat all operation from both sides of the canopy (i.e., from point cloud model generation to motion planning and execution), which increased the TPC to 213 s/vine.

The variability caused by the nonselective prepruning is shown in Figure 20. After this operation, the standard deviation of the bud distribution per cane was found to be ±2.08. Based on the statistics from Section 5.1 (orange data lines), only 95 out of 268 canes (35%) needed to be pruned. Under the assumption of ideal perception and manipulation capabilities where all pruning locations were detected and pruned, leaving exactly 4 buds per cane, in canes with more than 4 buds the best achievable deviation would be 0.97 standard deviation (Figure 20, red data line). In reality, because of some discrepancies in the pruning point detection and motion planning/execution pipelines, not all canes were consistently pruned. Out of 95 prunable canes, only 83 were successfully pruned, yielding a TPA of 87% (Figure 20, green data line). However, even with 87% TPA, the standard deviation decreased to ±1.03, which is a significant reduction in variance given that the prepruning step overpruned 45% of the prunable canes. In Section 6 we further discuss the source and potential improvements of the current system.

The factor that ultimately determines the success criteria for a pruning robot is its ability to remove canes. In other words, all the steps in the perception and motion planning leading up to the final execution of the cutting action become significant only if the target cut-point gets successfully cut. Commonly used metrics such as the TPA described above only quantify the ratio of success

Table 6. Combination of different design choices that affect TPE.

Combination	$\eta_{registration}$		$\eta_{localization}$		$\eta_{execution}$		TPE	TPC
All inclusive	$\eta_{bud\ detection}$ $\eta_{3D\ reconstruction}$	0.95 0.96	$\eta_{cane\ segmentation}$ $\eta_{cut-point\ identification}$	0.87 0.97	$\eta_{planning}$ $\eta_{execute}$	0.95 0.84	0.61	245
Single side pruning	$\eta_{bud\ detection}$ $\eta_{3D\ reconstruction}$	0.95 0.96	$\eta_{cane\ segmentation}$ $\eta_{cut-point\ identification}$	0.87 0.97	$\eta_{planning}$ $\eta_{execute}$	0.67 0.59	0.3	137
Full vine obstacle	$\eta_{bud\ detection}$ $\eta_{3D\ reconstruction}$	0.95 0.96	$\eta_{cane\ segmentation}$ $\eta_{cut-point\ identification}$	0.87 0.97	$\eta_{planning}$ $\eta_{execute}$	0.84 0.73	0.47	177
Single stereo	$\eta_{bud\ detection}$ $\eta_{3D\ reconstruction}$	0.75 0.68	$\eta_{cane\ segmentation}$ $\eta_{cut-point\ identification}$	0.64 0.68	$\eta_{planning}$ $\eta_{execute}$	0.92 0.84	0.17	105
No-TSP	$\eta_{bud\ detection}$ $\eta_{3D\ reconstruction}$	0.95 0.96	$\eta_{cane\ segmentation}$ $\eta_{cut-point\ identification}$	0.87 0.97	$\eta_{planning}$ $\eta_{execute}$	0.92 0.84	0.59	164

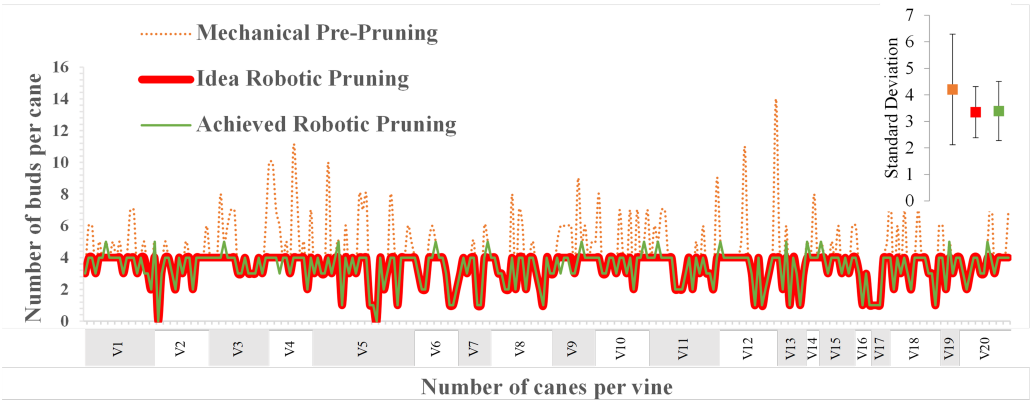


Figure 20. A variability plot that shows the distribution of all buds per cane for every field vine after prepruning, under the assumption of ideal robotic pruning (red data line), and achieved results with the robotic pruner (green data line). The error plot on the right side shows the standard deviation of bud distribution about the mean bud counts on all three cases.

or failure in completing the pruning tasks. To incorporate the effects of intermediate steps and to better describe the overall performance of the pruning robot, we introduce a new metric called Total Pruning Efficiency (TPE). The TPE is a multiplicative combination of several efficiency terms that at high level include perception (3D registration, cut-point detection), motion planning, and execution efficiencies as shown in Eq. (7). In this metric, all efficiencies are converted accuracies or success rates normalized to a number between 0 and 1. For instance, the bud detection efficiency is essentially the accuracy of detecting buds where $\eta_{bud\ detection} = 0.95$ represents 95% detection accuracy compared to ground truth values (see Table 5). Similarly, $\eta_{planning} = 0.95$ represents 95% success rate in the motion planner's convergence to a solution. As shown in Table 6, the TPE is especially valuable in narrowing down system bottlenecks as well as to justify different design choices.

$$TPE = \eta_{registration} * \eta_{localization} * \eta_{execution} \tag{7}$$

$$\eta_{registration} = \eta_{bud\ detection} * \eta_{3D\ reconstruction}$$

$$\eta_{localization} = \eta_{\ cane\ segmentation} * \eta_{cut-point\ identification}$$

$$\eta_{execution} = \eta_{planning} * \eta_{execute}$$

Table 6 summarizes TPE for various possible combinations of hardware, software, and pruning strategies. The first row with the cell labeled "all inclusive" incorporates all system components described in this paper. Here, the TPE amounted to 0.64 even with higher accuracies in the registration and localization pipeline but with relatively low manipulator execution efficiency. In

single side pruning, we only considered pruning a vine from one side. Although the $\eta_{registration}$ and $\eta_{localization}$ efficiencies remained similar, pruning only from one side mainly affected $\eta_{execute}$ efficiency as nearly 32% of pruning location were out of reach. This decreased the TPE significantly to 0.3. Considering the full point cloud model of vines as an obstacle mainly affected the motion planning and execution efficiencies. With more occupied space in the robot's workspace, the sampling-based planner (RRT-connect) took a significant amount of time as well as attempts to converge to a solution. Significant delays were also observed in the motion execution, and most joint configurations looked unnatural and complex. The TPE in this case was 0.47 with a TPC of 177. With just one stereo pair, we observed the most drastic effect on TPE. As it affected the perception part at the beginning of the pruning cycle, the error propagated to localization and motion execution stages. Here, the TPE was only 0.17. Finally, nearest-neighbor-based TSP optimization exhaustively minimized the total point-to-point distance traveled while visiting all pruning locations. Without TSP, the longest possible cut-routes increased the TPC by nearly 20% while the rest of the efficiencies remained similar. The single side, full vine model as an obstacle, the model from a single stereo, and the pruning point sequencing without TSP were analyzed in a simulator, virtually pruned from a single side, and based on the data of the same vines collected in this study.

5.4. Navigation

As mentioned in Section 4.5, the robot drove to each pruning location, remained stopped while pruning, and started moving again to the next vine location. Given that the width of the vineyard row was only 1.8 m wide and the mobile robot with the linear slider and the arm was approximately 1.2 m, maintaining a consistent distance from the canopy and remaining parallel to the rows were critical requirements. The autonomous navigation system was first tested on the accuracy of stopping at each pruning location. We used 20 vines whose positions were marked prior to the trial using the RTK-GPS. During the test, the robot stopped as expected in all locations and the average position of the robot while stopped was used to calculate its distance to the desired vine location. The longitudinal root mean squared error obtained in this case was 0.28 m, which was acceptable given that the pruning task was accomplished successfully in all 20 vines. Laterally, we observed average cross-track errors of 0.07 m for in-row navigation and maximum deviation errors of 0.29 m, which mainly occurred while turning. The average heading angle error was 10.76 degrees, and when stopped, the robot remained parallel to the canopy. This positioning facilitated the 3D reconstruction and motion planning algorithms for the pruning task, as all the pruning points were horizontally equidistant to the cameras for imaging and the arm for the actuation.

We also evaluated the capability of the autonomous navigation system in the larger section of the vineyard with multiple rows. To this aim, the robot was commanded to drive skipping one row

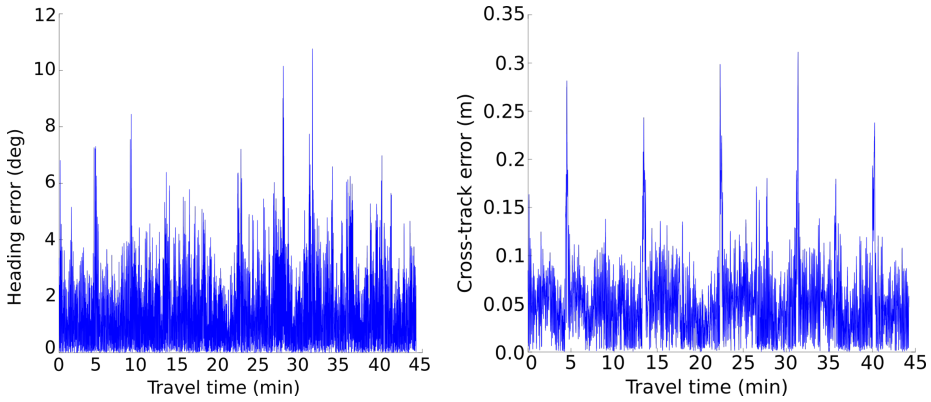


Figure 21. Cross-track (right) and heading (left) errors of the path followed by the vehicle when compared with the desired trajectory generated by the planner.

along the yellow route depicted in Figure 11. The total travel distance driven was approximately 1571.13 m at an average speed of 0.58 m/s and the robot drove autonomously for approximately 45 minutes with no intervention. In both trails, a similar heading angle and cross tracking errors as in the pruning section were observed.

6. Discussion

In general, robots that interact with their environment pose very challenging problems to solve. In particular for robotic pruning, biologically driven surrounding and indeterminant growth habits of vines add more challenges in perceiving and interacting with the environment. It took us three years of effort with two hardware revisions and numerous software modifications to achieve the results reported in this paper. This section summarizes the key lessons learned, capabilities and limitations, future enhancements to our existing system, and some remarks to guide further research.

The first requirement in the perceptual capabilities of the pruning robot is accurate and complete 3D models of vines. With multiple views (14 different viewpoints), the scan-match based 3D reconstruction approach was able to generate precise models of the vines. In the generated 3D modes, canes that are thin structures with diameters as small as 4 mm were clearly visible with few fragments in its structure. The top slanted camera was a necessary addition, which greatly helped to minimize missing information in the occluded regions by adding point clouds from views that were not seen from the front facing camera. Thus, with adequate overlapping from multiple viewpoints, the complete 3D reconstruction of the vines was possible from just one side of the canopy. However, this approach not only required frequent stereo calibration, but it also required manual tuning of several parameters to maintain a relatively consistent size of the registered point clouds for real-time processing. However, modern commercial vineyards typically have consistency in row width and vine spacing, and are equipped with mechanical means to simplify vine complexity at scale. These factors made it possible to tightly control field experiments such as maintaining constant distance between the robot and vines to achieve consistent results even with heuristically chosen parameters.

In this study, complex vine structures were simplified by manually prepruning with a machine. This step not only facilitated the perception pipeline, but the overall pruning operation. Despite the heuristic-based choices of multiple parameters, the 3D reconstruction method seems to be applicable to uncut, highly vigorous and cluttered vines (see Figure 22). However, it can be argued that such a vine could potentially have higher occlusion that could lead to incomplete or missing canes and affect TPE. This limitation could be handled with an in-hand camera system to explore regions of high occlusion and iteratively add missing links. In recent history, deep learning-based point cloud registration (Elbaz et al., 2017) has shown promising results to register noisy point cloud data without accurate initial alignments and could potentially eliminate frequent calibration and

Figure 22. The 3D reconstruction (left) of a complex vine (right) with multicameras and scan match-based approach described in Section 4.2.2.

initialization requirements. However, such a supervised approach could potentially require larger training samples to achieve good results.

Large datasets to training machine learning models for deep learning-based computer vision is a bottleneck in specialty crop industry and agriculture in general. The combination of vast amounts of cultivars and variations within those varieties makes collection and maintenance of labeled datasets for supervised machine learning extremely challenging. The consistency in image exposure and color achieved with the active light camera proved to greatly reduce variance caused in images by ambient lighting. As a consequence, the training sample size was reduced by multiple folds to achieve similar bud detection results when compared to models trained with larger datasets [see (Silwal et al., 2021) for more details]. The availability of public datasets with 3D plant models with proximal sensors is even rarer in agriculture. As hand-labeling of a large 3D dataset to segment canes from the rest of the vines was very resource-intensive, we refrained from the state-of-the-art deep networks and opted for classical machine learning with SVM. With the combination of singular value decomposition and SVM as a binary classifier, the region-growing algorithm was robust at segmenting dormant canes. The training of the SVM model required relatively small but hand-engineered features, and the generation of data for training and the training process itself could be done in a few minutes. Throughout our three years of development and testing, we only needed to train the model once, and it seemed to work equally well on simple as well as complex vine structures. The region-growing algorithm essentially exploits the nature of vines. It utilizes buds that are naturally present in vines as seeding points for growing regions in the segmentation process. As all vine varieties have canes with buds, with minor tweaks we expect this algorithm to be adaptable to most vine architectures with a relatively small amount of data for retraining. This could be significantly useful as vines are high-value crops, and the vine industry plays a major role in the specialty crops industry.

When pruning a vine, professional workers only keep canes that are healthy and within a certain diameter. These quality attributes of canes are currently not included in our work. Another key limitation in our current computer vision pipeline is the potential effects from wind. For the latter, we observed an average wind speed up to 12 miles/h (mph) (5.3 m/s) and some gusts up to 25 mph (11 m/s). As vines in commercial vineyards are very rigidly supported by metal posts (vertically) and trellis wire(s) (horizontally), small winds gusts (up to 12 mph or 5.3 m/s) seemed to have a minimum effect in the 3D reconstruction of prepruned vines. However, higher wind speeds could arguably cause significant issues in the registration process for any scan match-based approach, especially in vines with longer canes (not prepruned) and regions farther away from the rigid trunk and cordon supports. To minimize such an effect, we selected spur pruning to retain fewer buds per cane where the cut-points were closer to the supportive structures and were minimally affected by wind. For other pruning rules such as cane pruning where longer sections of cane need to be retained, affects from wind could be a significant problem. For the former, we currently consider all canes as healthy and viable. Although measuring cane diameter is relatively straightforward from the stereo images, assessing cane quality/health would require additional sensing ability. More advanced camera systems such as hyperspectral or thermal imaging technologies in conjunction with end-to-end deep networks could potentially provide robust solutions.

The addition of a prismatic base to the kinematic chain of the 6-DoF robot arm seems to have added several advantages. The motion planning and execution parts with the initial and final approach to the pruning locations were attempts to generate natural-looking motion of the arm as well as to cautiously interact with the vine structure. The initial planner was RRT-connect that positioned the end-effector 15 cm from the final destination. It converged to a solution in almost all cases (99%). However, the shortcomings in the manipulation end in this work are mainly attributed to the Cartesian path planner where 100% of the interpolated trajectory were sometimes not achievable. Furthermore, in some cases, the ROS's in-built Cartesian planner generated jerky motions that in some cases caused the tip/side of the end-effector to push the cane rather than securing it in-between the cutting blades.

The current state-of-the-art and the interest of the research community to control robot arms in a complex environment are reinforcement learning (RL). Similar to deep learning in computer vision, deep reinforcement learning policies tend to provide end-to-end solutions to manipulation tasks and could likely reduce dependency on heuristically set parameters and behaviors for pruning. Furthermore, with deep RL, more sophisticated capabilities that are not possible with existing sampling-based methods such as learning to prune from expert demonstrations could generalize pruning across different vine varieties and architectures. Our recent efforts in using RL policies to manipulate a robot arm for pruning can be found in (Yandun et al., 2021).

As described in Section 5.2.6, to prune the remaining 32% of vines, the robot had to repeat all processes from the other side. Although a robot arm with a longer reach could solve this issue, vine architectures with uniform cane distribution on a single side and well-defined vine to vine separation would be advantageous. Furthermore, viticultural practices are critical for reliable robotic pruning operations. Cane and spur pruning are the main pruning methods adopted by the industry. However, to maintain balance between yield, quality, and vegetative growth, accurate estimation of vine size is necessary. The estimation of vine size, which is often done by pruning weight estimation (Milkovich, 2021), determines the number of buds to retain per vine. This method of pruning is formally referred to as balance pruning, as the amount to prune is based on the capacity of the individual vine (Milkovich, 2021). None of the existing robotic prototypes, including this work, have implemented such a strategy. However, our approach of using bud detection and pruning strategy to retain a fixed number of buds per cane sets us on the right path to achieving balanced pruning.

Finally, we selected an MPC controller for navigation in this work mainly for the four following reasons: i) it has produced good results in autonomous navigation for a variety of vehicles and driving conditions (Sakhdari and Azad, 2018; Amer et al., 2017); ii) its formulation naturally allows us to constrain the optimization problem to obtain desired practical results; iii) it produced a smoother navigation in terms of overshooting and cross-track error when compared with approaches like pure pursuit in prior tests; and iv) as it is a model-driven strategy, we can increase its complexity including the dynamics of the vehicle or other variables for future research. It is worth noting that the second point was particularly useful for this application, as we limited the control effort to obtain maneuvers that reduced the risk of damaging the surrounding vegetation. Although just GPS way point following seemed robust for this application, the inclusion of a local sensor for navigation as well as safety critical features such as obstacle detection and avoidance, and compliance with farm vehicle and field workers, are currently ongoing developments.

7. Conclusions

In this work, we presented a combination of tools, techniques, and system development details of an autonomous vine pruning robot as a follow-up pruner. Highly vigorous Concord vines in a commercial vineyard were mechanically prepruned to ease robotic operations. The foci of attention here were not only to develop a system mostly utilizing off-the-shelf hardware components for a proof-of-concept prototype, but mostly to understand what is takes to robotically prune grape vines. The key technical challenges that we addressed in this work were robust imaging capability in the outdoors and data-efficient machine learning models for processing vine structures. The illumination invariant camera system proved to be a valuable component as consistent image data were acquired at any lighting condition. This also led to fewer training sample for detecting buds in images and eased 3D reconstruction. Results from the field study show that even complex vine structures could be accurately modeled from single side imaging. The integrated system robustly identified pruning location and pruned 87% of the canes successfully, with an average cycle time of 213 s/vine from two sides and 137 s/vine from one side. Improved pruning efficiency will require robustness in manipulation and advanced sensing capabilities to assess cane health and vine size for balance pruning. The mechanical design with a redundant manipulator was enough to address a single vine and could have multiple uses throughout the growing season, such as selective shoot thinning and harvesting.

Acknowledgments

This research was supported by United States Department of Agriculture, National Institute of Food and Agriculture, Specialty Crop Research Initiative under project no: USDA-NIFA-SCRI-2015-09334. The authors would like to acknowledge Zania Pothen, Tanvir Parhar, Harjatin Baweja, and the CLEREL field staff for their coordination and support during the field experiments.

ORCID

Abhisesh Silwal https://orcid.org/0000-0002-1710-6704
Francisco Yandun https://orcid.org/0000-0001-9313-7302
Anjana Nellithimaru https://orcid.org/0000-0001-5732-4688
Terry Bates https://orcid.org/0000-0002-6719-9435
George Kantor https://orcid.org/0000-0001-7088-8533

References

Abouzahir, M., Elouardi, A., Latif, R., Bouaziz, S., & Tajer, A. (2018). Embedding slam algorithms: Has it come of age? *Robotics and Autonomous Systems*, 100:14–26.

Allgöwer, F., & Zheng, A. (2012). *Nonlinear Model Predictive Control*, volume 26. Birkhäuser.

Alston, J. M., Lapsley, J. T., & Sambucci, O. (2018). Grape and Wine Production in California. *California Agriculture: Dimensions and Issues*, pages 1–28.

Amer, N. H., Zamzuri, H., Hudha, K., & Kadir, Z. A. (2017). Modelling and control strategies in path tracking control for autonomous ground vehicles: a review of state of the art and challenges. *Journal of Intelligent & Robotic Systems*, 86(2):225–254.

Bac, C., van Henten, E., Hemming, J., & Edan, Y. (2014). Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. *Journal of Field Robotics*, 31(6):888–911.

Bates, T. (2008). Pruning level affects growth and yield of New York concord on two training systems. *American Journal of Enology and Viticulture*, 59(3):276–286.

Bates, T. (2014). Mechanical crop control in new york 'concord' vineyards target desirable crop load levels. In *International Symposium on Physiological Principles and Their Application to Fruit Production 1177*, pages 259–264.

Bates, T., & Morris, J. (2009). Mechanical cane pruning and crop adjustment decreases labor costs and maintains fruit quality in New York 'concord' grape production. *HortTechnology*, 19(2):247–253.

Bates, T. R. (2017). Mechanical crop control in New York 'Concord' vineyards target desirable crop load levels. In *Acta Horticulturae*, volume 1177, pages 259–264.

Bechar, A., & Vigneault, C. (2016). Agricultural robots for field operations: Concepts and components. *Biosystems Engineering*, 149:94–111.

Botterill, T., Green, R., & Mills, S. (2013). Finding a vine's structure by bottom-up parsing of cane edges. In *2013 28th International Conference on Image and Vision Computing New Zealand (IVCNZ 2013)*, pages 112–117. IEEE.

Botterill, T., Paulin, S., Green, R., Williams, S., Lin, J., Saxton, V., Mills, S., Chen, X., & Corbett-Davies, S. (2017a). A robot system for pruning grape vines. *Journal of Field Robotics*, 34(6):1100–1122.

Botterill, T., Paulin, S., Green, R., Williams, S., Lin, J., Saxton, V., Mills, S., Chen, X. Q., & Corbett-Davies, S. (2017b). A Robot System for Pruning Grape Vines. *Journal of Field Robotics*, 34(6):1100–1122.

Calvin, L., & Martin, P. (2010). The U.S. Produce Industry and Labor: Facing the Future in a Global Economy. Technical report, U.S. Department of Agriculture.

Chen, C., He, Y., Gu, F., Bu, C., & Han, J. (2015). A real-time relative probabilistic mapping algorithm for high-speed off-road autonomous driving. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6252–6258. IEEE.

Chen, C., Wang, B., Lu, C. X., Trigoni, N., & Markham, A. (2020). A survey on deep learning for localization and mapping: Towards the age of spatial machine intelligence. *arXiv preprint arXiv:2006.12567*.

Chitta, S., Sucan, I., & Cousins, S. (2012). Moveit![ros topics]. *IEEE Robotics & Automation Magazine*, 19(1):18–19.

Corbett-Davies, S., Botterill, T., Green, R., & Saxton, V. (2012). An expert system for automatically pruning vines. In *ACM International Conference Proceeding Series*, pages 55–60. ACM.

De-An, Z., Jidong, L., Wei, J., Ying, Z., & Yu, C. (2011). Design and Control of an Apple Harvesting Robot. *Biosystems Engineering*, 110(2):112–122.

Derpanis, K. G. (2010). Overview of the ransac algorithm. *Image Rochester NY*, 4(1):2–3.

Diago, M., & Tardaguila, J. (2015). A New Robot for Vineyard Monitoring. *Wine & Viticulture Journal*, 30(3):38–42.

Ding, Y., Wang, L., Li, Y., & Li, D. (2018). Model predictive control and its application in agriculture: A review. *Computers and Electronics in Agriculture*, 151:104–117.

Donald, B. R. (1984). Motion planning with six degrees of freedom. Technical report, MASSACHUSETTS INST OF TECH CAMBRIDGE ARTIFICIAL INTELLIGENCE LAB.

Ebrahimi, M. A., Khoshtaghaza, M. H., Minaei, S., & Jamshidi, B. (2017). Vision-based pest detection based on SVM classification method. *Computers and Electronics in Agriculture*, 137:52–58.

Economic Research Service (2016). Annual Report FY 2017. https://www.ers.usda.gov/about-ers/plans-and-accomplishments/ers-annual-report-fy-2017/.

Elbaz, G., Avraham, T., & Fischer, A. (2017). 3d point cloud registration for localization using a deep neural network auto-encoder. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4631–4640.

Fennimore, S., & Doohan, D. (2008). The Challenges of Specialty Crop Weed Control, Future Directions. *Weed Technology*, 22(2):364–372.

Fidelibus, M., El-kereamy, A., Zhuang, G., Haviland, D., Hembree, K., & Stewart, D. (2018). Sample Costs to Establish and Produce Table Grapes. Technical report, University of California, Davis.

Fitzgibbon, A. W. (2003). Robust registration of 2d and 3d point sets. *Image and Vision Computing*, 21(13-14):1145–1153.

Fleet, A. (2021). National average gas prices rise steadily in 2021. https://www.automotive-fleet.com/10134992/national-average-gas-prices-rise-steadily-in-2021.

Fountas, S., Mylonas, N., Malounas, I., Rodias, E., Hellmann Santos, C., & Pekkeriet, E. (2020). Agricultural robotics for field operations. *Sensors*, 20(9):2672.

Gollakota, A., & Srinivas, M. (2011). Agribot-A Multipurpose Agricultural Robot. In *Annual IEEE India Conference*, pages 1–4, Hyderabad, India. IEEE.

Gongal, A., Amatya, S., Karkee, M., Zhang, Q., & Lewis, K. (2015). Sensors and Systems for Fruit Detection and Localization: A Review. *Computers and Electronics in Agriculture*, 116:8–19.

He, L., & Schupp, J. (2018). Sensing and Automation in Pruning of Apple Trees: A Review. *Agronomy*, 8(10):1–18.

Hellman, E. (2019). Parts of the grape vine: Flowers and fruit. https://grapes.extension.org/parts-of-the-grape-vine-flowers-and-fruit/(Accessed: August 2021).

Houle, D., Govindaraju, D. R., & Omholt, S. (2010). Phenomics: the next challenge. *Nature Reviews Genetics*, 11(12):855–866.

Howell, G. S. (2001). Sustainable grape productivity and the growth-yield relationship: A review. *American Journal of Enology and Viticulture*, 52(3):165–174.

Jeh, G., & Widom, J. (2002). Simrank: a measure of structural-context similarity. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 538–543.

Johnson, J., & CourtneyRoss (2016). The Cost of Vineyard Labor. https://www.goodfruit.com/the-cost-of-vineyard-labor/(Accessed: November 2021).

Karantanellis, E., Arav, R., Dille, A., Lippl, S., Marsy, G., Torresani, L., & Elberink, S. O. (2020). Evaluating the quality of photogrammetric point-clouds in challenging geo-environments–a case study in an alpine valley. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43:1099–1105.

Katyara, S., Ficuciello, F., Caldwell, D. G., Chen, F., & Siciliano, B. (2021). Reproducible pruning system on dynamic natural plants for field agricultural robots. In *Human-Friendly Robotics 2020: 13th International Workshop*, pages 1–15. Springer.

Kazmi, W., Foix, S., Alenyà, G., & Andersen, H. J. (2014). Indoor and outdoor depth imaging of leaves with time-of-flight and stereo vision sensors: Analysis and comparison. *ISPRS journal of photogrammetry and remote sensing*, 88:128–146.

Kicherer, A., Klodt, M., Sharifzadeh, S., Cremers, D., Töpfer, R., & Herzog, K. (2017). Automatic image-based determination of pruning mass as a determinant for yield potential in grapevine management and breeding. *Australian Journal of Grape and Wine Research*, 23(1):120–124.

King, J. (2021). Pruning Grapes in Home Gardens: Some Basic Guidelines. Technical report, Washington State University.

Kuffner, J. J., & LaValle, S. M. (2000). Rrt-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*, volume 2, pages 995–1001. IEEE.

Li, P., Lee, S. H., & Hsu, H. Y. (2011). Review on fruit harvesting method for potential use of automatic fruit harvesting systems. In *Procedia Engineering*, volume 23, pages 351–366.

Li, Y., Xia, C., & Lee, J. (2009). Vision-based Pest Detection and Automatic Spray of Greenhouse Plant. In *IEEE International Symposium on Industrial Electronics*, pages 920–92, Seoul. IEEE.

Marin, R. D. C., Botterill, T., & Green, R. D. (2015). Gibbs sampling for 2d cane structure extraction from images. In *2015 6th International Conference on Automation, Robotics and Applications (ICARA)*, pages 461–465.

Milkovich, M. (2021). Using pruning power for vine balance. https://www.goodfruit.com/using-pruning-power-for-vine-balance/(Accessed: September 2021).

Millan, B., Paz, D., Aquino, A., Palacios, F., & Tardaguila, J. (2019). Vineyard pruning weight assessment by machine vision: towards an on-the-go measurement system. *OENO One*, 53(2).

Moon, D., Chung, S., Kwon, S., Seo, J., & Shin, J. (2019). Comparison and utilization of point cloud generated from photogrammetry and laser scanning: 3d world model for smart heavy equipment planning. *Automation in Construction*, 98:322–331.

Morris, J. R. (2007). Development and commercialization of a complete vineyard mechanization system. *HortTechnology*, 17(4):411–420.

Mousazadeh, H. (2013). A technical review on navigation systems of agricultural autonomous off-road vehicles. *Journal of Terramechanics*, 50(3):211–232.

National Agricultural Statistics Service (2019). 2017 Census of Agriculture. Technical report, United States Department of Agriculture.

Papadakis, P. (2013). Terrain traversability analysis methods for unmanned ground vehicles: A survey. *Engineering Applications of Artificial Intelligence*, 26(4):1373–1385.

Paulin, S., Botterill, T., Lin, J., Chen, X., & Green, R. (2015). A comparison of sampling-based path planners for a grape vine pruning robot arm. In *2015 6th International Conference on Automation, Robotics and Applications (ICARA)*, pages 98–103. IEEE.

Pothen, Z., & Nuske, S. (2016). Automated assessment and mapping of grape quality through image-based color analysis. *IFAC-PapersOnLine*, 49(16):72–78.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 28:91–99.

Sakhdari, B., & Azad, N. L. (2018). Adaptive tube-based nonlinear mpc for economic autonomous cruise control of plug-in hybrid electric vehicles. *IEEE Transactions on Vehicular Technology*, 67(12):11390–11401.

Silwal, A., Parhar, T., Yandun, F., & Kantor, G. (2021). A robust illumination-invariant camera system for agricultural applications. *arXiv preprint arXiv:2101.02190*.

State, N. Y. (2021). New York State's Minimum Wage. https://www.ny.gov/new-york-states-minimum-wage/new-york-states-minimum-wage(Accessed: November 2021).

Stewart, G. W. (1993). On the early history of the singular value decomposition. *SIAM Review*, 35(4):551–566.

Striegler, R. K. (2021). Mechanization in the vineyard. https://viticulture.unl.edu/viticulture/2014%20Vineyard%20Mechanization%20NE%20Winery%20and%20Grape%20Growers%20Forum.pdf(Accessed: November 2021).

Sunusi, I. I., Zhou, J., Wang, Z. Z., Sun, C., Ibrahim, I. E., Opiyo, S., Soomro, S. A., Sale, N. A., Olanrewaju, T., et al. (2020). Intelligent tractors: Review of online traction control process. *Computers and Electronics in Agriculture*, 170:105176.

Tabb, A., & Medeiros, H. (2017a). A robotic vision system to measure tree traits. In *IEEE International Conference on Intelligent Robots and Systems*, volume 2017-Sept, pages 6005–6012. IEEE.

Tabb, A., & Medeiros, H. (2017b). A robotic vision system to measure tree traits. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6005–6012. IEEE.

Tabb, A., & Medeiros, H. (2018). Automatic segmentation of trees in dynamic outdoor environments. *Computers in Industry*, 98:90–99.

Uzes, D. M., & Skinkis, P. A. (2016). Factors influencing yield management of pinot noir vineyards in oregon. *Journal of Extension*, 54(3):11.

Vision Robotics Corp. (2020). Intelligent Autonomous Grapevine Pruner. https://www.visionrobotics.com/vr-grapevine-pruner(Accessed: November 2021).

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.

Wurm, K. M., Hornung, A., Bennewitz, M., Stachniss, C., & Burgard, W. (2010). Octomap: A probabilistic, flexible, and compact 3d map representation for robotic systems. In *Proceedings of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, volume 2.

Yandun, F., Parhar, T., Silwal, A., Clifford, D., Yuan, Z., Levine, G., Yaroshenko, S., & Kantor, G. (2021). Reaching pruning locations in a vine using a deep reinforcement learning policy. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE.

Zhang, Z., Gerke, M., Vosselman, G., & Yang, M. Y. (2018). A patch-based method for the evaluation of dense image matching quality. *International Journal of Applied Earth Observation and Geoinformation*, 70:25–34.

How to cite this article: Silwal, A., Yandun, F., Nellithimaru, A., Bates, T., & Kantor, G. (2022). Bumblebee: A path towards fully autonomous robotic vine pruning. *Field Robotics, 2*, 1661–1696.

Appendix: A brief economic study to compare cost between different pruning technologies

This appendix summarizes a brief economic study that highlights the cost associated with manual pruning operation and the benefits of robotic pruning both as a hand follow-up operation and a fully independent and autonomous system. Table 7 shows typical vineyard dimensions for three varieties of vine, including Riesling, Concord, and Vignoles, which are commonly grown across the United States. The variables evaluated include row width, vine separation, cordon height, number of trellis wires, and average number of vines per acre. These measurements were taken at the same commercial vineyard site used for testing the robotic pruner described in this paper. To quantify human pruning speed, an experienced professional pruner was asked to prune 30 sample of each vine variety at a regular pace. On average, it took 150 seconds to prune a vine with an average of 14 canes to cut.

The numeric values (see Tables 8 and 9) used in this brief study are mostly inspired from (Striegler, 2021). Tables 8 and 9 have more details on additional numeric values and their function in the calculation and comparison of four types of pruning technology viz. hand pruning, mechanically assisted hand pruning, mechanically assisted robotic pruning, and fully autonomous pruning. Additionally, we also assumed 25% increase on any outdated costs and wages used by (Striegler, 2021). We also assume minimum wages of $15 per hour [based on NY minimum wage (State, 2021)] and the cost of fuel is taken as a national average of $2.38 for 2021 from (Fleet, 2021). For robotic pruning, we assume the robot runs on fuel for an 18-h-long operation to match the labor counterparts of three pruning crews working 6 hours each. The calculations are detailed in Tables 8 and 9. Our calculations show that the cost of hand pruning per acre was $672, which was the highest cost amongst all technologies, as expected. In mechanically assisted pruning, a dedicated hardware system preprunes vines, which eases the work environment for pruners at the cost of the mechanical system. Here, the cost of mechanically assisted and hand follow-up operation together summed up to $459.25 per acre. In mechanically assisted robotic pruning, robots autonomously perform the follow-up pruning operation after vines are mechanically prepruned. Although at a glance it seems

Table 7. Various vine architecture measurements in the test vineyards.

	Vine type			
Measurement unit: feet (m)	**Riesling**	**Vignoles**	**Concord**	**Average**
Row width (W)	9 (2.74)	9 (2.74)	9 (2.74)	9 (2.74)
Vine separation (L)	6 (1.83)	4 (1.22)	9 (2.74)	6 (1.83)
Cordon height (H)	2 (0.61)	6 (1.83)	6 (1.83)	6 (1.83)
No. of trellis	4	1	1	-
No of vines per Acre	807	1210	537	850
Average cane per vine	11	13	18	14
Average manual pruning speed (sec/vine)	142.8	95.4	214.2	150

Table 8. Cost associated with hand and mechanically assisted pruning.

Hand pruning		**Mechanically assisted pruning**	
		Mechanical prepruning	**Hand follow-up**
Labor (H/Acre)	32 hrs.	0.65 equipment h + 2.4 labor hour	3 laborers for total of 18 h
Minimum wage (per h)	$15		
Benefit	40%		
Direct cost (per h)	$21		
Total Cost (per acre)	**$672**	**$81.25**	**$378**
		Combined Cost (per acre)	**$459.25**
Initial investment*	-	$30 000	-

Table 9. Cost associated with robotic pruning.

Fully Autonomous Pruning		**Mechanical Prepruning**
Labor (H/Acre)	-	
Minimum wage (per h)	-	
Benefit	-	
Run time	18 h.	0.65 equipment h + 2.4 labor hour
Robot fuel (9 gals @ 0.5 gals/h)	$21.42	
Generator fuel (9 gals @ 0.5 gals/h)	$21.42	
Lubrication cost ($2.1/h)	$37.80	
Total Cost (per acre)	**$80.64**	$81.25
Combined Cost (per acre)		**$161.90**
Initial investment*	**$115 000**	$30 000

*The robotic pruning prototype is estimated to cost $115,000 US. The $30,000 US cost is the estimated value of the OXBO pruning head attachable to farm vehicles.

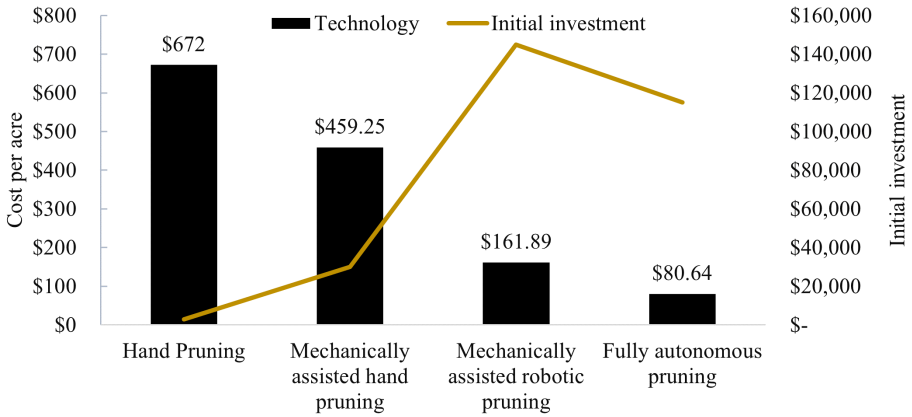


Figure 23. Cost breakdown for different pruning technologies.

to increase initial investments, it drastically decreased per acre cost of pruning to $161.9. With a fully autonomous system capable of pruning vines end-to-end without any assistance, the cost of pruning goes down to just $80.64 per acre. As evident from Tables 8 and 9, and depicted in Figure 23, assisted robotic technology or fully independent robotic solutions could decrease the cost of pruning by nearly 4 to 8 times, respectively. Thus, robotic systems for pruning have a clear advantage over existing technology/practice, and in today's economy and in the long-term they could prove to be profitable.