2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Optimal view point and kinematic control for grape stem detection and cutting with an in-hand camera robot

Sotiris Stavridis* and Zoe Doulgeri

Abstract— **In this work, a methodology to find the best view of a grape stem and approach angle in order to crop it is proposed. The control scheme is based only on a classified point cloud obtained by the in-hand camera attached to the robot's end effector without continuous stem tracking. It is shown that the proposed controller finds and reaches the optimal view point and subsequently the stem fast and efficiently, accelerating the overall harvesting procedure. The proposed control scheme is evaluated through experiments in the lab with a UR5e robot with an in-hand RealSense camera on a mock-up vine.**

I. Introduction

Recent advances in sensing, autonomy, learning and control have lead to an increasing use of robotics in agriculture [1], [2]. The workspace in the agricultural sector, i.e. farming fields, greenhouses, is characterized by uncertainties, unknowns and unmodelled states. Furthermore, the crops and their stems are usually occluded by the environment around them which negatively affects their visibility by a perception system and the ease of reaching to either examine or crop to collect them. Moreover, the variability between the individual fruits cannot be accurately modeled due to the structural and position uncertainties that are present. Especially for grape harvesting, the complex structure of the vineyards, i.e presence of trunks, branches, heavy foliage and supporting structures, as well as the clustered berries fruits, deems it more challenging for harvesting and only a few works in the literature consider it [3], [4]. An autonomous mobile manipulator has been developed in [3] that performs grape harvesting in a vineyard, while in [4] a mobile robot with two arms working in parallel each harvesting a grape is deployed in a vineyard. However, both of these works do not consider the problem of occlusions during harvesting, which is reported as a challenging and non-trivial issue in agricultural robotics [5]–[7].

In this work we consider the problem of finding the optimal feasible view point in order to properly view the stem of a grape and reach to cut it. A method is developed that maximizes the distance of the stem view from surrounding occlusions and subsequently finds the best cutting pose to crop the stem given the viewed scene, while respecting the kinematic constraints of the robot. In this paper, preliminary in-lab experiments with a manipulator on a mock-up grape vine are presented.

The research leading to these results has received funding from the European Community's Framework Programme Horizon 2020 under grant agreement No 871704, project BACCHUS.

Authors are with the Automation & Robotics Lab, School of Electrical & Computer Engineering, Aristotle University of Thessaloniki, Greece. Emails: {sotistav@ece.auth.gr, doulgeri@ece.auth.gr}
*Corresponding Author

In general, many methods have been proposed for avoiding occlusions to unveil and properly view an object of interest with an in-hand camera robot in various applications. They may be divided into two main categories: a) reactive approaches that modify a given trajectory to increase the visibility of the object of interest [6]–[10], and b) view planning approaches that sequentially plan view points to optimally view the object of interest [11]–[14].

In the agricultural sweet pepper harvesting scenario discussed in [6], an optimization function incorporates the pixels of the visible object of interest and a measure of manipulability. However, to compute the gradient of the optimization function, a 9 camera grid fixture attached to the robot's end-effector was employed. Extending [6], a machine learning strategy is introduced in [7] that employs a Convolutional Neural Network (CNN) to estimate the gradient of the optimization function. This adaptation allows the method to function with just a single camera setup. However, the network was trained and tested only through simulations, limiting its applicability in the actual field. A control-based approach for approaching and unveiling grape stems was proposed in [8]. This method relies on continuous camera feedback and stem recognition, requiring good lighting conditions and reliable stem recognition models. A reactive strategy for occlusion avoidance during target tracking using an autonomous drone equipped with a camera was developed in [9]. The obstacles are represented as spheres and the target is defined as a point with a predetermined position, while also requiring complete knowledge of the environment and the robot dynamics. In [10], an RGB-D based occlusion avoidance method of an object of interest was proposed. However, a complete view of the region of interest is required at the initial sensor position, as well as object recognition at each time frame.

In [11], an exploration-based view planning method is proposed that optimizes a scene coverage metric with a path length cost function to unveil an occluded grape stem in a few steps. However, singularity constraints are not considered and the solution is computational heavy and time consuming, owing to the exploratory strategy. In [12], a strategy was developed that switches between exploration sampling in order to detect the fruits and sampling around the detected fruits to determine their size and position. A voxel map of fruit regions is created to sample candidate view points around them which are evaluated based on the expected information gain by applying a heuristic utility function. In [13] a spherical projection is used to find an unoccluded view to a point of an object of interest and

iteratively compute the next best view for full scene coverage by locally maximizing the number of connected vertices to the current view of the generated visibility graph. In [14], a task involving eye-in-hand grasping is examined, employing a CNN to generate a grasping feasibility map. However, to determine the optimal viewpoint, the approach necessitates exploring the task space, involving traversing through different viewpoints and computing the grasping map. A different approach is introduced in [15], to physically unveil the stem. Given an RGB image from the in hand camera robot, a deep neural network is trained to generate a reference trajectory to physically unveil the stem and the grape by a second manipulator.

In this paper, in contrast to the aforementioned works, instead of adopting a time consuming exploratory strategy, or a reactive approach that requires continuous camera feedback, we propose a methodology to find the optimal view point around a grape stem that facilitates the harvesting procedure without requiring any further feedback from the camera. Additionally, our solution respects the kinematic constraints of the robot which is also not considered in previous works.

II. Problem Description & Proposed Solution

We consider a velocity controlled robot with joint position $\mathbf{q} \in \mathbb{R}^n$ equipped with a camera and a cutter with position and rotation matrix with respect to the world denoted by $\{\mathbf{p}_{cam}, \mathbf{R}_{cam}\}$ and $\{\mathbf{p}_{cut}, \mathbf{R}_{cut}\}$. We assume a perception system that can classify the points in the workspace that belong to the target grape, and its stem while also detecting the surrounding occlusions, e.g. [16], [17] respectively. We further assume that the robot starts from an initial position in front of the grapevine with the target grape being detected by the in-hand camera. This can be easily achieved with a mobile manipulator. Additionally, we assume partial visibility of the stem when the camera is at a close distance.

We address the problem of finding the optimal view point $\mathbf{p}_F$, away from occlusions and robot singularities based on the perception information, namely the point cloud data acquired by the in-hand camera while centering the camera view to the stem's position $\mathbf{p}_{sb}$. Moreover, we address the problem of finding the best stem cutting pose $\{\mathbf{p}_{cut,d}, \mathbf{R}_{cut,d}\}$ to crop the stem given the viewed scene, while respecting the kinematic constraints of the robot. Notice that the grasping of the grape is not considered in this work, however it may be performed either unimanually with an integrated cutter/gripper [18], or bimanually by a second manipulator equipped with a suitable gripper[1].

Our proposed solution includes an initial camera approaching strategy to achieve partial stem visibility. By moving closer towards the grape at a proper perception distance d we can ensure a reliable detection of points belonging to the stem leading to a stem location estimate. After the optimal view point $\mathbf{p}_F$ is found the in-hand camera reaches it with its view axis centered to the stem $\mathbf{p}_{sb}$ so that the stem is now properly viewed and its position $\mathbf{p}_{sb}$ updated if needed. Given this view the best cutting pose is calculated and can be in general reached for harvesting. Details of the proposed solution are given in the following subsections.

[1]https://cordis.europa.eu/project/id/871704 & https://bacchus-project.eu/

Point-cloud Subsets	Description
whole scene $\mathcal{W}$	point cloud of the whole scene
stem $\mathcal{S}$	point cloud of the stem $\mathcal{S} \subseteq \mathcal{W}$
obstacles $\mathcal{O}$	non-stem points $(\mathcal{W} - \mathcal{S})$ within r_o distance from $\mathbf{p}_{sb}$
proj. obstacles $\mathcal{O}_p$	Point in $\mathcal{O}$ projected upon the unit sphere centered at $\mathbf{p}_{sb}$
Sampled sphere $\mathcal{G}$	Sampled points on the unit sphere centered at $\mathbf{p}_{sb}$
Sampled proj. obstacles $\mathcal{O}_n$	Each point in $\mathcal{O}_p$ is assigned to its closest in $\mathcal{G}$, $\mathcal{O}_n \subseteq \mathcal{G}$
View points $\mathcal{F}$	Non-obstacle points in $\mathcal{G}$, $\mathcal{G} - \mathcal{O}_n$
Front View points $\mathcal{F}_h$	Viewpoints between camera and vine, $\mathcal{F}_h \subseteq \mathcal{F}$
Reachable View points $\mathcal{F}_r$	Non singular viewpoints, $\mathcal{F}_r \subseteq \mathcal{F}_h$

TABLE I: Point-cloud Classification.

A. Point cloud processing for optimal view point computation

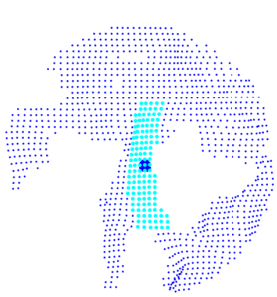
Table I summarizes the classified/calculated point-cloud subsets. Set $\mathcal{W}$ defines the set of the point-cloud points captured by the camera, subset $\mathcal{S} \subseteq \mathcal{W}$ denotes the point-cloud of the detected stem and its mean point considered as the stem position $\mathbf{p}_{sb}$. Since only the area surrounding the stem is of interest, from the overall scene point cloud $\mathcal{W}$, we filter out all points that are above a certain distance r_o from $\mathbf{p}_{sb}$ and remove the stem points $\mathcal{S}$ resulting in the set $\mathcal{O}$ which defines the set of potential occlusions (Fig.1a). Each point $\mathbf{p}_{o_i} \in \mathcal{O}$ with $i = 1, ..., n_O$ is projected on the unit sphere centered at $\mathbf{p}_{sb}$ using $proj_{\mathrm{S}}(\mathbf{p}_{o_i}) = \frac{\mathbf{p}_{o_i} - \mathbf{p}_{sb}}{\|\mathbf{p}_{o_i} - \mathbf{p}_{sb}\|} + \mathbf{p}_{sb}$ to yield the set $\mathcal{O}_p$ (Fig.1b). The optimal view point which is the point farthest from all occlusion points in $\mathcal{O}_p$ and may be found by solving the following optimization problem

$$\mathbf{p}^* = \underset{\mathbf{p} \in \mathrm{S}(\mathbf{p}_{sb})}{\operatorname{argmax}} \left\{ \min_{\forall i} \{\|\mathbf{p} - proj_{\mathrm{S}}(\mathbf{p}_{o_i})\|\} \right\}, \qquad (1)$$

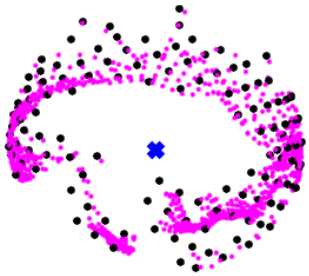
with some proved optimization method, e.g [13], where $\mathrm{S}(\mathbf{p}_{sb})$ denotes the unit sphere centered at $\mathbf{p}_{sb}$.

To reduce the complexity of (2), we sample the sphere into n_l number of points yielding set $\mathcal{G}$, using the Fibonacci lattice methodology [19] which provides a near optimal way to sample a sphere, as well as an efficient way to compute the nearest sampled points of a given point on the sphere. Each point in $\mathcal{O}_p$ is assigned to the nearest point of $\mathcal{G}$ yielding set $\mathcal{O}_n \subseteq \mathcal{G}$ (Fig.1b). The points of $\mathcal{G}$ that do not belong in $\mathcal{O}_n$ is the set of candidate stem view points $\mathcal{F}$ (Fig.1c) as they do not occlude the stem point $\mathbf{p}_{sb}$.

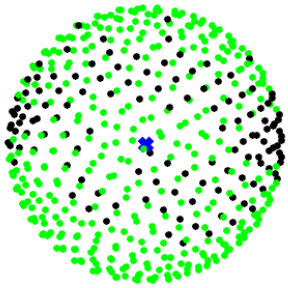
Due to the structure of the vineyards, reaching behind the rows would lead to collisions with the trunks, branches and foliage, as well as the supporting structures of the grapevine. We apply Principle Component Analysis (PCA) on subset $\mathcal{O}$ to fit a representative plane in order to split the scene into two



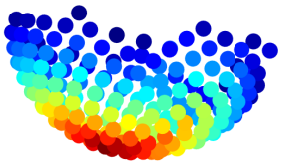
(a) Scene's point-cloud $\mathcal{W}$. Obstacles subset $\mathcal{O}$: blue. Stem subset $\mathcal{S}$: cyan. $\mathbf{p}_{sb}$: Blue $\times$



(b) Proj.obstacles $\mathcal{O}_p$: magenta. Sampled points near proj.obstacles $\mathcal{O}_n$: black.



(c) Sphere surface sampling ($n_l = 500$). View points $\mathcal{F}$: green. Near-obstacles sampled points $\mathcal{O}_n$: black.



(d) View points spherical mapping: Red to blue - Farthest to closest from occlusions.

Fig. 1: The point-cloud sets used in this work.

areas, the area between the camera and the grapevine and the area behind the grapevine. The points in $\mathcal{F}$ that belong to the front area are considered valid, while the points in the back are considered unreachable and are removed from $\mathcal{F}$ yielding $\mathcal{F}_h \subseteq \mathcal{F}$, ensuring that the determined view points always reside within the camera's clearly visible field of view. The resulting plane is placed on $\mathbf{p}_{sb}$, hence the previously defined sphere becomes a hemisphere. In Fig.1*d* a hemispherical mapping is shown that visualizes the minimum distance of every point in $\mathcal{F}$ from the nearest point in $\mathcal{O}_n$.

However, some of the candidate view points might be unreachable due to singularity constraints of the manipulator. A singularity metric is incorporated in $\mathcal{F}_h$ in order to find the reachable best stem view point. The singularity metric that is utilized in this work is the condition number $cond(\mathbf{J})$ which is the ratio of the maximum over the minimum singular value of the robot Jacobian. Other metrics of singularity or manipulability may be used as well [20]. View points in $\mathcal{F}_h$ that are below a certain threshold c_{th} at both the view point position and its respective cutting pose, which is calculated as explained in the next section, are removed from $\mathcal{F}_h$ resulting in the subset of the reachable candidate view points $\mathcal{F}_r \subseteq \mathcal{F}_h$. The point $\mathbf{p}_F$ with the largest value/distance (dark red in Fig.1d) is the solution to:

$$\mathbf{p}_F = \underset{\mathbf{p}\in\mathcal{F}_r}{\operatorname{argmax}} \left\{ \min_{\forall i} \{\|\mathbf{p} - \mathbf{p}_{n_i}\|\} \right\}. \tag{2}$$

where $\mathbf{p}_{n_i} \in \mathcal{O}_n$ with $i = 1, ..., n_n$, which is a problem similar to (2), but using the sampled point cloud sets.

After determining point $\mathbf{p}_F$, the perception distance d of $\mathbf{p}_{sb}$ from the camera may be selected by scaling $\mathbf{p}_F$ from the unit sphere using:

$$\mathbf{p}_{F,d} = \frac{\mathbf{p}_F - \mathbf{p}_{sb}}{\|\mathbf{p}_F - \mathbf{p}_{sb}\|} d + \mathbf{p}_{sb}. \tag{3}$$

Remark 1. *Notice that by using the sampling approach the number of occlusions points is significantly reduced from n_O, which is usually large $n_O \gg n_l$, to $n_n < n_l$, allowing for a quick solution. Additionally, since n_l is a user selected parameter, it may be chosen small or large enough depending on the time and accuracy requirements of the particular task.*

B. Reaching & centering kinematic control

In order to control the distance from the camera towards the stem and its position around the stem independently, it is convenient to transform the problem into spherical coordinates and design the control law in the spherical coordinate system. The origin of the spherical system is placed at the stem point $\mathbf{p}_{sb}$. The best view point $\mathbf{p}_{F,d}$ and the camera position $\mathbf{p}_{cam}$ are thus transformed into the spherical coordinate system using:

$$\begin{bmatrix} \rho \\ \theta \\ \phi \end{bmatrix} = \begin{bmatrix} \|\mathbf{p}\| \\ \operatorname{atan2}(p_y, p_x) \\ \operatorname{atan2}(\sqrt{p_x^2 + p_y^2}, p_z) \end{bmatrix} \tag{4}$$

for a given point $\mathbf{p} = [p_x, p_y, p_z]$, while ρ, θ and ϕ are the radial distance, the azimuth angle and the polar angle respectively. To generate the reference velocity for the end-effector, we utilize a first order dynamical system for each spherical coordinate.

$$\mathbf{v}_{sr} = \begin{bmatrix} -k_\rho(\rho - \rho_d) \\ -k_\theta(\theta - \theta_d) \\ -k_\phi(\phi - \phi_d) \end{bmatrix} \tag{5}$$

where k_ρ, k_θ and k_ϕ are positive control gains for the radial, azimuthal, and polar coefficients respectively, while ρ_d, θ_d and ϕ_d are the respective spherical coordinates of the target pose. The spherical reference velocity $\mathbf{v}_{sr}$ is transformed back to the Cartesian coordinates with:

$$\mathbf{v}_{cr} = \begin{bmatrix} \frac{p_x}{\|\mathbf{p}\|} & \frac{p_y}{\|\mathbf{p}\|} & \frac{p_z}{\|\mathbf{p}\|} \\ \frac{-p_y}{p_x^2+p_y^2} & \frac{p_x}{p_x^2+p_y^2} & 0 \\ \frac{p_x p_z}{\rho^2\sqrt{p_x^2+p_y^2}} & \frac{p_y p_z}{\rho^2\sqrt{p_x^2+p_y^2}} & -\frac{p_x^2+p_y^2}{\rho^2\sqrt{p_x^2+p_y^2}} \end{bmatrix}^{-1} \mathbf{v}_{sr} \tag{6}$$

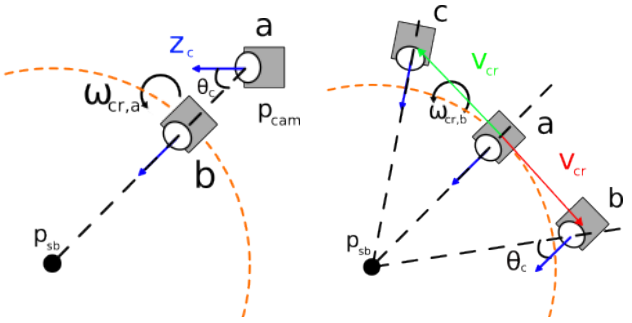
for a given point $\mathbf{p}$. For the centering objective, the following control signal is utilized:

$$\omega_{cr} = \underbrace{k_c \theta_c \mathbf{k}}_{\omega_{cr,a}} + \underbrace{\mathbf{S}(\mathbf{v}_{cr}) \frac{\mathbf{e}_{pc}}{\|\mathbf{e}_{pc}\|^2}}_{\omega_{cr,b}} \tag{7}$$

where $\mathbf{e}_{pc} = \mathbf{p}_{sb} - \mathbf{p}_{cam}$, $k_c \in \mathbb{R}_{>0}$ is a constant positive gain, $\theta_c \in [0, \pi/2]$ and $\mathbf{k}$ are the angle and axis of the minimum rotation between $\mathbf{e}_{pc}$ and the camera view axis $\mathbf{z}_c$, which can be calculated by the following expressions:

$$\theta_c = \cos^{-1}\left(\frac{\mathbf{z}_c^\intercal \mathbf{e}_{pc}}{\|\mathbf{e}_{pc}\|}\right), \quad \mathbf{k} = \frac{\mathbf{S}(\mathbf{z}_c)\mathbf{e}_{pc}}{\|\mathbf{S}(\mathbf{z}_c)\mathbf{e}_{pc}\|} \tag{8}$$

with $\mathbf{S}(\mathbf{z})$ the skew symmetric matrix of the corresponding vector $\mathbf{z}$. The first part $\omega_{cr,a}$ regulates the centering of the stem to the camera (Fig.2a), while the second $\omega_{cr,b}$ compensates for the translational velocity of the camera. Figure 2b depicts the effect of $\omega_{cr,b}$. From the initial position (a) solely with camera translational motion, position (b) is reached with misalignment, whereas when using $\omega_{cr,b}$ the robot reaches position (c) with the correct alignment.



(a) $\omega_{cr,a}$: Camera rotates to align its z-axis with $\mathbf{p}_{sb} - \mathbf{p}_{cam}$ **(b)** Green: with $\omega_{cr,b}$, Red: without $\omega_{cr,b}$

Fig. 2: Effect of the centering signal (7).

Finally, to compute the reference joint velocities to be commanded to the robot

$$\dot{\mathbf{q}}_r = \mathbf{J}^{\dagger}\mathbf{V}, \tag{9}$$

is used, where $\mathbf{J} = \mathbf{J}_{cam}$ is the robot Jacobian at the camera frame and $\mathbf{V} = \mathbf{V}_c = \begin{bmatrix}\mathbf{v}_{cr}^{\intercal} & \omega_{cr}^{\intercal}\end{bmatrix}^{\intercal}$.

After finding the best view point and viewed the stem from it, to facilitate the cutting motion the cutter should adopt a proper orientation with respect to the stem to reach to cut it. Let the orientation of the cutter be expressed by the rotation matrix $\mathbf{R}_{cut} = \begin{bmatrix}\mathbf{a}_c & \mathbf{u}_c & \mathbf{n}_c\end{bmatrix}$ and let $\mathbf{s}$ be the axis of the stem with $\mathbf{p}_{sb}$ as origin which, assuming that it lies on a near straight line, may be computed by applying PCA on the stem point cloud $\mathcal{S}$. As depicted in Fig.3, instead of aiming for the cutter to be orientated normal to the stem axis $\mathbf{s}$, which may lead to collisions, as seen in Fig.3b, we aim to align the cutter axis $\mathbf{n}_c$ with $\mathbf{p}_{sb} - \mathbf{p}_{F,d}$ and the cutter axis $\mathbf{a}_c$ with the normal vector of the plane defined by vectors $\mathbf{s}$ and $\mathbf{p}_{sb} - \mathbf{p}_{F,d}$ (Fig.3c) equal to

$$\mathbf{s}_n = \frac{\mathbf{S}(\mathbf{s})(\mathbf{p}_{sb} - \mathbf{p}_{F,d})}{\|\mathbf{S}(\mathbf{s})(\mathbf{p}_{sb} - \mathbf{p}_{F,d})\|}. \tag{10}$$

Thus the target cutting orientation is expressed by the rotation matrix:

$$\mathbf{R}_{cut,d} = \begin{bmatrix}\mathbf{s}_n & -\mathbf{S}(\mathbf{s}_n)\frac{\mathbf{p}_{sb}-\mathbf{p}_{F,d}}{\|\mathbf{p}_{sb}-\mathbf{p}_{F,d}\|} & \frac{\mathbf{p}_{sb}-\mathbf{p}_{F,d}}{\|\mathbf{p}_{sb}-\mathbf{p}_{F,d}\|}\end{bmatrix} \tag{11}$$

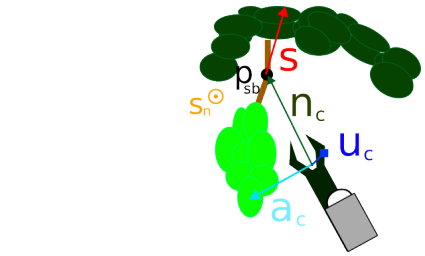
In this way a linear path approach and minimum rotation would suffice to achieve the target.

After aligning with the stem, the cutter reaches the stem point $\mathbf{p}_{cut,d} = \mathbf{p}_{sb}$ to crop it. For the translational part (5) is used to compute the reference cutter velocity $\mathbf{v}_{cut}$, while for the computation of the reference rotational velocity the following quaternion error is used:

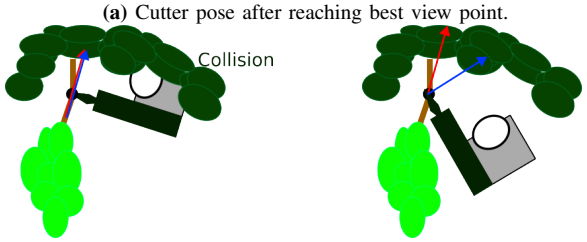
$$\begin{bmatrix}0 & \omega_{cut}^{\intercal}\end{bmatrix}^{\intercal} = -k_l log_Q(\mathbf{Q}_{cut}\mathbf{Q}_{cut,d}^{-1}) \tag{12}$$

where log_Q is the quaternion logarithm function, and $\mathbf{Q}_{cut}$ and $\mathbf{Q}_{cut,d}$ are the Quaternion forms of the rotation matrices $\mathbf{R}_{cut}, \mathbf{R}_{cut,d}$ respectively.

The reference joint velocities to be commanded to the robot are then computed using (9) with $\mathbf{J} = \mathbf{J}_{cut}$ the robot Jacobian at the cutter frame and $\mathbf{V} = \mathbf{V}_{cut} = \begin{bmatrix}\mathbf{v}_{cut}^{\intercal} & \omega_{cut}^{\intercal}\end{bmatrix}^{\intercal}$.



(a) Cutter pose after reaching best view point.



(b) Probable collision if $\mathbf{u}_c$ fully aligns with $\mathbf{s}$. **(c)** Behaviour when using (11)

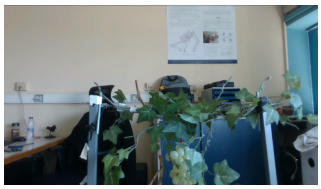
Fig. 3: Difference between aligning the cutter up axis $\mathbf{u}_c$ with the stem axis $\mathbf{s}$ (b) and aligning $\mathbf{a}_c$ with $\mathbf{s}_n$ (c).

III. In-Lab Experimental Validation

The efficacy of the proposed methodology is demonstrated in a lab experiment utilizing a mock-up vine with plastic leaves and grapes, as shown in Fig.4. A UR5e robot was utilized equipped with a Realsense d415 RGB-D camera and a non-articulated 3d printed cutter as the end-effector. Various cases of occlusions can be seen in the attached video with expected optimal view points to the left, right and front of the stem have been considered. The grape is positioned at $\mathbf{p}_g = [-0.86 \quad -0.06 \quad 0.735]^{\intercal}$ with a small variance between the runs and the initial position and orientation of the camera is such that the camera is facing the grape i.e. $\mathbf{p}_{cam,0} = [-0.2984 \quad -0.0584 \quad 0.8807]^{\intercal}$ and $\mathbf{Q}_{cam,0} = [0.5063 \quad -0.4855 \quad -0.4865 \quad 0.5209]^{\intercal}$ respectively. The view points (initial approach and best view point) were chosen to be at $d = 0.3m$ distance from $\mathbf{p}_{sb}$, since at that distance the camera depth measurements were more reliable. The control parameters were chosen as $k_c = k_l = 1.8, k_\rho = 0.8, k_\theta = k_\phi = 0.6$ while the control cycle was $T_s = 2ms$. For an adequate sphere sampling, the sphere was sampled at $n_l = 500$ points. Grape and stem detection methods are reported in [16] and [17] respectively and have been used in an integrated solution in the field with high success rate [21] where also the split of the scene around the grape to front and back has been successfully validated. In this work, we use a red colored tube wrapped around the stem for stem detection by color and an AprilTag for the initial grape

position detection in order to focus on the efficacy of the proposed solution.

(a) Initial setup



(b) Initial camera view

(c) View after initial approach

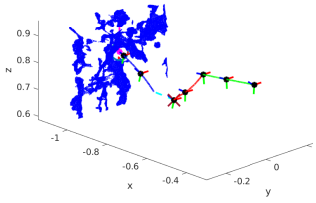
(d) View from best view point

(e) Final robot configuration

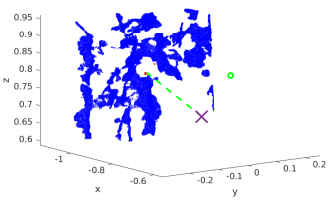
(f) Final cutter position

Fig. 4: Example of an experimental trial run

In Fig.5 the camera and the cutter paths, as well as the observed point clouds of the stem and the occlusions of a single experimental run are shown. The camera at its initial position is far from the stem and is unable to reliably detect it (Fig.4b). Thus, at the beginning the camera from its initial position and orientation approaches by a distance of $d = 0.3m$, a point above the detected grape by $8cm$ in order to detect the stem (green path in Fig.5a). Once the stem is partially detected (Fig.4c), the best feasible view point is computed and is reached by the camera while also centering the stem (red path in Fig.5a). Notice how this path is curved in the Cartesian space owning to the use of spherical coordinates. A second detection of the now



(a) Camera and cutter's paths

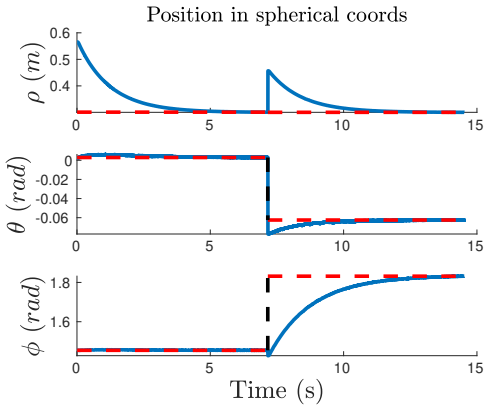


(b) Point clouds of the viewed scene and computed best view point

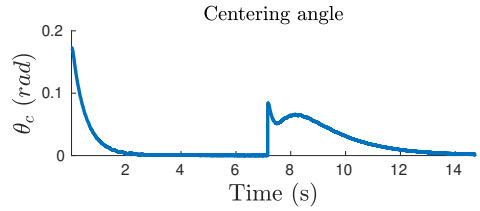
Fig. 5: Path and point clouds from a single experimental run

unveiled stem is performed in order to more reliably compute its position $\mathbf{p}_{sb}$ and its axis $\mathbf{s}$ (Fig.4d), and then the cutter proceeds to the target cutting pose to cut it (blue path in Fig.5a). The final configuration of the robot and position of the cutter are shown in Fig.4e and Fig.4f respectively.

In Fig.5b the point cloud of the viewed scene (blue: occlusions, red: stem), after the initial camera approach, is shown. The camera position is indicated by the green circle and the computed best view point by the purple cross. The dashed green line shows the view direction of the camera at the best view point, which offers an unobstructed view of the stem.

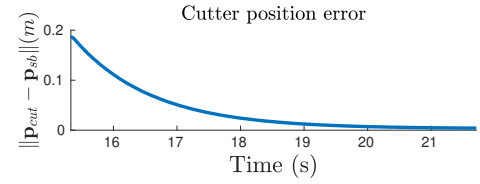


(a)

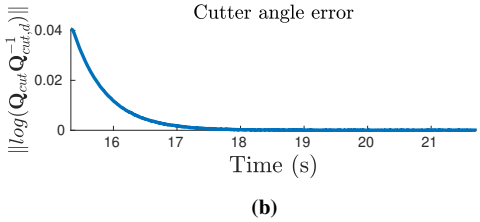


(b)

Fig. 6: (a) Spherical coords position of camera w.r.t $\mathbf{p}_{sb}$, (b) centering angle



(a)



(b)

Fig. 7: (a) Cutting position error, (b) orientation error.

Figure 6 shows the spherical coordinates of the camera frame for the initial approaching motion (from $0s$ to $7s$) and the motion towards the best view point (from $7s$ to $15s$ (Fig.6a), as well as the centering angle of the camera view

axis (Fig.6b). Notice that the angular coefficients of the initial motion remain constant since the camera moves in a straight path for that part of the overall motion. Further notice the discontinuity at $7s$ which is due to the calculation of the stem position $\mathbf{p}_{sb}$ from the point cloud. It can be seen that both tasks achieve their objectives, as the initial and the best view point (dashed red lines) are reached with the camera view centered at the stem. Moreover, Fig.7 shows the distance of the cutter from $\mathbf{p}_{sb}$ during the cutting motion and the angle error norm between the frame of the cutter and the target cutting pose, which are clearly reaching zero at $21.5s$ completing the cutting objective.

For the experiment, 22 different cases have been generated, one for each trial, by placing the leaves around the grape in different positions and with varying levels of stem occlusion. In 19 of the trials a suitable view pose was determined and the cutting pose was successfully reached resulting in a success rate of 86.36%. The failures were due to either incorrect stem detection, or heavy occlusions leading to limited small free space areas that did not allow for proper approaching towards the stem. In 12 of the trials several candidate view points with better views, i.e. farther from occlusions, surpassed the singularity threshold and were rejected, demonstrating the necessity of utilizing such a metric. Additionally, in all successful cases a significant increase of the number of the visible stem points has been observed at the optimized view point compared to the initial stem view with a mean increase of 191.14 points. Furthermore, between the initial and the best view, the stem mean position difference is $0.9cm$, with a maximum of over $3cm$, while the mean angle difference between the two axes is 30^o, suggesting the necessity of viewing the stem from a good view point. Finally, the average total time of the trials was $T_{total} = 21.25s$ with standard deviation $0.524s$ demonstrating a near constant time for each grape harvest, owing to the deterministic approach of finding the best view point.

IV. Conclusions

This work addresses the problem of viewing, reaching and cropping a grape stem under occlusions and kinematic constraints using a robot manipulator equipped with an in-hand camera. The proposed solution uses the point cloud from a single view to compute the best view point towards the stem, and a second view from the best view point to more reliably determine the cutting pose that requires minimal motion of the cutter. The effectiveness of the proposed approach is demonstrated in multiple in-lab experiments with varying cases of occlusions, where a success rate of 86.36% on properly viewing and reaching the stem was achieved. Future plans involve the integration of this work to an integrated mobile manipulator system and validate its performance with field tests in an actual vineyard.

References

[1] H. Zhou, X. Wang, W. Au, H. Kang, and C. Chen, "Intelligent robots for fruit harvesting: recent developments and future challenges," *Precision Agriculture*, vol. 23, pp. 1856–1907, Oct 2022.

[2] L. Droukas, Z. Doulgeri, N. L. Tsakiridis, D. Triantafyllou, I. Kleitsiotis, I. Mariolis, D. Giakoumis, D. Tzovaras, D. Kateris, and D. Bochtis, "A survey of robotic harvesting systems and enabling technologies.," *J Intell Robot Syst*, vol. 107, Jan. 2023.

[3] E. Vrochidou, K. Tziridis, A. Nikolaou, T. Kalampokas, G. A. Papakostas, T. P. Pachidis, S. Mamalis, S. Koundouras, and V. G. Kaburlasos, "An autonomous grape-harvester robot: Integrated system architecture," *Electronics*, vol. 10, no. 9, 2021.

[4] Y. Jiang, J. Liu, J. Wang, W. Li, Y. Peng, and H. Shan, "Development of a dual-arm rapid grape-harvesting robot for horizontal trellis cultivation," *Frontiers in Plant Science*, vol. 13, 2022.

[5] A. Gongal, S. Amatya, M. Karkee, Q. Zhang, and K. Lewis, "Sensors and systems for fruit detection and localization: A review," *Computers and Electronics in Agriculture*, vol. 116, pp. 8–19, 2015.

[6] C. Lehnert, D. Tsai, A. Eriksson, and C. McCool, "3d move to see: Multi-perspective visual servoing for improving object views with semantic segmentation," 2018.

[7] P. Zapotezny-Anderson and C. Lehnert, "Towards active robotic vision in agriculture: A deep learning approach to visual servoing in occluded and unstructured protected cropping environments," *IFAC-PapersOnLine*, vol. 52, no. 30, pp. 120–125, 2019. 6th IFAC Conference on Sensing, Control and Automation Technologies for Agriculture AGRICONTROL 2019.

[8] D. Papageorgiou, L. Koutras, and Z. Doulgeri, "A controller for reaching and unveiling a partially occluded object of interest with an eye-in-hand robot," in *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*, pp. 254–260, 2022.

[9] B. Penin, P. R. Giordano, and F. Chaumette, "Vision-based reactive planning for aggressive target tracking while avoiding collisions and occlusions," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3725–3732, 2018.

[10] I. Cuiral-Zueco and G. López-Nicolás, "Dynamic occlusion handling for real time object perception," in *2020 5th International Conference on Robotics and Automation Engineering (ICRAE)*, pp. 13–18, 2020.

[11] T. Yi, D. Zhang, L. Luo, and J. Luo, "View planning for grape harvesting based on active vision strategy under occlusion," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2535–2542, 2024.

[12] T. Zaenker, C. Smitt, C. McCool, and M. Bennewitz, "Viewpoint planning for fruit size and position estimation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3271–3277, 2021.

[13] R. Border and J. D. Gammell, "Proactive estimation of occlusions and scene coverage for planning next best views in an unstructured representation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4219–4226, 2020.

[14] D. Morrison, P. Corke, and J. Leitner, "Multi-view picking: Next-best-view reaching for improved grasping in clutter," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8762–8768, 2019.

[15] A. Sidiropoulos and Z. Doulgeri, "From rgb images to dynamic movement primitives for planar tasks," in *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, pp. 1–8, 2023.

[16] I. Kleitsiotis, I. Mariolis, D. Giakoumis, S. Likothanassis, and D. Tzovaras, "Anisotropic diffusion-based enhancement of scene segmentation with instance labels," in *Computer Analysis of Images and Patterns*, pp. 383–391, Springer International Publishing, 2021.

[17] G. Zampokas, I. Mariolis, D. Giakoumis, and D. Tzovaras, "Residual cascade cnn for detection of spatially relevant objects in agriculture: The grape-stem paradigm," in *Computer Vision Systems*, pp. 159–168, Springer Nature Switzerland, 2023.

[18] E. Vrochidou, V. N. Tsakalidou, I. Kalathas, T. Gkrimpizis, T. Pachidis, and V. G. Kaburlasos, "An overview of end effectors in agricultural robotic harvesting systems," *Agriculture*, vol. 12, no. 8, 2022.

[19] B. Keinert, M. Innmann, M. Sänger, and M. Stamminger, "Spherical fibonacci mapping," *ACM Transactions on Graphics*, vol. 34, no. 6, pp. 1–7, 2015.

[20] N. Vahrenkamp, T. Asfour, G. Metta, G. Sandini, and R. Dillmann, "Manipulability analysis," in *2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012)*, pp. 568–573, 2012.

[21] S. Stavridis, L. Droukas, and Z. Doulgeri, "Bimanual grape manipulation for human-inspired robotic harvesting," *Accepted in IEEE/ASME Transactions on Mechatronics*, 2024.