A controller for reaching and unveiling a partially occluded object of interest with an eye-in-hand robot

Dimitrios Papageorgiou, Leonidas Koutras and Zoe Doulgeri

***Abstract*— In this work, a control scheme for approaching and unveiling a partially occluded object of interest is proposed. The control scheme is based only on the classified point cloud obtained by the in-hand camera attached to the robot's end effector. It is shown that the proposed controller reaches in the vicinity of the object progressively unveiling the neighborhood of each visible point of the object of interest. It can therefore potentially achieve the complete unveiling of the object. The proposed control scheme is evaluated through simulations and experiments with a UR5e robot with an in-hand RealSense camera on a mock-up vine setup for unveiling the stem of a grape cluster.**

I. Introduction

As opposed to the structured industrial environment, farming fields are unstructured and the objects of interest (OOI), e.g. fruits, stalks, or stems, are in many cases either partially or completely occluded, when the robot attempts to acquire information related to them. Furthermore, in most cases, the fruit cannot be concretely modelled, as it involves structural uncertainties and its location is not known a priori [1]. Therefore, the challenge of visually unveiling the OOI, by tackling the problem of avoiding possible occlusions, e.g. leaves or branches, is widely reported and is considered as a non-trivial issue in agricultural robotics [1]–[3].

In this work we consider the problem of approaching and unveiling a partially occluded object of interest by an eye-in-hand robot to enable a subsequent task of grasping or cutting. We assume a perception system that can on-line classify the obtained points of the cloud to those belonging to the object of interest and the rest of the objects that potentially occlude it and develop a novel method to achieve increasing visibility of the partially occluded object by moving the manipulator. This method is developed to be integrated in the mobile bi-manual robot depicted in Fig. 1, performing grape harvesting. The method can be generalized to multiple applications involving humanoid robots attempting to reach an occluded OOI, while operating in cluttered, unstructured environments. In this paper, the initial lab tests with a static manipulator are presented.

II. Related literature and contributions

The methods for visually unveiling the OOI, by addressing the problem of occlusions considering a robot with an in-hand camera, can be divided in three main categories: RGB image based optimization approaches, 3D scene geometry based approaches and machine learning based approaches.

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, Dept. of Electrical & Computer Engineering, Aristotle University of Thessaloniki, Greece {dimpapag, kleonidas, doulgeri}@ece.auth.gr

Fig. 1: Mobile bi-manual harvesting robot

A. RGB image-based optimization approaches

According to the optimization-based approaches, which exploit only RGB information, an optimization function is defined, and the motion of the robot aims at maximizing (or minimizing) this function. In the context of visual unveiling of the OOI, one commonly selected optimization function is the visible area of the object of interest, i.e. the number of pixels of the OOI within the image captured by the camera. More specifically, in [2], an agricultural harvesting scenario is considered and the optimization function involves the pixels of the visible OOI, as well as a manipulability measure. However, for finding the gradient of the optimization function, which defines the direction towards the function's maximization, 9 cameras placed in a 3x3 grid-fixture were utilized and attached to the robot's end-effector. On the other hand, methods relying on a single camera for maximizing an optimization function involve either a predefined scanning path to obtain multiple views [4], which may be time and energy consuming, or probing strategies for exploration [5], which yield suboptimal solutions.

B. 3D scene geometry-based approaches

Methods that utilize information about the 3D scene geometry are the most popular solutions amongst the robotics literature. According to such approaches, the control signal which actuates the in-hand camera is based on the information about the 3D geometry of the environment, acquired via appropriate sensors, e.g. RGB-D cameras or LiDARs.

Most of the methods of this category assume a known and modelled environment and/or involve a known target (i.e. an object with known and predefined geometry/structure) [6]–[12]. More specifically, in [6], an autonomous drone equipped with a camera is considered, with the obstacles being modelled by spheres and the target (i.e. the OOI) being a point-feature having a known position. In [7], the authors propose a control scheme for self-occlusion avoidance during visual servoing, considering a complete knowledge of the object's model. In [8], a trajectory planning algorithm for reaching a specific target is presented, which assumes that the complete occlusion-free space is known and/or is computable a priori. Similarly, in [9], the complete geometry of the obstacles (clutter) within the environment is assumed to be known, e.g. by the CAD in an industrial task, based on which the occlusion-free space is analytically calculated, considering a sphere-like visual target with a known position. Following this line of thought, the approaches of [10] and [11] also assume that the 3D model of the target object and the obstacles in the workspace are known a priori. Lastly, in [12], although the obstacles are not modelled, the normal vectors at each point of the surface of the OOI are assumed to be known or computable, which implies that the OOI surface is either pre-modelled or can be modelled on-line.

In [13], a method which does not involve the modelling of the environment or the target is proposed. However, it is applicable only in the case of a 2D navigation of a platform, as a number of virtual 1-dof curves are defined around the robotic platform, having a geometry which cannot be directly extended to the 3D space.

C. Machine learning based approaches

A machine learning approach is proposed in [1], built on top of [2]. In particular, in [1] as opposed to [2], a convolutional neural network (CNN) is utilized, to estimate the gradient of the optimization function, making the method also applicable by utilizing only a single camera setup instead of 9 cameras required in [2]. However, the training of the CNN requires a large and representative training dataset. Although this dataset can be gathered via a simulated environment, transferring the trained controller to the field is still a challenging task. In [14], an eye-in-hand grasping task is considered and a CNN is utilized for calculating a grasping feasibility map. However, to find the "next best view", the method requires the exploration of the task space (e.g. to pass through various viewpoints), and the calculation of the grasping map at each view point of the task space, making this method time and energy consuming.

Finally, a work which cannot be easily classified in one of the aforementioned categories is [15], in which a wavelet transformation is utilized in order to perform a visual servoing task by avoiding occlusions. However, this method requires the knowledge of the exact pattern or model of the OOI, to assess the similarity via the wavelet transformation. Unfortunately, this information is usually not available in robotically assistive farming applications.

D. Main contributions

As opposed to the RGB image-based optimization approaches and the machine learning based approaches, the proposed method does not involve task space exploration motions and does not require the availability of multiple in-hand cameras. Furthermore, as opposed to the geometry-based approaches, the proposed method does not require the models of the OOI or the surrounding obstacles but their related point clouds. To the best of the authors knowledge there are no previous methods for unveiling an OOI that do not involve exploration motions, do not require the models of the OOI and the environment or they utilize only one RGB-D in-hand camera.

III. Problem description

Consider the availability of an N-dof robotic manipulator with an RGB-D camera attached to its end-effector. Let $\mathbf{q} \in \mathbb{R}^N$ be the vector of the joint variables of the manipulator, with N being the number of joints. In the case of a mobile manipulator the degrees of freedom include those of the mobile platform (3 dof). In the rest of the paper, most variables are expressed in the camera frame. Expressions in other frames (e.g the world frame) will be denoted by a left superscript.

Let ${}^0\mathbf{T}_c \in SE(3)$ be the homogeneous transformation expressing the generalized pose of the camera frame $\{\mathrm{C}\}$ in the world inertial frame $\{0\}$, involving its position ${}^0\mathbf{p}_c(\mathbf{q}) \in \mathbb{R}^3$ and its orientation ${}^0\mathbf{R}_c(\mathbf{q}) \triangleq [\mathbf{x}_c\ \mathbf{y}_c\ \mathbf{z}_c] \in SO(3)$. Let us further denote by $\mathbf{v}_c$ the generalized body velocity of the camera's frame consisted of the translational velocity ${}^0\mathbf{R}_c^T {}^0\dot{\mathbf{p}}_c$ and the rotational velocity $\mathbf{S}(\boldsymbol{\omega}_c) = {}^0\mathbf{R}_c^T {}^0\dot{\mathbf{R}}_c$, with $\mathbf{S}(.)$ denoting the skew-symmetric matrix. We assume that the system is aware of the center of the region of interest (ROI) that is defined as a sphere and includes at least one object of interest (OOI), that should not necessarily be visible by the robot when it is away from the ROI. For example, a grape cluster in the center of ROI has an associated stem that may not be visible at start.

Let $\mathbf{p}_r \in \mathbb{R}^3$ be the center of the ROI and $r \in \mathbb{R}_{>0}$ be its radius. We assume that given the point-cloud perceived by the RGB-D camera, the perception algorithm can on-line classify the points of the cloud in two categories: a) those belonging to the OOI and b) the rest of the point-cloud which may belong to surfaces that occlude the OOI. For instance, in the grape harvesting case, the OOI is the stalk of the grape cluster, while the rest of the point cloud may be points of the leaves, branches, supporting wires or even the rest of the fruit. Let $\mathbf{p}_i \in \mathbb{R}^3,\ i = 1, ..., n$ be the currently visible points of the OOI and $\mathbf{p}_{o,j} \in \mathbb{R}^3,\ j = 1, ..., m$ be the rest of the points which belong to surfaces that may occlude the observation of the OOI.

We assume a robot that provides a position or velocity control interface which is true for most of the commercially available collaborative robots. In the kinematic level, the controlled system can be expressed in the joint space as $\dot{\mathbf{q}} = \mathbf{J}^{\dagger}(\mathbf{q})\mathbf{u}_c$, where $\mathbf{J}^{\dagger} \in \mathbb{R}^{N\times 6}$ is a pseudo inverse of the end-effector Jacobian matrix $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{6\times N}$ and $\mathbf{u}_c \in \mathbb{R}^6$

is the control input that takes the form of the body reference velocity of the camera's frame and which should be designed in order to achieve the following objectives:

1) The end-effector has to smoothly reach the region of interest (ROI), which can be mathematically expressed as: $\lim_{t\to\infty}\|\mathbf{p}_r(t)\|\to \overline{r} \leq r$, for some $\overline{r} > 0$.
2) To maintain the center of ROI at the center of the field of view, as much as possible, which can be mathematically expressed as: $\theta(\mathbf{p}_r) \to 0$, for $t \to \infty$, where $\theta(\mathbf{p}_r) \triangleq \cos^{-1}\left(\frac{\mathbf{z}^\intercal \mathbf{p}_r}{\|\mathbf{p}_r\|}\right)$ is the angle between $\mathbf{p}_r$ and the camera's view direction which we assume without loss of generality that coincides with the z-axis of the camera's frame $\mathbf{z} = [0\ 0\ 1]^\intercal$. Notice that θ cannot be more than $\frac{\pi}{2}$, since the center of ROI $\mathbf{p}_r$ is within the field of view of the camera.
3) To maximize the visible area of the object of interest, i.e. to maximize the perceived number of points of the point-cloud of OOI.

Objective 1 aims at moving the end-effector to any of the points of the ROI while objective 2 aims at aligning the z-axis of the camera with the vector pointing to the center of ROI. Clearly these two objectives can be achieved by reaching any point in ROI with specified two orientation degrees of freedom (DOF) of the camera frame. Given the task redundancy the third objective is set to select the point in ROI that visually unveils the OOI as much as possible.

IV. Proposed control scheme

The controller superimposes two control signals $\mathbf{v}_{c1}$, $\mathbf{v}_{c2}$ as follows:

$$\mathbf{u}_c = \mathbf{v}_{c1} + \mathbf{v}_{c2}, \tag{1}$$

in the form of end-effector body velocities which are:

- $\mathbf{v}_{c1} \in \mathbb{R}^6$: ROI reaching & centering control signal, which is designed to fulfill the first two control objectives (Objective 1 and Objective 2).
- $\mathbf{v}_{c2} \in \mathbb{R}^6$: OOI active perception control signal, which is designed to fulfill the last objective (Objective 3), which is related to the maximization of the visible area of the OOI.

Each signal, acting alone, aims at achieving the respective objective. The proposed control scheme is designed so that $\mathbf{v}_{c2} = \mathbf{0}$ when no points of the OOI are visible, i.e. $n = 0$. Then the ROI reaching & centering controller is acting alone. Such a case may occur during the start of the motion when the camera is far from the ROI and the OOI starts being visible only when it approaches or enters the ROI.

A. ROI reaching & centering control signal

Notice that control objectives 1 and 2 can be achieved by appropriately translating and rotating the end-effector camera. Mathematically, the first objective is fulfilled when the camera position ${}^0\mathbf{p}_c$ converges to the manifold ${}^0\Omega$:

$${}^0\Omega \triangleq \{{}^0\mathbf{p}_c \in \mathbb{R}^3 : f({}^0\mathbf{p}_c - {}^0\mathbf{p}_r) \leq 0\}, \tag{2}$$

with $f(\mathbf{x}) \triangleq \mathbf{x}^\intercal\mathbf{x} - r^2$ for any $\mathbf{x} \in \mathbb{R}^3$. Since ${}^0\mathbf{p}_r = {}^0\mathbf{R}_c\mathbf{p}_r + {}^0\mathbf{p}_c$ the manifold (2) condition can be expressed in the camera frame C as $f(\mathbf{p}_r) \leq 0$.

Objective 2 regards the orientation of the camera. It is achieved when the camera's z-axis points to the center of the ROI.

To achieve these objectives, we propose the control signal:

$$\mathbf{v}_{c1} \triangleq \begin{bmatrix} k_p \max(0,\ f(\mathbf{p}_r))\mathbf{p}_r \\ k_o\theta\mathbf{k} \end{bmatrix}, \tag{3}$$

where for the translational part we use the approach of [16] and $k_p, k_o \in \mathbb{R}_{>0}$ are constant positive gains for translation and orientation respectively and $\theta \in [0, \frac{\pi}{2}]$ and $\mathbf{k}$ are the angle and axis of the minimum rotation between $\mathbf{p}_r$ and $\mathbf{z} \triangleq [0\ 0\ 1]^\intercal$, which can be calculated by the following expressions:

$$\theta = \cos^{-1}\left(\frac{\mathbf{z}^\intercal\mathbf{p}_r}{\|\mathbf{p}_r\|}\right),\ \mathbf{k} = \frac{\mathbf{S}(\mathbf{z})\mathbf{p}_r}{\|\mathbf{S}(\mathbf{z})\mathbf{p}_r\|}, \tag{4}$$

where $\mathbf{S}(\mathbf{z})$ is the skew symmetric matrix of the $\mathbf{z}$.

Notice that $\mathbf{v}_{c1}$ is continuous and its value is zero if and only if $f \leq 0$ and $\theta = 0$.

Proposition 1. *Setting $\mathbf{u}_c = \mathbf{v}_{c1}$ achieves asymptotic convergence to a point that fulfills Objectives 1 and 2.*

The proof can be found in the Appendix.

B. Visually unveiling OOI active perception control

Assume the simple case depicted in Figure 2. Let $\mathbf{p}_i$ be a visible point of the object of interest (OOI). Let $\mathbf{p}_o$ be the center of a spherical obstacle that can potentially occlude the OOI for a camera pose. The ray from the camera $\mathbf{p}_c$ to $\mathbf{p}_i$ shown Figure 2 is in the camera's field of view since $\mathbf{p}_i$ is visible. The vector normal to this ray from $\mathbf{p}_o$ intersects the ray at point $\hat{\mathbf{p}}$. Let $\hat{r} \in \mathbb{R}_{\geq 0}$ be the distance along the normal from the obstacle surface to $\hat{\mathbf{p}}$. It is clear that the maximum visible spherical region centered at $\mathbf{p}_i$ has a radius r_v which is always greater or equal to $\hat{r}$. Comparing instances A and B of Figure 2, notice that if the distance of the camera from $\mathbf{p}_i$ remains constant or less, the greater the value of $\hat{r}$, the greater the volume of the neighborhood of $\mathbf{p}_i$ that is unveiled.

Based on this observation, we design a control signal for the camera's body velocity $\mathbf{v}_{c2}$ which can be seen as a rotation of the camera around $\mathbf{p}_i$ in order to guarantee that $\hat{r} > d_c$, and that its derivative is positive, i.e., that $\hat{r}$ will increase so that the unveiled region around $\mathbf{p}_i$ will increase.

Consider the general case of $\mathbf{p}_i \in \mathbb{R}^3, i = 1, ..., n$ currently visible points and $\mathbf{p}_{o,j} \in \mathbb{R}^3, j = 1, ..., m$ obstacles. Given the aforementioned idea we draw our inspiration from our previous work [17] where virtual constraints are imposed on the surface of a forbidden region given as a point cloud via repulsive artificial potentials for surgical applications. We propose the utilization of a barrier artificial potential field around each obstacle, which induces a virtual repulsive velocity $\mathbf{u}_{i,j}$, acting at $\hat{\mathbf{p}}_{i,j}$, which is the nearest point of the

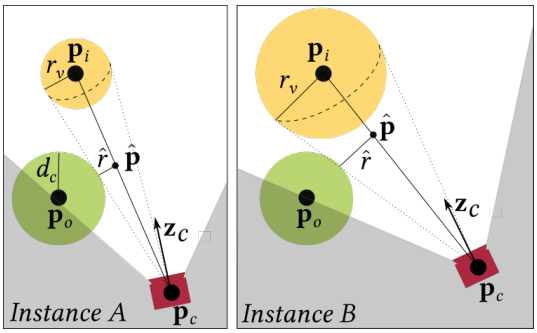


Fig. 2: Basic concept involving a single point of the OOI and a single point as an obstacle.

i-th ray from the j-th obstacle and is calculated by:

$$\hat{\mathbf{p}}_{i,j} \triangleq \begin{cases} \mathbf{0}, & \text{if } \frac{\mathbf{p}_i^\intercal}{\|\mathbf{p}_i\|}\mathbf{p}_{o,j} \leq 0 \\ \frac{\mathbf{p}_i\mathbf{p}_i^\intercal}{\mathbf{p}_i^\intercal\mathbf{p}_i}\mathbf{p}_{o,j}, & \text{if } \frac{\mathbf{p}_i^\intercal}{\|\mathbf{p}_i\|}\mathbf{p}_{o,j} \in (0, \|\mathbf{p}_i\|) \\ \mathbf{p}_i, & \text{if } \frac{\mathbf{p}_i^\intercal}{\|\mathbf{p}_i\|}\mathbf{p}_{o,j} \geq \|\mathbf{p}_i\| \end{cases}. \quad (5)$$

The proposed barrier artificial potential function, depicted in Figure 3, is designed to induce a repulsive velocity only within a predefined distance $d_0 \in \mathbb{R}_{>0}$ from the obstacle surface; it is defined as follows:

$$V(\hat{r}_{i,j}) \triangleq \begin{cases} \frac{1}{2}\ln^2\left(\frac{d_0^2}{d_0^2-(d_0-\hat{r}_{i,j})^2}\right), & \text{if } \hat{r}_{i,j} < d_0 \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where

$$\hat{r}_{i,j} \triangleq \|\hat{\mathbf{p}}_{i,j} - \mathbf{p}_{o,j}\| - d_c$$

with $d_c \in \mathbb{R}_{>0}$ the obstacle's spherical radius that is selected for minimizing the coverage of the empty space between neighboring points of the obstacle cloud. Notice that the value of the artificial potential tends to infinity when the surface of the obstacle is reached from outside in order to guarantee obstacle/occlusion avoidance.

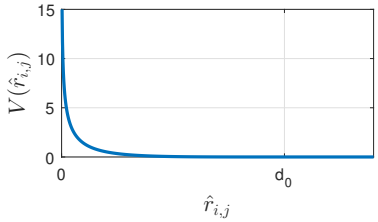


Fig. 3: Artificial potential function for $\mathbf{v}_{v2}$.

The virtual commanded repulsive velocity is then given by

$$\mathbf{u}_{i,j} \triangleq -\left(\frac{\partial V(\hat{\mathbf{p}}_{i,j})}{\partial \hat{\mathbf{p}}_{i,j}}\right)^\intercal = \begin{cases} \frac{2}{d_0^2-(d_0-\hat{r}_{i,j})^2}\ln\left(\frac{d_0^2}{d_0^2-(d_0-\hat{r}_{i,j})^2}\right)\mathbf{e}_{i,j}, & \text{if } \hat{r}_{i,j} < d_0 \\ \mathbf{0}_3, & \text{otherwise} \end{cases} \quad (7)$$

where $\mathbf{e}_{i,j} \in \mathbb{R}^3$ is given by:

$$\mathbf{e}_{i,j} \triangleq (d_o - \hat{r}_{i,j})\frac{\hat{\mathbf{p}}_{i,j} - \mathbf{p}_{o,j}}{\|\hat{\mathbf{p}}_{i,j} - \mathbf{p}_{o,j}\|}. \quad (8)$$

which is a vector at the direction of the normal to the ray with magnitude proportional to the distance along the normal from the obstacle's surface.

The virtual repulsive velocity (7) possesses the following properties:

- $\|\mathbf{u}_{i,j}\| \neq 0$, if and only if $0 < \hat{r}_{i,j} < d_0$, and $\mathbf{u}_{i,j} \neq \mathbf{0}$ if and only if $\hat{r}_{i,j} \geq d_0$, which means that the signal will not be affected by rays that are not within the range of influence of the j-th obstacle point, defined by d_0.
- $\|\mathbf{u}_{i,j}\| \to \infty$, when $\hat{r}_{i,j} \to 0$, i.e. when the i-th ray approaches the surface of the j-th obstacle. Notice that $\|\hat{\mathbf{p}}_{i,j} - \mathbf{p}_{o,j}\|$ cannot be less than d_c, as it reflects the accuracy of the RGB-D camera, by definition.
- $\mathbf{u}_{i,j}$ is continuous with respect to $\hat{\mathbf{p}}_{i,j}$.
- However, $\mathbf{u}_{i,j}$ is, in general, not continuous in time, as $\mathbf{p}_i, \mathbf{p}_{o,j}$ and even n, m depend on the point-cloud perceived by the RGB-D camera during its motion; as more points of the OOI, previously occluded, enter into the field of view, it is possible that some of them induce a non-zero control signal. This discontinuity can be remedied by a first order low-pass filter.

To synthesize the total proposed control signal $\mathbf{v}_{c2}$, we calculate the angular velocity $\boldsymbol{\omega}_{i,j}$, which is induced by the virtual repulsive velocity $\mathbf{u}_{i,j}$ around an axis passing from $\mathbf{p}_i$ and defined by the cross product of the directions of $\mathbf{u}$ and $\mathbf{p}_i$. This angular velocity is given by:

$$\boldsymbol{\omega}_{i,j} \triangleq \begin{cases} \frac{\mathbf{S}(\hat{\mathbf{p}}_{i,j}-\mathbf{p}_i)}{\|\hat{\mathbf{p}}_{i,j}-\mathbf{p}_i\|^2}\mathbf{u}_{i,j}, & \text{if } \frac{\mathbf{p}_i^\intercal}{\|\mathbf{p}_i\|}\mathbf{p}_{o,j} \in (0, \|\mathbf{p}_i\|) \\ \mathbf{0}_3, & \text{otherwise} \end{cases}. \quad (9)$$

By summing the $\boldsymbol{\omega}_{i,j}$-s acting on the i-th pivot point, we get the total angular velocity for the i-th OOI point, which is given by:

$$\boldsymbol{\omega}_i \triangleq \sum_{j=1}^{m} \boldsymbol{\omega}_{i,j} \in \mathbb{R}^3. \quad (10)$$

Lastly, to synthesize the total control signal $\mathbf{v}_{c2}$, the $\boldsymbol{\omega}_i$-s are superimposed after calculating the corresponding linear velocity at the end-effector. This superimposition is given by:

$$\mathbf{v}_{c2} \triangleq k\sum_{i=1}^{n}\begin{bmatrix}\mathbf{S}(\mathbf{p}_i) \\ \mathbf{I}_3\end{bmatrix}\boldsymbol{\omega}_i \in \mathbb{R}^6, \quad (11)$$

where $k \in \mathbb{R}_{>0}$ is a positive tunable gain.

Taking the time derivative of the artificial potential (6), we find that: $\dot{V} = -\frac{\|\mathbf{p}_i\|-\|\hat{\mathbf{p}}_{i,j}\|}{\|\mathbf{p}_i-\hat{\mathbf{p}}_{i,j}\|}\mathbf{u}_{i,j}^\intercal\mathbf{u}_{i,j}$, which is less or equal than 0, given that $\|\hat{\mathbf{p}}_{i,j}\| \leq \|\mathbf{p}_i\|$, which is true by construction. This means that V is bounded and that $\hat{r}_{i,j}$ is increasing within the area of influence, due to the fact that $V(\hat{r}_{i,j})$ is a decreasing function of $\hat{r}_{i,j}$ (see Fig. 3). Therefore, $\mathbf{p}_i$ will not be occluded by the obstacle centered at $\mathbf{p}_{o,j}$. Furthermore, given that the linear velocity of $\mathbf{v}_{c2}$ is orthogonal to $\mathbf{p}_i$, which implies that $\|\mathbf{p}_i\|$ remains constant, the maximum radius of visibility around $\mathbf{p}_i$ is increased with the proposed control signal. As a result, the progressive unveiling of more points of the OOI occurs in a chain-reaction manner, i.e. by unveiling progressively more and more OOI points.

V. Performance analysis through simulations

To assess the performance of the proposed controller, which involves the superposition of the two body camera/end-effector velocities (1), we tested it for 100 randomly 3D generated scenes with $m = 5$ spherical obstacles that may occlude the OOI. We considered a linear segment as being the OOI. In particular, the procedure of the random scene generation is the following:

- The five obstacle points are randomly placed within a cube with an edge of 0.5m. In particular, this procedure can be described by: ${}^0\mathbf{p}_{o,j} = \mathbf{U}_3^\intercal(-0.5, 0.5) + [1.8 \; -0.7 \; 0.1]^\intercal$ where $\mathbf{U}_3(-0.5, 0.5) \in \mathbb{R}^3$ symbolizes the 3-dimensional random uniform process between -0.5 and 0.5.
- The camera at the robot's end-effector starts from the same initial pose, which is ${}^0\mathbf{p}_c = \mathbf{0}_3$ and ${}^0\mathbf{R}_c = [0\ 0\ 1; 1\ 0\ 0; 0\ 1\ 0]$.
- The linear segment which corresponds to the OOI is defined in a parametric form as: ${}^0\mathbf{p}_L(\sigma) = {}^0\mathbf{p}_0 + \sigma({}^0\mathbf{p}_1 - {}^0\mathbf{p}_0)$, where ${}^0\mathbf{p}_0 = [3 \;\; -0.1 \;\; 0.6]^\intercal$ and ${}^0\mathbf{p}_1 = [3\ 0.1\ 1.4]^\intercal$ and $\sigma \in [0,\ 1]$.
- The center of the ROI is $\mathbf{p}_r = 0.5(\mathbf{p}_0 + \mathbf{p}_1)$ and the radious is set to $r = 3.262$, which means that the camera is initially within the ROI.

A representative generated scene and the solution achieved by the proposed method are depicted in Figure 4. The histogram of the initial visible percentage of visible OOI at the beginning of the simulations is depicted in Figure 5a. Notice that the mean value of the initially visible portion of the OOI is 48.09% for the 100 randomly generated scenes.

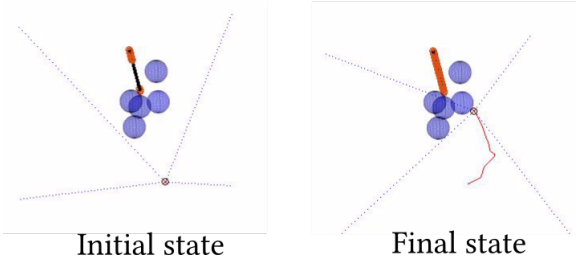
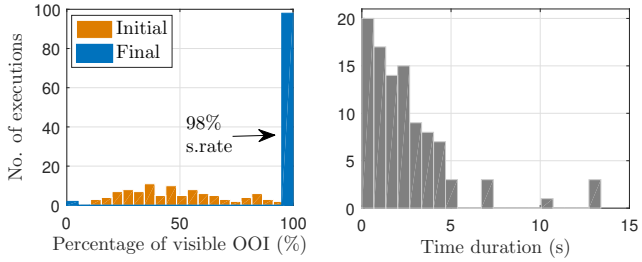


Fig. 4: Automatically generated scene. The OOI is denoted with black line and its visible part with red line, while the blue spheres represent the possible obstacles.

The parameters used in the simulations are $d_c = 0.1$, $d_0 = 0.3$, $k = 0.04$, $k_p = 5, k_o = 200$, while the control signal is filtered via a first order low pass filter having a pole at -10.

The histogram of the visible percentage of the OOI at the end of the motion under the proposed control scheme is depicted in Figure 5a. Considering as a success only the case where the whole OOI is visually unveiled, the success rate of the proposed controller reached 98%. The histogram of the duration required for the motion is depicted in Figure 5b for all 100 simulation runs. It is clearly visible that the duration of motion is bounded, i.e. the system reaches an equilibrium and the mean duration of motion is 2.53s, with the selected values of the control parameters.



(a) Success rate **(b)** Duration

Fig. 5: Histograms of the evaluation through 100 simulated scenes.

Through the simulation results it is evident that the control signal for the ROI reaching, i.e. $\mathbf{v}_{c1}$, does not reduce the efficiency of the controller which is responsible for the visual unveiling of the OOI, i.e. $\mathbf{v}_{c2}$, while it is worth noting that the camera remained in the vicinity of the OOI and within the ROI during its motion, due to $\mathbf{v}_{c1}$.

VI. Experimental evaluation

For the experimental evaluation of the proposed method, a lab setup was built with a mock-up vine, involving a plastic grape cluster and leaves, as illustrated in Figure 6. For emulating realistic field conditions, a fan was utilized for artificial wind. We consider a pre-cutting scenario, in which the arm with the cutting tool is equipped with an in-hand camera at its end-effector. Our aim is to approach the region of interest (ROI), which is defined as a sphere with radius $r = 0.4$m around the center of the grape cluster and visually unveil the stalk of the grape, which is considered to be the object of interest (OOI). In order to distinguish the points of the point-cloud which belong to the stalk (OOI), a red plastic tube was utilized, as the identification of the stalk is out of the scope of this work. The points not belonging in this red tube are considered to be obstacles. To make the scenario more realistic, we further assume that the OOI is detectable only within a range of 0.5m by the perception algorithm and the experiment starts outside of this range, i.e. without any points of the OOI being initially detected.

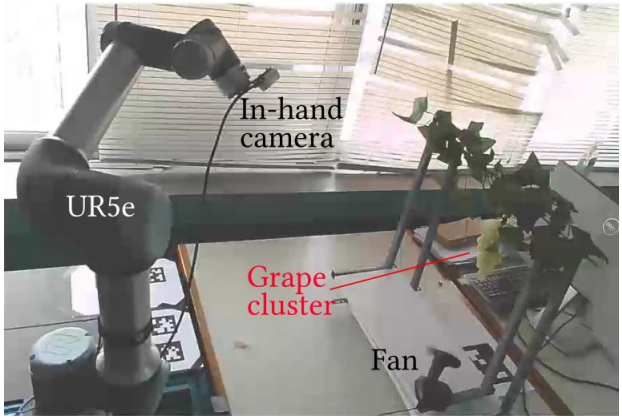
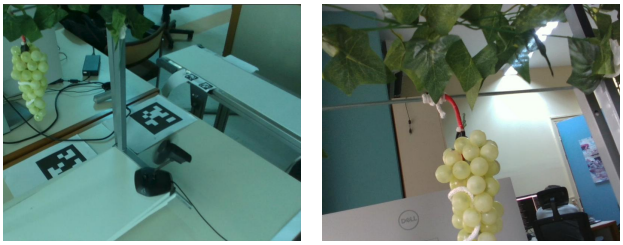


Fig. 6: Experimental setup.

A UR5e robot, under position control, fixed on a static

desk was utilized with a control cycle of 2ms. The reference trajectory was given as $\dot{\mathbf{q}} = \mathbf{J}^{\dagger}(\mathbf{q})\mathbf{u}_c$ where $\boldsymbol{u}_c$ was given by (1). The RGB-D camera was capturing the RGB point-cloud with a framerate of 15fps, a resolution of 1280x720 pixels and the point-cloud was uniformly under-sampled 5 times for decreasing the total number of points. Furthermore, the control code for (11) was parallelized[1], employing 6 parallel threads of the CPU, utilizing the OpenMP library in C++. Owing to the parallel implementation, $\mathbf{v}_{c2}$ is generated with a control cycle of approximately 67ms. In contrast, the ROI reaching & centering control signal, i.e. $\mathbf{v}_{c1}$, which does not generally involve a high computational load, had a control cycle of 2ms, i.e. it was synchronized with the control cycle of the robot. To superimpose these two control signals as dictated by (1), sample-and-hold was utilized for $\mathbf{v}_{c2}$. The control parameters were selected to be $d_c = 10^{-3}$m, $d_0 = 0.01$m, $k = 0.05$ for (7)-(11), $k_p = k_o = 5$ for (3) and a pole of $a = -2.5$ for the low pass filtering of the signal.

The initial camera view of the stalk is shown in Figure 7a. Notice that initially no points of the stalk are detected by the system, due to the initial distance from the stalk being more than the sensor's range. The path followed by the camera at the end-effector with the proposed controller is shown in Figure 8a and the final resulted camera view is provided in Figure 7b. Notice that due to the fact that no points are initially visible/detectable by the system, $\mathbf{v}_{c1}$ acts alone and therefore the robot moves initially towards the center of ROI until reaching the range in which the first points of the OOI are detected, i.e. until entering the red sphere in Fig. 8a).



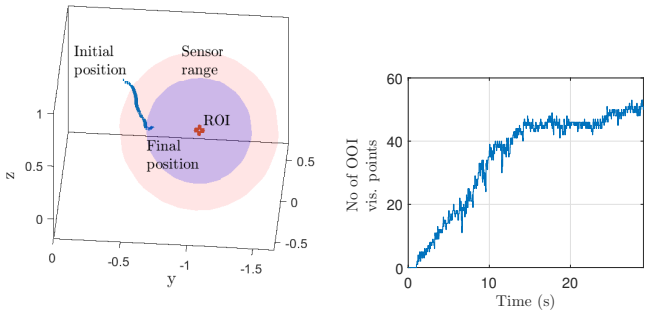
(a) Initial. **(b)** Final.

Fig. 7: Initial and final camera view.

The number of visible points of OOI during the motion evolution, under the proposed control scheme, is depicted in Fig. 8b. It is clearly visible that the number of visible points is increasing, which directly implies an increase at the visible area of the OOI from the camera, validating the results of the evaluations through simulations. In Figure 9, the time evolution of distance between the end-effector and the center of ROI is depicted, as well as the orientation error $\mathbf{k}\theta$ for centering of the view. Notice that the distance asymptotically reaches the radius of the ROI and the orientation error smoothly reaches zero, which demonstrates the fulfilment of the control objectives 1 and 2 respectively.

Experimental results demonstrate that the control signal for the ROI reaching & centering, i.e. $\mathbf{v}_{c1}$, and that responsible for the visual unveiling of the OOI, i.e. $\mathbf{v}_{c2}$, can be successfully superimposed achieving all set objectives for ROI reaching & centering and visual unveiling of the OOI.

[1]The code parallelization involves the summation of (11).



(a) Path of the in-hand camera. **(b)** Number of visible points during the experiment.

Fig. 8: Experimental reults.

Initial field-tests of the implementation of the proposed control scheme utilizing a mobile bi-manual robot with self-collision and joint limit avoidance are conducted on a real vineyard and can be found in the following link: `https://youtu.be/YFMtkUO2r7A`.

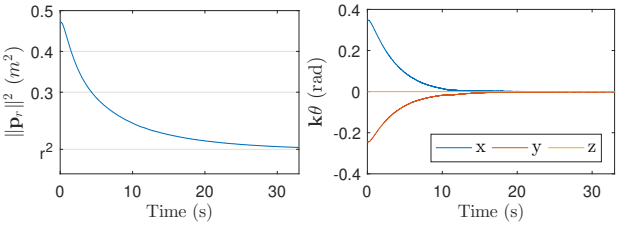


Fig. 9: Distance from the center of ROI and centering orientation error.

VII. Conclusions

This work addresses the problem of reaching and visually unveiling an object of interest by a robot with a camera in hand based on the point cloud perceived by the camera. The proposed solution is synthesized by superimposing two body velocities for the camera/end-effector. The control scheme effectiveness for reaching the vicinity of the object of interest while simultaneously optimizing the view of the object is revealed from both the simulation study and the experimental results. In particular, a success rate of 98% is shown to be achieved in simulations while the object of interest is shown to be significantly unveiled starting from just a few visible pixels, as demonstrated by the experimental results. Our future plans involve the extensive field-tests of the implemented method utilizing the BACCHUS mobile bi-manual robot on a real vineyard.

Appendix: Proof of Proposition 1

Equations (3), (4) can be written in the world frame as follows:

$$ {}^0\mathbf{v}_{c1} \triangleq \begin{bmatrix} k_p \max(0,\ f({}^0\mathbf{e}_p)) {}^0\mathbf{e}_p \\ k_o \theta {}^0\mathbf{k} \end{bmatrix}, \tag{12} $$

with ${}^0\mathbf{e}_p = {}^0\mathbf{p}_r - {}^0\mathbf{p}_c$, where ${}^0\mathbf{p}_c$ is the camera position and

$$ \theta = \cos^{-1}\left(\frac{\mathbf{z}^\intercal {}^0\mathbf{R}_c^\intercal {}^0\mathbf{e}_p}{\|{}^0\mathbf{e}_p\|} \right),\ {}^0\mathbf{k} = \frac{\mathbf{S}({}^0\mathbf{R}_c\mathbf{z}) {}^0\mathbf{e}_p}{\|\mathbf{S}({}^0\mathbf{R}_c\mathbf{z}) {}^0\mathbf{e}_p\|} \tag{13} $$

with ${}^0\mathbf{R}_c$ the rotation matrix corresponding to the camera orientation.

The kinematic system is $[{}^0\dot{\mathbf{p}}_c^\intercal\ {}^0\boldsymbol{\omega}_c^\intercal]^\intercal = {}^0\mathbf{v}_{c1}$. We will now prove that the directions of ${}^0\mathbf{e}_p$ and ${}^0\mathbf{k}$ remain constant during the system evolution. Differentiating $\frac{{}^0\mathbf{e}_p}{\|{}^0\mathbf{e}_p\|}$ we get:

$$ \frac{d}{dt}\frac{{}^0\mathbf{e}_p}{\|{}^0\mathbf{e}_p\|} = \frac{1}{\|{}^0\mathbf{e}_p\|}\left(\mathbf{I}_3 - \frac{{}^0\mathbf{e}_p}{\|{}^0\mathbf{e}_p\|}\frac{{}^0\mathbf{e}_p^\intercal}{\|{}^0\mathbf{e}_p\|} \right) {}^0\dot{\mathbf{e}}_p \tag{14} $$

But since ${}^0\dot{\mathbf{e}}_p$ is in the direction of $\frac{{}^0\mathbf{e}_p}{\|{}^0\mathbf{e}_p\|}$ from (12), (14) is equal to $\mathbf{0}$. Defining ${}^0\mathbf{a} \triangleq \mathbf{S}({}^0\mathbf{R}_c\mathbf{z}) {}^0\mathbf{e}_p = ({}^0\mathbf{R}_c\mathbf{z}) \times {}^0\mathbf{e}_p$, with $\times$ denoting the cross product operator and differentiating the second equation of (13) we get:

$$ {}^0\dot{\mathbf{k}} = \frac{d}{dt}\frac{{}^0\mathbf{a}}{\|{}^0\mathbf{a}\|} = \frac{1}{\|{}^0\mathbf{a}\|}\left(\mathbf{I}_3 - {}^0\mathbf{k}\,{}^0\mathbf{k}^\intercal \right) {}^0\dot{\mathbf{a}} \tag{15} $$

where

$$ {}^0\dot{\mathbf{a}} = {}^0\boldsymbol{\omega}_c \times {}^0\boldsymbol{R}_c\boldsymbol{z} \times {}^0\mathbf{e}_p + \left({}^0\boldsymbol{R}_c\boldsymbol{z} \right) \times {}^0\dot{\mathbf{e}}_p \tag{16} $$

Using the double cross product identity we get:

$$ {}^0\dot{\mathbf{a}} = ({}^0\mathbf{e}_p^\intercal {}^0\boldsymbol{\omega}_c) {}^0\boldsymbol{R}_c\boldsymbol{z} - ({}^0\mathbf{e}_p^\intercal {}^0\boldsymbol{R}_c\boldsymbol{z}) {}^0\boldsymbol{\omega}_c + \left({}^0\boldsymbol{R}_c\boldsymbol{z} \right) \times {}^0\dot{\mathbf{e}}_p \tag{17} $$

From the second equation of (12) and (13), ${}^0\boldsymbol{\omega}_c$ is orthogonal to ${}^0\boldsymbol{R}_c\boldsymbol{z}$, thus ${}^0\mathbf{e}_p^\intercal {}^0\boldsymbol{\omega}_c = 0$. Furthermore, ${}^0\dot{\mathbf{e}}_p$ is in the direction of ${}^0\mathbf{e}_p$ and thus $\left({}^0\boldsymbol{R}_c\boldsymbol{z} \right) \times {}^0\dot{\mathbf{e}}_p$ is in the direction of ${}^0\mathbf{k}$. Since ${}^0\boldsymbol{R}_c\boldsymbol{z}$ is also in the direction of ${}^0\mathbf{k}$, we infer that ${}^0\dot{\mathbf{a}}$ is in the direction of ${}^0\mathbf{k}$. Using the fact that ${}^0\boldsymbol{R}_c\boldsymbol{z}$ is in the direction of ${}^0\mathbf{k}$ we can conclude that ${}^0\dot{\mathbf{k}}$ is zero. This implies that the second row of (12) which can be interpreted as the logarithmic orientation error, is the error between the camera orientation ${}^0\mathbf{R}_c$ and a constant orientation ${}^0\mathbf{R}_d$ such that $\theta {}^0\mathbf{k} = \log {}^0\mathbf{R}_d {}^0\mathbf{R}_c^\intercal$.

The angular velocity corresponding to the orientation error ${}^0\widetilde{\mathbf{R}} \triangleq {}^0\mathbf{R}_d {}^0\mathbf{R}_c^\intercal$ is equal to ${}^0\widetilde{\boldsymbol{\omega}} \triangleq {}^0\boldsymbol{\omega}_d - {}^0\widetilde{\mathbf{R}} {}^0\boldsymbol{\omega}_c$. Since ${}^0\mathbf{R}_d$ is constant, ${}^0\boldsymbol{\omega}_d = \mathbf{0}$. We can also utilize the Rodrigues' rotation formula to get ${}^0\widetilde{\mathbf{R}} {}^0\boldsymbol{\omega}_c = (\mathbf{I}_3 + \mathbf{S}({}^0\mathbf{k})(\sin\theta) + \mathbf{S}^2({}^0\mathbf{k})(1-\cos\theta)) {}^0\boldsymbol{\omega}_c = \boldsymbol{\omega}_c$. Therefore ${}^0\widetilde{\boldsymbol{\omega}}$ is equal to $-{}^0\boldsymbol{\omega}_c$. However the following equation holds [18]:

$$ {}^0\widetilde{\boldsymbol{\omega}} = \dot{\theta} {}^0\mathbf{k} + \sin\theta {}^0\dot{\mathbf{k}} + (1 - \cos\theta)\mathbf{S}({}^0\mathbf{k}) {}^0\dot{\mathbf{k}} \tag{18} $$

Since ${}^0\dot{\mathbf{k}} = \mathbf{0}$, (18) implies ${}^0\boldsymbol{\omega}_c = -\dot{\theta} {}^0\mathbf{k}$. and thus:

$$ \dot{\theta} = -k_o \theta \tag{19} $$

Define the Lyapunov-like function W:

$$ W = \frac{1}{4}\max\left(0, f({}^0\mathbf{e}_p)\right)^2 + \frac{1}{2}\theta^2 \tag{20} $$

Differentiating W using (12) and (19) yields:

$$ \dot{W} = -k_p \max\left(0, f({}^0\mathbf{e}_p)\right)^2 {}^0\mathbf{e}_p^\intercal {}^0\mathbf{e}_p - k_o\theta^2 \tag{21} $$

Notice that $\dot{W}$ is zero, if and only if both the Objectives 1 and 2 are satisfied and it is negative in all other cases.

REFERENCES

[1] 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.

[2] C. Lehnert, D. Tsai, A. Eriksson, and C. McCool, "3d move to see: Multi-perspective visual servoing towards the next best view within unstructured and occluded environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 3890–3897.

[3] A. Gongal, S. Amatya, M. Karkee, Q. Zhang, and K. Lewis, "Sensors and systems for fruit detection and localization," *Comput. Electron. Agric.*, vol. 116, no. C, p. 8–19, Aug. 2015.

[4] S. D. Roy, S. Chaudhury, and S. Banerjee, "Active recognition through next view planning: a survey," *Pattern Recognition*, vol. 37, no. 3, pp. 429–446, 2004.

[5] M. Farrokhsiar, G. Pavlik, and H. Najjaran, "An integrated robust probing motion planning and control scheme: A tube-based mpc approach," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1379–1391, 2013.

[6] 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.

[7] N. Cowan, J. Weingarten, and D. Koditschek, "Visual servoing via navigation functions," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 521–533, 2002.

[8] S. LaValle, H. Gonzalez-Banos, C. Becker, and J.-C. Latombe, "Motion strategies for maintaining visibility of a moving target," in *Proceedings of International Conference on Robotics and Automation*, vol. 1, 1997, pp. 731–736 vol.1.

[9] M. A. Baumann, D. C. Dupuis, S. Leonard, E. A. Croft, and J. J. Little, "Occlusion-free path planning with a probabilistic roadmap," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 2151–2156.

[10] M. Kazemi, K. Gupta, and M. Mehrandezh, "Global path planning for robust visual servoing in complex environments," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 326–332.

[11] D. Nicolis, M. Palumbo, A. M. Zanchettin, and P. Rocco, "Occlusion-free visual servoing for the shared autonomy teleoperation of dual-arm robots," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 796–803, 2018.

[12] 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)*, 2020, pp. 13–18.

[13] A. Khelloufi, N. Achour, R. Passama, and A. Cherubini, "Sensor-based navigation of omnidirectional wheeled robots dealing with both collisions and occlusions," *Robotica*, vol. 38, no. 4, p. 617–638, 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)*, 2019, pp. 8762–8768.

[15] M. Ourak, B. Tamadazte, O. Lehmann, and N. Andreff, "Direct visual servoing using wavelet coefficients," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 3, pp. 1129–1140, 2019.

[16] C. Cheah and D. Wang, "Region reaching control of robots: Theory and experiments," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 974–979.

[17] T. Kastritsi, D. Papageorgiou, I. Sarantopoulos, S. Stavridis, Z. Doulgeri, and G. A. Rovithakis, "Guaranteed active constraints enforcement on point cloud-approximated regions for surgical applications," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8346–8352.

[18] L. Koutras and Z. Doulgeri, "Exponential stability of trajectory tracking control in the orientation space utilizing unit quaternions," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.