Visual Servoing for Vine Pruning based on Point Cloud Alignment

Fadi Gebrayel, Martin Mujica, Patrick Danès

Visual Servoing for Vine Pruning based on Point Cloud Alignment

Fadi Gebrayel[1], Martin Mujica[1] and Patrick Danès[1]
[1]*LAAS-CNRS, Université de Toulouse, CNRS, UPS, Toulouse, France.*
{*fadi.gebrayel, martin.mujica, patrick.danes*}*@laas.fr*

Keywords: Visual Servoing, Computer Vision, ICP, Vine Pruning, Agriculture Robotics

Abstract: This paper addresses the challenge of vine pruning, a crucial and laborious task in agriculture, using robotic technologies and vision based feedback control. The complex structure of vines makes visual servoing difficult due to challenges in 3D pose estimation and feature extraction. A novel approach to vision based vine pruning is proposed, based on the combination of Iterative Closest Point (ICP) point-cloud alignment and position-based visual servoing (PBVS). Four ICP variants are compared within PBVS in vine pruning scenarios: standard ICP, Levenberg–Marquardt ICP, Point-to-Plane ICP, and Symmetric ICP. The methodology includes a dedicated ICP initial guess to improve alignment speed and accuracy, as well as a procedure for generating reference point clouds at pruning locations. Live experiments were conducted on a Franka Emika manipulator equipped with a stereo camera, involving three real vines under laboratory conditions.

1 INTRODUCTION

Pruning is a critical task in vineyard management, which determines the quality and yield of the harvest. Traditionally, this task is laborious, time-consuming, and uncomfortable for workers, as it must be performed in cold weather and requires awkward postures. While robotic solutions have been proposed to pruning, fully autonomous systems have yet to replace manual pruning. Indeed, the complexity and variability of the structure of a vine hinders its perception as well as the planning and control of the motion of the cutting tool. To address this challenge, robots must integrate control systems with feedback from exteroceptive sensors, so as to adapt to environmental changes such as unexpected relative motion of the vine (due to vibrations, collisions, etc.).

The typical workflow of an autonomous vine pruning robot consists of the following cycle: autonomous movement to the next vine stock; perception of the plant, geometric and semantic modeling; determination of pruning poses; collision-free planning and control of the cutting device(s) motion towards these poses; cutting operation. However, in real-life conditions, vibrations or collisions with unperceived vine elements are likely, requiring sensor-based feedback (e.g., visual or force) to ensure successful cutting.

In research laboratories, several initiatives have emerged towards vine pruning robots. The prototype (Botterill et al., 2017) entails advanced computer vision, machine learning and motion, but includes no local vision-effort control of the end-effector cutting tool. According to the authors, tests show that the long chain of interdependent components limits feasibility. The Bumblebee robot (Silwal et al., 2022) also features such advanced techniques, together with a robust mechanical design. A significant effort is put on autonomous navigation, but no local exteroceptive control of the pruning shears is envisaged. In (Yandun et al., 2021), the robot's reference movement is generated by combining reinforcement learning and inverse kinematics, yet the authors explicitly mention that its execution is done in open-loop. Though targeting a less integrated solution, (Katyara et al., 2021) presents perception, cutting points learning, motion planning, but also addresses vine rods dynamics, admittance control and passivity maintenance.

As pruning is a complex task from a motion planning viewpoint, (You et al., 2020) analyzes ways of reducing planning time and sequencing the cutting points. Among the used common control laws, vision-based or vision-force feedback control strategies have very recently been considered in isolation, for fruit pruning (Zahid et al., 2021) or harvesting (You et al., 2022; Li et al., 2022). The harvesting technique (Gursoy et al., 2023) uses dual robotic arms guided by visual data to detect fruits and tree trunk. Joint velocity control is implemented through hierarchical quadratic programming, facilitating collision

avoidance with previously identified trunks. However, no visual information is provided, which could be used for feedback control. Besides, a visual servoing approach is proposed in (Mehta et al., 2014) for harvesting tasks, which features a nonlinear controller with fruit motion compensation.

The recent work (Zhang et al., 2020) incorporates iterative closest point (ICP) based point cloud registration into visual servoing. Though some robustness to illumination changes is obtained, the considered environments are large and dense, in contrast to a pruning scene. Therefore, the present work evaluates ICP based visual servoing strategies for vine pruning. The contributions are the following: endow ICP and position based visual servoing (ICP-PBVS) with an adaptive initial guess and real time capabilities to overcome the challenges of 3D pose estimation in the pruning context.; provide a thorough comparison of the influence of some ICP techniques on servoing performance (Besl and McKay, 1992; Fitzgibbon, 2003; Chen and Medioni, 1992; Rusinkiewicz, 2019); propose an approach to the generation of a synthetic reference point-cloud, as well as an experimental evaluation on complex vine pruning tasks. Importantly, our approach is well suited to a broader class of agricultural tasks, where mobile robotic manipulators (on tractors, etc.) must accurately reach cutting/gripping points.

In the sequel, an accurate 3D model of the vine is available a priori (through point-cloud registration (Choi et al., 2015) or along Section 4.4.3). It includes the reference (desired) pruning poses (e.g., through Simonit & Sirch vine pruning rules). The aim is to successfully position the robot pruner end-of-arm tooling (EOT) in spite of likely disturbances: vine motion due to past small collisions, platform vibrations, among others. Nonrestrictively, local vine rigidity around pruning poses is assumed.

Next Sections 2 and 3 provide theoretical background on ICP-PBVS and expand the algorithm. Section 4 outlines and analyzes experimental results on a Franka Emika Panda 7DOF manipulator robot equipped with a eye-in-hand stereo camera. Section 5 concludes the paper.

2 FUNDAMENTALS

2.1 Visual servoing

Visual servoing can be stated as the regulation to zero of the error (Chaumette and Hutchinson, 2006)

$$e(t) = s(m(t), a) - s^*, \tag{1}$$

with: s the vector of features used for feedback; s^* its reference value; $m(t)$ a vector of image measurements (e.g., from a eye-in-hand camera); a a vector of additional known parameters (camera, objects...). When s is the camera-to-target relative pose, the control strategy is termed "position based visual servoing" (PBVS). It requires a pose estimation algorithm. Conversely, visual servoing can be specified without resorting to pose estimation, by building s and s^* with the current and reference values of 2D features extracted from the image. The consequent "image based visual servoing" (IBVS) is known to be computationally cheaper and to offer higher robustness to calibration errors. Yet, its design may be difficult, e.g., when feature extraction/matching is troublesome.

Consider the positioning of a robot pruner EOT using visual feedback from a eye-in-hand camera. Dynamics of the camera (or EOT) are often neglected, so the control input is set to the camera velocity screw

$$v = [v_c, \omega_c]^T, \tag{2}$$

i.e., its translation and rotation velocities (expressed in its frame). The standard controller synthesis method is as follows: (i) set the open-loop model

$$\dot{s} = Lv, \tag{3}$$

where the so-called interaction matrix L is generally parameterized by s and some 3D pose variables; (ii) select the ideal closed-loop dynamics of e, e.g., the stable, first-order linear and decoupled equation

$$\dot{e} = -\lambda e,\ \lambda \in \mathbb{R}_+; \tag{4}$$

(iii) deduce the feedback control law

$$v = -\lambda \hat{L}^+ e, \tag{5}$$

with $\hat{L}$ a matrix gain that approximates L, and $\hat{L}^+$ its pseudo-inverse. Potential pitfalls of such approaches are well-known: for PBVS, the induced linear path in the space of pose variables may make the targets entailed in pose estimation leave the sensor field-of-view; in IBVS, the exponential decoupled convergence of 2D features to their reference values may imply unacceptable 3D motion of the camera.

The reference camera frame F_{c^*}, the current camera frame F_c, and the object frame F_o are central to classical PBVS. In the sequel, ${}^i t_j$ stands for the expression in F_i of the translation vector from the origin of F_i to the origin of F_j, with $i, j \in \{c^*, c, o\}$. Similarly, ${}^i R_j$ and ${}^i H_j = \begin{pmatrix} {}^i R_j & {}^i t_j \\ 0 & 1 \end{pmatrix}$ stand for the rotation and homogeneous matrices associated to the rigid transform that turns F_i into F_j. Defining (θ, u) as the (angle,unit_vector) pair equivalent to ${}^{c^*} R_c$, one sets

$$s = ({}^{c^*} t_c, \theta u),\ s^* = 0,\ e = s. \tag{6}$$

The poses depicted by ${}^{c^*}H_o$ and ${}^{c}H_o$ must be computed from visual data, in order to deduce ${}^{c^*}H_c$. For the above selection of s, the interaction matrix L comes as (Chaumette and Hutchinson, 2006)

$$L=\begin{pmatrix} {}^{c^*}R_c & 0 \\ 0 & L_{\theta u} \end{pmatrix}; L_{\theta u} = I_3 - \frac{\theta}{2}[u]_\times + (1-\frac{\operatorname{sinc}\theta}{\operatorname{sinc}^2\frac{\theta}{2}})[u]_\times^2, \quad (7)$$

what leads to the conventional PBVS feedback

$$v_c = -\lambda\, {}^{c^*}R_c^T\, {}^{c^*}t_c, \quad \omega_c = -\lambda\theta u, \quad (8)$$

with $[u]_\times$ the cross-product tensor associated to vector u.

2.2 3D Point Cloud Alignment

Getting the rigid transformation from the reference camera frame F_{c^*} to the actual camera frame F_c can be viewed as finding the relative pose that aligns the point cloud observed at F_{c^*} to that observed at the F_c. This relative pose can be obtained through iterative point cloud alignment methods.

Iterative Closest Point (ICP) The ICP algorithm iteratively selects corresponding points from two point clouds and computes the relative homogeneous transform which minimizes the sum of squared distances between pairs of matching points. A threshold limits the maximum matching distance for points with no correspondent. This process is detailed in Algorithm 1.

Inputs: point cloud $A = \{a_j\}$
point cloud $B = \{b_j\}$
initial guess homogeneous matrix H_0

Output: homogeneous matrix H that aligns point clouds A and B

Parameters: point cloud size N
threshold *threshold*

$H \leftarrow H_0$
while *not_converged* **do**
  for $j \leftarrow 1$ **to** N **do**
    $m_j \leftarrow FindClosestPointInA(Hb_j)$
    if $\|m_j - Hb_j\| \leq threshold$ **then**
      $w_j \leftarrow 1$
      else $w_j \leftarrow 0$;
    end
  end
  $H \leftarrow \arg\min_{\check{H}} J(\check{H}) := \sum_j w_j \|\check{H}b_j - m_j\|^2$
end

Algorithm 1: Standard ICP algorithm

Levenberg–Marquardt ICP The LM-ICP (Fitzgibbon, 2003) employs a general-purpose nonlinear optimization solver, the Levenberg–Marquardt algorithm, to minimize the registration error $J(\check{H})$ supporting Algorithm 1. This improves the convergence speed of ICP without increasing the computation time.

Point-to-plane ICP The PP-ICP (Chen and Medioni, 1992) entails the distance from any point to the plane tangent to the other point cloud at its matching point. The registration error to be minimized writes as

$$J_{\text{PP-ICP}}(\check{H}) := \sum_j w_j \big(\eta_j.(\check{H}b_j - m_j)\big)^2 \quad (9)$$

where η_j stands for the surface normal at m_j. This results in improved accuracy and faster convergence.

ICP with symmetric objective function The SYMM-ICP (Rusinkiewicz, 2019) incorporates a symmetrized version of the point-to-plane ICP registration error. In other words, each plane entailed in the objective function is defined from the tangent planes (or, equivalently, the surface normals) at both points within a corresponding pair. The registration error writes as

$$J_{\text{SYMM-ICP}}(\check{H}) := \sum_j w_j \big((\eta_{a,j}+\eta_{b,j}).(\check{R}b_j - \check{R}^{-1}a_j + \check{t})\big)^2 \quad (10)$$

where $\check{t}, \check{R}$ term the translation vector and rotation matrix of the homogeneous matrix decision variable $\check{H}$, and $\eta_{a,j}, \eta_{b,j}$ respectively stand for the surface normal at points a_j, b_j.

3 ICP BASED PBVS

Accurately estimating vine shoot poses is undeniably challenging. Irregular shoot shapes, potential occlusions, and varying lighting conditions constitute significant hurdles. Moreover, the lack of specialized branch models and the critical real-time processing requirement intensify the challenge.

In the vein of (Zhang et al., 2020), a point cloud based PBVS is introduced, where the ICP algorithm estimates the pose that aligns the current and reference point clouds. Its schematic diagram is shown on Figure 1. The ICP block compares a predefined reference point cloud with the current point cloud obtained through the vision system. It outputs the resulting homogeneous transformation between the reference and current camera frame, possibly using an initial guess (detailed further) to launch the computation. The PBVS block takes this matrix and, following the developments in 2.1, computes the camera

velocity screw. Finally, this control signal delivered by the outer visual servo loop is turned into the robot end-effector velocity screw, which constitutes the set-point of the inner robot controllers.

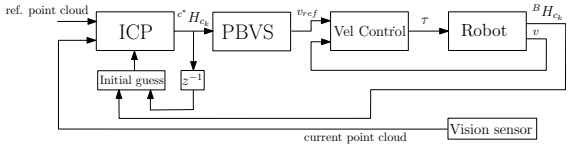


Figure 1: Schematic diagram of the proposed ICP based PBVS.

As aforementioned, the ICP algorithm requires an initial guess to initiate the point cloud alignment. The default implementation sets it to the identity matrix, which is clearly nonoptimal. An online initial guess selection method is described below, based on an approximation of the camera pose and ICP output.

Let ${}^{c^*}H_{c_{k-1}}$ and ${}^{c^*}H_{c_k}$ stand for the homogeneous matrices from F_{c^*} to the camera frames $F_{c_{k-1}}, F_{c_k}$ at respective visual servoing iterations $k-1, k$. Let ${}^{B}H_{c_{k-1}}$ and ${}^{B}H_{c_k}$ be the homogeneous matrices from the robot base frame F_B to $F_{c_{k-1}}, F_{c_k}$. At iteration $k-1$, ICP outputs an estimate of ${}^{c^*}H_{c_{k-1}}$. Similarly, robot kinematics enables approximations to ${}^{B}H_{c_{k-1}}$ and ${}^{B}H_{c_k}$ at iterations $k-1$ and k. Using the same notations for approximations and genuine values of homogeneous transforms, an initial guess for ICP at iteration k comes as

$$ {}^{c^*}H_{c_k} = {}^{c^*}H_{c_{k-1}} ({}^{B}H_{c_{k-1}})^{-1} \, {}^{B}H_{c_k}. \qquad (11) $$

4 EXPERIMENTS ON REAL VINES

4.1 Experimental setup and procedure

This section presents the experimental analysis of point-cloud based PBVS techniques for vine pruning. The setup is composed of a Franka Emika Panda robot endowed with the ROS middleware and a eye-in-hand Realsense D405 camera. This camera relies on stereovision, and is supported by ROS. The inner Franka controllers are implemented on a real-time kernel at 1kHz. The outer visual feedback is implemented in free-running mode, so its rate may vary, e.g., depending on the ICP behavior. Real vine stocks, with increasing complexity, are used for experimental tests. As the focus is to analyze and assess the proposed visual servoing framework, the environment is controlled (stable light conditions and incorporation of a textured background, to enhance the point cloud quality). The system runs on an Intel i7 processor under Ubuntu 20.04 with ROS noetic, PCL (Holz et al., 2015) and ViSP (Marchand et al., 2005) libraries (Figure 2).

Figure 2: Experimental setup composed of the robot Franka Emika, the Realsense camera D405 and one vine branch.

All point clouds are built from rectified RGB images. Their complexity is reduced by downsampling. Points beyond a specified depth threshold are removed. Irregular points are also removed by a statistical outlier filter.

In real-world scenarios, the reference (desired) point cloud around each cutting pose would typically be obtained as a fragment of the prior 3D model of the vine. To simplify comparisons and focus on servoing, it is hereafter obtained by first positioning the camera at the desired position.

Figure 3 illustrates an overview of the process. In particular, Figures 3a and 3b display the starting and reference RGB images. Importantly, these are not used for feedback, but just illustrate the experimental conditions. In Figure 3c, the reference (obtained in the condition of Fig. 3b) and starting (obtained in the condition of Fig.3a) point clouds are respectively shown in white and in color. Finally, Figure 3d shows how the reference and current point clouds overlap each other after convergence.

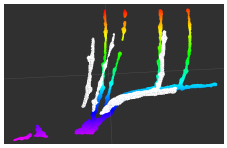
The first experiments (Sections 4.2, 4.3) have been run on non-dense branches (Figure 3). They share a common point cloud and reference position. The aim is to set a fair comparison of the ICP-PBVS methods for distinct starting camera poses. A second set of experiments addresses denser, more complex, vine structures (Section 4.4) as well as moving vines so as to get closer to real-life scenarios subject to external disturbances.

4.2 Influence of ICP initial guess

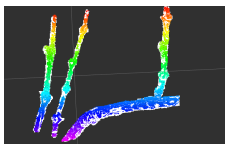
This section discusses the incorporation into ICP of the initial guess suggested in Section 3. First, it has been observed that this ICP initialization enables a much faster outer visual based feedback loop: the resulting increase, around 15 Hz for all proposed ICP

(a) RGB Image at the initial pose.

(b) RGB Image at the reference pose.



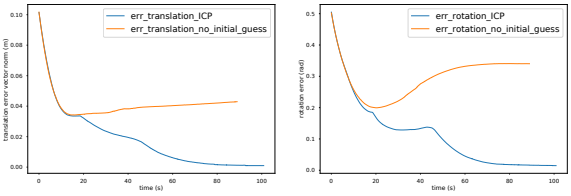
(c) Reference and initial point clouds.



(d) Reference and current point clouds at the end of the test.

Figure 3: Overview the experimental conditions, with the initial and reference images and their point clouds.

variations, is significant, especially considering that the camera frequency (i.e., the maximum admissible outer-loop frequency) is 60 Hz. Benefits include not only real time performance, but also feedback system accuracy or even stability. Indeed, lengthy ICP computations may increase the visual servoing period, potentially causing the robot to drift from the ideal continuous-time closed-loop dynamics (4), especially for dense vines. A limiting behavior may even be reached, where the vine leaves the camera field of view. A comparison of a servoing task in a simple vine with and without our proposed initial guess is shown in Figure 4, illustrating the mentioned behavior. Consequently, this adaptive ICP initial guess is henceforth incorporated in all ICP-PBVS.



(a) Linear error (b) Rotation Error

Figure 4: ICP-PBVS with and without initial guess.

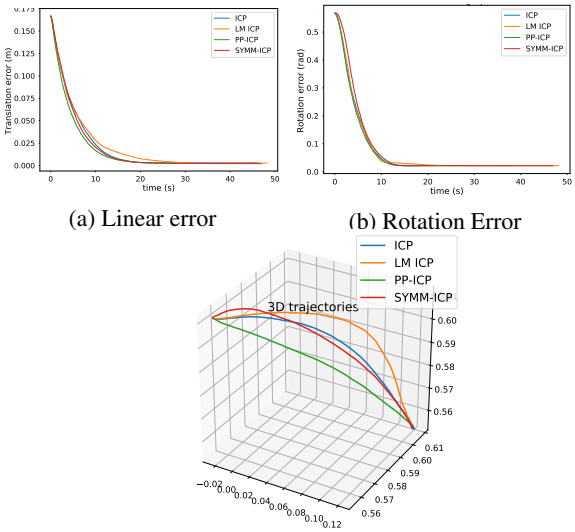
4.3 Comparison of different ICP methods

Table 1 summarizes the quantitative evaluation of a four-test experiment on sparsely structured branches. Each test includes significant translation and/or rotation differences between starting and reference poses, potentially making visual servoing fail due to differing viewpoints of the vine.

SYMM-ICP-PBVS outperforms other methods regarding settling time in the first test. However, PP-ICP-PBVS demonstrates faster convergence in the second and fourth tests. So does LM-ICP-PBVS in the third test. Convergence failures in the third and fourth tests show that both PP-ICP-PBVS and SYMM-ICP-PBVS encounter difficulties due to significant initial rotation errors, while LM-ICP seems sensitive to significant translation errors. Nevertheless, ICP-PBVS converges in all cases.

As for accuracy, concerning the translation and rotation Final Static Error (FSE), no definitive conclusion can be drawn globally, as no servoing method is uniformly better. Translation and rotation errors during servoing are depicted in Figure 5 for all the ICP variants in the second test case, along with the 3D trajectories. The smoothest errors are exhibited by PP-ICP-PBVS.

The drift of the induced 3D trajectories (shown on Figure 5c) w.r.t. the theoretical straight line can be attributed to unexpected abrupt errors between the homogeneous transforms estimated by ICP (or its variants) and their genuine values, which in turn lead to abrupt changes in the camera control input. It can also stem from a change of rate of the outer visual feedback: the longer the computation of the homogeneous matrix by the ICP, the longer the zero-order-hold of the camera velocity screw produced by the outer controller, and the less suited the resulting setpoint to the inner robot controller. The proposed ICP initial guess reduces this problem.



(a) Linear error (b) Rotation Error

(c) Camera 3D trajectory for each ICP method

Figure 5: Second test, non-dense branche structure: comparison of different ICP-PBVS methods.

While not being the fastest method, in view of its consistent convergence performance the ICP-PBVS

Test 1 - inial translation error: 0.184749 (m) initial rotation error: 0.351915 (rad)				
	ICP	LM ICP	PP-ICP	SYMM-ICP
Translation FSE (m)	0.003072	0.003333	X	0.002912
Rotation FSE (rad)	0.002162	0.003612	X	0.004765
Convergence time (s)	24.894065	33.757214	X	14.266417
Test 2 - initial translation error: 0.166275 (m) initial rotation error: 0.567889 (rad)				
Translation FSE (m)	0.002969	0.002691	0.002524	0.002115
Rotation FSE (rad)	0.022047	0.021488	0.022144	0.021275
Convergence time	18.307688	26.528462	17.730364	18.238572
Test 3 - initial translation error: 0.034708 (m) initial rotation error: 0.550662 (rad)				
Translation FSE (m)	0.002027	0.001722	X	X
Rotation FSE (rad)	0.02138	0.022581	X	X
Convergence time	16.85218	16.624503	X	X
Test 4 - initial translation error: 0.316327 (m) initial rotation error: 0.12695 (rad)				
Translation FSE (m)	0.002007	X	0.003296	0.004129
Rotation FSE (rad)	0.019513	X	0.01839	0.016704
Convergence time	8.834689	X	6.744938	7.37105

Table 1: Comparative experiment campaign, involving a non-dense branch structure. For each test, the best results for each criterion are highlighted in green.

strategy has been selected for all the next experimental evaluations.

4.4 Complex cases

4.4.1 Moving vine branches

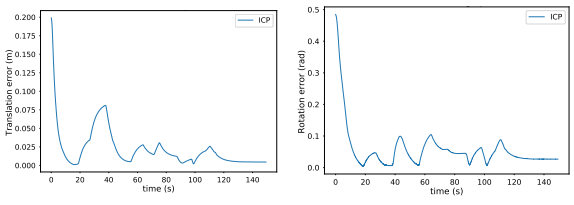
As aforementioned, the interest of the approach lies in its ability to handle unexpected variations that produce a discrepancy w.r.t. prior knowledge. So, experiments have been carried out to study how ICP-PBVS can cope with vine vibrations and unexpected movements (due for instance to unexpected collisions of the robot and the vine). Therefore, after reaching convergence, the vine is abruptly moved. Nevertheless, the camera is successfully driven to the pose where the current and reference point clouds are aligned. This can be seen in the error curves shown in Figure 6: sudden error increases (induced by sudden vine movements) are followed by an exponential convergence decay to zero. A more detailed analysis of this experimental scenario can be found in the companion video. [1]

4.4.2 Dense vine branches

After initial tests on real vines, experiments were extended to a more intricate environment characterized by densely packed branches, as illustrated in Figure 7.

[1] Video link

Three tests were conducted with two setups, considering the same reference point cloud (i.e., the same final pose), but with different initial conditions. Vine in test 2 (Figure 7b, 7d), presents an increased complexity due to the presence of numerous branches and residual dried grapes, thereby intensifying perception challenges. Table 2 displays the results, including errors and convergence time, obtained from the conducted tests. Moreover, Figure 8 presents the 3D trajectories of the three tests in each campaign, highlighting the successful convergence of ICP-PBVS in all tests from various starting positions towards the reference point. The findings reveal acceptable errors, notably in translation, albeit with slightly larger rotation errors, particularly evident in the second dense setup. Moreover, the observed prolonged convergence time may be attributed to the fact that the controller's gain was reduced to handle the higher computational demand arising from larger point clouds associated with densely branched vines. These findings highlight the effectiveness of ICP-PBVS in challeng-



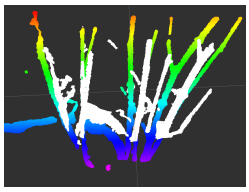
(a) Linear error (b) Rotation Error

Figure 6: Evolution of translation and rotation errors when unexpected vine movement is introduced.

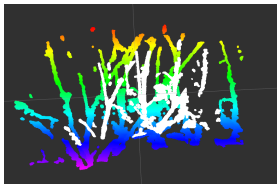
ing vine environments and its potential for pruning tasks, while providing insight on possible improvements for fast and real outdoors agricultural tasks.

(a) Test 1: RGB image at the initial pose.

(b) Test 2: RGB image at the initial pose.



(c) Test 1: Reference and initial point clouds.

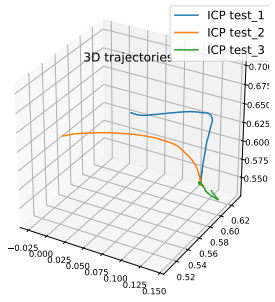


(d) Test 2: Reference and initial point clouds.

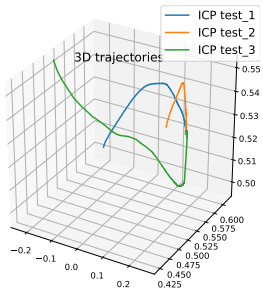
Figure 7: Overview of the case of dense branches tests, with the initial and reference images and their point clouds.

Test with dense vine 1			
Init. error (m, rad)	(0.193, 0.35)	(0.167, 0.566)	(0.034, 0.55)
Trans. FSE (m)	0.002914	0.001775	0.003168
Rot. FSE (rad)	0.00734	0.00563	0.045401
Conv. time (s)	73.682	59.895	42.364
Test with dense vine 2			
Init. error (m, rad)	(0.185, 0.22)	(0.20, 0.168)	(0.037, 0.47)
Tran. FSE (m)	0.001577	0.001477	0.003454
Rot. FSE (rad)	0.011115	0.011791	0.015248
Conv. time (s)	99.234	89.485	125.401

Table 2: First and second experiment campaign results, involving a dense branch structure.



(a) First test campaign



(b) Second test campaign

Figure 8: 3D Trajectories with dense branches.

4.4.3 Sim-to-Real Reference Point Cloud

As previously stated, in real-world applications the reference point cloud for each pruning pose must be derived from a 3D model of the vine. Thus we built a fine 3D model of the vine before the first pruning process, using the same camera. Subsequently, this 3D model was included in a simulated environment enabling the virtual selection of any pruning pose, so that all the reference point clouds could be recovered. These data were then transferred to the real-world servoing process, to apply the same visual feedback control as above. Figure 9 delineates the complete framework. The simulation was done in Gazebo. The pre-built 3D vine model was integrated as a mesh. The pruning pose was defined at a reference pose 15 cm from the cutting point, where the simulated Realsense D405 camera was positioned. In Figure 9, it can be seen how the synthetic reference point cloud is given to the online controller, and how the robot moves to align them using ICP-PBVS, resulting in the convergence to the cutting pose, as in the previous experiments.

5 CONCLUSIONS

This work presented a working real-time ICP based PBVS (ICP-PBVS) for vine pruning, along with its analysis for several ICP variants. Contrary to the proprioception based execution of a planned trajectory, visual servoing can adjust the cutting tool positioning in real time when facing unexpected environment changes. As the 3D trajectory followed by the pruning tool is of interest in the considered agriculture context, PBVS is preferred over IBVS. The proposed incorporation of a relevant initial guess reduces the ICP computation time. A side effect is the increase of the feedback control rate, which in turns positively impacts the closed-loop system stability thanks to shorter zero-order-hold of the control signal.

Experiments were conducted, combining four ICP variants with PBVS. Standard ICP showed the best performance in tests on simple vines, with further successful experiments on moving and complex vines. On the whole, ICP-PBVS adapts well to disturbances, making it suitable for pruning task and broader agricultural applications. However, the processing of big point clouds and the computational cost may remain an issue. Finally, an approach to generate the reference point cloud through simulation at the desired pose was presented and evaluated.

Future work will focus on alternative alignment methods to reduce computing time, improved control laws keeping the point cloud in the camera view, and tests in real-world outdoor conditions with more complex factors like lighting and wind.

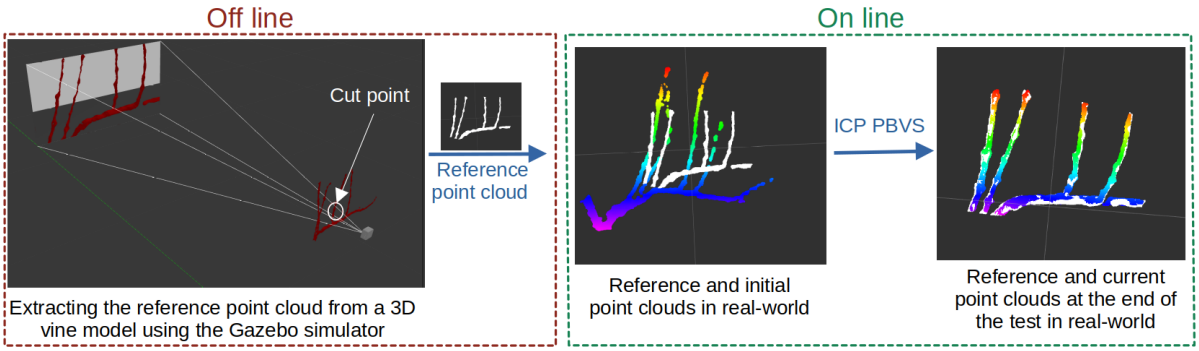


Figure 9: Sim to real Reference point cloud extraction scheme.

ACKNOWLEDGEMENTS

This work was supported by the "Défi Clé Robotique centrée sur l'humain" funded by La Région Occitanie, France.

REFERENCES

Besl, P. and McKay, N. (1992). Method for registration of 3-D shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611. Spie.

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

Chaumette, F. and Hutchinson, S. (2006). Visual servo control. I. basic approaches. *IEEE Robotics & Automation Magazine*, 13(4):82–90.

Chen, Y. and Medioni, G. (1992). Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3).

Choi, S., Zhou, Q.-Y., and Koltun, V. (2015). Robust reconstruction of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5556–5565.

Fitzgibbon, A. (2003). Robust registration of 2D and 3D point sets. *Image and Vision Computing*, 21(13-14).

Gursoy, E., Navarro, B., Cosgun, A., Kulić, D., and Cherubini, A. (2023). Towards vision-based dual arm robotic fruit harvesting. In *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, pages 1–6.

Holz, D., Ichim, A., Tombari, F., Rusu, R., and Behnke, S. (2015). Registration with the point cloud library: A modular framework for aligning in 3-D. *IEEE Robotics & Automation Magazine*, 22(4).

Katyara, S., Ficuciello, F., Caldwell, D., Chen, F., and Siciliano, B. (2021). Reproducible pruning system on dynamic natural plants for field agricultural robots. In Saveriano, M., Renaudo, E., Rodríguez-Sánchez, A., and Piater, J., editors, *Int. Workshop on Human-Friendly Robotics 2020*. Springer.

Li, T., Yu, J., Qiu, Q., and Zhao, C. (2022). Hybrid uncalibrated visual servoing control of harvesting robots with RGB-D cameras. *IEEE Trans. on Industrial Electronics*.

Marchand, E., Spindler, F., and Chaumette, F. (2005). ViSP for visual servoing: A generic software platform with a wide class of robot control skills. *IEEE Robotics & Automation Magazine*, 12(4).

Mehta, S., MacKunis, W., and Burks, T. (2014). Nonlinear robust visual servo control for robotic citrus harvesting. *IFAC Proceedings Volumes*, 47(3):8110–8115.

Rusinkiewicz, S. (2019). A symmetric objective function for ICP. *ACM Trans. on Graphics*, 38(4).

Silwal, A., Yandun, F., Nellithimaru, A., Bates, T., and Kantor, G. (2022). Bumblebee: A path towards fully autonomous robotic vine pruning. *Field Robotics*, (2).

Yandun, F., Parhar, T., Silwal, A., Clifford, D., Yuan, Z., Levine, G., Yaroshenko, S., and Kantor, G. (2021). Reaching pruning locations in a vine using a deep reinforcement learning policy. In *IEEE Int. Conf. on Robotics and Automation (ICRA'2021)*, Xi'an, China.

You, A., Kolano, H., Parayil, N., Grimm, C., and Davidson, J. (2022). Precision fruit tree pruning using a learned hybrid vision/interaction controller. In *IEEE Int. Conf. on Robotics and Automation (ICRA'2022)*, Philadelphia, PA.

You, A., Sukkar, F., Fitch, R., Karkee, M., and Davidson, J. R. (2020). An efficient planning and control framework for pruning fruit trees. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 3930–3936.

Zahid, A., Mahmud, M. S., He, L., Heinemann, P., Choi, D., and Schupp, J. (2021). Technological advancements towards developing a robotic pruner for apple trees: A review. *Computers and Electronics in Agriculture*, 189:106383.

Zhang, S., Gong, Z., Tao, B., and Ding, H. (2020). A visual servoing method based on point cloud. In *IEEE Int. Conf. on Real-time Computing and Robotics (RCAR'2020)*.