2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)
June 28-30, 2023. Seattle, Washington, USA

# Low-cost, accurate robotic harvesting system for existing mushroom farms

Panagiotis Mavridis[1], Nikolaos Mavrikis[1], Athanasios Mastrogeorgiou[1,2], Panagiotis Chatzakos[3]

***Abstract*— This paper presents a robotic mushroom harvesting solution, consisting of an actuated scanning vision system integrated into a gantry robot. The system is capable of performing segmentation and pose estimation of the mushrooms on Dutch shelves commonly used in growing farms worldwide. The vision system employs an active stereo RGB-D camera able to capture a 360° scene of the mushroom bed, providing a high quality reconstruction of the mushroom caps. The YOLOv5 algorithm is used for the detection and size classification of the mushrooms, while a two-step model-fitting method is developed for the pose estimation task. The actuated carriage is compact, designed for operation in real mushroom-growing farms and intended to be used together with a soft gripper. The robot has five actuated degrees of freedom (DoFs), three for the linear motion on the shelves, and two DoFs for achieving the desired orientation for the gripper. In a real harvesting scenario, the robot sequentially scans the selected areas and accurately places the gripper in the appropriate angle of attack utilising our pose estimation method together with our visual servoing module for minor adjustments. The results were promising on all trials using 3D printed white button mushrooms on real soil.**

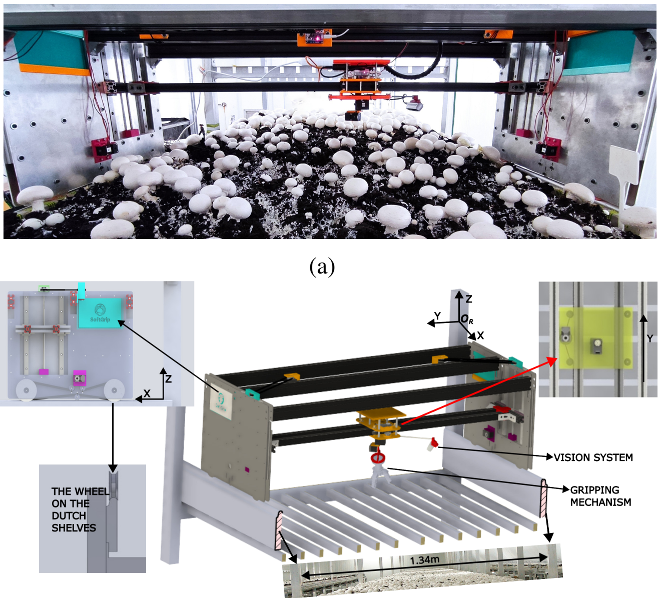
***Index Terms*— mushroom harvesting; agricultural robotics; computer vision; deep learning; visual servoing.**

## I. Introduction

According to [1], the mushroom industry is facing difficulties in maintaining profit margins due to the increasing cost of raw materials, and the application of outdated and inefficient cultivation methods. In such industries, automation is the key towards reducing labour costs, boosting production and improving working conditions [2]. Among the mushroom industry, the white button mushroom (Agaricus bisporus) is the most widely cultivated variety ($15-36\%$ of total market share) [3]. These mushrooms have a delicate body that can be easily damaged or blemished, leading to reduction in their value. As a result, a robotic solution capable of efficiently harvesting white button mushrooms, will bear tremendous economic benefits in the sector [4]. Such a system should be equipped with: (a) a vision system capable of scanning mushroom beds, estimating the size as well as the pose of the individual mushrooms and (b) a motion system capable of positioning a gripper in the correct angle of attack, for successful outrooting.

Several computer vision approaches for mushroom detection and segmentation have been implemented so far. The YOLOv5 [5] detection algorithm is used for real-time tracking of mushrooms on a conveyort belt in [6] as well as for growth status inspection in [7]. Concerning Agaricus bisporus mushrooms, a convolutional neural network (CNN) is utilised for mushroom size estimation [8], while in [9] a solution that detects overlapping mushrooms is proposed which applies a Canny operator, a Harris corner detection and the branch definition algorithm. Finally, [10] introduces a vision pipeline, that performs detection and pose estimation of mushrooms, using one RGB-D camera and ICP [11] registration. All mentioned approaches have not been integrated into an autonomous mushroom harvester, designed for existing mushroom farms [12]. In addition, besides the detection and localization tasks, the developed mushroom picking systems so far [13], [14], [15], [16] do not focus on the pose estimation of the mushrooms, which is an important step for the correct and successful outrooting.

In this work, we present a complete mushroom picking solution (Fig. 1a) consisting of: (a) an actuated stereo vision system for accurate mushroom detection, segmentation, classification & pose estimation which is orders of magnitude cheaper compared to laser scanning solutions [17] (b) an actuated gantry robot designed for existing mushroom farms [12], that is able to position a gripper in the correct angle of attack to successfully outroot a selected mushroom.



(a)

(b)

Fig. 1: (a) The robot on the Dutch shelves, (b) its CAD.

[1] Panagiotis Mavridis, Nikolaos Mavrikis & Athanasios Mastrogeorgiou are with TWI-Hellas, {panagiotis.mavridis, nikolaos.mavrikis, athanasios.mastrogeorgiou}@twi.gr, 280 Kifisias Ave., 152 32 Halandri, GR.

[2] Athanasios Mastrogeorgiou is with the Control Systems Lab, School of Mechanical Engineering, NTUA, GR. (email: amast@central.ntua.gr)

[3] Panagiotis Chatzakos is with the AI Innovation Centre, University of Essex, p.chatzakos@essex.ai, Wivenhoe Park, Colchester CO4 3SQ, UK.

978-1-6654-7633-1/23/$31.00 ©2023 IEEE

The rest of the paper is organised as follows: Firstly, the Cartesian robot is presented followed by the mushroom detection and pose estimation pipeline. Secondly, the harvesting procedure is described as well as the software developed to carry out the harvesting task. In turn, experimental results validate the accuracy of the proposed method while in the conclusion section the developed mechatronic solution is summarised and future research direction is proposed.

## II. Cartesian Robot Overview

The proposed robot design takes into account the $1.34m$ wide Dutch shelves commonly used in mushroom farms worldwide that are usually stacked one on top of the other [12]. Our robotic solution was designed to fit between two consecutive shelves and move along consecutive shelving bridges. As planned, a soft gripper will be integrated into the robot [18] suitable for delicate mushroom harvesting. Additionally, the robot is equipped with an actuated low-cost vision system consisting of one camera that can rotate and scan the mushroom bed, performing mushroom detection and pose estimation tasks.

The actuation system consists of eight motors: two for the motion along the X axis, two for the Z axis and one for the Y axis (coordinate system presented in Fig. 1b & 2). One motor is responsible for the rotational motion of the camera system while two motors are used for the picking mechanism. The robot moves along the X axis with the use of two identical actuated pulley systems. The reason for having two synchronised motors for the X axis motion is to avoid using a transmission shaft which could possibly interfere with the cameras field of view (FoV). Concerning the Z axis motion, two identical mechanisms consisting of a drive pulley/driven pulley system and a screw-nut mechanism are used, while for the motion along the Y axis a timing belt tensed around a single pulley is utilised. The Y axis motion is guided by V-wheels moving along T-slot aluminium extrusions. All mechanisms are presented in Fig. 1b.

A pair of $1254mm$ long, aluminium extrusions are used as guides for the Y axis motion. These parts can easily be replaced by bars of different lengths to change the width of the robot and make it compatible with a set of Dutch shelves other than the common one, i.e. $1.34m$. Currently, the robot is $1386mm$ wide to fit in the standard shelving bridge.

A major design task was the creation of wheels that would allow the robot to move across the Dutch shelves without losing track. The wheels are placed on the circular bump of the structural element of the Dutch shelves (Fig. 1b) and enclose it with a pair of flanges to maintain the robot's motion on track. The mechanisms used for the X and Z axis motion are placed between two aluminium sheets, which are $3mm$ thick, in a $2cm$ wide gap between them. The more compact the assembly, the bigger the area of the mushroom bed that can be utilized. The chassis of the robot consists of aluminium sheets and extrusions leading to a total mass of $25kg$. The low mass of the robot, as well as the use of polyurethane wheels and aluminium parts were deliberate choices that lead to sufficient corrosion and wear resistance.

The vision system is attached under the unit that moves along the Y axis (orange part in Fig. 1b). The camera's pitch angle is 60° (Fig. 2). This decision improved our initial approach presented in [19], and produced better reconstruction results of the mushroom caps as the camera had better view of the upper and the lateral side of the caps. During scanning, the camera's distance from the mushroom bed is $0.3m$. These values were selected, taking into account the camera's specifications [20] for optimal performance. During scanning, if the camera is close to one of the aluminium sheets, it cannot complete a full rotation for the scanning procedure. Therefore, the camera rotates 180°, when these areas (red in Fig. 2) are scanned. A vision system with two cameras would be twice as fast but not able to scan these areas (while rotating, one of the cameras would collide with the aluminium sheets of the robot's chassis).

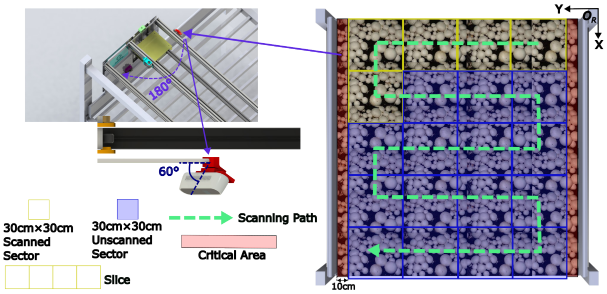


Fig. 2: The robot's scanning path and the rotational motion of the vision system.

Concerning the gripper, two actuated DoFs are needed so that all desired angles of attack can be reached. As a result, two motors are responsible for the yaw and pitch rotational motions of the picking mechanism. Given the coordinates $(x_m, y_m, z_m)$ of the cap and the yaw and pitch angles of the mushroom (Fig. 3), the (1-5) produce the desired position and orientation of the gripper.

$$yaw_{gripper} = \pi + \phi \tag{1}$$

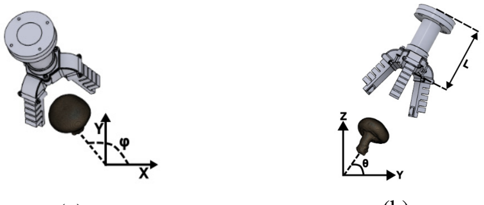
$$pitch_{gripper} = \pi + \theta \tag{2}$$

$$z_{gripper} = z_m + L \sin \theta \tag{3}$$

$$x_{gripper} = x_m + L \cos \theta \cos \phi \tag{4}$$

$$y_{gripper} = y_m + L \cos \theta \sin \phi \tag{5}$$

The $\phi$ and $\theta$ angles are the yaw and pitch angles of the



(a) (b)

Fig. 3: (a) Mushroom's yaw and (b) pitch angle.

mushroom respectively while $L$ is the distance between the last joint of the picking mechanism and the palm of the gripper (Fig. 3).

## III. Mushroom Detection and Pose Estimation

### A. Scanning and 3D Reconstruction

For the scanning procedure, the bed is initially divided into slices of $30cm \times 134cm$ while each slice is divided into scanning sectors of $30cm \times 30cm$ (Fig. 2). The robot moves to the center of each sector and performs a scan using the actuated vision system. Specifically, it captures 18 snapshots of the scene. Each snapshot consists of an aligned pair of RGB and Depth Map images. Between each snapshot, the camera system rotates 20° (Fig. 4). At the end of the scanning of the sector, the position of the camera is turned 340° relative to its starting position. Finally, the vision system rotates back to the initial position, ready to scan the next sector.

The RGB-D snapshots from the scans are converted to point clouds that are stitched together. The successive point clouds must have an overlapping area, so that the registration algorithm (ICP) can tightly align them [21]. Our reconstruction method iterates and stitches two consecutive point clouds using the ICP Point-to-Plane algorithm. ICP needs an estimation of the initial transformation between the point clouds to perform a tight alignment, which can be calculated using Global Registration algorithms. In our case, the initial transformation between two successive point clouds is known and its 20° respective to the center of rotation of the vision system. The result is a colorized point cloud with a complete 3D reconstruction of the mushrooms in that sector.

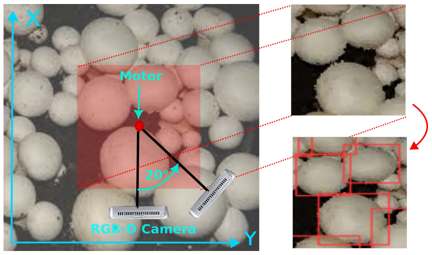


Fig. 4: The scanning, 3D reconstruction and detection procedure of the actuated vision system.

### B. Detection, Localisation and Size Classification

A mushroom detection and localisation procedure is developed, based on YOLOv5m [5] algorithm, to retrieve the position of the mushrooms and classify them in four size categories. The training dataset for YOLOv5m consists of 500 overhead images of 3D printed mushrooms, placed on real soil. The training procedure followed is described in detail in [22].

The source image for the YOLOv5 detection method must have an overhead position perspective for the detection results to be accurate (mushroom cap size estimation, etc.). However, adding one camera to the vision system was deemed redundant because the required overhead RGB image can be extracted from the colourised 3D reconstruction by using a virtual camera in the Open3D [23] environment. The virtual camera is configured to have the same intrinsic parameters as the scanning camera. Its position in the virtual environment is on the center of rotation of the vision system and its orientation is perpendicular to the reconstructed scene. The overhead RGB image from the virtual camera is used as the source for the YOLOv5m detection model. The result is a bounding box for each detected mushroom. The coordinates of the center of the bounding box $(x_p, y_p)$, are used to extract the $(x_m, y_m, z_m)$ coordinates of the detected mushroom from the reconstructed point cloud in the corresponding pixels. To avoid cases where the point cloud has no information in this point, e.g.: due to noise associated with the RGB-D camera, we extract a $3x3$ array of pixels around $(x_p, y_p)$. In turn, we retrieve the corresponding point cloud values and calculate the average value of the Z coordinate. The resulting 3D coordinates of the target mushroom $(x_m, y_m, z_m)$ are transformed to the gripper's coordinates using (3-5).

During the harvesting procedure, the mushrooms have to be grouped in size categories, e.g. $40mm$, $50mm$, $60mm$ and $90mm$ in diameter. Our approach performs classification to these four categories while this information can later be used in the harvesting plan.

### C. Orientation Estimation

After extracting the 3D location of the target mushroom, our method estimates its orientation using a two-step model-fitting approach. A dataset of mushroom models has been created offline by scanning 3D-printed mushrooms tilted in predefined orientations around the Z and X-axis (see Fig. 5). This results in 33 scans, i.e.: 8 scans from 0° to 360° with a 45° step around the Z-axis multiplied by 4 for the X-axis rotations (10° to 40° with a step of 10°) plus one model that is not tilted. The normal vector of the mushroom cap of each model is stored, deriving from its orientation. The point clouds of the models are converted and stored as K-D Trees, structures optimised for nearest neighbour search [24].

During runtime, the point cloud of each detected mushroom is segmented using the predicted bounding box. Next, the normal vector of the sample mushroom point cloud has to be computed. Based on experiments, it is safely concluded that the small curvature of the top part of the mushroom cap can be represented by a plane surface. By employing the RANSAC algorithm we segment the plane with the largest support (e.g. $ax + by + cz + d = 0$) on the sample mushroom point cloud and compute its perpendicular vector $\vec{s} = (a, b, c)$.

Initially, the angle difference between the normal vector of the sample and the normal vector of each model, is calculated. Specifically, the angle $\theta_i$ between the normal vector of the sample $\vec{s}$ and the stored normal vector of the $i^{th}$ model $\vec{m}_i$ is given by (6).

$$\theta_i = \langle \vec{s}, \vec{m}_i \rangle = \arccos\left(\frac{\vec{s} \cdot \vec{m}_i}{\|\vec{s}\|\|\vec{m}_i\|}\right) \quad (6)$$

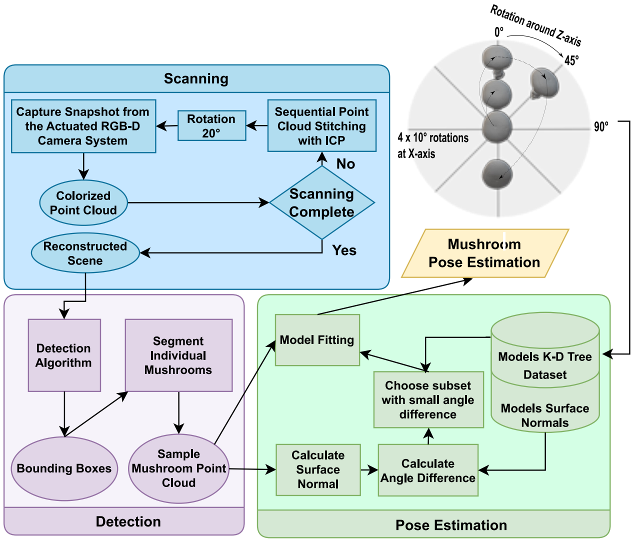


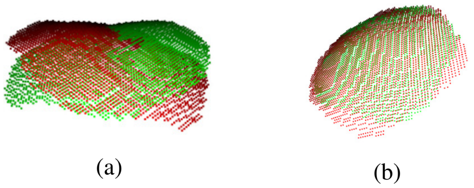
Fig. 5: Pose estimation pipeline.

If the difference $\theta_i$ exceeds a predefined threshold, those models are excluded from the next model-fitting comparison. This leads to an initial orientation estimation of the sample mushroom and avoids unnecessary comparisons with every model in the dataset.

Secondly, we initiate a nearest neighbour search between every point $p_s$ in the sample point cloud and the remaining models in the dataset. The point clouds of the models have been converted to K-D Trees to achieve fast nearest neighbour search. The search is hybrid, meaning that the criteria are: 1) the distance within a radius from the query point $p_s$ and 2) the number of neighbours within this radius. Specifically, the search algorithm is developed to check if for every $p_s$ in the sample there is at least one point from the model point cloud, within a specific radius of $2mm$.

Finally, a fitness score is calculated between 0.0 and 1.0, indicating how ”tight” is the overlap between a model and the sample (Fig. 6). The fitness is calculated using (7) where the $no\,of\,correspondences$ indicates the number of points in the model that satisfy the search criteria.

$$fitness = \frac{no\,of\,correspondences}{no\,of\,points\,in\,sample} \tag{7}$$


(a)

(b)

Fig. 6: (a) Bad fit and (b) Good fit between the sample (green) and a model (red).

The candidate model with the highest fitness score corresponds to the orientation of the sample mushroom. The pose estimation pipeline is summarised in Fig. 5.

### D. Visual Servoing

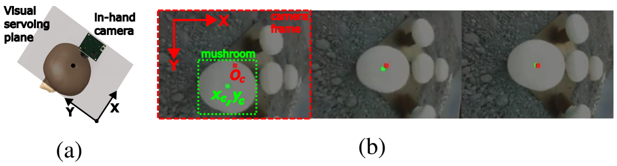
In most cases, the centre of the bounding box from the detection phase does not equal the centre of the corresponding mushroom cap, due to the mushroom's various possible orientations and the overhead perspective of the source RGB image. Therefore, after the initial approach to the mushroom with inverse kinematics, a visual servoing procedure takes place, aiding the approach to the mushroom's centre with higher precision, using a micro-camera that is intended to be embedded in the palm of the soft gripper.

When the gripper is positioned close to the target mushroom, the YOLOv5m detection module starts receiving the camera's feedback in real-time and calculates the bounding box of the target mushroom. The end effector's yaw and pitch angles are given by the orientation estimation procedure and (1-2). The robot is controlled to move on the plane that is tangential to the top of the mushroom model with the highest fitness score (Fig. 7a) and to perform visual servoing so that the center of the bounding box $(x_c, y_c)$ and the centre of the camera frame $O_c$ coincide (Fig. 7b). Utilising the in-hand camera and the trained YOLOv5m mushroom detector, the pixel error ($e_{pxx}$, $e_{pxy}$) between the centre of the bounding box of the target mushroom and the centre of the camera frame, is calculated. Two P controllers with gains $K_{px}$, $K_{py}$ are employed to move the robot on the $O_c$ plane and minimize ($e_{pxx}$, $e_{pxy}$). The motion on the visual servoing plane (Fig. 7a) is achieved by translating the pixel error to displacement in X, Y and Z axes, using (8-10).

$$\Delta z = K_{px} e_{pxx} \cos\theta \tag{8}$$

$$\Delta x = -K_{px} e_{pxx} \sin\theta \cos\phi - K_{py} e_{pyy} \sin\phi \tag{9}$$

$$\Delta y = K_{px} e_{pxx} \sin\theta \sin\phi - K_{py} e_{pyy} \cos\phi \tag{10}$$



(a)

(b)

Fig. 7: (a) The visual servoing plane and (b) the visual servoing progress as the controller aligns the centre of the bounding box of the detected mushroom (green) with the centre of the camera frame (red).

## IV. Harvesting Procedure

The developed robot is capable of performing the harvesting procedure presented in Fig. 8, consisting of four phases: *Initialisation*, *Scanning*, *Detection* and *Harvesting*. During the *Initialisation* phase the robot performs homing. In turn, the mushroom bed is divided to slices and scanning sectors (Fig. 2). During the *Scanning* phase the robot moves to the first slice and iteratively scans its sectors until the whole area of the slice is covered. For the *Detection* phase, the reconstructed images of the previous step are used to estimate the pose and mushroom size group in the slice. This results in a dataset of mushroom targets, containing the following information for each one: 1) 3D Position, 2) Orientation, 3) Size group. The mushrooms are sorted according to their Z coordinate. During the *Harvesting* phase, the harvesting plan combines the information of mushroom height and size in order to pick first the mushrooms that are bigger and in the upper layer, making the outrooting of the mushrooms in the lower layers easier. The proposed scanning and harvesting path of a mushroom bed are shown in Fig. 2.

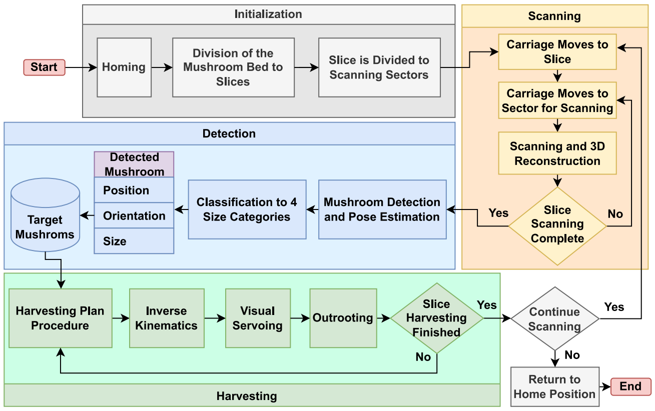

Initialization
Start
Homing
Division of the Mushroom Bed to Slices
Slice is Divided to Scanning Sectors
Scanning
Carriage Moves to Slice
Carriage Moves to Sector for Scanning
Scanning and 3D Reconstruction
Slice Scanning Complete
Yes
No
Detection
Detected Mushroom
Position
Orientation
Size
Target Mushrooms
Classification to 4 Size Categories
Mushroom Detection and Pose Estimation
Harvesting Plan Procedure
Inverse Kinematics
Visual Servoing
Outrooting
Slice Harvesting Finished
Yes
No
Continue Scanning
Yes
No
Return to Home Position
End
Harvesting

Fig. 8: The proposed harvesting procedure.

## V. Software Architecture

The computer vision software modules and the carriage control software modules are hosted on a PC running Ubuntu 20.04 LTS (Focal Fossa) and ROS Noetic [25]. The reconstruction and pose estimation procedures were developed using Open3D [23] and ROS.

Separate ROS nodes have been developed for the scanning and reconstruction procedures. A */camera_node* is used for controlling the RealSense D435i scanning camera, configured to produce images with a resolution of $848x480$ at $30FPS$. The */scanning_node* rotates the vision system and receives the RGB-D snapshots from the camera. The */reconstruction_node* is responsible for: (a) receiving the RGB-D snapshots, (b) converting them to point clouds and (c) performing the point cloud stitching. The point cloud stitching procedure runs in parallel with the scanning, and it takes $\sim 19secs$ for the 3D reconstruction of a whole sector. The */detection_node* receives the reconstructed scene, performs mushroom detection and produces the 3D localisation results. Finally, the */orientation_estimation_node*, receives the reconstructed scene and the bounding boxes of the detected mushrooms and estimates their orientation.

Regarding the software for the robot's motion, low-level */position_control* and */velocity_control* nodes have been developed that use the Dynamixel C++ API to communicate with the motors. The developed nodes are adjusted for velocity control along the X,Y and Z axes, while the actuators of the gripper and the vision system are configured for position control. Additionally, a */carriage_controller_node* is developed, which encapsulates a ROS action server that receives the pose of a mushroom target. By utilising the inverse kinematics (1-5) and the in-hand camera feedback the end effector is aligned with the centre of the target mushroom.

## VI. Experimental Results

### A. Experimental Setup

All the experiments were conducted using a mushroom bridge provided by [12] that is identical to those used in the mushroom growing farms. An end effector was 3D printed and a micro-camera was attached to it. The camera was identical to the one destined for in-hand use in the gripper proposed in [18], (Fig. 9). The total length of the end-effector (3D printed part and camera) equals the length $L$ of the soft gripper. The camera provides visual feedback during the visual servoing procedure to control the end effector towards the correct position for outrooting.

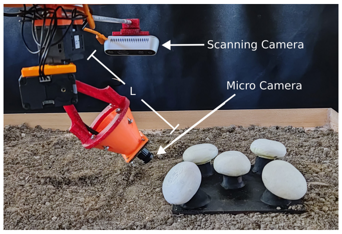


Fig. 9: The experimental setup.

### B. Orientation Estimation Experiments

Experiments for the vision system were conducted, to evaluate the proposed orientation estimation method. The mushrooms were placed on a 3D printed test base (black base in Fig. 9) and the developed pipeline calculated their orientation. The estimation was compared to the actual orientation of the sample mushroom which could be calculated manually by the user. The results for 20 trials are presented in Table I. The error was $\sim 5°$ around the X-axis and the execution time for the orientation estimation of a sample mushroom was $\sim 0.1s$.

TABLE I: The average error, execution time and number of comparisons of our orientation estimation method.

| Samples | Error Avg | Error Std Dev | Execution Time Avg | No of Compared Models Avg |
|---|---|---|---|---|
| 20 | $5.86°$ | $2.35°$ | $0.1176s$ | 8.7 |

### C. Visual Servoing Experiments

The accuracy of the visual servoing module was experimentally evaluated. Initially, the pose of a sample mushroom in Fig. 9 was estimated using the pipeline presented in Fig. 5 and the end-effector approached it using (1 - 5). In turn, the visual servoing started in order to align the in-hand micro-camera and the target mushroom. The camera frames $(397 \times 286)$ are passed to the YOLOv5m detection model that produced the bounding box of the mushroom at $30FPS$. The distance in pixels between $(x_c, y_c)$ and $O_c$ (Fig. 7b), on the visual servoing plane (Fig. 7a), was calculated online. The procedure stopped when the distance in pixels between $(x_c, y_c)$ and $O_c$ was equal or bellow $11px$ taking into account the adaptability of the soft gripper to objects. Fig. 10, presents the visual servoing results on the sample mushroom. The desired and actual position, of the linear motions of the robot in global coordinates $O_R$ is shown in Fig. 10 (column 1) while in Fig. 10 (column 2) is depicted the pixel error on the visual servoing plane which reaches the applied threshold.

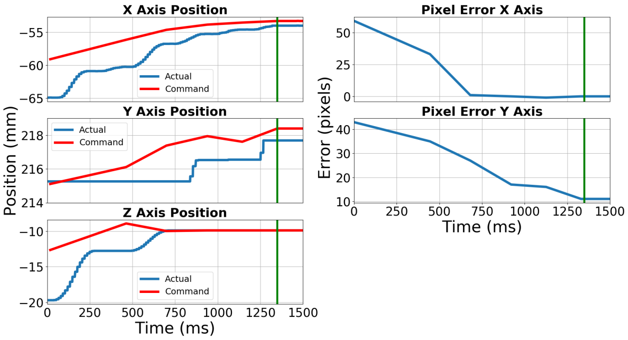


Fig. 10: The desired and actual position of the robot (left) and the errors in pixels (right), during the visual servoing process. The threshold is reached at $t = 1,35s$ (green line).

## VII. CONCLUSIONS

In this work, a low-cost autonomous mushroom harvesting system is presented. The robot is designed to work with a soft gripper, suitable for the delicate harvesting of white button mushrooms. The developed system is compact so that the chassis does not interfere with the crop, can be easily integrated in existing mushroom growing farms and has been tested in common, $1.34m$ wide, Dutch shelves. Concerning the low-cost vision system, accurate detection and pose estimation of the mushrooms is achieved by employing one active stereo RGB-D camera that can rotate 360°. The results were promising, the error was $\sim 5°$ which is easily handled by a soft gripper taking into account its robust adaptability to objects. Lastly, utilizing visual servoing and an in-hand micro camera, the gripper is guided to the optimal angle of attack. As soon as the soft gripper is ready for experiments, the robot's harvesting performance on real mushrooms (mushrooms picked per minute) will be evaluated.

## ACKNOWLEDGMENT

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N° 101017054 (SoftGrip). The authors would like to thank Irene Kalesi for her valuable contribution to this work.

## REFERENCES

[1] "Mushroom growers facing 'massive crisis'." https://www.independent.ie/. Accessed: 2022-04-15.

[2] R. Masey, J. Gray, T. Dodd, and D. Caldwell, "Guidelines for the design of low-cost robots for the food industry," *Industrial Robot: An International Journal*, vol. 37, pp. 509–517, 10 2010.

[3] D. Royse, J. Baars, and Q. Tan, *Current Overview of Mushroom Production in the World: Technology and Applications*, pp. 5–13. wiley, 08 2017.

[4] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey," *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 688–716, 2018.

[5] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, K. Michael, TaoXie, J. Fang, imyhxy, Lorna, Z. Yifu, C. Wong, A. V, D. Montes, Z. Wang, C. Fati, J. Nadar, Laughing, UnglvKitDe, V. Sonck, tkianai, yxNONG, P. Skalski, A. Hogan, D. Nair, M. Strobel, and M. Jain, "ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation," Nov. 2022.

[6] A. Lin, Y. Liu, and L. Zhang, "Mushroom detection and positioning method based on neural network," in *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, vol. 5, pp. 1174–1178, 2021.

[7] B. Wei, Y. Zhang, Y. Pu, Y. Sun, S. Zhang, H. Lin, C. Zeng, Y. Zhao, K. Wang, and Z. Chen, "Recursive-yolov5 network for edible mushroom detection in scenes with vertical stick placement," *IEEE Access*, vol. 10, pp. 40093–40108, 2022.

[8] C.-P. Lu and J.-J. Liaw, "A novel image measurement algorithm for common mushroom caps based on convolutional neural network," *Computers and Electronics in Agriculture*, vol. 171, p. 105336, 2020.

[9] S. Yang, B. Ni, W. Du, and T. Yu, "Research on an improved segmentation recognition algorithm of overlapping agaricus bisporus," *Sensors*, vol. 22, p. 3946, May 2022.

[10] N. L. Baisa and B. Al-Diri, "Mushrooms detection, localization and 3d pose estimation using rgb-d sensor for robotic-picking applications," 2022.

[11] S. M. Rusinkiewicz and M. Levoy, "Efficient variants of the icp algorithm," *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pp. 145–152, 2001.

[12] "Teagasc — agriculture and food development authority." https://www.teagasc.ie/crops/horticulture/mushrooms/. Accessed: 2022-04-15.

[13] J. Rong, P. Wang, Q. Yang, and F. Huang, "A field-tested harvesting robot for oyster mushroom in greenhouse," *Agronomy*, vol. 11, p. 1210, 06 2021.

[14] X. Hu, Z. Pan, and S. Lv, "Picking path optimization of agaricus bisporus picking robot," *Mathematical Problems in Engineering*, vol. 2019, pp. 1–16, 09 2019.

[15] B. Jia, S. Yang, and T. Yu, "Research on three picking arm avoidance algorithms for agaricus mushroom picking robot," in *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications( AEECA)*, pp. 325–328, 2020.

[16] S. Yang, B. Jia, T. Yu, and J. Yuan, "Research on multiobjective optimization algorithm for cooperative harvesting trajectory optimization of an intelligent multiarm straw-rotting fungus harvesting robot," *Agriculture*, vol. 12, p. 986, 07 2022.

[17] "Mycionics – intelligent mushroom harvesting." https://mycionics.com/. Accessed: 2023-01-01.

[18] "Softgrip." https://cordis.europa.eu/project/id/101017054/results. Accessed: 2023-01-01.

[19] P. Mavridis, A. S. Mastrogeorgiou, and P. Chatzakos, "Mushroom detection and pose estimation using instance segmentation and model fitting," in *2022 ICRA Workshop on Agricultural Robotics and Automation*.

[20] "Depth camera d435i." https://www.intelrealsense.com/depth-camera-d435i/. Accessed: 2023-01-01.

[21] J. Sánchez-Pérez, E. Torres, M. Jimenez, and N. Santiago, "Practical considerations for implementing icp registration algorithms in a 360o 3d point cloud stitching system," 01 2018.

[22] "Train custom data." https://github.com/ultralytics/yolov5/wiki/Train-Custom-Data. Accessed: 2023-01-01.

[23] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.

[24] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, p. 509–517, sep 1975.

[25] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng, "Ros: an open-source robot operating system," vol. 3, 01 2009.