Cooperative pepper picking: a Case Study on Yield Estimation in Indoor Agriculture

Marsela Polic, Antun Ivanovic, Matko Orsag [*†]

Abstract

Our case study focuses on indoor robotic farming. Inspired by recent promising results in sim-to-real transfer we built a realistic simulation environment combining a ROS-compatible physics simulator (Gazebo) with a realistic rendering cycles engine from Blender. Without loss of generality, we focus on a sweet pepper harvesting task and showcase and analyze the technological pipeline necessary to conduct such a mission. The pipeline starts from aerial robotics control and trajectory planning, combined with deep learning-based pepper detection, a clustering approach for yield estimation, and mission planning for harvesting using a heterogeneous team of robots.

Introduction

Today, as we witness obvious consequences of global warming, indoor farming is becoming an important tool to mitigate the problems farmers face with unpredictable and often extreme weather patterns. Traditional indoor agriculture enables farmers to provide their crops with optimal weather conditions, but outdoor impact, predominantly sunlight, still plays an important role in the plants' growth. Modern indoor farming completely eliminates the outside weather conditions, emulating optimal weather conditions controlling both the climate and the sunlight. While climate control can be considered as a solved problem, replacing manual labor is an active research topic.

From the robotics point of view, indoor farming provides a level of structure in the environment which is an important step towards fully autonomous operation.Farming industry is extremely labor intensive, and the job requirements often fit the category of dangerous, dull, and dirty, making them ideal for automation. Labor is even more important when considering organic farming. To reduce the use of pesticides, organic agriculture requires a lot more manual care, with a comparably smaller agricultural output. The obvious economical consequence of such a production system is a higher cost of organic food. Again, turning to robotics and automation can help alleviate the costs.

Deployment of robots on big farms is not a new concept, but rather a fast-growing industry that focuses on big machines tailored for specific crops and use cases. This approach is profitable only on a large-scale production system. For example, harvesting robots, even in indoor farms, are designed as mobile manipulators. Such an approach requires an extremely sophisticated framework capable of precise navigation towards the plant and manipulation in varying conditions. On the other hand, in automotive industry, which is a golden standard for robotized production, a product is guided towards the robot workspace, and not the other way around. In this paper we present a case study deploying a heterogeneous team of robots working together harvesting peppers.

The heterogeneous team of robots consists of an aerial inspection robot, mobile robots that carry plants around the farm, and a robotic manipulator which treats the plants. We focus on the harvesting problem and present a pipeline of technologies used to execute such a task, however the same pipeline can be applied to other problem as well. We provide the details of the simulation environment used to simulate the complete missions. Finally we provide the results of simulation analysis using a realistic scenario of sweet pepper harvesting.

Problem description

The setup considered in this use case analysis consists of a heterogeneous team of robots: a static robot manipulator positioned at its workstation, and mobile ground robots. They rely on a UAV for surveillance and data collection. The surveillance aerial manipulator is a UAV with eye-in-hand RGB-D camera on the manipulator's end effector. The UAV scans the greenhouse regularly, providing the control system with the recordings of the current state of the crop. Here, the UAV recordings are used for yield estimation, i.e. fruit counting for harvesting mission planning. Pepper detection and counting is realized using deep and unsupervised learning methods on RGB images and organized pointclouds provided by the UAV mounted RGB-D camera.

For the multi-robot system, two workstations are envisioned on the sides of the manipulator, enough for two mobile robots to operate. The UGVs carry plants to the work-

[*]Authors are with Faculty of Electrical Engineering and Computing, University of Zagreb [marsela.polic, antun.invanovic, matko.orsag] @fer.hr

[†]This work has been supported by Croatian Science Foundation under the project Specularia UIP-2017-05-4042

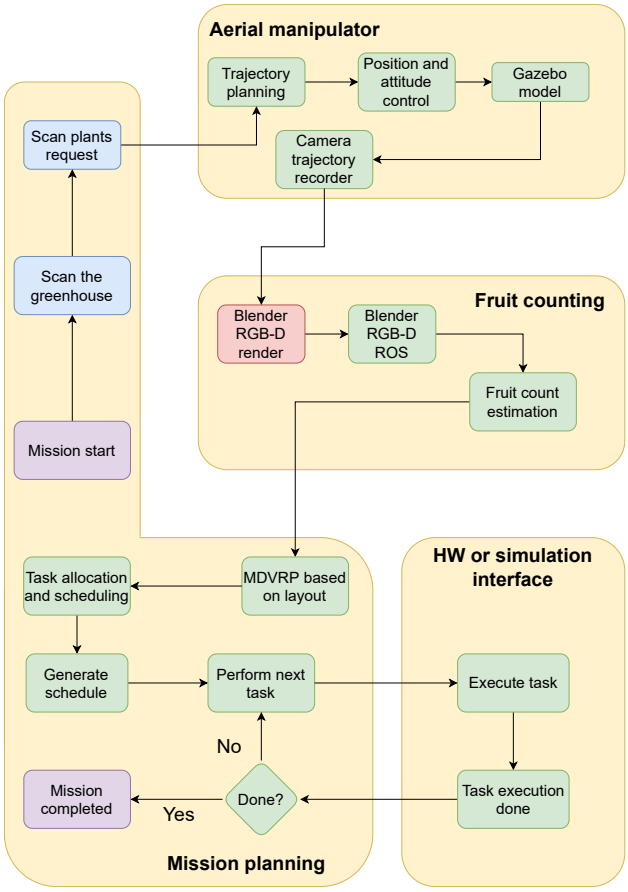

Figure 1: The functional diagram representing the system pipeline. At the mission start, a request for scanning the plants is generated, upon which the aerial manipulator executes a scan trajectory in the greenhouse. The gathered camera position and orientation is used through Blender to generate a realistic dataset for fruit counting. The mission then starts executing tasks and reports when all tasks are completed.

stations of the static robot manipulator arm. Since roughly one half of the plant is reachable by the manipulator, some activities (including harvesting) require cooperation during plant treatment procedure, in that the UGV rotates the plant so that the other half of the plant can be treated as well.

Simulation setup

Gazebo was used as a state of the art robotic simulation environment, thanks to the simple and straight-forward interface for control through the ROS. However, due to limitations of visual and depth sensor modeling in Gazebo environment, and the complexity of modeling a world containing a large number of detailed plant models, the simulation pipeline was extended with Blender modeling and simulation. The complete pipeline is described with a schematic in Fig. 1, where ROS and Gazebo related modules are colored green, and the Blender part is in red.

ROS/Gazebo As can be seen in the schematic in Fig. 1, most of the software solutions are implemented within ROS environment, controlling the Gazebo robotic setup. This organisation enables simple transfer of the software modules to the real hardware when deployment is ready in the real world. The aerial manipulator model in Gazebo performs a scanning trajectory, with the end-effector tip position simulating camera motion. Based on the camera information, fruit yield is estimated per plant, and provided to the mission planner. The mission planner generates a schedule for the robot(s) involved in the greenhouse cultivation. The speed of mission execution is a function of available resources and the setup.

Blender The model of the greenhouse in Blender consists of rows of tables carrying pepper growth containers. Each row consists of 4 tables carrying a single pepper plant. The plants are modeled using the basic building blocks developed for synthetic dataset generation, namely realistic pepper models, planar leaf models with realistic textures, flower models, plant stem, and a plastic pot. The plants are generated procedurally with a random particle generator, varying pepper, flower and leaf number, position, and orientation with respect to the plant stem. More info on the setup we use to train the AI to detect peppers can be found in (Polić, Tabak, and Orsag 2021), where we successfully demonstrated pepper detection method trained using synthetic and actual pepper images.

Perhaps the most important part of the Blender model utilized in this paper is the animated camera system, simulating an RGB-D sensor for the Gazebo robot simulation. The simulation pipeline links the motion of the end effector of the aerial manipulator, with an eye-in-hand RGB-D camera, to camera animation in the Blender environment model. The RGB image of the scene is rendered using built-in Blender rendering machines. The depth image is generated from the depth map calculated within Blender cycles rendering engine. During trajectory execution, the camera viewpoints are rendered and output into a folder, along with the generated depth images and corresponding camera positions. This data is processed offline in ROS, transforming the recorded depth image into an organized pointcloud. The pointcloud is published along with the corresponding RGB image and global camera transformation. This format corresponds to the output of a real RGB-D camera, and can be used used by the detection and counting software package in ROS. The detection and counting is implemented in Python, relying on the TensorFlow and Pythons Sklearn module.

Enabling technologies

In this paragraph we outline the key technologies developed to succesfully deploy the robots on a sweet pepper harvesting task.

Aerial manipulator motion planning

To estimate the fruit number in the greenhouse, it is necessary to scan and inspect plants. In this paper, an aerial manipulator with an RGB-D camera mounted on the end-effector is considered for the task. Since the layout of the

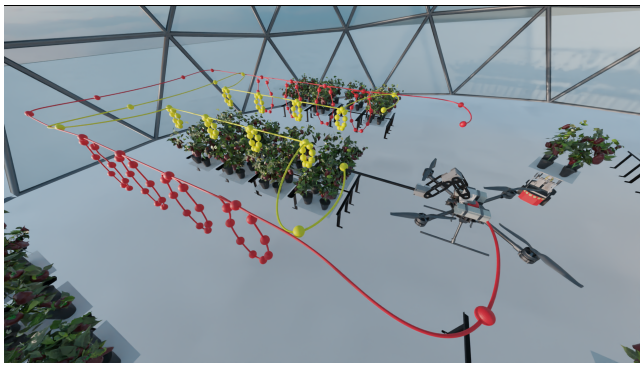

Figure 2: An example of the planned elliptical trajectory used to scan a single row of the structured greenhouse. Red lines and spheres denote the trajectory and waypoints of the UAV body, while the yellow color denotes the end-effector trajectory.

greenhouse is a-priori known, it is possible to scan a single plant container unit or a whole structured row. The gathered data is then used in the fruit counting algorithm.

Scan waypoints Each plant in a greenhouse row has to be scanned from multiple angles to obtain informative data for fruit counting. An example of planned waypoints and trajectory for a single row is depicted in Fig. 2. In short, the aerial manipulator moves around the row and scans each plant with an elliptical trajectory.

Precisely scanning each plant requires carefully planned end-effector motion, which mostly relies on the dimensions of growth containers. The elliptical shape of the waypoints is based on the plant position and dimensions. At each waypoint of the ellipse, the yaw of the UAV directs the end-effector towards the plant's centroid, pointing the end-effector upwards or downwards to get a better overview of the plant.

One row of the greenhouse consists of multiple plants. To perform a scan of a single row, it is necessary to scan each plant from both sides of the row. Therefore, an elliptical set of waypoints is planned for each plant, based on dimensions and number of plants in the row. Additional waypoints for navigating around the row are also included to yield a smooth trajectory.

Planned waypoints are considered to form a path of n waypoints:

$$\mathscr{P} = \left\{ \mathbf{p}_i \mid \mathbf{p}_i \in \mathbb{R}^{4+M}, i \in (0, 1, \ldots, n) \right\}, \qquad (1)$$

where $\mathbf{p}_i = \begin{bmatrix} x & y & z & \psi & \mathbf{q}_{\mathrm{M}}^T \end{bmatrix}^T$ is a single waypoint containing position and orientation of the UAV, as well as joint positions of the M dimensional manipulator.

Scan trajectory Based on the generated path from equation (1), a trajectory that respects dynamic constraints of the system is planned. Namely, the Time Optimal Path Parametrization by Reachability Analysis (TOPP-RA) algorithm (Pham and Pham 2018) that operates on the numerical integration approach is employed. As input the algorithm requires a set of positions with velocity and acceleration constraints for each degree of freedom. The output is a smooth trajectory $\mathscr{T}$:

$$\mathscr{T} = \left\{ \mathbf{t}(t) \mid \mathbf{t}(t) \in \mathbb{R}^{3(4+M)}, t \in (0, t_{end}) \right\}, \qquad (2)$$

where $\mathbf{t}(t) = \begin{bmatrix} \mathbf{p}^T & \dot{\mathbf{p}}^T & \ddot{\mathbf{p}}^T \end{bmatrix}^T$ denotes a trajectory point containing position, velocity and acceleration of each degree of freedom, and t_{end} denotes the duration of the trajectory.

Fruit counting

An important enabler of robotic agriculture is just in time detection, used during all plant hygiene operations. As most state of the art solutions, in this work, we rely on a commercial RGB-D camera (Fu et al. 2020). The red peppers are detected in 2D RGB images using a deep learning model. A MobileNet based Single Shot Detector(SSD), pretrained on the COCO dataset, is trained for object detection task in 2D RGB images. For the purposes of network training, a synthetic dataset is generated procedurally in Blender, in order to mitigate the cost of labeling a large training dataset (Hinterstoisser et al. 2019; Khan et al. 2019). Synthetic dataset generation has recently found applications in agriculture for various crops and cultures (Di Cicco et al. 2017; Olatunji et al. 2020; Zhang, Wu, and Chen 2021), including a synthetic dataset for the *C. annuum* semantic segmentation tasks (Barth et al. 2018). The transfer learning for the network first conducted on the synthetic dataset is followed by additional fine tuning on a small dataset of real, manually labeled images. The 2D detection pipeline produces bounding boxes that denote positions of detected peppers in the image, combining both detection and depth information of the RGB-D cameras. Various methods have been developed over the recent years for 3D pose estimation, such as surface normal estimation for grasp position optimisation (Lehnert et al. 2017), and peduncle model fitting in harvesting (Sa et al. 2017). In geometric model fitting methods, the detected fruit is modelled with geometric primitives such as cylinders and ellipsoids (Lehnert et al. 2016).

In this work, such high precision is not necessary, hence approximate 3D positions of the detected peppers are obtained applying the 2D detection bounding boxes on the organised pointcloud output from the camera depth channel. From among the filtered points representing a single detected pepper, a centroid point can be chosen as a reasonable approximation of the pepper position. It should be noted that this way, a point on the peppers' surface is chosen, resulting in varying position estimates, for the same pepper, depending on the camera perspective with respect to the pepper.

The harvesting mission is planned based on the ripe fruit count. The aerial manipulator executes a trajectory that enables recording the plants from multiple perspectives. Most of the fruit, unless heavily occluded, is recorded, and detected, from several perspectives during UAV motion. As stated, these detections do not match perfectly, since the positions of the pepper surface are not at the same global position. A counting method based on unsupervised learning is devised, that provides an estimate of the yield.

Upon trajectory execution, a set of detections is collected. In these detections, subsets of data are separated for each greenhouse row, using known layout of the greenhouse. Furthermore, with a predefined UAV trajectory, a known camera frame rate, and an estimate of the possible pepper yield, random subsampling is conducted on the detection dataset, filtering out most of false positive detections. For an expected yield of up to 10 peppers uniformly distributed across the plant body, this filtering step retains approximately 5-10 detections of each fruit. The remaining detections are augmented using random noise, to a fixed size set that enables unsupervised learning methods to properly separate the search space. We have found that 500 points are sufficient to properly separate the data, ranging in fruit count to up to 40 peppers per row (10 fruit per plant), with detection variation at centimeter level, and spread over approx. $2 \times 0.5 \times 0.5$m. From among the 500 points, the remaining false positive detections (e.g. pot, table, and other detected outside the plant bounding volumes) are filtered using known greenhouse layout. In case the greenhouse layout is not a-priori known, outlier removal methods can be used for the false positive filtering (Breunig et al. 2000).

On the augmented dataset of detections, the OPTICS algorithm is deployed as a clustering method (Ankerst et al. 1999) that separates the dataset by defining core points. These core points, i.e. cluster centres, are considered as peppers. The inputs to the algorithm are maximum distance ϵ, and minimum cluster size $MinPts$. The parameter ϵ represents the maximum distance of the cluster points to the core point, in order to be considered cluster members. In our case, this is set to 4 cm. The second parameter, $MinPts$, is a requirement on the cluster size, i.e. on the minimum number of detections to be considered a pepper. Empirical results in manipulator harvesting experiments showed that at least 2 detections from various viewpoints were needed to estimate the pepper position reliably. Due to the dataset augmentation to the fixed size of 500 points, the minimum number of detections is a function of the initial detections dataset size n_{init}, i.e. the original requirement of 2 detections increases by the factor of $\lceil 500/n_{init} \rceil$. The method clusters the points satisfying the provided conditions, and leaves the remaining points undefined.

Thanks to the organised structure of the greenhouse, the positions of pepper growth containers (pots) are known (or can be known if a similar detection was deployed for pot detection). Then, the detected peppers are counted in the two sub-spaces reachable by the robot manipulator from either side of the growth table/manipulation desk. This information is stored in an organised form of a yaml file, that is then interpreted by the mission planner.

Simulation results and conclusion

The greenhouse layout we used for simulation is shown in Fig. 4. The structure consists of eight tables, each with four pepper plant containers. The stationary manipulator workstation is located in the center of the structure. For this layout, we ran several simulations with a random number of peppers per plant. We tested the entire proposed pipeline generating the UAV trajectory for greenhouse inspection, counting the fruits from the video rendered in Blender, and planning the mission based on the inputs from the pepper counting.

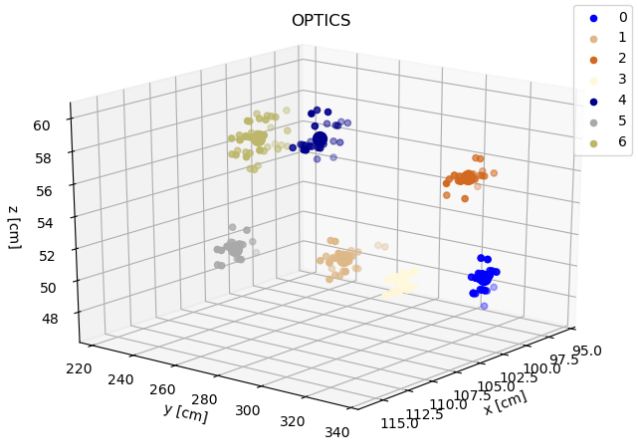

Figure 3: Detected peppers shown in different colors, along with the detections that determined the cluster core points. The detections are generated with a Gaussian random noise around the smaller set of actual detections, until a dataset of fixed size 500 is generated. The undefined points in the dataset are not shown for clarity.

Figure 4: The layout of the proposed greenhouse structure consisting of eight tables, each with four pepper plant containers. The stationary manipulator workstation is located in the center of the structure.

To test the planner with a random number of fruits per pepper plant, we selected 20 of the 40 available plants that bore fruit. For these plants, a random number of peppers between 1 and 5 was generated from a uniform distribution for the left and right sides of the plant. The simulation setups for this use case in terms of number of peppers ranged from 3 to 6 peepers per plant. For every scenario the yield estimation was correct with up to 10% margin of error.

References

Ankerst, M.; Breunig, M. M.; Kriegel, H.-P.; and Sander, J. 1999. OPTICS: Ordering points to identify the clustering

structure. *ACM Sigmod record*, 28(2): 49–60.

Barth, R.; IJsselmuiden, J.; Hemming, J.; and Van Henten, E. J. 2018. Data synthesis methods for semantic segmentation in agriculture: A Capsicum annuum dataset. *Computers and electronics in agriculture*, 144: 284–296.

Breunig, M. M.; Kriegel, H.-P.; Ng, R. T.; and Sander, J. 2000. LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 93–104.

Di Cicco, M.; Potena, C.; Grisetti, G.; and Pretto, A. 2017. Automatic model based dataset generation for fast and accurate crop and weeds detection. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5188–5195. IEEE.

Fu, L.; Gao, F.; Wu, J.; Li, R.; Karkee, M.; and Zhang, Q. 2020. Application of consumer RGB-D cameras for fruit detection and localization in field: A critical review. *Computers and Electronics in Agriculture*, 177: 105687.

Hinterstoisser, S.; Pauly, O.; Heibel, H.; Martina, M.; and Bokeloh, M. 2019. An annotation saved is an annotation earned: Using fully synthetic training for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 0–0.

Khan, S.; Phan, B.; Salay, R.; and Czarnecki, K. 2019. ProcSy: Procedural Synthetic Dataset Generation Towards Influence Factor Studies Of Semantic Segmentation Networks. In *CVPR Workshops*, 88–96.

Lehnert, C.; English, A.; McCool, C.; Tow, A. W.; and Perez, T. 2017. Autonomous sweet pepper harvesting for protected cropping systems. *IEEE Robotics and Automation Letters*, 2(2): 872–879.

Lehnert, C.; Sa, I.; McCool, C.; Upcroft, B.; and Perez, T. 2016. Sweet pepper pose detection and grasping for automated crop harvesting. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2428–2434. IEEE.

Olatunji, J.; Redding, G.; Rowe, C.; and East, A. 2020. Reconstruction of kiwifruit fruit geometry using a CGAN trained on a synthetic dataset. *Computers and Electronics in Agriculture*, 177: 105699.

Pham, H.; and Pham, Q. 2018. A New Approach to Time-Optimal Path Parameterization Based on Reachability Analysis. *IEEE Transactions on Robotics*, 34(3): 645–659.

Polić, M.; Tabak, J.; and Orsag, M. 2021. Pepper to fall: a perception method for sweet pepper robotic harvesting. *preprint*.

Sa, I.; Lehnert, C.; English, A.; McCool, C.; Dayoub, F.; Upcroft, B.; and Perez, T. 2017. Peduncle detection of sweet pepper for autonomous crop harvesting—combined color and 3-D information. *IEEE Robotics and Automation Letters*, 2(2): 765–772.

Zhang, K.; Wu, Q.; and Chen, Y. 2021. Detecting soybean leaf disease from synthetic image using multi-feature fusion faster R-CNN. *Computers and Electronics in Agriculture*, 183: 106064.