Reinforcement learning with space carving for plant scanning

Antonio Pico Villalpando[1], Matthias Kubisch[1], David Colliaux[2], Peter Hanappe[2], and Verena V. Hafner[1]

[1]Adaptive Systems Group, Humboldt-Universität zu Berlin, Germany
[2]Sony CSL Paris, France

Abstract

Optimal plant reconstruction is an essential element in automating our future agriculture. Computerized inspection of proper growth, nutrition, or pest infestation has become mandatory in fully autonomous in-door or micro-farm settings, shifting from fixed to moving camera systems. In industrial environments, plant scanning must work efficiently with a limited number of significant images to become economically viable. We present an adaptive learning algorithm for agricultural plant inspection robots, in particular, a specific type of reinforcement learning that we developed for our micro-farming platform created within the EU project ROMI. We suggest a new approach to 3D plant reconstruction by integrating the space carving technique with categorical Deep Q-Networks. Space carving leverages images captured from various positions to create a binary voxel grid, representing the occupied and unoccupied spaces of the scanned object. The proposed method incorporates partial 3D reconstructions of plants obtained through space carving, which get compared to a ground truth model to calculate the reward and guide scanning policies. We explain the algorithmic details and the 3D reconstruction technique in design, implementation, and evaluation. Experimental results confirm our approach's effectiveness in improving the 3D plant reconstruction process, highlighting its potential for further applications in agriculture and related fields.

1. Introduction

3D plant reconstruction has become an essential tool for understanding plant growth and development, facilitating advances in agriculture and plant breeding [4, 7]. Accurate and efficient 3D modelling techniques provide valuable information for monitoring plant health, optimizing resources, and improving yield predictions [16]. It has crucial applications for plant phenotyping as well as agricultural robotics. For example, it can be used in automated phenotyping set up to track the growth [20] and geometry of plants [19] or to guide the arm of a robot harvester [12]. In particular a good reconstruction of the plants enables further processing like plant 3d segmentation [6].

Various methods have been proposed to reconstruct 3D plant models, including ToF, stereo vision or LiDAR [13] or photogrammetry and space carving [21]. However, these techniques often require a significant amount of time and manual intervention, limiting their applicability in large-scale and real-time scenarios. Photogrammetry and space carving require the acquisition of a collection of RGB images from various viewpoints, which has a cost in terms of time and computing power. Some of these images may be more informative and it is thus important to collect the least amount of images which make the 3d reconstruction most accurate. This adaptive planning of the trajectory of the camera is also important in an exploration framework, when capturing images with drones for example, where it is known as next-best view planning, and information based strategies [3] or reinforcement learning (RL) algorithms [5] have been proposed. Active vision and RL algorithms have also shown potential in optimizing image acquisition policies for 3D reconstruction tasks [8] [5]. RL algorithms enable an agent to learn through trial and error, finding the best strategy to complete a task while maximizing a cumulative reward [14]. Recently, researchers have started to explore the integration of RL algorithms with 3D reconstruction techniques to improve efficiency and accuracy [17] [10]. Space carving is a 3D reconstruction technique that constructs a binary voxel grid by processing images captured from various positions, determining whether spaces are occupied by the scanned object or not [11]. Space carving has been used in various applications, including object recognition and scene reconstruction [1] [9] [22].

In this paper, we propose a novel approach to 3D plant reconstruction by integrating space carving with reinforce-

ment learning algorithms. We investigate the potential of space carving for obtaining partial 3D reconstructions of plants and use them to guide the scanning policies through reward calculation. We present the details of the reinforcement learning algorithm, the space carving technique, and the design, implementation, and testing of the algorithm. Lastly, experimental results demonstrate the effectiveness of the proposed method in achieving accurate and efficient 3D plant reconstructions.

2. Methods

2.1. Reinforcement learning

Reinforcement learning (RL) involves mapping situations to actions to maximize a reward signal, with the agent discovering rewarding actions through trial and error. The agent and environment interact in discrete time steps, with the environment providing states and rewards, and the agent choosing actions. The agent's goal is to maximize expected cumulative rewards using a policy, which maps states to actions.

A critical component of RL is the action-state value function, Q value, representing long-term rewards. This function is represented by a deep artificial neural network when deep reinforcement learning techniques are used. In this work, we implemented an enhanced Q-learning algorithm called Categorical DQN (Deep Q-Network) (known as C51).

2.2. Space carving

The space carving algorithm [11, 15] creates a 3D model of a physical object by iteratively removing voxels from a solid volume (Fig. 1). The initial solid cube consists of a 3D grid of voxels, all marked as undetermined (coded by value, $o = 0$). Each time a new picture is acquired, a binary mask segments the plant and the background and all voxels projecting to background regions of the image are switched to unoccupied ($o = -1$) and those belonging to the plant region switched to occupied ($o = 1$). By iteratively carving the space in such a manner, the 3d reconstruction of the plants gets more and more refined as the number of images taken increases.

3. Experimental setup

3.1. Agent and environment

The ROMI Plant Imager[1] is a phenotyping station consisting of a camera mounted on a cartesian arm [18]. To accelerate the learning process, we use a Virtual Plant Imager in the following.

In our experiment, the camera (agent) moved in a semi-sphere space (see Fig. 2) around a plant, taking discrete 2-degree steps and always pointing to the centre. With three additional vertical levels, there were 720 possible positions for the agent.

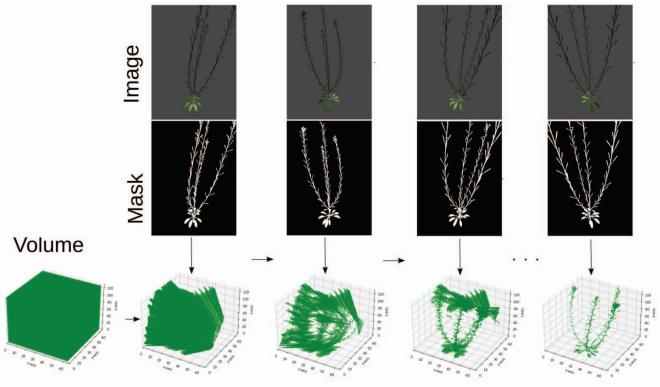


Figure 1. Different steps for creating a 3D model of a plant using space carving.

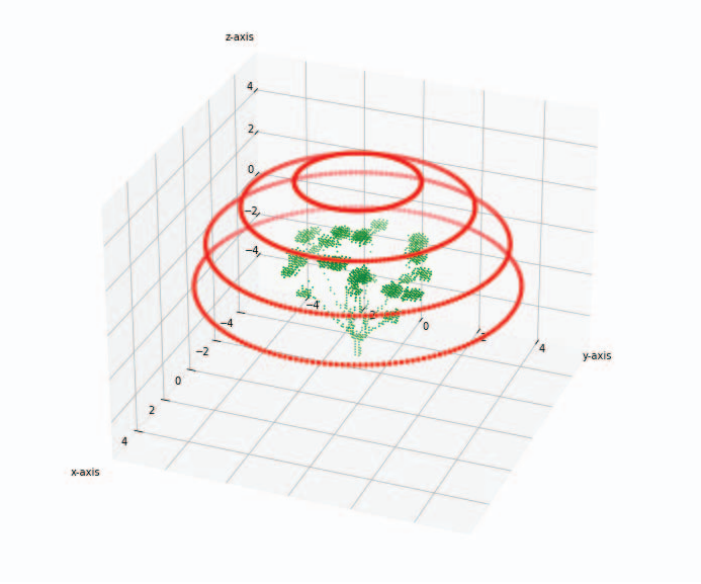


Figure 2. Representation of the camera setup. Every red point represents one possible position of the camera, resulting in a total of 720 images.

We created 8 simulated environments for agent interaction, comprising 720 images each with real-world camera coordinates, sourced from various plant and geometric models in Blender[2]. Five environments were allocated for training the scanning algorithm, and three for testing (see Fig. 3).

3.2. Goals, actions, states, and rewards

3.2.1 Goal

The primary goal of reinforcement learning is to optimize the agent's actions to achieve a defined objective. In these

[1] https://docs.romi-project.eu/plant_imager/

[2] https://www.blender.org/

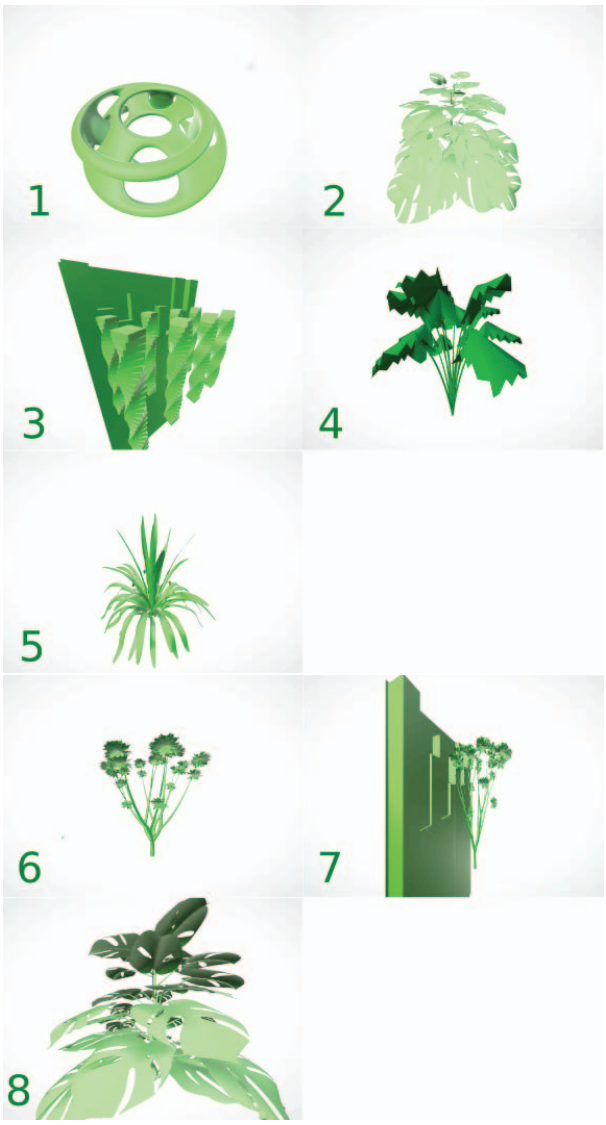


Figure 3. Five environments used for training (1-5) and three for testing (6-8) the reinforcement learning scanning algorithm.

experiments, the aim is to obtain the best 3D reconstruction of a plant using a limited number of 2D images taken at different camera positions. The goal can also be viewed as collecting the optimal set of images for the space carving algorithm.

To evaluate reconstruction quality, a high-resolution reconstruction for each environment was used as ground truth. Comparing current reconstructions to ground truth during training guided the agent toward an optimal solution. The agent learns to prioritize actions that bring reconstructions closer to ground truth, under the constraint of allowed images. Actions are defined as discrete camera arm movements along a semi-sphere trajectory around the plant, detailed in the next section.

3.2.2 Actions

At every time step, the agent could perform a single action, which corresponded to a horizontal-vertical movement pair. The horizontal movement (circle around the plant) was restricted to be anticlockwise and had the following possible values which were chosen empirically:

[0, 2, 4, 7, 9, 11, 14, 16, 18, 21, 23, 26,
28, 30, 33, 35, 37, 40, 42, 45, 67, 90]

These values indicate how many steps the agent has to move around the plant from the current position. Every step represents a movement of 2° along the above-mentioned circular path. The vertical movement was composed of the values [-3, -2, -1, 0, 1, 2, 3], which indicate the number of steps the agent has to move vertically (there are four different levels) from the current position. The combination of horizontal and vertical movements results in 140 different action pairs that the agent is able to execute.

3.2.3 States

The RL algorithm relies on rewards and states to define the more suitable actions to execute. In this work, two RL systems were designed: one using seen images as state and another using the current 3D reconstruction of the plant object. For the image-based state, the last three images collected by the agent were converted to grayscale and resized to 84x84 pixels (see Fig. 4).

The state representing the current carved volume comprised a 64x64x64 tensor (see Fig. 4). Each tensor element represents a voxel with three possible values: -1 for empty voxels, 0 for undetermined voxels, and 1 for solid voxels.

3.2.4 Rewards

A reinforcement learning algorithm requires the definition of a reward function to lead the learning process. Here, we defined the reward as the improvement (difference) of similarity between the current state of the carved volume and a known volume of the plant under analysis (ground truth). This is quantified by the Intersection over Union:

$$IoU = \frac{R_i \cap R_{gt}}{R_i \cup R_{gt}}$$

where R_{gt} and R_i are the sets of voxels for the ground truth reconstruction and the reconstruction at iteration i respectively. This calculation was made at every time step in the training process.

3.3. DQN model

We implemented Categorical DQN (C51) in this work. Traditional Q-learning algorithms learn a Q function to pre-

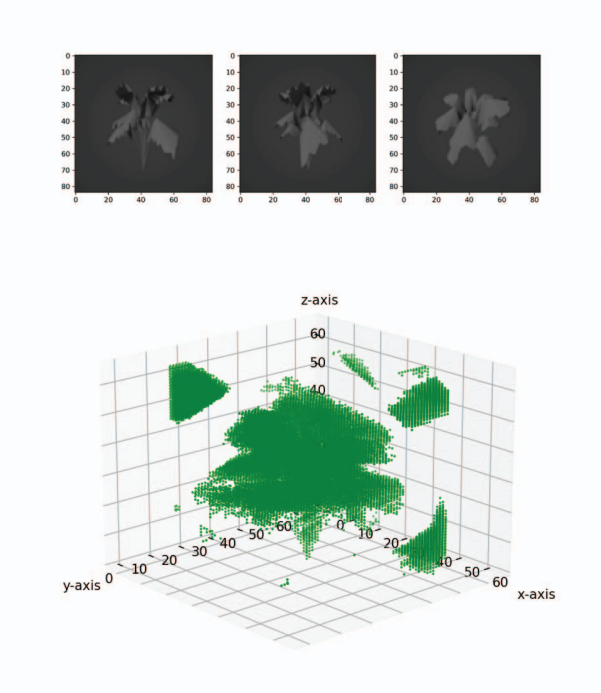


Figure 4. State represented by images (grey) and represented by volume (green). A partially reconstructed plant is also shown.

dict long-term rewards from a certain action and state, using the average of observed rewards. However, this discards valuable information about the environment. To address this, C51 predicts a histogram model for the Q-value probability distribution instead of just the expected value. This approach improves stability during training and enhances performance [2].

The model structure (Fig. 5) consists of three main parts:

- **Experience Replay**
  This is a ring buffer used to keep a history of samples collected by the agent. It is only required in the training process in order to improve sample efficiency (samples can be used more than once for training the model). Additionally, selecting batches of samples from different episodes avoids feeding the model with highly correlated data, which improves the convergence of the system. Our models used a buffer of 50000 elements with a batch size of 64.

- **Q-function**
  This is a neural network that predicts the probability distribution of state-action values. In training mode, it is fed from the experience replay buffer with batches of samples, while in normal mode it gets samples directly from the environment. This network consisted of 3 layers of 3D convolutions (for processing the carving volume), followed by 2 fully connected layers. The same architecture applies to the models with images as states, with the only difference being that 2D convolutions were used before the Fully Connected (FC) layers.

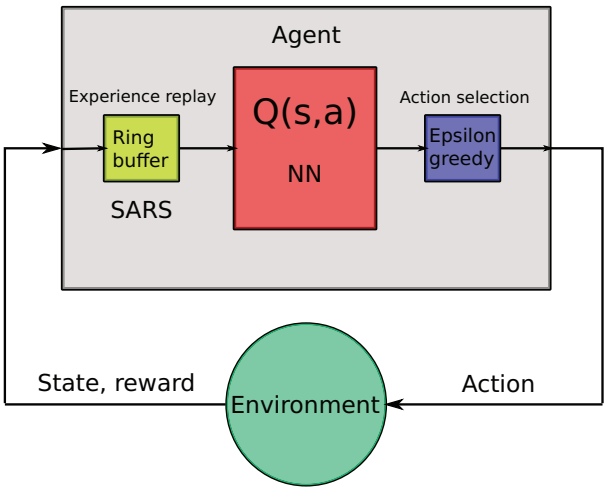


Figure 5. Reinforcement learning architecture.

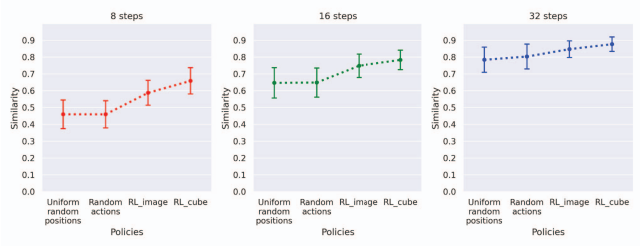
- **Action selection**
  This module utilises the epsilon greedy algorithm for selecting the action the agent is to execute. This algorithm selects the action with the best expected cumulative reward obtained from the Q function within a specified probability. Otherwise, it chooses an action randomly. This random action selection is useful in the training process because it helps the agent to explore new action combinations that could lead to finding better policies.

4. Results

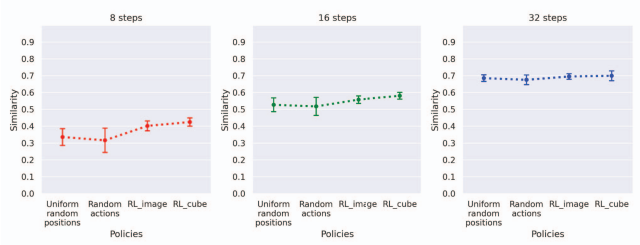
Two types of reinforcement learning agents were defined: one using images as state and another using the 3D reconstruction volume. For each, three policies collecting 8, 16, and 32 images were trained, resulting in six policies. Each movement had to generate an image, with no movement allowed without taking one.

During training, an environment was randomly chosen (see Fig. 3), and data collected for one episode, and repeated for every episode. The goal was to learn a policy for good 3D reconstruction in training environments, expected to work for unseen environments.

The six trained policies were tested in 200 episodes with random initial positions across environments. Figures 6a, 6b, 8a, 8b, and 8c display the mean and standard deviation of similarity between the reconstructed volume and ground truth for environments 3 and 4, and 6, 7, and 8, respectively. To evaluate if agents learned to find good image sets, two random policies were tested: one selecting positions



(a) Training results for environment 3 for 8, 16, and 32 steps.



(b) Training results for environment 4 for 8, 16, and 32 steps.

Figure 6. Similarities (IoU) between the 3D reconstructions and the ground truth model for the training environments 3 and 4. Four different policies are compared: one selecting random positions, one executing random actions, one RL trained using images as inputs, and one RL trained using the reconstruction volume.

randomly with a uniform distribution and another executing random actions.

Trained policies achieved better 3D reconstructions than random ones in most tests. However, using more images diminished this advantage. For instance, in environment 4 with 32 images, all policies had similar results (Fig. 6b). As more images are used, finding better camera views becomes less important since object coverage improves. This is supported by 8-image tests, where the IoU difference was significantly better than random policies. In more constrained environments, learned policies tend to find better camera views.

By comparing the learned policy using images as state against the one using the reconstructed volume, it is shown that their performance is very similar in all environments except for environment 3, where the policy using the volume achieved an IoU up to 10 percent better than the one using images (Fig. 6a).

The test plots revealed that learned policies did not perform well compared to random ones in environment 6 (Fig. 8a), except for the 8-image test using images as state. However, in environment 7 when the plant was placed next to a wall, learned policies achieved better results (Fig. 8b). This could be due to the fact that environment 6's regular, symmetrical plant structure makes any viewpoint equally valuable for 3D reconstruction. In contrast, the wall in environment 7 obstructs some viewpoints, making random camera

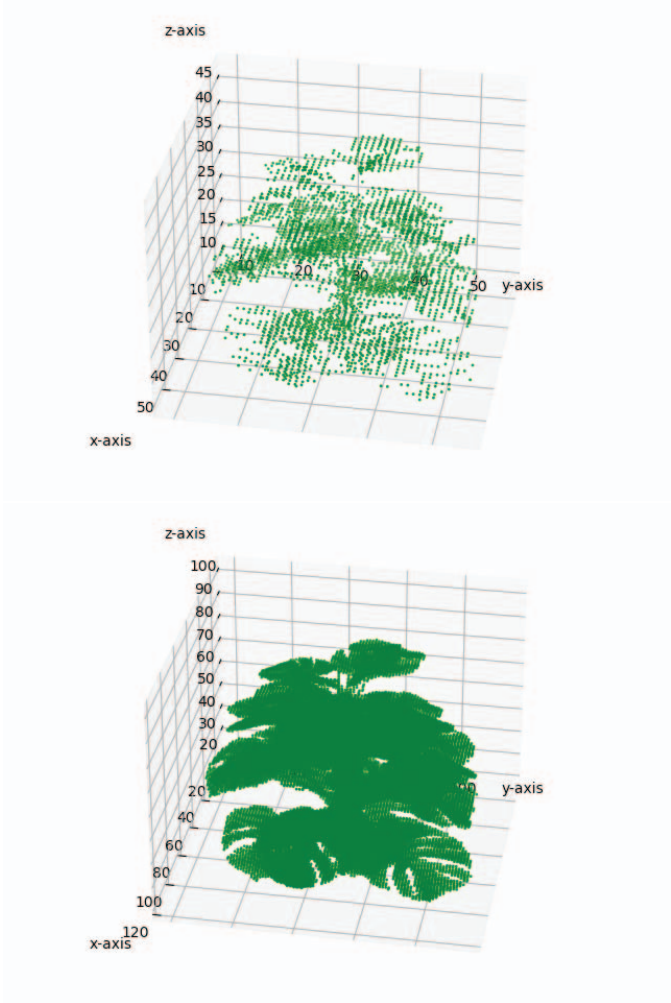
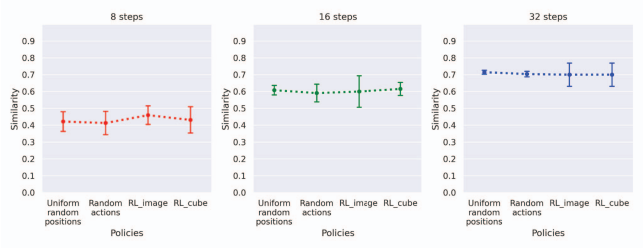


Figure 7. Voxel volume (top) used for training the algorithm using environment 2 and its post-processed version (bottom).

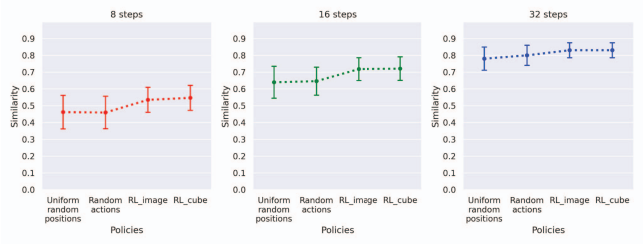
placement less effective, while trained policies can move to better positions.

In Environment 8, all policies had low IoU (Fig. 8c), as expected, since the camera's closer proximity to the plant requires more images to cover the entire volume. Interestingly, the performance gap between random and trained policies grows with more images taken in this environment. This could be due to the need for many more images to cover the volume, where taking a few images from any viewpoint provides little information. Once a sufficient number of images are taken, the impact of collecting images from good viewpoints is reflected in the resulting similarity ratio. Examples of reconstructions for environments 5, 6, 7, and 8 are shown in Fig. 9.

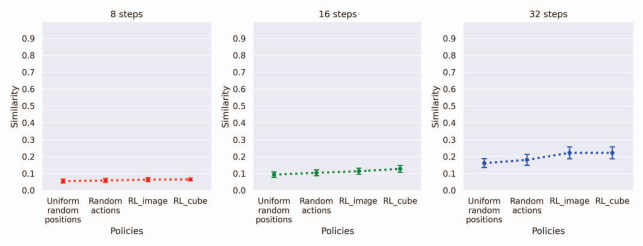
The reconstructed volumes shown previously were set at dimensions of 64x64x64 voxels to expedite the training process, which might not be optimal for generating final 3D models. This can be addressed by applying a post-



(a) Test results for environment 6 for 8, 16, and 32 steps.



(b) Test results for environment 7 for 8, 16, and 32 steps.



(c) Test results for environment 8 for 8, 16, and 32 steps.

Figure 8. Similarities (IoU) between the 3D reconstructions and the ground truth model for the test environments. Four different policies are compared: one selecting random positions, one executing random actions, one RL trained using images as inputs, and one RL trained using the reconstruction volume.

processing step to the collected images. Alongside saving the images, the algorithm should also save the camera positions where the images were taken, allowing space carving to be reapplied for generating volumes with a higher voxel density. Fig. 7 illustrates the volume used for training and the resulting post-processed volume.

5. Conclusion

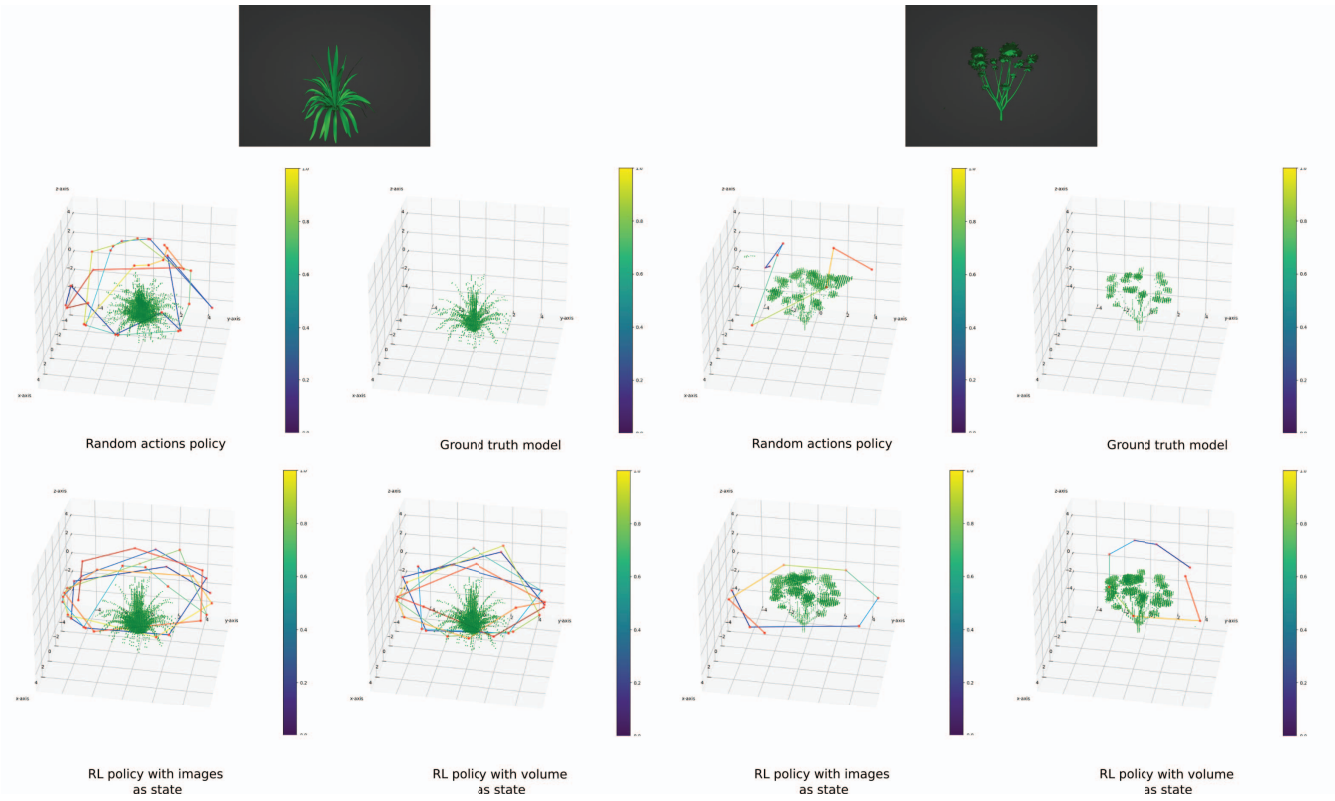
This work demonstrates an adaptive reinforcement learning algorithm developed for the micro-farming platform within the EU project ROMI, focusing on the innovative integration of space carving techniques with reinforcement learning algorithms for 3D plant reconstruction. Space carving uses images captured from various positions to create a binary voxel grid representing the occupied and unoccupied spaces of the scanned object.

The experimental results demonstrate the effectiveness of this approach in improving the 3D plant reconstruction process, showcasing its potential for broader applications in agriculture and related fields. Future work will focus on adapting the algorithms for real-world environments and integrating them with real robotic platforms, enabling more sophisticated 3D reconstruction systems and facilitating further research on sensor fusion and data-driven improvements to the reconstruction process.

Ultimately, this research highlights the effectiveness of reinforcement learning in 3D plant reconstruction and its ability to adapt to new situations. The ongoing work in refining the approach and expanding its use in agricultural robots and 3D reconstruction will lead to further advancements and practical applications in the field.

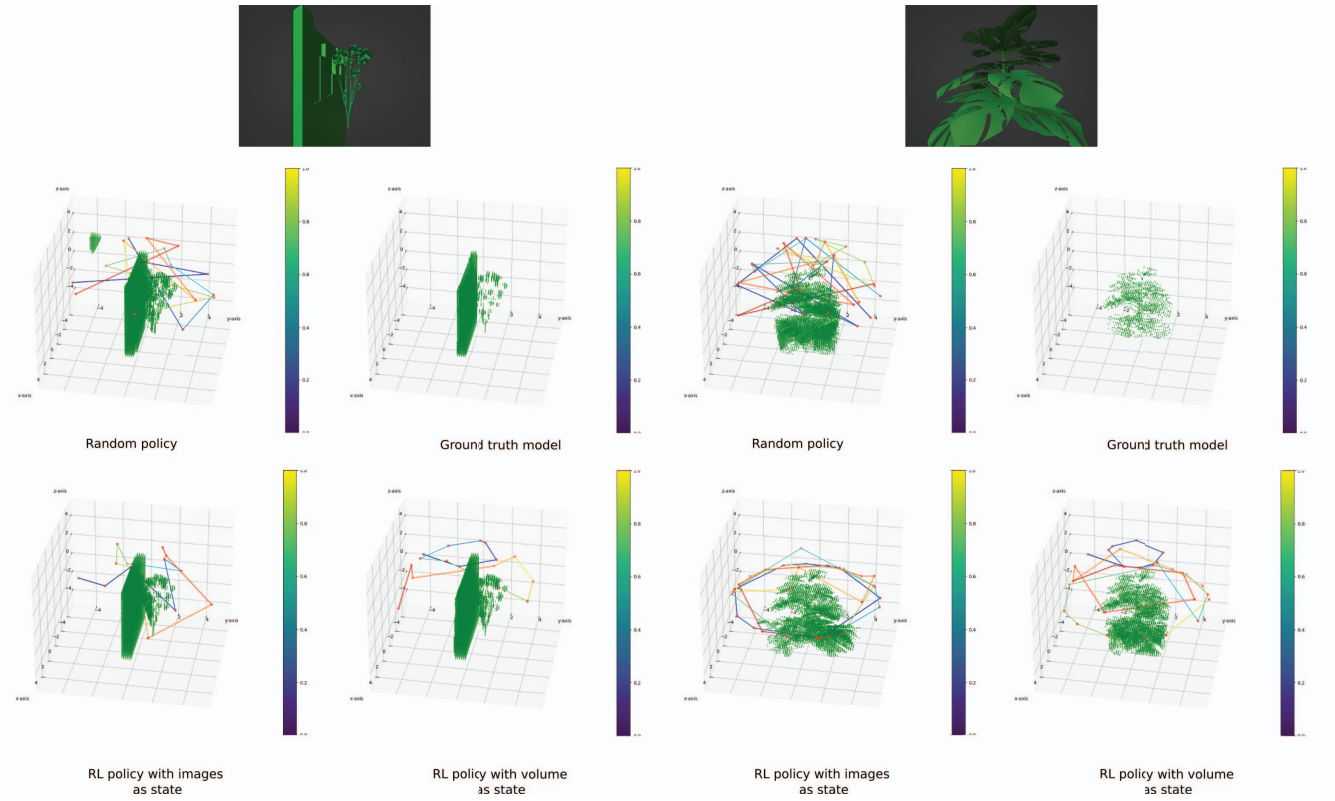
Acknowledgements

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 773875 (EU-H2020 ROMI, Robotics for Microfarms).



(a) Environment 5 with 32 images.

(b) Environment 6 with 8 images.



(c) Environment 7 with 16 images.

(d) Environment 8 with 32 images.

Figure 9. 3D reconstructions for different environments.

References

[1] Murat D Aykin and Shahriar Negahdaripour. Three-dimensional target reconstruction from multiple 2-d forward-scan sonar views by space carving. *IEEE Journal of Oceanic Engineering*, 42(3):574–589, 2016. 1

[2] Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International conference on machine learning*, pages 449–458. PMLR, 2017. 4

[3] Andreas Bircher, Mina Kamel, Kostas Alexis, Helen Oleynikova, and Roland Siegwart. Receding horizon" next-best-view" planner for 3d exploration. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 1462–1468. IEEE, 2016. 1

[4] Yann Chéné, David Rousseau, Philippe Lucidarme, Jessica Bertheloot, Valérie Caffier, Philippe Morel, Étienne Belin, and François Chapeau-Blondeau. On the use of depth camera for 3d phenotyping of entire plants. *Computers and Electronics in Agriculture*, 82:122–127, 2012. 1

[5] Mustafa Devrim Kaba, Mustafa Gokhan Uzunbas, and Ser Nam Lim. A reinforcement learning approach to the view planning problem. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6933–6941, 2017. 1

[6] Helin Dutagaci, Pejman Rasti, Gilles Galopin, and David Rousseau. Rose-x: an annotated data set for evaluation of 3d plant organ segmentation methods. *Plant methods*, 16(1):1–14, 2020. 1

[7] Jonathon A Gibbs, Michael Pound, Andrew P French, Darren M Wells, Erik Murchie, and Tony Pridmore. Approaches to three-dimensional reconstruction of plant shoot topology and geometry. *Functional Plant Biology*, 44(1):62–75, 2016. 1

[8] Jonathon A Gibbs, Michael P Pound, Andrew P French, Darren M Wells, Erik H Murchie, and Tony P Pridmore. Active vision and surface reconstruction for 3d plant shoot modelling. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 17(6):1907–1917, 2019. 1

[9] Raia Hadsell, J Andrew Bagnell, Daniel F Huber, and Martial Hebert. Accurate rough terrain estimation with space-carving kernels. In *Robotics: Science and Systems*, volume 2009, page 62, 2009. 1

[10] Seiya Ito, Byeongjun Ju, Naoshi Kaneko, and Kazuhiko Sumi. independent single-view 3d object reconstruction using reinforcement learning. In *VISIGRAPP (5: VISAPP)*, pages 811–819, 2022. 1

[11] Kiriakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. *International journal of computer vision*, 38:199–218, 2000. 1, 2

[12] Christopher Lehnert, Inkyu Sa, Christopher McCool, Ben Upcroft, and Tristan Perez. Sweet pepper pose detection and grasping for automated crop harvesting. In *2016 IEEE international conference on robotics and automation (ICRA)*, pages 2428–2434. IEEE, 2016. 1

[13] Lei Li, Qin Zhang, and Danfeng Huang. A review of imaging techniques for plant phenotyping. *Sensors*, 14(11):20078–20111, 2014. 1

[14] ML Littman and AW Moore. Reinforcement learning: A survey, journal of artificial intelligence research 4, 1996. 1

[15] David Israel Lovi. Incremental free-space carving for real-time 3d reconstruction. 2011. 2

[16] Stefan Paulus, Jan Dupuis, Anne-Katrin Mahlein, and Heiner Kuhlmann. Surface feature based classification of plant organs from 3d laserscanned point clouds for plant phenotyping. *BMC bioinformatics*, 14(1):1–12, 2013. 1

[17] Daryl Peralta, Joel Casimiro, Aldrin Michael Nilles, Justine Aletta Aguilar, Rowel Atienza, and Rhandley Cajote. Next-best view policy for 3d reconstruction. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*, pages 558–573. Springer, 2020. 1

[18] Guido Schillaci, Antonio Pico Villalpando, Verena V Hafner, Peter Hanappe, David Colliaux, and Timothée Wintz. Intrinsic motivation and episodic memories for robot exploration of high-dimensional sensory spaces. *Adaptive Behavior*, 29(6):549–566, 2021. 2

[19] David Schunck, Federico Magistri, Radu Alexandru Rosu, André Cornelißen, Nived Chebrolu, Stefan Paulus, Jens Léon, Sven Behnke, Cyrill Stachniss, Heiner Kuhlmann, et al. Pheno4d: A spatio-temporal dataset of maize and tomato plant point clouds for phenotyping and advanced plant analysis. *Plos one*, 16(8):e0256340, 2021. 1

[20] Weinan Shi, Rick van de Zedde, Huanyu Jiang, and Gert Kootstra. Plant-part segmentation using deep learning and multi-view vision. *Biosystems Engineering*, 187:81–95, 2019. 1

[21] Suresh Thapa, Feiyu Zhu, Harkamal Walia, Hongfeng Yu, and Yufeng Ge. A novel lidar-based instrument for high-throughput, 3d measurement of morphological traits in maize and sorghum. *Sensors*, 18(4):1187, 2018. 1

[22] Timothée Wintz, David Colliaux, Peter Hanappe, and Sony CSL Paris. Automated extraction of phyllotactic traits from arabidopsis thaliana. In *CVPPP workshop, ECCV*, volume 2018, page 6, 2018. 1