Simulation environment for the deployment of robots in precision agriculture

J. Rodriguez and D. Nardi

Abstract— **The use of robotic platforms for precision agriculture applications is a popular research topic nowadays. The advantages offered by these platforms have been shown in several of the tasks performed manually by operators. The deployment of robots for crop inspection and evaluation is one of the most studied problems in precision agriculture. The robots explore the crops to collect the necessary data for evaluating the vegetation status. The task requires a complex platform setup which depends of the mission requirements, the type of terrain and vegetation. The trial and error stage for obtaining the data that are actually useful requires significant time and effort in a real scenario. The use of a realistic simulation environment for testing may reduce the cost and time invested. In this work, we developed a *testing simulation environment* for precision agriculture robotics tasks. We focused on the problem of detecting the downy mildew in sunflowers crops. This type of parasite can be detected by analyzing morphological parameters such as the leaf color and the plant height. In particular, we approach the problem of estimating the sunflowers height by generating a 3D model of the crops. The obtained results are compared with the ground truth model of the plant to determine the accuracy of the estimation. Different configurations of the testing setup are evaluated to determine the optimal configuration to perform a successful 3D reconstruction.**

I. INTRODUCTION

Crop inspection and evaluation in precision agriculture is key to estimate the health of the crops. The manual inspection process is time demanding and requires human effort. Farmers have to do a one by one plant evaluation in order to extract the information to estimate the crops state. In the last years, UAVs and computer vision algorithms have been used to collect the necessary data to perform crop analysis [1]. The approach has improved the data collection process in terms of time and costs. In general, the UAV overflies the terrain while capturing top view images from the fields. Then, by using image processing the vegetation indexes are computed for crop health estimation. The UAV carries on board a specialized set of sensors for image capturing. In the simple case, RGB cameras are used to compute the soil adjusted vegetation index (SAVI) and the Triangular Greenness Index (TGI) [2]. However, a deep inspection requires the use of hyperspectral cameras to compute additional parameters such as the normalized difference vegetation index (NDVI) [3], the leaf area index (LAI) and the plant biomass. Additionally, morphological information of the vegetation can be obtained by using dedicated camera sensors and LiDAR. Crop height and size are used to determine the presence of diseases such as the downy mildew parasite. Performing the inspection using UAVs is in general an easy task, however, it depends on the weather conditions and the growth stage of the plants. This fact reduces the time window for evaluating the crop status and testing novel methodologies and algorithms in a real scenario. Further, in order to obtain data that can be effectively used by post-processing to extract the information of interest, the sensor setup and configuration need to be obtained through trial phase is demanding in time and cost. The use of a simulation environment for crop data acquisition could solve both testing and calibrating issues by offering an available software platform for testing, thus reducing substantially operational costs. This document is divided as follows: Section II discusses the role and features of the simulation environment for our application. Section III introduces the proposed solution. In section IV, some preliminary results are reported, and in the last section the conclusions and future work are presented.

Rodriguez and Nardi are with the Department of Computer, Control, and Management Engineering Antonio Ruberti, Sapienza University of Rome, Italy. Email:{jrodriguezg,nardi}@diag.uniroma1.it

II. SIMULATION TOOLS

Robotic simulators are typically used for testing robots in challenging scenarios under safety conditions. Currently, these simulation tools are mainly used for robotics modeling, sensor calibration, and testing novel concepts, models, and algorithms. The use of simulators supports the evaluation of the system capabilities without endangering the hardware components. For example, testing in simulation reduces the trial and error phase to determine the optimal sensor configuration. The success of this calibration depends on the fidelity of the sensors with respect to the real model and the accuracy of the virtual representation. Currently, most of the simulators include several models of the robot platforms, which allow for a direct software transfer from the virtual model to the real robot. Furthermore, simulators offer tools for sensor modeling. The developers are able to create and modify the sensor models given their own requirements. In this way, the users are able to evaluate the best hardware solution for their application. Additionally, the simulators support testing in different types of environments. In a real scenario modifying the work area can be expensive and time demanding especially when there are significant differences between the scenarios. Currently, most of the robotic simulators include a direct integration of CAD models into the scenes. This feature simplifies the task of modifying the scenarios by importing 3D models from different sources.

The most important features to consider for a robotic simulator are the *cost*, *transferability*, *compatibility* and *an accurate representation of the simulation scene*. The cost divides the simulators into two main categories: the commercial simulators and the open-source platforms. Among the most used commercial simulators, there are the RoboDK and Webots [4].

The open-source simulators support easy sharing and integration capabilities. Users are able to access and use models, scenarios, and algorithms already created by other developers which boost the implementation of their own solution.

Transferability reduces the time used for migrating, adapting and testing the solution developed in the real robot. This feature depends on the accuracy of the robot model. For instance, given an accurate robot representation, the controllers and sensor calibration can be directly transferred, while preserving to a good extent the behavior obtained from the simulations. For instance, Gazebo [5] includes the model of several robotic platforms that use ROS. Gazebo and ROS can interact in parallel to perform a simulation with the same parameters, and control architectures included in the real robots. In this way, the users are able to transfer their projects directly to the robots significantly reducing additional implementation steps.

Compatibility allows for the integration of the simulator with additional libraries, frameworks, and software. In this way, the developers are able to use the features and capabilities from different sources. For example, the V-REP [6] simulator can work in parallel with frameworks such as Matlab and ROS, which are suitable solutions to handle the vision and control part of the simulation.

Finally, by having a solid representation of the scene the users are able to study the scene parameters that may affect the robot behavior in a real scenario. This feature benefits the sensor calibration and platform configuration by including the possible uncertainties to reject.

Game engines have been used for robotic simulation in the context of human-robot interaction [7], and in robot soccer and rescue competitions [8]. These engines were created to design and develop video games. However, their high quality graphical engine and powerful physics engines are suitable to emulate collisions and body dynamics of robots. Unreal Engine (UE4) is an open-source platform for gaming developing. The accuracy and quality of the simulator fit well with the requirements needed for robotic simulation. Besides, the engine can work in parallel with ROS to add robotic functionalities to the world actors. The block work-flow of the engine is pretty intuitive and allows to easily exploit most of the advantages offered by the engine. UE4 has been previously used in the AI and robotics area for automatic dataset generation in [9], training, and validation algorithms for autonomous vehicles [10],[11], UAV-based tracking [12], and the evaluation of robotic 3D mapping systems [13].

Crop simulation for precision agriculture requires a solid scene representation to evaluate the system behavior under specific conditions such as image blurring, lens distortion, ambient noise and wind effects. The UE4 engine offers these capabilities making it a suitable choice in our context. Hence we chose it for our application. In this work, we use the UE4 engine for crop simulation in precision agriculture. In particular, we aim to perform 3D reconstruction of the crops using synthetic data. Additionally, a trial and error process is performed to determine the optimal sensors and mission parameters setup.

III. Proposed solution

We propose a crop simulation environment for precision agriculture developed in UE4. Our work is inspired by the approach followed in [9]. The main goal is to generate an accurate simulation model of the field to perform a 3D reconstruction of the crops. We focus our attention on the problem of plant height estimation using a UAV. The proposed solution generates a model of the crops that enables the extraction of height information. The UAV collects images of the field by using a monocular camera. Then the images are used to perform the 3D reconstruction. The scenario can be modified by adding, moving and importing new objects in the scene as well as by adjusting the platform setup and the mission trajectory. In this way, the user is able to quickly create new testing scenarios. Our solution uses the Photoscan software [14] to perform the 3D reconstruction. The following sections explain the development of the simulation environment, the platform configuration and testing phase, and the crop 3D reconstruction that is exploited for height estimation.

A. Simulation environment

The simulation environment is a unique scene that includes the necessary lighting, physics, and models for the simulation. We use 3 types of objects in the simulator: the landscape, the object models and the actor that represents the UAV. The object models were downloaded from CS repositories and then imported into the scene. In order to obtain a more realistic representation of the plants, additional textures were considered and modified by including normal and spectral layers. The UAV is simulated by a camera actor that follows a trajectory defined by the user. In the initial phase of our work, we focused on acquiring the necessary image information from the onboard camera to build the 3D reconstruction of the crops, and therefore we did not need a realistic visual model of the UAV into the scene. The images are captured at 4.5Hz using the model of an undistorted pinhole camera with a resolution of 1195x677 pixels. Figure 1 shows the perspective view of the simulation environment.

B. 3D reconstruction and height estimation

In general, there are a number of technical issues to consider in order to generate a 3D modeling using images.

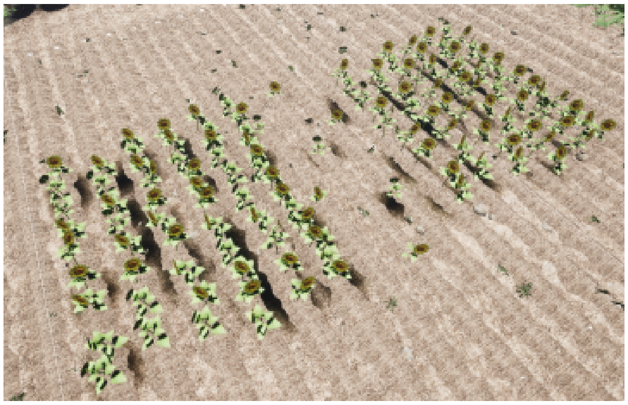

Fig. 1. Perspective view of the crops simulation environment for precision agriculture

First, a complete reconstruction of the goal requires pictures taken from different perspectives. Besides, the use of low-resolution imagery affects the outcome of the 3D modeling process. Finally, the type and amount of relevant features in the images is crucial to perform image association. These considerations constrain our problem as follows: (i) by overflying the terrain at a constant altitude, the collected images are taken from a top perspective and therefore the complete reconstruction of the crops is limited, (ii) once the plants are in the last growing stages, the number of key features in the images is reduced because the canopy surface becomes thick and homogeneous. The aim of our the 3D reconstruction of the field is to generate the model of the crops in order to extract the height information. For this purpose, it is not necessary to build the lateral part of the crops body. Indeed, an accurate 3D representation of the canopy surface is enough to determine the crop height. Regarding the lack of features on the canopy surface, our initial hypothesis is to use a high-resolution RGB camera to obtain the required amount of features to perform the image matching process.

The height estimation is performed by using the point cloud obtained from Photoscan. From the software, the digital elevation map (DEM) can be generated to determine the height of the canopy. However, in this project, we preferred to use the point cloud of the terrain as it includes more information about the scenario. Once the cloud is extracted, an algorithm classifies the points that belong to the canopy surface. Then, the mean of the canopy surface points is computed to obtain an average value of height. This is a first indication of the possibility of detecting geometric features that support the analysis of plant health.

C. Mission configuration

The addition to the issues for 3D reconstruction described in the previous section, the camera configuration and the setup of the mission parameters can affect the 3D modeling process. Therefore, a preliminary phase is required to determine the configuration that leads to the best results for 3D reconstruction. First, a camera calibration stage is performed as it is required by Photoscan in order to perform a correct image alignment. Then, the parameters for the mission setup such as the speed, the mission trajectory, and the flying altitude are evaluated. To this end, the trial and error capability offered by the simulation environment can be fully exploited.

IV. Results

For our experiments, we used a fixed value for the camera parameters. In UE4, the user is able to define the angle of view and the image resolution. Then, by using simple geometry the focal length and the center of the camera are computed. We chose an angle of 90 degrees and the maximum image resolution available. Then, we proceed to determine the mission parameters to improve the data collection process. The use of splines in UE4 simplifies the trajectory computation: the user defines a set of waypoints and then selects the type of path to follow. The trajectory is chosen in such a way that the UAV flies over the entire field. The flying velocity is fixed in order to capture enough images of the crops without adding blur effects. For our first experiments, we focus on determining the best flying altitude to perform 3D reconstruction. We performed tests by changing the altitude between 5m, 10m, 15m, and 20m. Regarding the plant model, we use sunflowers of 1 meter and 2 meters. The difference between the sunflower size is used to include the dwarfism effect given by the downy mildew presence on the crops. Our goal would be to detect plants of different height in the same scene. However, as a first step, we focus on computing the height of plants with the same size. An algorithm determines which points of the cloud belong to the canopy surface by using a threshold η. The points classification is shown in figure 2. From the simulation, the healthy and sick sunflowers have an average height of 2.1m and 1m measured from the center of the flower. The maximum height of the plants are 1.3m and 2.4m respectively. These values are used as ground truth to determine the accuracy of the estimation.

The simulation results are reported in table I. We evaluate the accuracy of the average height and the maximum height estimation. The best outcomes are obtained by fixing the UAV flying altitude 5 meters above the crops height. In this case, the average height obtained has an accuracy of 98% for both types of sunflowers. The maximum height estimation report an error of 6% . The tests at 10 m depict reasonable outcomes by having an error around 9% with both types of sunflowers. However, the outcomes of flying at an altitude above 10m significantly affect the accuracy of the estimation. The accuracy decreases when the distance between the camera and the plant increases. In particular, by flying at 15 m the accuracy dropped almost to 60%.

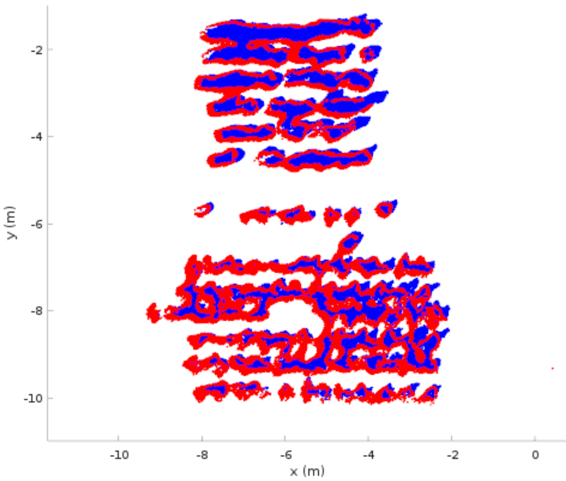

Fig. 2. Heightmap of the canopy. The blue zone represents the highest parts of the canopy surface while the red part depicts the bottom of the canopy.

Flying altitude	Healthy crops (2m)		Sick crops (1m)	
	avg (m)	max (m)	avg (m)	max (m)
5m	1.96	2.31	0.97	1.38
10m	1.95	2.46	0.89	1.08
15m	0.86	1.19	0.42	0.62
20m	0.65	0.92	0.30	0.38

TABLE I

HEIGHT CROPS ESTIMATION RESULTS BY VARYING THE FLYING ALTITUDE

V. CONCLUSIONS

We developed a simulation scenario for precision agriculture that includes a crop realistic model and a UAV camera actor to perform data collection for crop inspection. A 3D reconstruction map of the scene is generated to extract the required information to determine the vegetation status. Further, a set of trial and error tests were performed to establish the best parameter configuration and mission plan. The experimental results depict that the height estimation accuracy is improved by flying at an altitude of 5m above the canopy. Our simulation scenario offers advantages in terms of (i) time-saving, (ii) mission setup estimation, and (iii) scene adaptability in case of different scenario requirements. The next step in our research is the design of a methodology to differentiate between the sick and the healthy plants in the crops using the point cloud information.

REFERENCES

[1] C. Zhang and J. M. Kovacs, "The application of small unmanned aerial systems for precision agriculture: a review," *Precision Agriculture*, vol. 13, no. 6, pp. 693–712, Dec 2012. [Online]. Available: https://doi.org/10.1007/s11119-012-9274-5

[2] T. McKinnon, , and P. Hoff, "Comparing rgb-based vegetation indices with ndvi for agricultural drone imagery, 2017," *Agribotix LLC*.

[3] S. Candiago, F. Remondino, M. De Giglio, M. Dubbini, and M. Gattelli, "Evaluating multispectral images and vegetation indices for precision farming applications from uav images," *Remote Sensing*, vol. 7, no. 4, pp. 4026–4047, 2015. [Online]. Available: http://www.mdpi.com/2072-4292/7/4/4026

[4] O. Michel, "Webotstm: Professional mobile robot simulation, 2004," *arXiv preprint cs*, vol. 412052.

[5] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 3, Sept 2004, pp. 2149–2154 vol.3.

[6] E. Rohmer, S. P. N. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 1321–1326.

[7] C. Bartneck, M. Soucy, K. Fleuret, and E. B. Sandoval, "The robot engine #x2014; making the unity 3d game engine work for hri," in *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, Aug 2015, pp. 431–437.

[8] M. Zaratti, M. Fratarcangeli, and L. Iocchi, "A 3d simulator of multiple legged robots based on usarsim," in *RoboCup 2006: Robot Soccer World Cup X*, G. Lakemeyer, E. Sklar, D. G. Sorrenti, and T. Takahashi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 13–24.

[9] M. Di Cicco, C. Potena, G. Grisetti, and A. Pretto, "Automatic model based dataset generation for fast and accurate crop and weeds detection," *arXiv preprint arXiv:1612.03019*, 2016.

[10] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, M. Hutter and R. Siegwart, Eds. Cham: Springer International Publishing, 2018, pp. 621–635.

[11] A. Dosovitskiy, G. Ros, F. Codevilla, A. López, and V. Koltun, "Carla: An open urban driving simulator," *arXiv preprint arXiv:1711.03938*, 2017.

[12] M. Müller, V. Casser, J. Lahoud, N. Smith, and B. Ghanem, "Sim4cv: A photo-realistic simulator for computer vision applications," *International Journal of Computer Vision*, pp. 1–18, 2018.

[13] J. Du, C. Mouser, and W. Sheng, "Design and evaluation of a teleoperated robotic 3-d mapping system using an rgb-d sensor," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 5, pp. 718–724, May 2016.

[14] L. Agisoft and R. St Petersburg, "Agisoft photoscan," *Professional Edition*, 2014.