DART-Sim: An End-to-End Simulation Testbed for the Study of Deep Learning Powered Autonomous Drones for Precision Pollination of Tomatoes in a Greenhouse Environment

Rachel Bai
The Webb Schools

Abstract—**Pollination is one of the most important factors in the agricultural process. However, due to the decline of natural pollinators and the rise of large-scale greenhouse agriculture, many farmers are not able to rely only on natural pollinators like bees. In order to meet the demand for greenhouse crop production, a robust method for artificial pollination is necessary. A robotic pollination system based on an autonomous drone system may present a viable approach to precision robotic pollination in a commercial greenhouse environment. In the development of a system with this complexity, a simulation-based testbed offers advantages over direct physical implementations. This research is aimed to develop a capable and sophisticated simulation testbed, DART-Sim (Drone-based Autonomous Robotic Testbed for Simulation) that allows for the development of a drone-based autonomous pollination robot with sub-centimeter precision powered with deep-learning algorithms. DART-Sim features VSLAM algorithms such as monocular ORB-SLAM2 integrated with state-of-the-art (SOTA) deep learning algorithms like YOLO v5 for object detection, which enables sub-centimeter precision in localization and mapping. It is based on the popular ROS framework integrated with Gazebo, Rviz, and MoveIt. The working prototype testbed DART-Sim 1 is capable of a full-scale simulation of robotic artificial pollination system in a simulated indoor greenhouse environment for tomato plants with GPS-denied autonomous navigation. It also features a proof-of-concept prototype of an end-effector using a novel sprayer-based pollination mechanism as an attachable drone payload. This research demonstrates the feasibility and viability of a simulation-based approach for studying and building an autonomous precision pollination robot.**

I. Introduction

Insect pollination of crops is critical for commercial crop production and the security of the food production chain. In the US, the estimated value of pollinator-dependent crops is over $50 billion per year [2]. However, declining natural bee populations and an increasing demand for pollinators have resulted in increasingly expensive pollination costs for farmers. In tomato greenhouses, many areas lacking natural bee populations, such as areas like Maoxian County in Sichuan, China, are reliant on artificial pollination methods [3]. In order to address the decline of natural pollinators, an increasing number of farmers have turned to artificial pollination methods such as hand pollination or robotic pollination systems. There are many commercial crops that robotic pollination systems may be used for. However, tomatoes, the most common commercially grown greenhouse crop, present the most economically viable and accessible option for the basis of an autonomous robotic pollination system.

In the controlled environment of a tomato greenhouse, drone pollination is feasible but not currently practical or economically viable when compared to natural pollinators. Tomato flowers, which range from 1-2 cm across, often form alongside developing or ripe fruits, present a significant challenge for robotic pollination systems. A high threshold of precision and sophisticated image processing is necessary to detect the locations of mature tomato flowers accurately and precisely. Recent developments in drone-based artificial pollination systems have focused on improving the pollen delivery system or on implementing a physical end-to-end system, but not on precise or autonomous drone navigation [4]. A more efficient and precise process of navigation, flower detection, and pollination is necessary in order to allow for the development of a viable drone pollination. In this paper I present **DART-Sim**, an end-to-end simulation-based testbed that assists in the development of a precise and effective drone-based artificial pollination system. Using a novel method of mapping and localization with ORB-SLAM2 integrated with deep-learning object detection programs such as YOLO v5, sub-centimeter precision in localization and pollination of tomato flowers can be achieved. Using a deep-learning powered classification system for tomato plants in a greenhouse environment provides a map of tomato flowers which can be integrated with a depth map generated alongside ORB-SLAM2 for more precision in mapping and localization. A sprayer with two degrees of freedom and a secondary camera for precision tuning of the nozzle is also integrated into the system as part of the drone model, allowing for a wider range of possible locations for pollination. DART-Sim integrates these tools and frameworks into a unified simulation testbed, assisting in the development of a precise and efficient autonomous drone pollination system. Because of the flexible nature of the simulation testbed, the software, models, and image data pipelines can be modified or added on to in order to tackle other problems such as fruit disease identification or as a crop yield estimator.

II. Related Work

Farmers affected by declining bee populations have turned to artificial pollination methods, which can be separated into two groups: methods that involve human labor, and autonomous methods. In manual pollination, workers pollinate each flower by hand, a tedious and time-consuming task. Because of the scarcity and price of human labor in the agricultural process, manual pollination is also not feasible for large-scale farming. For large-scale farming in closed areas such as greenhouses, autonomous pollination presents a better solution to the problem of agricultural pollination. However, the efficiency and feasibility of different autonomous pollination methods differs. Passive methods of artificial pollination, such as wind-pollination [5], may be more cost-efficient in the short term, but advances in robotics and computer vision have made robotic autonomous pollination systems more feasible.

One of the advances that have made autonomous robotic pollination possible is efficient real-time V-SLAM, or Visual SLAM. Developments in Visual Odometry and developments in RGB-D Visual SLAM have allowed autonomous mobile robots to operate with greater speed, precision, and effectiveness, allowing for the development of fully autonomous robotic pollination systems [6].

There have been several different approaches to solving the problem of robotic artificial pollination. Various types of robotic pollination systems have already been developed. Most research into pollination robots use ground-based robots equipped with a highly maneuverable arm to individually pollinate flowers. BrambleBee, a ground-based robot, uses a soft-tipped end-effector to spread pollen on bramble flowers [5]. In the realm of drone-based pollination, multiple proof-of-concept drone-based pollination systems have also been researched, but most lack the intelligence and autonomy that is necessary for precise and effective robotic pollination [4,6]. Chechetka et al. proposed a gel to transport and deliver pollen, but lacked the precision needed to efficiently pollinate even large flowers [4]. Craigie et al. proposed a fully autonomous drone-based pollination system. However, their project failed to successfully pollinate any due to low precision in localization and navigation [1]. The limitation of the current model of drone-based pollination lies in its imprecise navigation and localization, which is needed for a commercially viable robotic pollination system.

Successful robotic pollination also requires an end-effector tailored to the movement and precision of a robotic system. As seen in Craigie et al., an end effector that requires a high degree of precision and stabilization is not feasible for fast and effective autonomous pollination [1]. To solve this issue, a sprayer-based system with two degrees of freedom allows for a drone-based pollination system to successfully pollinate flowers precisely from a wider range of locations.

Deep Learning-based object detection systems such as YOLO v5 use residual learning in convolutional neural networks to classify objects with greater accuracy and range than is possible with a traditional image recognition system [7]. In the flower detection process for artificial pollination, different types of flowers in multiple stages of growth may exist in the same space. In this kind of environment, a traditional color-based flower detection algorithm would fail to detect and correctly classify the correct and mature flowers. Deep learning based systems, although more resource-intensive, are vastly superior to traditional image recognition systems in classification accuracy.

III. Problem Statement

The objective of this research was to create a simulation testbed for the development of an autonomous robotic pollination system for use in a commercial tomato greenhouse environment. Due to the constraints of a greenhouse environment in which GPS navigation is denied, corridor spaces are narrow, and dense vegetation results in highly variable terrain, any successful drone-based robotic systems require extremely precise robotic localization, mapping, and navigation. Simulating the robotic system in a model greenhouse environment allows for more focus on the software framework and image recognition components rather than on real-life implementation, allowing for proof-of-concept end-to-end systems to be developed more quickly and effectively.

This system primarily focuses on a pollen delivery method, which is only one of the parts needed for a fully autonomous pollination system. A pollen collection system which can be adapted to the needs of the autonomous pollination system is proposed but not focused on in this research.

IV. Background

Pollination refers to the transfer of pollen from the stamen to the stigma of the same species of plant to allow for fertilization.

Autonomy refers to the ability of a robotic system to operate without human input during a set period to complete a specified set of tasks.

Object Detection refers to the use of computer vision and image processing in the detection of specific instances of a certain class of object.

Visual SLAM (VSLAM) refers to the specific type of SLAM(Simultaneous Localization and Mapping) system that uses 3D vision to perform localization and mapping functions. It can be divided into monocular and binocular SLAM methods.

Robotic simulation testbed refers to a platform on which software and virtual models can be integrated for the design, experimentation, and validation of a robotic system in a simplified, replicable environment.

V. Technical Approach

A. Overview

The simulation testbed DART-Sim consists of a simulation-based approach to building, training, and validating an autonomous robotic drone system powered by ROS, Gazebo, RViz, ORB-SLAM2, MoveIt! and deep learning based object detection with YOLO v5 for precise localization, mapping, navigation, and pollination. This system, which consists of several disparate pieces of software tied together using the ROS framework, presents an end-to-end simulation of a drone-based robotic system in a greenhouse environment. Because of the structure of the simulation testbed and ROS, the software pieces are largely modular, where SLAM modules or object detection algorithms can largely be substituted with other similar slam modules. This ensures that the testbed remains flexible and capable of remaining up to date with the rapid speed of new developments in robotic simulation.

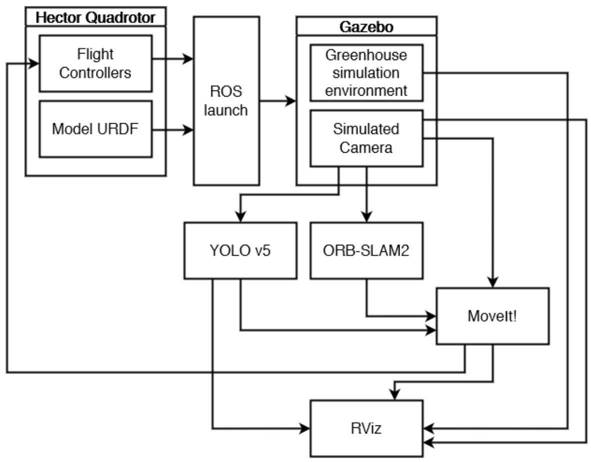

Fig. 1. DART-Sim System Diagram

B. Design of Drone

1) Software: The drone uses the Hector-Quadrotor ROS package, which includes the physical parameters and plugins for the drone to be physically simulated in Gazebo with ROS. Using the flight controller plugin and the URDF files that Hector Quadrotor provides, the package can easily be simulated in Gazebo and interact with the other software parts of the simulation testbed through ROS nodes and topics. With the forward-facing RGBD Kinect camera, the drone sends dual RGB and Depth camera image streams during a Gazebo simulation, which is used in the precise localization of the drone and mapping of its environment. The drone model is capable of full autonomy through integration with ORB-SLAM2 and MoveIt!, a motion planning framework, which includes capabilities for exploration, navigation, and smooth trajectory planning through a series of waypoints. The simulated camera is able to be visually seen in RViz through either the RGB channel, Depth channel, or with annotated feature points from ORB-SLAM2's debug image.

2) Hardware: The simulated model for the drone system was taken from the Hector Quadrotor package, which was modified for use with ROS Noetic. The base quadrotor model was modified to use a Kinect camera, which includes a monocular RGBD camera and processes the depth sensor data into a camera-generated depth point-cloud. This camera was placed horizontally along the bottom edge of the quadrotor in order to better capture the environment without image interference from the rotors blocking the camera view. The payload was placed using Blender onto the bottom of the drone, where it is able to move with a full range of motion and has access to flowers. The smaller camera mounted directly above the nozzle allows for the possibility of two-stage maneuvering in the robotic system to allow for even greater precision in flower location and direct pollination.

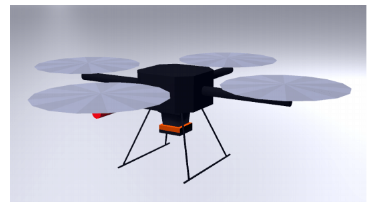

Fig. 2. Base Hector Quadrotor Model

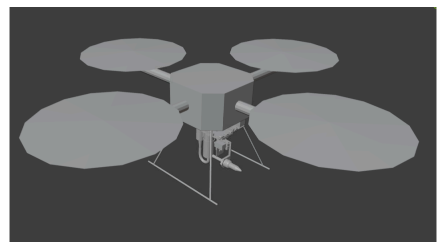

Fig. 3. Hector Quadrotor Model integrated with payload model

C. Autonomy

1) Localization and Mapping: SLAM, or Simultaneous Localization and Mapping, describes a set of algorithms that allow a robot with no previous knowledge of a space to build a map of its environment while at the same time using the map to localize itself. In ORB-SLAM2, a real-time complete SLAM system for both monocular and stereo cameras, a sparse 3D reconstruction of the environment is created. ORB-SLAM2 was chosen for its ease of integration into ROS and excellent performance in a wide range of environments. However, any similar SLAM system, such as LSD-SLAM, could be directly integrated into the SLAM navigation pipeline because of the modular design of DART-Sim [8]. Because of the performance of the Kinect camera in detecting walls and rough objects, the depth point cloud map generated by Kinect was used as a base map, which the ORB-SLAM2 point cloud can be placed on top of. The use of ORB-SLAM2 together with the Kinect depth map allows the simulated robotic system to precisely map out the dense and complex plants and flowers in a greenhouse environment. This enhanced SLAM system also has the ability to integrate Deep Learning-based object detection algorithms such as YOLO v5 into the SLAM pipeline to track the precise locations of flowers and gather information about the health or quantity of flowers in the greenhouse environment. This deep learning based algorithm can either be used in parallel with the depth point cloud and ORB-SLAM generated map or overlaid together to streamline the image processing pipeline. In DART-Sim, both methods are viable and easily implementable.

2) Object Detection: A Deep Learning approach in the SLAM pipeline with Yolo v5 allows for precise flower detection with greater accuracy than traditional hand-tuned algorithms. Traditional flower detection models locate flowers using color and size, and require more work in hand tuning the algorithm to perform well. However, using the Yolo v5 object detection system allows the system to achieve much better detection rates and precision with the downside of more complex real-time processing and longer training times. As well as higher accuracy, a deep-learning based object detection system can be expanded to include different objects and characteristics, as well as being more robust in adapting to different lighting and camera conditions. These factors are crucial for building a viable and effective object detection system for use in both DART-Sim and in the real world. In this research, a dataset of images of tomato plants was used in the training of a model to be used as part of the simulation testbed. Roboflow, a computer vision dataset generator and image annotation tool, was used to annotate, pre-process, and augment the tomato image dataset. This dataset was loaded and trained with Google Colab, where the results of the training and validation were recorded. The implementation of YOLO v5 as a deep learning-based object detection system for tomato flowers in various conditions allows for potential integration with DART-Sim.

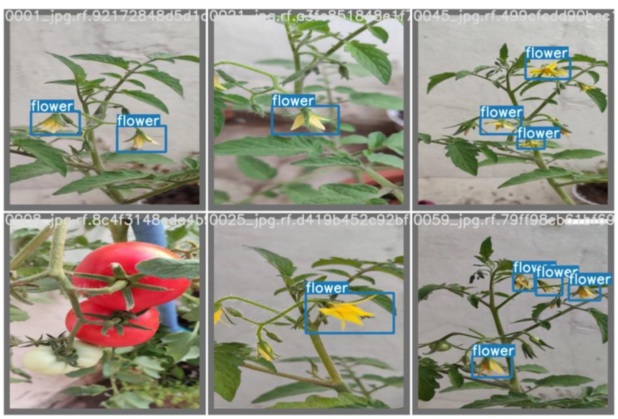

Fig. 4. Training batch of ground truth data

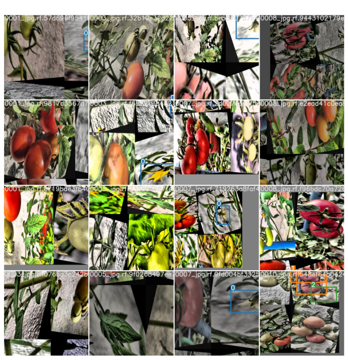

Fig. 5. Training batch of augmented tomato image images

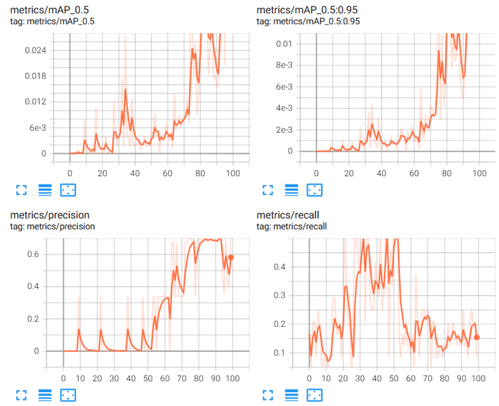

Fig. 6. YOLO v5 initial training results

3) Navigation: Autonomous navigation of the drone was achieved using the ROS Navigation Stack and MoveIt!, a motion planning framework. Using MoveIt!, a URDF model of the drone was configured and the parameters used in the path planning and exploration algorithms. MoveIt! provides a plan for the movement of joints of the robot to achieve a certain motion from this configuration file. In DART-Sim, the depth point cloud and ORB-SLAM2 were used to generate a map of the greenhouse environment, which MoveIt! used to plan and execute a planned path between a set of defined points. Hector Explorer, which uses frontiers to explore and map out a system, was used in the simulation testbed. Using MoveIt! navigation waypoints, each location of the flower can be converted into a waypoint which MoveIt! uses to plan out a smooth trajectory through all points.

4) Image Processing Pipeline: The integrated DART-Sim image processing pipeline from the simulated camera in Gazebo to the navigation system in MoveIt! is at the core of achieving autonomy of the drone-based robotic system in an unknown greenhouse environment [Fig. 7]. First, raw RGBD image data is collected from the Kinect camera mounted on the drone and split into two image topics in ROS to be fed into different algorithms [Fig. 8]. ORB-SLAM2 and the deep learning-based object detection system both use the RGB channels of the image data, but MoveIt! uses the depth point cloud directly with Octomap to map out the environment directly from the scanned depth images [Fig. 9]. Using the map built from the depth cloud data and the sparse point cloud that ORB-SLAM2 generates as seen in [Fig. 10], a MoveIt! Navigation map is built [Fig. 11].

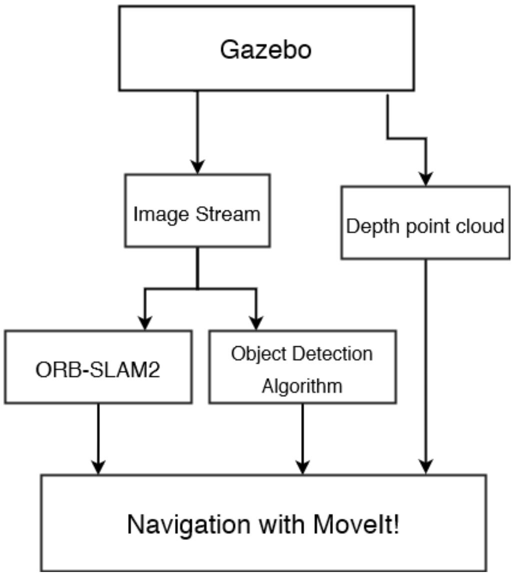

Fig. 7. Gazebo to MoveIt! Navigation pipeline

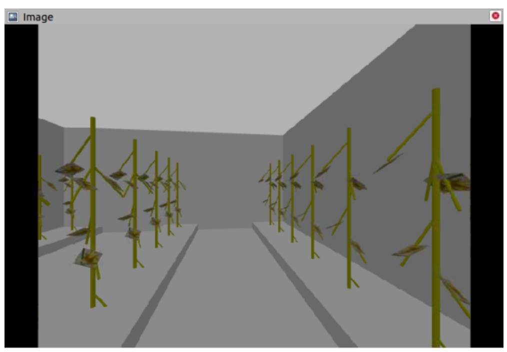
Fig. 8. Raw RGB image through simulated camera mounted on drone

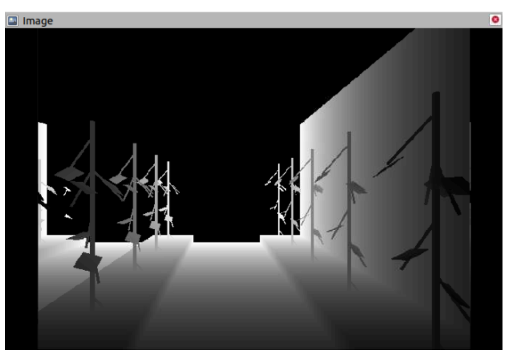
Fig. 9. Depth image through simulated camera mounted on drone

D. *End-Effector*

The payload model was designed using Autodesk Fusion. A sprayer end-effector consisting of an adjustable sprayer nozzle, diaphragm pump, and 20 mL syringe, was designed. Two degrees of freedom in the movement of the sprayer was achieved using a pan-tilt system with two micro servos with a 120 degree operation range. Above the nozzle, a Raspberry Pi camera was placed to assist in flower identification and precise adjustments of the nozzle. This monocular camera can be easily replaced with any other desired camera in the simulation testbed if needed. The proposed real-life control method for the end-effector uses a Raspberry Pi 4 to process image data from the camera and control the servos. In DART-Sim, this movement planning and image channel is accomplished through ROS, Gazebo, and MoveIt!. In this way, precise spraying in a wide range of angles can be accomplished in the simulated testbed. In order to validate the viability of the end-effector system in pollinating flowers, an initial prototype of the payload model was 3D printed and assembled, where it performed well in initial testing of spray precision and

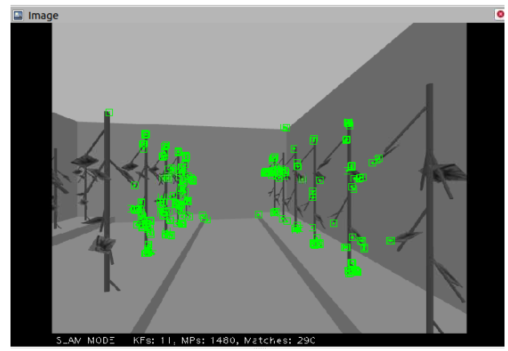
Fig. 10. Annotated ORB-SLAM debug image through simulated camera mounted on drone

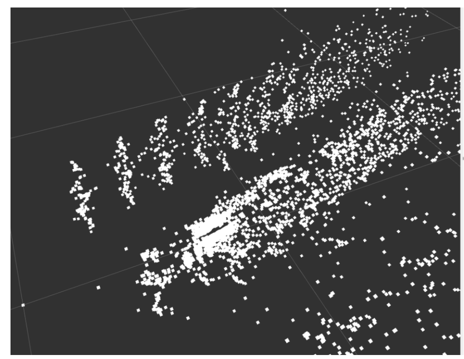
Fig. 11. Point cloud generated by ORB-SLAM2 without post-processing

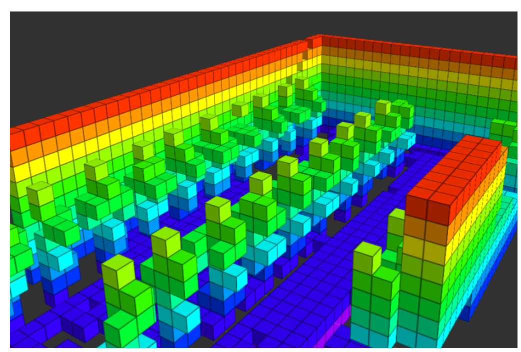
Fig. 12. MoveIt! Navigation map generated from Kinect depth point cloud data

movement. The proposed end-effector system is attached to the main drone body which allows for power and data transfer between the end-effector and main drone body.

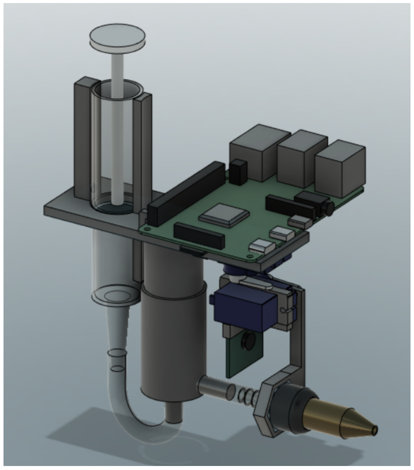

Fig. 13. Sprayer System Payload Model

E. Environment

1) Design: DART-Sim uses a drone-based pollination system which is placed into a 20x40 ft. rectangular greenhouse simulation environment modeled after a standard commercial greenhouse. It features fully grown tomato plants placed in rows at 3 ft. intervals from each other with 5 ft. of space between rows. Tomato plants with semi-realistic branching patterns and real-life images of tomato flowers were placed at regular intervals inside of the greenhouse environment. This environment was created for use in the development of a robotic drone pollination system as well providing enough realism to validate the drone system. Using the ORB-SLAM2 algorithm integrated with a deep-learning model such as YOLO v5 through a monocular RGBD Kinect camera mounted on the drone body, a detailed map of the modeled environment can be created along with the precise positions of mature flowers.

2) Simulated Model: The greenhouse model was constructed with the dimensions of a standard real-life greenhouse which was 40 ft by 20 ft, with 5 ft between rows of tomato plants and 3 ft between the center of each plant. Using these dimensions, a simplified model of the greenhouse was constructed in Autodesk Fusion, then processed in Blender and imported into Gazebo, where the model was able to be loaded in a launch file. In order to simulate the patterns of tomato plant flower and fruit growth, random points on each plant were chosen and flowers from a tomato image database were placed onto the model through Gazebo. A method for placing flower and tomato images programmatically onto each tomato plant to simulate a more natural greenhouse was pursued but not implemented in DART-Sim v1.

F. Implementation

The simulation of a drone-based robotic system was accomplished using ROS Noetic on Ubuntu 20.04 as a framework for Gazebo, RViz, MoveIt!, ORB-SLAM2, Hector Quadrotor, YOLO v5, and the various components of the simulation. Using Gazebo as the main physics simulation software, a greenhouse model was simulated alongside a fully built quadcopter model through the ROS package Hector Quadrotor. Each part of this system is essential in the construction of the simulation testbed, and the implemented system based primarily on a ROS framework allows DART-Sim to provide several services for development, testing, and validation of a drone-based robotic pollination system through the simulation system.

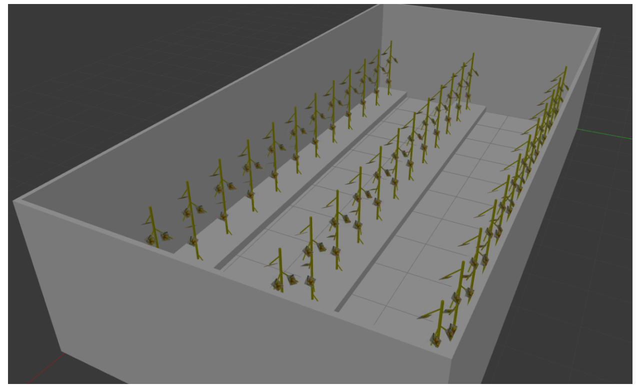

Fig. 14. The greenhouse model in gazebo

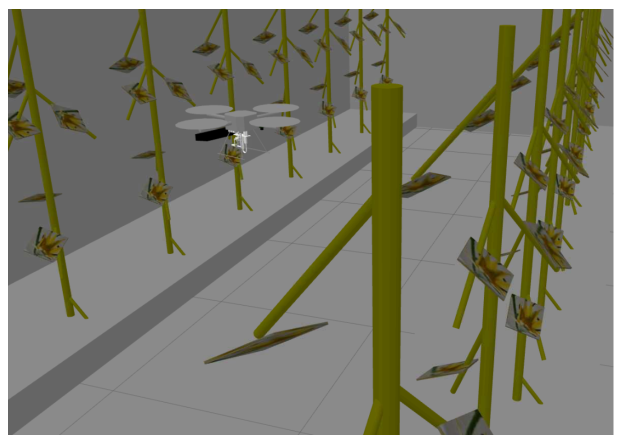

Fig. 15. Simulation with Hector Quadrotor in greenhouse environment

VI. Results

DART-Sim allows for the development of a sophisticated end-to-end drone-based pollination system. Photo-accurate tomato flowers placed at random intervals on each plant can be tracked by the drone system using a forward-facing monocular camera. The location accuracy of the system can be estimated in comparison to the known location of flowers in the simulated environment. In a potential simulation, the drone system can precisely map out and localize itself inside of a greenhouse environment using ORB-SLAM2 and the depth point cloud map generated by the Kinect camera, as well as precisely determine the location of the tomato flowers with

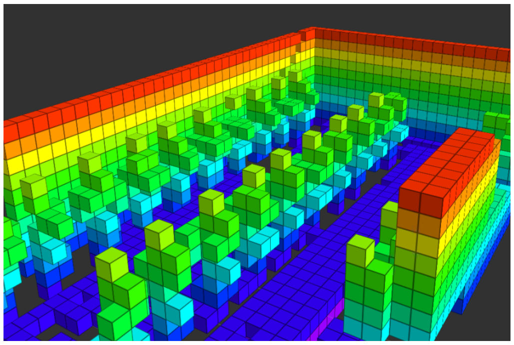

Fig. 16. MoveIt! trajectory planning and mapping

deep learning object detection algorithms such as YOLO v5. The system is also able to achieve full autonomy with MoveIt!, which provides a framework for the drone system to operate autonomously and use efficient trajectory plans in the dense environment of the greenhouse. For an end-to-end simulation, the payload sprayer system was incorporated into the drone as a rigid body connected to the drone and able to be used in the simulation tests, showing the viability of such a payload in a precision pollination system.

VII. Conclusion

In this paper, DART-Sim, a simulation testbed for assisting in the development of a drone-based artificial pollination robot with full autonomy for in a greenhouse environment is described and tested in a ROS-based simulation environment. With the incorporation of a deep learning approach in the navigation pipeline for fast and precise object detection with YOLO v5, the simulation testbed allows for drone-based systems to achieve greater accuracy and precision than other comparable autonomous pollination robots. Using this testbed, tomato flowers in a simulated dense greenhouse environment are able to be detected and localized with greater precision than with traditional methods. The integration of the payload into the Hector Quadrotor drone system in a way that still minimized collision with the environment and allowed for the drone system to pollinate flowers presents a ready-made tool for use in the development of a drone-based robotic pollination system. DART-Sim presents an end-to-end robotic simulation system for developing drone-based robotic pollinators, which allows it to be easily translated to real-world environments. With the increasing feasibility of robotic pollination, this paper presents the capabilities of DART-Sim, a simulation testbed for the development of the drone-based pollination systems of the future.

VIII. Future Work

In the future, the methods described in this paper may be used in the development and testing of a real-life prototype of an artificial pollination drone system. One concern in the migration to the real world is localization and navigation. ORB-SLAM2 was chosen for its real-time performance and the ability to track visual features extremely well, but other SLAM systems such as LSD-SLAM may be more suited for this greenhouse environment and increase the accuracy and precision of drone localization. Because navigation was not a main focus, a simple navigation and collision detection system was used. However, due to the complexity of a real-life greenhouse environment, it is likely that more work would be needed in order to accomplish satisfactory navigation results in a real-world environment. The development of a faster and more accurate flower detection algorithm is necessary for integration in a drone with limited computing resources, unlike the desktop computing power of the simulated system. Other avenues for further research include building more true-to-life greenhouse and plant models, expanding the capabilities of the simulated system, or tackling other vision-based agricultural problems, such as fruit disease detection, using DART-Sim as a framework. While DART-Sim presents a complete simulation testbed, there are many areas for improvement and further research in this area.

Acknowledgment

I would like to thank Professor Yutao He and Professor Zachary Dodds for their invaluable advice and encouragement during this research project. I would also like to thank Mr. Andrew Hamilton and Ms. Carissa Deranek for their continued support.

References

[1] C. Craigie, Z. Seker, P. Guglielmino, and N. Stallings, "Autonomous Drone Pollination," Worcester Polytechnic Institute., 2021.

[2] J. R. Reilly, D. R. Artz, D. Biddinger, K. Bobiwash, N. K. Boyle, C. Brittain, J. Brokaw, J. W. Campbell, J. Daniels, E. Elle, J. D. Ellis, S. J. Fleischer, J. Gibbs, R. L. Gillespie, K. B. Gundersen, L. Gut, G. Hoffman, N. Joshi, O. Lundin, K. Mason, C. M. McGrady, S. S. Peterson, T. L. Pitts-Singer, S. Rao, N. Rothwell, L. Rowe, K. L. Ward, N. M. Williams, J. K. Wilson, R. Isaacs, and R. Winfree, "Crop production in the USA is frequently limited by a lack of pollinators," Proceedings of the Royal Society B: Biological Sciences, vol. 287, no. 1931, p. 20200922, 2020.

[3] U. Partap and T. Ya, "The Human Pollinators of Fruit Crops in Maoxian County, Sichuan, China: A Case Study of the Failure of Pollination Services and Farmers' Adaptation Strategies," Mountain Research and Development, vol. 32, no. 2, pp. 179–186, May 2012.

[4] S. A. Chechetka, Y. Yu, M. Tange, and E. Miyako, "Materially Engineered Artificial Pollinators," Chem, vol. 2, no. 2, pp. 224–239, Feb. 2017, doi: 10.1016/j.chempr.2017.01.008.

[5] H. Williams et al., "Autonomous pollination of individual kiwifruit flowers: Toward a robotic kiwifruit pollinator," J. Field Robotics, vol. 37, no. 2, pp. 246–262, Mar. 2020, doi: 10.1002/rob.21861.

[6] K. Yousif, A. Bab-Hadiashar, and R. Hoseinnezhad, "An Overview to Visual Odometry and Visual SLAM: Applications to Mobile Robotics," Intell Ind Syst, vol. 1, no. 4, pp. 289–311, Dec. 2015, doi: 10.1007/s40903-015-0032-7.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," arXiv:1506.02640 [cs], May 2016, Accessed: Nov. 8, 2021. [Online]. Available: http://arxiv.org/abs/1506.02640

[8] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM," in Computer Vision – ECCV 2014, Cham, 2014, pp. 834–849. doi: 10.1007/978-3-319-10605-2_54.