DESIGN OF A ROBOTIC PLATFORM FOR PLANT ROOT IMAGING AND BIOMASS ESTIMATION

A Dissertation
Presented to
The Academic Faculty

By

Randall Kliman

In Partial Fulfillment
of the Requirements for the
Master of Science Degree
in the School of Engineering
Department of Electrical and Computer Engineering

Georgia Institute of Technology

December 2022

DESIGN OF A ROBOTIC PLATFORM FOR PLANT ROOT IMAGING AND BIOMASS ESTIMATION

Thesis committee:

Dr. Yongsheng Chen
Environmental Engineering
Georgia Institute of Technology

Dr. Ye Zhao
Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Callie Hao
Electrical and Computer Engineering
Georgia Institute of Technology

Date approved: December 12, 2022

ACKNOWLEDGMENTS

I would like to thank the members of my thesis committee for their help in preparation of this work. Dr. Yongsheng Chen, who has supervised me through this entire project, Dr. Ye Zhao, who helped point me in the direction of valuable resources, and Dr. Callie Hao, who has graciously agreed to be on my committee.

I would also like to thank to Dr. Patricio Vela, who gave me both technical help and the motivation to push myself, and Dr. Anqi Wu, who gave me valuable technical help on this project. I would also like to thank many of the members of the Environmental Engineering department who took me in and taught me about a discipline I had very little knowledge in. Special thanks to Tom Igou, Andrew Sharkey, and Yuankai Huang, and Mohammed El-Metwally.

This study was partially supported by the U.S. Department of Agriculture (Award No. 2018-68011-28371); National Science Foundation-U.S. Department of Agriculture (Award No. 2020-67021-31526); and National Science Foundation (Award No. 2112533). Any views and conclusions contained herein are those of the author, and do not necessarily represent the official positions, express or implied, of the funders.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Plant health and analysis has historically been a difficult undertaking due to both cost barriers and complex experimental setup. Our project focuses on analyzing the growth of lettuce plants grown in hydroponic systems. One of the main indicators of plant health is root biomass which is difficult to measure without the removal of the plant from its growth medium and destruction of the roots and leaves. In this work, we discuss the creation of an automated system for the estimation of plant root and eventually shoot biomass over its lifecycle. By using a robotics and computer vision-based system, we are able to estimate the biomass of a plant nondestructively. This system uses a robotic arm and turntable to capture 40 images at equidistant angles around a hydroponically grown plant. These images are then processed into silhouettes and used in a voxel-based volumetric 3D reconstruction method which outputs the resulting 3D model. We use a space carving method with a ray-tracing based optical correction method to create high accuracy 3D models. Analysis of this model shows that the underlying algorithm reconstructs the plant root structure accurately and gives a robust measurement of the plant root volume, which can be calibrated to indicate biomass.

CHAPTER 1
INTRODUCTION AND BACKGROUND

1.1 Background

In the field of hydroponics, plant health analysis as well as the collection of various plant related metrics is important both in the research lab and in industry. Specifically, estimations of plant responses to nutrient availability are important for determining how well plants will grow in a specific environment. Knowledge of these factors help inform how an environment can be controlled to optimize plant productivity and is the basis for fields such as Controlled Environment Agriculture (CEA).

Various studies have attempted to link root morphology and plant biomass to nutrient uptake[1], but few have attempted to do so with high temporal resolution. The main method for analyzing root nutrient uptake has been through in-situ soil studies as well as field-size studies [2]. Unfortunately, these studies have poor temporal resolution as soil cores must be taken at longer intervals due to their invasive nature. Infrequent measurement leads to a lower fidelity model of plant root growth which can lead to both overestimates or underestimates of plant nutrient application. Under-application can lead to lower crop yields, but over-application has been found to be toxic to local environments [3]. Higher temporal resolution allows for a more precise application of nutrients to plants and less environmental impact.

Historically, most root analysis methods are either invasive [4], [5] or very expensive to remain non-invasive [6][7][8][9]. As referenced above, older methods rely on taking soil cores and sieving out dirt to see how root structures are progressing [citation]. Recently, a few studies have been able to break through this cost barrier using advances in 3D model generation through the use of turntable-based 3D scanning methods [10], [11]. These stud-

ies have successfully shown that in-situ methods can be used to create high quality 3D root models. Plant imaging still poses many challenges, however. Many voxel-based methods are limited by hardware constrains such as memory or speed [cite]. Other methods focus only on the root system or the shoot system [12]. Very few methods consider the impact of optical distortion due to canister-style systems [13]. Additionally, no method focuses on the volumetric aspect of analysis, instead focusing on Root Structure Architecture (RSA).

1.2 Problem Statement

The current research being done in Georgia Tech's Daniel laboratory involves detailed analysis and experiments into nutrient uptake by lettuce plants grown in a hydroponic environment. This is mainly done by varying nutrient concentrations to columns of vertically grown lettuce plants and measuring the resulting plant matter. While this process provides useful data about plant growth, it only observes the consequences of nutrient uptake rather than the mechanisms themselves. Specifically, only leaf matter is measured because the root structures are to difficult to remove from their growth medium. Therefore, a closer look at the growth of root structures over time is needed. This project aims to look deeper into how plant root structures and plant leaf structures evolve over time as well as how these systems react to varying nutrient concentrations.

1.3 Objectives

The main objectives for this project were as follows:

- Build a root imaging system that can be used to accurately estimate root biomass and link this biomass to the nutrients in various nutrient solutions.
- Prove that the root imaging system described in [10] or something similar can be reproduced and improved by the introduction of a robotic arm.
- Examine the relationship between soil nutrients, root growth, and leaf growth.

- To generate a comprehensive dataset of plant images that can be used by others for future analysis and research.

Completing these objectives will lead to the creation of much new knowledge and prove that the underlying systems and methods that this project uses can be replicated and scaled up successfully.

CHAPTER 2
LITERATURE REVIEW

A look into existing literature shows that work on root imaging research is extensive, but studies linking root growth and nutrient uptake using 3D imaging is very limited. Most studies are interested in finding a link between root morphology and plant behavior[14][15]. Others attempt to use root imaging to quantify genotypic trends.[16]. Our project is the first of its kind which directly studies the link between root growth, leaf growth, and nutrient uptake using 3d imaging analysis. Quantitative methods for assessing root structures first appear in [5], called "Shovelomics". This paper uses a method for root measurement that requires the plant to be pulled out of the soil and cleaned in order to be directly measured using a combination protractor and ruler. From there, conclusions can be made about the health of the plant as well as other common metrics[17][16]. Other methods for plant imaging include hyper-spectral imaging, MRI [6], and X-ray tomography[18] which can give conclusions in a non-invasive manner, though are very expensive and require specialized equipment to image. A more accessible method, however, is to image plants in-situ using an RGB camera and plants grown in specialized containers [10], [11]. This method allows for a much less complex imaging rig, only using a rotating pedestal and a regular RGB camera or webcam. Here, plants are grown in gellan gum which keeps roots rigid like in soil while also providing a clear view of the roots themselves. Other approaches image the roots at very fine detail and aim to investigate roots at a cellular level. [19]. Moussus et. al develop a 3D printed method for growing plants and monitoring their growth during germination.

As for root imaging and analysis, a few solutions already exist. One method uses machine learning to pull features from the root structures [20]. Another, called Dynamic Roots, can pull root features out of 2D plant root images using voxel analysis [21]. [10]

also has a program called RootReader3D that is able to create a 3D model of the root structure and generate various statistics that can be exported. An analysis tool was developed by researchers at Georgia Tech called "GiaRoots" which is an overall solution for reconstructing and analyzing the structure of imaged roots. This approach, however, can only generate analysis of images rather than 3D reconstructing the roots[22]. GiaRoots also includes a web interface.

After taking a look at the literature, it is clear that our problem is not a wholly unique one. However, because [10] and [11] had great success in their experiments, we aim to replicate their work and find ways to improve it using modern computational methods.

CHAPTER 3
APPROACH AND METHODOLOGY

Our approach has three main aspects. First we developed a novel way to grow plants for imaging that does not use a gel or agar solution, but rather uses a liquid nutrient solution. Second, a 3D scanning system was built that included a robotic arm, a camera, a turntable, and a lighting system all supported by a Raspberry Pi. Lastly, we developed a python algorithm for processing the resulting images into a 3D reconstruction of the plant root structures.

3.1 Plant Cultivation and Growth

First, plants are sufficiently germinated before being transferred to a custom designed and 3d printed plant suspension device henceforth referred to as a ”net pot”. Seeds were first inserted into a rock wool substrate and kept moist to ensure germination would occur. Because of the variability within seeds, roughly three times the number of needed plants were germinated in a rock wool medium to ensure successful growth. Once grown, the rock wool was carefully removed using tweezers, taking extra care not to damage any existing root structures. These plants were then transferred to the net pot and held in place by two layers of black felt. The felt used must be black so that it does not create artifacts while imaging. Two layers of felt are used so that the plant does not fall over during growth and disturb the root structures.

The net pot was custom designed as a way to suspend the root structure over a 6” diameter by 6” tall acrylic canister. This size was chosen to accommodate the large and extensive root structures that lettuce plants grow. The net pot fits over the top edge of the acrylic canisters and has a few functions. First, it has a mesh grid with an opening in the middle to let the root structure grow large without interfering with the grid. As plant roots

grow, their roots grow thicker. Earlier designs did not have this opening and were prone to root constriction. Additionally, removing plants from the older design damaged the mesh and new net pots would need to be produced. The mesh top as well as the use of porous felt allows for the aeration of the water. Early experiments into plant growth had difficulty due to the lack of water aeration. Figure 3.1 illustrates how the net pots and canisters are assembled.

Figure 3.1: Growing Canisters. Left: Canister with net pot. Center: Canister and net pot with felt. Right: Canister with net pot, felt, and growing radish plant.

The water in the canisters was a custom nutrient solution that Dr. Chen's lab had been using in other experiments with nutrient solutions. Table 3.1 lists the recipe used to create the control solution. For our experiments, variations on the solutions were planned to test the plant response to changing nutrient availability and to measure nutrient uptake through volumetric information.

3.2 Imaging Setup and Dataset Collection

To image the lettuce plants, the canisters were placed on a custom designed and 3d printed turntable. This turntable is powered by a NEMA stepper motor and a DRV8825 stepper motor driver controlled by a Raspberry Pi. The imaging setup for this project is built into a 2020 aluminum extrusion box frame that allows for easy fixture and movement of experimental components. The turntable is affixed to one end of the frame and a robot

Table 3.1: Standard nutrient solution for plant growth. For some compounds, a pre-mixed water concentrate was not available, so its corresponding solid salt form was used instead. Also, compounds H_3BO_3, $MnSO_4 \cdot H_2O$, $Na_2MoO_4 \cdot 2\,H_2O$, $CuSO_4 \cdot 5\,H_2O$, and $ZnSO_4 \cdot 7\,H_2O$ were mixed together into a single "Micronutrient Mix".

Compound	Concentrate Volume (mL)	Salt Mass (g)	Concentration (g/L)
$MgSO_4 \cdot 7\,H_2O$	44.93	-	0.2461
$Ca(NO_3)_2 \cdot 4\,H_2O$	51.42	-	0.4859
NH_4NO_3	-	0.69872	0.038
KH_2PO_4	45.00	-	0.136
KNO_3	-	7.9553	0.442
H_3BO_3	4.50	-	0.0009
$MnSO_4 \cdot H_2O$	4.50	-	0.0008
$Na_2MoO_4 \cdot 2\,H_2O$	4.50	-	0.00006
$CuSO_4 \cdot 5\,H_2O$	4.50	-	0.00009
$ZnSO_4 \cdot 7\,H_2O$	4.50	-	0.0006
$C_{14}H_{21}FeN_3O_{10}$	45.17	-	0.01

arm with a mounted camera is affixed to the other end. The orientation and position of the robot arm was chosen to allow for the greatest range of motion between the front of the imaging tank and the top. Thus, the camera can effectively capture images of both the root structure and shoot structure. For each image set 80 images of the plant were taken: 40 images of the roots and 40 images of the shoots. These images are stored on a Raspberry Pi and then sent to an external computer for processing.

The imaging rig is designed as follows: A robotic arm with a camera on its end-effector is mounted one one side of a cubic metal frame using a custom designed 3D printed mount. On the other side of the frame, a light fixture with diffusion fabric is mounted. The light fixture itself is also 3D printed and was designed with a mount that allows it to snap easily onto the metal frame. In the middle, a custom designed turntable sits on a set of metal frame pieces that allow it to be adjusted horizontally within the space. Figure 3.2 shows the design in full. Small protrusions in the turntable feet allow them to slot into the grooves of the 2020 framing. Figure 3.3 shows what the setup looks like in its entirety. While there is also a set of lights behind the canister aid in silhouette generation, there is also a light fixture placed in the center of the top face of the frame which points down and shines

light on the front of the canisters. This allows the camera to more easily see the fiducial markers and aids in camera localization. Both of the lighting systems are simply small strips of NeoPixels that have been cut to required lengths. The backlight is a 2D array of NeoPixels, while the fiducial illuminator is simply one strip of NeoPixels. The entire setup is controlled by a Raspberry Pi which manages the lighting, turntable stepper motors, camera, and robotic arm control. Figure 3.4 diagrams the wiring for a custom control board known as a ”hat”.

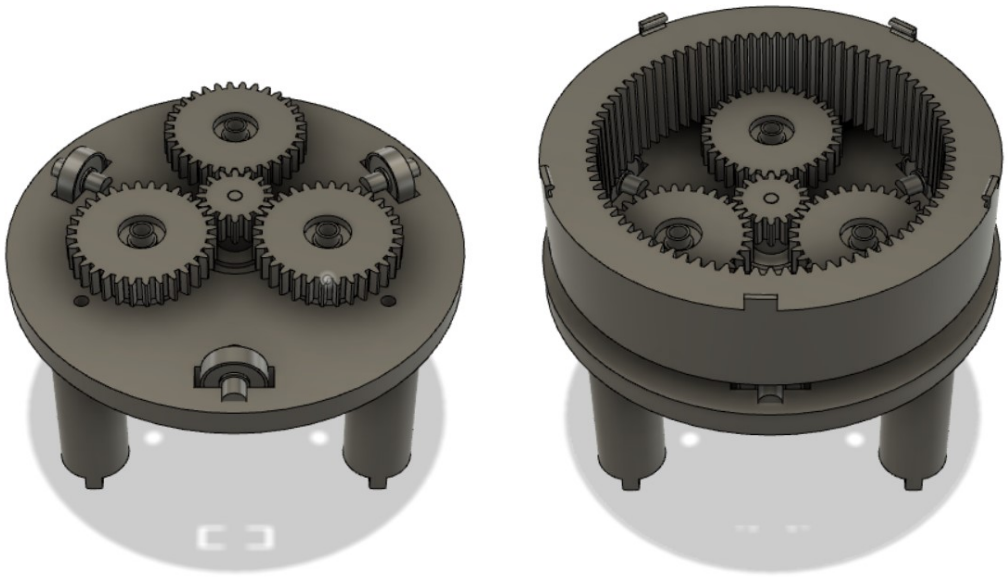


Figure 3.2: Turntable design. Left: Exposed turntable without ring gear. Right: Turntable with ring gear

The imaging process is simple. First, a plant canister is placed onto the turntable. Next a custom python script is run which turns on both the light systems. The robot arm moves into position, and begins to take pictures. After taking one picture, the turntable rotates 9° and another image is taken. This process repeats untill all images have been taken. Because we do not want the roots to move significantly between images, the turntable runs very slowly, at roughly 0.01 seconds/microstep. This translates to one photo taking roughly 4 seconds (including taking the image), meaning that the image set would take 2.6 minutes. In practice, however, this process takes closer to 6 minutes due to the time to take an image

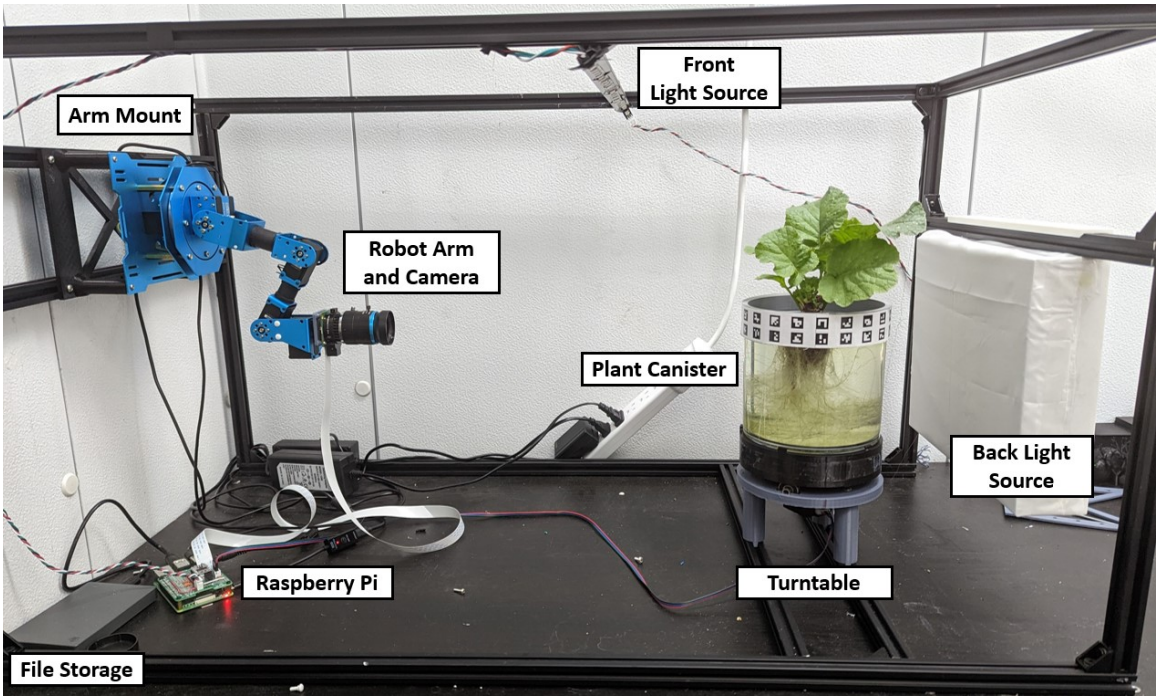


Figure 3.3: Imaging instrumentation

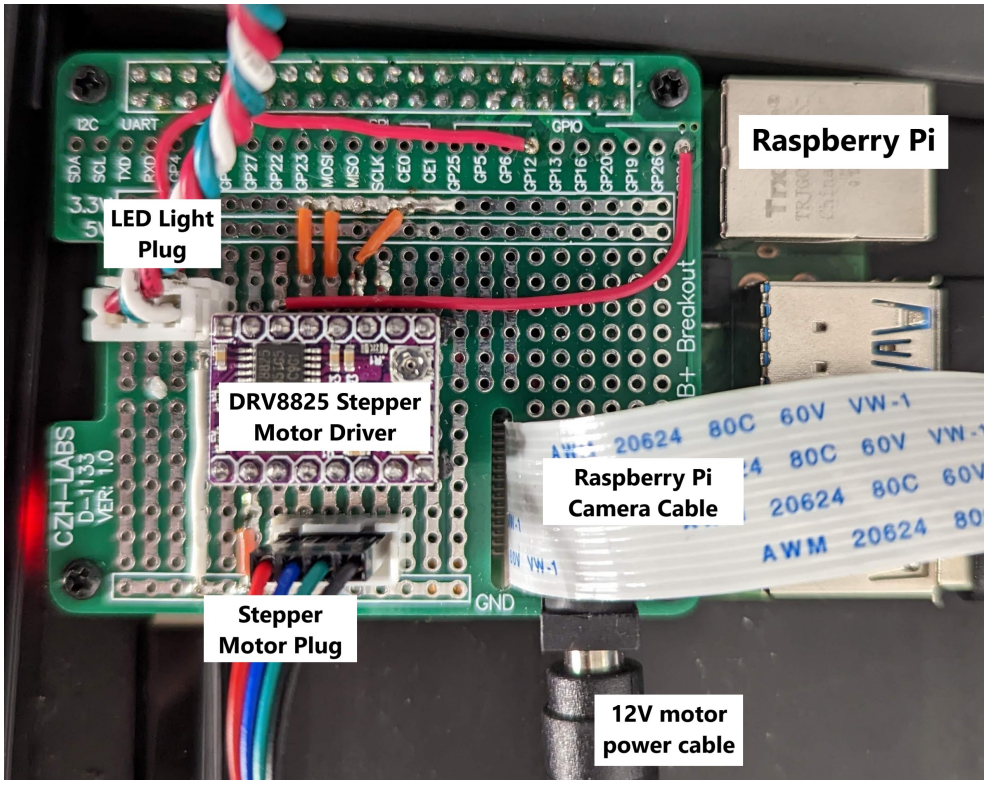


Figure 3.4: Raspberry Pi electronics connections. Note: wiring is done on a protoboard shield that sits on top of the Raspberry Pi.

and other internal hardware delays.

3.3 Robot Arm Control

To image the plants, we use a camera mounted to the end effector of a robotic arm. Previous works use a static camera to capture images, as this allows for less complex calibration needs and are less prone to movement errors. However, this only works when all of the parts of the experimental setup are precisely and accurately known. Additionally, taking images from multiple views (other than rotation around the center axis) requires additional cameras. We aimed to have this system self-contained for simplicity, so the Raspberry Pi would manage all cameras. It is difficult to multiplex cameras, so the alternative was to use a manipulator to move the camera around in space. While somewhat more complex in setup and control, we would be able to get images from any number of views by specifying a desired position and orientation. This is relatively unnecessary for taking root images, but it was helpful for fine-tuning the camera position in 3D space for the best root photos. Additionally, the manipulator allows for the imaging of the plant matter which was a potential research avenue. The end effector also allows for future expansion of the camera to possibly include cameras that can measure spectrums other than visual such as infrared.

The specific robotic arm used for this project was a LewanSoul Robotic xArm. Originally, the robot arm had a gripper affixed to its end, but this was removed in favor of a Raspberry Pi High Quality Camera. To move the camera, classic manipulator controls were used. While the arm itself has five degrees of freedom, only three of them are needed to control the position of the camera. For this application we consider the robot arm as a three link manipulator in 2D space. This allows for the specification of an x and y position as well as an orientation θ. Given desired position (x_d, y_d) and orientation θ, as well as link lengths l_1, l_2, l_3 as defined in Figure 3.5, we can calculate the robot arm angles as follows:

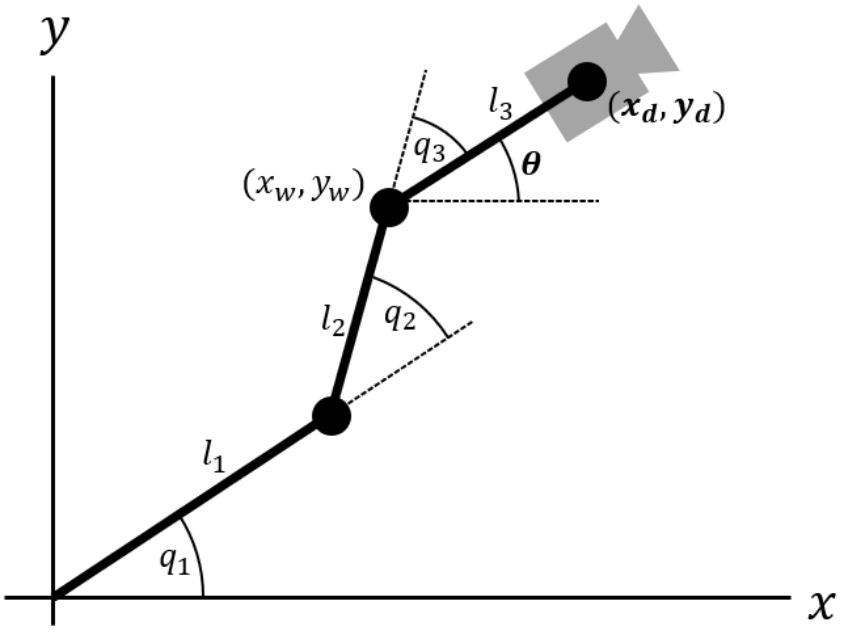


Figure 3.5: Inverse kinematics for the 3-link robotic arm.

First, we find the wrist position and distance:

$$x_w = x_d - l_3 cos(\theta) \tag{3.1}$$

$$y_w = y_d - l_3 sin(\theta) \tag{3.2}$$

$$r = ||x_w + y_w||_2 \tag{3.3}$$

Then we can find some intermediary angles:

$$\alpha = atan2(y_w, x_w) \tag{3.4}$$

$$\beta = acos(\frac{l_1^2 + r^2 - l_2^2}{2rl_1}) \tag{3.5}$$

$$\gamma = acos(\frac{l_1^2 + l_2^2 - r^2}{2l_1 l_2}) \tag{3.6}$$

Then we can find the final angles:

$$q_1 = \alpha + \beta$$

$$q_2 = \gamma - \pi$$

$$q_3 = \theta - q_1 - q_2$$

3.4 3D reconstruction

Many methods exist for 3D reconstruction, which include Structure from Motion (SfM)[23], Space Carving [24], and Neural Radiance Fields (NeRF)[25]. Unfortunately, SfM and NeRF proved difficult to use due to the nature of root and shoot morphology. SfM and NeRF work by detecting and matching features between images. Root structures, however, do not have many identifiable features and small perturbations between images can make it difficult to match the few features they do have. This creates a problem when localizing the camera positions and causes very poor results and even failure on the collected data sets. Thus, because SfM and NeRF were unavailable to use, Space Carving was determined to be the most suitable method for 3D reconstruction. Specifically, we use silhouette-based space carving[26] to achieve this. To implement this method, a library was written in python using extensive use of the OpenCV and NumPy packages.

3.4.1 Camera Localization

Most plant imaging works use static cameras with known positions within their imaging space and take pictures of plant systems at known locations. This leads to high precision camera localization and little uncertainty about camera position relative to the subjects they capture. Unfortunately, because we use a camera mounted to the end-effector of a low-cost robotic arm, this same localization certainty is not guaranteed. This problem is made worse by imperfections in the experimental design of the imaging frame and turntable. To solve this issue, a strip of ArUco fiducial markers was affixed to the outside perimeter of

each net pot. By knowing the positions of each marker in the real world as well as the position of each marker center in the captured images. This is usually done using a method called Perspective-n-Point (PnP), which has been conveniently implemented as a function in OpenCV. For PnP to work, four or more markers must be detected. The easiest way to do this was to make the marker strip two markers high so that many markers could be seen. This creates a sort of grid that can be easily found by the detector algorithm. A previous attempt only used one ring of markers, and suffered from major issues in reconstruction. Figure 3.6 compares faulty localization using only one ring of fiducial markers to proper localization using two rings of fiducial markers.

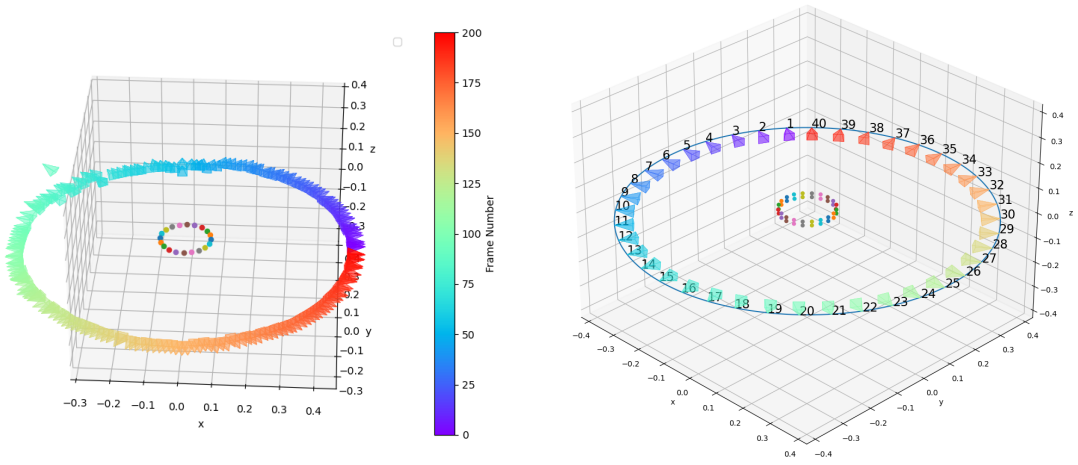


Figure 3.6: Camera Pose Estimations for each camera and the circle of best fit. The dots in the center are the positions of the fiducial markers. The left image shows faulty camera localization. where positions are not in correct alignment. Note: This localization was taken with 200 images per rotation. The right image shows proper camera alignment for 40 images. A ring of best fit is shown in blue.

This marker grid can be easily printed on any sort of conventional paper-based printer and wrapped around the net pot, guaranteeing that the markers are at precise relative positions to each other. This grid can be seen in Figure 3.1. With the camera extrinsic generated, camera projection matrices can be created by multiplying the 3x4 camera intrinsic matrix with a 4x4 extrinsic matrix. This projection matrix allows us to project any point onto the camera and check if the pixel that corresponds to that point has a value of 0 or 1.

3.4.2 Silhouettes

Many methods exist for obtaining high quality silhouettes of root structures. Options include simple thresholding, hysteresis thresholding, and normalized intensity thresholding [27]. These methods work well for very thin root structures, but fail to account for root vegetables such as radishes, potatoes, and carrots. Additionally, they require high quality lighting conditions and only work on images where the root system is the only thing visible. We propose a method of thresholding that combines the thin root preservation of hysteresis thresholding with the large structure detection of simple thresholding. First, a global threshold is found using Otsu's method. [28]. Then, we use a Sobel filter to find the edges of the root structures. The edges are then fed into a hysteresis threshold to create a binary silhouette image of these edges. For thin root structures, these edges usually cover the width of the roots, thus the root edges are equivalent to finding the roots themselves. However for larger root structures and especially in the case of the radishes that we imaged, the parent root structure tended to be very thick, meaning that our silhouettes would be missing the centers of some root structures. This would cause the root structures to only retain their thin sections. In contrast to the edges, simple thresholding has the opposite problem. Very thin root structures are translucent and cause the light from the backlight to shine through and make the roots appear brighter than other parts of the root. When applying a global threshold to an image of root structures, this property causes thin root structures to disappear while thicker root structures stay present in the silhouette. By combining the global threshold silhouette with the edge silhouette, we fill large holes and preserve thin root structures. Figure 3.7 shows how the combination of methods provides the best result. Lastly, the resultant image is passed through an OpenCV open filter to remove as many pixel-wide holes that may persist in the thin root structures. This often leads the root structures to be thicker than before, however this is usually counteracted by the agressive nature of the space carving method.

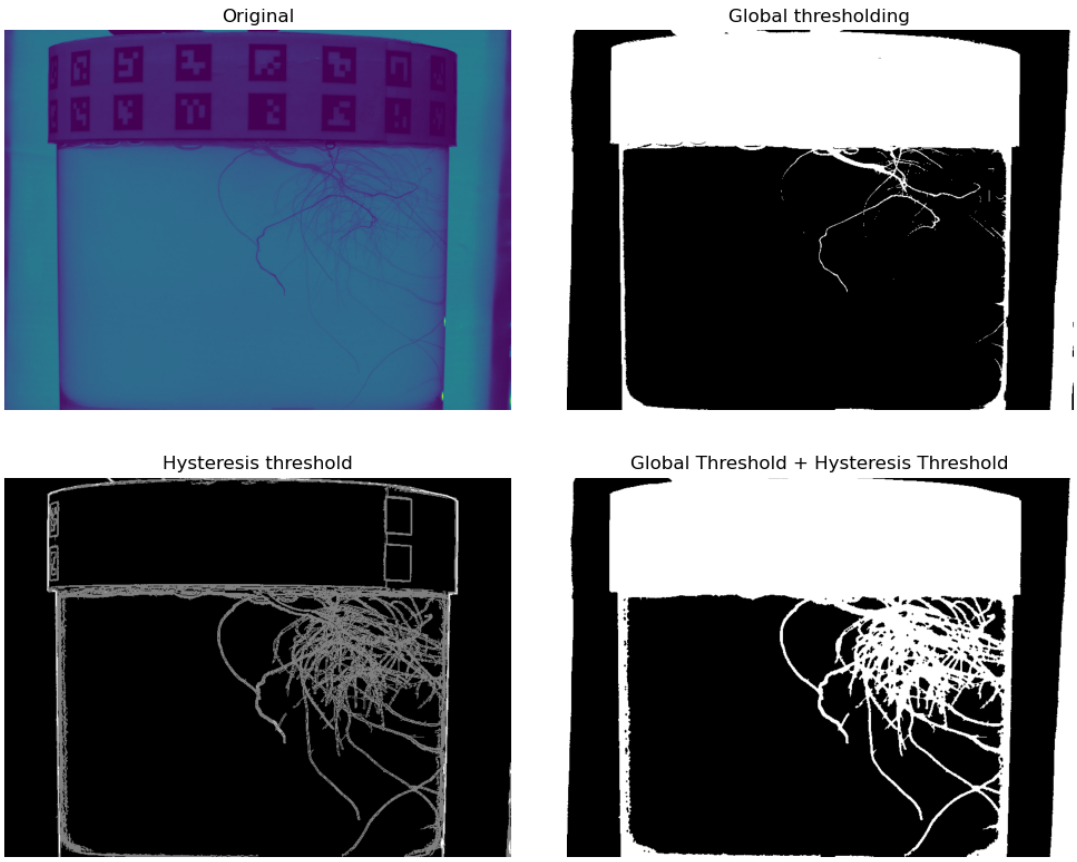


Figure 3.7: Silhouetting process to achieve full root silhouettes. Note: the whole canister is present, but using a raytracing technique, only the area inside the canister will be visible

3.4.3 Raytracing

One major challenge of imaging hydroponic systems is optical refraction due to various growth media. In our experiment, plants are grown hydroponically in acrylic canisters. While imaging, refraction causes significant distortion which leads to the loss of modeling fidelity. To remedy this, we developed a raytracing algorithm based on [13]. For each view, we use the camera extrinsics calculated in subsection 3.4.1. 480,000 (800x600) rays are cast from the camera at a simulated cylinder with the same dimensions as the growth canister. After collision and refraction with both the outside and inside surfaces of the cansiter, a set of points located on the close side of the inside cylinder surface is generated as well as their resulting ray vectors. Figure 3.8 illustrates the raycasting process. Then, for

each voxel at every view, these rays are cast towards virtual voxels and checked for which rays intersect them. Using a 3x800x600 ray tensor where the first dimension is a 3D vector and the other 2 dimensions are the shape of our images, we can create a corresponding 800x600 binary mask based on whether the ray intersected a voxel. The resulting mask can be applied using a bitwise AND operation to our original silhouette to determine the proportion of the voxel we see. This allows for a much more accurate estimation of voxel membership in our model.

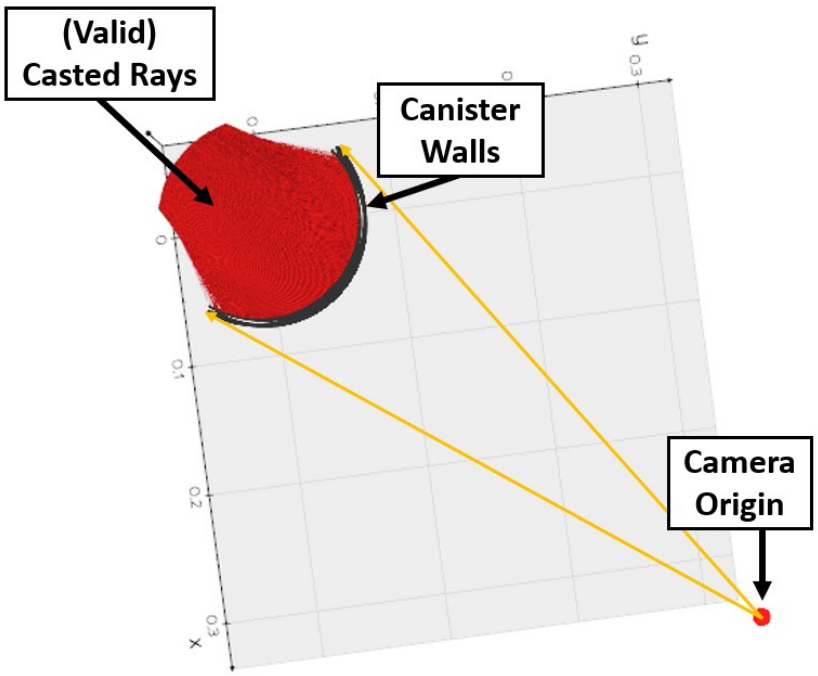


Figure 3.8: Raytracing Visualization. Rays leave the camera origin and are casted (yellow lines) towards the canister walls. Rays refract through these walls and end up as the red rays

3.4.4 Octree Space Carving

To facilitate space carving, an octree-based model was used based on Scharr's [12] paper on octree-based silhouette based space carving. The octree model has a few advantages. First, it is memory efficient for storing high quantities of spacial data. Second, it stores in-

formation about the size of each voxel which is useful for occupancy tests which will be discussed later. To summarize Sharr's method, one large voxel is created with a specified bounding box and is then projected onto a silhouette. The projected 2D voxel then is checked for how much of the silhouette it contains. If silhouette pixels fill the bounding box of the projected voxel, it is marked as part of the object. If the voxel contains some but not all of the silhouette, it is split into eight voxels. Otherwise, the voxel is marked as empty. For our algorithm, we leverage the raytracing method in subsection 3.4.3 to find voxel membership. Then each split voxel undergoes this process again until a certain desired resolution is reached. This is done from each silhouette angle until the root structure emerges. The model implemented for this project uses Sharr's "Mark-and-refine" method projects all voxels for each view at each step and marks which voxels need to be split. Then it splits them and creates a queue for the voxels to be checked. Figure 3.9 illustrates the process and the resulting high resolution voxel cloud. This method is chosen over a static grid of points due to memory and speed metrics. A static voxel grid with a predefined number of voxel points performs significantly worse due to access times and limited memory space.

One optimization that was made to the octree was to represent it as a dictionary rather than an octree. Comparing the two, we find that an octree has an average access time of $O(d)$, where d is the depth of the tree. For us, d is usually between 8 and 10, as this depth allows us to reach pixel-level resolution. On the other hand, python's dictionary has and average access time of $O(1)$ [29], making the process of iterating through voxels more efficient by a factor of almost 10.

3.5 3D model analysis

Once the 3D model has been generated, it can be analyzed. The octree implementation used allows for the access of a voxel's bounding box, can be used for precise calculation of total model volume. For each voxel box, we can calculate its volume by calculating the

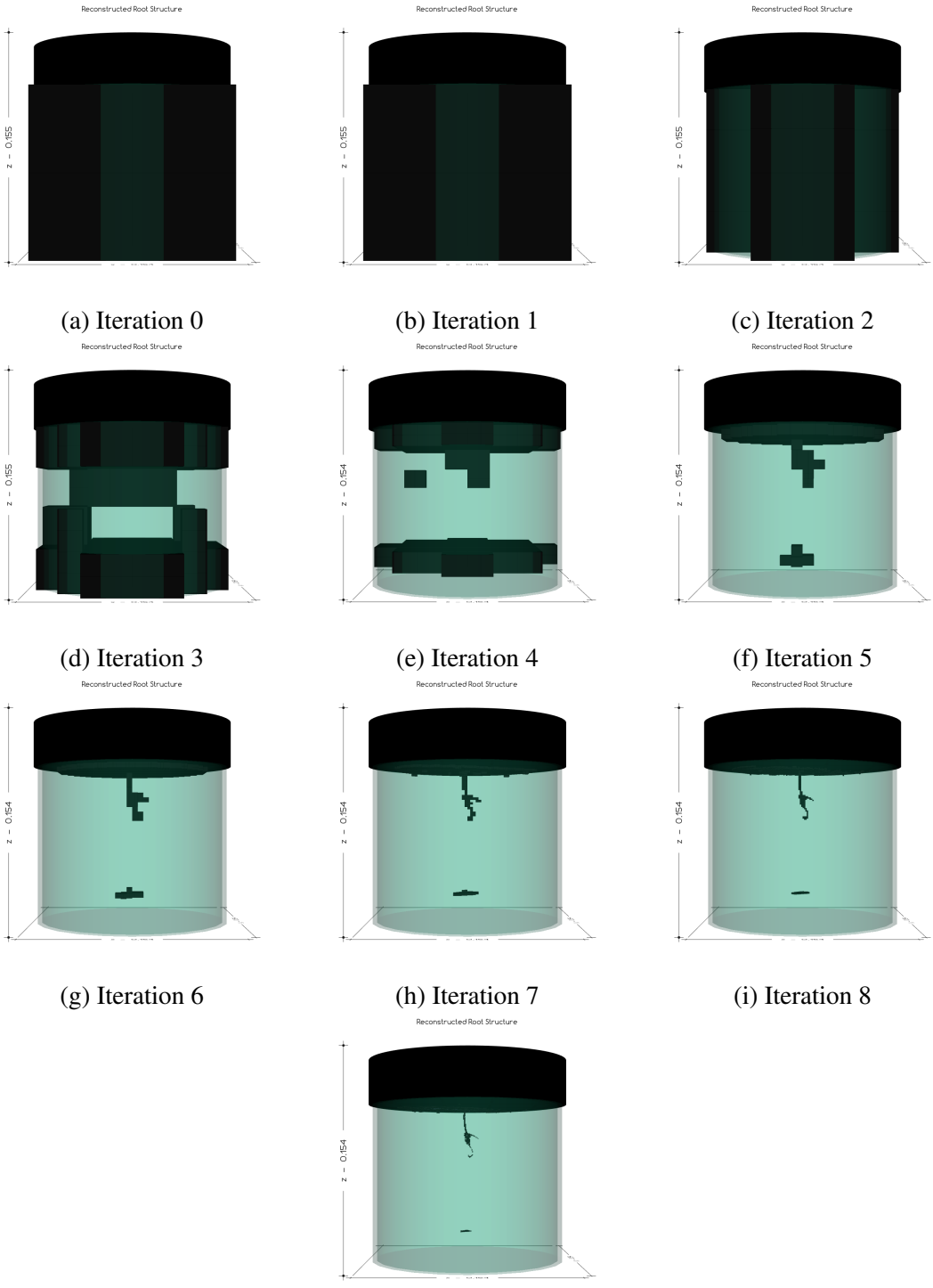
side lengths and then multiplying them together. The calculation for volume V of visual hull H is as follows:

$$V(H) = \sum_{v \in H} |(v_{x_{max}} - v_{x_{min}})(v_{y_{max}} - v_{y_{min}})(v_{z_{max}} - v_{z_{min}})|$$

As can be seen in Figure 3.10, the size should approach the real volumetric value given enough iterations.

We are also able to calculate the total dimensions of the root structure by finding the minimum and maximum voxel bounds in the visual hull in x, y, and z. For the synthetic root structure, the dimensions of the reconstructed roots were accurate to their real-world dimensions within $\approx 4\%$.

To aid in visual analysis, a visualization tool was developed using the `vedo` python library[30]. The tool includes a visualization of the canister and net pot which the root structure is displayed inside.



(a) Iteration 0 (b) Iteration 1 (c) Iteration 2

(d) Iteration 3 (e) Iteration 4 (f) Iteration 5

(g) Iteration 6 (h) Iteration 7 (i) Iteration 8

(j) Iteration 9

Figure 3.9: Progressive Space Carving. A large cube (black) is progressively split into smaller cubes until the target shape is reached.

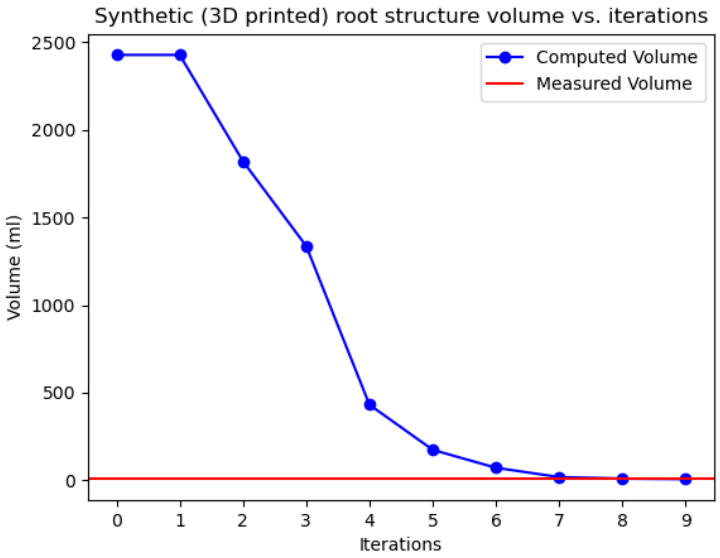


Figure 3.10: Estimated volume of a synthetic (3D printed) root structure. The real volume of the root structure is 8ml and after 9 iterations, the algorithm computes the volume to be 7.13ml. This is very close, and the error can be attributed to poor camera pose estimation as well as aggressive voxel culling. Much of the synthetic root's volume is in it's top which has been clipped slightly during reconstruction. This is caused by poor camera pose estimation as well as strange camera angles.

CHAPTER 4
EXPERIMENTATION AND RESULTS

We sought to evaluate our system's performance in a proof of concept experiment. The idea was to grow a large number of plants and create a comprehensive and longitudinal dataset that showed the algorithm could perform on a variety of different plant sizes and topologies. These shapes would range from very small and simple to very large and complex. This experiment was also intended to calibrate our nondestructive system by using the data to find a rough density value of roots and of shoots. As a result, subsequent experiments would not have to destroy their plants to measure their biomass.

4.1 Experimental Design

We oversaw a longitudinal study of 18 radish plants over the course of 30 days. Every 5 days, 3 plants were pulled off of its shelf and imaged through our imaging rig. Using a triplicate system, we are able to capture some of the variability in the plants and generate a better idea of what a plant looks like at a certain point in its growing cycle.After imaging, a plant was cut where its roots entered the canister from the felt layers. Both the roots and the shoots were weighed at this point to determine their wet masses. Then, the plants were packaged in an aluminum foil sheet, labeled, and left to sit in an oven at a constant temperature of 100° Celsius so that any water in the plants would evaporate. The plants were left in for at least 2 days to make very sure that no water was left in the plants. Foil sheets were then taken out of the oven and the dry plant matter was removed and weighed.

4.2 Experiment Results

4.2.1 Plant Data

Our experiment was run from October 1st through October 31st. Radishes were harvested on October 6th, 11th, 16th, 21st, 26th, and 31st. We observed that the radishes struggled with slow growth early on, but after October 16th, radish plants tarted to grow significantly larger. This also corresponded to significant root growth. The roots of these plants displayed a high amount of branching, as if they were looking for more nutrients. We also observed that the total percentage of water of each plant increased from roughly 92% in the beginning of the experiment and leveled off at between 94-96% for the remainder of the experiment. Roots tended to contain more water, between 96% and 97% water, while shoots tended to contain comparatively less water, between 92% and 94% water. Figure 4.1 summarizes how the plants grew over time.

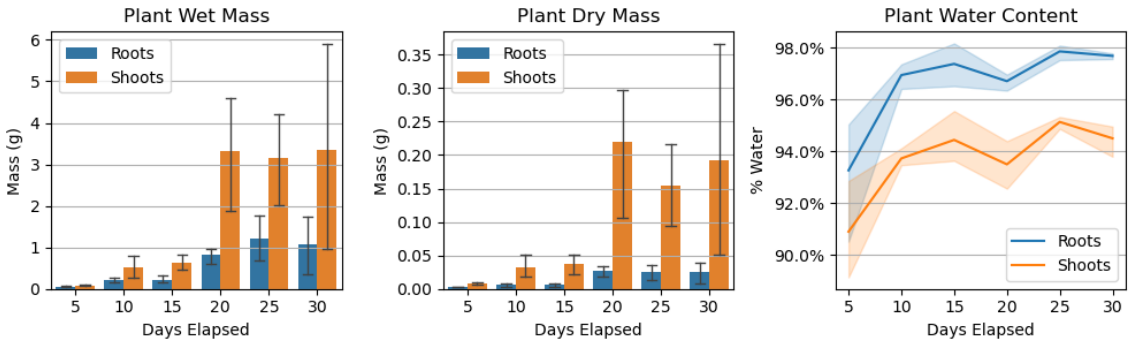


Figure 4.1: Plant growth experimental results. Left: Wet Masses, Center: Dry Masses, Right: Water Content

Algae growth was also a major issue. Nearly halfway through the experiment, algae began to appear on the top of the canisters. Soon after, this algae made its way into the canisters and around the root structures. Figure 4.2 shows how the algae grew in the net pot felt and on the inner surface of the plant canisters. Algae growth is a significant issue because it not only consumes the nutrients in the nutrient solution, but also causes occlusion in the water making it difficult to create good silhouette images. This occlusion can take

the form of either cloudy water which makes roots more difficult to see the further they are from the camera, or thick films which grow on the canister surface and create opaque occlusions. Thankfully, these larger occlusions tend to be smaller and are filtered out during the space carving algorithm.

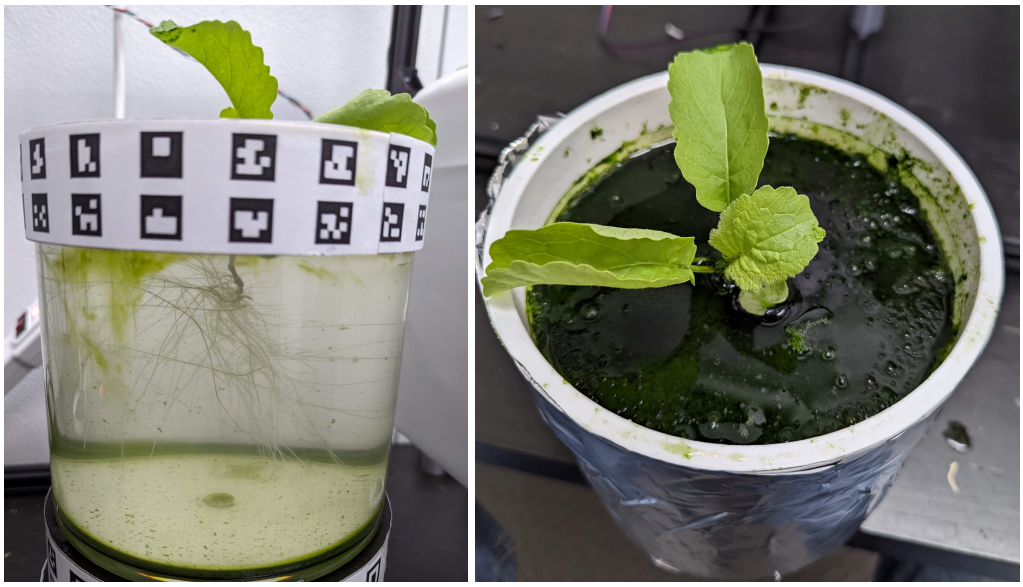
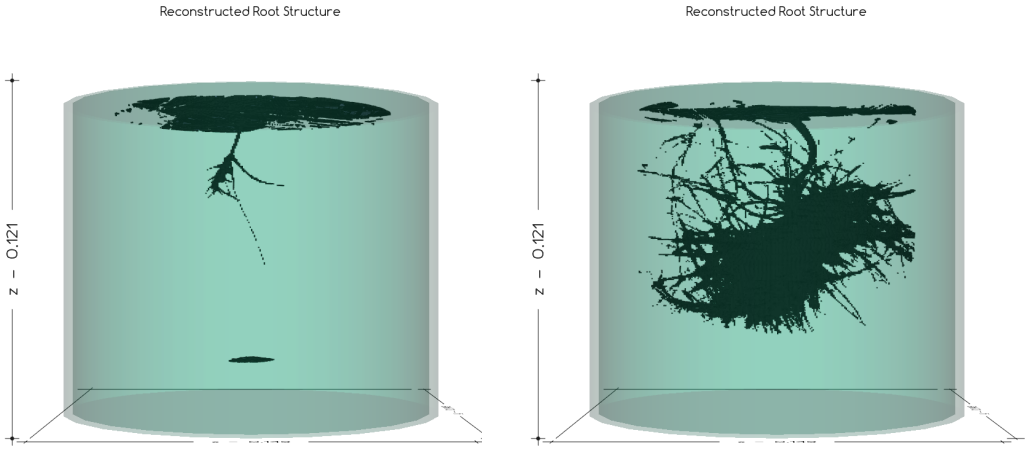


Figure 4.2: Canister algae growth

4.2.2 3D Reconstruction

Each of the 18 plant image sets were run through the 3D reconstruction algorithm. Because of the algorithm's long run time, a batch process was run on Georgia Tech's PACE super computing cluster. This allowed the processing to run continuously without interruption. Also due to the processing time, the resolution was limited to 0.27mm rather than the theoretically possible 70 microns. The resulting 3D reconstructions and volumetric estimations were of mixed quality. This quality can be measured both qualitatively, in terms of how much the reconstructed root looks like the original root structure, but also quantitatively in terms of how much volume the algorithm reports the roots to have. Because of the nature of the 3D reconstruction algorithm, quantitative results are often indicators of qualitative results.

Figure 4.3a shows an example of a good reconstruction. In this example, the root structure can be clearly seen though is missing some volume that should be there. Some voxels can be found at the bottom of the canister, and other voxels can be found at the top of the canister. These are extraneous voxels, i.e. voxels which are not part of the root structure itself, but a proper post processing program could be developed to allow a user to remove them. They are clearly defined and easy to identify as not part of the root structure. Figure 4.3b on the other hand shows a poor reconstruction. The high complexity of the root structure makes it difficult for the silhouetting algorithm to find where specific roots are and generates large blobs of roots. This error carries into the 3D reconstruction and large 3D blobs appear which have thin root structures emanating from them. Looking through the dataset, we find that 6 of the roots are qualitatively "good" reconstructions, and 12 of the roots are qualitatively "poor".



(a) Good Reconstruction (b) Poor Reconstruction

Figure 4.3: Qualitative reconstructive comparison. Left: A good reconstruction. Roots are easy to see and extraneous voxels are clearly not part of the root structure. Right: A poor reconstruction. Blobbing during silhouetting causes huge reconstructive masses and volumetric overestimations.

Another phenomenon present in these reconstructions is that reconstructed voxels never reach the full radius of the root structure. Figure 4.4 shows this phenomenon in full. This

appears to be caused by the optical distortion of the canisters, as objects at the left and right edges of the canister are not visible to the camera due to refraction. So, objects at this radius will not have full photoconsistency and are removed.

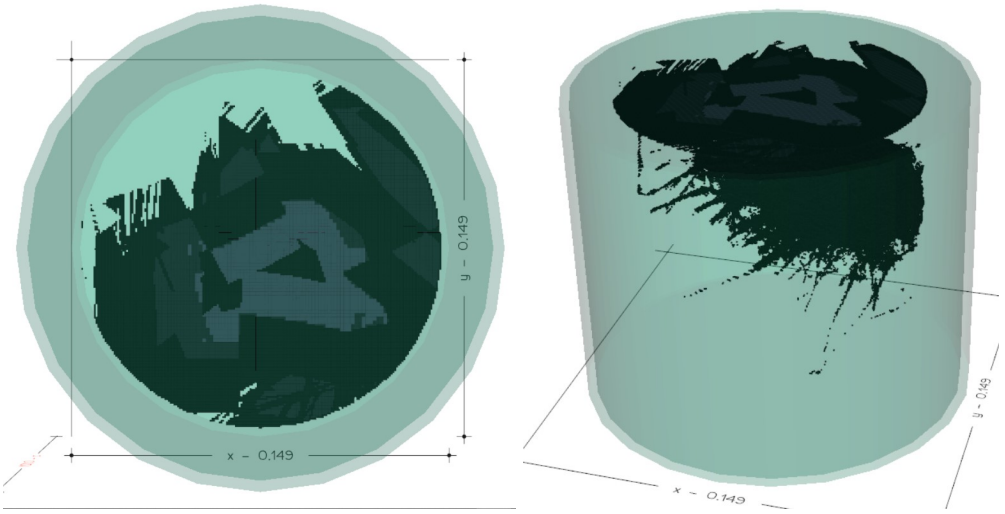


Figure 4.4: Observed voxel reduction due to optical distortion. The diameter of observed voxels is lower than the diameter of the canister even while the roots for this imageset reach the edges of the canister.

As can be seen in Table 4.1, plant root volume tends to increase over time as expected. The calculated volumes, however, become very large and inconsistent in the latter half of the experiment. Using the volumes of the root structures and the measured wet mass of these root structures, we can calculate a rough density estimate. This is done by dividing the calculated volume of our root structures by the measured root mass. Aggregating all of the data together, we arrive at an average density of **0.034 g/ml**. We can also track the density of the plants over time. Based on the wet and dry mass data, we can expect that root density will decrease over time, as the percentage of water increases over time. We also know that plant matter sinks in water, meaning that it is denser than water. Thus, a relative increase in water mass will lead to a decrease in density. Table 4.1 shows how this density changes over time.

The densities calculated in Table 4.1 show that the system does not provide fully accurate volumetric estimations for all types of plants. While the structure of the roots is often

Table 4.1: Measured root volumes, masses, and densities. Roots densities fall significantly short of the density of water, despite being made of mostly water.

Root #	Days Elapsed	Volume (ml)	Mass (g)	Density (g/ml)	Root % Water
1	5	1.790	0.035	0.019	90.52%
2	5	2.302	0.059	0.025	95.04%
3	5	5.940	0.063	0.011	94.24%
4	10	1.000	0.119	0.119	97.06%
5	10	6.370	0.240	0.038	96.42%
6	10	3.563	0.262	0.073	97.36%
7	15	7.704	0.333	0.043	97.45%
8	15	3.352	0.148	0.044	98.17%
9	15	6.655	0.187	0.028	96.52%
10	20	73.420	0.956	0.013	96.82%
11	20	69.090	0.929	0.013	96.35%
12	20	31.250	0.590	0.019	96.97%
13	25	116.740	1.772	0.015	97.97%
14	25	17.290	0.696	0.040	98.09%
15	25	46.020	1.149	0.025	97.53%
16	30	68.230	1.180	0.017	97.79%
17	30	-	0.3414	-	97.57%
18	30	-	1.7325	-	97.71%
Density of Water:				**0.997**	

qualitatively correct based on human comparison, the computed volumes are qualitatively incorrect. Extraneous voxels, aggressive voxel culling, and optical distortion cause high variability in the volume calculation and this volume tends to be a great overestimate. This overestimation leads to the density of the plant roots to be much lower than water which is incorrect for two main reasons. First, root mass has been measured to be roughly 96% water by mass, so we should expect root density to be relatively close to the density of water. Instead the computed densities are closer to 10% of water's. Secondly, roots have been observed to sink in water, meaning that they are more dense. However, the computed densities of the root structures are much less than water.

While the underlying algorithm may work for synthetic objects, a number of confounding variables in real root structures cause the methodology to create invalid results.

CHAPTER 5

DISCUSSION

5.1 Plant Cultivation

Cultivating plants using the method described in the methodology has proven to be an effective approach. Germinating 3 times amount of plants than needed gives a good selection of plants to choose from. Based on the growth of the plants, the canister-based growing system is effective and allows plants little constraint in how they grow. One issue that this system faces, however, is the lack of support that the felt provides. Two layers were chosen as a way to provide strength, however it is clear that this is not enough. As plants age, they become much larger and their weight can cause them to fall over. Figure 5.1 shows an example of this. The plant has fallen over and the root structures are growing towards the side of the canister rather than in the center or towards the bottom. This leads the roots to grow in the area of the canister which has the most optical distortion, and thus will cause more inaccuracy in the final model.

Figure 5.1: Fallen Plant. Left: Plant as seen from above, it has fallen over to its right side. Right: Plant as seen from below. As a result of falling over to the right, its roots are now concentrated on the left side of the canister.

When running the experiment, it is clear that the plant growth started off fast, but slowed

towards the end of the experiment. This makes sense, as no new nutrients were added to the nutrient solution meaning that all growth after a certain point was due to light, water, and air. This could also be seen by looking at the central roots: the normal growing time for a radish is around 30-45 days, however no radish fruit was produced. Other outcomes include root structures towards the end of the experiment that were long and thin. One hypothesis is that the radish plant produced more thin roots rather than fewer thick roots as a way of searching its growing space for nutrients. It is likely that the roots would have a completely different structure, possibly having bearing fruit, if a nutrients solution with constant concentrations existed in the canister.

Measuring the dry plant matter also had difficulties as the process for weighing the dry matter is no trivial task. This dry plant matter tends to stick to the aluminum foil and is quite difficult to remove. The process requires using a metal instrument to scrape the matter off the foil which often causes the foil to tear and matter to be lost and unaccounted for. Also, not all plant matter can be removed from the foil, and a residue is often left on the foil that may contain dried plant matter. For subsequent experiments, the following method should be used to measure this inaccuracy:

1. Weigh the aluminum foil without any plant matter
2. Weigh the aluminum foil with the plant matter
3. Dry the plant matter in the oven for 2 days.
4. Weigh the aluminum foil with the dry matter
5. Weigh the aluminum foil after scraping the most possibly dry matter off the foil.

The difference from the weights in step 1 and step 5 give a measure of how accurate the dry matter measurement is. This is a rough idea of how much matter is left on the foil and can be added back to the dry matter measurement if desired.

5.2 Instrumentation and Robotics

The instrumentation for this project performed to expectations and allowed for robust robotic control and highly accurate (micron-level) accuracy in camera localization. Robotic control is passable and the inverse kinematic model was sufficient for basic positioning of the robotic manipulator. One issue, however, was the robot's inability to correct orientation or move sideways. This is a limitation of the kinematic model, and future versions could implement the inverse kinematics of this robot to give it 6 degrees of freedom (x,y,z,θ, ω, ψ). This process is outlined in Appendix C. The robot arm also proved to be relatively unnecessary for the purposes of root imaging, as better results could have been produced by using a static camera. While this is true, a robot arm is still useful for the general instrumentation setup as the flexibility from the arm allows multiple camera angles to be taken. This could be used either for taking images of both the roots and shoots, but also could be used to generate macro images of thin root fibers. The robot arm also has a motor on the end which could be used to swap different camera types out seamlessly, allowing for multispectral imaging of a subject. The turntable and lighting systems also performed adequately for the needs of the experiment.

5.3 3D Reconstruction

The 3D reconstruction algorithm described in this work is accurate in some respects while being inaccurate in others. First, the algorithm can accurately and precisely determine where voxels are in the canister space and for an arbitrary shape with perfect silhouetting, can provide a high quality digital twin of an object. Unfortunately, not all silhouettes are perfect, and a number of issues with both the experimental design and reconstruction algorithm cause our methodology to create inaccurate results.

The first issue that must be addressed is distortion. Because voxels near the edges of the canisters are not visible, significant inaccuracy is caused in volumetric estimation. This

problem should be solved using an ”undistortion tank” where canisters are submerged in a water-filled rectangular tank so that light reaching the camera does not refract around a curved surface. Our methodology originally considered the idea and tested it, but a number of issues with it were found. Water often damaged the fiducial markers, and very specific camera angles were required such that both the root structures and the fiducial markers could be seen. Also, many studies that use this technique like [10] and [27] neglect to account for the distortion inherent to the tank. Because light is refracting into water, the rectangular tank distorts images slightly by magnifying them. Thus, raytracing would still be required to correct for this magnification in software.

The second issue to be addressed is extraneous voxels. This can partially be fixed by changing the camera angle so that the top and bottom surfaces of the canisters are not visible. This means that the root structures are seen straight on. It also would reduce the algorithm runtime as there are less useless voxels needed to be processed. However, extraneous voxels are still possible and so a user interface for removing these voxels would be useful. Vedo's toolbox[30] should support mouse selections natively, so this seems to be a realistic solution.

The last and possibly the most important issue to be addressed is aggressive voxel culling. One of the most prevalent issues in voxel based shape carving is the tendency for misalignment in camera positions to cause sections of a visual hull to be wrongfully removed. This problem is even worse in our case as thin, wire-like objects are most susceptible to wrongful removal. To fix this, we could adopt the regularized visual hull method from [27]. To adapt this into our methodology, a regularized visual hull would be computed during every refinement operation. This makes sure that all pixels in an image are covered by a voxel, even if the voxel is not photo-consistent for all 40 images. We attempted this method, however this process tends to cause a vast overestimate of volume as shown in Figure 5.2. This is likely due to the octree nature of the visual hull. Further research is required to find a better way to do this, but topology repair is very important to reaching an

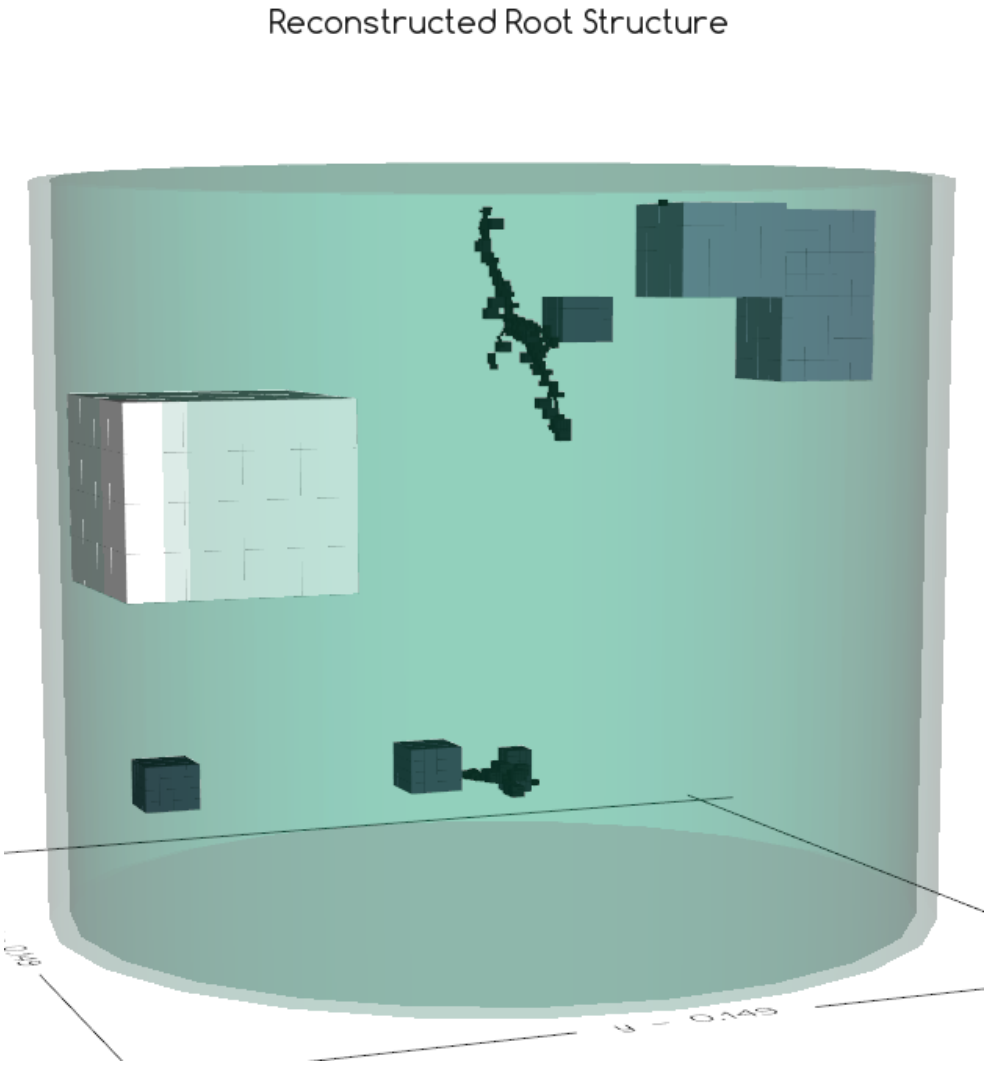


Figure 5.2: Reconstructed 5 day old radish root. Extra boxes in the image correspond to large voxels that cause massive volumetric overestimates.

accurate estimation of plant biomass.

While the methodology provided in [27] may work given more time, it has significant drawbacks when applied to an octree based model. In the octree model described in this work, re-added voxels must be iteratively space carved to avoid overestimation. This adds a significant number of new voxels, and consequently more raytracing operations. Usually by iteration 4 or 5, the topology repair algorithm takes just as much time as the space carving algorithm, meaning that the runtime has at least doubled. After iteration 5, topology repair tends to take more time than the space carving algorithm. One possible solve for this is to

use the structure of the reconstructed root structure to our advantage. If the root structure is first space carved without topology repair, we have an underestimate of root volume and much of the root structure has been lost. We can then define an area around the root structure, the volume a distance d away from the hull as our new search space. We can break this search space up into the smallest level of voxels and use the algorithm described in [27] to re-add voxels. This avoids having to process a significant number of voxels that are not likely to be part of the root's visual hull. Lopez et. al. [31] uses a similar principle, but instead further simplifies the root structure using structural properties and then refines their model using a combination of space carving and path optimization. This may be a useful direction.

Our proposed methodology also suffers from a high time to model completion which makes the claim of high-throughput less defensible. In order to generate results in a timely manner, we lowered the resolution from 70 microns down to 1/3 of a millimeter. Unfortunately the time complexity of the model operates on an exponential curve, roughly quadrupling the number of voxels at every iteration. Figure 5.3 documents the run times for each iteration of an example root structure with an expected time fit line. This time to completion is vastly greater than other approaches in the literature. [27] claims to have completed their 3D reconstructions in seconds. Their assertion, however, seems outlandish when considering that their program attempts volumetric carving using a fixed grid. Our algorithm, however, performs significantly faster than a traditional fixed voxel grid. Replacing our octree-based data structure with a fixed grid structure, the time to completion increases from roughly a day, to nearly 16 days. This means that our method is a significant speed-up which in our experiment allows reconstruction results to be available before the next set of root structures is measured.

Raytracing proved to be the greatest bottleneck for algorithm performance. The mark step which includes the raytracing computation took roughly 0.05s on an NVIDIA GTX1080 GPU and 0.017s on an NVIDIA A-100 GPU. One possible approach to improve this could

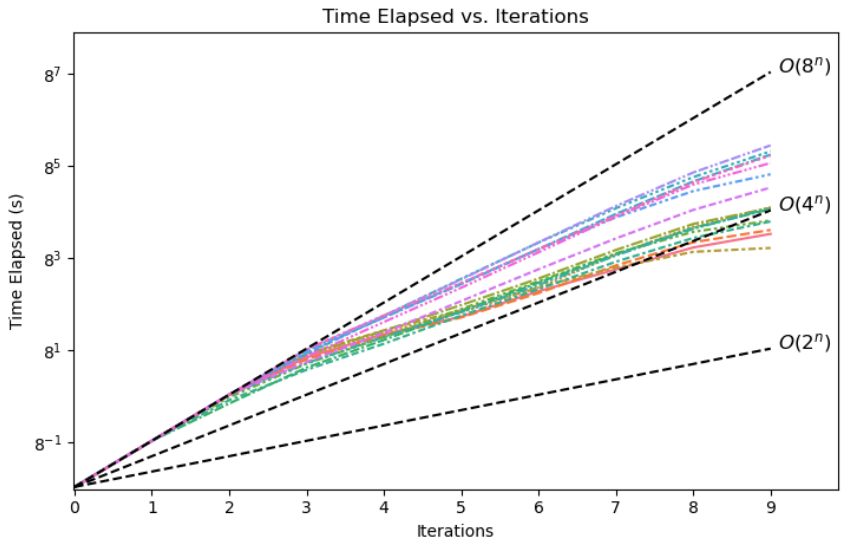


Figure 5.3: Reconstruction time analysis. The y axis is in log scale. The timings are not quite exponential, but approximately quadruple every iteration.

be finding a mathematical mapping between the voxels in the growth cylinder and pixel locations. Theoretically this can be done with perfect alignment of the camera's image plane to be tangential to cylinder's surface and for the center of the image to be perfectly aligned with the axis of rotation of the imaged cylinder. Figure 5.4 shows how this can be done. Using raycasting, we can cast our rays through the cylinder. Because of the geometry of the cylinder, the rays will go from pointing away from each other, to pointing towards each other. From here, we can draw a triangle using the edges of the cylinder's inside surface as well as the intersection point of all the rays. Knowing these values allows us to create a 3x4 projection matrix that can project nearly any point within the cylinder to this imaginary intersection point. By projecting a plane a distance l_1 away from the point, we can effectively create an imaginary undistorted camera that can see the points within the cylinder without distortion. This image is reversed around the z axis, however, so it will need to be reversed again in order to properly compare to the original images. From there, we can use Sharr's [12] method of finding the bounding box of a voxel to determine if the voxel needs to be split or not. This would significantly speed up the raytracing process, likely 100x. The main issue with this approach is that it requires extremely precise positioning of the

camera. Thankfully because we use a robot arm in this project, a control algorithm can be used to find the positioning of the camera using the printed fiducial markers and correcting it to the desired one. This method is only useful for canisters, however, and will not work with an undistortion tank due to the refraction rays pointing outwards rather than inwards.

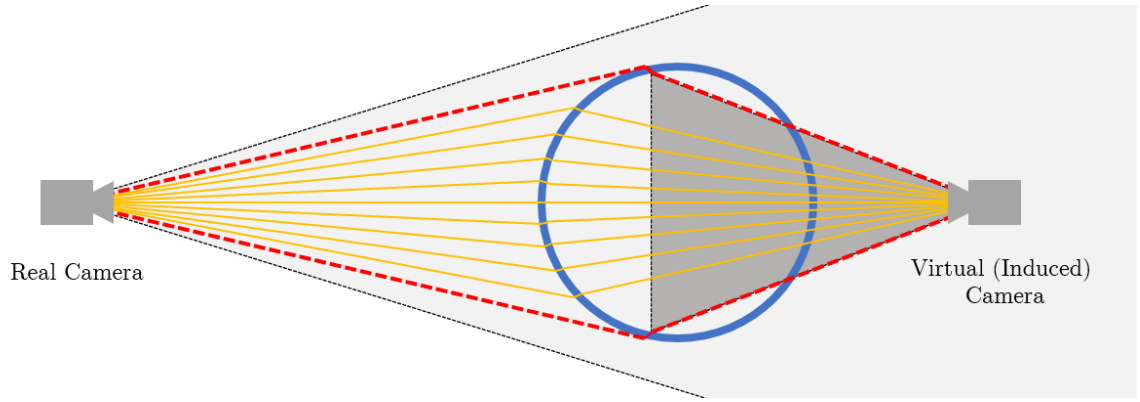


Figure 5.4: Virtual camera created by raytracing. Using the furthest valid rays, we can construct the bounding planes for the image space.

CHAPTER 6
CONCLUSION AND FUTURE WORK

For this project, we expanded upon previous methods to create an accurate and memory efficient method for creating 3D models of plant root and leaf structures. Our model outperformed [13] in resolution and expanded the methods found in [12] to also image plant root systems. We were able to create a functional 3D scanning tool using a robot arm that at the very least can generate high resolution images of plant roots nondestructively. Plants were shown to have significant growth in the designed canister system, even when subjected to nutrient deficiency. These canisters and their net pot toppers allowed for the accurate estimation of camera positions and aided in the relative accuracy of the 3D reconstruction. Using the developed algorithm, simple plant roots could be accurately reconstructed, while more complex root structures struggled. However, the algorithm was able to create a digital twin of a root structure using accurate methods, and reconstructed plants were dimensionally accurate. This is an improvement to other methods, as without raytracing their structure may be correct, but statistics like root length and splitting angle are inaccurate. Many of the methods described in this work are not novel, but their combination is, creating a new holistic method for plant root reconstruction. While our methodology did not produce accurate results on the experimental dataset, it requires only a few modifications to work properly. The underlying algorithm can work in a few ideal conditions, so generalizing it should not be too difficult a task. Optical distortion and aggressive voxel culling remain the two largest hurdles to overcome. Even so, the algorithm has great promise as a very accurate plant biomass estimator.

The computational methodology in this project resulted in a highly accurate model for each plant, but this was not without significant limitations. First, this entire system was written in Python with extensive use of the NumPy package. Python is known for being

a slower language compared to others, and our method's runtime may benefit from being rewritten in a faster language like C++. One optimization that was used to alleviate this issue was to use GPU acceleration to make the raytracing significantly faster, however this approach is not available on all computers. Future work could also focus on improving the runtime of the ray tracer, as it's accuracy means that it performs significantly slower than the method proposed in [12].

The robotic platform can also be expanded beyond its current capabilities. Currently an RGB camera is the only sensing device attached to the robot's end effector. However, a number of image-creating sensing devices could be mounted to the end effector to measure a wide array of multi-spectral phenomenon. The first and most obvious is to measure plant productivity both in the roots and shoots using the Normalized Difference Vegetation Index (NDVI)[32] which relies on infrared sensing.

While this system can create a high-resolution model of plant roots structures, this system has difficulty scaling to the size that the hydroponics industry requires. This system cannot necessarily be used for industrial systems that need to grow thousands of plants at a time, but has potential as a tool for fine tuning the nutrient solutions that would be present in large scale systems. We envision industry using a system like this in a research and development setting to estimate how a random sample of plants would react to different nutrient concentrations. Going forward from this project, there are a number of other research directions and unfinished ideas that can be pursued. On the computational side, skeletonization of these root structures can be done. This would allow for a more robust idea of how plants are growing and how plants are exploiting their environment. The system was also only used to analyze plant root structures, but can easily be modified to include 3D modeling of plant shoot structures. This would only require modifying the existing code to remove raytracing and instead use Sharr's bounding box method[12] to find voxel membership. On the plant biology side, we did not get to directly measure how plants respond to changes in their environment (specifically nutrient changes) due to time constraints. Using our sys-

tem, this should be a relatively easy task to undertake, especially since this application was the intended function of the project in the first place. An experiment for doing this was planned, however and focused on varying Potassium. Potassium was chosen for this, as it causes the least variation in plant growth. This would allow us to quantify the small changes that varying potassium would have.

While the system is not perfect, it still has a lot of potential. In the future, a wide variety of experiments could be run using this system or one like it. One interesting avenue for research includes investigating how plants interact with each other while growing as well as their impact on the environment. Co-planting has many advantages environmentally and could prove to be a useful way to mitigate environmental degradation[33][34]. However, few if any approaches exist to actually model these plant root interactions in space, whether 2D or 3D[35]. Additionally, our platform provides the ability to create a longitudinal data sets for plant root interaction which can help illuminate how plants compete and cooperate while growing together. Other experiments could include investigating the relationship between the microbiome around roots and their effect on root development and plant productivity.

Appendices

APPENDIX A

EXPERIMENTAL DATA

Table A.1: Experimental data for the first run of calibration experiments.

Date	Radish #	Root Mass (Wet)	Shoot Mass (Wet)	Total Wet Mass	Root Mass (Dry)	Shoot Mass(Dry)	Total Mass (Dry)	Root % water	Shoot % water	Total %Water
6-Oct	1	0.0348	0.098	0.1328	0.0033	0.007	0.0103	90.51724	92.85714	92.2439759
6-Oct	2	0.0585	0.0893	0.1478	0.0029	0.0097	0.0126	95.04274	89.13774	91.47496617
6-Oct	3	0.0625	0.0785	0.141	0.0036	0.0073	0.0109	94.24	90.70064	92.26950355
11-Oct	4	0.1192	0.2694	0.3886	0.0035	0.0176	0.0211	97.06376	93.46696	94.57025219
11-Oct	5	0.2399	0.787	1.0269	0.0086	0.0504	0.059	96.41517	93.59593	94.25455254
11-Oct	6	0.2618	0.5222	0.784	0.0069	0.0307	0.0376	97.3644	94.12103	95.20408163
16-Oct	7	0.3329	0.81104	1.14394	0.0085	0.0516	0.0601	97.44668	93.6378	94.74622795
16-Oct	8	0.1478	0.4775	0.6253	0.0027	0.0212	0.0239	98.17321	95.56021	96.17783464
16-Oct	9	0.1869	0.6339	0.8208	0.0065	0.0372	0.0437	96.5222	94.13157	94.67592593
21-Oct	10	0.956	3.4461	4.4021	0.0304	0.2563	0.2867	96.82008	92.56261	93.48719929
21-Oct	11	0.9287	4.5946	5.5233	0.0339	0.2969	0.3308	96.34974	93.53807	94.01082686
21-Oct	12	0.5903	1.8825	2.4728	0.0179	0.1055	0.1234	96.96764	94.39575	95.0097056
26-Oct	13	1.7722	4.2089	5.9811	0.0359	0.2158	0.2517	97.97427	94.87277	95.79174399
26-Oct	14	0.6957	2.0083	2.704	0.0133	0.0938	0.1071	98.08826	95.32938	96.03920118
26-Oct	15	1.1488	3.228	4.3768	0.0284	0.1546	0.183	97.52786	95.21066	95.8188631
31-Oct	16	1.18	3.1363	4.3163	0.0261	0.1581	0.1842	97.78814	94.95903	95.73245604
31-Oct	17	0.3414	0.9746	1.316	0.0083	0.0511	0.0594	97.56883	94.75682	95.48632219
31-Oct	18	1.7325	5.9009	7.6334	0.0396	0.3661	0.4057	97.71429	93.79586	94.68519926

APPENDIX B

PLANT CULTIVATION PROCESS

Described and shown here is the cultivation process for growing radish or other plants in the canister-based growing system described in this work.

1. A number of plant seeds are set in rock wool to germinate them. For radish plants, only one seed is required per wool cube, however more seeds may be necessary for other plant species. Once the seeds have been set in the rock wool, the rock wool is placed in a germination box and water is poured into it. Water must also be gently poured over the top of the seeds to simulate a soft rain. This can be done by pouring water over the back of one's hand which makes the water drop onto the plants more gently than directly pouring the water on it.

2. The plants are left to sit for as many days needed. For radishes, seeds must be germinated between 4-6 days.

3. Once fully germinated, radishes are assessed to find the group of seedlings which are the most similar. These seedlings are gently removed from their rock wool and can be weighed if desired.

4. Each seedling is placed into the felt growing medium that sits atop the net pot. This is a delicate process and extra care must be taken to prevent damage to the root system. Two circular pieces of felt are placed on top of each other and a hole is cut in the center. This can be easily done by folding the felt in half and cutting a hole in the middle of the straight edge. Once the hole has been made the roots of the seedling can be carefully guided through this hole. The most effective way to do this is by using tweezers to pull the root through the hole. Next the felt can be placed on top

of the net pot and the roots let down into the canister. *Note: One must be careful to make sure the roots reach the water and go fully into the canister. If not careful, roots can become trapped between the felt and the net pot.*

5. The canister is filled with the nutrient solution. The water level is filled up and over the net pot to ensure that the felt stays moist.

6. The canister is then wrapped with aluminum foil to reduce algae growth potential.

7. Each canister is placed onto the grow shelf under lights. The lights are scheduled from 6:00 AM to 8:00PM, giving the plants 14 hours of sunlight.

Figure B.1: Germinated radish seedlings growing in rock wool. Not all seeds germinated. Roughly 60 seeds were planted, but only $\approx$ 30 were viable and of those, only 18 were chosen for the experiment

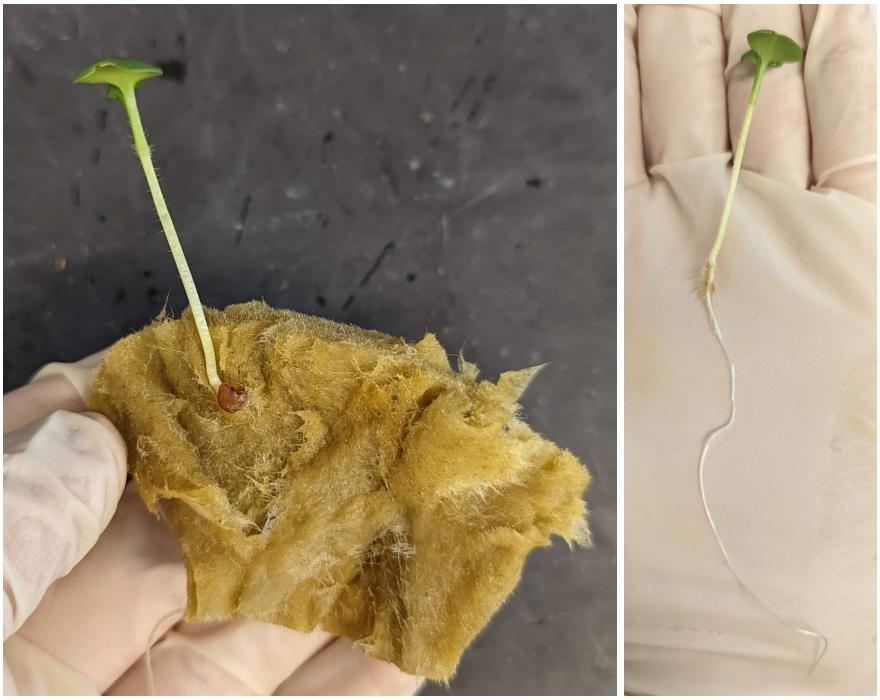


Figure B.2: Plant removal from rockwool. Rockwool should be pulled apart first, and then delicately removed to avoid damage to roots. Lightly tugging at the roots can aslo be helpful in loosening it.

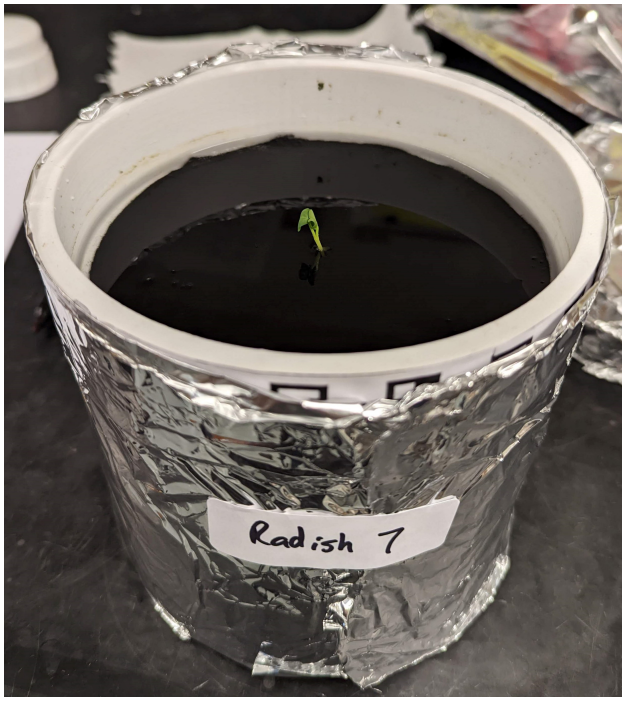


Figure B.3: Fully Prepared plant canister. Seedling is implanted into felt and placed in canister, then aluminum is wrapped around the canister and labelled.

APPENDIX C

3D XARM KINEMATICS AND ROBOT CONTROL

This chapter covers how to control the imaging arm to make the camera plane tangential to the canister and centered on the canister's rotational axis. This should allow the raytracing to collapse to a projection which theoretically can improve performance by roughly 4 times. The following method first finds the desired end effector configuration and then utilizes a similar method to the Newton-Raphson method [36] to solve for the inverse kinematics.

C.1 xArm Kinematics

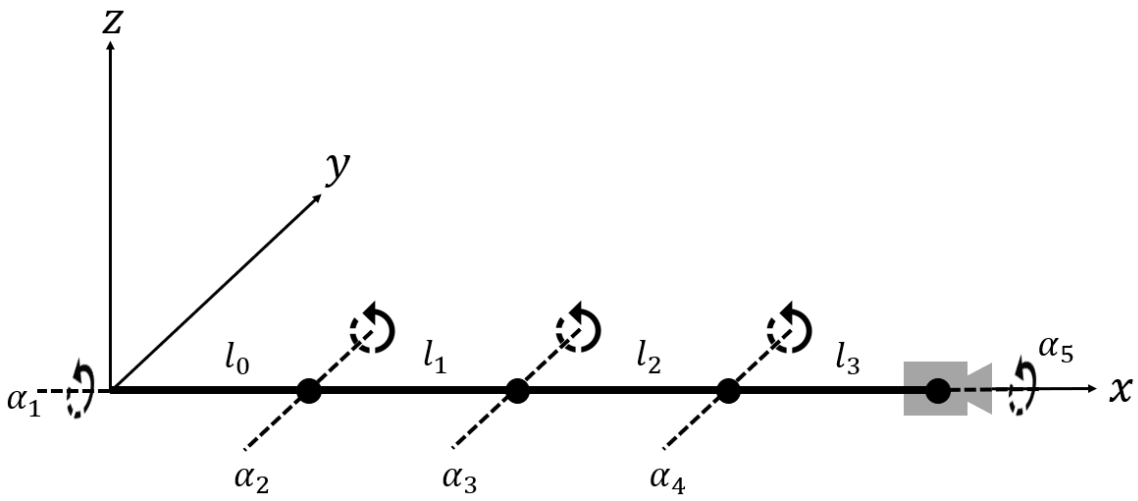


Figure C.1: xArm 6DOF Kinematic Chain. Dashed lines represent rotational axes. The camera is mounted to the end of the last link.

To start, let us define the kinematic chain for our robot. The xArm can be modeled as a 5 joint manipulator arm with 4 links. Let us define these links as l_0, l_1, l_2, and l_3. We can also define the configuration of the end-effector as $g_e(\alpha)$, where

$$\alpha = [\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5]^T$$

To model the structure of the robot, $g_e(\alpha)$ can be defined as a kinematic chain of links where each link is defined using a Denavit–Hartenberg(DH) matrix. In a DH matrix, the rotation of each link is encoded by 3x3 rotation matrix R and the displacement of each link across an axis is 3x1 vector d. Because we define the home position of our robot to be facing straight out in the x axis, all of our displacements will be in the x axis. Thus, a DH matrix will look like the following:

$$g = \left[\begin{array}{c|c} R_{3x3} & d_{3x1} \\ \hline 0_{1x3} & 1_{1x1} \end{array}\right]_{4x4}$$

To get the configuration of the end effector, all we must do is multiply the DH matrices for every link together:

$$g_e^b(\alpha) = g_1^b(\alpha_1) g_2^1(\alpha_2) g_3^2(\alpha_3) g_4^3(\alpha_4) g_5^4(\alpha_5)$$

Note: the subscript defines the frame we are interested in and the superscript defines the frame we are relative to. We can then calculate the DH matrix for each link:

$$g_1^b(\alpha_1) = \left[\begin{array}{ccc|c} 1 & 0 & 0 & l_0 \\ 0 & \cos(\alpha_1) & -\sin(\alpha_1) & 0 \\ 0 & \sin(\alpha_1) & \cos(\alpha_1) & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

$$g_2^1(\alpha_2) = \left[\begin{array}{ccc|c} \cos(\alpha_2) & 0 & 0\sin(\alpha_2) & l_1 \\ 0 & 1 & 0 & 0 \\ -\sin(\alpha_2) & 0 & \cos(\alpha_2) & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

$$g_3^2(\alpha_3) = \left[\begin{array}{ccc|c} \cos(\alpha_3) & 0 & 0\sin(\alpha_3) & l_2 \\ 0 & 1 & 0 & 0 \\ -\sin(\alpha_3) & 0 & \cos(\alpha_3) & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

$$g_4^3(\alpha_4) = \left[\begin{array}{ccc|c} \cos(\alpha_4) & 0 & 0\sin(\alpha_4) & l_3 \\ 0 & 1 & 0 & 0 \\ -\sin(\alpha_4) & 0 & \cos(\alpha_4) & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

$$g_5^4(\alpha_5) = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_5) & -\sin(\alpha_5) & 0 \\ 0 & \sin(\alpha_5) & \cos(\alpha_5) & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

Link 1 rotates around the x axis and is displaced by link length l_0. Link 2 rotates around the y axis anchored at the end of Link 1 and is displaced by link length l_1. Link 3 rotates around the y axis anchored at the end of Link 2 and is displaced by link length l_2. Link 4 rotates around the y axis anchored at the end of Link 3 and is displaced by link length l_3. Link 5, the end effector, rotates around the x axis anchored the end of Link 4.

Now that the robot's forward kinematics have been defined, we can look at the inverse kinematics. This will let us control the end effector to any position within the robot's reach, also known as its workspace. There are a number of methods to do this including Pieper's method [37], and the inverse Jacobian method. The inverse Jacobian solution is more straightforward, and therefore the method we will use. The Jacobian J as defined in a robotics context allows us to correlate $\dot{x}$, the velocity and twist of the end effector, with $\dot{\alpha}$, the change in the robot's actuator angles. The equation is as follows:

$$\dot{x} = J\dot{\alpha}$$

This is useful, however we want to do the opposite of this: obtain changes in actuator angles according to a change in position/twist.

$$\dot{\alpha} = J^{-1}\dot{x}$$

Now we can turn changes in velocity into changes in actuator angle, provided J is invertible. For fully actuated 6-DOF robots, this is not an issue. However, our robot is attempting to achieve 6-DOF with only 5 acutators, meaning that it is an underactuated robotic manupulator. Thus, our Jacobian will be a 6x5 matrix instead of a 6x6 and will not be invertible. Normally this would render the problem impossible, but we can utilize the pseudoinverse method, also known as the Moore-Penrose method[38] to allow us to take the inverse of our 6x5 matrix. We can define this pseudoinverse as:

$$J^{\dagger} = J^{T}(JJ^{T})^{-1}$$

Our new equation becomes:

$$\dot{\alpha} = J^{\dagger}\dot{x}$$

Thus, our Jacobian is transformed into a 6x6 matrix, and is invertible. With this, The xArm can be controlled to any arbitrary configuration from any other configuration. Because we plan to use the Newton-Raphson method, we will want to use the body Jacobian [39]. The body jacobian is defined as follows where n is the number of robot joints:

$$J = \begin{bmatrix} J_v \\ J_\omega \end{bmatrix}_{6xn}$$

Here, we define two sub-jacobians. Jacobian J_v defines the jacobian for linear velocities, and Jacobian J_ω defines the jacobian for angular velocities. J_v is defined as follows:

$$J_v = \begin{bmatrix} \frac{\partial x}{\partial \alpha_1} & \frac{\partial x}{\partial \alpha_1} & \cdots & \frac{\partial x}{\partial \alpha_n} \\ \frac{\partial y}{\partial \alpha_1} & \frac{\partial y}{\partial \alpha_1} & \cdots & \frac{\partial y}{\partial \alpha_n} \\ \frac{\partial z}{\partial \alpha_1} & \frac{\partial z}{\partial \alpha_1} & \cdots & \frac{\partial z}{\partial \alpha_n} \end{bmatrix}_3 xn$$

Finding Jacobian J_ω is done differently. [39] explains that this jacobian is found by taking the corresponding rotational column of g. For example, we can find $\hat{\omega}_3$, the angular velocity of the 3rd link, by taking the 2nd column of the rotation matrix for g_3^b as the 2nd column corresponds to rotation around the y axis.

We can now find the Jacobian for our system. First, we must find J_v. To do this, we find our translation vector. This is done by taking the x, y, and z coordinates from g_e:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} l_0 + l_1 + l_3 \cos(\alpha_2 + \alpha_3) + l_2 \cos(\alpha_2) \\ \sin(\alpha_1)\ (l_3 \sin(\alpha_2 + \alpha_3) + l_2 \sin(\alpha_2)) \\ -\cos(\alpha_1)\ (l_3 \sin(\alpha_2 + \alpha_3) + l_2 \sin(\alpha_2)) \end{bmatrix}$$

Finding the Jacobian,

$$J_v = \begin{bmatrix} 0 & -l_3 \sin(\alpha_2 + \alpha_3) - l_2 \sin(\alpha_2) & -l_3 \sin(\alpha_2 + \alpha_3) & 0 & 0 \\ \cos(\alpha_1)\ (l_3 \sin(\alpha_2 + \alpha_3) + l_2 \sin(\alpha_2)) & \sin(\alpha_1)\ (l_3 \cos(\alpha_2 + \alpha_3) + l_2 \cos(\alpha_2)) & l_3 \cos(\alpha_2 + \alpha_3)\ \sin(\alpha_1) & 0 & 0 \\ \sin(\alpha_1)\ (l_3 \sin(\alpha_2 + \alpha_3) + l_2 \sin(\alpha_2)) & -\cos(\alpha_1)\ (l_3 \cos(\alpha_2 + \alpha_3) + l_2 \cos(\alpha_2)) & -l_3 \cos(\alpha_2 + \alpha_3)\ \cos(\alpha_1) & 0 & 0 \end{bmatrix}$$

Next, we find Jacobian J_ω. Using the method in [39], we can find all the required $\hat{\omega}_i$.

$$J_\omega = \begin{bmatrix} 1 & 0 & 0 & 0 & \cos(\alpha_2 + \alpha_3 + \alpha_4) \\ 0 & \cos(\alpha_1) & \cos(\alpha_1) & \cos(\alpha_1) & \sin(\alpha_1)\ \sin(\alpha_2 + \alpha_3 + \alpha_4) \\ 0 & \sin(\alpha_1) & \sin(\alpha_1) & \sin(\alpha_1) & -\cos(\alpha_1)\ \sin(\alpha_2 + \alpha_3 + \alpha_4) \end{bmatrix}$$

C.2 Image Plane Tangentiality

Now that we can control our robot to any arbitrary configuration, we want to find the configuration that will make the camera's image plane tangent and centered on the canister's rotational axis. This configuration will have a few constraints. First, the z axis from the camera plane must be normal to the canister's surface, and therefore orthogonal to the canister's rotational axis. Second, the camera plane must be aligned vertically with the canister's rotational axis. However, because the image plane's y axis points down, it must be negatively aligned. We can describe these constrains using the canister's basis vectors $\hat{x_r}, \hat{y_r}, \hat{z_r}$ and the camera's basis vectors $\hat{x_c}, \hat{y_c}, \hat{z_c}$. Here, the operator $< \cdot, \cdot >$ describes a dot product.

$$< \hat{z_r}, \hat{z_c} >= 0$$

$$< \hat{z_r}, \hat{y_c} >= -1$$

Since we are using this method to correct a misalignment in an existing camera configuration, the position of this configuration will stay the same. Thus, we can describe this configuration in the camera frame as a product of frame changes. First, the frame is rotated around the y axis to align $\hat{z_r}$. Then, the frame is rotated around the x axis to align $\hat{y_c}$ Next, the frame is displaced in x by the radius of the original configuration and z by the height of the original configuration. Lastly, the frame is rotated around the z axis to return the frame to its original angle relative to the canister. The process can be described by the following equations. g_d refers to the desired configuration and x_c, y_c, z_c refer to the original positions of the camera frame before configuration correction.

$$\theta = \arctan(y_c, x_c)$$

$$h = z_c$$

$$r = \sqrt{x_c^2 + y_c^2}$$

$$g_d = \begin{bmatrix} R_z(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & \begin{bmatrix} r \\ 0 \\ h \end{bmatrix} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_x(-\pi/2)R_y(-\pi/2) & 0 \\ 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & r \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & h \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} -\sin(\theta) & 0 & -\cos(\theta) & r\cos(\theta) \\ \cos(\theta) & 0 & -\sin(\theta) & r\sin(\theta) \\ 0 & -1 & 0 & h \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Now that we have the desired configuration in the canister frame, we want to find the configuration change relative to the robot, as this will help us find our $\dot{\alpha}$. We can do this by calculating the configuration change relative to the camera, and then finding this change relative to the end effector. First, we must find where the desired frame is relative to the camera frame, as the difference will become our configuration change.

$\dot{g_c^c} = g_d^c = g_p^c * g_d^p$

For each frame, its subscript refers to the frame and the superscript refers to the reference frame it is relative to. For example, g_a^b refers to frame a relative to reference frame b. g_d^p refers to the desired configuration in the canister(plant) frame. g_p^c refers to the canister in the plant frame. We do not have this explicitly, but can get it by taking $(g_c^p)^{-1}$ since g_c^p is just the camera extrinsic which we have.

With the camera frame change known, we can find the end effector change. g_c^e refers to the camera frame relative to the end effector frame. Since the camera frame is rotated in

two axes relative to the end effector, we can find g_c^e as

$$g_c^e = \begin{bmatrix} R_x(-\pi/2)R_y(\pi/2) & 0 \\ 0 & 1 \end{bmatrix}$$
$$= \begin{bmatrix} 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Because we are defining movement in the camera frame, we must use the adjoint to find this movement in the end effector frame.

$$\dot{g_e^e} = g_c^e * \dot{g_c^c} * (g_c^e)^{-1}$$
$$= g_c^e * g_p^c * g_d^p * (g_c^e)^{-1}$$

With our configuration change found, we can covert this to a change in angle. We do this by finding the Euler angles of this configuration change matrix. If our rotation matrix is

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Then the 3 Euler angles are

$$\theta_x = atan2(r_{32}, r_{33})$$
$$\theta_y = atan2(-r_{31}, \sqrt{r_{32}^2 + r_{33}^2})$$
$$\theta_z = atan2(r_{21}, r_{11})$$

We have arrived at our end effector $\dot{x}$:

$$\dot{x} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \theta_x \\ \theta_y \\ \theta_z \end{bmatrix}$$

C.3 Configuration Control and the Newton-Raphson method

. With our configuration change vector $\dot{g}_e^e$ and our Jacobian found, we can write a control algorithm based on the Newton-Raphson method to get our end effector to our desired position. This algorithm is as follows (based on the pseudocode from [36][1]:

Algorithm 1 Newton-Raphson Inverse Kinematic Control

$\alpha(0) \leftarrow \alpha_0$ from an initial configuration g_0
$\dot{x}_b \leftarrow \log g_e^e$ where $\log(\cdot)$ is the transformation of an SE(3) lie group to a lie algebra.
$v_b \leftarrow$ linear velocity components of $\dot{x}_b$
$\omega_b \leftarrow$ angular velocity components of $\dot{x}_b$
while $||v_b|| > \epsilon_\omega$ or $||v_b|| > \epsilon_v$ **do**
    $\alpha(k+1) = \alpha(k) + J^\dagger \dot{x}_b$
end while

[1]This method and a good in depth explanation video can be found at https://modernrobotics.northwestern.edu/nu-gm-book-resource/6-2-numerical-inverse-kinematics-part-2-of-2/#department

APPENDIX D

IMAGING DATASETS

The full datasets with all images of plant roots and shoots and some example reconstructions can be found at [Link]. Here is a table that links the folder names to the root numbers. Note: Radish 17 and 18 cannot be reconstructed because their camera pose estimations are faulty. However, they can still be run through a silhouetting algorithm.

Radish #	Folder Name
1	2022-10-06 11-13-37
2	2022-10-06 12-04-32
3	2022-10-06 12-27-12
4	2022-10-11 11-13-34
5	2022-10-11 11-29-08
6	2022-10-11 12-25-31
7	2022-10-15 20-58-57
8	2022-10-15 21-25-48
9	2022-10-15 21-44-28
10	2022-10-21 09-06-53
11	2022-10-21 09-25-08
12	2022-10-21 09-54-31
13	2022-10-26 14-14-56
14	2022-10-26 14-48-47
15	2022-10-26 15-24-58
16	2022-10-31 10-37-54
17	2022-10-31 10-51-07
18	2022-10-31 11-06-19

APPENDIX E

ADDITIONAL 3D DESIGNED PARTS

Here are additional parts that were designed for the project. They are not crucial to the design process, but were necessary for frame mounting.

Figure E.1: Lightbox design. This part was printed in two and uses screws to affix them together. A light panel made of a cardboard backing and neopixels is affixed to the back plane. The top section is designed as a compliant mechanism to clip onto a 2020 aluminum extrion

Figure E.2: Camera Mounting Bracket. This allows the Raspberry Pi High Quality Camera to be mounted to the end effector of the xArm.

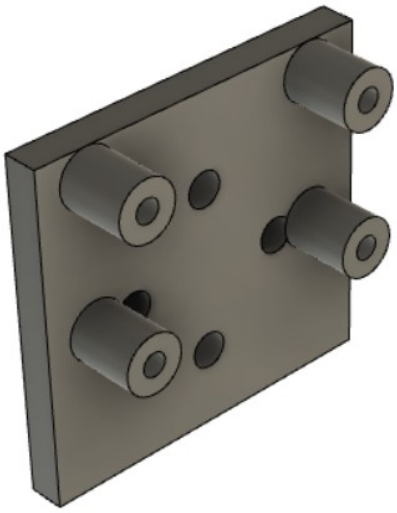
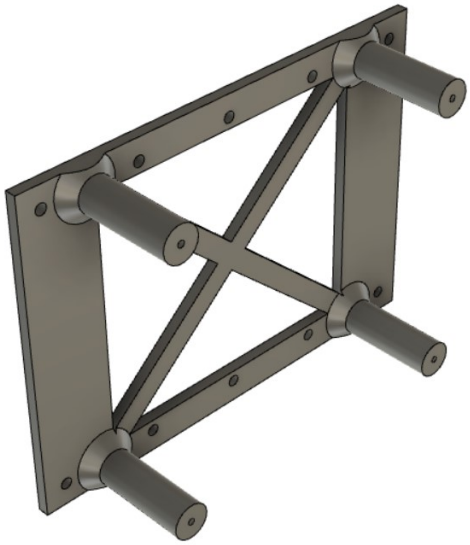


Figure E.3: Robotic Arm Mount. This allows the robotic arm to be mounted to the 2020 Aluminum frame. Rows of holes are for screws which attach the mount to the inside slots of a framing rod. Large standoffs are used to get the arm closer to the turntable. The part is mostly hollow to speed up printing time.



REFERENCES

[1] M. Kamper and A. Claassens, "Exploitation of soil by roots as influenced by phosphorus applications," *Communications in Soil Science and Plant Analysis*, vol. 36, pp. 393–402, Jan. 2005.

[2] J. J. Aguilar *et al.*, "Capturing in-field root system dynamics with RootTracker," *Plant Physiology*, vol. 187, no. 3, pp. 1117–1130, Jul. 2021. eprint: https://academic.oup.com/plphys/article-pdf/187/3/1117/43392901/kiab352.pdf.

[3] W. K. Dodds *et al.*, "Eutrophication of u.s. freshwaters: Analysis of potential economic damages," *Environmental Science & Technology*, vol. 43, no. 1, pp. 12–19, 2009, PMID: 19209578. eprint: https://doi.org/10.1021/es801217q.

[4] A. Bucksch *et al.*, "Image-Based High-Throughput Field Phenotyping of Crop Roots," *Plant Physiology*, vol. 166, no. 2, pp. 470–486, Sep. 2014. eprint: https://academic.oup.com/plphys/article-pdf/166/2/470/38114571/plphys_v166_2_470.pdf.

[5] S. Trachsel, S. M. Kaeppler, K. M. Brown, and J. P. Lynch, "Shovelomics: High throughput phenotyping of maize (zea mays l.) root architecture in the field," *Plant and Soil*, vol. 341, no. 1, pp. 75–87, Apr. 2011.

[6] D. van Dusschoten *et al.*, "Quantitative 3D Analysis of Plant Roots Growing in Soil Using Magnetic Resonance Imaging," *Plant Physiology*, vol. 170, no. 3, pp. 1176–1188, Jan. 2016. eprint: https://academic.oup.com/plphys/article-pdf/170/3/1176/37230472/plphys_v170_3_1176.pdf.

[7] G. Bodner, A. Nakhforoosh, T. Arnold, and D. Leitner, "Hyperspectral imaging: A novel approach for plant root phenotyping," *Plant Methods*, vol. 14, no. 1, 2018.

[8] S. Teramoto, S. Takayasu, Y. Kitomi, Y. Arai-Sanoh, T. Tanabata, and Y. Uga, "High-throughput three-dimensional visualization of root system architecture of rice using x-ray computed tomography," *Plant Methods*, vol. 16, no. 1, 2020.

[9] R. J. Flavel, C. N. Guppy, S. M. Rabbi, and I. M. Young, "An image processing and analysis tool for identifying and analysing complex plant root systems in 3d soil using non-destructive analysis: Root1," *PLOS ONE*, vol. 12, no. 5, 2017.

[10] R. T. Clark *et al.*, "Three-Dimensional Root Phenotyping with a Novel Imaging and Software Platform," *Plant Physiology*, vol. 156, no. 2, pp. 455–465, Mar. 2011. eprint: https://academic.oup.com/plphys/article-pdf/156/2/455/37143384/plphys_v156_2_455.pdf.

[11] S. Fang, X. Yan, and H. Liao, "3d reconstruction and dynamic modeling of root architecture in situ and its application to crop phosphorus research," *The Plant Journal*, vol. 60, no. 6, pp. 1096–1108, 2009. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1365-313X.2009.04009.x.

[12] H. Scharr, C. Briese, P. Embgenbroich, A. Fischbach, F. Fiorani, and M. Müller-Linow, "Fast high resolution volume carving for 3d plant shoot reconstruction," *Frontiers in Plant Science*, vol. 8, 2017.

[13] T. Masuda, "3d shape reconstruction of plant roots in a cylindrical tank from multiview images," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 2149–2157.

[14] J.-c. YANG, H. ZHANG, and J.-h. ZHANG, "Root morphology and physiology in relation to the yield formation of rice," *Journal of Integrative Agriculture*, vol. 11, no. 6, pp. 920–926, 2012.

[15] R. Tayade, S.-H. Kim, P. Tripathi, Y.-D. Choi, J.-B. Yoon, and Y.-H. Kim, "High-throughput root imaging analysis reveals wide variation in root morphology of wild adzuki bean (vigna angularis) accessions," *Plants*, vol. 11, no. 3, p. 405, 2022.

[16] S. Sultan, J. Snider, A. Conn, M. Li, C. N. Topp, and S. Navlakha, "A statistical growth property of plant root architectures," *Plant Phenomics*, vol. 2020, pp. 1–11, 2020.

[17] *Root trait inventory*, https://roots.ornl.gov/data-inventory.

[18] A. Piovesan, V. Vancauwenberghe, T. Van De Looverbosch, P. Verboven, and B. Nicolaï, "X-ray computed tomography for 3d plant imaging," *Trends in Plant Science*, vol. 26, no. 11, pp. 1171–1185, 2021.

[19] M. Moussus and M. Meier, "A 3d-printed arabidopsis thaliana root imaging platform," *Lab on a Chip*, vol. 21, no. 13, pp. 2557–2564, 2021.

[20] R. Yasrab, J. A. Atkinson, D. M. Wells, A. P. French, T. P. Pridmore, and M. P. Pound, "RootNav 2.0: Deep learning for automatic navigation of complex plant root architectures," *GigaScience*, vol. 8, no. 11, Nov. 2019, giz123. eprint: https://academic.oup.com/gigascience/article-pdf/8/11/giz123/30489573/giz123.pdf.

[21] O. Symonova, C. Topp, and H. Edelsbrunner, "Dynamicroots: A software platform for the reconstruction and analysis of growing plant roots," *PLOS ONE*, vol. 10, e0127657, Jun. 2015.

[22] T. Galkovskyi *et al.*, "Gia roots: Software for the high throughput analysis of plant root system architecture," *BMC Plant Biology*, vol. 12, no. 1, 2012.

[23] "The interpretation of structure from motion," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 203, no. 1153, pp. 405–426, Jan. 1979.

[24] K. Kutulakos and S. Seitz, "A theory of shape by space carving," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 1, 1999, 307–314 vol.1.

[25] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, *Nerf: Representing scenes as neural radiance fields for view synthesis*, 2020.

[26] A. Mulayim, U. Yilmaz, and V. Atalay, "Silhouette-based 3-d model reconstruction from multiple images," *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, vol. 33, no. 4, pp. 582–591, 2003.

[27] Y. Zheng, S. Gu, H. Edelsbrunner, C. Tomasi, and P. Benfey, "Detailed reconstruction of 3d plant root shape," in *2011 International Conference on Computer Vision*, 2011, pp. 2026–2033.

[28] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[29] *Time complexity - python wiki*, https://wiki.python.org/moin/TimeComplexity, Mar. 2022.

[30] M. Musy *et al.*, *Marcomusy/vedo: 2022.4.1*, version v2022.4.1, Oct. 2022.

[31] L. D. Lopez, D. Shantharaj, L. Liu, H. Bais, and J. Yu, "Robust image-based 3d modeling of root architecture," *Computer Graphics International 2011*, 2011.

[32] *Landsat normalized difference vegetation index*, https://www.usgs.gov/landsat-missions/landsat-normalized-difference-vegetation-index.

[33] D. Wu *et al.*, "Diversified effects of co-planting landscape plants on heavy metals pollution remediation in urban soil amended with sewage sludge," *Journal of Hazardous Materials*, vol. 403, p. 123 855, 2021.

[34] H. J. SCHENK, "Root competition: Beyond resource depletion," *Journal of Ecology*, vol. 94, no. 4, pp. 725–739, 2006.

[35] C. Cabal, R. Martínez-García, A. de Castro Aguilar, F. Valladares, and S. W. Pacala, "The exploitative segregation of plant roots," *Science*, vol. 370, no. 6521, pp. 1197–1199, 2020. eprint: https://www.science.org/doi/pdf/10.1126/science.aba9877.

[36] K. M. Lynch and F. C. Park, *Modern Robotics: Mechanics, planning, and Control*. Cambridge University Press, 2019.

[37] D. L. Pieper, “The kinematics of manipulators under computer control,” Ph.D. dissertation, Computer Science Department, Stanford University, 1968.

[38] R. Penrose, “A generalized inverse for matrices,” *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 51, no. 3, pp. 406–413, 1955.

[39] *Jacobian: Ros robotics*, https://www.rosroboticslearning.com/jacobian.