ACQUISITION OF LEAF POINT CLOUD DATA VIA STEREO IMAGES AND COMPUTER VISION METHODS

By

Daniel Gros

Senior Thesis in Computer Engineering

University of Illinois at Urbana-Champaign

Advisors: Professor Narendra Ahuja & John M. Hart

August 2023

Abstract

With the evolving nature of climate change and the consequential plant growth issues around the world it is becoming increasingly important to be able to create the next generation of plants that will be able to withstand the changing environment. At UIUC researchers are experimenting with genetic modifications on plants and tracking their growth to measure the effectiveness of these modifications on the plant's ability to survive in drier and hotter climates[???]. Metrics that communicate the quality of growth can be gathered from a leaf's phenotype data, including its area. This research project aims at making extraction of phenotype data easier, as in the case of leaf area, by providing researchers with an accurate and precise model of individual leaves.

Completion of the research project and application to practical software involved ____ broad fields of work. The first was training a machine learning model to be able identify individual leaves; accomplished using the open source Matterport implementation of Mask R-CNN. And the second _____.....

This research project and the resulting software produced by it demonstrate an effective workflow for acquiring leaf point clouds with birds-eye-view stereo images. These products are a step in allowing leaf phenotype measurements to be taken with ease, helping researchers track plant growth and improve their experiments. This research will provide a tool to use in helping the transition to a world with an ever changing and worsening climate.

Code will be made available.

Keywords: Mask R-CNN, Instance Segmentation, Sorghum, Geodesic, Point Cloud, Plant Phenotype

For an extensive survey of the filed we refer the reader to [16, 18]

"out of the scope of this paper" for things I don't need to cover

Commented [DG1]: Remember this is also development of a workflow

Page Break

Commented [DG2R1]: Talk more about the work that went into developing this workflow

Commented [DG3R1]: Redo without the leaf area/with work

Commented [DG4]: Add reference (from john)

Acknowledgments

I want to thank the following people for the help with my Senior Thesis:

Professor Ahuja for agreeing to be my thesis advisor and sticking with me despite the issues along the way including topic change and completion delay.

John M. Hart for being my mentor throughout the thesis project, helping guide the project and my work, and being a source of encouragement.

Jeremy Ruhter for being my contact in the greenhouse, for working with me to setup the ZED camera, and for being patient with all the issues during setup.

Tracy Johnson for providing space and plants for us to photograph and for being very informative relating to Sorghum characteristics.

Yunxuan Yang for helping with the annotation of images and Aaron Gros for helping with the visualization of point clouds.

Contents

1. Introduction

Crop production is key for our society to provide feed, food, and other resources. During the growth of a plant, the development of its functional body is affected by a dynamic process between its genotype, the performed management, and the environment [4]. Thus, plant scientists and breeders continuously assess phenotypic traits as an expression of the genotype for individual plants to generate new genetic variations of crops that show desired traits. Outside greenhouses, this in-field assessment is conventionally done manually, which is time-consuming [20]. In contrast to traditional phenotyping performed manually, vision-based systems have the potential for an objective and automated assessment with high spatial and temporal resolution, providing a method of performing the same assessment at a larger scale, in less time, and more objectively. One of such systems' objectives is to detect and segment individual leaves of each plant since this information correlates to the growth stage and provides phenotypic traits.[1] With growing concerns for the effects of climate change on crop production these studies are becoming increasingly popular and increasingly important.

……specific to my thesis, thesi statement (hypothesis here)…….

1.1 Background

The increasing severity of climate change is significantly impacting plant growth. Changing temperature, precipitation quantity and timing, atmospheric carbon dioxide levels, soil moisture, and nutrient availability can affect photosynthesis, respiration, and water use affecting plant growth [2]. As a result plant growth worldwide is at an exponential decline affecting the ability of plants to produce the expected yields. Consequently, research is being conducted in many institutions including the efforts of Professor Leakey at the University of Illinois at Urbana-Champaign to genetically modify plants to be able to withstand these changing environments. These experiments require large amounts of phenotype data to be consistently observed and measured by a team of field data collectors. The manual methods of collecting the phenotype data can slow the research down, limit the scale of the research, and even affect the research negatively via imprecise data recordings and larger measurement errors. As a result, current phenotyping methods are a bottleneck in plant gene research [3].

For the visual components of the recorded phenotype data, computer vision methods are being utilized to facilitate the data collection process. For example, one of the important metrics that communicates effective growth in plants is leaf size. Measuring leaf area is difficult under field conditions and common practice is to remove, scan, and measure leaves individually [4]. This is a slow and tedious process prone

Commented [DG5]: Am I doing area / size / width?

Commented [DG6]: Area vs width

to error. Approaches have been developed that utilize computer vision methods to automate and increase the reliability of the measurements. However, these computer-vision-based approaches remain in their infancy and can be inaccurate when it comes to leaves outside of the laboratory which are in dense clusters, overlap, and can curve and curl. This research project aims at making extraction of phenotype data easier, as in the case of leaf area, by providing researchers with an accurate and precise model of individual leaves.

Commented [DG7]: Width vs area

This research project aims to accomplish this in an accessible and noninvasive manner: without the need for specialized equipment and without the need for a significant number of people to take the measurements by hand. This can be accomplished with the use of stereoscopic images. After performing initial image processing, 2D stereo images can be used to perceive depth the same way our eyes perceive depth. With this perception of depth one can generate depth maps, point clouds, and meshes. As a result, solely with a pair of offset 2D images, it is possible to digitize the 3D structure of a cluster of leaves. The challenging part is isolating individual leaves that can be measured.

1.2 Research Goals

The aim for this research project is to enable researchers who are conducting experiments on plants that require the collection of phenotype data to perform the acquisition of data more easily and quickly.

The research goals are to define a method and develop a program that researchers can use to acquire leaf models for plants in environments with dense leaf clusters.

Questions this research will address are:

What is an effective process at isolating and modeling a leaf from stereo images?

What is the implementation for isolating and modeling a leaf from stereo images?

Commented [DG8]: John: This section should include all of what you did, including choosing greenhouse and plants, setting up camera system, obtaining the data, extracting data, etc. (Same as you did later in MethodologyOrrj)

Commented [DG9R8]: I thought maybe I would only include that which can be useful for other people to use? Like the goal wasn't to find a greenhouse room or set up the zed camera, the goal was to find and implement a process that can acquire a point cloud of an individual leaf?

Commented [GD10]: Only pointing out that the code is generic, not really specific to leaves, so it could be applied?? Or is that not a good idea?

Commented [DG11R10]: Contribution, here talk about applying state of the art equipment, state of the art computer vision methods to address problem of automating phenotyping data colleciton

Commented [DG12R10]: Expand

Commented [DG13R10]: Extract area, ...

Commented [DG14R10]: Anad "other biological investigations into leaves and properties"

1.3 Significance

With the growing effects of climate change crop yields are declining and researchers are hard at work developing new breeds of plants to thrive in the new environments. This project intends to provide these researchers with the tools to conduct this research more effectively and efficiently.

Although, this project serves as a first step in the automation of leaf phenotype data collection, the defined process and developed program can be altered for other applications that require modeling of objects. This could include other types of research: automating food harvesting, facilitating detection of large tumors, etc. This project can serve as a first step in development in the space of object modeling with stereo images.

1.4 Limitations

Although the goal of the project is to define and develop a program that can be used by plant genetic researchers to collect phenotyping data on plants certain factors in the development of the project limit the use case of the end product.

The entirety of the project was done in partnership with a research team that works on sorghum. As a result, the machine learning model used to identify individual leaves was trained on sorghum leaf data and is intended to only be accurate at identifying leaves for sorghum plants. In addition, all data used for the training consisted of healthy and well-lit leaves. Applying the model to environments outside of well-lit and healthy sorghum would require a larger dataset and further training of the model.

Commented [DG15]: John: Why is this section needed? It starts off thesis on a bad note? Want reader to be excited about what is coming next...

Commented [DG16R15]: I just followed online guides of how to format and structure a thesis, should I remove?

1.5 Structure

Chapter one has outlined the purpose of the research project and what it entails. It focuses on describing at a high level the problem the project intends to address, the method to address the problem, the potential applications of the project, and some limitations of the project.

Chapter two outlines the survey conducted to determine the most effective design choices. It describes the different methods for performing instance segmentation on leaves and the different methods for measuring object width on 3D models of objects.

Commented [GD17]: Might remove

Chapter three describes the methodology used to accomplish the project goals. This chapter steps through the setup involved for data acquisition, the process in creating, training, and using the instance segmentation model, and the code written with the aim of extracting the 3D structure of an individual leaf for future measurement of leaf aspects.

Chapter four communicates the level of effectiveness of the research project in obtaining a precise and accurate result. It defines the results that pertain to the output of the Mask R-CNN model and the output of the leaf model acquisition process.

Chapter five analyzes the results outlined in chapter four. It describes the findings of the project and the work that can be done to continue the project and answer further research questions.

Chapter six outlines the answers to the research questions and how the research project contributed to the technical space. This final chapter also describes the opportunities for further research utilizing the research from this project and extending its functionality and application.

2. Literature Review

During the establishment of the structure of the project, two areas needed surveying to inform the direction of the project. First was determining the appropriate method for performing instance segmentation on a leaf: 3D vs 2D and which algorithm within the appropriate category was best suited for the task. Second revolved around the methods in which a 3D model of a leaf could be utilized to acquire area measurements for the project's overarching goal of acquiring leaf phenotype data like area via a 3D model of a leaf.

2.1 Instance Segmentation on Leaves

A survey into the current instance segmentation methods for leaves resulted in two common approaches: 3D training and inference on point clouds [3], [5], [6] and the traditional training and inference on 2D images [4], [7], [8].

2.1.1 3D Point-Based Model for Instance Segmentation on Leaves

Current point cloud instance segmentation approaches on leaves involve extending a deep neural network that consumes point cloud data. One of the foundational approaches is called PointNet developed by C Qi, H Su et al. This network provides a unified approach able to perform object classification, part segmentation, and semantic segmentation with point clouds. The architecture is visualized in Figure 1. The classification network takes n points as input, applies input and feature transformations, aggregates point features by max pooling, which results in output scores that are classification scores for k classes. The segmentation network extends the classification network, concatenates global and local features, and outputs per point scores [9].

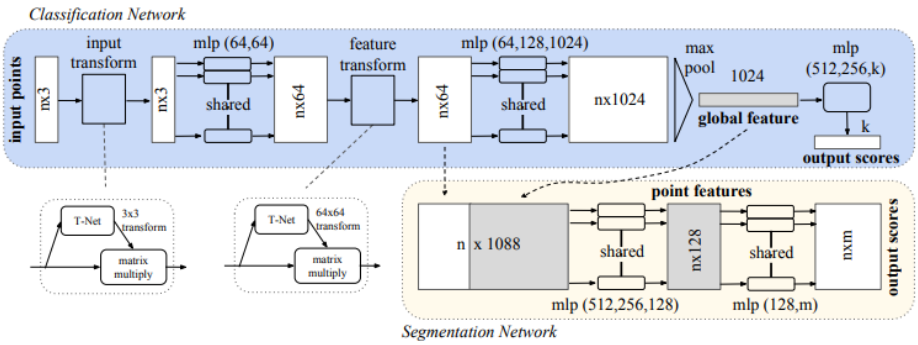


Figure 1. PointNet Architecture [9]

This architecture is applied to plant and leaf instance segmentation in the three following papers. Y. Li et al developed a process of segmenting point clouds with a focus on high-throughput data acquisition before deep learning. The data acquisition was performed using the MVS-Pheno platform to acquire high-throughput data and an internally built labeling tool, Label3DMaize, was used for the labeling. For the deep learning segmentation, Y. Li et al used PointNet to implement stem-leaf and organ instance segmentation achieving mean precision scores over 0.91 and f1-scores over 0.85 [3]. D. Li et al similarly

used PlantNet and improved upon it with a 3D Edge Preserving Sampling strategy for preprocessing input points, a Local Feature Extraction Operation module based on dynamic graph convolutions, and a semantic-instance Feature Fusion Module. The results outperformed other instance segmentation deep learning networks by over 13.3 percent [5]. K. Turgut et al compared the latest available point cloud segmentation approaches on instance segmentation with rosebush including PointNet and its improved version PointNet++. They utilized a labeled rosebush dataset and a synthetic model. The addition of the model improved the results of the model [6].

The papers demonstrate the effectiveness of point cloud instance segmentation. However, this effectiveness comes with its conditions. Y. Li et al outline that the great performance of the model is mainly due to a dataset of high quality and diversity [3]. This condition is particularly relevant as Y. Li et al and D. Li et al express how there is a shortage of well-labeled point cloud datasets in the plant domain [3], [5]. Consequently, pursuit of a point-based convolutional neural network model involves the acquisition of a high quality and diverse dataset. Acquisition of high quality point cloud data is not an accessible process and requires specific environmental conditions and tools: no wind disturbances, indoors, even and consistent lighting [3]. Not only is the acquisition of this data much more difficult, the annotation of this data and procedures for constructing 3D models for training are both much more time demanding and error-prone than the image-based model counterpart [6].

2.1.2 2D Image-Based Model for Instance Segmentation on Leaves

Most current 2D image instance segmentation approaches on leaves involve extending and training the Mask R-CNN model for the purpose of segmenting leaves. K. He et al's Mask R-CNN is self-described as "simple to train and […] easy to generalize to other tasks" and is the reason it's widely used for all types of applications including instance segmentation with leaves. Another improvement of Mask R-CNN compared to its older counterparts is its ability to accurately perform instance segmentation even among overlapping objects.

Mask R-CNN is an extension of Faster R-CNN which is an extension of Fast R-CNN. Fast R-CNN takes as input an image and object proposals. The network processes these with convolutional and max pooling layers that produce a feature map. The feature map is used to calculate softmax probability estimates and four numbers that encode refined bounding-box positions for each of the K object classes [10]. Faster R-CNN attaches a "Region Proposal Network"(RPN) before the Fast R-CNN architecture which reduces the number of object proposals and quickens the model [11]. Mask R-CNN leaves the RPN intact but adds a parallel process along with the Fast R-CNN architecture. The parallel process outputs a binary mask prediction for each region of interest along Fast R-CNN [12].

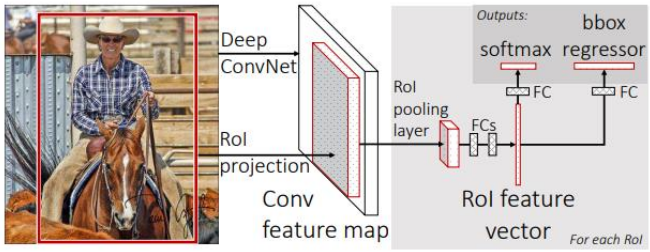


Figure 2. Fast R-CNN Architecture [10]

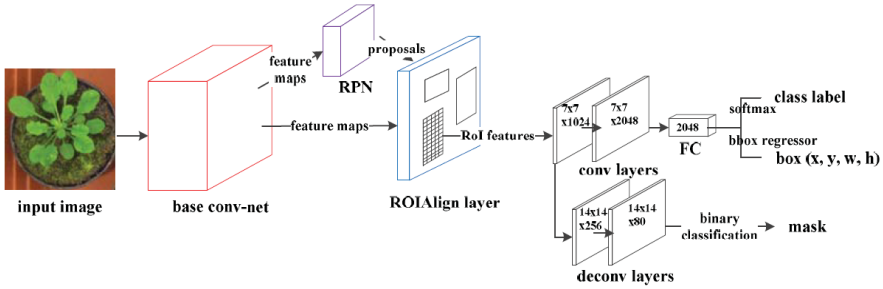


Figure 3. Mask R-CNN Architecture [7]

All together, Mask R-CNN includes three functional branches of classification, bounding-box (bbox) regression and mask branches, and the instance segmentation process is as follows. Firstly, input one image into a base convolutional-net to obtain the feature maps. Secondly, a Region Proposal Network (RPN) recommends a set of Regions of Interest (ROIs) that may include instances based on the feature maps. Thirdly, realign each ROI feature through a ROIAlign layer. Finally, predict the object category, box location and binary mask (indicates each pixel as target or background) for each ROI in parallel.[7]

This architecture is applied to plant and leaf instance segmentation in the three following papers. L. Xu et al tested different extensions of Mask R-CNN and trained the models to segment and count the number of leaves. The resulting model was able to accurately count leaves, even when overlapping, when feeding the model top view photos of individual round-leafed plants [7]. J. Schrader et al developed an android application that obtains the leaf size via instance segmentation with Mask R-CNN after placing picked leaves on a solid background [4]. J Champ et al extended Mask R-CNN for the fine detection and segmentation of weeds and crops [8].

The papers and application of Mask R-CNN to a plethora of segmentation applications demonstrate its effectiveness for a variety of objects in difficult environments. L Xu et al outlines that the reason for Mask R-CNN's effectiveness in leaf segmentation can be attributed to the RPN network that takes multi-scale anchors for the region of interest propositions, making leaves with different sizes, ocluded leaves, and leaves of different shapes recognizable [7]. Although Mask R-CNN is more accurate under these

conditions compared to other models, it can still produce erroneous results with small leaves and heavy occlusion. That said, the model's great effectiveness at segmenting different kinds of leaves with small amounts of data makes it an attractive model to use with leaf segmentation.

2.1.3 Selected Instance Segmentation Approach

Despite the greater precision of 3D point-based models that utilize the extra dimension for segmentation, the approach is not feasible for this project as it requires the acquisition of high quality and diverse data with intrusive technologies like LiDAR scanners in heavily controlled environments. The constraints of this project include data acquisition solely obtained through stereoscopic images in environments with dense foliage that cannot be altered. This makes it difficult to develop accurate and reliable 3D point-based models for this project. In addition, one of the project's aims is for the continual application to other research teams with varied environments and plant types, for example using images from row farming as training data. The difficulty in acquisition and annotation of 3D point-based datasets makes that path unappealing. This project will perform the instance segmentation tasks via the 2D image-based model Mask R-CNN. This approach will allow the research team the flexibility to collect data in changing environments, dense foliage, without tampering with the plants, and without the need of excessive time in annotation.

2.2 Leaf Area Measurements on Point Clouds and Meshes

Commented [DG18]: Only mention width in this section, rest area, "because of the curvature of the leaf, width is easier than area…."

Commented [DG19]: Only mention width in this section, rest area, "because of the curvature of the leaf, width is easier than area…."

To better understand the workflow that will be used in conjunction with this research project to obtain phenotype data, a survey into the methods of obtaining leaf area using a leaf mesh / point cloud was conducted. Since the shape of leaves is complex and varying, the survey at first consisted of finding a shortcut of some sort, a way to approximate the area of a leaf without needing to calculate the area: Quinby's paper provided some direction. "Each leaf blade (lamina) was measured for maximum width and length after each had attained maximum size. Leaf blade areas were calculated by multiplying length by width by .75. This formula was shown by Stickler et al. (1961) to give a close approximation to the actual leaf blade area as determined by a planimeter." [13] Stickler et al.'s paper provides an easier process at approximating leaf area that only requires finding a leaf's length and width, a process that seems much more manageable. Unfortunately, a process that obtains leaf area through simpler means than calculating it with need of a leaf's width and length, that is only via its width, seems to not be possible as data shows there is no consistent correlation between leaf width and leaf length. Collected data shown in Table 1 provided by the "……." Group at UIUC demonstrates this finding.

Table 1. Sorghum Leaf Length and Width Measurements and Their Ratio

Length (cm)	Width (cm)	Ratio
57.76	7.62	7.58005249
82.08	8.2296	9.97375328
91.2	9.7536	9.3503937
44.688	6.4008	6.9816273
80.56	8.5344	9.43944507
85.12	8.2296	10.3431516
53.808	7.62	7.06141732
51.68	6.096	8.47769029
75.392	7.62	9.89396325
63.84	6.096	10.4724409
54.72	4.572	11.9685039
54.72	5.4864	9.97375328
51.68	7.62	6.78215223
66.88	9.144	7.31408574
66.88	7.62	8.77690289
89.68	11.5824	7.74278215
70.832	9.4488	7.49640166
89.072	9.4488	9.42680552
88.16	11.5824	7.61154856
88.16	10.668	8.263967

Calculations of a distance across a 3D surface irrespective of rigid deformations (commonly occurring in leaves), also known as a geodesic distance, are often calculated using shortest path algorithm due to the graph-like nature of point clouds and meshes. Most existing shortest path algorithms fall under single-source shortest path algorithms and all-pairs shortest path (APSP) algorithms. When dealing with large datasets it's common practice to use single source shortest path algorithms and methods which utilize heuristic and landmarks strategies to minimize the otherwise heavy performance load. Traditionally algorithms like Dijkstra's, Breadth-First-Search, and Bellman Ford have been used [14]. Graph walks as well as other papers outline more effective ways of calculating geodesic distance involving differentiable and more time efficient methods as well as handling noisier data [15]. Due to the smaller nature of the leaf datasets in this project, it would not be inadvisable to use APSP algorithms to assist with the calculation of leaf width. However, this is out of the scope of this project.

Conducting this survey has demonstrated methods of calculating phenotype data from a leaf's point cloud or mesh, proving the usefulness of such data.

3. Methodology

The project is split into six parts: data collection, point cloud and image generation, point cloud projection, acquisition of leaf mask, scaling of leaf mask, and point cloud cropping.

The data collection phase involves determining what data to collect, where to collect the data, and how to collect the data. As the project was performed in partnership with a lab working on sorghum genetic modifications, the collected data are images of sorghum plants at different stages of growth. The data was collected in two different greenhouse rooms where genotype experiments were being performed. And the data was collected via a stereo camera called the "ZED" camera by Stereo Labs. The data is collected in a proprietary SVO format that needs to be processed by the ZED SDK to obtain more useful data.

The generation of the point cloud of the scene is obtained by first extracting the coordinates and color values of all the points from the SVO recordings using the ZED SDK and provided API, and then formatting the data into a data structure that is easy to read and manipulate. The 2D image generation is performed using the provided ZED SDK API.

The point cloud projection …..

The acquisition of a leaf mask is obtained by training an instance segmentation model to segment a leaf with annotated images of sorghum leaves at different stages of its lifecycle, in three different rooms, with three different distances between the camera and the plants.

The scaling of the leaf mask is performed via experimentally defining cutoff planes in the point cloud and a ratio between the dimensions of the mask and the point cloud. The method of scaling the leaf mask in this project is imprecise and error prone and a point of improvement for future work.

The point cloud cropping is performed by using the leaf mask as a filter in the length-width coordinate system of the point cloud.

Figure 4 describes the flow of the research project's product.

Commented [DG20]: John: Add sentence or two on how these will be used in your overall approach.

Commented [DG21]: John: Don't need to say this yet, leave for evaluation section.

Commented [GD22]: Will mention "There is a potential issue here that you should mention. If there is occlusion the points in the crop by cause issue with the distance measurement calculations, but I assume these crops would be able to be thrown out using the Reject Leaf algorithm." In the future work and limitations sections of the discussion section

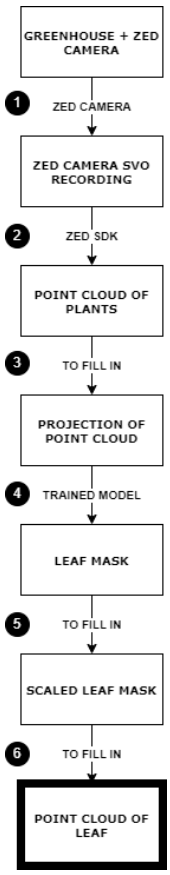


Figure 4. Project Workflow

3.1 Data Collection

3.1.1 Greenhouse Setup

The research project was performed with a focus on its application to the Sorghum plant. The first stage of data collection was performed in a predefined environment by the genetic researchers this project was done in partnership with. This environment was a room in the University of Illinois Plant Biology Greenhouse filled with different genetic varieties of Sorghum. Figure 5 showcases the room at the start of the data collection process.

Commented [DG23]: John: In the future section, you can say Maize is similar so methods can also be used on those plants as well.

Figure 5. First Greenhouse Room at Week 0 of Collection

After 9 weeks the Sorghum plants in the first greenhouse room matured to a point where there were no more significant changes in leaf size. Figure 6 showcases the room at this point.

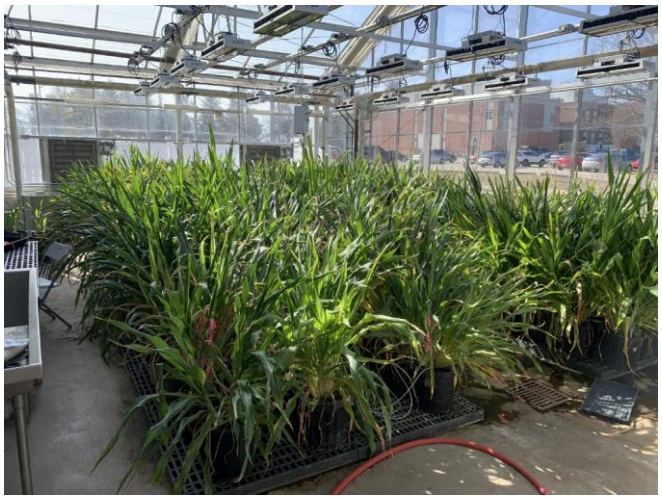


Figure 6. First Greenhouse Room at Week 9 of Collection

As a result of the unchanging leaf size, the setup was moved to a second greenhouse room to obtain more useful training data. In the second room Sorghum plants with no genetic modifications at different points in the growth cycle were grown and photographed. The second room focused on young sorghum plants as technical issues caused loss of data in the early days of data collection in the first room. Figure 7 showcases the setup in the second room.

Figure 7. Second Greenhouse Room at Week 0 of Collection

Figure 8. Second Greenhouse Room at Week 0 of Collection

Figure 9. Second Greenhouse Room at Week 17 of Collection

Commented [DG24]: Remove because this involves the zed camera which comes next in the section?

3.1.2 ZED Camera and Data Collection Setup

Once the subjects of the data collection were defined and ready to be photographed, the next step was setting up the data collection tool. The ZED camera and ZED box, a processing unit used for the camera, were fastened onto a wooden plank, and placed near the ceiling. Figure 8 is an image taken of the camera and box fastened to the wooden plank attached near the ceiling.

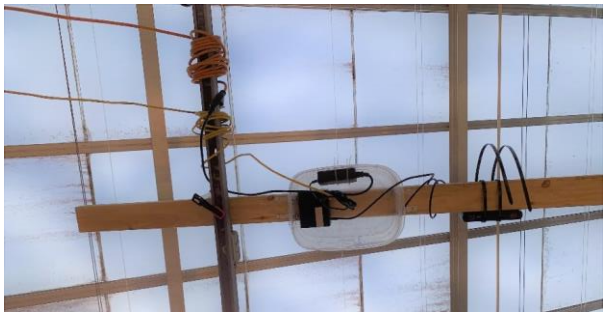


Figure 10. Bottom-Up View of ZED Camera and ZED Box on Plank

To access the ZED box, the ZED SDK installed onto the box, and the connected ZED camera, a switch was set up to be used as an interface between a laptop and the ZED box. The ZED camera was connected to the ZED box through a USB type A 3.0 cable. The ZED box was connected to a switch through an ethernet cable. And the switch was connected to a laptop through an ethernet cable. Figure 9 showcases the switch used and the power strip used to power the ZED Box, ZED Camera, and switch.

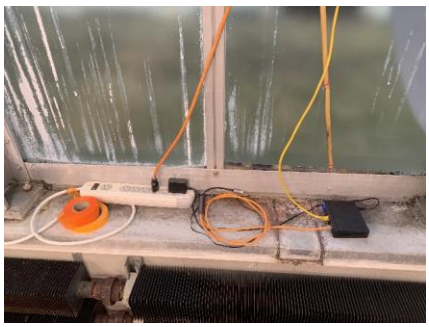


Figure 11. Switch connected to ZED Box

This initial part of setup involved many issues. One of the biggest issues was networking related. Initial attempts at connecting the ZED box to the school's network via the switch to be able to access the camera remotely were unsuccessful as the school's infrastructure didn't allow it. Other local networking issues arose involving the static vs dynamic IP configuration of the ZED box.

Once access to the ZED box and ZED camera was established the calibration and basic setup of the camera to be able to take appropriate recordings was the next step. Utilizing the ZED API integrated with the ZED SDK on the ZED Box, it was simple to connect to the ZED Camera and capture a recording. Some of the image settings weren't adequate for the scene and had to be tweaked, including the exposure, gamma, and white balance. There seemed to be a few bugs with the camera that made this process difficult. The 'auto' versions of these settings were often ineffective. Figure 10 and 11 show common occurrences when capturing images with the ZED camera where the auto exposure and auto white balance wouldn't behave adequately.

Figure 12. ZED Image Capture at 'auto' exposure

Figure 13. ZED Image Capture at 'auto' white balance

Even when exposure and white balance settings were manually modified, the behavior sometimes didn't change. The conclusion arrived after experimenting with resolving the issue was that manually entering settings required more 'camera on' time to calibrate. When directing the ZED camera to take longer captures, these manual setting modifications were enacted and resulted in appropriate images as showcased in Figure 12.

Figure 14. ZED Image Capture properly calibrated

Once the ZED camera was calibrated to take the appropriate photos, the next step was writing and setting up the scripts that would automatically collect the recordings throughout the next few months. This involved utilizing the ZED SDK to take SVO recordings. These recordings are various frame captures from the ZED stereo camera and can be processed by the ZED SDK to create plain 2D images, point clouds, meshes, and other renderings not relevant in this project. The code that directs the camera to

capture SVO recordings involves initializing the camera with the appropriate parameters and camera settings and then simply recording the frames.[1]

In order to capture SVO recordings recurringly over the course of a few months, a bash script was written that ran the SVO recording capture script in the background. At first the 'nohup' linux command was used to run recordings in the background. However, this command led to some issues and thus the 'screen' linux command was used for the background operation. An issue relating to camera overheating arose as well. The only permanent solution found was making the resolution of the recordings 1080 p instead of the 2K used in the recordings in the beginning of the project.

3.1.3 Control Photos

Once data collection was setup, control and calibration photos were taken. These were taken to determine the precision and accuracy of the point clouds generated from the camera's recordings and in case software calibration was needed. An artificial plant and a ruler at different distances from the camera were photographed. Figure 15 showcases an image of an artificial plant. Figure 16 shows an image of a ruler directly below the camera to measure inaccuracy as the measurement target gets further from the camera. Figure 17 shows an image of a ruler in the far-right side of the image to measure distortion when nearing the edges of the camera's field of view.

Figure 15. Calibration Photo of Artificial Plant

Commented [DG25]: Remove if not used in the results/discussion section

[1] For more information read Appendix A

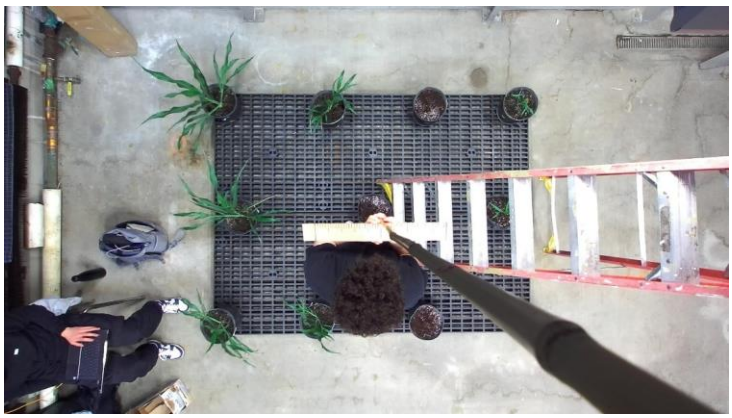


Figure 16. Calibration Photo of Ruler at 150cm from Camera

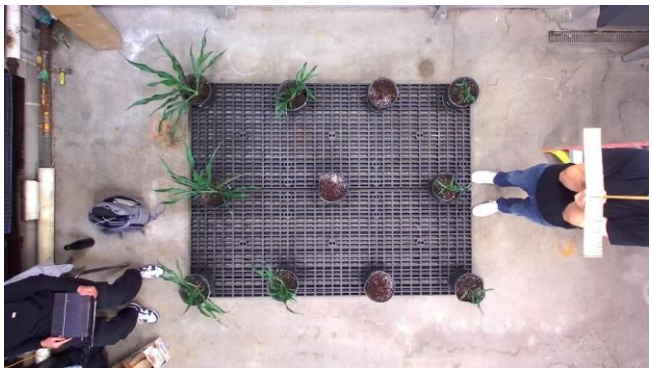


Figure 17. Calibration Photo of Ruler at 150cm from Camera on Right

3.1.4 Validation

As demonstrated in this section, validation of the data collection process occurred throughout the setup. The ZED SDK preinstalled onto the ZED Box was utilized to generate the previously shown 2D images. Validation of the effectiveness of the camera and the point clouds it produces is in the following section, “Point Cloud Generation.”

3.2 Point Cloud Generation

3.2.1 ZED SDK Setup

The ZED SDK contains software that's able to interpret the proprietary SVO recordings and generate point clouds from them. To process the SVO recordings more quickly and away from the Greenhouse Room, it was necessary to install the ZED SDK onto a machine other than the ZED Box. Standard setup for the ZED SDK often is performed on the host machine's OS. This was the path taken in the project: downloading the ZED SDK on a utilized anaconda virtual environment. However, complications arose related to the CUDA dependency: the version was incompatible with the ZED SDK. CUDA's intertwined relationship with the operating system and graphics card of the host machine made it difficult to continue the standard installation path for the ZED SDK. To avoid needing to make a great number of changes to the host machine, an alternative approach was taken. A publicly available Docker image with the ZED SDK was installed onto the system and a container was run from it. Within this container it was possible to make use of the ZED SDK and generate the point clouds from the collected recordings.

Additional work was done in setting up the docker container to make it persistent and to make SVO recording data saved within the docker container persist even if the container were to stop using Docker Volumes. For more details refer to Appendix A.

3.2.2 Point Cloud Generation and Visualization using ZED SDK

Once the ZED SDK was setup data and the recordings were transferred into the container with the ZED SDK installed, sample code provided with the ZED SDK was run. The output was of an OBJ file type with specified vertices and their colors.

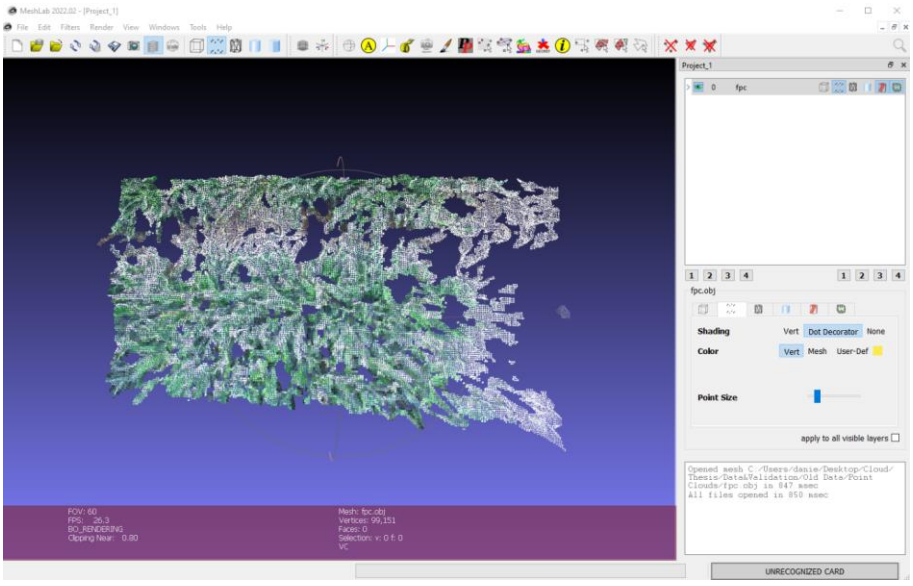


Figure 18. Visualization of Point Cloud with Sample ZED SDK Code

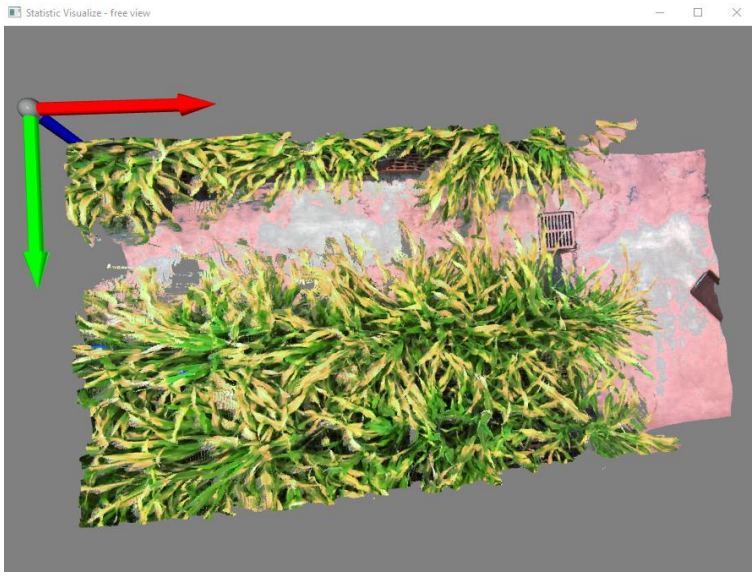


Figure 19. Visualization of Point Cloud with Custom Point Cloud Generation Code

The ZED SDK contains the functionality for developing the point cloud data and abstracts it to make it easy to use. Attributes like 'VIEW.LEFT' and 'MEASURE.XYZRGBA' allow easy extraction of 2D images and point clouds from SVO recordings. An example 2D Image taken from the left camera extracted from an SVO recording is shown in figure 13. An example point cloud produced by the SDK is shown in figure 14.

Commented [DG26]: Footer to appendix with code

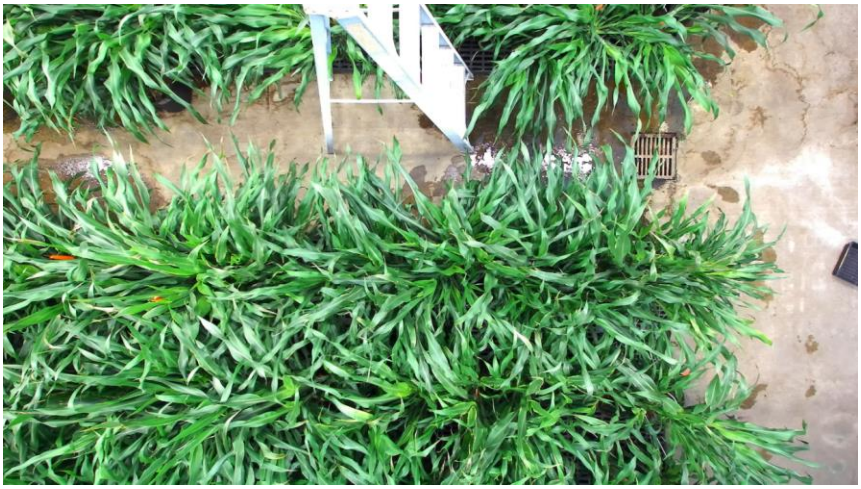


Figure 20. RGB 2D Left Stereo Image

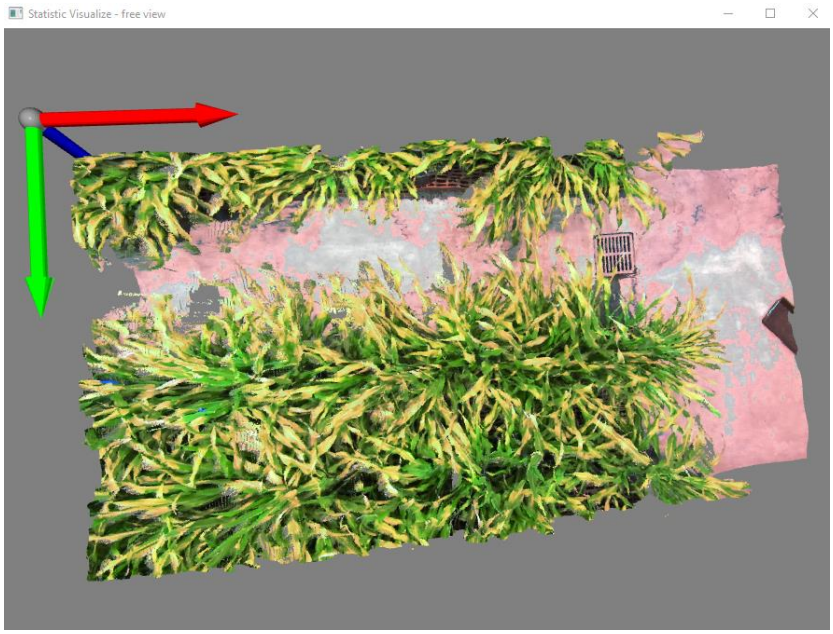


Figure 21. Point Cloud

The 2D images are exported in JPG format. The point cloud is exported as an OBJ file utilizing the sample code in the ZED SDK examples. However, to make the data easier to manipulate in later sections of the project, custom code was written to extract the point cloud data into a basic text file that only includes the X, Y, Z coordinates of the points and the R, G, B, A color values.[2] Another benefit of the custom code is that it provided a greater quantity of data points leading to smoother visualizations. Figure ____ shows a point cloud generated with the native export point cloud code in the SDK rendered in Mesh Lab on the right, compared to a point cloud generated with the custom point cloud extraction code rendered with Open 3D on the left.

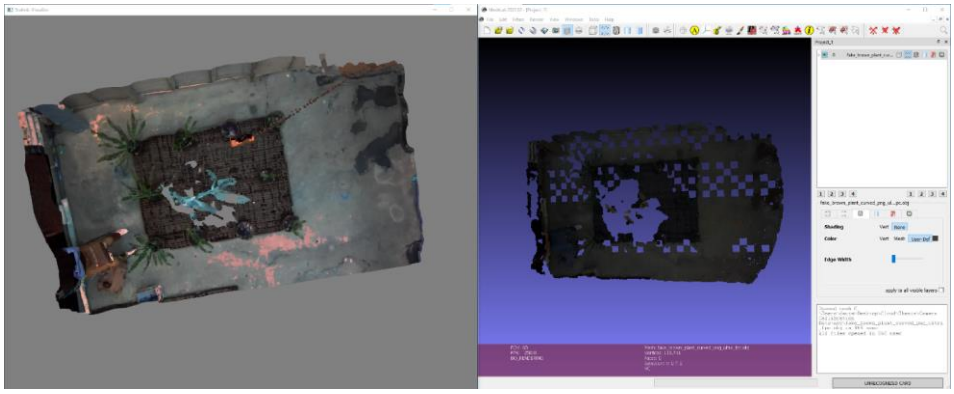


Figure 22. Point Cloud

3.2.3 Validation of Data Collection and Point Cloud Generation

To verify the efficacy of the ZED Camera and the Point Cloud Generation Software

[2] For more information read Appendix A

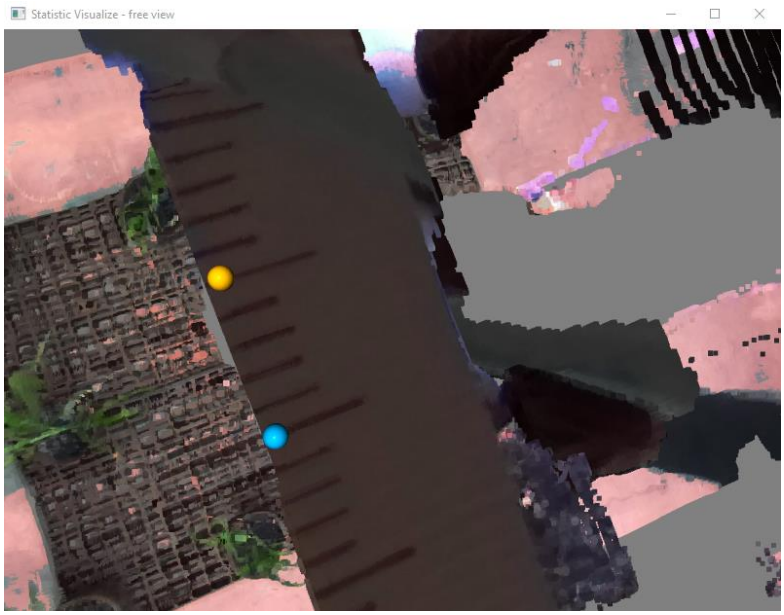


Figure 23. Point Cloud

The results of this validation are detailed section 4.1 of this paper.

3.3 2D Representation of Point Cloud

Applying a 2D mask to a 3D point cloud can have its complications even when constraining the point cloud cropping to only the x and y dimensions. To remedy such complications, the point cloud is transformed to its two-dimensional representation. The process involves four components:

- The point cloud's x and y axis are scaled to the dimensions of the stereo image.
- The dimensions of the stereo image are represented by a 2D array where each item in the array represents one pixel. The points in the point cloud are each paired with a position in the 2D array.
- Points that fall in the same x-axis interval and y-axis interval (aka points that are overlap), are placed in a list at the corresponding interval in the 2D array
- Points in the point cloud are paired to a position in the 2D array based on …… angle …..

3.4 Leaf Mask Acquisition with Instance Segmentation Model

With the raw 2D images and point cloud as input and an instance of an individual leaf as the desired output, an instance segmentation model is required for the next portion of the project. To best utilize the instance segmentation model, high quality and diverse annotated data is used to train, validate, and test the model. So, there's three ideal components to the data: high quality, annotated, and diversity. The images captured were as high quality as the ZED camera and ZED box allowed. The annotation is required after acquisition of the high-quality images to help train the model detect the objects labeled. The diversity of the data is acquired by capturing images in three different greenhouse rooms and by utilizing different cameras, ZED camera and go pro. It is now relevant to mention that data captured for another project with the go pro camera was utilized in this project for training the instanced segmentation model to add diversity. Diversity of the data can also be added through image augmentation. This process involves transforming raw images through color modifications, rotations, scaling, etcetera and feeding these transformed images into training the model. The sequence described is visualized in figure 18.

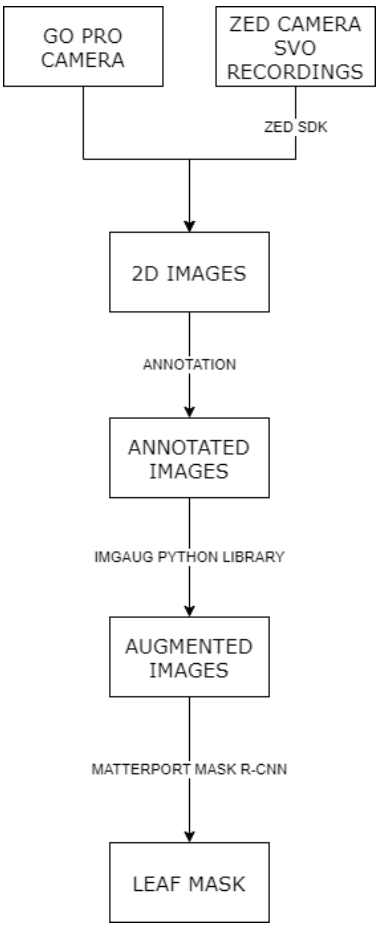


Figure 24. Training Mask R-CNN Model Workflow

What's left to complete the workflow visualized in figure 18 is annotation, augmentation, and using the Mask R-CNN model for training and inferencing.

3.4.1 Image Annotation

For accomplishing the annotation task and annotating the 2D images obtained from the ZED camera's SVO recordings, this project used the open source VGG Image Annotator. As no edits to the source code were required for this project and the tool is available for use online, this project utilized the online version of VGG Image Annotator. Annotation was accomplished by importing an image, annotating clear

and flatter leaves with the polygon tool, and exporting the annotations in a .JSON file. Figure 19 showcases the tool being used.

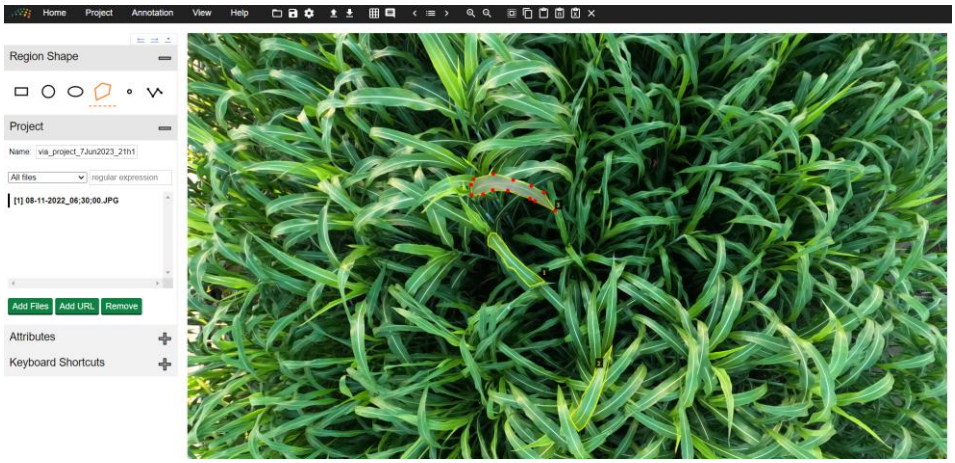


Figure 25. VGG Image Annotator

3.4.2 Image Processing Pre-Training

First the annotation masks are cropped to avoid processing the excess zeros around the mask and minimize computation when training the model. Then these annotation masks are applied to the images.

Figure 26. Annotated Image

Second the image augmentation is applied to the images. The augmentation of the images was accomplished via the python imgaug library. The transformations used on the images include rotation, vertical/horizontal flipping, color changes, blurring, and scaling. Figure 20 showcases one of the images transformed via a few of the image augmentation methods.

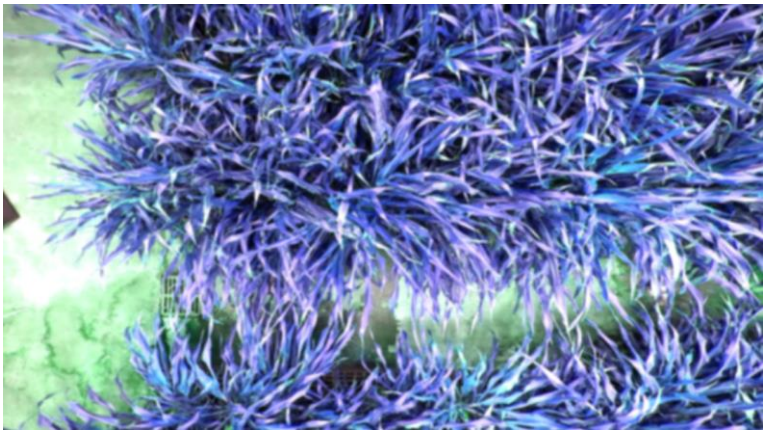


Figure 27. Augmented Image

3.4.3 Model Training

With the processed images ready for training, the following step requires using Mask R-CNN to train a model than can detect and segment individual leaves. An open-source implementation of Mask R-CNN[3] with tensorflow and keras from Matterport was used to implement the instance segmentation model. The model was trained on a remote machine with access to two GPUs for training and image processing purposes. The remote machine operated on two Nvidia GPUs and CUDA version 9.0.

Table 1 outlines the configurations for the matterport implementation of Mask R-CNN. Almost all parameters are the default ones. The only one changed parameter is that of "GPU_COUNT" to specify the number of GPUs this project had access to.

The model was trained on the 2017 COCO dataset as a baseline and the sorghum plant dataset built for this project. The model was trained for 30 epochs each with 100 steps. The standard output of training for the last few epochs is displayed in figure 22.

```
Epoch 27/30
100/100 [==============================] - 98s - loss: 0.9792 - rpn_class_loss: 0.0913 - rpn_bbox_loss: 0.2543 - mrcnn_class_loss
: 0.2577 - mrcnn_bbox_loss: 0.1972 - mrcnn_mask_loss: 0.1787 - val_loss: 2.0546 - val_rpn_class_loss: 0.2573 - val_rpn_bbox_loss:
 0.6220 - val_mrcnn_class_loss: 0.5047 - val_mrcnn_bbox_loss: 0.4324 - val_mrcnn_mask_loss: 0.2382
Epoch 28/30
100/100 [==============================] - 96s - loss: 0.8778 - rpn_class_loss: 0.0789 - rpn_bbox_loss: 0.2172 - mrcnn_class_loss
: 0.2377 - mrcnn_bbox_loss: 0.1743 - mrcnn_mask_loss: 0.1698 - val_loss: 2.1627 - val_rpn_class_loss: 0.3218 - val_rpn_bbox_loss:
 0.7100 - val_mrcnn_class_loss: 0.4506 - val_mrcnn_bbox_loss: 0.4298 - val_mrcnn_mask_loss: 0.2504
Epoch 29/30
100/100 [==============================] - 96s - loss: 1.0018 - rpn_class_loss: 0.0833 - rpn_bbox_loss: 0.2555 - mrcnn_class_loss
: 0.2689 - mrcnn_bbox_loss: 0.2060 - mrcnn_mask_loss: 0.1881 - val_loss: 2.0673 - val_rpn_class_loss: 0.2507 - val_rpn_bbox_loss:
 0.6826 - val_mrcnn_class_loss: 0.4385 - val_mrcnn_bbox_loss: 0.4552 - val_mrcnn_mask_loss: 0.2403
Epoch 30/30
100/100 [==============================] - 90s - loss: 0.9733 - rpn_class_loss: 0.0801 - rpn_bbox_loss: 0.2287 - mrcnn_class_loss
: 0.2653 - mrcnn_bbox_loss: 0.2097 - mrcnn_mask_loss: 0.1896 - val_loss: 2.1218 - val_rpn_class_loss: 0.3128 - val_rpn_bbox_loss:
 0.7225 - val_mrcnn_class_loss: 0.4270 - val_mrcnn_bbox_loss: 0.4244 - val_mrcnn_mask_loss: 0.2351
```

Figure 28. Training Output

ELABORATE IN THIS SECTION

3.4.4 Model Prediction

When using the model to predict the masks of leaves on the test dataset, the same configuration parameters as in table 1 were used.

Using the model to predict the leaf masks in an image involves 3 main stages. Stage 1 is using the Region Proposal Network (RPN), running a binary classifier on anchors that covers the entire image, to output region proposals. Region proposals are bounding boxes covering regions that are most likely the location of a leaf to be segmented. Stage 2 is the Proposal Classification. Stage 2 classifies the region proposals per class, including class probabilities and bounding box regressions. After stage 2 some refinement is done: adjusting bounding boxes, filtering out low confidence detections. After this stage 3 is takes the detections from the previous layer and runs the mask head to generate segmentation.

[3] For more information on how the model is trained read "Mask R-CNN" [12]

At the end of the process, after running a test image through the training process, some leaves are precisely masked. This can be viewed in Figure 23.

Figure 29. Test Image with Inferenced Leaf Masks

3.5 Leaf Mask Scaling

The leaf masks that result from the instance segmentation model's inference are not scaled appropriately to be able to apply the masks onto the projected point cloud. During the inference process the model resizes and compresses the images for ease and efficiency of data processing. The resulting leaf masks cannot be applied to the stereo images or projected point cloud before scaling the masks appropriately.

…… (probably about how the model resizes, I need to do reverse)

Ohhhhhh I just realized, the 1024 by 1024 is the size of the mask but not the size of the image the mask is applied to, the top and bottom are black bars, I need to:

- Apply same resizing mechanism to my images
- Write a script to determine the actual height (without the black bars) of the image -> this will be the actual height (without the black bars) of the mask

3.6 Point Cloud Cropping

With a 2D representation of the point cloud and a leaf mask that scaled appropriately to this 2D representation, all the components needed to crop the point cloud and acquire a point cloud of an individual leaf are available.

Applying the scaled leaf bitmask on the 2D representation of the point cloud yields the 2D representation of the point cloud of an individual leaf. Reversing the process of projecting the point cloud to a 2D array, the 2D array is projected back to its original structure.

4. Research Results

All the five steps outlined in the methodology chapter have their own technique for validation. This chapter will outline the results of each section of the project and demonstrate how each section was validated.

4.1 Data Collection and Point Cloud Generation & Rendering

In order to validate that the camera produces accurate data and the renderings accurately capture the information in the SVO recordings a set of control photos were taken with the ZED camera to compare real and vision-based measurements. An example of these control recordings are displayed as a 2D image and a point cloud rendering in Figures _____.

After generating the point cloud of the control recordings using the ZED sdk and rendering the point cloud using a point cloud visualization tool powered by the open 3d python library, vision-based measurements were taken along the ruler to populate metrics that help demonstrate the precision of depth, length, and width vision-based measurements. Two areas of loss of precision were identified: distance from camera and distance to the edge of the field of view (FOV) of the camera. The hypotheses of the impact of these metrics were that a higher distance from the camera would correlate with lower precision of measurements and data collected closer to the edge of the camera FOV would be impacted by lens distortion and contribute to lower precision of measurements.

Example renderings of control recordings taken to determine the impact distance from camera has on measurement precision are shown in Figures ____.

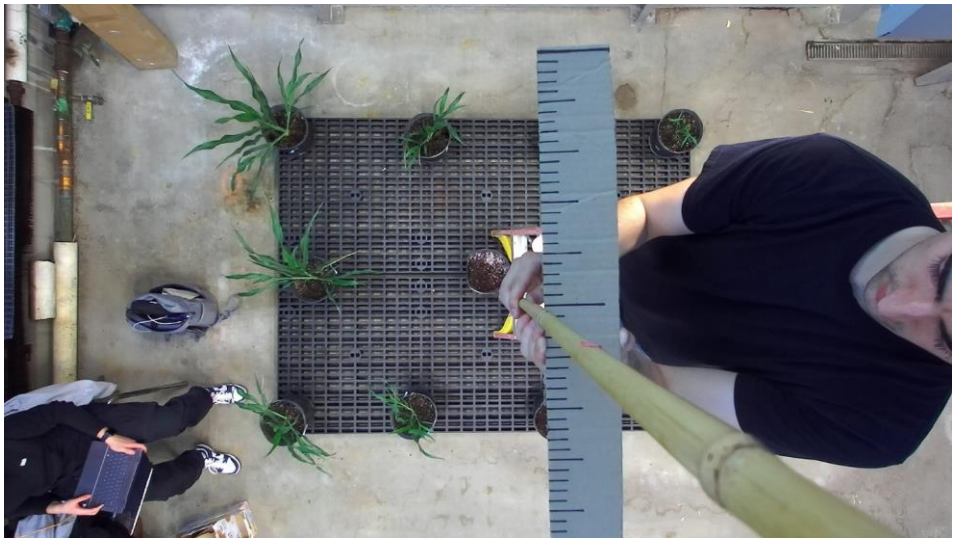


Figure 30. Control Photo of a Ruler 50 cm From the Camera's Lens

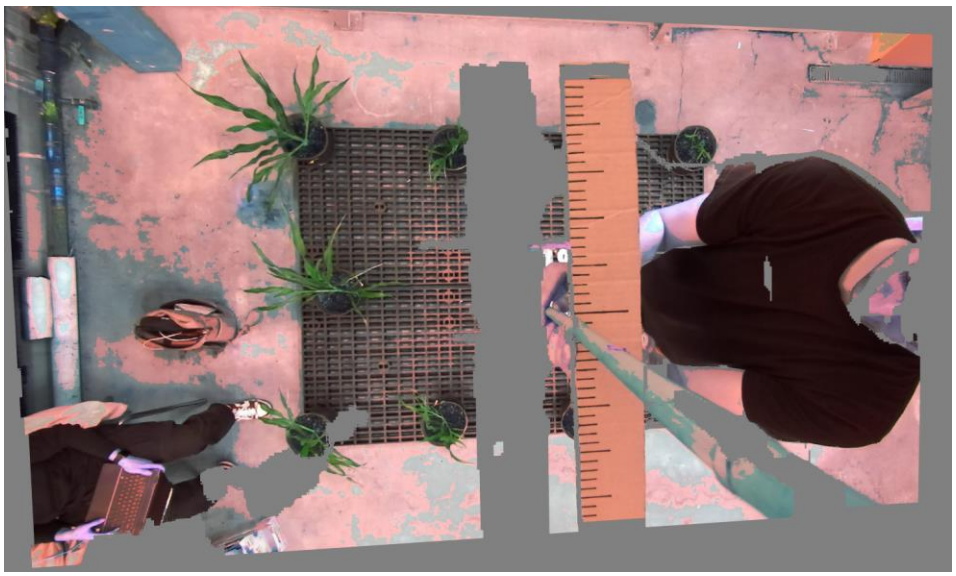


Figure 31. Point Cloud Rendering of a Ruler 50 cm From the Camera's Lens

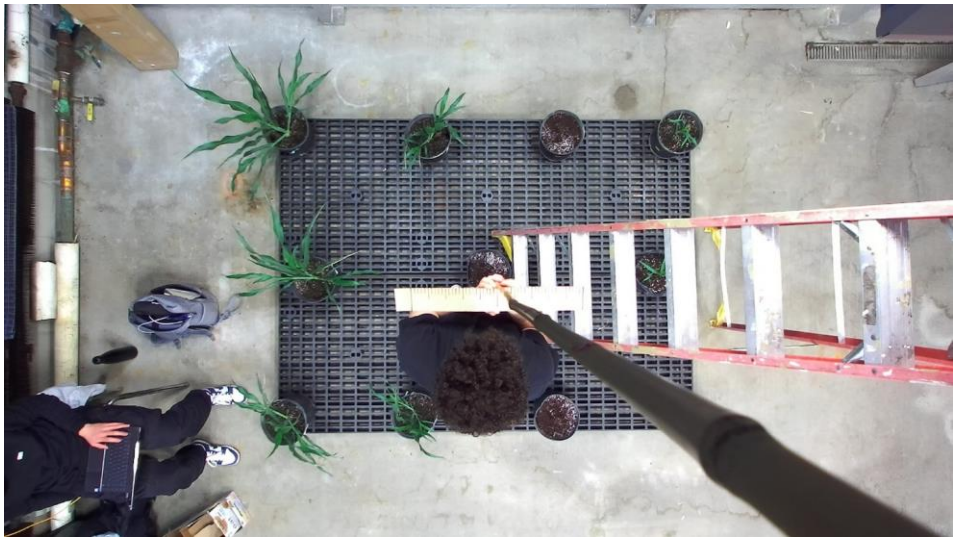


Figure 32. Control Photo of a Ruler 150 cm From the Camera's Lens

Figure 33. Point Cloud Rendering of a Ruler 150 cm From the Camera's Lens

Table 1 and 2 showcase data intended to demonstrate trends with regards to precision dependent on the measurement's distance to the camera. Table 1 showcases depth measurements taken across the different recordings. Table 2 showcases length/width measurements taken across the different recordings. Vision-based measurements taken at 25 cm from the camera are unpopulated as the point clouds generated from the recordings were not able to generate points of objects at 25 cm or closer to the camera. The generated point clouds included objects at 50 cm and further from the camera. These measurements are populated and are used to determine trends in the precision of recordings and rendered point clouds across the different dimensions.

Table 2. Depth Measurements Taken at Different Distances from Camera

Real Distance	Point Cloud Distance	|Expected – Actual|	Percent Error
25 cm	-	-	-
50 cm	50.87 cm	0.87 cm	1.74 %
75 cm	76.02 cm	1.02 cm	1.36 %
100 cm	102.94 cm	2.94 cm	2.94 %
125 cm	127.84 cm	2.84 cm	2.27 %
150 cm	162.42 cm	12.42 cm	8.28 %

Table 3. Length/Width Measurements Taken at Different Distances from Camera

Distance From Camera	Real Distance	Point Cloud Distance	|Expected – Actual|	Percent Error
25 cm	1 cm	-	-	-
50 cm	1 cm	1.0127 cm	0.0127 cm	1.27 %
75 cm	1 cm	1.0246 cm	0.0246 cm	2.46 %
100 cm	1 cm	1.0335 cm	0.0335 cm	3.35 %
125 cm	1 cm	1.0845 cm	0.0845 cm	8.45 %
150 cm	1 cm	1.1850 cm	0.1850 cm	18.50 %

The associated graph displaying the trend of the data in Table 1 is shown in figure ___. The associated graph displaying the trend of the data in Table 2 is shown in figure ____.

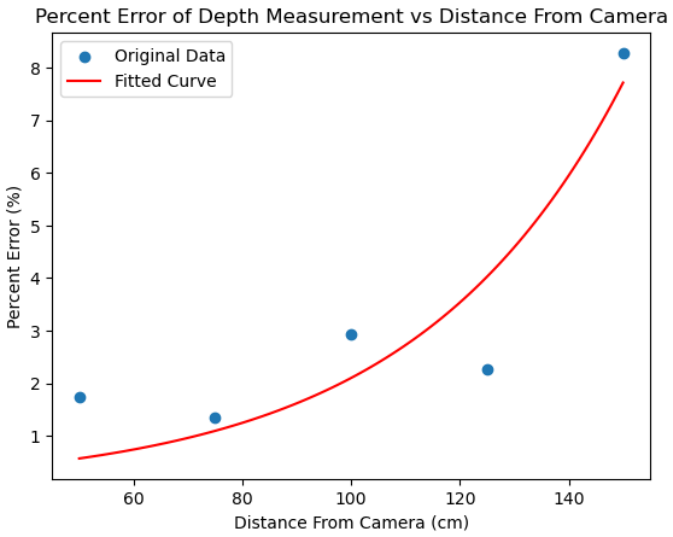


Figure 34. Best Fitting Curve Graph of Percent Error of Depth Measurement vs Distance from Camera

$$y = 0.156416 * \exp(0.025988 * x)$$

Figure 35. Best Fitting Curve Equation of Percent Error of Depth Measurement vs Distance from Camera

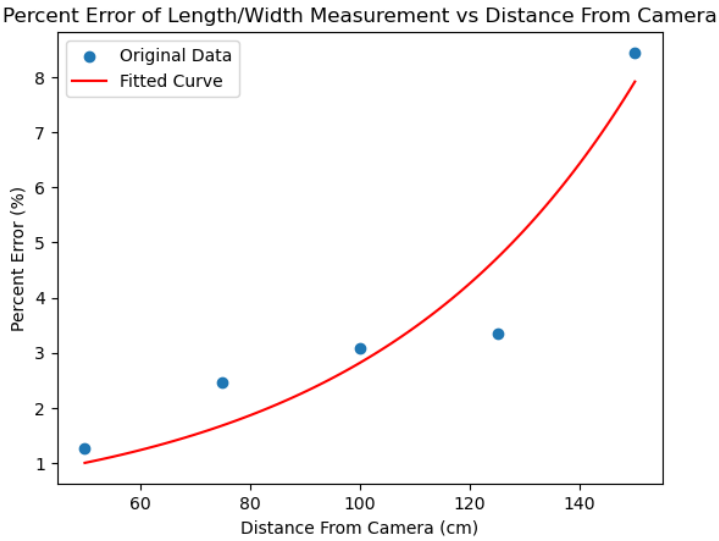


Figure 36. Best Fitting Curve Graph of Percent Error of Length/Width Measurement vs Distance from Camera

$$y = 0.356454 * \exp(0.020674 * x)$$

Figure 37. Best Fitting Curve Equation of Percent Error of Length/Width Measurement vs Distance from Camera

Example renderings of control recordings taken to determine the impact distance from FOV of the camera has on measurement precision are shown in Figures ____.

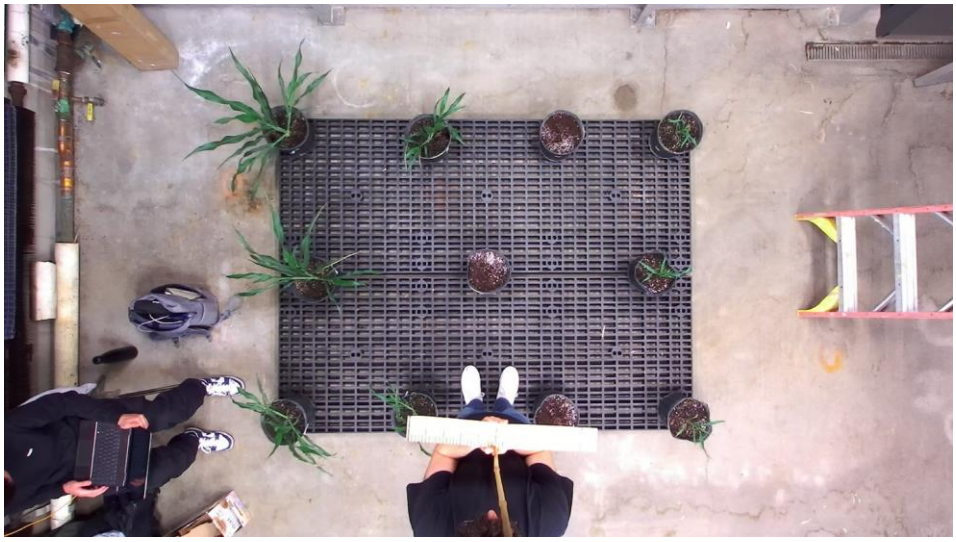


Figure 38. Control Photo of a Ruler 150 cm From the Plane of The Camera's Lens

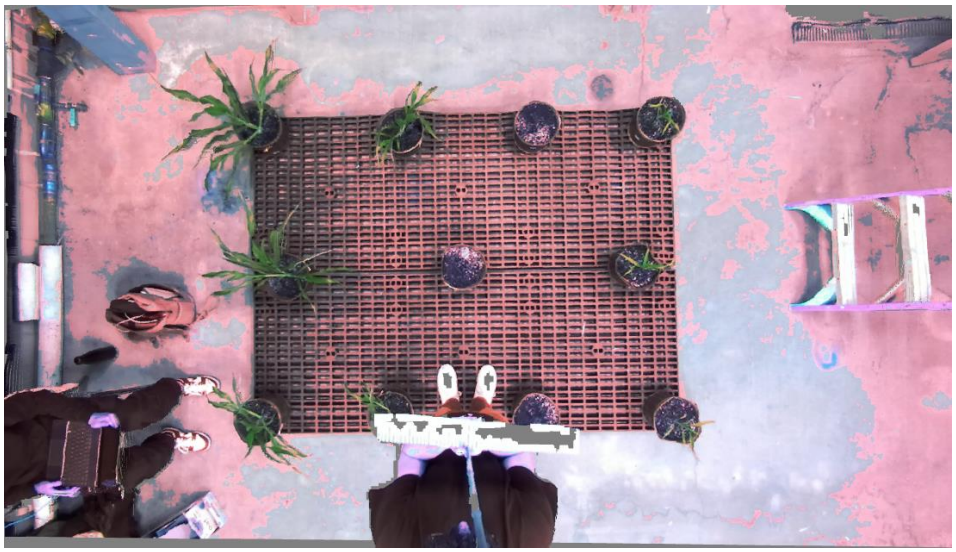


Figure 39. Control Photo of a Ruler 150 cm From the Plane of The Camera's Lens

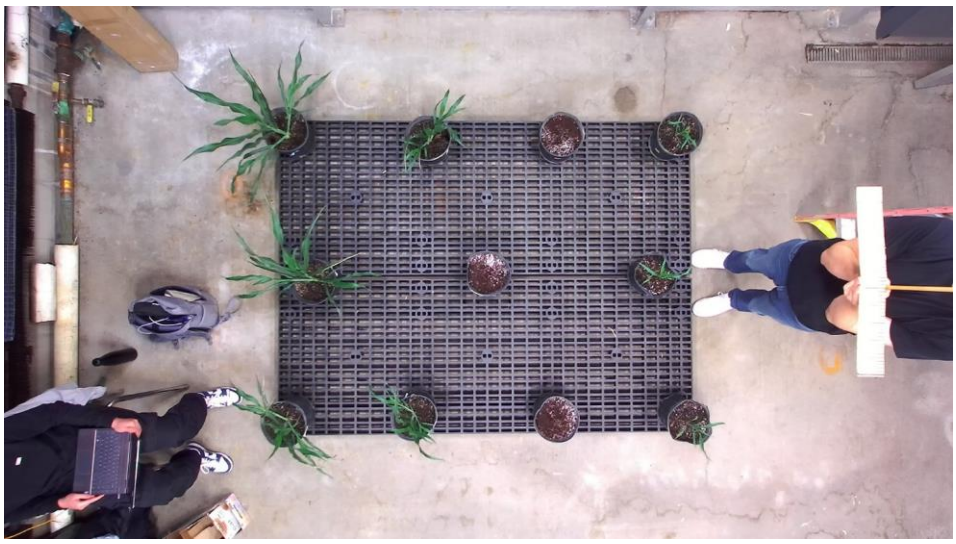


Figure 40. Control Photo of a Ruler 3__ cm From the Plane of The Camera's Lens

Figure 41. Control Photo of a Ruler 3__ cm From the Plane of The Camera's Lens

Table 3 showcase data intended to demonstrate trends with regards to precision dependent on the measurement's distance to the FOV of the camera. Table 3 showcases length/width measurements taken across the different recordings. The vision-based measurements are populated and are used to determine trends in the precision of recordings and rendered point clouds.

Table 3 showcases data intended to demonstrate trends with regards to precision dependent on the measurement's closeness to the edge of the FOV of the camera. Table 3 showcases length/width measurements taken across the different recordings. …

Table 4. Length/Width Measurements Taken at Different Locations in View and Distances from Camera ***** INCLUDE MORE MEASUREMENTS, MENTION HOW SIDE MEASUREMENTS WERE GOOD BECAUSE THOSE WERE CLOSE ENOUGH TO VIEWING EDGE**

Distance to Closest Edge	Real Distance	Point Cloud Distance	|Expected – Actual|	Percent Error
160 cm	1 cm	1.0845 cm	0.0845 cm	8.45 %
150 cm	1 cm	1.1850 cm	0.1850 cm	18.50 %
110 cm	1 cm	1.3877 cm	0.3877 cm	38.77 %

The associated graph displaying the trend of the data in Table 1 is shown in figure ___. The associated graph displaying the trend of the data in Table 2 is shown in figure ____.

The chart demonstrating the precision of the vision-based measurements relative to the actual measurements is shown in figure _____.

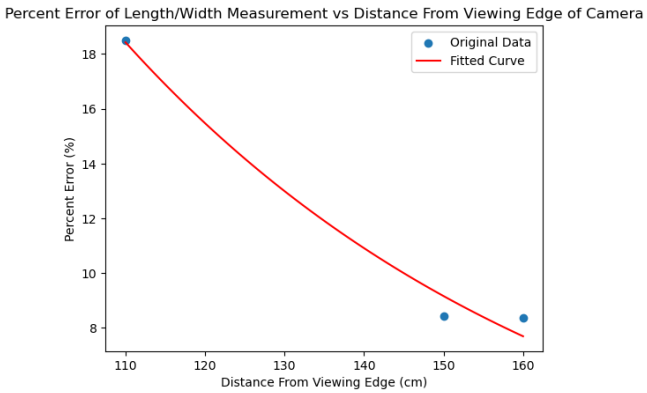


Figure 42. Best Fitting Curve Graph of Percent Error of Length/Width Measurement vs Distance to Viewing Edge of Camera

$$y = 125.820914 * \exp(-0.017463 * x)$$

Figure 43. Best Fitting Curve Equation of Percent Error of Length/Width Measurement vs Distance to Viewing Edge of Camera

4.2 2D Representation of Point Cloud

4.3 Leaf Mask Acquisition with Instance Segmentation Model

In order to validate that the trained model produces predictions that are precise and accurate, the project uses one of the Matterport Mask-RCNN sample's model validation jupyter notebooks to

quantitatively demonstrate the quality of the predictions. …… (talk about and show figures of these demonstrations)

FIGURE _____

In order to validate that an individual leaf mask is being isolated and can be used as a filter, the mask has been converted into a black and white image of which the background is white and the mask is black. As figure ____ demonstrates, the individual mask matches exactly with one of the leaf masks represented in the Matterport Mask-RCNN model validation file.

FIGURE_____

4.5 Leaf Mask Scaling

The validation for the leaf mask scaling

4.6 Point Cloud Cropping

The validation for the point cloud cropping consists of using a point cloud of a known shape filtered by a known mask of a known shape as inputs, producing an expected result. The point cloud of a known shape will be that of a cube, as shown in figure _____. The mask will be that of a circle, as shown in figure _____.

The result of filter the cube with a circle is shown in figure _____

5. Discussion

The goal of this research is to define a method and develop a program that researchers can use to acquire leaf models for plants in environments with dense leaf clusters. The results of the project demonstrate successful aspects and areas needing improvement. Although the precision and accuracy of some measurements doesn't make the product of this project ready to be shared, the results give empirical evidence that the method utilized is able to go through the process of acquiring leaf models for plants in environments with dense leaf clusters.

5.1 Data Collection and Rendering

5.1.1 Key Findings

Figure ____ showcases the findings from the data collection and rendering validation. The results showcase greater inaccuracy as the object moves further from the camera and closer to the edges of the camera's view. That said, the accuracy of measurements even 3 meters away and close to an edge of __, is stil ___%. With this information, it is conclusive that the precision of vision-based measurements with the ZED camera is high and performing vision-based measurements with the ZED camera recordings is reliable.

Commented [DG27]: Charts in discussion section, not results?

5.1.2 Limitations

Although the precision of most relevant points are high. There are many limitations observed from the visualization of the produced point clouds.

- Too few points using native exporting tools from zed sdk
- Loss of precision when depth changes quickly, more loss the greater the depth change
- Loss of points closer than 50 cm away from camera
- Ruler is reflective

Commented [DG28]: Put these only here and methodology or also results section?

5.1.3 Future Work

Greater experimentation with the attributes in the zed sdk to see if it does better with depth changes

5.2 2D Representation of Point Cloud

5.2.1 Key Findings

TO DO

5.2.2 Limitations
TO DO

5.2.3 Future Work
TO DO

5.3 Leaf Mask Acquisition with Instance Segmentation Model

5.3.1 Key Findings
TO DO

5.3.2 Limitations
TO DO

5.3.3 Future Work
TO DO

5.4 Leaf Mask Scaling

5.4.1 Key Findings
TO DO

5.4.2 Limitations
TO DO

5.4.3 Future Work
TO DO

5.5 Point Cloud Cropping

5.5.1 Key Findings

TO DO

5.5.2 Limitations

"There is a potential issue here that you should mention. If there is occlusion the points in the crop by cause issue with the distance measurement calculations, but I assume these crops would be able to be thrown out using the Reject Leaf algorithm."

5.5.3 Future Work

TO DO

With the point cloud of the leaf, distances are calculated between edge points and the midline of the leaf to determine which edge points define the widest part of the leaf. These edge points are utilized to find the widest width.

6. Conclusion

6.1 Research Answers

The research aim for this research project is to enable researchers who are conducting experiments on plants that require the collection of phenotyping data to perform the acquisition of data more quickly and precisely. Specifically, by defining a method and developing a program that researchers can use to acquire leaf width for plants in environments with dense leaf clusters.

The research addressed the question posed in the introduction relating to the design of the project.[4]

…… The instance segmentation algorithm that is most effective for leaf segmentation in environments with dense leaf clusters is the Mask R-CNN model….. The best way to train a model to segment measurable leaves out of a dense cluster with overlapping and curling leaves is by …….. The best methodology for obtaining leaf width while overcoming the problems leaf curling causes is through the calculation of geodesic distances between points in a point cloud …… The most effective process at accomplishing the task of measuring leaf width talking raw stereo images as input and the appropriate width measurements as the output is outlined in figure 26.

[4] For more information read section '1.2' of this paper

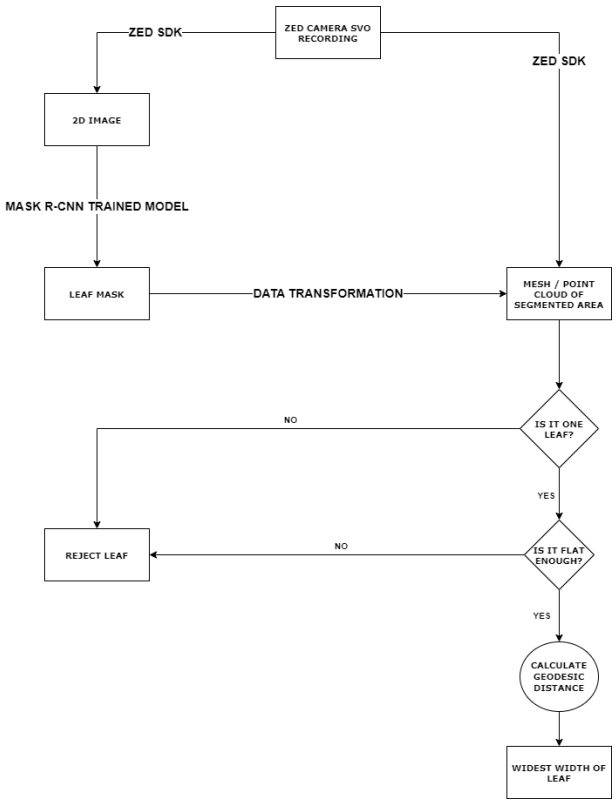


Figure 44. Next Steps Workflow

6.2 Contribution

This project provides plant researchers with a tool to track leaf width in their studies. One of these studies is that of genetically modifying Sorghum to be more resistant to the changes caused by climate change. This tool facilitates plant researchers in conducting their studies quicker, more accurately, and on a larger scale. The greater success of these studies will help prepare humanity for the worsening climate including assisting the potential food scarcity issue.

Although this project was specifically designed for obtaining leaf width for plant genetic researchers, the process and program can be altered and applied to other applications requiring measurements of

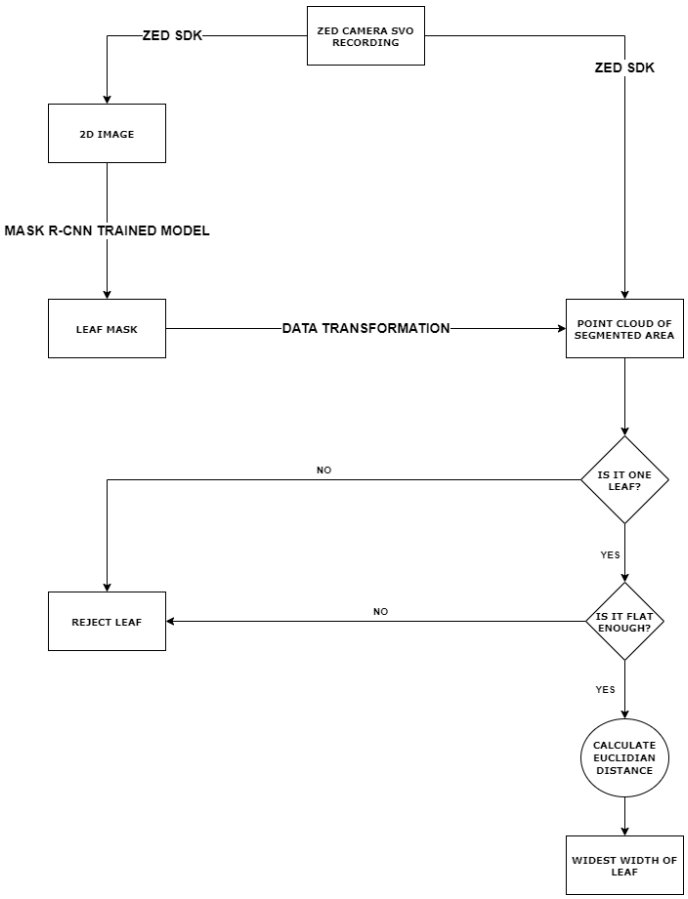
Commented [DG29]: John: Can mention that it is extendable to other plants, like maize as well.

objects. This could include other types of research: automating food harvesting, facilitating detection of large tumors, etc. This project serves as a baseline for future development and process improvement in the space of object measurement with stereo images. And though the project has some flaws it serves as a first step in the construction of widely applicable object measurement capabilities.

6.3 Further Research Opportunities

In planning the

Some studies focus on the Multi-view CNNs (Boulch et al., 2018; Guerry et al., 2017; Kalogerakis et al., 2017), hoping to indirectly realize the understanding of 3D data by strengthening the connection between 2D and 3D via 2D CNNs (Jin et al., 2018a). The main difficulties of this method include that it is hard to determine the angle and the quantity of projection from a point cloud to a 2D image, and how to re-project the segmented models from 2D to 3D space. [PlantNet a dual fun]


ZED CAMERA SVO RECORDING
ZED SDK
ZED SDK
2D IMAGE
MASK R-CNN TRAINED MODEL
LEAF MASK
DATA TRANSFORMATION
POINT CLOUD OF SEGMENTED AREA
IS IT ONE LEAF?
NO
YES
REJECT LEAF
NO
IS IT FLAT ENOUGH?
YES
CALCULATE EUCLIDIAN DISTANCE
WIDEST WIDTH OF LEAF

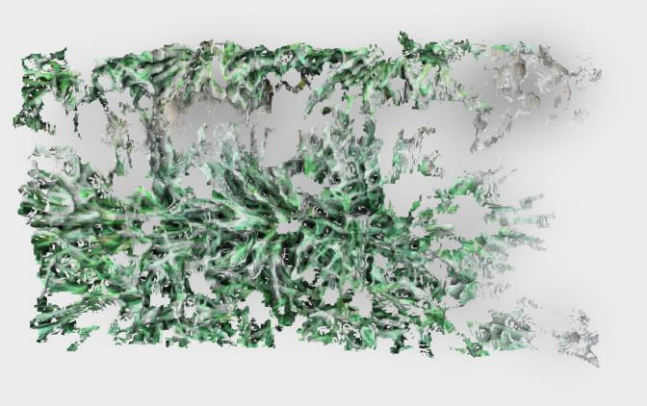


Figure 45. Mesh

Appendix A: ZED Camera Image Capture and Processing

A.1 Overview

TO DO

A.2 Code

TO DO

```
cam = sl.Camera()

init = sl.InitParameters()
init.camera_resolution = sl.RESOLUTION.HD2K
init.depth_mode = sl.DEPTH_MODE.ULTRA
init.SDK_verbose = 1
init.SDK_verbose_log_file = "/home/user/Documents/SVO_recording/SDK_log_file/log_file"
# other initial parameters have been left blank because default value is adequate

recording_param = sl.RecordingParameters(path_output, sl.SVO_COMPRESSION_MODE.H265)

runtime = sl.RuntimeParameters()
cam.set_camera_settings(sl.VIDEO_SETTINGS.BRIGHTNESS, -1)
cam.set_camera_settings(sl.VIDEO_SETTINGS.CONTRAST, -1)
cam.set_camera_settings(sl.VIDEO_SETTINGS.HUE, -1)
cam.set_camera_settings(sl.VIDEO_SETTINGS.SATURATION, -1)
cam.set_camera_settings(sl.VIDEO_SETTINGS.SHARPNESS, -1)
cam.set_camera_settings(sl.VIDEO_SETTINGS.GAMMA, -1)
cam.set_camera_settings(sl.VIDEO_SETTINGS.GAIN, 0)
cam.set_camera_settings(sl.VIDEO_SETTINGS.EXPOSURE, 5)
cam.set_camera_settings(sl.VIDEO_SETTINGS.WHITEBALANCE_TEMPERATURE, -1)
# other video settings have been left blank because default value is adequate

frames_recorded = 0

while frames_recorded < 100:
    if cam.grab(runtime) == sl.ERROR_CODE.SUCCESS :
        frames_recorded += 1
```

Figure 46. Capturing an .SVO recording with the ZED Camera

```
input_type = sl.InputType()
input_type.set_from_SVO_file(SVO_input_path)

init = sl.InitParameters(input_t=input_type, SVO_real_time_mode=False)

```

```
cam = sl.Camera()

runtime = sl.RuntimeParameters()
mat = sl.Mat()

cam.retrieve_image(mat, sl.VIEW.LEFT)
fd = cv2.imwrite(output_path_image, mat.get_data())

cam.retrieve_measure(mat, sl.MEASURE.XYZ)

pc_list = []

for i in range(0, mat.get_width()):
    for j in range(0, mat.get_height()):
        err, val = mat.get_value(i,j)

        if not np.isnan(val[0]) and not np.isinf(val[0]):
            pc_list.append(val[0:3])
```

Figure 47. ZED SDK Code for processing SVO recording into 2D image and point cloud

```
augmentation = iaa.SomeOf((4), [
    iaa.Fliplr(0.5),
    iaa.Flipud(0.5),
    iaa.OneOf([iaa.Affine(rotate=90),
               iaa.Affine(rotate=180),
               iaa.Affine(rotate=270)]),
    iaa.Multiply((0.8, 1.5)),
    iaa.GaussianBlur(sigma=(0.0, 5.0)),
    iaa.WithColorspace(
        to_colorspace="HSV",
        from_colorspace="RGB",
        children=iaa.WithChannels(0, iaa.Add((0, 50))))
])
```

Figure 48. Image Augmentation Configuration

Appendix B: 2D Representation of Point Cloud

B.1 Overview

TO DO

B.2 Code

TO DO

Appendix C: Instance Segmentation Model

C.1 Overview

TO DO

C.2 Code

TO DO

Table 5. Configurations for Matterport's Mask R-CNN Model

BACKBONE	resnet101
BACKBONE_STRIDES	[4, 8, 16, 32, 64]
BATCH_SIZE	1
BBOX_STD_DEV	[0.1 0.1 0.2 0.2]
COMPUTE_BACKBONE_SHAPE	None
DETECTION_MAX_INSTANCES	100
DETECTION_MIN_CONFIDENCE	0.9
DETECTION_NMS_THRESHOLD	0.3
FPN_CLASSIF_FC_LAYERS_SIZE	1024
GPU_COUNT	2
GRADIENT_CLIP_NORM	5
IMAGES_PER_GPU	1
IMAGE_CHANNEL_COUNT	3
IMAGE_MAX_DIM	1024
IMAGE_META_SIZE	14
IMAGE_MIN_DIM	800
IMAGE_MIN_SCALE	0
IMAGE_RESIZE_MODE	square
IMAGE_SHAPE	[1024 1024 3]
LEARNING_MOMENTUM	0.9
LEARNING_RATE	0.001
LOSS_WEIGHTS	{'rpn_class_loss': 1.0, 'rpn_bbox_loss': 1.0, 'mrcnn_class_loss': 1.0, 'mrcnn_bbox_loss': 1.0, 'mrcnn_mask_loss': 1.0}
MASK_POOL_SIZE	14
MASK_SHAPE	[28, 28]
MAX_GT_INSTANCES	100
MEAN_PIXEL	[123.7 116.8 103.9]
MINI_MASK_SHAPE	(56, 56)

NAME	leaves_zed
NUM_CLASSES	2
POOL_SIZE	7
POST_NMS_ROIS_INFERENCE	1000
POST_NMS_ROIS_TRAINING	2000
PRE_NMS_LIMIT	6000
ROI_POSITIVE_RATIO	0.33
RPN_ANCHOR_RATIOS	[0.5, 1, 2]
RPN_ANCHOR_SCALES	(32, 64, 128, 256, 512)
RPN_ANCHOR_STRIDE	1
RPN_BBOX_STD_DEV	[0.1 0.1 0.2 0.2]
RPN_NMS_THRESHOLD	0.7
RPN_TRAIN_ANCHORS_PER_IMAGE	256
STEPS_PER_EPOCH	100
TOP_DOWN_PYRAMID_SIZE	256
TRAIN_BN	FALSE
TRAIN_ROIS_PER_IMAGE	200
USE_MINI_MASK	TRUE
USE_RPN_ROIS	TRUE
VALIDATION_STEPS	50
WEIGHT_DECAY	0.0001

Appendix D: Leaf Scaling

D.1 Overview
TO DO

D.2 Code
TO DO

Appendix D: Point Cloud Cropping

D.1 Overview

TO DO

D.2 Code

TO DO

References

[1] J. Weyler, F. Magistri, P. Seitz, J. Behley, and C. Stachniss, "In-Field Phenotyping Based on Crop Leaf and Plant Instance Segmentation," *Proc. - 2022 IEEE/CVF Winter Conf. Appl. Comput. Vision, WACV 2022*, pp. 2968–2977, 2022, doi: 10.1109/WACV51458.2022.00302.

[2] G. C. Nelson *et al.*, "Climate change effects on agriculture: Economic responses to biophysical shocks," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 111, no. 9, pp. 3274–3279, Mar. 2014, doi: 10.1073/PNAS.1222465110/SUPPL_FILE/SD03.TXT.

[3] Y. Li *et al.*, "Automatic organ-level point cloud segmentation of maize shoots by integrating high-throughput data acquisition and deep learning," *Comput. Electron. Agric.*, vol. 193, p. 106702, Feb. 2022, doi: 10.1016/J.COMPAG.2022.106702.

[4] J. Schrader, G. Pillar, and H. Kreft, "Leaf-IT: An Android application for measuring leaf area," *Ecol. Evol.*, vol. 7, no. 22, pp. 9731–9738, Nov. 2017, doi: 10.1002/ECE3.3485.

[5] D. Li *et al.*, "PlantNet: A dual-function point cloud segmentation network for multiple plant species," *ISPRS J. Photogramm. Remote Sens.*, vol. 184, pp. 243–263, Feb. 2022, doi: 10.1016/J.ISPRSJPRS.2022.01.007.

[6] K. Turgut, H. Dutagaci, G. Galopin, and D. Rousseau, "Segmentation of structural parts of rosebush plants with 3D point-based deep learning methods," *Plant Methods*, vol. 18, no. 1, pp. 1–23, Dec. 2022, doi: 10.1186/S13007-022-00857-3/TABLES/11.

[7] L. Xu, Y. Li, Y. Sun, L. Song, and S. Jin, "Leaf instance segmentation and counting based on deep object detection and segmentation networks," *Proc. - 2018 Jt. 10th Int. Conf. Soft Comput. Intell. Syst. 19th Int. Symp. Adv. Intell. Syst. SCIS-ISIS 2018*, pp. 180–185, Jul. 2018, doi: 10.1109/SCIS-ISIS.2018.00038.

[8] J. Champ, A. Mora-Fallas, H. Goëau, E. Mata-Montero, P. Bonnet, and A. Joly, "Instance segmentation for the fine detection of crop and weed plants by precision agricultural robots," *Appl. Plant Sci.*, vol. 8, no. 7, p. e11373, Jul. 2020, doi: 10.1002/APS3.11373.

[9] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation." pp. 652–660, 2017.

[10] R. Girshick, "Fast R-CNN." pp. 1440–1448, 2015. Accessed: Jun. 05, 2023. [Online]. Available: https://github.com/rbgirshick/

[11] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, Accessed: Jun. 05, 2023. [Online]. Available: https://github.com/

[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Mar. 2017, doi: 10.1109/TPAMI.2018.2844175.

[13] J. R. Quinby, "Leaf and Panicle Size of Sorghum Parents and Hybrids1," *Crop Sci.*, vol. 10, no. 3, pp. 251–254, May 1970, doi: 10.2135/CROPSCI1970.0011183X001000030013X.

[14] R. A. Potamias, A. Neofytou, K. M. Bintsi, and S. Zafeiriou, “GraphWalks: Efficient Shape Agnostic Geodesic Shortest Path Estimation.” pp. 2968–2977, 2022.

[15] A. Agathos and P. Azariadis, “Optimal Point-to-Point geodesic path generation on point clouds,” *Comput. Des.*, vol. 162, p. 103552, Sep. 2023, doi: 10.1016/J.CAD.2023.103552.