

AN ABSTRACT OF THE FINAL REPORT OF

Josyula Gopala Krishna for the degree of Master of Science (M.S.) in Robotics
presented on June 8, 2023.

Title: Branch Reconstruction and Modelling For Pruning

Abstract approved: _____

Cindy Grimm

Successful pruning relies on accurately identifying the 3D structure of tree branches and leaders. However, this task is arduous in an agricultural setting due to the complexity of scenes, the presence of clutter, and variable weather conditions. This project addresses these challenges by leveraging advancements in 2D image segmentation and utilizing RGB and Depth data from a camera to generate precise 3D geometry for the tree. To achieve accurate tree structure identification, we employ a pre-trained MaskRCNN instance segmentation model trained on a smaller labeled dataset to obtain tree segmentation masks. Our algorithm leverages this mask and depth information to extract the geometry of branches and leaders, providing a comprehensive understanding of the tree's structure. Subsequently, we use three methods to estimate the branch radius, enabling the construction of a detailed 3D mesh representation. This mesh is then reprojected onto the image, resulting in a refined segmentation mask that captures the intricate shape of the

tree. The algorithm demonstrates remarkable performance, with a 97% accuracy in branch radius estimation and a success rate of 79.2% in generating masks for unseen data. These results showcase the algorithm's effectiveness in handling the challenges posed by agricultural environments and significantly contribute to the advancement of accurate pruning in cherry tree farms and enhance overall crop management.

©Copyright by Josyula Gopala Krishna
June 8, 2023
All Rights Reserved

Branch Reconstruction and Modelling For Pruning

by

Josyula Gopala Krishna

A FINAL REPORT

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science (M.S.)

Presented June 8, 2023
Commencement June 2023

Master of Science (M.S.) final report of Josyula Gopala Krishna presented on June 8, 2023.

APPROVED:

Major Professor, representing Robotics

Director of the School of Mechanical, Industrial, and Manufacturing Engineering

Dean of the Graduate School

I understand that my final report will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my final report to any reader upon request.

Josyula Gopala Krishna , Author

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to my advisor Dr. Cindy Grimm who provided invaluable support and guidance throughout this research. I am deeply grateful to Dr. Joe Davidson and Dr. Stefan Lee for their continuous support, mentorship, and invaluable insights. Their expertise and encouragement have been instrumental in shaping the direction of this project.

I would also like to extend my gratitude to all my lab friends for their camaraderie, stimulating discussions, and unwavering support. Their contributions, both intellectually and emotionally, have been invaluable throughout this journey.

I am indebted to everyone who has supported me, whether through feedback, discussions, or lending a helping hand during the various stages of this research. Your input and encouragement have been incredibly valuable, and I am thankful for your presence in my life.

I would like to acknowledge the funding agencies that have provided financial support for this project. Their investment has enabled me to pursue this research and make meaningful contributions to the field.

I am grateful to Oregon State University for providing a conducive academic environment and the necessary resources for conducting this research. The institution's commitment to excellence has been a driving force behind my academic and research endeavors.

Lastly, I would like to express my profound appreciation to my family. Their unwavering love, support, and understanding have been my pillar of strength

throughout this journey. I am grateful for their patience, encouragement, and belief in me.

To all those mentioned and to anyone else who has played a part in this endeavor, thank you from the bottom of my heart. Your contributions have made a significant impact on this research, and I am truly grateful for your support.

TABLE OF CONTENTS

	<u>Page</u>
1 Introduction	1
1.1 Motivation	1
1.2 Contribution	1
1.3 Organization	2
2 Related Work	3
2.1 Segmentation	3
2.2 Pruning	4
2.3 Automatic Labelling	4
2.4 Structure Extraction	5
3 Background	7
3.1 MaskRCNN	8
3.2 Camera Projection Matrices	9
3.3 3D Mesh Generation	12
3.4 Width Estimation	14
3.5 Method	15
4 Results	17
5 Discussion	22
5.1 Modes of Failure	22
6 Conclusion	24
Bibliography	25

LIST OF FIGURES

<u>Figure</u>		<u>Page</u>
3.1	In (a) Image Formation and (b) Image Projection is shown, images taken from [18]	10
3.2	The pixels along the width of the branch are projected to world coordinates using the inverse of the projection matrix, the distance between all the points projected into the world coordinates is averaged to obtain the width, as an example in the figure, the distance between two points p_1 and p_2 in the image is shown to be projected into the world coordinates p_{1_w}, p_{2_w} and the distance between them is calculated using the 2-norm	14
3.3	A segment of the leader across the z-axis is projected into the world frame and a circle is fit using RANSAC, this procedure is repeated along the length of the mask, and the median is taken to be an approximation of the treewidth.	15
3.4	The width estimation using triangulation is described in the above picture, process starts by getting the average depth along the branch and using this to calculate the pixel density per meter to estimate the width of the branch	16
4.1	Left to right shows the actual RGB image, segmentation mask, and the edges of the branch.	17
4.2	The three Bezier curves are shown in the image, the bezier is fit to align with the center and the bezier on the boundaries is offset from the center to align with the boundaries of the leader.	18
4.3	The radius estimates across the samples for all three approaches are shown in the figure, RANSAC estimates are observed to be most divergent due to the variation in the data for circle fit. While radius calculation along the branch has been shown to closely approximate the radius of the branch. Triangulation is shown to give the best results.	19

LIST OF FIGURES (Continued)

<u>Figure</u>		<u>Page</u>
4.4	The histogram plot for radius estimates from triangulation and radius along the width of the branch is shown in the figure, the actual radius is also plotted in red, the triangulation gives a mean of 15.76mm while the other method gives a mean of 15.96mm while the actual radius is 15.48mm	20
4.5	Mesh generated from the branch Bezier estimates	20
4.6	Segmentation mask produced by projecting the mesh back to image	21
5.1	The original image, the mask, and the reconstructed mesh are shown here, the reconstruction is inaccurate due to the split mask	22
5.2	The width could be not estimated due to the depth unavailability at the boundaries	23

Chapter 1: Introduction

1.1 Motivation

Sweet cherry tree pruning is a vital operation for tree health and throughput of the farm, this is an extremely labor-intensive operation, and due to the decline in farming labor, there's a drive toward developing automatic pruning systems, Automatic pruning systems, rely heavily on machine vision and instance segmentation, an example of such a system could be found in our work[22]. In this system, we use a UR5 robot arm equipped with a cutter at the end to effector for accurate pruning of the cherry trees. These types of systems are expected to perform in all weather conditions, therefore, training it on diverse data is crucial for accurate segmentation. Therefore we develop an algorithm that can leverage a model trained on a smaller dataset which might produce incomplete or partially correct masks and produce accurate 3D structure and labeled data for training by leveraging the RGB and Depth information along with mask output from the model.

1.2 Contribution

The objective of this project is two-fold:

- Extract branch geometry information such as the radius of the branch, and the structure of the branch.

- Produce improved image segmentation masks using the RGB and Depth data from an Intel RealSense Camera.

and

1.3 Organization

The next chapter discusses related work in segmentation and trunk width estimation of the branches. Later we provide the algorithm details and finally show the results of the approach tested in the lab setup.

Chapter 2: Related Work

2.1 Segmentation

Due to the recent success of computer vision in many applications, image segmentation has been a central task in many applications such as autonomous driving, clustering, medical image analysis, etc. Essential task of image segmentation is to partition the image into "segments" pixel-by-pixel, although numerous algorithms have been developed in the past, deep learning based segmentation architectures have shown uncompromisingly accurate results for a wide range of applications[14].

A deep learning architecture consists of a set of "convolutional layers" which extract various features of the images such as the objects, edges, color or texture-based information while the data is convolved through these layers to produce a bounding box and a pixel-by-pixel segmentation mask.

For an agriculture-based setting recent work on segmentation for apple tree branches detection and segmentation was done in [11] where the authors utilize a very simple architecture called SegNet to produce 92% accurate predictions from the images, the authors manually annotate 210 images for training, this number is really insufficient for modern day architectures such as MaskRCNN and Mask2Former, which require a large amount of data usually in order of thousands to train.

2.2 Pruning

Pruning refers to cutting of the dormant branches during winter or summer of the tree to improve the yield of the crop and maintain the health of the crop. Successful Pruning entirely depends on the accurate segmentation of the branches from the image, as noted in [22]. A majority of pruning systems are designed to work after visual processing of the data as noted in [4].

2.3 Automatic Labelling

A large amount of publicly available data for the agricultural domain is labeled manually for example PlantVillage a data set by Penn State University for plant disease detection has been labeled manually by domain experts [15], an extensive list of other manually labeled data resources can be found here [12]. It is known that labeling in the agricultural domain is a difficult task due to the nature of data present in the domain, which is usually irregularly shaped and occluded with the elements in the domain. Since the supervised training techniques require a large amount of training data which is accurately labeled to obtain results, this task get very laborious.

Recently authors of [20] have developed an image annotation tool that can assist the user in the annotation of the images in a semi-automatic or fully automatic fashion, providing the user with multiple ways of annotating such as polygons, key points, and freeform annotation and generates the labels in COCO format, which uses the MaskRCNN[7] a popular deep learning framework as the backend.

However, it is noted this tool has not been made to be available publicly yet.

In the paper [3] authors introduce a method to perform unsupervised labeling for weed detection in this work, authors use two recent unsupervised deep clustering algorithms, Joint Unsupervised Learning of Deep Representations and Image Clusters (JULE) and Deep Clustering for Unsupervised Learning of Visual Features (DeepCluster) to automatically annotate 17,095 images to reduce manual annotation, an alternate approach for training without labels but from a similar domain is called transfer-learning [17] and is yet to see its application in the agricultural domain.

The techniques of automatic labeling is still being explored widely and are a topic of major interest for the graphics and vision community in general, In this project we approach automatic image labeling for pruning the cherry trees where there is no human in the loop, unlike the previous work in [20] and would generate a 3D mesh and other desiderata such as leader width estimate which unsupervised techniques such as [1] and [3] do not provide, the labels of the images are produced in COCO format [9] for training a wide range of deep learning models.

2.4 Structure Extraction

Structure extraction of the plants has been an area of interest in monitoring the plants and their applications in modeling farms. Recent methods have focused on template matching and 3D reconstruction. For example, the work in [13] obtains 3D structure using a mix of precompiled templates and finding the nearest match

from the templates for a reconstructed point cloud, while the work in [10] uses the methods of multi-view geometry to reconstruct the plants. while traditional methods focused on image processing techniques to produce a 3D reconstruction. For example, the work of [23] follows a complex procedure using wavelets to remove noise from the background and use a custom procedure to 3D reconstruct the roots of the plants. Volumetric methods were used in the work of [16]. Comparatively, this work provides a much simpler method to extract the 3D structural information from a single image.

Chapter 3: Background

The algorithm has 4 phases outlined below:

- A pre-trained deep learning model such as MaskRCNN is used to extract the segmentation masks for the leader and the side branch, and the depth data is used to refine the segmentation mask region.
- A 2D Bezier curve with estimated width is fit to the mask such that it is center aligned with mask, this curve would be offset such that it aligned with image edges
- Three techniques to estimate the 3D width of the segmented region
 - Find the points on the extremities of the branch and calculating the distance between them.
 - Fit a circle for the 3d-point cloud along the vertical leader and select the median radius.
 - Use triangulation to find the width of the branch.
- Construct a 3D mesh using Frenet frame of 3D curve with radius
- The mesh is reprojected to the image to create the segmentation mask.

In this section, we shall go through each of these phases in detail and to understand the background required for this project

3.1 MaskRCNN

MaskRCNN[7] is a widely used image segmentation model which provides the capabilities of semantic and panoptic segmentation, for semantic segmentation the model produces labels pixel-wise for each object, while in panoptic segmentation, the semantic segmentation labels, as well as the object instances, are produced as the output. The MaskRCNN framework operates in two stages

Stage 1: This stage consists of two networks, a backbone (ResNet, VGG, Inception) and a region proposal network. which are used to obtain a set of region proposals. Region proposals are regions in the feature map which contain the object.

Stage 2: This in stage the network predicts bounding boxes and object classes for each of the proposed regions obtained in Stage 1. Each proposed region can be of a different size.

A more detailed account can be found in the original paper of MaskRCNN[7].

We use a trained version of MaskRCNN with a set of 371 images on which we labeled 5 classes of foreground objects: leaders, side branches, spurs, an “other” branch category used for tree branches that either extended off the side of the image or did not belong in any of the aforementioned categories, and nonbranch objects (primarily wires, wooden posts, and ribbons), which is trained using the detectron2 framework [21] with a ResNeXT backbone more details can be found in [22].

An alternate choice to MaskRCNN is Mask2Former which is a recently proposed

framework [2] that works on a transformer-based backend.

3.2 Camera Projection Matrices

A pinhole camera model is used as the defacto standard for projection i.e. world to image and deprojection i.e. image to world coordinates, An image is formed in a camera when an object placed at a distance 'd' in front of a camera with a focal 'f' the axis along the camera is focal length is taken to be 'Z-axis' and perpendicular to it is taken as 'Y-axis' from the figure by law of similar triangels we get

$$\frac{h'}{h} = \frac{f}{d}$$

this gives the relationship between the actual object size, the object size in the image formed and the focal length, distance.

Lets consider the 3D view of this scenario shown in 3.1a

Similar triangles in the above figure reveal

$$\frac{x}{f} = \frac{X_c}{Z_c} \Rightarrow x = f \frac{X_c}{Z_c}$$

And similarly, we find

$$y = f \frac{Y_c}{Z_c}$$

Now, we want to establish what pixel coordinates (u, v) correspond to these values of (x, y) . First, we note that the so-called principal point $(x = 0, y = 0)$ has the pixel coordinates $(u_0, v_0) = (W/2, H/2)$. Since x and y are measured in meters,

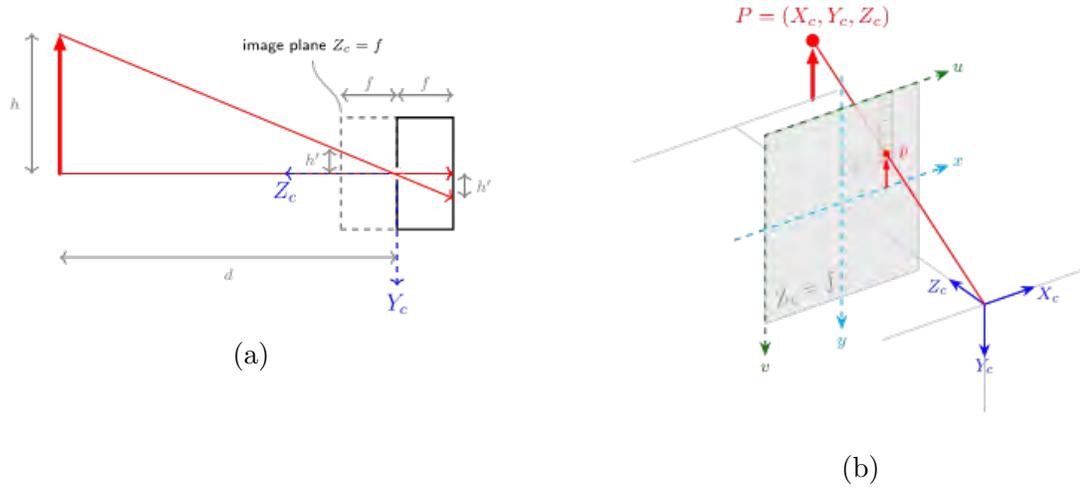


Figure 3.1: In (a) Image Formation and (b) Image Projection is shown, images taken from [18]

we need to know the width and height of one "sensor pixel" in the image plane measured in meters. If the pixel width in meters is k_u and the pixel height is k_v , then

$$u = u_0 + \frac{1}{k_u}x = u_0 + \frac{f}{k_u} \frac{X_c}{Z_c}$$

$$v = v_0 + \frac{1}{k_v}y = v_0 + \frac{f}{k_v} \frac{Y_c}{Z_c}$$

If we define $\alpha_u = f/k_u$ and $\alpha_v = f/k_v$ we can formulate the above equations in matrix form

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{Z_c} \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix}$$

Typically $\alpha_u = \alpha_v = \alpha$.

$$\mathcal{K} = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}$$

The matrix \mathcal{K} is called the intrinsic camera matrix, which gives a relation between the camera coordinate system and the pixels (X_c, Y_c, Z_c) . Another notable transformation in the literature is the world coordinate frame, A point in the world coordinate system is mapped to a point in the camera coordinate system via a rotation matrix \mathcal{R} and a translation t .

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \end{pmatrix} = \mathcal{R} * \begin{pmatrix} X_w \\ Y_w \\ Z_w \end{pmatrix} + t$$

We can write this transformation law just using matrix multiplication

$$\begin{pmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{pmatrix} = \begin{pmatrix} R_{xx} & R_{xy} & R_{xz} & t_x \\ R_{yx} & R_{yy} & R_{yz} & t_y \\ R_{zx} & R_{zy} & R_{zz} & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix} = \mathbf{T}_{cw} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}$$

where we defined the transformation matrix \mathbf{T}_{cw} in the last equality. It is great that we can relate the coordinates of different reference frames just by using a matrix. This makes it easy to compose several transformations, for example, to go from world to camera coordinates and then from camera to road coordinates.

It also enables us to get the inverse transformation using the matrix inverse: The transform from camera to world coordinates is given by the matrix $\mathbf{T}_{wc} = \mathbf{T}_{cw}^{-1}$. To conclude this section, we combine Eq. (3) and (4) into one single equation:

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} R_{xx} & R_{xy} & R_{xz} & t_x \\ R_{yx} & R_{yy} & R_{yz} & t_y \\ R_{zx} & R_{zy} & R_{zz} & t_z \end{pmatrix} \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}$$

Or by defining the intrinsic camera matrix \mathbf{K} and the extrinsic camera matrix $(\mathbf{R} | \mathbf{t})$

$$\lambda \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{K}(\mathbf{R} | \mathbf{t}) \begin{pmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{pmatrix}$$

The content in this section is adapted from [18], for a more detailed overview of the subject, it is recommended to read [6].

The intrinsic and extrinsic matrix is readily available in the IntelTMRealSense[®] SDK [8].

3.3 3D Mesh Generation

3D Mesh generation is a widely studied topic in computer graphics. We use the Frenet-Serret frame method to generate 3D meshes from the Bezier curve fit to

the segmentation mask. There always exists an instantaneous reference frame along a curve called the Frenet-Serret Frame, for point along the Bezier curve the Frenet Frame described in terms of \vec{T} , \vec{N} , \vec{B} Tangent, Normal and Bi-Normal respectively, can be obtained using the following equations:

$$\begin{aligned}\vec{T} &= \frac{\vec{x}'(t)}{\|\vec{x}'(t)\|} \\ \vec{B} &= \frac{\vec{x}'(t) \times \vec{x}''(t)}{\|\vec{x}'(t) \times \vec{x}''(t)\|} \\ \vec{N} &= \vec{B} \times \vec{T}\end{aligned}$$

where the points are sampled points of along a curve instead of functions, namely piecewise linear curves (connecting the samples by lines). Therefore we need to calculate the derivatives by means of finite differences. Thus, the tangent vector for each point \vec{x}_i of our curve can be calculated by

$$\vec{T}_i = \frac{\vec{x}_{i+1} - \vec{x}_{i-1}}{\|\vec{x}_{i+1} - \vec{x}_{i-1}\|}$$

If we have a point on the center of the bezier as $p_c = (x_c, y_c, 0)$ with a radius using the frenet-frame we can construct the 3D mesh as follows:

$$\mathcal{X}_p = \begin{bmatrix} \vec{N} & \vec{B} \\ \vec{T} & p_c \end{bmatrix} * \begin{bmatrix} x(t) \\ y(t) \\ 0 \end{bmatrix} \quad (3.1)$$

Further reading can be done at [5]

3.4 Width Estimation

Three methods to estimate the width of the branch are used in this project.

Width 1:

In this method points on the branch boundaries are projected to world coordinates and the distance between them is calculated. as shown in the figure below:

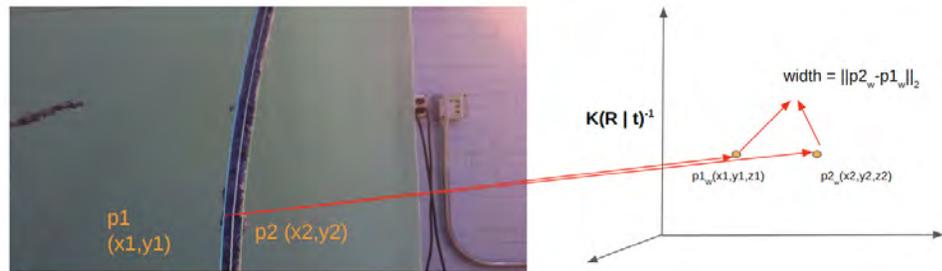


Figure 3.2: The pixels along the width of the branch are projected to world coordinates using the inverse of the projection matrix, the distance between all the points projected into the world coordinates is averaged to obtain the width, as an example in the figure, the distance between two points $p1$ and $p2$ in the image is shown to be projected into the world coordinates $p1_w, p2_w$ and the distance between them is calculated using the 2-norm

Width 2: A circle is fit to the point cloud and the width is the radius of the circle that is fit to the pointcloud.

Width 3:

Width is estimated using the viewing angle of the camera and the pixel distance between two points of the branch, by obtaining a pixel per meter value from the equation

$$(b - a) = 2 * d * \tan\left(\frac{\alpha}{2}\right)$$

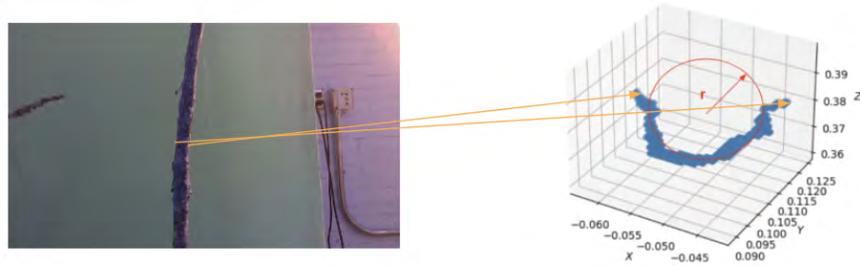


Figure 3.3: A segment of the leader across the z-axis is projected into the world frame and a circle is fit using RANSAC, this procedure is repeated along the length of the mask, and the median is taken to be an approximation of the treewidth.

where $(b - a)$ is the segment width, d is the average depth and α is the viewing angle. This procedure is shown in Figure. 3.4 and further details can be found in [19].

3.5 Method

The steps in the algorithm are as follows

1. Using the MaskRCNN described in this chapter is used to obtain the mask
2. A Bezier curve is fit to the mask of the segmented region to align with the boundaries of the mask.
3. Width estimation can be done in two ways
 -
 - Finding the points on the extremities of the branch and calculating the distance between them

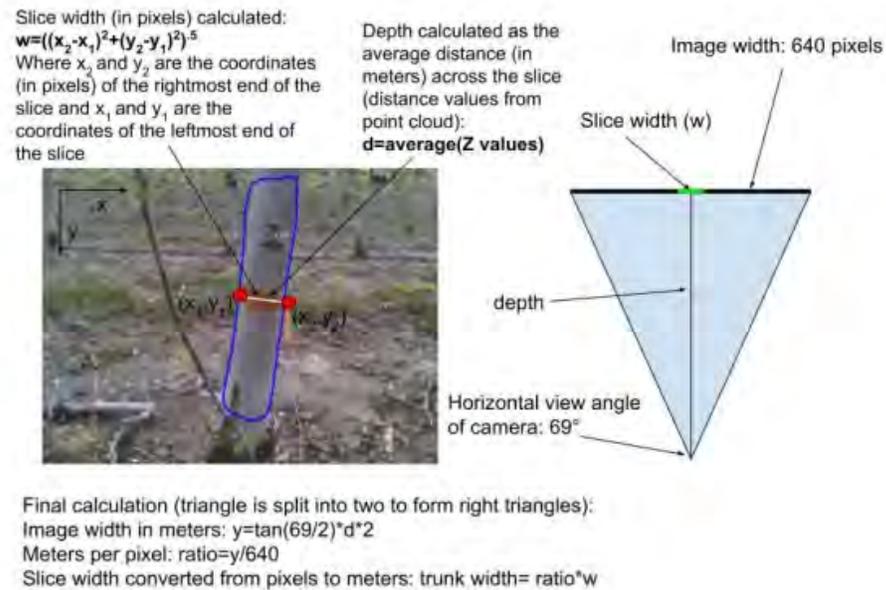


Figure 3.4: The width estimation using triangulation is described in the above picture, process starts by getting the average depth along the branch and using this to calculate the pixel density per meter to estimate the width of the branch

- Fit a circle for the 3d-point cloud along the vertical leader and select the median radius.
 - Triangulation method
4. After the width is obtained a frenet frame is constructed along the bezier curve to build a 3D mesh representation of the mask.
 5. The mesh thus obtained is projected to the image to create the segmentation mask and produce coco annotations

Chapter 4: Results

The dataset is collected using the Intel D435i camera on an in lab setup with 3 trees, for the results we shown the radius estimates for the leader of a tree, whose mean branch width is 15.48mm which is obtained by taking measurements along the length of the leader and averaging them. The data is collected as a bag file and is processed using the Intel RealSense [8] SDK.

A MaskRCNN has been used to extract masks in the first phase, and this mask is used to identify the region of interest in the image, such as the side-branch or leader and an edge contour image is extracted, as shown in Figure.4.1



Figure 4.1: Left to right shows the actual RGB image, segmentation mask, and the edges of the branch.

In the next phase, a Bezier is fit to the edges of the image as shown in the Figure. 4.2 the Bezier curve fit to the center of the image is offset to align with the boundaries of the image, giving a localization of the branch in the image.

The branch is now projected into 3D world frame coordinates using the depth information from the camera and the width is estimated using each of the three



Figure 4.2: The three Bezier curves are shown in the image, the bezier is fit to align with the center and the bezier on the boundaries is offset from the center to align with the boundaries of the leader.

techniques described in the previous section. The branch widths obtained using the three methods are shown in Figure 4.3, it can be seen that RANSAC method may produce the most outliers while triangulation and actual branch width estimates produce much closer results to the ground truth. Further examination of these two techniques can be seen in Figure.4.4.

Triangulation width estimation produces the most accurate estimates, with an average of 15.76mm while the actual branch width method produces an average width of 15.96 for a dataset of 235 images. After the branch width is obtained, the 3d mesh is constructed, an example construction of the 3D mesh is shown in the figure below:

The mask segment obtained after running and projecting the mesh back to the image is shown in the figure below:

The mean running time per image analyzed for 235 examples is found to be 1.68 seconds.

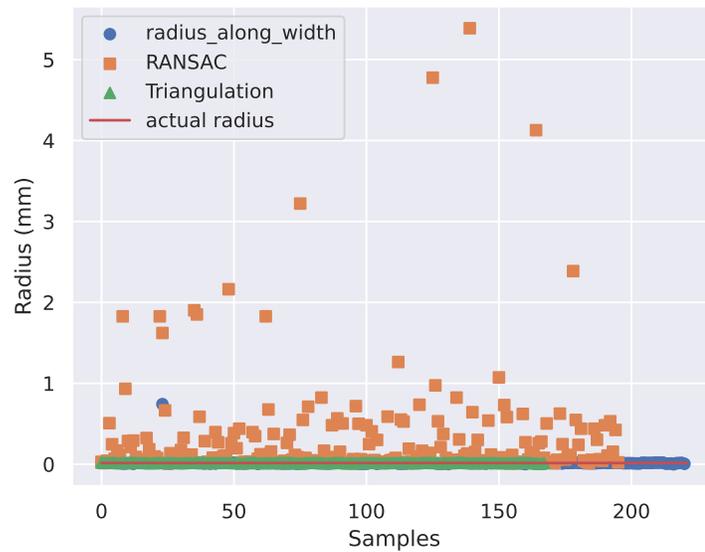


Figure 4.3: The radius estimates across the samples for all three approaches are shown in the figure, RANSAC estimates are observed to be most divergent due to the variation in the data for circle fit. While radius calculation along the branch has been shown to closely approximate the radius of the branch. Triangulation is shown to give the best results.

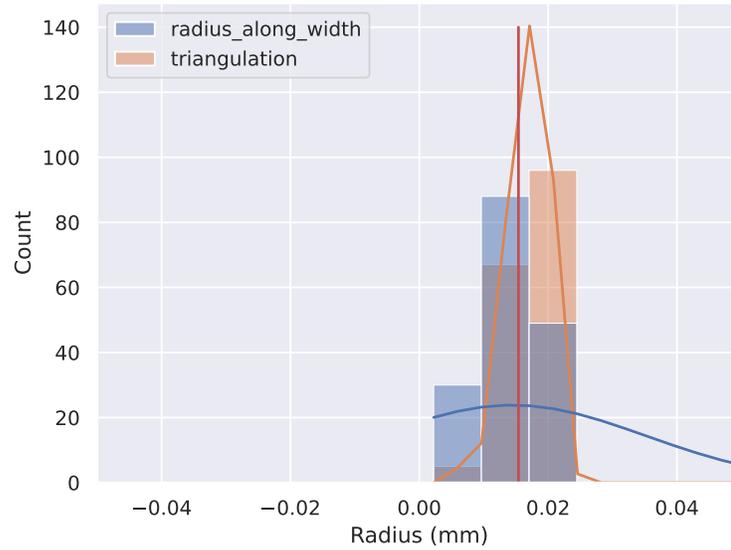


Figure 4.4: The histogram plot for radius estimates from triangulation and radius along the width of the branch is shown in the figure, the actual radius is also plotted in red, the triangulation gives a mean of 15.76mm while the other method gives a mean of 15.96mm while the actual radius is 15.48mm



Figure 4.5: Mesh generated from the branch Bezier estimates



Figure 4.6: Segmentation mask produced by projecting the mesh back to image

Chapter 5: Discussion

5.1 Modes of Failure

The method presented might not produce accurate 3D structure in the following cases:

(a) The mask is split (not clean or missing) 2D curve matches to “average” between the masks of the leader and side branch and hence the 2D curve is not aligned with the original image. 5.1

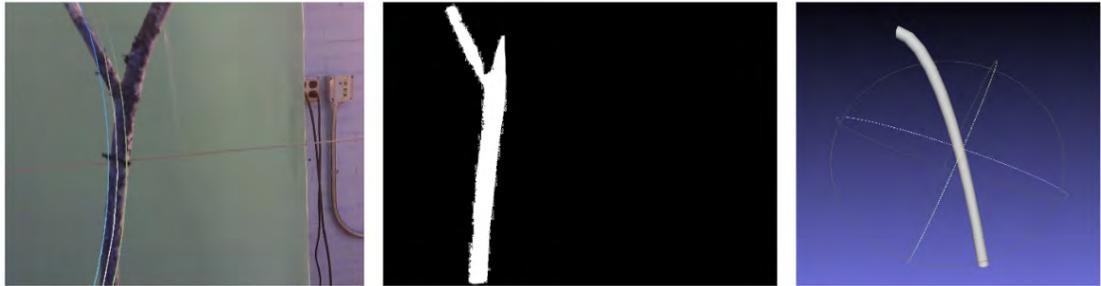


Figure 5.1: The original image, the mask, and the reconstructed mesh are shown here, the reconstruction is inaccurate due to the split mask

(b) When the depth data is not available along the branch then the width estimation of the branch fails 5.2.

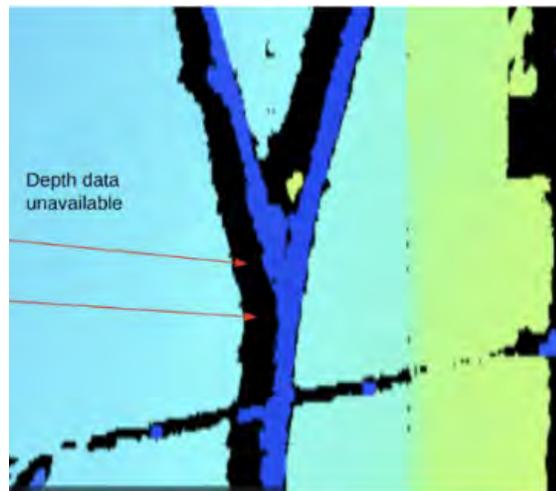


Figure 5.2: The width could be not estimated due to the depth unavailability at the boundaries

Chapter 6: Conclusion

The project showcases an approach to do 3D reconstruction, width estimation and segmentation mask generation relying only on RGB and depth data, the mesh could also be used to augment existing 3D reconstruction systems, another possible direction of utility is to import these meshes into PyBullet for training. A possible future research direction would be to use this method directly for L-System-based modeling to produce more realistic trees.

Bibliography

- [1] M. Dian. Bah, Adel Hafiane, and Raphael Canals. Deep learning with unsupervised data labeling for weeds detection on uav images, 2018.
- [2] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation, 2022.
- [3] Alessandro dos Santos Ferreira, Daniel Matte Freitas, Gercina Gonçalves da Silva, Hemerson Pistori, and Marcelo Theophilo Folhes. Unsupervised deep learning and semi-automatic data labeling in weed discrimination. *Computers and Electronics in Agriculture*, 165:104963, 2019.
- [4] Kadedghe G. Fue, Wesley M. Porter, Edward M. Barnes, and Glen C. Rains. An extensive review of mobile agricultural robotics for field operations: Focus on cotton harvesting. *AgriEngineering*, 2(1):150–174, 2020.
- [5] Andrew S. Glassner, editor. *Graphics Gems*. Academic Press Professional, Inc., USA, 1990.
- [6] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, 2 edition, 2003.
- [7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.
- [8] Intel™. Intel™realsense®. <https://github.com/IntelRealSense/librealsense>, 2023.
- [9] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

- [10] Lu Lou, Yonghuai Liu, Minglan Sheng, Jiwan Han, and John H. Doonan. A cost-effective automatic 3d reconstruction pipeline for plants using multi-view images. In Michael Mistry, Aleš Leonardis, Mark Witkowski, and Chris Melhuish, editors, *Advances in Autonomous Robotics Systems*, pages 221–230, Cham, 2014. Springer International Publishing.
- [11] Yaqoob Majeed, Jing Zhang, Xin Zhang, Longsheng Fu, Manoj Karkee, Qin Zhang, and Matthew D. Whiting. Apple tree trunk and branch segmentation for automatic trellis training using convolutional neural network based semantic segmentation. *IFAC-PapersOnLine*, 51:75–80, 2018.
- [12] Normaisharah Mamat, Mohd Fauzi Othman, Rawad Abdoulghafor, Samir Brahim Belhaouari, Normahira Mamat, and Shamsul Faisal Mohd Hussein. Advanced technology in agriculture industry by implementing image annotation technique and deep learning approach: A review. *Agriculture*, 12(7):1033, Jul 2022.
- [13] Elias Marks, Federico Magistri, and Cyrill Stachniss. Precise 3d reconstruction of plants from uav imagery combining bundle adjustment and template matching. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2259–2265, 2022.
- [14] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey, 2020.
- [15] Sharada P. Mohanty, David P. Hughes, and Marcel Salathé. Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7, 2016.
- [16] Alex Reche-Martinez, Ignacio Martin, and George Drettakis. Volumetric reconstruction and interactive rendering of trees from photographs. *ACM Trans. Graph.*, 23(3):720–727, aug 2004.
- [17] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning, 2018.
- [18] Mario Theers and Mankaran Singh. Algorithms for automated driving. <https://thomasfermi.github.io/Algorithms-for-Automated-Driving/LaneDetection/CameraBasics.html>, 2021.

- [19] T. Wang, P. Sankari, J. Brown, A. Paudel, L. He, M. Karkee, A. Thompson, C. Grimm, J.R. Davidson, and S. Todorovic. Automatic estimation of trunk cross sectional area using deep learning. *Collaborative Robotics & Intelligent Systems Institute, Oregon State University*, 2023.
- [20] Patrick Wspanialy, Justin Brooks, and Medhat Moussa. An image labeling tool and agricultural dataset for deep learning, 2020.
- [21] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019.
- [22] Alexander You, Nidhi Parayil, Josyula Gopala Krishna, Uddhav Bhattarai, Ranjan Sapkota, Dawood Ahmed, Matthew Whiting, Manoj Karkee, Cindy M. Grimm, and Joseph R. Davidson. An autonomous robot for pruning modern, planar fruit trees, 2022.
- [23] Ying Zheng, Steve Gu, Herbert Edelsbrunner, Carlo Tomasi, and Philip Benfey. Detailed reconstruction of 3d plant root shape. In *2011 International Conference on Computer Vision*, pages 2026–2033, 2011.