Computer Vision-Based Phenotyping in Agriculture: Leveraging Semantic Information for Non-Destructive Small Crop Analysis

Harry Freeman
CMU-RI-TR-23-59
August 2, 2023

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
George Kantor, *chair*
Michael Kaess
Daniel McGann

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

Abstract

Fast and reliable non-destructive phenotyping of plants plays an important role in precision agriculture, as the information enables farmers to make real-time crop management decisions without affecting yield. These decisions encompass a wide range of tasks, including harvesting, disease and pest management, quality control, and scientific breeding.

To non-destructively phenotype crops, computer and stereo-vision based methods are commonly used, as they are low-cost and resolve finer details compared to other systems such as LiDAR. However, most approaches are targeted towards large and sparsely populated crops, where occlusions, wind, and sensor error pose less of a challenge.

In this thesis, we tackle the problem of using computer vision to non-destructively phenotype smaller crops by leveraging semantic information. First, we present a method for creating 3D models of sorghum panicles by using seeds as semantic 3D landmarks. To evaluate performance, we develop an unsupervised metric to assess point cloud reconstruction quality in the absence of ground truth. We then use the model to estimate seed count, and demonstrate that this method outperforms extrapolating counts from 2D images, a common approach used in similar applications.

Next, we present a computer vision-based method to measure sizes and growth rates of apple fruitlets. With images collected by a hand-held stereo camera, our system fits ellipses to fruitlets to measure their diameters. To measure growth rates, we utilize an Attentional Graph Neural Network to associate fruitlets across days. We provide quantitative results on data collected in an apple orchard, and demonstrate that our system is able to predict abscise rates within 3% of the current method with a 7 times improvement in speed, while requiring significantly less manual effort.

Finally, we build upon our sizing pipeline by designing a robotic system to make the sizing process fully autonomous. We present a next-best-view planning approach targeted towards sizing smaller fruit. We utilize semantically labeled regions of interest to sample viewpoint candidates, along with an attention-guided information gain mechanism to generate optimal camera poses. Additionally, a dual-map representation is used to improve speed. When sizing, a robust estimation and clustering approach is introduced to associate fruit detections across images. We demonstrate that our system can effectively size small fruit in occluded environments.

Acknowledgments

I would first like to express gratitude to Eric Schneider for being such a great desk-mate and for his invaluable partnership on the sorghum project. I also want to thank John Kim and Mark Lee for their efforts in collecting ground truth data.

I would also like to convey a wholehearted thank you to Abhi Silwal for his indispensable help on the fruitlet sizing project, in addition to Mohamad Qadri and Zack Rubinstein. In addition, I want to thank Dan Cooley, Jon, Clements, Paul O'Connor, and the University of Massachusetts Amherst Cold Spring Orchard for their welcoming hospitality and assistance during our field testing.

I want to thank all my labmates and friends for creating such a fun, supportive, and engaging community. I also want to give a very special thank you to my parents, sister, grandparents, and beloved family members for their steadfast inspiration and ongoing encouragement.

Lastly, I would like to extend my deepest appreciation to my advisor George Kantor for his invaluable advice and unwavering support throughout my Master's experience.

Funding

I would like to thank our sponsors and partners for their generosity and commitment towards this work. The support of ARPA-E TERRA DE-AR0001134, USDA NIFA 20216702135974, USDA NIFA 2020014691022394, NSF Robust Intelligence 1956163, and USDA NIFA AIIRA AI Research Institute 2021-67021-35329 was fundamental in making the research presented throughout this thesis possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

Recent advancements in computer vision have allowed farmers to deploy autonomous plant monitoring solutions to more efficiently inspect vast quantities of crops. Computer vision-based systems provide fast and reliable information for downstream tasks, such as harvesting [25, 63], phenotyping [11, 81], and yield prediction [15, 19, 73], ultimately allowing farmers to make real-time crop management decisions.

Phenotyping is particularly important because it enables agricultural specialists to observe specific characteristics of plants and assess their overall quality and health. In agriculture, a phenotype refers to the observable characteristics of a plant as a result of the interaction between its genetic makeup and the environment. Phenotypes encompass various traits such as size, shape, color, internal temperature, and many other quantitatively measurable features. By analyzing phenotypes, agriculturalists can make faster and more informed decisions for a variety of tasks, including disease and pest management, quality control, and scientific breeding.

Because many phenotypes are observable, agriculturalists have set out to integrate computer vision-based systems to automate the labor-intensive tasks required to collect phenotyping data. Examples include fruit and seed counting [57, 82], measuring plant height and stalk size [1, 111], biomass prediction [101], and disease classification [33], which otherwise would be completed by teams of people using visual inspection or manual tools. However, most of these automated phenotyping systems are either

dedicated to larger crops or are destructive. Non-destructive phenotyping is often preferred, as it is a non-invasive approach that preserves genetic material, allows researchers to track plant development over time, and does not interrupt growth. Yet it is challenging to non-destructively phenotype smaller, more densely populated crops. This is because they are more difficult to detect and track as a result of their small size and occlusions from the surrounding environment. As well, wind and sensor error have a more significant effect, making the tasks of data association and 3D modelling much more challenging.

In this thesis, we evaluate automated computer vision-based methods for the purpose of non-destructively measuring phenotypes of smaller grains and fruit, specifically sorghum seed counts and apple fruitlet sizes. We do this by leveraging semantic information to improve tasks such as global registration, association, and viewpoint planning. Although we focus on these two crops, the presented methods could be used to phenotype other fruits and grains of comparable size.

1.2 Contributions

In Chapter 4, we present a computer vision pipeline for non-destructive seed counting of sorghum panicles for early forecasting of yield. Accurate forecasting is valuable for sorghum breeding programs, as it would allow faster decision-making on variant suitability, which could expedite the current five-year breeding process [42]. Seed count would be a valuable phenotypic trait, but it is currently not possible to sample in a non-destructive way. In contrast to the large and separated fruits typically inspected, sorghum seed counting is more challenging from a computer vision perspective. This is because the seeds are much smaller than typically studied crops, averaging 3.3mm in diameter, making them difficult to detect and track. In addition, there is significantly more occlusion due to the dense packing and clutter from husks. Although there has been work on 2D image-based instance counts for other crops [24, 48, 57], it is still difficult to obtain a high accuracy count with sorghum. To address these issues, we create an accurate 3D model of a sorghum panicle from multiple stereo views using seeds as semantic 3D landmarks in reconstruction. To evaluate the model in the absence of ground truth, we present an unsupervised metric for assessing point cloud reconstruction quality. We then use the model to estimate seed count. Using

our proposed method, we acquire a more realistic count than using 2D image-based approximations. Our specific contributions are

- A novel 3D reconstruction method that utilizes seeds as semantic 3D landmarks to produce an accurate model of a sorghum panicle.
- A new metric for assessing point cloud reconstruction quality in the absence of ground truth.
- A novel method for extracting seed counts from point clouds, which involves extending 2D image processing techniques into 3D to identify local maxima in a density map.

In Chapter 5, we introduce a computer vision-based system for measuring the sizes and growth rates of apple fruitlets from single stereo-image pairs. Measuring growth rates of apple fruitlets is important because it allows apple growers to determine when to apply chemical thinners to their crops in order to optimize yield. The current practice of obtaining growth rates involves using calipers to manually record sizes of hundreds to thousands of fruitlets across multiple days. Due to the number of fruitlets that need to be sized, this method is laborious, time-consuming, and highly subject to human error. With images collected by a hand-held stereo camera, our system segments and fits ellipses to fruitlets to measure their diameters. To automate the measurement of growth rates, we develop a novel Graph Neural Network [29] approach that uses semantic features for temporal fruit association, which, to the best of our knowledge, is the first of its kind used in agriculture. We demonstrate that our method produces measurements comparable to those taken by using calipers, with the ability to reduce manual effort and significantly improve speed. Our key contributions are

- A computer vision-based system to detect, segment, and size apple fruitlets.
- An Attentional Graph Neural Network approach for temporal fruit association.
- Experiments on data collected in a commercial apple orchard.

Despite the improved speed and reduced effort, there are some limitations with this approach. For one, it is not fully autonomous. Human effort is still required to capture reasonable, unoccluded images of the fruitlets. As well, only a single image is used to size the fruit, whereas using information from multiple images may lead to improved results. In Chapter 6, we address these issues by designing a robotic system

to make the sizing process fully autonomous. This task is challenging because fruitlets grow in very occluded environments. There are leaves, branches, and wind, and often fruitlets will occlude one another. The robot needs to be able to reason about the environment and determine the optimal viewpoints to capture images. We achieve this using a next-best-view (NBV) planning approach targeted towards sizing smaller fruit. Previous NBV planners in agriculture are designed to size larger fruit that are sparsely populated. They rely on low-resolution maps and naive association methods that do not generalize across smaller fruit sizes. To overcome these limitations, we present an NBV planning approach that utilizes regions of interest for viewpoint sampling and an attention-guided mechanism for calculating information gain. In addition, we integrate a dual-map representation of the environment that significantly speeds up expensive ray casting operations while maintaining finer occupancy information. To address the challenges of data association in the presence of wind and sensor error, we introduce a robust estimation and graph clustering approach. We demonstrate that our planning method improves sizing accuracy compared to another state-of-the-art planner used in agriculture. Our main contributions are

- A novel next-best-view planner that uses coarse and fine occupancy maps with an attention-guided information gain metric to capture images of smaller fruit.
- A robust estimation and graph clustering approach to associate fruitlet detections across images in the presence of wind and sensor error.
- Quantitative evaluation on data collected by a real robotic system in a commercial apple orchard.

Chapter 2

Related Work

2.1 3D Reconstruction of Plants

For 3D reconstruction in agriculture, most works have been dedicated towards mapping larger fields and rows [13, 83] as opposed to single plants. Orchard rows are reconstructed in [90] by merging views using cylinders fit to trunks. This does not adapt well to sorghum as the stems are too small to fit geometric shapes to. Additionally, localized views of flowers and vines are captured in [74, 94], but they do not get a complete 360° scan. For single plant modelling, reconstructed point clouds have been used to phenotype plants in [12, 37, 61, 79]. However, the datasets used were captured using a high-precision laser scanner which is not available for sorghum nor adaptable to work in the field.

2.2 Fruit and Seed Counting

There has been a significant amount of work dedicated towards counting in agricultural settings. Mapping and estimating the yield of mangoes in occluded environments with a Faster R-CNN [86] detector is presented in [96] and [60]. Mapping and counting grapes by fitting spheres to point clouds in 3D is presented in [71]. While these methods work in their respective domains, they do not extend well to sorghum, where the seeds are smaller and the levels of density and occlusions are much higher, making

them harder to consistently segment and fit shapes to.

There has also been relevant work in estimating seed counts for smaller crops from single 2D images. Counting rice and soybeans with density maps using convolutional neural networks is addressed in [24] and [57] respectively. However, the rice and beans have been stripped from the plant and laid out such that there are few occlusions. Density maps have also been used to count corn kernels on the cob, where the final count is proportional to the density map count as a result of corn's symmetric shape [48]. Similarly, [72] uses a KD-Forest approach to detect grapes in clusters using keypoint-based features and subsequently estimates yield using a scale factor. These methods do not adapt well to counting sorghum seeds because of the panicle's asymmetric shape. Additionally, the authors of [69] use a YOLOv4 [4] network to count sorghum heads in a field in aerial drone images, but do not count seeds on individual panicles.

2.3 Fruit Sizing

There has been significant work dedicated towards sizing fruit in agriculture. In the work of [15], calibration spheres are placed on trees and used as reference scales to estimate the sizes of segmented apples. Similarly, Wang *et al.* [107] are able to size fruits in the field with a smartphone by placing a reference circle of known size behind the fruit. While these methods only require simple segmentation and sizing algorithms, they do not extend well to fruitlets as it is impractical to place reference objects behind hundreds of fruit in occluded environments.

Approaches have also been developed to size fruits in 3D. Reconstruction-based methods are used by [46, 104], where 3D models are created from multiple sensor measurements. However, these methods are computationally expensive and do not perform well with occlusions where reconstructions are often incomplete. To address these issues, automated shape completion methods have been implemented by [54, 62] which fit superellipsoids to accumulated point clouds. These methods either rely on successive frame alignment algorithms, such as Iterative Closest Point, which fail in agricultural environments due to the dynamic structure of plants, or use expensive ray casting operations which are slow when performed at finer resolutions. 3D sizing is performed by [28] where the major-axis of an apple is fit to 3D points collected

from a single camera image and time-of-flight sensor. However, the performance is poor, achieving an accuracy of 69.1%. The work of [31] also sizes apples from single images by fitting spheres in 3D. This would not adapt well to apple fruitlets due their small size making it challenging to capture enough of the fruit's surface.

As an alternative to 3D sizing, 2D photogrammetric methods have been adopted in agriculture. These methods either directly estimate the widths of fruits [28] or fit ellipses to measure the heights and widths [82, 106]. This is advantageous as sizes are able to be extracted quickly from 2D images without the need to aggregate information from multiple views. However, the presented approaches use simple detection and segmentation modules, relying on color information and non-deep features. While these methods work well in their respective domains, they would fail when trying to detect and segment fruitlets. This is because the proximity of fruitlets are much closer together, and their colors blend in with the surrounding leaves.

2.4 Spatio-Temporal Fruit Association

There has been limited work on spatio-temporal association in agriculture, which is required to track growth rates. Temporal plant association methods have been proposed for large rows of crops [10, 14, 21]. However, these methods assume static scenes during data collection and take advantage of the structural similarity of rows across days, which does not hold for apple fruitlets which move, grow, and fall off.

For spatio-temporal association of individual fruit, there has been effort dedicated towards both fruit tracking [60, 114] and identification from different camera views [27]. However, these methods either rely on images taken from consecutive frames, or maintain 3D knowledge about the current scene. Hondo *et al.* [40] develop a deep learning approach to size apples and track their growth over time. However, images are captured from a camera fixed in place, and fruit identification is performed by comparing center coordinates of segmented apples. This does not extend well to apple fruitlets as a result of their small size, close proximity, and number of fruit needed to be sized. The works of [12, 37, 61, 79] focus on the 4D registration of individual plant components. However, the datasets were acquired in a lab setting using a high-precision laser scanner, which does not adapt well to the field.

Recently, a method was proposed for temporal fruit registration of strawberries

using a new feature descriptor that takes into account the position of neighboring fruit [87]. This method, however, assumes a similar depth structure of the point clouds as a result of the captured images being taken by a robot driving down a row of crops. This is not the case for apple fruitlets, where a robotic arm would have to reach in and capture images, resulting in vastly different camera poses.

2.5 Next-Best-View Planning

Next-best-view (NBV) planners are used to determine the next best camera pose that maximizes information gain. Alternative to coverage path planners, which rely on prior information to compute optimal camera poses to cover the area of known static maps [22, 47, 65, 75, 78, 84], NBV planners are used to explore unknown and dynamic environments. They rely on information gain metrics [20, 44] to evaluate candidate viewpoints to effectively explore unknown areas of interest.

There have been previous works dedicated towards NBV planning in agriculture, where approaches either use only current sensor information or build a volumetric map of the environment to determine the next-best-view. Examples of the former include the work of [53], who use a 3D camera array to capture images to size a target fruit of interest. The next best camera pose is selected by computing a gradient to determine the optimal direction in which the visibility of the target is increased. While this method is effective at avoiding local occlusions, it cannot extend globally when multiple targets need to be imaged. As well, in [89], a viewpoint planning approach is presented to count the number of apples in a cluster. They determine the optimal camera pose by maximizing entropy of a set of world hypotheses informed by previous images. However, the method makes simplified occlusion assumptions that do not extend well when sizing smaller fruit.

More commonly, volumetric-based planning approaches have been adopted in agriculture, where occupancy information stored inside a volumetric map is used to guide the planning process. These methods often combine occupancy and region of interest (ROI) information when determining the next best camera pose. In the work of [97], detected ROIs in the form of segmented apples are used to evaluate viewpoints based on a weighted sum of exploration information. This information is then used as input into a Decentralized Monte Carlo Tree Search [3] planner to plan

a sequence of viewpoints for multiple robot arms. However, computing a sequence of viewpoints can be computationally expensive. Zaenker *et al.* [112] address this issue by using ROIs to more informatively sample viewpoints. The information gain for each viewpoint is then approximated by casting rays through the volumetric map. While this is a more greedy approach, as only single points are evaluated instead of a complete path, the computational cost is much less. The authors of [64] try to further reduce the computational complexity by using automated shape completion along with a viewpoint dissimilarity metric to estimate information gain, replacing the need for ray casting. However, this does not extend well to smaller fruit, where shape completion methods traditionally fail as a result of the inability to capture enough of the fruit's surface.

An attention-guided NBV planning approach is presented by Burusa *et al.* [8], where attention regions defined around different plant components are used to restrict which voxels contribute to the information gain. The authors demonstrate that thid led to improved results regarding both reconstruction accuracy and speed. However, the location and size of the attention regions were assumed to be prior knowledge, which is not the case when sizing unknown fruit. As well, they did not consider ROI information when viewpoint sampling, and sample viewpoints on a cylindrical sector which was also assumed to be known beforehand.

Chapter 3

Preliminaries

3.1 Illumination-Invariant Flash Stereo Camera

Object detection and semantic segmentation are traditionally challenging in agricultural environments due to varying illumination conditions. The appearance of fruits and plants in images taken at different times of day may be significantly altered as a result of inconsistent sun position, brightness of day, and shadows caused by moving clouds and neighboring branches and leaves. As a result, it is difficult for networks to generalize, as they are required to have large amounts of training data, which is often difficult to obtain. While data augmentation, transfer learning, and visual pre-training [88] help reduce the amount of data needed to reliably train these networks, having consistent lighting across images significantly boosts performance.

To address these issues, and make the trained networks more robust, all acquired stereo images used throughout this thesis were captured using an in-hand version of the illumination-invariant flash stereo camera presented by Silwal *et al.* [93]. This is an active lighting-based camera system that is able to generate uniform images across varying lighting conditions. Two images taken in our most recent field test can be seen in Fig. 3.1, with one taken in the dark around midnight and one in bright daylight around noon.

A flash-based stereo camera system is used to extract 3D information because of its reliability and low cost. This is common practice in agriculture, as stereo cameras resolve finer details compared to other systems such as LiDAR [99, 100, 109].

Figure 3.1: Two images taken with the in-hand illumination-invariant flash stereo camera from [93], spaced approximately 12 hours apart. Images were taken on 05/19/2023 around midnight (left) and noon (right).

In addition, flash-based stereo systems are more resilient to varying illumination conditions, where RGB-D sensors inconsistently perform [28, 106].

3.2 Stereo Re-Projection

Throughout this thesis, we assume a pinhole camera model to describe the projection of 3D points onto the image plane. A pinhole camera is a mapping $\mathbf{P}$ between a homogeneous 3D world coordinate $\mathbf{X} = [X \quad Y \quad Z \quad 1]^T$ and a 2D image coordinate $\boldsymbol{x} = [x \quad y \quad 1]^T$ represented as

$$\begin{aligned} s\boldsymbol{x} &= \mathbf{P}\mathbf{X} \\ \mathbf{P} &= \mathbf{K}\mathbf{T_c} \\ \mathbf{K} &= \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned} \tag{3.1}$$

Here, $\mathbf{K}$ is the intrinsic camera matrix with focal lengths f_x and f_y along the x and y axes respectively and with principal point (c_x, cy). $\mathbf{T_c} \in SE(3)$ is the pose of the camera in the world frame, and s is a scaling factor.

Prior to data collection, the stereo camera is calibrated to find the left and right intrinsic matrices $\mathbf{K_l}$ and $\mathbf{K_r}$, in addition to $\mathbf{T_r^l} \in SE(3)$ which represents the geometric relationship between the left and right cameras. When a stereo image is captured, distortion is removed using a simple radial and tangential distortion model. The images are then rectified using $\mathbf{K_l}$, $\mathbf{K_r}$, and $\mathbf{T_r^l}$ to make the epipolar lines horizontal. The relationship between a homogeneous 3D world coordinate $\mathbf{X}$ and its corresponding 2D image coordinates in the rectified left and right images $\boldsymbol{x_l}$ and $\boldsymbol{x_r}$ can be described as

$$\begin{aligned} s_l \boldsymbol{x_l'} &= \mathbf{K_l T_c X} \\ s_r \boldsymbol{x_r'} &= \mathbf{K_r T_c X} \\ \mathbf{K_l} &= \begin{bmatrix} f_x' & 0 & c_x' & 0 \\ 0 & f_y' & c_y' & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ \mathbf{K_r} &= \begin{bmatrix} f_x' & 0 & c_x' & -f_x' b \\ 0 & f_y' & c_y' & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned} \tag{3.2}$$

where f_x', f_y', and (c_x', c_y') are the focal lengths and principal point after rectification, and b is the baseline of the rectified stereo cameras. Because the epipolar lines are horizontal, $\boldsymbol{x_l'} = [x_l' \quad y' \quad 1]$ and $\boldsymbol{x_r'} = [x_r' \quad y' \quad 1]$, and the disparity d is $x_l' - x_r'$.

To re-project a point in a stereo image, we estimate the disparity using RAFT-Stereo [59]. RAFT-Stereo is a state-of-the-art deep learning-based stereo matching network that outperforms traditional disparity generation methods, such as SGBM [38], and does not require fine-tuning. This is advantageous as the network does not need to be retrained between datasets. After the disparity is estimated, the point's 3D world coordinate $\mathbf{X}$ can be calculated as

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{b}{d} \begin{bmatrix} x_l' - c_x' \\ y' - c_y' \\ f_x' \end{bmatrix} \tag{3.3}$$

Chapter 4

Sorghum Seed Counting

All work in this chapter was done in collaboration with Eric Schneider.

4.1 Motivation

Non-destructive sorghum seed counting would be beneficial towards experimental plant breeding, as it would enable farmers to rank variants earlier in the growing season without disturbing growth. In this work, we reconstruct a 3D model of a sorghum panicle using multiple viewpoints that span 360° around the stalk. The model is then used to estimate seed count. Due to the small scale of seeds, the model must achieve a high level of accuracy in order to obtain a reasonable count estimate.

4.2 Reconstruction and Seed Counting

4.2.1 System Overview

In order to generate a high-quality 3D model of a sorghum panicle, we set up an automatic data collection process by attaching the flash stereo camera from Section 3.1 to the wrist of a UR5 robot arm. The robot follows a circular trajectory around the panicle as shown in Fig. 4.1, which results in 360° images of each panicle as illustrated in Fig. 4.2. From all images taken, we spatially downsample to only consider images $\mathbf{I}_i \in \mathbb{I}$ and poses $\mathbf{T}_i \in \mathbb{T}$ in the shape of a double ring, spaced 5cm apart, leaving

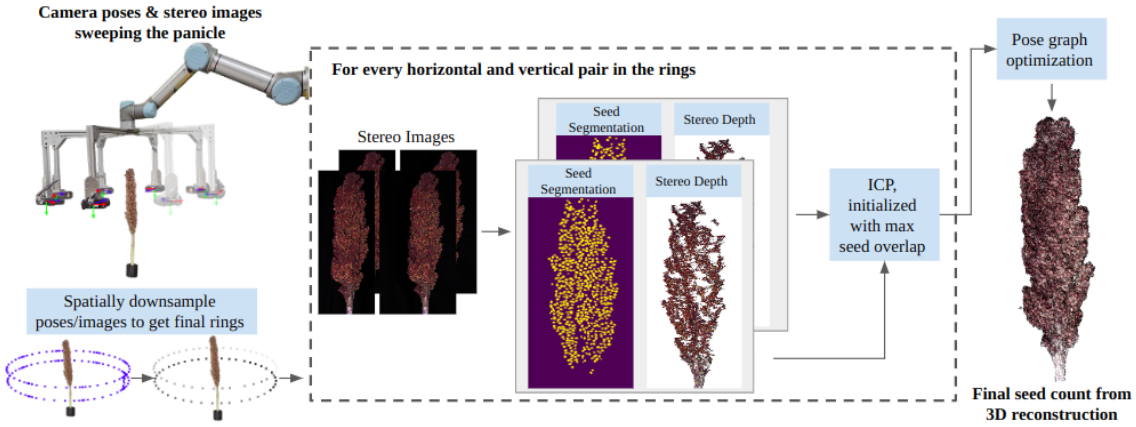


Figure 4.1: 3D Reconstruction pipeline for the sorghum stalk.

roughly 85 images per panicle. A double ring was used because the camera field of view could not capture the entire panicle height. We then use RAFT-Stereo [59] to construct point clouds for each frame. Using Iterative Closest Point (ICP) on segmented seeds, we construct a pose graph that is then optimized to create the final high quality point cloud $\mathbf{C}$. Lastly seed masks are combined between all images $\mathbf{I}_i \in \mathbb{I}$ to obtain a final seed count.

4.2.2 Instance Segmentation

Given a stereo-image pair, we acquire a 3D point cloud semantically labeled with individual sorghum seeds. This is achieved through instance segmentation on 2D images, and re-projecting the masks onto the 3D points. Because of the time-consuming nature of hand-segmenting individual seeds, ten 1440×1080 sorghum images across different species were used to train the network. To augment the dataset and improve inference, each image was split into 120×90 smaller tiles with 50% overlap. An ImageNet-1K pretrained CenterMask [52] network was fine-tuned on the resulting data. Centermask was selected because it qualitatively outperformed Mask R-CNN [36] on our dataset.

During inference, instance segmentation is performed on the 120×90 tiles which are then merged. Masks with low confidence scores are immediately dropped, as well as masks that are at the boundary of the tiles. The remaining masks across tiles are merged if the intersection over union (IOU) is high enough. If not, the higher

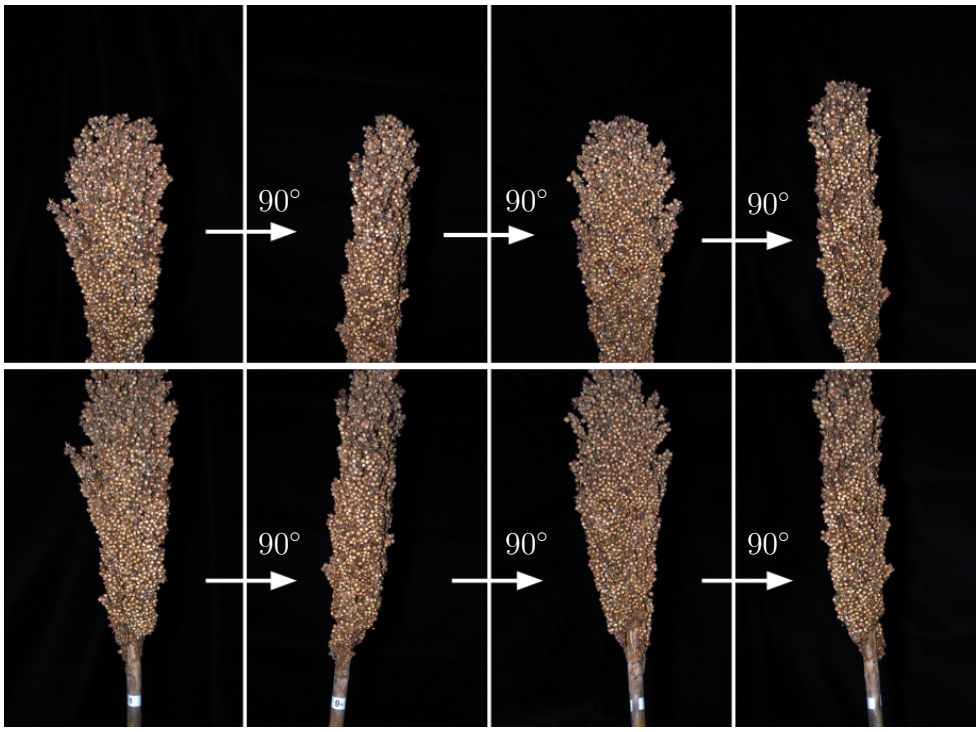


Figure 4.2: Example of a subset of images separated by 90° for both the top and bottom rings.

confidence mask is chosen. After inference, seed masks are re-projected onto the 3D point cloud.

In order to choose high-quality seed points, the re-projected seeds are filtered. First, seeds where more than 15% of the segmented pixels have invalid disparity are removed. Then, seeds which have more than 15% of their points dropped by a radial outlier filter are discarded. The remaining seed points are used as semantic 3D landmarks throughout reconstruction and counting, and the median point of each seed cloud is treated as the seed center.

4.2.3 Global Registration

We jointly register point clouds of a sorghum panicle imaged from different viewpoints via pose graph optimization [16]. One challenge is that the clouds are dense, and ICP on the full cloud performs poorly due to bad correspondences, an example of ICP falling into local minima. Instead, we choose a limited set of high-quality points

in the cloud and run ICP only on those points, somewhat analogous to performing optical flow on higher quality landmarks like SIFT features. The set of good seeds from image $\mathbf{I}_i$ are used as node $\mathbf{P}_i$ in the pose graph. Edges are created between neighboring frames by finding the local transform between the two point clouds of seed centers using ICP. Pose graph optimization is then performed to refine the final camera poses using the Levenberg-Marquardt algorithm [55]. An example of a reconstructed panicle is shown in Fig. 4.3(b).

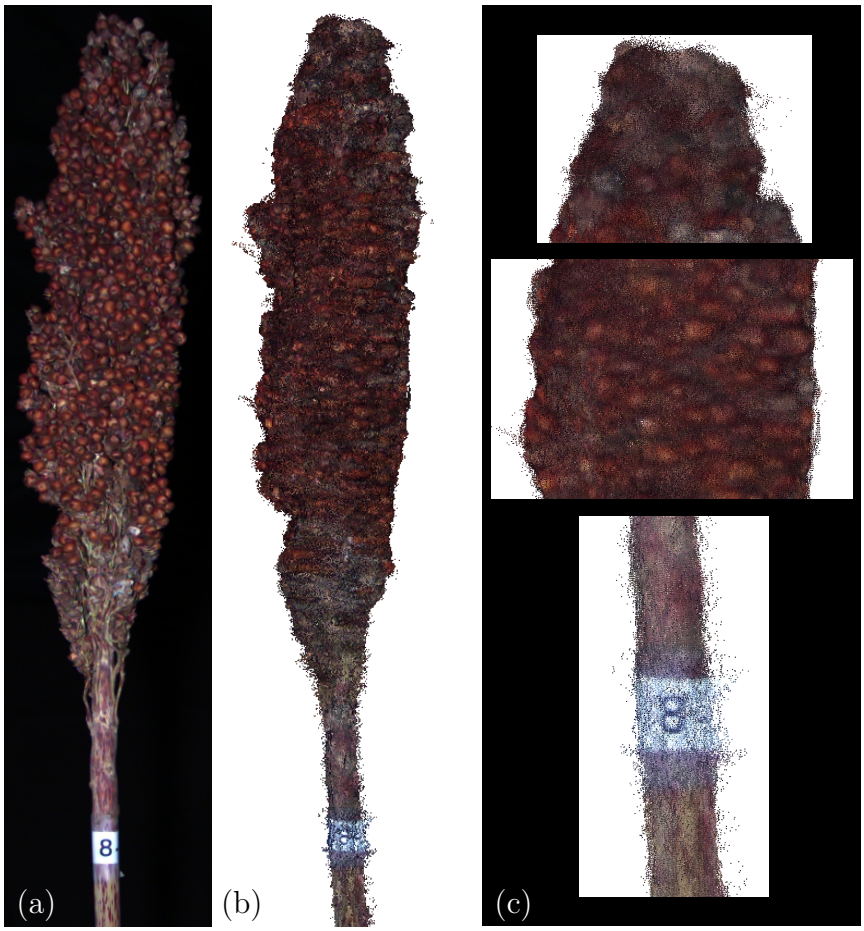


Figure 4.3: Example reconstruction results. (a) one of the original RGB images, (b) the colorized point cloud, (c) zoomed view of the colorized point cloud at the stem, mid-body, and tip. Some points of interest include the "8" on the stem label, and the body outline which matches the RGB outline well.

We observe that using camera poses from arm kinematics to initialize ICP yields poor results on the scale of seeds. This is due to error in extrinsic camera parameters,

despite using a standard hand-eye calibration process. Hence, we refine the camera pose priors by maximizing seed mask overlap. The seed masks of two neighboring nodes $\mathbf{P}_i$ and $\mathbf{P}_j$ are projected into a common image frame, at the average pose between $\mathbf{T}_i$ and $\mathbf{T}_j$. We search for the pixel shifts that yield the maximum IOU of seed masks as shown in Fig. 4.4. The *No Shift Maximize* ablation test in Fig. 4.14 shows that this IOU maximization improves reconstruction.

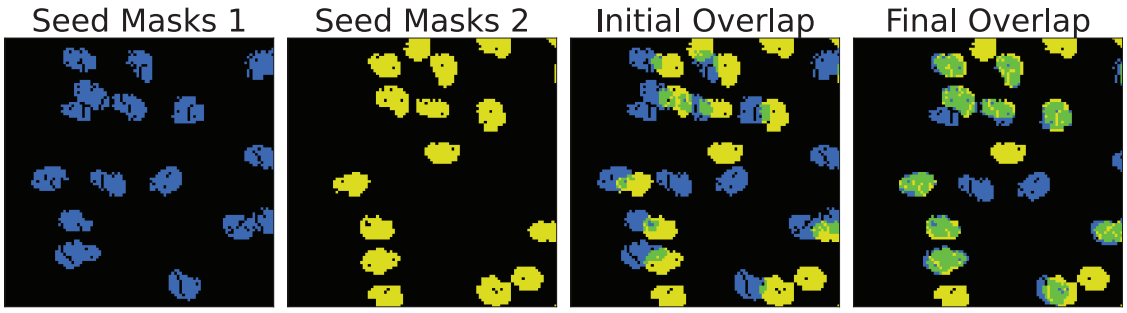


Figure 4.4: Matching mask structure with maximum IOU. Seed masks 1, seed masks 2, and their intersection are colored blue, yellow, and green in respective order.

4.2.4 Counting

In order to obtain a final seed count, we use the 3D model to ensure that a single true seed segmented in multiple images will be counted only once. The following 3D counting method performs this combination of 2D counts while handling the close proximity of neighboring seeds, spurious detections, and noise in the point cloud.

First, 3D seed centers are clustered using density-based spatial clustering (DBSCAN) [23], as shown in Fig. 4.5(d). To count the seeds in each cluster, we adopt the concept of 2D image smoothing and apply it to 3D point clouds. In image processing, a 2D Gaussian filter smooths an image by calculating a weighted average around each pixel's neighborhood. We take this idea and extend it to 3D. In our method, each seed center in the cluster is treated as a unit-impulse, and each impulse is smoothed around a volume of space using a 3D Gaussian sphere. Areas of space near multiple centers will have a higher density than those that are further away or near fewer centers. An example of this density map can be seen in Fig. 4.6(c).

Once the density values are calculated for the cloud points, the final step to

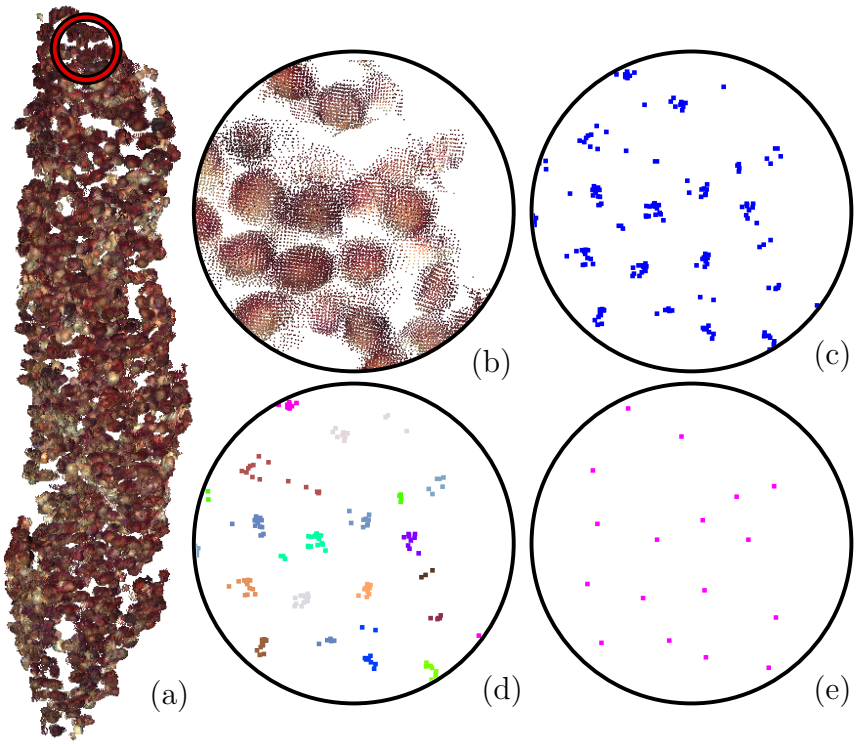


Figure 4.5: (a) An example of a final point cloud seed mask, (b) zoomed seeds, (c) seed centers, (d) seed centers clustered with DBSCAN, and (e) final seed sites.

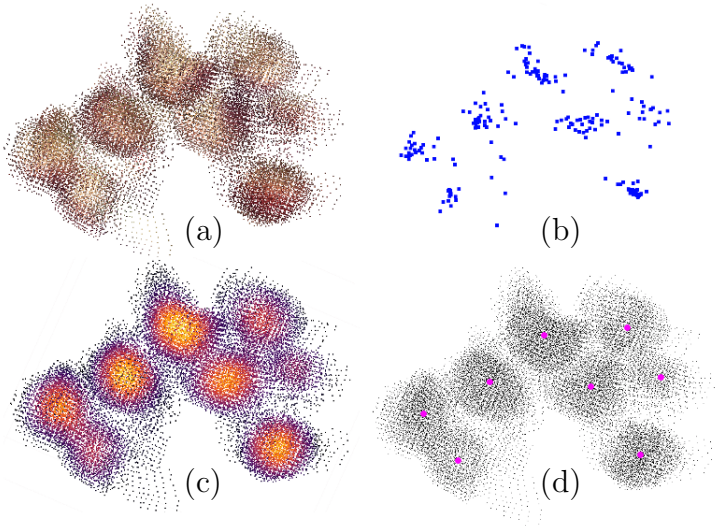


Figure 4.6: (a) Seed point cloud that has been put in a single cluster by DBSCAN, (b) seed centers from individual images, (c) seed points weighted by seed-center density, and (d) local maxima (pink) that have been chosen as seeds.

calculate the number of seeds in each cluster is to find the local maxima within a defined radius. This is a type of non-maximal suppression (NMS) on the density values. Each local maximum corresponds to a unique seed and is treated as the location of the seed's center, as shown in Fig. 4.6(d). Once all local maxima are found for each cluster, the total number of maxima becomes the final seed count. Fig. 4.7 gives an example of this seed-detection process on an entire panicle.

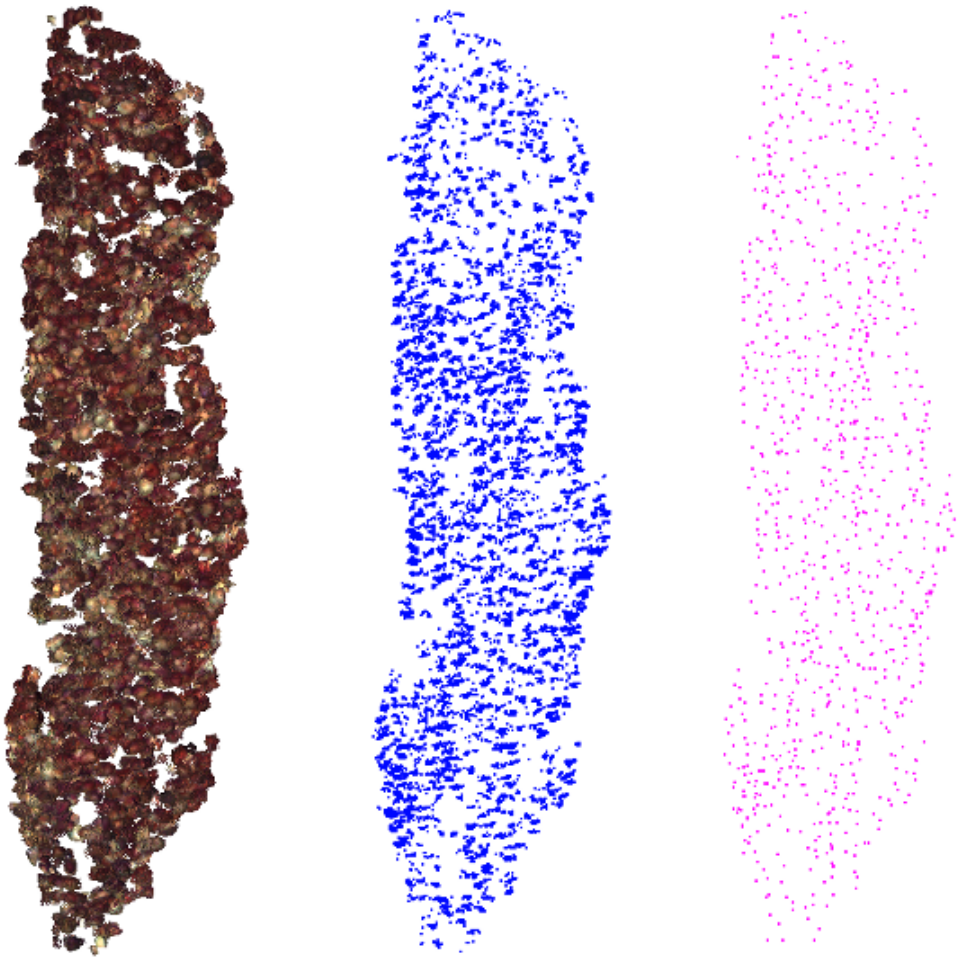


Figure 4.7: From the cloud of masked seed instances (left), we find detected seed centers from all views (middle). After identifying maxima in the density cloud, the final filtered seed positions are given (right).

4.3 Unsupervised 3D Reconstruction Metric

Several prior works [113, 115] discuss quantitative reconstruction evaluation in the absence of ground truth, but they require that the final output to evaluate against is a mesh. Our reconstruction method produces a dense point cloud, so we developed and validated a novel cloud-only rendering-based method for assessing reconstruction quality in the absence of ground truth. We compare a small circle of pixels sampled from an RGB image $\mathbf{I}_i \in \mathbb{I}$, centered on a sampled seed against a projected render of the same seed made using the full reconstructed point cloud. A sampling function λ is defined so that K seeds are sampled per image along the center of the vertical axis where the projections are cleanest. This method experimentally indicates relative levels of noise in the reconstructed point clouds by comparing rendered sections to the original RGB images. Fig. 4.8 visualizes the point sampling process, and example renders are displayed in Fig. 4.9.

To validate this framework, noise was purposefully introduced in the refined camera poses after global registration when creating the reconstructed cloud. This noise in the final transforms effectively simulates poor reconstruction quality. A variety of comparisons were then run on pairs of RGB image patches and the corresponding rendered patches. We evaluated all combinations of RGB/grayscale and normalized/un-normalized patches with respect to their intensities, gradients, and Laplacians. Example outputs can be seen in Fig. 4.9.

The two metrics that responded the best to the introduced noise were the mean-squared error (MSE) on normalized grayscale gradients and the Structural Similarity [108] (SSIM) on normalized grayscale Laplacians, as shown in Fig. 4.10. Two examples of our image-to-render comparison with their corresponding MSE and SSIM scores are shown in Fig. 4.11.

Our reconstruction quality metrics "$\alpha\beta$-MSE" and "$\alpha\beta$-SSIM" are defined as follows. For each image, λ samples K seeds from $\mathbb{S}_i$, where $\mathbb{S}_i$ are the seeds in image $\mathbf{I}_i$. For a sampled seed $s_{ik} \in \mathbb{S}_i$, the image patch α_{ik} and rendered patch of the point cloud β_{ik} are generated, both of which are grayscaled and normalized. The MSE and SSIM of α_{ik} and β_{ik} are calculated and then averaged over all seeds and panicles.

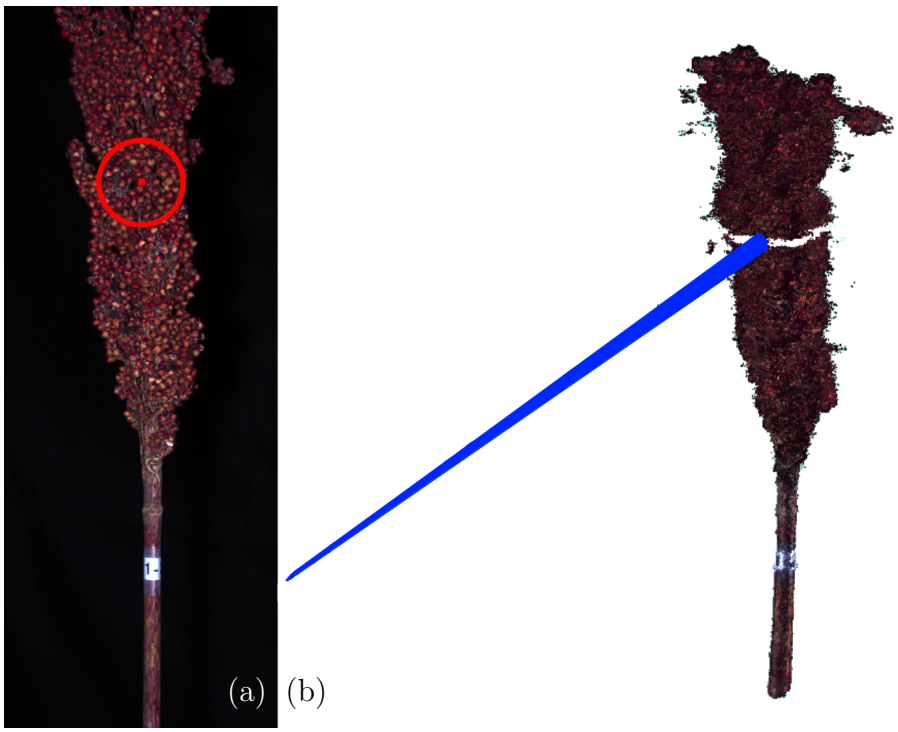


Figure 4.8: (a) RGB image of a sorghum panicle, where a single seed (highlighted in red) has been selected by the sampling function λ. (b) Visualization of the render projection, where a cone (blue) reaching out from the render origin selects only the points around the chosen seed.

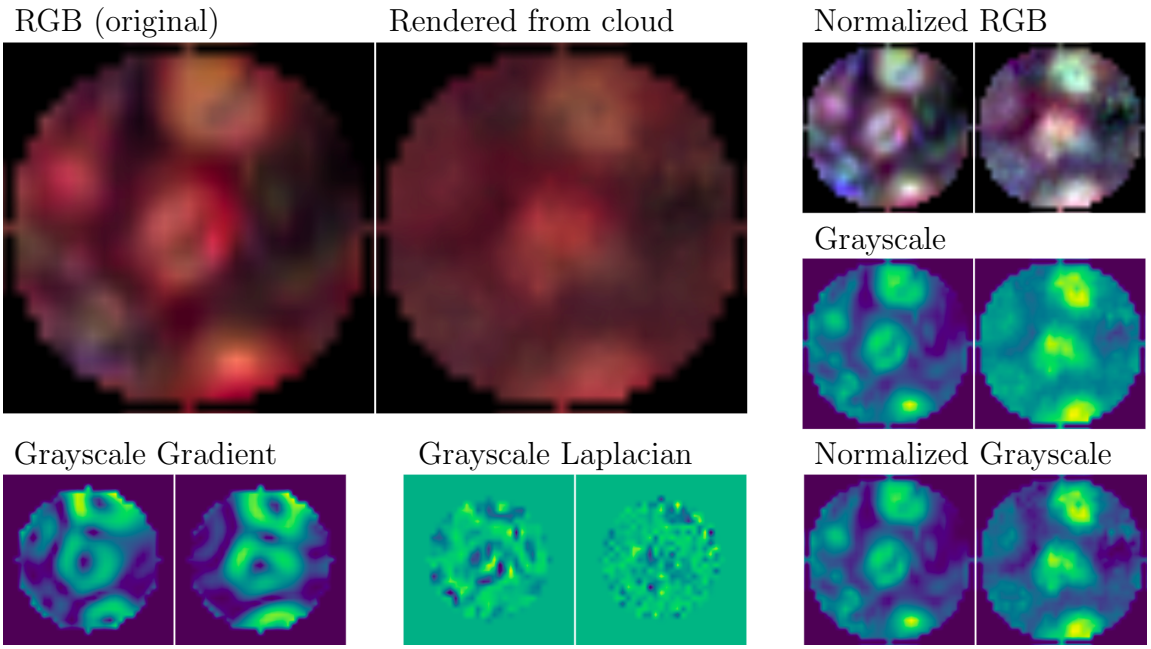


Figure 4.9: Examples of the image operations that were explored when finding patch comparisons most sensitive to reconstruction noise.

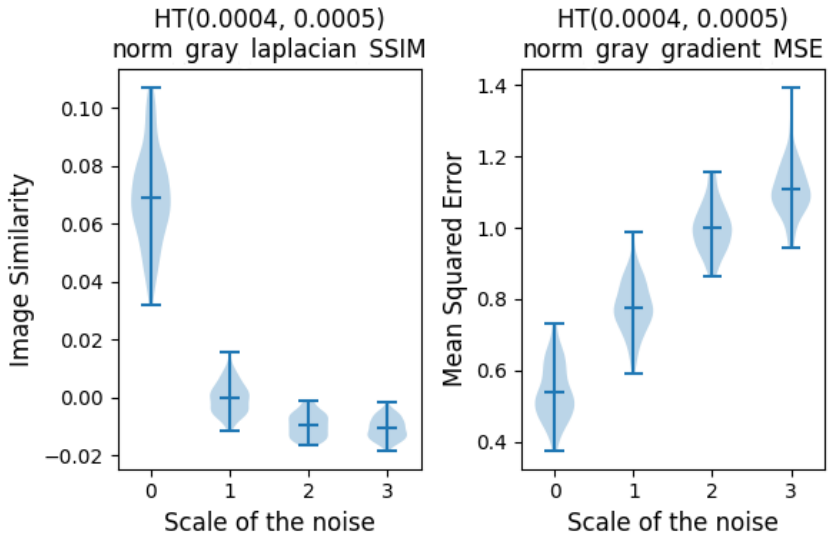


Figure 4.10: Response of chosen metrics to introduced noise. Noise took the form of homogeneous transforms, with translational noise drawn from a Gaussian $\mathcal{N}(0, \sigma = \text{scale} * 0.4\text{mm})$ and rotational angle noise drawn from a Gaussian $\mathcal{N}(0, \sigma = \text{scale} * 0.5\text{mrad})$. After the random transforms the cloud was recalculated and rendered.

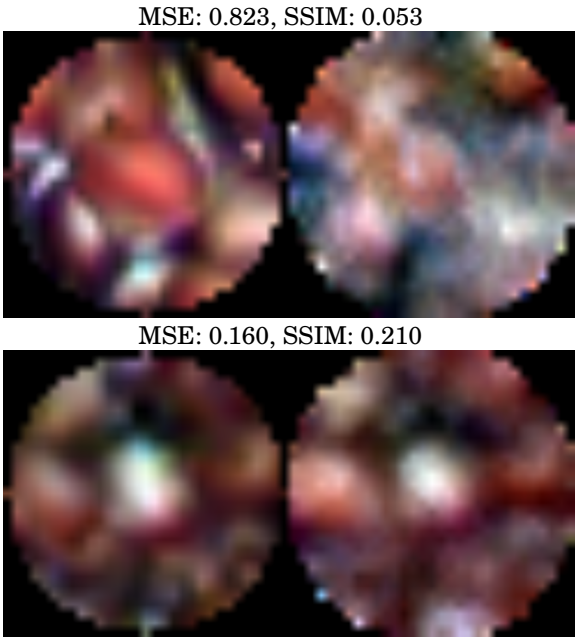


Figure 4.11: Qualitative examples of the reconstruction metrics. On the left are image patches, on the right are patches rendered from the reconstructed point cloud. Patches are normalized so each channel has min/max values of 0/255.

$$\text{MSE}_{ik} = \frac{1}{N} \sum_{pixels} \left[\nabla \alpha_{ik} - \nabla \beta_{ik}\right]^2 \tag{4.1}$$

$$\alpha\beta\text{-MSE} = \frac{1}{P} \sum_{p} \frac{1}{IK} \sum_{i} \sum_{k \in \lambda(\mathbb{S}_i)} \text{MSE}_{ik} \tag{4.2}$$

$$\text{SSIM}_{ik} = \text{SSIM}\big(\mathcal{L}\big(\alpha_{ik}\big), \mathcal{L}\big(\beta_{ik}\big)\big) \tag{4.3}$$

$$\alpha\beta\text{-SSIM} = \frac{1}{P} \sum_{p} \frac{1}{IK} \sum_{i} \sum_{k \in \lambda(\mathbb{S}_i)} \text{SSIM}_{ik} \tag{4.4}$$

Here ∇ is the image gradient, $\mathcal{L}$ is the image Laplacian, $\frac{1}{IK} \sum_i \sum_{k \in \lambda(\mathbb{S}_i)}$ indicates an average over sampled seeds in all images, and $\frac{1}{P} \sum_p$ indicates an average over all panicles.

4.4 Experiments and Results

4.4.1 Dataset

Our dataset consists of stereo images of 100 sorghum panicles. There were 10 panicles from 10 different species as seen in Fig. 4.12(a). To evaluate our proposed method, we manually stripped panicles (Fig. 4.12(c)) and counted all seeds using an automatic seed counting machine[1] (Fig. 4.12(d)), which serves as ground truth. The process of stripping seeds, removing husks, and counting took significant effort, on average 40 minutes per panicle, which reinforces the usefulness of an automated method for yield estimation.

Random errors in the seed count include some lost seeds that fell off panicles between image collection and hand-counting. Affecting the count in the opposite direction, some unremoved husks were counted as seeds by the counting machine, despite manual efforts to separate seeds from husks. We expect the effect on the ground truth to be small. The stereo images, camera poses, human-labeled seed segmentations, panicle weights, and ground truth seed counts can be found in our dataset[2]. Fig. 4.13 provides an example of the images, depth data, and segmentation results from our dataset.

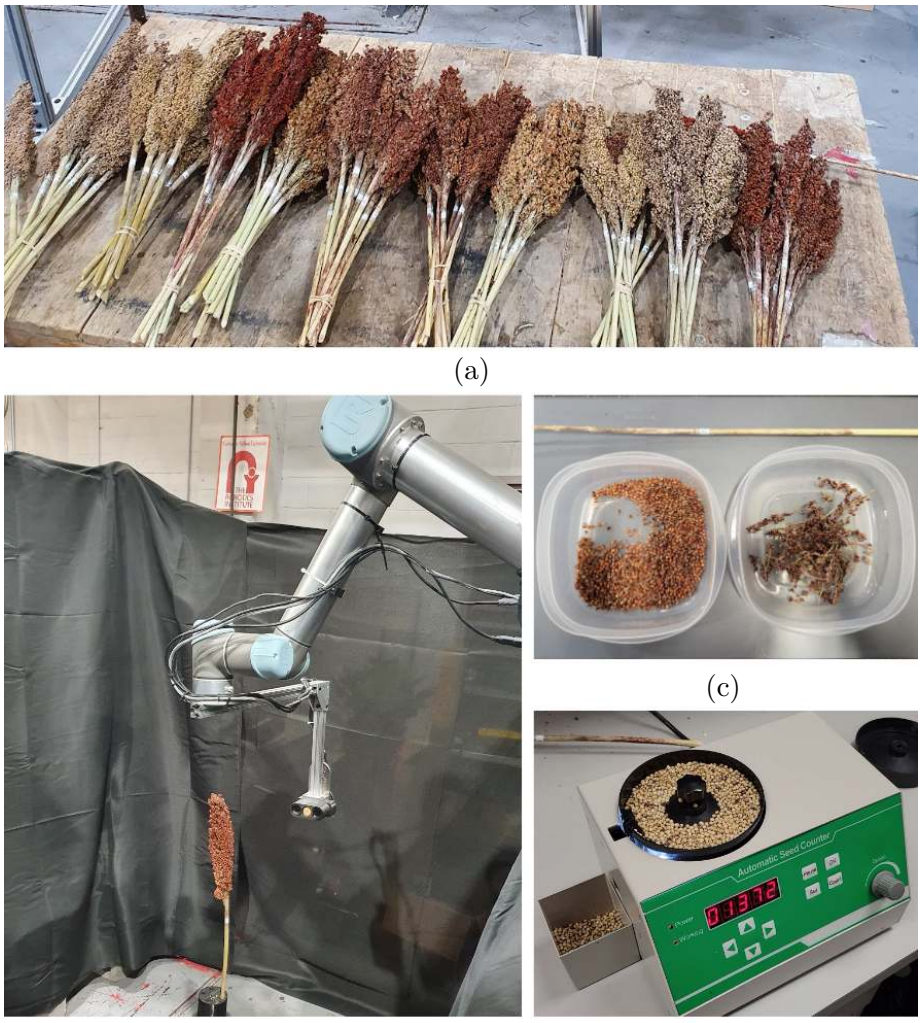
4.4.2 Reconstruction Results

We assess the effectiveness of our approach with ablation tests using the reconstruction metrics described in Section 4.3. Below references to "$\alpha\beta$-MSE" and "$\alpha\beta$-SSIM" are referring to these specific operations on image and rendered patches. Note that growing $\alpha\beta$-MSE (error) and dropping $\alpha\beta$-SSIM (similarity) both indicate a worse match. Fig. 4.14 shows results of ablation tests on reconstruction quality.

1. *Our Method*: Our final method. All experiments below are modifications to this approach. This had the best average $\alpha\beta$-MSE and $\alpha\beta$-SSIM scores.
2. *No Shift Maximize*: The mask overlap maximization discussed in Section 4.2.3

[1]Wadoy Automatic Seeds Counter, Sly-C

[2]High-Resolution Stereo Scans and Segmentation Data of 100 Sorghum Panicles at https://labs.ri.cmu.edu/aiira/resources/



(a) (b) (c) (d)

Figure 4.12: (a) 100 sorghum panicles from 10 different sorghum species. (b) Our data collection system, a stereo camera attached to the UR5 robot arm. (c) Seeds were manually stripped and (d) counted using a seed counting machine.

is not used. This resulted in a slight decrease in reconstruction quality.

3. *No Final Optimize*: The pair-wise ICP transformations discussed in Section 4.2.3 are still used to adjust cameras relative to the first frame, but the final optimization is not applied.
4. *Full-Cloud ICP*: Instead of running pair-wise ICP on masked seed points, ICP was run on the full point clouds. This test showed a significant drop in

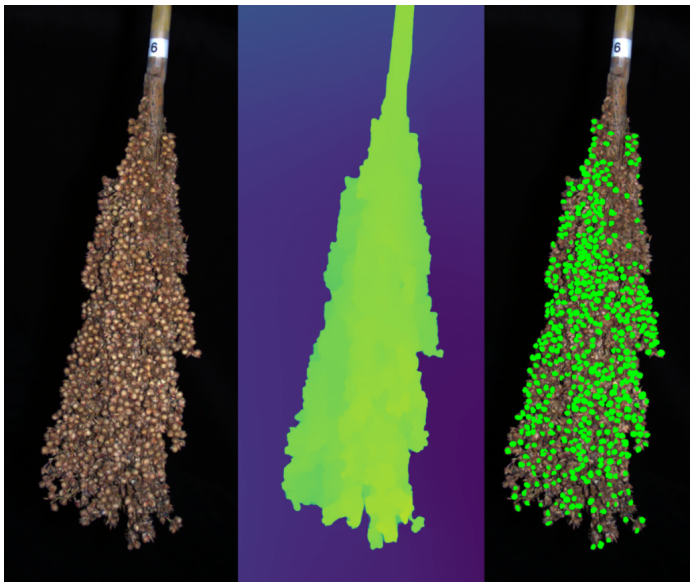


Figure 4.13: Visualized example of the images, depth data, and hand-segmentations in our sorghum dataset.

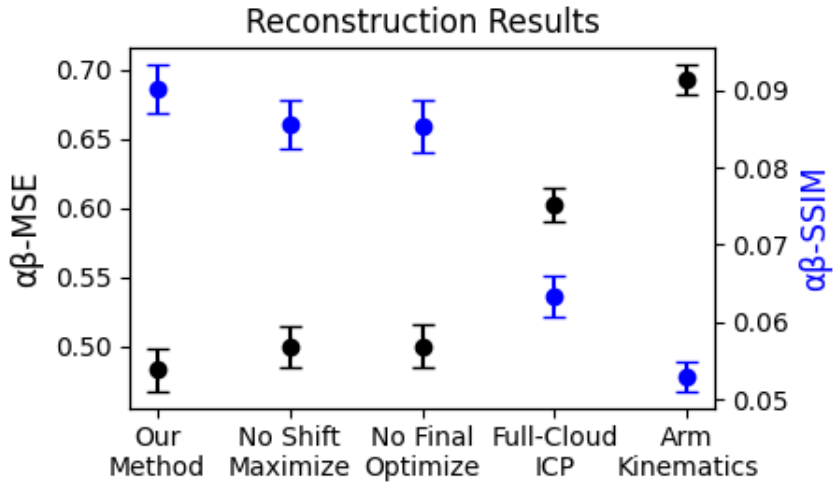


Figure 4.14: Noise metric results showing growing error and dropping similarity for reconstruction experiments. The vertical bars are the 95% confidence intervals for the mean of the per-panicle scores.

reconstruction quality.

5. *Arm Kinematics*: Views were combined using the arm kinematics, with no pose optimization. Although kinematically reconstructed panicles could be used for

applications like collision avoidance, they had the worst reconstruction scores and could not be used for counting. Single seeds were clearly represented in multiple 3D locations, "smeared" cylindrically around the panicle.

The best reconstruction results came from pose graph optimization using ICP on points determined to be high-quality seeds. This did better than ICP naively done using the full cloud from each image. Our hypothesis on why full-cloud ICP is worse is that sorghum is very organic and complex, and picking out meaningful, high-quality areas for ICP to operate on reduces the likelihood of ICP falling into local minima. The required quality of reconstruction depends on the application. When using 3D structure to identify overlaps in 2D segmentation, decreasing reconstruction quality will lead to counting errors as identifications of the same seed drift apart in space.

4.4.3 Counting Results

As shown in Fig. 4.15, the count produced by our method has a strong linear fit to the ground truth count, with an R^2 of 0.875. The 10-fold RMSE using a 75/25 train/test split calculates an average prediction error of 295 seeds. There will always be some error in non-destructive counts, since sorghum panicles have internal, hidden seeds that cannot be seen from an outside view. The only way to expose all seeds is to strip them off the panicles, a destructive and time-consuming process.

Another characteristic worth measuring for sorghum is its yield weight, which represents a sellable quantity of the crop. The fit between count and seed weight is still reasonably representative, with an R^2 linear fit of 0.819 in Fig. 4.16, but it fits slightly less well than seed count, likely due to variations in seed density across panicles. The 10-fold RMSE using a 75/25 train/test split calculates an average prediction error of 8.5 grams per panicle.

4.4.4 Benefits of 3D Data over 2D

In [48], it was shown that it is sufficient to take a 2D count of one side of an ear of corn and scale that to a full kernel count. To test this, ears were rotated around their long axis by 90° increments and imaged, and it was found that single-image kernel counts had low variation because kernels were generally evenly distributed. In

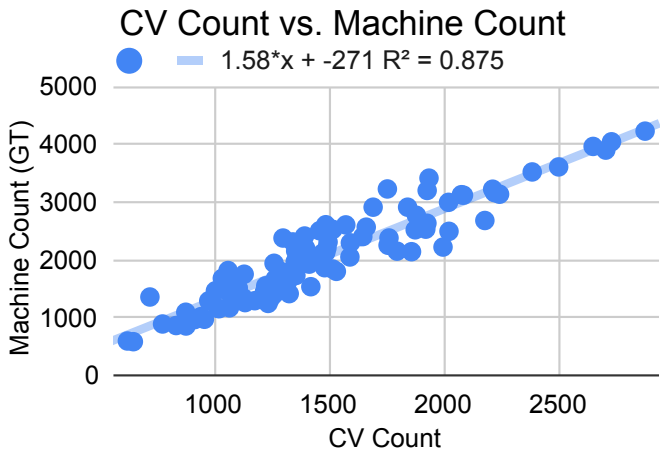


Figure 4.15: Fit between our method's count (Computer Vision/CV Count) and the ground truth count as described in Section 4.4.1.

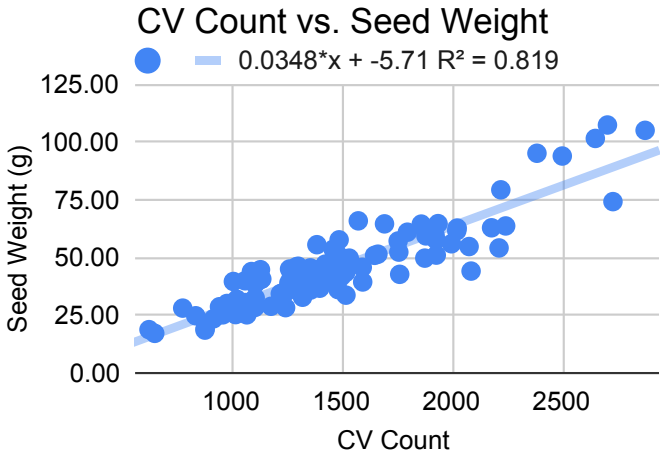


Figure 4.16: Fit between counted seeds and seed weight, which is the weight of seeds after they have been stripped off a panicle and cleaned of husks.

contrast, sorghum is more complex in shape, and therefore has more variation when a full count is extrapolated from a single image. In Fig. 4.17 and Fig. 4.18 we compare the predictiveness of 2D and 3D counts.

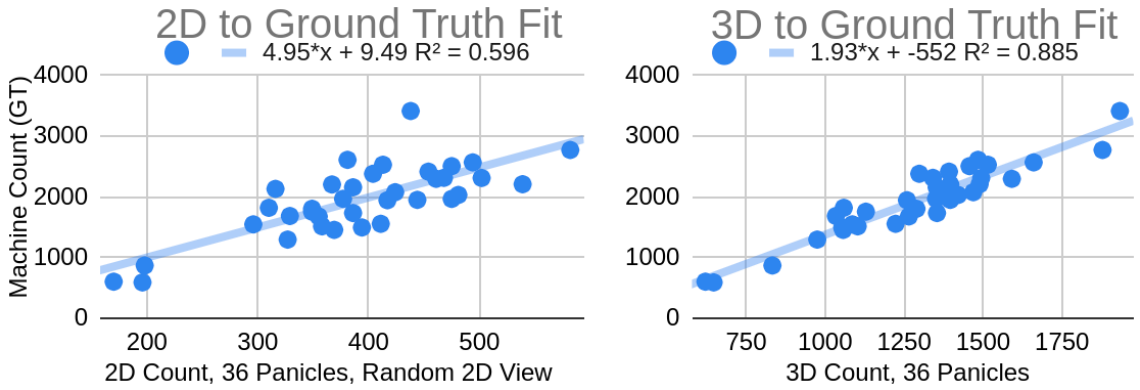


Figure 4.17: Comparison of 2D and 3D counts fit to ground truth. 2D count comes from a single image per available panicle and has a lower R^2 score, indicating worse predictive performance for linear regression. The 10-fold RMSE for these 2D and 3D counts are 353 and 204 respectively.

It may seem unfair to compare 2D and 3D extrapolation because 3D methods have more data available (dozens of images vs. a single image), but it is important to evaluate for hardware considerations. Getting images surrounding a plant for 3D reconstruction is more costly in terms of system complexity, requiring the camera to be actuated rather than fixed to a mobile base such as a tractor, so it is important to assess what relative benefit the 3D method brings.

In order to test the extrapolation principle, we obtained 2D segment counts from images spaced 90° apart. This was complicated by the fact that some panicles were too tall to be captured in a single frame. To avoid trying to combine segmentation counts from multiple images, we only use counts where the full panicle is visible in four 2D views. 36 out of the 100 panicles met this criteria, enough to get a reasonable representation.

As seen in Fig. 4.17, 3D counts have a significantly better linear fit to the ground truth counts, with an R^2 of 0.885 compared to 0.596 for 2D counts (sampled randomly from the 90° separated views), demonstrating that 3D count is a better predictor of the desired feature. The variation in 2D count within each panicle can be seen in Fig. 4.18. There are significant variations in extrapolated counts within each panicle,

often stretching to 20-40% of the ground truth value.

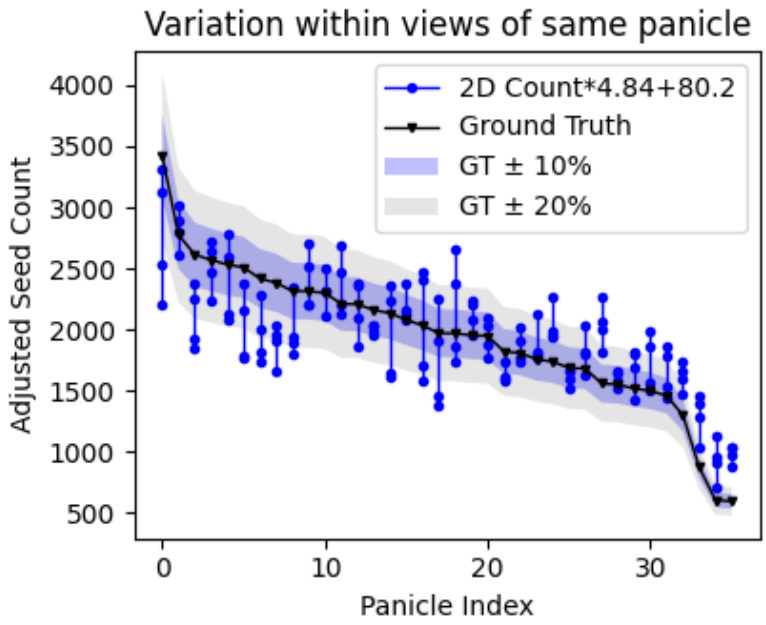


Figure 4.18: Variation across viewpoints among the 36 panicles, using a linear fit to extrapolate from 2D count to an estimated full count. Linear fit parameters have been recalculated to use all four 90° separated images per panicle instead of a random one as in Fig. 4.17. R^2 on the increased views was 0.634.

4.5 Discussion

One of the benefits of this approach is the integration of segmentation counts across multiple 2D views, using the 3D model to determine which detections are unique. Future detection and segmentation improvements could be folded in to improve estimates while still getting the benefit of view combination. However, the use of multiple views is an intensive process, and uses many images of each panicle. It would be worthwhile to find the minimal image set that could reliably create a high-quality model, reducing runtime and resource requirements. Dense panicle models could also be put to other uses - in addition to extracting counts and volumetric information, other phenotyping or health characteristics could be evaluated, perhaps based on

color, or texture. The model could also be used to plan physical interactions between robots and the modelled crop.

Although these images were captured in a lab setting, in-field image capture from an arm mounted on a mobile base would also be possible. Because sorghum panicles often grow close together, future work would have to be dedicated towards stalk isolation through manipulation or model reconstruction without a full 360° scan. As well, we took images with a black background to simplify color-based foreground segmentation. To work in the field, the system would have to be able to reason about which instances belong to the panicle of interest.

Chapter 5

Apple Fruitlet Sizing and Growth Rate Tracking

5.1 Motivation

Measuring growth rates of apple fruitlets is important because it enables farmers to better control their annual yield. It is standard practice to thin apple trees to prevent them from developing a pattern of alternative year bearing in order to produce a more consistent yearly harvest. To predict the effect thinning application has when applied to trees, the Fruitlet Growth Model developed by Greene *et al.* [30] is used to determine how often farmers need to spray. The model takes into account the growth rates of a subset of fruitlets over multiple days. The fruitlets with growth rates greater than 50% of the fastest growing fruits are predicted to persist the thinning. Farmers calculate the percentage expected to abscise and use this information to determine when to apply thinning application. Therefore, increasing the sampling size and producing more consistent measurements will lead to more accurate and reliable yield predictions.

The sizing method most commonly used in practice involves identifying each individual fruitlet, using a digital caliper to hand-measure sizes, and manually entering the data into a spreadsheet so that growth rates can be tracked. We will refer to this sizing process as the caliper method. The number of fruitlets typically

sized using the caliper method lies in the range of high hundreds to low thousands. Sizing is performed twice on each fruitlet per thinning application: once three to four days after application, and again seven to eight days after application. Taking this many hand-measurements is not only labor-intensive, but highly subject to human error. It is inefficient and time-consuming for a human to record caliper readings for hundreds to thousands of fruitlets across multiple days, making some farmers hesitant to adopt the approach. Manually associating fruitlets across different days is also very challenging; fruitlets are likely to have moved or fallen off, resulting in them being mis-identified which negatively affects growth estimates. Moreover, using calipers creates variability when measuring asymmetrically shaped fruit, leading to inaccurate measurements which become more pronounced as different workers are employed to collect data. As a result, there is a need to automate this process to make sizing faster, more repeatable, and more accurate.

5.2 Sizing and Growth Rate Tracking

5.2.1 Tagging Methodology

The Fruitlet Growth Model requires growers to determine the number of trees and the number of clusters per tree they want to size. A cluster is a group of fruitlets that grow out of the same bud (Fig. 5.1), with typically two to six fruitlets per cluster. The diameters of each fruitlet in every selected cluster are measured two to three times per thinning application.

We hang AprilTags [77] next to each cluster to identify the selected fruitlets. AprilTags were selected because they allow for fast cluster identification in computer vision systems. They are easy to detect, and can be used as additional semantic information for fruitlet association as demonstrated in Section 5.2.4.2. Each fruitlet in the cluster is assigned a unique id for identification and tracking its size (Fig. 5.1).

5.2.2 Camera Setup

To facilitate data collection, we designed a custom setup that consists of the stereo camera from Section 3.1 connected via USB to a laptop to save the captured images.

Figure 5.1: An example of a fruitlet cluster. An AprilTag is hung next to the cluster, and each fruitlet receives a unique id which is written on the back for identification.

A phone is mounted on the back and connected via USB-C to the same laptop to allow the user to visualize and assess the quality of the images in real-time, as shown in Fig. 5.2.

5.2.3 Fruitlet Sizing

Our apple fruitlet sizing pipeline is based on the one presented by Qadri [85]. Fruitlets are detected, segmented, and sized by fitting an ellipse. The diameter of each fruitlet is calculated using the baseline of the stereo camera, the minor axis of the fit ellipse, and the extracted disparities. An overview of our system is shown in Fig. 5.3.

The main difference in our approach and the one in [85] is we replace the MADNet [98] network with RAFT-Stereo [59]. We also replace the Faster R-CNN [86] network with Mask R-CNN [36] to predict bounding boxes around all fruitlets in the image. This is because Mask R-CNN can also be used for tag segmentation, as discussed in Section 5.2.4.2, without requiring an additional network pass. Only the bounding box classification head is used for fruitlet detection, and the mask segmentation head is ignored. We use a customized detectron2 [110] Mask R-CNN implementation.

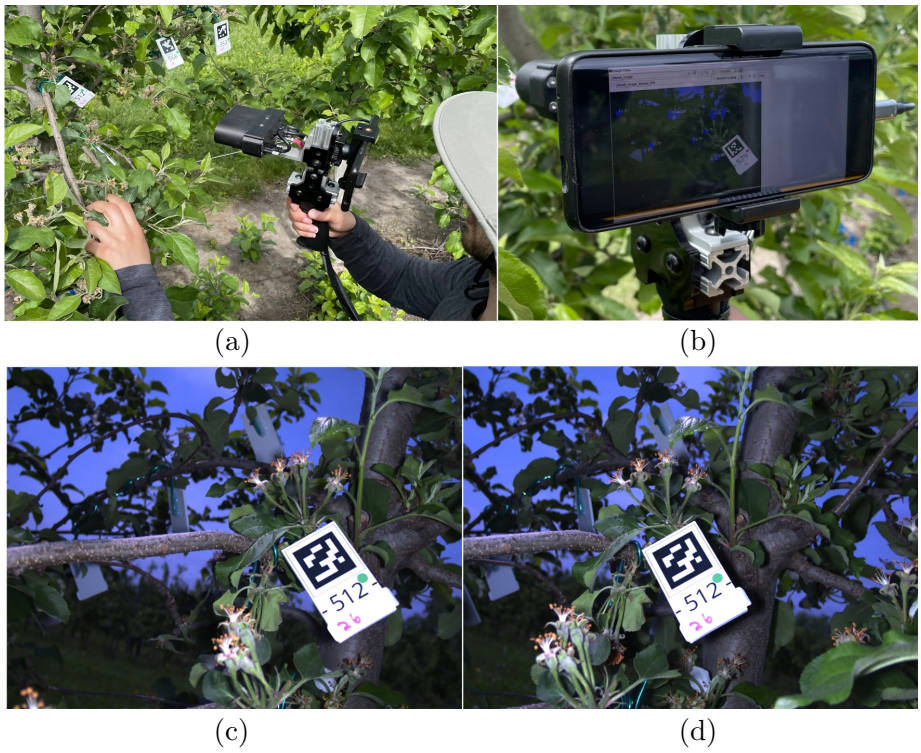


Figure 5.2: (a) Hand-held flash stereo camera. (b) Phone is mounted and connected via USB-C to display images to the user in real-time. Captured left (c) and right (d) stereo images are shown.

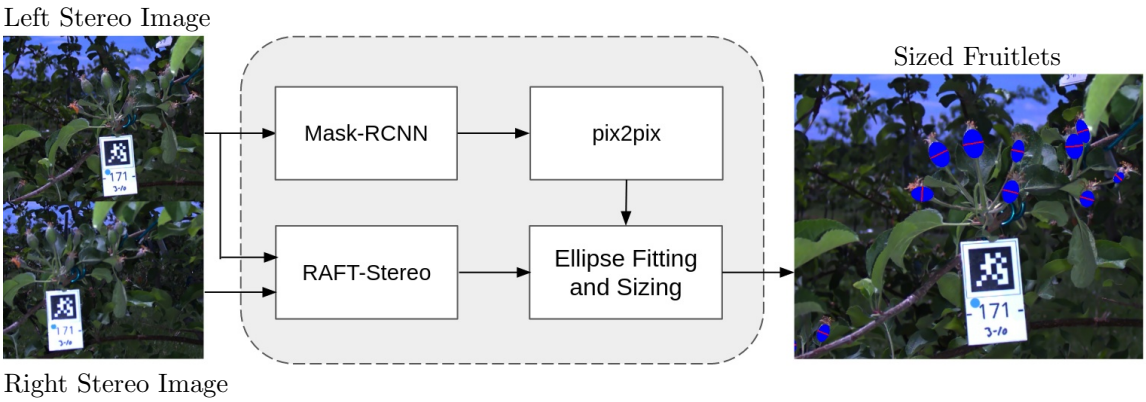


Figure 5.3: Fruitlet sizing pipeline. Fruitlets are detected using Mask R-CNN bounding box classification head and segmented using pix2pix. Ellipses are fit to the segmented fruitlets and sized using disparity values extracted by RAFT-Stereo.

To segment, we train a pix2pix [45] Conditional Generative Adversarial Network (CGAN). Each individual bounding box is cropped and passed to the pix2pix generator, which outputs a single segmentation mask for the fruitlet (Fig. 5.4).

The reason the proposed detection and segmentation networks are used, instead of a single instance segmentation Mask R-CNN network, is because of the challenges of obtaining labelled ground truth data, which is a common issue in agriculture [2]. Hand-segmenting fruitlets for the required number of images to train is a time-consuming task, and is difficult to outsource as a result of the required domain-specific knowledge. It is more data-efficient to train a network to semantically label individual fruitlets [85]. A CGAN architecture was chosen as the segmentation network because CGANs have demonstrated previous success in operating in agricultural environments [76, 80, 85]. As well, pix2pix qualitatively performed better than Mask R-CNN for fruitlet segmentation when trained on low amounts of labelled data.

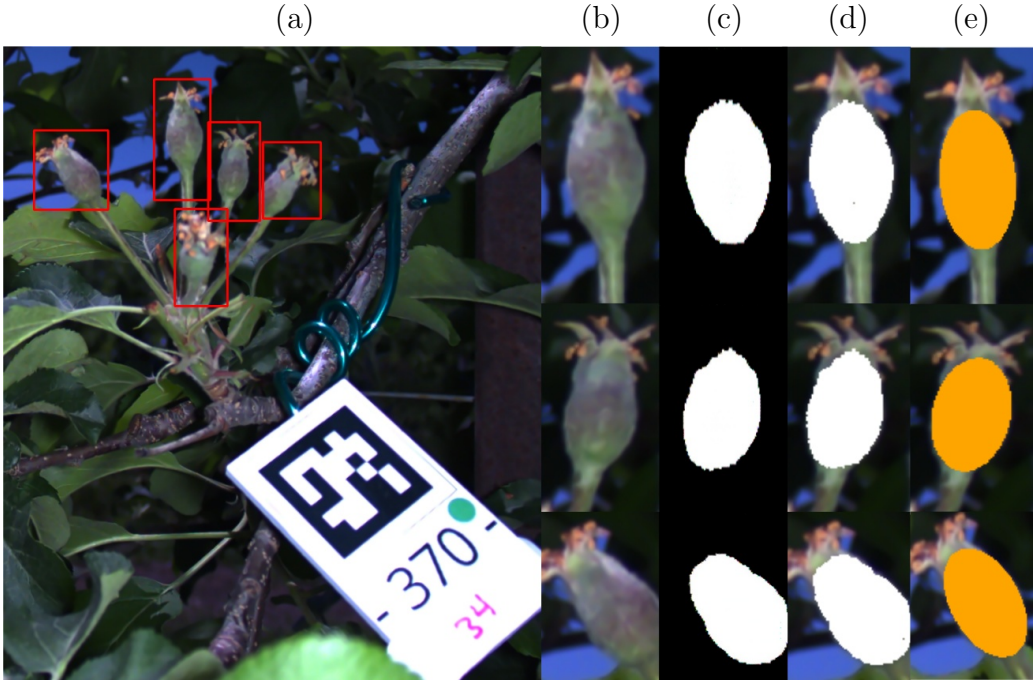


Figure 5.4: Segmentation of apple fruitlets. Detected fruitlets (a) are cropped (b) and passed to the pix2pix generator which outputs a segmentation mask (c) and (d) to be used for ellipse fitting (e).

An ellipse is fit to the segmented image following the process demonstrated in Fig. 5.5. First, a binary threshold is applied to the pix2pix output. Next, the contour

surrounding the segmented points is extracted. Lastly, an ellipse is fit using the OpenCV [5] fitEllipse function, which uses a constrained least squares formulation. The result is each ellipses' canonical parameters, including the length of the minor axis ma. The size is calculated as

$$\text{size} = \frac{ma \times b}{d} \tag{5.1}$$

where b is the baseline of the stereo camera and d is the max disparity value found in a square region around the center of the segmented fruitlet. The derivation of Equation 5.1 can be found in [85].

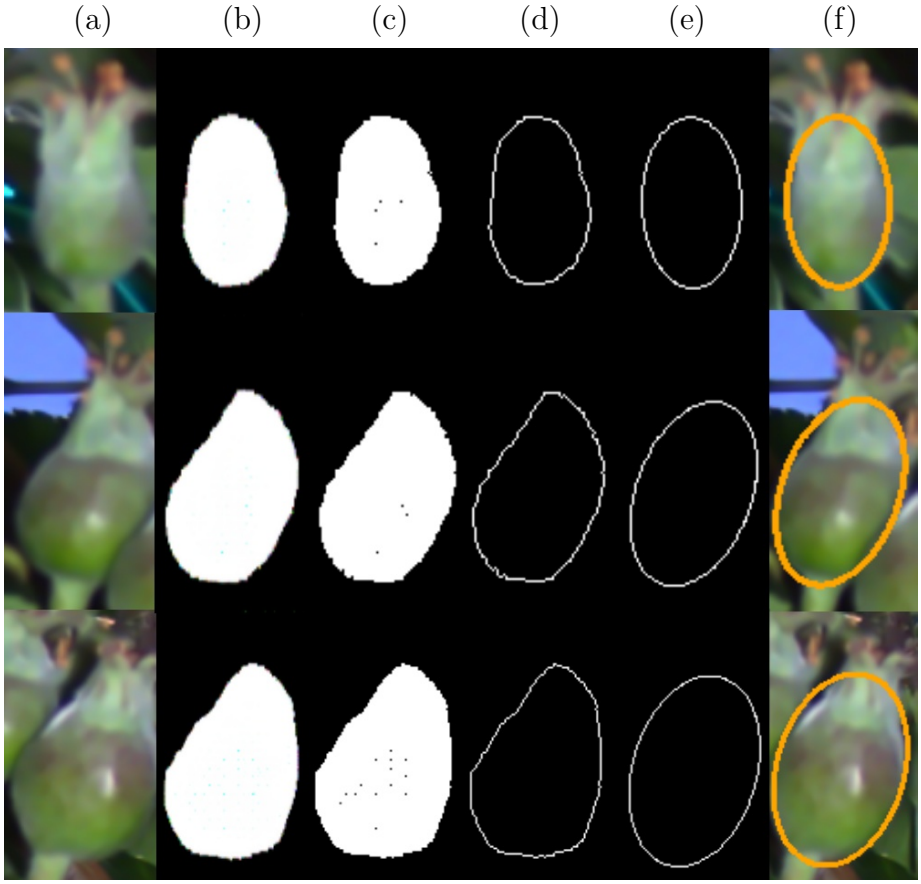


Figure 5.5: Fruitlet ellipse fitting. The pix2pix output (b) is thresholded (c) and a contour is fit around the segmented image (d). An ellipse is fit using the OpenCV fitEllipse function to produce (e) and (f).

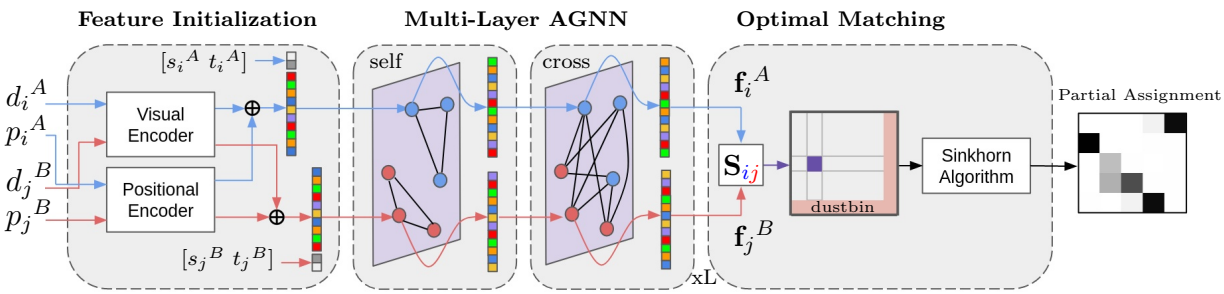


Figure 5.6: Temporal fruit association network architecture. Local features are mapped to deep vectors using visual and positional descriptor encoders, and the result is concatenated with node classification scores and tag information to build the initial node feature vectors. The feature vectors are updated through a series of L alternating self and cross attention layers, and the result is passed through an optimal matching layer to find the optimal partial assignment.

5.2.4 Temporal Fruit Association

While the system presented in Section 5.2.3 helps automate the fruitlet sizing process by removing the need for hand caliper measurements, very little research has been done to compare sizes across different days. As a result, human effort is still required to determine which fruitlets in one image match to fruitlets in the other so that growth rates can be properly tracked. This task is time-consuming and requires great attention to detail. If we wish to move towards full automation of the fruitlet sizing process, alternative solutions must be found.

It is challenging to accurately associate fruitlets in images taken on different days. This is because the fruitlets may have moved, fallen off, or changed appearance, and the images are unlikely to have been taken from the same camera pose. To address these issues, we introduce a Graph Neural Network (GNN) [29] based system for temporal fruit association. Given two images of the same cluster taken on different days, the task is to identify which fruitlets in one image match to those in the other. We use a GNN over traditional Convolutional Neural Network (CNN) based architectures because CNNs assume the input to be a grid-like structure, where convolutions are performed on neighboring pixels. This assumption does not accurately represent fruitlet association as there is no guarantee of spatial locality between fruitlets, especially across images. GNNs have shown impressive results when operating on unstructured data, where node feature embeddings are propagated along

edges in a process known as feature aggregation. They have successfully been applied to multi-object tracking tasks [7, 105] and feature matching from different camera poses [91, 92]. There have been several GNN architectures introduced [32, 49, 68] each with their own unique feature aggregation methods designed for their desired tasks. Particularly, Graph Attention Networks (GATs) [103] have gained a lot of traction as they are more generalizable and are able to employ attention mechanisms in order to assign importance to neighboring nodes without requiring knowledge about the graph structure upfront.

To perform cluster association, we utilize an Attentional Graph Neural Network, a type of GAT introduced by Sarlin *et al.* [91]. The main difference in our problem formulation and the one in [91] is that instead of matching keypoints consisting of single pixels, we are matching fruitlet detections which span multiple pixels. The novelty in our approach lies in i) our local feature extractor (Section 5.2.4.3), which extracts visual and positional descriptors from fruitlet and tag detections; ii) our node feature vector initialization (Section 5.2.4.4), which encodes the local features using CNNs and directly injects classification score and tag information onto the resulting feature vector; and iii) our mechanism to apply loss to exclusively clustered fruitlets (Sections 5.2.4.2 and 5.2.4.8) due to the difficulties in labelling ground truth data. The temporal fruit association network can be seen in Fig. 5.6.

5.2.4.1 Formulation

Consider images A and B taken on different days with detected fruitlets $\mathbf{F}^A$ and $\mathbf{F}^B$ belonging to the same cluster C. $\mathbf{F}^A$ and $\mathbf{F}^B$ have M and N fruitlets respectively, index by $\mathcal{A} := \{1, ..., M\}$ and $\mathcal{B} := \{1, ..., N\}$. The objective is to match fruitlets in $\mathbf{F}^A$ with fruitlets in $\mathbf{F}^B$. Similar to [91], each fruitlet must adhere to a set of constraints: i) must have at most a single correspondence in the other image; ii) may be unmatched as a result of fruitlets falling off, occlusions, missed detections, or incorrectly detected fruitlets.

5.2.4.2 Detection, Segmentation, and Feature Map Extraction

An image $\mathrm{I} \in \{A, B\}$ of target cluster C is passed through a Mask R-CNN network to detect fruitlet bounding boxes $\mathbf{B}^{\mathrm{I}}$ and detect and segment tags $\mathbf{T}^{\mathrm{I}}$. As in Section

5.2.3, only the bounding box classification head is used to detect fruitlets, whereas the mask segmentation head is also used to segment the AprilTag. An additional class output is added to identify fruitlets that belong to the imaged cluster. In other words, the network identifies the clustered fruitlets $\mathbf{F}^{\mathbf{I}}$ by determining which $b_i \in \mathbf{B}^{\mathbf{I}}$ belong to C. This is necessary for three reasons: first, the Fruitlet Growth Model requires only the fruitlets belonging to the tagged cluster to be measured; second, our ground truth data contains only measured growth rates for clustered fruitlets; and third, for network training it is infeasible to accurately label fruitlet matches for every fruitlet in hundreds of images, and is it much more labor and time-efficient to focus on only clustered fruitlets. An example of the AprilTag identification and Mask R-CNN output can be seen in Fig. 5.7.

Fruitlets are segmented using pix2pix, and tag $\tau \in \mathbf{T}^{\mathbf{I}}$ associated with C is identified by AprilTag id. The resulting feature maps from the ResNet-101 [35] Feature Pyramid Network (FPN) [58] are later used for local feature extraction (Section 5.2.4.3).

Figure 5.7: Example fruitlet and tag detection. Fruitlets are classified as cluster (red) or non-cluster (green). The cluster tag (orange) is identified by AprilTag id.

5.2.4.3 Local Feature Extraction

Contrary to [91], whose local features are a combination of visual descriptors and keypoint positions, we embed additional disparity and segmentation information onto the keypoints. As a result, our local features consist of both visual and positional descriptors. We denote $\mathbf{A}^{\mathrm{I}} = \{\mathbf{B}^{\mathrm{I}} \cup \tau\}$ as the association set. For an $a_i \in \mathbf{A}^{\mathrm{I}}$ with visual descriptor $\mathbf{d}_i$ and positional descriptor $\mathbf{p}_i$, the local feature is $(\mathbf{d}_i, \mathbf{p}_i)$.

For each $a_i \in \mathbf{A}^{\mathrm{I}}$, visual descriptors $\mathbf{d}_i$ are constructed using the extracted feature maps from Section 5.2.4.2 and the ROIAlign [36] operation. ROIAlign is used because it is able to output a feature vector of fixed size regardless of the size of the detected bounding box, while maintaining important semantic information. The feature map at the appropriate pyramid level, determined by the bounding box size, is passed to ROIAlign which outputs an $\mathbb{R}^{256\times7\times7}$ feature vector. To extract positional descriptors $\mathbf{p}_i$, the disparities, segmentations, and x and y pixel locations of the bounding boxes are normalized, stacked, and resized, resulting in an $\mathbb{R}^{4\times64\times64}$ positional vector. Disparity values are used to allow the network to reason about 3D information in the image, without requiring the need to re-project points onto 3D space. Segmentations provide semantic information regarding which pixels in the bounding box are of greater importance. A visualization of our local feature extractor is shown in Fig. 5.8.

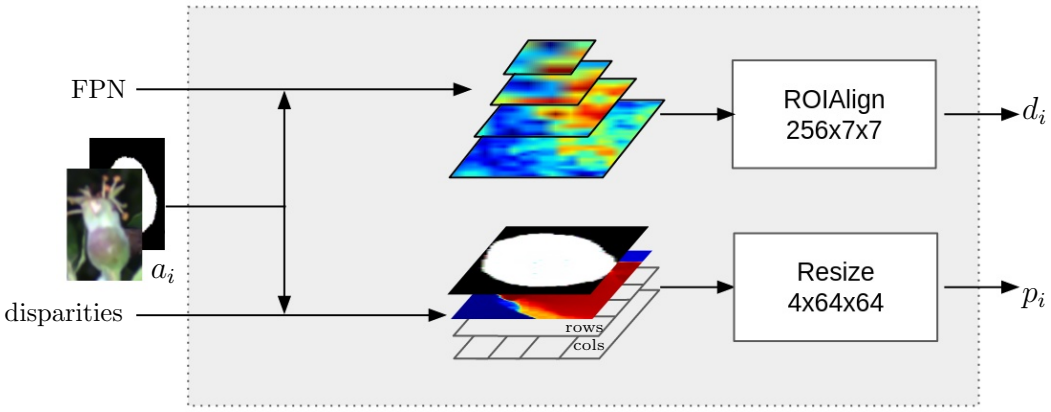


Figure 5.8: Local Feature Extractor. The Mask R-CNN feature maps are cropped and passed to ROIAlign to build the visual descriptor. Positional descriptors are built by stacking the bounding box pixel locations with the cropped disparity values and segmentations, and are resized to a fixed shape.

5.2.4.4 Feature Vector Initialization

The local features for each detection consist of visual and spatial information across multiple pixels. We use CNNs to both reduce the spatial dimensionality and embed the information into deeper features. This allows the network to simultaneously reason about both appearance and 3D position. The initial node feature vectors are defined as

$$^{(0)}\mathbf{x}_i = [\mathrm{CNN}_{\mathrm{d_{enc}}}(\mathbf{d}_i) + \mathrm{CNN}_{\mathrm{p_{enc}}}(\mathbf{p}_i) \parallel s_i \parallel t_i] \tag{5.2}$$

where s_i is the Mask R-CNN cluster prediction score from Section 5.2.4.2 and allows the network to reason about the importance node i. t_i indicates if the node corresponds to a fruitlet $b_i \in \mathbf{B}^{\mathrm{I}}$ or tag τ. Directly injecting t_i into the initial feature vector enables the network to better reason about the spatial relationship between fruitlet and tag nodes.

$$t_i = \begin{cases} 1, & \text{if } a_i \equiv \tau \\ 0, & \text{otherwise} \end{cases} \tag{5.3}$$

5.2.4.5 Attentional Graph Neural Network

Drawing inspiration from [91], we build a multiplex graph [70] whose nodes are $a_i \in \{\mathbf{A}^A \cup \mathbf{A}^B\}$ with initial feature vectors $^{(0)}\mathbf{x}_i \in {}^{(0)}\mathbf{X}$. The graph has two types of edges: self-edges E_{self} that connect all nodes to all other nodes from the same image; and cross-edges E_{cross} that connect all nodes to all nodes from the other image. Message Passing Neural Networks [26] are used to propagate information across edges. For each layer $\ell \in \{0, 1, ..., L\}$, the node feature vectors are updated as

$$^{(\ell+1)}\mathbf{x}_i = {}^{(\ell)}\mathbf{x}_i + ^{(\ell)}\mathrm{MLP}([^{(\ell)}\mathbf{x}_i \| \mathbf{m}_{E\to i}]) \tag{5.4}$$

where $\mathbf{m}_{E\to i}$ is the aggregation of messages arriving from edges $\{j : (i, j) \in E\}$ with $E \in \{E_{\mathrm{self}}, E_{\mathrm{cross}}\}$. Edges alternate each layer between E_{self} and E_{cross}, and each layer ℓ has its own learned MLP.

5.2.4.6 Self-Attentional Aggregation

Messages are aggregated using the Self-Scaled Dot Product Attention mechanism presented in [102]. Scaled Dot-Product Attention is faster and more space efficient than traditional additive attention, and has demonstrated great success in sequence-based relational tasks. Following the notion of retrieval systems, for each edge $\{j : (i, j) \in E\}$, queries $\mathbf{q}_i$ are mapped against a set of keys $\mathbf{k}_j$ to assign weights to values $\mathbf{v}_j$. The message is then the sum of the weighted values

$$\mathbf{m}_{E \to i} = \sum_{\{j:(i,j)\in E\}} \alpha_{ij} \mathbf{v}_j \tag{5.5}$$

where α_{ij} is the softmax over query-key similarity scores

$$\alpha_{ij} = \text{Softmax}_j(\mathbf{q}_i^{\mathrm{T}} \mathbf{k}_j) \tag{5.6}$$

The queries, keys, and values are learned linear projections, with each layer having its own projection parameters. Multi-head attention is used as presented in [102]. The final matching descriptors are calculated from learned linear projections

$$\begin{aligned} \mathbf{f}_i^A &= \mathbf{W} \cdot^{(L)} \mathbf{x}_i^A + \mathbf{b}, \ \ \forall i \in \mathcal{A} \\ \mathbf{f}_j^B &= \mathbf{W} \cdot^{(L)} \mathbf{x}_j^B + \mathbf{b}, \ \ \forall j \in \mathcal{B} \end{aligned} \tag{5.7}$$

5.2.4.7 Optimal Matching Layer

Once final matching descriptors $\mathbf{f}_i^A$ and $\mathbf{f}_j^B$ are calculated, the partial assignment problem must be solved. We use the same optimal transport method as [91] to solve for the partial sum assignment matrix $\mathbf{P} \in [0, 1]^{M \times N}$. A score matrix $\mathbf{S} \in \mathbb{R}^{M \times N}$ is built representing the pairwise score similarity of matching features

$$\mathbf{S}_{ij} = < \mathbf{f}_i^A, \mathbf{f}_j^B > , \ \ \forall (i, j) \in \mathcal{A} \times \mathcal{B} \tag{5.8}$$

where $< \cdot, \cdot >$ denotes the inner product. Dustbins are then added to allow for the assignment of unmatched keypoints. The score matrix $\mathbf{S}$ is augmented to $\overline{\mathbf{S}} \in \mathbb{R}^{(M+1)\times(N+1)}$ by appending a new row and column filled with a single learnable

parameter:

$$\overline{\mathbf{S}}_{i,N+1} = \overline{\mathbf{S}}_{M+1,j} = \overline{\mathbf{S}}_{M+1,N+1} = z \in \mathbb{R} \tag{5.9}$$

We define the constraints for the augmented partial assignment matrix $\overline{\mathbf{P}}$ as

$$\begin{aligned} \overline{\mathbf{P}}\mathbf{1}_{N+1} &= [\mathbf{1}_M^T \ N]^T \\ \overline{\mathbf{P}}\mathbf{1}_{M+1} &= [\mathbf{1}_N^T \ M]^T \end{aligned} \tag{5.10}$$

where M and N are appended because each dustbin should have as many matches as there are detections in the other set. We attempt to maximize the total score $\sum_{(i,j)} \overline{\mathbf{S}}_{i,j}\overline{\mathbf{P}}_{i,j}$ while satisfying the constraints in Equation 5.10. The above optimal transport problem is solved using the Sinkhorn algorithm [18]. After T Sinkhorn iterations, $\mathbf{P} = \overline{\mathbf{P}}_{1:M,1:N}$ is recovered and fruitlet nodes $\mathbf{F}_A$ and $\mathbf{F}_B$ identified by the Mask R-CNN class outputs are matched.

5.2.4.8 Loss

As a result of the Attentional Graph Neural Network and Optimal Matching Layer being fully differentiable, partial assignment predictions can be backpropagated all the way to the local features. Using the same negative log-likelihood loss as [91] with ground truth matched labels $\mathcal{M} = \{(i,j)\} \subset \mathcal{A} \times \mathcal{B}$ and unmatched labels $\mathcal{I} \subseteq \mathcal{A}$, $\mathcal{J} \subseteq \mathcal{B}$, we calculate the partial assignment loss as

$$\begin{aligned} \mathcal{L} = &- \sum_{(i,j)\in\mathcal{M}} \log \overline{\mathbf{P}}_{i,j} \\ &- \sum_{i\in\mathcal{I}} \log \overline{\mathbf{P}}_{i,N+1} - \sum_{j\in\mathcal{J}} \log \overline{\mathbf{P}}_{M+1,j} \end{aligned} \tag{5.11}$$

Unlike in [91], where datasets are created by applying homographies and each keypoint is well-defined in either M, I, or J, our ground truth labels only consist of fruitlets belonging to the target cluster. Therefore, $\mathcal{M}$, $\mathcal{I}$, and $\mathcal{J}$ only consist of clustered fruitlet correspondences, and as a result not all nodes contribute to the loss.

5.3 Experiments and Results

5.3.1 Dataset

5.3.1.1 Data Collection

Our dataset consists of stereo images taken of 252 clusters along with their caliper measurements. The data was collected at the University of Massachusetts Amherst Cold Spring Orchard. The clusters were evenly distributed between three apple varieties of Fuji, Gala, and Honeycrisp. Each cluster was imaged on four different days spread out over an eight day period: 05/18/2021, 05/21/2021, 05/23/2021, and 05/25/2021, which we refer to as Day 1, Day 4, Day 6, and Day 8 respectively. During data collection, the clusters were tagged with an AprilTag, and each fruitlet was assigned a unique id to track across multiple days. A human manually operated the hand-held stereo camera (Fig. 5.2), collecting a sequence of images of each cluster (Fig. 5.9). Data from 42 clusters was used to train the detection, segmentation, and association networks. For the remaining 210 clusters, a single image from each cluster per day was manually selected and used for evaluation. This was done to simulate a human taking a single image in the field. Hand measurements were collected for each cluster using a digital caliper (Fig. 5.10).

Figure 5.9: Example subset of an image sequence of a cluster captured in the field.

Taking measurements with hand calipers naturally results in random errors. This is because measurements will vary as the caliper is rotated around the fruitlet as a result of its asymmetrical shape. As well, the measurements depend on how tightly the caliper is closed around the fruitlet, which can easily change across days. With no exact quantifiable number to express this variation, we asked apple growers how much

this measurement can vary. Their response was between 1-1.5mm for two separate measurements of the same fruit taken on the same day. This is significant, as the sizing process specified by the Fruitlet Growth Model begins when fruitlets are as small as 6mm. This further demonstrates the need for alternative solutions.

Figure 5.10: Example hand caliper measurement of a fruitlet.

5.3.1.2 Annotation Labelling

To train the Mask R-CNN network, 600 images were labelled with bounding boxes around every fruitlet and polygons around each tag. For the pix2pix network, 300 cropped fruitlets were hand-segmented with a binary mask. Both labelled datasets were divided into training, validation, and test sets with a 70/15/15 split respectively. When labelling bounding boxes, each fruitlet was classified as either a cluster or non-cluster fruitlet to be used when training Mask R-CNN.

5.3.1.3 Association Labelling

To label the data used to train the fruitlet association network, a custom tool was developed (Fig. 5.11). Two images of annotated fruitlets and tags on different days are placed side by side, and the user matches the clustered fruitlets by assigning ids. 400 images were labelled to train the network, and datasets were divided into 70/15/15 training, validation, and test splits.

Due to the limited size of the dataset, the data was augmented in several ways. Images were randomly flipped horizontally, and labelled bounding boxes were randomly

Figure 5.11: Custom tool for fruitlet association labelling. (a) Images of the same cluster taken on different days are placed side by side with bounding boxes displayed and clustered fruitlets outlined. (b) The user is able to select and assign matching ids to each fruitlet in the cluster.

shifted and scaled without requiring the width-height aspect ratio to be maintained. To make the network more robust to various graph structures, in each training batch nodes were randomly dropped from the graph and cluster prediction scores randomly shifted. Augmenting the data had a significant improvement on the performance of our network, as demonstrated in Fig. 5.15.

5.3.2 Fruitlet Sizing

We evaluate our fruitlet sizing approach presented in Section 5.2.3. After all fruitlets are sized, outliers are removed from both the caliper measurements and the outputs from our pipeline using a Z-score threshold of 3. This was necessary to mitigate both substantially incorrect caliper recordings and cases where the trained networks failed to generalize. The number of outliers removed were less than 0.5% of all fruitlets sized. The distribution of measured sizes using our computer vision sizing pipeline (CVSP) and the caliper method (CM) are reported Fig. 5.12. and Table 5.1. While the computer vision pipeline consistently produces slightly larger results on all days, the growth trends are similar across the 8 day period.

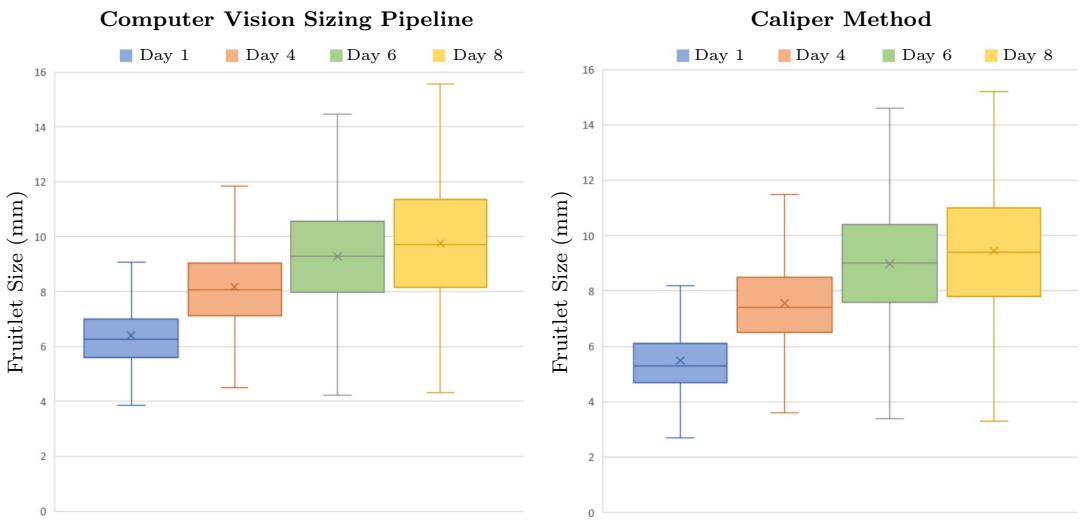


Figure 5.12: Distribution of computer vision and caliper method measured fruitlet sizes. The "x" symbol indicates the mean and the horizontal line indicates the median.

	CVSP Mean	CVSP Med	CVSP Std	CM Mean	CM Med	CM Std
Day 1	6.41	6.25	1.45	5.45	5.30	1.21
Day 4	8.18	8.07	1.63	7.58	7.40	1.68
Day 6	9.32	9.29	2.15	9.01	9.10	2.19
Day 8	9.83	9.80	2.35	9.53	9.60	2.37

Table 5.1: Mean, median, and standard deviations (mm) of our computer vision sizing pipeline and caliper measurements

Table 5.2 shows the mean absolute error (MAE) and the mean absolute percentage error (MAPE) of the computer vision sizes compared against the caliper method. The MAE is just over 1mm on Day 1, and remains under 1mm for the subsequent days. As well, the MAPE is largest on the first day, and reduces as the fruitlets grow, falling under 10% on Day 6. One possible reason for this is it is more difficult to measure disparity values and segment smaller fruitlets. On the other hand, the errors may stem from the inconsistencies in measuring ground truth. The size variations from using calipers would have a more significant effect when the fruitlets are small.

The measurements produced by our method have a high correlation with the caliper measurements, with an R^2 score of 0.826 (Fig. 5.13).

The ultimate goal is to be able to measure fruitlet growth rates to determine when thinning application should be applied. Fruitlets with growth rates less than 50% of the fastest growing fruits are predicted to abscise [30]. Therefore, we evaluate our

	CM Mean (mm)	MAE (mm)	MAPE (%)
Day 1	5.45	1.04	21.3
Day 4	7.58	0.749	11.1
Day 6	9.01	0.691	8.67
Day 8	9.53	0.719	8.45

Table 5.2: MAE and MAPE of our CVSP compared to CM. Mean caliper measured sizes are provided for reference.

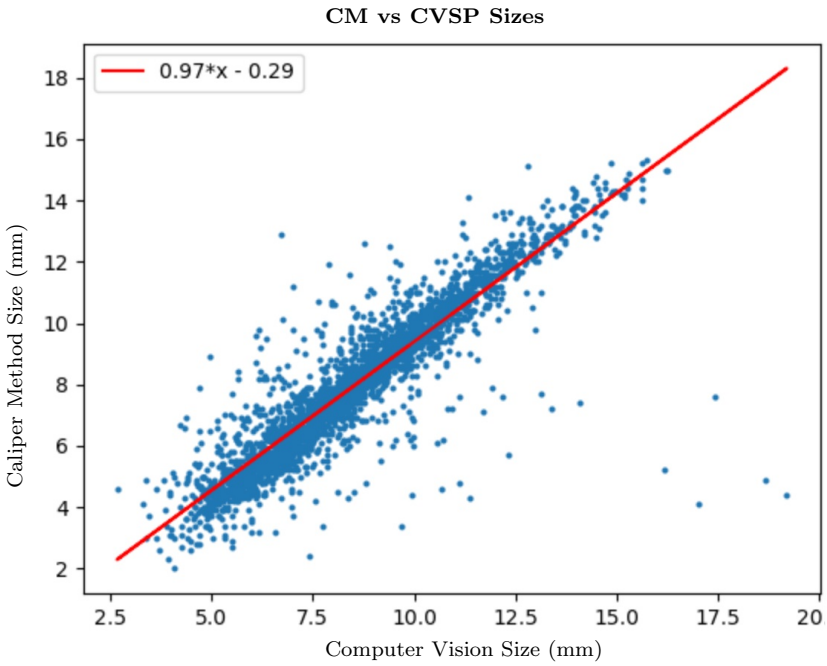


Figure 5.13: Linear fit between CM and CVSP measured sizes.

method's ability to predict the percentage of fruitlets that will abscise and compare the results to the caliper method. According to the Fruitlet Growth Model, fruits should not be measured until they have a diameter of at least 6mm, and measurements should be spaced three to four days apart. We select the date range of Day 4 to Day 8 because both sizing and timing requirements are satisfied. The median growth rate of the top 15% fastest growing fruitlets are used to determine the drop percent. The results can be seen in Table 5.3. The computer vision method predicts that an almost equivalent percentage of fruitlets will abscise, with less than a 1% difference compared to the caliper method. This is what best demonstrates the effectiveness of our approach. Growers could potentially draw the same conclusions about when

to spray using our system as they would using the current method, and without the need to manually size each fruitlet. Fig. 5.14 shows the distributions of growth rates over the date range used. The distributions follow similar trends.

	CVSP MFG	CM MFG	CVSP AP	CM AP
Day 4-8	3.76mm	4.00mm	55.6%	54.8%

Table 5.3: Evaluation of growth rates measured using CVSP and CM. Abscise percent (AP) is calculated using the median of the growth rates of the top 15% fastest growing fruitlets (MFG).

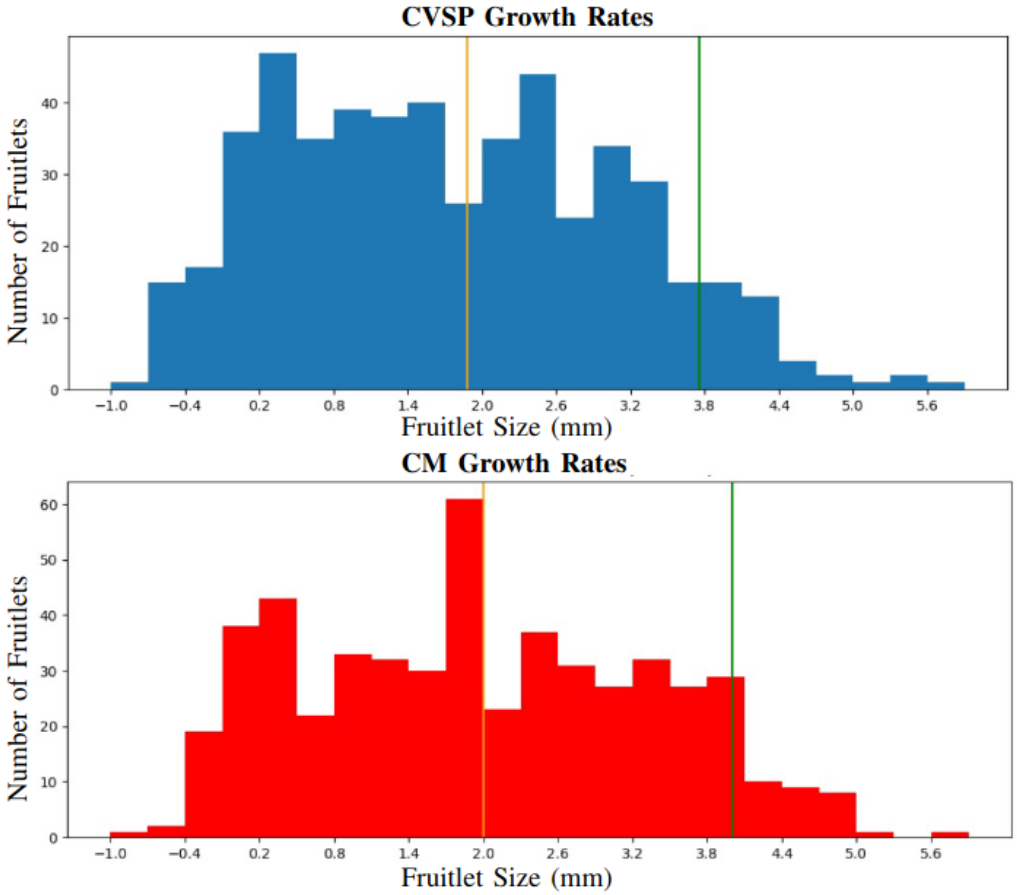


Figure 5.14: Distribution of growth rates over Day 4-8. The green bar represents the median growth of the top 15% fastest growing fruits. The orange bar indicates 50% of this value which is used to calculate the abscise percentage.

We assess the speed of our pipeline. Table 5.4 shows the distribution of sizing times from image input to sizing output. We ran our evaluation on an NVIDIA GeForce RTX 3070 GPU. The average processing time is approximately 4.34s per image, with RAFT-Stereo consuming a majority of the time with an average of 4.05s. The speed of RAFT-Stereo and hence our pipeline will be affected by the GPU used.

	Mean (s)	**Med (s)**	**Std (s)**
RAFT-Stereo	4.05	4.03	0.144
Mask R-CNN	0.205	0.203	0.0198
pix2pix	0.0683	0.0650	0.0400
Sizing	0.0189	0.0184	0.00686
Total	4.34	4.32	0.152

Table 5.4: Runtimes of different CVSP modules.

We asked apple growers how long it usually takes to size fruitlets with calipers. The response we received is that it requires a minimum of 30s per cluster. Based on this preliminary study, our computer vision pipeline was able to size fruitlets 7 times faster compared to hand caliper measurements. We plan to evaluate this performance improvement on a larger scale in the future.

5.3.3 Temporal Fruit Association

We evaluate the performance of our temporal fruit association network. The precision, recall, and matching scores are computed against the partial assignment matching threshold and reported in Fig. 5.15. Matching score is the average ratio of correct associations over the total number of fruitlets belonging to the target cluster. Our network achieves strong performance, with a matching score of 95.1% at a matching threshold up to 0.8. Qualitative examples can be seen in Fig. 5.17.

We also run ablation tests to prove the validity of our design choices. The matching scores with varying match thresholds are reported for the following experiments:

i *Cross-Day Fruitlet Association*: Our primary method from Section 5.2.4.

ii *Simple Positional Descriptor*: Segmentation and disparity information is removed from the positional descriptor $\mathbf{p}_i$, leaving only pixel spatial information.

iii *No Tag Feature Initialization*: The tag information t_i is not concatenated onto the initial feature vector (Equation 5.2).

iv *No Score Feature Initialization*: The score information s_i is not concatenated into the initial feature vector (Equation 5.2).

v *No Data Augmentation*: The data augmentation methods described in Section 5.3.1.3 are removed from training.

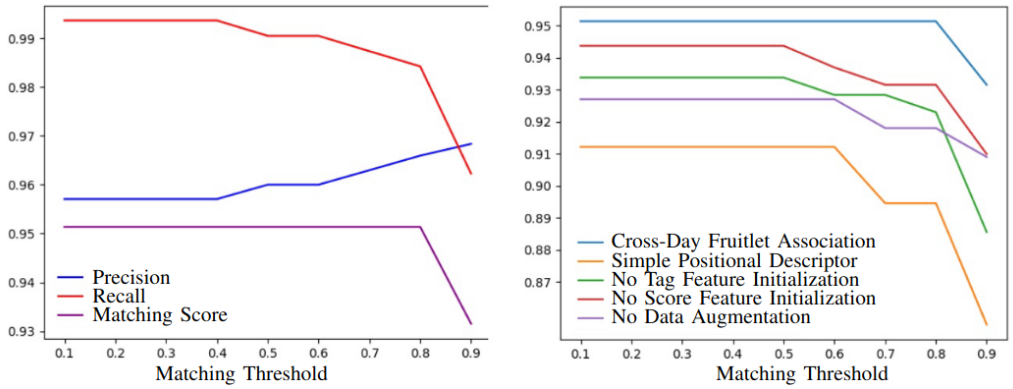


Figure 5.15: Left: precision, recall, and matching score for the temporal fruit association network. Right: ablation study of our temporal fruit association network. Our presented network achieves the highest matching score.

As demonstrated in Fig. 5.15, each of our design choices has a positive effect on matching score. While embedding disparity and segmentation into the positional descriptor leads to the greatest improvement, data augmentation and directly injecting classification score and tag information into the node feature vector all provide additional value, ultimately resulting in a matching score above 95%.

5.3.4 Automated Growth Tracking

We evaluate growth rates measured using a full end-to-end fruitlet growth measurement pipeline (FGMP) and the caliper method across the Day 4 to Day 8 date range. The growth measurement pipeline consists of both fruitlet sizing and association, for a fully automated growth tracking system with no manual fruitlet identification. Outliers are removed from both measurements using a Z-score threshold of 3, with the number of outliers being less than 2.5% of the total measured growth rates. The results are

shown in Fig. 5.16 and Table 5.5. Our end-to-end pipeline predicts approximately the same number of fruitlets will abscise compared to the caliper method, with less than a 3% difference. Apple growers would be able to draw similar conclusions about when to spray using our automated approach that does not require any manual sizing or fruitlet identification.

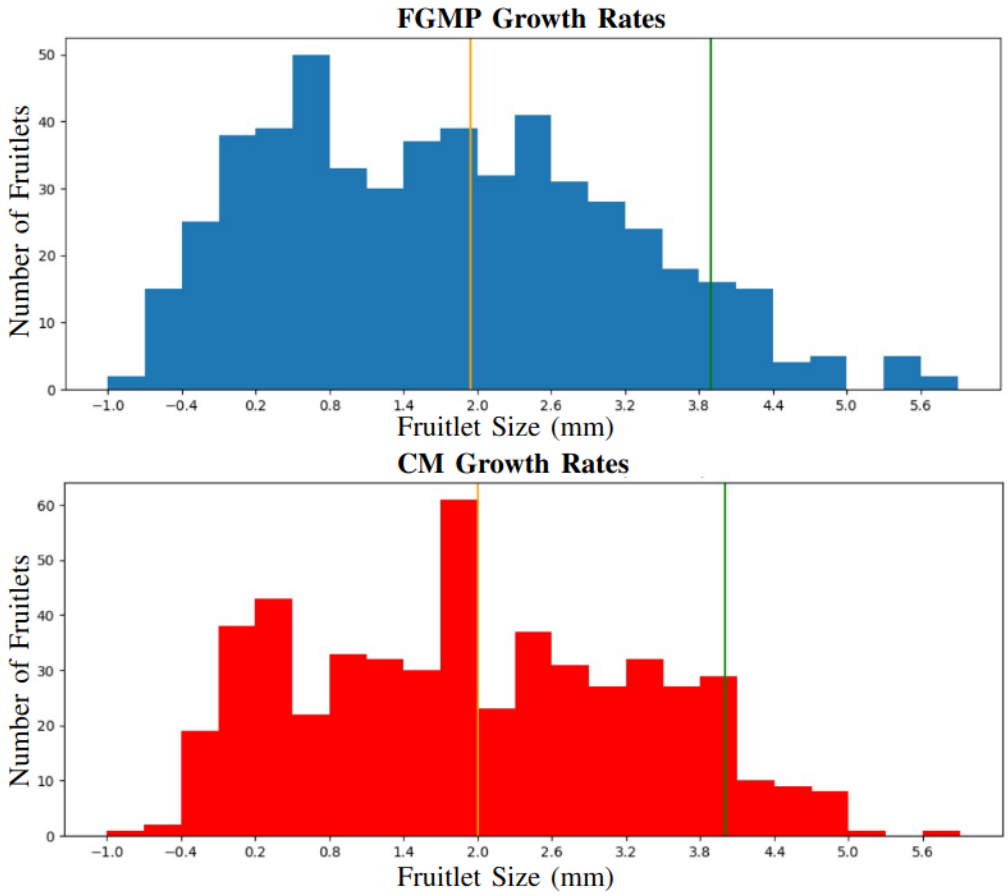


Figure 5.16: Distribution of growth rates over Day 4-8. The green bar represents the median growth of the top 15% fastest growing fruits. The orange bar indicates 50% of this value which is used to calculate the abscise percentage.

The association network took an average of 0.570s to process a pair of images with a standard deviation of 0.0224s.

	FGMP MFG	CM MFG	FGMP AP	CM AP
Day 4-8	3.89mm	4.00mm	57.6%	54.8%

Table 5.5: Evaluation of growth rates measured using FGMP and CM. Abscise percent (AP) is calculated using the median of the growth rates of the top 15% fastest growing fruitlets (MFG).

5.4 Discussion

We present an alternative approach to sizing and measuring the growth rates of apple fruitlets. We have demonstrated that our computer vision-based method is able to produce similar results as the current caliper method used in practice. Most notably, we are able to predict similar abscise rates with only a single stereo image per cluster, without any human effort required to label fruitlets or take caliper measurements. The advantage our system brings is a faster and less labor intensive approach that produces comparable results.

While our approach produces promising results, there is still work needed in order to make it adoptable by growers. For one, it requires the use of a stereo camera that is able to take quality images in light varying environments. We used our custom-made illumination invariant camera system that is not freely available. As well, to process results in real-time, a computationally sufficient device must be carried out in the field and connected to the camera. This brings challenges as the wiring makes it difficult to maneuver around the cluster. In an ideal scenario, lightweight models small enough to run mobile devices would be used. While it is possible to replace the backbone of Mask R-CNN with lightweight networks [43, 56], RAFT-Stereo still requires sufficient computational resources. Alternatively, one advantage to our fully automated approach is images can be collected and processed offline. This would require connecting the camera to a device with sufficient memory capacity to save the captured images to be processed at a future point in time.

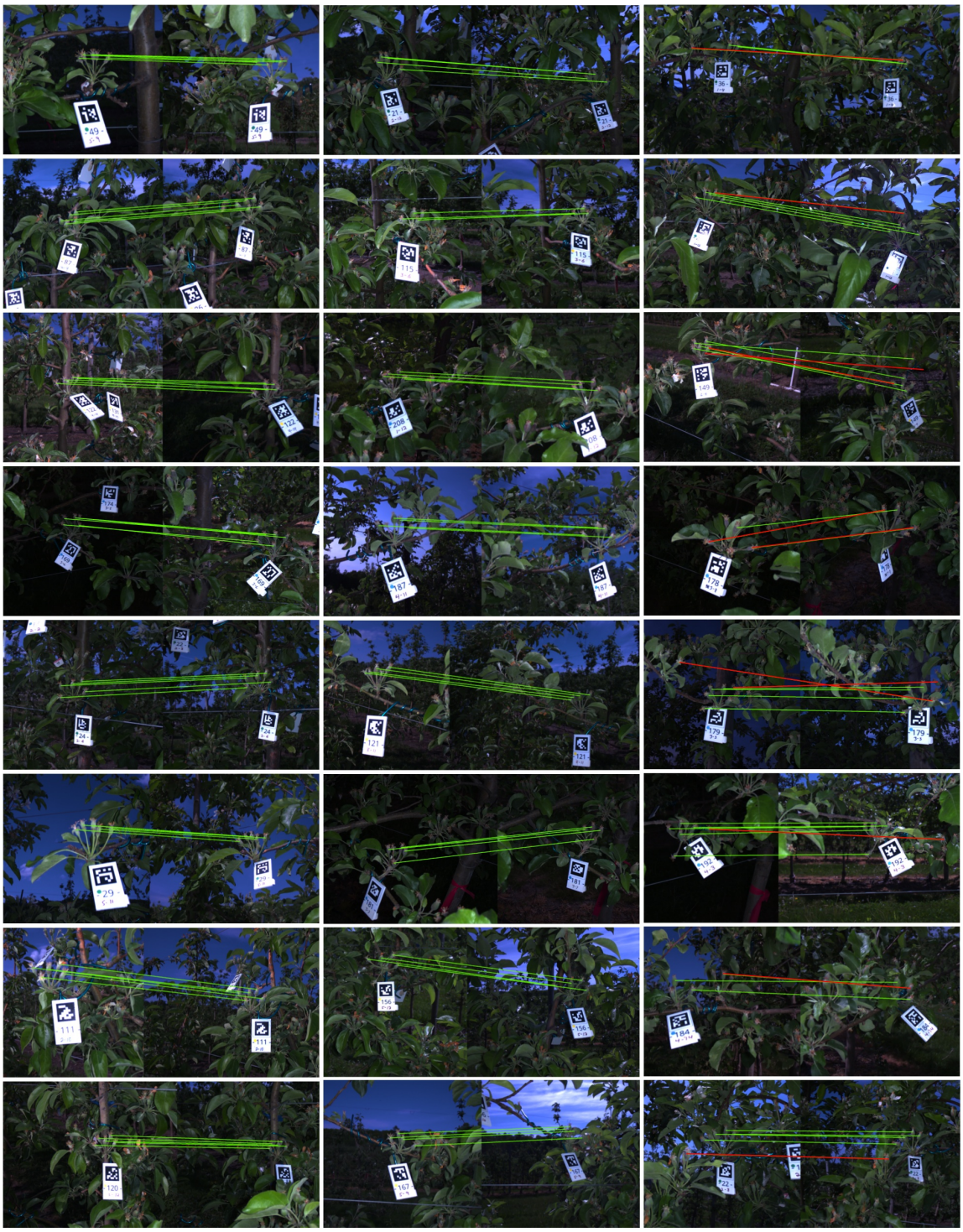


Figure 5.17: Examples of temporal fruitlet associations. Left column: correctly associated fruitlets. Middle column: correctly associated fruitlets when a fruitlet is either occluded or has fallen off. Right column: incorrect association examples.

Chapter 6

Apple Fruitlet Sizing with Next-Best-View Planning

6.1 Motivation

In Chapter 5, we presented a method for sizing apple fruitlets from images captured using a hand-held stereo camera. Despite the significant time and labor improvements compared to using calipers, the system is not fully autonomous as a human is required to capture images in the field. In addition, the approach is limited to using only a single image to size the fruit. It would be beneficial to combine information from multiple views to handle the cases where not every fruitlet is fully visible from a single camera pose. In this chapter, we address these issues by designing a robotic system to autonomously capture images. This is achieved using a next-best-view (NBV) planning approach. The NBV planner has to be able to reason about the environment and determine where the end-effector should go to capture images that will ultimately allow the fruitlets to be sized.

6.2 System Overview

An overview of our system can be seen in Fig 6.1. Our pipeline is composed of two stages. In the first stage, NBV planning is used to capture images. Images are

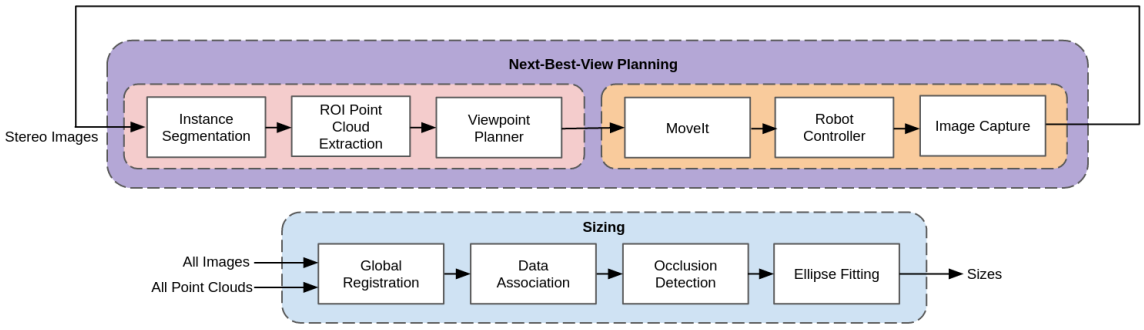


Figure 6.1: Overview of our next-best-view planning and sizing pipeline.

captured using the flash stereo camera as described in Section 3.1. The camera is attached to the end of a 7 DoF robotic arm (Fig. 6.2) consisting of a UR5 and linear slider [94]. When planning, the fruitlets are first segmented, then a point cloud of the scene is extracted with region of interest (ROI) information. The viewpoint planner then updates a dual-map representation of the environment, which is used to sample candidate viewpoints and determine the next best pose for the end-effector based on expected utility. A path is planned to the target pose using the MoveIt framework [17] which is executed by the robot controller. The process repeats until the specified planning duration is exceeded.

The second stage is apple fruitlet sizing, which is performed once viewpoint planning is complete. To account for re-projection error resulting from sensor noise and wind, all point clouds are globally registered using a robust estimation method. Data association between images is then performed using Highly Connected Subgraph Clustering [34] to account for outliers and spurious detections. Lastly, occlusion boundaries are detected which are used to fit ellipses and size the fruitlets.

6.3 Next-Best-View Planning

6.3.1 Instance Segmentation

We replace the detection and segmentation networks from Section 5.2.3 with a single Mask R-CNN [36] network. Because of the time-consuming effort required to label ground truth data for segmenting fruitlets, training was performed in two stages. In the first stage, only the bounding box predictor is trained with the mask head loss

Figure 6.2: In-hand flash stereo camera [93] attached to a 7 DoF robotic arm [94].

set to 0. In the second stage, all weights are frozen except the mask head, which is trained on a subset of fruitlets in each image. Example training images can be seen in Fig. 6.3.

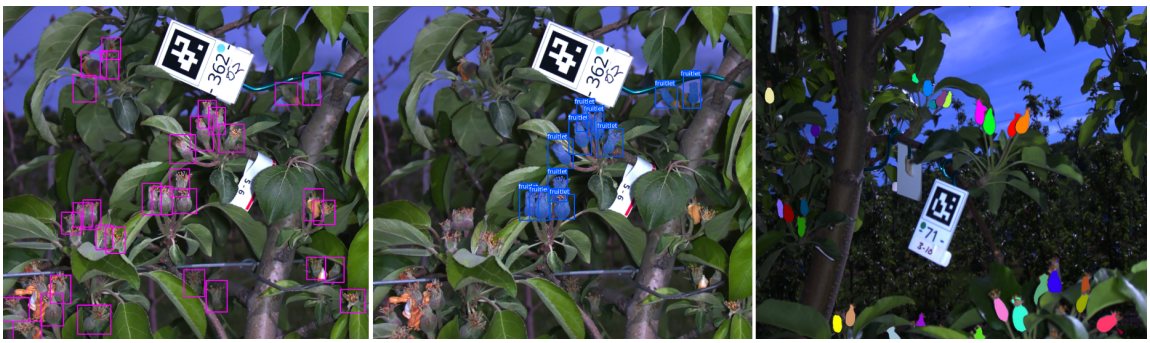


Figure 6.3: Left: Stage 1 - bounding box predictor trained on all fruitlets. Middle: Stage 2 - mask head trained on a subset of fruitlets. Right: Example inference result after training.

6.3.2 ROI Point Cloud Extraction

We extract a point cloud semantically labelled with regions of interest using the segmented fruitlets and estimated disparities. Disparity estimation is performed using the RAFT-Stereo faster implementation [59]. This implementation significantly reduced inference time from 4s to 0.7s compared to the one used in Section 5.2.3.

To reduce noise, the point cloud undergoes a two-stage filtering process. First, a bilateral filter is used to smooth the depths while preserving edges. Second, we apply a depth discontinuity filter presented in [95] to help mitigate the effect of discontinuities on depth measurements.

6.3.3 Viewpoint Planner

Our viewpoint planner adopts ROI and attention-based mechanisms from previous works in order to size smaller fruit. Sampling viewpoint targets in the vicinity of known ROIs [66, 97, 112] has demonstrated strong results with regards to both map coverage and sizing. In addition, Burusa *et al.* [8] showed that restricting attention to specific plant-occupied regions of space when calculating information gain led to improved results with regard to both reconstruction accuracy and speed.

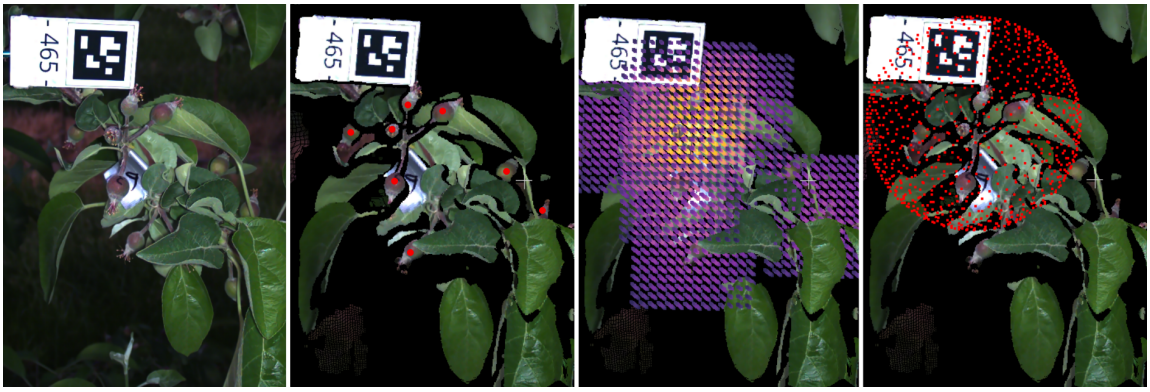
In our implementation, viewpoint targets are sampled using a modified ROI targeted sampling approach from [112]. Possible viewpoint candidates are then evaluated using an attention-guided information gain formulation, and the viewpoint with the maximum utility is selected as the next best camera pose.

Our viewpoint planner utilizes a dual-map representation of the environment in the form of coarse and fine octrees. This representation allows us to significantly speed up expensive ray casting operations resulting in more effective planning time.

6.3.3.1 Workspace and Sampling Tree Generation

Our viewpoint planner uses both workspace and sampling trees to generate viewpoint candidates. The workspace tree defines the valid end-effector poses the robot is able to reach, while the sampling tree identifies the areas of interest that viewpoint targets should be sampled from. To generate the workspace tree, ten million randomly generated joint configurations were sampled in simulation.

Because fruitlets are sized in clusters, we need a sampling tree large enough to encompass every fruitlet in the cluster, but not so large that it includes additional fruitlet clusters. Restricting the size of the sampling tree is beneficial as it reduces planning time and allows viewpoints to better capture the fruitlets of interest. We autonomously generate our sampling tree by detecting the position of the target cluster and fitting a sphere around it. This is an attentive spatial sampling approach similar to the one used by [67] who fit spheres around pre-determined points of interest. An example of the process can be seen in Fig. 6.4.



(a) (b) (c) (d)

Figure 6.4: Sampling tree generation example. (a) Original RGB image with AprilTag hung near cluster. (b) Extracted point cloud with detected fruitlet centroids. (c) Density map created by smoothing around centroids. (d) Sampling tree created from density map local maxima and neighboring fruitlets.

To detect the target cluster, we improve upon our method used in Section 5.2.4 by utilizing 3D information. Similar to Section 5.2.1, an AprilTag is hung near the cluster of interest. The position of the AprilTag need not be consistent across clusters, as long as the desired cluster to be sized is the closest cluster to the tag. An initial image is captured to detect the tag and surrounding fruitlets which are re-projected to 3D. Borrowing our approach from Section 4.2.4, we create a density map by treating each fruitlet center as unit-impulse and applying a 3D Gaussian filter to smooth the region of space around it. Cluster centers are found using local maxima, and each fruitlet is assigned to the closest cluster based on its Euclidean distance.

The cluster whose local maximum is closest to the AprilTag is used as the target cluster. The sampling tree is then constructed by fitting a sphere that encompasses

all of the assigned fruitlets and is centered around the local maximum. To limit the effect of possible incorrect assignments, the sphere radius is restricted to a maximum of 4cm, a quantitative value measured when field testing.

The area spanned by the sampling tree is additionally used as the Attention Region used to compute utility gain as described in Section 6.3.3.4. This is an improvement over the method presented by Burusa *et al.* [8], who assume the location and size of the attention regions are prior knowledge, which is not typically the case when sizing unknown fruit.

6.3.3.2 Dual-Map Representation

State-of-the-art NBV planners in agriculture use expensive ray casting operations to update their volumetric maps and calculate information gain. While popular libraries such as Octomap [41] have been designed to speed up these operations through optimized logic and multi-threading, they are still slow when performed at millimeter-level resolutions. This is not ideal, as sizing apple fruitlets at lower resolutions results in information loss due to the inability to accurately represent regions of interest when sampling viewpoint targets and calculating information gain.

To overcome this, we use a dual-map representation. One limitation of previous ROI-based planners is that they only use a single resolution map to represent the environment. This is inefficient, as ROIs are usually confined to certain areas of space. Instead, our planner maintains two maps of the environment: a coarse octree that stores occupancy information at lower resolution, and a fine octree that stores both occupancy and ROI information at higher resolution. The coarse octree spans the entire observation space and is used to approximate the occupancy of voxels outside the Attention Region, whereas the fine octree is restricted to within the Attention Region and is used to identify which voxels are in the vicinity of the fruitlets of interest. This allows the planner to more efficiently evaluate occupancy and occlusions when motion and viewpoint planning while providing sufficient resolution to plan around fruitlet ROIs. A visualization of our dual-map representation can be seen in Fig. 6.5. To construct the coarse and fine octrees, we use the OctoMap framework [41] and the ROI-extended implementation presented by Zaenker *et al.* [112] respectively.

When ray casting, for both updating the environment model and calculating

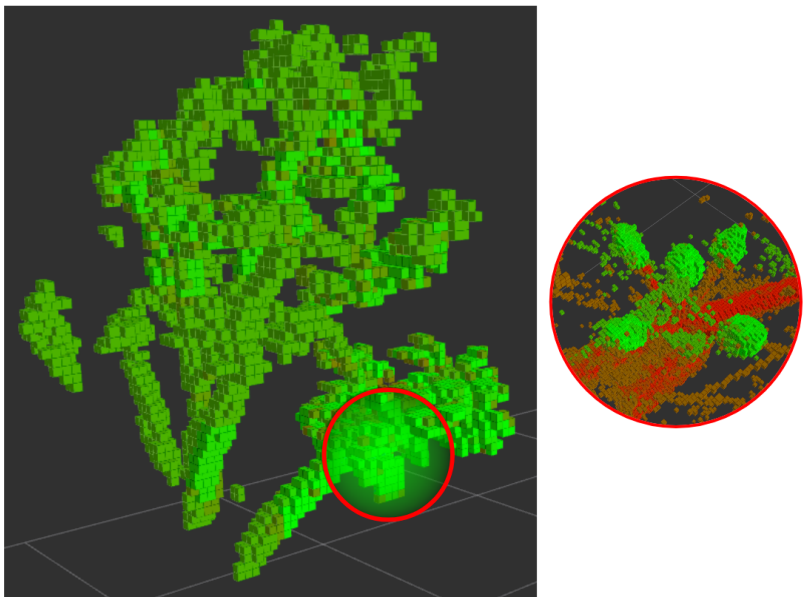


Figure 6.5: Visualization of dual-map representation. Left: Coarse octree that stores occupancy information and spans the entire observation space. Right: Fine octree that stores occupancy and ROI information (green) within the Attention Region.

information gain, rays are cast through nodes in the coarse octree outside the Attention Region, and the fine octree inside the Attention Region, as demonstrated in Fig. 6.6. This significantly improves the computational cost of ray casting by reducing the number of traversed voxels when using higher map resolutions.

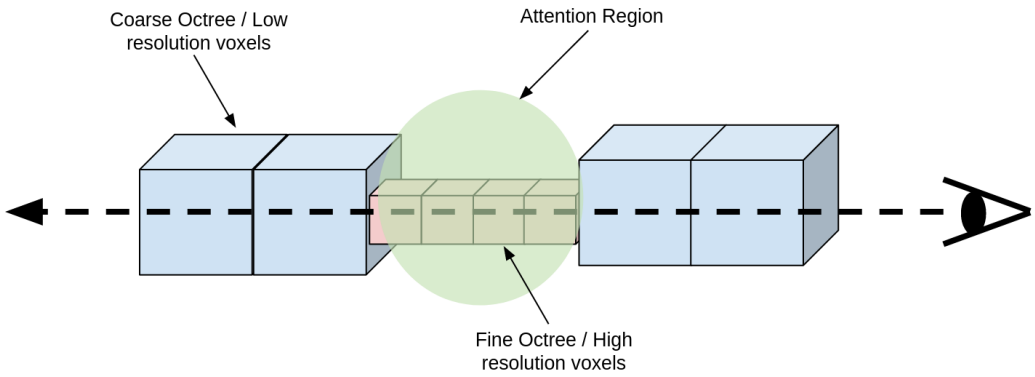


Figure 6.6: Dual-map ray casting implementation. When ray casting, the coarse and fine map are used outside and inside the Attention Region respectively

It is worth noting that because octrees inherently support multi-resolution plan-

ning, this approach could be implemented using a single map. However, the downside to using a single octree is that the resolution of the coarse nodes is restricted to a multiplicative power of two of the resolution of the fines nodes. Because no nodes are added to both the coarse and fine planning trees, maintaining two maps provides more fine-grained control over the planning resolutions while adding a trivial amount of additional overhead.

6.3.3.3 Viewpoint Sampling

Our viewpoint sampling method is inspired by ROI targeted sampling presented by Zaenker *et al.* [112]. First, ROI frontier voxels within the sampling tree are identified and used as target candidates. An ROI frontier is a frontier in the vicinity of a known ROI, where a frontier is the region between an empty voxel and unknown space. To determine ROI frontiers, the 6-neighborhoods of all ROIs are checked for free nodes. For all resulting free nodes, their 6-neighborhoods are checked for unknown neighbors. All free nodes that have at least one unknown neighbor are used as viewpoint targets.

For each viewpoint target, candidate viewpoints are sampled from a partial Fibonacci sphere centered around the target within a specified sensor range. The direction of each viewpoint is oriented towards the target. Viewpoints with positions that do not lie within the workspace tree are discarded.

6.3.3.4 Viewpoint Evaluation

Once all candidate viewpoints are sampled, their estimated information gain (IG) is calculated. For each viewpoint, multiple rays spanning the field of view of the camera are cast from the viewpoint along the direction towards the target using both the coarse and fine planning trees, as described in Section 6.3.3.2. Rays terminate when they encounter an occupied voxel or exceed a maximum distance.

Our IG metric is an attention-guided version of the Unobserved Voxel IG presented by [112]. To apply our attention mechanism, the IG for ray $r \in R$ depends on the number of unknown voxels along the ray that lie in the Attention Region $N_{A,u,r}$. The IG of the ray is $N_{A,u,r}$ divided by the total number of nodes along the ray that are inside the Attention Region $N_{A,r}$. Unknown voxels outside the Attention Region do not contribute. The final IG is averaged across all rays. This attention-based

approach ensures that only unknown voxels that lie within our sizing area of interest contribute to the IG, which results in more informative planning. Equations 6.1, 6.2, 6.3, 6.4, and 6.5 describe how the information gain for a viewpoint is calculated, where X_r is the set of voxels traversed by ray r, and A is the set of voxels that lie within the Attention Region. A visualization for a single ray is provided in Fig. 6.7.

$$I(x) = \begin{cases} 1, & \text{if } x = \text{unknown} \\ 0, & \text{otherwise} \end{cases} \tag{6.1}$$

$$N_{A,r} = \sum_{x \in X_r \cap A} 1 \tag{6.2}$$

$$N_{A,u,r} = \sum_{x \in X_r \cap A} I(x) \tag{6.3}$$

$$IG_r = \begin{cases} \frac{N_{A,u,r}}{N_{A,r}}, & \text{if } N_{A,r} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{6.4}$$

$$IG = \frac{1}{|R|} \sum_{r \in R} IG_r \tag{6.5}$$

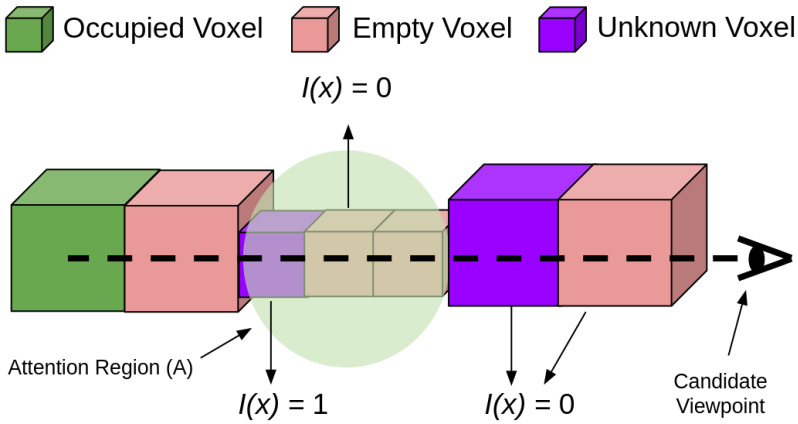


Figure 6.7: Attention-guided information gain formulation for a single ray. Only unknown voxels inside the Attention Region contribute to the information gain.

We also compute a cost C to move to the viewpoint, which is the Euclidean distance between the current camera position and the position of the viewpoint. The

Algorithm 1 Viewpoint Planning

Parameter: u_t

```
for Planning Duration do
    r_s = roiTargetSample();
    h_s = makeHeap(r_s);
    while h_s ≠ ∅ and peek(h_s) > u_t do
        vp = pop(h_s);
        if moveToPose(vp) then
            break;
        end if
    end while
end for
```

cost is scaled by a constant α. The Euclidean distance is used as a cost approximation because computing the joint trajectory for every viewpoint is time-consuming. The final utility of the viewpoint is

$$U = IG - \alpha \cdot C \tag{6.6}$$

6.3.3.5 Viewpoint Selection

Our viewpoint planning algorithm is described in Algorithm 1. Viewpoint candidates are sampled using ROI targeted sampling. A max-heap is created from the viewpoints with order determined by their utility value from Equation 6.6. The planner iterates through the heap until a viewpoint is found with a utility greater than a pre-determined threshold and that the motion planner can find a successful path to. If no viewpoints are left in the heap, new viewpoints are sampled. The process of viewpoint planning and capturing images repeats until the desired planning duration is exceeded.

6.4 Apple Fruitlet Sizing

Sizing is performed once next-best-view planning is complete. This is advantageous as it allows us to use information from all extracted segmented images and point clouds. The sizing pipeline consists of four stages, as depicted in Fig. 6.1. First, to account for sensor noise and wind, point clouds are globally registered using a robust

estimation method. Next, fruitlet detections are associated across images using a graph clustering approach. Lastly, occlusion boundaries are found, and ellipses are fit to the least occluded fruitlets which are used to calculate size.

6.4.1 Global Registration

In each frame, there is an offset in the re-projected point cloud as a result of both sensor noise and wind, as visualized in Fig. 6.8. Qualitatively, we have observed translational shifts in the points clouds typically within the range of 3-5mm, but up to as large as 1cm. While this error may be negligible for larger fruit, is is non-trivial when sizing apple fruitlets, which are often only a few millimeters apart.

(a) (b) (c)

Figure 6.8: Point cloud offset example. Images (a) and (b) with similar camera poses have a noticeable offset in the point clouds (c) as a result of sensor noise and wind.

Using standard methods for point cloud registration, such as Iterative Closest Point between frames, will often fail as a result of the sparse viewpoints and lack of similar structure in the point clouds. Similarly, directly solving for partial assignment using methods such as the Hungarian Algorithm [51] will not work because not every fruitlet is detected in every image due to occlusions.

To globally register the point clouds, we adopt a robust estimation approach. Robust estimation is a popular method used in Simultaneous Localization and

Mapping to reduce the effect of outliers in the assumed model. In our case, outliers represent inconsistent detections across images as a result of occlusions and false positive detections. In our approach, we use robust estimation to globally register all point clouds simultaneously, instead of frame by frame. This produces more reliable results as fruitlet detections across the span of all images are more consistent than between individual frames.

We formulate the task as a nonlinear least-squares problem. The objective is to find a transformation $\mathbf{T_i}$ for each frame $i \in \{1, ..., N\}$ that correctly aligns the re-projected point cloud $\mathbf{P_i}$. For each image $\mathbf{I}_i$, the centroid $c_{ij} \in \mathbb{R}^3$ of every fruitlet $s_{ij} \in \mathbf{S_i}$, $j \in \{1, ..., M_i\}$ is calculated by taking the mean x, y, and z coordinates of the re-projected segmentation masks. Our objective is to find the set of transformations $\mathbf{T} = \{\mathbf{T_1}, ..., \mathbf{T_N}\}$ that minimizes

$$\underset{\mathbf{T}}{\text{minimize}} \frac{1}{2} \sum_{i=1}^{N} \sum_{j=i+1}^{N} \sum_{k=1}^{M_i} \sum_{l=1}^{M_j} \rho(||\mathbf{T_i} c_{ik} - \mathbf{T_j} c_{jl}||^2) \tag{6.7}$$

where ρ is the robust kernel that we select to be a scaled arctan function as presented in Equation 6.8. The optimization problem is solved using a trust-region reflective algorithm [6, 9] with transformations initialized as the camera poses.

$$\rho(x) = \alpha \cdot \arctan\left(\frac{x}{\alpha}\right) \tag{6.8}$$

After solving for the new transformations $\mathbf{T}$, the point clouds correctly align, as shown in Fig. 6.9.

6.4.2 Data Association

After globally registering the point clouds, we can associate the fruitlet detections across images. We need the association method to be robust to spurious and missed detections. To accomplish this, we utilize Highly Connected Subgraph (HCS) clustering [34]. HCS clustering is a recursive graph-clustering method that partitions a graph into subgraphs by taking the minimum cut. If the number of edges in the minimum cut is greater than half the number of nodes in the graph, the graph is determined to be highly connected and is a cluster. Otherwise, the edges in the minimum cut are removed, and the process repeats for each new subgraph.

Figure 6.9: Re-projected point clouds before (left) and after (right) global registration.

We construct a graph $G(V, E)$ consisting of nodes V and edges E. We treat the world-transformed centroid of each detected fruitlet in each image as a node.

$$V = \{\mathbf{T_i} c_{ij}, \forall\ i \in \{1, ..., N\} \text{ and } j \in \{1, ..., M_i\}\} \tag{6.9}$$

And edge e_{ij} is created between nodes v_i and v_j if i) the detections corresponding to the centroids are from different images, and ii) the euclidean distances between the nodes is less than a threshold τ.

$$E = \{e_{ij} \forall\ i, j \text{ s.t. } \text{Im}(i) \neq \text{Im}(j) \wedge \|v_i - v_j\| \leq \tau\} \tag{6.10}$$

$\text{Im}(i)$ represents the image corresponding to node v_i. The first edge requirement enforces that two detections in the same image cannot associate to the same fruitlet. The second edge requirement creates asymmetry in the graph and enforces only reasonable associations. We then cluster the graph using HCS clustering as described in Algorithm 2. Each resulting cluster represents a fruitlet association. Visualizations can be seen in Fig 6.10 and Fig. 6.11.

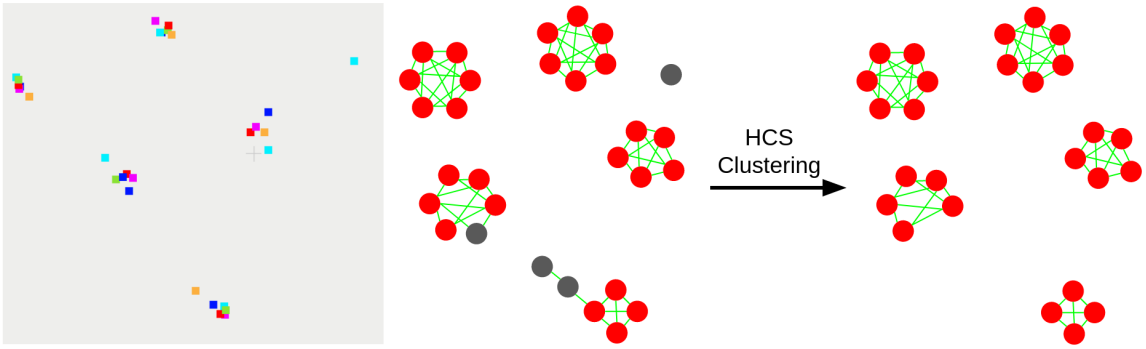
6.4.3 Occlusion Detection and Ellipse Fitting

To determine the best image to use to size each fruitlet, we estimate their occlusion boundaries. This is done using the contours of the segmented fruitlets and the depth

Algorithm 2 Highly Connected Subgraph Clustering (HCS)

Parameter: $G(V, E)$

1: $(H, \overline{H}, C) = MINCUT(G)$
2: **if** G, C is highly connected **then**
3: return G
4: **else**
5: return $HCS(H) \cup HCS(\overline{H})$
6: **end if**



(a) (b) (c)

Figure 6.10: HCS Clustering example. Point cloud of centroids (a) are used to build the graph (b) consisting of false detections (grey). HCS clustering removes the false detections (c) and each subgraph represents an associated fruitlet across images.

Figure 6.11: Data association example. Each color represents the same fruitlet associated in different images.

images. For every pixel in each contour, its 2D normal is calculated in the direction opposite the fruitlet center. The depth values are compared for a specified window size along both directions of the normal. If the depth values along the direction of the normal are smaller, the pixel is determined to be an occlusion boundary. An example can be seen in Fig. 6.12. Due to the close proximity of the fruitlets and the noise in the depth maps, it is challenging to precisely detect which pixels lie at occlusion boundaries. This approximation provides reasonable performance for our application.

Figure 6.12: Occlusion detection example. Top - occlusion boundaries (red) are detected for each fruitlet in the cluster. Bottom - the least occluded images of each fruitlet are used to fit an ellipse and estimate size.

The least occluded fruitlet from a given association across all images is used to size. This is determined by calculating the ratio of occluded pixels to the number of pixels in the contour. An ellipse is fit to the unoccluded pixels, and the size is calculated using the same method as described in Section 5.2.3. A visualization can

be seen in Fig. 6.12.

6.5 Simulated Experiments and Results

6.5.1 Environment

For simulation, we use a Gazebo [50] simulated environment. Fruitlets are modelled as ellipsoids and each fruitlet in a cluster is assigned a unique color, as shown in Fig. 6.13. The different coloring of the fruitlets is used only for instance segmentation and has no effect on data association.

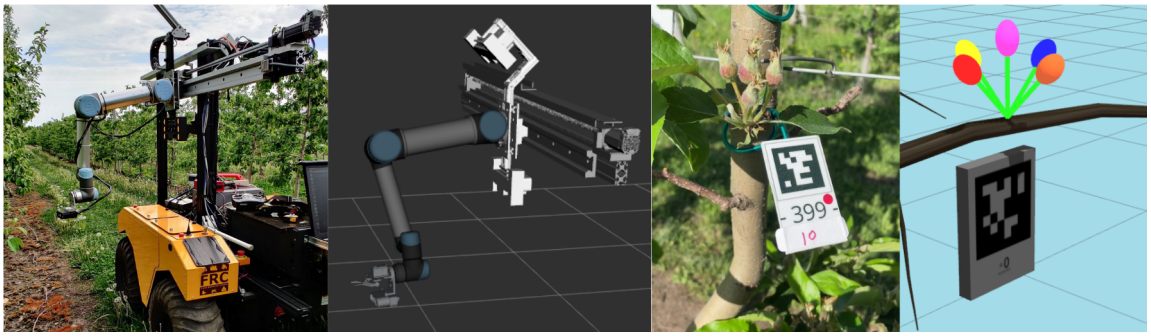


Figure 6.13: Example 7 Dof robotic arm and fruitlet cluster in simulated environment.

Images are captured using a simulated RGB-D camera, and fruitlets are segmented by HSV thresholding. A distance threshold is additionally applied to separate different instances of the same color belonging to different clusters.

The reason an RGB-D camera is used in simulation is because of the difficulties in obtaining accurate disparity information from stereo images in simulated environments. The lack of texture on the ellipsoids makes it difficult to estimate disparity using classic stereo matching methods such as SGBM [39], and the simulated data is too far out of distribution for deep learning-based stereo matching networks. For the purposes of simulation, it was much simpler to use an RGB-D camera, where we could add random noise to the depth image to better replicate the behavior of using stereo in the real-world.

Similarly, we used the different coloring of fruitlets to make instance segmentation more straightforward. This was a simpler approach than training an instance

segmentation network on simulated data, and was more robust than applying color detection and clustering on single colored fruits due to the injected noise in the depth images. This is a fair design choice because the focus of this work is not dedicated towards instance segmentation in simulation.

6.5.2 Dataset

Experiments in simulation were performed on synthetic apple trees. Three different structural topologies with varying levels of foliage were created using SpeedTree[1] and imported into Gazebo. For each tree, 16 trials were run with randomly generated clusters for a total of 48 trials. Clusters were randomly generated with 3-6 fruitlets and varying poses. The sizes of the fruitlets were also generated at random with diameters ranging from 7mm-14mm. For each trial, a single cluster is selected and simulated AprilTag hung in near proximity to it, as shown in Fig. 6.14.

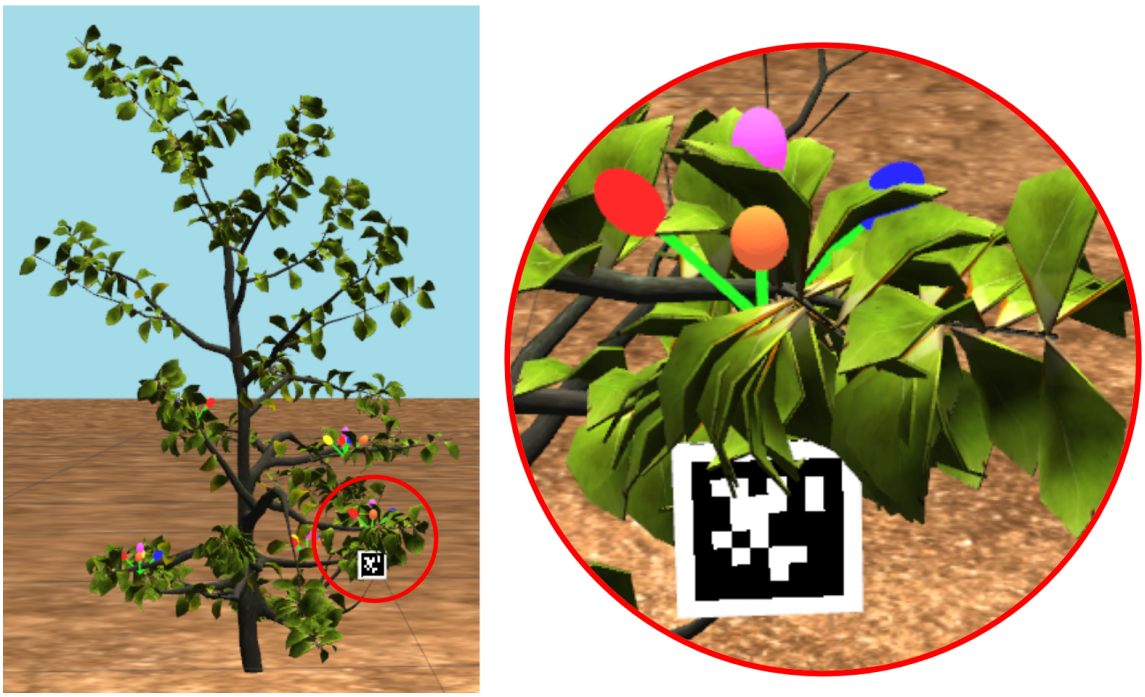


Figure 6.14: Example simulated tree and cluster.

[1] www.speedtree.com

6.5.3 Sizing Results

We assess the effectiveness of our viewpoint planner with ablation tests and comparing it to a slightly modified version of the planner presented in [112]. All planners were given a combined planning and execution time of two minutes to move around the world and capture images. The methods evaluated include

1. *Fruitlet Viewpoint Planner (FVP)*: Our method as described in Section 6.3.3. Coarse and fine octree resolutions of 1cm and 3mm were used.
2. *Single Map Fruitlet Viewpoint Planner (SM-FVP)*: The dual-map representation of the environment from Section 6.3.3.2 was removed and replaced with a single octree of 1cm resolution.
3. *Non-Attention-Guided Fruitlet Viewpoint Planner (NAG-FVP)*: The attention mechanism used to calculate information gain, as described in Section 6.3.3.4, is removed. This is achieved by redefining A in Equations 6.2, 6.3, and 6.4 as the set of all voxels. All unknown voxels along the ray contribute to the information gain. Coarse and fine octree resolutions of 1cm and 3mm were also used.
4. *ROI Viewpoint Planner (RVP)*: The ROI Viewpoint Planner presented by Zaenker *et al.* [112]. The planner was slightly modified to sample viewpoints using only ROI targeted sampling. Exploration sampling was removed because it was unnecessary due to the small volume defined by the sampling tree. An octree resolution of 1cm was used. This method is equivalent to removing both the dual-map representation and the attention-guided utility as described by *SM-FVP* and *NAG-FVP*. Note that our sizing method from Section 6.4 is still used rather than the one presented in [112].

We report the match percent (MP), which is the total percentage of fruitlets that can be matched with ground truth. In addition, we compare the mean absolute error (MAE) and mean absolute percentage error (MAPE) between the measured and ground truth sizes. We also record the rounded mean number of images (RMNI) captured using all planners. The results can be seen in Table 6.1.

$$\text{MAE} = \frac{1}{N}\sum_{n=1}^{N} |\text{measured} - \text{gt}| \tag{6.11}$$

$$\text{MAPE} = \frac{100}{N} \sum_{n=1}^{N} \frac{|\text{measured} - \text{gt}|}{\text{gt}} \tag{6.12}$$

	FVP (ours)	***SM-FVP***	***NAG-FVP***	***RVP***
MP (%)	**95.8**	90.7	87.5	95.4
MAE (mm)	**0.613**	0.834	0.983	0.841
MAPE (%)	**6.17**	8.33	9.92	8.36
RMNI	6	**7**	6	**7**

Table 6.1: Simulated match percent, mean absolute error, mean absolute percentage error, and rounded mean number of images for all planners.

Our FVP achieves the highest match percent and lowest mean absolute and mean absolute percentage errors compared to all ablations. Our planner is able to size 95.8% of fruitlets, and achieves a mean absolute percentage error of 6.17%.

Analyzing the ablations, SM-FVP and RVP were able to capture more images on average. This is expected, as casting rays through the fine octree at 3mm resolution slows down computation compared to using only a single octree at 1cm resolution. NAG-FVP achieved the worst results across all metrics. This is because of the additional computation required to maintain the fine map without the benefits of attention-guided information gain.

In addition, there is a trade-off between the SM-FVP and RVP. The RVP is able to size more fruitlets compared to SM-FVP, but has slightly larger mean absolute and mean absolute percentage errors. This could be explained by the RVP being encouraged to explore more, as voxels outside the Attention Region contribute the information gain. Because SM-FVP applies the attention mechanism, it is incentivized to more accurately explore previously detected ROIs inside the Attention Region.

Regardless of the benefits of SM-FVP and RVP, our FVP effectively demonstrates that, despite capturing fewer images on average, combining a dual-map representation with attention improves results.

We also plot the linear fit between the measured and and ground truth sizes in Fig. 6.15. Our FVP achieves the highest linear fit with an R^2 score of 0.907. SM-FVP, NAG-FVP, and RVP all achieve R^2 scores of 0.824, 0.746, and 0.767 respectively.

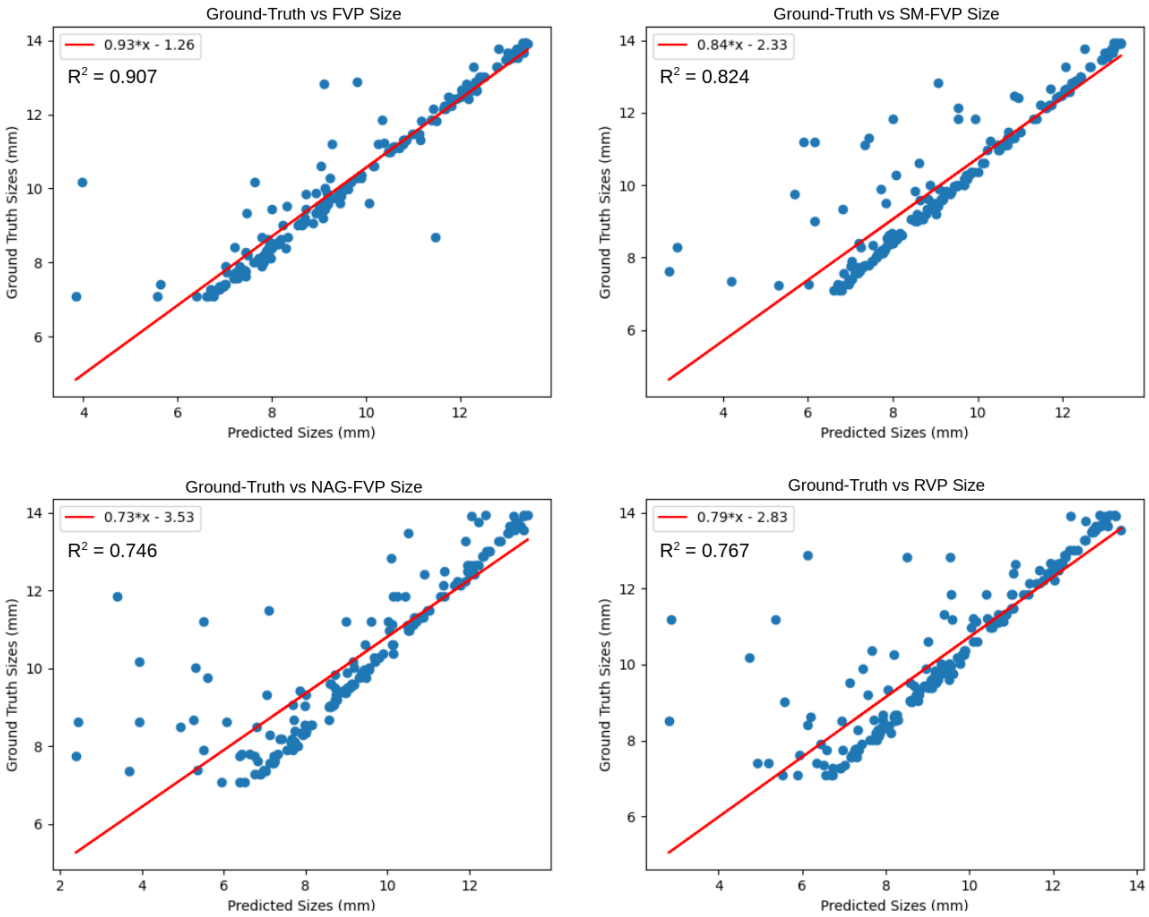


Figure 6.15: Linear fits between simulated ground truth and predicted sizes.

6.6 Real-World Experiments and Results

6.6.1 Dataset

Real-world experiments were conducted at the University of Massachusetts Amherst Cold Spring Orchard. The robotic system presented in Section 6.2 was used to size 30 McIntosh apple clusters on 05/22/2023 using our Fruitlet Viewpoint Planner as described in Section 6.5.3. Caliper measurements were also recorded to be used as ground truth. The instance segmentation network from Section 6.3.1 is trained using 600 stereo images of Fuji, Gala, and Honeycrisp clusters taken on previous field tests.

6.6.2 Sizing Results

We report the match percent (MP), mean absolute error (MAE), and mean absolute percentage error (MAPE) between the measured and ground truth sizes, shown in Table 6.2. In addition, the linear fit between the ground truth and measured sizes can be seen in Fig. 6.16.

	FVP (ours)
MP (%)	88.1
MAE (mm)	0.811
MAPE (%)	7.24

Table 6.2: Real-world match percent, mean absolute error, and mean absolute percentage error for our FVP.

Our Fruitlet Viewpoint Planner demonstrates impressive results on the real-world experiments. The planner achieves a mean average percentage error of 7.24% and an R^2 score of 0.909 between the predicted and caliper-measured sizes.

In addition, 88.1% percent of fruitlets were able to be matched to ground truth and sized. While this is a strong result, it could be improved. Through qualitative inspection, many missed instances were a result of the Mask R-CNN network missing detections. This is likely because no McIntosh apples were used in the training or validation sets. With additional training data the performance on this metric would likely improve.

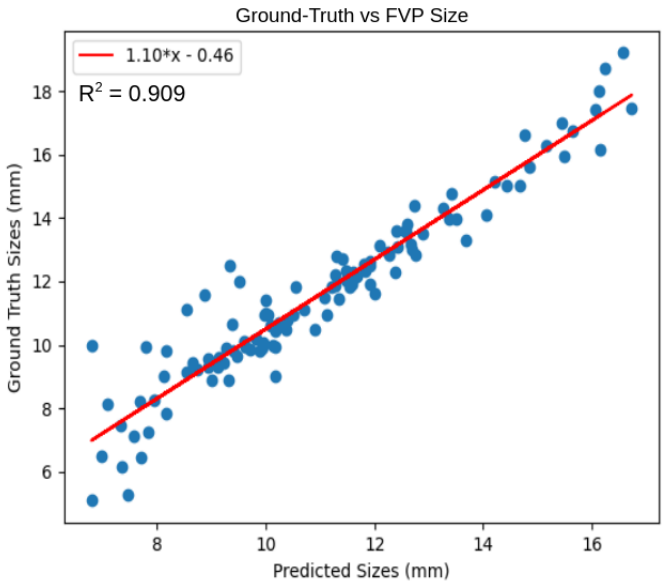


Figure 6.16: Linear fit between caliper-measured and predicted sizes for our FVP.

6.7 Discussion

We present an approach to autonomously capture images and size apple fruitlets using next-best-view planning. Our Fruitlet Viewpoint Planner produces measurements with a strong linear fit to hand caliper measurements, and our global registration and data association methods are robust to wind, sensor error, and false positive and negative detections. One of the benefits of our system is that it could extend beyond sizing apple fruitlets. The proposed methods could also be used to phenotype other young fruit of comparable size, allowing growers to make more informed decisions for a variety of downstream tasks.

Our method could also be adapted to integrate global viewpoint coverage to size multiple clusters of fruit at a time. Instead of using an AprilTag to identify a cluster, global broad scans of trees could be performed to identify high-density fruit regions using the method described in 6.3.3.1. Our viewpoint planner could then be used to capture finer coverage of each region. To achieve this, our dual-map representation could be extended to include multiple fine octrees and Attention Regions.

Chapter 7

Conclusions

It is challenging to non-destructively phenotype smaller crops in agriculture. By leveraging semantic information in the form of seed centers, segmentations, and volumetric regions of interest, we were able to improve the tasks of global registration, temporal fruit association, and viewpoint planning to non-destructively phenotype smaller grains and fruit within the context of three applications:

- Chapter 4 presented an end-to-end pipeline to create an accurate 3D model of a sorghum panicle, which in turn is used estimate seed count. To evaluate our model without ground truth, we introduced an unsupervised point cloud reconstruction metric. The linear fit between counts generated from our model and ground truth were well-correlated, with an R^2 of 0.875.
- Chapter 5 presented a method to size and track growth rates of apple fruitlets from single images collected with a hand-held stereo camera. Sizing was performed by segmenting and fitting ellipses to the fruit, and semantic feature maps were used as part of an Attentional Graph Neural Network to associate fruit detections across days. Our sizing pipeline was able to produce comparable sizes to using hand calipers, achieving a linear fit with an R^2 score of 0.826, but with a 7 times improvement in speed. More importantly, our full end-to-end growth measurement pipeline was able to predict abscise rates within 3% of the caliper method. This demonstrates that our method would allow apple growers to draw similar conclusions about when to apply chemical thinners to crops

without the need to manually size or associate the fruit.

- Chapter 6 presented a next-best-view planning method to make the sizing process fully autonomous, along with a data association approach that is robust to sensor noise and wind. We show in simulation that our dual-map and attention-guided next-best-view planner outperforms all ablation tests and a state-of-the-art ROI planner. As well, our next-best-view planner and sizing pipeline demonstrate strong results in real-world experiments. The sizes produced by our method achieve a mean absolute error and mean absolute percentage error of 0.613mm and 6.17% respectively compared to caliper measurements. 88.1% of fruitlets are able to be sized, and the measurements achieve an R^2 score of 0.909 against ground truth.

Chapter 8

Future Work

For sorghum modelling and seed counting, future work needs to be dedicated to getting the pipeline working in the field. Our images were captured in a lab setting with a black cloth surrounding the panicle, making the tasks of imaging and segmentation easier. To successfully capture a 360° ring of images, the robot would have to be able to either navigate around or manipulate the neighboring panicles in the field. Alternatively, our proposed modelling and counting methods could be adapted to work with partial surface coverage of the panicle. Additionally, more intelligent methods for target-stalk identification and both background and foreground segmentation would need to be integrated in order to use the target seed centers as semantic landmarks.

For apple fruitlet sizing, sizing is still performed using the least occluded image. Valuable information is lost from the images that are not used to size. It would be beneficial to combine information from multiple images to create a 3D model of an apple fruitlet. This could be achieved by fitting 3D shapes such as quadrics to the globally registered and associated point clouds. As well, we only size one cluster at a time, and require that the AprilTag be present in the original image. Future work could be dedicated towards autonomous exploration, where multiple fruitlet clusters are found in high-density regions and are appropriately sized using our next-best-view planning approach.

Bibliography

[1] Yin Bao, Lie Tang, Matthew W. Breitzman, Maria G. Salas Fernandez, and Patrick S. Schnable. Field-based robotic phenotyping of sorghum plant architecture using stereo vision. *Journal of Field Robotics*, 36(2):397–415, 2019. doi: https://doi.org/10.1002/rob.21830. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21830. 1.1

[2] R. Barth, J. IJsselmuiden, J. Hemming, and E.J. Van Henten. Data synthesis methods for semantic segmentation in agriculture: A capsicum annuum dataset. *Computers and Electronics in Agriculture*, 144:284–296, 2018. ISSN 0168-1699. doi: https://doi.org/10.1016/j.compag.2017.12.001. URL https://www.sciencedirect.com/science/article/pii/S0168169917305689. 5.2.3

[3] Graeme Best, Oliver M Cliff, Timothy Patten, Ramgopal R Mettu, and Robert Fitch. Dec-mcts: Decentralized planning for multi-robot active perception. *The International Journal of Robotics Research*, 38(2-3):316–337, 2019. doi: 10.1177/0278364918755924. URL https://doi.org/10.1177/0278364918755924. 2.5

[4] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *CoRR*, abs/2004.10934, 2020. URL https://arxiv.org/abs/2004.10934. 2.2

[5] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 5.2.3

[6] Mary Ann Branch, Thomas F. Coleman, and Yuying Li. A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems. *SIAM Journal on Scientific Computing*, 21(1):1–23, 1999. doi: 10.1137/S1064827595289108. URL https://doi.org/10.1137/S1064827595289108. 6.4.1

[7] Martin Buchner and Abhinav Valada. 3d multi-object tracking using graph neural networks with cross-edge modality attention, 2022. URL https://arxiv.org/abs/2203.10926. 5.2.4

[8] Akshay K. Burusa, Eldert J. van Henten, and Gert Kootstra. Attention-driven active vision for efficient reconstruction of plants and targeted plant parts, 2022.

2.5, 6.3.3, 6.3.3.1

[9] Richard H. Byrd, Robert B. Schnabel, and Gerald A. Shultz. Approximate solution of the trust region problem by minimization over two-dimensional subspaces. *Math. Program.*, 40(1-3):247–263, 1988. doi: 10.1007/BF01580735. URL https://doi.org/10.1007/BF01580735. 6.4.1

[10] Luca Carlone, Jing Dong, Stefano Fenu, Glen C. Rains, and Frank Dellaert. Towards 4 d crop analysis in precision agriculture : Estimating plant height and crown radius over time via expectation-maximization. 2015. 2.4

[11] Akshay L. Chandra, Sai Vikas Desai, Wei Guo, and Vineeth N. Balasubramanian. Computer vision with deep learning for plant phenotyping in agriculture: A survey. *CoRR*, abs/2006.11391, 2020. URL https://arxiv.org/abs/2006.11391. 1.1

[12] N Chebrolu, F Magistri, T Läbe, and C Stachniss. Registration of spatio-temporal point clouds of plants for phenotyping. *PLoS ONE*, 16(2), 2021. 2.1, 2.4

[13] Nived Chebrolu, Philipp Lottes, Alexander Schaefer, Wera Winterhalter, Wolfram Burgard, and Cyrill Stachniss. Agricultural robot dataset for plant classification, localization and mapping on sugar beet fields. *The International Journal of Robotics Research*, 36:027836491772051, 07 2017. doi: 10.1177/0278364917720510. 2.1

[14] Nived Chebrolu, Thomas Läbe, and Cyrill Stachniss. Robust long-term registration of uav images of crop fields for precision agriculture. *IEEE Robotics and Automation Letters*, 3(4):3097–3104, 2018. doi: 10.1109/LRA.2018.2849603. 2.4

[15] Hong Cheng, Lutz Damerow, Yurui Sun, and Michael Blanke. Early yield prediction using image analysis of apple fruit and tree canopy features with neural networks. *Journal of Imaging*, 3(1), 2017. ISSN 2313-433X. doi: 10.3390/jimaging3010006. URL https://www.mdpi.com/2313-433X/3/1/6. 1.1, 2.3

[16] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5556–5565, 2015. doi: 10.1109/CVPR.2015.7299195. 4.2.3

[17] David Coleman, Ioan Alexandru Sucan, Sachin Chitta, and Nikolaus Correll. Reducing the barrier to entry of complex robotic software: a moveit! case study. *CoRR*, abs/1404.3785, 2014. URL http://arxiv.org/abs/1404.3785. 6.2

[18] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transportation distances. 2013. doi: 10.48550/ARXIV.1306.0895. URL https://arxiv.org/abs/1306.0895. 5.2.4.7

[19] Bini Darwin, Pamela Dharmaraj, Shajin Prince, Daniela Elena Popescu, and Duraisamy Jude Hemanth. Recognition of bloom/yield in crop images using deep learning models for smart agriculture: A review. *Agronomy*, 11(4), 2021. ISSN 2073-4395. doi: 10.3390/agronomy11040646. URL https://www.mdpi.com/2073-4395/11/4/646. 1.1

[20] Jeffrey A. Delmerico, Stefan Isler, Reza Sabzevari, and Davide Scaramuzza. A comparison of volumetric information gain metrics for active 3d object reconstruction. *Auton. Robots*, 42(2):197–208, 2018. doi: 10.1007/s10514-017-9634-0. URL https://doi.org/10.1007/s10514-017-9634-0. 2.5

[21] Jing Dong, John Gary Burnham, Byron Boots, Glen C. Rains, and Frank Dellaert. 4d crop monitoring: Spatio-temporal reconstruction for agriculture. *CoRR*, abs/1610.02482, 2016. URL http://arxiv.org/abs/1610.02482. 2.4

[22] Theresa Marie Driscoll. Complete coverage path planning in an agricultural environment. 2011. 2.5

[23] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of 2nd International Conference on Knowledge Discovery and*, pages 226–231, 1996. 4.2.4

[24] A. Feng, H. Li, Z. Liu, Y. Luo, H. Pu, B. Lin, and T. Liu. Research on a Rice Counting Algorithm Based on an Improved MCNN and a Density Map. *Entropy (Basel)*, 23(6), Jun 2021. 1.2, 2.2

[25] Davinia Font, Tomàs Pallejà, Marcel Tresanchez, David Runcan, Javier Moreno, Dani Martínez, Mercè Teixidó, and Jordi Palacín. A proposal for automatic fruit harvesting by combining a low cost stereovision camera and a robotic arm. *Sensors*, 14(7):11557–11579, 2014. ISSN 1424-8220. doi: 10.3390/s140711557. URL https://www.mdpi.com/1424-8220/14/7/11557. 1.1

[26] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. *CoRR*, abs/1704.01212, 2017. URL http://arxiv.org/abs/1704.01212. 5.2.4.5

[27] A. Gongal, A. Silwal, S. Amatya, M. Karkee, Q. Zhang, and K. Lewis. Apple crop-load estimation with over-the-row machine vision system. *Computers and Electronics in Agriculture*, 120:26–35, 2016. ISSN 0168-1699. doi: https://doi.org/10.1016/j.compag.2015.10.022. URL https://www.sciencedirect.com/science/article/pii/S016816991500335X. 2.4

[28] A. Gongal, M. Karkee, and S. Amatya. Apple fruit size estimation using a 3d machine vision system. *Information Processing in Agriculture*, 5(4):498–503, 2018. ISSN 2214-3173. doi: https://doi.org/10.1016/j.inpa.2018.06.002. URL https://www.sciencedirect.com/science/article/pii/S2214317317302408. 2.3,

3.1

[29] M. Gori, G. Monfardini, and F. Scarselli. A new model for learning in graph domains. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 2, pages 729–734 vol. 2, 2005. doi: 10.1109/IJCNN.2005.1555942. 1.2, 5.2.4

[30] Duane W Greene, Alan N Lakso, Terence L Robinson, and Phillip Schwallier. Development of a fruitlet growth model to predict thinner response on apples. *HortScience*, 48(5):584–587, 2013. 5.1, 5.3.2

[31] Eleonora Grilli, Roberto Battisti, and Fabio Remondino. An advanced photogrammetric solution to measure apples. *Remote Sensing*, 13(19), 2021. ISSN 2072-4292. doi: 10.3390/rs13193960. URL https://www.mdpi.com/2072-4292/13/19/3960. 2.3

[32] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. *CoRR*, abs/1706.02216, 2017. URL http://arxiv.org/abs/1706.02216. 5.2.4

[33] Sunil S. Harakannanavar, Jayashri M. Rudagi, Veena I Puranikmath, Ayesha Siddiqua, and R Pramodhini. Plant leaf disease detection using computer vision and machine learning algorithms. *Global Transitions Proceedings*, 3(1):305–310, 2022. ISSN 2666-285X. doi: https://doi.org/10.1016/j.gltp.2022.03.016. URL https://www.sciencedirect.com/science/article/pii/S2666285X22000218. International Conference on Intelligent Engineering Approach(ICIEA-2022). 1.1

[34] Erez Hartuv and Ron Shamir. A clustering algorithm based on graph connectivity. *Information Processing Letters*, 76:175–181, 12 2000. doi: 10.1016/S0020-0190(00)00142-3. 6.2, 6.4.2

[35] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. URL https://arxiv.org/abs/1512.03385. 5.2.4.2

[36] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017. URL http://arxiv.org/abs/1703.06870. 4.2.2, 5.2.3, 5.2.4.3, 6.3.1

[37] Karoline Heiwolt, Cengiz Öztireli, and Grzegorz Cielniak. Statistical shape representations for temporal registration of plant components in 3d, 2023. 2.1, 2.4

[38] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008. doi: 10.1109/TPAMI.2007.1166. 3.2

[39] Heiko Hirschmuller. Stereo processing by semiglobal matching and mutual

information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008. doi: 10.1109/TPAMI.2007.1166. 6.5.1

[40] Takaya Hondo, Kazuki Kobayashi, and Yuya Aoyagi. Real-time prediction of growth characteristics for individual fruits using deep learning. *Sensors*, 22(17), 2022. ISSN 1424-8220. doi: 10.3390/s22176473. URL https://www.mdpi.com/1424-8220/22/17/6473. 2.4

[41] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 2013. doi: 10.1007/s10514-012-9321-0. URL https://octomap.github.io. Software available at https://octomap.github.io. 6.3.3.2

[42] Leland Ralph House. A guide to sorghum breeding. *ICRISAT*, 1985. 1.2

[43] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017. URL https://arxiv.org/abs/1704.04861. 5.4

[44] Stefan Isler, Reza Sabzevari, Jeffrey Delmerico, and Davide Scaramuzza. An information gain formulation for active volumetric 3d reconstruction. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3477–3484, 2016. doi: 10.1109/ICRA.2016.7487527. 2.5

[45] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2018. 5.2.3

[46] Tushar Jadhav, Kulbir Singh, and Aditya Abhyankar. Volumetric estimation using 3d reconstruction method for grading of fruits. *Multimedia Tools and Applications*, 78:1613–1634, 2018. 2.3

[47] Wei Jing, Di Deng, Zhe Xiao, Yong Liu, and Kenji Shimada. Coverage path planning using path primitive sampling and primitive coverage graph for visual inspection, 2019. 2.5

[48] Saeed Khaki, Hieu Pham, Ye Han, Andy Kuhl, Wade Kent, and Lizhi Wang. Deepcorn: A semi-supervised deep learning method for high-throughput image-based corn kernel counting and yield estimation. *Knowledge-Based Systems*, 218:106874, 2021. 1.2, 2.2, 4.4.4

[49] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016. URL http://arxiv.org/abs/1609.02907. 5.2.4

[50] N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on*

Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566), volume 3, pages 2149–2154 vol.3, 2004. doi: 10.1109/IROS.2004.1389727. 6.5.1

[51] Harold W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, March 1955. doi: 10.1002/nav.3800020109. 6.4.1

[52] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *CVPR*, 2020. 4.2.2

[53] Chris Lehnert, Dorian Tsai, Anders Eriksson, and Chris McCool. 3d move to see: Multi-perspective visual servoing for improving object views with semantic segmentation, 2018. 2.5

[54] Christopher Lehnert, Inkyu Sa, Christopher McCool, Ben Upcroft, and Tristan Perez. Sweet pepper pose detection and grasping for automated crop harvesting. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2428–2434, 2016. doi: 10.1109/ICRA.2016.7487394. 2.3

[55] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944. 4.2.3

[56] Hao Li, Aozhou Wu, Wen Fang, Qingqing Zhang, Mingqing Liu, Qingwen Liu, and Wei Chen. Lightweight mask r-cnn for long-range wireless power transfer systems. In *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–6, 2019. doi: 10.1109/WCSP.2019.8927856. 5.4

[57] Yue Li, Jingdun Jia, Li Zhang, Abdul Mateen Khattak, Shi Sun, Wanlin Gao, and Minjuan Wang. Soybean seed counting based on pod image using two-column convolution neural network. *IEEE Access*, 7:64177–64185, 2019. doi: 10.1109/ACCESS.2019.2916931. 1.1, 1.2, 2.2

[58] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *CoRR*, abs/1612.03144, 2016. URL `http://arxiv.org/abs/1612.03144`. 5.2.4.2

[59] Lahav Lipson, Zachary Teed, and Jia Deng. RAFT-Stereo: Multilevel Recurrent Field Transforms for Stereo Matching. *arXiv preprint arXiv:2109.07547*, 2021. 3.2, 4.2.1, 5.2.3, 6.3.2

[60] Xu Liu, Steven W Chen, Chenhao Liu, Shreyas S Shivakumar, Jnaneshwar Das, Camillo J Taylor, James Underwood, and Vijay Kumar. Monocular camera based fruit counting and mapping with semantic data association. *IEEE Robotics and Automation Letters*, 4(3):2296–2303, 2019. 2.2, 2.4

[61] Federico Magistri, Nived Chebrolu, and Cyrill Stachniss. Segmentation-based 4d registration of plants point clouds for phenotyping. In *2020 IEEE/RSJ*

International Conference on Intelligent Robots and Systems (IROS), pages 2433–2439, 2020. doi: 10.1109/IROS45743.2020.9340918. 2.1, 2.4

[62] Salih Marangoz, Tobias Zaenker, Rohit Menon, and Maren Bennewitz. Fruit mapping with shape completion for autonomous crop monitoring, 2022. URL https://arxiv.org/abs/2203.15489. 2.3

[63] Christopher McCool, Inkyu Sa, Feras Dayoub, Christopher Lehnert, Tristan Perez, and Ben Upcroft. Visual detection of occluded crop: For automated harvesting. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2506–2512, 2016. doi: 10.1109/ICRA.2016.7487405. 1.1

[64] Rohit Menon, Tobias Zaenker, and Maren Bennewitz. Viewpoint planning based on shape completion for fruit mapping and reconstruction, 2022. URL https://arxiv.org/abs/2209.15376. 2.5

[65] Gonzalo Mier, João Valente, and Sytze de Bruin. Fields2cover: An open-source coverage path planning library for unmanned agricultural vehicles. *IEEE Robotics and Automation Letters*, 8(4):2166–2172, 2023. doi: 10.1109/LRA.2023.3248439. 2.5

[66] Riccardo Monica and Jacopo Aleotti. Contour-based next-best view planning from point cloud segmentation of unknown objects. *Autonomous Robots*, 42: 443–458, 2018. 6.3.3

[67] Riccardo Monica, Jacopo Aleotti, and Stefano Caselli. A kinfu based approach for robot spatial attention and view planning. *Robotics and Autonomous Systems*, 75:627–640, 2016. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2015.09.010. URL https://www.sciencedirect.com/science/article/pii/S0921889015001980. 6.3.3.1

[68] Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. *CoRR*, abs/1810.02244, 2018. URL http://arxiv.org/abs/1810.02244. 5.2.4

[69] Lawrence Mosley, Hieu Pham, Yogesh Bansal, and Eric Hare. Image-based sorghum head counting when you only look once, 2020. 2.2

[70] Peter J Mucha, Thomas Richardson, Kevin Macon, Mason A Porter, and Jukka-Pekka Onnela. Community structure in time-dependent, multiscale, and multiplex networks. *science*, 328(5980):876–878, 2010. 5.2.4.5

[71] Anjana K Nellithimaru and George A Kantor. Rols: Robust object-level slam for grape counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 2.2

[72] Stephen T. Nuske, Kyle Wilshusen, Supreeth Achar, Luke Yoder, Srinivasa G.

Narasimhan, and Sanjiv Singh. Automated visual yield estimation in vineyards. *J. Field Robotics*, 31:837–860, 2014. 2.2

[73] Stephen T. Nuske, Kyle Wilshusen, Supreeth Achar, Luke Yoder, Srinivasa G. Narasimhan, and Sanjiv Singh. Automated visual yield estimation in vineyards. *J. Field Robotics*, 31:837–860, 2014. 1.1

[74] Nicholas Ohi, Kyle Lassak, Ryan Watson, Jared Strader, Yixin Du, Chizhao Yang, Gabrielle Hedrick, Jennifer Nguyen, Scott Harper, Dylan Reynolds, Cagri Kilic, Jacob Hikes, Sarah Mills, Conner Castle, Benjamin Buzzo, Nicole Waterland, Jason Gross, Yong-Lak Park, Xin Li, and Yu Gu. Design of an autonomous precision pollination robot. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7711–7718, 2018. doi: 10.1109/IROS.2018.8594444. 2.1

[75] Timo Oksanen and Arto Visala. Coverage path planning algorithms for agricultural field machines. *Journal of Field Robotics*, 26(8):651–668, 2009. doi: https://doi.org/10.1002/rob.20300. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20300. 2.5

[76] J.R. Olatunji, G.P. Redding, C.L. Rowe, and A.R. East. Reconstruction of kiwifruit fruit geometry using a cgan trained on a synthetic dataset. *Computers and Electronics in Agriculture*, 177:105699, 2020. ISSN 0168-1699. doi: https://doi.org/10.1016/j.compag.2020.105699. URL https://www.sciencedirect.com/science/article/pii/S0168169920310206. 5.2.3

[77] Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation*, pages 3400–3407, 2011. doi: 10.1109/ICRA.2011.5979561. 5.2.1

[78] Stefan Oßwald, Philipp Karkowski, and Maren Bennewitz. Efficient coverage of 3d environments with humanoid robots using inverse reachability maps. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pages 151–157, 2017. doi: 10.1109/HUMANOIDS.2017.8239550. 2.5

[79] Haolin Pan, Franck Hétroy-Wheeler, Julie Charlaix, and David Colliaux. Multi-scale space-time registration of growing plants. In *2021 International Conference on 3D Vision (3DV)*, pages 310–319, 2021. doi: 10.1109/3DV53792.2021.00041. 2.1, 2.4

[80] Tanvir Parhar, Harjatin Baweja, Merritt Jenkins, and George Kantor. A deep learning-based stalk grasping pipeline. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6161–6167, 2018. doi: 10.1109/ICRA.2018.8460597. 5.2.3

[81] Stefan Paulus, Jan Behmann, Anne-Katrin Mahlein, Lutz Plümer, and Heiner Kuhlmann. Low-cost 3d systems: Suitable tools for plant phenotyping. *Sensors*,

14(2):3001–3018, 2014. ISSN 1424-8220. doi: 10.3390/s140203001. URL https://www.mdpi.com/1424-8220/14/2/3001. 1.1

[82] Juan Manuel Ponce, Arturo Aquino, Borja Millan, and José M. Andújar. Automatic counting and individual size and mass estimation of olive-fruits through computer vision techniques. *IEEE Access*, 7:59451–59465, 2019. doi: 10.1109/ACCESS.2019.2915169. 1.1, 2.3

[83] Ciro Potena, Raghav Khanna, Juan Nieto, Roland Siegwart, Daniele Nardi, and Alberto Pretto. Agricolmap: Aerial-ground collaborative 3d mapping for precision farming. *IEEE Robotics and Automation Letters*, 4(2):1085–1092, 2019. doi: 10.1109/LRA.2019.2894468. 2.1

[84] Danial Pour Arab, Matthias Spisser, and Caroline Essert. Complete coverage path planning for wheeled agricultural robots. *Journal of Field Robotics*, n/a (n/a). doi: https://doi.org/10.1002/rob.22187. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.22187. 2.5

[85] Mohamad Qadri. Robotic vision for 3d modeling and sizing in agriculture. Master's thesis, Carnegie Mellon University, Pittsburgh, PA, August 2021. 5.2.3, 5.2.3, 5.2.3

[86] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016. 2.2, 5.2.3

[87] Alessandro Riccardi, Shane Kelly, Elias Marks, Federico Magistri, Tiziano Guadagnino, Jens Behley, Maren Bennewitz, and Cyrill Stachniss. Fruit tracking over time using high-precision point clouds. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9630–9636, 2023. doi: 10.1109/ICRA48891.2023.10161350. 2.4

[88] Gianmarco Roggiolani, Federico Magistri, Tiziano Guadagnino, Jan Weyler, Giorgio Grisetti, Cyrill Stachniss, and Jens Behley. On domain-specific pretraining for effective semantic perception in agricultural robotics, 2023. 3.1

[89] Pravakar Roy and Volkan Isler. Active view planning for counting apples in orchards. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6027–6032, 2017. doi: 10.1109/IROS.2017.8206500. 2.5

[90] Pravakar Roy, Wenbo Dong, and Volkan Isler. Registering reconstructions of the two sides of fruit tree rows. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9. IEEE, 2018. 2.1

[91] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. *CoRR*, abs/1911.11763, 2019. URL http://arxiv.org/abs/1911.11763. 5.2.4, 5.2.4.1, 5.2.4.3, 5.2.4.5, 5.2.4.7, 5.2.4.8, 5.2.4.8

[92] Yan Shi, Jun-Xiong Cai, Yoli Shavit, Tai-Jiang Mu, Wensen Feng, and Kai Zhang. Clustergnn: Cluster-based coarse-to-fine graph neural network for efficient feature matching, 2022. URL https://arxiv.org/abs/2204.11700. 5.2.4

[93] Abhisesh Silwal, Tanvir Parhar, Francisco Yandun, Harjatin Baweja, and George Kantor. A robust illumination-invariant camera system for agricultural applications. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3292–3298, 2021. doi: 10.1109/IROS51168.2021.9636542. (document), 3.1, 3.1, 6.2

[94] Abhisesh Silwal, Francisco Yandún, Anjana K. Nellithimaru, Terry Bates, and George Kantor. Bumblebee: A path towards fully autonomous robotic vine pruning. *CoRR*, abs/2112.00291, 2021. URL https://arxiv.org/abs/2112.00291. (document), 2.1, 6.2, 6.2

[95] Arjun Singh, James Sha, Karthik S. Narayan, Tudor Achim, and Pieter Abbeel. Bigbird: A large-scale 3d database of object instances. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 509–516, 2014. doi: 10.1109/ICRA.2014.6906903. 6.3.2

[96] Madeleine Stein, Suchet Bargoti, and James Underwood. Image based mango fruit detection, localisation and yield estimation using multiple view geometry. *Sensors*, 16(11):1915, 2016. 2.2

[97] Fouad Sukkar, Graeme Best, Chanyeol Yoo, and Robert Fitch. Multi-robot region-of-interest reconstruction with dec-mcts. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9101–9107, 2019. doi: 10.1109/ICRA.2019.8793560. 2.5, 6.3.3

[98] Alessio Tonioni, Fabio Tosi, Matteo Poggi, Stefano Mattoccia, and Luigi Di Stefano. Real-time self-adaptive deep stereo, 2018. URL https://arxiv.org/abs/1810.05424. 5.2.3

[99] Nikos Tsoulias, Dimitrios S. Paraforos, George Xanthopoulos, and Manuela Zude-Sasse. Apple shape detection based on geometric and radiometric features using a lidar laser scanner. *Remote Sensing*, 12(15), 2020. ISSN 2072-4292. doi: 10.3390/rs12152481. URL https://www.mdpi.com/2072-4292/12/15/2481. 3.1

[100] James P. Underwood, Calvin Hung, Brett Whelan, and Salah Sukkarieh. Mapping almond orchard canopy volume, flowers, fruit and yield using lidar and vision sensors. *Computers and Electronics in Agriculture*, 130:83–96, 2016. ISSN 0168-1699. doi: https://doi.org/10.1016/j.compag.2016.09.014. URL https://www.sciencedirect.com/science/article/pii/S0168169916308249. 3.1

[101] Sebastian Varela, Taylor Pederson, Carl J. Bernacchi, and Andrew D. B.

Leakey. Understanding growth dynamics and yield prediction of sorghum using high temporal resolution uav imagery time series and machine learning. *Remote Sensing*, 13(9), 2021. ISSN 2072-4292. doi: 10.3390/rs13091763. URL https://www.mdpi.com/2072-4292/13/9/1763. 1.1

[102] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. URL http://arxiv.org/abs/1706.03762. 5.2.4.6, 5.2.4.6

[103] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks, 2017. URL https://arxiv.org/abs/1710.10903. 5.2.4

[104] Yawei Wang and Yifei Chen. Fruit morphological measurement based on three-dimensional reconstruction. *Agronomy*, 10(4), 2020. ISSN 2073-4395. doi: 10.3390/agronomy10040455. URL https://www.mdpi.com/2073-4395/10/4/455. 2.3

[105] Yongxin Wang, Xinshuo Weng, and Kris Kitani. Joint detection and multi-object tracking with graph neural networks. *CoRR*, abs/2006.13164, 2020. URL https://arxiv.org/abs/2006.13164. 5.2.4

[106] Zhenglin Wang, Kerry B. Walsh, and Brijesh Verma. On-tree mango fruit size estimation using rgb-d images. *Sensors*, 17(12), 2017. ISSN 1424-8220. doi: 10.3390/s17122738. URL https://www.mdpi.com/1424-8220/17/12/2738. 2.3, 3.1

[107] Zhenglin Wang, Anand Koirala, Kerry Walsh, Nicholas Anderson, and Brijesh Verma. In field fruit sizing using a smart phone application. *Sensors*, 18(10), 2018. ISSN 1424-8220. doi: 10.3390/s18103331. URL https://www.mdpi.com/1424-8220/18/10/3331. 2.3

[108] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861. 4.3

[109] Ulrich Weiss and Peter Biber. Plant detection and mapping for agricultural robots using a 3d lidar sensor. *Robotics and Autonomous Systems*, 59(5):265–273, 2011. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2011.02.011. URL https://www.sciencedirect.com/science/article/pii/S0921889011000315. Special Issue ECMR 2009. 3.1

[110] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 5.2.3

[111] Sierra N. Young, Erkan Kayacan, and Joshua M. Peschel. Design and

field evaluation of a ground robot for high-throughput phenotyping of energy sorghum. *Precision Agriculture*, 20(4):697–722, 2019. doi: https://doi.org/10.1007/s11119-018-9601-6. URL https://link.springer.com/article/10.1007/s11119-018-9601-6. 1.1

[112] Tobias Zaenker, Claus Smitt, Chris McCool, and Maren Bennewitz. Viewpoint planning for fruit size and position estimation. *CoRR*, abs/2011.00275, 2020. URL https://arxiv.org/abs/2011.00275. 2.5, 6.3.3, 6.3.3.2, 6.3.3.3, 6.3.3.4, 6.5.3, 4

[113] Guoxian Zhang and Yangquan Chen. A metric for evaluating 3d reconstruction and mapping performance with no ground truthing. In *ICIP*, 2021. 4.3

[114] Wenli Zhang, Jiaqi Wang, Yuxin Liu, Kaizhen Chen, Huibin Li, Yulin Duan, Wenbin Wu, Yun Shi, and Wei Guo. Deep-learning-based in-field citrus fruit detection and tracking. *Horticulture Research*, 9, 02 2022. ISSN 2052-7276. doi: 10.1093/hr/uhac003. URL https://doi.org/10.1093/hr/uhac003. uhac003. 2.4

[115] Xu Zhao, Rui Wu, Zhong Zhou, and Wei Wu. A new metric for measuring image-based 3d reconstruction. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 1030–1033, 2012. 4.3