Quantifying the Spatial Distribution of Avocados for Optimising Yield Estimation

A thesis presented in fulfilment of the requirements for the degree of Master of Engineering in Mechatronics at Massey University, Manawatū, New Zealand.

JACK WILLIAMS

2024

Abstract

This thesis addresses the need for accurate yield estimation in commercial avocado orchards, leveraging insights from prior research on AI technology for avocado counting. The literature review identified a gap in understanding the spatial distribution of avocado fruit, leading to the development of a 3D positional data collection methodology. In this project a coordinate mapping device employing a laser distance measure and high precision rotary encoders was developed, tested and later implemented in a commercial Hass avocado orchard, resulting in a unique 3D positional dataset of 2,909 measured avocados from 27 trees. The results provide valuable insights into how avocados are distributed on the trees, leading to recommendations for an ideal image capturing procedure. It was found that 98% of all fruit would be included in a field of view extending out to half the row spacing along the row direction and 2.4 m in any other direction from the tree's base. Challenges such as occlusion prompt the need for further research, including simulations and trials based on the collected dataset. Recommendations are made to guide future research towards refining image-capturing procedures for implementing image-processing AI, leading to efficient and accurate individual tree counts. These counting methods can be employed in conjunction with traditional or satellite-based block sampling techniques, facilitating an accurate overall block yield estimation and providing growers with the necessary insights to make informed commercial harvest decisions.

Acknowledgements

I would like to express my appreciation to my supervisors Andrew East and Gabe Redding, for their guidance and feedback throughout this project. Special thanks to Peter Jeffery who readily dedicated many hours to the data collection process. I'd also like to thank Anna Bailey for allowing me to conduct trials on her personal avocado tree, as well as the team at King Avocado orchard, especially Claudia Hermosillo for her invaluable assistance. Finally, I'd like to thank my wife Linda for her continued support during the year, and valued contribution to the editing of this thesis.

List of Abbreviations

AI	Artificial Intelligence
CAD	Computer Aided Design
CMD	Coordinate Mapping Device
CMM	Coordinate Measuring Machine
CNC	Computer Numeric Control
D-H	Denavit-Hartenberg
FOV	Field of View
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GUI	Graphical User Interface
IDE	Integrated Development Environment
IMU	Inertial Measurement Unit
KDE	Kernel Density Estimate
LCD	Liquid Crystal Display
LiDAR	Light Detection and Ranging
LOS	Line of Sight
LPS	Local Positioning System
PPR	Pulses Per Revolution
PTFE	Polytetrafluoroethylene
RKT	Real Time Kinematic
SPI	Serial Peripheral Interface
UART	Universal Asynchronous Receiver/Transmitter
3D	Three Dimensions/Dimensional

Table of Contents

List of Figures

List of Tables

CHAPTER 1
INTRODUCTION

1.1 Background

Hass avocados are an important export for New Zealand, with increasing demand both domestically and internationally [1, 2], and over 2000 commercial growers across the North Island [3]. Yield estimation is a critical aspect of avocado growing as it enables growers to plan and manage their harvest efficiently [1]. Accurate yield estimation helps growers to optimise their resources, such as labour, transportation, and storage facilities which are essential for the efficient management of harvest and post-harvest activities [4, 5]. With an accurate estimation of yield, growers can determine the best time and quantity for harvest, ensuring that the fruit is picked at the right maturity level, resulting in optimum quality and quantity of produce. Because Hass avocados can be stored on the tree [6], yield estimation also enables growers to forecast their future harvests, which is valuable information for marketing, sales, and financial planning. Furthermore, an accurate yield estimation can help avocado growers make informed decisions regarding the allocation of resources, including fertilizers, water, and pest control measures which can significantly impact crop yield [1, 4]. Overall, yield estimation is crucial for avocado growers as it provides valuable insights and data for managing and optimising their crop production effectively. There is a need to develop more accurate and efficient methods for yield estimation in avocado orchards to help growers make better management and commercial planning decisions.

This master's project is a continuation of a previous year's research master's by Kyle Macadam that was conducted in collaboration with MAF Digital Lab and King Avocado [7]. Macadam's thesis and literature review explored the current methods employed by avocado growers for pre-harvest avocado yield estimation as well as new technologies such as remote sensing, unmanned aerial systems, computer vision, and artificial intelligence that could provide more reliable and precise yield estimates, reducing the time and cost associated with manual counting. The research and findings of Macadam ultimately led him to develop an artificial intelligence model capable of identifying avocados within a photograph that serves as an intermediary step in overcoming the problem of yield estimation. For this reason, the literature review included in this thesis

will not thoroughly explore alternate methods of avocado yield estimation that Macadam determined to be less suitable but will build upon the findings of his research and the AI model he developed for detecting avocados in an image. This model was trained on 512 by 512-pixel images and could provide a real-time avocado count within a new image with 88% accuracy. During some trials, the model was able to predict some fruit per tree counts within 5% of the actual harvest count. Further training and refinement could be conducted on this model. However, the main problem that now prevents this model from providing a viable solution to yield estimation is determining how images should be captured of the trees within orchards that would allow the implementation of the model to estimate the overall number of avocadoes on a tree and orchard block.

1.2 Aim of Study

The research presented in this thesis addresses the need for accurate yield estimation in commercial avocado orchards. Building upon insights gained from a previous master's study [7] and the following literature review, further research is required to optimise methods for capturing visible spectrum photographs of Hass avocados in New Zealand orchards.

While a photographic approach that aims to count every avocado on the trees seems exhaustive, it poses challenges such as overlapping images, leading to potential double-counting of fruit. Moreover, the issue of occlusion caused by branched and foliage further complicates the photography process, as some fruits may remain obstructed and thus uncountable. By understanding the spatial distribution of fruit, this research aims to address these challenges to refine an image capture procedure within the canopy and guide future research on yield estimation using AI.

In the absence of existing data on the spatial distribution of avocados, the primary objective of this study is to research methods for accurately measuring the 3D position of fruit, aiming to quantify and gain insight into their spatial distribution. Without an accessible means for collecting the required positional data, the design, development, and validation of a measuring device are pivotal for this research, enabling the collection of a unique 3D dataset of avocado positions. This dataset will serve as a foundation for optimizing an image-capturing procedure in commercial avocado orchards.

1.3 Thesis Outline

This thesis begins with a literature review that investigates existing methods of yield estimation, associated challenges, and the significance of accurate estimation for commercially cultivated avocados. It delves into current technologies in commercial growing and yield estimation, as well as potential technologies for collecting spatial data to advance AI yield estimation through optimized image-capturing techniques.

The methodology section encompasses the development and testing of a Coordinate Mapping Device (CMD) capable of capturing accurate positional data of avocados. The methods used to validate the accuracy of the device through all stages of development are outlined, and results from a small-scale trial on a local avocado tree are presented that demonstrate its feasibility to efficiently measure avocados on a tree. The comprehensive methodology used to deploy the CMD at the King Avocado orchard and gather positional data from 27 trees is also presented, detailing aspects such as the measurement setup, required equipment, data-saving procedures, and orchard specifications.

In the results and discussion section, the collected dataset is discussed and evaluated for trends in fruit distribution, shedding light on how images should be captured to represent the true number of avocados on a tree accurately. The analysis also addresses challenges like fruit overlap from neighbouring trees and issues such as occlusion caused by branches and foliage.

Finally, conclusions and recommendations are then presented based on the findings from the results. These recommendations aim to guide the development of a yield estimation solution through future research into an optimised image-capturing procedure required to implement the existing AI counting model.

CHAPTER 2
LITERATURE REVIEW

2.1 Literature Review Introduction

Given the established importance of accurate yield estimation for commercial growers and the technological advancements outlined in prior research [7], this literature review aims to explore and optimize methodologies for capturing visible spectrum photographs of Hass avocados cultivated in New Zealand orchards. The ultimate objective is to process these images using a pre-trained AI model [7] for efficient avocado counting. The research specifically focuses on methodologies to collect positional data of avocados as they mature on the tree. This data is crucial for understanding the spatial distribution of the fruit, offering insights into optimal tree photography and guiding future research necessary for the development of a robust yield estimation process. The literature review will delve into current yield estimation methods employed by commercial growers and investigate pertinent details related to avocado growth that may impact the design of a 3D measuring device capable of quantifying the spatial distribution of the fruit.

The literature review will also examine different sampling techniques used to estimate yields in orchards. The findings of this literature review will identify the uses of artificial intelligence models applied to estimate the yield of crops from photographs. This information will be used to identify the most appropriate methodology to achieve the objectives of this research. The literature review will also identify gaps in the literature and highlight potential areas for future research.

2.2 Yield Estimation Techniques

Many of the methods employed by avocado growers to estimate their yield before harvest involve physically counting the number of fruit on a sample of trees and extrapolating the count to estimate the total yield of the orchard. However, this method is prone to errors as it relies on visual estimates, which can vary depending on the observer's experience, perspective, and fatigue level [8, 9]. As a result, manual yield estimation can be inaccurate, leading to over or underestimation of the actual yield, which can have significant economic impacts on growers [1, 4]. A manual count provides a snapshot of the fruit count at only one specific time. However, commercial growers can also benefit

from to receiving multiple yield estimations throughout the year, so it is crucial that the methods they employ minimise cost, labour and time [9].

Although capturing images within an avocado tree canopy for use with AI to estimate yield is the main scope of the project, existing yield estimation methods involving manual labour will be explored to highlight their limitations and the need for a more modern and scalable solution. Other developing areas of yield estimation not directly related to the image capture and AI processing of avocados will also be investigated which may provide relevant information and research that can benefit this master's project.

2.2.1 Manual Counting

Manually estimating the yield of a commercial orchard for a given year is a labour-intensive process involving workers traversing between trees and blocks of an orchard to visually count the number of avocados within a canopy. Many growers report this method to be unreliable by itself, and that it always includes an element of “feel” and guesswork [9]. Some growers report they achieve better results by gauging the relative size and distribution of fruit within the trees as well as the strength of the flowering events in comparison to previous years, rather than attempting to count the number of fruit [9].

Although most orchards do gather some form of manually acquired yield estimate, this cannot be reliably used by itself to make quantitative predictions on harvest yield but is instead used in conjunction with past years' counts that were gathered with as much consistency as possible to provide a comparative estimate. It is common practice to utilise extensive historical records of individual blocks of avocado trees within commercial orchards to aid in current yield estimation. These records not only consist of previous harvest counts at a block level but also include the variety, rootstock, age, number of trees, harvest weight, etc., that aid growers in identifying trends between past yields and growing conditions [9]. This allows growers to make predictions based on historical data rather than relying solely on manual tree counts for their overall yield estimation, which usually includes significant errors.

Alternate methods of manual yield estimation of avocado trees have been explored to gather a standardised fruit density count of a given tree in conjunction with its capacity

[1, 10]. This method requires estimating the canopy volume of a tree and using a standardised set of binoculars from a specific distance away from the canopy to find the yield density of an isolated section of the tree. Error arises from this method because of factors such as inaccurate canopy measurements, nonuniform fruit distribution throughout a tree, and a dense canopy of leaves obscuring the fruit from view. The problem of fruit being obscured by leaves and branches within a tree will also be evident when capturing images for AI processing, but through the collection of position and line of sight data of fruit within a canopy, trends and patterns may be discovered that provide insight into optimal angles and positions from which photos can be taken.

The methods involving manual estimation remain inaccurate even when grower experience and historical data are taken into account [4]. A study conducted by the University of New England across five avocado orchard blocks in 2016 and 2017 compared the yield estimation error between the manual count and harvest count to range from 51% under to 125% over [4]. Given the economic impact that accurate yield estimation has on commercial growing and harvesting decisions, there is a significant opportunity for alternate methods of yield estimation with a wide margin for success.

2.2.2 Photography

Visual spectrum photography and computer vision techniques are widely used for crops such as apples because they lend themselves well to the concept of 2D farming, where the canopy structure and fruit distribution can be easily captured in 2D images [11-13]. These crops typically have a more uniform canopy and fruit distribution, allowing for straightforward image analysis and accurate yield estimation.

Similar techniques have also been used on non-2D crops such as mangos where the fruit can be seen and photographed from the outside of the canopy [14, 15]. In contrast, crops like avocados present unique challenges due to their complex canopy structure, irregular fruit distribution, and foliage characteristics. The dense foliage of avocado trees can obstruct the visibility of fruits, making it difficult to capture accurate images solely in the visual spectrum [16].

A student project from California Polytechnic State University investigated image-capturing techniques and computer vision to tackle the problem of yield estimation for individual avocado trees [16]. This project utilised thermal imaging cameras at specific times of the day in conduction with visual spectrum photography to further isolate the avocadoes. Results showed that the visible avocadoes were successfully segmented from the background and a total count within individual photographs could be provided. It was also found that it is unlikely that a camera can capture all the avocados in a tree within its field of view because of the many fruits that are hidden within and around the canopy. This was compensated for by incorporating a prediction model with the measured canopy size and the fruit density estimate from the thermal images. Although this method incorporates more advanced computer vision tools to identify fruit density in areas of a tree's canopy, it still includes the same error and complexities introduced by manually estimating the fruit density with optical equipment and multiplying it by a measured canopy volume [10].

It is evident that the ability to capture fruit density at a given point in the tree's canopy will not be sufficient to estimate the total number of avocados on a tree. Research is needed to understand the distribution of avocados within trees and identify optimal methods for capturing photographs to estimate yields without relying on labour-intensive measurements or assuming consistent fruit distribution.

2.2.3 Artificial Intelligence

Artificial intelligence (AI) plays a significant role in agriculture for yield estimation by leveraging advanced algorithms and data analysis techniques [17, 18]. AI systems can process large amounts of agricultural data, including satellite imagery, weather patterns, soil conditions, and crop characteristics [17, 18]. By analysing these data sets, AI algorithms can identify patterns, correlations, and trends that contribute to crop yield [19].

One common approach is to use machine learning algorithms to train AI models on historical agricultural data [18, 20]. These models learn from the data and develop the ability to predict crop yields based on various factors. For example, they can analyse satellite images to assess crop health, monitor vegetation indices, detect diseases or pests,

and estimate crop biomass. It is important to note that this technique relies on predictive methods rather than a quantitative evaluation of current data to accurately estimate yield.

There are multiple studies done on the feasibility of using AI to identify and count fruit including avocadoes within a visual spectrum photograph [5, 7]. This research successfully demonstrates the ability of AI to identify fruit within an image, however, the problem of optimizing an image-capturing process to best apply these AI models to achieve an overall yield estimation on an entire tree has not been addressed in detail.
Further research is required to develop this technology beyond simply counting the number of avocados in an image, into a capable yield estimation tool that can be applied across an entire orchard block.

2.2.4 Satellite

Satellite imagery is employed to assess the general health of individual trees or specific areas within a crop block, primarily to identify correlations between canopy health and yield based on historical records [4, 21-23]. The use of satellites can be beneficial for yield estimation to the extent that it can aid in optimising block sampling methods but is always used in conjunction with on-ground sampling methods [23].

Satellite imagery alone does not provide independent quantitative counts of avocado yield that are devoid of other variables. Instead, it serves as a tool to observe trends and patterns in canopy health and yield, allowing growers to make informed assessments regarding block sampling.

2.2.5 Multispectral Imaging

Multispectral imaging is a remote sensing technique that involves capturing images of an object or area using multiple discrete bands of the electromagnetic spectrum. Each band corresponds to a specific range of wavelengths, allowing for the analysis of various spectral properties of the target [24-27].

Multispectral and hyperspectral imaging has been successfully used for the evaluation of diseases, fatty acid and mineral nutrient concentration, and ripeness of avocadoes [28-30]. These evaluations are conducted on avocados post-harvest [28, 30], or from

externally captured images where the features of interest could be identified by canopy health [29, 31, 32].

Although avocados may have a unique spectral reflectance relative to the canopy, the problem of occlusion remains. For multispectral imaging to be used as a yield estimation tool that does not suffer from the same effects of canopy and cluster obscuration that visual spectrum photography does, the spectral properties of avocados and the canopy must be such that the avocadoes can be 'seen' at a particular wavelength behind the canopy features. No research was found that suggests such conditions and no examples of multispectral imaging for avocado yield estimation were found that are independent of other variables such as canopy health.

Future research investigating the potential for hyperspectral imaging to overcome problems of occlusion may be worthwhile, however, given the current complexity and cost of this technology, its use is not justified in the context of this research over visual spectrum photography.

2.2.6 LiDAR

LiDAR (Light Detection and Ranging) is an active remote sensing technique widely used in various fields, including agriculture [33, 34]. It operates by emitting laser pulses and measuring the time it takes for the reflected light to return to the sensor, allowing for the precise measurement of distances and the creation of detailed 3D maps of the environment [35-38].

In agriculture, LiDAR has gained attention for its ability to capture the structure of crops, including tree canopies, with high accuracy and resolution [39, 40]. By emitting laser beams from ground-based sensors, LiDAR can generate a point cloud of millions of data points, representing the surface and objects within the scanned area. Some crops such as apples have fruit with unique reflective properties to its surrounding canopy, allowing this data to be processed to derive information about the size and number of fruit [33, 34].

The use of LiDAR as a growing aid for avocado crops is limited to canopy size and density evaluations [41, 42], and no sources were found that incorporated LiDAR as a

yield estimation tool. This is likely due to limitations posed by dense foliage, overlapping branches and fruit clusters, making it challenging for LiDAR to accurately distinguish and count individual fruits.

2.3 Avocado Distribution Simulation

By developing a reliable method to identify the position of avocados within a tree, the spatial distribution of avocados can be quantified. This knowledge can then be used in determining the optimal camera angles required for photographing the tree and subsequently processing and counting the avocados using an AI model. Conducting simulations based on the established distribution patterns will allow for an evaluation of the most effective camera positions to capture comprehensive images of the fruit within the tree. This, in turn, will facilitate the successful implementation of the visual spectrum photography AI model as a robust and accurate tool for estimating avocado yield.

2.3.1 Avocado Tree Specifications

2.3.1.1 Tree Size

Avocado trees can reach impressive heights and exhibit considerable variability in their size [43]. Mature avocado trees typically range in height from 5 to 12 meters, with some exceptional specimens growing even taller [43-45]. Allowing avocado trees to grow to their full height is usually not practical for commercial growing which is why they are usually pruned to more manageable and efficient heights between 2 to 3 meters tall [45-47].

The size of avocado trees can also vary widely in terms of trunk diameter and canopy spread. Trunk diameters can range from a few centimetres to over 30 centimetres [48], while canopy spreads can extend up to 6 meters or more [44]. The variability in avocado tree size can be influenced by various factors, including avocado cultivar, environmental conditions, pruning practices, and overall tree health [44].

Due to the large variability in the size of avocado trees, this research primarily focuses on Hass avocado trees sourced from King Avocado orchards in New Zealand. However, the aim is to develop data capture techniques that can be applied to avocado trees of different sizes. By considering the specific characteristics of Hass avocado trees, the goal

is to create a methodology that can be adapted for other avocado cultivars and orchards globally. This approach ensures wider applicability and the potential to gain insights into avocado tree dynamics and fruit distribution beyond the scope of this particular orchard.

2.3.1.2 Fruit Distribution

Limited research exists regarding the spatial distribution patterns of avocados during their growth on trees. The significant variability of orchard practices, including tree age, height, and pruning, is likely to affect fruit development and the potential influence of occlusion on image capture methodologies. Due to the lack of quantifiable data on the spatial distribution of avocados and an understanding of how it might be influenced by variability in growing practices, this remains an area requiring further research.

2.3.1.3 Canopy Features

Occlusion caused by surrounding canopy and branches is a prominent problem mentioned in the literature, especially with avocadoes [5, 16]. The dense foliage and overlapping branches of avocado trees can hide the fruit from direct view, leading to underestimation or misinterpretation of the actual yield. Growers have also reported that fruit is difficult to see when trying to manually estimate the yield on larger trees [9]. Efforts to improve visibility for photographing fruit by using blowers to move leaves have been explored but yielded limited success. [5].

The presence of neighbouring trees in an orchard can significantly impact the optimal viewing of a specific tree's canopy [19]. The canopies of nearby trees may appear in the background, leading to potential inaccuracies in yield estimation. This can occur when additional fruits, located behind the targeted tree, are inadvertently counted, thus affecting the accuracy of the estimation process.

One study attempted to address the problem of fruit occlusion in yield estimation by developing an AI model that could predict the relative amount of fruit that would be left uncounted due to occlusion, utilizing canopy features extracted from photographs [49]. Features such as the proportion of partially occluded fruit in an image were used to train the model to make these predictions with ground truth data provided by manually counting avocados on the trees. Ultimately, this predictive model was unable to increase

the effectiveness of yield estimation and no correlations were identified between parameters such as partial fruit occlusion and the overall number of hidden fruit [49].

It is evident that collecting distribution data and investigating canopy features and their influence on the visibility and occlusion of avocados from various positions will provide valuable insights for applying and optimizing AI fruit detection methods and enhance the accuracy of yield estimation in avocado trees.

2.3.2 Data Collection

There is no existing data on the distribution of avocados as they grow on the tree. A method for capturing the 3D positional data of avocados in trees is needed to understand how they are distributed. Such data can be used to create simulations to determine the best photographing practices for capturing images containing all the avocadoes on a tree in the fewest number of photographs possible with minimal overlapping and occlusion.

The following section outlines various technologies and methods that hold potential for use in 3D measuring applications within the canopy of avocado trees.

2.3.2.1 IMU

An Inertial measurement unit (IMU) is a device that is widely used in the fields of robotics, motion capture and navigation, that combines multiple sensors, such as accelerometers, gyroscopes, and sometimes magnetometers, to measure and estimate an object's orientation, velocity, and acceleration in three-dimensional space [50-53].

IMUs rely on integrating acceleration and angular rate measurements over time to estimate position and orientation [52, 53]. However, errors in the measurements, such as noise, biases, and drift, accumulate during the integration process, leading to position errors that grow over time [50, 51, 54, 55].

To overcome the effects of integration drift, IMUs are often used in conjunction with other technologies such as GPS that can correct for drift periodically and are only used to provide reliable positional data over short distances [50, 51, 55]. Without these compensation methods, an IMU will have inherent positional errors that compound over

time, making it difficult to capture accurate positional data over more than a few seconds at a time. The use of an IMU for the application of mapping the position of avocados in a tree would involve physically moving an IMU device from a known datum point to the position of each avocado in a tree. This process would take several minutes if not hours for a single tree, meaning positional error would result in very inaccurate results if drift could not be corrected for.

2.3.2.2 GPS

GPS (Global Positioning System) is a satellite-based navigation system that provides precise positioning, velocity, and timing information anywhere on Earth. It operates by using a network of satellites in space that transmit signals to GPS receivers on the ground. By receiving signals from multiple satellites and measuring the time it takes for the signals to reach the receiver, GPS can determine the receiver's location [56-59].
In terms of horizontal positioning, GPS can achieve accuracies ranging from a few meters to a few centimetres, depending on the technology used [57, 58]. With the use of differential GPS (DGPS) or Real-Time Kinematic (RTK) techniques, which involve corrections from additional reference stations or base stations, horizontal accuracies can be improved to centimetre-level accuracy [60-63].

The accuracy of vertical positioning GPS is generally lower compared to horizontal positioning, resulting in greater errors when it comes to its use for 3D positional recording [64, 65]. GPS signals can also be obstructed or weakened when objects, such as trees or buildings, block the line of sight between the satellites and the GPS receiver [56, 66]. Under a dense tree canopy, the GPS receiver may not receive signals from enough satellites, resulting in a weak or unreliable GPS fix. The use of GPS from within the canopy of an avocado tree would result in poor or non-existent GPS signal resulting in extremely inaccurate positional data even with high-end equipment, making it unfeasible for capturing the position of avocados on a tree.

2.3.2.3 LPS

Local Positioning Systems (LPS) is a positioning technology that provides location information in local or confined areas where GPS signals may be obstructed or weakened. LPS uses its own transceiver base stations in place of satellites which can help address

occlusion problems by providing accurate position measurements in areas where GPS signals are blocked or significantly degraded in conditions such as the inside of buildings [66-68], or within the canopy of a tree.

LPS technology ranges in cost and ability from hobbyist-style indoor setups to high-end setups with the ability to maintain centimetre precision positional measurements [66, 69]. Although LPS can solve the problems caused when GPS satellite signals cannot be received, an unobstructed direct line of sight is still required between the mobile transceiver and at least two stationary transceivers that are set up in the operating environment [66]. This means that to use LPS as a method of measuring the positions of avocados within a tree, these transceiver base stations would have to be set up within the canopy such that any point within the tree would be in direct line of sight with at least two of these transceivers. Overcoming the problem of canopy and branch occlusion would require a large number of base stations and a comprehensive setup process for every tree to be evaluated.

2.3.2.4 Mechanical Arm

A kinematic coordinate measuring arm, also known as a portable coordinate measuring machine (CMM), is a precision measurement device used to accurately capture three-dimensional (3D) coordinates of objects or surfaces [70, 71]. It consists of a mechanical arm with multiple articulated joints, allowing it to reach different positions and orientations [71]. These arms are commonly used in industries such as manufacturing, aerospace, and automotive, where precise dimensional measurements are required for quality control, inspection, reverse engineering, or assembly verification. They can be portable and flexible, allowing measurements to be taken directly on workshop floors, in the field, or in various environments.

The accuracy of a kinematic device relies on precision-manufactured linkages and joints that are consistent with the kinematic model used to calculate the position of their end effector [70, 71]. A CMM with multiple linkages and joints introduces compounded error due to mechanical play, encoder and kinematic modelling error. For use within an avocado tree canopy, a kinematic arm with several linkages combined to create sufficient

reach and manoeuvrability to navigate around branches would be complicated to build, deploy, and manipulate.

2.3.2.5 Laser Distance and Orientation Kinematics

Laser distance measuring devices are handheld devices that use laser technology to accurately measure distances between two points. They provide a quick and convenient way to obtain precise measurements in a variety of applications and are commonly used in construction, architecture, interior design, real estate, landscaping, and various other fields where accurate distance measurements are needed. Laser distance measures work by emitting a laser beam towards a target surface and measuring the time it takes for the laser beam to reflect back to the device [72, 73]. By knowing the speed of light, the device can calculate the distance based on the time-of-flight principle [72, 73].

A combination of laser distance measures and pitch and yaw measuring rotary encoders enables precise 3D position measurements of an object relative to the device's mounting point. This is achieved by employing forward kinematics principles, similar to those used in kinematic arms, but instead of relying on a physical arm with a known length, the laser distance measure provides the necessary readings. Past research has investigated the use of these techniques to measure 3D coordinates of large storage yards for quickly and efficiently evaluating their volume, yielding accurate results [74]. Laser measuring devices have also been used within the agricultural industry to provide information on crop parameters such as height, canopy volume and density [75].

Small, low-cost, and portable laser measures can be easily obtained from hardware stores, providing accuracy within a few millimetres [73, 76]. However, it is important to note that as the measurement distance increases, the accuracy of these devices diminishes rapidly [77]. Therefore, it is crucial to consider this factor when selecting a laser measure for a specific application and when verifying its accuracy.

One limitation associated with utilizing laser measures and rotary encoders for 3D mapping is their reliance on an unobstructed line of sight to the target object. However, this challenge can be addressed by establishing multiple points of known coordinates around the tree's base, enabling the device to be mounted strategically. This approach not

only mitigates occlusion issues caused by the canopy and branches but also offers valuable insights into the visibility of different avocados and potential patterns of occlusion. By systematically setting up these reference points, a more comprehensive understanding of the distribution and visibility of avocados within the tree can be gained, enhancing the accuracy and reliability of the mapping process.

2.3.2.6 LiDAR

LiDAR holds the potential to measure the positional data of objects, including avocados within a tree's canopy. However, its applicability is hindered by its indiscriminate capture of 3D data points from all elements in its field of view, encompassing branches and leaves. Consequently, an essential step involves categorizing these data points to distinguish avocado surfaces from other elements. With such classification, the LiDAR could be used independently from image processing AI to provide a direct yield estimation of a tree. Yet, the challenge lies in developing a model to identify avocado points within the LiDAR point cloud, which would necessitate significant annotation and training.

This avenue for research stands as a distinct opportunity beyond the scope of the visual spectrum photography AI model and remains unsuitable for gathering data concerning the spatial distribution of avocados to optimize image capture processes and implement the existing avocado counting AI model effectively.

2.4 Orchard Block Sampling Methods

Orchard block sampling methods are used for yield estimation in orchards. They involve systematically sampling trees within a designated block to obtain representative data while reducing the total number of trees that need to be evaluated. These methods provide more cost-effective insights into average yields and variability and allow for better allocation of resources and monitoring of the impact of management practices on productivity.

2.4.1 Block Specifications

An avocado orchard block refers to a specific area within an avocado orchard that is dedicated to the cultivation and growth of avocado trees. It typically consists of a

contiguous grouping or arrangement of avocado trees, often planted in rows or a specific pattern, within a defined section of the orchard [78, 79]. Orchard blocks are designed to facilitate efficient management, irrigation, pest control, and harvesting operations [9, 78, 79].

Growing guides for avocados have suggested that an optimal rectangular block size for avocado orchards is approximately 80 by 120 meters [79]. However, it is important to consider that the suitability of this block size may vary depending on the topography and contours of the land where the orchard is established.

The plant density of an avocado block is determined by the spacing between its trees. Across different growers and techniques, this tree spacing can vary considerably between 2.5 and 12 meters [80-82]. Traditional spacing is considered to be 7 by 7 meters [81, 82], with high-density plantations maintaining much smaller spacings of 2.5 by 5 meters [80, 82], and some experimental plantations with 2.5 by 2.5-meter spacing [81].

2.4.2 Visual Selection

Several sources including an orchard management guide [79], a case study of six Australian industry growers [9], and crop estimation recommendations by the California Avocado Commission [83], highlight that it is not feasible to manually count every tree in an orchard. Instead, avocado orchards that can contain tens of thousands of trees, utilise various methods of block sampling. This can involve selecting trees that growers feel are representative of the overall block, selecting a range of low to medium and high-yield individual trees for counting, and then extrapolating to estimate the total yield of a block of trees. These representative areas within a block are identified by an experienced grower who will walk through the blocks multiple times to gauge the health and yield strength of the trees [9]. It is also common practice to utilise historical records to compare current and previous estimations to highlight inconsistencies caused by manual counting and sampling errors [79].

One particular grower reported that from the trees he selected to sample he would only count fruit from an estimated quarter of the tree canopy [9]. This reduced the time required to manually count each tree significantly, and to minimise sampling error a

different quarter from each tree would be counted each time. There could be potential for this technique to also be applied with image capturing techniques when applying an AI counting model.

These block sampling methods greatly reduce the number of trees that need to be counted, however, they also introduce compounding error to the already inaccurate manual counting estimations. By utilising block sampling methods, the sampling time and employee fatigue levels can be reduced, but human error is not eliminated, and overall yield estimation accuracy remains a consequential problem that needs to be addressed.

2.4.3 Satellite

By analysing satellite imagery, researchers and growers can obtain information on tree health, canopy density, and other relevant factors. This data assists in the selection of representative sample trees for yield estimation, improving the accuracy and efficiency of the sampling process compared to traditional manual methods.

Research by the University of New England led to the development of a grower-friendly app that utilized satellite imagery to compare canopy health within avocado orchard blocks to select as few as 20 trees to sample with an overall claimed yield accuracy of 93% [23]. This service still relies on growers providing manual estimates of the selected sample trees, however with the greatly reduced sample numbers, it becomes more practical to diligently count individual trees. Relying solely on satellite imagery does not offer standalone quantitative counts of avocado yield that are unaffected by other variables, however, it can be used as a valuable tool for observing trends and patterns in canopy health and yield, enabling growers to make informed evaluations regarding block sampling strategies [4, 21-23].

The optimization of individual tree-counting methods using AI and the integration of satellite-aided block sampling techniques show promise in improving efficiency and reducing sampling requirements in agricultural practices. Achieving an optimal balance between these approaches can potentially lead to improved accuracy and cost-effectiveness in yield estimation. Assessing the trade-offs between the precision of individual tree counting and the representative sampling provided by satellite-aided block

sampling methods will contribute to refining these techniques for more efficient and reliable yield estimation in the future.

2.5 Conclusion of Literature Review

In conclusion, this literature review provides a comprehensive overview of agricultural yield estimation, focusing on the integration of machine learning models, satellite imagery, and block sampling techniques for the Hass avocado variety. The review emphasizes the inefficiency of current manual yield estimation techniques. It highlights the potential of AI-based approaches to enhance efficiency and accuracy in yield estimation while acknowledging the limitations of satellite imagery as a standalone tool for independent quantitative counts.

Furthermore, the review identifies key research gaps in understanding the spatial distribution patterns of avocados within trees and the influence of orchard practices on fruit development and occlusion. There is a need for improved methods to capture images within the avocado tree canopy, considering factors such as occlusion, overlap, and fruit distribution. This highlights the importance of capturing distribution data and investigating canopy features to enhance the effectiveness of AI fruit detection methods.

By addressing these gaps, future research can contribute to the development of more efficient and reliable yield estimation methods in avocado orchards. This includes refining image capture techniques, investigating the on-tree spatial distribution of avocados, and evaluating optimal sampling methods. The findings of this literature review provide a foundation for further investigation and highlight potential areas for future research and commercial solutions that will ultimately benefit avocado growers by streamlining the yield estimation process.

CHAPTER 3
METHODOLOGY

Informed by the insights gathered from the literature review, it is evident that a comprehensive understanding of avocado spatial distribution is necessary to optimise an image-capturing procedure that allows for the implementation of the counting AI previously developed. This necessitates a device capable of measuring the 3D position of avocados as they mature on the tree, and through the literature search, led to the identification of laser measurement technology, used in conjunction with rotary encoders, as the most viable means to fulfil the research objectives.

This chapter outlines the technological considerations and development process of a Coordinate Mapping Device (CMD) designed for capturing the 3D positions of avocados on trees. It covers the integration of complementary technologies alongside the CMD, along with the validation testing procedures and results aimed at ensuring the device and the associated data collection methodology meet the required accuracy standards. The chapter concludes by discussing the methodology applied to deploy the developed technology in an orchard environment for data collection.

3.1 Design and Manufacturing

This section covers the considerations and processes involved in designing and manufacturing a device capable of collecting the necessary positional data for this research.

3.1.1 Design Considerations

In considering the design approach, precision requirements and the choice of technology play pivotal roles. Insights from the literature review, alongside information on tree specifications provided by King Avocado, guide decisions on technology selection and design capability, ensuring usability as a data collection device within the project budget and timeline.

3.1.1.1 Tree Size

To tailor the CMD to the diverse range of avocado trees grown in commercial orchards, insights were drawn from collaboration with Claudia Hermosillo at King Avocado, based on provided photos and measurements. Critical parameters informing the design include an optimal harvesting tree height set at 3.5 m for efficiency, the variability in tree size spanning 2 m to 6 m in height, trunk diameters typically observed to be within 0.3 m, and King Avocado's practice of trimming lower branches within 600 mm of the ground. These specifications were integral in shaping the design of the CMD, ensuring its adaptability to the diverse dimensions of avocado trees within the orchard.

3.1.1.2 Positional Precision

Provided that the ideal height of avocado trees for commercial growing and harvesting was within 3.5 m, the CMD should be capable of measuring the position of avocados within this distance range with appropriate accuracy. Factors such as wind and variability in the measured point on each avocado will introduce errors to measurements regardless of the precision of the device itself. The compounding error of these factors led to a target of maintaining plus or minus one-centimetre precision up to a distance of 4 meters. A device capable of this precision would exceed the requirements of the positional data given its intended use for understanding the spatial distribution of avocados. It would also still be capable of measuring avocados on trees from the upper height range of 6 m with a precision smaller than the width of an average avocado (6 to 8 cm, based on measurements taken at the King Avocado orchard).

3.1.1.3 Canopy Conditions

Measuring avocados within the canopy of a tree poses challenges related to spatial constraints. Therefore, any technology employed in the data capture procedure must be compact to fit beneath the branches and foliage of the trees. Another consideration is the brightness of the laser used for measurement. While the canopy may offer some shelter from sunlight, certain areas within the tree will inevitably be exposed. To address this concern, the laser should be sufficiently bright to remain visible and functional even in direct sunlight.

3.1.1.4 Origin

Measurements taken of avocados in a tree need to have a common origin point so that the distribution of all avocados relative to each other and the position of the tree can be evaluated. This origin needs to be standardised throughout the measurement of all trees and will be referenced to the global position of the tree (e.g. oriented north and level).

3.1.2 Coordinate Mapping Device

The coordinate mapping device (CMD) developed to obtain the relative coordinates of avocado locations on a tree consists of two freely movable rotary axes resulting in gimbal-like movement of the laser measuring device mounted on top, enabling the user to direct the laser by hand to point at a target. The rotational position of the two axes is then read by two separate rotary encoders, which allows the orientation of the laser to be known. Together with the distance reading of the laser measuring device, this provides sufficient information to calculate the positional coordinates of the laser point on the target object relative to the origin of the device.

3.1.2.1 Laser Distance Measure

Laser measuring devices utilize the time of flight of light. This involves measuring the time it takes for light to be emitted from the device, reflect off an object, and then be received again by the device. This information is then used to calculate the distance the light travelled, providing the distance of the measured object from the device.

3.1.2.1.1 Selection

The selection of the device for providing linear distance measurements was based on considerations of availability, cost, and claimed accuracy. A cost-effective DeWalt DW055PL laser measuring device, readily available at a local hardware store, was chosen. This compact device featured a built-in rechargeable battery and was advertised to have an accuracy of ±6 mm at 10 m, with a maximum range of 16 meters.

The compact size of the DeWalt device, in comparison to other available options, made it suitable for the intended application. Additionally, the USB-rechargeable battery allowed for convenient charging during field use in orchards if needed. Importantly, the

laser dot produced by the device was sufficiently bright to be visible outdoors even in sunny conditions.

3.1.2.1.2 Laser Accuracy Validation

To assess the accuracy of the device, a testing procedure was conducted using a tape measure, measuring distances up to 8 m. The process involved measuring the distance from a target object (a white cardboard box) in half-meter increments with the tape measure and subsequently re-measuring the same distance with the laser device. Testing was carried out both indoors and outdoors on a cloudless day under direct sunlight, with three readings taken at each position to evaluate any fluctuation in readings by the laser measure.

Testing results (see Appendix B) demonstrated that during indoor conditions the laser device was, at worst, accurate within a range of +1 to +3 mm over a distance of 8 m, with no variation observed between repeated measurements, which comfortably meets the required centimetre precision over a distance of 4 m.

An identical test conducted outdoors under direct sunlight revealed a notable reduction in accuracy and an increase in variation between repeated measurements at the same distances. Reading fluctuations are likely attributed to ambient light interfering with the device's receiver, resulting in the accuracy of the device worsening to a range of -58 to +67 mm over an 8 m distance at its worst.

It should be noted that the results depict the worst-case measurement scenario, involving bright sunlight directly on a gloss white target surface – conditions that are unlikely to be replicated under the shaded canopy of an avocado tree. Additionally, the more significant error was evident only beyond 4 m, and within this distance, the error remained within ±17 mm. Consequently, based on these tests, the device was deemed suitable for meeting the specified requirements and was selected for use as part of the 3D coordinate mapping device.

3.1.2.2 Rotary Encoders

Two rotary encoders allow the orientation of the laser device to be known and thus the 3D position of the laser dot to be measured. The precision of the measured orientation depends on the resolution of the rotary encoders used. Therefore, the selection of suitable encoders for use in the device is essential.

3.1.2.2.1 Resolution

To maintain centimetre precision over a distance of four meters, the necessary step resolution of the encoders can be calculated as follows:

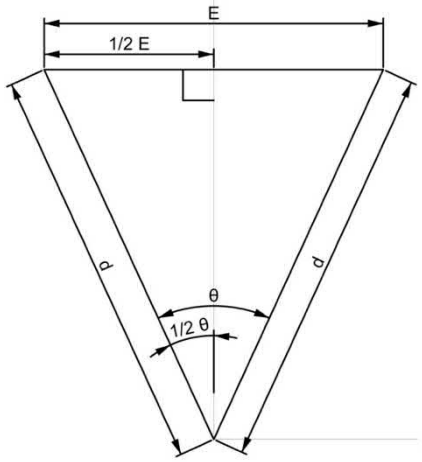


Figure 3.1 – Encoder angle to linear error calculation

$$0.5\theta = \tan^{-1}\frac{0.5E}{d}$$

$$\theta = 2\tan^{-1}\frac{E}{2d}$$

$$\theta \leq 2\tan^{-1}\frac{1cm}{2 \times 400cm}$$

$$\theta \leq 0.143\ldots°$$

$$PPR \geq \frac{360°}{\theta}$$

$$PPR \geq 2513$$

These calculations provide the required rotary encoder step resolution of less than 0.143 degrees between readings or pulse per revolution count (PPR) of at least 2513. For comparison, to achieve this resolution with a physical analogue scale with indicator lines separated by one millimetre (visually observable separation), the scale would have to be over 0.8m in diameter. This is impractical for use within the canopy of avocado trees; hence, a high-resolution rotary encoder must be used instead.

Commercially available digital encoders are offered at various bit resolutions suitable for various applications. The higher the bit resolution, the greater the number of unique digital rotary positions or PPR the encoder is capable of outputting as shown below:

$$10\ bit = 1024\ PPR$$
$$12\ bit = 4096\ PPR$$
$$13\ bit = 8192\ PPR$$
$$14\ bit = 16384\ PPR$$

Considering the previously calculated PPR of 2513, it is determined that a 12-bit rotary encoder with a PPR of 4096 is sufficient. This choice surpasses the required angular resolution by over one and a half times, making it more suitable compared to the next readily available encoders with a lower bit resolution of 10, which would fall short of meeting the precision requirements.

3.1.2.2.2 AMT Series Absolute Encoder

An AMT222A-V rotary encoder, equipped with a radial-mounted cable connector, and a 12-bit single-turn absolute resolution, operates on a 5V power supply and communicates via a full-duplex Serial Peripheral Interface (SPI) [84]. This configuration is ideal for measuring absolute positions around a single rotation, providing the orientation of the coordinate mapping device. The 5V power supply and SPI communication make it conducive to seamless integration with common, low-cost microcontrollers like Raspberry Pi or Arduino.

As per the product datasheet, the accuracy of these rotary encoders is specified as 0.2 degrees [85]. It is important to note that encoder accuracy differs from resolution; while the encoder has a 12-bit resolution, reporting 4096 pulses over one rotation or every 0.088 degrees, there may be an error between the measured and actual angles. This angular error determines the accuracy of the encoder. With an accuracy of 0.2 degrees over a distance of four meters, the linear error can be calculated as follows:

$$0.5\theta = \tan^{-1}\frac{0.5E}{d}$$
$$E = 2d\ \tan 0.5\theta$$
$$E = 2 \times 400\ \tan(0.5 \times\ 0.2)$$
$$E = 1.396 \ldots cm$$

This error falls within the target error range of plus or minus one centimetre over four meters. Given the absence of a similar rotary encoder with higher resolution, within the budget constraints of this master's project, the AMT222A-V encoder was deemed suitable, and two units were purchased for use on the coordinate mapping device.

3.1.2.3 Microcontroller

The use of a microcontroller is required to read and store digital readings from the two rotary encoders within the CMD. It can also be used to temporally store the distance measurement from the laser device to be saved for later calculation of the measured coordinate point in conjunction with the rotary angles.

Ideally, the CMD should be a stand-alone device capable of saving measured point data without the need to be connected to a computer or power supply. However, the impracticality arises from the laser measure device used, lacking a straightforward way to output distance readings other than its LCD display. While intercepting the signal within the device as it sends the laser distance value to its display is theoretically possible, no simple method to achieve this was found, rendering it a needlessly complicated and time-consuming task. Alternatively, the distance reading from the laser measure would have to be manually input either into the CMD or an external computer by the user. To maintain synchronization between laser measurements and rotary encoder readings, a microcontroller can record the rotary positions of the CMD when commanded by an external computer (laptop), where the user manually inputs the distance reading on the laser measure display. Since wireless technologies like Bluetooth would introduce unnecessary complexities to a device that already necessitates the user's close proximity for manipulation within the canopy of a tree, a wireless setup was not perused.

For these reasons, a simple Arduino Uno microcontroller was chosen to connect to the rotary encoders and send their positional data to a laptop via serial communication. This setup was enclosed in a 3D-printed case and connected to the two rotary encoders with two ribbon cables, as depicted in Figure 3.2.

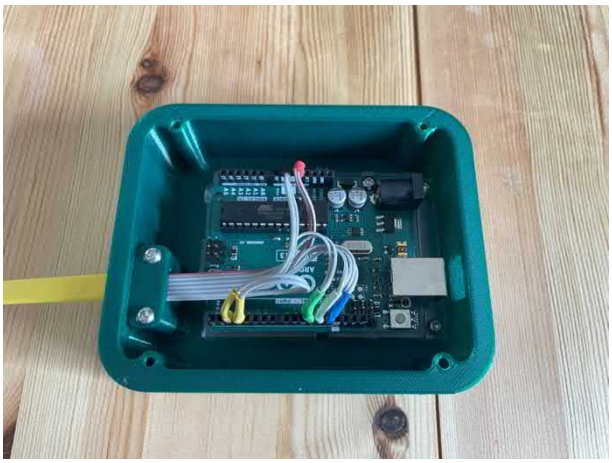


Figure 3.2 – Arduino Uno microcontroller connected to output leads from rotary encoders

3.1.2.4 Mechanical Design

The functional design of the CMD is relatively straightforward, necessitating only two movable axes connected to rotary encoders. However, the challenge in crafting a device capable of sustaining measurement accuracy lies in achieving tight tolerances to minimize backlash and play in the moving axes.

3.1.2.4.1 CAD

A CAD model (Figure 3.3) was created using a 3D modelling software by Dassault Systèmes called SolidWorks. The design was tailored around the selected rotary encoders and laser measuring device, emphasizing compactness for ease of use beneath a tree.

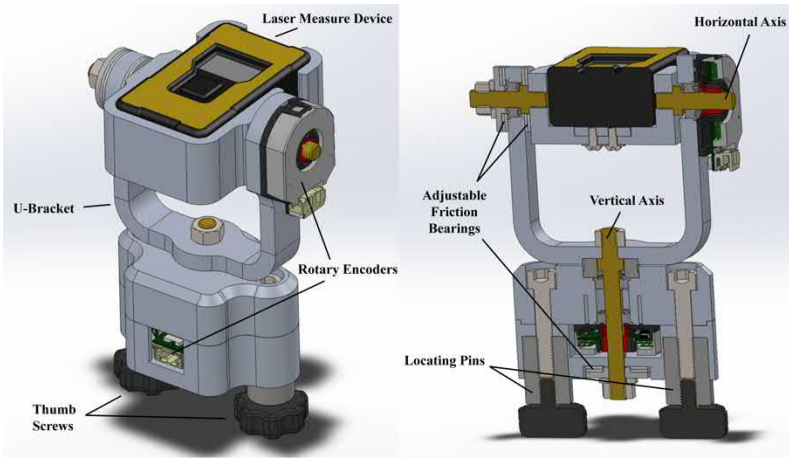


Figure 3.3 – CAD design (a) with cross-section (b)

Due to the accuracy requirements of the device, the design was intentionally over-engineered to ensure rigidity. This involved incorporating oversized ball bearings, resulting in a robust device with precise rotational movements.

The device's body comprises two aluminium parts that can be securely bolted together, encapsulating the main axis, bearings, and rotary encoder. A U-bracket, mounted on top of the main axle, supports the bearings and rotary encoder for the horizontal axis. The U-bracket design includes ample clearance for the laser measuring device and its mount to achieve a full 360 degrees of unrestricted rotation. This, combined with the vertical axis of rotation, enables the device to be oriented to face any direction within 3D space.

3.1.2.4.2 Rotary Axis

The cross-section view in Figure 3.3(b) illustrates the shaft and bearing arrangement for both rotational axes, which were manually machined to meet ISO-h6 tolerance, ensuring a close-fitting clearance fit with the ball bearings. Each axis incorporates two flanged bearings to facilitate smooth rotational movement while minimizing radial play. Additionally, an oversized thrust bearing has been integrated into the main Z-axis to eliminate axial shaft movement and minimize any play in the U-bracket.

Both axes feature multiple low-friction PTFE washers, tightened against each other with a lock nut. This setup provides a controlled and adjustable level of rotating friction between the PTFE washers, restricting the free movement of each axle. This is crucial to enable the axles to rotate freely enough for manual adjustment of the laser's position while ensuring sufficient friction to prevent unintended movement or sagging under the device's weight or resistance from the cables connected to the rotary encoders.

3.1.2.4.3 Mounting Pins

Attached to the underside of the device are two precision-machined ISO-H7 locating pins. These pins are securely fastened to the device through dedicated locating pockets, enabling the CMD to be mounted to a mounting frame in a repeatable and accurate manner. The inclusion of two thumb screws facilitates the mounting of the CMD to a frame without the requirement for additional tools, ensuring ease of use in an orchard setting.

3.1.2.4.4 Manufacturing

Utilizing the precise CNC machines available at the Massey University engineering workshop, the device's body components were machined from solid aluminium. This enabled the production of parts with tight tolerances, and minimal play or deflection, ensuring the structural rigidity and accurate measurements by the device.

The components were machined using a Haas VF-2SS CNC milling machine, equipped with features like tool offset compensations and live thermal compensation. This ensured the fabrication of parts with a high degree of dimensional accuracy, maintained within 0.01mm.

In Figure 3.4(a), the fully assembled device is presented, featuring rotary encoders, the laser measure device, connecting ribbon cables, and thumb screws for convenient mounting. An aluminium mounting block was also machined, offering the flexibility to attach the CMD to any standard camera tripod if required.

(a) (b)

Figure 3.4 – Complete CNC machined assembly of CMD (a), with protective case (b)

3.1.2.5 Protective Case

The CMD is designed for outdoor use in orchards, where exposure to water and dust is a potential risk to its sensitive electronic components. To safeguard the device from these

conditions during transportation and use, a water-resistant carry case was made using 3D printing, as shown in Figure 3.4(b).

3.1.2.6 Forward Kinematics

By applying forward kinematics, equations can be derived to express the 3D coordinates of the laser dot based on the distance and rotary angles of the CMD. The kinematic diagram (Figure 3.5) illustrates the physical geometry of the CMD, incorporating revolute joints measured by rotary encoders for pitch and yaw. Additionally, unchanging link offsets represent the device's physical dimensions, along with a variable link offset for the laser measure's distance reading. The coordinate origin of the CMD is situated at the centre of its vertical pivot axis at the base of the mounting point.

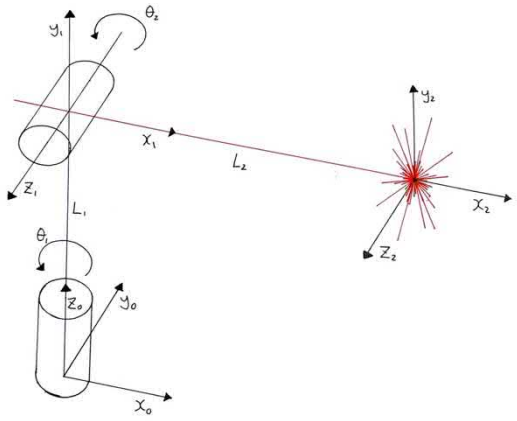


Figure 3.5 – Kinematic diagram

Using the Denavit-Hartenberg (DH) convention the following equations were derived from the kinematic diagram, where d_1 is the measured distance reading from the laser measure, θ_1 is the angle measured by the roll rotary encoder and θ_2 is the angle measured by the pitch rotary encoder:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} (d_1 - 0.02965)\cos\theta_1 \cos\theta_2 \\ (d_1 - 0.02965)\sin\theta_1 \cos\theta_2 \\ (d_1 - 0.02965)\sin\theta_2 + 0.09472 \end{bmatrix}$$

Detailed derivations for these equations are available in Appendix A. Based on these equations, a simple Python function was written that takes the two rotary angles and the laser measure distance reading and returns X, Y, and Z coordinates of the measured point

relative to the device origin point. This functionality enables the CMD to be directed at any 3D point within the geometric line-of-sight constraints of the device, facilitating the storage of distances to that same point in the X, Y, and Z directions.

3.1.2.7 Setting Device Zero

Zeroing the axis angles of the CMD is essential to ensure that the rotary encoders read zero when the device is pointing straight ahead, as defined by the kinematic diagram.

The pitch of the device was zeroed at a distance of 6.5 m on a building floor by measuring the height of the laser emitter at the device and ensuring the laser dot was positioned at the same height on the opposing wall. The yaw was set based on the geometry of the device by positioning it squarely against a wall and ensuring the laser remained parallel across the length of a perpendicular wall.

Without a dimensionally and geometrically known structure more accurate than the walls of a building, it was not possible to quantify the accuracy of this zeroing procedure. However, any error will only impact the offset of measured coordinates relative to the device's origin. This will result in a slight shift of all measured avocados relative to the trunk but will have a minimal effect on the usefulness of the collected data.

3.1.3 Mounting Frame

The CMD facilitates the definition of any 3D point or avocado relative to its origin. To convert these measurements into meaningful data, a method is necessary to define coordinates relative to an unchanging origin point related to the tree positions. This unchanging origin is set as the centre of the tree trunk, 300 mm above ground level (providing ground clearance for the CMD). To achieve this, a mounting frame was developed for the CMD to have multiple known positions around the base of an avocado tree for mounting and applying correct offsets to the relative coordinates.

3.1.3.1 Design

The design of the frame was informed by consultations with Claudia Hermosilla from King Avocado, who provided valuable data on the orchard's tree dimensions and

characteristics. The consideration included a range of tree sizes, from smaller to larger extremes, with insights into the potential trunk diameter and the presence of low branches.

The largest trees in the orchard were estimated to have trunk diameters up to 300 mm, and some trees exhibited low branches emanating from the trunk. In response to these considerations, a circular steel ring was designed in CAD with an inner diameter of 600 mm, as depicted in Figure 3.6(a). This dimension was chosen to accommodate the majority of avocado trees, providing additional space for potential branches, while not being overly cumbersome to handle under a tree.

Additionally, the frame was tailored to fit within ground clearance beneath the tree, aligning with King Avocado's pruning practice of trimming branches below approximately 600 mm from the ground. This configuration allows the CMD to be mounted as low as 100 mm above the ground, ensuring flexibility and adaptability in various orchard settings.

(a) (b) (c)

Figure 3.6 – Mounting frame CAD model (a). Completed frame with adjustable legs (b). Multiple feet options for various ground conditions (c)

The frame was designed for maximum rigidity to prevent warping or bending that could introduce measurement errors. Constructed from 8 mm thick plasma-cut steel plates, the frame exhibited sufficient rigidity without the need for additional steel bracing, as initially considered during the design phase.

3.1.3.2 Adjustable legs

The frame was equipped with six adjustable legs to conform to the ground contours under a tree. Made from aluminium tubing in two varying lengths (Figure 3.6(b)), the legs could be interchanged for greater height adjustability. Despite having six mounting sleeves, only three legs were utilized at a time to facilitate easy levelling of the device and ensure even weight distribution. The sleeves, featuring bicycle seat clamps, offered infinite height adjustability within the length constraints of the legs. The result is a frame capable of adapting to roots or uneven terrain that can be repeatedly positioned 300 mm from ground level around different trees. Figure 3.6(c) displays two 3D-printed foot designs that could be swapped based on ground conditions, with larger feet suitable for soft or wet ground to prevent sinking and to maintain stable measured coordinates.

3.1.3.3 Mounting Points

Eight mounting positions for the CMD were integrated into the frame by drilling and reaming holes on a milling machine with a digital readout providing 0.01 mm dimensional resolution. These reamed holes ensured a precise and repeatable fit with the CMD's locating pins, subsequently secured in place by hand-tightened thumb screws (see Figure 3.7).

Figure 3.7 – CMD mounted on the frame

Each mounting point, spaced at 45-degree intervals around the frame, offers numerous positions for CMD attachment, reducing occlusion issues arising from branches, leaves, or adjacent avocados during measurements. Mounting points are numbered 1 through 8, on the frame, allowing the user to input the correct mounting position into the GUI. This

numbering system allows the correct geometric offset to be saved along with coordinate measurements. The code subsequently translates the CMD coordinate origin to the central position of the mounting frame.

3.1.3.4 Locating Bolts

The frame consists of two detachable halves, facilitating mounting around a tree trunk. For consistent locating points throughout the measurement of various trees, both halves must consistently mate together in the same position. Figure 3.8(a) shows the use of precision shoulder bolts and machined mating surfaces, ensuring a precise and repeatable connection between the two halves.

(a) (b) (c)

Figure 3.8 – Frame mounting surface allowing halves to be bolted together (a). Compass for global alignment of frame yaw (b). Spirit level for global alignment of vertical axis of frame (c)

Flatness across the top surface of the frame was attained by aligning all components on a flat stone before welding the steel pieces together. Although ideal machining processes such as precision surface grinding were not employed due to machinery limitations, subsequent validation testing confirmed that the frame's error was within an acceptable limit.

3.1.3.5 Global Origin

A repeatable method for setting up the frame around each tree relative to a global coordinate system is essential to ensure the uniform orientation of measured avocados across different trees in an orchard. This global alignment offers insights into avocado growth relative to external factors, such as the sun's position, and facilitates accurate plotting of measurements from various trees.

The chosen global coordinate system aligns the X-axis with magnetic north and the Z-axis at a normal to the horizontal plane. Alignment of the frame with this coordinate system is achieved using a compass and spirit levels.

3.1.3.5.1 North

The use of a navigation compass, combined with a 3D-printed mounting bracket, enabled the fixation of the compass to a specific point on the frame as shown in Figure 3.8(b). This arrangement could then be aligned with magnetic north when the frame was installed under a tree.

Consideration was given to magnetic interference during the design of this setup. Given the substantial amount of steel comprising the frame, placing the compass too close could lead to magnetic interference overpowering the Earth's magnetic field, resulting in inaccurate readings. To counteract this, the compass was positioned at a location devoid of magnetic interference, allowing it to point to magnetic north. The frame was then gradually moved closer to the compass until any observable movement of the compass needle occurred. This movement was noted to happen within a range of 150 mm, and a mounting bracket was designed to keep the compass beyond this distance from the frame.

3.1.3.5.2 Level

Two spirit levels (Figure 3.8(c)) were attached to each half of the frame, offering a level indication on the most accessible side for the user. The frame's height could be adjusted by manipulating each leg, allowing it to be positioned under the tree until achieving a level orientation facing north. This process, replicable for any tree in an orchard, guaranteed the accurate orientation of the coordinate system used to measure avocados on trees.

3.1.4 Tripod Measurement System

The developed frame for mounting the CMD under the tree offers multiple points around the tree's base for avocado measurements. Despite this system, potential occlusion from branches, leaves, and other avocados makes it uncertain that every avocado on all trees

will have a direct line of sight to the CMD laser. To address this, having a backup method to measure problematic avocados from an alternate origin point is prudent.

A practical solution for mounting the CMD in various positions to measure challenging points is to use a camera tripod. With an adjustable tripod, the CMD can be set up outside the tree at different heights depending on the location of the target avocado and any obstacles in the way. However, using a tripod introduces challenges in defining local coordinate measurements of the device with the global coordinate frame of the tree being measured. This challenge is particularly evident when measurements from different tripod setups need to be remapped to the same coordinate system.

3.1.4.1 Datum Points

To address this challenge, three fixed datum points were strategically positioned around the tree. These points remain constant throughout the entire measurement process, regardless of the number of different measurement positions. At the start of a new measurement position, each datum point was measured and saved by the CMD. In subsequent measurement positions, the same three datum points were measured, and the offset in the measured coordinates of the datum points across different positions was used to remap both the datum points and any measured avocados. The specific process and programming techniques employed for remapping these coordinate frames are further explained in the validation section.

Utilizing spheres as datum points offers the advantage of being measurable from any location. By adding a distance offset equivalent to the radius of the sphere to the laser measurement, the centre of the datum points can be accurately recorded. The datum points designed for this purpose were constructed by brazing stainless steel ornamental spheres onto steel pegs, which can be inserted into the ground around the tree being measured. Datum points were sandblasted to achieve a matte finish (see Figure 3.9(a)) to reduce their reflectivity and mitigate potential light interference that could introduce errors to the laser measurements.

(a) (b)

Figure 3.9 – Datum points sandblasted to minimise surface reflectivity (a). Datum points painted unique colours for ordering purposes when staked around a tree (b)

To facilitate identification during the remapping process, each datum point was painted a distinct colour, as depicted in Figure 3.9(b). This differentiation becomes crucial when conducting measurements for remapping, ensuring that individual datum points are recognized.

The accuracy of the remapping process relies on the repeatability of measured datum points. It is crucial to precisely aim the CMD at the centre of each datum point and apply the correct radius offset for each measurement of the datum points. Placing the datum points as far apart as reasonably possible around the tree helps mitigate aiming errors and enhances the accuracy of the remapping process.

3.2 Programming

The programming aspect of this research project, involving the coordinate mapping device, graphical user interface, and data processing scripts, constitutes a substantial part of this work. Python, implemented in the PyCharm IDE, was used for all programming tasks, except for the Arduino Uno Microcontroller code, which was written in C++ using the Arduino IDE.

While not directly influencing the practical application of the research results, this section is necessary for comprehending the data collection process and the transformation of data into usable 3D coordinates, as well as understanding the potential errors within the final datasets.

3.2.1 Microcontroller Code

The CMD employs an Arduino Uno to facilitate the communication between the two 12-bit ATM series absolute encoders and a portable laptop for data storage and processing. The Arduino is responsible for reading and writing data from the encoders, providing regulated 5V power to them, and is powered through the USB connection with the laptop. Notably, the Arduino does not store any data internally; its role is to relay the SPI data from the encoders to the computer over USB.

3.2.1.1 Serial Communication

The code to operate the Arduino microcontroller was obtained freely from CUI Devices on their ATM series encoder product page [85], and it was adapted to manage the control and reading of two rotary encoders (full code included in Appendix E). This code utilizes the SPI protocol for controlling and reading data from the encoders and Arduino UART to transmit that data to the connected PC. During initial testing, data was continuously read through the Arduino serial monitor. Later, it was read on request through a program scripted in Python using serial communication.

Key parameters in the code include a baud rate of 115200, a clock rate of 500 kHz, and a read delay of 500 milliseconds. This setup allows for a new reading from both encoders to be transmitted via Arduino UART twice every second. The chosen read rate aligns with the nature of the coordinate measuring process, where individual measurements are taken of avocados, allowing sufficient time for aiming the device between each measurement.

3.2.1.2 Encoder Zeroing

Establishing the zero or home position of the rotary encoders is a one-time requirement, defining the orientation of the coordinate system of the CMD. Once set, it remains constant throughout the entire data-capturing process and will likely remain the same for the lifespan of the device. A zeroing functionality was incorporated into the Arduino microcontroller code, enabling it to send a zeroing command to both encoders upon Arduino power-up. Once this zeroing process is accomplished with the device appropriately positioned, the zeroing functionality was disabled to prevent the inadvertent resetting of its zero position every time the CMD was connected to USB during use.

3.2.2 GUI

A Graphical User Interface (GUI) was developed in Python using Tkinter, a built-in Python library for creating custom GUIs. This GUI was crucial for the data collection process in this research, facilitating the quick and repetitive saving of positional information for avocados in an orchard, along with tree, row, and block details. Operable on a portable laptop connected to the CMD via USB, the GUI allows data to be read from the CMD on command.

Figure 3.10 illustrates the GUI used for all validation testing, data reading, and saving. It is segmented into three control sections for text file manipulation, serial port settings, and displaying/saving avocado or datum point measurements.

Figure 3.10 – Graphical User Interface for data collection from CMD

The code for the GUI is implemented as the main Python file that is executed from the PyCharm IDE. Subsequent functions for data processing or file handling are invoked from this main file as the user interacts with the GUI by clicking buttons or entering text.

3.2.2.1 Tree Files

In the 'Tree Files' section of the GUI, users can create or load a new text file within the PyCharm project, representing an avocado tree under measurement. The user can include a description in the text file, capturing information such as tree numbers, rows, blocks,

and other relevant details to be saved with the tree's positional data. After creating a new tree file or loading an existing one, a 3D measurement can be taken by the CMD and saved as a new line of text in the respective tree file.

3.2.2.2 Serial Ports

Serial port control enables the user to view, open, and connect to the correct serial port associated with the CMD as it is connected to the laptop. This flexibility allows the CMD to be connected to different USB ports on the same device or to those on a new computer, each with unique serial port names.

3.2.2.3 Avocado Mapping

After connecting the CMD and loading a text file through the GUI, the user can initiate the process of saving 3D points using the avocado mapping section of the GUI. Here, the user can input all the parameters associated with the setup and measurement of the CMD.

3.2.2.3.1 Measurement Type

This dropdown menu allows the user to specify whether a measured point is an avocado or a datum point. This information is saved with the coordinate data and is used during the datum transformation process for tripod-mounted measurements.

Below this menu is a text input for the datum diameter. This value is set by default to 50 mm, which is the diameter of the three datum points made for this task and does not need to be changed as long as the same datum points are used. If the measurement type is set as a datum measurement, this diameter value will be used to add the correct offset (the radius of the datum) to the laser distance reading when it is pointed at the spherical datum points. This results in the measured point being the centre of the datum point as opposed to its surface, ensuring the same 3D point is measured regardless of the position of the CMD.

3.2.2.3.2 Point Name/No

The number of the avocados being measured can be manually input into this text box. Alternatively, this field will automatically be populated with ‘1’ when the GUI is first launched and increase by one after an avocado measurement is taken. Unless an avocado

measurement needs to be deleted and retaken, this number doesn't require input from the user.

For datum point measurements, this field needs to be input with the colour of the datum point ('Gold,' 'Silver,' or 'Black'). This convention preserves the order of datum points across different tripod mounting positions and is crucial for correct remapping during later data processing. Similar to the avocado numbers, this field will automatically be populated with 'Gold' for the first datum measurement and will then iterate through 'Silver' and 'Black' for the remaining two datum points. After the last datum point, the field will populate with the number of the next avocado to be measured; however, at any point, these default text inputs can be changed to accommodate incorrect or revised measurements by the user.

3.2.2.3.3 Laser Reading

The user is required to input the reading from the laser distance measurement device into this field while aligning the CMD with the target. This input is used in forward kinematic calculations to determine the coordinates of the measured point.

3.2.2.3.4 Frame Mounting Point

In this drop-down menu, the user selects the point on the mounting frame where the CMD is attached, applicable for both datum and avocado measurements. This information is saved with the measured point data and is crucial for applying the correct transformation, aligning the coordinate system of the measured point with the centre of the frame or tree trunk.

If the CMD is mounted on a tripod, selecting the 'custom offset' option allows the user to input X, Y, and Z coordinate offsets, which are also saved with the measured point.

3.2.2.3.5 Custom Offset

The custom offset fields allow the user to include approximate X, Y, and Z offsets of tripod measurement positions relative to the tree origin. These values can be estimated with a tape measure by measuring the distance between the tree trunk and the tripod mounting point of the CMD along the X, Y, and Z axes of the tree origin. This rough

estimate is not utilized for any coordinate transformation process but serves as a record of where the measurements were taken from.

The rough custom offset is saved as a precaution in case of processing errors or for a sanity check on the data after recording. Additionally, it acts as a naming convention to identify points (avocado or datum) belonging to a specific tripod setup, ensuring consistency across measurements from the same origin.

3.2.2.3.6 Saving and Reading

Once all the steps are complete, the current 3D coordinates of the laser dot can be viewed or saved. Clicking the 'Save Coordinates' button prompts the GUI to read the current position of the rotary encoders and calculate the coordinates of the laser dot based on the distance input. This coordinate data is then saved to the loaded text file along with point and origin information as shown in the example below:

--- Zone 2, Block 58, Row 16, Spacing 5x2, Counted row 3 ---
- Avocado No:1, X:0.278, Y:0.411, Z:0.699, Origin:One

Alternatively, by clicking the 'Show Position' button, the GUI displays the current orientation of both rotary encoders along with the calculated coordinates of the measured point. This feature can be used to verify whether the device is working correctly before saving coordinates.

3.2.2.4 Error Handlings

All Python functions called by the GUI buttons incorporate various levels of error handling. After a button is pushed, error or confirmation messages are displayed at the bottom of the GUI. These messages alert the user of an error or confirm that an operation has been successfully completed. Table 3.1 provides a comprehensive list of all the error handling and confirmation feedback messages integrated into the GUI.

Table 3.1 – GUI error/confirmation messages

Message	Details	Action
Error: Unsupported platform	An environment error was raised due to an unknown computer operating system	Run Python script from Mac, Windows, or Linux OS.
Error: Invalid data	Encoder data being read from the Arduino is not in the expected format and may be corrupted.	Check the encoder and USB connections. Arduino or encoders may be malfunctioning.
Error: Encoder moved	The device has moved while taking the measurement causing encoders to rotate.	Retake measurement without moving the device.
Error: Read error	The data received is incomplete.	Check the encoder and USB connections. Arduino or encoders may be malfunctioning.
Error: Directory does not exist	Loaded tree text file or directory for saving data does not exist.	Ensure the 'Tree Data' directory is within the PyCharm project directory and the text file being loaded exists.
Error: File already exists	The tree file being created already exists.	Load a file instead or create a new file with an alternate name.
File Created	A new file has been created and is loaded automatically	None
Error: No ports found	No serial ports have been found on the computer.	Ensure CMD is connected to the computer via USB and ports are functioning.
Ports loaded	Computers serial ports successfully loaded.	None
Error: Please select valid port	A valid port has not been loaded and selected from the drop-down port menu	Load and select the correct serial port corresponding to the CMD
Serial port opened	The serial port has been successfully loaded and opened	None
Error: Please input valid datum diameter	The text input for datum diameter is invalid	Input the correct numeric value (mm).
Error: Port not open	A serial port has not been opened.	Load and select the correct serial port corresponding to the CMD

Error: No file selected	No tree file has been created or loaded.	Load an existing tree file or create a new tree file to save data.
Error: Please input valid laser reading	The text input for laser distance is invalid	Input the correct numeric value (m).
Error: Please select origin	The origin point for the CMD has not been specified	Select a frame origin point or custom offset from the drop-down menu
Error: Please input a valid custom offset	The text input for custom offset coordinates is invalid	Input the correct numeric value for X, Y, and Z (m)
Reading Encoders…	Encoder data is being read from the Arduino	Do not move the CMD
Position Shown	Data has been successfully read and is being displayed by the GUI. Data has not been saved.	None
Written to file	Data has been successfully read and saved to the specified tree file.	None

Implemented error handling prevents the unintentional saving of incomplete or corrupted data, which could pose a significant setback during the data collection process. The provided feedback ensures that the user can confidently save valid data to the correct location every time.

Error messages proved invaluable on the first day of data collection at the orchard. A loose cable connection was causing one of the rotary encoders to fail in sending data to the Arduino. The displayed read error code promptly highlighted the issue, allowing for a quick resolution before continuing with the data collection. Without this error notification, a substantial amount of potentially faulty data might have been recorded, necessitating the remeasurement of trees.

3.2.3 Reading Data

To extract data from the encoder positions communicated by the Arduino UART serial, a Python script utilizing Python Serial was developed. When the user initiates the save operation on the GUI, a function is called to read data from the serial port. Multiple

readings are taken from each encoder, to be recorded and cross-verified, ensuring their consistency. This verification mechanism allows for the detection of errors in case any of the encoders are inadvertently moved during a reading.

Additional error checks are implemented to identify incomplete or incorrectly formatted data. Such errors can arise from encoder malfunctions, faulty connections to the Arduino, or malfunctions in the Arduino or USB connection.

After confirming a valid, stable reading, it is converted into an angle in degrees based on its bit value (ranging from 0 to 4095). Subsequently, this angle is processed alongside the laser distance input using the forward kinematic equations of the device.

3.2.4 Saving Data

Coordinate data is systematically stored in text files designated for individual trees. Users can create new tree files, incorporating information about the tree's condition, number, location in the orchard, and other relevant details, which is saved at the top of the text file. Subsequently, new point measurements are appended as new lines to the file, featuring the point type, name, measured coordinates, and origin point in a standardized format.

Accurate point naming conventions are crucial for later correct data reading and processing. For instance, a first datum point measurement must be recorded as a datum point and named 'Gold.' Any deviation in naming can lead to errors returned by the datum transformation functions. Fortunately, the impact of such errors is minimized, as all points are saved to text files before processing. This allows for corrections to naming mistakes at any time without affecting the measured data.

The laser distance reading and encoder positions are the only pieces of information not saved to the text file. After the forward kinematics equations are used to calculate the coordinates of the measured point, only the X, Y, and Z coordinates are stored with the point information in the text files.

3.2.5 Data Processing

The data processing occurs independently of the data collection process. This means that all measurements taken with the CMD are saved as coordinate points relative to the device's origin. A separate Python program was written to read the saved information in the tree text files and transform all the coordinate points according to their specified origin. This ensures that all measured points become relative to the same origin—specifically, the centre of the tree's trunk at the height of the frame.

Given that the frames are consistently set up level and north-facing, and the spacing of orchard blocks is known, it allows avocados from multiple trees to be plotted correctly relative to one another. This aids in gaining insight into how avocados from different trees may overlap.

3.2.5.1 Coordinate Transformation

Coordinate point rotations were applied using NumPy matrix multiplication Python functions. The following equations were used to apply rotations about the X, Y, and Z axis respectively:

$$\begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\begin{bmatrix} x_r \\ y_r \\ z_r \end{bmatrix} = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

These equations are the matrix representation of two-dimensional Pythagorean calculations used to solve for the coordinates of a point from the angle of its vector. Combined, these rotation matrices can be used to apply any rotation in 3D space.

For frame-mounted transformations, only the last equation is needed to apply a rotation about the Z axis. For datum point transformations, the order in which rotations are applied need to be considered to ensure rotations can be applied and reversed correctly during the remapping process. Translations are simply applied by adding offsets to the coordinates of a point to be translated.

3.2.5.2 Frame Position Remapping

The geometry of the frame is used to apply rotations and translations to measured coordinates based on their mounting positions on the frame. There are eight mounting points on the frame which position the CMD radially facing outwards, 260 mm away from the centre of the trunk in 45-degree increments. A list of X, and Y offsets, and angles corresponding to each point was created using the frame's geometry.

An offset from the list was employed to apply the correct rotation and translation to a local CMD coordinate based on the frame offset saved with the measurement.
Figure 3.11 illustrates four arbitrary local coordinate points (plotted in purple) processed for all eight frame origin points (all other coloured points). As expected, the transformed points are shifted and rotated to match the geometry of the frame mounting positions while maintaining their positions relative to one another.

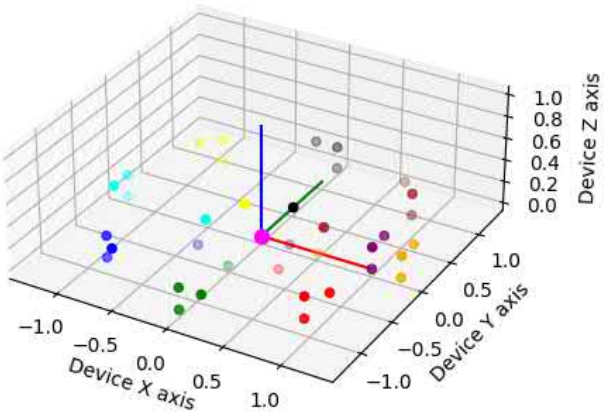


Figure 3.11 – Remapping of arbitrary 3D points based on frame geometry

When mounting the CMD on a frame point, it is essential to position it in such a way that its local X-axis is directed away from the tree trunk. This orientation is crucial for the correct interpretation of measured local coordinates to the global reference frame.

3.2.5.3 Datum Point Remapping

The datum point remapping process involves establishing reference positions for all three datum points based on a known origin, typically a frame mounting point. Once these reference datum points are established, subsequent measurements can be taken from various tripod positions, provided all three datum points have a clear line of sight to the CMD.

For each new tripod mounting position, it is crucial to measure all three datum points before proceeding with avocado measurements. The data is initially saved in local CMD coordinates and is later remapped to align with the target coordinate points of the tree.

In Figure 3.12, the remapping process is illustrated step by step. The target datum points are represented with a green plane and have been transformed based on their known frame position, resulting in correctly positioned points relative to the frame coordinate system. The local tripod-measured datum points are depicted with a red plane. As they have been measured from a new tripod position, they occupy a different 3D position.

The blue plane represents the red datum plane at each step of the transformation process. The overall shape of the triangular plane formed by both the target and tripod datum points remains similar, illustrating the relative position of the datum points as they are distributed on the ground. This process ensures that measurements from different tripod positions are accurately mapped to the same coordinate system.

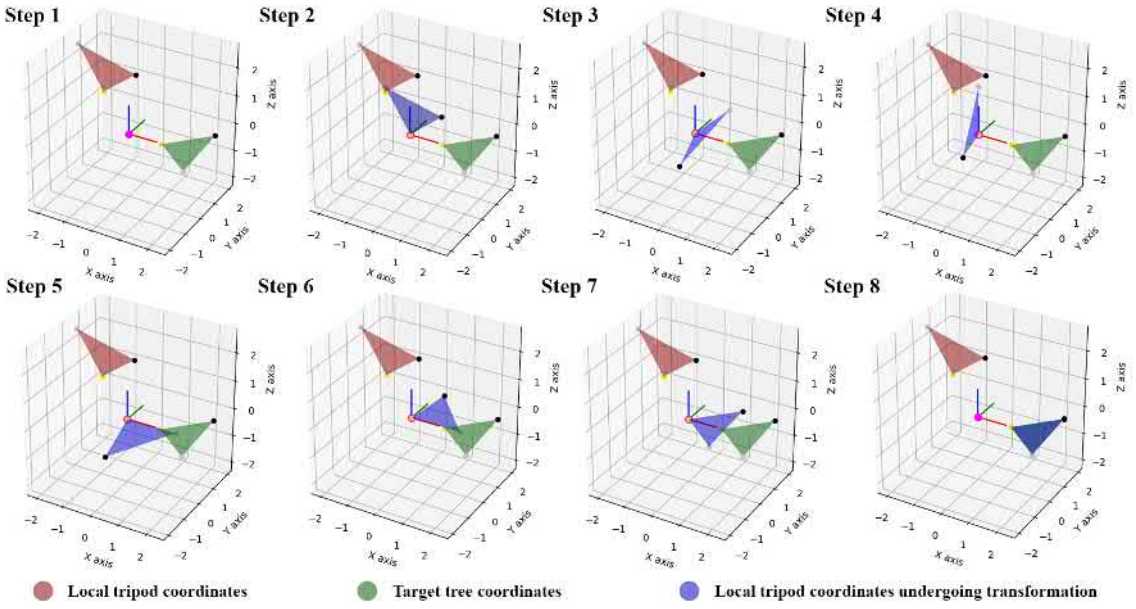


Figure 3.12 – Step by step datum remapping process.

The method developed for remapping the three datum points from one coordinate frame to another involves a flattening procedure, where the datum points of both coordinate frames are translated to the global zero point and aligned flat with the global coordinate axis. The rotations needed to move the target datum points to this position are recorded and then applied in reverse order to the datum points being remapped. The process is implemented in Python through the following steps:

1. Establishing Target Datums

   The target datum points (green) are initially plotted in the correct global coordinate system, alongside the red datum points that require remapping.

2. Translation to Global Origin

   All remapping datum points are translated such that the gold datum point is located at the global origin zero point.

3. Rotation about Global Z-Axis

   Remapping points are rotated about the Z-axis until the silver point coincides with the Z/X plane or when its Y coordinate equals zero.

4. Rotation about Global Y-Axis

Remapping points are rotated about the Y-axis until the silver point coincides with the Z/Y plane when its X coordinate equals zero.

5. Additional Y-Axis Rotation

Remapping points are rotated -90 degrees about the Y-axis to align the vector formed by the gold and silver datum points with the X-axis.

6. Rotation about Global X-Axis

Remapping points are rotated about the X-axis until the silver point coincides with the X/Y plane or when its Z coordinate equals zero. The remapping datum points are now aligned with the global coordinate system.

Steps 2 to 6 are also applied to the target datum points, and all rotations and translations are recorded. This results in both the target and remapping planes aligned in the same way with the global axis while the transformations and offsets required to move both planes to the original position of the target are known.

7. Reversing Rotations

The rotations needed to align the target datum plane to its original orientation within the global coordinate system are applied in reverse order to the remapping points.

8. Inverse Translation

The inverse of the translation required to align the target gold datum to the global origin is applied to the remapping points.

After these steps, the remapping points are rotationally aligned with the target datum points, sharing the same gold point coordinates. This entire transformation process is also applied to any avocado measurements taken from the same coordinate system as the remapped datum points.

The outcome is that both datum and avocado measurements, taken from any tripod mounting position, are remapped through a best-fit approximation to three target datum points. Remapped gold points always have zero error, while the silver and black points exhibit increasing error due to the imperfect measurement of the datum points across the two reference frames. This error is saved with the data during the remapping process, enabling the user to assess the accuracy of the datum measurements and the remapping process.

A potentially more accurate method would involve remapping the datum points based on their centroid point rather than corner by corner. This approach would average the error between each of the three datum points, rather than having zero error for the first point and gradually increasing error for the next two. Despite the potential improvement in accuracy, the current transformation method's error was deemed acceptable in testing. Due to time constraints in this research project, a centroid-based remapping approach was not pursued.

3.3 Validation

This section covers the validation procedures and results used to ensure all mechanical, electrical and programming elements of the CMD and data collection process are functioning as expected, within an acceptable error.

3.3.1 CMD Validation

In addition to the one-dimensional distance validation that was conducted with the laser measure previously, the fully assembled 3D measuring device would also require further validation to quantify any error that may be present in the rotary encoders and the mechanical pivots and linkages. 3D validation can be achieved by measuring known points in 3D space and comparing the measured coordinates with the actual coordinates of the known points. Such a test would also verify the kinematic equations and code used to process the raw distance and orientation data to provide a coordinate measurement.

3.3.1.1 Setup

The selection of suitable validation points for the device involved deliberation on potential known measuring points. Consideration was given to using the corners of a

room's floor and ceiling as reference points. However, this approach hinged on the assumption that the theoretical dimensions and geometry of the structure were accurate or that there existed a calibrated device of sufficient precision to measure these points and determine the CMD's relative position.

In the absence of a structure with known points to centimetre precision or an additional calibrated measuring device, an alternative validation method was devised, utilizing a high-precision CNC machine.

The CNC machine employed in the validation process was calibrated to a precision of within 10 micrometres, covering a volumetric travel of over 0.12 cubic meters. The setup involved fixing a target point on the machine that could be consistently relocated to eight known points, representing the corners of a theoretical box with dimensions of 1 by 0.8 by 0.15 meters. The target point was established by marking a strip of tape with a small spot on the main spindle head of the machine, as illustrated in Figure 3.13.

Figure 3.13 – CNC machine used to create known coordinate points

To measure the predefined points, the CMD was positioned on a tripod at a suitable distance from the CNC machine (Figure 3.14). Each of the eight designated machine positions was then measured using the CMD, and the resulting coordinates were recorded. Subsequently, these measured points were compared against the known machine positions to assess the error in the measurements. This procedure was repeated two additional times, each with a distinct orientation of the device relative to the CNC machine. This not only provided a more comprehensive validation of the device's field of

view (FOV) but also increased the number of measured points available for error evaluation.

Figure 3.14 – Measuring positions of CMD from three different arbitrary orientations

Measurements were captured with the device positioned up to four meters away from the CNC machine. This approach aimed to validate the accuracy of relative measured points at the upper limit of the distance the device would eventually be employed for measurements.

3.3.1.2 Error results

All 24 points were recorded and processed utilizing the GUI and Python code developed for avocado measurements. The resulting coordinates were plotted, generating a 3D distribution representing the three boxes created by the CNC machine, as illustrated in Figure 3.15. A visual inspection of the plot indicates that the points form three identical boxes, positioned at each arbitrary orientation as anticipated.

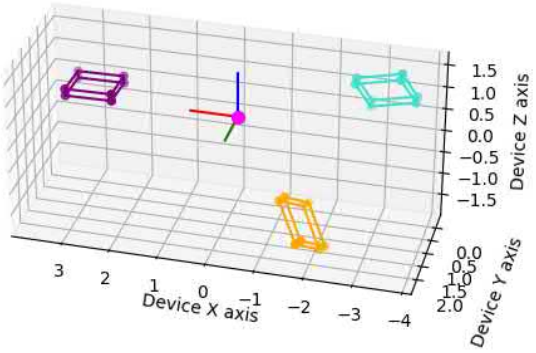


Figure 3.15 – 3D plot of measured CNC positions

With this dataset, the relative error between measured points can be calculated by determining the distance between the measured points and comparing it to the known distance moved by the CNC machine. This was achieved by calculating the length of each of the 12 edges of the boxes using the following equation:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

This equation utilizes the 3D coordinates of two points to compute the length of the vector between them. Table 3.2 provides the disparity between the edge length measured by the CMD and the actual distance moved by the CNC machine for each measured box. Additionally, it includes the maximum, average, and standard deviation calculated from the absolute error values.

Table 3.2 – Error between measured and actual edge length (mm)

Edge	Box 1	Box 2	Box 3
1	6.2	-7.5	-5.2
2	3.9	-3.6	2.5
3	0.3	-15.0	-8.0
4	-0.3	-2.8	0.6
5	7.1	2.9	-0.6
6	6.3	-11.1	-7.2
7	1.0	-2.3	-2.0
8	-1.7	-16.4	-8.9
9	-1.4	-0.9	-1.5
10	2.0	-3.5	4.8
11	4.1	-2.8	2.5
12	3.1	-5.6	-1.5
Maximum	**7.1**	**16.4**	**8.9**
Average	**3.1**	**6.3**	**3.8**
SD	**2.4**	**5.2**	**3.0**

The obtained data revealed that within a measurement distance of four meters, the average absolute error between points is less than 4.4 mm, with a maximum of 16.4 mm. Considering that this error is based on the distance between two measured points, and it involves compounding errors from two measurements, the error in a single measurement would be half of these values. Consequently, it results in an average error of 2.2 mm and a maximum error of 8.2 mm. These values also account for any user error associated with

aiming the laser dot precisely at the designated position on the CNC machine for each measurement. The full coordinate dataset and error results can be found in Appendix C.

Based on this test, it can be reasonably concluded that any measurement taken by the CMD within a range of four meters will be well within the centimetre precision required for measuring avocados on trees.

A crucial consideration for this test is that it doesn't validate the accuracy of measurements taken by the device relative to its origin; it only assesses the accuracy of measured points relative to each other. The global accuracy of measurements is influenced by the zero-setting procedure of the device and the one-dimensional distance validation of the laser measuring device. Hence, any error in the absolute measurements taken by the device will be consistent, resulting in a shift in the coordinate frame or origin point of measured avocados. This shift remains the same across different trees being measured by the same device and will have a minimal impact on understanding the spatial distribution of avocados on their respective trees.

3.3.2 Mounting Frame Validation

3.3.2.1 Theoretical Geometry Compensation

The evaluation of errors in the frame mounting points and the code responsible for transforming the coordinate systems of those mounting points included marking five fixed points at various positions on the inside walls of a house. The avocado mapping device recorded the coordinates of each of these measurement points from every origin point on the frame (Figure 3.16). Subsequently, the measured points were transformed to the global coordinate system of the frame, based on the theoretical positions of the origin points as defined in the CAD model of the frame.

Figure 3.16 – Measuring points on the wall from all frame positions

In an ideal scenario, this process would yield identical global coordinates for each measured point from all eight origin points. However, the error was quantified as the length of the vector between a measured point from origin one and the same measured point from a different origin. The error of each measured point is detailed in Table 3.3.

Table 3.3 – Frame mounting point deviation measurements (mm)

Measured Target Point	**Frame Mounting Points**								**Distance from Frame**
	One	**Two**	**Three**	**Four**	**Five**	**Six**	**Seven**	**Eight**	
1	0	26	15	24	31	15	18	8	~2600
2	0	61	19	32	41	20	30	21	~6100
3	0	26	15	14	10	12	17	7	~2400
4	0	19	8	6	1	20	19	8	~2200
5	0	40	24	32	40	26	46	20	~5100
Max	**0**	**61**	**24**	**32**	**41**	**26**	**46**	**21**	
Average	**0**	**34**	**16**	**21**	**26**	**18**	**26**	**13**	
SD	**0**	**17**	**6**	**11**	**15**	**5**	**12**	**7**	

Results revealed a maximum deviation of only 61mm, observed in a point measured over 6 m away from the frame. This range is nearly twice the maximum expected height or width of the trees to be measured. Consequently, it can be reasonably assumed that the CMD can measure the position of an avocado at this range from any point on the frame with an error as small as the width of one avocado.

3.3.2.2 Measured Geometry Compensation

The observed error stems from slight imperfections in the construction of the frame, likely resulting from minor warping during the welding process. One potential improvement could involve recording the 3D rotations and translations necessary to align points measured from the numbered frame origins to the correct central frame origin. This approach mirrors the process used for aligning coordinate systems with datum points, enabling a more precise representation of the frame's geometry compared to theoretical CAD geometry.

While the ideal solution would involve access to a large surface grinding machine to enhance the flatness and accuracy of each mounting point, such resources were not available. However, it is feasible to measure and quantify the deviation of the measurement points from their theoretical geometry and apply this offset to coordinate measurements from the frame. Given the unchanging geometry of the frame, this correction could be implemented either before or after data collection took place at the King Avocado orchard.

After evaluation of all sources of compounding error, it was determined that the error introduced by the frame was acceptable for the purpose of measuring the 3D position of avocados. Consequently, additional measured compensation for the frame was deemed unnecessary.

3.3.3 Tripod Measurement Validation

This section covers the validation process of the entire datum point remapping process required to take measurements with the CMD from a tripod-mounted origin. All aspects of this process, including the physical measurements and the coordinate remapping code, are evaluated to uncover any compounding errors.

To begin the validation test, datum points were placed in a random configuration on the ground, spaced several meters apart. These points were then measured by the CMD, positioned arbitrarily relative to the datum points. Datum measurements, along with their corresponding colours (Gold, Silver, Black), were recorded to ensure accurate identification for the remapping process. This initial measurement position served as the

global origin, to which all subsequent measurement origins would be converted. It is depicted as a green plane in Figure 3.17. Subsequently, the CMD was relocated to a new arbitrary position and orientation for measuring the same datum points, illustrated as red a plane in Figure 3.17.

This procedure was replicated for two additional datum point layouts, introducing variations in layout conditions. Changes included reversing the order in which datum points were positioned on the ground and placing the CMD both inside and outside the triangle formed by datum points during measurements. This deliberate variation ensures the robust validation of the remapping procedure, demonstrating its efficacy across diverse scenarios without requiring specific positions or ordering of the datum points.

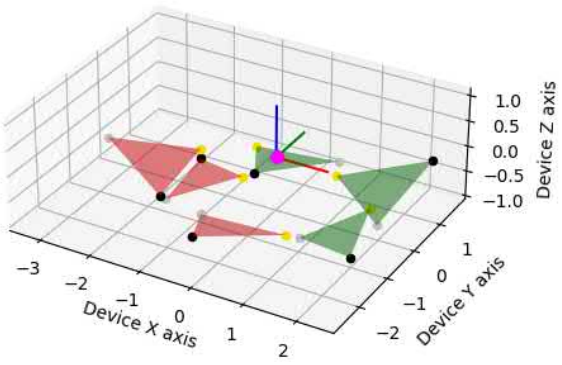


Figure 3.17 – Three different datum setups measured from different origins

In Figure 3.17, three distinct red-green triangle pairs are depicted, each associated with its unique datum point layout. In an ideal scenario where the programming and datum measuring procedure are perfect, the remapping process would precisely transform all the red triangles onto their respective green target triangles, resulting in zero offset between their gold, black, and silver points.

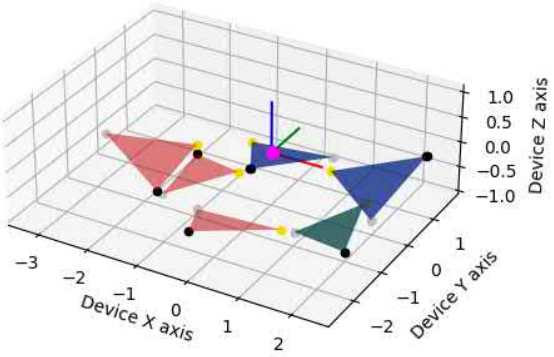


Figure 3.18 – Datum points after remapping onto target origins

Figure 3.18 illustrates all the datum points after undergoing the remapping process to their target coordinate frames. The remapped datum points are denoted by blue triangles, closely overlapping with the target green datum points. Following the completion of the remapping process, the error between the target and transformed points is automatically saved into a text file by the Python program. This enables users to assess the maximum error in the transformation process, which will also be reflected in any avocados transformed concurrently.

The error for each point transformed in this validation test is listed below in Table 3.4. The error in these transformations naturally increases from the gold to silver and black points due to the way the remapping is programmed. The gold points are always translated first directly onto their respective target point, and subsequent rotations are applied about this same point to align the other two points as closely as possible. The full measured and transformed coordinate dataset can be found in Appendix D.

Table 3.4 – Datum transformation validation error

Error (mm)	**Gold**	**Silver**	**Black**
Layout 1	0	7	16
Layout 2	0	4	2
Layout 3	0	6	12
Max	**16**		
Average	**5**		
SD	**6**		

These results show an acceptable amount of error across all points, with a maximum error of 16 mm and an average of only 5 mm. This falls within the accepted error produced by the measuring frame itself, indicating that a tripod-mounted CMD used in conjunction with these datum points and this coordinate remapping process will be capable of measuring avocados within an acceptable margin of error.

3.3.4 Compounding Error

The experimental error, derived from the mounting frame, datum remapping process, and the CMD, is summarized in Table 3.5. The measurement frame contributes the maximum expected error for any measurement type, reaching 61 mm, with an average expected error of 19 mm. As this frame error is determined using CMD measurements, it inherently represents their compounding errors, which is also applicable to the listed datum errors.

Table 3.5 – Worst-case compounding error arising from all aspects of the measurement process (mm).

	Average Error	**Max Error**
CMD	4	17
Frame	19	61
Datum	5	16
Maximum	**19**	**61**

It is important to note that these validation tests encompass any potential user error stemming from accidental movement of the tripod or frame supporting the CMD, as well as inconsistencies with manually aiming the CMD laser onto target points. Given that the frame validation process includes measurements beyond 4 m from the CMD, it is concluded that the expected error from an average coordinate measurement would be much less than the maximum summarized in Table 3.5.

Based on these results and considerations the expected error of any measurement taken within an orchard tree would be within the width of an average avocado. Therefore, the CMD along with the implemented measurement procedures and technology were deemed to be suitable for capturing 3D coordinate points of avocados on the tree to quantify and gain insights into their spatial distribution.

3.4 Single Tree Trial

To further validate the device before its deployment into the orchard for full-scale data collection, a small-scale trial was conducted on a local avocado tree in Palmerston North (Figure 3.19). This privately grown tree is approximately five meters tall and estimated to be around 20 years old.

Figure 3.19 – Trial tree located in Palmerston North

A small-scale trial would further validate the equipment, measurement and data processing of the entire procedure and include a photographic comparison between a real-world photograph and a simulated photograph rendered based on the measured avocado coordinates.

3.4.1 Setup

Measurements were taken from tripod mounting positions underneath the tree, utilizing the three datum points to remap the coordinate frame from different mounting points.

Figure 3.20 – Trial measurement setup with datum points (1, 2, 3), data collection PC (4), tripod (5), and CMD (6).

3.4.1.1 Frame

The mounting frame was not employed for this device due to the large trunk of the tree and its lower propagating branch, rendering it too large for the frame to fit around. Given that the trunk size of this tree was much larger than what could be expected from commercially grown orchard trees, and considering the frame had already undergone thorough indoor validation testing, its inclusion in this test was not deemed crucial.

3.4.1.2 Datum Points

Figure 3.20(1, 2, 3) shows the datum points placed out around the base of the tree with roughly 2-3 meters spacing between them. A datum spacing aids in the remapping process by reducing the positional error of measured avocados that have been remapped to another datum coordinate system.

3.4.1.3 Tripod

The positioning of the tripod is not critical for taking measurements, as long as it offers a mounting point for the CMD with a direct line of sight to all three datum points and the avocados to be measured.

For the first position, the tripod was placed next to the base of the tree (Figure 3.20(5)), and its approximate position along the X, Y, and Z axes relative to the centre of the tree trunk was recorded using a tape measure. The orientation of all axes was estimated by aligning the X-axis with north and the Z-axis with the global vertical axis. These origin offsets were input into the GUI, saved with any measurements taken from that position, and served as a rough approximation to orient measured avocados from the first tripod position to the position of the tree.

Once a sufficient number of measurements were taken, the tripod was then moved to another position on the other side of the tree, providing a line of sight to all three datum points. Additional datum point and avocado measurements were then taken, which could be remapped to the first tripod coordinate position.

3.4.2 Data Collection

To measure datum points and avocados, the CMD was mounted to the tripod and manually aimed by the user. The red laser dot from the laser measure device was used to target the centre of the base of avocados (Figure 3.21(b)). Avocados were measured by scanning from right to left along a vertical line starting at the first avocado. This systematic approach was employed to help remember which avocados had already been measured, although it still relied on the user's memory, which could be unreliable, especially when measuring multiple trees over a full day of work.

Datum points were measured by aiming the laser at the centre of each datum point in the order of Gold, Silver, and Black to maintain the naming convention for the remapping program.

(a) (b)

Figure 3.21 – Aiming the CMD by hand at a target avocado (a). Laser dot visible on the base of an avocado (b)

A total of 48 avocados and 6 datum points were measured from two tripod positions, covering roughly half of the avocados on one side of the tree. The saved data from this trial can be found in Appendix F.

The time required to collect this dataset was 13 minutes for equipment setup and 40 minutes for taking measurements. At this rate, it would be possible to measure 50 avocados on one tree within an hour. The total time needed to measure a tree in an orchard greatly depends on factors such as the number of avocados on the tree, as well as the size,

shape, and occlusion of the tree. However, based on this test, it is feasible to capture a reasonable dataset of trees using this method.

On the day of measuring this tree, wind conditions were observed to be causing avocados to be blown up to one avocado's width from their natural hanging position, increasing the difficulty of taking measurements and introducing further error. Bright sunlight conditions and wind-induced movement of avocados caused fluctuations in the laser reading, with values varying up to 30mm for avocados further away from the device. This additional source of error was attempted to be minimized by observing the fluctuations of the laser reading for a few seconds and approximating an average value.

3.4.3 Photographic Validation

The main objective of this master's project is to collect spatial data that can serve to gain insights into an optimal real-world photography procedure. This data will facilitate the simulation of photographs of the measured trees from different angles or camera profiles. To ensure the reliability of this process, simulated images will be compared to real-world photographs taken of the tree.

Ideally, a perfect result would depict avocados in the simulated image exactly as they appear in the real image, establishing confidence in simulated images generated from the positional data collected through the previously outlined methodology.

3.4.3.1 Photo Capturing

The real photograph was captured using an iPhone 12 mini, facing along the X-axis, and positioned approximately from the same origin point as the first tripod position (X = 0, Y = 1, Z = 1), with a 45-degree upward inclination. It's important to note that the phone's orientation is a rough estimate based on tape measurements and a levelling application, introducing a certain degree of error into the photographic validation comparison.

Figure 3.22 – Photograph taken on iPhone 12 mini

Figure 3.22 shows the real-world validation image that includes a portion of the measured avocados during the trial. A simulated image based on the measured positional data of these avocados should include the same avocados rendered in the same frame positions, allowing for some error due to uncertainty in the camera position and simulated camera intrinsic parameters.

3.4.3.2 Camera Intrinsics

Simulating a camera involves applying the camera's intrinsic parameters to a standard pinhole camera model, programmed in Python. This model, based on the known 3D position of a single point light source within its field of view (in this case, an avocado), calculates the corresponding pixel coordinates that the light source would project onto the theoretical camera sensor (see Figure 3.23).

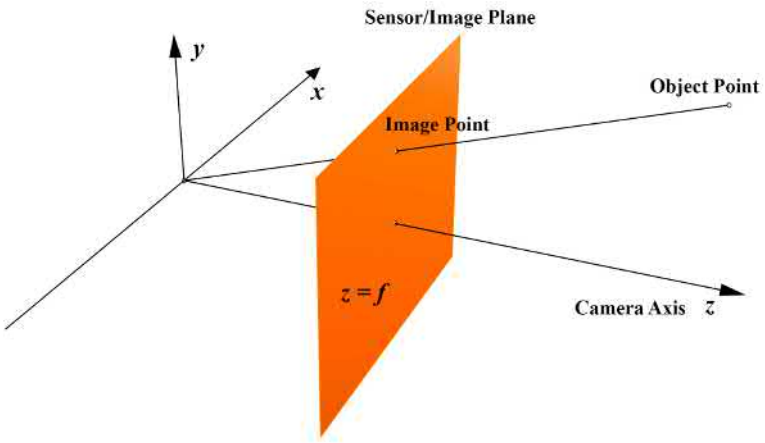


Figure 3.23 – Pinhole Camera Model

A camera matrix based on the pinole model has the following format, where c_x and c_y are offsets that define the principal point in the image, and f is the focal length of the pinhole camera.

$$Camera\ Matrix = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

By default the pinhole camera model has the camera axis or lens direction aligned with the Z-axis. To aim the simulated camera, its orientation and position can be included in a more detailed camera matrix or alternatively, the object points being processed can be transformed relative to the standard camera origin.

The matrix parameters are calculated from the image height and width, and the horizontal FOV. The pixel size also needs to be known to convert 3D object coordinates in millimetres to units of pixels. The camera matrix based on specified camera intrinsic parameters can then be multiplied with a 3D object point to calculate the position of the object on the image plane:

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix}$$

Once an image point has been calculated from the object point, its coordinates are either saved for image rendering or discarded if it does not project onto the sensor image plane

of the camera model. The calculations and matrix multiplication of the camera model were done with Python packages NumPy and OpenCV for image rendering. The Python code written to process and render 3D avocado points into an image is included in Appendix G.

The simulated image used for comparison to the real image was created based on the camera intrinsic parameters of the iPhone 12 Mini used to take the real-world photograph [86, 87].

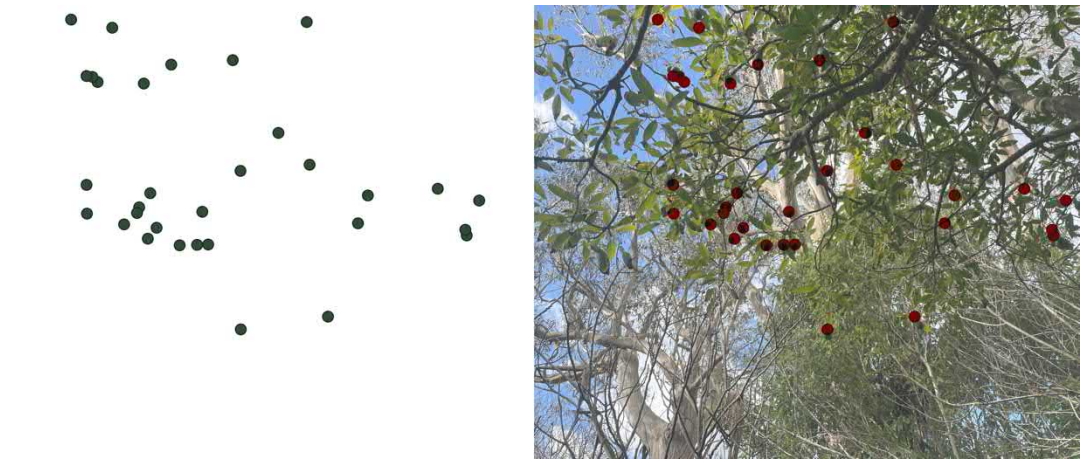
The sensor height and width correspond to the maximum pixel resolution of the phone, and the horizontal FOV was calculated based on an equivalent focal length of 28 mm [86]. The width of the sensor pixel was not specified by the manufacturer, however, a standard 1.7-micrometre pixel size [88] was used as a baseline to render the first images and was manually adjusted until the FOV of the simulated and real images was similar. The final intrinsic parameters used to render the simulated image are shown below in Table 3.6.

Table 3.6 – Camera intrinsic parameters used in image simulation

Intrinsic parameters of iPhone 12 Mini		**Units**
Horizontal FOV	1.1722	Radians
Sensor Width	4032	Pixels
Sensor Height	3024	Pixels
Pixel Size	0.0014	Millimetres

3.4.3.3 Simulated Image

The avocado points measured during this single tree trial were transformed to align the simulation pinhole camera model with the orientation of the real-world comparison photograph. Subsequently, the points were processed with the camera matrix and rendered into a 2D image as green circles using OpenCV, resulting in the image shown in Figure 3.24(a).



(a) (b)

Figure 3.24 – Rendered image based on simulated camera and measured avocado coordinates from single tree trial (a). Superimposed real and rendered comparison images (b)

3.4.3.4 Comparison

Figure 3.24(b) shows the real image captured on the iPhone 12 mini overlayed with the simulated image. The rendered avocado points are shown as red dots over the real image. The simulated image points are closely positioned over their corresponding real avocados with up to two avocado widths offset in some cases.

The two comparison images were not expected to be perfectly aligned due to compounding errors arising from factors such as the real-world camera position uncertainty, the estimated pixel size of the iPhone camera used in the intrinsic calculations, and the windy conditions during avocado point measurements. With these sources of error taken into consideration, the results of this image comparison trial provide clear validation of the ability of the CMD, the data capturing and processing procedure to measure the position of avocados in a tree in such a way that allows images to be simulated based on that data that would adequately represent an equivalent real-world photograph taken of the same tree.

These results show the promising potential of being able to simulate the fruit distribution within photographs of measured avocado trees (or similar fruit trees), which could then be used for the optimisation of image-capturing procedures and counting using AI.

3.4.4 Trial Based Recommendations

The initial single tree trial provided valuable insights into the reliability and practicality of using the device for data collection. The results indicated that measuring 50 avocados within an hour on a single tree was feasible. Extrapolating this rate of data collection to an average per-tree avocado count of 95, as provided by King Avocado across two orchard blocks, it was estimated that data could be collected from 21 trees over a 40-hour working week. To maximize the dataset from the orchard, it was recommended to allocate at least five 10-hour days for data collection. However, adverse weather conditions were acknowledged as a factor that could impact or impede data collection.

The timing estimate was based on the first tree measurement, and it was anticipated that the data collection process would become more efficient with an optimized approach developed over multiple days of measurements.

Certain challenges were identified during the trial, such as the difficulty of visualizing the red laser dot in the canopy, especially in bright sunlight. This issue was expected to be less pronounced in smaller commercial orchard trees. Another challenge was the need for a reliable method to keep track of measured avocados, prompting the consideration of marking individual avocados with paint or stickers.

Additionally, it became evident that having a small table or box alongside the tree, along with a portable camping chair, would create a more comfortable setup for the laptop user to input data and take measurements. It was recommended to have a second person to aim the CMD from under the tree and relay distance measurements verbally to the laptop user, enhancing the efficiency of the process.

3.5 Orchard Data Collection

The orchard data collection took place at the King Avocado orchard in Kaitaia, Northland, under the supervision of Claudia Hermosilla, the orchard manager. This is the same orchard from which images were captured for the training of the avocado counting AI during the previous master's project [7]. It was chosen for its multiple blocks of avocado trees that are grown to a commonly adopted commercial standard spacing and size which

represent the direction of commercial growers around the world, as well as their previous cooperation and contribution towards this project.

Figure 3.25 – King Avocado Orchard [89]

Figure 3.25 shows a satellite view of the orchard with the measured blocks highlighted. All rows of avocado trees within the orchard are longitudinally aligned with true North/South. The magnetic declination at the orchard's location is positive 18 degrees [90], which is used to orientate and plot avocados from different trees that were measured using the measurement frame that is set up to align with magnetic north.

Data collection was conducted from the 11th to the 15th of September 2023, spending on average 10 hours a day taking measurements. King Avocado stated for any growing season they aim to count their on-tree fruit between March and April to aid in commercial export planning. The positional data collection was unable to be conducted during this time due to the CMD device not being completed and validated by this time. For the purposes of positional measurements, the same fruit remains on the trees and will only begin dropping from December, resulting in a minimal effect on the data's applications for yield estimation by measuring in September.

One predicted difference in the positional data between these growing months is that the increased size of the fruit may cause them to hang slightly lower on the branches during September, however, this difference will likely have an insignificant effect on the collected dataset.

3.5.1 Block Selection

Due to the time constraints of this project, the number of trees able to be measured is somewhat limited. Because of this, it is important to select the most relevant trees that most completely represent the future of commercial avocado growing within New Zealand and the world to extend the applications of the unique dataset.

The literature review of this thesis indicated that commercial growers aim to maintain orchard trees below 3 meters tall [45-47]. More economical spacing around 2.5 by 5 meters is also becoming more prevalent over traditional spacing of 7 by 7 meters. Two blocks were chosen from the King Avocado orchard for their smaller, more economical trees and spacing. All avocados measured were of the Hass variety.

3.5.1.1 Block 58

The first block to be measured is shown below in Figure 3.26. This block consisted of trees ranging in height between approximately 1m and 3.5m with row spacing of 5m with 2m spacing between each tree.

Table 3.7 – Block 58 specifications

Block Specifications	
Spacing	5x2 m
Average Tree Height	~3 m
Row Orientation (magnetic bearing)	18°
Raised Bed Rows	0.5 m
No Counting Rows	6
Planted	Feb 2017

There are 43 rows in the block with a maximum of 30 trees per row, tapering down to 5 trees along the shortest row. Trees within this block typically have a very dense canopy

with lots of occlusion caused by branches and leaves within the tree which made measuring the avocados challenging and often required the use of a tripod-mounted setup for the CMD.

King Avocado currently utilises a manual counting method that involves selecting specific rows to count by hand and from which to extrapolate an average fruit per tree estimate. This block includes 6 of these counting rows that had recently been counted by King Avocado on the 1st of March 2023.

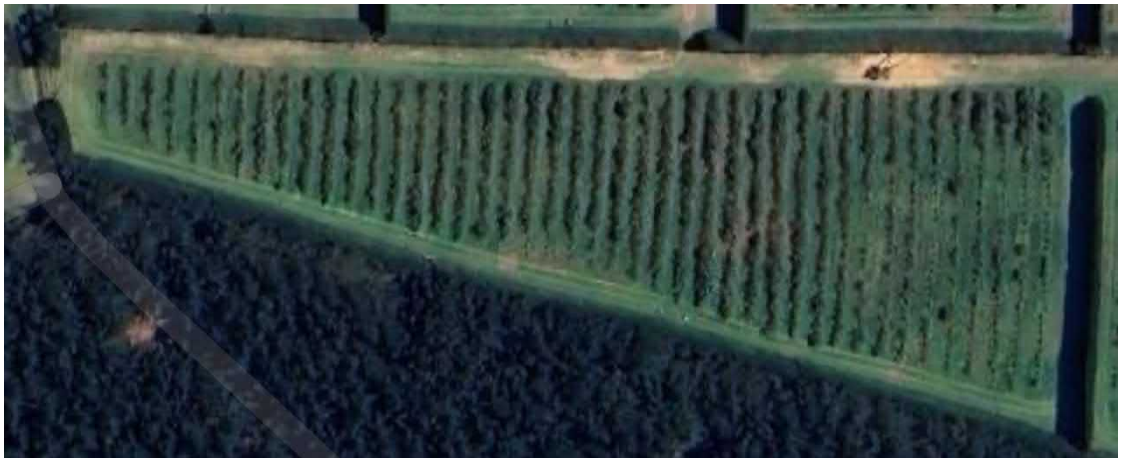


Figure 3.26 – Orchard Block 58, Zone 2 [89]

This block was also one of the blocks photographed by Kyle Macadam for annotation and training of the avocado counting AI model developed to identify avocados in an image.

3.5.1.2 Block 49

To create a dataset with more variety in block configurations and tree size and age, a second block was also measured. This block had slightly larger trees averaging approximately 3.5 m in height and a larger square spacing of 3.5 x 3.5 m

This is a considerably smaller block with only 17 rows and a maximum of 23 trees per row.

Table 3.8 – Block 49 specification

Block Specifications	
Spacing	3.5x3.5 m
Average Tree Height	~3.5 m
Row Orientation (magnetic bearing)	18°
Raised Bed Rows	N/A
No Counting Rows	3
Planted	May 2008

This block includes three counting rows that had been counted by King Avocado on the 14th of March 2023. The trees within this block typically had much fewer fruit than block 58, and had much less dense internal foliage which made measuring the avocados significantly easier. Measuring these trees could take as little as 30 minutes whereas the trees from block 58 regularly took over 1.5 hours to measure.

Figure 3.27 – Orchard Block 49, Zone 3 [89]

3.5.2 Row Selection

Only trees from the designated counting rows of the orchard blocks were measured, which would allow a direct comparison between King Avocados manual counts and the counts resulting from the 3D measurements.

Because of the limited number of trees being measured due to time constraints, not all rows could be sampled. The counting rows that were chosen for sampling were considered based on factors such as health and size to ensure they were representative of the overall block. Rows 16 and 24 were chosen for measurement from block 58, as well as row 14 from block 49.

3.5.3 Tree Selection

To prevent any measurement bias that could result from human selection of what seemed to be the best or easiest trees to measure, a more consistent method of tree selection was specified.

It was noted by King Avocado that trees that grow at the edges of an orchard block tend to produce more fruit due to the more favourable light conditions, than the majority of the trees within the block. For this reason, the first 2 trees at the end of any row were not counted. Where possible, trees were also measured in sets of three. This would allow any overlap between avocados on neighbouring trees to be plotted and evaluated. This would provide valuable insight into how a single photograph might include avocados from two trees at once, resulting in unreliable estimates.

Based on these considerations the final sampling procedure involved measuring the 3rd, 4th, and 5th trees from a given row, skipping the next three trees, and then measuring another three trees etc, until the edge trees at the end of the row were reached. There were some cases where this procedure was interrupted and modified slightly to accommodate abnormalities within a row that were clearly not representative of the block. Healthy blocks and rows were chosen for data collection, however, there were still some trees within these blocks that had various degrees of defoliation due to mite damage. Such trees were either avoided completely or the level of damage was noted in the measurement file

and photographed. Additionally, some rows included pollinator trees which were also skipped.

In block 58, trees selected from rows 16 and 24 adhered strictly to the established selection procedure. However, in block 49, due to time constraints, only trees from row 14 were measured, leading to necessary deviations. The selection criteria began with the last tree in the row (tree 23). The two edge trees (23 and 22) were skipped, followed by measuring the subsequent three trees (21-19) as per the procedure. The next three trees (18-16) were then skipped. Due to the presence of a pollinator and an unrepresentative small tree with no fruit among the next two trees, trees 15 and 14 were skipped as well. Subsequently, trees 13, 12, and 11 were measured, and because tree 5 was also a pollinator, only trees 10 and 9 were skipped instead of the next three. The final trees in the row measured were 8, 7, and 6. Images of each tree measured in the orchard can be found in Appendix I.

3.5.4 Data Capture Procedure

The procedure for measuring a single tree was developed partially from the first trial tree in Palmerton North and was refined through trial and error as each tree was measured in the King avocado orchard.

The resulting methods were used to produce accurate 3D measurements with a consistent global origin, whilst minimising the amount of avocados missed or having repeated measurements. The measurement procedure is ideally carried out by two persons – one to operate the laptop, and another to aim and read out the laser reading of the CMD. A single person can collect data, however this would involve constant navigation from the underside of the trees to operate the laptop, which would increase the data collection times considerably.

3.5.4.1 Equipment

The equipment used for data collection can be found in Appendix K. This includes everything necessary for taking avocado measurements using the frame and tripod datum methods.

3.5.4.2 Tree photography

Two photos were taken from the sides of every tree that was measured. This is not strictly for image simulation and comparison purposes as with the first test tree, but serves as a record of the tree's features to go along with the 3D avocado positional data. Features such as the size, canopy density and tree health can be known and used alongside the positional data to evaluate potential trends between these factors and the distribution of avocados.

3.5.4.3 Frame Setup

When a tree to be measured had been identified, the first step was to set up the measurement frame around the base of the tree as shown in Figure 3.28. The frame can be separated into halves by removing its shoulder bolts with a spanner and Allen key. The two halves are then placed around the trunk of the tree and bolted back together. If the frame is unable to fit underneath the tree for any reason, a tripod must be used instead. However, this was not necessary for any of the trees measured in the orchard.

Figure 3.28 – Frame setup

Using the compass attachment, the frame is oriented to align with magnetic north for every tree. The compass is always mounted to point one on the frame to maintain a consistent coordinate system between trees, allowing their avocados to be plotted next to each other based on the row orientation.

The frame is set up as low to the ground as possible to ensure there is ample room between the mounting points for the CMD and the lowest hanging avocados on the tree. The three short frame legs can be used for this and adjusted up or down to accommodate uneven

ground. If the ground is especially uneven, the larger legs can be used for greater height adjustment. In the event that an obstacle such as a tree root or rock interferes with the placement of the frame legs, they were moved to another of the six leg mounting points on the frame.

For softer soil conditions, the large 3D-printed feet were used to prevent the frame from sinking into the soil, which could cause the coordinate system of the frame to shift between avocado measurements. Once the frame had been set up, care was taken to prevent any disturbance to the position of the frame itself or the ground supporting it.

3.5.4.4 Datum Setup

The placement of the datum points is non-critical, as long as they are placed in such a way as to minimise occlusion from branches and raised beds of orchard rows. The pegs should be placed as far apart as practical while allowing direct line of sight to measurement points. This minimises the error in remapping coordinate frames from tripod measurements. Order of placement does not matter, but once the datum points have been placed around the tree, care needs to be taken to ensure they are not moved or bumped until there are no more measurements on that tree to be taken.

(a) (b)

Figure 3.29 – Datum points placed around frame setup (a). One datum placed on the raised bed for greater tripod visibility (b)

A distance of approximately five meters between datum points was used when the row and tree conditions allowed it. In the case of the raised beds, it was found that placing one datum point on top of the row bed next to the tree (Figure 3.29), and the other datum

points off to the side of the row on the lower ground, resulted in adequate spacing where all three points were easily visible from most locations outside the tree.

3.5.4.5 Laptop Setup

The laptop was set up on a portable crate next to the tree being measured between the block rows. A small camping chair and a five-meter-long data cable allowed the user to sit comfortably away from the canopy of the tree while controlling the CMD through the laptop. From there the CMD is connected to the laptop over USB and the corresponding serial port is opened using the Python data collection GUI.

Figure 3.30 – Laptop setup with camping chair and crate to allow the user to record readings

3.5.4.6 File Creation

Before collecting any data from a tree a new text file was created that belongs to that tree where all subsequent coordinate measurements will be saved.

These files were created in the GUI based on a naming convention that includes the zone, block, and tree number (e.g. “tree Z2B58R16_T3.txt”), which ensures the 3D data can be traced back to the correct tree.

A description was also added to the text file upon creation that included the block features such as spacing and which counting row the tree belonged to. Additional descriptions included details like mite damage to the tree, its size and overall health. Any details about the tree that make it differ from the rest of the block were noted, in case of potential correlations to the positional data of the measured fruit.

3.5.4.7 Aiming at Target Points

The procedure for aiming at target points (avocados and datum points) was standardised where possible to maintain consistency throughout the data collection process.

When saving coordinates from the GUI it is important not to move the CMD during this process, as it can result in an incorrect coordinate reading or a read error. To prevent this, the laptop user would communicate to the CMD aiming person to let them know when a point had finished saving so they could then move on to aiming at the next point.

3.5.4.7.1 Avocados

Where possible, avocados were measured from the centre of the fruit at the lowest point as shown in Figure 3.31. By maintaining a consistent measurement point for all avocados, they will be more accurately recorded relative to one another and an offset could also be applied if desired to plot the avocados closer to their true position based on an average avocado size.

Figure 3.31 – Laser measuring the centre of the bottom of an avocado

3.5.4.7.2 Datum points

A procedure was also developed for measuring datum points in an attempt to record points as close to the centre of the datum as possible. This was done by first aiming the laser at the stem of the datum point to find the correct horizontal position of the CMD. The laser was then aimed upwards until the laser dot was positioned vertically central on the datum point sphere. When possible the user could position their eye directly behind the CMD

while aiming at the target datum point. This resulted in the reflection of the laser point being perceived as brightest when the laser was pointed directly at the centre of the sphere (the red light from the laser is reflected directly back to the CMD when aimed at the centre), serving as a useful indicator for aiming.

Figure 3.32 – Aiming the laser dot at the centre of the datum point for an accurate measurement

A pre-set offset is applied in the GUI code when taking datum point measurements which shifts the measured point from the surface of the datum sphere to its centre.

3.5.4.7.3 Reading laser

Often the laser measuring device would produce fluctuation distance readings in brightly lit conditions or when target avocados moved in the wind. These errors could be minimised by shading the CMD and physically holding target avocados to prevent movement. Any remaining fluctuation necessitates user estimation for an average laser reading.

3.5.4.8 First Measurements

Once the frame, datum points and tree text file have been set up accordingly, the first coordinate measurements can be recorded. Before taking any other measurements, an initial datum measurement needs to be recorded from a frame mounting point. These first datum measurements were used to align the coordinate systems of any subsequent tripod measurement positions to the original frame coordinate system. This was necessary for aligning all measured points with the standardised global coordinate system of the tree that is based on magnetic north and a level X/Y plane.

3.5.4.9 Taking Frame Measurements

To take a measurement from the mounting frame, the CMD is positioned on any of the eight mounting points and secured in place using thumb screws (Figure 3.33).

Figure 3.33 – CMD mounted on the frame with thumb screws

The mounting point of the CMD was selected in the GUI so that the coordinate could be saved along with its origin point. This is important for remapping frame measurements to have an origin at the centre of the frame (the centre of the trunk).

The CMD was manually aimed at the target object and the laser distance reading was relayed to the laptop user to save the coordinate from the specified frame position.

(a) (b)

Figure 3.34 – Frame mounting CMD taking avocado coordinate reading (a). Aiming CMD underneath low tree canopy (b)

Aiming the CMD from the frame was difficult when working on smaller trees (Figure 3.34(b)). Branches and leaves preventing direct LOS to avocados would also have to be held out of the way where possible, or an external tripod measurement point would have to be used when none of the eight frame mounting points would suffice.

3.5.4.10 Taking Tripod Measurements

Tripod measurements were used when certain avocados could not be seen from the frame. Often these were avocados higher up within the tree or on the outer branches. The tripod can be set up anywhere from where the device has a line of sight to the three datum points and the target avocado. Whenever a tripod measurement point is set up, the first three recorded points need to be datum points in the correct order. 'Datum Point' should be selected as the measurement type in the GUI and the datum points should be measured in order of Gold, Silver, and Black and the points should be named exactly so. The naming of the measured datum points is critical for the remapping post-processing of the tree files, and the naming convention should be observed when taking measurements. However, mistakes can easily be corrected by editing the text file prior to the remapping process, without any data loss.

When saving measured points from a tripod, their origin needs to be set to 'custom and manually input offset coordinates from the frame centre need to be specified. This offset is not used for positional re-mapping (the datum point coordinates achieve this) but can be used as an informative reference. The custom offset, however, must be identical for each point measured from the same tripod position. The custom offset is estimated by measuring the distance along the global X, Y, and Z axis between the centre of the tree trunk to the tripod mounting point.

Figure 3.35 – CMD mounted on a tripod next to the tree row.

Once the datum points have been measured, the measurement type can be set to ‘Avocado’ in the GUI and any number of avocados can be measured from the same tripod mounting point. If the tripod needs to be moved to measure more avocados, the setup process of recording the initial three datum points needs to be repeated from the new tripod position. This can be done as many times as required to measure all the avocados on the tree.

3.5.4.11 Branch labelling

After the initial single tree trial, it was apparent that a method for identifying which avocados had been measured and which avocados had not was necessary to prevent overcounting or missing certain avocados.

Initially, the solution for this was to apply a sticker to every avocado after it had been measured as shown in Figure 3.36(a). This was very time-consuming and impractical for higher-growing avocados that could not be reached.

(a) (b)

Figure 3.36 – Avocado labelling with stickers (a). The final method of labelling only using ribbon to mark branches (b)

The final method used was a process of labelling the branch that avocados belonged to. Avocados on outer branches were measured first and their branch was marked with ribbon tape (Figure 3.36(b)) to indicate all avocados propagating from that branch had been measured. This was repeated for all branches down to the main trunk of the tree until there were no more unmarked branches.

3.5.4.12 Data Processing

The relative positional data from the CMD collected in the field was written to text files. This was then processed later by remapping frame points and aligning datum points if tripod measurements were taken and written to a new file. Because the raw data was left unchanged there is no possibility of data loss from the remapping process.

The data collected at the orchard was periodically processed with the remapping Python code to check that the measured points (datum and avocado) were not producing any errors and could be plotted correctly. This was done at the end of each day to ensure everything was working as expected.

From there, the processed data can be plotted, evaluated for trends, and photographed (simulated) from any angle to optimise an image-capturing procedure for applying the AI counting model. The results of this analysis are described in the following results and discussion chapter.

CHAPTER 4
RESULTS AND DISCUSSION

This section presents the results of the avocado tree measurements conducted over five days in the King Avocado orchard. The data collection process, which aimed to evaluate the spatial distribution of avocados within selected trees, faced some challenges related to weather conditions and technical issues. Despite these challenges, a total of 27 trees were successfully measured, providing a dataset comprising 2197 avocados from block 58 and 712 avocados from block 49. The average number of fruit per tree across both blocks was 108 avocados which took an average of 90 minutes to measure. Subsequently, the spatial distribution of avocados is discussed, examining patterns and implications for optimizing image-capturing procedures as well as discussing potential applications where this unique dataset may be useful beyond the scope of yield estimation for commercial avocado growers.

4.1 Count Comparison

The manual counts for the 27 measured trees, conducted by Claudia Hermosilla from King Avocado, were carried out on the 1st of March for block 58 and on the 14th of March for block 49. These manual counts were compared with the counts resulting from the 3D positional data collection in mid-September. Due to the meticulous marking methods and the extended time devoted to measuring each tree during the 3D measuring process, the resulting counts can be considered as precise as practically achievable.

Figure 4.1 illustrates a comparison between counts from positional data collection and manual counts by King Avocado, revealing a rise in deviation as the number of avocados on the tree increases. This is attributed to challenges in maintaining diligence and accuracy during manual counts of larger quantities. Consequently, the trees with higher fruit counts contribute significant error to the overall manual yield estimation across an entire block, resulting in increased expected error when the average fruit per tree count rises. Detailed count datasets for both methods are available in Appendix H.

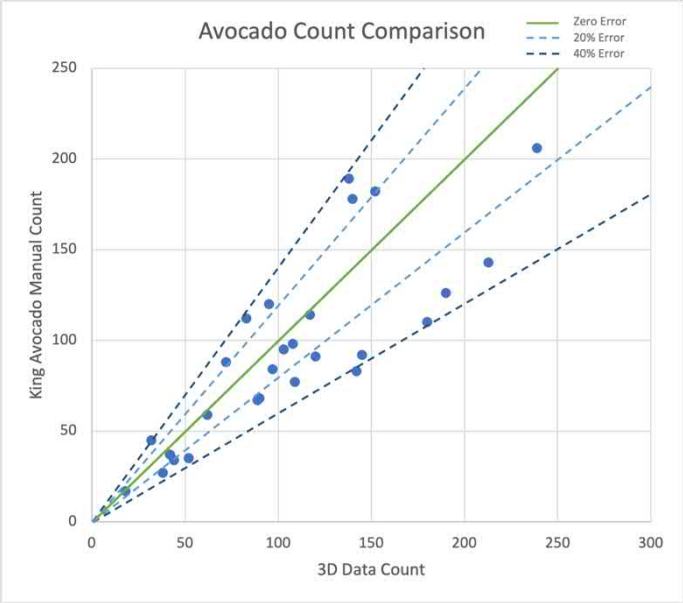


Figure 4.1 – Orchard vs 3D Data Count Comparison

The analysis of the 27 measured trees revealed a maximum error of 41% for overestimated counts and -42% for underestimated counts. The average fruit per tree count of the manual counts by King Avocado and the actual count showed an average error of -11% across all measured trees, indicating a tendency to undercount the number of avocados on a tree. As the average error takes into account both undercounted and overcounted trees, its consistency cannot be presumed across all manual counts. The absolute average error between manual and actual counts, at 24%, could lead to variations in the overall average error of a count depending on the extent to which upper and lower errors offset each other.

This discrepancy further highlights the necessity for a more accurate method of yield estimation at the individual tree level. With this level of error evident in a tree-by-tree comparison before extrapolation across an entire block, it becomes clear that substantial inaccuracies can be introduced into overall yield estimations for an orchard, potentially leading to misguided commercial planning decisions.

4.2 Fruit Overlap

Between the neighbouring trees that were measured, there were often cases where avocados would be closer to a neighbouring tree than their own tree. Such observations

bear significance for the optimisation of image capture procedures aimed at obtaining accurate yield estimates for individual trees, while avoiding the inadvertent inclusion of avocados stemming from neighbouring trees. An alternative methodology could involve considering overlapping fruit instances from an adjacent tree as attributable to the primary tree, an approach that would rely on the FOV of the captured images to extend only to the space halfway between the two trees based on a known block spacing.

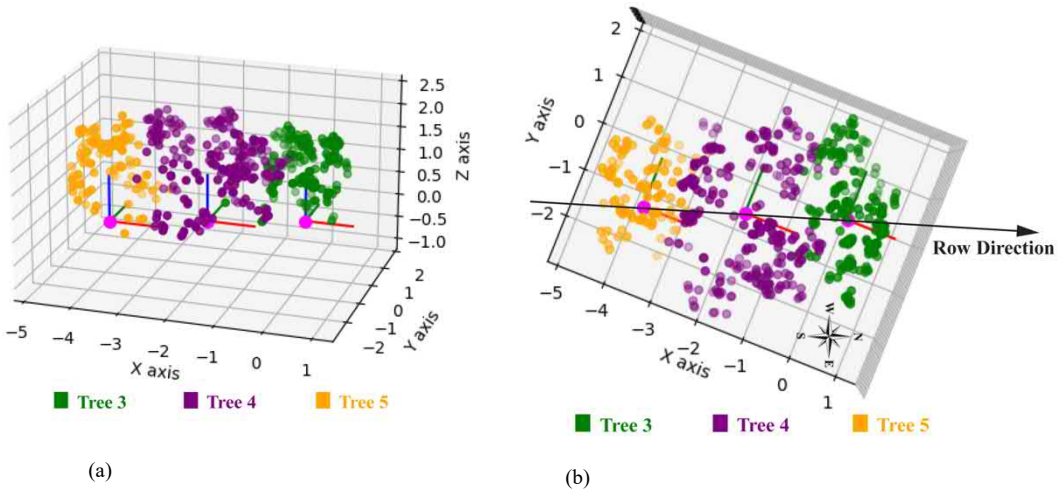
4.2.1 Neighbouring Trees

In assessing fruit overlap between neighbouring trees, the avocado groups from the measured trees were systematically plotted. This involved applying a coordinated X/Y offset, factoring in both block spacing and a magnetic declination of 18 degrees, to account for the row alignment with true north. For block 58, with trees spaced 2m apart, the applied offset between adjacent trees on the same row was X minus 1.9m and Y minus 0.62m. Similarly, for block 49, with trees spaced 3.5m apart, the offset was X minus 3.33m and Y minus 1.08m. The need for this offsetting procedure arises from the fact that the measured coordinate systems are aligned with magnetic north rather than the orchard rows. By aligning them based on spacing and declination, we ensure an accurate representation of the spatial distribution and potential overlap between avocados from neighbouring trees within the orchard blocks.

A total of nine sets of three adjacent trees were measured and plotted, including six sets from block 58 and three sets from block 49. This direct visualization offers insights into the actual tree-by-tree overlap across both blocks.

4.2.1.1 Trees 3-5 (Row 16, Block 58)

The initial set of three adjacent trees from block 58 (Figure 4.2) reveals a moderate amount of fruit overlap between trees, characterized by fruit skewing noticeably outwards in the direction perpendicular to the row. This occurrence is likely attributed to branches from neighbouring trees restricting each other's extension along the row due to the relatively small 2m spacing.



(a) (b)

Figure 4.2 – Z2B58R16 trees 3,4,5 side view (a), and bird's eye view (b)

Additional plots used to evaluate actual fruit overlap for the eight remaining sets of adjacent trees are available in Appendix J.

4.2.1.2 Trees 9-11 (Row 16, Block 58)

Trees 9-11 offer limited insight into overlap characteristics, mainly because of the abnormality of the small tree 10. However, fruit from the other two trees is observed to extend much further along the rows, filling the vacant space left by tree 10. This observation suggests that the spacing might be too narrow for optimized growth. If trees had more space, they might naturally extend and produce more fruit along the row direction.

4.2.1.3 Trees 15-17 (Row 16, Block 58)

In this group, there is minimal overlap between trees 15 and 16, but a considerable overlap between trees 16 and 17. This is likely attributed to tree 16 having more space to extend its branches, encroaching over the smaller tree 17. This once again highlights
a natural tendency for tightly spaced trees to extend beyond the halfway point between adjacent trees when given the opportunity.

4.2.1.4 Trees 3-5 (Row 24, Block 58)

Trees 3-5 exhibit a relatively clean segregation of fruit between the trees, occurring at approximately the halfway point between them. This pattern remains consistent, even for the smaller tree 3 in this instance.

4.2.1.5 Trees 9-11 (Row 24, Block 58)

This set of trees includes two smaller trees, 11 and 10, with no observable overlap between them. Tree 10 exhibits a distinct skewing perpendicular to the row direction, particularly noticeable in some far outer fruit in the Y direction. Additionally, the larger Tree 9 shows a minor amount of fruit overlapping into the space of Tree 10.

4.2.1.6 Trees 15-17 (Row 24, Block 58)

These trees also exhibit similar skewing to previous groups, although there is minimal overlap except for a few fruit between trees 16 and 17. The skewed fruit distribution with minimal overlap is likely caused by interference of outer branches between trees, which is not immediately visible in plots due to a lack of fruit on the outer extremities of these branches.

4.2.1.7 Trees 21-19 (Row 14, Block 49)

The initial three trees from block 49 exhibit no fruit overlap. This is likely attributed to the larger tree spacing of 3.5m, as well as the relatively small size of the trees in this particular group.

4.2.1.8 Trees 13-11 (Row 14, Block 49)

This group similarly displays minimal to no fruit overlap, showcasing a more consistent circular fruit distribution for each tree.

4.2.1.9 Trees 8-6 (Row 14, Block 49)

The last group from block 49 exhibits minimal fruit overlap between trees and a relatively even circular distribution around trees 6 and 8. In contrast, tree 7 displays an unusual skewing along the direction of the row. This set of trees was also noticeably narrow along the perpendicular row direction, a characteristic not observed in other sets in either block.

This could potentially be attributed to larger trees from adjacent rows hindering the extension of branches further away from the row.

The implications of the observed tendency for trees to grow unevenly skewed perpendicular to the row direction are significant, especially in small-spacing blocks. In such cases, a photography procedure is likely to involve a total Field of View (FOV) that extends farther out to the sides of each row than it does between the trees. This is in contrast to trees from block 49, which exhibit a more even fruit distribution, making them more suitable for a rotationally symmetrical image-capturing procedure.

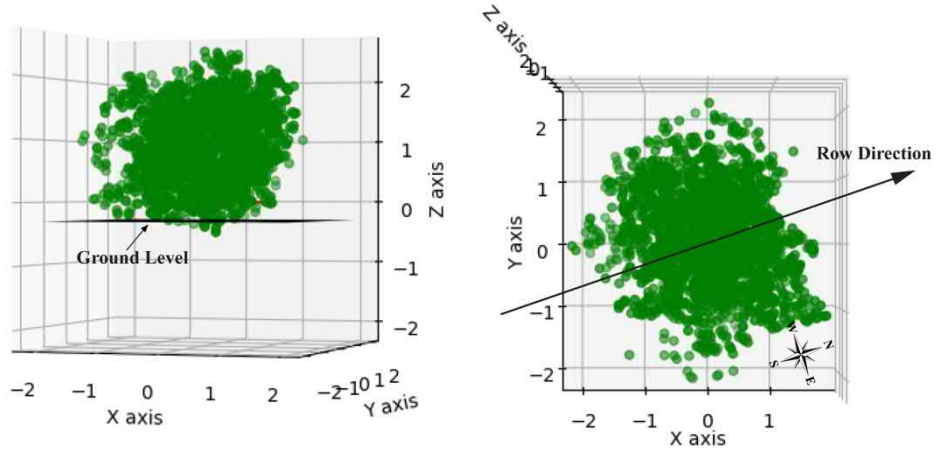
Observations regarding overlapping fruit reveal the potential for notable overlap, especially in closer-spaced trees. These conditions must be carefully considered when devising a photography procedure if accurate individual tree counts are required. Alternatively, when the focus is solely on accurate counts across an entire block, methods for attributing overlapping fruit to the nearest tree could be implemented.

4.2.2 Combined Plots

An alternative evaluation method is to combine the 3D avocado data from all measured trees into one plot with the same origin. This approach can be utilized on a block-by-block basis to assess the potential overlap of all measured fruit from the origin and half the tree spacing along the block rows. However, the drawback of this method is that it does not consider the overlap of fruit relative to adjacent tree branch conditions; instead, it examines all overlapping fruit regardless of neighbouring trees.

4.2.2.1 Block 58

Figure 4.3 depicts all measured avocados from block 58 combined into a single plot with the same origin. These plots reveal a roughly symmetrical spherical distribution of fruit.



(a) (b)

Figure 4.3 – All avocados from block 58 plotted together. Side view (a). Bird's eye view (b)

The overlapping fruit within this combined plot is identified by a horizontal deviation, measuring the distance in the direction of the rows from the centre of the origin, exceeding half of the tree spacing (1m for block 58). Python code was written to quantify this overlap, accounting for the magnetic declination, to identify all points that fall outside these specified conditions and calculates the percentage of overlapping fruit for each block.

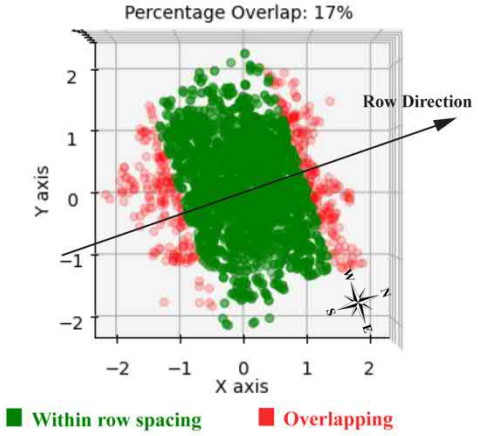


Figure 4.4 – Fruit within adjacent tree space in block 58 (bird's eye view)

In Figure 4.4, the fruit distribution reveals overlapping fruit highlighted in red. Approximately 17% of all avocados in block 58 extend beyond the designated tree spacing, with some instances of overlap exceeding 1m.

4.2.2.2 Block 49

All fruit from the trees measured in block 49 was also plotted similarly, as depicted in Figure 4.5. These plots reveal a relatively uniform distribution of fruit, characterized by more horizontal spread in relation to the height when compared to block 58.

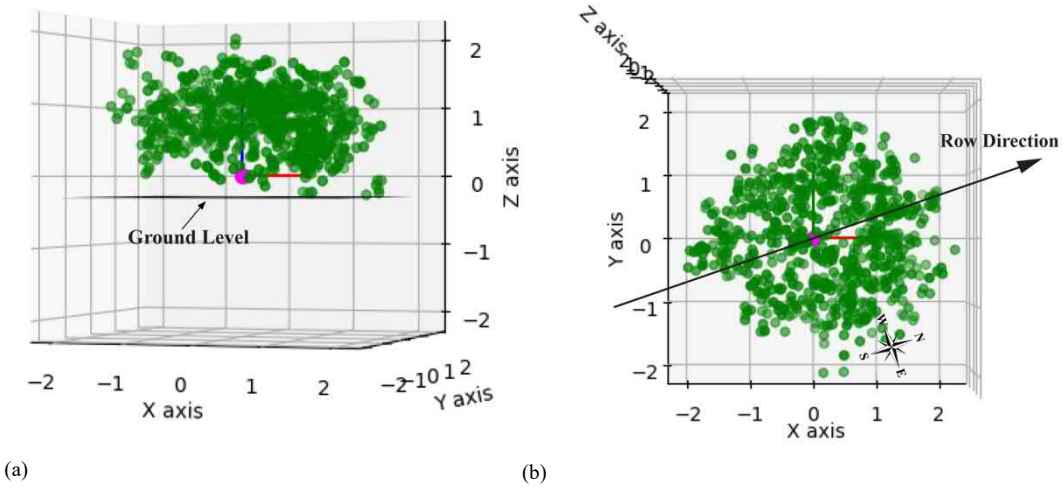


Figure 4.5 – All avocados from block 49 plotted together. Side view (a). Bird's eye view (b)

The overlapping fruit was identified based on a horizontal deviation of 1.75m for block 49, aligning with the direction of the rows. This criteria resulted in a considerably lower percentage of overlapping fruit, amounting to only 4%.

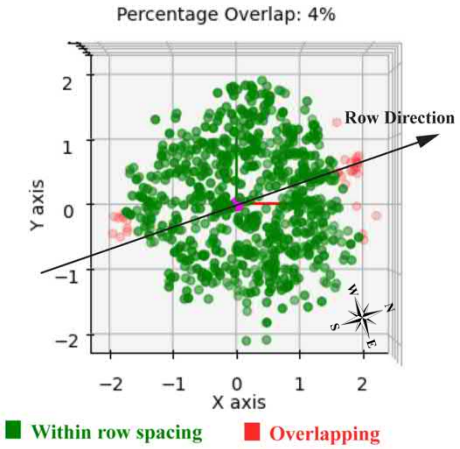


Figure 4.6 – Fruit within adjacent tree space in block 49 (bird's eye view)

The consolidated plots underscore the potential for error arising from fruit overlap into the space of adjacent trees. The mere presence of overlapping fruit could introduce up to 17% error in individual tree yield estimates when using photographs processed by AI. To address this concern during the optimization of image capture for AI processing, strategies such as applying correction factors or employing more precise methods to attribute overlapping avocados to adjacent trees should be considered. In any case, it is clear that a photography procedure should encompass images that avoid crossing the central perpendicular line between trees in a row within their field of view.

4.3 Spatial Distribution

The spatial distribution of fruit is crucial in determining optimal image capture locations and angles for an effective count. Frequency plots provide a valuable tool for assessing the occurrence of measured avocados at specific spatial points around the tree. By examining upper and lower percentiles, one can identify acceptable boundaries and exclude extremes to achieve a target accuracy in yield estimation. This approach allows for a more informed strategy in selecting viewpoints for image capture, ensuring a comprehensive representation of all fruit on the trees.

Plots were generated by assessing the deviation along all three coordinate axes from the origin point, employing data bins with a width of 200mm. To aid in data visualization, the plots incorporate a Kernel Density Estimate (KDE), which produces a continuous probability density curve. Additionally, key percentiles, including the 1st, 5th, 25th, 50th, 75th, 95th, and 99th, are depicted on the plots. These percentiles aid in identifying the mean position, as well as the outermost positions of the fruit on the trees, providing a comprehensive understanding of their spatial distribution from which inferences can be made pertaining to image-capturing procedures as well as potential insight into better growing practices.

4.3.1 Row Direction

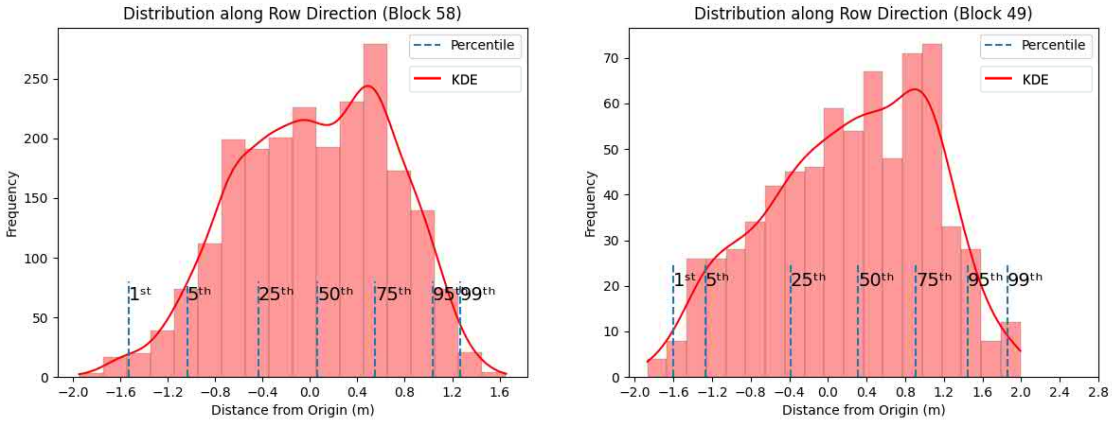


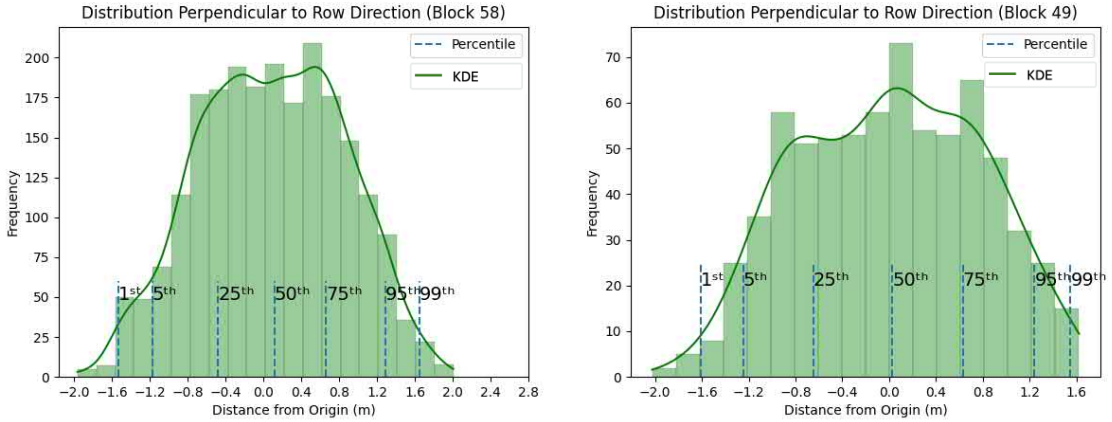
Figure 4.7 – Distribution along the direction of rows of all fruit belonging to block 58 (a), and block 49 (b)

The distribution of measured fruit along the row direction exhibits a Gaussian-like pattern for both blocks. In the case of block 58, with a 5 by 2m spacing, 98% of the fruit lies within 1.2m in any direction along the row axis. For block 49, with a 3.5 by 3.5m spacing, the same majority is positioned within 1.8m. Despite the additional 1.5m spacing between trees in block 49, compared with block 58, the increased spread of the majority of fruit along the row axis is only marginal. However, these cut-off distances for the 98% majority exceed half the spacing between each tree for both blocks, a factor previously discussed as the ideal field of view (FOV) limits for image capturing. This is crucial to avoid overlapping into the shared space between two trees, thereby mitigating the risk of duplicative fruit counting and ensuring an accurate yield estimate. Hence, an image-capturing procedure should exclude images capturing avocados at these outer limits, adhering instead to the boundaries set by the row spacing.

An intriguing observation is that the mean (50^{th} percentile) of the distribution is not zero but approximately 0.3m towards the north-facing side for block 49 and approximately 0.08m for block 58. This deviation is evident in the skew of the bell curve towards the right in both plots. This phenomenon holds significance in exploring more optimal growing conditions, as it indicates increased yield on the north-facing side of trees. Exposure to the sun is a known contributor to the postharvest quality of Hass avocados [91, 92], the development of flowers and fruit set [93, 94], and, according to Claudia Hermosillo from King Avocado, it also has a noticeable impact on yield (observed from

trees growing on slopes allowing full exposure of the trees to the sun). The likely cause of the skewed distribution along the row direction is the enhanced sun exposure of northern-facing fruit and foliage in an orchard planted in the southern hemisphere.

4.3.2 Perpendicular to Row Direction



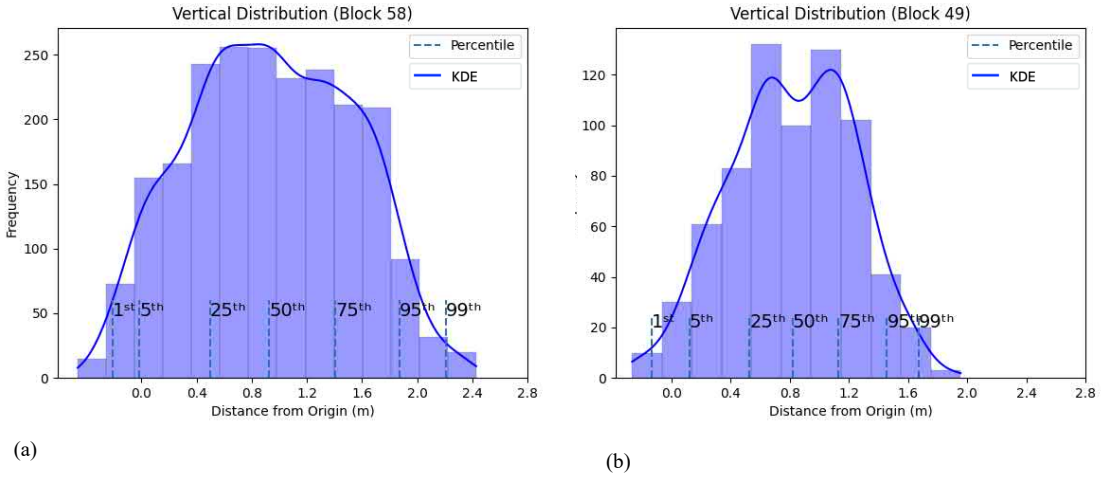
(a) (b)

Figure 4.8 – Distribution along the Y-axis of all fruit belonging to block 58 (a), and block 49 (b)

The fruit distribution along the direction perpendicular to the rows across both blocks similarly exhibits a Gaussian-like pattern, with the 1st and 99th percentiles closely situated within 1.6m. This suggests that an image-capturing procedure incorporating an FOV extending up to 1.6m in either direction from the tree's trunk, perpendicular to the row direction, will encompass over 98% of the fruit. This holds true for both orchard blocks.

Contrary to the previous plots for the row direction, the mean or 50th percentile for both perpendicular direction plots approximates the zero position with more symmetrical distributions from the trunk of the trees. This aligns with the hypothesis correlating sunlight exposure and yield, as either side of the tree should receive a roughly equal duration of sunlight due to the sun rising on the right side of the rows and setting on the left side of the rows.

4.3.3 Vertical



(a) (b)

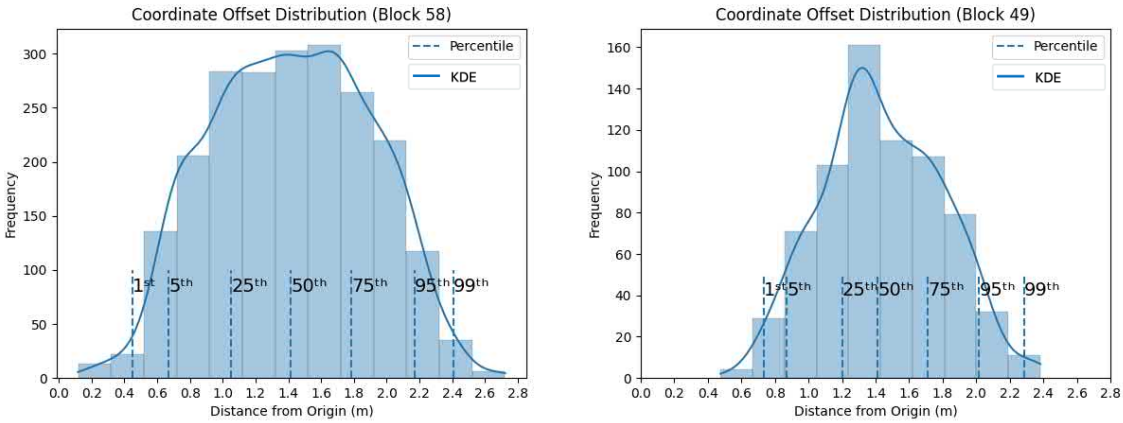
Figure 4.9 – Distribution along the vertical axis of all fruit belonging to Block 58 (a), and Block 49 (b)

Along the vertical axis, the mean height of fruit on the trees is approximately 0.8m, with 98% growing between -0.2m and 2.2m for block 58. This same lower range applies to block 49 but with a lower upper limit of 1.8m. Negative height values result from the measurement frame's height defining the tree's coordinate system, approximately 0.3m above the ground. In some cases, trees were grown on mounds, allowing the recording of lower-hanging fruit even further down.

Based on this information, an all-encompassing photography procedure would need to accommodate vertical variation of no more than 2.4m from ground level.

4.3.4 Distance from Origin

An additional frequency plot was generated for the calculated distance of each measured point from the origin. This analysis integrates all three measurement axes, offering a more comprehensive 3D insight into the offset distance of each fruit from the tree trunk. It establishes the necessary limits for a spherical image capturing FOV from the origin of the tree.



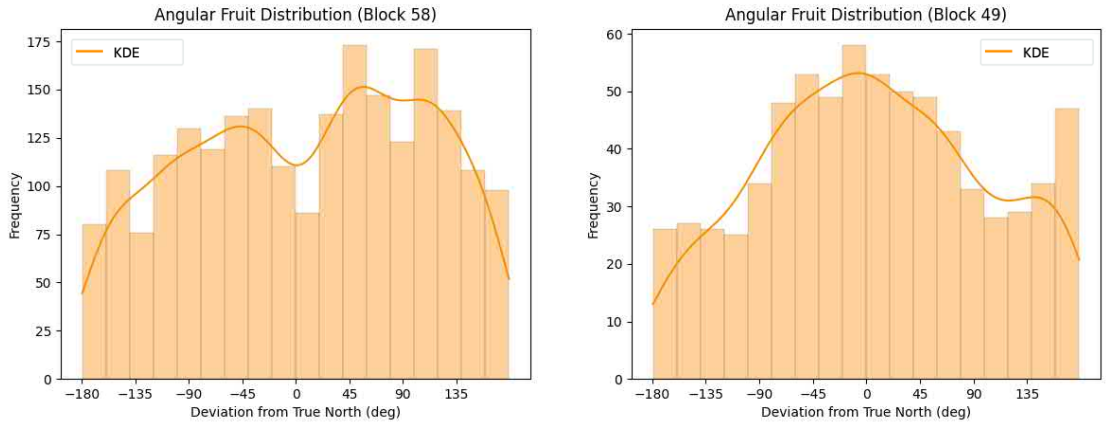
(a) (b)

Figure 4.10 – Distribution of origin offset distance of all fruit belonging to block 58 (a), and block 49 (b)

Results from these plots indicate a mean distance from the tree origin of 1.4m for both orchard blocks. The upper limits encompassing 98% of fruit ranged from 0.4m to 2.4m for block 58 and between 0.7m and 2.3m for block 49. During the data collection process, some avocados were observed to be extremely close to the frame, and in some cases, inside the frame itself. However, upon analysis of these plots, it is evident that this extremely close fruit falls within the first percentile and can likely be excluded in an optimized photography procedure.

4.3.5 Angular

Angular distribution plots show the deviation of the measured fruit from true north. The measurement is expressed as the angle formed by the fruit on a horizontal plane relative to a north-south line, measured counter-clockwise from above.



(a) (b)

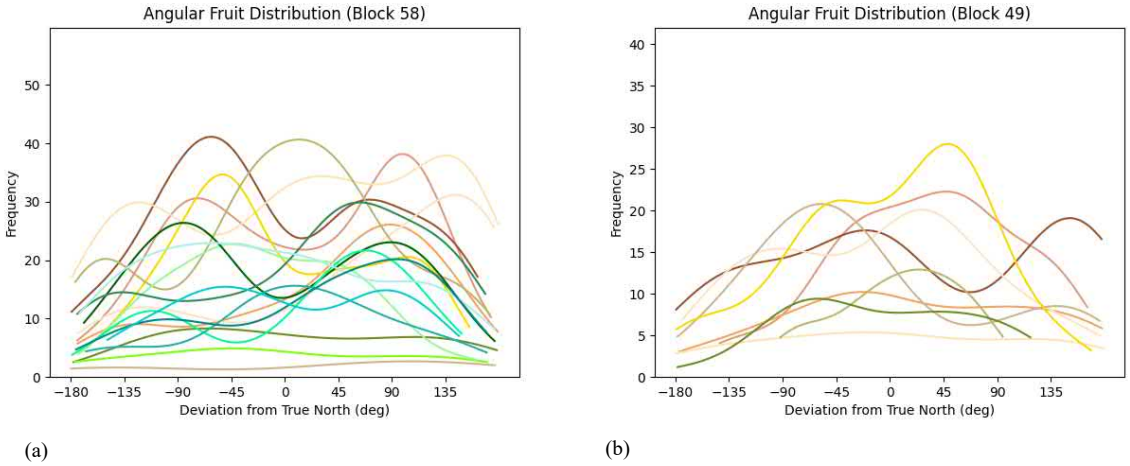
Figure 4.11 – Angular distribution of fruit from block 58(a), and block 49(b)

These figures further emphasize the pattern of increased fruit growth on the Northern side of the tree due to sun exposure, particularly evident in block 49 as depicted in Figure 4.11(b) by a prominent KDE curve peaking at zero degrees (North). The plot for block 58 also reveals a bimodal curve rising towards the zero-degree point with a noticeable dip at the top. This abrupt decrease in fruit frequency at the zero-degree point is likely attributed to the tight tree spacing of 2 m along the row direction. This arrangement causes the northernmost areas of the trees to be shaded from direct sunlight by the tree planted to the north, while the areas just 45 degrees to either side still receive adequate light for more optimal flower development and fruit set.

These findings suggest that a 2 m spacing between trees along the north-south direction may not be permitting adequate light to reach the northern face of trees. This could potentially lead to a suboptimal ratio of planting area to yield in the block.

4.3.6 Variation between Trees

The subsequent plots illustrate the angular distribution of fruit for all 27 individual trees measured across both blocks. While aggregating these individual trees has offered insights into the overall average fruit distribution, it is pertinent to acknowledge the variations across individual trees. As depicted in Figure 4.12, significant disparities in the angular spatial distribution of fruit are evident among different trees, encompassing not only differences in total per-tree yield but also variations in the locations of peak angles.



(a) (b)

Figure 4.12 – KDE curves for angular distributions of each individual tree. Block 58 (a), and block 49 (b)

Significant variation among individual trees persists across all spatial distribution axes, similar to the observed angular distribution, as demonstrated in the supplementary plots found in Appendix L.

4.4 Discussion

The spatial distribution findings reveal key insights into the arrangement of avocados on trees within the orchard blocks. The analysis of the fruit distribution along the row direction highlights a Gaussian-like pattern, indicating that the majority of fruit is concentrated within a certain distance along the row axis. However, the cut-off distances for the 98% majority exceed half the spacing between trees, emphasizing the need for careful consideration in image-capturing procedures to avoid overlapping into the space between adjacent trees.

Based on the data collected from the King Avocado orchard, a procedure that includes an FOV out to a 2.4 m radius from the trunk and does not include fruit that grow more than half the spacing between two trees will be capable of providing sufficient images to include 98% of the fruit on the tree. This assumes the procedure avoids self-intersecting photographs leading to duplicate counts and ensures no occlusion of fruit within the captured images. This inclusive FOV accommodates the range of trees observed in the two orchard blocks, each with their respective tree spacings, which were pruned to maintain a tree height below 3 m. While the required FOV for an optimal photography procedure may vary based on tree height and spacing in a specific orchard block, the findings provided here are based on commonly accepted optimal growing conditions expected to persist in future practices, ensuring broad relevance across commercial blocks.

4.4.1 Occlusion

While a comprehensive positional dataset was able to be collected manually, occlusion remains a challenge for future image-capturing procedures using AI for yield estimation. Occlusion may lead to undercounting, but this could potentially be mitigated by employing a multi-directional photography approach. This involves targeting the same points in space (avocados) multiple times, either by tracking the fruit between frames with image processing algorithms or by using a correction factor to account for duplicates.

However, this approach necessitates additional research, testing, and more advanced data collection procedures for individual trees. Simplicity and ease in image capturing are crucial considerations to ensure costs remain less than the value of an accurate yield estimation in an orchard block.

A simpler approach to address the problem of occlusion could involve deriving a correction factor from experimental trials. In these trials, an optimal image-capturing procedure would be applied across several trees in a block, while manual fruit counts are recorded simultaneously using a thorough counting and marking procedure, similar to those outlined in the methodology section used to collect positional data. Images of the trees are then processed using the previously developed AI model [7] to obtain a total fruit count per tree. The resulting AI counts could then be compared to the actual manual counts, and a correction derived to adjust for the average variation between the two methods. Due to considerable variation of shape and size between avocado trees, developing a robust correction factor may require a significant sample of trees to accurately extrapolate counts across an entire block.

4.4.2 FOV Overlap

Overlap between images captured during a photography procedure also needs to be considered. All avocados within a 2.4 m radius from the trunk, perpendicular to the rows and within the tree spacing, should be encompassed within the overall field of view (FOV) of the images. However, the presence of overlapping FOV may lead to duplicated avocados and subsequent overcounting.

Preventing this overlap while still capturing all avocados belonging to individual trees is a challenge that will need to be addressed through the simulation of camera angles based on the positional dataset of avocados collected in this research. Optimised solutions might involve taking photographs radially outward from the trunk at precise angles to align the edges of their FOV while incorporating cameras with a limited depth of field to blur avocados beyond a specified range. However, the impact of blurred avocados on images processed by counting AI [7] is unknown and requires further investigation to determine the feasibility of this approach.

4.4.3 Photography Device

After developing an optimal photography procedure, either through simulation or real-world trials, a practical device could be designed to streamline the process of photographing individual trees. This might involve a handheld frame positioned beneath the canopy of avocado trees, equipped with multiple cameras to capture images from predetermined angles simultaneously.

4.4.4 Block Sampling

Alongside the development of accurate and efficient individual tree counting methods, there is likely a need for the extrapolation of counts across entire blocks, as growers typically operate on a block basis rather than focusing solely on individual trees. This need arises unless a rapid image-capturing procedure can be devised to make acquiring counts for all trees in a block practical. This could involve methods such as cameras mounted on vehicles traversing between rows or autonomous ground/airborne robots programmed to capture images from specific angles.

Individual tree counts could be incorporated into traditional block yield estimation methods, involving counting predetermined rows in a block and extrapolating those counts across the entire block. While simple to implement, manual yield estimates using these techniques are known to be inaccurate [4, 8, 9], partly due to imperfect manual tree counts and preselected counting rows that may not represent the rest of the block accurately. An AI-based individual counting method has the potential to eliminate the error from imperfect manual counts, and further trials may reveal that the error from row extrapolation becomes insignificant, rendering this technique more feasible.

Another method of extrapolating individual tree counts across a block or orchard is a satellite imagery-based sampling technique developed by the University of New England [4, 21-23], as discussed in the literature review section. This technology compares the canopy health of trees in a block to determine the optimal trees to be selected for counting and extrapolation across the rest of the block. With a claimed yield estimation accuracy of 93% that still relies on manual counting of individual trees, this technology could likely be implemented alongside an AI-based counting solution to achieve yield estimates of much higher accuracy.

4.4.5 Further Applications of Data

The applications of this dataset extend beyond identifying an optimal, highly accurate image-capturing procedure, offering quantifiable means for testing alternative experimental photography methods. Simulations based on the dataset could also provide insights into the accuracy of a single photograph taken from within the canopy, as well as determine a correction factor necessary to convert AI counts into usable yield estimates. This has the potential to lead to an exceptionally rapid photography procedure that can be manually implemented, requiring minimal or no extrapolation across the rest of the block.

In addition to its intended use in optimizing image capture for avocado trees, this unique 3D positional data has the potential to contribute to various aspects of commercial agriculture development beyond yield estimation. Preliminary investigations into the spatial distribution of avocados uncovered a correlation between yield and sun exposure, influenced by shading from neighbouring trees and the row orientation of planted blocks. These findings highlight the potential for the data to assist growers in making informed decisions about different growing practices.

The creation of a digital twin based on the collected positional orchard data could open avenues for various applications in research [95-97]. A digital recreation of the orchard can be utilised for advanced simulations and modelling to optimise orchard management practices. For instance, a digital twin could be employed to simulate the impact of different environmental conditions, such as sunlight exposure or variations in tree spacing, on avocado growth and yield distribution. Additionally, it provides a virtual platform for testing and refining image-capturing procedures, helping to advance AI-based counting methods as well as enabling researchers and growers to explore and analyse the orchard's spatial characteristics without physical presence.

CHAPTER 5
CONCLUSIONS AND RECOMMENDATIONS

5.1 Summary

This thesis addresses the critical need for accurate yield estimation in commercial avocado orchards, leveraging insights from previous research and developed AI technology for avocado counting within visual spectrum photographs [7]. Identifying a knowledge gap in the spatial distribution of avocado fruit, the research focuses on collecting 3D positional data of avocados during their growth, essential for refining image capture procedures within the canopy and guiding future yield estimation research.

The methodology introduces a Coordinate Mapping Device capable of accurate 3D positional data collection using a laser measurement device. A mounting frame establishes a known coordinate system around avocado trees, enabling the evaluation of positional data for individual trees and facilitating comparisons between neighbouring trees. The feasibility of the developed methodology is demonstrated through validation tests and a small-scale trial on a local avocado tree, forming the basis for simulating optimized image capture procedures.

Implementation at the King Avocado Orchard yields a unique 3D dataset of over 2,909 measured avocados from 27 trees across two Hass avocado blocks. Results showcase trends in fruit distribution along the vertical, row, and perpendicular row directions. Gaussian-like distributions along the horizontal plane reveal insights, with 98% of fruit concentrated within a 2.5 m distance from the tree trunk's centre at ground level. Considerable fruit overlap between adjacent trees underscores the need for an optimal image-capturing procedure, suggesting a field of view within half the tree spacing in the row direction from each tree trunk. This approach attributes overlapping fruit to the nearest tree, ensuring 98% inclusion in the photography field of view.

The dataset and findings contribute to future research in optimizing image-capturing procedures in avocado orchards, facilitating machine learning applications for visual spectrum photography yield estimation. The thesis also explores extrapolation methods for individual tree counts across entire orchard blocks, considering the integration of

satellite imagery-based sampling technology, Crop Count [21-23], as a complementary yield estimation tool alongside the discussed individual tree counting method.

5.2 Recommendations

Based on the dataset collected in this research, there is a need for further simulations based on the positional data. These simulations would involve the creation of theoretical cameras and renders of digitized fruit on individual trees, adhering to experimental image-capturing procedures derived from the recommended spatial field of view. The objective is to develop an optimized photography procedure that reduces the number of required photographs while minimizing overlapping fruit and ensuring the inclusion of a significant percentage of all fruit attributed to each tree. This could subsequently lead to the development of a device capable of efficiently implementing this optimal image-capturing procedure within an orchard environment. Such a device could manifest as a handheld tool deployable within the canopy by a user or an array of cameras attachable to the tree base for simultaneous image capturing.

Challenges arising from fruit occlusion remain a subject for future research to address. This would involve implementing the developed image-capturing procedure in an orchard setting, paired with manually acquired fruit counts for individual trees. By comparing AI-generated counts from captured images with manual individual tree counts, a correction factor could be derived to account for occluded fruit.

Beyond yield estimation, this dataset holds promise for optimizing the cultivation practices of commercial avocado orchards. The 3D positional data encompasses implications for growing conditions such as sunlight and tree spacing, offering insights into fruit yield variations across different areas of the trees. Importantly, the methodology outlined in this research is not limited to avocado orchards, but could also be extended to other crops where understanding the spatial distribution of fruit proves advantageous.

REFERENCES

1. Mokria, M., et al., *Fruit weight and yield estimation models for five avocado cultivars in Ethiopia*. 2022: Environmental Research Communications.

2. Avocado, N.Z. *Avocado industry continues strong growth*. 2021; Available from: https://industry.nzavocado.co.nz/stronggrowth/#:~:text=New%20Zealand%27s%20avocado%20industry%20value,in%20the%20New%20Zealand%20market.

3. Authority, T.N.Z.H.E. *Avocado*. 2020; Available from: https://www.hea.co.nz/2012-05-11-03-05-28/avocado-trade.

4. Robson, A., M.M. Rahman, and J. Muir, *Using Worldview Satellite Imagery to Map Yield in Avocado (Persea americana): A Case Study in Bundaberg, Australia*. 2017, University of New England.

5. Vasconez, J.P., et al., *Comparison of convolutional neural networks in fruit detection and counting: A comprehensive evaluation.* 2020.

6. Spalding, D.H., *Storage of Avocados.* 1976: p. 109-113.

7. Macadam, K., *Viability of Avocado Yield Quantification using Machine Vision and Machine Learning*, in *MAF Digital Labs*. 2022, Massey University.

8. Bergh, B.O. and M.J. Garber, *Avocado Yields Increased by Interplanting Different Varieties*. 1964.

9. Margetts, J., *Crop Forecasting Case Studies*. 2016, P2P Business Solutions.

10. Köhne, S., *Yield Estimation Based on Measureable Parameters.* 1985.

11. Prengaman, K. and T. Mullinax. *Computer vision systems can count apples and provide a new perspective on crop load*. 2017; Available from: https://www.goodfruit.com/computer-vision-systems-can-count-apples-and-provide-a-new-perspective-on-crop-load/#:~:text=Designed%20in%20the%20United%20Kingdom,heat%20map%20that%20shows%20the.

12. Pixofarm. *Easily count fruits in your orchard*. 2023; Available from: https://www.pixofarm.com/fruit-counting.

13. Gongal, A., M. Karkee, and S. Amatya, *Apple fruit size estimation using a 3D machine vision system*. 2018.

14. Qureshi, W.S., et al., *Machine vision for counting fruit on mango tree canopies.* 2016.

15. Anderson, N.T., et al., *Estimation of Fruit Load in Australian Mango Orchards Using Machine Vision.* 2021.

16. Baker, W.J., et al., *Automated Avocado Yield Forecasting Using Multi-Modal Imaging*. 2017.

17. Talaviya, T., et al., *Implementation of artificial intelligence in agriculture for optimisation of irrigation and application of pesticides and herbicides.* 2020.

18. Javaid, M., et al., *Understanding the potential applications of Artificial Intelligence in Agriculture Sector.* 2022.

19. He, L., et al., *Fruit yield prediction and estimation in orchards: A state-of-the-art comprehensive review for both direct and indirect methods.* 2022.

20. Linaza, M.T., et al., *Data-Driven Artificial Intelligence Applications for Sustainable Precision Agriculture.* 2021.

21. AG, G. *World-first yield forecasting technology offers avocado growers 93% accuracy*. 2023; Available from: https://www.growag.com/highlights/article/world-first-yield-forecasting-technology-offers-avocado-growers-93-accuracy.

22. Agrifutures. *Horticulture and space- based technologies*. 2023; Available from: https://agrifutures.com.au/wp-content/uploads/2021/04/21-046.pdf.

23. Circul8. *CropCount, A forecast tool farmers can count on.* 2023; Available from: https://www.circul8.com.au/our-work/crop-count/.

24. Marini, F. and J.M. Amigo, *Unsupervised exploration of hyperspectral and multispectral images.* 2019.

25. Amigo, J.M., *Hyperspectral and multispectral imaging: setting the scene.* 2019.

26. Insight, O. *Multispectral Imaging*. 2023; Available from: https://www.oceaninsight.com/knowledge-hub/measurement-techniques/imaging/#:~:text=Multispectral%20imaging%20(MSI)%20involves%20capturing,spectral%20content%20from%20that%20data.

27. Optics, E. *Hyperspectral and Multispectral Imaging*. 2023; Available from: https://www.edmundoptics.com/knowledge-center/application-notes/imaging/hyperspectral-and-multispectral-imaging/.

28. Kämper, W., et al., *Rapid Determination of Nutrient Concentrations in Hass Avocado Fruit by Vis/NIR Hyperspectral Imaging of Flesh or Skin.* 2020.

29. Chumbimune, S.Y., et al., *Methodology for Avocado (Persea Americana Mill.) Orchard Evaluation Using Different Measurement Technologies.* 2022.

30. Pinto, J., H. Rueda-Chacón, and H. Arguello, *Classification of Hass avocado (persea americana mill) in terms of its ripening via hyperspectral images.* 2019.

31. Sankaran, S., et al., *Can High-Resolution Satellite Multispectral Imagery Be Used to Phenotype Canopy Traits and Yield Potential in Field Conditions?* Transactions of the ASABE, 2021. **64**(3): p. 879-891.

32. Ye, X.-J. and K. Sakai, *Fruit Yield Estimation Through Multispectral Imaging.* 2012.

33. Gené-Mola, J., et al., *Fruit detection in an apple orchard using a mobile terrestrial laser scanner*, in *Biosystems Engineering*. 2019.

34. Debnath, S., M. Paul, and T. Debnath, *Applications of LiDAR in Agriculture and Future Research Directions.* 2023.

35. Wasser, L.A. *The Basics of LiDAR - Light Detection and Ranging - Remote Sensing*. 2022; Available from: https://www.neonscience.org/resources/learning-hub/tutorials/lidar-basics.

36. Neoge, S. and N. Mehendale, *Review on LiDAR technology*. 2020.

37. Sharma, B. *What is LiDAR technology and how does it work?* 2022; Available from: https://www.geospatialworld.net/prime/technology-and-innovation/what-is-lidar-technology-and-how-does-it-work/.

38. Odunlade, E. *What is LiDAR and How does it Work*. 2019; Available from: https://circuitdigest.com/article/what-is-lidar-and-how-does-lidar-works.

39. Moreno, H., et al., *On-Ground Vineyard Reconstruction Using a LiDAR-Based Automated System.* 2020.

40. Omasa, K., F. Hosoi, and A. Konishi, *3D lidar imaging for detecting and understanding plant responses and canopy structure.* 2007.

41. Murcia, H.F., S. Tilaguy, and S. Ouazaa, *Development of a Low-Cost System for 3D Orchard Mapping Integrating UGV and LiDAR.* 2021.

42. Westling, F., J. Underwood, and S. Örn, *Light interception modelling using unstructured LiDAR data in avocado orchards.* 2018.

43. Menzel, C.M. and M.D.L. Lagadec, *Increasing the productivity of avocado orchards using high-density plantings: A review.* 2014.

44. DUFORD, M.J. *How big does an avocado tree get*. 2023; Available from: https://www.homefortheharvest.com/how-big-does-an-avocado-tree-get/.

45. Stassen, P.J.C., B. Snijder, and Z.J. Bard, *Results Obtained by Pruning Overcrowded Avocado Orchards.* 1999.

46. Thorp, T.G. and B. Stowell, *Pruning Height and Selective Limb Removal Affect Yield of Large 'Hass' Avocado Trees.* 2001.

47. Wolstenholme, B.N. and A. Sheard, *Avocado Tree Vigour and Size Dictate Orchard Planting Density: A South African Perspective.* 2012.

48. Roe, D. and S. Köhne, *Performance of Commercially Grown 'hass' Avocado on Clonal Rootstocks at Westfalia Estate, South Africa.* 1999.

49. Koirala, A., K.B. Walsh, and Z. Wang, *Attempting to Estimate the Unseen—Correction for Occluded Fruit in Tree Fruit Load Estimation by Machine Vision with Deep Learning.* 2021.

50. Kim, S.U., et al., *Robust methods for estimating the orientation and position of IMU and MARG sensors.* 2021.

51. Kok, M., J.D. Hol, and T.B. Schon, *Using Inertial Sensors for Position and Orientation Estimation.* Foundations and Trends in Signal Processing, 2017. **11**: p. 1-153.

52. Filippeschi, A., et al., *Survey of Motion Tracking Methods Based on Inertial Sensors: A Focus on Upper Limb Human Motion.* 2017.

53. Hindle, B.R., J.W.L. Keogh, and A.V. Lorimer, *Inertial-Based Human Motion Capture: A Technical Summary of Current Processing Methodologies for Spatiotemporal and Kinematic Measures.* 2021.

54. Kok, M. and T.B. Schon, *Magnetometer Calibration Using Inertial Sensors.* IEEE Sensors Journal, 2016.

55. NovAtel, *IMU Errors and Their Effects.* 2014.

56. Khamkar, V., et al., *An Arduino Based Indoor & Outdoor Positioning System (IOPS).* International Journal of Scientific Engineering and Applied Science (IJSEAS), 2018. **4**(3).

57. Insight, G. *How accurate is GPS technology*. 2016; Available from: https://help.gpsinsight.com/deep-dive/how-accurate-is-gps-technology/.

58. Admin, L. *How Accurate Are GPS Coordinates*. 2022; Available from: https://landairsea.com/blogs/news/how-accurate-are-gps-coordinates-1.

59. Central, G., *What is GPS: Getting Started.* 2023.

60. GNSS. *RTK GPS: Understanding Real-Time Kinematic GPS Technology*. 2023; Available from: https://globalgpssystems.com/gnss/rtk-gps-understanding-real-time-kinematic-gps-technology/.

61. Seidle, N. *What is GPS RTK*. 2023; Available from: https://learn.sparkfun.com/tutorials/what-is-gps-rtk/all.

62. Chivers, M. *Differential GPS Explained*. 2003; Available from: https://www.esri.com/about/newsroom/arcuser/differential-gps-explained/.

63. Koenig, F. and D. Wong, *Differential Global Positioning System (DGPS) Operation and Post-Processing Method for the Synchronous Impulse Reconstruction (SIRE) Radar*. 2007.

64. Wing, M.G. and J. Frank, *Vertical measurement accuracy and reliability of mapping-grade GPS receivers.* 2011.

65. Higgins, M.B., *Heighting With GPS: Possibilities and Limitations.*

66. Robotics, M. *https://marvelmind.com/download-2/line_of_sight/*. 2023; Available from: https://marvelmind.com/download-2/line_of_sight/.

67. Dawood, M., et al., *Low Cost Local Positioning System (LPS) -Design and Development for vehicular tracking.* 2013. **45**: p. 439-446.

68. Hoppe, M.W., et al., *Validity and reliability of GPS and LPS for measuring distances covered and sprint mechanical properties in team sports.* 2018.

69. Makerfabs. *ESP32 UWB(Ultra Wideband)*. 2023; Available from: https://www.makerfabs.com/esp32-uwb-ultra-wideband.html.

70. Dupuis, J., C. Holst, and H. Kuhlmann, *Improving the kinematic calibration of a coordinate measuring arm using configuration analysis.* 2017.

71. Creaform3d. *What is a CMM? Everything You Need to Know About Coordinate Measuring Machines and Their Types.* 2023; Available from: https://www.creaform3d.com/blog/what-is-cmm-and-their-types/.

72. Fonseca, J.A.S.D., et al., *Distance Measurement Systems Using Lasers and Their Applications.* 2017.

73. Spike. *Laser Measuring Guide*. 2021; Available from: https://spike.ikegps.com/laser-measuring-guide/.

74. He, H., et al., *Research on Laser Distance Measurement Technology for Remaining Volume of Large Open- pit Material Yard.* 2020.

75. Ehlert, D., R. Adamek, and H.-J. Horn, *Vehicle Based Laser Range Finding in Crops.* 2009.

76. Supply, E. *How Accurate Are Laser Distance Measurements Tools?* 2023; Available from: https://www.engineersupply.com/How-Accurate-Are-Laser-Distance-Measurements-Tools.aspx.

77. Zhang, C., et al., *Method of improving large-scale measurement accuracy of laser tracker based on photogrammetry.* 2019.

78. AVERETT, W.E., *Tree Thinning the Avocado Grove by the Block System.* 1949.

79. Heijden, D.v.d. and R. Kok, *Orchard Management Guide.* 2022.

80. Snijder, B. and P.J.C. Stassen, *Manipulation of Avocado Trees to Control Tree Size a Four Year Progress Report.* 1998.

81. Avocado, L. *High Density Plantings*. 2020; Available from: https://www.lynwoodavocado.co.nz/high-density-plantings.

82. Marakas, G. *Avocado Tree Climate and Soil Requirements – Planting Avocado Trees*. 2023; Available from: https://wikifarmer.com/avocado-tree-climate-and-soil-requirements-planting-avocado-trees/.

83. Commission, C.A. *Crop Estimation Instructions*. 2020; Available from: https://www.californiaavocadogrowers.com/industry/crop-statistics/crop-estimation-instructions.

84. Devices, C. *AMT22 Series*. 2023; Available from: https://www.cuidevices.com/product/motion-and-control/rotary-encoders/absolute/modular/amt22-series.

85. Devices, C. *AMT22 Modular Absolute Encoder Datasheet*. 2023.

86. Dxomark. *Apple iPhone 12 mini Camera test*. 2021; Available from: https://www.dxomark.com/apple-iphone-12-mini-camera-review-performance-in-your-pocket/#:~:text=Key%20camera%20specifications%3A,mm%20measured)%20f%2F2.4%20lens.

87. Apple. *iPhone 12 mini - Technical Specifications*. 2023; Available from: https://support.apple.com/kb/SP829?locale=en_AU.

88. VisionDoctor. *Sensor and pixel sizes of CCD and CMOS sensors*. 2022; Available from: https://www.vision-doctor.com/en/camera-technology-basics/sensor-and-pixel-sizes.html.

89. Google-Maps, *Google Maps Satellite Image*. 2023.

90. Magnetic-Declination.com. *Magnetic declination in Awanui, New Zealand*. [cited 2024; Magnetic declination in Awanui, New Zealand]. Available from: https://www.magnetic-declination.com/New%20Zealand/Awanui/1880284.html.

91. Allan B Woolf, J.H.B., Ian B Ferguson, *Preharvest exposure to the sun influences postharvest responses of 'Hass' avocado fruit.* 1999.

92. A.B. Woolf, I.B.F., L.C. Requejo-Tapia, L. Boyd, W.A. Laing, A. White, *Impact of Sun Exposure on Harvest Quality of 'hass' Avocado Fruit.* 1999.

93. Rankel, K. *Does My Avocado Need Direct Sunlight*. 2023; Available from: https://greg.app/avocado-direct-sunlight/#:~:text=The%20intensity%20and%20duration%20of,flowers%20into%20creamy%2C%20delicious%20avocados.

94. *Avocado flowering: key aspects of management*. 2021; Available from: https://www.seipasa.com/en/blog/avocado-flowering-key-aspects-of-management/.

95. Paula Catala-Roman, E.A.N., Jaume Segura-Garcia, Miguel Garcia-Pineda, *Harnessing Digital Twins for Agriculture 5.0: A Comparative Analysis of 3D Point Cloud Tools.* 2024.

96. Peyman Moghadam, T.L., Everard J. Edwards *Digital Twin for the Future of Orchard Production Systems.* 2020.

97. Jihong Yan, X.L., Siyang Ji, *Design and Implementation of Workshop Virtual Simulation Experiment Platform Based on Digital Twin.* 2024.

APPENDIX A – FORWARD KINEMATICS

Parameters of Denavit-Hartenberg Convention (D-H):

$$A_i = Rot_{z,\theta_i} \cdot Trans_{z,d_i} \cdot Trans_{x,a_i} \cdot Rot_{x,\alpha_i}$$

$$A_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

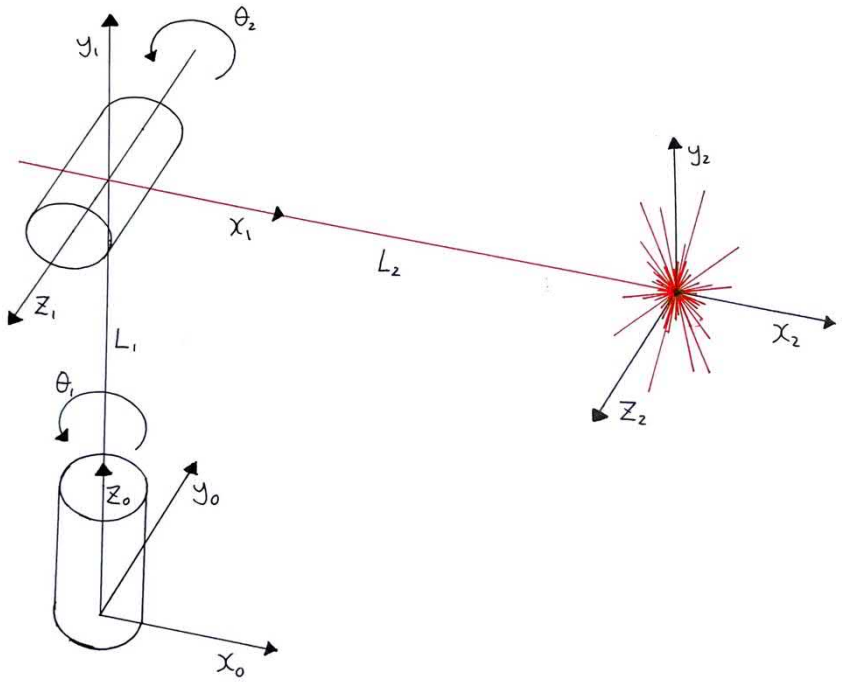
$$\theta_i = link\ angle\ around\ z\ axis$$

$$d_i = link\ offset\ along\ z\ axis$$

$$a_i = link\ length\ along\ x\ axis$$

$$\alpha_i = link\ twist\ about\ x\ axis$$

Kinematic Diagram:



D-H Parameters of device:

Link	$\mathbf{a_i}$	$\boldsymbol{\alpha_i}$	$\mathbf{d_i}$	$\boldsymbol{\Theta_i}$
1	0	90 °	L_1*	Θ_1*
2	L_2	0	0	Θ_2*

*Variable

Link 1 transformation matrix:

$$A_1 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 90° & -\sin 90° & 0 \\ 0 & \sin 90° & \cos 90° & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & 0 \\ \sin\theta_1 & \cos\theta_1 & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 90° & -\sin 90° & 0 \\ 0 & \sin 90° & \cos 90° & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 \cos 90° & \sin\theta_1 \sin 90° & 0 \\ \sin\theta_1 & \cos\theta_1 \cos 90° & -\cos\theta_1 \sin 90° & 0 \\ 0 & \sin 90° & \cos 90° & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Link 2 transformation matrix:

$$A_2 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & 0 \\ \sin\theta_2 & \cos\theta_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & L_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & L_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & L_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

End effector/laser dot transformation matrix:

$$T_2^0 = A_1 \cdot A_2$$

$$T_2^0 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 \cos 90° & \sin\theta_1 \sin 90° & 0 \\ \sin\theta_1 & \cos\theta_1 \cos 90° & -\cos\theta_1 \sin 90° & 0 \\ 0 & \sin 90° & \cos 90° & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & L_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & L_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^0 = \begin{bmatrix} \cos\theta_1\cos\theta_2 - \sin\theta_1\cos 90°\sin\theta_2 & -\sin\theta_2\cos\theta_1 - \sin\theta_1\cos 90°\cos\theta_2 \\ \sin\theta_1\cos\theta_2 + cos\,\theta_1\, cos\, 90°\sin\theta_2 & \cos\theta_1\cos 90°\cos\theta_2 - \sin\theta_2\sin\theta_1 \\ \sin 90°\sin\theta_2 & \cos\theta_2\sin 90° \\ 0 & 0 \end{bmatrix} \rightarrow$$

$$\rightarrow \begin{matrix} \sin\theta_1\sin 90° & L_2\cos\theta_1\cos\theta_2 - L_2\sin\theta_1\cos 90°\sin\theta_2 \\ -\cos\theta_1\sin 90° & L_2\sin\theta_1\cos\theta_2 + L_2\cos\theta_1\cos 90°\sin\theta_2 \\ \cos 90° & L_2\sin 90°\sin\theta_2 + L_1 \\ 0 & 1 \end{matrix} \Bigg]$$

X,Y,Z coordinates of End effector/laser dot:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} L_2\cos\theta_1\cos\theta_2 - L_2\sin\theta_1\cos 90°\sin\theta_2 \\ L_2\sin\theta_1\cos\theta_2 + L_2\cos\theta_1\cos 90°\sin\theta_2 \\ L_2\sin 90°\sin\theta_2 + L_1 \end{bmatrix} = \begin{bmatrix} L_2\cos\theta_1\cos\theta_2 \\ L_2\sin\theta_1\cos\theta_2 \\ L_2\sin\theta_2 + L_1 \end{bmatrix}$$

$$If\ \ d_1\ is\ the\ laser\ distance\ reading, and\ the\ X\ and\ Z\ axis\ offsets$$

$$as\ defined\ by\ the\ CAD\ model\ are$$

$$-0.02965\ meters\ and\ 0.09472\ meters\ respectivly,$$

$$then\ \ L_1 = 0.09472$$

$$and\ \ L_2 = d_1 - 0.02965$$

$$\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{z} \end{bmatrix} = \begin{bmatrix} (\boldsymbol{d_1 - 0.02965})\boldsymbol{\cos\theta_1\cos\theta_2} \\ (\boldsymbol{d_1 - 0.02965})\boldsymbol{\sin\theta_1\cos\theta_2} \\ (\boldsymbol{d_1 - 0.02965})\boldsymbol{\sin\theta_2 + 0.09472} \end{bmatrix}$$

APPENDIX B – LASER MEASURE VALIDATION

Indoor distance validation results:

Actual Distance (mm)	**Laser Distance Min Reading**	**Laser Distance Max Reading**	**Error Range**
500	502	502	+2
1000	1001	1001	+1
1500	1501	1501	+1
2000	2002	2002	+2
2500	2502	2502	+2
3000	3002	3002	+2
3500	3501	3501	+1
4000	4001	4001	+1
4500	4502	4502	+2
5000	5003	5003	+3
5500	5503	5503	+3
6000	6003	6003	+3
6500	6502	6502	+2
7000	7003	7003	+3
7500	7502	7502	+2
8000	8003	8003	+3
		Max Error Range (mm)	**+1 to +3**

Outdoor distance validation results:

Actual Distance (mm)	**Laser Distance Min Reading**	**Laser Distance Max Reading**	**Error Range**
500	497	501	+1 -3
1000	1000	1003	+3
1500	1500	1505	+5
2000	1997	2001	+1 -3
2500	2505	2511	+11
3000	3009	3017	+17
3500	3484	3499	-16
4000	4007	4013	+13
4500	4551	4565	+65
5000	4942	4968	-58
5500	5547	5555	+55
6000	6006	6022	+22
6500	6498	6512	+12 -2
7000	6993	7067	+67 -7
7500	7483	7524	+24 -17
8000	7981	8017	+17 -19
		Max Error Range (mm)	**+67 to -58**

APPENDIX C – CMD VALIDATION DATA

CNC measured points No. 1:

--- 8 point George validation test. Box 1 x 0.8 x 0.15m. Position relatively flat, north facing towards George ---

- Avocado No:1, X:2.517, Y:-0.318, Z:0.419, Origin:(x:0.0, y:0.0, z:0.0)
- Avocado No:2, X:2.536, Y:0.488, Z:0.417, Origin:(x:0.0, y:0.0, z:0.0)
- Avocado No:3, X:3.539, Y:0.459, Z:0.385, Origin:(x:0.0, y:0.0, z:0.0)
- Avocado No:4, X:3.516, Y:-0.341, Z:0.388, Origin:(x:0.0, y:0.0, z:0.0)
- Avocado No:5, X:2.513, Y:-0.318, Z:0.262, Origin:(x:0.0, y:0.0, z:0.0)

CNC measured points No. 2:

--- 8 point George validation test. Box 1 x 0.8 x 0.15m. Position not flat, South-West-Down towards George ---

- Avocado No:1, X:-1.401, Y:1.665, Z:-0.718, Origin:(x:0.0, y:0.0, z:0.0)
- Avocado No:2, X:-1.763, Y:1.09, Z:-1.126, Origin:(x:0.0, y:0.0, z:0.0)
- Avocado No:3, X:-2.26, Y:1.775, Z:-1.652, Origin:(x:0.0, y:0.0, z:0.0)
- Avocado No:4, X:-1.901, Y:2.345, Z:-1.249, Origin:(x:0.0, y:0.0, z:0.0)
- Avocado No:5, X:-1.289, Y:1.662, Z:-0.822, Origin:(x:0.0, y:0.0, z:0.0)

CNC measured points No. 3:

--- 8 point George validation test. Box 1 x 0.8 x 0.15m. Position not flat, South-Up towards George ---

- Avocado No:1, X:-2.451, Y:0.026, Z:1.254, Origin:(x:0.0, y:0.0, z:0.0)
- Avocado No:2, X:-2.709, Y:-0.334, Z:0.594, Origin:(x:0.0, y:0.0, z:0.0)
- Avocado No:3, X:-3.657, Y:-0.191, Z:0.887, Origin:(x:0.0, y:0.0, z:0.0)
- Avocado No:4, X:-3.398, Y:0.162, Z:1.547, Origin:(x:0.0, y:0.0, z:0.0)
- Avocado No:5, X:-2.452, Y:0.158, Z:1.184, Origin:(x:0.0, y:0.0, z:0.0)

Measured vs known edge length (m):

Edge	Box 1	Box 2	Box 3	Actual Length
1	0.8062	0.7925	0.7948	0.8
2	1.0039	0.9964	1.0025	1
3	0.8003	0.7850	0.7920	0.8
4	0.9997	0.9972	1.0006	1
5	0.1571	0.1529	0.1494	0.15
6	0.8063	0.7889	0.7928	0.8
7	1.0010	0.9977	0.9980	1
8	0.7983	0.7836	0.7911	0.8
9	0.9986	0.9991	0.9985	1
10	0.1520	0.1465	0.1548	0.15
11	0.1541	0.1472	0.1525	0.15
12	0.1531	0.1444	0.1485	0.15

Edge length error (m):

Edge	Box 1	Box 2	Box 3
1	0.0062	-0.0075	-0.0052
2	0.0039	-0.0036	0.0025
3	0.0003	-0.0150	-0.0080
4	-0.0003	-0.0028	0.0006
5	0.0071	0.0029	-0.0006
6	0.0063	-0.0111	-0.0072
7	0.0010	-0.0023	-0.0020
8	-0.0017	-0.0164	-0.0089
9	-0.0014	-0.0009	-0.0015
10	0.0020	-0.0035	0.0048
11	0.0041	-0.0028	0.0025
12	0.0031	-0.0056	-0.0015

Edge length absolute error (mm):

	Box 1	Box 2	Box 3
Maximum	7.1	16.3559	8.9229
Average	3.1160	6.1945	3.7636
SD	2.3959	5.2030	2.9599

APPENDIX D – DATUM VALIDATION DATA

Untransformed datum measurements:

--- Datum point Validation test. Includes three groups of different datum layouts, each measured from two different origins. Note saved origin coordinates are for identification purposes only and do not represent the actual offsets. ---

- Datum Point:Gold, X:0.95, Y:0.432, Z:-0.31, Origin:One
- Datum Point:Silver, X:2.29, Y:-0.641, Z:-0.34, Origin:One
- Datum Point:Black, X:2.188, Y:1.692, Z:-0.281, Origin:One
- Datum Point:Gold, X:-1.552, Y:0.057, Z:-0.307, Origin:One
- Datum Point:Silver, X:-3.225, Y:-0.358, Z:-0.336, Origin:One
- Datum Point:Black, X:-1.318, Y:-1.694, Z:-0.298, Origin:One

- Datum Point:Gold, X:1.645, Y:0.38, Z:-0.739, Origin:Two
- Datum Point:Silver, X:1.046, Y:-1.062, Z:-0.734, Origin:Two
- Datum Point:Black, X:2.081, Y:-1.179, Z:-0.744, Origin:Two
- Datum Point:Gold, X:1.006, Y:-1.467, Z:-0.481, Origin:Two
- Datum Point:Silver, X:-0.543, Y:-1.627, Z:-0.457, Origin:Two
- Datum Point:Black, X:-0.154, Y:-2.576, Z:-0.276, Origin:Two

- Datum Point:Gold, X:-1.128, Y:1.295, Z:-0.737, Origin:Three
- Datum Point:Silver, X:0.406, Y:1.54, Z:-0.741, Origin:Three
- Datum Point:Black, X:-0.674, Y:0.429, Z:-0.734, Origin:Three
- Datum Point:Gold, X:-0.968, Y:0.51, Z:-0.925, Origin:Three
- Datum Point:Silver, X:-1.73, Y:-0.851, Z:-0.917, Origin:Three
- Datum Point:Black, X:-1.94, Y:0.687, Z:-0.922, Origin:Three

Transformed datum measurements with error:

--- The points in this file have been transformed based on their datum points or frame origin ---

--- Datum point Validation test. Includes three groups of different datum layouts, each measured from two different origins. Note saved origin coordinates are for identification purposes only and do not represent the actual offsets. ---

--- Transformed Datum to Target Datum Error: Gold 0.0, Silver 0.007, Black 0.016 ---

Gold, 1.21, 0.432, -0.31, One
Silver, 2.55, -0.641, -0.34, One
Black, 2.4480000000000004, 1.692, -0.281, One
Gold, 1.21, 0.432, -0.31, One
Silver, 2.5554814, -0.64538884, -0.34012273, One
Black, 2.4593897, 1.6806155, -0.28131202, One

--- Transformed Datum to Target Datum Error: Gold 0.0, Silver 0.004, Black 0.002 ---

Gold, 1.6157412319027589, -1.0783400782009827, -0.739, Two
Silver, 0.1725362915010154, -1.6744310947412422, -0.734, Two
Black, 0.8216603166302661, -2.489018106668145, -0.744, Two
Gold, 1.6157413, -1.07834, -0.739, Two
Silver, 0.17627406, -1.6728873, -0.73401284, Two
Black, 0.82337123, -2.4888828, -0.74401003, Two

--- Transformed Datum to Target Datum Error: Gold 0.0, Silver 0.006, Black 0.012 ---

Gold, 1.295, 0.8679999999999999, -0.737, Three
Silver, 1.54, -0.6659999999999999, -0.741, Three
Black, 0.42899999999999994, 0.41400000000000003, -0.734, Three
Gold, 1.295, 0.868, -0.737, Three
Silver, 1.5410045, -0.6722905, -0.74101645, Three
Black, 0.42265165, 0.40418825, -0.7339904, Three

Datum point transformation error (mm):

Error (mm)	Gold	Silver	Black
Layout 1	0	7	16
Layout 2	0	4	2
Layout 3	0	6	12
Max	**16**		
Average	**5**		
SD	**6**		

APPENDIX E – ARDUINO CODE

Arduino read/write code (modified):

```
/*
 * AMT22_SPI_Sample_Code.ino
 * Company: CUI Inc.
 * Author: Jason Kelly
 * Version: 2.0.1.0
 * Date: August 20, 2019
 *
 * This sample code can be used with the Arduino Uno to control the AMT22 encoder.
 * It uses SPI to control the encoder and the Arduino UART to report back to the PC
 * via the Arduino Serial Monitor.
 * For more information or assistance contact CUI Inc for support.
 *
 * After uploading code to Arduino Uno open the open the Serial Monitor under the Tools
 * menu and set the baud rate to 115200 to view the serial stream the position from the AMT22.
 *
 * Arduino Pin Connections
 * SPI Chip Select Enc 0:   Pin  2
 * SPI Chip Select Enc 1:   Pin  3
 * SPI MOSI                 Pin 11
 * SPI MISO                 Pin 12
 * SPI SCLK:                Pin 13
 *
 *
 * AMT22 Pin Connections
 * Vdd (5V):                Pin  1
 * SPI SCLK:                Pin  2
 * SPI MOSI:                Pin  3
 * GND:                     Pin  4
 * SPI MISO:                Pin  5
 * SPI Chip Select:         Pin  6
 *
 *
 * This is free and unencumbered software released into the public domain.
 * Anyone is free to copy, modify, publish, use, compile, sell, or
 * distribute this software, either in source code form or as a compiled
 * binary, for any purpose, commercial or non-commercial, and by any
 * means.
 *
 * In jurisdictions that recognize copyright laws, the author or authors
 * of this software dedicate any and all copyright interest in the
 * software to the public domain. We make this dedication for the benefit
 * of the public at large and to the detriment of our heirs and
 * successors. We intend this dedication to be an overt act of
 * relinquishment in perpetuity of all present and future rights to this
 * software under copyright law.
 *
 * THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
 * EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
 * MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
 * IN NO EVENT SHALL THE AUTHORS BE LIABLE FOR ANY CLAIM, DAMAGES OR
 * OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
 * ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
 * OTHER DEALINGS IN THE SOFTWARE.
 */
```

```
/* Include the SPI library for the arduino boards */
#include <SPI.h>

/* Serial rates for UART */
#define BAUDRATE        115200

/* SPI commands */
#define AMT22_NOP       0x00
#define AMT22_RESET     0x60
#define AMT22_ZERO      0x70

/* Define special ascii characters */
#define NEWLINE         0x0A
#define TAB             0x09

/* We will use these define macros so we can write code once compatible with 12 or 14 bit encoders */
#define RES12           12
#define RES14           14

/* SPI pins */
#define ENC_0            2
#define ENC_1            3
#define SPI_MOSI        11
#define SPI_MISO        12
#define SPI_SCLK        13

void setup()
{
  //Set the modes for the SPI IO
  pinMode(SPI_SCLK, OUTPUT);
  pinMode(SPI_MOSI, OUTPUT);
  pinMode(SPI_MISO, INPUT);
  pinMode(ENC_0, OUTPUT);
  pinMode(ENC_1, OUTPUT);

  //Initialize the UART serial connection for debugging
  Serial.begin(BAUDRATE);

  //Get the CS line high which is the default inactive state
  digitalWrite(ENC_0, HIGH);
  digitalWrite(ENC_1, HIGH);

  //set the clockrate. Uno clock rate is 16Mhz, divider of 32 gives 500 kHz.
  //500 kHz is a good speed for our test environment
  //SPI.setClockDivider(SPI_CLOCK_DIV2);   // 8 MHz
  //SPI.setClockDivider(SPI_CLOCK_DIV4);   // 4 MHz
  //SPI.setClockDivider(SPI_CLOCK_DIV8);   // 2 MHz
  //SPI.setClockDivider(SPI_CLOCK_DIV16);  // 1 MHz
  SPI.setClockDivider(SPI_CLOCK_DIV32);    // 500 kHz
  //SPI.setClockDivider(SPI_CLOCK_DIV64);  // 250 kHz
  //SPI.setClockDivider(SPI_CLOCK_DIV128); // 125 kHz

  //start SPI bus
  SPI.begin();
}

void loop()
{
  //create a 16 bit variable to hold the encoders position
  uint16_t encoderPosition;
```

```
  //let's also create a variable where we can count how many times we've tried to obtain the position in case there are errors
  uint8_t attempts;


  //if you want to set the zero position before beggining uncomment the following function call
  // setZeroSPI(ENC_0);
  // setZeroSPI(ENC_1);

  //once we enter this loop we will run forever
  while(1)
  {
    //set attemps counter at 0 so we can try again if we get bad position
    attempts = 0;

    //this function gets the encoder position and returns it as a uint16_t
    //send the function either res12 or res14 for your encoders resolution
    encoderPosition = getPositionSPI(ENC_0, RES12);

    //if the position returned was 0xFFFF we know that there was an error calculating the checksum
    //make 3 attempts for position. we will pre-increment attempts because we'll use the number later and want an accurate count
    while (encoderPosition == 0xFFFF && ++attempts < 3)
    {
      encoderPosition = getPositionSPI(ENC_0, RES12); //try again
    }

    if (encoderPosition == 0xFFFF) //position is bad, let the user know how many times we tried
    {
      Serial.print("Encoder 0 error. Attempts: ");
      Serial.print(attempts, DEC); //print out the number in decimal format. attempts - 1 is used since we post incremented the loop
      Serial.write(NEWLINE);
    }
    else //position was good, print to serial stream
    {

      Serial.print("Encoder 0: ");
      Serial.print(encoderPosition, DEC); //print the position in decimal format
      Serial.write(NEWLINE);
    }

    //////////again for second encoder//////////////////////////

    //set attemps counter at 0 so we can try again if we get bad position
    attempts = 0;

    //this function gets the encoder position and returns it as a uint16_t
    //send the function either res12 or res14 for your encoders resolution
    encoderPosition = getPositionSPI(ENC_1, RES12);

    //if the position returned was 0xFFFF we know that there was an error calculating the checksum
    //make 3 attempts for position. we will pre-increment attempts because we'll use the number later and want an accurate count
    while (encoderPosition == 0xFFFF && ++attempts < 3)
    {
      encoderPosition = getPositionSPI(ENC_1, RES12); //try again
    }

    if (encoderPosition == 0xFFFF) //position is bad, let the user know how many times we tried
    {
```

```
      Serial.print("Encoder 1 error. Attempts: ");
      Serial.print(attempts, DEC); //print out the number in decimal format. attempts - 1 is used since we post incremented the loop
      Serial.write(NEWLINE);
    }
    else //position was good, print to serial stream
    {

      Serial.print("Encoder 1: ");
      Serial.print(encoderPosition, DEC); //print the position in decimal format
      Serial.write(NEWLINE);
    }

    //For the purpose of this demo we don't need the position returned that quickly so let's wait a half second between reads
    //delay() is in milliseconds
    delay(500);
  }
}

/*
 * This function gets the absolute position from the AMT22 encoder using the SPI bus. The AMT22 position includes 2 checkbits to use
 * for position verification. Both 12-bit and 14-bit encoders transfer position via two bytes, giving 16-bits regardless of resolution.
 * For 12-bit encoders the position is left-shifted two bits, leaving the right two bits as zeros. This gives the impression that the encoder
 * is actually sending 14-bits, when it is actually sending 12-bit values, where every number is multiplied by 4.
 * This function takes the pin number of the desired device as an input
 * This funciton expects res12 or res14 to properly format position responses.
 * Error values are returned as 0xFFFF
 */
uint16_t getPositionSPI(uint8_t encoder, uint8_t resolution)
{
  uint16_t currentPosition;       //16-bit response from encoder
  bool binaryArray[16];           //after receiving the position we will populate this array and use it for calculating the checksum

  //get first byte which is the high byte, shift it 8 bits. don't release line for the first byte
  currentPosition = spiWriteRead(AMT22_NOP, encoder, false) << 8;

  //this is the time required between bytes as specified in the datasheet.
  //We will implement that time delay here, however the arduino is not the fastest device so the delay
  //is likely inherantly there already
  delayMicroseconds(3);

  //OR the low byte with the currentPosition variable. release line after second byte
  currentPosition |= spiWriteRead(AMT22_NOP, encoder, true);

  //run through the 16 bits of position and put each bit into a slot in the array so we can do the checksum calculation
  for(int i = 0; i < 16; i++) binaryArray[i] = (0x01) & (currentPosition >> (i));

  //using the equation on the datasheet we can calculate the checksums and then make sure they match what the encoder sent
  if ((binaryArray[15] == !(binaryArray[13] ^ binaryArray[11] ^ binaryArray[9] ^ binaryArray[7] ^ binaryArray[5] ^ binaryArray[3] ^ binaryArray[1]))
          && (binaryArray[14] == !(binaryArray[12] ^ binaryArray[10] ^ binaryArray[8] ^ binaryArray[6] ^ binaryArray[4] ^ binaryArray[2] ^ binaryArray[0])))
    {
```

```
      //we got back a good position, so just mask away the checkbits
      currentPosition &= 0x3FFF;
    }
  else
  {
    currentPosition = 0xFFFF; //bad position
  }

  //If the resolution is 12-bits, and wasn't 0xFFFF, then shift position, otherwise do nothing
  if ((resolution == RES12) && (currentPosition != 0xFFFF)) currentPosition = currentPosition >> 2;

  return currentPosition;
}

/*
 * This function does the SPI transfer. sendByte is the byte to transmit.
 * Use releaseLine to let the spiWriteRead function know if it should release
 * the chip select line after transfer.
 * This function takes the pin number of the desired device as an input
 * The received data is returned.
 */
uint8_t spiWriteRead(uint8_t sendByte, uint8_t encoder, uint8_t releaseLine)
{
  //holder for the received over SPI
  uint8_t data;

  //set cs low, cs may already be low but there's no issue calling it again except for extra time
  setCSLine(encoder ,LOW);

  //There is a minimum time requirement after CS goes low before data can be clocked out of the encoder.
  //We will implement that time delay here, however the arduino is not the fastest device so the delay
  //is likely inherantly there already
  delayMicroseconds(3);

  //send the command
  data = SPI.transfer(sendByte);
  delayMicroseconds(3); //There is also a minimum time after clocking that CS should remain asserted before
we release it
  setCSLine(encoder, releaseLine); //if releaseLine is high set it high else it stays low

  return data;
}

/*
 * This function sets the state of the SPI line. It isn't necessary but makes the code more readable than
having digitalWrite everywhere
 * This function takes the pin number of the desired device as an input
 */
void setCSLine (uint8_t encoder, uint8_t csLine)
{
  digitalWrite(encoder, csLine);
}

/*
 * The AMT22 bus allows for extended commands. The first byte is 0x00 like a normal position transfer, but the
 * second byte is the command.
 * This function takes the pin number of the desired device as an input
 */
void setZeroSPI(uint8_t encoder)
{
  spiWriteRead(AMT22_NOP, encoder, false);
```

```
  //this is the time required between bytes as specified in the datasheet.
  //We will implement that time delay here, however the arduino is not the fastest device so the delay
  //is likely inherantly there already
  delayMicroseconds(3);

  spiWriteRead(AMT22_ZERO, encoder, true);
  delay(250); //250 second delay to allow the encoder to reset
}
/*
 * The AMT22 bus allows for extended commands. The first byte is 0x00 like a normal position transfer, but the
 * second byte is the command.
 * This function takes the pin number of the desired device as an input
 */
void resetAMT22(uint8_t encoder)
{
  spiWriteRead(AMT22_NOP, encoder, false);

  //this is the time required between bytes as specified in the datasheet.
  //We will implement that time delay here, however the arduino is not the fastest device so the delay
  //is likely inherantly there already

  delayMicroseconds(3);

  spiWriteRead(AMT22_RESET, encoder, true);

  delay(250); //250 second delay to allow the encoder to start back up
}
```

APPENDIX F – TRIAL DATA

Single tree trial data (untransformed):

--- This is Anna's test tree. 5 m tall. Wind blowing avocados up to one avocados' width off position. Laser distance fluctuating up to 30 mm for long readings ---

- Datum Point:Gold, X:0.773, Y:1.385, Z:-0.685, Origin:(x:0.0 y:1.0 z:1.0)
- Datum Point:Silver, X:-2.527, Y:-0.004, Z:-0.638, Origin:(x:0.0 y:1.0 z:1.0)
- Datum Point:Black, X:1.179, Y:-3.411, Z:-0.722, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:1, X:2.534, Y:-1.849, Z:2.496, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:2, X:2.501, Y:-1.837, Z:2.524, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:3, X:2.304, Y:-1.909, Z:2.636, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:4, X:2.294, Y:-1.637, Z:2.762, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:5, X:2.447, Y:-1.157, Z:2.861, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:6, X:2.311, Y:-0.957, Z:2.398, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:7, X:2.207, Y:-0.67, Z:2.95, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:8, X:1.927, Y:-0.405, Z:2.965, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:9, X:2.533, Y:-0.676, Z:1.755, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:10, X:2.259, Y:-0.097, Z:1.478, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:11, X:1.915, Y:-0.112, Z:2.492, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:12, X:0.974, Y:-0.042, Z:2.108, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:13, X:0.844, Y:-0.385, Z:2.224, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:14, X:0.432, Y:-0.274, Z:1.955, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:15, X:0.442, Y:-0.326, Z:1.978, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:16, X:-0.042, Y:0.086, Z:1.423, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:17, X:0.001, Y:-0.153, Z:1.775, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:18, X:-0.008, Y:-0.186, Z:1.73, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:19, X:2.216, Y:0.102, Z:2.1, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:20, X:2.215, Y:0.174, Z:2.096, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:21, X:1.772, Y:0.12, Z:1.932, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:22, X:2.124, Y:0.275, Z:2.006, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:23, X:2.189, Y:0.449, Z:2.229, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:24, X:2.103, Y:0.476, Z:2.043, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:25, X:1.916, Y:0.464, Z:2.263, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:26, X:2.055, Y:0.555, Z:2.283, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:27, X:2.118, Y:0.579, Z:2.297, Origin:(x:0.0 y:1.0 z:1.0)

- Avocado No:28, X:1.909, Y:0.585, Z:1.972, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:29, X:1.642, Y:0.707, Z:1.776, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:30, X:1.345, Y:0.621, Z:1.646, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:31, X:1.024, Y:0.205, Z:3.055, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:32, X:1.351, Y:0.334, Z:2.86, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:33, X:1.313, Y:0.465, Z:2.537, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:34, X:0.939, Y:0.563, Z:2.4, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:35, X:1.309, Y:0.736, Z:2.548, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:36, X:1.298, Y:0.772, Z:2.588, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:37, X:1.326, Y:0.826, Z:2.656, Origin:(x:0.0 y:1.0 z:1.0)
- Avocado No:38, X:0.903, Y:0.765, Z:2.41, Origin:(x:0.0 y:1.0 z:1.0)
- Datum Point:Gold, X:2.851, Y:-2.94, Z:-0.766, Origin:(x:-1.5 y:-1.0 z:1.0)
- Datum Point:Silver, X:2.507, Y:0.632, Z:-0.672, Origin:(x:-1.5 y:-1.0 z:1.0)
- Datum Point:Black, X:-1.843, Y:-1.888, Z:-0.737, Origin:(x:-1.5 y:-1.0 z:1.0)
- Avocado No:1, X:1.809, Y:-0.142, Z:0.505, Origin:(x:-1.5 y:-1.0 z:1.0)
- Avocado No:2, X:1.768, Y:-0.103, Z:0.536, Origin:(x:-1.5 y:-1.0 z:1.0)
- Avocado No:3, X:2.31, Y:0.135, Z:0.64, Origin:(x:-1.5 y:-1.0 z:1.0)
- Avocado No:4, X:2.247, Y:0.183, Z:0.645, Origin:(x:-1.5 y:-1.0 z:1.0)
- Avocado No:5, X:2.055, Y:1.107, Z:0.752, Origin:(x:-1.5 y:-1.0 z:1.0)
- Avocado No:6, X:1.68, Y:1.153, Z:0.699, Origin:(x:-1.5 y:-1.0 z:1.0)
- Avocado No:7, X:1.632, Y:1.31, Z:0.733, Origin:(x:-1.5 y:-1.0 z:1.0)
- Avocado No:8, X:1.494, Y:1.215, Z:0.991, Origin:(x:-1.5 y:-1.0 z:1.0)
- Avocado No:9, X:1.454, Y:1.239, Z:1.027, Origin:(x:-1.5 y:-1.0 z:1.0)
- Avocado No:10, X:1.091, Y:1.313, Z:0.997, Origin:(x:-1.5 y:-1.0 z:1.0)

APPENDIX G – CAMERA INTRINSICS

Python camera simulation code:

```
import numpy as np
import math as m
import cv2 as cv
from Read_edited_data import read_edited_points

pi = m.pi


def transform_camera(points, x_rot, y_rot, z_rot, x, y, z):

    x_rot, y_rot, z_rot, x, y, z = -x_rot, -y_rot, -z_rot, -x, -y, -z

    x_rotation = np.array([[1, 0, 0],
                           [0, m.cos(x_rot), -m.sin(x_rot)],
                           [0, m.sin(x_rot), m.cos(x_rot)]])

    y_rotation = np.array([[m.cos(y_rot), 0, m.sin(y_rot)],
                           [0, 1, 0],
                           [-m.sin(y_rot), 0, m.cos(y_rot)]])

    z_rotation = np.array([[m.cos(z_rot), -m.sin(z_rot), 0],
                           [m.sin(z_rot), m.cos(z_rot), 0],
                           [0, 0, 1]])

    H = x_rotation @ y_rotation @ z_rotation

    for point in range(np.shape(points)[1]):
        points[:, point] = points[:, point] @ H

    points[0] += x
    points[1] += y
    points[2] += z

    return points


def define_camera(horizontal_fov, sensor_width, sensor_height, pixel_size):
    """
    Defines a camera based on input parameters and returns same parameters with a camera matrix
    :param horizontal_fov: float in radians
    :param sensor_width: integer in pixels
    :param sensor_height: integer in pixels
    :param pixel_size: float in meters
    :return:
    """

    # define camera
    h_fov = horizontal_fov  # rad
    w = sensor_width  # px
    h = sensor_height  # px
    px_to_m = pixel_size  # dimension in meters of one square pixel

    # calculate camera matrix
    cx = w / 2 - 0.5
    cy = h / 2 - 0.5
    f = cx / (m.tan(h_fov / 2))
    print("focal length (m): {}".format(f*pixel_size))
```

```
    cam_mtx = np.array([[f, 0, cx],
                        [0, f, cy],
                        [0, 0, 1]])

    return h_fov, w, h, px_to_m, cam_mtx


def photograph_points(defined_camera, points, background_directory):
    """
    Takes a photograph with a given camera of focal point position 0,0,0 looking out along the z axis.
    To shift the camera angle, transform the points before calling this function.
    :param defined_camera: camera defined in 'define_camera' funtion
    :param points: 3d points to be photographed
    :param background_directory: An image directory or None for black background
    :return:
    """

    h = defined_camera[2]
    w = defined_camera[1]
    px_to_m = defined_camera[3]
    cam_mtx = defined_camera[4]

    if background_directory is None:
        image = np.zeros((h, w, 3), dtype="uint8")  # create black background image
        image += 255        # make background white
    else:
        image = cv.imread(background_directory)
        image = cv.resize(image, (w, h), interpolation=cv.INTER_AREA)

    points /= px_to_m  # convert coordinate units from meters to pixels
    image_points = cam_mtx @ points  # get camera projection of vectors
    image_points = (image_points / image_points[2]).T  # divide by z and transpose for image

    in_frame = 0
    out_frame = 0
    for pixel in image_points:  # color vertices pixels
        x = pixel[0]
        y = pixel[1]

        x = int(-x + w)     # flips x-axis direction about image origin to align camera and image coordinates
        y = int(-y + h)     # flips y-axis direction about image origin to align camera and image coordinates

        if w > x >= 0 and h > y >= 0:       # only draws if the point is within frame
            # image[y, x] = (0, 0, 255)
            image = cv.circle(image, (x, y), 35, (57, 74, 54), -1)
            image = cv.circle(image, (x, y), 35, (28, 41, 30), 5)

            in_frame += 1

        else:
            out_frame += 1

    print("Avocados in frame: {}\nAvocados out of frame {}\n".format(in_frame, out_frame))
    return image


avocados = read_edited_points("Tree Data Edited", "tree Anna's Test Tree transformed")[0]

# transform to orientate camera along x_axis
# no offset is applied because the origin point is set at first measurement frame
avocados = transform_camera(avocados, 0, 0, -pi/2, 0, 0, 0)
avocados = transform_camera(avocados, -pi/4, 0, 0, 0, 0, 0)
```

```
camera = define_camera(1.1722, 4032, 3024, 0.0000014)

photo = photograph_points(camera, avocados, None)

cv.imshow("image", photo)
cv.waitKey(0)
```

APPENDIX H – AVOCADO COUNT DATA

Orchard avocado count comparison:

Tree	3D Data Count	Orchard Manual Count	Variation (%)
Z2B58R16_T3	180	110	-39%
Z2B58R16_T4	213	143	-33%
Z2B58R16_T5	109	77	-29%
Z2B58R16_T9	142	83	-42%
Z2B58R16_T10	18	17	-6%
Z2B58R16_T11	239	206	-14%
Z2B58R16_T15	145	92	-37%
Z2B58R16_T16	190	126	-34%
Z2B58R16_T17	52	35	-33%
Z2B58R24_T3	32	45	41%
Z2B58R24_T4	108	98	-9%
Z2B58R24_T5	138	189	37%
Z2B58R24_T9	152	182	20%
Z2B58R24_T10	90	68	-24%
Z2B58R24_T11	72	88	22%
Z2B58R24_T15	140	178	27%
Z2B58R24_T16	95	120	26%
Z2B58R24_T17	83	112	35%
Z3B49R14_T6	97	84	-13%
Z3B49R14_T7	120	91	-24%
Z3B49R14_T8	62	59	-5%
Z3B49R14_T11	103	95	-8%
Z3B49R14_T12	89	67	-25%
Z3B49R14_T13	38	27	-29%
Z3B49R14_T19	117	114	-3%
Z3B49R14_T20	42	37	-12%
Z3B49R14_T21	44	34	-23%
Maximum Variation			41%
Minimum Variation			-42%
Average Fruit Per Tree	**108**	**95**	**-11%**
Absolute Average Error			24%

APPENDIX I – ORCHARD TREE PHOTOGRAPHS

All images are taken from the side view, at 0.5 times zoom, on an iPhone 12 Mini:

B58R16 Tree 3

B58R16 Tree 4

B58R16 Tree 5

B58R16 Tree 9

B58R16 Tree 10

B58R16 Tree 11

B58R16 Tree 15

B58R16 Tree 16

B58R16 Tree 17

B58R24 Tree 3

B58R24 Tree 4

B58R24 Tree 5

B58R24 Tree 9

B58R24 Tree 10

B58R24 Tree 11

B58R24 Tree 15

B58R24 Tree 16

B58R24 Tree 17

B49R14 Tree 6

B49R14 Tree 7

B49R14 Tree 8

B49R14 Tree 11

B49R14 Tree 12

B49R14 Tree 13

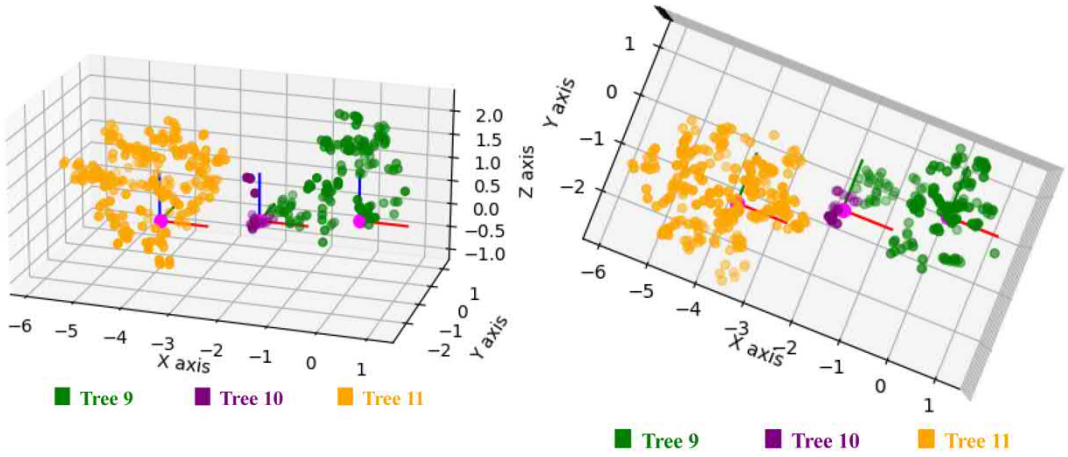
B49R14 Tree 19

B49R14 Tree 20

B49R14 Tree 21

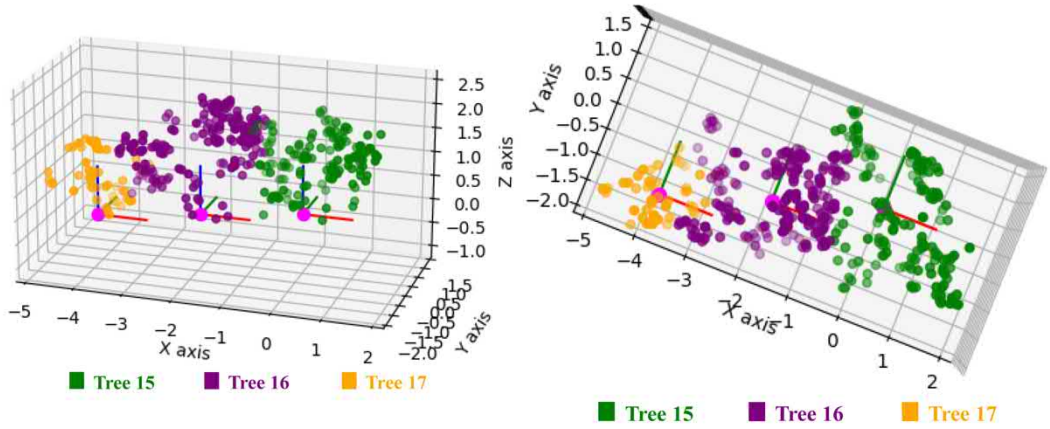
APPENDIX J – NEIGHBOURING TREE PLOTS

Z2B58R16 trees 9,10,11 side view (a), and birds eye view (b):



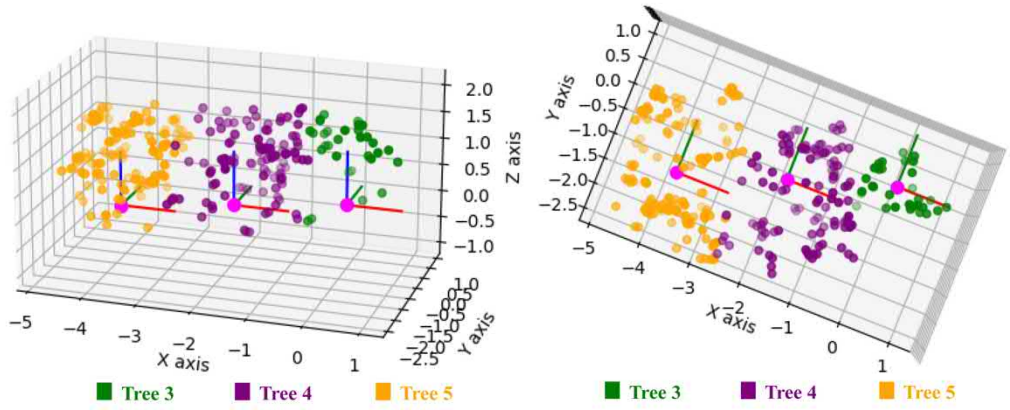
(a) (b)

Z2B58R16 trees 15,16,17 side view (a), and birds eye view (b):



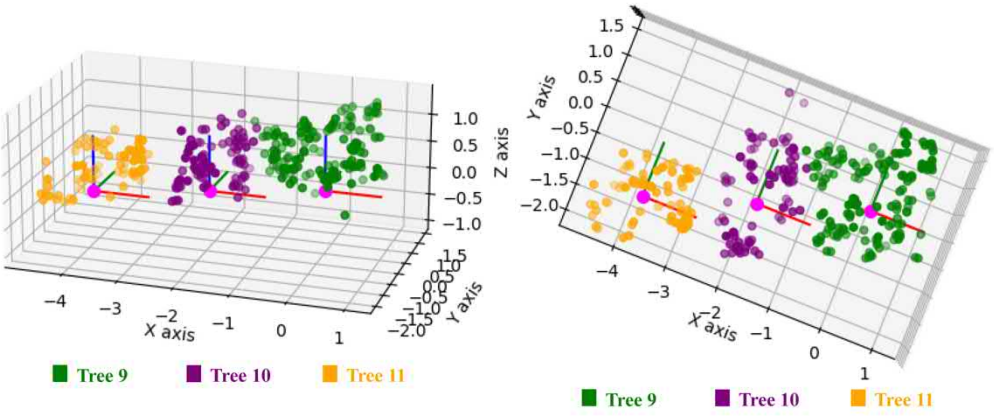
(a) (b)

Z2B58R24 trees 3,4,5 side view (a), and birds eye view (b):



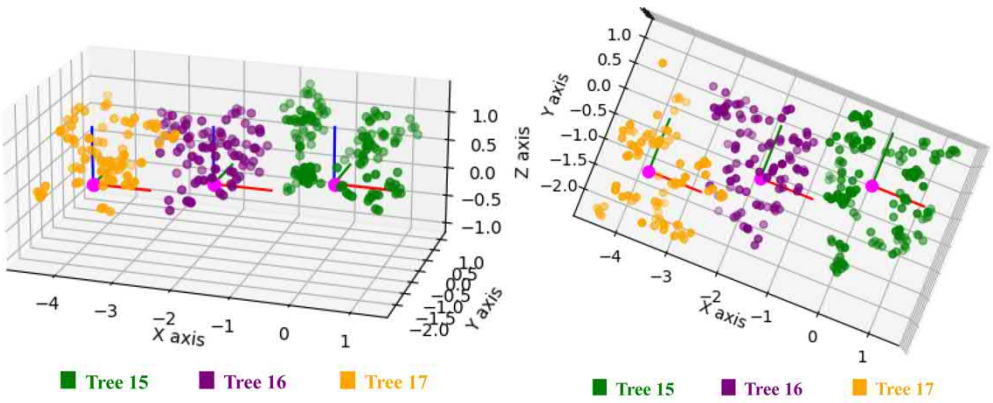
(a) (b)

Z2B58R24 trees 9,10,11 side view (a), and birds eye view (b):



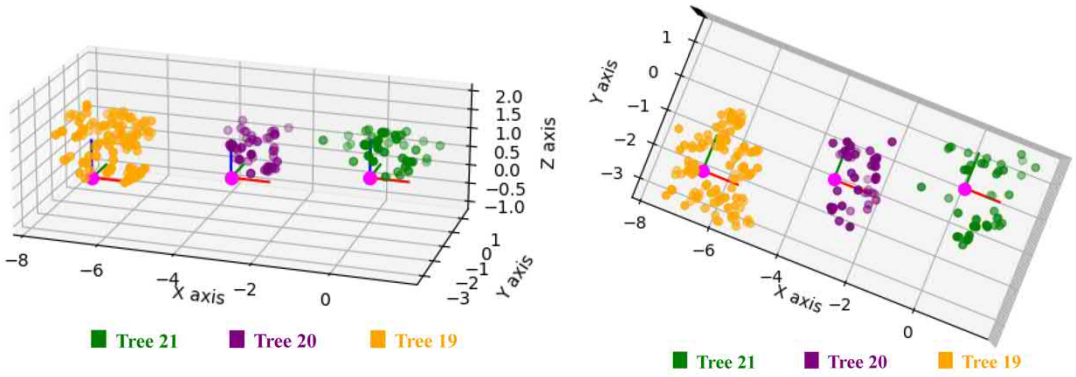
(a) (b)

Z2B58R24 trees 15,16,17 side view (a), and birds eye view (b):



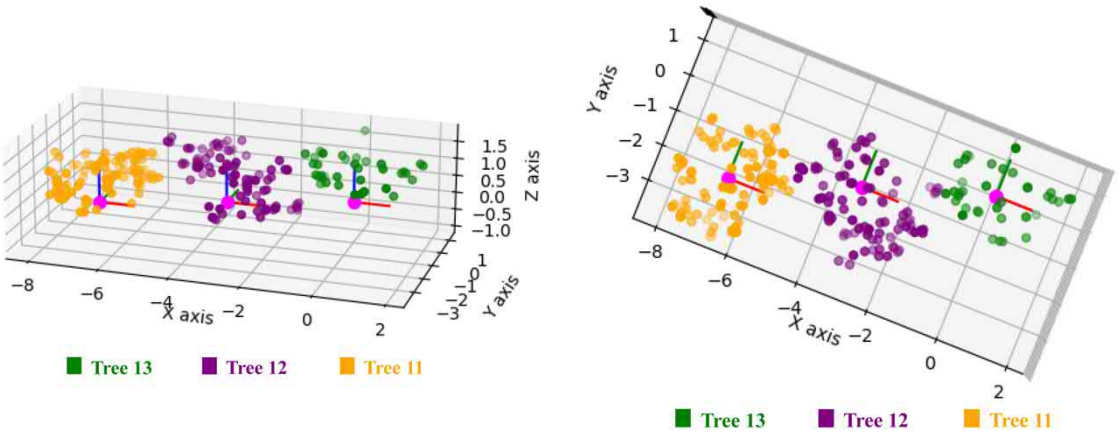
(a) (b)

Z3B49R14 trees 21,20,19 side view (a), and birds eye view (b):



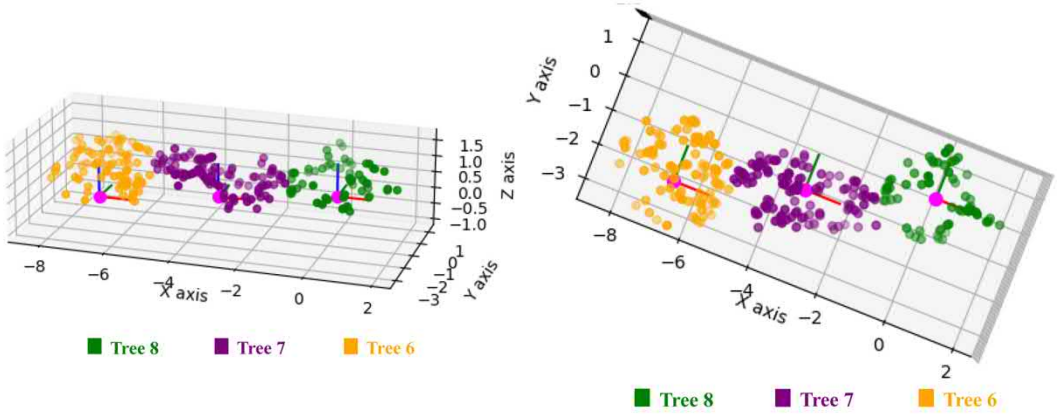
(a) (b)

Z3B49R14 trees 13,12,11 side view (a), and birds eye view (b):



(a) (b)

Z3B49R14 trees 8,7,6 side view (a), and birds eye view (b):



(a) (b)

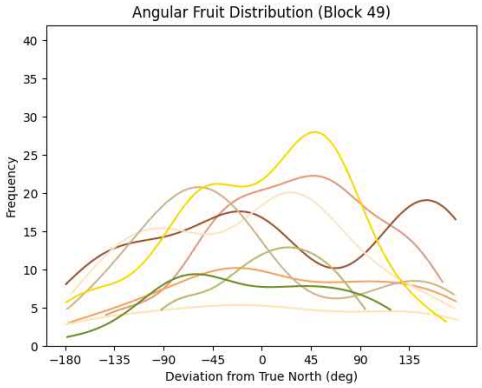
APPENDIX K – DATA COLLECTION EQUIPMENT LIST

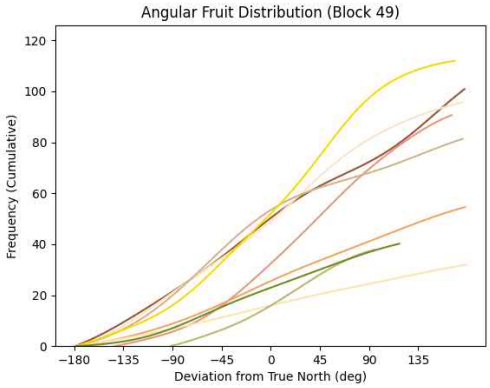
List of required equipment to collect position data with CMD:

Equipment List
Laptop
Battery pack
Laptop charging cable
Laser measure charging cable
USB data cable (5m)
Portable table/crate
Camping chair
CMD
CMD tripod mount
Tripod
Measurement frame
Frame compass attachment
Frame legs (long)
Frame legs (short)
Frame feet (large)
Frame feet (small)
Frame spanner and Allen key
Datum points (3)
Marking tape/ribbon
Tape measure

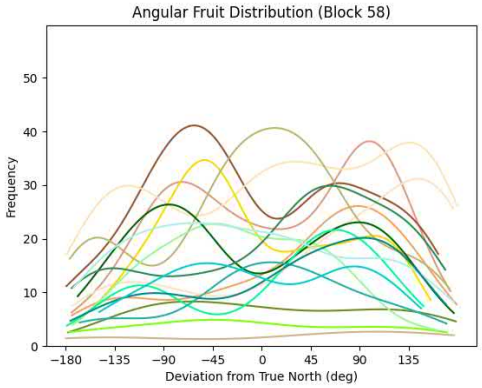
APPENDIX L – INDIVIDUAL TREE DISTRIBUTIONS

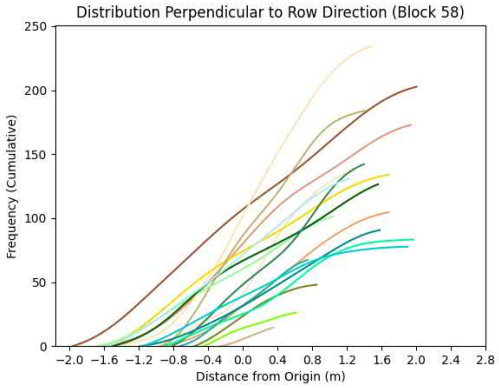
KDE curves for individual trees, showing angular fruit distribution of Block 49 (frequency & cumulative frequency):



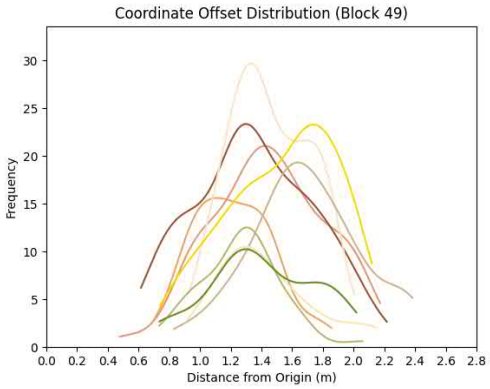


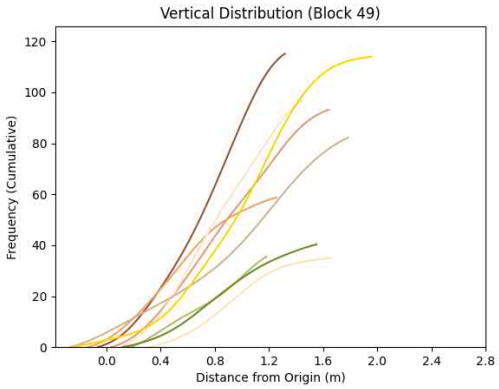
KDE curves for individual trees, showing angular fruit distribution of Block 58 (frequency & cumulative frequency):



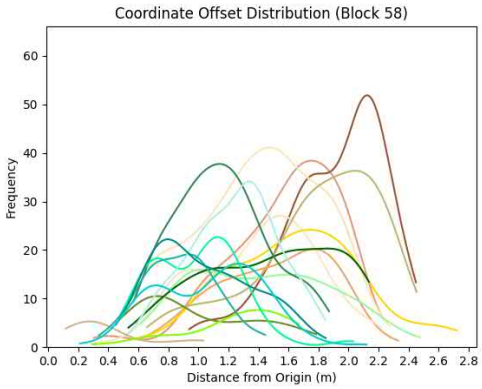


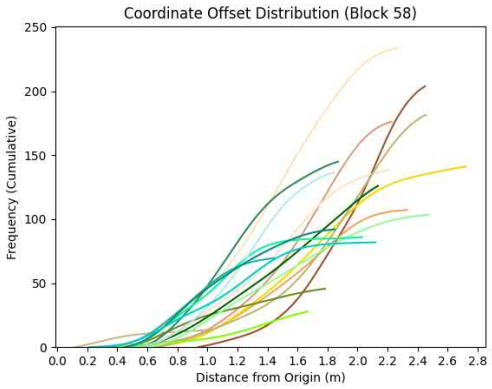
KDE curves for individual trees, showing distance of fruit from origin of Block 49 (frequency & cumulative frequency):



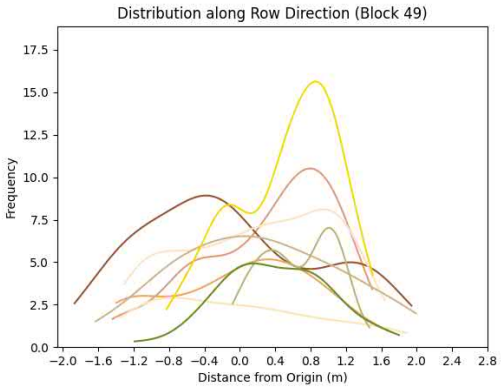


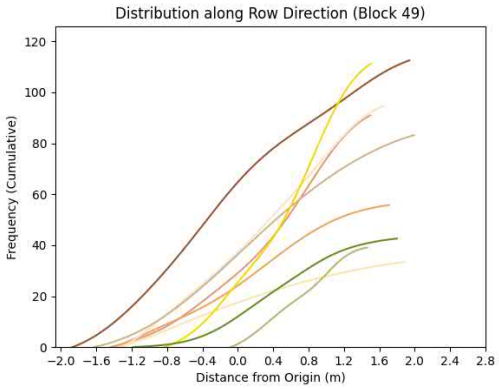
KDE curves for individual trees, showing distance of fruit from origin of Block 58 (frequency & cumulative frequency):



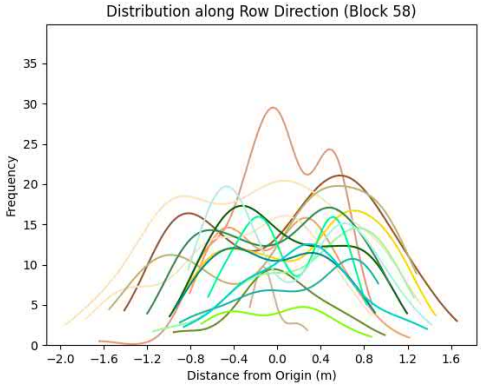


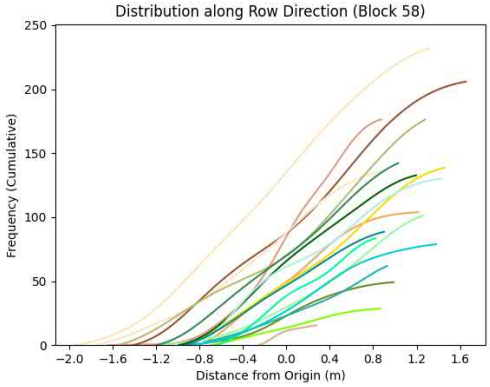
KDE curves for individual trees, showing fruit distribution along row direction for Block 49 (frequency & cumulative frequency):



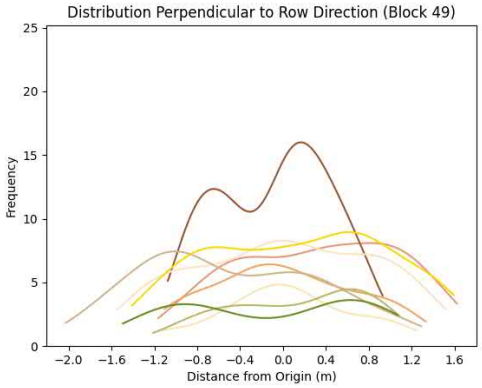


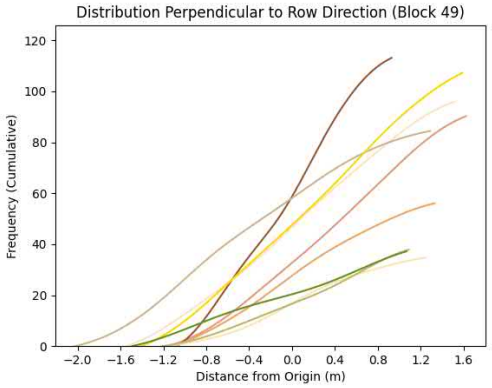
KDE curves for individual trees, showing fruit distribution along row direction for Block 58 (frequency & cumulative frequency):



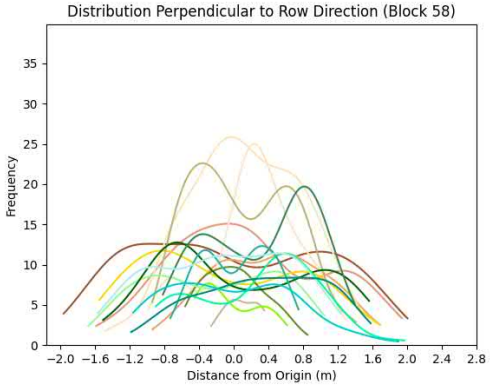


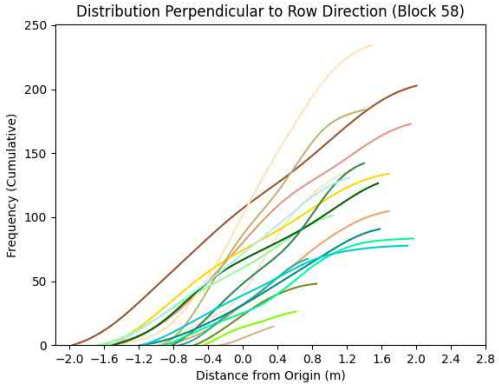
KDE curves for individual trees, showing fruit distribution perpendicular to row direction for Block 49 (frequency & cumulative frequency):



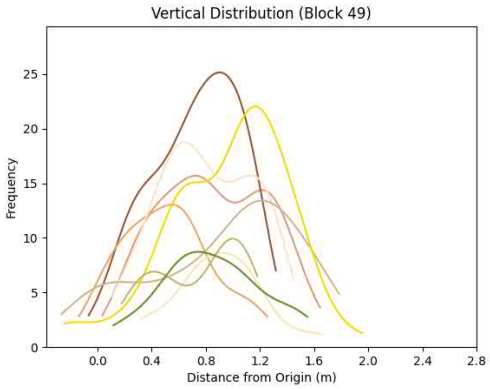


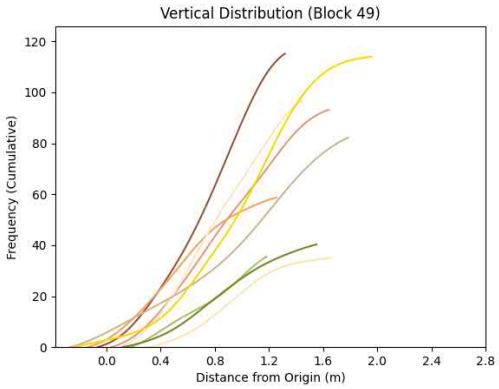
KDE curves for individual trees, showing fruit distribution perpendicular to row direction for Block 58 (frequency & cumulative frequency):





KDE curves for individual trees, showing vertical fruit distribution of Block 49 (frequency & cumulative frequency):





KDE curves for individual trees, showing vertical fruit distribution of Block 58 (frequency & cumulative frequency):