Time-of-flight Perception Pipeline for Selective Green Asparagus Harvesting: Theory and Application

A thesis

submitted in fulfilment

of the requirements for the Degree

of

Doctor of Philosophy

at the

University of Waikato

by

Matthew Christopher Scott Peebles

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

University of Waikato

2021

Abstract

Generally declining labour markets, coupled with a significant increase in the projected global population have raised concerns over a potential food scarcity crisis. In response to this, the agricultural industry is currently undergoing a technological revolution, especially with respect to robotic harvesting.

Green asparagus, due to its unique physiology and labour intensive harvesting requirements, has been a long-time crop of focus in this field. Despite the relatively large body of work surrounding the development of selective robotic asparagus harvesting solutions, no such machine has yet ascended to the level of being commercially competitive. A critical component of a prospective selective robotic asparagus harvester, that is missing from contemporary machines, is a robust vision system, capable of detecting, and localising, harvest-eligible spears in real-time from a commercial asparagus field.

This thesis presents a novel perception pipeline for time-of-flight images that can achieve this task, and describes the implementation of the proposed perception pipeline into two functional robotic harvesters, AHR-1 and AHR-2, capable of harvesting green asparagus spears in a real-world commercial setting.

The proposed perception pipeline achieves spear detection by firstly, segmenting the soil plane from an input pointcloud of an asparagus row, generated with a time-of-flight camera. Two plane detection methods, namely RANSAC and a modified version Hyun's method (MHM) were investigated for this purpose. A detailed evaluation of the performance of each method on cluttered scenes revealed both RANSAC and MHM to be suitable for soil plane segmentation of asparagus beds, with both methods demonstrating similar RMSE error across a variety of scenes. The stability of model predictions made by RANSAC was generally found to be lower than that achieved by MHM, particularly for high-clutter scenes. This was determined to be due to the non-deterministic nature of RANSAC coupled with high degrees of soil plane occlusion in high clutter scenes.

Following soil plane removal, non-asparagus points pertaining to rocks, weeds, and other field debris are then filtered from the scene. This is achieved by coarse filtering input points based on the output of a FRCNN model. The

various FRCNN models utilised in this work were trained with a novel dataset of labelled images, collected from various asparagus farms throughout New Zealand and California, USA. Evaluation of these models revealed a typical maximum F1 score of 0.73, providing reasonable frame-by-frame identification of asparagus features.

The remaining point clusters, representing each asparagus spear in the scene, are then filtered to remove flying pixels; a typical artifact of time-of-flight imaging. A novel geometric filter, named the closest point filter (CP filter), was developed for this task. Based on laboratory testing, it was found that this filter achieved a 68% reduction in the mean standard deviation of intra-cluster distance with respect to ground-truth positions, resulting in a significant improvement in the accuracy of base point predictions.

The proposed perception pipeline is implemented inside a ROS framework, and deployed on two robotic harvesting platforms. The first platform, AHR-1, was developed as a proof-of-concept system. This system provided a wealth of knowledge which was utilised to develop a more complete prototype asparagus harvester, AHR-2. The design of AHR-1 and AHR-2 was largely informed by the shortcomings of existing asparagus harvesting robots from the literature, particularly with respect to their harvesting, and detection strategies.

The literature surrounding robotic asparagus harvesters does not provide an adequate method for objectively evaluating the performance of such systems. Consequentially, performance metrics pertaining to existing robotic harvesters are relatively opaque, particularly with respect to the selective nature of the harvesting task. These inadequacies necessitated the development of a novel evaluation method for the evaluation of a selective asparagus harvester, which was utilised to measure the performance of both AHR-1 and AHR-2.

Several field trials of AHR-1 and AHR-2 were conducted on farms throughout New Zealand and California, USA. The resulting analysis revealed that AHR-2 achieved state-of-the-art performance, harvesting 45.9% of all harvest-eligible spears with a precision of 87.2% at its nominal ground speed of 0.3m/s. The vision system detected 97% of all harvest-eligible spears with a precision of 74.5% at this speed. When operating at a ground speed of 0.7m/s AHR-2's harvesting rate fell to 22% with 95% precision. The corresponding drop in the vision systems detection rate was relatively small, dropping to 92.5% with 87.2% precision at a ground speed of 0.7m/s. From this it was concluded that the vision system outperformed the available hardware. The perception pipeline proposed by this thesis achieved state-of-the-art performance at a variety of ground speeds. The pipeline demonstrated a robustness to the unstructured nature of commercial asparagus fields, providing spear locations which successfully facilitated the robotic harvest of green asparagus spears.

Acknowledgements

Firstly, I would like to acknowledge Callaghan Innovation, and Robotics Plus Limited for funding this research, and providing technical guidance. Dr. Shen Hin Lim is also acknowledged for the supervision and academic mentor-ship he provided throughout the project. Dr. Lim's guidance was invaluable to the success of this research.

I would also like to thank the various asparagus growers who were involved throughout this research for their patience and contributed knowledge, time, and resources. In particular, I would like to acknowledge Matt Carnachan of Kaimai Fresh, Tony Rickman of Boyds Asparagus, Geoff Lewis of Tendertips NZ, and Aaron, Aric, and Jake Barcellos of Ag-Bar Ag Enterprises. Without the support of these growers this research would not have been possible.

Finally, I would like to acknowledge Josh Barnett for the remarkable mechanical, and mechatronic design efforts he contributed to this project. The robotic systems designed, and constructed by Mr. Barnett were paramount to the success of this research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The United Nations estimate that the global population will reach 9.8 billion people by the year 2050 [1]. As the lifeblood of civilisation, the agricultural industry is vitally important in sustaining this growth. However, despite the ever increasing demand for agricultural produce, global labour shortages are on the rise, particularly in developed nations [2, 3]. This mismatch between supply and demand imposes considerable food scarcity concerns on future populations [4]. In response, agricultural sectors, worldwide, are seeking technological solutions in the form of automation. Agricultural industries that are particularly dependent on human labour, such as the cultivation of green asparagus, are currently in dire need of such advances.

Modern advancements in technology have seen automation widely adopted by many industries, particularly in manufacturing [5], healthcare [6], and transport [7]. However, commercially viable agricultural robots are still relatively limited in both prevalence and application. As such, a large amount of research in this sector has been undertaken [8–26].

Various market leading suppliers of agricultural equipment already manufacture driverless/automated tractors for autonomous navigation and precision planting, however, these machines are limited in their utility and are far from being capable of automating complex agricultural tasks such as crop harvesting. In order to automate such tasks, significant advancements in hardware, perception methods, control, and mechanical design must be made. Harvesting

tasks often require complex, highly dexterous manipulation of unstructured environments, coupled with high performance vision systems capable of detecting, and localising target crops at human-competitive rates. Furthermore, in many cases subjective judgements about the eligibility of individual crops based on, for example size, colour or ripeness, need to be made. This requires decision making by sophisticated software systems.

Literature regarding robotic harvesting reveals a considerable amount of work focusing on individual aspects of these requirements, however, there is a limited prevalence of such research being adopted for commercial use, particularly as a fully integrated solution. Consequentially, research pathways aiming to explore the application of these works remain open, especially for use in commercial settings.

The maturity of research surrounding robotic harvesting, coupled with the dire needs of the industry make green asparagus an ideal candidate as a case study for robotic harvesting in an applied setting.

1.1 Green Asparagus Industry

Green asparagus has a rich history with respect to mechanised harvesting. As such, a great deal of research has been completed, and a number of commercial solutions proposed. Despite this, the asparagus industry has still not adopted any automated solutions. As a result, the industry is currently suffering under the aforementioned labour shortages. With the asparagus industry projected to be valued at over 36 billion USD by 2027 [27], there is a large incentive for automation.

Asparagus is a perennial plant that sprouts from underground structures known as “crowns”, and is harvested seasonally during the spring. Asparagus is a notoriously labour intensive crop to produce, due to its unique physiology. The emerging spears grow randomly and extremely quickly [28]. New Zealand growers, consulted during this research, estimate that their particular

asparagus varieties, mainly "Jersey Giant" and "Pacific 2000", grow at a rate of between 7 and 10cm per day [29]. The spears must be harvested before they reach a height of approximately 23cm to ensure they remain edible. Further growth of the spears will result in the crop maturing past an edible point, eventually forming a fern like bush. During the off-season months the asparagus crop is allowed to mature completely, enabling the plant to produce the required carbohydrates to support sprouting in the following season. The fast growth rate of asparagus spears dictates that a significant portion of the field must be harvested daily in order to avoid loss of edible product.

In the New Zealand market, the cost of labour to achieve daily harvesting is prohibitively high. A major asparagus producer can expect to pay approximately 1 million NZD per year to service a 100 ha farm [30]. In addition to the direct costs of employing labour there are various other costs which can impact on the economic viability of the asparagus industry, particularly in developed nations where the cost of labour is high. These include health and safety compliance, administrative overheads, and the costs associated with a migrant workforce. Furthermore, the necessity of daily harvesting makes the asparagus industry particularly vulnerable to unforeseen labour shortages. Changes to immigration law, or worldwide events such as the 2020 Covid-19 pandemic can, therefore, significantly impact the economic viability of the industry. These issues have generated significant demand for a robotic harvesting solution.

1.2 Automating the Asparagus Harvest

Developing a robotic harvesting solution for asparagus requires an understanding of the manner in which the crop is commercially grown. On a commercial farm, asparagus crowns are typically planted in rows, either on flat ground or in mounded structures. Figure 1.1 shows an image of a commercial asparagus row in New Zealand.

Figure 1.1: Example of a typical asparagus row from a New Zealand farm.

The sporadic and random growth characteristics of asparagus spears dictate that only a small subset of spears are eligible for harvest at any particular time. Growers need to take care to ensure spears are harvested at the optimal length in order to maximise the saleable yield. Therefore, it is essential that shorter spears be allowed time to grow to a marketable height, and taller spears are not allowed to mature past saleability. This requires that human harvesters must make subjective judgements on the eligibility of each spear for harvest. Consequentially, indiscriminate bulk harvesting methods are not appropriate.

In order for a robotic harvester to mimic the ability of human harvesters, it must be capable of making selective judgments about the eligibility of each spear for harvest. This requires that such a system be capable of generating an accurate model of the asparagus bed, and interpreting this model to both detect, and localise eligible spears. Generating a geometrical model of an asparagus bed is a task well suited to range imaging techniques.

Stereo vision is a widely adopted range imaging technique that utilises a pair of cameras to detect features in a scene from two known perspectives. When the transform between these perspectives is known, the relative position of features in each image can be used to infer the distance between the camera

and scene elements at a number of pixels. An image that encodes these ranges is known as a depth map. Stereo vision techniques are useful for generating high precision, high resolution depth maps of scenes where the feature density is high, but are known to struggle when features are scarce; for example when imaging a large flat wall. Another known limitation of such techniques is the computational complexity involved with both feature matching, and stereoscopic ranging. These limitations impose considerable frame-rate limitations, especially for high resolution images.

Structured Light range imaging is another technique for generating depth maps from a scene. This technique involves projecting a known pattern of light onto a scene, and inferring the depth at each pixel based on the patterns distortion as it falls onto the scene. Cameras which utilise this technique tend to have moderate depth precision, and good resolutions and frame-rates, however, the resolution of the projected pattern can often impose significant limitations on the feature resolution which can be captured, limiting the minimum size of resolvable objects in the scene. It is possible to utilise high resolution projections, or multiple different projected patterns to improve the feature resolution of this technique, however these approaches often come at significant cost to the achievable frame-rate.

Scanning Laser Rangefinders (and LiDAR systems) are another range imaging technology that are widely utilised in robotic applications. These systems work by sampling the range between the device and various scene elements directly by sending a laser pulse at various angles and measuring the time it takes for the light to reflect from the scene. Typically, these sample points comprise a grid, from which a depth map can be constructed. These systems are characterised by high accuracy/precision, and range, however the mechanical nature of the laser reflector, coupled with the serial data acquisition strategy means that such devices tend to have extremely low resolution and frame rates. Industrial devices of this type are also very expensive.

Time-of-flight cameras are also a technology that utilise a relatively modern approach to range imaging. These cameras work by illuminating a scene with a light source and measuring the time it takes light to travel from the camera and reflect from various objects in the scene. Typically this is achieved by modulating the light source at a known frequency. Each pixel in the scene is then able to calculate the phase shift between incoming light, and the modulation signal, from which the time-of-flight can be inferred. Depth can then be calculated based on the time-of-flight calculated at each pixel, and the known speed of light through air. These cameras are capable of producing depth maps with moderate resolutions and precision at extremely high frame-rates. Additionally, the low computational complexity of the techniques utilised by these cameras means that hardware costs are relatively low.

An overview of sensor technologies for asparagus harvesting [31] revealed that Time-of-flight (ToF) imaging is a particularly well suited range imaging technique for this application and as such is the primary imaging technology utilised in this work.

As with any organic structure, commercial asparagus fields are relatively messy and unstructured. Such environments are known to be challenging for image processing techniques to accommodate. In order to realise a robotic harvester, the constituent imaging system must be capable of interpreting ToF data in a deterministic way and producing meaningful outputs which correspond to target spears.

1.3 Objective

The aim of this thesis is to present a novel perception pipeline for ToF images that can facilitate the robotic harvest of selected, harvest-eligible, green asparagus spears in a real-world commercial setting.

In order to achieve this the perception pipeline should:

- Detect, and localise asparagus spears in ToF images taken of commercial asparagus beds
- Generate a model which informs the robot about the form of each detected spear, and use this model to find the base point of each spear
- Make decisions based on this model to select harvest-eligible spears
- Track target points in world-space to enable coordination with harvesting hardware

1.4 Methodology

The research methodology in this work has a distinct focus on real-world application. As such, simulated environments were avoided, and all methods, systems, and algorithms developed were configured and tested based on data captured from real hardware, in both laboratory and commercial settings. Furthermore, all commercial asparagus rows, from which this data was collected, were not manicured, or prepared in any way that simplified either the detection/localisation or harvesting tasks.

Multiple data-collection platforms were developed throughout this research. These platforms ranged from static frames and hand-powered trolleys to tractor towed robotic platforms. Various datasets were collected by operating these platforms on commercial asparagus farms throughout New Zealand, and in Los Banos, California. These datasets included RGB images, encoder outputs, 3D pointclouds and infra-red (IR) intensity maps. Generally, such datasets were recorded during motion as a time-series. This approach allowed the recorded datasets to encapsulate various complicating factors, such as motion blur and mechanical vibration, often associated with motion.

A ROS (Robot Operating System) [32], environment was utilised to both allow visualisation of the recorded datasets, and to facilitate the "replaying" of data using the ROSBAG system. This enabled the developed systems to run in an "offline" setting, where input datastreams were populated by his-

torically recorded data. Rapid experimentation was then possible, facilitating the development and parameterisation of various algorithms.

A general philosophy of minimising the required processing time was adopted for this research, in order to realise an applicable, real-time solution. As such, the processing strategy involved systematic removal of irrelevant points from the ToF images, rather than generation of high fidelity models.

Once the system had reached an appropriate level of maturity, a proof-of-concept harvester was developed. This machine was operated in Los Banos, California and provided a wealth of knowledge about the operation of the developed perception pipeline. Additionally, field trials of the robotic platform allowed a clear understanding of the practical limitations of data-collection methods to be gained. This facilitated the development of a robust evaluation method.

Based on knowledge gained from the proof-of-concept harvester, a final asparagus harvesting robot was constructed. The aforementioned evaluation method was applied in order to objectively measure the performance of both the harvester and constituent perception system.

1.5 Principal Contributions

- A novel concept for a selective robotic asparagus harvester. This concept builds upon previously existing selective asparagus harvesting robots from the literature and addresses the limitations of these machines, such as poor detection resolution
- A novel perception pipeline for ToF images for the detection, and localisation of green asparagus spears in a commercial setting. The perception pipeline utilises modern computational hardware, and methods to achieve state-of-the-art performance with respect existing asparagus harvesting robots

- A method for utilising real-time convolutional neural network (CNN) based methods for coarse filtering of input data, allowing subsequent processing methods to operate over smaller, more information dense, datasets resulting in reducing processing time
- A frame-based approach to simultaneously track asparagus spears between frames and improve localisation precision
- A procedure for determining a ground truth dataset of an asparagus row for evaluation of a selective asparagus harvester. This procedure mitigates the considerable practical limitations imposed by the growth characteristics of asparagus spears, resulting in ground truth datasets that are significantly more comprehensive than existing datasets from the literature
- A method for evaluating selective harvesting robots as a binary classifier, allowing an objective evaluation of the robots performance. Evaluating robots with this method enables future work to generate meaningful comparisons of system performance. Additionally, application of this evaluation method to the robotic systems developed in this work provides an objective performance baseline for future researchers to build upon
- A geometric filter for mitigation of flying pixels from ToF representations of asparagus spears. This filter enables significant improvements to the geometric representation of the asparagus spears, improving the accuracy of the subsequent base point predictions
- A geometric method for determining asparagus spear models from point clusters. This method allows data intense pointcloud representations of target spears to be converted into lightweight spear descriptions, allowing a low overhead description of a scene to be constructed
- A modification to Hyun's method that improves the method's resilience to input noise at the cost of feature resolution. This allows the method

to operate on noisy pointclouds such as those captured under non-ideal imaging conditions

- An evaluation of FRCNN (Faster Region-based Convolutional Neural Network) applied to real-world asparagus beds. Additionally, the labelled datasets used to train the various FRCNN models utilised in this work are publicly available at: `https://github.com/MPeebles/AsparagusDatasets.git`

1.6 Published Work

- M. Peebles, J. J. Barnett, M. Duke, S. H. Lim, Robotic Harvesting of Asparagus using Machine Learning and Time-of-Flight Imaging Overview of Development and Field Trials, 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE2020), Hong Kong (online), 20 August 2020, Pages 1361-1366
- M. Peebles, S. H. Lim, B. McGuinness, M. Duke, Identification of Failure Conditions for Robotic Harvesters Vision System, Australasian Conference on Robotics and Automation 2019 (ACRA2019), Adelaide, Australia, 9 December 2019
- M. Peebles, S. H. Lim, M. Duke, B. McGuinness, Investigation of Optimal Network Architecture for Asparagus Spear Detection in Robotic Harvesting, 6th IFAC Conference on Sensing, Control and Automation for Agriculture (AgriControl), Sydney, Australia, 4 December 2019, Pages 283-287
- M. Peebles, S. H. Lim, L. Streeter, M. Duke, C. K. Au, Ground Plane Segmentation of Time-of-flight Images for Asparagus Harvesting, International Conference on Image and Vision Computing New Zealand 2018 (IVCNZ2018), Auckland, New Zealand, 19 Nov 2018, Pages 1-6

- M. Peebles, S. H. Lim, M. Duke, C. K. Au, 2018, Overview of Sensor Technologies Used for 3D Localization of Asparagus Spears for Robotic Harvesting, Applied Mechanics and Materials, Volume 884, Pages 77-85

1.6.1 Published Videos

1. AHR-1 during field trials in California: `https://youtu.be/KA85fJ9eM6g`
2. AHR-2 operation overview: `https://youtu.be/lbb67G1djvc`
3. AHR-2 during field trials in New Zealand: `https://youtu.be/2e2K36E8MI4`
4. AHR-2 top down view of end-effector in operation: `https://youtu.be/m3EeCMHMcWk`
5. AHR-2 operating at various speeds: `https://youtu.be/mJXiLBMOHTA`
6. CASE2020 Presentation: `https://youtu.be/KvMYxg1FU-o`
7. A case study video produced by New Zealand Trade and Enterprises (NZTE) describing AHR-1's contributions to NZ agritech: `https://www.youtube.com/watch?v=bY17Dx1eLT4`

1.7 Outline of Thesis

The thesis is structured as follows:

- Chapter 2 provides an overview of the existing literature surrounding agricultural robotics, CNNs and ToF imaging for crop detection. The chapter then presents literature surrounding robotic asparagus harvesting which is critically analysed with respect to the imaging technologies, processing methods, and evaluation techniques, in order to provide insight on the current state-of-the-art performance of existing robotic asparagus harvesters
- Chapter 3 presents the theoretical basis of the novel perception pipeline and provides insight into the philosophy by which it was developed

- Chapter 4 discusses implementation of the theoretical perception pipeline into a functional vision system. This chapter also discusses the integration of the vision system into an initial proof-of-concept robotic platform
- Chapter 5 discusses field trials of the proof-of-concept machine and subsequent development of both the hardware and software systems. Following this development, this chapter presents, and analyses the performance of a working robotic harvester
- Chapter 6 presents the conclusions of the thesis and speculates on the potential impact of future work

Chapter 2

Literature Review

This chapter begins by exploring the literature surrounding agricultural robotics and robotic harvesters. Following this, an exploration of CNN methods, and ToF imaging is presented.

The chapter then critically analyses the literature surrounding robotic asparagus harvesting, beginning by discussing previously existing selective asparagus harvesting robots, and the various technologies which comprise their respective asparagus detection systems. The discussion then expands to include literature pertaining to a multitude of asparagus spear detection methods which have been published outside the context of a full harvesting robot.

2.1 Agricultural Robotics

Advances in automation, sensor and GPS technologies are beginning to drive widespread adoption of advanced technologies in the agricultural industry worldwide [8–11]. Many industry leading providers of agricultural equipment, such as John Deere, already supply various autonomous, and semi-autonomous tractors, and other specialised equipment throughout the world. Today, the scope of such advanced systems is relatively limited, with their primary uses relegated to autonomous navigation and mapping for precision planting applications. However, with labour costs on the rise a great deal of work is underway which aims to increase the variety of tasks achievable by such machines.

A number of researchers are working on autonomous robots for the purposes of crop surveillance, weed spraying, and yield estimation. BoniRob [13, 14], a robotic weeder presented by Ruckelshausen et al. utilised an AI system to discriminate between weeds and various other plants. The system was equipped with a mechanical weeding tool, with which weed species were robotically removed. A similar system known as AgBot II [15], was developed by Bawden et al. which, in addition to mechanical weeding, was capable of chemical weeding or targeted fertilisation by spraying. These machines, although not yet commercial, are at a reasonably high level of development. In addition, a number of other research robots, such as, Hortibot [16], Autonome Roboter [17] and RIPPA [18] have been developed to investigate a variety of elements related to weeding and crop estimation. In general these machines are in a relatively applied state, existing as full robotic systems capable of interacting with real world farms.

In contrast to the aforementioned machines, robotic systems capable of more complex, or interactive, farming tasks, such as robotic harvesters are in a much less developed state [33]. Commercially operated robotic harvesters, such as Cerescon's SPARTER harvester for white asparagus spears, do exist, however generally only target crops were selection is not an issue, and the growing systems are simple. For these reasons machines capable of selective robotic harvesting remain generally under-developed. This is likely due to the high cost of the sensors and robotic elements required to achieve these more complex agricultural tasks as well as the relative immaturity of the constituent technology. A review of state-of-the-art robotic harvesting technologies, conducted by Bac and van Henten [12] reported that apples, tomatoes, strawberries, citrus fruit and green asparagus are the most targeted crops for automated harvesting research.

De-An et al. [19] published work on a robotic apple harvesting robot. The robot consists of a 5-DOF robotic arm, equipped with a vision system and pneumatically operated end-effector. The robot utilises a visual servoing

technique to maneuver the end-effector towards target apples for harvest. De An et. al. stated that their machine achieved a harvesting success rate of 77% with a per-fruit cycle time of 15s. While these results are commendable from a technical standpoint, the resulting machine is unlikely to be commercially competitive due to its slow cycle time and high harvesting failure rate when compared to human harvesters. Another apple harvesting robot was developed by Baeten et al. [20]. The machine utilised a 6-DOF industrial robot arm, with a silicone funnel shaped end effector. The vision system adopted a eye-in-hand approach, and localised target fruit based on visual servoing. The researchers reported a 80% detection and harvesting rate with a cycle time of 9s, offering promising improvements over De-An et. al's previous work, however likely still short of commercial viability.

Muscato and Prestifilippo [21] developed an orange picking robot, designed for 3D orchard trees. In a 3D orchard, trees are grown in their natural state, with foliage able to grow in all directions. This is in contrast to more modern 2D orchards, where trees are trained into relatively planar structures. The machine utilises a novel cylindrical harvesting arm that maximises the available reach of the robot. A eye-in-hand vision system, coupled with a novel end-effector is able to detect, localise and cut the stem of oranges from 3D trees. The eye-in-hand approach adopted by the researchers allows high detail close up images of target fruit to be evaluated throughout the harvesting process. The researchers reported a per-fruit cycle time of 8.7s; too high to enable the robot to replace human labour, falling short of the required cycle time of 6s per fruit reported in the study. Lee and Rosa [22] developed an orange harvester that aimed to minimise some of the difficulty associated with 3D trees with a novel canopy volume reduction technique. Their machine compressed section of the orange tree with hydraulic powered metal tubes. This approach concentrated the target oranges, allowing the pneumatically powered harvesting system to more effectively harvest the fruit, achieving a reported removal rate of 84%.

Kondo et al. [23] developed a robotic harvester targeting tomato clusters. The machine is designed to be operated inside a greenhouse environment and utilises a stereo vision pair, consisting of two CCD cameras to detect, localise and harvest tomato bunches. In order to avoid damage to the fruit, the robot targets the peduncle of a cluster of tomatoes, avoiding contact with the edible part of the fruit completely. The researchers reported a 65% detection success rate, however, more work is required to achieve satisfactory mechanical harvesting.

Barnett et al. [24] have published work on the development of a robotic pollination device for kiwifruit orchards. The machine consists of a mechanically operated boom, which is controlled to maintain a constant distance from the overhead kiwifruit canopy as it traverses the orchard. An onboard vision system detects kiwi-fruit flowers, and deploys a directed spray, containing pollen, at each target flower. The researchers reported that the robot is capable of detecting 70% of flowers, 80% of which were successfully sprayed, at a ground speed of 0.36m/s.

A robotic kiwifruit harvester, based on Scarfe's work [25], was reported by Williams et al. [26]. The work outlines the development of a four-armed robotic system capable of detecting, localising, and harvesting ripe kiwifruit from beneath the canopy of a commercial kiwifruit orchard. The results of this study stated that the harvester was able to harvest 51% of the kiwifruit crop, with a per-fruit cycle time of 5.5s. However, the researchers stated that a significant portion of missed fruit (25%) are lost due to mechanical problems with the gripper. Williams et al. speculate that further development will result in a system capable of harvesting 70% of the crop.

The robotic systems covered in this review are not yet commercially viable for a variety of reasons. Mainly, these systems lack the harvesting speed and detector accuracy required to rival human harvesters. Generally, it is difficult to ascribe concrete requirements for such robots to be considered commercially competitive due to the large number of factors which affect the industry,

and the complex interaction between technologies, growers and the communities they serve. However, metrics such as harvesting speed, operating hours, and machine cost are likely among the most important factors to consider. The limited adoption of these technologies in the commercial sector provide implicit proof that currently achievable harvesting rates, system robustness, and machine costs are not yet sufficient to justify the use of such robots in commercial settings. The most successful of the robots in this review have utilised convolutional neural network (CNN) based methods for the detection of various crops. The following section provides an overview of these methods applied to crop detection.

2.2 CNN Methods for Crop Detection

Machine learning is a booming area of research due to modern day advancements both computational power, and computational density [34]. CNNs, a subset of this space, have been shown to achieve state-of-the-art performance in a variety of image classification, and object detection tasks, particularly in unstructured environments [35–37]. As such, it is known that such methods are well suited for crop detection, however in an applied setting real-time performance is key. There are three noteable CNN architectures designed primarily for real-time performance, namely Faster Region-based CNN (FRCNN) [38], "You Only Look Once" (YOLO) [39], Yolov3 [40] and Single Shot Multibox Detector (SSD) [41]. Many researchers have developed a variety of CNN models based on these architectures for the purposes of crop detection.

FRCNN models were the most commonly identified during this review. "Deep-fruits" [42], a FRCNN model developed by Sa et al. achieved the F1 scores at shown in Table 2.1 at a framerate of 5FPS from images taken in both laboratory, and greenhouse conditions. F1 score is a common metric for CNN performance and is discussed in Section 4.3.2.2. The method utilised a multi-modal approach fusing the results of FRCNN detections of simultaneous

Table 2.1: F1 Scores achieved for various fruits by the "DeepFruits" [42] FR-CNN model.

Fruit	**F1 Score**
Strawberry	0.948
Mango	0.942
Apple	0.938
Avocado	0.932
Orange	0.915
Rock Melon	0.848
Sweet Pepper	0.828

RGB and NIR images. These scores are very impressive, approaching human performance.

Bargoti and Underwood [43] achieved similar performance by applying FR-CNN to fruit detection in orchards, achieving F1 scores of 0.904, 0.908 and 0.775 for the detection of apples, mangoes, and almonds respectively from images taken in commercial orchards. The researchers reported that the relative poor performance achieved for almonds in their study was due to the small size of the crop with respect to the image resolution.

FRCNN was also successfully applied to commercial kiwifruit orchards by Fu et al. [44]. The researchers achieved a recognition rate of 82.5%, 85.6%, 94.3%, and 96% for occluded, overlapping, adjacent, and separated fruit respectively, with a processing time of 0.274s. The results of this study demonstrate the resilience of CNN methods like FRCNN to accommodate unstructured environments.

Another study which highlights the resilience of FRCNN to unstructured environments was conducted by Gao et al. [45]. The study was concerned with detecting apples in-situ on trees grown using a SNAP (simple, narrow, accessible, and productive) architecture in a commercial orchard. The FRCNN model, developed in this study, achieved a mean average precision of 0.879 on

apples in various occlusion states involving the fruit, branches, leaves and grow-wires.

Researchers have also investigated modified versions of FRCNN that incorporate depth information in addition to the traditional RGB channels, achieving improved detector performance [46, 47]. These researchers achieved an average precision of 0.893 in the detection of apples, and a F1 score of 0.885 for the detection of passion fruit respectively.

Mask-RCNN [48] is a modification to FRCNN that outputs a segmentation mask for each detected instance. Chu et al. [49] achieved a F1 score of 0.905 with a detection time of 0.25s for the detection of apples . Although this F1 score is impressive, the added specificity that the segmentation mask provides seems to come at a significant cost in terms of processing time.

Ganesh et al. [50] applied Mask-RCNN to the detection of oranges in commercial orange groves. The researchers investigated how varying the colour space of training images, specifically RGB and HSV, effected the performance of the model. It was found that inclusion of both RGB and HSV colour spaces increased the F1 score from 0.881 (RGB only) to 0.887 (RGB + HSV).

Yu et al. [51] applied Mask-RCNN to the detection of strawberries, with an emphasis on non-structured environments. The researchers found that the model achieved 95.78% recall with 95.41% precision. Furthermore, it was reported that the average localisation error was $\pm$1.2mm. The researchers stated that this localisation error is far below precision requirement of a proposed harvesting end-effector. It can therefore be concluded that robotic harvesting is likely possible.

SSD, Yolo, Yolov3, and a novel extension of Yolo named "MangoYolo" were applied to the detection of Mangoes by Koirala et. al. [52]. Table 2.2 details the F1 scores and associated inference times achieved by the various models in this study. From this study it appears that both "one shot" detectors outperformed FRCNN in terms of both F1 score, and inference time. It is

Table 2.2: F1 Score and inference times of various models for the detection of mangoes in a study conducted by Koirala et al. [52].

Model Architecture	F1 Score	Inference Time (ms)
FRCNN	0.936	67
SSD	0.950	46
Yolo	0.916	20
YOLOv3	0.951	25
MangoYOLO	0.968	15

difficult however to draw conclusions on this performance without testing a wider variety of crops.

Vasconez et al. [53] conducted a comparison between FRCNN and SSD for the detection of avacados, lemons and apples under field conditions. The researchers applied the outputs of these networks to count the fruit present in video footage. It was determined that FRCNN achieved a counting performance of 93%, while SSD achieved 90% across all fruit. The underlying FRCNN and SSD models achieved a mean average precision of 77%, and 57% respectively.

The real-time CNN architectures utilised in these reviewed works have shown to be suitable for a variety of crop detection tasks and have shown resilience to unstructured real-world environments.

As identified in the previous chapter, ToF imaging has shown promise as a detection technology for robotic harvesting. In the following section an overview of ToF imaging for the purposes of crop detection is provided.

2.3 ToF Imaging for Crop Detection

The principal of ToF as a method for range-finding has existed for some time. Scanning laser rangefinders, and ultrasonic sensors are some examples of devices which utilise this method [31]. More advanced technologies exist which apply this principal on a per-pixel basis, allowing real-time acquisition of depth

maps. These devices are known as ToF cameras [54]. Research utilising these cameras increased five-fold from 2006-2008, and continues to rise due to a general decline in the overall cost of such devices [55, 56].

The release of Microsoft's Kinect V2 ToF camera in 2014, a piece of consumer electronics designed for the Xbox gaming console, has opened the door for widespread research into ToF applications by significantly reducing the cost of such technology [57–60]. As such a number of researchers have investigated application of the technology to the task of crop detection and mapping [61–63], however limited work has been done on utilising this data for crop localisation [64].

Various researchers have utilised ToF cameras as a way to augment existing vision systems, improving performance. Gai et al. [65] applied depth images, obtained from a ToF camera, to segment various instances of weeds, and crop in the foreground of images from the background soil. An increase in segmentation success rate from 87.2% (RGB only) to 96.6% was observed when depth information was included. A similar study by Gene-Mola et al. [66] augmented RGB data with depth information from a Kinect V2 ToF camera to train a modified FRCNN network for the detection of Fuji apples. The researchers stated that all range images taken during this study were taken during the night, as to avoid the well known limitations of ToF pixel saturation in direct sunlight [67]. A 4.46% improvement in F1 score was observed when utilising this information.

Researchers have also investigated the direct use of pointclouds generated from ToF cameras for the detection of crops. Li et al. [68] developed a system for detecting broccoli and green beans, with an emphasis on weedy conditions, that utilised a ToF camera. This work was conducted for the purpose of robotic weeding of ground crops. In order to mitigate the effect of direct sunlight, the researchers covered the imaged section of the field with a sunshade. The system achieved a detection success rate of 88.3% and 91.2% for broccoli and green beans respectively.

Wong and Lim [69] also developed a system which operated on pointcloud data from a ToF sensor for the detection and localisation of ground based crops for the purpose of robotic harvesting. The study focused on the detection of foremost spring onions within a row on commercial fields. The researchers reported that their system successfully located between $96.4\% - 100\%$ of crops from a test set of 1000 frames. This is a promising advancement for the robotic harvesting of such crops.

Crop mapping is another field where ToF cameras are being applied. Vázquez-Arellano et al. [70] utilised ToF cameras to develop a 3D reconstruction of maize crops for the purpose of crop phenotyping. The researchers utilised a robotic platform, called TALOS [71], equipped with a Microsoft Kinect V2 ToF camera for imaging maize crops in a greenhouse nursery. The robotic platforms position was tracked using a combination of an IMU and SPS930 base station. The detected maize crop positions agreed closely with the ground truth, with a mean and standard deviation of 3.4cm and $\pm$1.3cm respectively. A subsequent study [72] conducted by Vázquez-Arellano et al. aimed to determine the stem position and height of the reconstructed plants following application of this method. The researchers concluded that ToF imaging was an effective means for determining the stem locations, and height of maize crops.

This review has shown ToF imaging to be a promising new technology for the purposes of crop detection. In particular, ToF imaging seems suited for the detection of ground crops, where direct sunlight can be avoided.

The following section provides an extensive overview of the literature surrounding selective asparagus harvesters, as well as various detection methods for green asparagus spears. The literature is critically analysed with respect to the applicability of the proposed methods for a real-world commercial setting.

2.4 Current Asparagus Harvesters

Demand for a robotic selective asparagus harvester has existed since the early 1950s. Academic literature regarding the earliest machines is scarce, however a number patents can be sourced which describe the early developments. Machines such as the: Matteoli [73], Turkington [74], Chatagnier [75], Franzen [76], Marmorine-Lawson [77], and Lawson harvesters [78], as well as early versions of the Kim-Haws harvester [79, 80], utilised tactile sensors for the detection of asparagus spears. These sensors, although varying in their respective implementation, all operate on the principle of physical contact with the target spear to achieve detection. Little information is known about the specific implementation of the sensors used in these early machines. For this reason little is known about the specific sensor outputs. However, conceptually such sensors would seem limited to either binary go/no-go, or continuous analog outputs based on 'feeler' deflection. A 2010 review, conducted by Chen et al. [81] compared the efficacy of a variety of asparagus harvesting technologies. The researchers found that 'contact based' sensors tend to result in unreliable harvesting due to the unpredictable interaction of asparagus spears and the sensor elements. The researchers argued that the varying stiffness and diameter of individual spears greatly impacted the force observed by the sensor unit, leading to unreliability and loss of yield.

Moore's 1964 harvester [82] utilised an alternative approach for detecting asparagus spears. The machine utilised an air stream directed laterally across the asparagus row using a nozzle. The air stream was positioned such that the air pressure held open a nominally closed switch on the other side of the row. Passing spears of acceptable height would block the air stream, thereby closing the switch. Moore reported that this method malfunctioned frequently due to vibration and environmental wind. To improve the performance, Moore replaced the air based detection system with a more standard array of tactile sensors. The varying spear stiffnesses and diameters introduced significant error with this detection method. Finally, Moore replaced the tactile sen-

sors with a 'photoelectric unit' which presumably operated on the principle of spears blocking line-of-sight between a light source and a receiver unit. Moore reported trouble in optimising the combination of the receiver units sensitivity and the speed at which other electro-mechanical devices could operate.

Work continued on developing a robotic selective asparagus harvester through the 1980s and early 1990s in both the industrial and academic spheres. Arndt et al. working with the University of Wollongong's Centre for Advanced Manufacturing and Industrial Automation (CAMIA)[83, 84] were among the first to formally research the problem of selective asparagus harvesting. They developed the CAMIA harvester, which utilised similar machine design principals to Franzen's 1965 harvester, including the use of an actuated blade to cut the spear bases at an angle, and a rotating drum mechanism for spear retrieval. The spear detection system of the CAMIA harvester was modernised by replacing each of the tactile sensors with a photoelectric 'beam-type' sensor. These sensors were arranged in a series of gates that spanned across the asparagus row and operated on a similar philosophical principal to the early harvesters. The mounting height of each sensor was such that any spears of harvestable height, that passed through a gate due to the forward translation of the machine, would break the beam producing a binary 'go/no-go' response, thereby being detected. Machines developed by the industrial sector around this time, such as the Geiger-Lund harvester [85] as well as more modern advancements, such as the Kim-Haws harvester [86] have also utilised photoelectric 'beam-type' sensors in a similar way.

Photoelectric 'beam-type' sensors have seen a relatively high degree of success. Arndt et al. [83] reported that the CAMIA harvester achieved ground speeds of up to 10km/h and a total spear loss of 15% during field trials in Jugiong, New South Wales, Australia. It was reported that this loss was calculated based on the number of spears that were missed, dropped, or damaged by the harvester, presumably in comparison with the expected performance of manual harvesters. These results are impressive, however details regard-

ing the ground speed, and weather conditions experienced during the trials are unclear. In 1995 the Washington Asparagus Commission conducted a series of field trials to compare the harvesting performance of both the CAMIA and Haws harvesters with two non-selective harvesters, namely the Swather and Snapper [84]. The performance of the CAMIA harvester was poor during these trials as the geometry of the American beds was significantly different from the Australian beds for which the machine was designed. The researchers stated that commercialisation of their machine would solve many of the issues faced during these trials. Specifically, it was estimated that a 3-row (rather than 1-row) device, with more reliable sensors would be capable of 10% faster speeds and be capable of harvesting 80 acres of asparagus in 1000 hours. In contrast to the CAMIA harvester, the Haws harvester saw a relatively high degree of success during these trials, reportedly producing only 13% less yield than manual harvesting. However, the speed at which this was achieved is unclear. Additionally, the machine's multi-pass design philosophy means that the overall efficiency of the machine was likely quite low.

The Geiger-Lund harvester [85] underwent an independent field trial during April of 2006. This harvester utilises photoelectric 'beam-type' sensors similarly to the CAMIA and Haws harvesters. Likewise, the detection system consists of many gates which span the asparagus row, each gate containing spear sensors. The Geiger-Lund harvester differs slightly from the CAMIA and Haws harvester by using a pair of photoelectric 'beam-type' sensors for each gate. The sensors are arranged vertically such that the top sensor can detect if a particular spear is tall enough to qualify for harvest, and the bottom sensor can locate the base (cutting location) of the spear. This allows the Geiger-Lund harvester to accommodate for a wider range of spear morphologies.

An economic analysis of the Geiger-Lund harvester was performed by Clary et al. [87]. The researchers reported that, on average, the Geiger-Lund harvester was able to retrieve 70% of the marketable spears. An additional 10%

of spears were identified as 'stringers' which described situations where the harvesters cutting mechanism was not successful. The total detection rate of this system was therefore reported to be 80%. However, contradictions in figures published by Clary et. al. indicate potential methodological flaws in their analysis. It is reported that on average the Geiger-Lund harvester produced a pay-weight of approximately 50% of the expected yield from manual harvesting. If it is assumed that human harvesters are reasonably efficient it seems unlikely that such a yield could constitute 70% of all marketable spears. Furthermore, the reported 50% of pay-weight achieved by the harvester pertains to data from dates not published in the study. Instead, the related figure provides data pertaining to a separate period, during which an average of 40% of manual harvester's pay-weight was achieved.

In 2012 the University of Bremen and several industrial partners, with a large amount of support from the European Council, began work on the AmLight project [88]. The goal of this project was to develop a robotic selective asparagus harvester for commercial use. The project resulted in the AmLight harvester which utilised a stereo pair consisting of two CMOS based sensors for asparagus spear detection. The system worked by performing a white balance on the unrectified images from each camera in the stereo pair. This normalises the colour channels to allow for changes in ambient lighting conditions. Spears are then segmented from the background by colour, allowing spear base features to be detected in each image. These features can then be localised stereoscopically. The AmLight harvester reportedly achieved a detection success rate of 70% with a precision of $\pm$3cm during field trials. However, limitations imposed by the slow speed of the image processing system coupled with problems maintaining a straight line trajectory on the real-world fields meant that the AmLight harvester failed to harvest any spears during field trials.

The GARotics project followed on from the AmLight project with the goal of improving the asparagus detection system. Leu et. al. [89] presented an

overview of the updated system. The updated system replaced the stereo camera pair with a Microsoft Kinect V2 ToF camera. Such cameras provide depth information at each pixel, allowing 3D point-clouds of the scene to be constructed. Rather than relying on colour information for spear detection, like the AmLight harvester, the GARotics harvester analyses the pointcloud geometrically to detect asparagus spears. Leu et. al. reported the uses of Random Sample Consensus (RANSAC) [90] and 'Euclidean cluster extraction' for the detection of spears, however details on the implementation, and specific algorithms used are not known.

The CAMIA, Geiger-Lund and AmLight harvesters are reported to have harvested approximately 70% - 85% of all marketable spears. While these figures are impressive, the respective analyses failed to provide insight on the proportion of harvested spears which were not harvest eligible (false positive rate) for each machine. A definition of the false positive rate can be found in Section 5.4.3. It is paramount to the success of a robotic harvester that target spears be selectively harvested in order to preserve potential yield. Without knowing the false positive rate it is difficult to assess how selective each of these machines are. This is an extremely important factor because the selectivity of an asparagus harvesting robot greatly effects its ability to be commercially utilised.

Researchers have also worked on asparagus detection outside the context of a complete harvester. The following section presents the various sensing methods and technologies in such studies.

2.5 Asparagus Detection

Baylou et al. [91] developed a machine vision system for the detection of emergent tips of white asparagus spears. The system consisted of two CCD cameras arranged as a stero-pair directed laterally across the asparagus row. On the opposite side of the row a light-box was positioned that was used to

back light the images. This produced high contrast monochromatic images of the growing-mound and spear tip silhouettes. A binarisation process was used to define a clear contour between the background and scene elements. The threshold of binarisation is determined automatically by finding the mean y-value where the dark scene components separate from the light background and plotting a curve of this value for a series of different binarisation threshold values. The resulting curve is analysed to find a plateau and the corresponding binarisation threshold is used by the system. Spear tips are then identified geometrically by considering local peaks in the images and localised stereoscopically. While this approach is innovative, the researchers do not publish any metrics of performance and so it is difficult to judge the applicability of the proposed methods.

Another machine vision system for asparagus detection was developed in 1990 by Humburg et al. [92]. This system utilised a single monochromatic CCD camera for spear detection. The camera was mounted facing laterally across the asparagus row at a downward angle of 45°. Behind the camera system an array of incandescent bulbs and reflectors were positioned in order to illuminate the scene. The camera was fitted with an optical band-pass filter centered around the near infra-red range (100nm band centered at 850nm). This range was used as it was found to provide the highest contrast in reflectivity between asparagus spears and background elements. Furthermore, the infra-red component of light from incandescent bulbs is significant. A binarisation process is firstly applied to the images using a manually tuned threshold. The binarisation threshold was manually altered throughout testing in order to maintain good separation of spears and background elements. Blobs that remained after binarisation were analysed to separate spear features from noise. This was achieved by eliminating blobs that did not conform to specific shape requirements such as vertical span, width and size. Once spear features had been determined a perspective transform was used to determine real-world target locations based on image-space coordinates. The perspective transfor-

mation matrix was calculated by manual calibration and was determined for each row prior to testing. Humburg et al. [93] published an evaluation of field performance for this system. During this trial, six 15m long row segments were imaged. For each section a perspective transformation matrix was determined using standard calibration techniques. Data was collected by recording video of the asparagus bed as the machine translated down the row. Spear features were detected offline based on this video. This is primarily because the image processing could not be done in real-time. The researchers reported that the vision system correctly identified 86% to 97% of harvestable spears during this trial. Additionally, it is stated that the spears were localised within a 2.97cm by 5.39cm window with 95% confidence.

Grattoni et al. [94] developed a system for asparagus detection. Their device utilised a stereo pair consisting of two monochromatic CCD cameras. The cameras were orientated so that they faced laterally across the asparagus row, with the optical axis of each camera parallel to the ground plane. On the opposite side of the asparagus bed was a black backboard, which blocked the busy background of the adjacent row, creating high contrast between spear features and background in the images. The scene was front-lit with relatively high intensity light such that the images were close to two-toned, with foreground features and asparagus spears being bright and the background being dark. The images were binarised using a constant threshold. Due to the nature of the lighting setup, the binarisation threshold did not need much tweaking and the images were relatively robust to changes in ambient lighting conditions. Edge detection was applied to the binarised images, and the vertical contours used as asparagus features. This resulted in a vertical line for both the leading and trailing edges of each spear in the image. The fact that each asparagus spear had two associated contours caused a degree of ambiguity in assigning the correct leading and trailing edges to the correct asparagus spear in each of the stereo images. Additionally, occlusion of spears due to grass/weeds or other spears added to this ambiguity. These ambiguities were

resolved by applying an ‘asparagus detection algorithm’[95] which relied on previously developed contour detection methods. The stereo pair was also mounted to a linear rail which was orientated in the direction of the asparagus bed. This allowed images of the scene to be taken at various precise locations, which allowed ambiguous features to be resolved. Each spear could then be localised stereoscopically. The researchers reported a total processing time of 3.5s per image. However, the processor used in this work had a clock speed of 25MHz. A modern CPU would be capable of much faster processing times. The researchers stated that laboratory experiments indicate that the vision system should work in “real operating conditions”, however do not provide any metrics of performance.

A prototype asparagus harvester, designed for use in Japanese greenhouses was developed by Irie et al. [96]. The asparagus crowns in this work were planted in raised planter boxes, which were shaped similarly to traditional asparagus beds. Throughout each row, several mature asparagus ferns are allowed to grow. These spears are known as the ‘parent spears’ and are able to provide the asparagus crowns with sufficient energy to sprout new spears throughout the season. This results in a forest like structure consisting of immature spears, harvestable spears, and parent spears. Between each planter box exists a rail system that allows precise positioning, and translation of the prototype harvester. The detection system utilises what the researchers call a ‘3D Vision Sensor’. The system consists of a ‘TV camera’ (presumably a colour camera with relatively high resolution) and two laser plane projectors. The camera is mounted on the harvester such that it faces laterally across the asparagus row, with its optical axis parallel to the soil plane of the planter box. The laser plane projectors are positioned above, and below the camera with their projected planes orientated such that they are parallel to the soil plane of the planter box. This distance between the two laser planes is set to the minimum desired spear length of 230mm. Since the separation of the laser planes is known, areas in the image where the laser is present can be

localised trigonometrically. This results in a 3D representation of the scene, which consists of points where each laser plane intersects a target. Points from each scan plane are presumably matched in order to produce a spear model. The researchers reported a maximum observed range error of 8mm during laboratory trials of this detection system.

Irie et al. [97] published work on a different asparagus harvesting robot, which utilised a scanning laser rangefinder (SLR) for asparagus detection. The harvester consisted of a 4-DOF robotic arm mounted to a platform that could traverse the asparagus beds using the aforementioned rail system. The SLR was mounted on the wrist of the robot, just above the end effector. To detect spears the robot translates down the row with the SLR orientated such that the scanning plane is parallel to the soil plane of the planter box. The laser plane is positioned at the height of the minimum desired spear length, in this case 250mm. When the SLR detects an object of interest the platform stops and the end effector is rotated so as to allow the SLR to range various positions down the length of the object. Each successive scan is collated to generate a 3D representation of the object. If geometric analysis concludes that the object is a spear, base coordinates are then calculated and the arm is directed to harvest the spear. The researchers stated that a scanning operation takes 2.0s and the harvesting action for a single spear takes 2.4s. Little information is reported about the accuracy of this detection method.

SLR technologies were also utilised by Sakai et al. [98] for asparagus spear detection. Their apparatus consisted of a carrier platform, and two SLR devices separated vertically by a distance of 160mm, orientated such that both their scanning planes were parallel to the soil plane of the planter boxes. The SLR assembly was translated vertically during the scanning procedure using an electric actuator. The scanning procedure took approximately 4s. Range measurements from both scanners were collated into a 3D representation of the scene. Points in this 3D representation were converted from their native polar coordinate system into a Cartesian coordinate system centered at the

laser origin. An image representing the partial derivative of the range data with respect to the row direction was constructed, and a threshold applied in order to discriminate between various objects in the scene. This method operates on the assumption of a large and sudden range disparity between spear features and background elements, and that spear features are predominantly vertical. The researchers reported that the system correctly identified 90% of the isolated spears, 76% of spears that cause occlusion of another spear, and 40% of spears that were frontally occluded.

A novel detection strategy, employing a multitude of monoscopic RGB cameras, for the detection of green asparagus spears was developed by Kennedy et al. [99]. The method begins by imaging a section of an asparagus bed with three cameras, arranged in a known geometry. The images are taken from within an enclosed environment, illuminated with artificial light in order to minimise the effect of variable lighting conditions. Images from each camera are processed in order to determine a probability map, highlighting pixels which likely pertain to asparagus spears. This is achieved by first migrating the colour-space of the image from RGB to YCbCr. Gamma correction using $\gamma = 1.5$ is then applied. The three principle axes from a principal component analysis are then computed, and the second axis is utilised as the probability map. Each image is then transformed based on the known camera extrinsics to a common base-plane. The resulting image, referred to by the authors as a "chicken foot image" contains multiple overlapping representations of each spear from the perspective of each camera. The compound image then undergoes a process of hypothesis estimation. A thresholding process, and subsequent morphological operations, are applied to "clean up" the image. These processes result in a singular blob for each asparagus spear, the eccentricity, size and location of which can be used to infer the individual spears position, height and width. Based on a field trial, the researchers concluded that the proposed method finds asparagus cutting points robustly, with a high level of accuracy.

Certain assumptions made by Kennedy et al. have, however, oversimplified the detection task. Firstly, it is stated in the study that asparagus spears grow straight out of the ground with no obscuring foliage. While conceptually this is true, real world settings are rarely this simple and reasonable weed growth is to be expected in even the most well maintained field. The inclusion of weeds or foreign debris will have a significant effect with respect to calculating the probability map, resulting in potential detection failure further down the perception pipeline presented in the study. Furthermore, the researchers stated that the method precludes horizontal spears from being detected. While this is useful in some cases, such as for ignoring fallen spears as implied in the study, there are potential cases where harvestable spears could be missed because of this. For example, spears which have not grown straight up out of the ground, but are slanted horizontally across the bed. Finally, the researchers failed to demonstrate the effectiveness of their method for dense spear clusters. Morphological methods utilised during the proposed hypothesis generation are unlikely to be able to effectively discriminate between such spears.

2.6 Summary

This chapter has discussed research regarding harvesting robots, CNNs and ToF imaging for crop detection.

It is clear from this review that robotic harvesters are in a generally under-developed state with respect to the requirements of the agricultural industry. Many of the robotic harvesters investigated in this review utilised off-the-shelf robotic manipulators, and vision systems that were generally lacking in robustness. As such, the current state of robotic harvesting is too slow, and unstable to be adopted for commercial use. It was found that robotic harvesters that were purpose built solutions generally fared better than their off-the-shelf counterparts. It is speculated that this is because such end-to-end solutions allow significant simplification of the task space, enabling closer to

real-time performance. Robotic harvesting tasks where crop-by-crop selective decision making was required generally resulted in poor performance among the reviewed harvesters. This highlights the difficulty of selective harvesting.

CNN methods for real-time image classification, and object detection were investigated with an emphasis on applications involving crop detection for harvesting. In general such methods were shown to achieve state-of-the-art performance for a variety of crops. Additionally, such methods were generally robust to changes in ambient lighting conditions, and unstructured agricultural environments. In particular, these methods proved to be capable of dealing with heavily occluded features. Of the model architectures reviewed, FRCNN was by far the most prevalent. Overall FRCNN seems well suited for the task of asparagus segmentation in images from commercial asparagus beds.

ToF imaging for crop detection was also investigated. It was determined that applications of ToF technology is a relatively new area of research, due to recent cost reductions associated with inclusion of the technology in the consumer market. From the reviewed studies, it is clear that ToF cameras are relatively sensitive to direct sunlight as they are susceptible to pixel saturation. As such researchers commonly conducted imaging tests during the night, or within indoor or otherwise sheltered environments. This limitation seemed to dictate that the technology was most appropriate for ground crops, such as broccoli, lettuce and spring onions. ToF imaging enabled excellent segmentation, and localisation results for these crops in the reviewed literature. In many cases these results were achieved despite sub-optimal weedy field conditions. As such, it is expected that ToF imaging is well suited for detecting asparagus spears in a commercial setting.

A critical analysis of various asparagus harvesting machines, and spear detection systems was conducted. Review of these machines revealed that current detection systems are not adequate to facilitate the selective harvesting task. With the exception of the AmLight and GaRotics harvesters, the most applied asparagus harvesters detect asparagus spears based on photoelectric sensors

mounted on a number of discrete gates. This detection strategy, while cheap and extremely fast, does not allow sufficient nuance about the field structure to be understood. This necessitates a needlessly aggressive harvesting routine which is likely to inflict significant collateral damage to the crop, failing the selective element of the harvesting task. Unfortunately, the literature makes no mention of the actual false positive rate of such harvesters.

With regard to the AmLight and GaRotics harvesters, little information is published. It was suspected, that the performance of these machines is limited when dealing with weeds, or other complicated field structures based on publicly available video footage of these machines in operation.

The following chapter presents a novel perception pipeline for the purpose of detecting and localising green asparagus spears in a commercial setting.

Chapter 3

The Perception Pipeline

The preceding chapter provided a critical analysis of the literature regarding asparagus harvesters and perception systems. The analysis revealed that ToF technology has seen some success in the detection of ground crops, even in the presence of weeds and other elements present in unstructured fields. The mechanics of ToF imaging offer a variety of advantages over contemporary perception technologies, such as high frame-rates, low computational complexity and the ability to operate in low-light scenes. ToF imaging is particularly promising for the detection of green asparagus spears since many of the effects stemming from the disadvantages of ToF imaging such as, the limited detection range, poor background light rejection, and limited resolution can be mitigated due to the crops unique growing environment.

This chapter outlines the basics of ToF imaging and provides a detailed description of a novel perception pipeline capable of processing images from a ToF camera in real-time for the detection, localisation and classification of asparagus spears in a commercial setting. The resulting software implementation of this pipeline is referred to as the "vision system" and its implementation, and development into a functional asparagus harvesting robot is presented in Chapter 4. The notion of real-time performance, in the context of an asparagus harvesting robot, is discussed in the following section.

3.1 Real-time Performance

In order for a robotic system to successfully harvest an asparagus spear the system must be capable of:

1. Data acquisition from sensors such as cameras and encoders
2. Processing of input data to detect relevant target spears
3. Localisation of target spears
4. Selection of target spears based on harvest eligibility and target scheduling
5. Path planning of end-effector
6. Manipulation of end-effector to facilitate the harvest of a target spear

In general, current agricultural robots have not seen wide adoption by the commercial sector, for a variety of complex reasons as discussed in Chapter 2. The complexity of these factors preclude a rigid prescription of required performance metrics, although cycle time, system robustness, and system cost are likely major contributors to commercial viability. It is therefore reasonable to expect that these metrics in particular should be a focus for improvements in order to make such systems commercially competitive. The speed at which the above operations can be performed greatly influences the maximum speed at which the overall robot can operate, and therefore greatly affects the spear throughput that can be achieved. For this reason, it is vital that each of these operations operate as quickly as possible in order to maximise the robot's performance.

For the purposes of this work the term "real-time" is utilised to encapsulate the concept of operating "as fast as possible". The limits of what can be considered real-time performance are expected to be determined empirically through analysis of overall system performance.

Despite the inability to prescribe hard performance metrics, it is instructive to determine a ballpark understanding of acceptable ground speed in order to contextualise the problem in a commercial setting. One asparagus farm that

was utilised in this work had approximately 100 hectares of growing area, consisting of 600km of row length. During harvesting season this farm currently employs upwards of 80 workers, costing an estimated $\$1,000,000$ per year. Allowing for a break-even period of two years, the total cost of a robotic solution should be less than $\$2,000,000$. If each robot has a total cost of $\$80,000$ this means that the daily harvesting task must be conducted by a total of 25 machines, requiring that each robot harvest approximately 24km of asparagus row per day. To achieve this, each machine needs to operate at a minimum of 1km/h, assuming that the machine can operate 24 hours per day. A more reasonable operational period of 10 hours per day prescribes an operation speed of 2.5km/h. Based on these figures, it can be concluded that a prospective asparagus harvesting robot should have an operating speed of between 0.278m/s and 0.694m/s.

3.2 Time of Flight Imaging Principles

ToF cameras are a class of active, scannerless LiDAR devices that leverage the relatively constant speed of light to generate a depth-map representation of a scene. A ToF camera consists of both a specialised image sensor, and a light source; typically a laser in the near-infrared range (800-2500nm). The camera operates by illuminating a scene with light from the on-board light source and measuring the round trip time of the light as it reflects from various objects in the scene. Applying this method on a per-pixel basis allows a depth value for every pixel to be resolved. The resulting image, which encodes both depth, and IR intensity at each pixel, is known as a depth-map of the scene. Such an image has the capacity to convey 3D information about the scene in the form of a pointcloud, formally defined in Section 3.2.2.

Most modern ToF cameras utilise a phase-correspondence method, rather than a simple "one-shot" method for capturing the depth information. The phase correspondence method works by modulating the light source periodi-

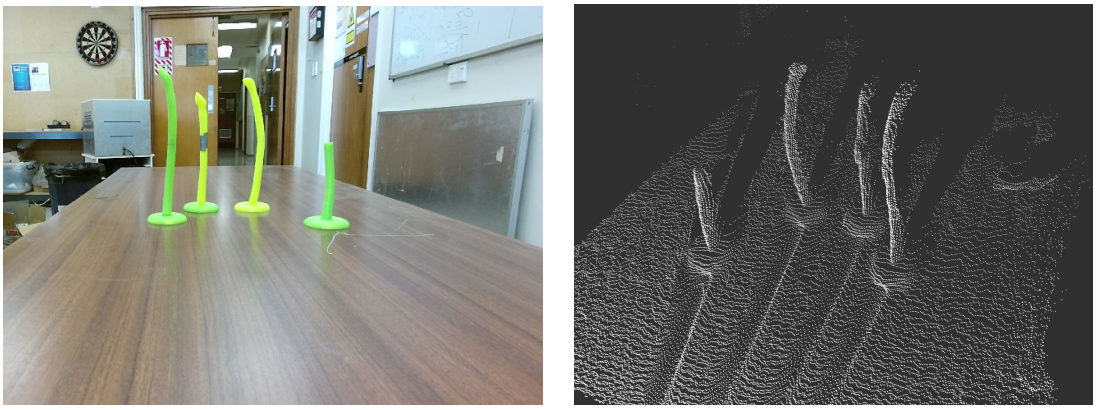
cally, and measuring the phase-shift between the modulated light signal and the returning signal reflected from the scene at each pixel on the image sensor over a number of frames. Generally such a camera operates by taking several raw phase images in quick succession, followed by a proportionally large processing time, during which a depth map is calculated. This approach can deliver depth-images at reasonably high frame rates, as high as 60 FPS for industrial grade cameras.

The mechanics of ToF cameras leave them vulnerable in a number of ways. These inherent weaknesses are described in the following subsection.

3.2.1 Weaknesses of Time-of-Flight Imaging

ToF cameras are known to exhibit multi-path errors when imaging scenes with surfaces that are reflective in the near-IR range. Multi-path errors occur when photons from the camera's light source reflect from multiple surfaces before returning to the camera. This causes the perceived range measurement to be erroneously high, as photons take a much longer time to travel through multiple reflections than they do for a more direct route. The camera has no way of determining whether the ToF of any given photon corresponds to a direct, or multi-path reflection, and therefore must simply assume a direct path. Figure 3.1(a) shows an RGB image of a table with a series of 3D printed spears on top. The surface of this table is reflective to IR light. This causes photons from the ToF camera to exhibit multi-path errors as shown by the "grooves" in Figure 3.1(b).

Motion artifacts are another known weakness of ToF imaging. There are two main effects motion can have on ToF images. Intra-frame motion blur can occur if the ratio between the speed of motion, and the integration time of the pixels is sufficiently high. More pertinent to crop detection however, are the inter-frame errors, which occur as a result of mismatching phase values between the constituent phase images of each frame. These effects are especially pronounced around the edges of foreground objects, where discon-



(a) (b)

Figure 3.1: Images demonstrating the effect of multi-path errors. Image (a) shows an RGB image of the scene, and (b) shows the resulting pointcloud. The trenches in front of each "spear" in the pointcloud are the result of multi-path errors due to the reflective surface of the tabletop.

tinuous changes in range measurements are observed. The direction of motion also plays a part in the severity of this type of error. Transverse motion, for example, is much more likely to result in inter-frame motion artifacts than axial motion.

Flying pixels are also an error characteristic of ToF imaging. Similarly to the inter-frame errors associated with motion artifacts, flying pixels occur around the edges of foreground elements where a discontinuous change in range can be observed. Due to the discrete nature of the pixel grid, it is often not possible to cleanly associate an object's edge with any given pixel. In such cases, noise in the sensor can become a deciding factor in the representation of the objects edge. This can cause a similar phase correspondence mismatch, causing pixels at the edge of the object to be assigned incorrect depth information which fluctuates significantly from frame-to-frame.

A ToF image sensor needs to be able to identify the modulated light signal, originating from the camera, in order to make range measurements. Although most ToF cameras utilise a rather narrow bandwidth of light as the signal carrier, it is common that scenes can contain additional sources of light in the carrier bandwidth. These sources typically arise from reflections of incandes-

cent surfaces or broad spectrum light sources such as the sun. Modern ToF sensors mitigate this interference by constructing each pixel as a pair of photo receivers. This enables one receiver from each pair to act as a control, allowing the camera to operate on the signal differential. This approach works well for low levels of background light interference, but breaks down upon saturation of the control receiver. As the amount of background light interference increases the size of this differential decreases, significantly hampering the signal-to-noise ratio of the device near saturation. This is problematic because the precision of the analog to digital converter used to interpret these signals is limited, resulting in poor precision in the resulting range measurement. These limitations result in ToF cameras exhibiting poor performance in outdoor environments. It should be noted, however, that these issues stem from the limited dynamic range of the photo receivers rather than some inherent feature of outdoor environments themselves. Appropriate tuning and hardware specifications can, therefore, produce a camera capable of performing well in such environments.

A drawback of the "phase-correspondence" approach adopted by most modern ToF cameras is a hard limit on the maximum operating range. This limitation arises due to the fact that a pure sinusoidal signal that is ϕ radians out of phase is indistinguishable from a signal that is $2\pi + \phi$ radians out of phase, that is the phase shift is modulo 2π. Any range measurement larger than the distance represented by a phase shift of 2π radians is therefore ambiguous. Some modern ToF cameras utilise many superimposed frequencies for their modulation in order to extend the unambiguous range, however since the same limitation conceptually applies to all periodic signals cameras that utilise such advancements still exhibit limited operating range, usually on the order of 1-5 meters.

The depth-maps generated by ToF cameras are often presented as 3D point-clouds. The following subsection provides a theoretical definition of a "point-cloud" and describes how they are created from these depth-maps.

3.2.2 Pointclouds

A pointcloud, $\mathbf{P}$, is defined as a collection of elements, $\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n \in \mathbf{P}$ in a 3D Euclidean space. As such, each of these elements contains a 3D vector component, $\mathbf{p}_i^{\text{xyz}} \in \mathbb{R}^3$ which describes a position in the space, and can also contain additional components such as a colour vector $\mathbf{p}_i^{\text{RGB}} \in \mathbb{N}^3 \cap [0, 255]^3$. This work deals mostly with pointclouds consisting of elements of the form $\mathbf{p}_i = [\mathbf{p}_i^{\text{xyz}}, \mathbf{p}_i^{\text{RGB}}]$, denoted as a "XYZRGB pointcloud". Likewise, pointclouds that have elements of the form $\mathbf{p}_i = \mathbf{p}_i^{\text{xyz}}$ are denoted simply as "XYZ pointclouds".

Depthmaps captured from ToF cameras can be converted into pointclouds by utilising the intrinsic parameters of the camera. If it is assumed that the scene is translationally, and rotationally aligned with the image-space coordinates of the depth-map, a 3D pointcloud can be generated as:

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = z \begin{bmatrix} \frac{1}{f_x} & 0 & 0 & 0 \\ 0 & \frac{1}{f_y} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \\ \frac{1}{z} \end{bmatrix} \tag{3.1}$$

where $\begin{bmatrix} x & y & z \end{bmatrix}^{\text{T}}$ denotes the three principal Cartesian coordinates of a point in a Euclidean space, f_x and f_y denote the focal lengths of the camera and $\begin{bmatrix} u & v \end{bmatrix}^{\text{T}}$ are the corresponding image-space coordinates of the point in the depth-map.

XYZRGB pointclouds are generated by performing image registration between an auxiliary RGB image and the depthmap generated by the ToF sensor. The relevant colour information can then be appended to each element in the pointcloud. Many ToF cameras have an integrated RGB camera that is automatically registered to the depth-map in order to provide XYZRGB data. Typically, this registration is performed using an internal perspective transformation matrix calibrated by the camera manufacturer.

There are two common paradigms used for storing pointclouds in software, namely either *unorganised* or *organised* pointclouds. The main differences between these paradigms arise based on how the elements are stored in memory.

In an *unorganised* pointcloud, elements are stored as a 1D vector, while in an *organised* pointcloud points are stored in a I_1 by J_1 2D array such that each element in the array corresponds to a u, v coordinate from the original depth-map. *Organised* pointclouds are denoted by ${}^*\mathbf{P}$ and are defined as:

$${}^*\mathbf{P} = \begin{bmatrix} {}^*\mathbf{p}_{1,1} & \cdots & {}^*\mathbf{p}_{1,J_1} \\ \vdots & \ddots & \vdots \\ {}^*\mathbf{p}_{I_1,1} & \cdots & {}^*\mathbf{p}_{I_1,J_1} \end{bmatrix}$$

The structure of an *organised* pointcloud is advantageous because the proximity between points in the scene is preserved. This allows for efficient nearest neighbour point searches over areas of the pointcloud with smooth and continuous depth gradients. However, this rigid structure requires that the pointcloud be dense; that is comprised of exactly $I_1 \times J_1$ valid elements. This is problematic when pointclouds contain invalid points as a result of real-world imaging, or if elements need to be removed as part of the processing pipeline. A typical approach to solving this issue is to simply replace the numerical data pertaining to such elements with a NaN (not a number) or equivalent "null" structure. Such an approach, however, is not capable of fully realising the computational savings associated with element removal, such as results from pointcloud filtering or segmentation. This is because an *organised* pointcloud will always contain $I_1 \times J_1$ elements, requiring a minimum of $I_1 \times J_1$ iterations to process. The only exception to this result is in the special case where the filtered pointcloud can be represented by an axis-aligned subset of the original depthmap. In contrast, an *unorganised* pointcloud does not face these issues as invalid, or unwanted elements can simply be dropped from the structure. This allows certain algorithms to be much more performative using *unorganised* pointclouds. The obvious drawback with *unorganised* pointclouds is that they do not retain point correspondence with the depth map, making certain operations less efficient.

The proposed perception pipeline is intended to operate on input pointclouds, generated by a ToF camera to detect, and localise asparagus spears.

The following section presents the fundamentals of the proposed perception pipeline.

3.3 Fundamentals of the Proposed Perception Pipeline

The goal of a successful asparagus perception pipeline is to take a 3D point-cloud representation of a scene, and to discriminate between those points which correspond to asparagus spears, and those which belong to background elements, or foreign debris such as weed growth. Furthermore, points pertaining to individual spears should be separable so that each spear can be analysed separately.

Real-time performance is a fundamental constraint on the proposed pipeline. For this reason it is essential that unnecessary data points be eliminated from the pointcloud to limit unnecessary computation. One approach, which conforms to this core philosophy, is to model the perception pipeline as a feature filter. Such an approach aims to detect asparagus spears by systematically removing non-conforming points from the scene until only asparagus points remain. Similarly, individual spears can then be separated from the set of asparagus-related points by iteratively filtering points from each respective spear into individual groups or clusters. A high-level overview of the proposed perception pipeline is described by Figure 3.2.

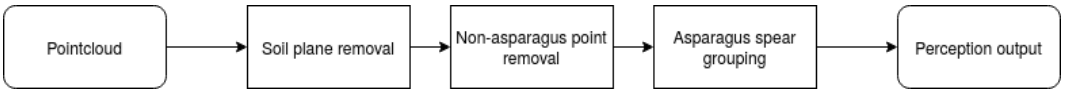


Figure 3.2: High-level overview of the perception pipeline.

A common non-asparagus feature present in all scenes is the ground or soil. As mentioned above, asparagus is generally grown in either flat or mounded beds. For this thesis, the proposed perception pipeline is limited in scope to flat asparagus beds. The reason for this is twofold. Firstly, a flat, or

planar asparagus bed is geometrically simpler and less variable than a mounded bed which simplifies the image processing task. Secondly, the most local, and therefore accessible farms to Hamilton/Waikato, where this research was conducted, tend to use flat growing beds. Accepting this assumption the task of filtering out the ground or soil from a scene becomes one of plane detection. This is a logical first step in a prospective perception pipeline because a large majority of points in each scene belong to the soil plane. Filtering these points early will greatly reduce the number of points to be processed in subsequent steps.

Following the removal of the soil plane a typical scene will consist of only points pertaining to asparagus spears, and foreign debris such as rocks, dirt clumps and weeds. The next step is to filter out non-asparagus features. The points which remain from this process all relate to various asparagus spears in the scene. It is important that each individual spear in the scene is filtered out separately from these remaining points in order to determine the cut location, and morphology of each individual spear.

Each of the core elements of the proposed pipeline, described in Figure 3.2, are presented in detail in the following subsections, beginning with soil plane removal.

3.3.1 Soil Plane Removal

The soil plane is expected to be a dominant feature present in all asparagus beds. Due to the unstructured nature of real-world asparagus beds, the soil surface may not be strictly planar. Additionally, noise from the ToF camera, especially in the direction of the cameras optical/depth axis, will contribute significant noise to the captured scene. Consequentially, methods utilised for the soil plane detection need to be robust to noise, and possible perturbations of the soil plane. Additionally, since the captured pointcloud is a 2.5D representation of the scene, any weed growth, spears, or foreign debris that are present in the scene will occlude parts of the soil plane that fall in the

shadows of the camera's light source. Consequentially, points not belonging to the plane are not necessarily balanced by points pertaining to the soil plane. For this reason, a plane-of-best fit approach to modelling the soil plane is not sufficient. Typically, an asparagus bed, like most organic constructions, lack straight edges and planar surfaces. The planar surface of the asparagus bed has been constructed to be this way. It is therefore assumed that the soil plane is the largest, or most dominant, planar feature in any given scene. As such, the task of modelling the soil plane of the asparagus bed is reduced to the task of modelling the dominant plane in a scene. Once a valid soil plane model has been determined, points in the scene pertaining to the soil can be filtered out from the input pointcloud by applying a distance threshold.

Two methods for determining the dominant plane in a pointcloud were explored for this task, namely Hyun's Method, and Random Sample Consensus (RANSAC). The following subsections describe these methods in detail.

3.3.2 Hyun's Method

Hyun's method is a method for detecting planar features in pointcloud data that was developed by Yoo et al. [100]. Specifically their work is concerned with plane detection in images captured using a Microsoft Kinect V2 Camera (Kinect Camera), and has an emphasis on achieving real-time performance.

The input to the method is an organised, XYZ pointcloud, ${}^{*}\mathbf{P}$. Hyun's method begins by uniformly sampling a set of n_1 points from the input pointcloud. The points are sampled by selecting every α^{th} point in row order. That is the points are sampled such that the set of sampled points, $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_{n_1}]$ is comprised of elements, $\mathbf{s}_\xi = {}^{*}\mathbf{p}_{i_1,j_1}$, with $i_1 = \left\lceil \frac{(\xi-1)\alpha}{J_1} \right\rceil$ and $j_1 = [(\xi-1)\alpha + 1] - J_1(i_1 - 1)$, where α is a constant denoting the size of the sub-sampling. At each of the sampled points the vectors $\mathbf{k1}_\xi$ and $\mathbf{k2}_\xi$ are calculated as:

$$\mathbf{k1}_\xi = {}^{*}\mathbf{p}_{i_1-1,j_1} - {}^{*}\mathbf{p}_{i_1+1,j_1} \tag{3.2}$$

$$\mathbf{k2}_\xi = {}^{*}\mathbf{p}_{i_1,j_1+1} - {}^{*}\mathbf{p}_{i_1,j_1-1} \tag{3.3}$$

A local normal vector can then be calculated at each sampled point in $\mathbf{S}$ as:

$$\mathbf{n}_\xi = \mathbf{k1}_\xi \times \mathbf{k2}_\xi \tag{3.4}$$

Points in $\mathbf{S}$ are then grouped based on the correspondence between their associated normal vectors by considering all pairs of points, $\mathbf{s}^\dagger$ and $\mathbf{s}^\ddagger$ and checking the conditions:

$$\mathbf{n}^\dagger \cdot \mathbf{n}^\ddagger = 1 \tag{3.5}$$

$$(\mathbf{s}^\dagger - \mathbf{s}^\ddagger) \cdot \mathbf{n}^\dagger = 0 \tag{3.6}$$

Equation 3.5 is used to determine if two normal vectors, at $\mathbf{s}^\dagger$ and $\mathbf{s}^\ddagger$ are parallel and equation 3.6 is used to determine if the vector between two sampled points, $\mathbf{s}^\dagger$ and $\mathbf{s}^\ddagger$ is perpendicular to the normal vector $\mathbf{n}^\dagger$. Equation 3.6 is important because it allows the method to reject parallel, but not coincident planes from being grouped. Grouping is achieved iteratively as follows:

Let ϵ be the index of iteration. To begin, $\mathbf{n}_\epsilon$, where $\epsilon = 0$, is checked against all other normal vectors using equations 3.5 and 3.6. Any normal vectors which satisfy these equations have their corresponding sampled point, $\mathbf{s}_\epsilon$ saved in a vector, $\mathbf{l}_\epsilon$. This process is repeated for $\epsilon \in \mathbb{N} \cap [1, n_2]$ and each $\mathbf{l}_\epsilon$ is stored in an array $\mathbf{L}$. In practice strictly applying equations 3.5 and 3.6 yield poor results due to sensor noise. For this reason a parallelity threshold, ψ_1 and orthogonality threshold, ψ_2 were enforced such that equation 3.5 and 3.6 become:

$$\mathbf{n}^\dagger \cdot \mathbf{n}^\ddagger = \psi_1 \approx 1 \tag{3.7}$$

$$(\mathbf{s}^\dagger - \mathbf{s}^\ddagger) \cdot \mathbf{n}^\dagger = \psi_2 \approx 0 \tag{3.8}$$

The resulting array, $\mathbf{L}$, contains n_3 vectors, each consisting of points from $\mathbf{S}$ which lay on the same plane (within some tolerance). For each vector in $\mathbf{L}$ a plane equation of the form:

$$ax + by + cz + d = 0 \tag{3.9}$$

is fit by minimising the square error of points in each vector to their respective plane model using Singular Value Decomposition (SVD). The result of this

method is an array, $\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_{n3}]$, of vectors containing plane model coefficients $[a, b, c, d]$ for each vector.

Further refinement is then carried out to eliminate duplicate planes. This process is known as the "refinement stage". Each plane model in $\mathbf{M}$ is checked against all other plane models by applying the following criteria to every pair of plane models, $\mathbf{m}^\dagger$ and $\mathbf{m}^\ddagger$.

$$\mathbf{m}^\dagger \cdot \mathbf{m}^\ddagger = [a^\dagger, b^\dagger, c^\dagger] \cdot [a^\ddagger, b^\ddagger, c^\ddagger] = 1 \tag{3.10}$$

$$d^\dagger - d^\ddagger = 0 \tag{3.11}$$

Equation 3.10 is used to check that the two planes are parallel, and equation 3.11 is used to check that the distance between two planes is zero. If two planes are found which satisfy these conditions then the plane which contains the smallest number of points is removed.

Similarly to the previous step, real-world errors mean that thresholds need to be used to generate acceptable results. For this reason equations 3.10 and 3.11 become:

$$\mathbf{m}^\dagger \cdot \mathbf{m}^\ddagger = [a^\dagger, b^\dagger, c^\dagger] \cdot [a^\ddagger, b^\ddagger, c^\ddagger] = \psi_3 \approx 1 \tag{3.12}$$

$$d^\dagger - d^\ddagger = \psi_4 \approx 0 \tag{3.13}$$

during implementation, where ψ_3 and ψ_4 are the threshold of planar parallelity and distance respectively.

The result of Hyun's method is a list of plane model coefficients for every predicted plane in the scene. Hyun's method only discriminates between different plane features from a global perspective, using only the orientation, and offsets. This means that the method is unable to discriminate between two planar features that happen to have co-planar surfaces. In this case Hyun's method will report a single plane model which encompasses both plane segments.

The list of detected planes can then be arranged by the number of elements in the $\mathbf{l}$ vector of each plane. From this the plane with the largest number of elements can be selected. This is the dominant plane in the image.

3.3.3 Evaluation of Hyun's Method

Hyun's method was developed for use in detecting planar surfaces within the internal structure of boat hulls. Such environments allow the Kinect camera to image surfaces at relatively close range, without ambient lighting. These conditions are ideal for ToF imaging. In contrast, an asparagus harvester must operate outdoors, where interference is a concern. This means that the pointcloud quality will be lower for the proposed application.

To test the suitability of Hyun's method for plane detection in asparagus harvesting a mockup asparagus bed was constructed. The mockup asparagus bed, pictured in Figure 3.3, was constructed from MDF and filled with sand to simulate the sandy soil used in typical asparagus fields.

Figure 3.3: Mockup asparagus bed. Frame is constructed from MDF and Kinect mount is constructed from 3D printed PLA.

Pointcloud images captured from this setup proved to be too noisy for the standard Hyun's method under lab conditions. The range precision of the Kinect camera is approximately 10mm. Because of this, local points can differ significantly from frame-to-frame, resulting in wildly fluctuating **k1** and

k2 vectors. Fluctuations in these vectors result in inconsistent normal vector calculations, and a large number of erroneous planes being detected due to poor correspondence between normal vectors on the same planar feature. This problem can be alleviated by loosening the parallelity, orthogonality, and distance thresholds, ψ_1 ψ_2, ψ_3, and ψ_4. However, it was found that the degree to which these thresholds needed to be loosened to achieve coherent planes was such that the detected planes were poor descriptors of the scene.

Modifications were made to Hyun's method to solve this problem. These modifications are described in the following subsection.

3.3.4 Modifications to Hyun's Method

The application of soil plane detection for asparagus harvesting requires that only the dominant plane from the input pointcloud must be detected. Due to the structure of an asparagus bed, and the chosen mounting arrangements of the camera it is expected that every scene will consist of a singular planar feature which spans the majority of the image. The proposed modifications to Hyun's method take advantage of these factors by trading generality for consistency. The following modifications were made to Hyun's method:

- The refinement stage was simplified
- A linear regression, rather than straight line was used to define vectors **k1** and **k2**

The refinement stage utilised in standard Hyun's method is used to reject duplicate planes, and to group corresponding normal vectors into various planes within the scene. The application presented in this thesis does not require every planar feature in the image to be detected; only the largest plane needs to be detected. Therefore, once $\mathbf{L}$ is constructed, the vector $\mathbf{l}_{\max} \in \mathbf{L}$ with the largest number of elements is selected. Least squares regression is then used to fit a plane model to the elements of $\mathbf{l}_{\max}$ as stated above. The refinement steps described by equations 3.12 and 3.13 are not required because

only a single plane is being considered. There is no need to disambiguate similar planes.

When presented with a noisy pointcloud, the normal vectors, $\mathbf{n}$, are prone to error. This is because the construction of vectors **k1** and **k2** considers only the nearest neighbouring points from ${}^{*}\mathbf{P}$ to each sampled point $\mathbf{s}_{\xi}$. When the pointcloud is noisy the depth axis can fluctuate approximately $\pm$10mm. When imaging over the ranges expected for this application (approximately 1m) the spatial resolution of points in the lateral and vertical axes (i_1 and j_1) of the pointcloud are much smaller than this. Vectors **k1** and **k2** are therefore dominated by fluctuations in the depth axis of the pointcloud, resulting in wild fluctuations in the local normal vectors at each point in $\mathbf{S}$. A solution to this is to construct the **k1** and **k2** vectors based on a line-of-best-fit to several points which span the i_1 and j_1 directions of the pointcloud respectively. This approach will result in much more stable **k1** and **k2** vectors, however the achievable feature resolution will decrease considerably as more points are included in the line fitting. Likewise, the stability of the resulting **k1** and **k2** vectors will increase as more points are included in the line fitting. This application aims to detect the dominant plane in a pointcloud, and expects this plane to be expansive. Thus, high feature resolution is not required, enabling a large number of points to be included in the construction of **k1** and **k2**.

Construction of the **k1** and **k2** vectors begins with uniformly sampling the pointcloud ${}^{*}\mathbf{P}$ across the i_1 and j_1 axes. In the modified version, the set of sampled points, $\mathbf{S}$, are maintained in a 2D matrix such that each point can be described as $\mathbf{s}_{i_2,j_2} = {}^{*}\mathbf{p}_{i_1,j_1}$, where $i_1 = (i_2 - 1)\alpha + 1$, and $j_1 = (j_2 - 1)\beta + 1$ for values of α and β which satisfy $J_1 \mod \alpha = 0$ and $I_1 \mod \beta = 0$, where α and β are constants that denote size of the sampling grid in the i_1 and j_1 axes respectively. This differs from the original method, where the sample points were stored in a 1D vector. For each of the sampled points, $\mathbf{s}_{i_2,j_2} \in \mathbf{S}$ the set of points in ${}^{*}\mathbf{P}$ between ${}^{*}\mathbf{p}_{i_1,j_1}$ and ${}^{*}\mathbf{p}_{i_1+\beta,j_1}$ and the set of points between ${}^{*}\mathbf{p}_{i_1,j_1}$ and ${}^{*}\mathbf{p}_{i_1,j_1+\alpha}$ in the i_1 and j_1 directions respectively are found; $\mathbf{w1}_{i_2,j_2}$ and

$\mathbf{w2}_{i_2,j_2}$ respectively. Since the i_1 and j_1 axes are orthogonal and the point-sets $\mathbf{w1}_{i_2,j_2}$ and $\mathbf{w2}_{i_2,j_2}$ are exclusively sampled along these axes, the problem of fitting a line to each point-set in 3D simplifies to 2D line-fitting.

Since the vectors **k1** and **k2** are only required to compute the local normal vector, which is strictly a direction vector, only the slope of this line needs to be solved. That is, the magnitude of vectors **k1** and **k2** are irrelevant.

A 2D line can be represented as:

$$Y = m_L X + C \tag{3.14}$$

where, m_L denotes the gradient of the line in the (X, Y) plane and C is a constant. For a given pair of coordinates $[x_1, y_1]$ and $[x_2, y_2]$ there exists an m_L and C which describes a line in 2D that intersects both coordinates. This concept can be expanded to q elements and be expressed in matrix form as:

$$\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & ... \\ 1 & x_q \end{bmatrix} \begin{bmatrix} C \\ m_L \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ .. \\ y_q \end{bmatrix} \tag{3.15}$$

which can be expressed as:

$$\mathbf{A}\chi = \mathbf{g} \tag{3.16}$$

In cases where all q coordinates lay perfectly along a single line in 2D this overdetermined system can be solved for vector χ and the line equation can be resolved. However, when noise is considered such that the coordinates do not lay perfectly on a line a solution for χ does not exist. In this case χ can be approximated by finding a vector $\hat{\chi}$ which satisfies the condition:

$$\|\mathbf{A}\hat{\chi} - \mathbf{g}\|^2 \leq \|\mathbf{A}\chi - \mathbf{g}\|^2 \text{ for all } \chi \in \mathbb{R}^2 \tag{3.17}$$

Such a $\hat{\chi}$ is the least squares approximate solution. $\hat{\chi}$ can be solved by substituting for χ in equation 3.16:

$$\mathbf{A}\hat{\chi} = \mathbf{g} \tag{3.18}$$

Multiplying by $\mathbf{A}^{\mathrm{T}}$ gives:

$$\mathbf{A}^{\mathrm{T}}\mathbf{A}\hat{\chi} = \mathbf{A}^{\mathrm{T}}\mathbf{g} \tag{3.19}$$

From which it can be determined that:

$$\hat{\chi} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{g} \tag{3.20}$$

Since the columns of $\mathbf{A}$ are comprised of the coefficients of separate, independent variables from equation 3.15, the matrix $\mathbf{A}$ is full rank. In this case the peudo-inverse, $\mathbf{A}^{\dagger}$ of matrix $\mathbf{A}$ can be found as:

$$\mathbf{A}^{\dagger} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{T}} \tag{3.21}$$

Thus:

$$\hat{\chi} = \mathbf{A}^{\dagger}\mathbf{g} \tag{3.22}$$

Decomposing $\mathbf{A}$ using QR decomposition provides that:

$$\mathbf{A} = \mathbf{Q}\mathbf{R} \tag{3.23}$$

where $\mathbf{Q}$ is a rectangular matrix, the columns of which describe an orthonormal basis of $\mathbf{A}$, and $\mathbf{R}$ is an upper triangular matrix. Typically $\mathbf{Q}$ is calculated using the Gram-Schmidt process, and $\mathbf{R}$ is calculated as:

$$\mathbf{R} = \mathbf{A}\mathbf{Q}^{\mathrm{T}} \tag{3.24}$$

due to the orthonormal nature of $\mathbf{Q}$.

Substituting this decomposition into equation 3.20 results in:

$$\hat{\chi} = \left(\left(\mathbf{Q}\mathbf{R}\right)^{\mathrm{T}}\left(\mathbf{Q}\mathbf{R}\right)\right)^{-1}\left(\mathbf{Q}\mathbf{R}\right)^{\mathrm{T}}\mathbf{g} \tag{3.25}$$

$$= \left(\mathbf{R}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}\mathbf{Q}\mathbf{R}\right)^{-1}\mathbf{R}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}\mathbf{g} \tag{3.26}$$

Since $\mathbf{Q}$ is by definition orthonormal:

$$\mathbf{Q}^{\mathrm{T}} = \mathbf{Q}^{-1} \tag{3.27}$$

$$\mathbf{Q}^{\mathrm{T}}\mathbf{Q} = \mathbf{Q}^{-1}\mathbf{Q} = \mathbf{I} \tag{3.28}$$

Therefore,

$$\hat{\chi} = \left(\mathbf{R}^{\mathrm{T}}\mathbf{R}\right)^{-1}\mathbf{R}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}\mathbf{g} \tag{3.29}$$

$$= \mathbf{R}^{-1}\left(\mathbf{R}^{\mathrm{T}}\right)^{-1}\mathbf{R}^{\mathrm{T}}\mathbf{Q}^{\mathrm{T}}\mathbf{g} \tag{3.30}$$

$$= \mathbf{R}^{-1}\mathbf{Q}^{\mathrm{T}}\mathbf{g} \tag{3.31}$$

All elements of equation 3.31 are then known, and $\hat{\chi}$ can be solved.

The modified Hyun's method (MHM) generates much more stable model predicution than standard Hyun's method when applied to noisy pointclouds such as those taken from a ToF camera in an unstructured environment. This is because MHM considers a larger local region of each sample point from the input pointcloud, effectivly smoothing out local range fluctuations. For the application presented in this thesis MHM is more suitable than standard Hyun's method.

Another algorithm for detecting the dominant plane in a pointcloud is RANSAC. A description of RANSAC is presented in the following subsection.

3.3.5 Random Sample Consensus (RANSAC)

RANSAC is a method that is used to find dominant features in n-dimensional pointcloud data. When presented with a dataset that contains both inliers (points which represent the desired feature) and outliers (points which represent noise and background elements) RANSAC aims to find a feature description which contains the largest number of inliers. This means that RANSAC aims to find only the largest feature in the dataset. If the dataset contains many similar features each feature will need to be sequentially determined by iteratively applying RANSAC and removing inliers at each iteration.

RANSAC requires that features be defined explicitly as a mathematical model. For the application of determining the dominant plane in a 3D pointcloud a plane model is utilised, however, generally this method can extend into n-dimensions and utilise any mathematically definable feature.

A 3D plane can be expressed mathematically described by equation 3.9, repeated here for convenience:

$$ax + by + cz + d = 0$$

RANSAC begins by randomly sampling a subset of points $\mathbf{S}$ from an input *unorganised* pointcloud $\mathbf{P}$. The number of points sampled is the number of points required to fully define the feature model. In the case of a plane three points are required.

Model parameters a, b, c and d are then determined by fitting a plane to the points in $\mathbf{S}$. An exact solution to this plane fitting task will always exist because the number of points in $\mathbf{S}$ is equivalent to the number of parameters of the model. This results in a model prediction:

$$\mathbf{m} = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \tag{3.32}$$

The model prediction is then tested by calculating the error, e, between every point in $\mathbf{P} \backslash \mathbf{S}$. The specific error metric can vary per application, however for plane detection it is common to use the orthogonal distance between the plane described by $\mathbf{m}$, and a point, $\mathbf{p}_i^{xyz} = [x_i, y_i, z_i] \in \mathbf{P} \setminus \mathbf{S}$:

$$\rho = \frac{|ax_i + by_i + cz_i + d|}{\sqrt{a^2 + b^2 + c^2}} \tag{3.33}$$

All points with error values lower than some threshold ψ_5 are deemed inliers and stored in a vector, $\mathbf{I_0}$. If the number of inliers in $\mathbf{I_0}$ exceed some threshold, ψ_6, then the predicted model $\mathbf{m}$ is deemed to be a good fit. When this occurs a refined version of the model prediction, $\hat{\mathbf{m}}$ is generated by fitting a new model to all points in $\mathbf{S} \cup \mathbf{I}_0$. This model fitting is generally achieved using least squares regression. $\hat{\mathbf{m}}$ is then evaluated by calculating the mean squared error, E_r, between the plane described by $\hat{\mathbf{m}}$ and the points in $\mathbf{S} \cup \mathbf{I}_0$.

This procedure is iterated n_4 times in order to find an acceptable solution. Whenever a given iteration produces a well fitting model, and thus produces

a refined model, $\hat{\mathbf{m}}$, the corresponding mean squared error value, E_r, of that model is compared against models generated in previous iterations. If the current iterations E_r value is smaller than previous iterations then $\hat{\mathbf{m}}$ is updated as the new "best fit model". After n_4 iterations the model which produced the lowest E_r is returned.

RANSAC has a few well known limitations. Firstly, RANSAC is non-deterministic. This means that there is no guarantee that a reasonable solution will be found in a finite amount of time/iterations. This is because, at the core, the method relies on randomly sampling the dataset and eventually selecting a subset which contains only representative inliers. Representative, in this context requires that a point needs to not only be an inlier, but also be positioned close to its Gaussian mean position when considering the influence of sensor noise. When this is the case it is likely that the model prediction for a given iteration, $\mathbf{m}$, will generate a model that fits the data well. Furthermore, it is preferable that such a situation occur many times during the n_4 iterations so that a variety of refined models can be generated, allowing a more robust selection of the best fitting model.

The computational complexity of RANSAC varies depending on the size and dimensionality of the data, and the complexity of both the feature model, error function, and regression methods used. In order to achieve real-time performance the number of points which are operated on need to be minimised.

RANSAC is best suited when the dataset represents a scene with a small number of features which are relatively large. In this scenario the ratio of inliers to outliers is relatively large, corresponding to a high probability of randomly selecting a sample of representative inliers. This increases the likelihood that a reasonable solution will be found within n_4 iterations.

It is required the the method utilised for base plane segmentation be robust in the presence of weeds, and other field clutter in order to ensure the proposed perception pipeline is able to operate robustly in a commercial setting. The

following section presents an evaluation of both RANSAC and MHM with respect to field clutter, and camera angle.

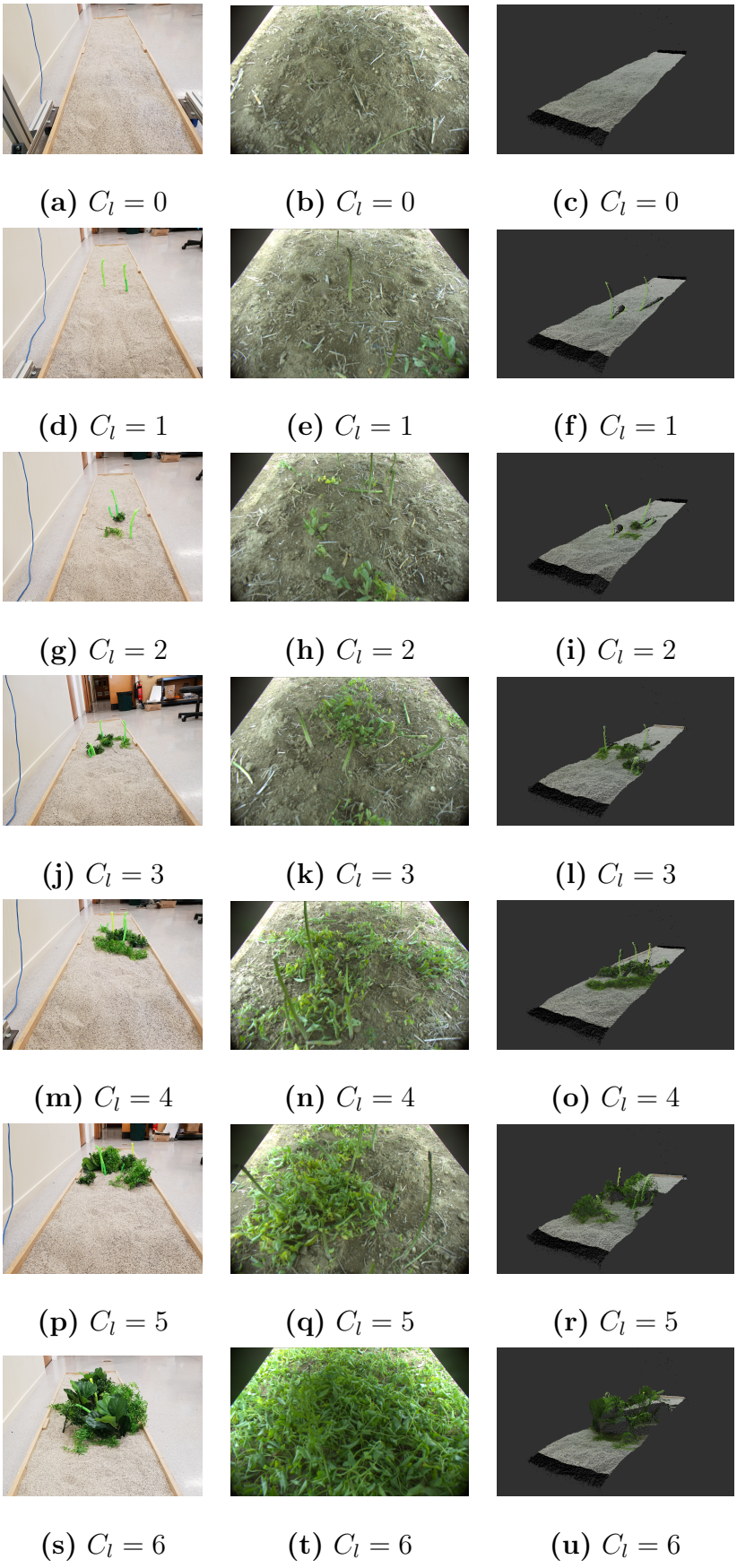
3.4 Effect of Field-Clutter and Camera Angles

An experiment was performed to compare the applicability of both RANSAC and MHM for soil plane detection. The experiment aimed to investigate the performance of each method against two variables; the degree of field clutter, and the angle between the soil plane and optical axis of the ToF camera.

A sandbox measuring 2.8m $\times$0.6m was constructed from MDF, simulating an asparagus bed. A Kinect camera was mounted to a frame at a height of 900mm above the surface of the sand pit. The mount, was configurable to allow various camera angles to be achieved. For this experiment the mounting angle of the camera, θ, was defined as the angle between the base of the sandbox and the optical axis of the camera.

Several scenes with varying degrees of field clutter were constructed using 3D-printed asparagus spears (PLA) and various plastic shrubbery. The degree of field clutter in each of these scenes was based on observations of weed growth observed on real-world asparagus beds. In total 7 scenes were constructed, each with an increasing degree of field clutter; These scenes were assigned a clutter level, $C_l \in \mathbb{N} \cap [0, 6]$. Figure 3.4 (a-u) shows each scene, as well as the corresponding pointcloud and image demonstrating the real-world basis.

For each scene a series of one minute recordings of the Kinect camera's pointcloud stream were generated. These recordings were made for θ values of 0°, 25°, 45°, 55°, and 65°, resulting in a dataset of 35 recordings in total. The coordinate system for the recorded pointcloud stream was orientated with the x-axis aligned down the length of the sandbox, y-axis aligned across the width, and the z-axis directed normal to the perceived soil plane. Additionally, the recorded pointclouds were cropped along the x, and y axes in order to simplify the scenes. A similar cropping process was expected to be applicable to real-



(a) $C_l = 0$ **(b)** $C_l = 0$ **(c)** $C_l = 0$

(d) $C_l = 1$ **(e)** $C_l = 1$ **(f)** $C_l = 1$

(g) $C_l = 2$ **(h)** $C_l = 2$ **(i)** $C_l = 2$

(j) $C_l = 3$ **(k)** $C_l = 3$ **(l)** $C_l = 3$

(m) $C_l = 4$ **(n)** $C_l = 4$ **(o)** $C_l = 4$

(p) $C_l = 5$ **(q)** $C_l = 5$ **(r)** $C_l = 5$

(s) $C_l = 6$ **(t)** $C_l = 6$ **(u)** $C_l = 6$

Figure 3.4: Images of scenes of various clutter levels. The first column shows images of the recreated lab apparatus, the second column shows images of the real-world scene from which the recreated scenes were based, and the last column shows a pointcloud capture of the scene from the ToF camera.

world asparagus rows, eliminating all points in the input pointcloud outside of the asparagus bed.

Both RANSAC and MHM aim to fit a plane of the form described in equation 3.9, reiterated here for convenience:

$$ax + by + cz + d = 0$$

to an input XYZ-pointcloud, ${}^*\mathbf{P}$. It was not possible to prescribe a ground truth model of the form in Equation 3.9 to the surface of the sandbox as it was not perfectly planar. For this reason both RANSAC and MHM were evaluated independently against their respective performance on the zero clutter ($C_l = 0$) scenes for each value of θ. This resulted in a set of five ground truth models for each method. Each of these ground truth models were generated by applying the respective method to a uniform sampling of 1000 frames from the relevant recording and averaging the resultant a, b c, and d parameters.

The performance of each method was then evaluated by calculating the root mean squared error (RMSE) between each of the ground truth models, $\mathbf{m}_g$ and a corresponding model prediction based on a cluttered scene ($C_l = 0-6$). The RMSE of each combination of method, θ and C_l was calculated as set out below.

To begin, an *organised* pointcloud of the scene, ${}^*\mathbf{P}^\eta$ was captured with the ToF camera and a uniform grid of n_5 2D points, $\mathbf{S}^\eta = [\mathbf{s}_1^\eta, \mathbf{s}_2^\eta, ..., \mathbf{s}_{n_5}^\eta]$ which span the x, y domain of ${}^*\mathbf{P}^\eta$ were determined. The method being tested was then applied to ${}^*\mathbf{P}^\eta$ in order to determine a predicted model, $\mathbf{m}_p$. For each point in $\mathbf{S}^\eta$, z coordinates, z_g and z_p were determined based on $\mathbf{m}_g$ and $\mathbf{m}_p$ respectively. The RMSE was then calculated as:

$$\text{RMSE} = \sqrt{\frac{1}{n_5}\sum_{i=1}^{n_5}(z_g - z_p)^2} \tag{3.34}$$

This procedure was conducted 100 times for each combination of method, camera angle and clutter level, on a uniform sampling of pointclouds from the relevant recordings using $n_5 = 500$. From each set of 100 RMSE values an

average RMSE, and standard deviation, σ_{RMSE} was determined. Figure 3.5 shows the graphed results.

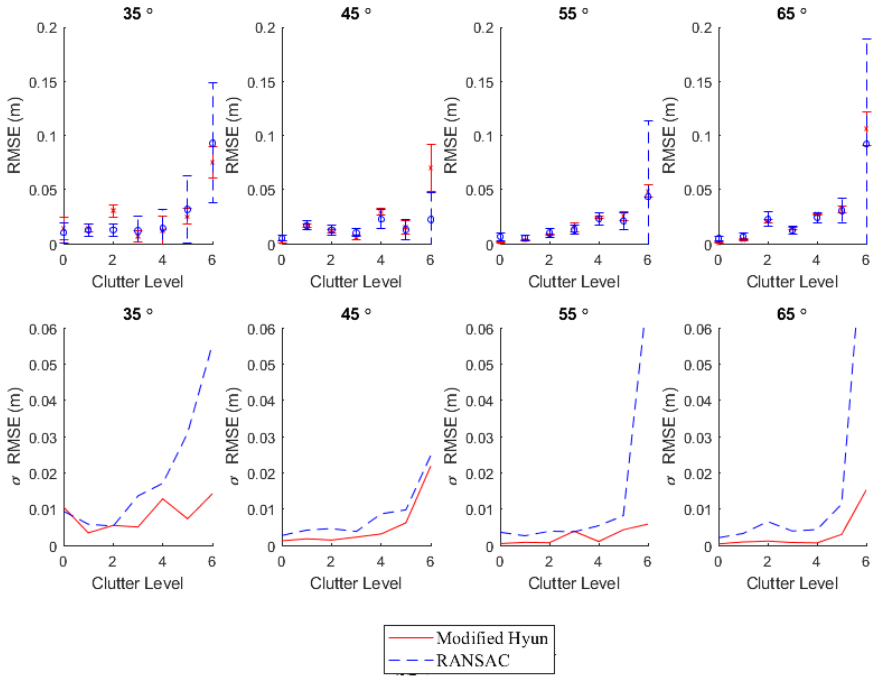


Figure 3.5: Average RMSE and σ_{RMSE} of predicted plane models for various camera angles and clutter levels

In general, the RMSE of planes predicted using both RANSAC and MHM increased with C_l. This trend, though ubiquitous across all values of θ, was most severe at the extremes, specifically at 25° and 65°. This was particularly true for $\theta = 25°$. The RMSE of plane models fit to data collected using a camera angle of 25° was an order of magnitude higher than all other angles tested. For this reason, results for $\theta = 25°$ were excluded as outliers. For completeness, plots of the RMSE and σ_{RMSE} for the 25° case have been included in Appendix A.4. It was concluded that the trend in RMSE could be explained by noise in the pointcloud for low values of θ, and lack of soil plane visibility for high values of θ. For shallow camera angles (low values of θ) a large amount of the scene is imaged. This results in a pointcloud consisting of a significant number of relatively long range points. These longer range points are subject to lower precision due to limitations of the Kinect camera.

Such images are therefore characterised by high soil plane visibility and higher noise. Conversely, images taken with steep camera angles (high values of θ) consist of points derived from a much smaller portion of the scene, and contain points much closer to the camera. Since these images have a limited field of view it is possible that high degrees of clutter can effectively occlude the soil plane. Such images are therefore characterised by low noise, and poor soil plane visibility. It can be seen in Figure 3.5 that for a zero clutter ($C_l = 0$) scene, both methods produce an RMSE of approximately 0.01m for the 35° case, and approximately 0m for the 65°. This difference in RMSE is due to the larger degree of noise present in the shallow angle images. Alternatively, when considering the performance these methods on high clutter scenes ($C_l = 6$) it can be seen that higher camera angles ($\theta = 65°$) tend to produce higher RMSE than more moderate camera angles ($\theta = 45°$-$55°$). Specifically, for $\theta = 65°$ MHM produced a RMSE that was approximately $32\% - 55\%$ higher than moderate camera angles. RANSAC demonstrated an increase of approximately $50\% - 70\%$, however the significance of this observation is indeterminate due to the extremely large corresponding σ_{RMSE} value.

σ_{RMSE} also exhibited an increasing trend with C_l, however the variance of predictions made by RANSAC were consistently higher than MHM and increased rapidly for higher values of C_l. Similarly to above, σ_{RMSE} was generally higher for shallow camera angles due to increased noise in the pointcloud. However, the σ_{RMSE} did not show significant change at higher values of θ; rather, σ_{RMSE} primarily increased with clutter level.

It was concluded that the limited precision of RANSAC predictions was due to the non-deterministic nature of the algorithm. As the degree of clutter increases, the number of points pertaining to the soil plane in the scene decreases due to occlusion. In such cases it becomes increasingly likely that any arbitrary "random sample" proposed by the algorithm results in a higher number of inliers the ground truth model. This results in inaccurate plane models being selected as the dominant plane. The inherent randomness in the

sampling process utilised by RANSAC means that a large variety of these inaccurate models are proposed over the 100 trials, manifesting as a high σ_{RMSE} value.

The execution times of both methods were also investigated and are presented in Table 3.1. The sampling schemes utilised by each respective method preclude direct comparison of execution times, however trends within each method can be observed. The execution times of each method was not affected by the clutter level, and was found to purely be a function of the camera angle. In general, both methods executed faster and exhibited less variance in execution time for shallow camera angles. However, both methods performed within the real-time constraints, with RANSAC and MHM achieving a best case frame rate of 33.8 and 178.6 respectively and a worst case frame rate of 18 and 79 respectively. It was concluded that the trend in execution times exhibited by both methods was due to the number of points present in the pointcloud. The Kinect camera has a fixed viewing angle. This means that images taken at shallow camera angles, where the maximum range of soil plane points is high, contain a large number of points from irrelevant parts of the scene. These points are then removed when the pointclouds are cropped. This results in a larger number of points being cropped for low angle images than for high angle images.

From this experiment it was concluded that both RANSAC and MHM are effective methods for soil plane detection in the experimental scenes. This forms a basis to believe such methods to be applicable in real-world scenes. Based on the results of this experiment, it is reasonable to expect these methods to perform best at moderate camera angles between 45° and 55°. The aforementioned mechanisms, which necessitate this, offer some insight about the effect that the intrinsic properties of the input pointcloud have on the performance of these methods, particularly with regard to the camera's field-of-view, and range precision.

Table 3.1: Execution times of RANSAC and MHM for a single plane prediction. σ_t is calculated based on the variance of execution time across all 100 trials for each respective camera angle.

RANSAC		
Angle	**Average Execution Time (s)**	σ_t**(s)**
25°	0.0296	0.0011
35°	0.0385	0.0006
45°	0.0508	0.0010
55°	0.0578	0.0016
65°	0.0557	0.0021

MHM		
Angle	**Average Execution Time (s)**	σ_t**(s)**
25°	0.0056	0.0002
35°	0.0075	0.0002
45°	0.0099	0.0003
55°	0.0113	0.0003
65°	0.0127	0.0005

Following soil plane removal, the proposed perception pipeline begins to filter out points from the remaining pointcloud that do not pertain to asparagus spears. This process is called non-asparagus point removal and is presented in the following section.

3.5 Non-Asparagus Point Removal

After the removal of soil plane points a pointcloud of a typical scene consists of a number of disconnected clusters of points pertaining to spears, weeds and other field debris. Depending on the roughness of the terrain, it is also possible that sections of the soil plane survive segmentation and are present in such pointclouds. Non-asparagus point removal is the process of identifying, and removing those remaining points which do not pertain to asparagus spears.

The proposed perception pipeline utilises a convolutional neural network operating on an RGB image of the scene to detect regions of the scene where asparagus features are present. These regions are used to filter out sections of the pointcloud where asparagus features are not found. The following subsections describe the process of neural network filtering.

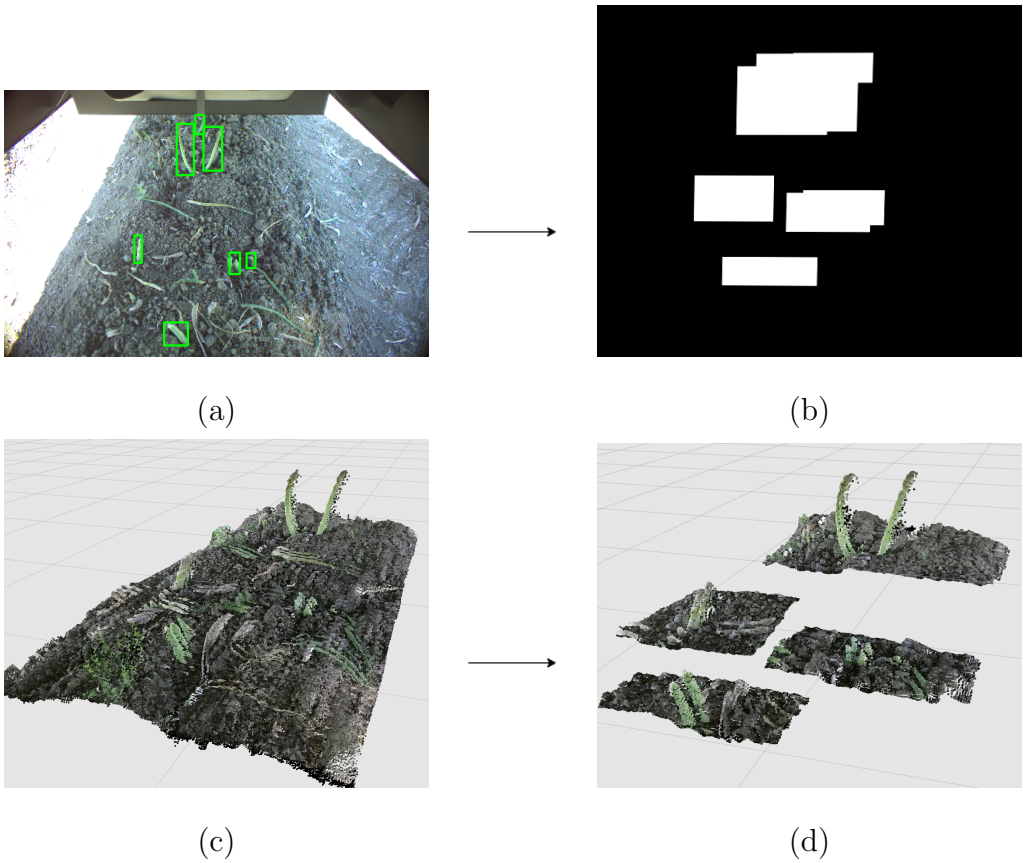
3.5.1 Neural Network Filtering

Convolutional neural networks (CNN) are a class of deep neural network that utilise both the data and structure of their inputs to generate inference. Such networks are often used for high level image processing tasks such as classification and object detection. The proposed perception pipeline utilises Faster RCNN (FRCNN) [38]; a CNN architecture which is designed for real-time object detection. FRCNN takes a single RGB image as an input and outputs a fixed number of "bounding boxes", each with an associated class prediction and confidence value. The number of bounding boxes produced per image is constant, and configurable before training of the model. The class prediction associated with each bounding box is used to determine the object type as-

sociated with the bounding box, allowing the model to be trained to detect and identify multiple types of objects at once. The confidence value associated with each bounding box provides a measure of how "sure" the model is that the proposed class is present in the subsection of the image bounded by the bounding box. In practice a confidence threshold is generally enforced so that only bounding boxes with high confidence are considered valid. A well trained model will produce bounding boxes with high confidence that completely enclose all instances of the corresponding classes in the image.

The perception pipeline utilises the bounding boxes generated by a FRCNN model to filter out parts of the pointcloud which do not contain asparagus spear features. This is achieved by generating a binary mask describing the regions of the input image where bounding boxes are present. This process is demonstrated in Figure 3.6 (a) and (b). The mask is then transformed into the image space of the pointcloud's depthmap using a homographic (perspective) transformation. This transformation requires registration between the pointcloud and RGB image. Specifics regarding the application of this registration are detailed in Section 4.5.1. Once the mask has been transformed a bitwise-and operation is applied between the transformed mask and depthmap of the pointcloud. This results in regions of the depthmap where asparagus features are not present being marked *False* and consequentially being removed from the corresponding pointcloud. Figure 3.6(c) and (d) demonstrate this process of neural network filtering.

The outputs of CNNs and other machine learning methods are highly dependent on their training data. It is difficult to ensure that a given training set will produce a model which generalises well to the domain of required use cases. Furthermore, the capacity for a specific model architecture to learn is limited by the depth/parameters of the network. Simply providing a larger and/or more diverse training set to a given CNN model will therefore not guarantee higher performance. These constraints mean that it is difficult to train a single FRCNN model to be an effective asparagus spear detector for



(a) (b)

(c) (d)

Figure 3.6: Example of the neural network filtering process. (a) shows an example image of an asparagus bed with annotations showing the predicted bounding boxes output by the FRCNN model. These bounding boxes are used to generate the binary mask shown in (b). The mask is then applied to the input pointcloud, shown in (c), in order to produce the resulting filtered pointcloud in (d).

all use cases. For this reason, a number of bespoke networks were utilised for use on individual fields and hardware platforms.

The following subsection discusses the configuration, training, and evaluation of an FRCNN model.

3.5.2 Labelling and Training

A dataset of 500-1500 images is first collected from the field on which the network is intended to operate. This is achieved by imaging a number of selected asparagus rows continuously, and selecting a uniform sampling of frames. In practice, the robotic platform on which the network is to be deployed, is maneuvered at its operational speed down the rows and a stream of the on-board camera is recorded. This is done for two reasons. Firstly, this ensures that the perspective of the images is as close to the intended use case as possible. Secondly, since the platform is moving at its operational speed, any motion artifacts or camera blur can be accounted for in the dataset. Both of these factors aim to create a training dataset that is as similar to the intended use case as possible, thereby increasing the generality of the network. Additional measures to generalise the training set were taken for some of the FRCNN models trained. These include:

- Collecting images from different times and days as to include images with a variety of lighting, weather conditions and plant conditions
- Randomly selecting the "direction of travel" of the platform while data collecting
- Selecting "difficult" sections of the field, where weed growth is high, or spear morphologies are unusual

After data collection, the images are split into three distinct groups; the training-set, test-set, and validation-set which contain approximately 75%, 12.5% and 12.5% of the images respectively. These proportions were chosen based on recommendations from the developers of the specific FRCNN imple-

mentation utilised in this work, discussed in Section 4.3.2. The training-set contains the majority of the images and will drive gradient decent during the training process. The validation-set is utilised during training to determine the total loss at the end of each iteration. It is important that the training-set and validation-set are distinctly separate to avoid testing on training data. The test-set is reserved to evaluate the network after training. The images are separated into these groups randomly, however the representation of each "type" of image is maintained. For example, a data-set containing 1000 images from "Row A", and 100 images from "Row B" will have 750 "Row A", and 75 "Row B" images randomly selected for the training-set. Additionally, "difficult" images are manually distributed across the three sets to ensure their proportional representation.

Each of the images in the dataset are then manually labelled. This process involves a human manually identifying all instances of asparagus spears in the images and determining the coordinates of each corresponding bounding box; in a sense the human is generating examples of what the FRCNN is desired to achieve. This is a labour intensive process involving considerable nuance. Section 4.3.2 discusses the implementation of FRCNN utilised for this work, and provides an example of labelling for a specific model. An evaluation of the resulting FRCNN model is also presented.

Another form of non-asparagus points which are filtered during this phase of the perception pipeline are flying pixels. The following subsection presents a novel geometric method which was developed to filter flying pixels.

3.5.3 Closest Point Filtering

Flying pixels are a common phenomena in ToF imaging that occur at the edges of objects where there is a discontinuous jump in range between the foreground and background elements. Such a scenario occurs at the edges of asparagus spears where the range between the spear and the background soil plane is discontinuous. The discontinuity in range can cause some of the frames

Figure 3.7: Pointcloud of two spears demonstrating a number of flying pixels. These pixels extend far behind the imaged spear in the direction of the cameras optical axis.

taken during the ToF imaging process to have significantly different ranges. When these ranges are averaged together the resulting range prediction can be erroneous, resulting in sharp edges from the scene appearing “smeared” in the range direction. Figure 3.7 shows an example of flying pixels in real-world ToF images of asparagus spears.

Flying pixels are problematic for the vision system because they obfuscate the true shape of the asparagus spear. This results in inaccurate base point predictions. It would be preferable if such points could be filtered out. There are two main ways this could be achieved. Firstly, this could be achieved by processing raw frames during acquisition and applying algorithms to detect the edges of target features in the scene. Secondly, the pointcloud could be filtered geometrically in order to remove the flying pixels.

The Kinect v2 camera does not provide access to raw frames. As such, a geometric filter was developed for the removal of flying pixels, named closest points filter (CP filter). The CP filter operates on individual point clusters

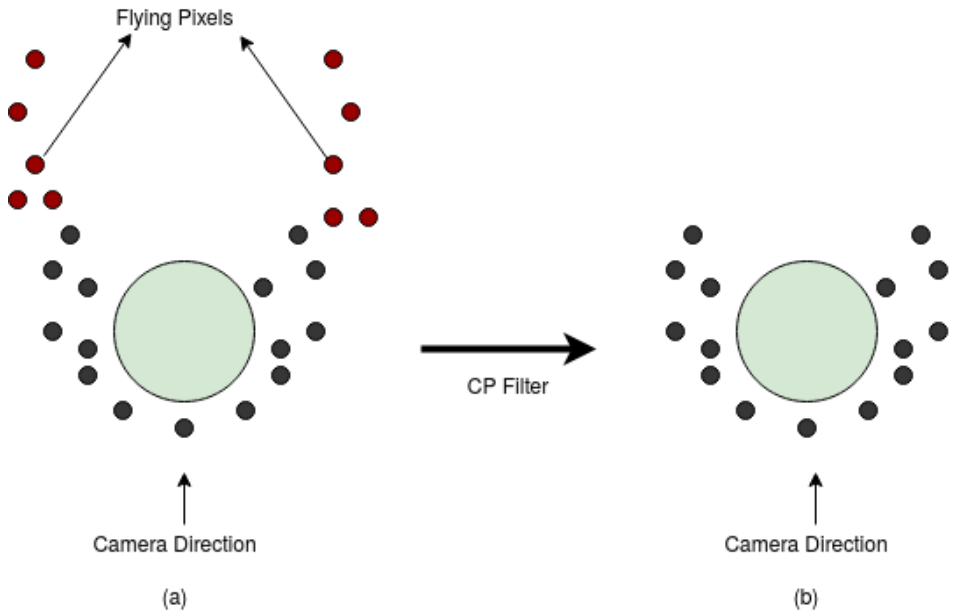


Figure 3.8: Illustration of the intended effect of the CP filter. The green circles in the figure represent a 2D slice of an asparagus spear. (a) illustrates what a typical pointcloud looks like prior to filtering, with the erroneous flying pixels shown in red. (b) demonstrates the desired output of the CP filter.

representing individual asparagus spears. The method takes advantage of the typical "horseshoe" geometry of a ToF image of an asparagus spear. Details about individual spear clustering and isolation can be found in Section 3.6.

Figure 3.8 demonstrates the flying pixels by illustrating a 2D slice through one of these structures. Figure 3.8 (a) shows an example of a pointcloud cluster prior to the application of the CP filter. The flying pixels, shown in red extend far beyond the asparagus spear in the direction of the optical axis of the camera. Figure 3.8 (b) shows an example of the desired pointcloud cluster after the CP filter is applied. The basic structure of the CP filter is outlined in Figure 3.9.

The filter operates on a list of pointclouds, each element of which represents a single asparagus spear. This list of pointclouds is provided by previous elements of the detection system. In essence each pointcloud in this list is a subset of the overall input pointcloud, selected to best represent a single asparagus spear.

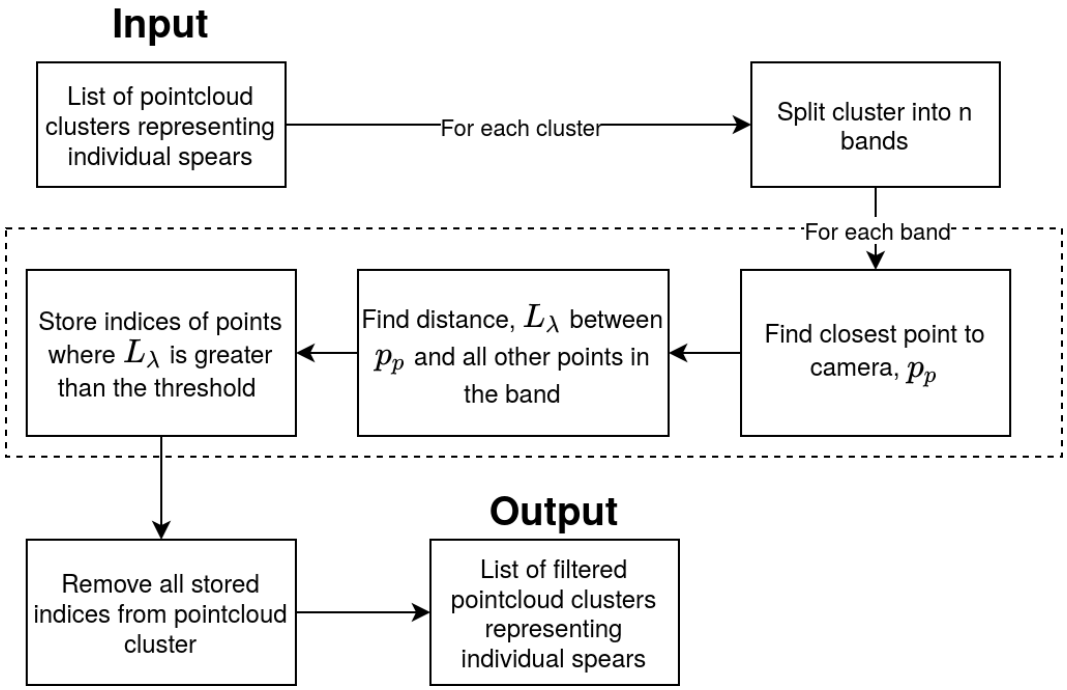


Figure 3.9: Overview of the CP filter algorithm.

The algorithm begins by splitting each pointcloud into a series of bands. This is achieved by generating a series of planes which are parallel to the soil plane (detected in previous elements) and separated equally in the direction that is normal to the soil plane. The planes are arranged such that the first, and nth planes are coincident to the bottom and top points in the pointcloud respectively. For this purpose, "bottom" means the lowest point projected onto the soil plane normal, and "top" means the highest point. Figure 3.10 shows an example of these planes overlaid on an example pointcloud.

For each plane a projection of all points in the pointcloud that are above the plane, but below the plane above is made. This results in a single "slice" of the asparagus pointcloud similar to that shown in Figure 3.8(a). Each of these projections is considered to be a "band" of the original pointcloud. Since there are no points above the highest plane there will be $n - 1$ bands for a given pointcloud.

Each band is then analysed independently in order to identify the flying pixels. Firstly, the closest point in the band to the z-axis of the camera is determined. This point is set as the pilot point, $\mathbf{p}_p$ for the band. The distance

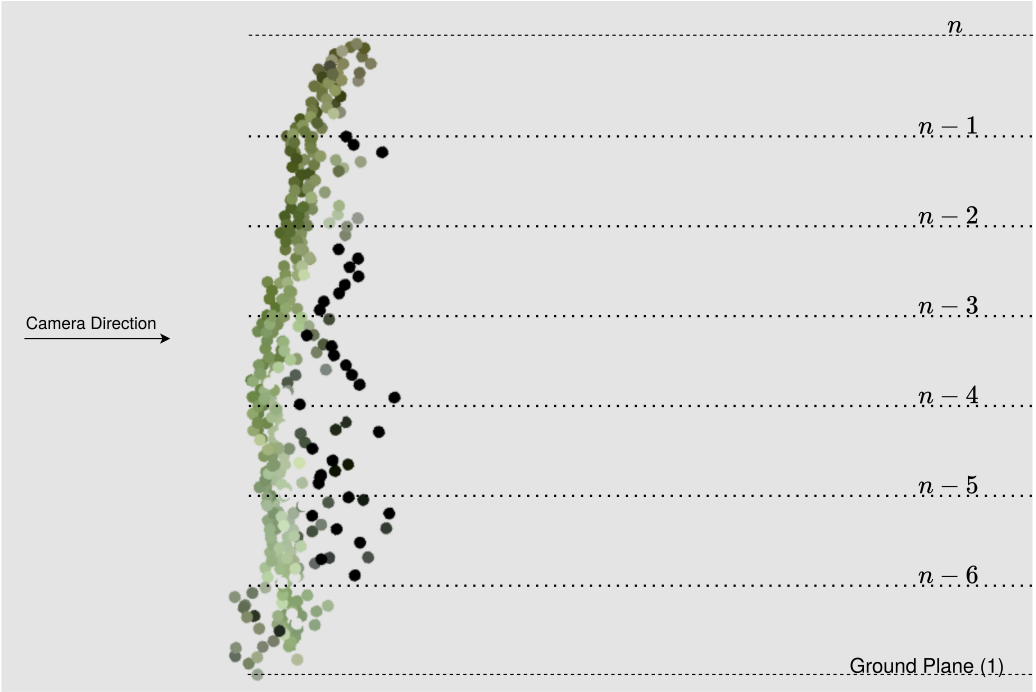


Figure 3.10: Side view of an asparagus pointcloud demonstrating the banding procedure. Each dotted line represents a plane that is normal to the ground plane.

between the pilot point and all other points in the band is then calculated. Since the points have been projected onto a 2D plane this distance is the length of the line connecting each point $\mathbf{p}_\lambda$ to the pilot point $\mathbf{p}_p$ and is calculated as:

$$L_\lambda = \|\mathbf{p}_p - \mathbf{p}_\lambda\| \tag{3.35}$$

In the original, 3D pointcloud this distance is the length of the projection of a line connecting each point to the pilot point and a plane which is parallel to the soil plane and coincident to the pilot point. The distance between each point in the band and the pilot point is then thresholded in order to eliminate points that are too far to be considered part of the asparagus surface. This results in a list of "outliers" for each band. It is then trivial to eliminate the corresponding points from each input pointcloud.

The closest point filter was tested using laboratory data. This experiment is presented in the following subsection.

3.5.4 Evaluation of the Closest Point Filter

An experiment was performed in which wooden dowels were arranged in a grid of known dimensions and imaged using a Kinect camera. Applying the closest point filter to the pointclouds resulted in more precise base point predictions.

Firstly, a calibration plate was manufactured from a sheet of aluminium. The sheet, pictured in Figure 3.11, had a regular grid of 20mm holes cut out of it such that wooden dowels (also with a diameter of 20mm) could be planted into the plate. This allowed various simulated scenes to be created with a high degree of accuracy and repeatability. The surface of the aluminium plate was painted a matte black in order to eliminate specular reflection, and mitigate multipath errors. The aluminium plate was placed in view of the camera and taped to the floor of the workshop to ensure that it did not move during imaging. The camera was positioned at a height of 900mm, at an angle of 45°, based on the results of the experiment presented in Section 3.4.

Experimental data was collected by placing an individual dowel into the plate and taking 20s long recordings of the Kinect's data stream for a number of different dowel positions. The resulting dataset contained recordings corresponding to 45 positions, arranged in a 5×9 grid. The spacing between the dowels in this grid was 100mm based on the construction of the calibration plate.

A dataset of 500 images of dowels in random locations was also collected and used to train a FRCNN model to allow the neural network filter to function for the test.

Each recording was partially processed by the perception pipeline, where both soil plane removal, and non-asparagus point removal were applied. The resulting pointcloud consisted of only those points pertaining to the wooden dowel from each recording. A number of points from each of these recordings were sampled and projected to the soil plane. Figure 3.12(a) shows the entire grid of points collected via this method.

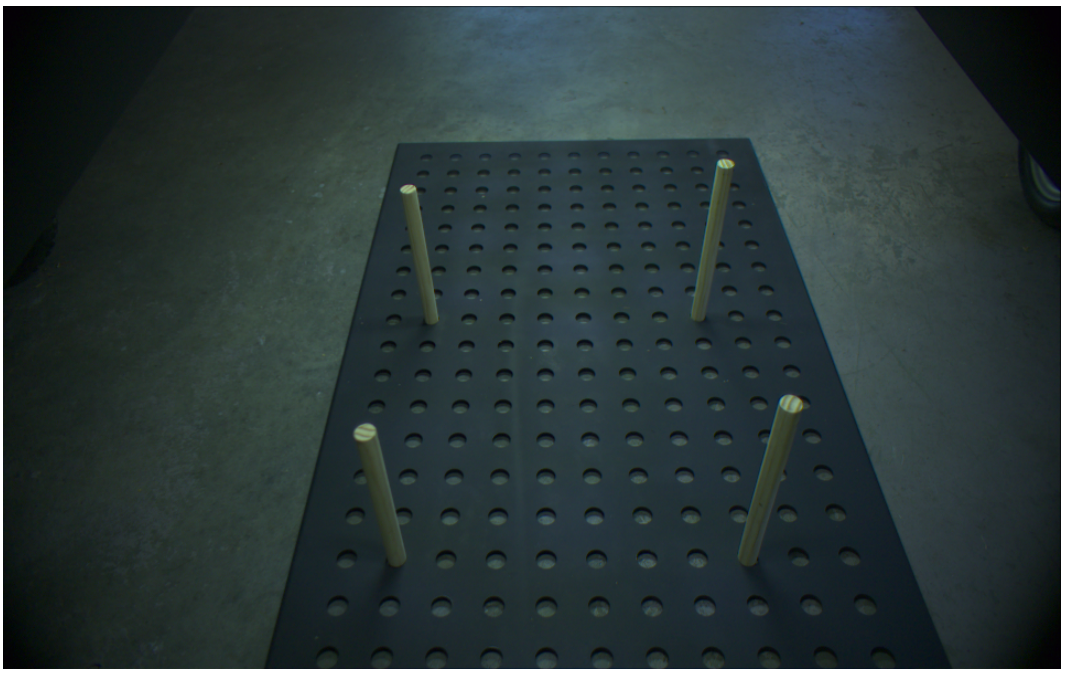
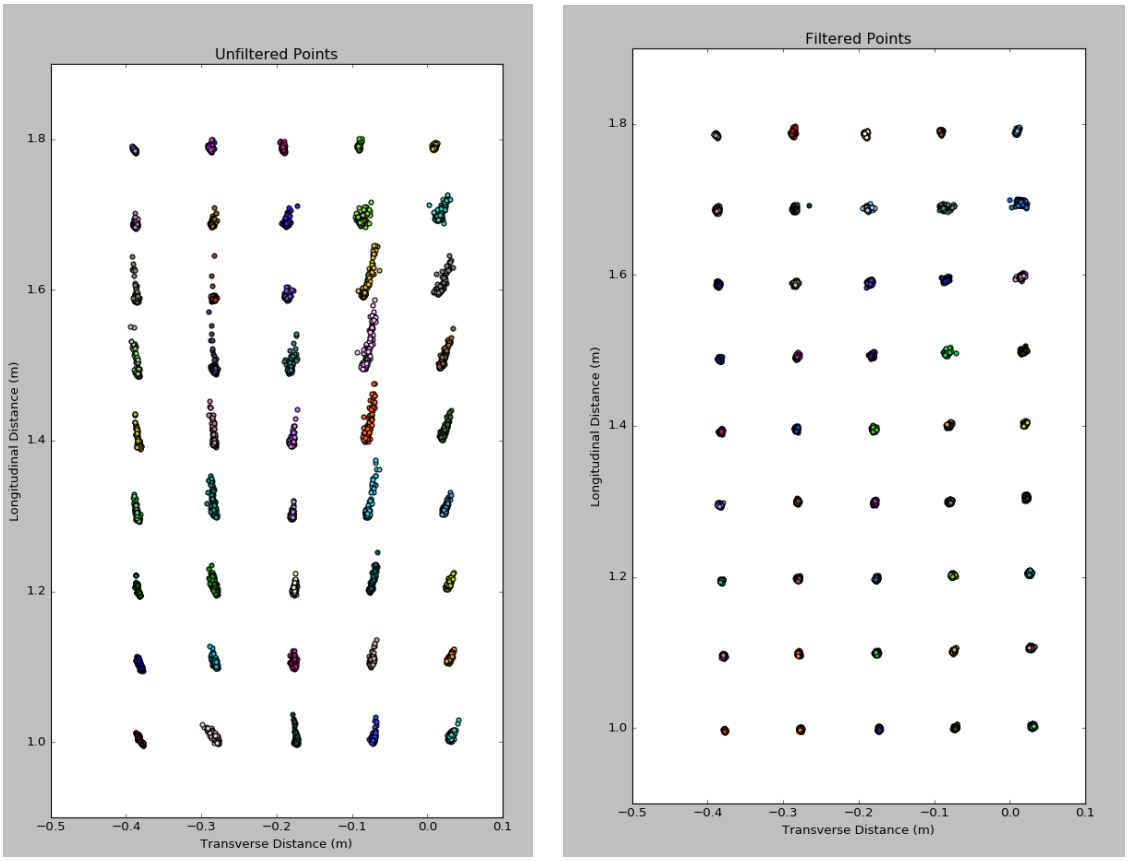


Figure 3.11: Aluminium calibration plate for evaluating the CP filter. Each 20mm hole is spaced 100mm apart, allowing the relative ground truth position between dowels to be determined.

It can be seen in Figure 3.12 that the points which describe the wooden dowel are significantly spread in the range direction of the camera. This is expected due to both the limited range precision of the Kinect camera, and the presence of flying pixels in the image. It seems that these effects are most pronounced at a range of 1.4-1.7m, and less pronounced both close to the camera, and far away. There are two effects occurring here, both of which occur due to the 45° mounting angle of the camera. Firstly, due to the perspective of the camera lens, objects that are further away from the camera take up less angular size in the depthmap. This results in objects that are far from the camera being represented by fewer points, effectively reducing the probability that an individual pixel will fall on the edge of an object. For dowels located at the furthest distance from the camera this results in a relatively cohesive set of points describing the position of the object. For dowels which are closest to the camera, the 45° mounting angle of the camera results in these objects being viewed from a much more extreme angle. Therefore, the potential differences in range seen at the discontinuous range boundaries surrounding the object



(a) **(b)**

Figure 3.12: Plots showing various points pertaining to wooden dowels collected during the evaluation of the CP filter. (a) shows the points prior to filtering and (b) shows the remaining points after the CP filter was applied.

are much smaller. Objects closer to the camera therefore have a much more limited spread of pixels. However, dowels in the middle of the scene (range of 1.4-1.6m) are affected significantly by the range precision of the camera and the presence of flying pixels. For these dowels the mean of all their points differ significantly from the ground truth.

This process was then repeated, and the resultant points passed through the CP filter. Figure 3.12(b) shows the result of applying the CP filter. The figure shows that the point clusters are much more concentrated than prior to filtering.

An analysis of these datasets was conducted in order to quantify the effectiveness of the CP filter. To begin, the standard deviation of points recorded at each position, $\sigma(H_{\lambda_2})$ for $\lambda_2 \in \mathbb{N} \cap [1, 45]$, was determined. This value provided a measure of how "spread out" the points in each cluster, H_{λ_2} were. This was achieved by calculating the geometric mean of all points within each cluster, $\mu(H_{\lambda_2})$, and finding the mean range between $\mu(H_{\lambda_2})$ and all other points in that cluster. The mean of all these mean ranges across all positions was then calculated as $\bar{\mu}(\sigma(H_{\lambda_2}))$

It is also important that the positional accuracy of each point cluster is maintained in order to facilitate accurate base-point predictions from the perception pipeline. Although the aluminium plate is manufactured accurately it is difficult to label ground-truth for this dataset. This is because it is difficult to align the real-world and camera frames. Figures 3.12(a) and (b) show that there is a slight rotational misalignment of the apparent grid of dowels with respect to the coordinate system of the camera. Without an understanding of this transform, analysis of the positional accuracy of these points is limited to a relative comparison between detected point clusters. The construction of the aluminium plate was such that the distance between each dowel was expected to be 100mm (100mm grid). In order to evaluate each of the datasets the mean and standard deviation of ranges, $\mu(R_{\lambda_2})$ and $\sigma(R_{\lambda_2})$, between each

Table 3.2: Evaluation of the CP filter. The table shows both the intra-cluster mean of standard deviations in range, as well as the mean distance, and standard deviation of distance, between clusters.

	Intra-Cluster	**Inter-Cluster**	
	$\bar{\mu}(\sigma(H_{\lambda_2}))$	$\bar{\mu}(R)$	$\bar{\sigma}(\mu(R))$
Unfiltered	6.5	101.943	5.005
Filtered	2.1	101.227	4.370

cluster mean and its four nearest neighbours was calculated. The average of these values across all point clusters, $\bar{\mu}(R)$ and $\bar{\sigma}(\mu(R))$ was then found.

It can be seen from Figure 3.12(a) that the major axis of each cluster is pointing in the same direction. This is as expected since the range direction of the pointcloud image is more prone to error than the x and y coordinates. However, it can be seen that these major axes are pointing towards approximately the -0.15m position on the transverse axis. This offset is due to the fact that the ToF sensor on the Kinect camera is offset from the origin of the pointcloud it reports. Specific information regarding the internal construction for the Kinect camera are unavailable, however based on rough physical measurement of the camera it appears as though the pointcloud origin is centered at the top right corner of the camera case as shown in Figure 3.13.

There are some limitations with filtering points with CP filter. These limitations are discussed in the following subsection.

3.5.5 Limitations of CP Filter

The method filters out erroneous points based on distance. However, due to the inconsistent characteristics of pointclouds obtained at different locations in the image, coupled with the camera perspective the number of points which define an asparagus spear vary depending on the position of that spear in the field of view of the camera. Since the method removes points from the point cluster the amount of information defining a spear is reduced even further.

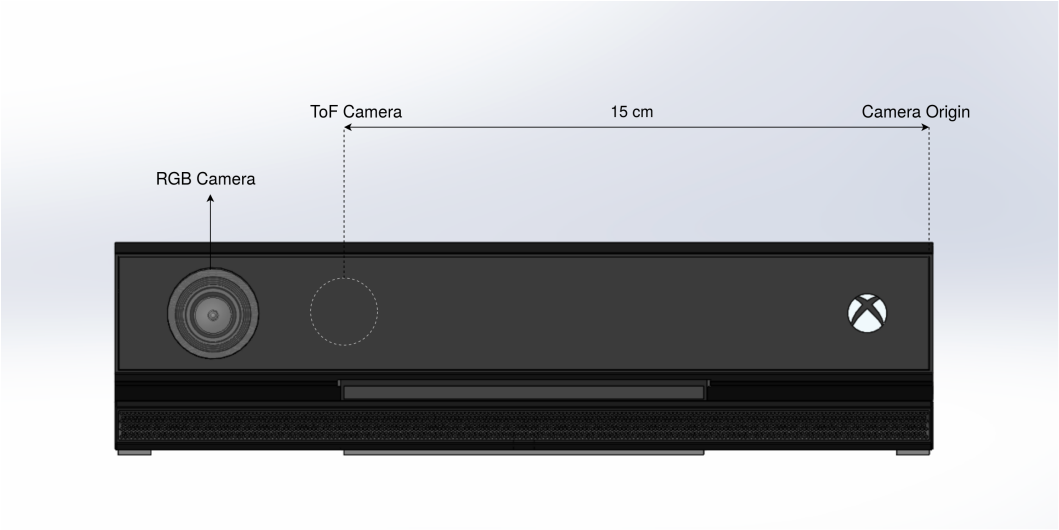


Figure 3.13: View of the Microsfot Kinect V2 camera. The camera reports pointcloud with an origin located at the top-right corner of the camera as shown.

Ideally, rather than removing such points, a correction could be made in order to bring the points back to the surface of the asparagus spear. The cylindrical nature of asparagus spears make the feature mapping component of such a method relatively simple. For example, a measure of the span of points in each band could be assessed by considering the spread of points along an axis which is perpendicular to the optical axis of the camera. This span could be used to inform the algorithm about the potential diameter of the asparagus spear slice that the band represents. All points in the band that are further than the distance threshold from the pilot point could then be shifted in the direction of this line such that the square error between all of these points and the theoretical asparagus spear slice is minimised.

Another limitation of this approach is that each of the band planes must be parallel. The method, therefore, does not extend well to cover complex spear geometries that cannot be well modelled by a piece-wise cylindrical prism. For example, spears that are significantly bent, or that wave significantly are not accounted for. Such spears result in multiple "sections" of the asparagus spear existing in a single band. This nullifies completely the assumptions made in processing the band, resulting in large numbers of inliers being removed.

More sophisticated analysis of the bands could alleviate this issue, although to achieve nth degree robustness it would likely be very computationally expensive. There is also the possibility that machine learning methods could be well suited to the task of separating inlier points from flying pixels, however if such an approach was to be utilised, better results could likely be achieved by applying the methods directly to the pointcloud cluster, avoiding the need for spear banding entirely. In the same vein, 3D model fitting to the entire pointcloud cluster could be considered, however the noise, and large variance in expected inputs from real fields could make this task very difficult to rigidly define. Additional limitations and suggested improvements to the CP filter, particularly non-geometric methods, are described in Appendix A.1.

Once the soil plane and non-asparagus points have been removed the original input pointcloud consists of sparsely separated groups of points pertaining to various asparagus spears in the scene. The following section presents the process of asparagus spear grouping.

3.6 Clustering and Spear Identification

In order to analyse each spear individually, the perception system must cluster each of these groups of points into a series of separate, isolated pointclouds. This process is known as clustering.

The general purpose of a clustering algorithm is to divide some input pointcloud $\mathbf{P} \in \mathbb{R}^3$ into a set of n_c sub-pointclouds or clusters, $\mathbf{\Xi} = \{\Gamma_1, \Gamma_2, ..., \Gamma_{n_c}\} \in \mathbf{P}$ such that all points, $\{\gamma_1, \gamma_2, ...\gamma_{m_c}\}$ in every $\Gamma \in \mathbf{\Xi}$, are geometrically cohesive in some way. Geometric cohesion merely means that the maximum Euclidean distance between any point and its nearest neighbour is less than some constant threshold, κ_r. The specific algorithm, named "Euclidean Cluster Extraction" is provided by Point Cloud Library (PCL) and functions as follows:

1. A Kd-tree [101] representation of the input pointcloud, $\mathbf{P}$, is generated. Kd-trees are a type of binary search tree, which are useful for facilitating efficient nearest neighbour searches of K-dimensional data.
2. The algorithm then sets up an empty list of clusters, C_E and an empty queue of points, Q_E, that need to be checked.
3. Some initial point $\mathbf{p}_i \in \mathbf{P}$ is selected and added to Q_E.
4. For every point $\mathbf{p}_i \in Q_E$:
   (a) Find every point in $\mathbf{P}$ which is within a sphere of radius κ_r from $\mathbf{p}_i$;
   (b) Each of these points is then checked to determine if it is already part of any cluster in C_E. If not, the point is added to Q_E.
5. Once every point in Q_E has been evaluated, all points are added to a new cluster Γ_{i_3} and stored in C_E. The point queue, Q_E is then cleared.
6. A new starting point $\mathbf{p}_i \in \mathbf{P} \notin C_E$ is added to Q_E and the process repeats from step 4 until all $\mathbf{p}_i \in \mathbf{P}$ have been assigned to a cluster.

The final step of the perception pipeline is to generate spear models based on each point cluster that results from Euclidean Cluster Extraction. The purpose of this procedure is to take each individual cluster of points, pertaining to a single spear from the input pointcloud, and to generate a sanitised representation of that spear to be utilised by downstream processes such as mapping software, or harvesting systems. The following subsection presents the method by which this is achieved.

3.6.1 Spear Modelling from Clusters

Line-of-best-fit approaches, including high-order polynomial fitting were initially considered as a method for modelling asparagus spears based on individual point clusters. However, it was found that typical spear morphologies required model fitting of such a high order that such an approach was not per-

formative. Additionally, base-point predictions made from such models were highly variable as these fitting methods are sensitive to inter-frame noise.

A simple method was developed that models each asparagus spear as a piece-wise set of a lines connecting a series of points along the length of the spear. The following procedure is used to develop this model for all clusters, Γ_{ϵ_2} for $\epsilon_2 \in \mathbb{N} \cap [1, n_c]$.

1. The cluster is separated into $n_m - 1$ bands by bisecting the cluster with n_m planes, which are normal to the soil plane, and uniformly distributed along the soil planes normal vector. This banding procedure is identical to the banding procedure utilised for the CP filter.

2. For each band, B_λ for $\lambda \in \mathbb{N} \cap [1, n_m - 1]$, the geometric centroid of all points contained by the band, q_λ, was calculated and added to a point-list associated with the cluster, $\mathbf{D}_{\Gamma_{\epsilon_2}}$. Additionally, an approximation of the width for each band was calculated by finding the span of all points in the projection of the band on to the lower bisecting plane.

3. The base-point, and top-point were then added to the front of $\mathbf{D}_{\Gamma_{\epsilon_2}}$ such that the base-point is the first entry. The base-point was determined by finding the intersection between the line connecting q_1 to q_0 and the soil plane. The top-point was determined by finding the furthest point from the soil plane in the projection of all points in B_{n_m-1} to the normal vector of the base plane.

The resulting point-list can be used to determine various information about each detected spear:

- The length of the detected spear can be calculated as the total length of line segments that connect the points in the point list
- The base-point of the asparagus spear is simply the first point in the point list
- The width estimations associated with each band allow a concept of spear width, and taper to be determined

- The piece-wise function, constructed by connecting the points in the point-list with linear line segments gives a good first-order approximation of the spear's morphology

3.7 Summary

This chapter has presented a novel perception pipeline for the detection of asparagus spears in a commercial setting. An overview of the pipeline is shown in Figure 3.14. The pipeline operates on a time-synchronised RGB image and pointcloud obtained through ToF imaging. There are three main operations performed by the pipeline in order to achieve this task, namely soil plane removal, non-asparagus point removal, and asparagus spear grouping/modelling.

Soil plane removal was achieved by calculating a plane model based on the input pointcloud, and applying a distance threshold to eliminate points that are too far from the modelled plane. Two methods were investigated for determining the soil plane model, namely MHM and RANSAC. The resilience of these methods with respect to field clutter was evaluated. It was concluded that both methods are suitable for soil plane segmentation in a commercial setting.

Non-asparagus point removal was achieved by firstly applying a FRCNN model to produce a set of bounding boxes, highlighting the areas in an RGB image where asparagus features were present. These bounding boxes were then utilised to segment corresponding areas of the input pointcloud, broadly cutting away many points not pertaining to asparagus spears. Euclidean Cluster Extraction was then applied to separate the remaining spear clusters from the pointcloud. A novel geometric point filter, the CP filter, was then applied to the point clusters in order to remove flying pixels and improve the precision of the point clusters.

Finally, a geometric method was applied to generate a spear model based on each of the remaining point clusters.

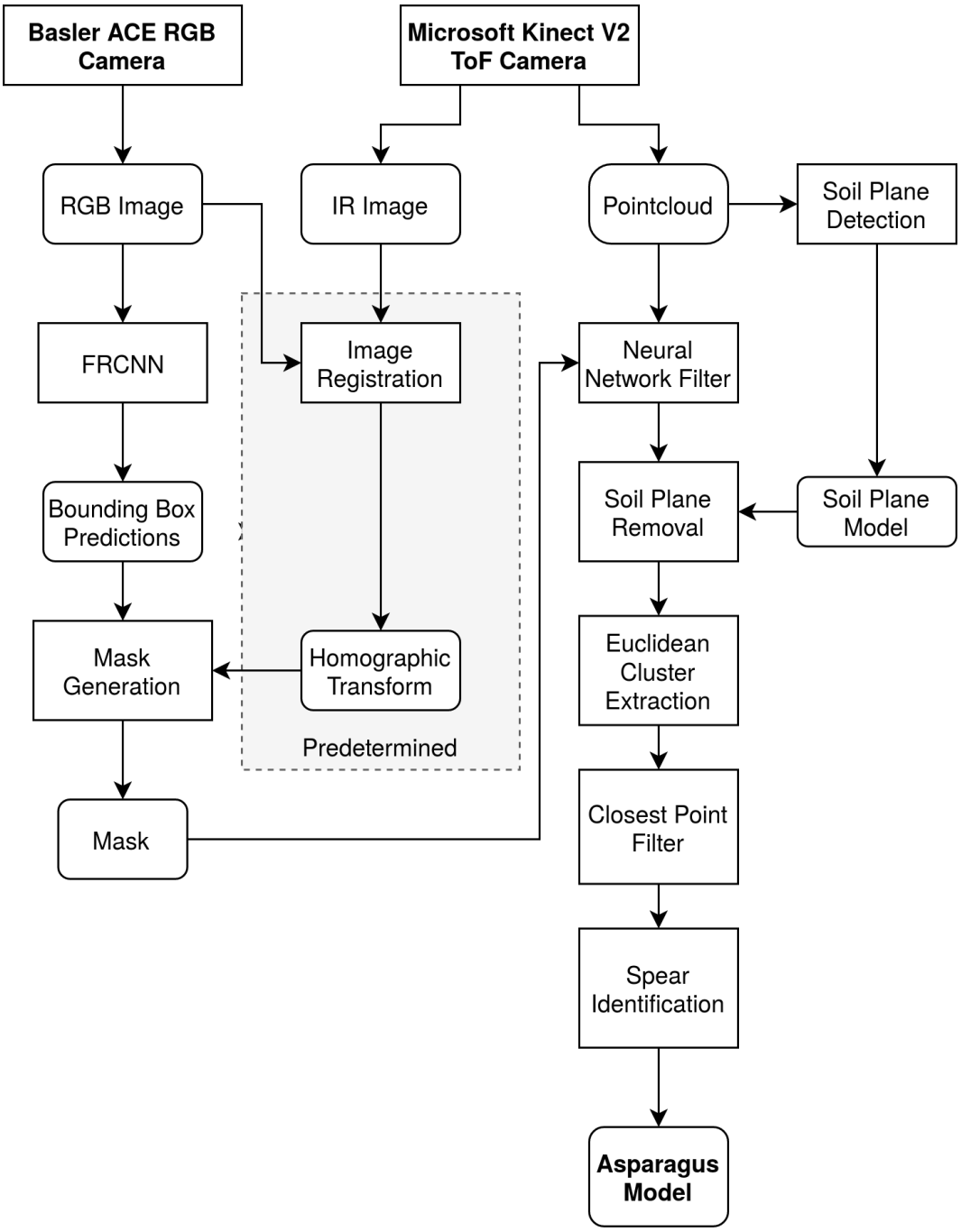


Figure 3.14: Outline of the full novel perception pipeline. The section marked as "predetermined" is calculated offline as described in Section 4.5.1.

In order to realise a working robotic harvester, a software implementation of this perception pipeline is required. This implementation, and resulting vision system are presented in Chapter 4.

Chapter 4

Vision System Integration

The previous chapter presented a theoretical perception pipeline for the detection of asparagus spears. This chapter focuses on the implementation of this pipeline into a stand-alone vision system and the subsequent integration of this system with a robotic platform to realise a working asparagus harvester.

The chapter begins by presenting a novel concept for a robotic selective asparagus harvester that utilises the novel perception pipeline formulated in the previous chapter. A proof-of-concept robotic harvester, named AHR-1, is then presented based on this concept. The chapter continues by discussing the hardware elements of AHR-1 including the frame, sensors, electronics and robotic elements. Software elements required to implement the perception pipeline into a working vision system are then discussed and the subsequent integration of the vision system into AHR-1 is presented. As part of this discussion the data acquisition, asparagus tracking system and robot state tracking modules are discussed, as well as the control elements required for overall control, such as the motion controller interface, and target scheduling solutions.

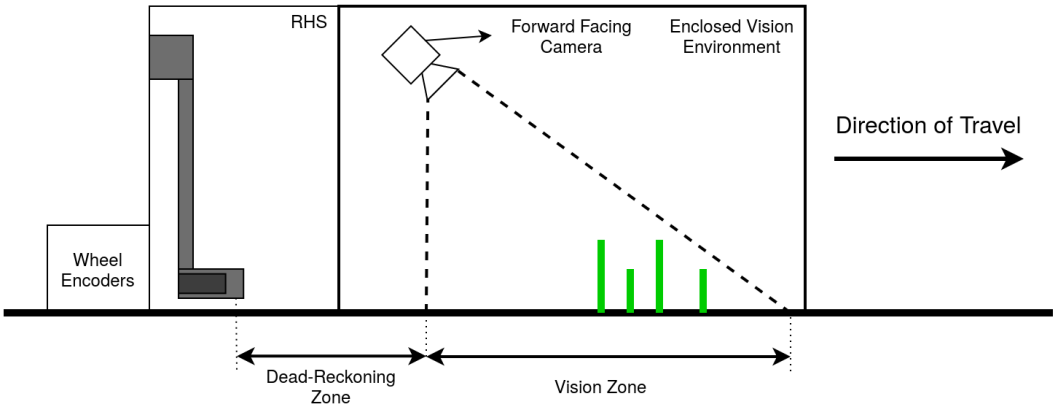


Figure 4.1: Overview of a novel concept for a robotic selective asparagus harvester. The diagram outlines the major components of the proposed system.

4.1 Novel Concept for a Robotic Selective Asparagus Harvester

A novel concept selective asparagus harvester was developed to allow a proof-of-concept of the vision system. Figure 4.1 illustrates the main components of this concept. The main goal of this concept was to enable a low-cost platform to be constructed in order to explore the integration of the vision system with hardware elements, and to understand the required end-effector precision and accuracy to enable the targeted asparagus spears to be harvested. The main features of the proposed concept harvester are:

- A tractor drawn carriage, designed such that it straddles a single row of asparagus, and travels down the direction of the row
- Wheel encoders mounted at each wheel used to track the travel down the row
- A forward mounted ToF and RGB camera for detecting oncoming spears
- A solid light-shield surrounding both cameras and the part of the asparagus row that is in their field-of-view. The light-shield is designed to block direct sunlight exposure with the part of the field being imaged in order to reduce over-exposure and saturation of the ToF sensor.

Additionally, the light-shield provides resistance to environmental wind, which can move spears during imaging

- A robotic harvesting system (RHS), consisting of a multi-axis linear translation stage mounted at the rear of the machine. This RHS is used to manipulate an end-effector to coordinate the harvesting of spears

The proposed asparagus harvesting robot operates as follows:

1. The robot translates down a single asparagus row, drawn by a standard tractor at a constant speed.
2. As the robot moves down the row both cameras continuously stream data to a processing unit housing the vision system.
3. Wheel encoders continuously stream data to the processing unit as the robot translates down the row. The processing unit uses the information to keep track of the robot's position.
4. The processing unit utilises base predictions from the vision system, and positional data from the wheel encoders to calculate when a target spear will be passing beneath the RHS. Commands are sent to the translation stage so that the end-effector is positioned to intercept the spear as the machine continues to translate forward.
5. The end-effector harvests the spear and transports it to storage.

In order to achieve real-time performance it is important that the required amount of data-processing is minimised. The perception pipeline, outlined in Chapter 3, is designed to operate on unobstructed images of the asparagus bed. It is therefore important that the "vision zone" of the harvester is clear at all times in order to eliminate the need for expensive pre-processing of input data. This means that it is not possible to have the RHS in the field-of-view of the camera, necessitating a period of dead-reckoning for each targeted spear between the time of detection, and time of harvesting. During the dead-reckoning period, the base position of each targeted spear is updated on an open loop basis, purely based on encoder data. This is bound to be less

accurate than a closed loop solution, where the target is tracked right up until the point of harvest, however, is likely much more preformative.

This concept addresses many of the weaknesses present in asparagus harvesting robots from the published literature. Firstly, many of the most successful asparagus harvesting robots, such as the Geiger-Lund and Haws harvesters, utilise multiple harvesting tools, and a gate-based detection strategy. This fundamentally limits the detection resolution of the machine, and guarantees high levels of collateral damage during the harvest. In contrast the proposed concept utilises a contiguous vision zone, and linear axis to eliminate this problem. Similarly, robots such as the GARotics harvester have previously opted for closed-loop detection methods that track target spears right up until the point of harvest. While this eliminates the need for dead-reckoning, the added computational strain involved with reasoning about the presence of the harvesting elements inside the vision zone is likely to impact detection time and limit the overall machines performance.

AHR-1 was designed, and built as a proof-of-concept machine based on this novel harvester concept. The relative low-cost approach adopted for AHR-1 resulted in the complexity of the harvesting elements being somewhat limited. Primarily, the machine served to validate the translation of image-space spear base predictions to world-space target locations. AHR-1 and its constituent components are described in the following sections.

4.2 AHR-1 Architecture

Based on the novel harvester concept presented in the previous section a functional robotic harvester was developed; This robot is referred to as AHR-1. A system diagram and section view and of AHR-1 can be seen in Figures 4.2 and 4.3 respectively, highlighting the the key components. An image of the harvester during field trials conducted in California during the 2019 season can be seen in Figure 4.4.

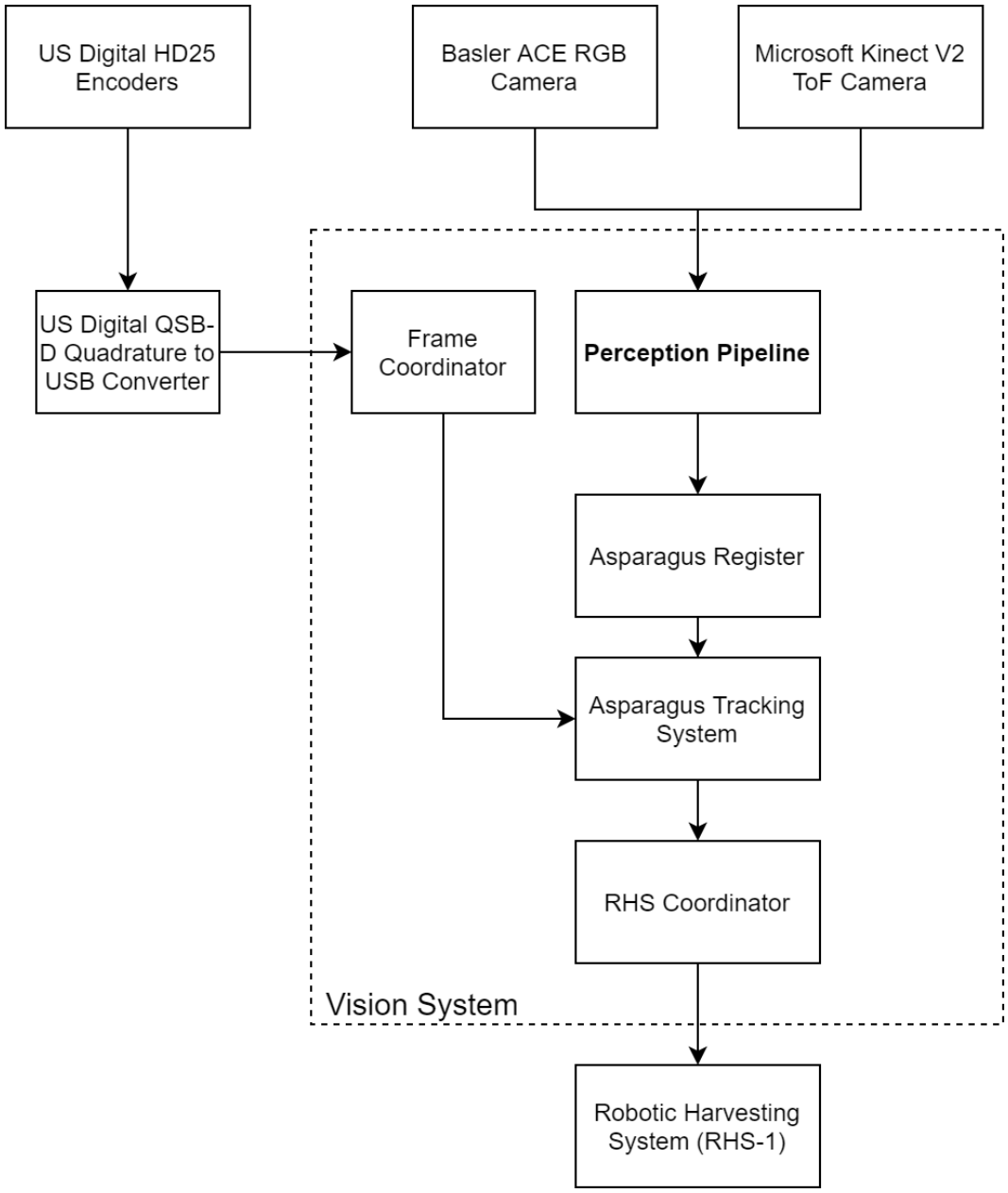


Figure 4.2: System diagram for AHR-1.

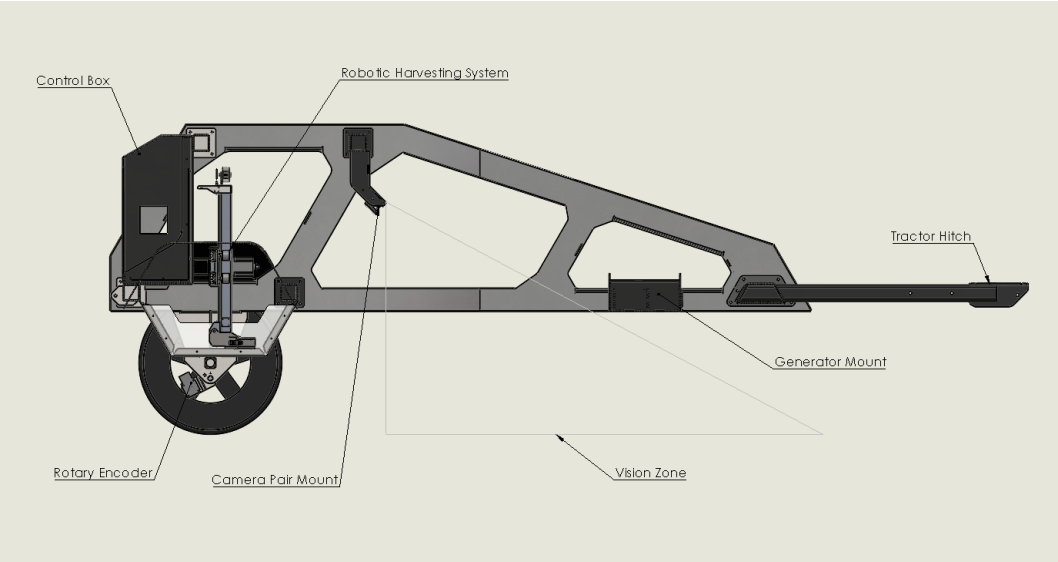


Figure 4.3: Section view of a CAD model of the harvester highlighting the various subsystems.

Figure 4.4: AHR-1 in operation on an asparagus row in Los Banos, California.

The frame of AHR-1 was constructed from a combination of standard steel sections and laser-cut folded steel sheet. The frame was built to be extremely robust, prioritising mechanical rigidity to material optimisation in order to minimise changes in the rigid offsets between the various components during operation. This allows rigid transformations between coordinate frames in software to better represent the actual physical system. More sophisticated data processing and more advanced sensor utilisation could allow similar performance from a more lightweight frame, however the optimisation of this balance was not a concern for this proof-of-concept. AHR-1 is powered by a 1.6kW petrol generator, mounted at the front of the machine near the draw bar. The generator is mounted atop a plate that is isolated from the main frame by rubber dampers in order to mitigate the propagation of vibration from the generator to other elements of the machine. At the rear of the machine are mounted a camera array, and RHS, as well as the main control box that houses the system's electronics.

The camera array is rigidly fixed to a steel camera mount near the rear of the frame. The camera array consists of a Kinect camera, and a Basler ACE RGB camera. The camera mount, shown in Figure 4.5, is located at a height of 950mm above the soil plane, and is orientated such that the optical axis of each camera is at 45° to the soil plane. The camera mount is configured such that the optical axis of each camera is aligned both vertically, and with the center of the asparagus row. This ensures that most targets are closer to the center of the image frame, mitigating the effects of transverse flying pixels due to the low range precision in the ToF image. Surrounding the camera array, and the section of asparagus bed being imaged is an aluminium light shield, which house several 24V LED strip lights used to provide consistent illumination during imaging while rejecting direct sunlight.

AHR-1 has a simple wheel assembly consisting of two wheels with diameters of approximatly 600mm, connected to two rigidly fixed stub axles. AHR-1 does not have a suspension system. Each wheel has a rotary encoder assembly

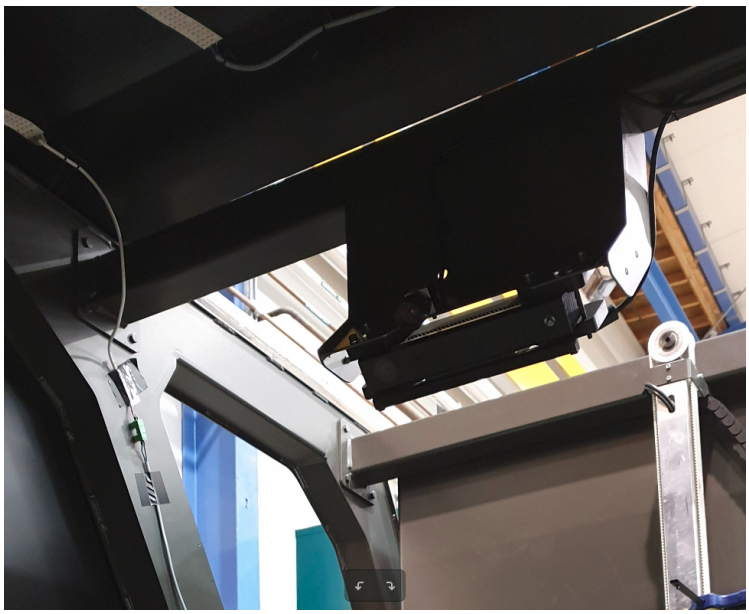


Figure 4.5: AHR-1's camera mount with Microsoft Kinect V2 and Basler ACE cameras attached.

connected underneath the axle. The encoder assembly consists of a spring loaded swing arm with a US Digital Digital HD25 rotary encoder mounted to the end. The output shaft of the encoder is connected to a rubber runner wheel that makes contact with the inner rim of the wheel. The spring arm ensures contact between the runner wheel and the inner rim of the wheel during operation. A US Digital QSB-D quadrature-to-usb converter is connected to the output of each encoder. This device converts the quadrature signals from the encoder to a digital number (unsigned long) and facilitates communication with the main system PC via a usb connection. The device provides several utilities for polling, resetting and configuring the operating resolution.

AHR-1 has a control box mounted at the rear which contains all of the electronic components for the RHS, vision system, and control environment. The control box contains power rails at:

- 230v 50hz AC from the generator
- 24V DC from AC-DC power supply
- 12V DC from AC-DC power supply

The main components housed in the control box are:

- Galil Motion controller used to coordinate motion of the RHS
- Servo drivers for each motor in the RHS
- Main system PC
- WiFi router used to allow remote access to the main system PC during operation

The specifications of the main system PC are:

- Intel I5-5700K Processor
- Nvidia Titan-X Pascal GPU
- EVGA 500W Power supply
- Samsung 950 EVO 500GB M.2 SSD
- 32GB RAM

4.2.1 Robotic Harvesting System (RHS-1)

The robotic harvesting system of AHR-1 (RHS-1) is a robotic device used to perform the asparagus harvest. The device consists of a 2-axis linear translation stage, a motion controller and an end-effector. The design of the 2-axis linear translation stage is based largely on Barnett's Linear Kiwifruit harvesting Robot (LHR) [102, 103]. Figure 4.6 shows a system diagram for the system.

The translation stage is comprised of a main rail, which spans laterally across the machine (y-axis) and a smaller vertical rail (z-axis) fixed to the carriage of the main rail. The system utilises an innovative T-drive mechanism to actuate both axes. A 24V stepper motor is mounted at each end of the main rail. The main carriage is fitted with four runner wheels, and a runner sprocket is attached to the top of the z-axis as shown in Figure 4.7. A belt is fitted such that it forms a T-shaped loop, engaging with the runner wheels, motor sprockets and runner sprocket. Both ends of the belt are fixed near the bottom of the z-axis. This configuration allows full control of both the y-axis and z-axis

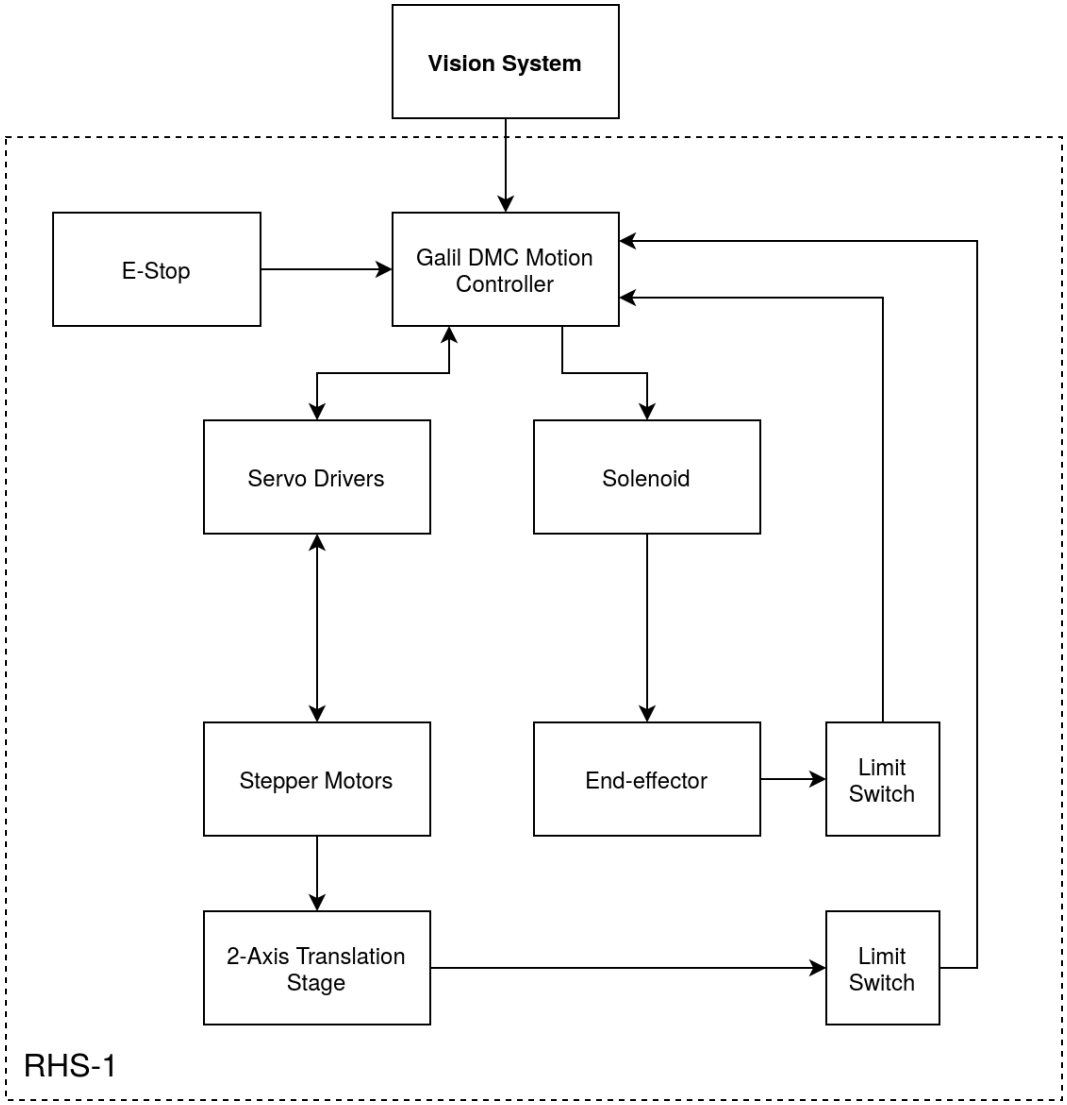


Figure 4.6: System diagram for RHS-1.

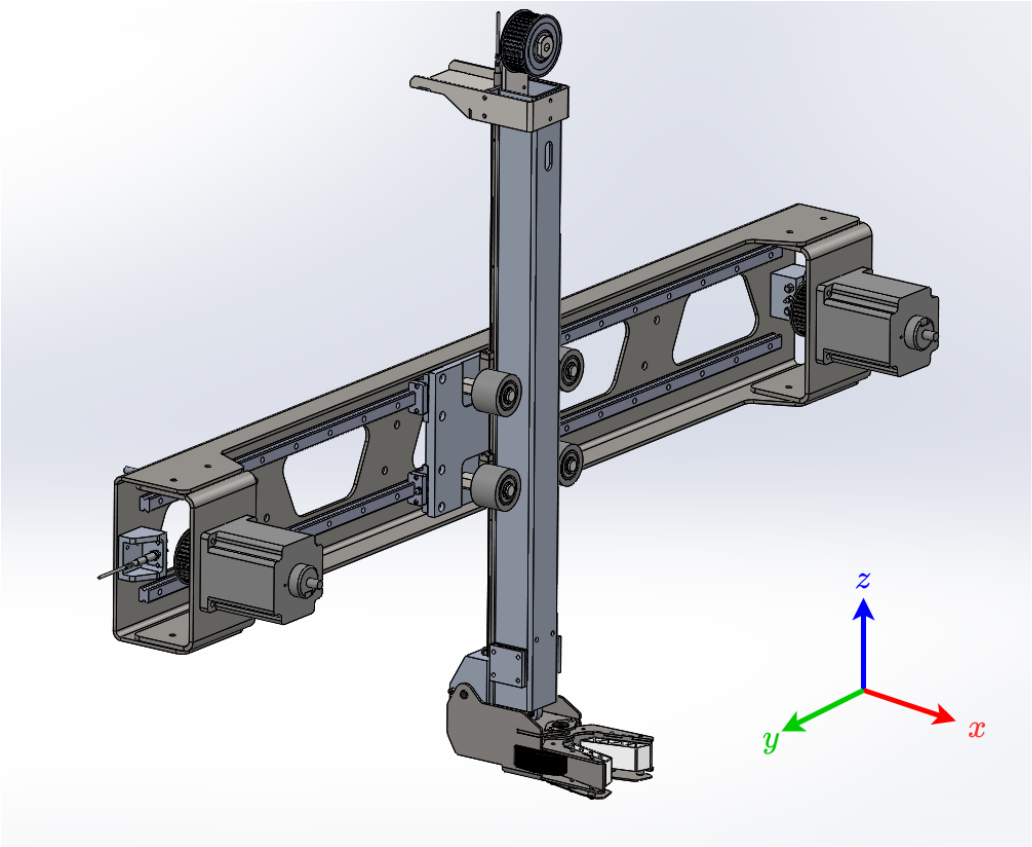


Figure 4.7: RHS-1. The device is a 2-axis linear rail system that can maneuver the robots end-effector to facilitate harvest.

based on the differential of motor speed. For example, pure vertical motion of the z-axis is achieved when the speed of rotation of both motors are equal in magnitude but opposite in direction, while pure translation in the y-axis is achieved when the motor speeds are equal in both magnitude and direction. Any combination of y-axis and z-axis motion can be achieved by some drive differential between these. The advantage of this configuration is that both motors remain stationary during operation, significantly reducing the inertia of the moving elements. This allows for greater speed to be achieved using less powerful motors and also mitigates force translated to the frame due to changes in momentum of the harvesting elements.

A Galil DMC motion controller was used to coordinate the motion of both motors, and provide GPIO allowing interaction with limit switches and actuation of external systems such as the end-effector. The motion controller allows several motion profiles to be written, and triggered from an external system

such as the system PC. The motion controller hardware is able to accurately monitor the speed and acceleration of the motors in order to provide closed-loop following of the motion profiles. These motion profiles can be made to accept runtime arguments, allowing the external system to control speed-limits and target positions dynamically. This approach allows low-level control of the motors, such as pulse-train generation, to be managed independently of the main system, allowing for accurate time-synchronisation between each motor, and robust error handling.

The end effector is a simple scissor-type mechanism constructed from laser-cut sheet steel elements. The device, shown in Figure 4.8, is powered using a 12V solenoid. The inner faces of the scissor mechanism are lined with compliant 3D printed pads that are designed to deform around a target asparagus spear. Underneath these pads are mounted a pair of steel blades, the edges of which overlap slightly when the end effector is in a closed position. During operation, the end effector is maneuvered to intercept a target spear at the base with the mechanism open. When the target spear is in the cutting-zone the mechanism is actuated and held in the closed position, resulting in the target spear being cut near the base and gripped by the end-effector. The end effector is then positioned over the closest deposit box and the spear is released.

In order to coordinate RHS-1 to harvest spears, the perception pipeline developed in Chapter 3 is be utilised to develop a working vision system capable of both tracking target spears frame-to-frame and scheduling targets. The following section presents the integration of the novel perception pipeline into AHR-1.

4.3 Software Framework

This section discusses the various software elements developed to integrate the perception pipeline into AHR-1. The previous chapter described the the-

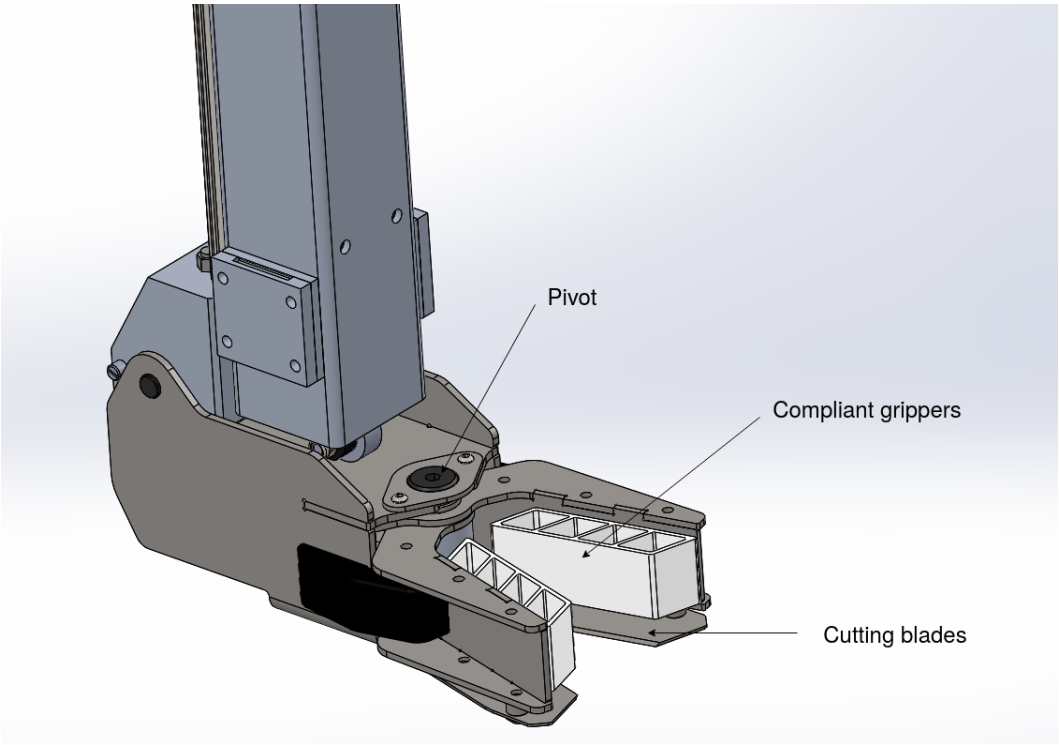


Figure 4.8: AHR-1's end-effector. The device is actuated with a 12V solenoid and simultaneously grips and cuts target asparagus spears.

oretical basis for the pipeline, however several specifics regarding the implementation of the methods were omitted. These specifics are discussed here. Additional subsystems are also described such as the data acquisition pipeline, state tracking, and asparagus tracking modules.

The perception pipeline, as described by Chapter 3, provides a method for predicting the base/cutting location of asparagus spears in a pointcloud obtained from ToF imaging. The perception pipeline operates on time-synchronised pairs of ToF and RGB images to achieve this task. Operation of the vision system in a real-time environment requires a robust data acquisition pipeline to provide a stream of time-synchronised image data. This synchronisation needed to be implemented in software as the Kinect camera does not support hardware triggering. Integrating the vision system into a physical asparagus harvester requires a number of additional problems to be solved. Namely these problems are:

1. Spear permanence and tracking.

2. Frame coordination.

3. Target scheduling.

The following subsections present the data acquisition pipeline and development of an FRCNN model utilised for Neural Network Filtering, as described in Section 3.5.1. Additionally the problems of spear permanence and tracking, frame coordination, and target scheduling are discussed.

4.3.1 Data Acquisition

AHR-1 utilises three different sensors, namely, a ToF camera, RGB camera, and rotary encoders at the wheels. Each of these devices communicate with the main system PC using a USB serial connection.

The Kinect camera is connected to the system PC via USB 3.0. Libfreenect2 [104], an open source, reverse engineered driver, is utilised for communication between the camera hardware and USB controllers. Raw data from the Kinect camera is communicated via a USB serial connection and deconstructed by the freenect2 driver to provide many different data streams, including a depth-map (512x424), RGB image (1920x1080) and IR image (512x424) stream. The depth-map data can be converted into pointcloud data by applying a transformation matrix, which accounts for the camera geometry and intrinsic parameters. This matrix is included with the freenect2 driver. The purely geometric, XYZ pointcloud generated from this transformation can be enhanced by registering each pointcloud with the corresponding RGB image, resulting in a XYZRGB pointcloud. Registration of the RGB image with the pointcloud data requires a further image-space transformation to be applied, the matrix for this is also hard-coded in the freenect2 driver. The Kinect camera, communicating through the freenect2 driver can deliver XYZRGB pointclouds at a rate of approximately 24 frames per second.

The Basler ACE RGB camera also connects to the main system PC via USB 3.0. Basler's Pylon 5 API is used to interface with the camera. The Pylon 5 API allows on-the-fly configuration of camera parameters such as frame rate,

exposure time, and white balance settings, as well as software methods for capturing image frames. The resolution of the Basler ACE camera utilised by AHR-1 is 1920x1080 pixels. AHR-1 operates the Basler ACE camera with a fixed exposure time and frame rate. The operating frame rate is 50 frames per second.

Communication with the wheel encoders is achieved via the US Digital QSB-D Quadrature to USB converters. Each wheel of AHR-1 is connected with an encoder, and QSB-D device. These devices connect with the main system PC as a serial device via USB 2.0. UDEV rules were utilised to create symbolic links to each device based on serial number. This allowed each device to be addressed by ID, making the system robust to changes in port discovery order. The QSB-D device is designed to communicate via ASCI strings sent over the serial interface. The device has an extensive list of commands for configuration, as well as commands for reading, and resetting the encoder count. AHR-1 utilised the device in a modulo-n count mode, with a count resolution of one count per quadrature cycle. The QSB-D device is capable of communicating at a rate of approximately 80 messages per second.

Time synchronisation of data is a requirement of the vision system. Due to the consumer grade nature of the hardware utilised by AHR-1, specifically the Kinect camera, hardware triggers are not supported. This means that time synchronisation must be achieved in software. Asynchronous data streams from each sensor are established and fed into a first-in-first-out buffer of a fixed size. In addition to storing the raw data, each buffer entry also includes a system timestamp marking the time of acquisition for each pointcloud, image, or encoder count. The Kinect camera reports at the slowest rate of all the sensors in the system. This means that for every pointcloud image the system receives there are multiple images and encoder readings. The system generates a stream of time synchronised data, or a "data frame" by matching RGB images and encoder counts from their respective buffers to pointclouds from

the Kinect buffer based on the proximity of the timestamps. The process for achieving this matching is as follows:

1. Grab the data from the end of the Kinect buffer. This data will be the oldest in the buffer and consists of both a pointcloud($\mathbf{P}$) and timestamp($T_{\mathbf{P}}$).
2. Trim all of the other buffers by removing all entries where the corresponding timestamp is older than $T_{\mathbf{P}}$. This is achieved by iteratively selecting the entry and the end of each buffer and removing the entry if the timestamp is older than $T_{\mathbf{P}}$.
3. For each of the other buffers search for an entry where the timestamp is within some threshold $t_{\max}$ of $T_{\mathbf{P}}$. If all of the other buffers has an acceptable entry the corresponding data from each of these entries is packaged together with the pointcloud $\mathbf{P}$ into a data frame.
4. The data frame is then passed on for processing and the entry from the end of the Kinect buffer is removed.
5. This process is repeated as long as there are entries in the Kinect buffer, resulting in a stream of data frames consisting of a set of pseudo-time-synchronised data.

The software synchronisation method described above imposes considerable time delays (approximately 18ms) on the acquisition pipeline. This reduces the effective frame rate of the incoming data-frames, introducing a significant bottleneck on the overall vision system.

A key component of the proposed perception pipeline is the Neural Network Filter, which utilises a FRCNN model to initially filter incomming pointclouds. The following subsection discussed the implementation of FRCNN utilised for AHR-1, and outlines the development and evaluation of an example network for clarity.

4.3.2 FRCNN Model Implementation

The vision system utilises Tensorflow as a machine learning framework to implement the FRCNN model. This implementation is contained within Tensorflow's object detection API [105, 106] and constituent "model zoo". The specific model utilised by the vision system is *fasterrcnninceptionv2coco*. This model consists of "Inception v2" convolutional layers [107] which were pretrained on Microsoft's COCO dataset [108]. Training, in this work, was limited in scope to the fully connected layers of the model. This, so called "transfer learning" approach, allows the generalised feature detectors learned through training on an extensive dataset like COCO to be retained in the lower levels of the network, greatly reducing the required number of training images. It is arguable that training the FRCNN model from scratch could result in more appropriate, task specific feature detectors arising. However, generating a dataset which is both large, and general enough for the task is not practical. The various labelled datasets produced throughout this work have been made publicly available at `https://github.com/MPeebles/AsparagusDatasets.git`.

The following subsection details the data collection, training and evaluation of a specific FRCNN model (*frcnn_usaNet1*) developed for use by AHR-1 during field trials in Los Banos, California. Details of the field trials can be found in Chapter 5.

4.3.2.1 Example FRCNN Model

The dataset for *frcnn_usaNet1* consists of 500 images taken from three separate asparagus rows. The labeled images can be found in the *"usa_01"* folder of the aforementioned Github repository. These rows, shown in Figure 4.9, were adjacent to each other and located at one end of the asparagus field. This is problematic for generalisation, however, operational constraints of AHR-1 precluded access to other rows. These limitations are discussed in Section 5.1.1.

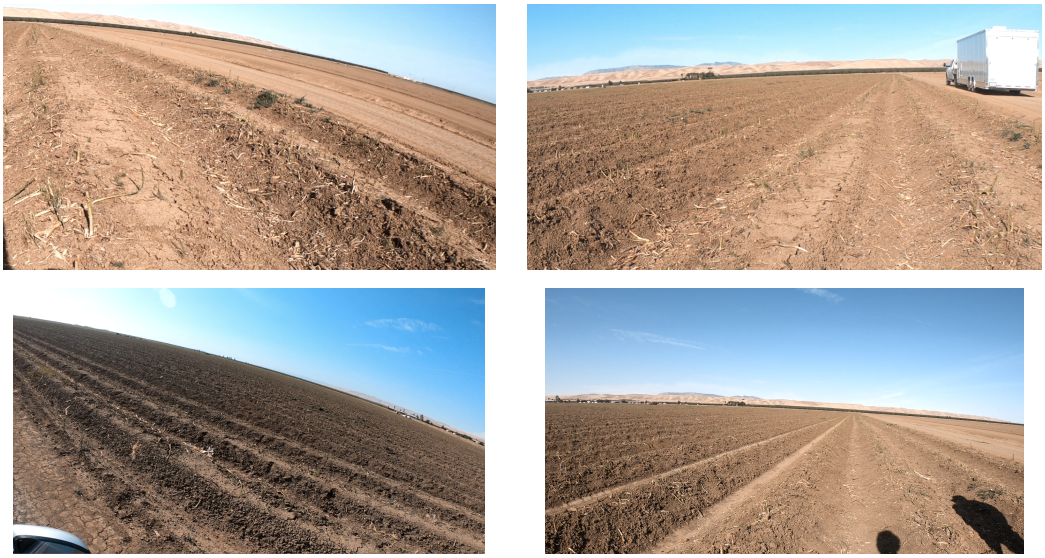


Figure 4.9: Images of test rows from which the training data for *frcnn_usaNet1* was collected. These rows were utilised for field trials of AHR-1 (see Chapter 5).

The training-set, test-set and validation-sets contained 447, 76, and 74 images respectively. These images were captured from the AHR-1 platform, at a height of approximately 1m from the ground, oriented at an angle of 45° to the soil plane and had a resolution of 576×360.

The FRCNN model was configured for a single class, dubbed "aspa", and a maximum number of detections per image of 50. The configuration file has been included in appendix A.2.

Consistency in labelling is important for achieving high performance in any CNN model. However, achieving a high level of consistency is challenging due to human error. Unstructured agricultural environments like asparagus fields are challenging to interpret, introducing a huge amount of subjectivity with regards to which features "count" as asparagus spears. This is due to the sporadic nature of asparagus, resulting in a large variety of spear sizes and morphologies being present in the scene. These factors mean that a single viewpoint, as presented in an image, can be ambiguous.

The dataset for *frcnn_usaNet1* was labelled by a single person using the following rule-set in an attempt to achieve the most consistent labelling possible:

1. All elements of the image which contain spear heads, or bracts (triangular patterns on the asparagus spear) should be labelled as asparagus spears. This means that fallen spears, which result from culling during previous harvesting runs should be counted.
2. Elements which are particularly small should not be labelled. This ensures that the resolution of labelled elements is large enough to reasonably contain enough information to describe asparagus spear features, helping the model to generalise. Determining which elements are considered "small" is a subjective task, achieved heuristically.
3. Each element should be labelled independently of all other elements. This means significant overlap between labels can exist when spears are clustered close together.
4. Only the visible features of each element should be labelled. This means that no inference should be made as to the structure of occluded elements.

The images were labeled in PASCAL VOC format [109] using LabelImage [110]. Figure 4.10 shows a number of labelled images from the *frcnn_usaNet1* dataset.

The *frcnn_usaNet1* model was trained with Tensorflow [106] v1.13.1 using a Nvidia Geforce Titan-X pascal GPU. The training curve, seen in Figure 4.11 shows that the model achieved a plateau in the normalised total loss after approximately 35,000 iterations. This plateau indicates that the model achieved a local minima, implying sufficient training time was allowed. The training curve has been smoothed using a Savitzky-Golay filter for clarity [111].

The *frcee_usaNet1* model was evaluated by investigating the precision-recall characteristics, and calculating the corresponding maximum F1 score. This analysis was performed based on images in the test set and is presented in the following subsection.

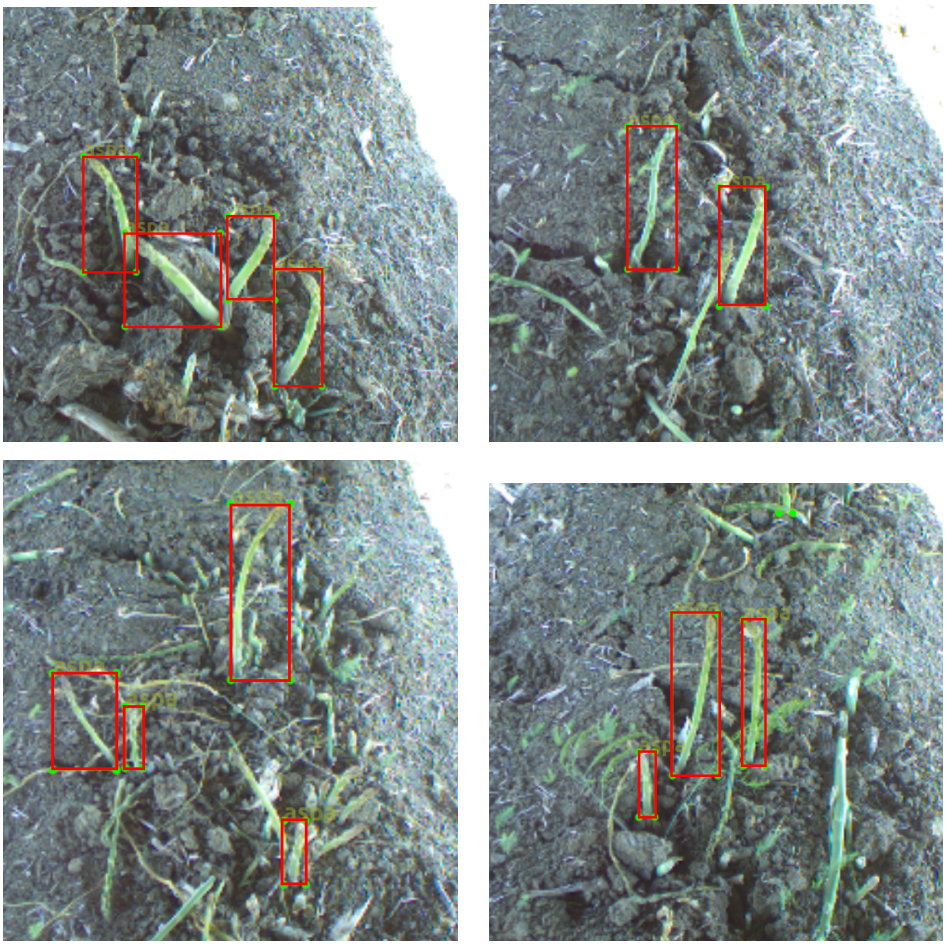


Figure 4.10: Example of various labelling decisions. These bounding boxes were defined manually utilising the rules set out in Section 4.3.2.1.

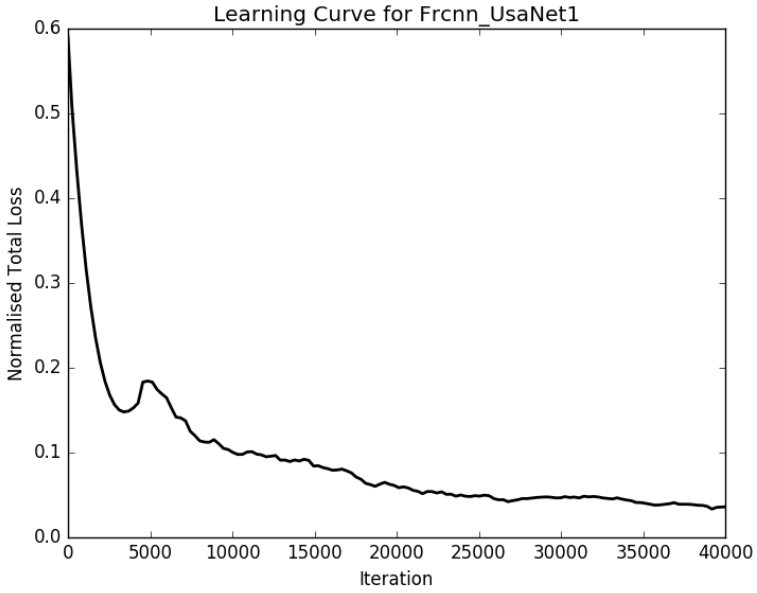


Figure 4.11: Training curve of the Frcnn_UsaNet1 model. This plot shows a plateau in the normalised total loss after approximately 35,000 iterations. The plateau indicates that the model has achieved a local minima.

4.3.2.2 Evaluation of *frcnn_usaNet1*

Precision and recall are fundamental metrics in pattern recognition and object detection. For a given object detector a series of predictions will result in a number of *true positive* (TP), *false positive* (FP), and *false negative* (FN) detections. The precision and recall of the detector can then be calculated as:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4.1}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4.2}$$

Precision provides a measure of the proportion of detections which are *true positives*. Precise detectors (detectors with high precision) are characterised by few false positive detections. Recall provides a measure of the proportion of all relevant objects which were detected. Detectors with high recall are characterised by few false negative detections.

The precision and recall of a detector are often inversely related. An object detector which selects every object in a scene will have high recall, because

it will not miss any target objects, but will have very low precision as many of the detections will be false. Likewise, a detector which does not provide many detections can have high precision, because less false predictions will be made, however, a correspondingly low recall as less of the target objects will be detected. Often the precision and recall characteristics of a given detector are decided by the confidence threshold, C_t. This threshold is used to determine the minimum confidence a given prediction must have before it is regarded as valid.

For low C_t values, models tend to exhibit lower precision and higher recall. Conversely, high C_t values correspond to high precision and low recall. The specific relationship between the precision and recall of a given model over a series of C_t values can be explored using a precision-recall plot.

To generate the precision-recall plot of a given model, the number of TP, FP and FN detections in the models test set was determined by considering the intersection over union (IOU) between the output of the model, and the corresponding labels. For a given bounding box, $\bar{A}$ and label, $\bar{B}$, IOU is defined as:

$$\text{IOU} = \left| \frac{\bar{A} \cup \bar{B}}{\bar{A} \cap \bar{B}} \right| \tag{4.3}$$

For all images in the test set the number of TP, FP, and FN detections were determined as follows:

1. For each bounding box in the image, $\bar{A}_\nu$, with a confidence value higher than C_t an IOU value was calculated for all k labels, $\bar{B}_\tau$ for $\tau \in \mathbb{N} \cap [1, k]$ in the image. These values were stored in a k dimensional list, $\bar{\mathbf{L}}_\nu$, of elements with the form (IOU, τ), where IOU is the IOU value calculated between the νth bounding box and τth label respectively. In order to enforce a minimum required overlap, any IOU value less than some threshold was set to zero. Each of these lists was then sorted in descending order based on the IOU value.

2. Each $\bar{A}_\nu$ then voted for their preferred label based on the first eligible element of $\bar{\mathbf{L}}_\nu$.

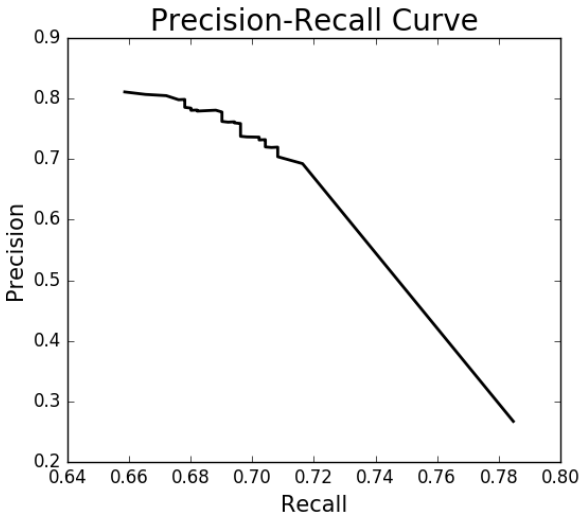


Figure 4.12: Precision recall plot for *frcnn_usaNet1*. Note that the axis limits have been moved from the origin for clarity.

3. Each label then selected the $\bar{A}_\nu$ with the highest IOU value from the set of bounding boxes that voted for it. All other $\bar{A}_\nu$ from this set were rejected and the IOU value for the associated label was set to zero.

4. Steps 2 and 3 were repeated until all remaining $\bar{A}_\nu$ had $\bar{\mathbf{L}}_\nu$ consisting only of elements with IOU values of 0, or all $\bar{A}_\nu$ had been selected by some label.

After this process the number of TP, FP and FN detections were determined as follows:

- All $\bar{A}_\nu$ which were associated with a label were considered TP
- All $\bar{A}_\nu$ which were not associated with a label were considered FP
- All labels which were not associated with any $\bar{A}_\nu$ were considered FN

Applying this procedure to all images in the test set allows a total number of TP, FP and FN detections for a given C_t to be determined. Precision and recall can then be calculated based on equations 4.1 and 4.2 respectively. Plotting these values for a series of C_t values generates the precision-recall plot shown in Figure 4.12.

As expected, the precision-recall plot for *frcnn_usaNet1* shows an inverse relationship between precision and recall. The point on this plot where the

detector operates is controlled by the confidence threshold C_t. For this application both precision and recall are considered equally important. In such cases a useful measurement of accuracy is the F1 score, defined as:

$$\text{F1} = 2 \times \left(\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \right) \tag{4.4}$$

By iterating over all points in the precision-recall curve the maximum F1 score, and corresponding C_t value was found to be 0.73 and 0.968 respectively. This reveals that predictions made by *frcnn_usaNet1* are most accurate when a C_t of 0.968 is applied.

After detecting individual spears the vision system must be capable of identifying and tracking spears on a frame-by-frame basis. Spear permanence and tracking is discussed in the following subsection.

4.3.3 Spear Permanence and Tracking

The proposed vision pipeline does not retain knowledge of previously detected asparagus spears on a frame-by-frame basis. This is problematic because AHR-1 is designed to image the asparagus row on a continuous basis such that subsequent images will contain the same spears, resulting in multiple detections for each spear. It is important that the system can keep track of each detected spear on a frame-by-frame basis to allow unique spears to be identified. This is the problem of **spear permanence**.

Another problem is **spear tracking**. As the robot translates down the asparagus row each unique spear that is detected needs to be tracked in 3D space to enable coordination with RHS-1. This problem is especially of concern as AHR-1 has a blind zone between RHS-1 and the vision-zone, requiring some degree of dead reckoning to intercept target spears.

Solving the problem of spear permanence means finding a robust, unchanging feature description of each unique spear which can reliably be used to identify an individual spear in any frame; a kind of spear fingerprint. AHR-1 utilises the base point prediction provided by the perception pipeline as this

feature descriptor. The philosophy of this decision is that any detected spear which shares a base point with a previously detected spear must be a secondary detection. For clarity, a **primary detection** is defined as the first detection made by the vision system of a specific individual spear, while **secondary detections** are defined as all other detections of the same spear in subsequent data frames. The forward motion of AHR-1 during operation means that the relative translation of this base point needs to be accounted for. This is achieved by the frame coordination system, detailed in Section 4.3.4, using feedback from the wheel encoders. This is a relatively first-order approach, as further geometric comparisons could be considered, however due to time limitations higher-order solutions have not been investigated.

AHR-1 begins by passing the first available data frame through the vision system. This results in a base point prediction for each of the detected spears. AHR-1 utilises an array, namely the Asparagus Register (AR), to keep track of detected spears. Table 4.1 shows an example of the AR during operation.

Table 4.1: Example of a typical asparagus register (AR) during operation.

ID	Point List	Length List	Base Point	Harvestable	Confirmed	Attainable
1	[[x_1,y_1, z_1],..,[x_n,y_n,z_n]]	[L_1 ,.., L_n]	[x,y,z]	TRUE	TRUE	TRUE
2	[[x_1,y_1, z_1],..,[x_n,y_n,z_n]]	[L_1 ,.., L_n]	[x,y,z]	TRUE	TRUE	FALSE
3	[[x_1,y_1, z_1],..,[x_n,y_n,z_n]]	[L_1 ,.., L_n]	[x,y,z]	FALSE	TRUE	FALSE
4	[[x_1,y_1, z_1],..,[x_n,y_n,z_n]]	[L_1 ,.., L_n]	[x,y,z]	FALSE	FALSE	FALSE

Each entry to of the AR is a *struct* representing a unique spear on the asparagus bed. The *struct* contains the following information:

- **ID:** A Unique ID number
- **Point List:** An array of $[x, y, z]$ positions. The first position in this list is the primary detection that populated the spear in the AR and all other points are subsequent secondary detections of the same spear
- **Length List:** An array of spear lengths based on the pointcloud geometry of each spear. The index of the entry matches the index of the

positions in the **Point List**. i.e. the first entry pertains to the length of the spear calculated during the primary detection

- **Base Point** A single $[x, y, z]$ position calculated as the geometric mean of all points in the **Point List**. This point is considered as the actual base location of the spear
- **Harvestable:** A Boolean flag describing if the spear is of a harvestable length. This flag is set based on the mean length from the **Length List**
- **Confirmed:** A Boolean flag describing if the number of detections which define the spear, i.e. the number of entries in the **Point List** and **Length List** is larger than some threshold, T_R. This flag allows spurious entries generated based on sensor noise or one-off false detections from the vision system to be ignored
- **Attainable:** A Boolean flag describing if the spear is attainable by RHS-1. This flag is set based on the spear being both inside the workspace of RHS-1, and based on the target scheduling system (see Section 4.3.5)

Once the data frame is passed through the vision system the resulting base points for each detected spear are evaluated, and used to populate the AR. As mentioned above, primary detections and secondary detections are differentiated based on the detected base point of each spear. This is achieved by applying a distance threshold, d_T, between each prospective base point ($\mathbf{b}_p$) and the accepted **Base Point** entry ($\hat{\mathbf{b}}_P$) for each spear in the AR such that detections where:

$$\left\| \mathbf{b}_p - \hat{\mathbf{b}}_P \right\| > d_T \tag{4.5}$$

are considered primary detections, while all other detections are considered secondary. When a primary detection is made a new entry is added into the AR, while any secondary detections result in the detected base point and length being added into the corresponding spears **Point List** and **Length List** respectively. Since asparagus spears are stationary with respect to the

ground, the calculations in Equation 4.5 are carried out in the world coordinate frame (see Section 4.3.4).

The AR is also continuously cropped by removing spears with **Base Locations** beyond the reach of RHS-1 (i.e. when the spear is behind AHR-1). Such spears can no longer be harvested by AHR-1 and therefore do not need to be tracked.

Following the above procedure on a frame-by-frame basis results in an AR that contains a real-time collection of unique asparagus spears which are currently located in either the vision region, or dead-reckoning zone of the robot. From the AR, target spears can be selected, and the relevant positional information sent to RHS-1 to coordinate harvest.

4.3.4 Frame Coordination

AHR-1 has many subsystems, each of which operate in their own frame of reference. Additionally, a fixed reference frame for the *world* must be considered in order to track AHR-1's motion down an asparagus row. Target asparagus spears also exist in the *world* reference frame. The base plane predictions which the perception pipeline generates are made in the reference frame of the input pointcloud, coincident with some fixed point on the ToF camera. The end-effector of RHS-1 is also a reference frame of importance, allowing the coordination of the end-effector with target spears. Each of these reference frames are rigidly offset from each other based on the geometry of the frame, and by the rotational offsets of RHS-1's motors. Integrating the vision system into a functional harvester requires keeping track of each of these reference frames in a time-synchronised manner. This is the problem of *frame coordination*. Figure 4.13 shows the hierarchy of reference frames utilized by AHR-1.

The *world* frame is at the top of the hierarchy and represents the fixed, global reference frame of the ground/row on which AHR-1 operates. Target

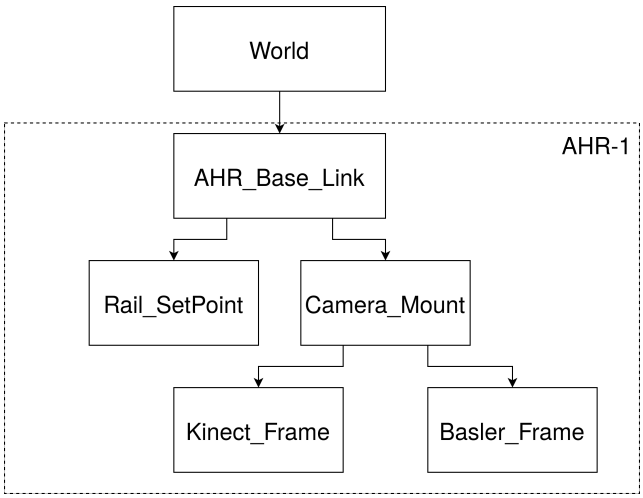


Figure 4.13: Reference frame hierarchy of AHR-1.

asparagus spears are located in this frame as they are fixed with respect to the ground.

The *AHR_Base_Link* frame is the origin of the robot in the world. AHR-1's origin is located at the home position of RHS-1's end effector, however this position is somewhat arbitrary. It is useful to position AHR-1's origin at the end-effector's home position because it eliminates the need for an additional "RHS" frame to represent the rigid body offset between the origin of RHS-1 and any other arbitrary point on AHR-1. The *AHR_Base_Link* frame is orientated similarly to the *world* frame, however the offset of *AHR_Base_Link* from the origin of the *world* frame is determined by both the rigid geometric offsets of the robots construction and the position which the harvester has travelled in the world, calculated based on feedback from the wheel encoders.

The *Rail_SetPoint* frame represents the desired pose of RHS-1's end effector. During operation RHS-1 is instructed to continuously move the end effector to meet this pose. The transformation between the *AHR_Base_Link* and the *Rail_SetPoint* frames is therefore determined by the desired end-effector position, relative to its home position at any point in time.

The *Camera_Mount* frame represents the mount on which both the Kinect and Basler cameras are connected. This frame is simply the result of a fixed rigid body offset from the *AHR_Base_Link*. This transformation involves both

a translation, and rotation and is determined during hand-eye calibration (see Section 4.5.3). Attached to the *Camera_Mount* frame are two additional frames that represent the origin of each camera. Data from the cameras is reported in their respective reference frame. The transformation between each of these camera frames and the *Camera_Mount* frame also represents a fixed rigid body transformation. This transformation is determined during calibration (see Section 4.5.1) and is used allow registration between each image.

4.3.5 Target Scheduling

RHS-1 is a relatively simple robotic system, designed to provide a proof of concept rather than operate at commercial rates. For this reason the harvesting throughput of RHS-1 is limited. The expected harvestable spears in a typical row are too numerous and densely clustered for RHS-1 to successfully harvest. AHR-1 therefore has to make decisions on which targeted spears are attainable by RHS-1 before coordinating the harvest. This problem is compounded by the continuous harvesting basis of AHR-1 because it means that targeted spears are continuously moving toward RHS-1 as the machine translates the asparagus row. The limitations of RHS-1 mean that the time needed for the end-effector to be positioned to intercept an oncoming targeted spear can be longer than the time it takes for that spear to pass beyond RHS-1, making harvest impossible. This imposes a point of no return for each targeted spear that is a function of the ground translation speed of AHR-1 and the position of the end-effector before initiating the move. This presents a problem of *target scheduling*, whereby AHR-1 must determine which spears to harvest from a continuous stream of targeted spears. Ideally, this decision should be made to optimise the number of harvested spears by considering future moves of RHS-1 and selecting an efficient set of harvest targets.

AHR-1 adopts a simple target scheduling strategy to solve this problem. When AHR-1 is ready to harvest a spear it begins by selecting the first qualifying spear with the lowest ID number from the AR as the new target. The

spear with the lowest ID number corresponds to the oldest spear in the AR as ID numbers are sequential. For a spear to qualify for targeting both its "harvestable" and "attainable" flags must be "True". AHR-1 then calculates an exclusion zone based on its current ground speed, and the cycle time of RHS-1 determined by:

$$\delta_{RHS} = \begin{bmatrix} (V_g T_{cy}) d_f \\ W \\ H \end{bmatrix} \tag{4.6}$$

where $\delta_{RHS} \in \mathbb{R}^3$ is the exclusion zone defined in the *Base_Link* coordinate frame, V_g is AHR-1's instantaneous ground speed, T_{cy} is the cycle time of RHS-1, and, W and H are the width and height of AHR-1's operating space respectively. d_f is an empirically determined safety factor which is used to extend the exclusion zone in order to ensure smooth operation and accommodate unforeseen delays. All non targeted spears that are within the exclusion zone at the time δ_{RHS} is calculated have their "attainable" flags set to false. The result is that all spears in the AR that will not be targetable by RHS-1 after the harvesting cycle is complete will be rejected from future targeting decisions. There are two reasons why a spear can be considered untargetable by RHS-1:

1. The spear is outside the workspace of RHS-1.
2. The spear is within RHS-1's workspace, however by the time the end effector moves into position to coordinate harvest the spear will no longer be in RHS-1's workspace.

RHS-1 is designed to harvest asparagus spears by intercepting them from the direction in which the robot is travelling. The system utilises two thresholds, namely the "down threshold", d_d and the "grab threshold", d_g to control when RHS-1 should initiate the intercepting downwards z-move, and grab functions respectively. Figure 4.14 illustrates this harvesting routine in more detail.

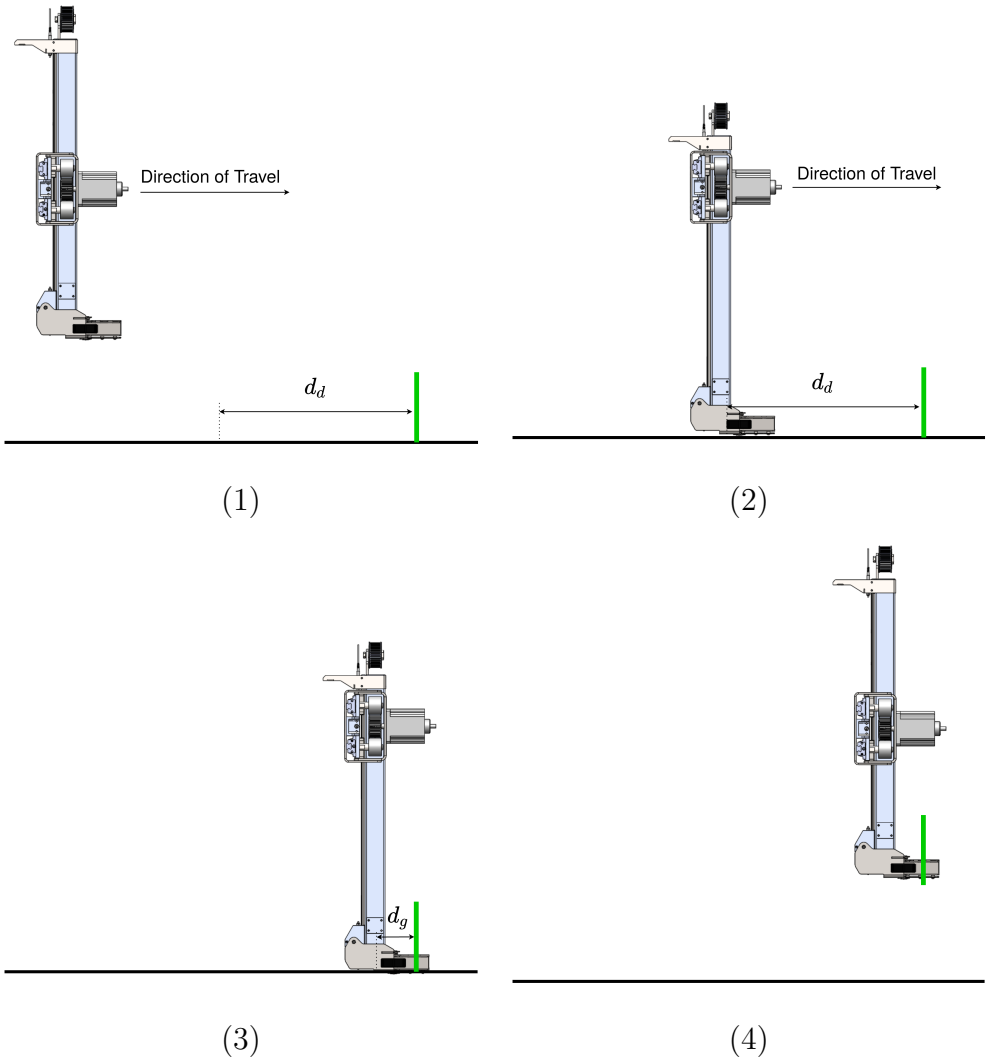


Figure 4.14: Demonstration of AHR-1's harvesting routine. (1) RHS-1 is in the "up" position as AHR-1 approaches a target spear. (2) when the end-effector is d_d away from the target spear RHS-1 moves to the "down" position ready to intercept the spear. (3) when the end-effector is d_g away from the target spear it grabs the spear, cutting near the base. (4) the end effector moves to the "up" position, carrying the spear away to be deposited.

All of the aforementioned software implementations are deployed in a ROS framework. The following section outlines the structure of the ROS network and provides a detailed description of the various nodes and services developed.

4.4 Robot Operating System (ROS) Network for AHR-1

AHR-1 utilises ROS [32] on the main system PC to direct program flow and to coordinate sensor data. A description of ROS can be found in Appendix A.3. The ROS network for AHR-1 is shown in Figure 4.15. The figure shows all of the ROS nodes utilised by AHR-1 and demonstrates the corresponding data-types output by each node. Additionally, the figure shows how data flows through the network, from the input devices to RHS-1.

The following subsections provide a description of each node in the ROS network.

4.4.1 Input Nodes

Kinect V2 Node, *Basler Node*, and *QSB Node* are the input nodes to the ROS network. These nodes directly interface with their respective hardware via the appropriate drivers/APIs and package the data into ROS messages. These nodes publish their data at approximate frame rates of 20, 50 and 80 respectively.

4.4.2 Frame Sync

The frame sync node subscribes to the topics published by the input nodes and implements the aforementioned frame synchronisation. When relevant data is available this node packages a pointcloud, RGB image and a count value for each encoder into a synchronised data frame.

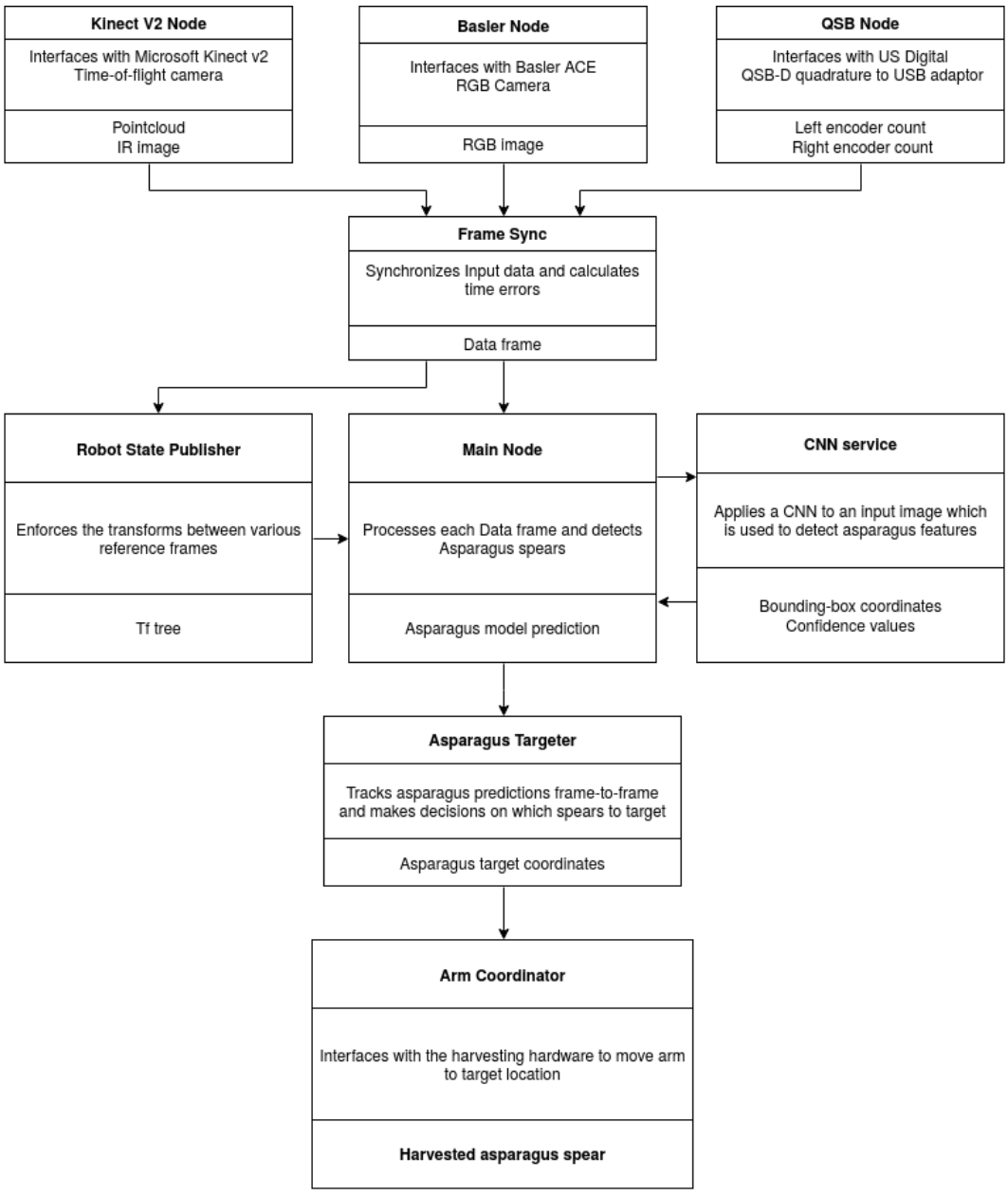


Figure 4.15: ROS network of the robotic asparagus harvester. Each box describes the function of a node or service, and describes the type of output generated. The arrows describe the flow of information through the network.

4.4.3 Robot State Publisher

ROS includes a robust method for managing various reference frames, namely the "tf" package. This package allows many different reference frames to be registered, and the corresponding rigid body transforms between the frames to be defined, and updated in real-time. This information is stored by "tf" in a structure known as a "tf-tree". The "tf" package also provides methods for transforming points (and pointclouds) between the coordinate frames.

Robot State Publisher is the node which AHR-1 uses to maintain the "tf-tree". To achieve this the Robot State Publisher subscribes to the frame sync node and utilises the encoder counts to update the transform between the world and rail frames. Robot State Publisher also subscribes to the *Asparagus Targeter* node in order to update the transform between the *rail-frame* and the *rail-set-point* frame.

4.4.4 Main Node

The main node utilises the synchronised data frames provided by the frame sync node to implement the vision system described in Chapter 3. This node also publishers the relevant pointclouds and markers necessary for visualisation.

4.4.5 CNN Service

The CNN service is a ROS service server which is called by the main node in order to implement the vision system. The service takes a single image and returns the bounding boxes determined by the FRCNN model.

4.4.6 Asparagus Targeter

The Asparagus Targeter node is used to maintain the AR and perform target scheduling and selection. The node receives information about individual spear detections based on the vision system from the main node. It uses this

information to maintain the AR and make targeting decisions based on the aforementioned procedure.

4.4.7 Arm Coordinator

The Arm Coordinator node implements a state machine which is used to direct RHS-1 to harvest targeted spears and deposit them in the nearest collection box. The node subscribes to the Asparagus Targeter for information on the targeted spear position.

4.5 Calibration

AHR-1 has three main subsystems that require calibration, namely the camera-pair, encoders and RHS-1. This section provides reasoning for these subsystem's calibration requirements, as well as an overview of the methodology used.

4.5.1 Camera-Pair Calibration

AHR-1 utilises both a Kinect camera, and a Basler ACE camera. It is intended that these cameras be mounted with fixed offsets from AHR-1's *"Base_Link"* frame with their optical axes aligned. The intrinsic parameters of each camera also differ significantly. These factors mean that the images produced by each camera (RGB and IR/depth-image) are taken from different perspectives and have significantly different fields-of-view and distortion characteristics. As such, registration between these two images is a requirement of the vision system.

Image registration was performed between the depth-image of the Kinect camera, and the RGB image from the Basler camera using a standard perspective transformation. This transform is used to map points from one plane, into another and can be mathematically expressed as:

$$\lambda_p \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{H_p} \begin{bmatrix} \hat{u} \\ \hat{v} \\ 1 \end{bmatrix} \tag{4.7}$$

where, λ_p is a constant scale factor, and $\mathbf{H_p}$ is a 3×3 homography matrix which maps $[\hat{u}, \hat{v}]$ points in the Basler's RGB image to $[u, v]$ points in the Kinect camera's depth-image. Applying this transform to all points in the RGB image results in a warped RGB image in the same coordinate space as the depth-image, thus resulting in the required registration.

The homography matrix $\mathbf{H_p}$ was estimated empirically by minimising the reprojection error from a set of n_k corresponding points, $\mathbf{\Omega} = \begin{bmatrix} \begin{bmatrix} \hat{u_\kappa} & \hat{v_\kappa} \end{bmatrix}^{\mathrm{T}} & \begin{bmatrix} u_\kappa & v_\kappa \end{bmatrix}^{\mathrm{T}} \end{bmatrix}$, where $\mathbf{\Omega} \in (\mathbb{N}^2, \mathbb{N}^2)^{n_k}$, and $\kappa \in \mathbb{N} \cap [1, n_k]$.

For a perspective transform of the form:

$$\mathbf{H_p} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \tag{4.8}$$

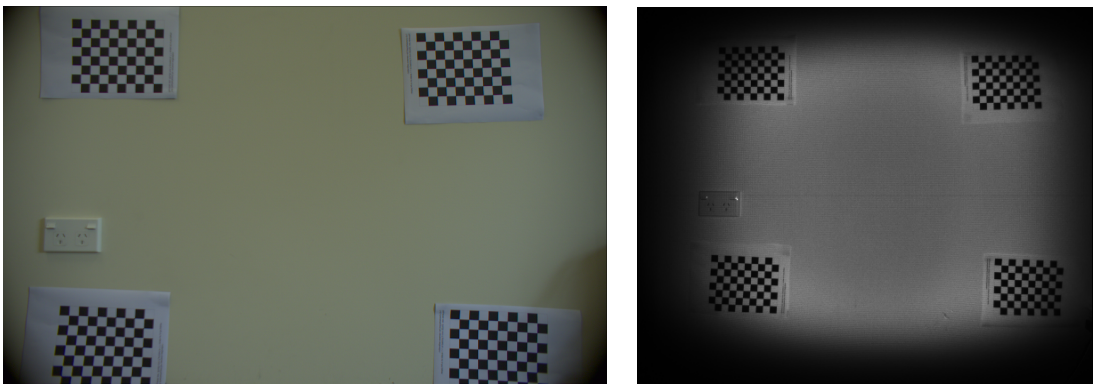
the reprojection error can be calculated by minimising the quantity:

$$\sum_i \left(u_\kappa - \frac{h_{11}\hat{u}_\kappa + h_{12}\hat{v}_\kappa + h_{13}}{h_{31}\hat{u}_\kappa + h_{32}v_\kappa + h_{33}} \right)^2 + \left(v_\kappa - \frac{h_{21}\hat{u}_\kappa + h_{22}\hat{v}_\kappa + h_{23}}{h_{31}\hat{u}_\kappa + h_{32}v_\kappa + h_{33}} \right)^2 \tag{4.9}$$

This was achieved with a least-squares regression using tools provided by OpenCV.

The set of corresponding points , $\mathbf{\Omega}$, were determined by imaging a series of standard calibration chessboard patterns with each camera connected to the mounting bracket. Figure 4.16 shows the captured images.

OpenCV tools were used to determine the image-space locations of the "black square" corners from each of the chessboards in each image. The correspondence between each detected point was determined geometrically. The resulting set of corresponding points was used as $\mathbf{\Omega}$ to estimate $\mathbf{H_p}$.



(a) **(b)**

Figure 4.16: Images used to calibrate the camera pair. **(a)** is an RGB image (1920 × 1080) from the Basler camera. **(b)** is an IR image (512 × 424) from the Kinect camera.

4.5.2 Encoder Calibration

AHR-1's encoders are used to track the forwards translation of the robot during operation. AHR-1 uses a constant to map encoder counts of each encoder to the respective linear distance traveled by each wheel. This relationship can be expressed as:

$$\begin{bmatrix} d_{\text{Left}} \\ d_{\text{Right}} \end{bmatrix} = \begin{bmatrix} \lambda_{\text{Left}} & 0 \\ 0 & \lambda_{\text{Right}} \end{bmatrix} \begin{bmatrix} E_{\text{Left}} \\ E_{\text{Right}} \end{bmatrix} \tag{4.10}$$

where d_{Left} and d_{Right} denote distance travelled in meters of the left, and right wheels respectively, λ is a calibration constant and E denotes the encoders count.

Encoder calibration constants (λ_{Left} and λ_{Right}) were determined empirically by manually rolling AHR-1 forward a number of different known distances and recording the corresponding E_{Left} and E_{Right} values; calculations were made based on simple rearrangement of equation 4.10. This procedure was repeated a number of times and the mean values of λ_{Left} and λ_{Right} were used.

4.5.3 RHS Hand-Eye Calibration

The *base_link* frame of AHR-1 is located at the home position of RHS-1's end effector. The *camera_mount* frame, on which both the Basler and Kinect cameras are mounted, is fixed at a rigid offset from the *base_link*. The pointcloud output from the Kinect camera is calibrated internally such that its local coordinate system the same scale as other world-based reference frames. That is, the distance d_{kinect} between two points in the pointcloud is the same as the distance d_{world} between the same two points in the world. Therefore, the transform between the *camera_mount* frame and AHR-1's *base_link* is simply a rigid body transformation. This transformation can be described as a 3D Affine transform between a point, $\mathbf{p}_{\text{cam}}^{\text{xyz}} = \begin{bmatrix} x_{\text{cam}} & y_{\text{cam}} & z_{\text{cam}} \end{bmatrix}^{\text{T}}$ in the *camera_mount* frame and a point $\mathbf{p}_{\text{base}}^{\text{xyz}} = \begin{bmatrix} x_{\text{base}} & y_{\text{base}} & z_{\text{base}} \end{bmatrix}^{\text{T}}$ in the *base_link* frame. This transform is represented mathematically as:

$$\mathbf{p}_{\text{base}}^{\text{xyz}} = \mathbf{A}_{\text{HE}}\, \mathbf{p}_{\text{cam}}^{\text{xyz}} \tag{4.11}$$

where $\mathbf{A}_{\text{HE}}$ is a 4×4 3D Affine transformation matrix of the form:

$$\mathbf{A}_{\text{HE}} = \begin{bmatrix} \mathbf{R}_{\text{HE}} & \mathbf{T}_{\text{HE}} \\ 0 & 1 \end{bmatrix} \tag{4.12}$$

In 4.12 $\mathbf{R}_{\text{HE}}$ is a 3×3 rotation matrix representing a combined rotation about all 3 principal axes of the camera_mount reference frame. This can be expressed as:

$$\mathbf{R}_{\text{HE}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_{\text{HE}} & -\sin\theta_{\text{HE}} \\ 0 & \sin\theta_{\text{HE}} & \cos\theta_{\text{HE}} \end{bmatrix} \begin{bmatrix} \cos\beta_{\text{HE}} & 0 & \sin\beta_{\text{HE}} \\ 0 & 1 & 0 \\ -\sin\beta_{\text{HE}} & 0 & \cos\beta_{\text{HE}} \end{bmatrix} \begin{bmatrix} \cos\gamma_{\text{HE}} & -\sin\gamma_{\text{HE}} & 0 \\ \sin\gamma_{\text{HE}} & \cos\gamma_{\text{HE}} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.13}$$

where θ_{HE}, β_{HE}, and γ_{HE} represent the angle of rotation about the three principal x, y, and z, axes, and $\mathbf{T}_{\text{HE}} \in \mathbb{R}^3$ is a column vector representing the translation between frame origins.

The 3D Affine transformation matrix, $\mathbf{A}_{\text{HE}}$, was estimated empirically based on least-squares regression of a set of corresponding points, $\mathbf{\Omega}_{\text{HE}}$, similarly to the camera-pair calibration described in the previous section. Each corresponding point was collected using the following procedure:

1. A wooden dowel is placed upright in the vision space of AHR-1 representing a model asparagus spear. The Kinect camera is then used to capture a pointcloud of the scene.
2. A point from the center of the circular top surface of the dowel is manually selected from the pointcloud and is transformed into the *camera_mount* frame.
3. Since the rail axis of RHS-1 is perpendicular to the forward direction of the robot the corresponding point from the *base_link* frame is determined based on a combination of forward distance based on wheel encoders, $\left[\, d_{\text{Left}} \;\; d_{\text{Right}} \,\right]^{\text{T}} \in \mathbb{R}^2$, and the 2 axis displacement of RHS-1 end effector, $\mathbf{d}_{\text{RHS}} \in \mathbb{R}^2$.
4. Firstly, the position of the end-effector, $\mathbf{d}_{\text{RHS}}$, is determined by moving RHS-1's end effector on both axes to a position that it is roughly inline with with the top of the wooden dowel. The end-effector remains in the open position during this time.
5. AHR-1 is then manually rolled forward until the dowel is inside the grippers of the end-effector. The average forward displacement based on both d_{Left} and d_{Right} is then calculated and stored as d_{Enc}.
6. Fine adjustments to both RHS-1's position, and the forward displacement are then made to center the dowel within the end-effector.
7. The corresponding point in the *base_link* frame is then determined as:

$$\mathbf{p}_{\text{base}}^{\text{xyz}} = \begin{bmatrix} d_{\text{Enc}} \\ \mathbf{d}_{\text{RHS}} \end{bmatrix} \tag{4.14}$$

AHR-1 uses a total of 10 corresponding points, generated from dowels with of varying heights to generate this transform.

4.6 Summary

This chapter presented a novel concept for a selective asparagus harvesting robot. The proposed concept was developed into a working proof-of-concept machine, namely AHR-1.

Additionally, this chapter discussed the implementation of the novel perception pipeline, presented in Chapter 3, into a stand-alone vision system. This integrated system solves the problems of spear permanence and tracking, frame coordination, and target scheduling, resulting in a robust system for detecting, localising, and coordinating the harvest of target asparagus spears.

The underlying ROS network was also presented in this chapter. These components combine to form a working robotic asparagus harvester. In the next chapter, a series of field trials, and subsequent evaluation of the proposed vision system are presented.

Chapter 5

Field Trials and Evaluation

Chapters 3 and 4 have presented a novel perception pipeline and discussed its integration into a complete vision system for the detection, localisation and targeting of green asparagus spears in a commercial setting. The integration of this proposed vision system into a working proof-of-concept harvester, AHR-1, is also discussed. This chapter begins by presenting field trials of of AHR-1 conducted in Los Banos, California. Analysis of AHR-1's performance during these trials are then used to explore the limits of the vision system and robotic hardware. This analysis results in various limitations being identified leading to the development of a second harvester prototype, AHR-2.

This chapter also offers a critique of existing evaluation methods found in the literature and proposes a novel evaluation method by which harvesting and detection performance can be objectively quantified. This evaluation method is applied to AHR-2 based on data collected from field trials in New Zealand.

5.1 Field Trials in California, USA

AHR-1 underwent a significant amount of lab trials prior to field testing, during which a basic system calibration and parameterisation was established. During this testing, real asparagus spears were planted in a sandbox apparatus similar to the one utilised for the evaluation of Hyun's method, outlined in Section 3.3.3. AHR-1 was maneuvered over this sandbox by hand, at speeds well below

the desired operating speed of 0.3m/s and RHS-1 was reliably able to harvest the planted spears.

The first field trials of AHR-1 were conducted during early May 2019. These field trials were conducted over a two week period on three different asparagus rows from a single farm near Los Banos, California (A-Bar Ag Enterprises). The goal of these field trials was to tune AHR-1's vision system, and investigate its performance in real-world conditions. This field trial was conducted several months before the New Zealand 2019 season. As such, these field trials were the first time AHR-1 was tested outside of a lab environment.

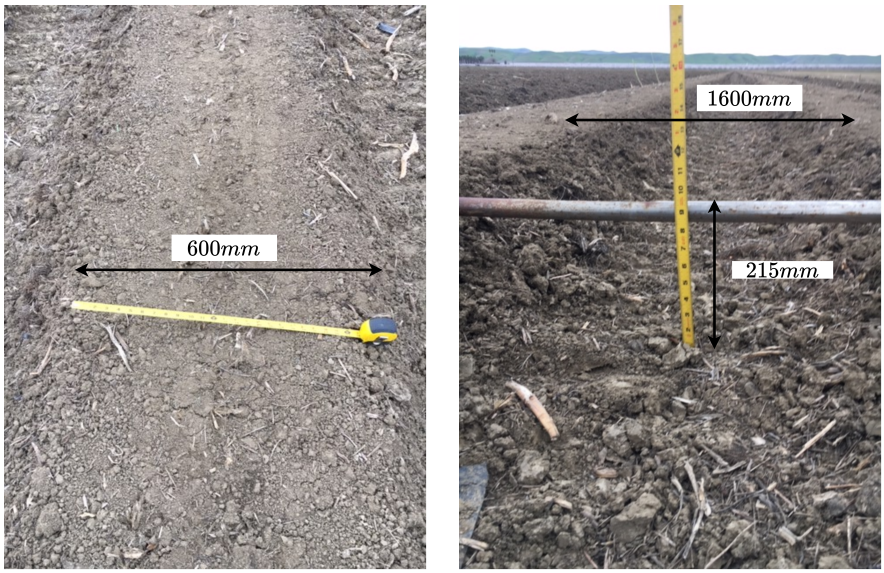
The US growers utilised mounded asparagus rows in contrast to the typical flat growing systems of New Zealand, for which AHR-1 was designed. Minor adjustments to both AHR-1 and the testing rows were required to ensure compatibility.

Since AHR-1's vision system was designed to operate on flat soil planes, the tops of each of the three testing rows were rolled flat, resulting in a pseudo trapezoidal geometry. This ensured that the immediate soil plane surrounding the asparagus spears was flat to allow the vision system to operate. Figure 5.1 shows the typical geometry of these rows after flattening. The following section describes the preparation of AHR-1 for operating on these trapezoidal rows.

5.1.1 Preparation of AHR-1

To accommodate the Californian asparagus rows a set of spacers were developed that raise the distance between the wheel hub, and body of the AHR-1. The spacers, shown in Figure 5.2, coupled with the adjustable tow-ball height provided by the tractor had the effect of raising the body of the machine close to the design height when operated on the trapezoidal rows.

These spacers were essential due to complications with the trapezoidal row geometry. Without spacers, the wheels of AHR-1 would make contact with the ground significantly lower than the height of the soil plane. This would result



(a) (b)

Figure 5.1: Row geometry of the trapezoidal test rows in Los Banos, California. (a) shows the top of a test row, measuring 600mm across. (b) shows the height of each flat section to be raised a distance of approximately 215mm, with 1600mm centre to centre spacing.

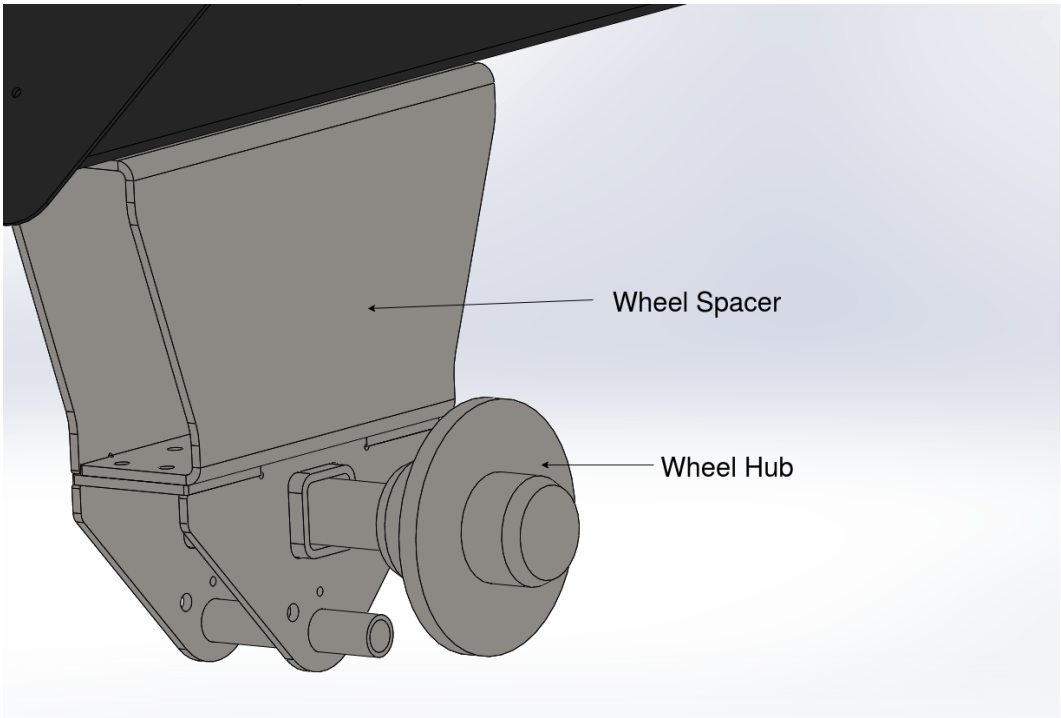


Figure 5.2: Wheel spacer fitted to AHR-1 in order to allow operation on trapezoidal Californian asparagus beds.

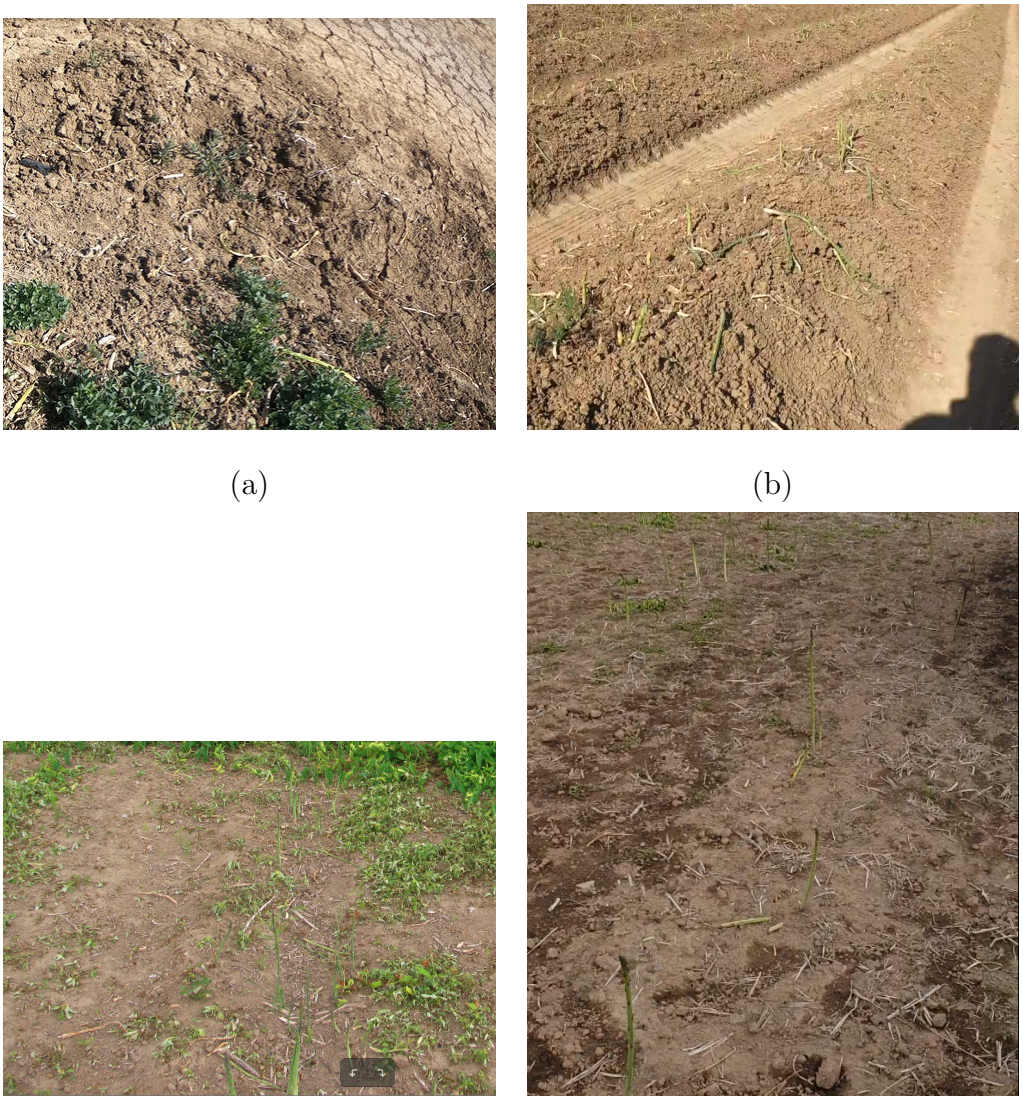
in the entire body of the machine being much closer to the soil plane than it was designed for. The problems with this are twofold. Firstly, the work-space of RHS-1 is relatively limited, meaning that raising the soil plane into this work-space would result in RHS-1 not having enough clearance between the soil plane and the end-effector at the upmost position, hindering operation. Secondly, raising the soil plane would have the effect of reducing the distance between the camera and the targeted section of the asparagus row. This would greatly reduce the size of the imaged section due to the camera's fixed field-of-view.

AHR-1 was transported from New Zealand to Los Banos via air freight. AHR-1 had to be disassembled due to volume restrictions imposed during transport, invalidating the previously established system calibration; the calibration process described in Chapter 4 was repeated in Los Banos after the machine was reassembled.

The dry Californian climate meant that the soil conditions were notably different than New Zealand farms. In particular, the typical size of dirt clumps were significantly larger, making the soil appear much more granular than the sandy loam for which AHR-1 was designed. Additionally, the weed species observed on the Californian farm differed from those observed in New Zealand. Figure 5.3 visually demonstrates the differences between the Californian and New Zealand farms. These differences in the appearance, and physical make-up resulted in previously trained FRCNN models achieving poor performance on the Californian asparagus rows. This necessitated the development of a field specific FRCNN model. This model was named *frcnn_usaNet1* and the specifics of its development are described in Section 4.3.2.1.

5.1.2 On Field Tuning

The performance of the perception pipeline is largely affected by the various input parameters and thresholds required by its constituent algorithms. A trial-and-error process of on-field tuning was performed in order to determine



(a) (b)

(c) (d)

Figure 5.3: Images demonstrating the visual difference between the appearance of New Zealand and Californian asparagus rows. Images (a) and (b) were taken from a Californian farm while (c) and (d) were taken from a New Zealand farm.

a working configuration of these parameters. It should be noted that although a considerable effort was made to optimise this configuration, time, cost and practicality prohibited an in depth exploration of the configuration space. As such the configuration used during this field trial was not formally optimised.

Initially, a series of vision-only runs were conducted in order to evaluate the performance of the vision system and newly determined system calibration. During these vision-only runs it was identified that a large, proportion of spears higher than the harvesting threshold h_t, of 200mm were intermittently being identified as non-harvestable by the vision system. As presented in previous chapters, the vision system calculates the height of each spear by summing the length of a series of line segments connecting several centroids along the length of spear. Noise present in the input pointcloud translates to significant noise in this length reading. Assuming this noise is Gaussian, any imaged spear with a length of h_t will have its length under-reported in 50% of detections. For this reason h_t should be chosen to be 2-Sigma lower than the actual desired spear height in order to ensure that less than 5% of detections are under reported. In practice quantifying the exact distribution of noise is difficult, and the offset between the desired spear height and h_t is determined heuristically through a process of trial-and-error. The intermittent under reporting of spear lengths observed during this preliminary field trial was considered to be caused by an increase in pointcloud noise resulting from real-world conditions. The previously calculated h_t value was determined based on input pointclouds captured under lab-conditions, where the noise distribution was much tighter. This meant that the previously calculated h_t value was higher than the 2-Sigma point for the noisier real-world conditions. A new h_t value was determined by iteratively lowering h_t until less intermittent detections were observed.

The asparagus tracking system, described in Section 4.3.3, also performed poorly during these initial vision-only trials. The system utilises a distance threshold, d_T, to track individual asparagus spears on a frame-to-frame basis. The significant increase in field complexity, coupled with the increased

pointcloud noise between real-world, and lab conditions resulted in previously determined d_T values being too small. This was observed as a number of false spears being added to the AR due to increased variance in frame-by-frame base point predictions for each spear. This problem was solved by iteratively increasing d_T until the number of false spear entries were manageable, however this solution comes at the cost of reduced feature resolution. The ability of the tracking system to deal with densely packed groups of spears was therefore hampered.

Several runs were also conducted in order to tune parameters for RHS-1. The California field trial was the first time the RHS-1 had operated outside of lab conditions. The minimum ground speed at which AHR-1 could be operated during these trials was dictated by the slowest gear of the tractors provided by A-Bar Ag Enterprises. This was approximately 0.3m/s, significantly faster than speeds achieved during previously hand-pulled lab trials.

The control logic of the RHS-1 allows for parameterisation of many kinematic constraints. Before operation, a series of manual moves of the end-effector were performed in order to explore the operable work-space of RHS-1 in the field. In this way a z-limit was imposed on RHS-1 such that ground collisions of the end-effector could be avoided. Similarly, the acceleration and maximum rail velocities were initially down-tuned, and progressively increased back to the design point over a number of test runs. This was done to reduce the risk of skipping steps due to the added impulse coming from AHR-1's interaction with the ground at these untested speeds.

Finally, tuning of the interaction between RHS-1 and vision system was performed. Firstly, the exclusion-zone coefficient, d_f, was determined, as introduced in Section 4.3.5, by operating the machine with harvesting operations disabled. In this mode of operation, RHS-1 would move to the y-position of each target, but not initialise any z movement or end-effector instructions. In this way the end-effector would simply hover above the target spears. By cross-referencing the visual output of the vision system with the motions of

RHS-1 it was possible to incrementally increase d_f until the desired y-position was achieved before each target spear was behind the end effector. Once AHR-1 was operating in this fashion it was determined that the exclusion zone was sufficiently large.

The grab and down offsets, d_g and d_d, as described in Section 4.3.5, were found to be poorly matched to the operating conditions of the field trial. This is not surprising as both parameters are heavily reliant on the machine's ground speed, which was significantly faster during the field trials than it was during lab tuning. Both of these parameters were independently tuned in the field with a trial-and-error based approach. Following this process, the minimum viable d_d was determined. This value represents the target position offset required to allow the z-axis move of RHS-1 to complete right as the end effector is passing the target spear. Increasing d_d higher than this minimum value results in the end effector moving into position ready to intercept spears earlier. The machine achieved better performance when operating with a d_d value slightly higher than the minimum value. The reason for this is discussed in the following subsection.

5.1.3 Evaluation Method of AHR-1

Once an initial system tuning was completed a large number of runs were conducted in order to evaluate the performance of AHR-1. During these runs fine adjustments were made to the system configuration. In this way a preliminary exploration of the configuration space was achieved, enabling an understanding of the most critical parameters for successful operation to be gained.

These runs were conducted over two days and one evening during the final week of the two week US excursion. The logistics of conducting these trials so far from home also restricted access to available human resources, and equipment. These limitations meant that devising a practical method for collecting ground truth field data was extremely difficult. This challenge is further exacerbated by the fast growth rate of asparagus spears in general.

Without ground truth data, meaningful quantitative evaluations are hard to achieve. The evaluation method adopted during this field trial was as follows:

1. Operate AHR-1 on a section of asparagus row, recording video footage of RHS-1 and target spears.
2. Review the video footage and tally the number of harvesting actions taken by RHS-1, and the number of these actions which result in a spear being harvested.

The state machine which dispatches RHS-1 is very strict in its operation. This allows the following assumptions to be made about the system:

- RHS-1 is only dispatched to harvest a spear following a positive detection from the vision system. For this reason, every harvesting action taken by RHS-1 can be associated with a single spear detection
- Successful harvesting of a spear requires that the vision system can detect, and track a target spear's position with at-least enough accuracy and precision that the spear is within the harvesting zone of the end effector following a harvesting move by RHS-1. A successful spear harvest can therefore be considered as evidence that the vision system is accurate enough for RHS-1's end-effector, however the ultimate precision of the vision system remains unknown

In the context of these assumptions, the proportion of RHS-1's harvesting actions that result in a harvested spear, P_s, gives some indication of the accuracy and precision of the vision system. Inexorably linked with this, P_s also provides some indication of how well the parameters d_f ,d_g and d_d are configured; these factors govern the interaction between the vision system and RHS-1.

While P_s does provide some quantitative measure of the machine's performance, it should be noted that this metric is far from complete. As such, the scope of the conclusions drawn from this evaluation are limited and many

aspects of the machine's performance cannot be investigated based on the collected data. Specifically, this evaluation is unable to draw conclusions regarding:

- The total number of detected spears and/or the effective precision/recall characteristics of the detector. This evaluation method does not make record of any correspondence between the vision system output, and spears which do not interact with RHS-1
- The percentage of eligible spears successfully harvested. No ground truth exists for spears which are not harvested during this evaluation. As such a definitive count of harvested versus missed spears of eligible height cannot be made. It is however, possible to make estimates of this figure from review of the video footage, however such estimates are expected to be inaccurate due to the subjective nature of interpreting 2D video frames as 3D spears
- The eligibility of harvested spears. All spears that are harvested by RHS-1 are assumed to result from a positive detection, however no ground truth data exists to determine if the harvested spear was of an eligible height at the time of harvest
- The ultimate accuracy and precision of the predicted base points. This method only allows us to determine that the accuracy and precision of the predicted base points are sufficient to facilitate harvest by RHS-1

Over the two days three separate system configurations were explored, and their performance evaluated. The standard configuration, resulting from initial on-field tuning, was used as a baseline for two additional configurations, namely *Config-1* and *Config-2*. Each of these configurations were arrived at by optimising a single individual system parameter, consequentially amplifying potential trade-offs in system performance.

Config-1 was determined by optimising the grab delay, d_g, of the end effector. It was determined, by video review that the standard configuration

Table 5.1: Evaluation of various system configurations during Californian field-trials.

Configuration	**Success**	**Fail**	**Success Percentage**
Standard	-	-	83.3
Config-1	80	8	91
Config-2	157	45	77.72

resulted in typical spear contact during harvesting being located at the rear of the gripping pads. This was not optimal as much of the pad length was under utilised, requiring higher precision from both the vision system and RHS-1 to achieve a successful harvest. By increasing d_g the average point of spear contact was moved closer to the center of the end-effector's gripper.

Config-2 was built off of *Config-1*, by incrementally reducing the down threshold, d_d. The intention was that a smaller d_d value would result in RHS-1 spending less time with the end-effector in the crop-space, minimising potential collateral damage. Additionally, it was expected that minimising d_d would allow for a smaller exclusion zone, d_f, resulting in higher spear yield due to increased RHS-1 utilisation.

Table 5.1 shows the results of the evaluations performed for each system configuration. It can be seen from the table that *Config-1* offered a significant performance increase over the standard configuration, while *Config-2* resulted in a relative decrease in performance.

Config-1 provided a performance increase because it improved the utilisation of the gripper. If the gripper pad length is G_p, moving the average point-of-contact between spear and gripper to the center of the gripper pad allows for longitudinal errors in the base point predictions to be $\pm G_p/2$. In contrast the standard configuration only accommodates errors of $+G_p$. Consequentially, spears which are closer than predicted collide with the back of the end-effector rather than being harvested. This result demonstrates that the

Table 5.2: Results of evaluation conducted under low-light conditions.

Configuration	**Success**	**Fail**	**Success Percentage**
Twilight	76	23	76.7
Night	64	47	57.7

longitudinal precision of the vision system must be at least $\pm G_p/2$ for 91% of spears detected for this run.

The performance decrease seen for *Config-2* was largely due to the fact that many asparagus spears are not perfectly straight or well formed. The vision system only considers the base point of each target spear and does not evaluate the spears actual morphology. There are a significant proportion of spears which are skewed, or bent and the end-effector is designed with a relatively small harvesting area in order to allow a more surgical operation, mitigating potential collateral damage. This however, precludes access to spears if a top-down approach vector is used as there is often not a straight line path to the base point of a given spear. When using the standard configuration, the approach vector of the end effector is much more lateral; the vector runs parallel to the ground in the direction of the machine's travel. Such an approach vector allows spears with base points slightly outside of the harvesting area to be "funneled" into the gripper by bending the spear slightly. This effectively reduces the precision requirement of the vision system, resulting in a larger number of spears being harvested. This effect is not as common when the approach vector is steep as is the case for *Config-2* which explains the relative performance decrease.

AHR-1's performance was also evaluated under twilight and night time lighting conditions using *Config-1*. Table 5.2 shows the result of both of these low-light trials.

These results demonstrate that the performance of AHR-1 decreased significantly with a reduction in ambient lighting. The vision-space of AHR-1 is equipped with artificial lighting, and the ToF camera is insensitive to lighting

outside of the IR spectrum. These factors make it unlikely that low ambient lighting conditions could have an appreciable effect on these components of the system. However, the FRCNN model (*frcnn_usaNet1*) utilised by this system is susceptible to visual differences between input images and the training dataset. Since no low-light images were included in the training set for *frcnn_usaNet1*, it is likely that the observed decrease in performance is due to poor or inconsistent performance of the FRCNN model.

The above evaluations revealed how various system configurations, and ambient lighting conditions effect the performance of AHR-1. However, without a ground-truth dataset the conclusions that can be drawn from these evaluations are limited. In general, these evaluations found that configurations which reduced the required precision of the vision-system resulted in a higher proportion of RHS-1 actions resulting in a successful harvest. This trend is trivial, and the corresponding drawbacks of increased collateral damage and reduced feature resolution cannot be evaluated without ground-truth data. Additionally, an inability to match individual spear detections to the harvested spear precludes an analysis of the actual harvesting success rate from being made. Some insight, however, can be drawn from these evaluations:

- The longitudinal precision of the vision system is at least $\pm G_p/2$, and the lateral precision is sufficient to facilitate harvest for 91% of spears detected during the trial run for *Config-1*
- The harvesting success is impacted significantly by the approach vector of the end effector. Evaluations of *Config-2* revealed that top-down approach vectors commonly result in failure due to the asparagus spear morphology
- Low-light images should be included in the training dataset of the FR-CNN model

5.2 Summary of AHR-1 Performance

AHR-1 successfully fulfilled its purpose as a proof-of-concept harvester, providing strong evidence that predictions made by the vision system are of sufficient quality to facilitate the harvesting of spears in real-world conditions. The Californian field trial demonstrated that base point predictions are within a precision of $\pm G_p/2$ in up to 91% of cases. These results were achieved under real-world conditions at a ground speed of 0.3m/s. Time pressure, coupled with limited equipment and human resources precluded the collection of ground-truth data during these trials. This has severely limited the quantitative analyses and subsequent conclusions which can be made about AHR-1's overall performance. However, based on video review and experience gained operating the machine, there are many qualitative aspects of the machines performance which are discussed below.

AHR-1 operated at a constant ground speed and utilises a RHS-1 which cannot move in the direction of travel. This means that prior to dispatching the end-effector for harvest the system must determine if the target spear will still be in the harvestable zone by the time the end-effector is in position. The exclusion zone, controlled by d_f, controls this minimum distance between RHS-1 and any new target spear. In practice this distance is about 0.5m when operating at 0.3m/s, imposing a maximum spear throughput of 2 spears per meter. The random distribution of harvestable spears means that there are many instances where multiple harvestable spears are positioned within a 0.5m section of row. In such scenarios AHR-1 is unable to harvest more than 1 spear from the group. Based on video review, this results in AHR-1 harvesting approximately 20% of all harvestable spears that it passes. It should be noted that this figure is arrived at by making subjective judgements on the eligibility of missed spears from a fixed angle video of AHR-1's operation. As such, a reasonable degree of uncertainty is expected, however a general understanding of the underlying limitations of AHR-1 can be formed.

Video review also reveals that AHR-1 occasionally misses spears when RHS-1 must make a *worst case* move. A *worst case* move is when the distance which the end-effector must travel is maximised, and as such the time required to complete the move is high. In such scenarios the allowed exclusion zone is not sufficient at an operating speed of 0.3m/s to ensure that the end effector can move into position fast enough. This problem can be fixed by increasing the size of the exclusion zone, however such a solution would exacerbate the problems outlined above. Since the frequency of *worst case* moves is reasonably low it is favourable to allow some degree of losses of this kind in order to keep the exclusion zone as small as possible.

Many of AHR-1's system parameters are heavily dependent on the operating speed of the robot. Due to the laborious nature of tuning the machine, it was deemed impractical to develop multiple system configurations for various operating speeds. Therefore, evaluations of AHR-1 provided little insight into the limits of the vision system with respect to ground speed. However, utilising real-world data collected during these evaluations enables a qualitative investigation of the vision system's performance in an offline setting. Recorded data-streams, obtained during the Californian field trials, were presented to the vision system at artificially increased rates and the resulting output from the vision system was analysed. Based on this analysis, no appreciable difference in detected spears was observed for effective ground speeds less than 1m/s.

The ROS infrastructure on which the system is built is indifferent to the source of the input data stream. There are, however, subtle differences between genuine data collected at higher grounds speeds, and recorded data being played back at an increased rate. Firstly, factors pertaining to the imaging dynamics, such as motion blur or mechanical vibration from higher speed operation are not accounted for when operating with recorded data. It is, however, not expected that significant errors are likely to arise due to imaging dynamics within the confines of ground speeds considered reasonable for agri-

cultural machinery. More pertinent to this analysis is the fact that increasing the playback rate of recorded data has the effect of artificially increasing the effective frame-rate of the data acquisition pipeline. As discussed in Section 4.3.1, data acquisition is the bottleneck which governs the rate of the overall vision system. Analyses based on increased playback rates are therefore likely to overstate the system's performance. With these factors considered this analysis shows that the vision system is likely capable of operating at speeds much faster than 0.3m/s if both the imaging, and harvesting hardware would allow.

The harvesting strategy of AHR-1 utilises the forward motion of the robot as a component of the end-effector's approach vector. This approach, although seemingly suitable for an operating speed of 0.3m/s, ties the harvesting dynamics of the end-effector to the speed of the machine. When operating at faster speeds, the time available for the end-effector to react becomes severely reduced, increasing the effective precision requirements of the base point locations provided by the vision system. This is problematic because the precision of predictions made by the vision system have shown to be inversely proportional to speed. Such a harvesting strategy is therefore unlikely to be successful at higher operating speeds.

Based on these observations it is clear that the vision system is reasonably effective at making base-point predictions to facilitate the harvest of spears. However, limitations in the available evaluation methods, coupled with hardware limitations of AHR-1 preclude a thorough understanding of the limitations, and ultimate capabilities of the vision system. For this reason, a second robotic platform, AHR-2, was developed which utilises an almost identical vision system with upgraded hardware. The following section outlines this development.

Figure 5.4: AHR-2 operating on a New Zealand asparagus row.

5.3 AHR-2 Architecture

AHR-1 was widely considered successful, both as a demonstration of the vision systems capabilities, as well as a proof-of-concept robotic platform. The success of AHR-1 during the Californian field trials garnered significant attention from various growers and representative organisations that were interested in understanding the potential commercial viability of an upgraded robotic system. AHR-2, shown in Figure 5.4, was constructed, in part, to explore this space. AHR-2 was also constructed to allow a more thorough investigation into the limits of the vision system.

AHR-1 achieved a relatively low spear throughput of approximately 2 spears per meter during the Californian field trials. This low throughput resulted in the machine harvesting only about 20% of harvestable crop per pass. The primary reasons for AHR-1's poor performance in this regard was determined to be due to limitations of RHS-1, specifically the rail speed and size of the exclusion zone. AHR-2 was constructed with a much upgraded robotic

harvesting system (RHS-2) to remedy these problems. A system diagram of AHR-2 and RHS-2 are shown in Figure 5.5 and Figure 5.6 respectivley.

RHS-2 includes a 3-axis linear rail system as seen in Figure 5.7. The system is designed to allow for much higher acceleration and maximum speed in the y-z plane. The additional x-axis moves a little over 1m/s. The inclusion of an additional axis is critical because it allows AHR-2 to *double back* for missed spears without the overall machine stopping. This significantly reduces the problems associated with AHR-1's exclusion zone, significantly improving the potential spear throughput of the machine.

Significant modifications were also made to the end-effector for AHR-2. Figure 5.8 shows a detailed view of the improved design. These modifications were made in response to both a performance review of AHR-1's end-effector as well as grower feedback. AHR-2's end-effector utilises a series of pneumatic rams to actuate the gripper and blade independently. This is in contrast to the design utilised by AHR-1, in which the cutting and gripping actions were mechanically linked. The cutting action performed by AHR-2 mimics the typical method utilised by manual harvesters; the blade is thrust into the ground near the base of the spear. This cutting method has several advantages over AHR-1's method. Firstly, decoupling the gripper from the cutting action allows the cutting force to be significantly higher without affecting the gripping dynamics. This allows for much cleaner cutting of the asparagus spears. Secondly, this harvesting method allows the end-effector to cut closer to the soil plane without risking collision with the ground. This is advantageous because it minimises the height of the "spear stumps" which many growers value as a method for reducing disease and maintaining a tidy soil bed.

The core of AHR-2's vision system is identical to AHR-1, however some improvements were made to RHS-2's control logic in order to facilitate the additional axis. Likewise, many of the previously constant control parameters were reformulated as functions of the robot's ground speed. This allows a wide variety of ground speeds to be tested without re-tuning the machine.

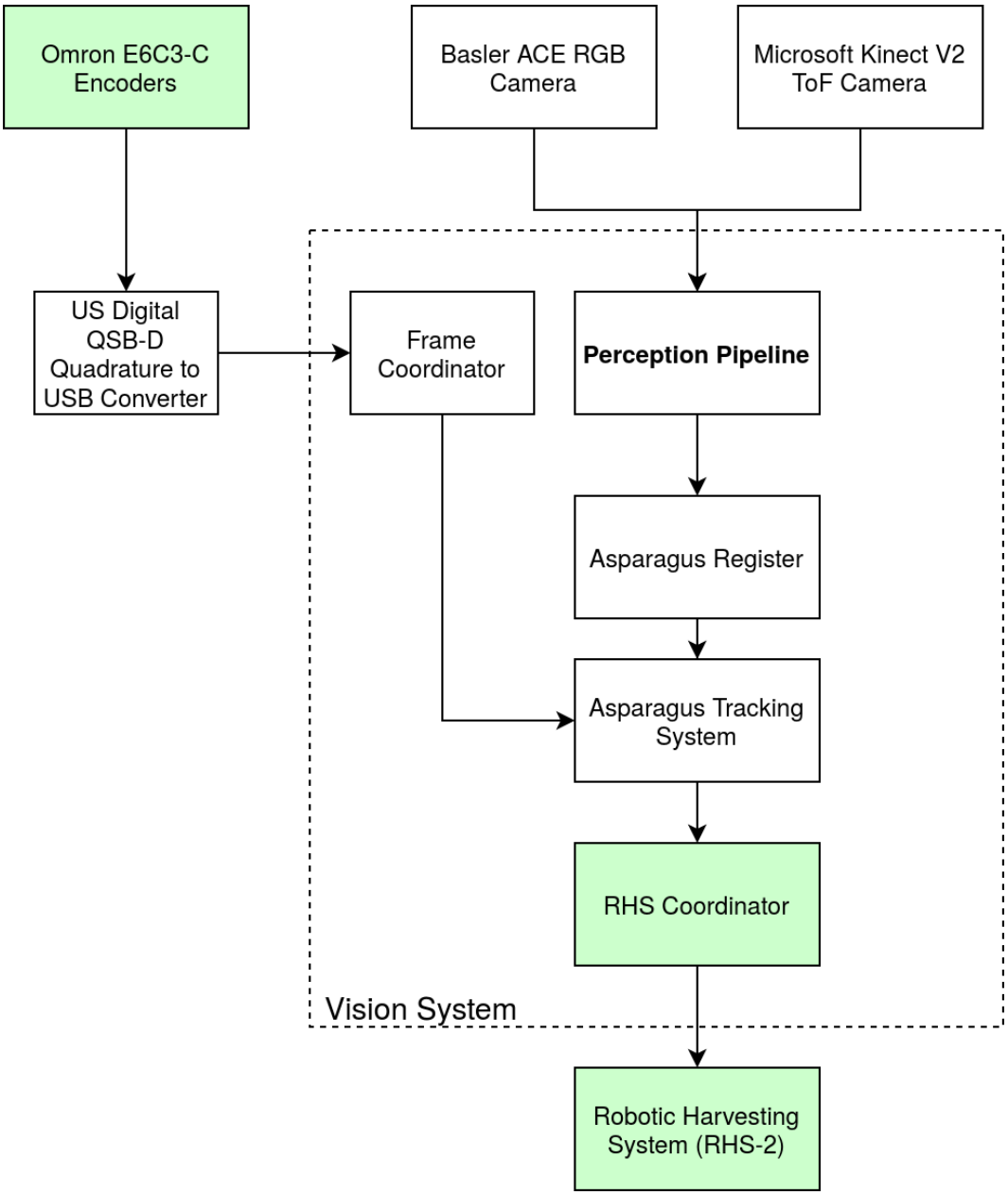


Figure 5.5: System diagram of AHR-2. The elements highlighted in green have changed from AHR-1.

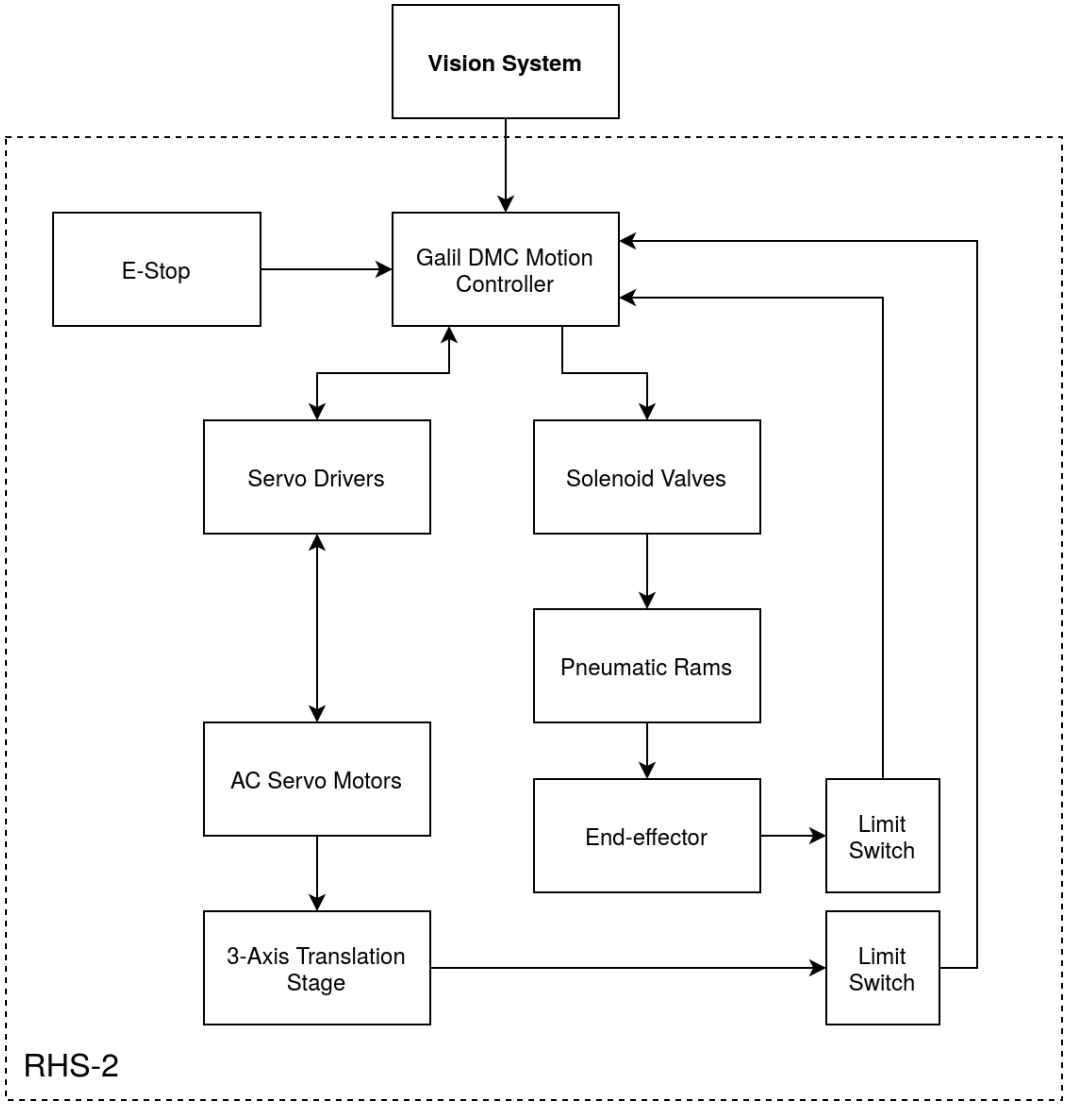


Figure 5.6: System diagram of RHS-2.

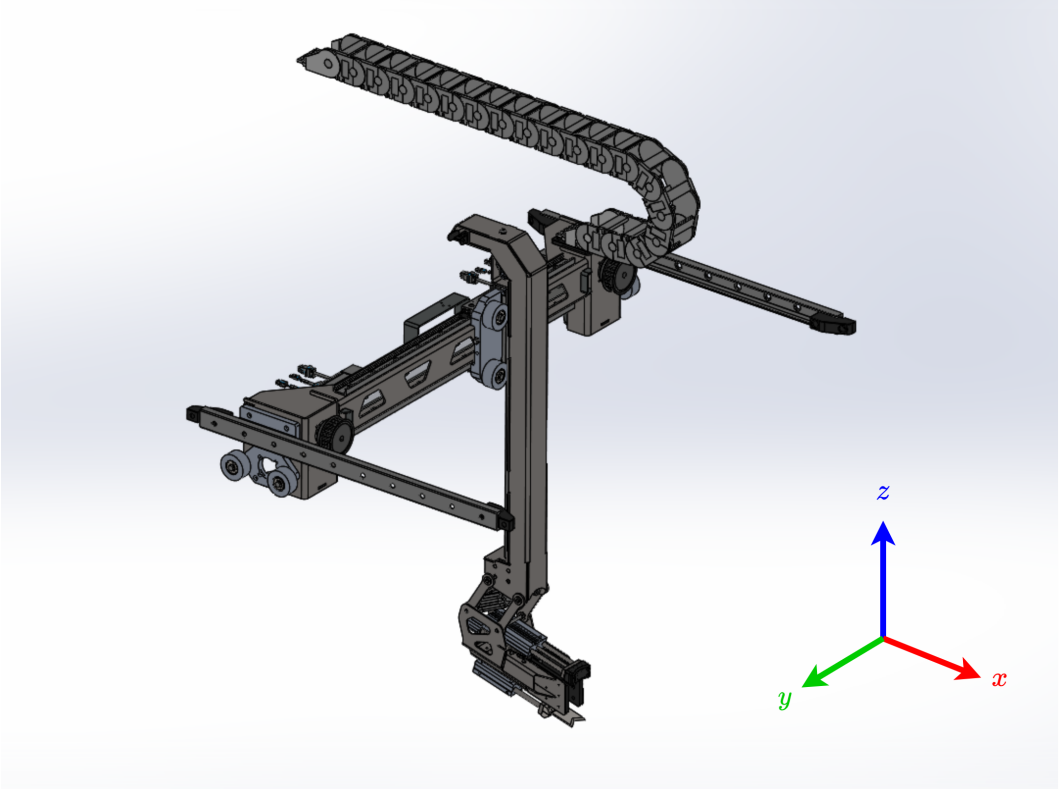


Figure 5.7: Image showing RHS-2, a 3-axis linear rail system.

Figure 5.8: AHR-2's end-effector. Gripping and cutting actions can be individually actuated via pneumatics.

The overall harvesting paradigm for RHS-2 is the same as it was for RHS-1. In general the paradigm is:

1. Identify a target spear.
2. Move in the y-axis to align the end-effector with the target spear.
3. Wait until the target spear is within the "down threshold", d_d of the end-effector.
4. Initiate a z-axis move to move the end-effector into position to intercept the spear.
5. When the spear is within the "grab threshold", d_g trigger the end-effector to grasp, and cut the spear.
6. Perform the drop-off routine to deposit the spear.

However, the addition of the x-axis to RHS-2 adds a degree of complexity. This is because a target spear can now be located behind the current position of the end-effector, requiring a negative x-axis move in order to position the end-effector at the required "down threshold". Furthermore, when the ground speed of the robot is considered it becomes increasingly important to account for the time required for RHS-2 to complete any given move. This is because the overall motion of the robot adds significant translation to the position of the target spear with respect to RHS-2. Figure 5.9 demonstrates this situation.

In order for the end-effector to be positioned a distance, d_d in front of the target spear at X_p the end-effector must move to position X_r. This position is displaced by a distance X_d which is the total distance translated by the target spear during the time RHS-2 is moving to X_r at the robot's current ground speed. Mathematically this is represented as:

$$X_r = X_p - X_d - d_d \tag{5.1}$$

however, this formulation is recursive as X_d depends on X_r:

$$X_d = (X_r / V_{\text{RHS}}) V_{\text{AHR}} \tag{5.2}$$

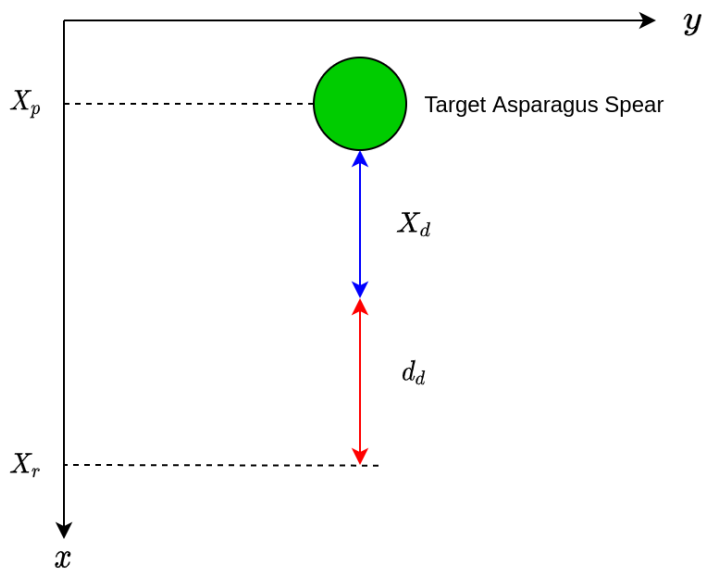


Figure 5.9: Diagram showing the calculation of the required down position for 3-axis RHS-2.

where V_{RHS} and V_{AHR} are the effective velocities of the end-effector and AHR-2 respectively. AHR-2 therefore utilises a numerical Newton-Raphson solver to estimate X_r over five iterations.

The grab factor, d_g was also expressed in terms of the ground speed for AHR-2. The time required for the end-effector to complete a grasp is relatively constant, controlled by the flow rates of various pneumatic systems. This means that the x-axis distance travelled by the end-effector in the time between a grasp command being initiated and the end-effector completing actuation can be calculated as:

$$X_g = V_{\text{RHS}} t_g \tag{5.3}$$

where t_g is the time required to complete a grasp and V_{RHS} is the ground speed of the robot. For high values of V_{RHS} this distance can be significant. Therefore, in order for the end-effector to grip a spear at the desired offset, d_g, AHR-2 sends a grip command at a position $d_g + X_g$ in order to account for the ground speed of the robot.

In order to gain an objective understanding of the performance of AHR-2 a robust evaluation method was required. The following section discusses this

need and presents a novel evaluation method for a selective robotic asparagus harvester.

5.4 Development of Evaluation Methods

Based on a review of evaluation methods utilised for various asparagus harvesting robots throughout the literature a number of critical flaws were identified. The performance of the CAMIA, Geiger-Lund, and Haws harvesters were all evaluated by counting the number of missed, dropped and damaged spears remaining in the field after robotic harvesting. Each of these harvesters are reported to have successfully harvested an impressively high percentage of marketable spears, with values ranging from 70% to 85%. However, the evaluation methods utilised in these studies make no attempt to quantify the false positive rate of each respective harvester. Such a figure is vitally important because it allows an understanding of how well selected the harvested spears are. Trivially, a simple bulk harvester, for example, could easily harvest nearly 100% of the marketable spears, however such a machine would also harvest all the shorter than marketable spears. This is not acceptable. These evaluation methods are critically flawed, in that they do not allow the context in which their reported harvesting rates are achieved to be determined.

Evaluation methods utilised by the AmLight, and subsequent GaRotics harvester are not well documented in the literature, although the stated performance metrics imply exactly the opposite problem to the methods stated above. Studies corresponding to these harvesters only report that the harvesting arm successfully harvested a spear 90% of the time. As such, no insight is provided on the overall capability of the harvester or constituent vision system.

It is vital that the evaluation methods utilised in this research are not flawed in these ways. The following sections outline the development of a novel evaluation method for an asparagus harvesting robot, and discuss the

philosophical ramifications of the various assumptions from which these methods were constructed.

5.4.1 2019 New Zealand Field Trials of AHR-1

A series of field trials were conducted with AHR-1 during the 2019 New Zealand asparagus season in order to explore various methods of evaluation. The main goal of these trials was to address the shortcomings of the evaluations performed during the California field trials and to develop a practical means by which ground-truth data could be collected and incorporated into the analysis. AHR-1 underwent a full reassembly, and system calibration after being transported back to New Zealand from the USA. Additionally, the wheel spacers which enabled the robot to operate on the Californian asparagus rows were removed to accommodate the flat growing systems found in the New Zealand.

Collecting ground-truth data in an unstructured agricultural environment is extremely challenging. Ideally, such a dataset would provide a means of directly comparing various spear metrics, such as spear height, diameter, form and base location, with values derived from direct measurement. However, encapsulating the entire structure of any section of asparagus row of appreciable length was considered impractical with the available resources. For this reason, only a limited number of spear metrics and/or proxy measurements can feasibly be determined.

During the 2019 NZ field trials of AHR-1, expert judgement was explored as a potential proxy for individual spear height and width measurements. Expert judgement is simply an appraisal of harvest eligibility performed by a skilled human asparagus harvester. Such experts routinely utilise individual judgement when selecting harvestable spears during manual harvesting. It is therefore reasonable to assume that an expert's decision on the harvesting eligibility of each individual spear is sufficient to indicate that a given spear is of appropriate height, and width to harvest; this approach is much more practical than physically measuring each spear.

Two expert asparagus harvesters volunteered to assist with the 2019 NZ field trials. Firstly, three different 30m long sections of asparagus row were selected from various areas of the available field in order to provide an accurate representation of the entire field. Spears from each section were then assigned an ID and an expert appraisal following the following procedure:

1. Starting from the beginning of a section, begin walking down the length of the row recording video footage of the asparagus spears.
2. Have an expert sequentially identify individual spears by either pointing, or touching the spear tip. For each new spear record the spear number, and have the expert declare either "harvest", "cull", or "leave".
3. Repeat this process for each expert, and each test section.

The three appraisal classes, "harvest", "cull" and "leave", can be described as follows:

- Harvest - The spear is of sufficient height, diameter and quality to be harvested
- Cull - The spear is either malformed, or not of sufficient diameter or quality to harvest. The spear should be cut and left in the field. This is a common operation during manual harvesting
- Leave - The spear is of sufficient quality but has not yet reached a harvestable height. The spear should be left undamaged to allow further growth before harvest

Following expert appraisal, AHR-1 was operated on each of the test sections and recordings of both the incoming data-stream as well as the outgoing stream of base-point predictions were made. Additionally, the corresponding harvesting actions taken by RHS-2 were recorded.

The resulting dataset consisted of a set of three corresponding videos for each of the three test sections. It was intended that review of this video footage would allow a correspondence between both expert appraisals, vision system output and harvesting outcome of each individual spear to be determined.

Determining this correspondence however, proved extremely challenging for a variety of unforeseen reasons.

The subjective element of each appraisal made it difficult to reach consensus on the eligibility of individual spears. Such scenarios were much more frequent than anticipated. This difference in opinion between experts is most readily attributed to differences in each grower's operation, specifically with regard to the markets to which they sell, and the maximum throughput capability of their available pack houses. These considerations mean that each grower has a different proclivity for accepting shorter and/or lower quality spears. Furthermore, these opinions are likely to shift throughout the season based on the performance of their respective crops and annual yield targets.

A further confounding issue with this analysis was that it was difficult to match individual spears between datasets. It was expected that experts would identify spears in a similar order as they traversed the row, allowing for simple matching based on the recorded video footage. However, in practice the spear selection methodologies differed wildly. Furthermore, the video footage of the expert's appraisals were captured by three different people; one videographer was assigned to each row section. This was done in the interests of time-saving, both for the expert's sake and to mitigate the influence of spear growth over the day, however, the inconsistency between each resulting video compounded the difficulty of spear matching between datasets.

The particular field on which these trials were conducted was also of reality low quality, housing a 15 year old crop which was set to be re-sown at the end of the season. As such, the health of the crop was much worse than typical fields in the region. This is a concern as any potential results which can be drawn based on these trials are unlikely to be representative of the majority of commercial asparagus fields.

Despite the difficulty, initial headway was made into analysing these datasets. However, during this time AHR-2 was under development, demanding considerable attention. It was determined that it was not worth continuing the

analysis, particularly due to the poor quality of the field on which the data was collected. The results of this analysis were therefore inconclusive, however certain conclusions can be drawn about the evaluation methods themselves.

Firstly, these trials demonstrated that expert appraisal cannot be relied upon as a proxy for ground truth data. The degree of disagreement was found to be much higher than anticipated, invalidating much of the collected data. Table 5.3 shows an excerpt from a dataset of video-matched spears, and their corresponding expert appraisals. Only 55% of the spears shown in this table had an agreeable appraisal.

Secondly, it was concluded that further effort must be made to ensure that individual spears can be matched between datasets. Without such a correspondence it is impossible to draw conclusions about the vision system's performance.

5.4.2 Field Trial and Evaluation of AHR-2

Field trials of AHR-2 were conducted during New Zealand's 2020 harvesting season. Prior to these trials, an iterative on-field tuning process was conducted in order to determine an optimum system configuration. The methodology for this tuning process closely resembled the on-field tuning of AHR-1 performed during the California field trials.

Previous evaluations of the vision system had failed to provide a great deal of quantitative insight primarily due to the lack of available ground truth data. Generating a ground-truth dataset was therefore the primary goal of the 2020 field trial.

A secondary goal of the 2020 field trial was to evaluate the overall performance of the robotic harvester. The results of this evaluation help to guide the development of future harvesters towards commercial viability.

As previously mentioned, generating a ground-truth dataset from an unstructured environment like an asparagus row is extremely difficult, and laborious. An ideal ground-truth dataset would encapsulate the entire form of

Table 5.3: Excerpt from a dataset of video-matched spears. The table demonstrates that expert appraisals often do not agree. This highlights the subjectivity of the selective harvesting task.

Spear ID	**Expert 1**	**Expert 2**	**Agreement**
1	Leave	Cull	
2	Leave	Cull	
3	Harvest	Harvest	✓
4	Leave	Leave	✓
5	Leave	Harvest	
6	Leave	Leave	✓
7	Harvest	Harvest	✓
8	Harvest	Cull	
9	Harvest	Harvest	✓
10	Harvest	Harvest	✓
11	Harvest	Cull	
12	Cull	Harvest	
13	Leave	Cull	
14	Cull	Cull	✓
15	Leave	Leave	✓
16	Leave	Leave	✓
17	Leave	Cull	
18	Harvest	Harvest	✓
19	Harvest	Cull	
20	Leave	Leave	✓

the asparagus row, allowing direct comparisons of spear base-points, widths and lengths predicted by the vision system to values derived from direct measurements of the real-world scenes. Such a dataset, however, is impractical to construct. Instead, a carefully selected subset of possible measurements should be selected to strike a balance between practicality and scene representation.

Various methods for determining the ground-truth base points of spears were conceptually explored. The two most considered methods for achieving this were:

1. Constructing a Cartesian grid over the asparagus row. This method involves marking sections of the asparagus row with a set of perpendicular axes, such that the base position of each spear can be measured. These positions could then be transformed into the robots. *base_link* to act as ground truth data.

2. Utilising GPS to determine base point locations. The starting position and orientation of the robot would then be used to transform the GPS positions into the robots *base_link* frame.

However, it was determined that the ultimate precision achievable by both of these methods was too low to enable meaningful conclusions about the vision systems accuracy. Additionally, the considerable cost associated with the equipment, and in-field set-up was considered prohibitive. It was therefore determined that ground-truth base points could not be obtained.

Without ground-truth base points it is not possible to draw conclusions about the ultimate precision of the vision system. However, the performance of the system can still be evaluated as a binary classifier if ground-truth data is available on each spears eligibility for harvest.

Initially, it was intended that the length, width, and z-height (vertical distance from the top of a spear to the ground plane) of each spear would be measured in order to determine the ground-truth harvest eligibility for each spear. Figure 5.10 shows an example of how both the length, and z-height of spears are measured. However, both the length and width measurements

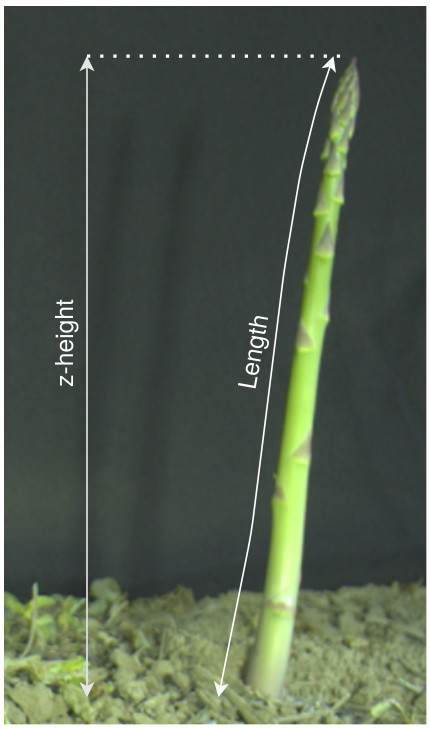


Figure 5.10: Demonstration of the difference between a spear's z-height and length.

proved prohibitively time-consuming to take. Consequentially, it was decided that only ground-truth z-height values could be collected.

The following procedure was followed to generate a dataset of ground-truth z-height values:

1. Mark out 6 test sections over several asparagus rows, each approximately 10m long.

2. Traverse each test section while a taking video recording of the asparagus spears.

3. While traversing each row sequentially identify, and label each asparagus spear by making physical contact with each respective spear tip and audibly counting such that the spear number is recorded in the video footage. Spears should be selected in order of appearance from the start of the test section, prioritising spears on the left hand side of the row. This is to aid with spear matching later in the process.

4. As each spear is counted have a second person measure that spears z-height. Every measurement should be audibly stated so that a record can be maintained on the video footage. Likewise, the physical act of measuring each spear should be visible in the video footage.

5. A third person should act as record keeper, writing down each z-height measurement and corresponding spear number.

During the 2020 field trials, this procedure was conducted on two separate farms, namely "Boyds Asparagus" and "Kaimai Fresh", resulting in a dataset of 761 ground-truth z-height measurements.

Corresponding datasets of vision system outputs, and harvesting outcomes for each test section were then generated by operating AHR-2 on each test section, and recording video of both RHS-2, and input data-streams. Three different ground speeds, specifically 0.3m/s, 0.5m/s and 0.7m/s were tested, each speed being utilised on two unique test sections. AHR-2 was operated on each test section immediately after the ground truth for that section was recorded to minimise the influence of spear growth. In order to ensure that the ground speed of AHR-2 remained constant on each test section, operation of the machine was initiated approximately 20m in front of each section. This allowed the machine to accelerate up to the test speed prior to engaging with each test section. Additionally, all asparagus spears within a small zone immediately before each test section were removed to ensure that RHS-2 would not be in motion prior to engaging with each test section.

During the 2020 field trial it was observed that the vision system achieved very few detections at an operating speed of 0.7m/s. It was determined that this was because the vision system was not being presented an adequate number of frames for each spear due to the relatively slow performance of the acquisition pipeline. In order to solve this problem the threshold, T_R which controls the number of spears required to define a spear was reduced whenever an operating speed of 0.7m/s was required.

Extensive review of the video footage was performed in order to determine if each individual spear in the ground-truth dataset was:

1. Detected and/or correctly classified by the vision system.
2. Harvested by RHS-2.

It was determined that for the purposes of this evaluation the z-height of a spear could be used as an adequate approximation of the spear's length. For the vast majority of spears this is a valid assumption. Applying a z-height threshold on the ground-truth data is therefore sufficient to determine the ground-truth harvest eligibility. This makes it possible to analyse AHR-2's performance as a binary classifier by comparing the expected outcome of each spear based on the ground-truth dataset with the actual outcome based on AHR-2's performance.

Following this process every spear in the ground-truth data set was assigned both a *detection*, and *harvest* status. The *detection* status represented how the spear was perceived by the vision system. Spears which were detected, were assigned as status of *"Green"* if they were classified as harvestable and *"Blue"* if they were not. Spears which where were not detected by the vision system were assigned a status of *"False"*. Likewise, spears which were successfully harvested were assigned a *harvest* status of *"True"* while spears which were not harvested were assigned a status of *"False"*.

Analysing AHR-2's performance as a binary classifier requires the determination of the number of true positive (TP), false positive (FP), and false negative (FN) actions (type I and type II errors). There are, however, three different perspectives from which this analysis can be performed. Each perspective requires different definitions of these metrics and offer different insights into the overall system performance. These perspectives can be explained as follows:

- **All Detections:** This perspective considers all detections made by the vision system and aims to analyse how well the vision system performs at

detecting and classifying asparagus spears in general. When considering this perspective:

  - **True positives:** are defined as all instances where a spear from the ground-truth dataset is detected, and correctly classified by the vision system. That means that all cases where the ground-truth z-height for a given spear is higher than the harvesting height threshold, h_t, and the *detection* status for that spear is *"Green"* are considered as *true positive* detections. Likewise, spears with ground-truth z-heights lower than h_t, and with a *detection* status of *"Blue"* are also considered true positives
  - **False positives:** are defined as all instances where a spear is detected, but incorrectly classified. That means all cases where the spears ground-truth z-height is higher than h_t, but the *detection* status is *"Blue"*, or cases where the ground-truth z-height is lower than h_t, but the *detection* status is *"Green"*
  - **False negatives:** are defined as all instances where a spear is not detected by the vision system. Any time a spears *detection* status is *"False"* is considered as a *false negative*

- **Targets Only:** This perspective only considers spears with ground-truth z-heights higher than h_t. Since all such spears are desired targets of AHR-2 it is useful to analyse the datasets from this perspective in order to understand the vision systems detection performance on targets independently from other detections. When considering this perspective:
  - **True positives:** are defined as spears with ground-truth z-heights greater than h_t, and with a *detection* status of *"Green"*
  - **False positives:** are defined as spears with ground-truth z-heights less than h_t and with a *detection* status of *"Green"*

  - **False Negatives:** are defined as spears with ground-truth z-heights greater than h_t which were not detected by the vision system. Such spears have a *detection* status of *"False"*

- **Harvests:** In order to evaluate the overall performance of the harvester it is useful to consider how well the ground-truth z-height data corresponds to the spears which were actually harvested during the trials. When considering this perspective:

  - **True positives:** are defined as spears with ground-truth z-heights greater than h_t that also have a *harvest* status of *"True"*
  - **False positives:** are defined as spears with ground-truth z-heights less than h_t that also have a *harvest* status of *"True"*
  - **False negatives:** are defined as spears with ground-truth z-heights greater than h_t that also have a *harvest* status of *"False"*

5.4.3 Determination of Precision and Recall Characteristics for AHR-2

Based on these definitions, precision and recall values can be calculated for each perspective with the previously defined equations 4.1 and 4.2:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

where TP, FP and FN denote the number of *true positives*, *false positives*, and *false negatives* for any given perspective. For each perspective the TP, FP, and FN detections were also categorised by speed. Table 5.4 shows the precision and recall values determined based on this analysis.

The precision and recall values corresponding to various speeds can be interpreted with respect to each perspective as follows:

- **All Detections:**

  - **Precision** is the proportion of detections made by the vision system which were correctly categorised
  - **Recall** is the proportion of all spears in the ground-truth dataset which were both detected and correctly categorised by the vision system

- **Targets Only:**
  - **Precision** is the proportion of all spears which were considered harvestable by the vision system that were actually harvestable
  - **Recall** is the proportion of all harvest eligible spears in the ground-truth dataset which were detected by the vision system and classified as harvestable

- **Harvests**
  - **Precision** is the proportion of all spears that were harvested by RHS-2 that were actually eligible for harvest
  - **Recall** is the proportion of all harvestable spears in all relevant test sections which were actually harvested by RHS-2

5.5 Discussion of Vision System Performance

Table 5.4 shows that the vision system achieved an *All Detection* recall of 0.549 with a precision of 0.830 at a ground speed of 0.3m/s. This means that of the 257 relevant spears for which ground-truth data was collected, the vision system detected and correctly classified 54.9%. Furthermore, 83% of all detections made by the vision system at this speed were correctly classified. The *Target Only* precision/recall characteristics were significantly higher, representing a 97.2% detection rate of all harvestable spears in the ground-truth dataset with a true positive rate of 74.5% at a speed of 0.3m/s. The large difference between these recall values indicates that the majority of failed detections cor-

Table 5.4: Precision and recall characteristics for various ground speeds based on 2020 Field trials of AHR-2.

All Detections			
Speed (m/s)	Entries	Precision	Recall
0.3	257	0.830	0.549
0.5	252	0.843	0.461
0.7	252	0.889	0.471
Targets Only			
Speed (m/s)	Entries	Precision	Recall
0.3	96	0.745	0.972
0.5	85	0.759	0.969
0.7	92	0.872	0.925
Harvests			
Speed (m/s)	Entries	Precision	Recall
0.3	79	0.872	0.459
0.5	66	0.957	0.338
0.7	86	0.905	0.226

respond to spears that are shorter than the harvesting height threshold, and are therefore not harvest eligible.

Both the *All Detection* and *Target Only* recall exhibited a decreasing trend with respect to ground speed. This suggests that as the speed of AHR-2 increases the number of spears from the asparagus row which are successfully detected decreases. During field trials it was identified that higher ground speeds resulted in less data frames being processed per asparagus spear due to limitations in the speed of the acquisition pipeline. This phenomena means that spears which are more difficult to detect are less likely to achieve the requisite number of frames to survive the filtering process. As previously mentioned, for a ground speed of 0.7m/s it was necessary to reduce the frame number threshold, T_R in order to achieve any detections at all. The observed inverse proportionality of detector recall to speed is therefore expected.

In contrast, the corresponding *All Detection* and *Target Only* precision values exhibited an increasing trend with ground speed. This indicates that as the ground speed of AHR-2 increases, the proportion of detections made by the vision system which were correctly categorised increased, particularly at a ground speed of 0.7m/s. This trend can also be explained as a result of the frame filtering process. In order for a spear to be detected by the vision system, it must be detected in at least T_R, out of $N \propto 1/V_{AHR-2}$ frames, where N is the total number of frames presented by the acquisition pipeline that contain the spear. N is therefore a function of the robot's ground speed (V_{RHS}). The most pronounced spears in any given scene tend to be the most consistently detected by the vision system. In general, the probability that any given spear is detected in a single frame seems to be proportional to the spear's height. It is clear from the binomial probability distribution that the probability of achieving at least T_R detections in N frames is proportional to the probability of each trial, and inversely proportional to $(T_R - N)$. The probability of a spear surviving the filtering process is therefore proportional to the spear's height, and inversely proportional to AHR-2's ground speed. As

the ground speed of the robot increases, the probability of a spear surviving the filtering process decreases faster for smaller spears. This means that at higher ground speeds, fewer overall spears are detected, and the majority of spears which are detected are generally more pronounced due to their greater height. Since such spears are easier for the vision system to detect this results in more conservative outputs, resulting in less *false positive* detections, but also less overall spears being detected. These factors manifest as an increase in precision and a decrease in recall.

AHR-2 achieved a *Harvest* recall of 0.459 with a precision of 0.872 at a speed of 0.3m/s. This means that the robot managed to harvest 45.9% of all harvest eligible spears from the relevant test section, and that 87.2% of all spears harvested in the section were eligible. Table 5.4 shows that as the robot's ground-speed increases the *Harvests* recall falls significantly, harvesting 22.6% of eligible spears at a ground speed of 0.7m/s. There are two main contributing factors to this performance decrease. Firstly, kinematic limitations of RHS-2 mean that the maximum spear throughput is limited. This trivially results in some degree of unavoidable loss at faster ground speeds where the required spear throughput is high. More interesting perhaps, is the consideration of potential deterioration in base point prediction quality leading to missed spears. At high ground speeds the number of frames defining each spear is restricted, and in the case of 0.7m/s the frame threshold T_R is set lower than the standard configuration. Since the base point prediction is averaged over a number of frames, it is conceivable that the quality of such predictions falls as the number of contributing frames is reduced leading to missed spears. Unfortunately, the ground-truth dataset collected for this evaluation did not contain ground truth base points. This makes it impossible to measure the quality of individual base point predictions and therefore such an effect remains speculative. Based on this analysis it is not possible to determine the relative contribution of each of these factors on the perceived decrease in *Harvest* recall.

The *Harvest* precision was found to be significantly higher than the *Target Only* precision at all ground speeds tested. This is unexpected because it means that the proportion of all spears harvested by RHS-2 which were truly harvest eligible was higher than the proportion of spears targeted by the vision system which were truly harvest eligible. This suggests that RHS-2 somehow prioritised spears from the target list which were ground-truth harvest eligible, despite the fact that AHR-2 does not have any system that intentionally prioritises targets in this way. A possible explanation for this is that the vision system often detects the most prominent spears from a scene first, resulting in the most prominent spears being targeted by RHS-2 before less prominent spears are even detected. Once a spear is targeted the system will not change targets until a harvesting routine is completed. This means that when RHS-2 finishes the current harvesting routine the next target spear can be behind the previous target's location. Even when operating at the slowest tested ground speed of 0.3m/s, kinematic limitations of RHS-2 mean that such spears are unlikely to be attainable. This means that less prominent spears which take longer to detect have a chance of being located in a position which precludes them from being targeted all together. Since this phenomena specifically effects less prominent and therefore shorter spears, it disproportionately effects *false positive* target predictions made by the vision system. This results in higher *Harvest* precision, but less overall harvesting actions. As such this effect is also responsible for driving down the *Harvest* recall at all ground speeds.

A visual exploration of vision system outputs was conducted in order to identify common conditions that lead to detection failure. The results of this exploration are presented in the following section.

5.6 Investigation of Various Failure Conditions of Vision System

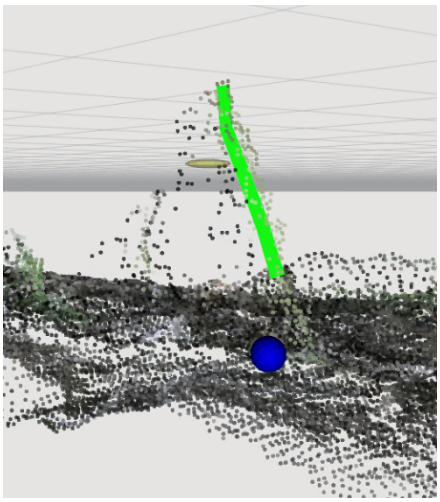
Based on a visual exploration of various detection failures it was identified that the weaknesses in the vision system fall into two main categories; clustering failures and segmentation failures. In total three types of clustering failures, denoted as C_1^E-C_3^E errors, and four types of segmentation failure, denoted S_1^E-S_4^E were identified.

5.6.1 Clustering Failures

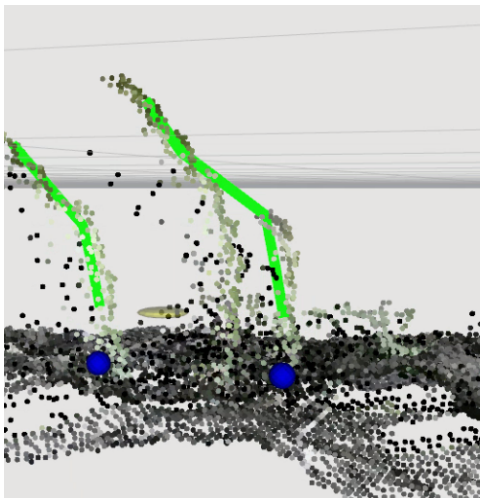
Clustering failures occur when the Euclidean cluster extraction method, outlined in Section 3.6, incorrectly combines points pertaining to multiple different spears into a single cluster. The vision system assumes that each cluster describes only a single asparagus spear. When the clustering algorithm fails in this way this assumption is invalidated, leading to a number of downstream problems resulting in detection failure.

C_1^E errors occur when points from one spear muddled in the flying pixels of another spear and consequentially included in the same cluster by the Euclidean clustering algorithm. Typically, such a scenario occurs in cases of frontal occlusion, when one spear is partially hidden from view of the camera by another spear. When multiple spears are clustered together in this way it is common for the occluded spear to be either fully, or partially filtered out by the closest point filter described in Section 3.5.3. This can result in both *false negative* detections as well as incorrect length/height classifications of individual spears. Figure 5.11 shows some examples of C_1^E errors.

C_2^E errors occur when either the base locations of two or more spears are very close or multiple spears are angled such that their tips are in close proximity. In such cases the Euclidean clustering algorithm incorrectly clusters both spears into a conjoint cluster. Unlike with C_1^E, points in such scenarios are often not filtered out by the closest point filter. Instead, the asparagus model



(a) Vision Output

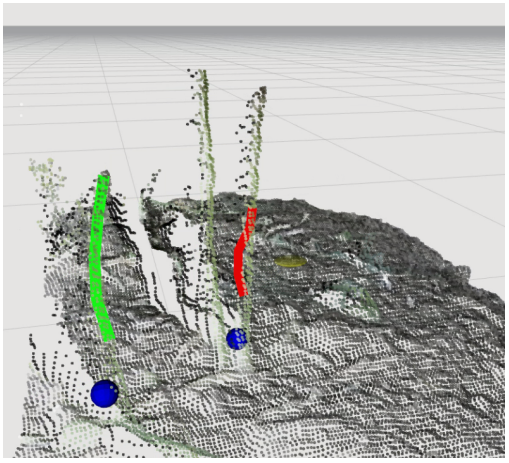


(b) Vision Output

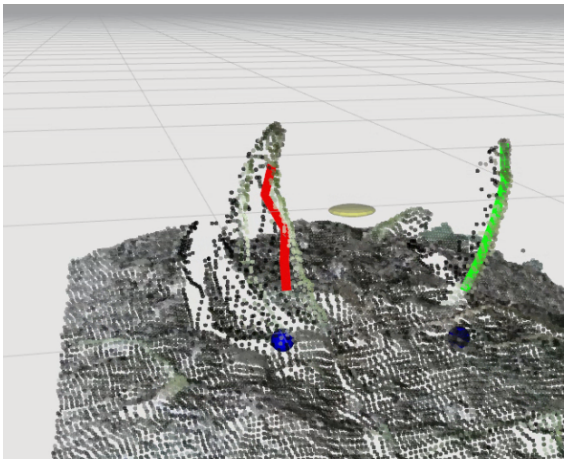
(a) Real-World Scene

(b) Real-World Scene

Figure 5.11: Examples of frontal occlusion resulting in C_1^E errors. In these cases points from the front pointcloud are clustered together with points from the occluded spear. The CP filter then removes the occluded section of the rear spear.



(a) Vision Output



(b) Vision Output

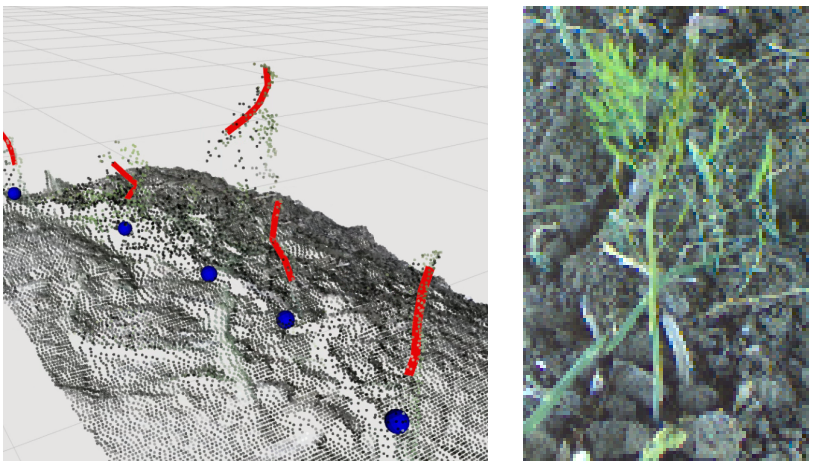
(a) Real-World Scene

(b) Real-World Scene

Figure 5.12: Examples of C_2^E errors. In these examples the proximity of either the base or tip is too close resulting in incorrect clustering. The resulting conjoint point cluster is then interpreted as a single spear by the perception pipeline, generating inaccurate base locations.

which informs the vision system output is fitted to the conjoined point cluster. This can result in wildly incorrect base point predictions and length/height estimates. Typically the predicted spear model lays somewhere in between all involved spears. Figure 5.12 shows some examples of C_2^E errors.

C_3^E errors occur when a spear is significantly malformed or beginning to fern. In such cases the resulting pointcloud is often extremely messy, invalidating many of the assumptions on which the perception pipeline is built. This results in base-point predictions that are unpredictable and inaccurate. In some cases each branching section of a ferning spear can be incorrectly in-



(a) Vision Output (b) Real-World Scene

Figure 5.13: Example of a C_3^E error. The spear shown in (b) has been incorrectly assigned multiple point clusters resulting in an extra spear being included in the vision system output, shown in (a).

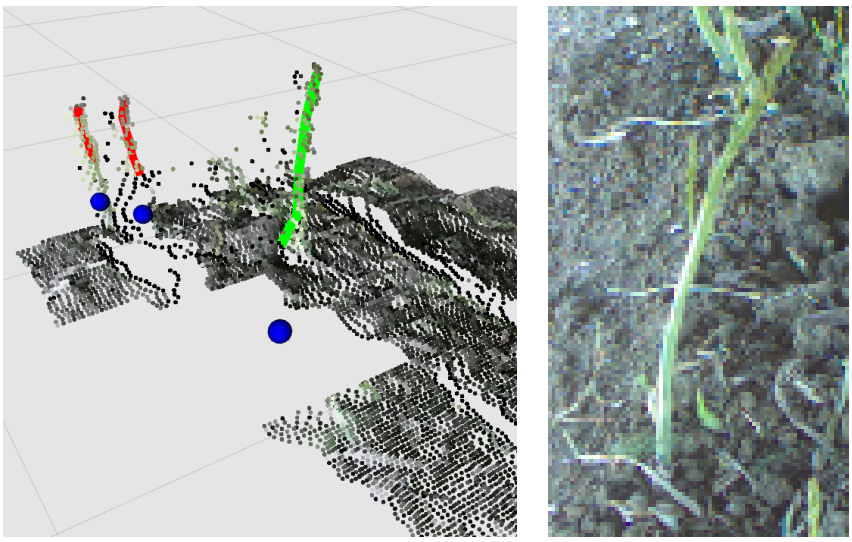
terpreted as multiple distinct clusters by the Euclidean clustering algorithm, resulting in multiple *false positive* detections. Figure 5.13 shows some examples of malformed spears which may result in C_3^E errors.

5.6.2 Segmentation Failures

Segmentation failures occur when the FRCNN model fails to perform as intended or the soil plane segmentation method incorrectly filters points from the input pointcloud. In such cases features of the input pointcloud may be incorrectly filtered leading to inappropriate inputs to the vision system. This can result in detection failure.

S_1^E errors occur when the FRCNN model fails to detect a given spear. In such cases the section of the input pointcloud which contains the spear can be filtered out, making detection of the spear impossible.

S_2^E errors occur when a spear is partially segmented by the FRCNN filter, resulting in a spear fragment being passed through the processing pipeline. Such scenarios occur when the FRCNN model fails to detect a given spear, but that spear is coincidentally partially included in the bounding box of another spear. Alternatively, spear fragments can also be created when spears



(a) Vision Output (b) Real-World Scene

Figure 5.14: Example of an S_2^E error. The gap in the pointcloud shown in (a) is due to a false negative detection by the FRCNN model. However the top half of the undetected spear has been included in the bounding box which relates to the two spears at the back of the image. This has generated a spear fragment which has been processed as a spear. The resulting base point location is inaccurate.

are partially filtered by the region filter. Spear fragments are problematic for the vision system as typically they only contain points from the tip of a spear. Consequentially, there is a large gap between the lowest points in the fragmented point cluster and the soil plane. When the base location of the fragmented point cluster is projected over this large distance to the base plane the results can be inaccurate, particularly if the affected spear morphology is not particularly vertical. Furthermore, the bottom of the fragmented point cluster is not subject to the same dead-band point removal, associated with soil plane segmentation, as a typical cluster. This means that there is significant frame-to-frame variation in the z-height of the lowest point in the fragmented cluster, further exacerbating the error present after the base is projected to the soil plane. Figure 5.14 demonstrates some examples of S_2^E errors.

S_3^E errors occur when the soil plane model is a particularly poor predictor of the ground surface in some local region. When the terrain of the asparagus bed is uneven, there can exist local areas of the bed that are far away from

the plane. Spears within this local region with base points originating far from the predicted plane can have a large proportion of points near their base removed as a result of soil plane segmentation. In such cases the affected spears can have their heights significantly misrepresented. In extreme cases this can result in the number of points defining the spear falling below the point number threshold, T_R, and being ignored completely. It is also possible that affected spears are reduced to a spear fragment, in which case the error transforms into a S_2^E error.

The final type of segmentation are S_4^E errors. These errors occur when the FRCNN model makes a false positive detection. This can result in non-asparagus features being included in the filtered pointcloud and passed down the processing pipeline. When this occurs points pertaining to the foreign features are processed as asparagus spears resulting in many C_3^E errors.

It should be noted that each of these error types were determined by examining single data frames recorded during the Californian field trials. The vision system utilises multiple frames to make decisions about target spears and their corresponding base point locations. For this reason the presence of any one of the errors does not preclude eventual detection of the affected spears. However, when the number of available frames are restricted, such as when the machine is operating at high ground speed, the probability of these errors resulting in poor detections is high.

5.6.3 Frequency of Various Error Types

An investigation was performed in order to understand the relative frequency of C_1^E, C_2^E, C_3^E, S_1^E, S_2^E, S_3^E, and S_4^E errors. A dataset consisting of $18,000$ data frames, representing a 500m long section of asparagus row recorded during the California field trials was analysed. 138 frames were randomly selected from this dataset, and the number of each type of error were determined based on manual inspection. Care was taken to ensure that the temporal distance

Table 5.5: Frequency of various errors from 138 randomly selected data frames.

Error Type	**Frequency**	**% of observations**
C_1^E	13	7.8
C_2^E	6	3.6
C_3^E	10	6.1
S_1^E	53	32
S_2^E	24	14.5
S_3^E	0	0
S_4^E	4	2.4
Unknown	55	33

between each of the 138 frames was a minimum of 1 second in order to ensure that each frame was unique. Table 5.5 shows the results of this investigation.

33% of the apparent failed detections observed during this analysis could not be attributed a definitive cause. These observations have been recorded as *unknowns* in the table. In many of these cases it was speculated that multiple of the aforementioned errors may have interacted to cause the observed failed detection. Additionally, very short spears were typically not detected. In these cases the points pertaining to the affected spears were filtered out during base plane segmentation. These cases were not recorded as C_3^E errors unless the predicted soil plane model differed significantly from the ground surface in the area local to the affected spear.

The analysis revealed that 73.6% of the *known* errors were segmentation errors, with S_1^E errors constituting 32% of all observations over the 138 frames. This suggests that *false negative* predictions made by the FRCNN model are a prevalent cause of detection failure on a frame-by-frame basis. S_2^E errors constituted 14.5% of observations making such errors more prevalent than any individual clustering error. Similarly to S_1^E errors, the main cause of S_2^E errors are due to poor performance of the FRCNN model. Based on these results it can be concluded that the majority of observed errors could be addressed by

improving the recall of the FRCNN model. However, improvements in recall are often accompanied by a loss in precision. Lower precision for the FRCNN model means a higher *false positive* rate, resulting in more S_4^E errors. Such errors are extremely unpredictable and likely more damaging to the overall performance of the vision system than S_1^E or S_2^E errors. Care should therefore be taken in balancing the precision/recall characteristics for any improved CNN model.

Spear fragments which are generated as result of S_2^E errors can cause inaccurate false positive detections. In such cases it would be preferable that the spear fragment be filtered out completely, resulting in a *false negative* detection. This could be achieved by altering the perception pipeline to enforce a maximum distance from cluster base to the soil plane.

There were relatively similar occurrences of (C_1^E-C_3^E) errors observed in this analysis. All of these types of errors are the result of failures in the Euclidean cluster extraction algorithm utilised by the vision system. Improvements to the cluster extraction strategy are therefore required to reduce these types of errors. Potential improvements are discussed in Section 6.2.

5.7 Summary

This chapter has presented a novel evaluation method for a selective asparagus harvester. This evaluation method was devised based on extensive data collection, and field testing of AHR-1 in California, USA, and the cumulative result of work presented in Chapter 3 and 4. The results of these field trials enabled the development of a prototype robotic asparagus harvester, AHR-2.

Evaluation of AHR-2 based on this method indicated that AHR-2 achieved state-of-the-art performance. The chapter also presented a visual evaluation of several data-frames from which several classes of detection failure condition were identified. Potential causes of these detection failures were also discussed. The final chapter of this thesis presents conclusions drawn from this work and

highlights critical areas where future work is likely to improve the system's performance.

Chapter 6

Conclusions and Future Work

This chapter presents a series of conclusions which are drawn based on the totality of this research. In addition, this chapter identifies several areas where future work is required and speculates on the potential commercial viability of this research.

6.1 Conclusion

This thesis presented a novel perception pipeline for ToF images that can facilitate the robotic harvesting of green asparagus spears in real-world conditions. In developing this perception pipeline a number of important academic challenges needed to be overcome.

The perception pipeline begins by removing points from an input pointcloud pertaining to the soil plane of the asparagus row. This operation requires a mathematical model describing the dominant plane in the input scene to be predicted. Two candidate algorithms, namely RANSAC and Hyun's method, were investigated to solve this problem.

Throughout this work, a Microsoft Kinect V2 ToF camera was utilised to generate the input pointclouds. It was found that Hyun's method, in its standard formulation, performed poorly on pointclouds captured from this camera. It was concluded that the method exhibited poor resilience to noise. In order to mitigate this issue a modification to Hyun's method was developed (MHM)

which improved the methods noise resilience at the cost of feature resolution. This was achieved by computing local surface normal vectors based on a pair of vectors constructed from a linear regression over a number of surface points, rather than simply the nearest points as in the original formulation.

An experiment was conducted which compared the performance of both MHM and RANSAC in scenes with varying degrees of clutter, and various camera angles. It was found that both methods exhibited increased error in their plane model predictions as the level of clutter in the scene was increased. This trend of increasing error was most prominent at the extremes of camera angles tested. It was concluded that this was the result of two factors which dominate at each extreme. Firstly, at shallow camera angles a large number of soil-plane points are positioned far from the camera, where ToF imaging is likely to achieve lower range precision. In contrast, for steep camera angles images contain points of higher precision, but represent a smaller portion of the scene due to the fixed field of view of the camera. This means that field clutter comprises a larger percentage of the visible scene, reducing the available inliers by occluding the soil-plane.

The stability of model predictions made by each method was also investigated by calculating the standard deviation in error over a set number of frames. It was found that RANSAC typically produces less stable model predictions than MHM, particularly at high clutter scenes. This was concluded to be due to the non-deterministic nature of RANSAC. For high clutter scenes, particularly for steep camera angles, it was concluded that RANSAC's low stability was due to an increased likelihood that an individual "random sample" proposed by the algorithm could achieve a significant number of inliers due to soil plane occlusion.

Due to the real-time nature of the application, the perception pipeline needs to avoid processing unnecessary data. A novel application of CNN methods was utilised to provide coarse filtering of input data in order to avoid processing areas of the scene with no asparagus features. This process, utilised bounding

box predictions, transformed into the IR image space of the Kinect camera to generate a binary mask that describes areas of the scenes where asparagus features are present. This allows efficient elimination of irrelevant points from the input pointcloud, significantly reducing the number of points to process during subsequent operations.

The implementation of the neural network filter utilised FRCNN as the object detector. This work has provided valuable academic contributions in the form of both a series of labelled datasets, and an evaluation of a standard FRCNN implementation in a real-world setting. These contributions will allow future researchers to conduct objective comparisons between existing CNN methods, and future developments.

Flying pixels were identified as problematic artifacts resulting from the mechanics of ToF imaging. To overcome inaccuracies contributed by these artifacts, this thesis presented a novel geometric method which removes flying pixels from input point clusters based on their spread in the range axis of the camera, projected to the soil plane. This method was able to achieve a 68% decrease in the mean standard deviation of intra-cluster distance when tested under laboratory conditions.

The perception pipeline produces a series of asparagus spear models, allowing the base point and z-height of harvest eligible spears to be identified and localised for robotic harvesting. To achieve this two major problems needed to be addressed, namely the problems of spear permanence, and spear tracking. The perception pipeline utilises a novel frame based approach that uses the frame-by-frame proximity of predicted base points to discriminate between primary, and secondary detections, allowing each individual spear in the scene to be matched on a frame-by-frame basis. By tracking the translation of the robot with wheel encoders each individual spear can then be tracked in world space to facilitate the harvest. Furthermore, this method enables more accurate base point locations to be determined for each spear, by considering the average of all secondary detections available. Since the number of secondary

detections for each spear is determined by the effective frame rate of upstream processes, it is expected that this approach will scale well with improvements to hardware and upstream processes.

This thesis has also presented a novel concept for an asparagus harvesting robot, on which the perception pipeline could be deployed. This concept addresses many of the weaknesses of existing asparagus harvesters in the literature. The University of Waikato utilised this novel concept, and perception pipeline to construct two purpose-built robotic platforms (AHR-1 and AHR-2), which were successfully operated on various commercial asparagus farms throughout New Zealand and in California, USA.

A novel evaluation method was also developed in order to address the inability of previously established methods from the literature to adequately consider the selective nature of the harvesting task. This novel evaluation method utilises the z-height of each spear as a proxy for harvest eligibility, and evaluates the performance of the robotic harvester and constituent vision system as a binary classifier. This significantly improves the practicality of generating a ground-truth dataset, allowing for a much more objective understanding of the harvesters performance.

This evaluation method was applied to AHR-2 during field trials conducted in New Zealand during the 2020 season. It was revealed that the vision system achieved state-of-the-art performance, successfully detecting 97%-92.5% of target spears with a precision of 74.5%-87.2% at a ground speeds of 0.3m/s-0.7m/s respectively. AHR-2 successfully harvested 45.9% - 22.6% of all harvest eligible spears with a precision of 87.2%-90.5% at ground speeds of 0.3m/s-0.7m/s respectively. Based on the evaluation, it was concluded that the observed loss in harvest recall at high ground speeds can be primarily attributed to limitations in the robotic hardware, rather than the perception pipeline.

On face value, a review of the existing literature reports higher harvesting recall among several machines than achieved by AHR-2. However, this thesis identified a number of methodological flaws in these studies which make direct

comparison of performance impossible. Evaluations of other harvesting robots described in the literature either fail to provide insight on the harvester's *false positive* rate, or are unable to describe the total harvesting recall achieved. This introduces doubt as to the ability of these machines to selectively harvest. With respect to the task of selective harvesting, AHR-2 achieved state-of-the-art performance.

6.2 Future Work

Despite achieving state-of-the-art performance, both the perception pipeline and resulting robotic harvester have various limitations which have been identified in this thesis. Addressing these limitations is expected to improve the performance of both systems considerably.

The thesis identified several classes of detection failure from a comprehensive set of recorded data-frames. It was found that these detection failures either stemmed from shortcomings in the Euclidean clustering algorithm, or the FRCNN model.

In most cases, clustering failures occurred when the pointcloud representations of multiple spears were in close proximity, or the number of points defining a feature was low. In such cases, distinguishing each individual spear based on only spherical proximity was not sufficient. If more sophisticated methods for clustering individual spear features from the input pointcloud are developed, the effective feature resolution of the detection system would be greatly improved.

Segmentation failures were found to be much more common than clustering failures and resulted from either failed detections made by the FRCNN model, or inclusion of unwanted features in the predicted bounding box. The field of research into CNN methods is vast and rapidly evolving. As such there are a myriad of potential advances from which the FRCNN model could benefit which were outside the scope of this thesis. It is likely that far greater

precision/recall characteristics are achievable from the FRCNN model than achieved in this work. Additionally, more surgical object detectors, such as Mask-RCNN, are worth exploring as a way to reduce the inclusion of unwanted field features. Object detectors that operate directly on pointcloud information are another potential line of inquiry.

This research utilised a Kinect camera which, while respected by many researchers that are interested in depth imaging, is understandably limited due to its status as a piece of consumer electronics. Industrial devices offer significant advantages over this device. Firstly, such devices are usually capable of hardware triggering which is advantageous as it allows hardware synchronisation between RGB and depth images, speeding up the acquisition pipeline considerably when compared to software based approaches. Industrial devices also allow custom lensing to be designed, which allows only the region of interest from the field to be focused onto the limited pixels of the camera. This vastly improves the effective resolution of the ToF camera, resulting in much higher feature resolution.

Evaluation of AHR-2 concluded that the vision system was capable of detecting 97% of all harvest eligible spears, successfully harvesting 45.9% of all harvest eligible spears at a ground speed of 0.3m/s. These values fell to 92% and 22.6% respectively for a ground speed of 0.7m/s. The corresponding harvester and detector precision remained relatively high at all ground speeds. It is clear from these results that the hardware on-board AHR-2 was the major limiting factor for performance. In particular, unavoidable loss of spears due to the constant ground speed was identified as a major flaw.

It is expected that if the robotic platform was self-driven, or otherwise capable of regulating the instantaneous ground speed that the machine could slow-down or stop to avoid missing targeted spears. This would result in a significant improvement to the harvesting recall with limited decrease in the harvesting precision. There are two possible design philosophies which can be explored based on this conclusion.

Firstly, the vision system could be integrated into a fully autonomous vehicle that is capable of navigating both the farm, and asparagus rows. Such a machine, however, is somewhat of an ultimate goal, requiring a considerable amount of additional research, development and funding. A more realistic next-step in this technology would perhaps be integrating the system into a human operated vehicle that is capable of regulating its speed automatically during harvesting. It is expected that such a vehicle could be developed with minimal additional research, and depending on cost, could provide growers with a commercially viable alternative to manual harvesting.

6.3 Concluding Remarks

Demand for agricultural automation is ubiquitous throughout the world. Widespread adoption of robotic systems, capable of achieving complex agricultural tasks, such as crop harvesting, is on the horizon. In order to realise such advancements the gap between theory and application must be bridged.

This thesis has presented a novel perception pipeline for green asparagus, which has demonstrated that robotic selective asparagus harvesting is technologically within our grasp. Such a result is a testament to the ever increasing advancements in computation, sensing and mechanical design, and begins to bridge the gap between theory and application.

References

[1] *World Population Prospects: The 2017 Revision*. Tech. rep. UN Department of Economic and Social Affairs, 2017.

[2] Kirsten Lovelock and Teresa Leopold. "Labour Force Shortages in Rural New Zealand: Temporary Migration and the Recognised Seasonal Employer (RSE) Work Policy." In: *New Zealand population review / New Zealand Demographic Society (Inc.)* 33 (Jan. 2009), pp. 213–234.

[3] Balwinder-Singh et al. "Agricultural labor, COVID-19, and potential implications for food security and air quality in the breadbasket of India". In: *Agricultural Systems* 185 (2020), p. 102954.

[4] *The future of food and agriculture - Alternative pathways to 2050*. Tech. rep. Food and Agriculture Organization of the United Nations, 2018.

[5] Jörgen Frohm et al. "Levels of Automation in Manufacturing". In: *Ergonomia - International Journal of Ergonomics and Human Factors* 30 (Jan. 2008), pp. 181–207.

[6] Thomas Davenport and Ravi Kalakota. "The potential for artificial intelligence in healthcare". In: *Future Hospital Journal* 6 (June 2019), pp. 94–98.

[7] Mike Daily et al. "Self-Driving Cars". In: *Computer* 50.12 (2017), pp. 18–23.

[8] L. Emmi et al. "Integration and Assessment of a Real Fleet of Robots". In: *New Trends in Robotics for Agriculture* (2014).

[9] S. M. Pederson et al. “Agricultural robots - system analysis and economic”. In: *Precision Agriculture* 7 (2006), pp. 295–308.

[10] Athanasios T. Balafoutis, Frits K. Van Evert, and Spyros Fountas. “Smart Farming Technology Trends: Economic and Environmental Effects, Labor Impact, and Adoption Readiness”. In: *Agronomy* 10.5 (2020).

[11] R. G. V. Bramley and J. Ouzman. “Farmer attitudes to the use of sensors and automation in fertilizer decision-making: nitrogen fertilization in the Australian grains sector”. In: *Precision Agriculture* 20 (2019).

[12] C. W. Bac et al. “Harvesting Robots for High-Value Crops: State-of-the-Art Review and Challenges Ahead”. In: *Journal of Field Robotics* 31 (July 2014).

[13] S. Sander. *BoniRob: An Autonomous Mobile Platform for Agricultural Applications*. 2015.

[14] A. Ruckelshausen et al. “BoniRob - an autonomous field robot platform for individual plant phenotyping”. In: *Precision Agriculture* 9(841) (2009).

[15] O. Bawden et al. “A lightweight, modular robotic vehicle for the sustainable intensification of agriculture”. In: *Australian Robotics & Automation Association ARAA* (2014).

[16] R. N. Jørgensen et al. “Hortibot: A system design of a robotic tool carrier for high-tech plant nursing”. In: *CIGR Ejournal* IX (2007).

[17] A. Ruckelshausen et al. “Autonome Roboter zur Unkrautbekämpfung”. In: *Zeitschrift für Pflanzenkrankheiten und Pflanzenschutz* (2006), pp. 173–180.

[18] R. Bogue. “Robots poised to revolutionise agriculture”. In: *Industrial Robot: An International Journal* 43 (Aug. 2016), pp. 450–456.

[19] Zhao De-An et al. “Design and control of an apple harvesting robot”. In: *Biosystems Engineering* 110.2 (2011), pp. 112–122.

[20] Johan Baeten et al. “Autonomous Fruit Picking Machine: A Robotic Apple Harvester”. In: *Springer Tracts in Advanced Robotics* 42 (July 2007).

[21] Giovanni Muscato et al. “A prototype of an orange picking robot: Past history, the new robot and experimental results”. In: *Industrial Robot: An International Journal* 32 (Apr. 2005), pp. 128–138.

[22] B. S. Lee and U. Rosa. “Development of a Canopy Volume Reduction Technique for Easy Assessment and Harvesting of Valencia Citrus Fruits”. In: *Transactions of the ASABE* 49 (2006), pp. 1695–1703.

[23] N. Kondo et al. “A Machine Vision System for Tomato Cluster Harvesting Robot”. In: *Engineering in agriculture, environment and food* 2 (2009), pp. 60–65.

[24] Henry Williams et al. “Autonomous pollination of individual kiwifruit flowers: Toward a robotic kiwifruit pollinator”. In: *Journal of Field Robotics* 37.2 (2020), pp. 246–262.

[25] Alistair Scarfe. “Development of an autonomous kiwifruit harvester”. PhD thesis. Massey University, 2012.

[26] Henry A.M. Williams et al. “Robotic kiwifruit harvesting using machine vision, convolutional neural networks, and robotic arms”. In: *Biosystems Engineering* 181 (2019), pp. 140–156.

[27] *Asparagus Market*. Tech. rep. Future Market Insights, 2017.

[28] C. W. Culpepper and H. H. Moon. “Changes in the composition and rate of growth along the developing step of asparagus”. In: *Plant Physiol* 14(4) (1939), pp. 677–698.

[29] *New Zealand Asparagus Council*. `http://asparagus.org.nz/`. Accessed: 10-06-2021.

[30] G. Lewis and C. Lewis. personal communication. May 6, 2016.

[31] M. Peebles et al. “Overview of Sensor Technologies Used for 3D Localization of Asparagus Spears for Robotic Harvesting”. In: *Applied Mechanics and Materials* 884 (2018), pp. 77–85.

[32] Stanford Artificial Intelligence Laboratory et al. *Robotic Operating System*. Version ROS kinetic kame. May 23, 2016.

[33] Redmond Shamshiri et al. “Research and development in agricultural robotics: A perspective of digital farming”. In: *International Journal of Agricultural and Biological Engineering* 11 (July 2018), pp. 1–14.

[34] Neil C. Thompson et al. *The Computational Limits of Deep Learning*. 2020.

[35] N. Marfu’ah and Arrie Kurniawardhani. “Comparison of CNN and SVM for Ship Detection in Satellite Imagery”. In: vol. 1. 2020.

[36] Parul Sharma, Yash Paul Singh Berwal, and Wiqas Ghai. “Performance analysis of deep learning CNN models for disease detection in plants using image segmentation”. In: *Information Processing in Agriculture* 7.4 (2020), pp. 566–574.

[37] José Naranjo-Torres et al. “A Review of Convolutional Neural Network Applied to Fruit Image Processing”. In: *Applied Sciences* 10.10 (2020).

[38] S. Ren et al. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. In: *Computer Vision and Pattern Recognition* (2015).

[39] Joseph Redmon et al. “You Only Look Once: Unified, Real-Time Object Detection”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 779–788.

[40] Joseph Redmon and Ali Farhadi. “YOLOv3: An Incremental Improvement”. In: *ArXiv* abs/1804.02767 (2018).

[41] Wei Liu et al. “SSD: Single Shot MultiBox Detector”. In: *CoRR* abs/1512.02325 (2015).

[42] Inkyu Sa et al. “DeepFruits: A Fruit Detection System Using Deep Neural Networks”. In: *Sensors (Basel, Switzerland)* 16 (2016).

[43] Suchet Bargoti and James Underwood. “Deep fruit detection in orchards”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 3626–3633.

[44] Longsheng Fu et al. “Kiwifruit detection in field images using Faster R-CNN with ZFNet”. In: *IFAC-PapersOnLine* 51.17 (2018). 6th IFAC Conference on Bio-Robotics BIOROBOTICS 2018, pp. 45–50.

[45] Fangfang Gao et al. “Multi-class fruit-on-plant detection for apple in SNAP system using Faster R-CNN”. In: *Computers and Electronics in Agriculture* 176 (2020), p. 105634.

[46] Longsheng Fu et al. “Faster R-CNN based apple detection in dense-foliage fruiting-wall trees using RGB and depth features for robotic harvesting”. In: *Biosystems Engineering* 197 (2020), pp. 245–256.

[47] S. Tu et al. “Passion fruit detection and counting based on multiple scale faster R-CNN using RGB-D images”. In: *Precision Agriculture* (2020), pp. 1–20.

[48] Kaiming He et al. “Mask R-CNN”. In: *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 2980–2988.

[49] Pengyu Chu et al. “DeepApple: Deep Learning-based Apple Detection using a Suppression Mask R-CNN”. In: *CoRR* abs/2010.09870 (2020).

[50] P. Ganesh et al. “Deep Orange: Mask R-CNN based Orange Detection and Segmentation”. In: *IFAC-PapersOnLine* 52.30 (2019). 6th IFAC Conference on Sensing, Control and Automation Technologies for Agriculture AGRICONTROL 2019, pp. 70–75.

[51] Yang Yu et al. “Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN”. In: *Computers and Electronics in Agriculture* 163 (Aug. 2019), p. 104846.

[52] A. Koirala et al. “Deep learning for real-time fruit detection and orchard fruit load estimation: benchmarking of ”MangoYOLO””. In: *Precision Agriculture* (2019), pp. 1–29.

[53] J.P. Vasconez et al. “Comparison of convolutional neural networks in fruit detection and counting: A comprehensive evaluation”. In: *Computers and Electronics in Agriculture* 173 (2020), p. 105348.

[54] Miles Hansard et al. *Time of Flight Cameras: Principles, Methods, and Applications*. Oct. 2012.

[55] A. Kolb et al. “Time-of-Flight Cameras in Computer Graphics”. In: *Computer Graphics Forum* 29.1 (2010), pp. 141–159.

[56] Manuel Vazquez-Arellano et al. “3-D Imaging Systems for Agricultural Applications - A Review”. In: *Sensors* 16.5 (2016).

[57] Péter Fankhauser et al. “Kinect v2 for mobile robot navigation: Evaluation and modeling”. In: *2015 International Conference on Advanced Robotics (ICAR)*. 2015, pp. 388–394.

[58] Elise Lachat et al. “Assessment and Calibration of a RGB-D Camera (Kinect v2 Sensor) Towards a Potential Use for Close-Range 3D Modeling”. In: *Remote Sensing* 7.10 (2015), pp. 13070–13097.

[59] Roanna Lun and Wenbing Zhao. “A Survey of Applications and Human Motion Recognition with Microsoft Kinect”. In: *International Journal of Pattern Recognition and Artificial Intelligence* 29.05 (2015), p. 1555008.

[60] Niyonsaba Eric and Jong-Wook Jang. “Kinect depth sensor for computer vision applications in autonomous vehicles”. In: *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*. 2017, pp. 531–535.

[61] Daichang Fu et al. “Automatic detection and segmentation of stems of potted tomato plant using Kinect”. In: *Sixth International Conference on Digital Image Processing (ICDIP 2014)*. Ed. by Charles M. Falco,

Chin-Chen Chang, and Xudong Jiang. Vol. 9159. International Society for Optics and Photonics. SPIE, 2014, pp. 18–22.

[62] Christopher Lehnert et al. "Sweet pepper pose detection and grasping for automated crop harvesting". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 2428–2434.

[63] A. Gongal et al. "Sensors and systems for fruit detection and localization: A review". In: *Computers and Electronics in Agriculture* 116 (2015), pp. 8–19.

[64] Longsheng Fu et al. "Application of consumer RGB-D cameras for fruit detection and localization in field: A critical review". In: *Computers and Electronics in Agriculture* 177 (2020), p. 105687.

[65] Jingyao Gai, L. Tang, and B. Steward. "Automated crop plant detection based on the fusion of color and depth images for robotic weed control". In: *J. Field Robotics* 37 (2020), pp. 35–52.

[66] Jordi Gené-Mola et al. "Multi-modal deep learning for Fuji apple detection using RGB-D cameras and their radiometric capabilities". In: *Comput. Electron. Agric.* 162 (2019), pp. 689–698.

[67] Joan Rosell-Polo et al. "Advances in Structured Light Sensors Applications in Precision Agriculture and Livestock Farming". In: *Advances in Agronomy* 133 (June 2015), pp. 71–112.

[68] Ji Li and Lie Tang. "Crop recognition under weedy conditions based on 3D imaging for robotic weed control". In: *Journal of Field Robotics* 35.4 (2018), pp. 596–611.

[69] Chee Kit Wong and P.P.K. Lim. "Processing of point cloud data from ToF camera for the localisation of ground-based crop". In: *International Conference on Mechatronics and Machine Vision in Practice, M2VIP 2012* (Jan. 2012), pp. 184–189.

[70] Manuel Vázquez-Arellano et al. "3-D reconstruction of maize plants using a time-of-flight camera". In: *Computers and Electronics in Agriculture* 145 (2018), pp. 235–247.

[71] David Reiser et al. *Crop Row Detection in Maize for Developing Navigation Algorithms Under Changing Plant Growth Stages*. Dec. 2016, pp. 371–382.

[72] M. Vázquez-Arellano et al. "Determination of stem position and height of reconstructed maize plants using a time-of-flight camera". In: *Comput. Electron. Agric.* 154 (2018), pp. 276–288.

[73] A. J. Matteoli. "Asparagus harvester". US Patent US2581119A. 1952.

[74] J. O. Turkington. "Asparagus harvesting machine". US Patent US2767544A. 1956.

[75] F. J. Chatagnier. "Asparagus harvester". US Patent US3066469A. 1961.

[76] D. E. Franzen. "Asparagus harvester". US Patent US3176456A. 1965.

[77] M. O. Marmorine and L. E. Lawson. "Asparagus harvester". US Patent US3328943A. 1967.

[78] L Lawson. "Asparagus harvester". US Patent US3717986A. 1973.

[79] S. K. Haws. "Stalk selective harvesting machine". US Patent US4003193A. 1975.

[80] S. K. Haws. "Machine for harvesting asparagus stalks and the like". US Patent US4064682A. 1976.

[81] S. Wang D. Chen Q. Zhang. "Current Status and Technical Challenges of Asparagus Mechanical Harvesting". In: *American Society of Agricultural and Biological Engineers (ASABE)* (2010).

[82] M. J. Moore. "Mechanical Harvesting - Asparagus". In: *American Society of Agricultural Engineers* (1964), pp. 261–266.

[83] G. Arndt, W. M. Perry, and R. Rudziejewski. “Advances in Robotised Asparagus Harvesting”. In: *Proceedings: 25th International Symposium on Industrial Robots* (1994), pp. 261–266.

[84] G. Arndt, R. Rudziejewski, and V. A. Stewart. “On the future of automated selective asparagus harvesting technology”. In: *Computers and Electronics in Agriculture* 16(2) (1997), pp. 137–145.

[85] W. J. Lund. “Asparagus harvester”. US Patent US4512145A. 1985.

[86] S. K. Haws. “Selective Harvester”. US Patent US20110126504A1. 2009.

[87] C. D. Clary et al. “Performance and Economic Analysis of a Selective Asparagus Harvester”. In: *Applied Engineering in Agriculture* 23(5) (2007), pp. 571–577.

[88] J. Strauß. *Development of an Automatic harvesting system for green asparagus with stalk detection in Ambient Light*. Tech. rep. Strauss Verpackungsmaschinen GmbH, 2014.

[89] A. Leu et al. “Robotic Green Asparagus Selective Harvesting”. In: *ASME Transactions on Mechatronics* 22(6) (2017), pp. 2401–2410.

[90] Martin Fischler and Robert Bolles. “Random Sample Consensus: A Paradigm for Model Fitting with Applications To Image Analysis and Automated Cartography”. In: *Communications of the ACM* 24 (June 1981), pp. 381–395.

[91] P. Baylou et al. “Detection and three-dimensional localization by stereoscopic visual sensor and its application to a robot for picking asparagus”. In: *Pattern Recognition* 17(4) (1994), pp. 377–384.

[92] D. S. Humburg and J. F. Reid. “A machine vision algorithm for identification of harvestable spears of asparagus”. In: *American Society of Agricultural Engineers* (1990).

[93] D. S. Humburg and J. F. Reid. “Field Performance of Machine Vision for the Selective Harvest of Asparagus”. In: *Journal of Commercial Vehicles* 100 (1991), pp. 81–92.

[94] P. Grattoni et al. “Automatic harvesting of asparagus: an application of robot vision to agriculture”. In: *Mobile robots* 8 (1994).

[95] P Grattoni and A Guiducci. “Contour coding for image description”. In: *Pattern recognition* 11(2) (1990), pp. 95–105.

[96] N. Irie et al. “Asparagus harvesting robot coordinated with 3-D Vision Sensor”. In: *IEEE International Conference on Industrial Technology* 1-3 (2009), pp. 408–413.

[97] N. Irie and N. Taguchi. “Asparagus harvesting robot”. In: *Journal of Robotics and Mechatronics* 26(2) (2014), pp. 267–268.

[98] H. Sakai et al. “Accurate position detecting during asparagus spear harvesting using a laser sensor”. In: *Engineering in Agriculture, Environment and Food* 6(3) (2013), pp. 105–110.

[99] “A Perception Pipeline for Robotic Harvesting of Green Asparagus”. In: *IFAC-PapersOnLine* 52.30 (2019). 6th IFAC Conference on Sensing, Control and Automation Technologies for Agriculture AGRICONTROL 2019, pp. 288–293.

[100] H. W. Yoo et al. “Real-Time Plane Detection Based on Depth Map from Kinect”. In: *IEEE ISR 2013* (2013), pp. 1–4.

[101] Jon Louis Bentley. “Multidimensional Binary Search Trees Used for Associative Searching”. In: *Commun. ACM* 18.9 (Sept. 1975), pp. 509–517.

[102] Josh Barnett. “Prismatic axis, differential-drive robotic kiwifruit harvester for reduced cycle time”. MA thesis. 2018.

[103] C. K. Au et al. “Workspace analysis of Cartesian robot system for kiwifruit harvesting”. In: *The international Journal of Robotics Research and Application* 47(4) (2020), pp. 503–510.

[104] Lingzhu Xiang et al. *libfreenect2: Release 0.2*. Version v0.2. Apr. 2016.

[105] Jonathan Huang et al. “Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 3296–3297.

[106] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015.

[107] C. Szegedy et al. “Rethinking the Inception Architecture for Computer Vision”. In: *Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 2818–2826.

[108] Tsung-Yi Lin et al. “Microsoft COCO: Common Objects in Context”. In: *Computer Vision – ECCV 2014*. Ed. by David Fleet et al. Cham: Springer International Publishing, 2014, pp. 740–755.

[109] M. Everingham et al. “The Pascal Visual Object Classes (VOC) Challenge”. In: *International Journal of Computer Vision* 88(2) (2010), pp. 303–338.

[110] Tzutalin. *LabelImg*. Git Code: https://github.com/tzutalin/labelImg. 2015.

[111] A. Savitzky and M.J.E. Golay. “Smoothing and Differentiation of Data by Simplified Least Squares Procedures”. In: *Analytical Chemistry* 36(8) (1964), pp. 1627–39.

Appendix A

A.1 Non-Geometric Approaches to Improve the CP Filter

The Kinect camera, utilised by the perception pipeline, does not provide access to the raw data frames which make up each ToF image. This limitation precluded certain non-geometric flying pixel filtering methods being investigated in detail. This section provides speculation on various non-geometric approaches which could be investigated in order to improve the CP filter.

One method of flying point filtering could be achieved by rejecting non-conforming raw frames prior to the integration step performed by the ToF camera. Such an approach would mitigate the effect of flying pixels significantly by limiting the influence of erroneous points from the eventually reported range average. However, whatever algorithm is to be implemented for identification and filtering out of erroneous frames would need to be aware of the features we are looking for, in this case asparagus spears. This is not an easy task and introduces a large potential for incorrect feature identification. It is questionable whether such an algorithm could be applied without significantly affecting the frame rate of the camera. Additionally, external factors such as the reflectively of the imaged surfaces, multi path errors and atmospheric disturbance can cause significant deterioration of the cameras range precision. The result can be significant noise in the range measurement of the camera, even when imaging flat and continuous surfaces. An algorithm that attempts to reduce the presence of flying pixels by thresholding discontinuous

range changes might struggle to accommodate situations such as this. This will likely require limiting the acceptable operating conditions of the camera such that the image quality is sufficiently high to isolate the kinds of discontinuous range changes expected at the edge of foreground elements. The kinds of scenes that need to be imaged in outdoor environments are dynamic, and vary significantly. Achieving ideal imaging conditions in these environments is extremely challenging.

Another method for removal of flying pixels could be to use traditional image processing techniques on either an RGB image, or the IR image provided by the Kinect camera. Sobel or Canny edge detection could be used on such images to provide a map of all hard edges in the scene. A perspective transformation between the original image, and the IR image provided by the Kinect camera can then be used to match those pixels in the edge map to their corresponding point in the pointcloud. If the Sobel/Canny operations provide a reasonable edge map for the scene then this approach would be very effective, however due to the variable lighting and unstructured environments this is likely difficult to achieve. Early testing of colour based binarisation, and edge detection methods applied to real world scenes proved difficult to tune. Often such edge maps are rife with random noise, and erroneous edge predictions. In such a scenario any point filtering built from this edge map would result in a large amount of points being incorrectly filtered. It is however possible that such a method could be used as a "first approach", as a way to limit the influence of further algorithms but such methods have not yet been investigated.

A.2 *frcnn_usaNet1* Configuration File

```
  model {
faster_rcnn {
  num_classes: 1
```

```
image_resizer {
  keep_aspect_ratio_resizer {
    min_dimension: 600
    max_dimension: 1024
  }
}
feature_extractor {
  type: "faster_rcnn_inception_v2"
  first_stage_features_stride: 16
}
first_stage_anchor_generator {
  grid_anchor_generator {
    height_stride: 16
    width_stride: 16
    scales: 0.25
    scales: 0.5
    scales: 1.0
    scales: 2.0
    aspect_ratios: 0.5
    aspect_ratios: 1.0
    aspect_ratios: 2.0
  }
}
first_stage_box_predictor_conv_hyperparams {
  op: CONV
  regularizer {
    l2_regularizer {
      weight: 0.0
    }
  }
```

```
  initializer {
    truncated_normal_initializer {
      stddev: 0.00999999977648
    }
  }
}
first_stage_nms_score_threshold: 0.0
first_stage_nms_iou_threshold: 0.699999988079
first_stage_max_proposals: 300
first_stage_localization_loss_weight: 2.0
first_stage_objectness_loss_weight: 1.0
initial_crop_size: 14
maxpool_kernel_size: 2
maxpool_stride: 2
second_stage_box_predictor {
  mask_rcnn_box_predictor {
    fc_hyperparams {
      op: FC
      regularizer {
        l2_regularizer {
          weight: 0.0
        }
      }
      initializer {
        variance_scaling_initializer {
          factor: 1.0
          uniform: true
          mode: FAN_AVG
        }
      }
```

```
        }
        use_dropout: false
        dropout_keep_probability: 1.0
      }
    }
    second_stage_post_processing {
      batch_non_max_suppression {
        score_threshold: 0.0
        iou_threshold: 0.600000023842
        max_detections_per_class: 20
        max_total_detections: 20
      }
      score_converter: SOFTMAX
    }
    second_stage_localization_loss_weight: 2.0
    second_stage_classification_loss_weight: 1.0
  }
}
train_config {
  batch_size: 5
  data_augmentation_options {
    random_horizontal_flip {
    }
  }
  optimizer {
    momentum_optimizer {
      learning_rate {
        manual_step_learning_rate {
          initial_learning_rate: 0.000300000014249
          schedule {
```

```
            step: 900000
            learning_rate: 2.99999992421e-05
          }
          schedule {
            step: 1200000
            learning_rate: 3.00000010611e-06
          }
        }
      }
      momentum_optimizer_value: 0.899999976158
    }
    use_moving_average: false
  }
  gradient_clipping_by_norm: 10.0
  fine_tune_checkpoint: "faster_rcnn_inception_v2_coco_2018_01_28/model.ckpt"
  from_detection_checkpoint: true
  num_steps: 200000
}
train_input_reader {
  label_map_path: "training/object-detection.pbtxt"
  tf_record_input_reader {
    input_path: "data/Train.record"
  }
}
eval_config {
  num_examples: 8000
  max_evals: 10
  use_moving_averages: false
}
eval_input_reader {
```

```
  label_map_path: "training/object-detection.pbtxt"
  shuffle: false
  num_readers: 1
  tf_record_input_reader {
    input_path: "data/Test.record"
  }
}
```

A.3 Robot Operating System (ROS)

ROS is a widely used, open source collection of tools, libraries and standards designed for the implementation of robotic systems. The main advantage of ROS is that it provides a robust messaging server which is useful for coordinating various sensors and overall control of the robot. ROS also facilitates multi-threaded execution automatically, providing a large performance boost with little programming effort.

A ROS network consists of nodes, topics, services and actions. Nodes are programs that run asynchronously, and usually on separate threads. These programs are used to implement the logic required to achieve various tasks. Nodes are organised into networks and communicate by sending messages under one of three paradigms; master-slave, client-server, or publisher-subscriber. The most common paradigm, and the one used for AHR-1 is the publisher-subscriber.

Nodes in a publisher-subscriber network do not communicate with each other directly. Instead, nodes communicate by constructing messages and labelling them with a name, called a topic. These messages are then published onto the network where subscriber nodes can listen for them on a specific topic. This paradigm is advantageous for robot implementation because each node operates independently , providing inherent stability to the overall program

flow. The independent operation also means that the system is inherently modular, aiding in development speed and robustness.

A ROS network can also contain services and actions. These constructs provide a way for nodes to request one-off actions to be performed and can be thought of as network-level functions. These functions can be called from within nodes, or via the command line. Services take in a number of arguments and provide some output. They are typically used for time-critical applications where the execution of a node depends on the output of the service call. Actions are used to trigger an asynchronous set of events that are typically used for longer, non-time-critical subroutines, such as for configuration or logging.

A.4 Effect of Field-Clutter and Camera Angles: (25° case)