Delft University of Technology
Master of Science Thesis in Computer Science: Software Technology

Object pose estimation for automation of plant extension in the greenhouses.

Matthijs Joost Wisboom

TU Delft Delft University of Technology

Object pose estimation for automation of plant extension in the greenhouses.

Master of Science Thesis in Computer Science: Software Technology

Embedded Systems Group
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Van Mourik Broekmanweg 6, 2628 XE Delft, The Netherlands

Matthijs Joost Wisboom
m.j.wisboom@student.tudelft.nl
matthijswisboom@gmail.com

09-06-2023

Author
Matthijs Joost Wisboom (m.j.wisboom@student.tudelft.nl)
(matthijswisboom@gmail.com)
Title
Object pose estimation for automation of plant extension in the greenhouses.
MSc Presentation Date
21-06-2023

Graduation Committee

Prof. dr. K.G. Langendoen	Delft University of Technology
Dr.ir. Y.B. Eisma	Delft University of Technology
It. S. Bosma	Lely

Abstract

The farming industry is undergoing a transformative shift towards efficiency and large-scale production. Manual labor is unable to meet the increasing demand, leading to the development of automation techniques, particularly in tasks like crop harvesting. While automating harvesting is an intriguing prospect due to its straightforward nature and tangible outcomes, other crucial tasks arise throughout a plant's lifecycle, depending on the plant type and growing environment. In greenhouses, plants such as tomatoes or bell pepper are grown vertically. As these plants grow, a constant working area is created that needs to be shifted down to keep producing crops. While most greenhouses use a string to suspend plants, a new Clipper method could ease the automation efforts for this task. This work investigates the feasibility of developing a system that detects and determines the position and rotation of clips in the plant extension process.

To achieve this feasibility, a neural network has been trained on a custom dataset with images containing 6-D pose annotations of a clip that do not require manual annotations. The designed system also executes a validation step with a second set of sensors in the form of a stereo camera. The entire methodology has been evaluated on a test setup of a greenhouse. The small and round properties of the clip cause a rotational error that does not fall in positional and rotational requirements set by a mechanical grabber on a robotic arm. Higher positional precision of annotation data and depth information from the stereo camera is necessary to make this methodology feasible.

Contents

Chapter 1

Introduction

The farming industry is evolving into an efficient, large-scale industry. With manual labor unable to cope with the growing demand, automation techniques for jobs such as harvesting crops are widely being developed [15, 16, 17]. Harvesting might be the most interesting task to automate as it is a reasonably straightforward task and yields concrete results. However, many other tasks are needed during a plant's life, depending on the plant type and growing environment. The greenhouse is one environment where proper plant care is essential. The Dutch greenhouse industry produces 10 times more vegetables per hectare than traditional outdoor farming methods [5]. To achieve this, plants like tomatoes, bell peppers, and cucumbers are grown vertically and are densely positioned next to each other. They can be grown any time of year, given the proper lighting and temperature in this sheltered environment, but they need to be handled with care so as not to cause any damage or infections to spread. These plants require constant attention to grow upwards optimally, allowing efficient crop growth along their main stem and optimizing their yield. Once the plants reach their desired height, they continue to grow but need to be lowered and moved to allow the new extension to grow new produce. This process alone needs to be executed regularly during a plant's lifespan and consumes many labor hours per week for a hectare of a greenhouse. As this process is repetitive yet skillful, automating the plant extension task would significantly reduce labor costs. This thesis will focus on automation efforts in the plant extension process of tomato plants, as this is the most commonly grown plant in Dutch greenhouses in 2022 [5]. More specifically, Section 1.1 will discuss the various methods currently used for plant extension with automation in mind. In Section 1.2, the challenges for automation in greenhouses are explained. After this, the problem statement will give a set of sub-questions to build on. This research is conducted at company Lely. The existing framework created there and other related work is discussed in Chapter 2.

1.1 Plant extension

Most Dutch greenhouses growing tomato plants do so using a hydroponics system. This efficient system uses a soil-less medium packed with nutrients that can be managed precisely for the plants to grow faster and give higher yields.

They are planted in horizontal rows above the greenhouse floor. The first stages of a plant's growth include reaching the top of the greenhouse and creating the desired density of plant stems. Depending on the plant extension method, vertical growth can be accommodated using various methods that use a suspended metal wire hung at the top of the greenhouse.

1.1.1 Hook & Wire

In the most common method used in manual labor, the plant stem is twisted around a rope that is hung from a suspended wire to force an upwards growth. This requires the careful guidance of the plant so as not to damage the stem or the growing produce. Once the stems have reached the desired height, they continue to grow, introducing an extra step besides twisting the plant around the wire. Periodically, the plant is lowered by giving more wire from a hook or spinner and moving it sideways on the suspension wire. A sketch of this setup can be seen in Figure 1.1 (left). This plant extension process needs to be applied to all plants roughly every two weeks. Even though this task is repetitive, the requirement of careful plant movement makes this task highly specialized to do quickly. With automation in mind, moving these stems like a skilled worker requires extensive sensing of the plants' structure. Weak points need to be identified to ensure that nothing on the plant will be damaged when handling the plant. For this reason, the Hook & Wire method will not be chosen to automate.

1.1.2 Clipper System

An alternative, systematic method is the Clipper system. This system only uses a rope in the first stage of the plant and is removed once the top is reached. Instead of twisting the plant, it is held up using two clippers attached to a metal rod that hangs off the top suspension wire, as seen in Figure 1.1 (right). A single clip in the greenhouse setting is shown in Figure 1.2. Once the plant needs to be extended, the following actions need to be taken:

1. The lower clip is removed from the rod.

2. The top clip is slid to the bottom of the rod.

3. A point on the newly grown stem is picked, and the first clip is clipped to the rod.

4. The rod is moved sideways on the suspension wire

In comparison to the Hook & Wire system, the clipper system is less reliant on a perfect understanding of its environment. In this system, the main challenges lie in detecting and positioning the clips and identifying a small section of the stem to place the clip back on. Moreover, automating this system will reduce the spread of viruses as a robotic system would only ever deliberately touch the clips and not the plants. For these reasons, the clipper system has been chosen as an automation strategy for the robotic system.

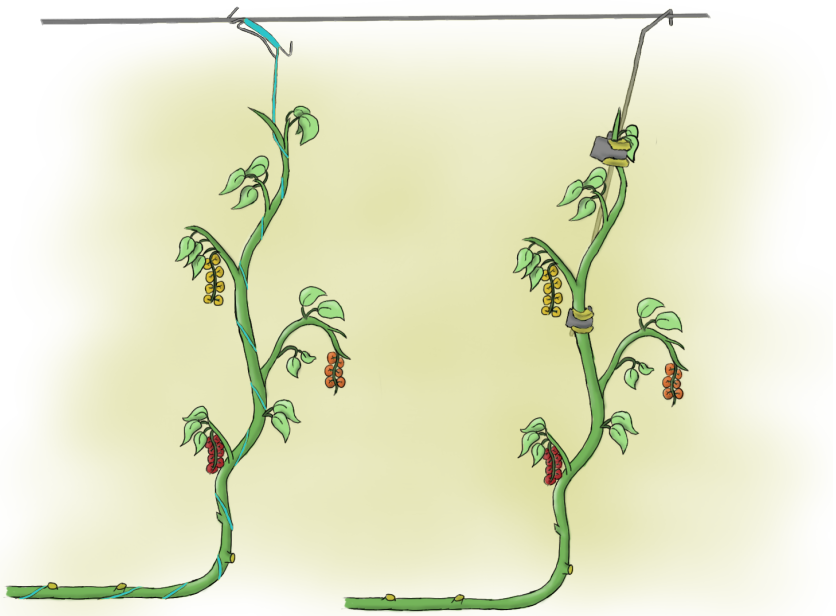

Figure 1.1: **Hook & Wire system (left) versus Clipper system (right).**

1.2 Environment

Each crop must be grown in specific environments to ensure the most efficient harvest possible. The greenhouse is built up in rows spaced 1.3 meters apart from each other [8]. As such, the distance to view the plant becomes relatively small. In the tomato crop, many different types can be grown. These types can vary in stem thickness, tomato size, leaf size, and brittleness, to name a few, which can affect the weight of the plant, its curvatures at the top, and the spacing between each plant. This diversity is important to take into account when developing any type of automation in the greenhouse. A typical greenhouse row is shown in Figure 1.3. The tubes that run along the floor of the greenhouse row are used for heating the greenhouse. As a second purpose, manual laborers use vertical carts on these rails to travel along the row and reach the top suspension wire of the greenhouse, which is situated roughly 4 meters above the ground. The clips are situated near the top of the greenhouse, which means lighting conditions can change drastically depending on the weather. Also, a grower can choose to use lighting to control photosynthesis. These lights are also not the same between greenhouses, varying between blue, red, white light, or a combination. These factors are taken into account in designing a strategy to automate the plant extension process.

Figure 1.2: **Clipper holding a tomato plant to a metal rod with measurements.**

1.3 Lely

The company for which this research is conducted is Lely [22]. Lely is known for its automation in the dairy industry, with its main campus situated in Maassluis. This area has many greenhouses growing tomato plants, and Lely developed an interest in greenhouse automation a few years ago. In recent years, they developed a robotic arm situated on a large vertical pole that drives across the greenhouse rails mentioned in Section 1.2 [27]. This setup is used to develop several end-effectors that can work on one plant at the same time. Lely strives to replace most of the tasks present in the greenhouse. This thesis will focus on the end-effector responsible for the plant extension using the clips mentioned in Subsection 1.3.1. The arm and the existing end-effector hardware are in the prototyping phase for this work to build on. The results of this work will be discussed in more detail in Chapter 2.

1.3.1 End-effector clipper

The end-effector that was developed for the robotic arm has been used for initial tests to gauge an accuracy requirement for the robotic arm to grab and place a clip successfully. The end-effector consists of three moving parts, an image of which can be seen in Figure 1.4. The main component is the gripper; this clamps the clip to be placed and grabbed from the rod and plant. The other two components are used to aid in this process. The Guider ensures the metal rod is positioned in the middle of the clip. So that when the gripper opens, the clip holds onto the rod. Lastly, the Pusher slides between the stem and rod to push the rod off the clamping beaks on the clipper. This is to ensure the clip does not get stuck on the rod.

1.4 Problem statement

The concept of automation of plant extension is relatively new, with no known working robot yet. Therefore, the methods to achieve this automation must be thoughtfully researched and tested. To take a clip off the metal rod with the current end-effector, there is a need for a method that is able to detect and estimate the position and rotation in a scene using a camera system. Therefore,

Figure 1.3: **Example of a Dutch greenhouse row**

this study focuses on the design and validation of such a method to investigate the feasibility of a system that automates the plant extension using the Clipper System. As such, this thesis will answer the following research question:

Is it feasible to develop a system that determines the position and rotation of clips in the plant extension process?

This thesis will answer this question by developing solutions to sense the clip and its environment accurately. Furthermore, this thesis will answer the following sub-questions:

- What are the existing technologies or systems available for object detection and pose estimation in a 3D environment?
- How accurate do the object detection and pose estimation algorithms need to be for successful clip grasping?
- What sensor technologies and data processing methods are most suitable for detecting and determining the position and rotation of clips in a greenhouse environment?
- What are the limitations of current object detection and pose estimation techniques in terms of their applicability to the greenhouse setting?

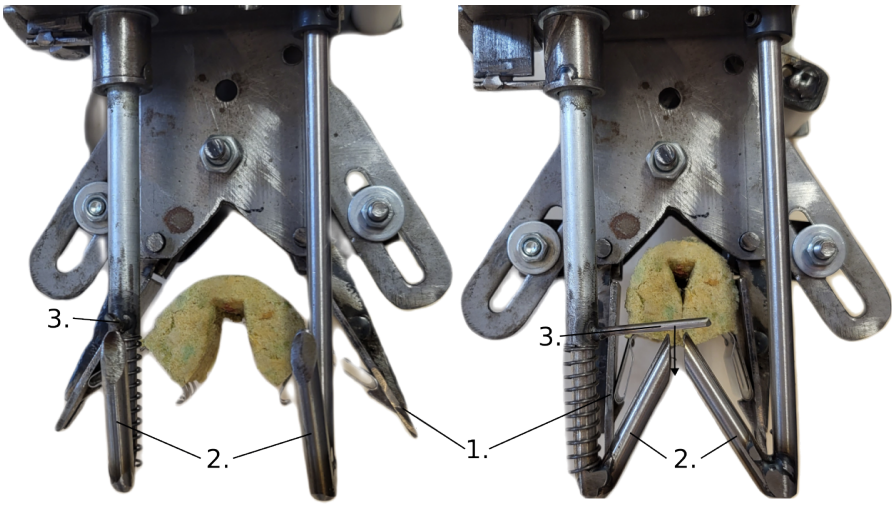

Figure 1.4: **Prototype clipper end-effector: open positions (left) vs. closed positions (right) of the Gripper (1), Guider (2), and Pusher (3) mechanisms controlled using servo motors.**

During this thesis, a detecting and positioning pipeline has been created to evaluate the effectiveness of the methods. This pipeline can be deployed in environments resembling those described in Section 1.2, which includes test environments like an office setup.

Chapter 2

Related Work

Greenhouse automation has become an increasingly important area of research due to the need for sustainable and efficient food production. An essential aspect is computer vision techniques for object detection, tracking, and classification. However, detecting the plant extension clips can be challenging due to the occlusion caused by the greenhouse environment. This chapter provides a review of related work in greenhouse automation and object detection in occluded environments by examining the current state-of-the-art techniques. These insights will be valuable for developing a detection pipeline that helps automate the plant extension process.

2.1 Lely technologies

Although Lely has not yet entered the market of greenhouse automation, their efforts in developing a robotic arm have already come a long way. Recent research and development in 2022 have resulted in a prototype setup that will serve as the basis for this project [27]. This section will give an overview of the capabilities of this robotic arm.

2.1.1 Philosophy

As mentioned in Section 1.3, Lely aims to automate multiple tasks. In contrast to developing a separate robot for each task, this project aims at creating a modular system that can do multiple tasks at once using multiple robotic arms. Multiple arms are attached to a pole stretching to the top of the greenhouse. Each arm can move independently of one other or work together on a single task. As an example, the pole might have three arms, the top arm being responsible for the plant extension process, the middle for removing unwanted stems, and the lowest for removing stems below the growing area.

2.1.2 SCARA

To achieve this philosophy, each arm is essentially manufactured in the same way, apart from a specialized end-effector. A Selective Compliant Articulated Robot Arm, or SCARA for short, is chosen as the design. This type of arm allows compliance in the X-Y directions but is rigid in the Z direction. So for

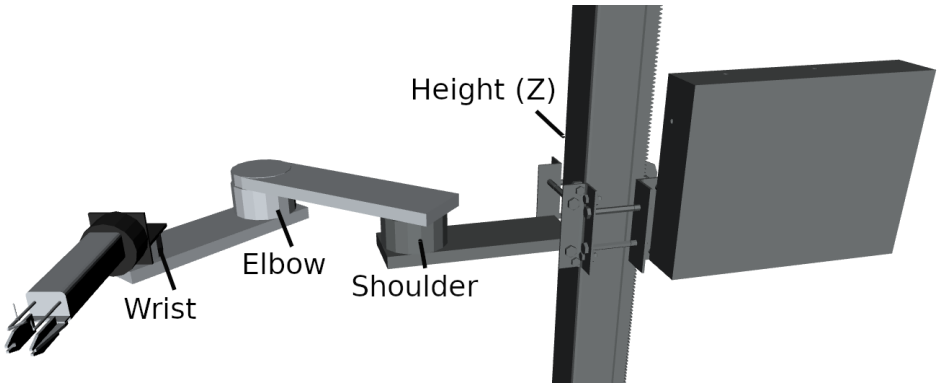

Figure 2.1: **SCARA design with 4 joints. The end-effector is attached to the wrist, with additional pitch or roll control depending on the application.**

	X	Y	Z
Location (mm)	14	30	5
Rotation	15°	15°	20°

Table 2.1: **Maximum allowed position and rotation difference to actual clip transformation.**

vertical movement, the pole is used. This type of design minimizes the vertical clearance needed when two arms are working in close proximity while retaining reachability in the greenhouse. An overview of the joints is shown in Figure 2.1. Each arm houses a computational unit that computes the inverse kinematics as well as any processing of sensing equipment.

2.1.3 Required accuracy

The arm has been evaluated using the end-effector clipper from Subsection 1.3.1. With a fixed rod location of the clipper known to the arm, the catching accuracy is 100%. However, when the clipper location or orientation is not precise enough, it can cause the mechanism to miss. The developed arm and end-effector can successfully grab a clip within the location and rotation requirements found in Table 2.1. These requirements will serve as a threshold to evaluate the feasibility of the design that will be discussed in Chapter 3.

2.2 Sensing technology

Determining the position and rotation of the clip is, as indicated by the previous section, a crucial part of taking the clip from the rod. This section will explore techniques used in other works to sense their environment and/or to find and locate objects of interest.

2.2.1 Sensor hardware

In order for a robotic system to function in a non-static environment, it needs to have data of its surroundings. A color camera is a good way to capture the surroundings of the system due to the wide range of information it has. Determining the position of anything with a single camera is often used in systems where objects are situated on a plane with a known depth or are always the same size. For the clipper application, the object to be found is always the same size. Thus, in theory, a single color camera can be sufficient to accomplish the task. However, to improve accuracy and deal with the surrounding plants that can vary in size dramatically, extra depth information is highly desirable. Previous works for greenhouse automation mostly use a Binocular setup with a camera-in-hand [29] to obtain an RGB-Depth image (RGB-D). This means the cameras are positioned near the end-effector and can utilize both Visual-based servoing and open-loop visual control. Stereo cameras can offer a high resolution and are relatively low cost. Other depth-sensing methods like Time-of-Flight or LiDAR are also valid options to consider. These laser sensors have higher accuracy and do not rely as much on features and lighting conditions as stereo vision, but they are more expensive, less modular, and have a lower resolution. While there are certainly options to consider for different applications, the choices are highly reliant on the methods and requirements.

2.2.2 Farm Technology Group Wageningen University

One source of inspiration is the Dutch Wageningen University (WUR), which introduced the SWEEPER bell pepper harvester [1]. This harvester scans each plant for crops ready to be harvested using a calibrated color threshold that can be set for each session, with shape size constraints as extra reject criteria. To mitigate ambient illumination effects, the robot applies a Flash-No-Flash (FNF) controlled illumination acquisition protocol [2]. The depth of the fruit is then determined using an RGB-D pixel-to-world transformation to calculate the center point of the pepper. After the location of the pepper is known, the end-effector needs to determine the rotation at which to cut off the pepper. To do this, the stem is detected using a deep neural network segmentation network and processed using a Canny edge detector and straight line detection using Hough transform. Using two viewpoints, there is enough information to position the robot such that the stem is behind the fruit, as seen from the camera. This pipeline achieved an accuracy of 73% with an acceptance of 25 degrees provided by the end-effector specifications and an average harvesting speed of 24 seconds per pepper. For this pipeline to work for the clipper application, both the stem and rod need to be segmented and located to determine the rotation of the clipper, with an alternative system needed to detect the clips. Achieving higher accuracy while working with smaller distances in the clipper system will be difficult but not impossible.

2.2.3 Harvey

Another bell pepper harvester is a robot called Harvey from Queensland University of Technology [19]. This robot employs a different method in the pose estimation of the cutting point. A deep neural network provides a segmenta-

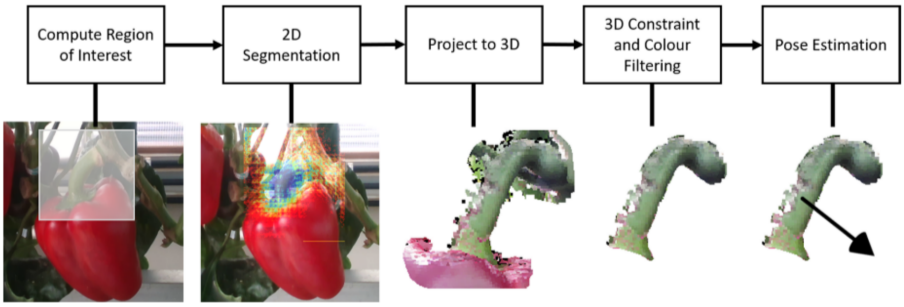

Figure 2.2: **Pipeline for bell pepper peduncle reconstruction from the Harvey robot [19]**

tion of both the bell pepper and the peduncle, which is the stem on which the pepper grows. Using an RGB-D camera, a 3D reconstruction of the peduncle is constructed, and its center point is calculated and used as the cutting point. The angle at which the robot should approach is calculated using an estimation of the normal vectors on the chosen cutting point. Figure 2.2 shows the pipeline of this process from their paper. Using this method, a maximum accuracy has been achieved of 84% in the detection of peduncles with 77% successful harvests. Although this method picks an arbitrary point on the peduncle that determines the rotation of the arm, objects can be segmented and reconstructed in 3D for further processing to give a better understanding of the environment. This will be an idea for reconstructing the clip in the environment to determine an approach angle.

2.2.4 Object Pose Estimation

Robotic solutions for object detection outside the greenhouse environment are also widely being researched and tested. Many modular systems use similar strategies, as discussed earlier, to identify a gripping point [25, 23]. However, in the clipper system, there is only one valid gripping point, and this is always in the same orientation relative to the clip. Therefore, the gripping point angle can be determined when the exact transformation is known. While much less perfected than detection and segmentation networks, various works have gone into 6-dimensional pose estimation of all kinds of objects. To encourage research in pose estimation and set up a uniform dataset format, Hodan et al. [13] published a benchmark for 6-Dimensional pose estimation of rigid objects from a single RGB-D input image (BOP). This platform allows methods to be tested with identical datasets, with a variety of objects and different levels of occlusion. Since 2018, the publishers of the benchmark have run a BOP Challenge for methods to be compared against each other. These challenges require methods to be run on known validation data as well as unkown validation data on fresh datasets; a leaderboard[1] shows the scores of the results on each dataset.

[1] https://bop.felk.cvut.cz/leaderboards/

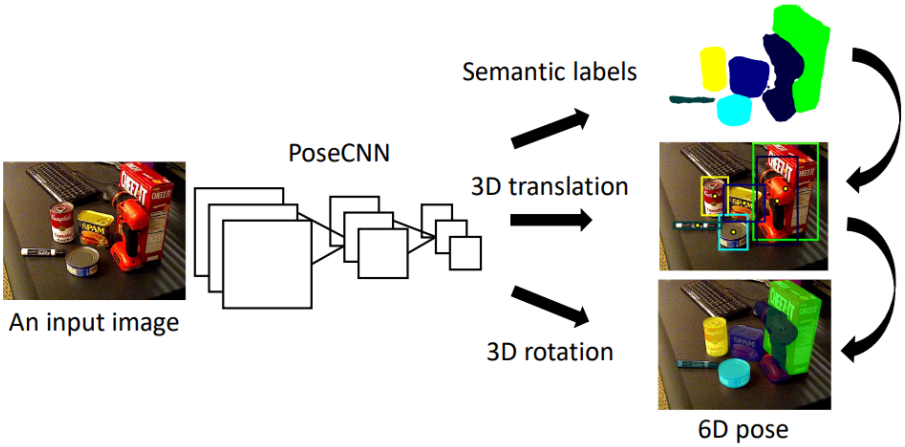

Figure 2.3: **PoseCNN for 6D object pose estimation, where the network is trained to perform three tasks: semantic labeling, 3D translation estimation, and 3D rotation regression [28].**

PoseCNN

One approach that many implementations consider their backbone structure is PoseCNN [28]. PoseCNN is a Pose Convolutional Neural Network. It combines template-based methods and feature extraction to gather helpful information in an image and feed this into a deep neural network. Based on a 3D object model, the network is trained on translation matrix $\mathbf{T}$ and rotation matrix $\mathbf{R}$. As Figure 2.3 shows the pipeline, semantic labeling is first applied to the feature map that produces confidence scores on each pixel for an object. Next, 3D translation estimation first uses regression to find the object center point using camera intrinsics. It then combines this with the semantic labeling in a Hough voting scheme to vote the final estimation of the object centroid in the image. Due to this setup, where all object pixels are considered, the center of the object can be occluded. Finally, the rotation is regressed using the translation gained from the last step and a loss function that computes the distance between the annotated and predicted rotation. Notice that this method does not require depth information to be available as it trains on the annotation data. In this paper, depth is an optional addition for the network to extract more features from but can be run without. Other works use the depth information to perform a final alignment during or after regression is finished to fit an object model into the 3D pixel points (Point cloud) and adjust regression accordingly [20, 18].

As the robotic arm with a camera setup could return relative positions and rotations from the camera, it can run parallel to other tasks as long as the processing power on the arm can handle the computation. Training a network like this on a specialized object requires some effort, but results may work well with a wide range of environments and lighting conditions.

2.2.5 Winner BOP 2022

While PoseCNN from the previous section has set up a great baseline, it is already relatively old and inaccurate compared to other state-of-the-art methods. Prior Geometry Guided Direct Regression Network (PGDRN) [21] was the winner of the most recent BOP Challenge in 2022. This network largely differs in the methods used to detect and match features. In this paper, a detailed 3D reconstruction of the object is used to compute prior features to compare to the input image. This image is first fed into a 2D Detection network to create a box around the object, after which color features are extracted using a ResNet [11] backbone with deconvolution layers and upsample layers. The color image and 3D model features are concatenated to end up with a prior-color feature map. This is then used to predict object regions and coordinates to regress the final transformation values. This method heavily relies on the 3D reconstruction of the object. Next, the network performs best when trained on large amounts of data, as other networks submitted to the BOP Challenge. The datasets in the BOP challenge are not very large, with around a thousand images per object. However, this paper, as well as many others, uses physically based rendering (PBR) techniques to acquire more training data [14]. All datasets in the BOP challenge have objects laid out on a flat surface. Using the PBR method, the scenes for which to train can be recreated using simulated gravity. Creating a rendered environment for the greenhouse, which is random, is difficult to achieve and requires large computing power. While this method works best on all datasets provided to the BOP challenge, it does not inherently mean this method will work best for the clipper system when rendered annotation data is not available.

2.3 Discussion

The greenhouse is an environment with many unknowns that require adaptability and redundancy. Even though the clipper is a rigid object, no assumptions should be made about the object's location, as damaging a plant might mean plant loss or infection spread. The current methods for environmental awareness, especially in the greenhouse, have come a long way but are not perfect. If the clipper method is to be feasible, the robot needs to achieve high accuracy to be able to replace a laborer. As the clipper method is not standard in a greenhouse yet, training networks like those discussed in Section 2.2.4 will need a setup for annotation of position and rotation in the greenhouse to achieve high-accuracy results for a feasible solution. Choosing a suitable Pose estimation network is a challenging task, as the benchmarks that have been set do not resemble the greenhouse environment in any way. The methods discussed also do not verify the prediction with an extra set of sensors which is an important aspect, as a wrong prediction can cause plant loss.

Chapter 3

Design

The robotic arm that Lely has developed serves as a starting point to tackle the clip extension process. While the entire clip extension process is interesting on its own, the current end-effector described in Section 1.4 is too long for the greenhouse to be able to maneuver between the plants. Tomato stems are separated between 20 and 40 cm and would, therefore, not fit the current 30 cm end-effector. For this reason, the feasibility of achieving the required location and rotation accuracy of the clip from Table 2.1 first needs to be reached before improvements should be made to the size of the end-effector. This chapter will show the design that locates the clips in terms of rotation and position in 3-D space. The first section will describe and provide a systematic overview of the system. Then, the vision pipeline and neural network that have been set up will be explained.

3.1 Systematic overview

Building upon the baseline robotic arm from Section 2.1, this section provides a systematic overview of the system that is developed. This section will first look at the separate components in the system, providing the communication network for a single arm. The camera setup will then be discussed, followed by how the image data is parsed and processed on the central computation unit.

3.1.1 Hardware

The hardware design of the entire system can be split into two parts. While this project only accounts for one robotic arm, the entire robotic system will mount multiple arms that each perform their own tasks. These arms are all positioned along one central cart pole that can move horizontally across the greenhouse rails. This cart is one of the two parts. However, since this part of the system is only used for creating stability for the arms and moving from plant to plant, it falls out of scope for this project. The focus here lies on the extension of a single plant and will, therefore, not be discussed any further. The second part is the robotic arm. This arm operates a total of 5 motors for positional control using inverse kinematics to position the end-effector to any position it is able to reach. The motors, as well as the end-effector, are controlled over a Controller

Area Network (CAN bus) with an M7 microprocessor to handle communication. The M7 serves as a middleman between the CAN components and the central command running Linux and Robotic Operating System 2 (ROS2). This is a flexible distributed framework used for the development of robotic systems. ROS2 allows for multiple components called nodes to interact with one another and be deployed on multiple machines across a network. Unlike its name, ROS2 is not an operating system but rather a framework with a set of tools on top of an installed Linux distribution. An NVIDIA Jestson is used as the central command unit for the arm. This module has all the necessary components to test and deploy various programs and is able to run large neural networks. With this system, there are little to no limitations in computation power for the development of the arm. The arm can move freely up and down the cart pole. This is because the communication is handled over a Local Area Network that is set up on the cart. Communication with the cart or any other device, such as a manual controller, is done over this network. Power to the system comes from four 12V batteries configured in series to obtain 48V that is required for the multiple motors. This voltage is then converted for the other components in the system. Power is delivered to each arm using two copper rails and sliding contacts to eliminate the need for a cable harness from the cart to the arm that may tangle up with the plants. An overview of the electrical components in the system can be seen in Figure 3.2.

3.1.2 Camera Setup

The camera setup that has been chosen for this application involves three cameras positioned on the wrist of the robotic arm from Section 2.1. This position has been chosen to view plants from multiple angles and can utilize active visual servoing when approaching clips. Two cameras are used to construct a three-dimensional map (point cloud) of the scene using a stereo-matching approach. These cameras are placed at a fixed distance from each other, called the baseline, and an image is taken from both simultaneously. A distortion matrix is applied to both images such that points in the world lay on one line in the real world. The two lines from both images can then be used for feature-matching, which is an algorithm matching pixels from left and right images. The 3D point can then be obtained using trigonometry. When objects are close by, the horizontal shift from one image to the other becomes greater and therefore requires a more extensive search area for feature matching. This causes the feature matching to produce fewer matches as near objects will be viewed with different angles, which causes the same features to not appear in both images. One way to solve this problem is to shorten the distance between the two cameras. This causes more corresponding features between the two images as the two images are more similar, and therefore more points can be calculated for the point cloud. The working area of the arm is relatively short, with the distance from the arm to the plant being less than 30 centimeters. This is while most off-the-shelf stereo cameras are designed for distances with a minimum of 30 centimeters. For this reason, a custom stereo-vision setup has been constructed using the OAK-FFC-3P modular baseboard from Luxonis [3]. The setup consists of two global shutter grayscale 1-megapixel cameras with a horizontal separation of 25-millimeter. The setup achieves a significantly higher density of the point cloud in close-range objects compared to a pre-configured 75-millimeter OAK-

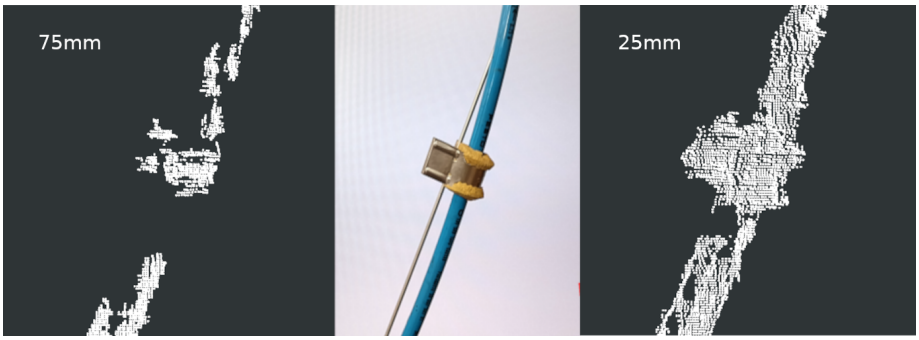

Figure 3.1: **3D points determined using stereo-matching with a baseline of 75mm (left) versus 25mm (right) on a single clip with a fake stem. Points further than 50 cm are not included to exclude the background.**

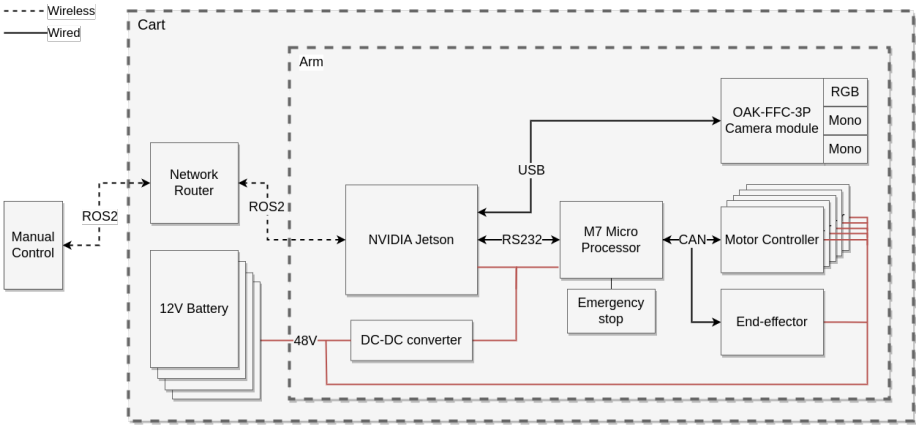

Figure 3.2: **Schematic overview of the electronic components with communication types**

D option, with on average 128 % additional points in the point cloud, a visual difference is shown in Figure 3.1. For additional information, a 12-megapixel color camera is positioned 25 mm left of the stereo pair. The modular baseboard is equipped with a Robotics Vision Core 2, which can perform actions such as stereo-matching to obtain a point cloud or run any AI model if converted to the right format. The ability of the camera to do its own calculations means the main computation unit can allocate more of its resources to other tasks.

3.2 Annotation pipeline

The system that will be used involves the use of a neural network to cope with the adaptability requirement of the greenhouse. An annotation pipeline is an essential component in the development of neural networks, especially for objects that are not in any dataset yet. Annotating is the process of labeling the input data to provide a neural network with ground truth information. However, the annotation process can be time-consuming and therefore expensive, making it

Figure 3.3: **Items are placed randomly on a table with a surrounding ring of markers that register an object's relative position for the LINEMOD dataset [12].**

challenging to obtain large amounts of accurately labeled data. This challenge is particularly significant when dealing with pose estimation. While 2D segmentation annotators can manage a few hundred annotations in a day with specialized tools, scaling this up to three-dimensional, including the object rotations, significantly increases the time needed for annotation. As a reference, the BOP datasets discussed in Chapter 2 consists of about a thousand real images, with some approaches adding more than ten thousand procedural generated images. Manually generating training data is both time and accuracy-wise not viable. There is a need for a pipeline to automate the process.

3.2.1 Scene annotations

The way most 6-D pose datasets are annotated is achieved in scenes. A scene consists of an object that needs to be annotated and a set of ARTags [9], often implementing ArUcO [10]. These markers are typically placed on a table in a circular pattern to ensure at least one marker is always visible when viewing objects from different angles. An essential factor between scenes is that the relative transformation between the object and marker is not constant. The reason for this is to negate any correlation between the markers and the object when training for a neural network. An example of this setup can be found in the LINEMOD dataset, Figure 3.3, which is the most common dataset to validate 6D Pose estimation networks on. The markers have a known length,

and their position and rotation relative to the camera can be calculated using the camera intrinsics. This information is useful when paired with the relative transformation from marker to object. For the duration of a scene, which in LINEMOD is around 50 images, only one annotation for an object is necessary to calculate the camera transformation for all other images in the scene. This method improves the speed of annotation dramatically and ensures a consistent measurement. As an addition to this annotation pipeline, a registration ring has been developed to ease the annotation even further. This ring consists of 5 ArUcO markers in a half-circle configuration, as shown in Figure 3.4. The ArUcO board to place in the scene consists of a grid of two by four to ensure the camera sees at least part of the markers to register its 3D transformation. The measurement becomes less accurate when the markers become smaller or fewer markers are present. This becomes specifically apparent when the single markers on the ring are less visible to the camera. The methods for detecting the markers use segmentation on the borders and inside patterns. An additional transformation to each marker position and rotation to the center of the clip is applied to return the same pose depending on the relative marker placement on the ring. When the markers become difficult to segment due to extreme angles or low lighting, the estimated position and rotation will be off. For this reason, a moving median of 10 measurements across multiple frames is taken for all visible ring markers to achieve an accurate pose from the combined markers.

Once the relative transformation from the ArUcO board to the ring is saved, the ring can be removed. Small magnets are embedded inside the ring that snaps on the clip and can be easily removed without the clip moving from its position. During annotation, the image, transformation from camera to clip, camera intrinsic, and depth are processed into the BOP format from Subsection 2.2.4. This step also generates a mask of the clip and a mask based on the clip visibility in the depth image that is used for training 2D classification. Using this pipeline, a dataset has been created using a variety of locations that offer different lighting conditions on and around the clip. Around three thousand annotations are generated across fifty scenes.

3.3 Pose Neural Network

Now that a pipeline has been set up to gather data, it is time to train a neural network on this data. The neural network must be able to detect and give rotation and translation vectors relative to the camera. For the clip application, several factors come into play. The network should not penalize the symmetric properties of the clip. Neural networks are evaluated based on a function, so this function must return the same score when rotated by 180 degrees around its symmetrical axis. Another factor to consider is the amount of training data that is needed to train these networks. Even though the annotation pipeline from Section 3.2 is efficient, the amount of training data most of the networks from Section 2.2.4 need up and around fifty thousand annotated images. These networks use procedurally generated images with only a portion of real images. With highly detailed 3D models, a virtual environment is created to train on. These environments replicate the position and occlusion that are found in the real world using physics simulation. However, creating a procedurally generated greenhouse setting that is realistic enough for training is outside of the scope

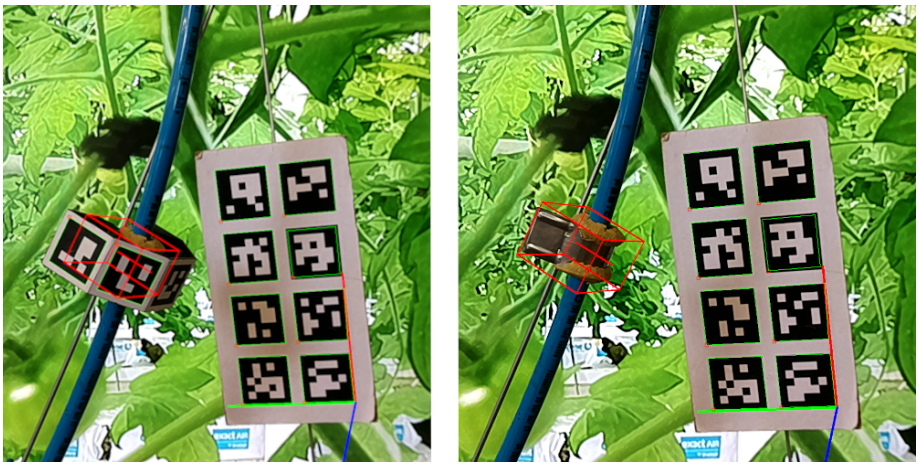

Figure 3.4: **Aruco ring that is attached on the clip using magnets to register the relative transformation to the AruCo board. After this, the ring is removed. An annotation can then be created if the board is in view.**

of this project. The chosen neural network, therefore, only uses real images for training.

3.3.1 EfficientPose

The neural network that has been chosen is EfficientPose [4]. This network is based on a 2D classification and bounding box detection network called EfficientDet [26]. The EfficientPose network extends EfficientDet to add two subnetworks to predict the rotation $\mathbf{R}$ and translation $\mathbf{t}$. Due to the small size of these subnets and their utilization of shared input feature maps with the existing classification and bounding box subnets, the added computational overhead is negligible. To combat the need for large datasets, the training of this network employs 6D augmentation on the real images in order to increase the amount of training data and help to converge the network to a more general solution. This network can add extra rotation, scaling, and shearing on the image and annotation to generate more data. A small error is introduced when objects are not in the center of the image. As the camera perspective on the 3D object changes, so does the projection onto the 2D image plane. However, the neural network has shown that the benefits from the additional data obtained with this 6-D augmentation outweigh the introduced error. These images are then also color transformed, like adjusting the contrast and brightness of the input image, which does not affect the annotated data.

For evaluating transformations, points (vertices) in a 3D model are used for evaluating the position. This 3D model has been created using Blender [7] and contains around three thousand vertices to depict the clip accurately; Figure 3.5 shows a render of the object. A common metric ADD(-S) [12] is used for evaluating the model. The average point distances between the 3D model point set $\mathbb{M}$, transformed using the ground truth $\mathbf{R}$ and $\mathbf{t}$, and the model point set transformed with the estimated $\tilde{\mathbf{R}}$ and $\tilde{\mathbf{t}}$, are calculated by this metric in Equation 3.1.

Figure 3.5: **A 3D model is used to evaluate the accuracy of the prediction. The model is mirrored along the horizontal and vertical axis. The model with stem (right) is used as a refinement step on the camera point cloud.**

Equation 3.2 displays the metric for symmetric objects. This metric finds the vertices with the lowest distance to the ground truth rather than matching the vertices between truth and predicted one-to-one. It has been found, however, that with the clip application, small errors induced in the annotation or augmentation step cause the shortest vertices between prediction and truth not to be the correct or symmetric counterpart. Therefore, the average over all these distances will cause a wrong convergence. To remedy this, only the minimum distance between the ADD measurement and ADD with the 180° rotated truth model is taken. This gives a final metric in Equation 3.3 where $\mathbf{x_{1m}}$ is the mirrored point in X (horizontal) and Z (vertical) direction. An estimate is considered correct if the average point distance is smaller than 5 millimeters, which is 10% of the diameter of the object.

$$\text{ADD} = \frac{1}{m} \sum_{x \in \mathbb{M}} \|(\mathbf{R}\mathbf{x} + \mathbf{t}) - (\tilde{\mathbf{R}}\mathbf{x} + \tilde{\mathbf{t}})\| \tag{3.1}$$

$$\text{ADD-S} = \frac{1}{m} \sum_{\mathbf{x_1} \in \mathbb{M}} \min_{\mathbf{x_2} \in \mathbb{M}} \|(\mathbf{R}\mathbf{x_1} + \mathbf{t}) - (\tilde{\mathbf{R}}\mathbf{x_2} + \tilde{\mathbf{t}})\| \tag{3.2}$$

$$\text{ADD-M} = \frac{1}{m} \sum_{\mathbf{x_1} \in \mathbb{M}} \min_{\mathbf{x} \in \{\mathbf{x_1}, \mathbf{x_{1m}}\}} \|(\mathbf{R}\mathbf{x_1} + \mathbf{t}) - (\tilde{\mathbf{R}}\mathbf{x} + \tilde{\mathbf{t}})\| \tag{3.3}$$

3.4 Pointcloud refinement

After the model has made a prediction from Section 3.3, the stereo cameras are used as a validation of the prediction. The camera setup from Section 3.1.2 publishes a point cloud. This is a set of 3D points that show the depth of the camera view. Picking out a plant extension clip from just this information is a hard feat, and therefore additional information from the neural network is used to focus a search attempt. Once a pose is returned from the neural network, a 6x6x6 centimeter box is taken from this position, which is just larger than the clip itself and uses all points within this box for an Iterative Closest Point approach (ICP) [6]. This ICP algorithm iteratively matches two point

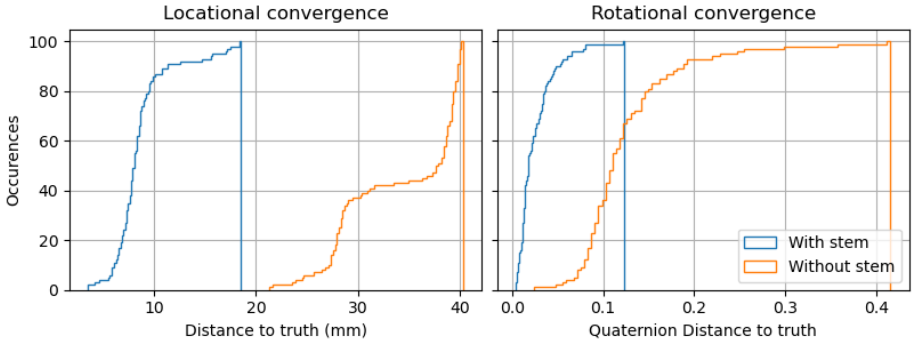

Figure 3.6: **Difference in the convergence between only the clip model on a point cloud and the clip model with a stem. This figure shows cumulative distances to the truth position and rotation of the clip. Both run from the truth position and rotation as the initial (predicted) pose. Distance is taken as Euclidean distance and rotation as Quaternion distance from Equation 4.1.**

clouds so that the combined distance between the first point cloud, the source frame, and the closest points in the second point cloud, the target frame, is minimized. For the greenhouse application, a point cloud is converted from the 3D model from Section 3.3 with the addition of a piece of stem to help converge to the correct vertical alignment. The clipper application does not need to find clippers not attached to a stem, so this assumption of all clips having a stem can be made. Without this stem, the round nature of the clip model causes large rotational errors to occur when running the algorithm on a noisy point cloud, as can be observed in Figure 3.6. The 3D model is positioned on the predicted transformation of the neural network and acts as the target frame for the ICP. The box point cloud is set as the source frame and is fitted onto the target frame. After the algorithm converges to a solution, the inverse transformation is applied to the output of the neural network prediction to reach the final transformation of the clip.

Chapter 4

Results

Using the design described in Chapter 3, a series of tests have been conducted to determine the accuracy and confidence of the system. This chapter will first examine the natural error of the camera system. Next, the neural network live prediction, or inference for short, is put to the test to see how accurate the initial guess's overall position is. After this, these results are compared to the Iterative Closest point approach that is run on the inference to view any correlation between the two.

4.1 Test setup

To measure the performance of the methods described in Chapter 3, a controlled test setup has been constructed that simulates the greenhouse setting. An image is taken from multiple arm locations on which the truth transformation of the clip is measured using the annotation ring and the inference. Lastly, an ICP refinement is run on the inference transformation on the point cloud, as described in Section 3.4. All measurements are transformed into a world frame so that the measurements from different arm locations can be combined and inconsistencies in the arm or camera can be spotted. The axis for rotation is measured in quaternions as the distance between two rotations and can be calculated using Equation 4.1, ranging between 0 (identical) and 1 (rotated 180 °). B_m is the mirrored point in the X (horizontal) and Z (vertical) direction to keep the symmetry of the clip into account. X, Y and Z define the location of the world coordinate. Where X is the axis that runs along the greenhouse row, Z is vertical, and Y is the depth from arm to plant. A top-down view of the set setup can be seen in Figure 4.1.

$$R_\Delta(A, B) = \min_{b \in \{B, B_m\}} 1 - (A_x \cdot b_x + A_y \cdot b_y + A_z \cdot b_z + A_w \cdot b_w)^2 \tag{4.1}$$

Distance is calculated using Pythagoras from Equation 4.2

$$L_\Delta(A, B) = \sqrt{(A_x \cdot B_x)^2 + (A_y \cdot B_y)^2 + (A_z \cdot B_z)^2} \tag{4.2}$$

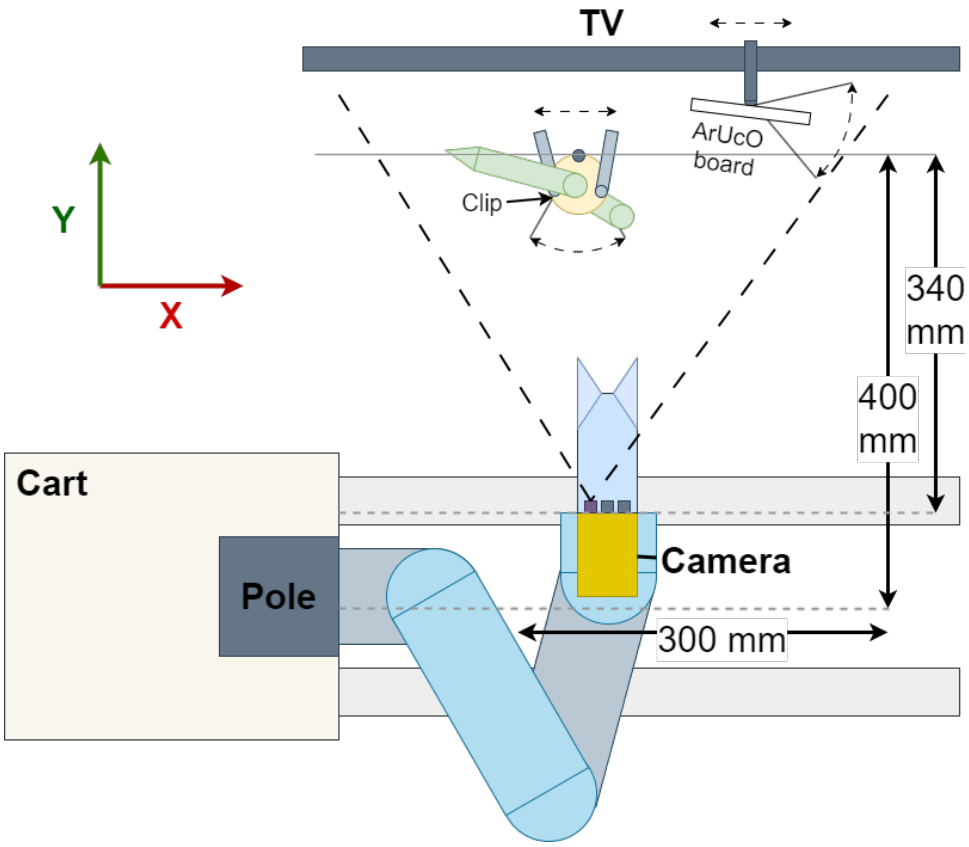

Figure 4.1: **Top-down view of the test setup. The clip and ArUcO board are placed in different positions and orientations per scene. Measurements are taken from 10 arm positions, 5 from 340 mm, and 5 from 400 mm distance to the clip. Arm positions are evenly spaced 300 mm in the X direction, with the camera pointing at the clip.**

4.2 Truth accuracy

As mentioned earlier, the arm takes measurements from multiple locations to view the scenes from multiple angles. For these tests, the accuracy error of the truth poses is measured. To do this, the ArUcO board from Figure 3.4 has been taped off so that only two outside markers are visible; the board is then placed horizontally. The distance between these two markers is 146 mm. Photos are then taken from many arm positions so that the two markers are in view of the camera. The position and Euclidean distance between the calculated markers are plotted in Figure 4.2. What can be spotted in the scatter plots is that there exists an error towards the back right side of the image of the camera that reaches 10 mm, likely due to the camera calibration not being configured perfectly. From these results, an expectation can be made that a prediction from the network or ICP will be between -5 and 10 mm off other measurements from different locations in the X and Y direction. Both the dataset and test setup have used the same camera that introduces this error.

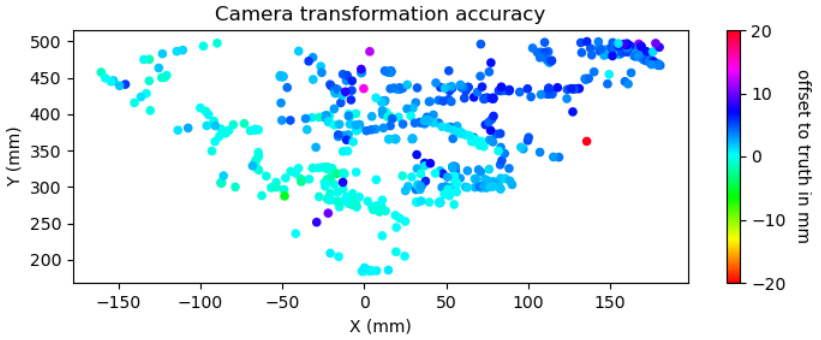

Figure 4.2: **Distance error of two ArUcO markers spaced 146 mm apart placed on different positions, dots are placed on the average position between the two markers. As the markers move further away (+Y) and to the right (+X), the calculated distance differs from the true distance.**

Trained on	< 6 *mm*	*Classification*
ADD-S	0.026	**0.996**
ADD	0.304	0.995
ADD-M	**0.353**	0.995

Table 4.1: **Prediction accuracy of the EfficientPose network trained on the different loss functions. The first column shows the fraction of the test set prediction with a distance lower than 6 mm according to the ADD-M metric. Classification is calculated as mean Intersection over Union (IoU)[24].**

4.3 Model accuracy

The section will show the accuracy of the designed system for individual measurements. First, the EfficientPose model has been trained using the three different loss metrics as described in Section 3.3. Some predictions on the test set from the best-performing network are visualized in Figure 4.3. The dataset has been split into an even training and test set, where the training set is 6-D and color augmented for every epoch for a total of 500 epochs. The training was performed on a system with a GTX 1080 Ti with 10 GB of VRAM. θ is a way to scale up the neural network with a larger width, depth, and input size [4]. The used system could not handle higher than $\theta = 0$, so this number was chosen. The network has been trained on three loss functions; ADD, ADD-S, and ADD-M from Section 3.3.1. The results of the test set after training are displayed in Table 4.1. The ADD-M metric returns the best network and will therefore be used in the remainder of the tests on images not included in either train or test set.

Figure 4.4 shows the spread of inference measurements to the true position of the clip. The biggest difficulty for the neural network is to determine the depth of the clip, with a mean of 12 mm from the true value and a standard deviation of 22 mm, while directions X and Z show a significantly less spread with both

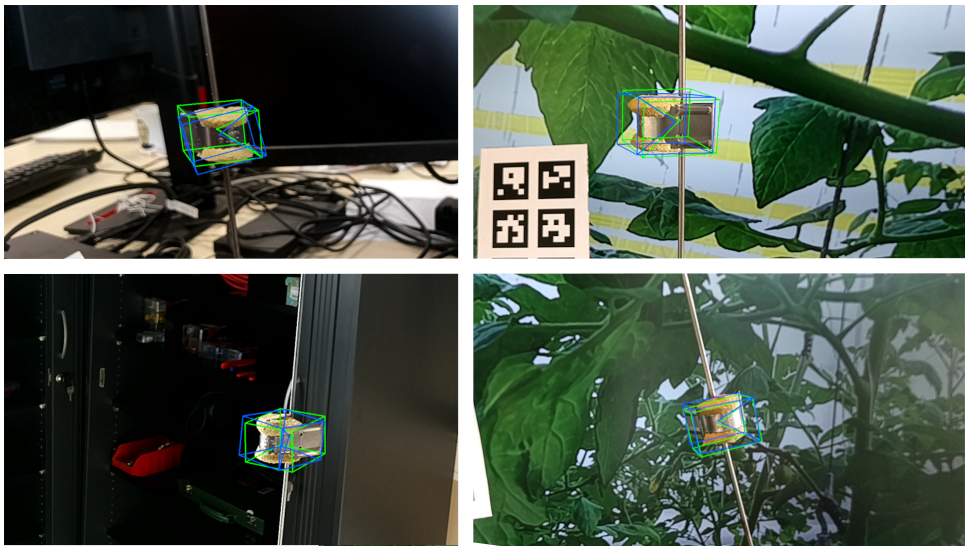

Figure 4.3: **The test set included clippers from different locations, with different lighting conditions to evaluate the training methods. Blue boxes indicate the prediction from the neural network, and green is the true transformation.**

mean within 2 mm and standard deviation of 7 and 3 mm respectively. The ICP step after inference does improve the measurements of depth to a mean of 7 mm and 13 mm standard deviation but worsens the X, and Z results. Figure 4.6 shows the cumulative error of both inference and ICP distance to the truth value. The average of the inference to truth and ICP to truth distance is taken for the ICP measurement to view whether adding the ICP step results in a lower distance per measurement. Only 30% of the attempts cause a fit that complies with the requirements. What is noticeable is that the ICP can not correct for rotational inaccuracies in the inference and is only marginally better in positional accuracy. To gain a better understanding of where the ICP approach is lacking, the clip rotation can be split up into X, Y, and Z axis, which stand for Yaw, Pitch, and Roll, respectively. The distribution of points is visualized in Figure 4.5. The horizontal axis tells the error of the inference, and the vertical axis is that of the ICP. The error is mostly due to the Roll, which is the axis that rotates around the metal rod from Section 1.3.1. The standard deviation of this error is 48 for inference and 77 for ICP. A likely explanation for this is the shape of the clip. While the clip is round at the front, it is not always visible on the point cloud. Thus depending on the detail at the time the point cloud is captured, the flat side of the model clip is fitted onto the front side of the point cloud clip.

After these tests were concluded, an investigation into the effect of the error in the annotated dataset was done. The work of EfficientPose [4] mentioned that using 6-D augmentation introduces a small error, but the added data outweighed the performance that this error introduced. As the Clipper is a small but detailed object, a small error could have a large impact on a converging solution of the EfficientPose network. Table 4.2 shows the difference between

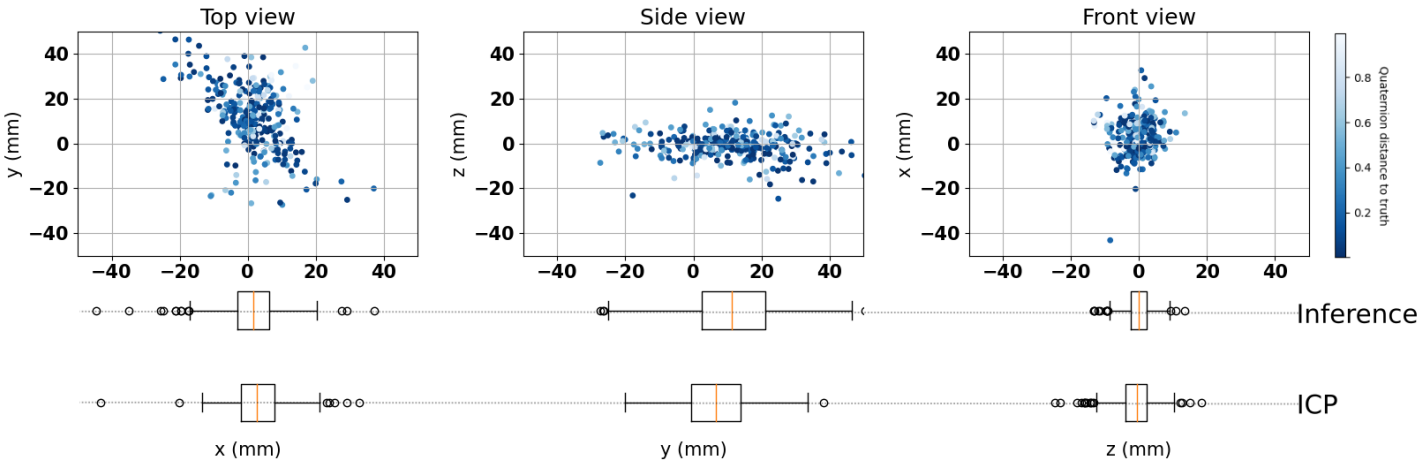

Figure 4.4: **Accuracy over the inference location with regards to the truth location at the center. Point color is determined by the accuracy of the rotational distance to the truth value; darker is better. Inference variance and deviation are shown on the boxplot.**

	With 6-D aug.	Without 6-D aug.
ADD-M	0.353	0.450
Translation Error Mean (mm)	9.728	8.031
Translation Error Std (mm)	8.868	8.034
Rotation Error Mean (degrees)	14.16	6.821
Rotation Error Std (degrees)	14.15	5.630

Table 4.2: **Accuracy of networks trained with ADD-M metric with and without 6-D augmentation on the training set.**

a network trained with and without 6-D augmented images. The error that is introduced causes large effects on the rotational accuracy.

These results have shown that errors introduced in the system, whether that to be in the annotation data, or used during execution of the system, have a large impact on the final prediction of the clip position and orientation. The fact that the clip is relatively small and round in comparison to objects that are presented in the BOP datasets, Since 6-D augmentation causes such a large impact on prediction while other objects do not suffer from this. It goes to show that an extra level of accuracy is required when it comes to annotating the data.

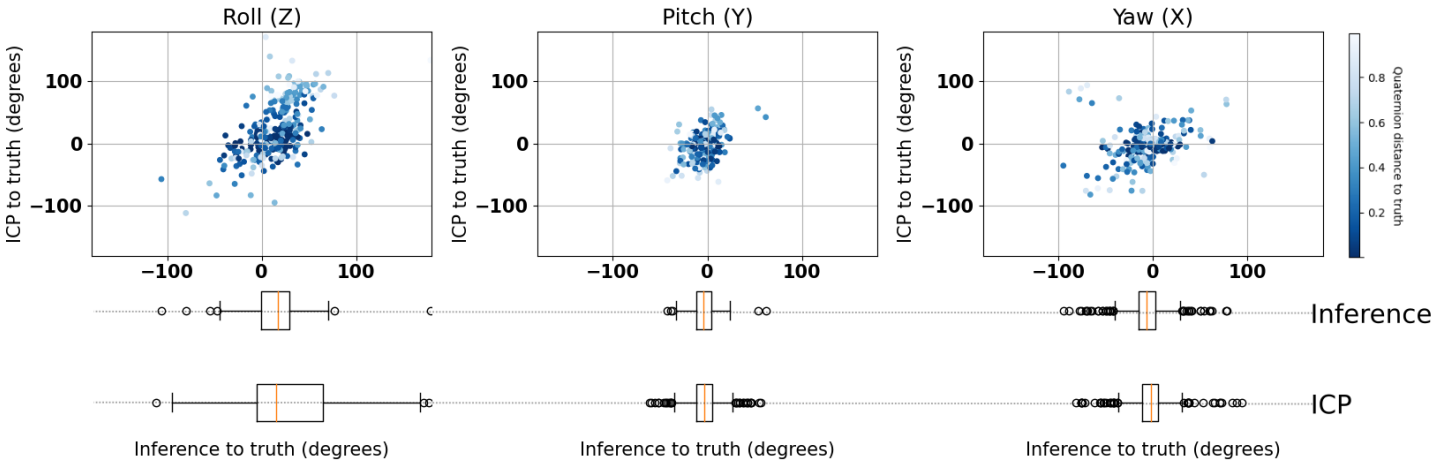

Figure 4.5: **Accuracy over the inference location with regards to the truth rotation at the center in Roll, Pitch, and Yaw. Point color is determined by the total quaternion distance; darker is better. Inference variance and deviation are shown on the boxplot.**

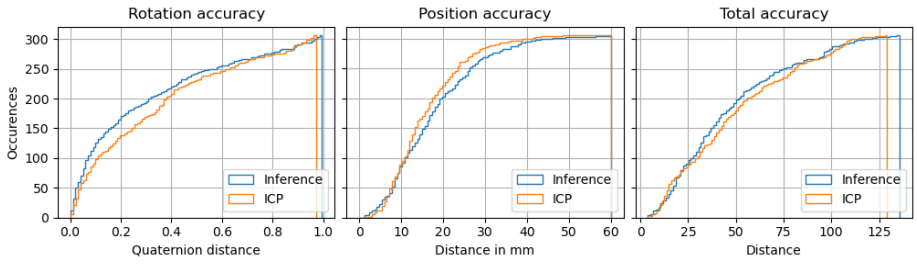

Figure 4.6: **Accumulative accuracy of the Rotation (left), Position (middle) and total (right) measured to the true pose of both inference and an Iterative Closest Point computed with the inference transformation as starting point.**

Chapter 5

Conclusions

This paper has conducted a feasibility study on effectively locating and orienting a clip for automating the plant extension process using the Clipper system, described in Section 1.3.1. To achieve this, a neural network that takes in an image and returns the position and orientation of objects has been trained on a custom dataset that does not require manual annotations. This annotation method uses a board of ArUcO positional markers and registers the position of the clip transformation based on a second set ArUcO markers. Using this methodology, three thousand annotations can be generated in a couple of days. A neural network has been used to predict the rotation and position of a clip in 3D. Due to the symmetric properties of the clip, different loss functions to train the neural network on have been tried and compared, where a custom one-to-one mirrored minimum function (ADD-M) returned the best-performing neural network. After this, the neural network was tested in a controlled greenhouse setup. As an extra refinement and validation step, a stereo camera has been used to perform an ICP algorithm, Section 3.4. For the study to succeed, the system's prediction accuracy must fall within the positional requirements of 14, 30, and 5mm for the X, Y, and Z directions, respectively. The system must also not exceed 15 degrees for X and Y and 5 degrees for Z. While the system does not meet these requirements at the moment, the results mentioned in Chapter 4 provide insight into why the system in its current state is not accurate enough. This chapter will discuss the weak and strong points in the pipeline and conclude on the feasibility of the chosen methodology.

5.1 Location

The system is able to detect the clip object very consistently with a 99.97% mean Intersection over Union (IoU), making false classifications an infrequent occurrence. In terms of positional accuracy, the network achieves mean values of 2, 2, and 12 millimeters for X, Z, and Y, respectively. These numbers are within the requirements, although many measurements still lie outside the allowed offset due to a large deviation. One explanation for this large deviation can be due to an incorrectly calibrated camera. As seen from the results in Figure 4.2, a small error is introduced in the annotations of the dataset. This error is worsened with the additional transformation from the ArUcO board to the clip,

which causes the error to be irregular depending on the position of both the ArUcO board and the clip. It is then understandable that the prediction does not converge to an offset but rather spreads around the center.

5.2 Rotation

Another likely effect of the error embedded in the camera is the bad rotation. While the rotation that was taken from the ArUcO markers had no significant enough error, the irregular location difference caused the neural network not to converge properly. This could be especially seen in training with the ADD-S metric, which can cope with all types of symmetry in an object. In theory, the ADD-S metric should perform as well as the ADD-M metric, which is specifically designed for the symmetric properties of the clip. Both keep the symmetry properties in mind, something the ADD metric does not do. However, the neural network always converged on a solution that had the Clipper on its side on the ADD-S metric since the closest point distance (rather than one-to-one distance) in this position and orientation was the lowest with the errors. While the other metrics did not have this specific problem, rotational features that do not match between annotations on similar locations could cause a less general solution.

5.3 ICP verification

Apart from the neural network, the ICP verification was meant to confirm the neural network prediction of the location and orientation. While in 30% of cases, this approach worked well, most observations on the point cloud lacked the detail necessary to fit the Clipper 3D model onto the point cloud successfully on the Y rotational axis. For this reason, using ICP refinement to validate a prediction is not recommended with the current depth estimation accuracy on the point cloud.

5.4 Final verdict

It is apparent that objects with such small sizes, like the Clipper, require a high-precision dataset to be able to determine the position and rotation of a Clipper consistently and within the requirements to be able to grab a clip in a greenhouse setting. At the current state, the setup cannot be deployed in the greenhouse due to inaccurate knowledge of the surroundings. The size and shape of the clip on the close distances found in a greenhouse setup require higher accuracy than objects the neural networks are tested on. Therefore, the feasibility of detecting and positioning a Clipper in the plant extension process can only be reached with more accurate data, both for the annotation pipeline and point cloud from the camera. The methods in this design show promising improving results but are not ready for deployment in a greenhouse at its current state.

Chapter 6

Future Work

As already mentioned in Chapter 5, the current state of the designed system does not reach the requirements for the end-effector to grab the Clipper consistently. This chapter will discuss the future work that can be done to improve this system and make it feasible to use in a greenhouse. Most importantly, the camera's intrinsics need to be improved to reduce the error in the dataset and make the neural network converge to a better solution. Either through a different camera or recalibration and testing again. As a requirement, the network should perform the same when trained using the ADD-M and ADD-S loss functions.

Other options can also be explored to achieve a more accurate prediction. One of these options is to explore 3D rendering of scenes as discussed earlier in Section 2.2.4 to generate additional data for the network. This data does not fall victim to a measurement error as the translation from camera to Clipper is extremely precise. With this extra data, other neural networks that utilize this data can also be explored and tried. In terms of point cloud approaches, further analysis can be conducted to validate and refine a prediction of the network.

During testing, some viewing angles of the clip did show enough detail and accuracy to be able to converge consistently to a good solution like those used in Figure 3.6. This can be used to position the arm to a point that is optimal for the point cloud. Another option to explore is the use of a different camera system. As discussed in Section 3.1.2, a Time of Flight camera may work better than the stereo camera that has been chosen. It would be interesting to find the difference in accuracy and detail in the greenhouse setting. Luxonis, the company behind the modular components for the camera, is developing more products, such as a Time-of-Flight sensor and a pre-configured short-range stereo camera utilizing two color global shutter cameras.

When the desired accuracy has been reached, the next step in the Clip extension process involves path planning to the grab position, avoiding any obstacles along the way. Testing the end-effector in the greenhouse would preferably be done with a mechanical design, as the current design is very elongated, making it difficult to maneuver between plants without hitting any plants. The environmental analysis was conducted in a greenhouse that uses the Hook & Wire method explained in Section 1.1.1. Therefore, no data has been gathered on the rotation angles of the clips when the plant grows. This information is useful when designing a new end-effector to determine the positions and angles it must reach in order to grab all possible orientations of the clip.

Bibliography

[1] Boaz Arad, Jos Balendonck, Ruud Barth, Ohad Ben-Shahar, Yael Edan, Thomas Hellström, Jochen Hemming, Polina Kurtser, Ola Ringdahl, Toon Tielen, et al. Development of a sweet pepper harvesting robot. *Journal of Field Robotics*, 37(6):1027–1039, 2020.

[2] Boaz Arad, Polina Kurtser, Ehud Barnea, Ben Harel, Yael Edan, and Ohad Ben-Shahar. Controlled lighting and illumination-independent target detection for real-time cost-efficient applications. the case study of sweet pepper robotic harvesting. *Sensors*, 19(6):1390, 2019.

[3] Bradley Dillon Brandon Gilles. Depthai hardware documentation 1.0.0 documentation.

[4] Yannick Bukschat and Marcus Vetter. Efficientpose: An efficient, accurate and scalable end-to-end 6d multi object pose estimation approach. *arXiv preprint arXiv:2011.04307*, 2020.

[5] CBS. Groenteteelt; oogst en teeltoppervlakte per groentesoort, Mar 2022.

[6] Dmitry Chetverikov, Dmitry Svirko, Dmitry Stepanov, and Pavel Krsek. The trimmed iterative closest point algorithm. In *2002 International Conference on Pattern Recognition*, volume 3, pages 545–548. IEEE, 2002.

[7] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018.

[8] JA Dieleman, Arie de Gelder, BA Eveleens, Anne Elings, Jan Janse, Peter Lagas, Tian Qian, JW Steenhuizen, and Esther Meinen. Tomaten telen in een geconditioneerde kas: groei, productie en onderliggende processen. Technical report, Wageningen UR Glastuinbouw, 2009.

[9] Mark Fiala. Artag revision 1, a fiducial marker system using digital techniques. *National Research Council Publication*, 47419:1–47, 2004.

[10] S Garrido-Jurado. R. mu ẽnoz-salinas, fj madrid-cuevas, and mj mar éın-jim éenez. automatic generation and detection of highly reliablefiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *2011 international conference on computer vision*, pages 858–865. IEEE, 2011.

[13] Tomas Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders Glent-Buch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, et al. Bop: Benchmark for 6d object pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 19–34, 2018.

[14] Tomáš Hodaň, Martin Sundermeyer, Bertram Drost, Yann Labbé, Eric Brachmann, Frank Michel, Carsten Rother, and Jiří Matas. Bop challenge 2020 on 6d object localization. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 577–594. Springer, 2020.

[15] Yanbin Hua, Nairu Zhang, Xin Yuan, Lichun Quan, Jiangang Yang, Ken Nagasaka, and Xin-Gen Zhou. Recent advances in intelligent automated fruit harvesting robots. *The Open Agriculture Journal*, 13(1), 2019.

[16] Zhuoling Huang, Sam Wane, and Simon Parsons. Towards automated strawberry harvesting: Identifying the picking point. In *Towards Autonomous Robotic Systems: 18th Annual Conference, TAROS 2017, Guildford, UK, July 19–21, 2017, Proceedings 18*, pages 222–236. Springer, 2017.

[17] JoonYoung Kim, HyeRan Pyo, Inhoon Jang, Jaehyeon Kang, ByeongKwon Ju, and KwangEun Ko. Tomato harvesting robotic system based on deep-tomatos: Deep learning network using transformation loss for 6d pose estimation of maturity classified tomatoes with side-stem. *Computers and Electronics in Agriculture*, 201:107300, 2022.

[18] Yann Labbé, Justin Carpentier, Mathieu Aubry, and Josef Sivic. Cosypose: Consistent multi-view multi-object 6d pose estimation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVII 16*, pages 574–591. Springer, 2020.

[19] Chris Lehnert, Chris McCool, Inkyu Sa, and Tristan Perez. Performance improvements of a sweet pepper harvesting robot in protected cropping environments. *Journal of Field Robotics*, 37(7):1197–1223, 2020.

[20] Yi Li, Gu Wang, Xiangyang Ji, Yu Xiang, and Dieter Fox. Deepim: Deep iterative matching for 6d pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 683–698, 2018.

[21] Chongpei Liu, Wei Sun, Keyi Zhang, Jian Liu, Xing Zhang, and Shimeng Fan. Prior geometry guided direct regression network for monocular 6d object pose estimation. In *2022 41st Chinese Control Conference (CCC)*, pages 6241–6246. IEEE, 2022.

[22] Lely Industries N.V. Lely, Mar 2022.

[23] Deepak Rao, Quoc V Le, Thanathorn Phoka, Morgan Quigley, Attawith Sudsang, and Andrew Y Ng. Grasping novel objects with depth segmentation. In *2010 IEEE/RSJ international conference on intelligent robots and systems*, pages 2578–2585. IEEE, 2010.

[24] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019.

[25] Ashutosh Saxena, Justin Driemeyer, Justin Kearns, and Andrew Ng. Robotic grasping of novel objects. *Advances in neural information processing systems*, 19, 2006.

[26] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020.

[27] Emiel van der Meijs. Automation of repetative tasks in the agricultural sector. 2022.

[28] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.

[29] Yuanshen Zhao, Liang Gong, Yixiang Huang, and Chengliang Liu. A review of key techniques of vision-based control for harvesting robot. *Computers and Electronics in Agriculture*, 127:311–323, 2016.