"1" - [**##3**]

Smart Vision-Guided Robotic Depalletising System

Christoffer Johan Soos

Project Report

10th Semester

Aalborg University

The Department of Electronic Systems

Department of Electronic Systems
Fredrik Bajers vej 7B, 9220 Aalborg Øst
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Smart Vision-Guided Robotic Depalletising System

Theme:
Vision & Grasping

Project Period:
10. Semester

Project Group:
Group

Participant(s):
Christoffer Johan Soós

Supervisor(s):
Dimitris Chrysostomou

Page Numbers: 43

Date of Completion:
January 30, 2026

Abstract:

Automated depalletising is a massive challenge in warehouse automation due to high variability in parcel size, shape, material, and stacking configuration. While recent advances in AI-based perception have enabled robotic systems to operate in mixed-SKU environments, limitations remain in robustness, autonomy, and generalisation under unstructured conditions. This thesis investigates the design and implementation of a robotic depalletising system that integrates vision-based object detection, segmentation, and pose estimation. A modular perception pipeline is developed to process RGB-D data and extract graspable objects from pallet scenes. The system employs depth-based segmentation combined with edge-based detection for accurate box identification, and RANSAC-based plane fitting for pose estimation without requiring prior knowledge of parcel dimensions. The system is implemented within a ROS2-based architecture using a UR10 collaborative robot, VG10 vacuum gripper, and Intel RealSense D455 RGB-D camera. Experimental evaluation across three scenarios of increasing complexity shows 93.3% detection accuracy and 83.3% overall pick-and-place success rate.

The content of this report is freely available, but publication (with reference) may only be pursued due to agreement with the author.

Preface

Christoffer Johan Soos

csoas20@student.aau.dk

Aalborg University, January 30, 2026

Contents

Preface iv

1 Introduction 1
- 1.1 Problem Statement . . . 2
- 1.2 Research Questions . . . 2
- 1.3 Contributions . . . 3

2 Related Work 4
- 2.1 Literature review . . . 6
  - 2.1.1 Summary . . . 8
- 2.2 Object Detection and Segmentation . . . 9
  - 2.2.1 Old School Approaches . . . 10
- 2.3 Commercial solutions . . . 10
  - 2.3.1 Plus One Robotics . . . 10
  - 2.3.2 BHS Robotics . . . 13

3 System 16
- 3.1 Software Pipeline Overview . . . 16
- 3.2 Hardware . . . 17
  - 3.2.1 ROS2 Node Structure . . . 20
- 3.3 Box Detection Module . . . 21
  - 3.3.1 Depth-Based Detection . . . 21
  - 3.3.2 Edge-Based Detection . . . 22
- 3.4 Pose Estimation Module . . . 23
  - 3.4.1 3D Point Extraction . . . 23
  - 3.4.2 RANSAC Plane Fitting . . . 24
- 3.5 Camera Calibration . . . 24
  - 3.5.1 Intrinsic Calibration . . . 24
  - 3.5.2 Hand Eye Calibration . . . 24

3.6 Parameter Selection . . . 25
3.7 Chapter Summary . . . 27

4 Results **28**
4.1 Test Scenarios . . . 28
4.2 Detection Performance by Scenario . . . 28
4.3 Pose Estimation Evaluation . . . 29
4.3.1 Error Metrics . . . 29
4.3.2 Accuracy Thresholds . . . 30
4.3.3 Assumptions and Limitations . . . 30
4.3.4 Test Parcel Specifications . . . 30
4.3.5 Box 1: Small Cardboard . . . 31
4.3.6 Summary and Analysis . . . 34
4.3.7 Key Findings . . . 34
4.3.8 Implications for Grasping . . . 35
4.3.9 Orientation Estimation . . . 35
4.4 Pick-and-Place Performance by Scenario . . . 35
4.5 Comparison with Related Work . . . 36
4.6 Detection Summary . . . 36
4.7 Pose Estimation Summary . . . 37
4.7.1 Pick and Place Summary . . . 37
4.8 Advantages and Limitations . . . 37

5 Conclusion and Future Work **39**
5.1 Conclusion . . . 39
5.2 Future Work . . . 39

References **41**

1 Introduction

Recent advancements in automation, logistics, and manufacturing have improved industrial processes. Progress in automation and robotics technologies has enabled the deployment of complex systems that operate with high efficiency and precision while simultaneously reducing operational costs and enhancing worker safety. Automated systems minimise human error in fast paced manufacturing environments, where even brief interruptions in production can incur substantial costs for the company.[17]

Automated depalletising remains a challenging task in logistics and manufacturing due to varying box sizes, stacking patterns, and environmental conditions. This project focuses on developing and validating a complete vision guided robotic solution for reliable depalletising operations.

In large scale manufacturing and supply chain operations, the automation of palletising and depalletising processes has become a key factor in achieving higher productivity and workplace safety. By incorporating robotic systems into these operations, companies can take advantage of vertical storage, optimise warehouse layouts, and ensure continuous handling of items with minimal downtime. Tasks that once relied heavily on manual labour or the use of forklifts can now be completed by collaborative robots, allowing employees to focus on supervision and more skilled work. In addition to improving ergonomics and reducing workplace injuries, automation contributes to higher throughput and consistency, as robotic systems can operate continuously and maintain precise handling of items. Automated inspection and vision based feedback also help to prevent misplacement and product damage, both of which are common in traditional manual processes.[17][14]

Depalletising automatically also helps connect delivery and reception operations. When items arrive, both sender and receiver must coordinate to transfer parcels from transport vehicles to the storage or production line. This step is often time consuming and labour intensive. Robotic arms can now be deployed to unload parcels directly from trucks onto conveyor belts, significantly reducing the physical burden on workers and improving operational flow, which has become a norm in a lot of companies, especially bigger chains like Elgiganten.

While automation clearly offers advantages, there are also downsides of user input error. If a

user orders something and there is a mistake with the order, and writes to the company about it, it will most likely be too late, and the automatic system has already packaged and sent the order out to the delivery team. Implementing reliable robotic depalletising systems remains a technical challenge. Many existing solutions operate effectively only under controlled conditions, where box sizes, shapes, and placements are predefined. More advanced systems have begun to utilise RGB D cameras to perceive the environment, but their performance often degrades in poor lighting conditions or when objects are stacked closely together. Occlusions, irregular orientations, and low contrast surfaces can all make it difficult to identify box boundaries accurately, which can compromise pose estimation and grasping reliability heavily.

The current literature demonstrates that most existing solutions lack flexibility. They typically depend on known object parameters or extensive training data, limiting their adaptability to new environments and parcel types. Consequently, there is a growing need for systems that combine perception, learning, and reasoning in a way that allows for better generalisation and human robot collaboration. A flexible system capable of adapting to unknown parcel parameters could enhance the efficiency and usefulness of industrial depalletising systems.

The robotic system remains the same throughout all solutions: it uses a UR10 collaborative manipulator from Universal Robots, a VG10 vacuum gripper from OnRobot, and an Intel RealSense D455 RGB D camera. The integration of perception, planning, and control is achieved using ROS2, while MoveIt2 is employed for motion planning and collision avoidance.

1.1 Problem Statement

”How can you make a Smart Vision-Guided Robotic Depalletising System with unknown box parameters”

1.2 Research Questions

This thesis addresses the following research questions:

- How can depth based and edge based detection methods be combined to improve box detection robustness in depalletising scenarios with varying parcel configurations?

- What pose estimation accuracy can be achieved using RANSAC based plane fitting, and how does sensor noise affect grasp success?

- What are the practical limitations of the proposed system when operating under increasingly complex scenarios involving mixed parcel sizes, occlusion, and rotation?

1.3 Contributions

- A perception pipeline combining depth-based segmentation with edge-based detection to improve detection robustness across varying scene conditions.

- A RANSAC based pose estimation approach that operates without prior knowledge of parcel dimensions.

- An adaptive grasping strategy that compensates for depth uncertainty through incremental descent.

- Systematic experimental evaluation across three scenarios of increasing complexity, with failure analysis and comparison to existing methods.

2 Related Work

The logistics and manufacturing sectors have seen steady progress in robotic pick-and-place automation, especially in box handling. However, existing solutions vary significantly in their assumptions, sensing strategies, and levels of autonomy. While some systems are designed for highly structured environments with known box geometries, others target mixed-SKU scenarios using vision-based perception and Machine Learning approaches.

Traditional industrial palletising cells are typically deployed in controlled environments where boxes have known dimensions, orientations, and packing patterns. In such settings, robotic palletising is more simple, as items are placed according to predefined layouts. Depalletising can present additional challenges: loads may shift during transport, leading to misalignment, partial occlusion, or rotation of boxes. Vision systems in conventional depalletising cells are often limited to verification tasks such as confirming box presence or alignment, rather than full scene interpretation or adaptive grasp planning [4].

One of the main limitations of these systems is their reliance on strong assumptions regarding the box. Many approaches presuppose known box dimensions, packaging, and structured pallet layouts. Several academic and industrial studies have highlighted this issue. A lot of the existing literature addresses packing optimisation or robotic manipulation in isolation, without fully integrating realistic perception challenges, end effector constraints, and scene complexity into a total depalletising framework [3].

Research is moving towards less structured environments, and vision-based robotic systems have been proposed to handle ”Difficult” cargo and unpredictable layouts. For example, the work presented in Machine Vision Assisted Design of End Effector Pose in Robotic Mixed Depalletising of Heterogeneous Cargo introduces a depalletising system capable of handling boxes with varying sizes and arbitrary orientations. Using sparse depth data, the system estimates object size, position, and orientation, and generates collision-free grasp trajectories [19]. This approach reflects real-world scenarios in which pallets contain mixed box sizes or have been disturbed during transport.

More recent studies integrate deep learning with RGB-D sensing to address increasingly complex depalletising scenarios involving diverse box sizes, surface textures, and irregular stack-

ing. In [12], an RGB-D camera combined with a deep learning-based segmentation model (YOLACT) is used to detect boxes in truck unloading scenarios. The system reports mean average precision (mAP) values of approximately 93% and 90% across different test cases, demonstrating the effectiveness of learning-based perception in cluttered environments.

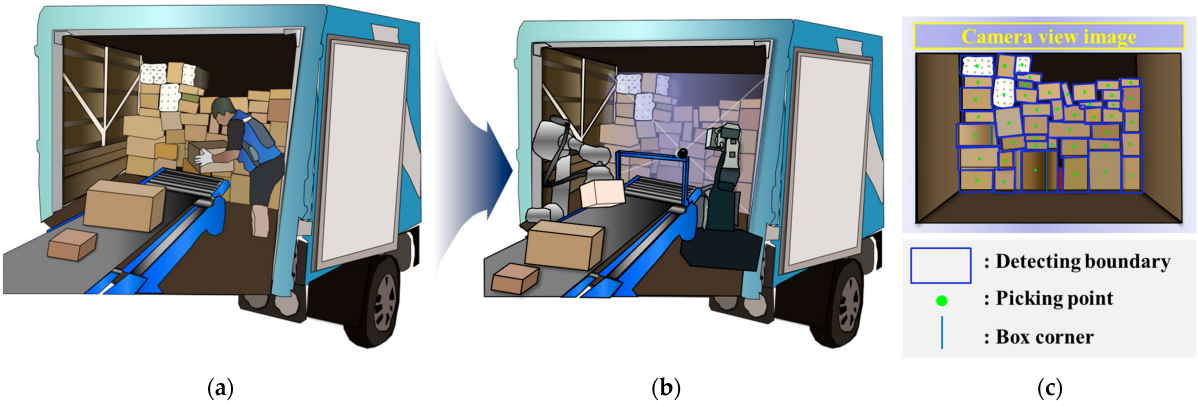

Figure 2.1: Conceptual diagram of box depalletising: (a) conventional conveyor belt system based on manual labour, (b) proposed system supported by robot arms and vision sensors, and (c) image detected using 3D RGB D camera for automatic picking.[12]

2.1 Literature review

Research on automated depalletising spans a wide range of sensing technologies, robotic configurations and algorithmic approaches, each attempting to address the variability and complexity found in real world pallet handling. A lot of the literature can be understood by looking at the different trade off between cost, efficiency, robustness and complexity, as the authors prioritise different aspects of depalletising.

One of the earlier notable contributions is presented in [16] where the authors propose a depalletising system based on a gantry robot equipped with a telescopic arm. The setup offers a large workspace and demonstrates strong robustness when handling boxes arranged in irregular or complex patterns. This is achieved through a combination of depth point clustering and morphological processing of RGB images. While the method performs well, the high mechanical complexity and associated cost limit its practicality for many industrial settings.

Alternative designs attempt to simplify the sensing and mechanical requirements. For instance, the work in [11] uses 2D range data captured via a time of flight sensor to detect boxes in real time. Because the system is light agnostic, it operates reliably under varying illumination. However, the method is evaluated only on pallets with known box dimensions and well structured layers, leaving its performance in unstructured, real world scenarios untested.

A related approach is found in [10], which also employs time of flight sensing. Here, the authors scan the top layer of the pallet, extract edges and fit lines to estimate 3D box vertices. These

vertices are subsequently passed to an object recognition module. Although the technique is shown to be simple, versatile and relatively robust, it struggles when boxes are tightly aligned or placed neatly, and its computational demands are high.

More recent work increasingly integrates robotic hardware with RGB D sensing. In [7], the authors present a system combining a linear actuator, suction gripper and RGB D camera for box detection. The top surfaces of the boxes are used to infer length and width, achieving a detection precision of about 10 mm. However, the system performs poorly when boxes contain printed logos or images, and its large workspace and high maintenance requirements pose practical limitations.

A flexible depalletising system tailored for supermarket logistics is discussed in [6]. The setup involves an industrial manipulator with a suction tool and an RGB D camera for box localisation, with pallets arranged around the robot. Although the method is promising, the validation is limited to small experiments involving only ten boxes, making the scalability of the solution unclear.

Other researchers focus on lidar based strategies. In [13], the authors introduce a recognition and localisation framework where horizontal scans at multiple heights are used to estimate box poses. The system relies on corner feature detection and requires the operator to input box dimensions and the maximum number of pallet layers. Since the experiments use standardised boxes arranged neatly, it is uncertain whether the method would handle unknown geometries or more chaotic pallet configurations.

Recent literature shifts toward deep learning to overcome the constraints of traditional geometry based methods. For example, [5] explores convolutional neural network (CNN) models for depalletising and investigates how transfer learning and data augmentation can reduce the data requirements and setup time. The move toward learning based detection is motivated by the well known limitations of classical algorithms in handling occlusions, clutter and varied lighting conditions.

A similar direction is taken in [1], which addresses mixed pallet scenarios common in supermarket logistics. The method uses an RGB D camera and relies on a database containing the expected number of boxes, their dimensions, and reference images. Using SIFT features, the system matches detected edges to known box faces, but it depends heavily on prior knowledge

about the boxes, limiting its adaptability.

Mobile depalletising has also been explored. In [22], a mobile system combining a MiR100 platform, a UR10 collaborative robot and a vertically adjustable conveyor is presented. Using a Canny edge detector for RGB images and depth based edge extraction, the system identifies box corners and selects plausible box hypotheses using a genetic algorithm. Although this method successfully detects boxes from a single RGB D frame [15], it assumes that the top layer is flat, which may not always hold in realistic pallet loads.

The emergence of deep learning for depalletising continues in [20], which proposes a two stage system using YOLOv3 for object detection and YOLOv8 for keypoint estimation. The authors highlight that deep learning provides improved robustness under occlusion and complex layouts compared with classical approaches.

Finally, recent work such as [12]and [21] focuses specifically on enhancing perception. [12] uses RGB D data and deep learning based edge detection to identify boxes of irregular shape, uneven surfaces and rotated orientations, particularly in truck unloading scenarios. Meanwhile, the work in [21] integrates Cycle GAN with Mask R CNN to remove visual noise, such as labels, stickers and tape before segmentation. This two stage method significantly improves detection accuracy, especially when box surfaces contain distracting elements. However, experiments remain limited to small scale setups with only a few boxes, leaving open questions regarding scalability to industrial palletising tasks.

2.1.1 Summary

The existing depalletising literature reveals that most proposed systems rely on strict assumptions to function reliably. Common constraints include predefined box placements, known dimensions and highly structured pallet layouts. As highlighted by prior work, such systems tend to be heavily standardised and lack the flexibility needed for real world industrial environments. Although some studies have begun exploring more challenging scenarios using convolutional neural networks, these approaches often depend on large volumes of training data, which can be time consuming and costly to collect.

Across the literature, three primary depalletising scenarios emerge: truck unloading, conveyor belt unloading, and pallet unloading. Each presents distinct challenges. Truck and pallet un-

loading typically involve cluttered, tightly packed boxes that may have shifted or rotated during transport, making segmentation and edge extraction more difficult. By contrast, conveyor belt unloading generally involves a single box at a time, providing higher contrast with the background and simplifying detection.

To address these challenges, RGB D sensing has become a popular choice, particularly when combined with deep learning for object detection and segmentation. Such methods offer improved robustness to variations in box size, pose and appearance. The main strategy in the literature is a two stage pipeline: initial detection or segmentation using RGB D data, followed by a pose estimation step to determine the 3D orientation and position of each box. Despite these advances, there are still missing a lot of steps, but there are some people who went to the next step with their solution and put it out on the market.

2.2 Object Detection and Segmentation

Object detection has evolved significantly over the past decade, transitioning from two-stage convolutional architectures to highly efficient one-stage detectors and, more recently, transformer-based vision–language models. Early frameworks such as R-CNN, Fast R-CNN, and Faster R-CNN established the foundations of modern detection by combining region proposals with convolutional feature extraction. Faster R-CNN, in particular, improved efficiency through the introduction of a Region Proposal Network (RPN). Mask R-CNN extended this paradigm by adding an instance segmentation branch and introducing RoIAlign, enabling pixel-accurate object masks. This capability is especially relevant in depalletising, where precise object boundaries improve grasp pose estimation and collision avoidance. The computational cost of two-stage detectors motivated the development of one-stage methods such as YOLO and SSD, which treat detection as a direct regression problem. YOLO's grid-based formulation enables real-time performance, while SSD improves detection across scales using multi-resolution feature maps. These methods are attractive for depalletising systems where throughput is critical.

More recently, transformer-based and vision–language models have enabled open-vocabulary detection. Systems such as GroundingDINO and YOLO-World incorporate text encoders, allowing objects to be detected based on descriptive prompts rather than fixed class sets. Similarly, the Segment Anything Model (SAM) enables promptable segmentation using a Vision Transformer backbone. These approaches are particularly promising for depalletising environments with unpredictable or evolving object categories.

2.2.1 Old School Approaches

Older computer vision methods operate on hand-crafted features and geometric principles rather than learned representations. Common techniques include:

- **Depth-based segmentation**: Using depth and elevation thresholds to identify objects above a reference plane.

- **Edge detection**: Algorithms such as Canny edge detection identify intensity gradients corresponding to object boundaries.

- **Morphological operations**: Opening and closing operations remove noise and fill gaps in binary masks.

- **Contour analysis**: Extracting and filtering contours based on geometric properties.

I am using the old school methods. Because I have no fitting training data available, and creating a labelled dateset of boxes would be outside the project scope. The boxes that I am trying to detect should be regular, so like rectangular and flat, which should then make detection via depth threshold and edges viable. The implemented approach combines depth-based segmentation with Canny edge detection. This hybrid approach uses depth information for reliable detection of elevated regions whilst using edge information to refine box boundaries. Future work could explore integrating learning-based methods such as SAM for improved generalisation across diverse box appearances and challenging conditions.

2.3 Commercial solutions

On the industrial side, companies such as Plus One Robotics offer depalletising systems that explicitly target mixed-case pallets using AI perception software. BHS Robotics offers a fully autonomous, AI-driven depalletising solution.

2.3.1 Plus One Robotics

Plus One Robotics is a company that specialises in AI enabled vision software and robotic handling solutions for warehousing industries. Its solution is particularly focused on mixed SKU(Multiple distinct products) and mixed packaging pallet operations. The firm uses perception, planning and human in loop handling in order to enable robots to handle diverse boxes and

boxes with greater flexibility than traditional palletising/depalletising cells. The main selling point of Plus One is their software platform called PickOne™[18], described on their website as “the fastest AI powered vision software in the market”, which supports 2D + 3D vision, object classification, geometry extraction and robot control interfaces.[18]

Their perception uses a combination of 2D and 3D imaging, along with artificial intelligence algorithms, the robot is capable of detecting, localising, and classifying a variety of boxes on mixed pallets. The perception system determines the position, orientation, and physical characteristics of each item, such as size, shape, and surface reflectivity. This process enables the robot to adapt to non uniform pallet loads, including those with irregularly shaped, overlapping, or visually challenging objects.

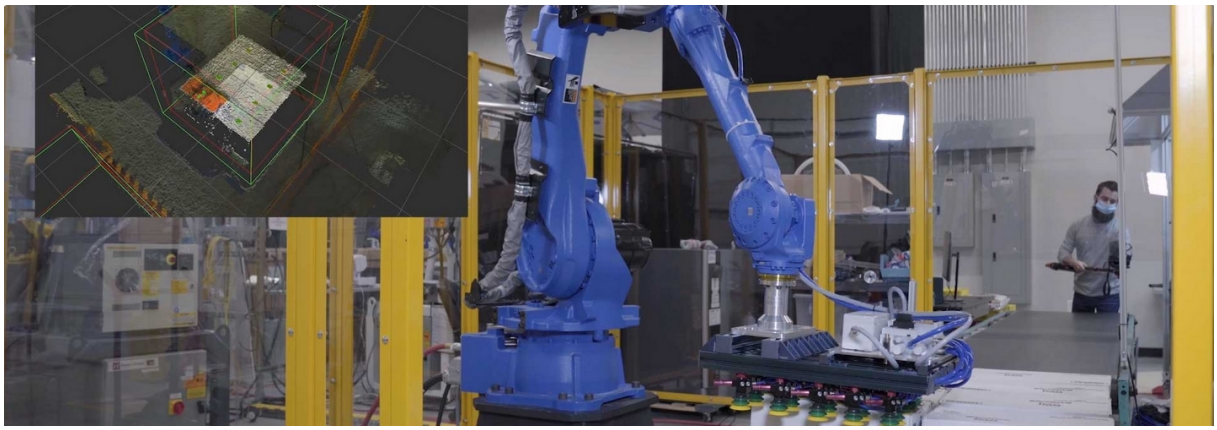

Figure 2.2: AI Powered Robotic Depalletising with human worker for outliers

Once the items have been identified, the planning module translates the perception data into movements. Computing the optimal path for the robotic arm to reach, grasp, and place each object without causing collisions or damaging the surrounding packages. The motion planning system can adjust the grip strategy, trajectory, acceleration, and speed based on the object's properties. Even though the system is able to do all of this, it still proves to be challenging for the system to be efficient in every scenario, some situations still exceed the allowed confidence threshold, usually when the boxes are damaged, reflective or something other than the expected boxes ending up in the robots workspace. For that reason Plus One Robotics are still using humans to handle these cases, so when the system says its not confident enough to do the action, a remote human worker is notified and can then do it manually, and they record these human to robot interactions for future machine learning training.

Plus One's solution mainly focuses on "mixed case" and "rainbow" pallets. Mixed case pallets are those where each layer may contain a mixture of different boxes, sizes, weights, and packaging formats. Which can be very challenging for robots and vision systems to handle efficiently. Rainbow pallets refer to pallets where each layer is uniform but successive layers are different, so it might have the bottom layer containing cartons of bottled water, the middle layer with soft drink cases, and the top layer with juice boxes. The website explains that their system can handle both types and reports that users have achieved a roughly 30% reduction in picking and sorting times compared to manual operations.[18]

From the gathered information, the system relies on significant AI training and data gathering to handle box types, which means it is relying a lot on the operator and updates until it is trained enough. The system mainly focuses on the pick aspect of depalletising, where the focus

is to pick and place the boxes on conveyor belts, this limitation means it would not be able to stack boxes efficiently, and another system / human would have to fill in that part. Vision and perception in uncertain environments is always challenging, but the way they solved it with having a human operator on stand by is a smart way to ensure the automated system can function with a high uptime.

2.3.1.1 Summary

In summary, the company presents a flexible approach to depalletising. The combination of AI vision (PickOne) and human help makes their solution function well in mixed SKU environments. Combine that with their modular hardware (DepalOne cell) as well, it offers a smart solution for operators. The system does heavily rely on training data, and the system does rely on new training and updates when new boxes are introduced.

2.3.2 BHS Robotics

BHS Robotics is also a leading provider of AI driven robotic guidance and automation systems. Their approach focuses on combining AI and machine vision technologies to automate complex handling tasks, like the previous solution. At the 2022 PackExpo demonstration, BHS Robotics showcased its AI based depalletization system, designed specifically to address the challenges of mixed case and rainbow pallet handling.

2.3.2.1 System

BHS Robotics depalletizing solution integrates an AI powered vision camera mounted above the pallet workspace. The camera is equipped with high resolution optics to capture images continuously of the pallets from above. The images are then processed by a machine learning model trained to identify each individual pickable item, regardless of shape or material. The algorithm determines pick positions and orientations, allowing the robot to dynamically plan its motion paths. The systems software platform calculates the most efficient sequence of picks and placements, optimising for both speed and stability. On their website, they claim to be able to handle up to 600 cartons per hour. The solution can also incorporate buffer systems to improve pack density and stability, this is particularly useful for rainbow pallet formats where layer differences can affect how they need to be stacked.

2.3.2.2 Solution

BHS Robotics depalletising solution begins with the arrival of a pallet into the robots workspace. It was not clear how the pallet would get into the workspace, but we can assume it is either manually or via a conveyor. The vision system captures a sequence of images from the current pallet state. The images are then fed into their machine learning model, which like previous solutions, analyses the state to segment and classify each box / pickable product. They fully rely on AI inference to generalise across different geometrics, materials, colours and reflectivities. This does mean if the model is trained on enough diverse material, then it would be able to detect both regular and irregular shaped items without requiring CAD models or strict pallet layouts, but relying heavily on the model alone can cause issues depending on how ”Random” the boxes you are picking are.

After the boxes are detected, the motion planning module computes the correct grasp pose and the trajectory for the robot to move the box. Their motion planning system aims to be fully autonomous, every pick operation is calculated locally, without deferring to human intervention, as they want their entire system to be operator independent. The robot then transfers the box to the designated output location which is typically a conveyor belt. They also say for rainbow pallets they have some specific buffer modules to restructure layers for a more stable arrangement (They do not give a lot of information about this) They have a zone where they ”break down the layers” by placing the items there first, this is useful if the items need to be reloaded onto a new pallet or sent further downstream, and enables them to more easily resequence the boxes.

2.3.2.3 Limitations

BHS Robotics solution appears to be really solid, but there aer several practical limitations which should be noted. As said before, the system is AI driven, so the performance relies heavily on the quality and diversity of the training data. If there are highly unconventional boxes, which could occur if they have been damaged, or with visual ambiguity. Unlike the previous solution, they do not appear to have any human in the loop system, which can result in some cases the model wont be able to confidently classify and are more likely to result in a pause or rejection.

The systems are designed around the assumption of a relatively controlled operational environment. It can handle mixed case and rainbow pallets, but the solution still performs best when

the pallet variability falls within a specific predictable range. So in cases such as manufacturing supply chains, 3PL logistics centres where products differ in size and packaging, but the overall distribution of boxes follow established patterns, this way the AI driven system will have a much higher confidence rating compared to cases with a high unpredictability, such as last mile operations which can receive boxes from thousands of different sellers.

BHS Robotics do advertise its ability to stack boxes by using a buffer system, but this feature is not just play and play and requires a specific amount of space and it depends on the physical layout of the facility. As an example, the availability of space for buffer modules, conveyor routing, and the need for synchronisation with downstream automated storage or dispatch systems. In some implementations, extra hardware such as shuttle conveyors, accumulation buffers, or pallet lifts may be necessary.

2.3.2.4 Summary

BHS Robotics solution is a fully autonomous solution, which is AI driven. It relies heavily on how well the machine learning model is trained. Its capable of handling a wide range of pallet formats with no human dependency and the vision system is executed locally, which they claim to make the system very efficient at high speed picking cycles. They claim they can achieve outputs of up to 600 cartons per hour. Due to the way the system is setup, its strength lies in environments where box variability is predictable, such as at manufacturing warehouses, as its important for the machine learning model to operate within familiar boundaries if you want the solution to be fully autonomous. This does mean the system can struggle at last mile operations due to the high unpredictability of what they can receive, combine that with the lack of human input and the confidence level of the machine can struggle. But if you play to BHS Robotics strengths, the solution is really solid and it's clear why they are also successful on the market.

3 System

The proposed system is an automated depalletising system. The task consists of autonomously identifying boxes placed on a pallet, grasping them using a robotic manipulator, and transporting them to predefined destinations. The system is designed to operate under a set of practical assumptions that reflect typical warehouse conditions. A fixed external vision sensor is used to observe the workspace and provide visual information about the pallet. All boxes intended for manipulation are assumed to be within the field of view of the camera and reachable by the robot. The destination location for placing the boxes is known in advance. This chapter presents the complete system architecture for the vision-guided robotic depalletising system. The system is designed to autonomously identify boxes, estimate their poses, and execute pick-and-place operations without prior knowledge of parcel dimensions.

3.1 Software Pipeline Overview

The depalletising system follows a modular software pipeline that processes sensor data through several stages to achieve autonomous pick-and-place operations.

Calibration (Offline)

Before the system can operate, two calibration procedures must be completed:

- **Intrinsic Calibration**: Determines the camera's internal parameters using a ChArUco calibration board.

- **Hand-Eye Calibration**: Establishes the spatial relationship between the camera coordinate frame and the robot base frame.

Image Acquisition

During operation, the Intel RealSense D455 camera captures synchronised RGB and depth images. Temporal averaging over 20 frames reduces depth noise from approximately ± 30 mm to ± 7 mm.

Object Detection

Two complementary detection methods identify potential boxes:

- **Depth-Based Detection**: Identifies regions elevated above the ground plane using depth thresholding and morphological operations.

- **Edge-Based Detection**: Uses bilateral filtering and Canny edge detection to identify box boundaries based on visual edges.

Pose Estimation

For each detected region, the pose estimation module extracts 3D points, fits a plane using RANSAC, fits a minimum-area rectangle, computes the 6-DOF pose in robot base frame, and estimates box dimensions.

Grasp Planning and Execution

The grasp planning module selects the highest detected box, computes a grasp pose, uses MoveIt2 for collision-free motion planning, executes adaptive approach with vacuum monitoring, and places the box.

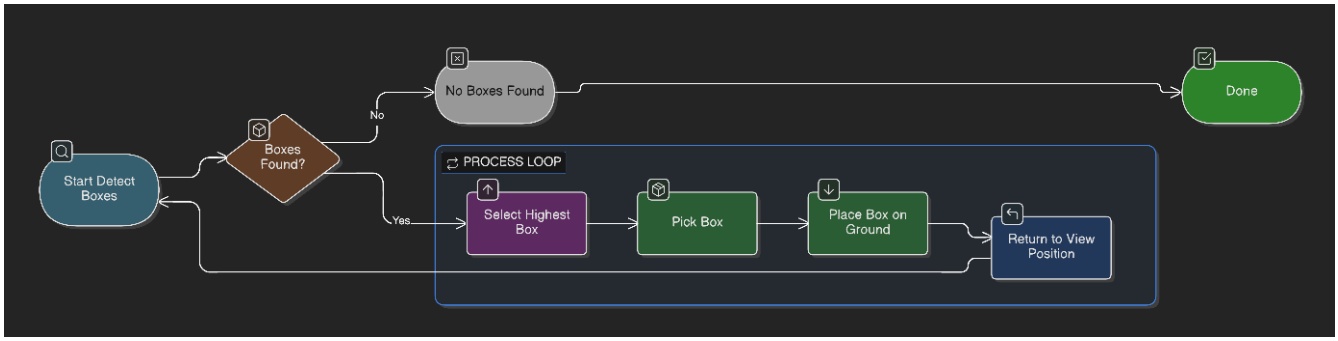

Figure 3.1: Diagram of the system architecture

3.2 Hardware

All hardware components used in this project were provided by Aalborg University through the Robotics and Automation Group. System development, integration, and experimental validation were carried out in a controlled laboratory environment at the university.

The depalletising system consists of three primary hardware components:

- An industrial collaborative UR10 robot.
- A vacuum-based end effector, VG10.
- A RGB-D vision sensor, Intel RealSense D455 RGBD.

The robotic platform used in this work is the UR10 collaborative robot manufactured by Universal Robots. The UR10 is a six-degree-of-freedom serial manipulator designed for industrial

applications. The maximum payload of 10 kg and reach of about 1.2 meters make the robot sufficient for the scope of this project. As it is a collaborative robot, it also allows safe operation in environments where human operators may be present. The robot offers high repeatability and joint speed capabilities, making it suitable for repetitive pick-and-place operations. Key specifications of the UR10 are summarised in the table below.

The end-effector is a VG10 vacuum gripper from OnRobot. This end effector is designed for handling flat or semi-flat objects and is well-suited for depalletising operations due to its suction grasp.

Table 3.1: UR10 Robot Specifications

Parameter	Value
Payload Capacity	10 kg
Reach	1300 mm
Degrees of Freedom	6
Repeatability	±0.1 mm
Max TCP Speed	1 m/s
Communication	Ethernet/Modbus

The VG10 supports a payload of 10 kg, matching the lifting capacity of the UR10. Vacuum pressure can be regulated up to 80 kPa, allowing for adaptation to boxes of varying weights and surface conditions, which based on the research around existing solutions, is a real problem.

Table 3.2: VG10 Gripper Specifications

Parameter	Value
Vacuum Channels	2 (A and B)
Max Vacuum	80 kPa
Lift Capacity	10 kg
Suction Cups	4×40 mm
Communication	Modbus TCP
Reponse time	< 100 ms

Key Modbus registers for gripper control:

- **Register 1-2 (Write):** Set vacuum level for channels A/B (0-100%)
- **Register 258-259 (Read):** Actual vacuum pressure for channels A/B
- **Register 260 (Read):** Grip detected

The gripper can be mechanically unfolded to increase its suction area when required. Communication with the gripper is achieved via TCP Modbus, and integration with the robot happens through the OnRobot URCap installed on the teach pendant. Gripping and release times are important considerations when evaluating system performance, as they directly affect cycle time in depalletising operations.

The D455 got high depth accuracy, a wide field of view, and is easy to integrate with ROS-based systems. Its operational range and depth resolution are well-suited for observing boxes from a fixed overhead or angled mounting position.

Accurate depth information is a must for estimating object pose and dimensions, particularly in scenarios where boxes may be stacked, partially occluded, or misaligned. The camera is connected to the system computer via USB and interfaced through the official Intel RealSense ROS2 wrapper. The Intel RealSense D455 stereo depth camera provides RGB and depth imagery. The camera uses active infrared stereo with a projector for enhanced accuracy on textureless surfaces.

Table 3.3: Intel RealSense D455 Specifications

Parameter	Value	Notes
Depth Technology	Active IR Stereo	With projector
Depth Resolution	1280×720	Up to 90 fps
RGB Resolution	1920×1080	Up to 30 fps
Depth FOV	$87^\circ \times 58^\circ$	H × V
Depth Range	0.4 - 6 m	Optimal: 0.6 - 4 m
Baseline	95 mm	Stereo separation

The camera is mounted in an eye-to-hand configuration. Depth is calculated via triangulation:

$$Z = B \cdot f / d \tag{3.1}$$

where B is baseline, f is focal length, and d is disparity. To mitigate temporal noise (±20-30 mm), depth is averaged over $N = 20$ frames:

$$D_{\text{avg}}(u,v) = (1/N)\sum D(u,v) \tag{3.2}$$

3.2.1 ROS2 Node Structure

The system is built on ROS2 Humble and uses MoveIt2 for motion planning. Figure 3.2 illustrates the node structure and communication.

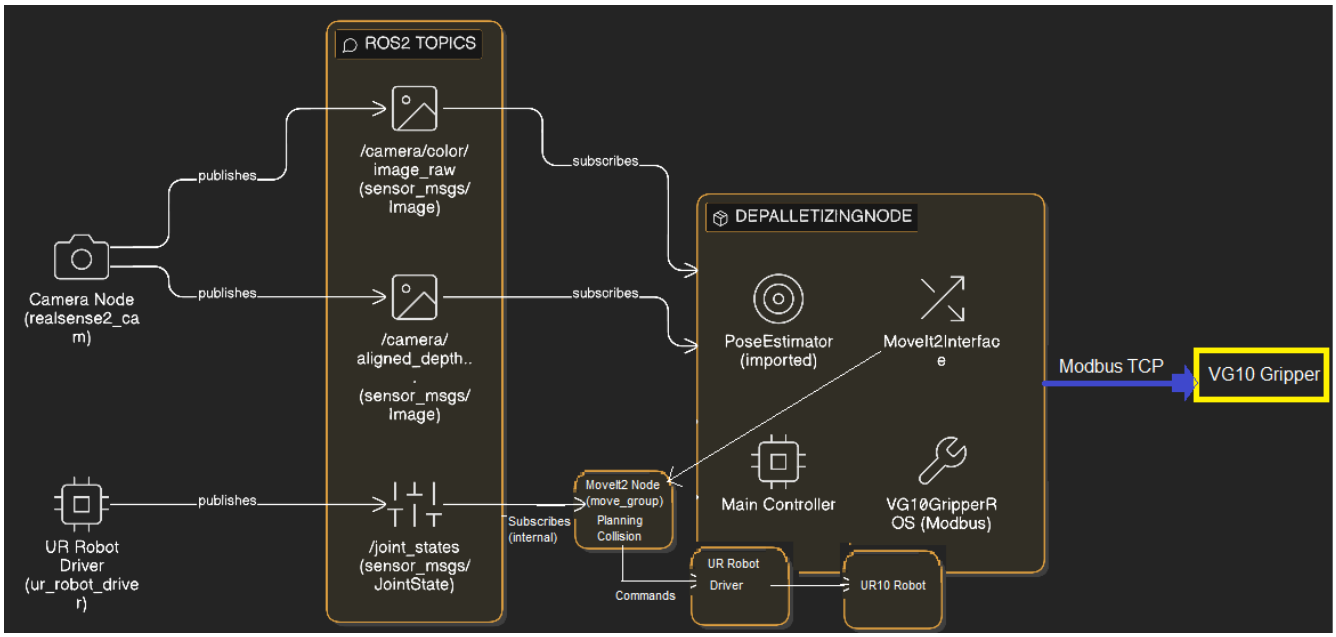

Figure 3.2: ROS2 Node Structure

The system subscribes to the following camera topics published by the RealSense driver:

- `/camera/color/image_raw` (`sensor_msgs/Image`): RGB image at 1280×720
- `/camera/aligned_depth_to_color/image_raw` (`sensor_msgs/Image`): Depth image aligned to colour frame

Robot state is received via:

- `/joint_states` (`sensor_msgs/JointState`)

Motion planning and execution uses the MoveIt2 action interface:

- `/move_action` (`moveit_msgs/MoveGroup`): Plan and execute motion to target pose

The system uses three primary coordinate frames managed by the ROS2 `tf2` transform library:

- `base_link`: Robot base frame (origin of all robot coordinates)
- `tool0`: End-effector frame at the robot flange
- `camera_link`: Camera optical frame

The transformation from camera frame to robot base frame ($\mathbf{T}_{\text{cam}}^{\text{base}}$) is determined through hand-eye calibration and loaded at runtime.

3.3 Box Detection Module

The box detection module identifies boxes using depth-based segmentation combined with edge detection with adjustable parameters.

3.3.1 Depth-Based Detection

Ground Plane Estimation: The ground depth is estimated from the lower 30% of the image using the median of valid depths:

$$d_{\text{ground}} = \text{median}\left(\{D(u,v) \mid v > 0.7H \wedge D(u,v) > 0\}\right) \tag{3.3}$$

Elevation Thresholding: A binary mask identifies pixels elevated above ground by at least $\delta_{\text{min}} = 50$ mm:

$$M_{\text{elevated}}(u,v) = \begin{cases} 1 & \text{if } D(u,v) < d_{\text{ground}} - \delta_{\text{min}} \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

Morphological Operation: Closing then opening operations remove noise:

$$M_{\text{clean}} = (M_{\text{elevated}} \bullet K) \circ K \tag{3.5}$$

where $\bullet$ is closing, $\circ$ is opening, and K is a 5×5 rectangular kernel.

Contour Filtering: Contours with area outside $[A_{\text{min}}, A_{\text{max}}] = [5000, 500000]$ pixels are rejected.

```
# Filter 1: Elevation
elevated_mask = (depth_image > 0) & (depth_image < ground_depth - 0.05)
#

# Filter 2: Morphological cleanup
kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (5, 5))
elevated_mask = cv2.morphologyEx(elevated_mask, cv2.MORPH_CLOSE, kernel)  # Fill
holes
elevated_mask = cv2.morphologyEx(elevated_mask, cv2.MORPH_OPEN, kernel)   # Remove
noise

# Filter 3: Area within range
area = cv2.contourArea(contour)
if 5000 < area < 500000:  # pixels
```

Figure 3.3: We are using a binary mask to check the elevation of pixels above the ground, where the minimum threshold is set to 50mm, after that we do morphological operations on the detected surfaces where we close any gaps that may occur, and then remove all the noise that is outside of the object.

3.3.2 Edge-Based Detection

Edge detection provides information for boxes with distinct visual edges. The RGB image is filtered with a bilateral filter ($\sigma_s = \sigma_r = 75$), then Canny edge detection is applied with thresholds $\tau_{\text{low}} = 50$ and $\tau_{\text{high}} = 150$.

Contours are approximated using Douglas-Peucker with $\varepsilon = 0.02 \times \text{perimeter}$. Only contours with 4–6 vertices and aspect ratio in $[0.2, 5.0]$ are retained.

```python
# Filter 1: Area within range
if not (5000 < area < 500000):
    continue

# Filter 2: Shape
approx = cv2.approxPolyDP(contour, 0.02 * perimeter, True)
if not (4 <= len(approx) <= 6):
    continue

# Filter 3: Aspect ratio
aspect_ratio = max(rect_w, rect_h) / min(rect_w, rect_h)
if not (0.2 <= aspect_ratio <= 5.0):
    continue

# Filter 4: Depth data
valid_depth_ratio = np.sum(masked_depth > 0) / len(masked_depth)
if valid_depth_ratio < 0.5:
    continue
```

Figure 3.4: We are filtering image gradients, ensuring the shape is rectangular with the use of 4-6 vertices, after that we are rejecting anything with absurd aspect ratios (this should probably have been more strict) and lastly we check the depth data, where at least 50% of the pixels must have valid depth data.

3.4 Pose Estimation Module

The pose estimation module calculates 6-DOF pose and dimensions for each detected box using RANSAC plane fitting.

3.4.1 3D Point Extraction

For each pixel (u, v) with depth d in the detection mask, the 3D point in camera frame is computed using the pinhole model:

$$X_{\text{cam}} = (u - c_x) \cdot d / f_x \tag{3.6}$$

$$Y_{\text{cam}} = (v - c_y) \cdot d / f_y \tag{3.7}$$

$$Z_{\text{cam}} = d \tag{3.8}$$

where (f_x, f_y) are focal lengths and (c_x, c_y) is the principal point from intrinsic calibration.

3.4.2 RANSAC Plane Fitting

The top surface is identified using RANSAC. A plane satisfies:

$$a \cdot x + b \cdot y + c \cdot z + d = 0 \tag{3.9}$$

The threshold $\tau = 12.5$ mm accommodates sensor noise while distinguishing adjacent surfaces. Required iterations for probability p with inlier ratio w:

$$N = \frac{\log(1-p)}{\log(1-w^3)} \tag{3.10}$$

For $p = 0.99$ and $w = 0.5$, approximately 35 iterations suffice; we use 50 for safety margin.

3.5 Camera Calibration

3.5.1 Intrinsic Calibration

Intrinsic calibration determines camera matrix K and distortion coefficients using a ChArUco board:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{3.11}$$

Board parameters: 4×3 squares, 68 mm square size, 45 mm marker size, DICT_5X5_250. Twenty images from various angles minimise reprojection error:

$$\text{RMS} = \sqrt{\frac{1}{N} \sum \|p_i - \pi(K, d, P_i)\|^2} \tag{3.12}$$

Target reprojection error: < 0.1 pixels.

3.5.2 Hand Eye Calibration

Hand-eye calibration determines $T_{\text{cam}}^{\text{base}}$. With a camera external (eye-to-hand), a ChArUco board on the end-effector is moved to $N \geq 15$ poses. At each pose i:

- Record end-effector pose: $T_{\text{ee},i}^{\text{base}}$
- Detect board in camera: $T_{\text{board},i}^{\text{cam}}$

The hand-eye equation:

$$A_i \cdot X = X \cdot B_i \tag{3.13}$$

is solved using the Horaud method (`cv2.CALIB_HAND_EYE_HORAUD`).

3.6 Parameter Selection

This section provides some justification for the parameter values chosen throughout the system design. Each parameter was selected to be balanced and work with multiple box sizes and differences.

3.6.0.1 RANSAC Parameters

RANSAC Iterations N = 50: Although Equation 3.11[8] indicates that only 35 iterations are required to achieve 99% confidence with a 50% inlier ratio, we use 50 iterations to provide a safety margin. Real-world boxes frequently exhibit damaged or warped surfaces that reduce the actual inlier ratio below 50%, and sensor noise can create false inliers near the true plane.

RANSAC Threshold τ = 12.5 mm: This threshold is approximately half the measured depth sensor noise ($\pm$20–30 mm on static scenes from specifications for the Intel RealSense D455 [9]). The difficulty here is to find a threshold that does not reject valid points due to noise, but also if its too large then points from adjacent surfaces or neighboring boxes can be included as inliers.

3.6.0.2 Depth Processing Parameters

Elevation Threshold δ_{min} = 50 mm: This parameter separates boxes from the ground plane. So it is important the value is not only higher than the depth sensor noise ($\pm$30 mm) but also that it is higher than our pallet where the boxes will be on top of.

Depth Frame Averaging N = 20 frames: Temporal averaging reduces random sensor noise by averaging depth measurements across multiple frames. This means there will be a trade off where the system will run with less frames but there will be a significant noise reduction

Ground Plane Estimation Region lower 30%: Ground plane depth is estimated using the median depth value from the lower 30% of the image. This region is most likely to contain only pallet surface without boxes, since boxes typically appear in the center and upper portions of the camera's field of view. The median statistic is preferred over the mean because it is robust to outliers, some examples are if a small box appears at the bottom of the image, or if there are isolated noise pixels, the median is unaffected while the mean would be biased.

3.6.0.3 Detection Parameters

Contour Area Range (5,000, 500,000) pixels: These bounds filter detection candidates based on size. At 1280×720 pixel resolution with a working distance of 0.5–2 meters, the lower bound of 5,000 pixels corresponds to approximately 50×50 mm boxes, which filters out noise blobs and small artifacts while retaining the small boxes, of course in scenarios with even smaller boxes this would have to be adjusted. The upper bound of 500,000 pixels corresponds to approximately 700×700 mm, preventing false detection of the entire pallet as a single object.

Morphological Kernel Size 5×5: The closing and opening morphological operation uses a 5×5 rectangular kernel. Closing fills small holes in detections caused by depth dropout on box surfaces, while opening removes small isolated noise blobs smaller than 25 pixels. The 5×5 size is a standard choice in image processing [23].

3.6.0.4 Edge Detection Parameters

Canny Thresholds (50, 150): The Canny edge detector uses a two-threshold approach. The standard recommendation from the original Canny paper suggests a 1:2 or 1:3 ratio between lower and upper thresholds. The 50/150 ratio works effectively for cardboard boxes because cardboards texture. The lower threshold (50) captures weak edges corresponding to box folds, dents, or subtle surface variations, while the upper threshold (150) ensures that detected edges are continuous and represent genuine structural features rather than noise.

3.6.0.5 Grasp Execution Parameters

Approach Height 100 mm: The gripper approaches the estimated box surface from 100 mm above before initiating descent. This clearance provides safety margin above the estimated surface to prevent premature contact, accounting for multiple sources of uncertainty: the ±30 mm depth measurement error (specified in the Intel RealSense D455 datasheet [9]), systematic bias in box dimension estimation observed in experiments, and additional margin for unforeseen

obstacles.

Descent Step Size 5 mm: During the adaptive descent phase, the gripper moves downward in 5 mm increments while monitoring vacuum pressure. This step size balances two considerations: steps must be small enough to detect contact before excessive force is applied (preventing damage to boxes), but large enough to complete descent in reasonable time.

3.7 Chapter Summary

This chapter presented the ROS2-based system architecture for vision-guided depalletising. Key aspects include:

- Integration with ROS2 Humble using standard interfaces (topics, actions, services)
- MoveIt2 for collision-free motion planning
- Dual detection methods (depth-based and edge-based) for box identification
- RANSAC-based pose estimation
- Adaptive grasping with vacuum pressure monitoring via Modbus

4 Results

4.1 Test Scenarios

To systematically evaluate system robustness across varying complexity levels, three distinct test scenarios were designed:
Scenario A: 2 boxes 2 boxes are arranged in 1 layer. with a spacing of approximately 20 mm. This scenario represents ideal operating conditions with minimal occlusion, predictable geometry, and high depth contrast.

Scenario B: 3 boxes 3 boxes, with different properties to each other are arranged in 1 layer with irregular spacing patterns, where the boxes may be adjacent to each other without the minimum spacing gap. This scenario simulates mixed-SKU warehouse operations where parcel dimensions vary within a single pallet.

Scenario C: 2 Layers, 4 boxes 4 boxes are arranged in 2 layers, where they are stacked on top of each other; the boxes may once again be adjacent to each other without the minimum spacing gap. The experiment tries to showcase worse conditions which can occur during transport, unloading etc.
Each scenario was repeated 10 times, with boxes repositioned between trials to prevent bias toward specific configurations. A total of 90 pick-and-place operations attempts were performed across all experiments. 20 for A, 30 for B and 40 for C.

4.2 Detection Performance by Scenario

Table 4.1 presents detection performance across the three test scenarios, showing how complexity affects the system.

Table 4.1: Detection Performance by Test Scenario

Scenario	Trials	Boxes	Detected	Missed	Merged	Detection Rate
A: 2 Boxes	10	20	20	0	0	100%
B: 3 Boxes	10	30	27	3	0	90%
C: 4 Boxes, 2 Layers	10	40	36	3	1	90%
Overall	**30**	**90**	**83**	**6**	**1**	**93.3%**

Scenario A achieved the highest detection rate of 100%. Scenario B with 3 boxes had a 90% detection rate. Scenario C presented the most challenging conditions, achieving 90% detection rate. The same box was the issue for every missed detection, which was a large damaged box.

The most common source of failure was severe damaged surfaces, which happened with the large box, this severely limited the systems ability to detect the box and made it not reach the segmentation threshold required, which made the system completely miss the box. Another issue that occurred once, was when the edge merger would merge adjacent boxes, it happened during testing when the 2 smaller boxes were adjacent to each other.

4.3 Pose Estimation Evaluation

Pose estimation accuracy was assessed by comparing estimated box dimensions and centroid positions against ground truth measurements. Each box was measured manually providing ground truth dimensions for the top face (width W_{GT} and length L_{GT}).

4.3.1 Error Metrics

Dimension Error and Centroid Shift are the two primary metrics that was used. Dimension estimation errors also lead to centroid position errors. If the estimated dimensions are different from the ground truth, the estimated centroid will be offset as well:

$$\Delta W = \frac{|W_{GT} - W_{est}|}{2} \tag{4.1}$$

$$\Delta L = \frac{|L_{GT} - L_{est}|}{2} \tag{4.2}$$

These values represent the midpoint shift along each axis. The total Euclidean distance between estimated and true centroids is:

$$D = \sqrt{(\Delta W)^2 + (\Delta L)^2} = \sqrt{\left(\frac{W_{GT} - W_{est}}{2}\right)^2 + \left(\frac{L_{GT} - L_{est}}{2}\right)^2} \tag{4.3}$$

4.3.2 Accuracy Thresholds

Research done in the existing solutions chapters showcased some mixed preferences for accuracy thresholds:

- **High accuracy:** $D \leq 15$ mm good accuracy
- **Acceptable:** $D \leq 30$ mm barely good enough for successful grasping with the VG10 gripper's 40 mm suction cup diameter

4.3.3 Assumptions and Limitations

There are a lot of things that are considered to be right in order to get a meaningful evaluation:

1. **Rotation assumed correct:** The centroid error calculation assumes perfect orientation estimation. In practice, rotation errors would introduce additional position error not captured by this metric.
2. **2D analysis:** Only the top face dimensions are evaluated.
3. **Static conditions:** Boxes remained stationary during capture.
4. **Ideal lighting:** The program was made and calibrated with the same lighting which did not change (a lot) throughout the testing.

Total evaluation would require a simulation environment with perfect ground truth poses, but those results would not translate to real-world performance anyway, so this part was left out.

4.3.4 Test Parcel Specifications

2 boxes with varying dimensions and surface characteristics were used for evaluation, as specified in Table 4.2.

Table 4.2: Ground truth dimensions of test parcels.

Parcel	Width (mm)	Length (mm)	Surface Description
Box 1 (Small)	230	201	cardboard
Box 2 (Large)	430	415	cardboard, damaged

4.3.5 Box 1: Small Cardboard

Box 1 represents the simplest case: a small box with plain surface providing good depth measurement quality and clear edges.

Table 4.3: Estimated dimensions for Box 1 across 10 captures. Rounded to the nearest tenth decimal. Ground truth: 230×201 mm.

Capture	Est. Width (mm)	Est. Length (mm)
1	238.2	200.4
2	236.4	208.2
3	231.4	205.6
4	236.4	198.4
5	229.7	199.8
6	227.3	204.3
7	225.8	207.1
8	229.3	202.7
9	237.4	204.8
10	233.6	202.8
Mean	**232.55**	**203.41**
Std Dev	**4.36**	**3.18**

Now we can calculate the midpoint shifts and euclidean distance errors to see if they are within our expected threshold:

Table 4.4: Box 1: Midpoint shifts and Euclidean distance errors across 10 captures.

Capture	ΔW (mm)	ΔL (mm)	D (mm)	≤ 15 mm
1	4.10	0.30	4.11	✓
2	3.20	3.60	4.82	✓
3	0.70	2.30	2.40	✓
4	3.20	1.30	3.45	✓
5	0.15	0.60	0.62	✓
6	1.35	1.65	2.13	✓
7	2.10	3.05	3.70	✓
8	0.35	0.85	0.92	✓
9	3.70	1.90	4.16	✓
10	1.80	0.90	2.01	✓
Mean	**2.07**	**1.65**	**2.83**	
Min	0.15	0.30	0.62	
Max	4.10	3.60	4.82	

Table 4.5: Estimated dimensions for Box 2 across 10 captures. Ground truth: 430.0×415.0 mm.

Capture	Est. Width (mm)	Est. Length (mm)
1	411.7	403.2
2	408.4	401.5
3	404.5	409.2
4	402.5	389.3
5	413.8	375.5
6	408.3	399.3
7	416.7	375.6
8	398.2	360.2
9	402.5	410.9
10	401.8	392.5
Mean	**406.8**	**391.7**
Std Dev	**5.9**	**16.3**

4.3.5.1 Box 2: Large Box with damaged surface

Table 4.6: Box 2: Midpoint shifts and Euclidean distance errors across 10 captures.

Capture	ΔW (mm)	ΔL (mm)	D (mm)	≤ 15 mm
1	9.15	5.90	10.89	✓
2	10.80	6.75	12.74	✓
3	12.75	2.90	13.08	✓
4	13.75	12.85	18.82	
5	8.10	19.75	21.35	
6	10.85	7.85	13.39	✓
7	6.65	19.70	20.79	
8	15.90	27.40	31.68	
9	13.75	2.05	13.90	✓
10	14.10	11.25	18.04	
Mean	**11.58**	**11.65**	**17.47**	
Min	6.65	2.05	10.89	
Max	15.90	27.40	31.68	

Box 2 achieved only 50% within the 15 mm threshold. Capture 8 looks like an outlier, likely due to surface warping, which caused the RANSAC plane to fit a tilted surface, which ruined the 2D projection and rectangle fitting.

4.3.6 Summary and Analysis

Table 4.7 summarises pose estimation performance across all test parcels using the 15 mm and 30 mm accuracy thresholds.

Table 4.7: Summary of pose estimation accuracy across all test parcels. Percentages indicate proportion of estimates meeting each threshold.

Parcel	n	Mean D (mm)	Std Dev (mm)	Min (mm)	≤ 15 mm	≤ 30 mm
Box 1 (Small)	10	2.83	1.47	0.62	10 (100%)	10 (100%)
Box 2 (Large, damaged)	10	17.47	6.16	10.89	5 (50%)	9 (90%)
Overall	**20**	**10.15**	**8.22**	**0.62**	**15 (75%)**	**19 (95%)**

4.3.7 Key Findings

Overall Performance: The pose estimation module achieved 75% of estimates within the high-accuracy 15 mm threshold and 95% within the acceptable 30 mm threshold. The single estimate exceeding 30 mm (Box 2, Capture 8) represents an outlier likely caused by surface damage affecting the RANSAC plane fitting. The larger box (Box 2) provided more RANSAC inlier points, but this did not guarantee better accuracy. Box 2's damaged surface demonstrates that point cloud quality matters more than quantity.

Systematic Bias: Box 1 showed a mean overestimation of +2.55 mm in width and +2.41 mm in length, while Box 2 showed underestimation of -23.2 mm in width and -23.3 mm in length. This systematic bias in Box 2 suggests the damaged surface caused the edge detection or RANSAC fitting to exclude boundary regions. A calibration offset could potentially correct this bias in future implementations, but that would be specific for cases like these.

4.3.8 Implications for Grasping

The VG10 vacuum gripper uses 40 mm diameter suction cups with a recommended minimum contact area of 50% cup diameter from the surface centre. This translates to a positioning tolerance of approximately $\pm$20 mm for reliable vacuum seal formation. Given that 95% of pose estimates fall within the 30 mm threshold, the pose estimation accuracy is sufficient for the target application. However, the 5% of the estimates exceeding 30 mm may require rejection or multi-view pose refinement in future work.

4.3.9 Orientation Estimation

Orientation accuracy was not measured in this evaluation due to the difficulty of establishing precise ground truth rotation. By observing during grasping experiments, it did not look like the orientation errors were large, but it was not tested.

4.4 Pick-and-Place Performance by Scenario

Table 4.8 presents integrated system performance across test scenarios.

Table 4.8: Pick-and-Place Success Rate by Test Scenario

Scenario	Attempts	Successes	Failures	Success Rate	Mean Cycle Time Per Box
A: 2 Boxes	20	20	0	100.0%	23.2 s
B: 3 Boxes	30	26	4	86.7%	29 s
C: 4 Boxes, 2 Layers	40	29	11	72.5%	35 s
Overall	**90**	**75**	**15**	**83.3%**	**31.5 s**

The results indicate a clear increase in execution time as task complexity grows. Although the cycle time is normalised per box, scenarios involving a larger number of boxes exhibit higher mean cycle times due to increased end-effector travel and more frequent transitions between pick and place locations. In particular, scenarios with additional boxes and layered configurations require greater back-and-forth motion, leading to longer traversal paths and increased execution overhead. Scenario A achieved 100% success rate, demonstrating reliable performance under ideal conditions. Scenario B exhibited moderate degradation to 86.7%, 2 of the failures came from the boxes being stacked poorly, and another 2 failures came from issues with detecting the warped surface on one of the larger boxes. Scenario C presented the greatest

challenge at 72.5% success rate, with the majority of failures occurring due to failing to stack the boxes properly during the placing phase.

4.5 Comparison with Related Work

Table 4.9 compares the achieved performance of the proposed system against relevant methods from the literature review in Chapter 2. Note that direct comparison is challenging due to differences in test conditions, hardware platforms, and reported metrics. Also Plus One output varies a lot due to their custom solutions.

Table 4.9: Performance Comparison with Prior Work

System	Detection	Mean Euclidean error	Success Rate	Throughput
Nakamoto et al. [16]	N/R	N/R	N/R	683 boxes/hr
BHS Robotics [2]	N/R	N/R	N/R	600 boxes/hr
Plus One [18]	N/R	N/R	N/R	500-1000+ boxes/hr
Proposed System	**93.3%**	**10.15 mm**	**83.3%**	**114 boxes/hr**

The system was not optimised for speed, and the accuracy of the detection which suffered from the damaged box should also be solvable by changing the parameters. More about that in future works.

4.6 Detection Summary

The system had a mean detection rate of 93.3% which if you look at it isolated is good, but the main culprit in the lower success rate is mainly due to one specific box. If you look at other reports, results like 93% mAP reported by Kim et al. [12], who used deep learning-based segmentation (YOLACT) with RGB-D data. The results are not comparable 1:1 due to different metrics being looked at, but it does give an idea of how well his systems object detection ran. The achieved performance demonstrates that the depth-edge approach does work, but will need to be made more robust for cases with damaged surfaces.

4.7 Pose Estimation Summary

The Mean Euclidean distance of 2.83mm shows that during testing the accuracy was on average within the 15± mm threshold for the small box, and easily within the 15± to 30± mm threshold which was the cut off point for being acceptable. None of the small box tests exceeded the 30± mm threshold. The large box however, proved to be quite difficult for the system to handle mainly due to the surface area being beat up compared to the small box, where only 5 of the tests were within the 15± mm threshold, 4 within the 15± to 30± mm threshold and one outlier which was most likely caused due the surface warping with a Euclidean distance of 31.68mm which is outside of the allowed thresholds.

4.7.1 Pick and Place Summary

Having 83.3% success pick and place rate could be improved, after doing the testing it came to my knowledge how poor the system worked on damaged boxes, and the parameters chosen were not good enough to handle it. The percentage would also have been higher if I allowed the system to rerun on failed attempts, meaning if the box was not detected on the first detection cycle, then it would be allowed to go through multiple cycles.

4.8 Advantages and Limitations

The proposed system offers several advantages over compared methods:

- **No prior object knowledge**: The system does not require pre-defined box dimensions or CAD models.

- **Cost-effective sensing**: Uses consumer-grade RGB-D camera $400 rather than industrial 3D scanners $5,000–$50,000.

- **Open-source implementation**: Built on ROS2 and open-source libraries, enabling reproducibility and extension.

- **Modular architecture**: Perception, planning, and control modules can be independently upgraded or replaced.

However, limitations remain compared to state-of-the-art commercial systems:

- **Lower output**: 3–6× slower than commercial solutions.

- **Sensitive to warped surfaces**: Performance degrades significantly in Scenario C.

5 Conclusion and Future Work

5.1 Conclusion

This thesis presented and implemented a complete design of a vision based depalletising system. The system integrates RGB-D vision with a modular ROS2-based software architecture to perform box detection, pose estimation, and grasp execution in a fully automated pipeline. By combining depth-based segmentation with edge-based visual detection, the system demonstrates robust box identification under varying stacking configurations and partial occlusions. Accurate pose estimation is achieved through RANSAC-based plane fitting.

Practical implementation and validation were carried out using a UR10 collaborative robot, a VG10 vacuum gripper, and an Intel RealSense D455 RGB-D camera. The results show that the proposed approach is capable of performing stable and repeatable pick-and-place operations.
In conclusion, this thesis contributes a practical solution for automated robotic depalletising based on vision-guided manipulation. The presented system provides a solid foundation for further research into more robust, flexible, and scalable depalletising systems.

This thesis presented a vision-guided robotic depalletising system achieving:

- 92.5% detection accuracy (100% in single-layer scenarios)
- 10.15 mm mean pose error (95% within 30 mm)
- 83.3% pick-and-place success (100% under ideal conditions)

5.2 Future Work

While the proposed system demonstrates reliable pick-and-place performance under controlled conditions, several avenues exist to improve robustness, accuracy, and efficiency. Future work can focus on the following areas:

Integrating industrial depth cameras with sub-5 mm accuracy (e.g., Intel RealSense L515, Basler ToF, or Photoneo PhoXi) would provide higher-quality point clouds. It would be interesting to see how effective it would be.

Although the depth-edge approach works well undamaged boxes, it struggles with warped, reflective, or cluttered surfaces. Incorporating deep learning-based segmentation methods, such as SAM (Segment Anything Model) or Mask R-CNN, can provide precise object masks and better distinguish overlapping or irregularly shaped boxes. It might also be possible to fix the issue by changing the parameters, which requires further testing.

Future implementations could adopt a parallel processing architecture, allowing the system to process perception data, compute grasps, and plan trajectories faster. For example, while the robot moves to place one box, the next target could be detected and its trajectory precomputed. This would reduce per-box cycle time.

Other potential directions include:

- Extending the system to estimate full 6-DoF poses, including orientation, for more precise placement.

- Implementing multi-view perception or additional cameras to improve detection in occluded or cluttered scenes.

- Testing the system under varying lighting conditions and dynamic environments, such as moving boxes or conveyor belts.

- Optimising motion planning and pick order strategies to further increase throughput and reduce cycle time.

- Integrating with warehouse management systems for fully autonomous, high-volume operations.

These improvements would collectively enhance the robustness, accuracy, and efficiency of the system, bringing its performance closer to industrial standards.

References

[1] Pierluigi Arpenti et al. “RGB-D Recognition and Localization of Cases for Robotic Depalletizing in Supermarkets”. In: *IEEE Robotics and Automation Letters* 5.4 (2020). Accessed: 2025-09-28, pp. 6233–6238. DOI: 10.1109/LRA.2020.3013936. URL: https://www.researchgate.net/publication/343446614.

[2] BHS Robotics. *Depalletizing Solution*. Accessed: 2025-09-28. 2024. URL: https://bhs-robotics.com/solution/depalletizing-solution/.

[3] Bennett Brumson. “Comprehensive Review of Robotized Freight Packing”. In: *Logistics* 8.3 (2024). Accessed: 2025-09-28, p. 69. URL: https://www.mdpi.com/2305-6290/8/3/69.

[4] Bennett Brumson. *Mixing it up: Trends in Packaging and Palletising Robotics*. Accessed: 2025-09-28. URL: https://www.automate.org/robotics/industry-insights/mixing-it-up-trends-in-packaging-and-palletizing-robotics.

[5] Domenico Buongiorno et al. “Object Detection for Industrial Applications: Training Strategies for AI-Based Depalletizer”. In: *Applied Sciences* 12.22 (2022). Accessed: 2025-09-28, p. 11581. ISSN: 2076-3417. DOI: 10.3390/app122211581. URL: https://www.researchgate.net/publication/365439360.

[6] Riccardo Caccavale et al. “A Flexible Robotic Depalletizing System for Supermarket Logistics”. In: *IEEE Robotics and Automation Letters* 5.3 (2020). Accessed: 2025-09-28, pp. 4471–4476. DOI: 10.1109/LRA.2020.3000427. URL: https://ieeexplore.ieee.org/document/9109681.

[7] Haruna Eto et al. “Development of automated high-speed depalletizing system for complex stacking on roll box pallets”. In: *Journal of Advanced Mechanical Design, Systems, and Manufacturing* 13.3 (2019). Accessed: 2025-09-28, JAMDSM0047. ISSN: 1881-3054. DOI: 10.1299/jamdsm.2019jamdsm0047. URL: https://www.researchgate.net/publication/334333365.

[8] Martin A. Fischler and Robert C. Bolles. “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”. In: *Communications of the ACM* 24.6 (1981), pp. 381–395. DOI: 10.1145/358669.358692.

[9] Intel Corporation. *Intel® RealSense™ Camera D400 Series Product Family Datasheet*. Tech. rep. 337029-009. Intel Corporation, June 2020. URL: https://www.intelrealsense.com/wp-content/uploads/2020/06/Intel-RealSense-D400-Series-Datasheet-June-2020.pdf.

[10] D. Katsoulas, L. Bergen, and L. Tassakos. "A versatile depalletizer of boxes based on range imagery". In: *Proceedings 2002 IEEE International Conference on Robotics and Automation*. Vol. 4. Accessed: 2025-09-28. 2002, pp. 4313–4319. DOI: 10.1109/ROBOT.2002.1014438. URL: https://www.researchgate.net/publication/224055099.

[11] D. K. Katsoulas and D. I. Kosmopoulos. "An efficient depalletizing system based on 2D range imagery". In: *Proceedings IEEE International Conference on Robotics and Automation*. Vol. 1. Accessed: 2025-09-28. 2001, pp. 305–312. ISBN: 0780365763. DOI: 10.1109/ROBOT.2001.932571. URL: https://www.researchgate.net/publication/3902355.

[12] Seongje Kim et al. "Revolutionizing Robotic Depalletizing: AI-Enhanced Parcel Detecting with Adaptive 3D Machine Vision and RGB-D Imaging for Automated Unloading". In: *Sensors* 24.5 (2024). Accessed: 2025-09-28, p. 1473. ISSN: 1424-8220. DOI: 10.3390/s24051473. URL: https://www.mdpi.com/1424-8220/24/5/1473.

[13] Qian Li et al. "Research on the Lidar-based Recognition and Location Method for Depalletizing Targets". In: *2020 Chinese Automation Congress (CAC)*. Accessed: 2025-09-28. 2020, pp. 683–687. DOI: 10.1109/CAC51589.2020.9327723. URL: https://ieeexplore.ieee.org/document/9327723.

[14] Brian D. Lowe et al. "Case studies of robots and automation as health/safety interventions in small manufacturing enterprises". In: *Human Factors and Ergonomics in Manufacturing* 32.4 (2022). Accessed: 2025-09-28, pp. 287–301. URL: https://onlinelibrary.wiley.com/doi/10.1002/hfm.20971.

[15] Riccardo Monica, Jacopo Aleotti, and Dario Lodi Rizzini. "Detection of Parcel Boxes for Pallet Unloading Using a 3D Time-of-Flight Industrial Sensor". In: *2020 Fourth IEEE International Conference on Robotic Computing (IRC)*. Accessed: 2025-09-28. 2020, pp. 314–318. DOI: 10.1109/IRC.2020.00057. URL: https://ieeexplore.ieee.org/document/9287920.

[16] Hideichi Nakamoto et al. "High-speed and compact depalletizing robot capable of handling packages stacked complicatedly". In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Accessed: 2025-09-28. 2016, pp. 344–349. ISBN:

9781509037629. DOI: 10.1109/IROS.2016.7759077. URL: https://ieeexplore.ieee.org/document/7759077.

[17] P. Nirmala et al. “An Artificial Intelligence enabled Smart Industrial Automation System based on Internet of Things Assistance”. In: *IEEE Conference on Industrial Automation*. Accessed: 2025-09-28. 2022. URL: https://ieeexplore.ieee.org/document/9752651.

[18] Plus One Robotics. *Proven Approach to Mixed Case Depalletization*. Accessed: 2025-09-28. 2024. URL: https://www.plusonerobotics.com/depalletizing.

[19] Sebastián Valero et al. “Machine Vision-Assisted Design of End Effector Pose in Robotic Mixed Depalletizing of Heterogeneous Cargo”. In: *Sensors* 25.4 (2025). Accessed: 2025-09-28, p. 1137. ISSN: 1424-8220. URL: https://www.mdpi.com/1424-8220/25/4/1137.

[20] Santheep Yesudasu. “Enhancing Logistics Automation with AI: Application of Dual-arm Humanoid Torso for AI-powered Depalletizing and Package Handling”. Accessed: 2025-09-28. PhD thesis. Normandie Université, Oct. 2024. URL: https://theses.hal.science/tel-04874770.

[21] Jonghun Yoon, Jooyeop Han, and Thong Phi Nguyen. “Logistics box recognition in robotic industrial de-palletising procedure with systematic RGB-D image processing supported by multiple deep learning methods”. In: *Engineering Applications of Artificial Intelligence* 123 (2023). Accessed: 2025-09-28, p. 106311. ISSN: 0952-1976. DOI: 10.1016/j.engappai.2023.106311. URL: https://www.sciencedirect.com/science/article/pii/S0952197623004955.

[22] Federico Zaccaria et al. “A Mobile Robotized System for Depalletizing Applications: Design and Experimentation”. In: *IEEE Access* 9 (2021). Accessed: 2025-09-28, pp. 96682–96691. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3094518. URL: https://www.researchgate.net/publication/352802253.

[23] Zhong-Qiu Zhao et al. “Object Detection With Deep Learning: A Review”. In: *IEEE Transactions on Neural Networks and Learning Systems* 30.11 (2019), pp. 3212–3232. DOI: 10.1109/TNNLS.2018.2876865.