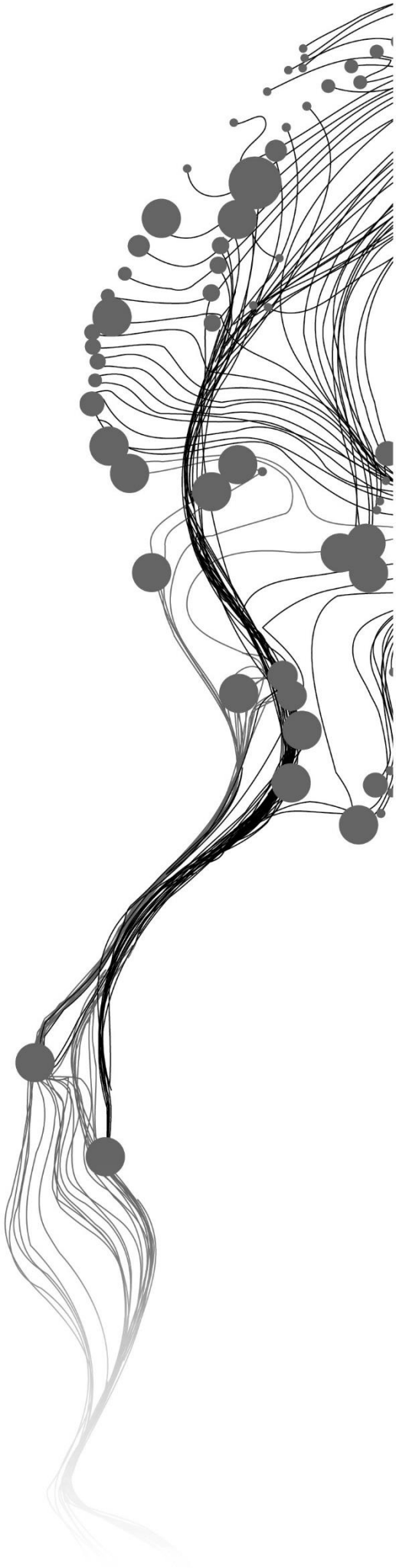
# DETECTION OF OBJECTS AND THEIR ORIENTATION FROM 3D POINT CLOUDS IN AN INDUSTRIAL ROBOTICS SETTING

DEVI DARSHANA SREDHAR
July 2021

SUPERVISORS:
Dr. Ville. V. Lehtola
Dr. Ir. S. J. Oude Elberink



# DETECTION OF OBJECTS AND THEIR ORIENTATION FROM 3D POINT CLOUDS IN AN INDUSTRIAL ROBOTICS SETTING

DEVI DARSHANA SREDHAR
Enschede, The Netherlands, July 2021

Thesis submitted to the Faculty of Geo-Information Science and Earth Observation of the University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation.
Specialization: Geoinformatics

SUPERVISORS:
Dr. Ville. V. Lehtola
Dr. Ir. S. J. Oude Elberink

THESIS ASSESSMENT BOARD:
Prof. Dr. Ir. M.G. Vosselman (chair)
Dr. Petri Rönnholm, Dept of Built Environment, Aalto University, Finland
Drs. J.P.G. Bakx
Dr. Ville. V. Lehtola
Dr. Ir. S. J. Oude Elberink

# ABSTRACT

Lidar techniques are highly suitable for employing in industrial setups such as in the automatic unloading of cargo containers. However, the restrictions on the sensor positions allow the cargo container to be scanned only from a certain position and angle. The varying point density in the point cloud data because of the scanning geometry affects the detection of individual instances of similar objects. Here, we study such a lidar system to detect and obtain the positions of the objects present in the scene to manipulate them.

This research leverages the available information from single-shot lidar representations of open cargo containers stacked with box-like objects. The study uses a direct point cloud segmentation technique as the baseline method and explores an alternate approach by employing a projection-based point cloud segmentation method to find a solution. The problem of varying point density is handled by increasing the footprint of the laser points using a uniform kernel during the projection of point cloud data to an image. The projected point cloud data is then segmented using the watershed method to detect the number of objects. The study also compares the two segmentation methods – the segment growing method used for direct point cloud segmentation and the watershed method. The results are evaluated quantitatively and qualitatively.

Furthermore, we obtain the object pose with six degrees of freedom and extract the object dimensions to be communicated to the robotic manipulator for unloading the container. With these properties, in future work, the objects could be identified in the real world.

**Keywords**: Industrial Robotics, 3D point clouds, Machine vision in container unloading, Projection-based point cloud segmentation, Single-view lidar scan, Range image segmentation, Watershed Segmentation.

# ACKNOWLEDGEMENTS

Foremost, I would like to thank my first supervisor Dr. Ville. V. Lehtola for his guidance, support and encouragement throughout the research period. His thought-provoking questions, valuable comments, and discussions since the early phases of my research have helped me shape and refine my work.

I want to thank my second supervisor, Dr. Ir. S.J. Oude Elberink who has also been highly supportive and guided me all through my work. I thank him for his patience, encouragement, suggestions and prompt response every time I had questions.

I am greatly indebted to my chair Prof. Dr. Ir. Vosselman for his critical evaluation and suggestions that contributed to the quality of my research. I also extend my thanks to drs. J.P.G Wan Bakx for his guidance in my academic life at ITC.

I extend my gratitude to all the teaching faculty and staff who made my experience at ITC enjoyable and ITC Excellence Scholarship for financially supporting my education.

Finally, I would like to thank my family for believing in me and all my friends for being with me during good and stressful times.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# 1. INTRODUCTION

## 1.1. Background

A large amount of cargo is transported worldwide, and its handling is a very tedious task when done manually (Vaskevicius, Pathak, & Birk, 2017). It is time-consuming and imposes potential health risks to the employees involved. There is a risk of damage that could be caused to the product during its handling. Unloading cargo thus calls for a logistics automation process to be in place to overcome the problems of health risks involved, compensate for the labor shortage and to speed up the process. An automated robotic system designed to unload containers should effectively handle goods of different sizes, shapes, and weights while at the same time, be able to manage a picking success rate similar to that of a human (Stoyanov et al., 2016). The robotic system must understand the items to be handled and the different circumstances under which they are found to automate the process successfully. The successful detection, unloading and caution towards not damaging the product need to be considered. A high level of autonomy is required, which can be achieved through systems capable of sensing the environment, understanding the objects, making decisions based on the obstacles, and interacting with the scene to maneuver the objects without constant human supervision. Even if the system is intended for a wide variety of goods oriented in any direction in space, the goods may become ungraspable due to scenarios such as movement during its transportation (Bonini et al., 2015). Therefore, even standardized loading cannot ensure reliable detection of objects during their unloading, as their positions are altered during transportation. The need for a robust object detection method to improve the process of automation and increase the usage of robotic systems for unloading cargo containers hence becomes essential. This benefits industrial applications and results in a user-friendly labor environment by removing the manual burden on the laborers. It could in turn be realized as positive business growth in such sectors.

The automated systems consist of a robot, a gripper mechanism and a unit to detect objects for unloading goods from a container (Kirchheim, Burwinkel, & Echelmeyer, 2008). When the target field is complex, the optimal selection of sensors for detection is vital for the type of application. In an industrial setting, the accurate position and dimensions of the individual objects in 3d space are required (Choi et al., 2012). Using a LiDAR (Light Detection and Ranging) system, data is collected with depth information and properties that define objects in 3d space. Such 3d data allows to recognize instances of scene objects and estimate each of their poses with six degrees of freedom (three translation and three rotation), which enables manipulating such objects precisely (Aldoma et al., 2013). The increasing research in 3d modeling and 3d object recognition techniques from laser point clouds make it suitable for employing it for the automation process in industrial setups (Elseberg, Borrmann, & Nüchter, 2011). Due to their short range, they can generate dense point cloud data to represent a scene (Soulard & Bogle, 2011). Industrial robots, which are employed for object grasping, work by detecting the objects and their positions.

Object detection for the cargo unloading process is considered a combined task of object recognition at the instance level and the estimation of each object's pose with respect to the sensor (Rudorfer, 2016). The point cloud of each object needs to be distinguished to extract pose details of objects. For this, the scene must be segmented to provide boundaries for each instance present in it. The segmentation task is performed on an acquired dataset to simplify and analyze the nature and the number of objects present in a scene (G. Vosselman & Maas, 2010). Segmenting algorithms can be extended to either perform semantic or instance segmentation based on the purpose of the application. Instance segmentation considers

multiple objects of the same class as individual instances as opposed to semantic segmentation (Elich, Engelmann, Kontogianni, & Leibe, 2019). The point cloud is segmented at the instance level, the pose and dimensions of each of them are estimated in order for the robotic system to understand the objects and unload them.

### 1.1.1. Problem Statement

This research focuses on automating the container unloading process using the system presented in **Figure 1-1**. In an industrial setup, the positioning of lidar and robotic systems is constrained by logistical and spatial elements. The position of the lidar system (red) used and the robotic manipulator (yellow) is annotated in the figure. It is considered for this study that the objects (cargo) to be detected are boxes of different sizes stacked from top to bottom of the same or different products. The dimensions are not uniform over the entire scene and the objects present vary in their alignment and orientation. It is challenging to develop a segmenting algorithm that would distinguish individual object instances with only small gaps between them to help estimate the pose details and dimensions. This would enhance the existing process of handling cargo. The figure below shows an automatic system unloading cargo boxes from an open container. **Figure 1-1** also visualizes the gaps between two objects are different (some boxes are compactly packed, while some have significant gaps ).

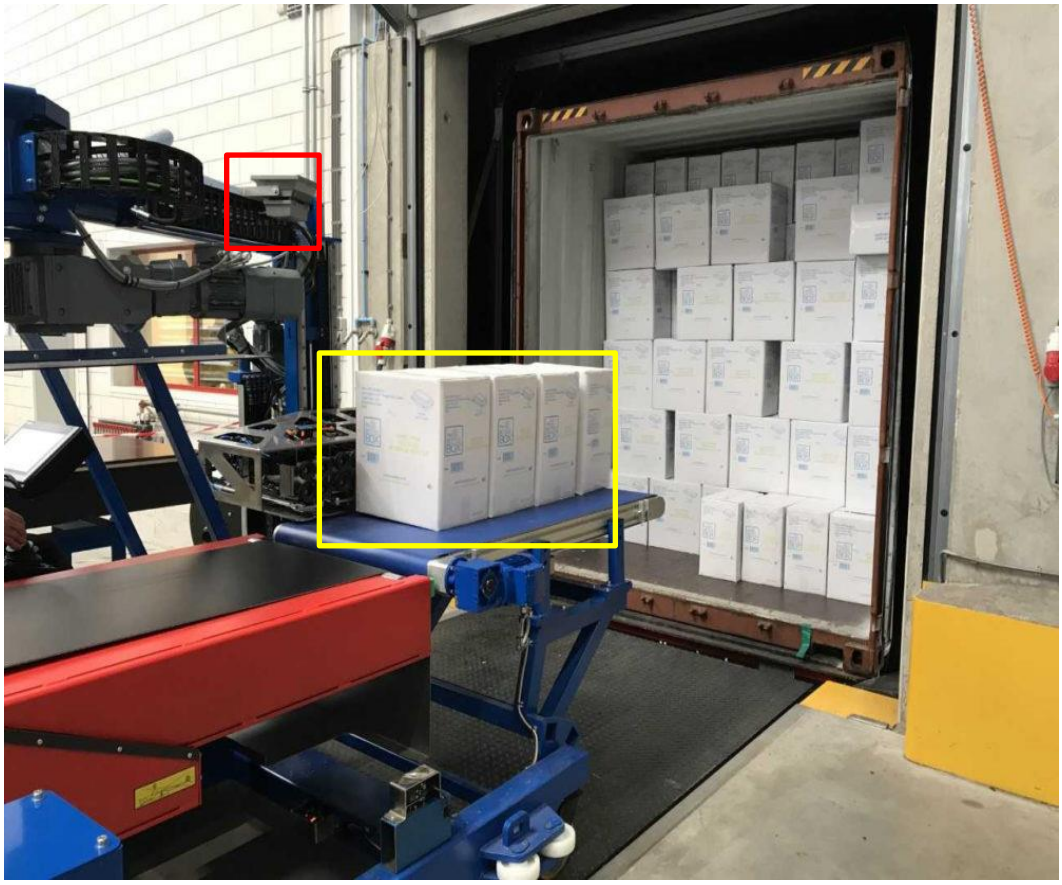


Figure 1-1 An automatic robotic system unloading cargo boxes from an open container; mounted lidar system highlighted in red; robot manipulator unloading four cargo boxes (yellow)

The container unloading scenario deals with objects belonging to the same class (cargo boxes). Since the cargo boxes are all characterized similarly, performing an instance-based segmentation adds more value to the specific task of auto unloading a cargo container. These box-shaped objects are compactly packed, but sometimes there are gaps between them. The acquired 3d point cloud of a similar scene is visualized in

**Figure 1-2**. If the gripper mechanism of the system were to position itself at the gaps, it might fail at picking the object (Doliotis, McMurrough, Criswell, Middleton, & Rajan, 2016). This increases the need for accurate boundaries around each target object. Thus, a segmentation algorithm that can distinguish each box object based on the thin boundaries present is required.

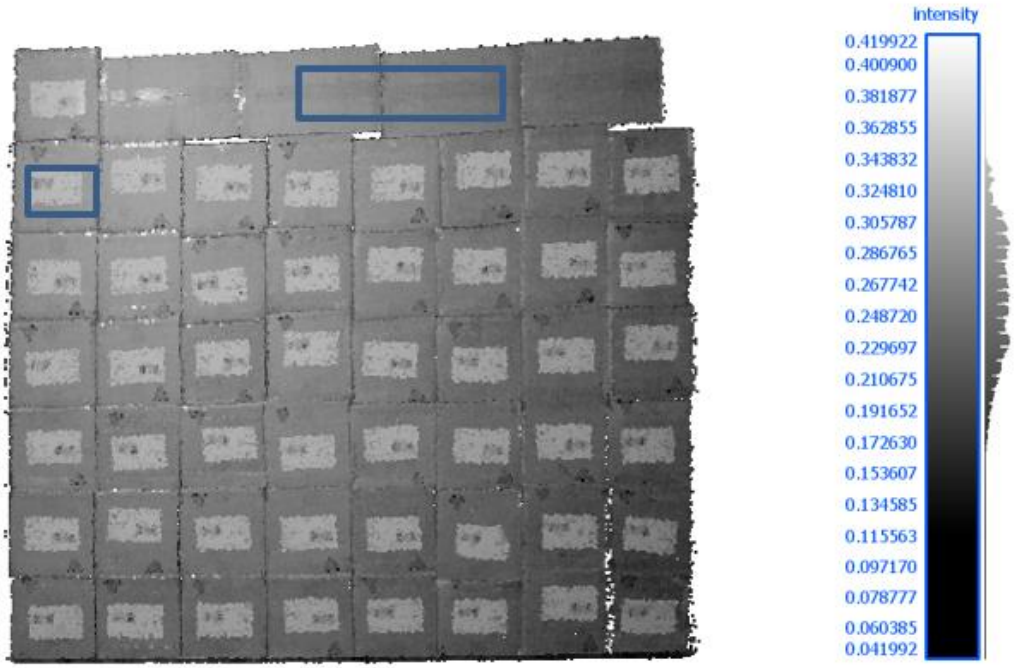



Figure 1-2 3d raw point cloud data representing the contents of an open cargo container, depicting multiple objects (colored based on laser intensity); labels and tapes on the boxes are visible (blue)

After segmenting the point cloud data, each detected instance's pose details are estimated. Errors in extracting pose information can damage the device by mispositioning the grasping arm of the robot (Vaskevicius et al., 2017). Furthermore, the objects of interest may not always be aligned in a straight line, making the pose estimation process even more difficult. Thus, the problem of pose estimation demands that the position of the object be known with its six degrees of freedom (6 DoF). They may also vary in their alignment (arrangement) and this requires us to know their dimensions. Moreover, these objects are not rich in texture or geometric features, making it more complex to employ feature-based methods (D. Liu et al., 2018). The accuracy of pose details relies on the accurate segmentation results. Hence, a combination of instance segmentation of objects of interest and their subsequent pose estimation with dimensions would better detect the target objects.

## 1.2. Research Identification

A series of steps are involved in the cargo unloading process; each step removes a single box or several of them at once. The steps are visualized by a scan which are one-shot representations of the scene elements. The robotic system decides which of the scene objects should be unloaded first for optimal results every time a scan is processed. All the objects visible should be detected with their poses and geometry at each scan to make a decision. The object geometry here refers to the dimensions of the object and the pose details are its 6 DoF.

The point cloud data is first segmented to simplify the task by using a segment growing technique which groups similarly characterized points. However, such direct methods cause problems when separating objects with thin boundaries, especially when density varies. One technique to identify objects from lidar data is to project the 3d point clouds into range images and then analyze using image analysis techniques (Ye, Wang, Yang, Ren, & Pollefeys, 2011). Without RGB data, the point cloud can be considered as an image with only a depth channel. These images can be segmented by exploiting the discontinuities in depth and surface normal orientation (Baccar, Gee, Gonzalez, & Abidi, 1996). The surface normal vector of laser points helps in differentiating each box type object, where there is a possible change in surface orientation detected. The surface orientation with respect to the beam direction results in different laser footprints and intensity values. A direct point cloud segmentation method is employed to understand the effects of varying footprints of the laser points on the segmentation process. The method is kept as a baseline and the study tries to find an alternative to segmenting the point cloud data by employing a projection-based point cloud method. The study then attempts to alleviate the problem of varying point density by increasing the footprint of the laser point during its projection onto the image. A comparison is also drawn on the lines of which approach is suitable for segmenting individual box objects stacked in a cargo container.

The main focus in automation processes is to reduce equipment costs and the computational effort involved in processing data from multiple scanning positions. The single-shot scans are thus explored in this study to understand the maximal accuracy such datasets can provide for the intended application. Furthermore, the study attempts to utilize the single-shot representations with a scanner fixed at a point to see whether the available information level is sufficient for segmenting the objects and finding their orientation. In such a case, the need for an effective segmentation technique to precisely distinguish the objects of interest from single view captures amidst the various corruptions that could be present in the dataset to extract pose details is necessary.

### 1.2.1. Research Objective

The research aims to develop an algorithm that will perform segmentation on the point cloud data to identify the number of objects ***'N'*** present. It aims to achieve maximal accuracy for the segmentation on one-shot scans of the scene. It will utilize only the properties available from the point cloud data. The segmentation is followed by estimating the object dimensions and the 6 DoF associated with each of them. The method aims to focus on using the most suitable point cloud property from the data to achieve better instance segmentation results for similarly characterized objects and estimate their pose. The sub-objectives designed to address the primary objective are -

1) To find the number of objects ***'N'*** with their dimensions - that are visible and therefore to be unloaded next.
2) To assess the accuracy of the method.
3) To extract pose estimates of the objects with six degrees of freedom.

### 1.2.2. Research Questions

1) Which point-cloud attribute(s) is the most suitable to differentiate the foreground objects from the background?
2) What method can be used to distinguish every object?
3) How can the problem of varying point density be addressed?
4) What are the total number of objects and their dimensions?
5) How are the objects of interest oriented in 3d space?
6) Does the segmentation work well to aid in recognizing the individual instances?

## 1.3. Thesis Structure

This document consists of seven chapters. Chapter 1 discusses the background and the motivation for carrying out this research. Chapter 2 briefly reviews the theoretical concepts and principles from the literature that are relevant to this study. Chapter 3 is about the datasets available for this study and the software involved in the different processes of the research. Chapter 4 elaborates the methodology and the various steps taken to achieve the objectives. Chapter 5 presents the analysis of results followed by Chapter 6 and Chapter 7, discussing the critical findings and suggesting recommendations for future work.

# 2. LITERATURE REVIEW

This section reviews the principles of lidar and the segmenting algorithms used in this research from the literature. The focus is on those studies that contribute to object detection in an industrial robotics setting. It explores the techniques for point cloud and image segmentation. Further, it also discusses the segmentation evaluation concepts for determining the accuracy of the segmentation results.

## 2.1. 3D Point Clouds

Point clouds are a three-dimensional representation of a scene in space. They can be acquired using lidar systems. A typical lidar system is equipped with a scanner mechanism and can be mounted on different platforms based on which they can be airborne, terrestrial, or mobile (Fernandez-Diaz et al., 2014). In this study, a pulsed lidar is used. It works by measuring the distance from the sensor to the objects of interest by emitting laser pulses and calculating the time taken for the pulse to reach back (Chazette, Totems, Hespel, & Bailly, 2016). A lidar captured scene is represented in the form of points, each having 3d information about its location along with laser intensity values. A large volume of such points makes a point cloud. Due to accurate and cost-effective data collection methods in the past years, it has been employed for various applications within the industrial automation domain (Jakovljevic, Puzovic, & Pajic, 2015). The quality of the point cloud data collected directly affects its processing. It is affected by the properties of the objects scanned, environment conditions, the hardware system and the scan geometry (Soudarissanane, Lindenbergh, Menenti, & Teunissen, 2011). For each point, its range, horizontal and vertical angles, the transmitted beam makes with the hit-surface are recorded, which depends on the relative position of the sensor system to the scene (Křemen et al., 2006). A laser beam hitting the target surface leaves a circular footprint while the surface at larger distances has an elongated footprint – **Figure 2-1**. The intensity of backscatter depends on the laser footprint and affects the processing of point cloud data.

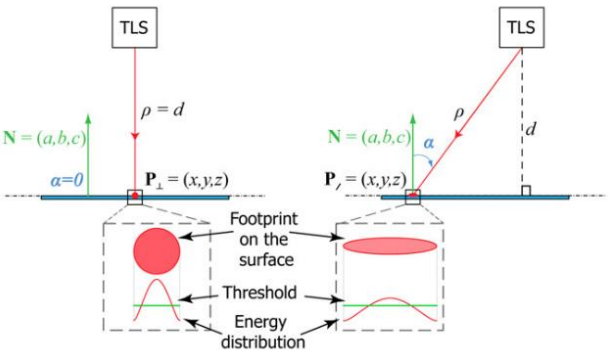



Figure 2-1 Reflection geometry. An illustration of the resulting footprint for a perpendicular laser beam (left); laser beam with some incidence angle (right) Source: (Soudarissanane et al., 2011)

Although advancements in scanner technology have led to accurate data collection, processing such data is still crucial for employing it in different application domains (Bia & Wang, 2010). The lidar points collected are distributed over the entire measurement area and for a container unloading scenario, the task then becomes segmenting this data into meaningful components. Segmentation remains one of the critical elements in point cloud data processing (Jakovljevic et al., 2015).

## 2.2. Object Detection in Industrial Robotics

Detecting objects in an industrial setting requires the robotic system to know the objects' pose and orientation details with respect to a reference frame or a sensor system. Therefore, understanding the observed environment and determining the number, attributes and pose of the objects within the

environment is one of the most challenging issues and aims that the machine vision community addresses. Carton-box detection is one of the most occurring scenarios in logistics automation, as most goods come packed in boxes of different sizes (Echelmeyer et al., 2011). However, there is no large-scale public dataset available to train and evaluate carton box models; Jinrong Yang et al. (2021) used open-sourced images to build a dataset but does not involve 3d detection. Although deep learning methods provide better results and are starting to replace the classical methods, they need substantially large training data and require high computational resources.

RGB-D (RGB color image and range information) usage for object recognition has shifted the focus on the classical 2d approach to analyzing the data with an additional parameter – range (Czajewski & Kolomyjec, 2017). The segmentation task can utilize the range information to get an accurate 3d pose and extract geometrical features of the segmented objects. The method proposed in Kuo et al. (2014) works by generating feature points and simulating the images for pose detection using a template-based matching algorithm. However, for the matching process, the models require distinctive feature points and in the case of recognizing a carton box, the lack of significant descriptors makes such methods less useful. Although a single range image provides valuable information, objects viewed from various viewpoints provide information across the views and combining this makes the object detection task more robust (Djelouah et al., 2015). Nevertheless, vision-based systems for industrial applications have challenges, such as varying illumination levels and occluded objects (Kim et al., 2012).

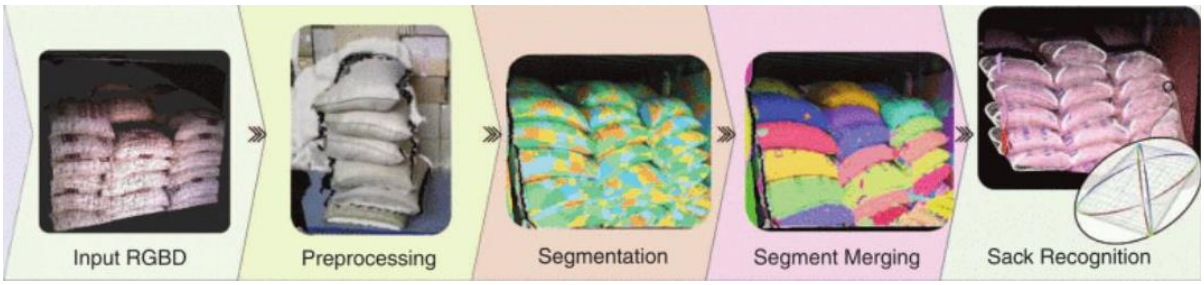



Figure 2-2 Instance segmentation of coffee sacks for object detection and retrieval by a robotic system using RGBD data. Source: (Stoyanov et al., 2016)

Three-dimensional point cloud data have advantages over the RGB-D images; they contain information about the volume, surface, location of the objects and enable the extraction of pose with 6 DoF. Nevertheless, the 3d point cloud data is highly unorganized and has varying point density. Studies have explored the data fusion approaches for object tracking from multiple single-shot representations (Dieterle et al., 2017). Dieterle et al. (2017) uses a combination of laser data and stereo images to make a data association and tracks the objects through multiple views. However, the usage of more datasets contributes to high computational costs. Thus, this study tries to find a method that exploits the single-shot lidar scans for the object detection task.

## 2.3. Segmentation

Segmentation is the process of dividing the scene objects into meaningful and recognizable elements. From a given set of points, the process of segmentation will group similarly characterized points into one homogenous group. The fundamental step in point cloud processing is to separate the background and the foreground points. The result of segmentation further helps in locating the position of the objects. Although the shape, size and other information of the objects can be determined, the 3d points clouds are

noisy, sparse and lack uniform structure. The non-uniform point density is caused due to the scanning geometry.

The segmentation of 3d data follows edge-based, region-growing or hybrid approaches. The 3d information in the captured laser points helps in distinguishing the different objects in the scene. However, if the objects of interest are more or less on the same plane, one dimension is effectively lost. The local surface attributes, such as surface normal, gradients and curvatures, define the weakly present edge geometry when the changes in surface properties exceed a given threshold (Rabbani, van den Heuvel, & Vosselman, 2006). The local surface attributes can be defined per point. Integrating the point feature values across the segments result in better segmentation results. The accurate calculation of the normal vector at each point is an essential step in 3d point cloud processing, also crucial to segmentation. Regression-based estimation for normal vector computation works by fitting a plane to k-nearest neighbors of a point using principal component analysis (PCA) (Jolliffe & Cadima, 2021). A method that is robust to outliers and works well on data with varying local density is required.

The major distinction between lidar data and 2d image data is that the 3d points are a highly unordered discrete set of points scattered in space. On the other hand, 2d images have high-density pixels while the point cloud has some areas with sparse points and for this reason, the 2d object recognition methods cannot be used straightforwardly on the 3d data (H. Li et al., 2019). Object detection in 3d point clouds can be divided into raw point cloud-based methods, projection-based methods and volumetric methods (Arnold et al., 2019). This study keeps a direct point cloud segmentation as the baseline method and it explores a projection-based point cloud segmentation method to see if it is better suited for the application at hand.

### 2.3.1. Point Cloud Segmentation

The task of direct point cloud segmentation is challenging due to its uneven density, high redundancy and unorderly structure (Nguyen & Le, 2013). The discontinuities in the surface represented by the points are the basis for edge-based segmenting techniques. On the other hand, the region-growing methods work by detecting continuous surfaces with homogenous or similar properties. Two-step approaches can obtain good segmentation results - a coarse segmentation, followed by a refinement step (Besl & Jain, 1988). The studies in the past have adopted similar approaches to the raw point cloud segmentation process. In (Tóvári & Pfeifer, 2005), the normal vectors, the distance from the point to the nearest plane and the distance between the current and candidate points are used to merge a point to the seed region. Ning et al. (2009) proposed a rough segmentation to group all points belonging on the same plane, followed by a more refined segmentation to get more detailed segmentation results with distance from a point to the local shape being the criterion. The area of the plane generated can be used as the criteria for seed region selection and then a suitable searching algorithm can be implemented to add the neighboring points to the seed region (Deschaud & Goulette, 2010). A single-point cloud segmenting technique will not provide a satisfactory result (George Vosselman, 2010). For a better outcome, a combination of methods is recommended.

### 2.3.2. Range Image Segmentation

A computer vision system must determine depth from the images that enter the system to recognize objects in three-dimensional space. A 3d point cloud can be mapped to a 2d grid by projection-based methods and this grid can be processed further (G. Yang et al., 2020). These methods reduce the dimension of the point cloud and, subsequently, the computational cost of its processing. Although the loss of information is inevitable while using these methods, image-based object detection methods are well researched in computer vision. A range image is one in which the grayscale values directly relate to the

depth information (Hoffman & Jain, 1978). Range images can be effectively segmented by utilizing the two main discontinuities that occur in them. The first one is step edges that indicate breaks in depth and roof edges that show discontinuities in the surface normal orientation (Baccar et al., 1996).

The watershed transform is based on the intuition that the local minima in the gradient image are considered catchment basins. When flooded from those points, watershed lines are built where water from two basins meets (Beucher, 1979); **Figure 2-3**. However, the application of watershed segmentation tends to over-segment the image because several local minima are identified (Meyer & Beucher, 1990). The drawback is overcome by starting the watershed from selected points, called the markers (Juntao Yang, Kang, Cheng, Yang, & Akwensi, 2020). In Yang et al. (2020) the tree crowns are selected as local maxima within a given window size and then inverted to consider the high points as minima to perform segmentation. There is no significant height variation in a planar surface to identify a local maximum within the object instance. In such a case, the Euclidean Distance Transform is applied to find the distance from each foreground pixel to the nearest background pixel (Ibrahim, Nagy, & Benedek, 2019). To further enhance the segmentation, one can integrate the repetitive, regular patterns found in the region of interest. The method in (Shen, Huang, Fu, & Hu, 2011) assumes the segments on the planar surface are aligned globally along either vertical or horizontal direction and it partitions the urban facades having repeating rectangular structures. The approach can also have the added advantage of solving the lack of uniform resolution over the entire area. The region scanned can have variable resolution depending on the scan geometry of the system; the objects at the center of the scan have better resolution than objects at the edge of the scan (Sithole, 2008). As the objects are stacked in rows, one above the other, they can be assumed to be aligned in the horizontal direction. This additional information of repeating linear patterns can refine the segmentation, where weak boundaries between two objects exist. Thus, a good approximation for the number of identified objects in the scene can be found.

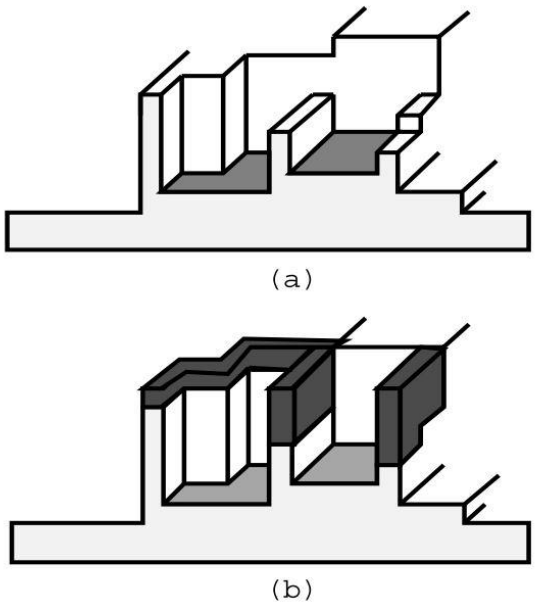



Figure 2-3 Figure 4 11 (a) A gradient image showing two regional minima (in dark); (b) Dams built to prevent the water from merging between the two adjacent catchment basins. Source - (Baccar et al., 1996)

## 2.4. Segmentation Quality Evaluation

The accuracy of obtained segmentation results is measured by comparing them to the ground truth data. The discrepancy between the results of segmentation and the ground truth reveals the quality of segmentation. When this discrepancy is small, the quality of segmentation is high. The discrepancies are of two types: geometric and arithmetic (Y. Liu et al., 2012). Geometric discrepancy occurs when the segmentation results are evaluated by comparing the boundaries of the predictions with the reference data. On the other hand, arithmetic discrepancies are based on the over and under segmentation results that the method may produce. It is evaluated directly by comparing with the total number of identified objects. Several methods could be employed for evaluation purposes, and visual inspection is considered one among them (B. Johnson & Xie, 2011). By visually comparing the segments, the user can determine the

qualitative accuracy. However, visual inspection of results does not provide quantitative evaluation and is subjective (Xueliang Zhang, Feng, Xiao, He, & Zhu, 2015).

For this study, a method that assesses both types of discrepancies is adopted. The metrics such as precision, recall and F1-score are some criteria for quantifying the efficacy of an algorithm (W. Li, Guo, Jakubowski, & Kelly, 2012). The percentage of correctly segmented components produced by the algorithm is called precision and the percentage of correctly obtained ground truth reference components is called the recall. While precision is more sensitive to the presence of incorrect elements, the latter is sensitive to the presence of undetected reference data. The F1-score is computed using these two percentages and potentially reveals the overall quality by considering the trade-offs of the two measures. The tri-partite measurements of precision, recall and F1-score help evaluate classification results. However, implementing it for the instance segmentation results is not direct. The Jaccard index (Deza & Deza, 2009, p.299) is used for this purpose. It is the IoU (intersection over union) score, which is based on region overlapping. The bounding boxes of the predicted results and the ones from ground truth are the two inputs based on which the IoU gives a score. The accuracy of the segmentation is evaluated quantitatively by setting a threshold on this score.

When there is no reference data available to compare against, they can be obtained manually (Douillard et al., 2011). Several open-sourced tools are available for creating ground truth labels for the objects present in the scene (Saleh et al., 2018; Nieto et al., 2021). For this study, LabelImg, an image annotation tool, is used to extract bounding boxes of the objects and is used as the reference data (Xiao et al., 2019).

# 3. DATA AND SOFTWARE

The properties of data used in this research are detailed in **Table 3-1** and a sample point cloud is shown in **Figure 3-1**. As for the evaluation of results, visual inspection methods have been employed to evaluate all the datasets. A quantitative evaluation is made on one dataset. The different datasets and their properties are discussed below.

## 3.1. Lidar data

The datasets used for this study are 3d point clouds that capture the contents of an open cargo container using a Sick LMS4000 lidar system mounted on a semi-automated robot. The absolute sensor position and the scanning angle of the laser system are not known. The sensor position is approximately 2.5 meters away from the top left corner of the cargo container. The point clouds are all one-shot captures. The lidar sees the container from the open end, so the points captured belong to the cargo container and the objects of interest. For this study, the acquired point clouds are converted to a standardized format by removing the points belonging to the sidewalls of the cargo container. Some point cloud datasets are scanned from different angles for the same cargo container. Some others are scanned after a row of objects have been removed, revealing the box objects that may be present behind the unloaded row of objects. This variation and combination of datasets allow for analyzing if the results vary under different circumstances for the same scene. The varying point density and scalar range from the sensor to the objects present in the scene are described below.

Table 3-1 Dataset description

| Data Type | Number of datasets | Data Format | Dataset Size | Total number of points in the dataset | Point spacing (in m) | Scalar range (in m) |
|---|---|---|---|---|---|---|
| Point Cloud Data | 10 | .ply | 2.09 GB | 774,992 – 1,133,298 points | 0.0038 – 0.024 | 2.37 – 3.59 |

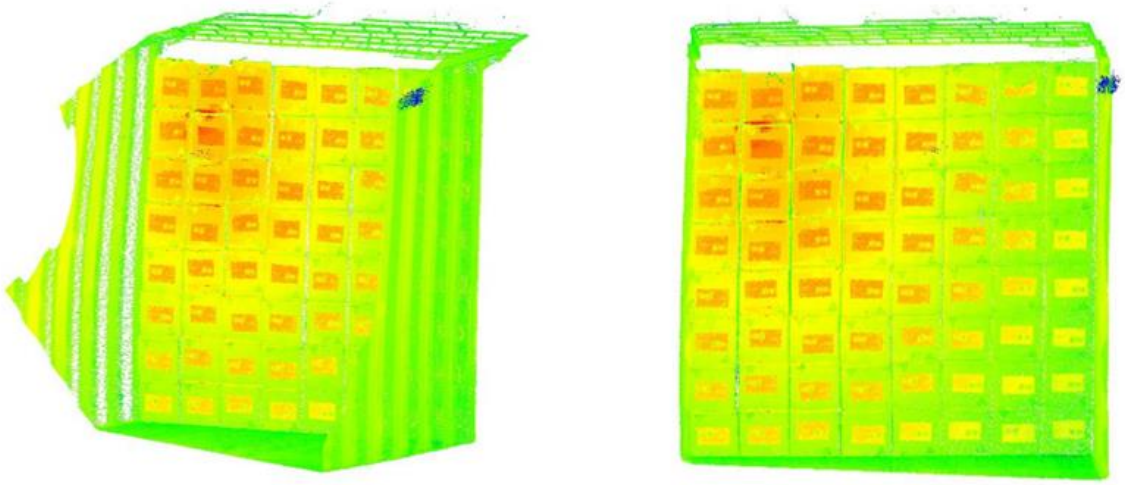


Figure 3-1 3d raw point cloud of an open cargo container with labels on the carton boxes visible (colored based on laser intensity ) – side view and front view (left to right)

The varying scalar range (in meters) from the sensor to the scanned objects from the scene is visualized in **Figure 3-2**.

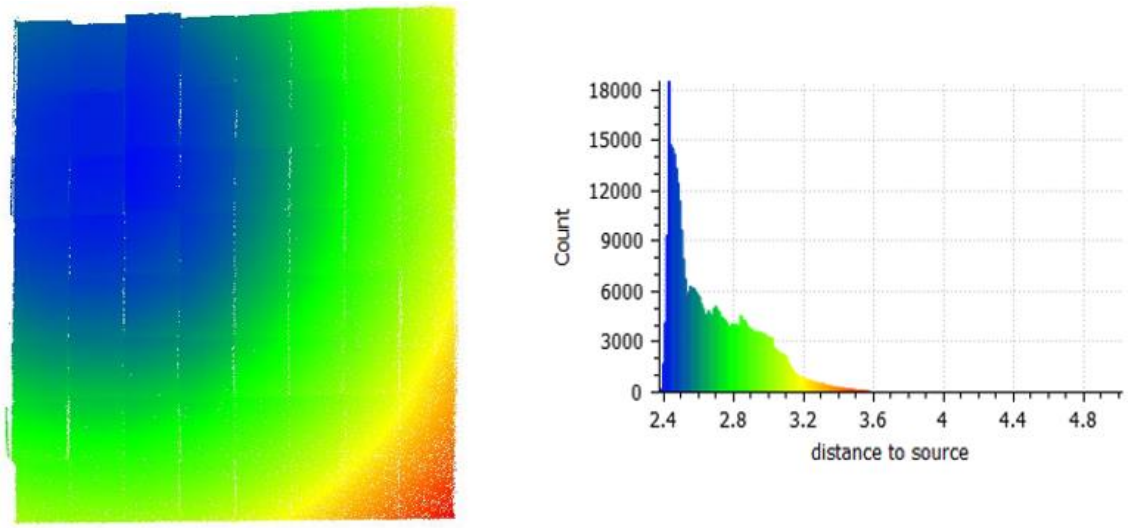



Figure 3-2 3d point cloud of an open cargo container in standard format (colored based on the scalar distance from the scanner) and its corresponding histogram (left to right)

### 3.2. Ground truth

When ground truth data is not available for evaluating the results obtained, one way is to generate them manually. The annotations are manually created by using an open-sourced toolbox from python – LabelImg[1]. It provides an interface to read image data and assign labels to each of the objects. The PASCAL Visual Object Classes (VOC) format is used for these annotated datasets. It outputs an XML file for each annotated image with bounding box information for all box objects present. The bounding box information is extracted as shown below –

$$\text{Bounding box information} = [x\_minimum, y\_minimum, x\_maximum, y\_maximum]$$

The generated bounding box (ground truth) is used to evaluate against the bounding box that the algorithm produces.

### 3.3. Software

The point cloud data is segmented directly using a segment growing technique. The results are visualized using the PCM (Point Cloud Mapper) program. The range image formed by projecting the point cloud data and the subsequent image segmentation is implemented using Python (3.7) programming language. PyntCloud and OpenCV are some python libraries used. The point cloud pre-processing for both methods is handled by Cloud Compare software. The chosen IDE (Integrated Development Environment) is Jupyter Notebook. The processes are all run on a Windows 64-bit machine with Intel Core i7-9750H CPU at 2.60 GHz with 16GB RAM.

[1] https://github.com/tzutalin/labelImg

# 4. METHODOLOGY

This chapter outlines the main steps taken to answer each of the research questions presented in chapter 1. **Figure 4-1** outlines the workflow to identify the objects distinctly and subsequently estimate the pose with six degrees of freedom for each of the identified objects along with their dimensions.

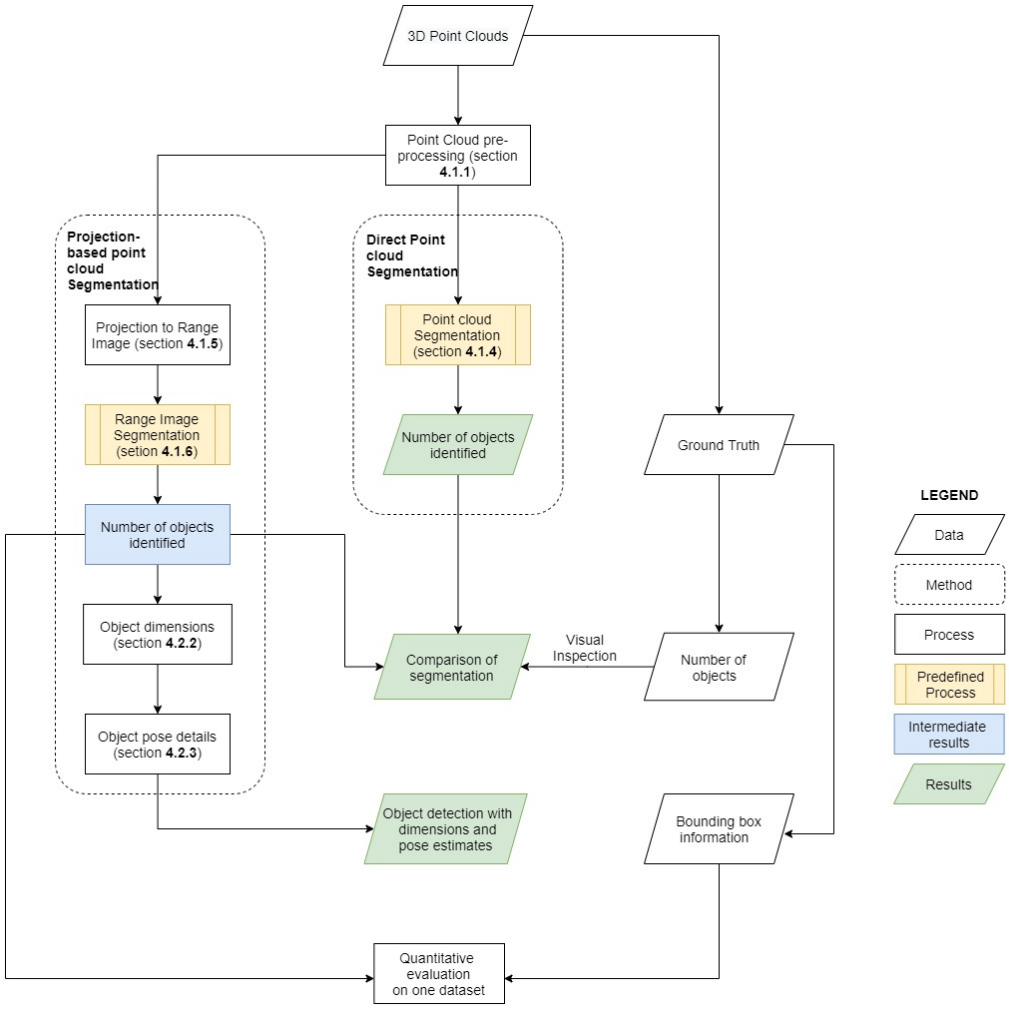



Figure 4-1 Overall workflow - overview of the steps involved in the methodology; section number included within parenthesis

## 4.1. Object Segmentation

The segmentation approach depends on the nature of the application, as discussed earlier. In the specific case of cargo-box unloading in industrial robotics, the entire scene is made of the same object. The possible difference is in the orientation and dimensions of such objects. This section describes the process involved in segmenting the 3d point cloud data and the range image obtained by projecting the 3d points. Initially, the point cloud dataset is analyzed for its properties to understand the dataset's quality and how it can be utilized further for the aimed application.

### 4.1.1. Data Pre-Processing: Downsampling the point density

The point cloud datasets available are one-shot representations of an open cargo container. **Figure 4-2** depicts one such dataset, illustrating the varying point density. The point cloud is sub-sampled using Cloud

Compare software to achieve close to uniform point density; **Figure 4-3**. The aim here is to visualize the borders to each box object. **Figure 4-4** visualizes the surface density in the downsampled point cloud using a radius of two centimeters with its histogram. The value used for thinning the point cloud is identified by visualizing the "variation in surface density". The goal is to see the surface variation between the points that belong to the box surfaces and the boundaries between them.

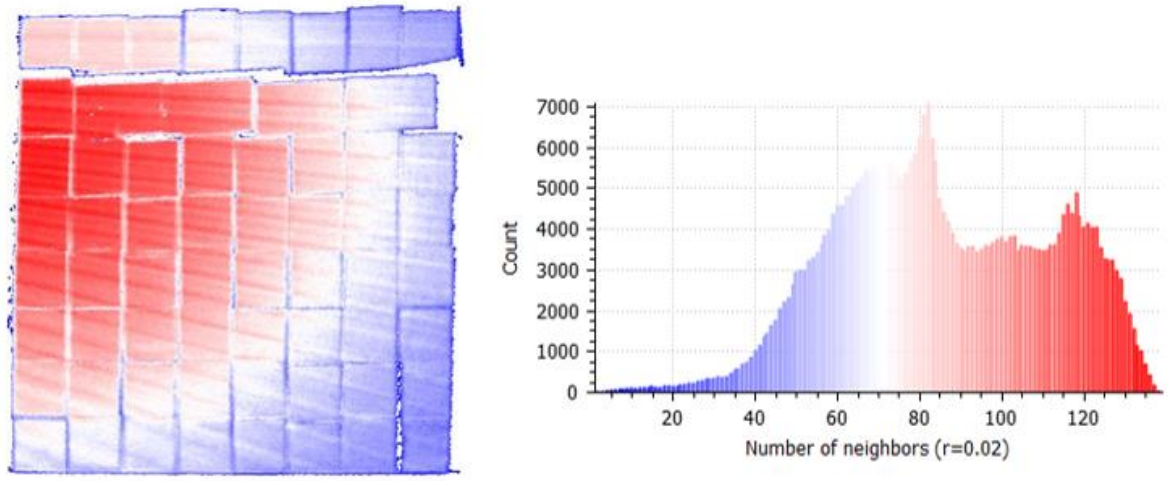



Figure 4-2 (a) Figure representing the varying point density in the point cloud data; (b) Histogram of the available point density

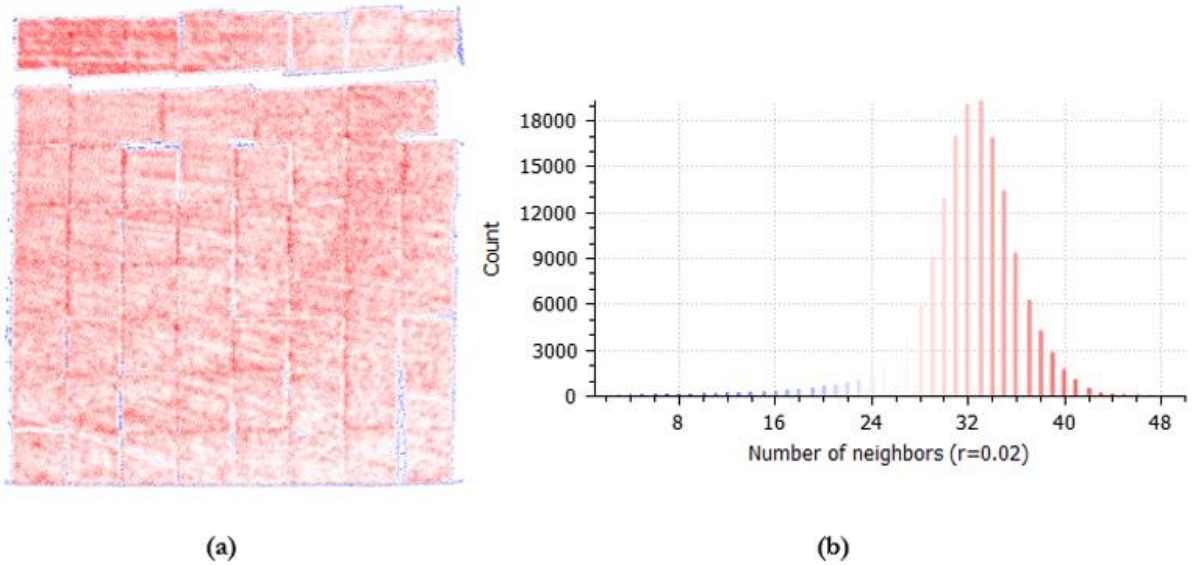



Figure 4-3 (a) Figure representing the point density after downsampling; (b) Histogram of the point density after downsampling

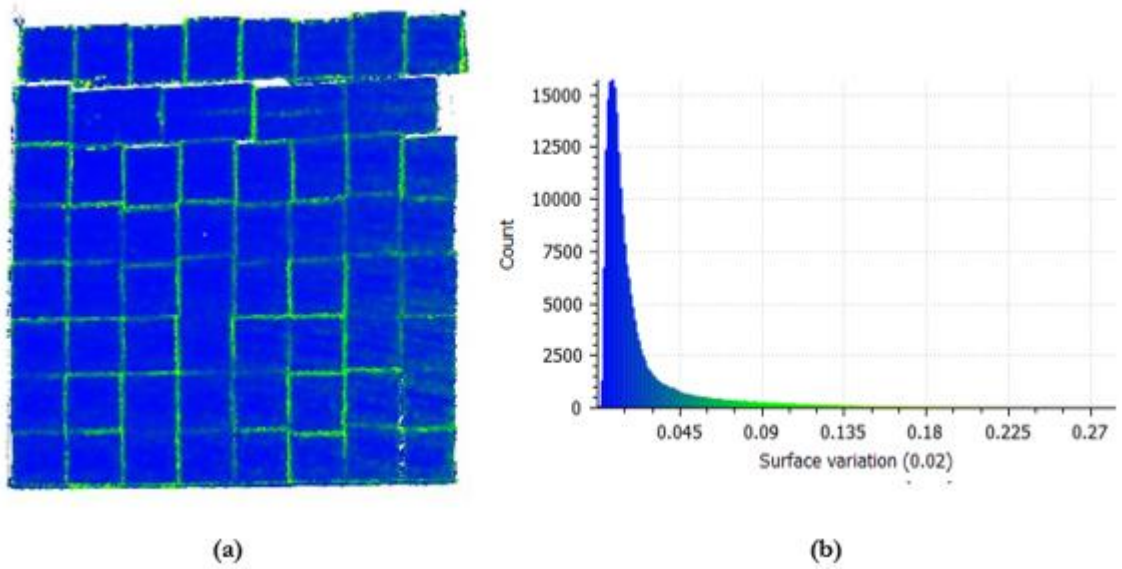



Figure 4-4 (a) Surface density on object surfaces and the boundaries after downsampling; b) Histogram of the surface density values after downsampling

### 4.1.2. Laser Intensity

The different attributes of the point cloud data are analyzed to answer research question 1). Laser point intensity is the measure of the laser pulse returned from the object surface and it depends on the properties of the object material (Song, Han, Yu, & Kim, 2002). The intensity information is a more specific feature than the normal vector, which considers a defined neighborhood around the point. Although it is valuable information for distinguishing the objects, the objects stacked in the cargo container have labels on the surface and tapes along their faces. The reflection from the labeled portion is high, visualized in **Figure 4-5**. The laser backscatter from the box's surface is similar to the backscatter from the labeled portion as the range from the scanning system increases. In the top-left region, the laser intensity nicely distinguishes the box surface and the box edges. In such an ideal case, a threshold value can separate the two. However, the positioning of the scanner system and the varying range causes the differences in the backscatter received. Hence, this property is not a good differentiator to separate the background and edge points in this case.

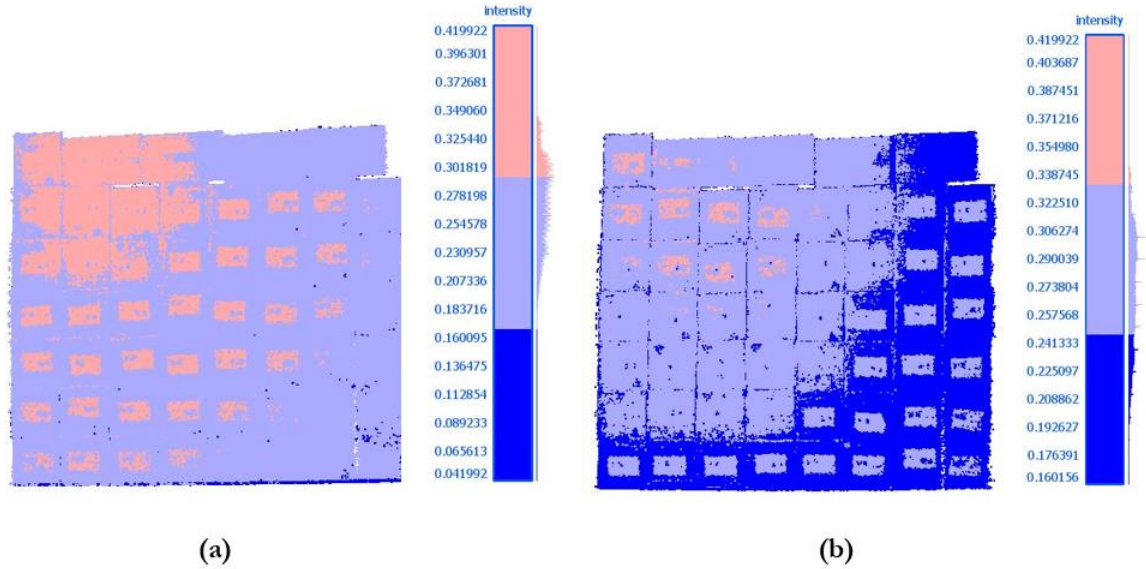



Figure 4-5 Point cloud colored based on laser point intensity – (a) available laser point intensity (labeled portion and some top-left boxes shown in light pink); (b) point cloud filtered with laser point intensity values above a threshold of 0.16 (labeled portion and some box surfaces shown in light blue, labeled portion of some top-left boxes shown in light pink)

### 4.1.3. Point-wise Surface Normal

The way an object's surface is oriented with respect to a defined plane is a valuable feature that helps in recognizing it uniquely. The unit normal vector ***'n-p'*** to a tangent plane at a point ***'p'*** represents the orientation of an object's surface with the defined plane. This unit normal vector, ***'n-p'*** can be determined by fitting a plane to a neighborhood of points; **Figure 4-6**. This neighborhood is user-defined and directly affects the calculation of the normal vectors. The normal vectors are not discrete when passing over an edge that is smooth or curved, as in this case. This slight variation in the computed surface normal vectors will help distinguish the points that are on the surface of the objects and the points that belong on the edges. Large neighborhoods provide stable normal vectors to points that belong on the object's surface, but the same is not true for the points on the edges. The optimal selection of neighborhood for the normal vector computation is thus essential.

For this study, the surface normal at each point is calculated using a plane local surface model that is robust to noise and performs better with edges that are not sharp. A neighborhood radius of two centimeters with the octree structure is selected. The orientation of the computed normal is kept parallel to one of the three main X, Y and Z axes in the positive direction. The alignment with each of the axes results in its respective surface normal orientations. The change in the normal vectors computed along the z-orientation with varying neighborhood radius is illustrated in **Figure 4-7**.

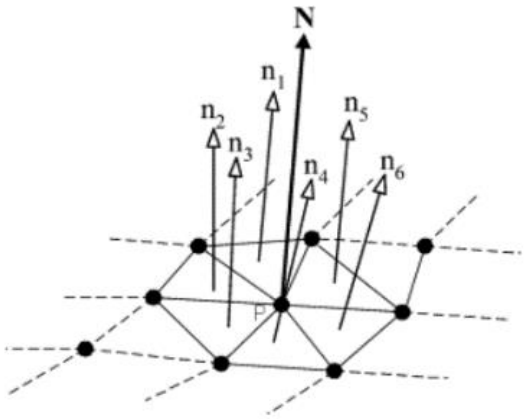



Figure 4-6 Computing normal (N) at a point P.
Source: (Woo, Kang, Wang, & Lee, 2002)

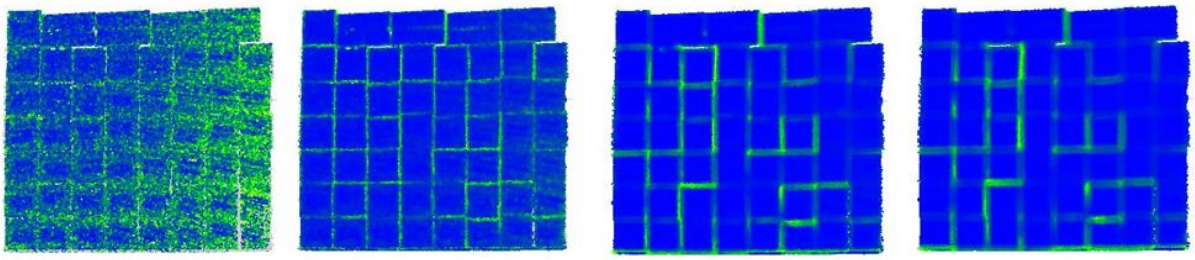



Figure 4-7 Point cloud colored based on the changes in normal vector on the object surfaces and boundaries, calculated with different neighborhood radius-(a)1 cm; (b)2 cm; (c)4 cm; and (d)5 cm

The normal vectors computed per point attempt to separate the foreground and boundary points upon analyzing the differences based on different point cloud attributes. The normal along the 'x' and 'y' axes show strong linear differences in the vertical and horizontal direction, respectively. **Figure 4-8** below shows the laser intensity, normal vector in x, y and z-directions used to color the point cloud to distinguish the background and foreground points. The local normal along the z-axis is the most suitable attribute for determining the boundaries around each box object from the entire box pile.

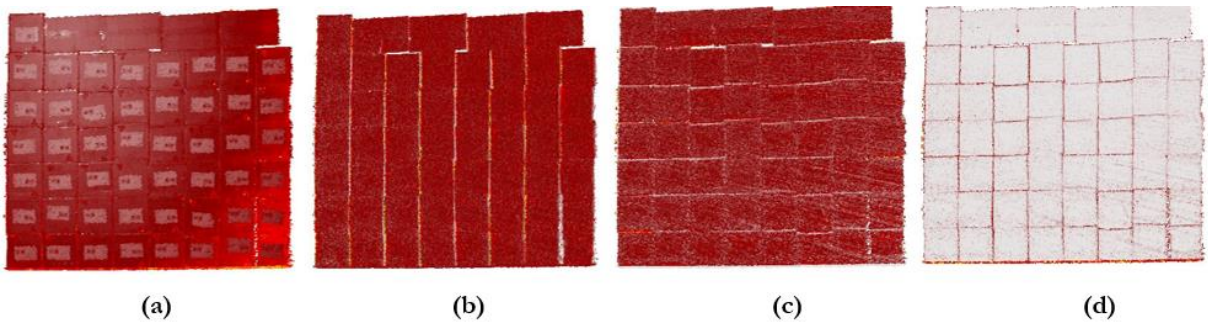



Figure 4-8 Point cloud of open cargo container colored based on (a) laser point intensity; (b) normal vector along x-direction; (c) y-direction; (d) z-direction

### 4.1.4. Point Cloud Segmentation

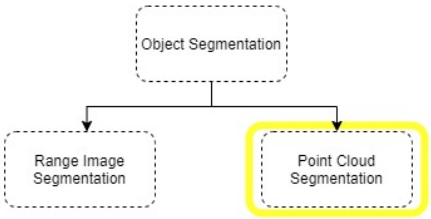



Figure 4-9 First step in the methodology pipeline - this sub-section deals with the highlighted box (in yellow)

The point cloud segmentation is carried out using a combination of segmenting techniques. The segment growing technique identifies the different objects based on the similarity of the point features within a given neighborhood from the seed points. The point feature z-normal vector is the chosen attribute to carry out the segmentation. Due to the high density of the point cloud data, the x, y and z- values of each lidar point differ at the millimeter level. Thus, the points are all scaled by a factor of ten. **Figure 4-10** shows the steps involved.

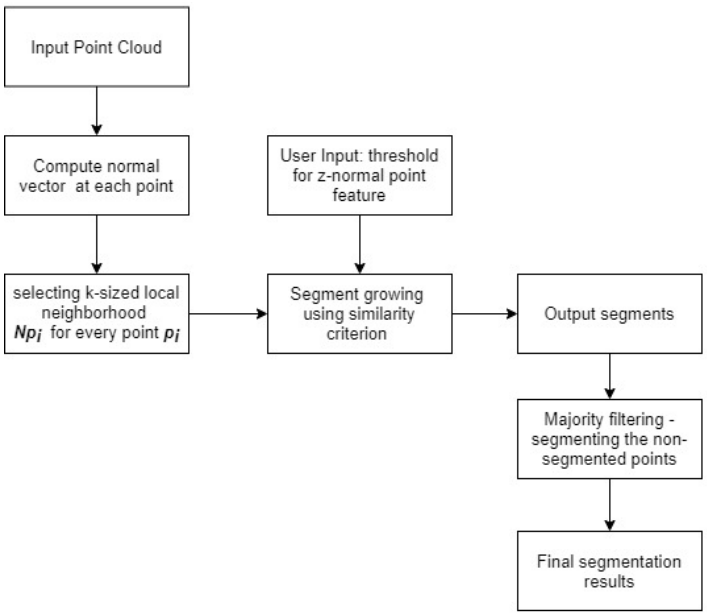



Figure 4-10 Flowchart outlining the process involved in direct point cloud segmentation using segment growing

The segmentation process starts by identifying the neighborhood of each point in the point cloud. The k-nearest neighbors (k-NN) algorithm is used to span the search for neighboring points. Growing seeds are defined as groups of nearby points that have comparable point feature values. The seed regions are initialized with a neighborhood of 30 points in this study; **Figure 4-12 (b)**. The condition used here to group points together is the similarity of z-normal vector feature values. The z-normal vector values are lower over the edges of the box objects while they are higher on the surface of the boxes. When the feature values are close to the initialized segment's average feature value, the candidate points are merged with the seed points, extending the segment. By experimenting, a threshold value of 0.90 for the z-normal vector is selected to distinguish a point from belonging to the object surface or its edge; **Figure 4-11 (d)**. The effects of the threshold value set on the z-normal vector are shown below.

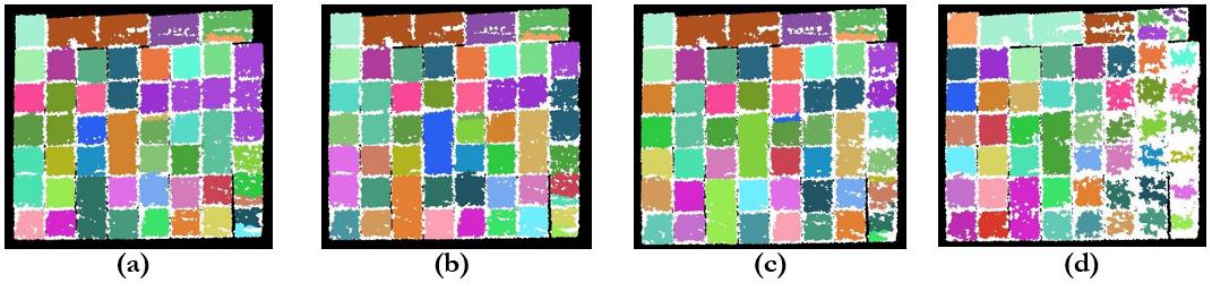



Figure 4-11 The results of segment growing with varying threshold values set on the z-normal vector feature with neighborhood size of 30 points – (a) 0.75; (b) 0.80; (c) 0.85 and (d) 0.90

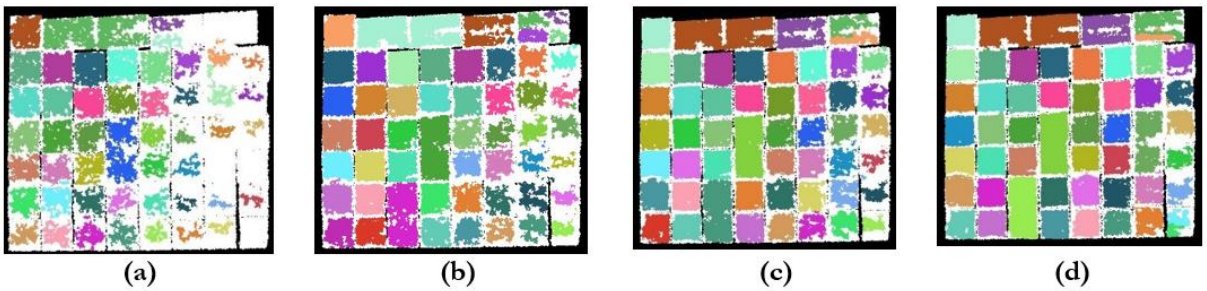



Figure 4-12 The results of segment growing with varying neighborhood size with threshold on z-nomal vector set at 0.90 – (a) 20 points; (b) 30 points; (c) 35 points and (d) 40 points

By experimenting, a value of 20 points for the neighborhood selection has a decreased change in the number of segments. As the neighborhood increases from 30, there is also a reduced effect on the number of segments. An increased neighborhood results in fewer isolated points that do not belong to any segment, but the results suffer from under-segmentation. Decreasing the neighborhood size leads to more points not being segmented at regions with low point density. The results of a combination of neighborhood values and the threshold values still contribute to many points not part of any of the segments.

Table 4-1 Tables with detected number of segments for the figures 4-11 and 4-12 (the chosen values highlighted in green)

| Threshold value on z-normal vector Figure 4-11 | Detected number of segments ( ground truth – 54) |
|---|---|
| 0.75 | 47 |
| 0.80 | 49 |
| 0.85 | 51 |
| 0.90 | 54 |

| Neighborhood size (in points) Figure 4-12 | Detected number of segments ( ground truth – 54) |
|---|---|
| 20 | 49 |
| 30 | 54 |
| 35 | 53 |
| 40 | 52 |

Different steps can follow up the segment growing results. The one selected for the case at hand is the majority filtering. To get smooth and more defined segments from the results of segment growing, the isolated points that do not belong to any segment are merged with the segment label that is most occurring within a defined neighborhood by the majority filtering technique. A neighborhood radius of one meter is chosen and a majority filtering is applied based on the segment labels obtained from the previous steps. The final results of point cloud segmentation using segment growing are presented in the results section.

#### 4.1.5. Range Image

The following function defines the range image as-

$$f = f(m, n) \qquad \text{Equation (1)}$$

Where,

- ***m*** denotes the row in the image,
- ***n*** represents the image columns and
- ***f (m, n)*** is a function of the laser point values (local normal vector computed along z-axis orientation)

When the ranging system employed is known, the vector ***(m, n, f (m, n))*** can be translated to a real-world spatial coordinate system. The points are projected onto the image plane ***(m,n)*** using virtual sensor coordinates to map the 3d points onto the range image. The sensor is assumed to be positioned on a plane passing through (0, 0, z-minimum) with normal vector n = (0, 0, 1) such that it is parallel to the X and Y axes.

For an orthogonal projection of the points onto the image plane, three points **P1, P2** and **P3** that belong to the stacked box pile on the point cloud are manually selected. The points are selected such that they belong to the three corners of a plane that these points could construct. The point cloud is then rotated around the normal to the plane generated by these three points, such that the normal of this plane is parallel to the global z-axis.

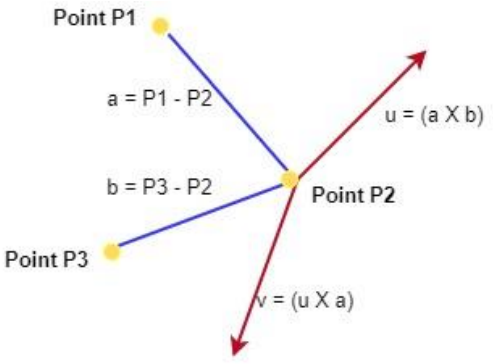



Figure 4-13 Manual selection of three points P1, P2 and P3 to compute the normal vector of the plane formed by the points

Where,

- ***P1, P2, P3*** are three points manually selected from the box pile; **Figure 4-13**
- ***u*** is perpendicular to ***a*** and ***b***
- ***v*** is perpendicular to ***u*** and ***a***
- ***(a,u,v)*** forms the new basis to which the points are to be transformed

$$A = R\,B \qquad \text{Equation (2)}$$

(or)

$$\mathrm{R}([\mathrm{a},\mathrm{u},\mathrm{v}]) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$Q' = R\,Q \qquad \text{Equation (3)}$$

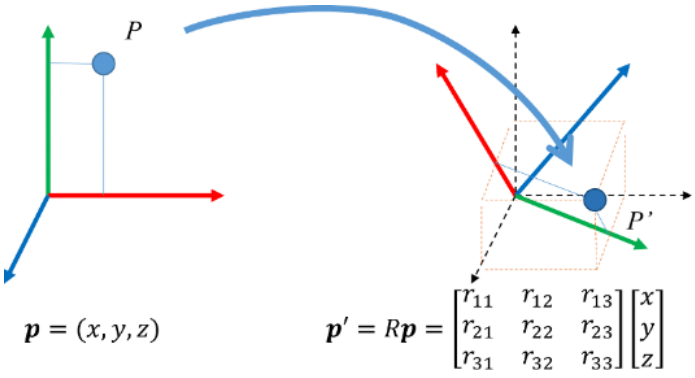



Figure 4-14 Figure[2] illustrating the transformation of point P through a rotation matrix R

Where,

- A - the old basis
- B - the new basis
- Q - point cloud
- R - transformation matrix (matrix R from **Figure 4-14**)
- Q' - transformed point cloud

As discussed in section 2.3.2, the 3d point cloud data can be projected and mapped onto 2d image planes. Next, the points are transformed to the new basis using the transformation matrix (3) generated in the previous step and then scaled to fit the dimensions of the range image. The image coordinate system defines the pixel positions ***(m,n)*** on the image. Optimal pixel size needs to be chosen to map the points to the image plane ***(m,n)***, as too small pixels can lead to loss of information and too large sizes could result in loss of pixel connectivity. The optimal pixel size is selected with the knowledge of the resolution of point cloud data (Hernández & Marcotegui, 2009).

To further minimize the effect of the varying point density, kernels of different sizes are used to increase the footprint of the laser point when projected on the image. A suitable footprint size is selected, which efficiently separates the two adjacent objects. The dimensions of the image increase with an increase in the footprint size as well. The resulting images of using footprints of different sizes are shown in **Figure 4-15**. Considering the trade-off between accuracy and time complexity, footprint size (k)=5 is used in this study. The point features to be used are computed before the projection onto the horizontal plane. This way, different channels can be introduced, and the number of features can be increased within the pixel. Later, these channels can be stacked together to be treated as an image of n-dimensionality. In this study, the z-normal vector values are mapped to the pixels and are then converted to grayscale tones to visualize it as an image.

[2] http://motion.cs.illinois.edu/RoboticSystems/CoordinateTransformations.html

As the point to pixel mapping does not have a one-to-one correspondence, multiple points may land on a single pixel. The normal vector contains information about the borderlines of each object; therefore the maximum z-normal vector value among the points that fall on the pixel is assigned to that pixel. This technique is similar to a depth-buffering algorithm, which works by mapping the values that are closer to the image plane to ensure correct surfaces occlude the other surfaces (G. S. Johnson, Lee, Burns, & Mark, 2005). In this research, the maximum value among the points landing on a pixel is retained for that pixel. Thus, the range image is now a 3d representation of the point cloud data where the pixel intensities are proportional to the z-normal vectors computed at each point.

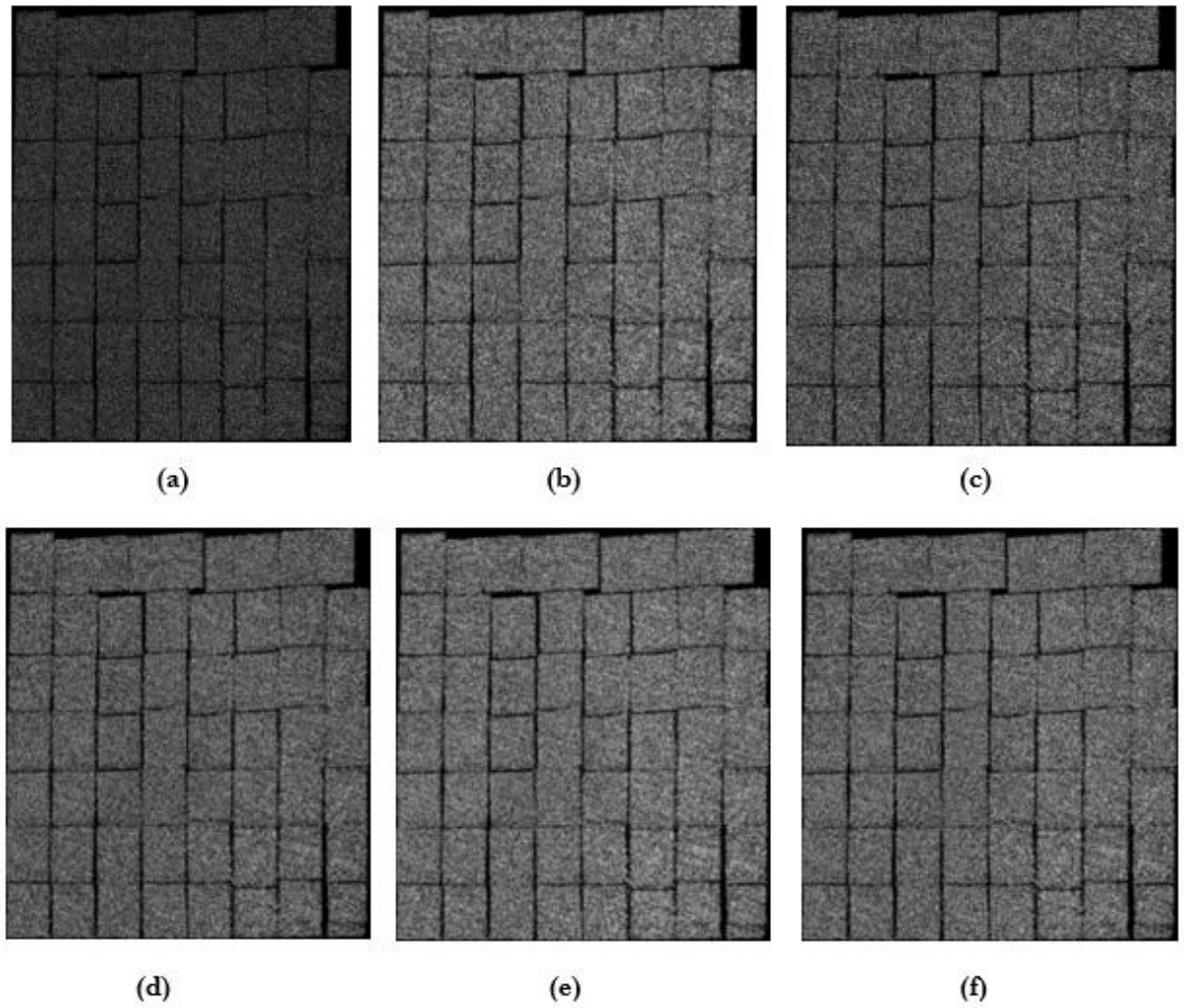



Figure 4-15 Range image projected from 3d point cloud with different footprint sizes and image resolutions described in table 4-1

The width of the cargo-loaded container (used in **Figure 4-15** and **Table 4-2**) of stacked cargo boxes measures 2.34 meters (measured approximately using Cloud compare). The pixel size for each of the above images is calculated with this width and displayed in the table below.

Table 4-2 Table showing corresponding pixel size for respective footprint size and image resolution used; the selected image resolution and footprint highlighted (green)

| Image (Figure 4-15) | Image resolution (in pixel) | The footprint used for landing points to the image (in pixel) | Size of one pixel (in pixel/cm) (size of container ~ 2.34cm) |
|---|---|---|---|
| (a) | 800x600 | 1x1 | 0.29x0.39 |
| (b) | 1200x1000 | 3x3 | 0.19x0.23 |
| (c) | 1400x1200 | 3x3 | 0.17x0.19 |
| (d) | 1400x1200 | 4x4 | 0.17x0.19 |
| (e) | 1600x1400 | 5x5 | 0.15x0.17 |
| (f) | 1650x1450 | 5x5 | 0.14x0.16 |

### 4.1.6. Range Image – Watershed Segmentation

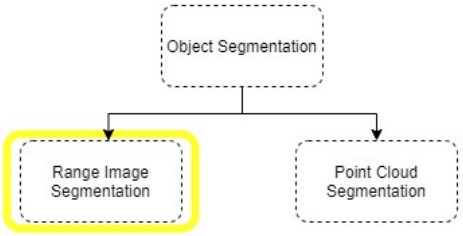



Figure 4-16 Next step in the methodology pipeline - this sub-section deals with the highlighted box (in yellow)

This section describes the process involved in the segmentation of objects using the watershed technique. The input to the method is the range image obtained by the projection of 3d points onto an image plane ***(m,n)***. This image-based method works on the 2d rectilinear grid points, which have the associated z-normal vector values assigned to them. The discontinuities in these values help determine the boundary and the foreground pixels. The following flowchart describes the processes involved in segmenting the image counterpart of the point cloud data.

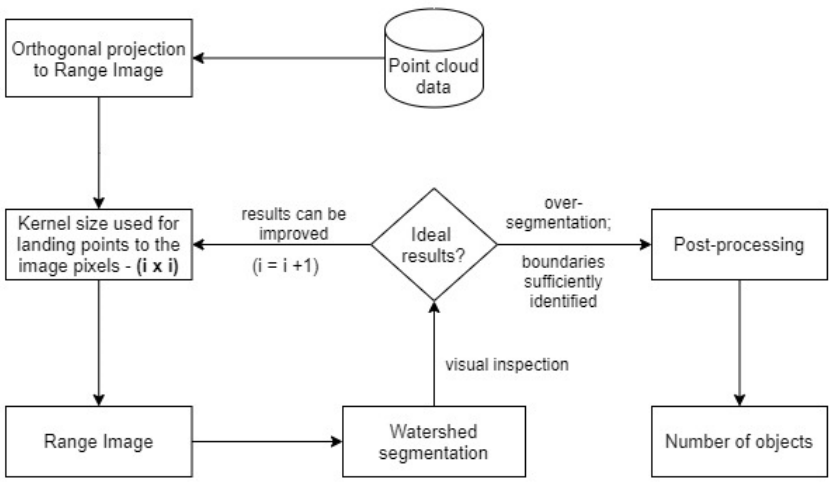



Figure 4-17 Flowchart outlining the process involved in Range Image Segmentation

As discussed in section 2.3.2, the watershed segmenting technique works by finding the low-intensity points in a grayscale image and starts to fill up until the water rises and meets the high-intensity points where barriers or segmenting lines are built. This essentially separates the two nearby objects. The method can have two approaches: top-down or bottom-up. The principle behind the top-down approach is that the maxima points are located, and the tracking is in the downward direction in search of the associated minima. The other starts at the bottom and continues to fill upward until the maxima are reached. A brief overview of the steps involved in this study –

1. The local maxima are identified by a binary thresholding technique, each of which will further form the catchment basins.
2. A distance transform function to compute the distance from the foreground pixel to the nearest pixel belonging to the background; the peaks are identified as the pixels with high values.
3. By using an inverse of the function at step 2, the peaks are flooded by using an upward descent until the boundaries are located.
4. The smaller segments are merged with neighboring segments using a statistical approach.
5. Refining the segment boundaries from step 4; by removing the overlap between adjacent segments.

#### 4.1.6.1. Background and Foreground Labeling

The obtained range image has noise that needs to be reduced. The edges of the objects of interest need to be preserved as they contain information on the boundaries between two adjacent regions. For this purpose, a Gaussian kernel is implemented with a window size of 3x3 pixels. The center pixel of the window has the largest value, and it decreases symmetrically as the distance from the center pixel increases. When the image is convolved, the boundaries are not suppressed as the horizontal and the vertical pixels of the Gaussian kernel have smaller values. After the application of a smoothing filter, the image is suppressed of noise to a considerable extent. The salt and pepper effect illustrated in **Figure 4-18 (b)** and the effect of noise removal in **Figure 4-18 (c).**

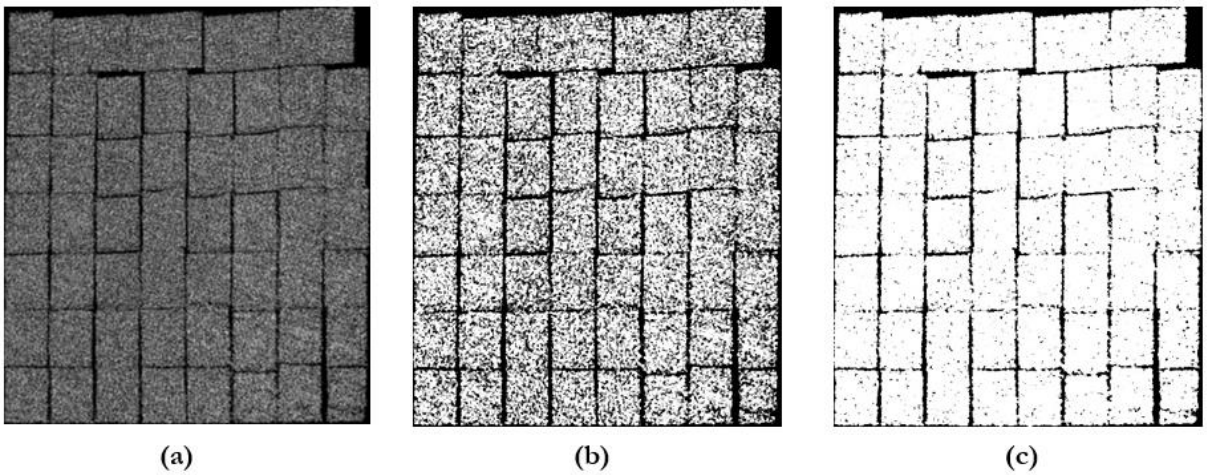



Figure 4-18 Selected range image - (a) in grayscale; (b) binary threshold image before noise removal; (c) binary threshold image after noise removal

As noted earlier, the zones dividing the adjacent catchment basins are known as the watershed lines. The initial step to applying the watershed algorithm is to produce a gradient image from the given grayscale image (Vo, Truong-Hong, Laefer, & Bertolotto, 2015). In a grayscale image, like that of our range image, there is a significant difference in the grayscale tones of the pixels that belong to the foreground and the

background. Thus, a binary thresholding filter is used to distinguish the two regions; **Figure 4-18 (c)**. The OTSU's binary thresholding algorithm is selected for this purpose. The algorithm works by finding an appropriate threshold by reducing the within-class variance using just the gray level histogram of the image (Goh, Basah, Yazid, Aziz Safar, & Ahmad Saad, 2018). It is set to assign the value one to the foreground objects and zero to the pixels belonging to the background. The range image is converted to a binary scale and the pixel values are now either zero or one. The values are stretched on a 256-bit range and it is expected that most of the background pixels have a value that corresponds to zero and the pixels that form the surface of the object have the grayscale value 256.

For an image $\boldsymbol{I}$ and its corresponding labeled image $\boldsymbol{L}$ –

- $\boldsymbol{I(m,n)}$ *and* $\boldsymbol{L(m,n)} \in \{\boldsymbol{0,1}\}$ (m,n is the pixel index) Equation (4)
- $\boldsymbol{L(m,n)} = \boldsymbol{0}$ (denotes the pixel as belonging to background) Equation (5)
- $\boldsymbol{L(m,n)} = \boldsymbol{1}$ (denotes the pixel as belonging to foreground) Equation (6)

However, there still exists a considerable amount of background pixels on the objects' surface. This mislabelling of the image pixels needs to be corrected, which would otherwise make the selection of local minima complex for the watershed algorithm. Morphological operators can be used to handle this problem on binary images (Jamil, Sembok, & Bakar, 2008). The application of morphological operators prepares the image for segmentation. The two morphological functions used here are erosion and dilation.

#### 4.1.6.2. Morphological Operation

Erosion (7) is an operation on the binary image $\boldsymbol{L}$, which uses a structuring element $\boldsymbol{M}$, such as a sliding window of size 3x3 or more. It changes the value of the pixel (m,n) in $\boldsymbol{L}$ from 1 to 0 if the result of moving $\boldsymbol{M}$ with (m,n) at its center is lesser than the value determined earlier. The size of $\boldsymbol{M}$ determines how much the object is thinned. On the other hand, dilation (8) is an operation using a similar structuring element $\boldsymbol{M}$ to slide through the binary image $\boldsymbol{L}$, where the pixel values are replaced if the value at its center is higher than the previously determined value. The object boundaries are grown using dilation. The size of the growing also depends on the window size. Erosion reduces the external noise; the dilation function is used for reducing the internal noise. While dilation reduces the holes within the object, it also increases the object boundaries. In order to preserve the object's actual bounds, an erosion function is applied post the application of dilation.

The two operations are run a suitable number of times over the image to achieve the expected results. In mathematical notation,

$$\boldsymbol{L} \ominus \boldsymbol{M} \quad \text{Equation (7)}$$

$$\boldsymbol{L} \oplus \boldsymbol{M} \quad \text{Equation (8)}$$

The next step is defining criteria for the upward descent, which the seeds need to follow. When the two nearby objects are nearly touching each other, a distance transform (DT) function can be used to separate each of them. The function works by computing the Euclidean distance to the closest background pixel for each of the pixels associated with the foreground. In this study, we have the objects of interest $\boldsymbol{S}$ and the complement $\boldsymbol{S'}$ refers to the black pixels belonging to the background. The $\boldsymbol{S'}$ are the pixels of interest to which the distance from $\boldsymbol{S}$ is calculated. The result of DT is a distance map $\boldsymbol{(D)}$ which has the values that are the distance from this pixel $\boldsymbol{(p)}$ to its nearest $\boldsymbol{S'}$, given by –

$$\boldsymbol{D(p) = \min\{d(p,q) \mid q \in S'\}} \text{ (Fabbri et al., 2008)} \quad \text{Equation (9)}$$

(where D is the distance map of image L)

$$\boldsymbol{d(p,q) = \sqrt{(p_x - q_x)^2 + \left(p_y - q_y\right)^2}} \qquad \text{Equation (10)}$$

The figure below shows an example of a distance image with numerical values. The values shown are squared representations of the Euclidean distance function.

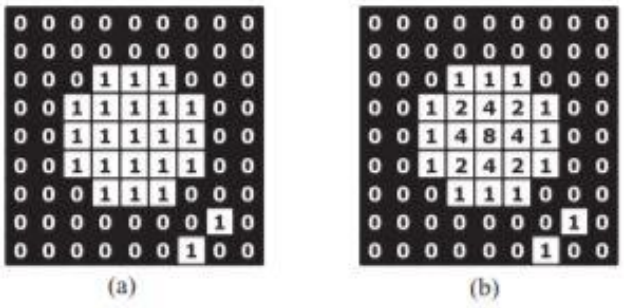



Figure 4-19 A numerical example of distance transform - (a) Binary image; (b) Euclidean distance computed from each pixel to its nearest black pixel. Source (Fabbri et al., 2008)

#### 4.1.6.3. Watershed Algorithm

Over-segmentation is an inevitable issue that results from watershed-based segmentation (Parvati, Prakasa Rao, & Mariya Das, 2008). This is overcome by using markers that determine the seed regions from where the flooding needs to start. Seed pixels are defined as pixels with magnitudes less than a threshold value. The water level starts rising from these seed points and by using a connected component, the non-marked pixels that are at the current level of water are now associated with the same label (Zhang et al., 2014). The watershed is applied to the calculated distance transform image such that the water starts filling up from the pixels identified as peaks and watersheds or dams are constructed to separate the adjacent objects where the water meets the object surface. The watersheds are visualized by drawing contour lines and identifying the segmented object by a unique segment number. Rectangular shapes are defined over each of the identified segments to extract compact structures from the generated contours. These rectangles are a result of fitting the bounding box to the generated contours. The bounding box of each segment returns the top-left pixel indices along with the width and the height of the segment.

Bounding box information = [*x_minimum, y_minimum, width, height*]

Over-segmented results are indeed better than under-segmented results as each object is identified at least once. In the next stage, statistical post-processing methods further remove the over-segmentation.

#### 4.1.6.4. Box-object Detection

The first step in post-processing is to separate the bounding boxes into ideal ones and the ones that are too small to be an actual segment. A threshold range to split the ideal and too-small segments are arrived at by plotting the histogram of the areas of the bounding boxes. The average segment area is found and a standard deviation of a suitable value that fits the data is selected, the minimum and maximum threshold ranges are computed. The next step is to merge the identified small segments based on a neighborhood criterion. The smaller segments are first sorted based on the x-pixel index value. Euclidean distance is computed to check the neighborhood of one segment with the candidate segments and those segments

that are well within the threshold distance range are merged. The threshold for the merge step is set by computing the distance between the current segment and all the other segments that could be a candidate for the merge function. The distances are averaged and a suitable percentage of this average distance is used as the minimum threshold. The smaller segments that have not been merged with any of the other segments are retained. The areas of the retained segments are checked and if their areas are more than half of the area of the segments considered to be ideal, they are kept otherwise discarded. The over-segmentation is aimed to be considerably reduced at this step.

Since the objects in real-time can not be found over one another, two adjacent bounding boxes need to be pruned in such a way that the segment areas no longer overlap; **Figure 4-20**. For this, the adjacent boxes are taken and the intersection over union (IoU) percentage is computed for each pair of boxes. The IoU values that are significant, that is, when two object areas are shared, they are labeled as overlapping. The area of the overlap region is calculated at this stage.

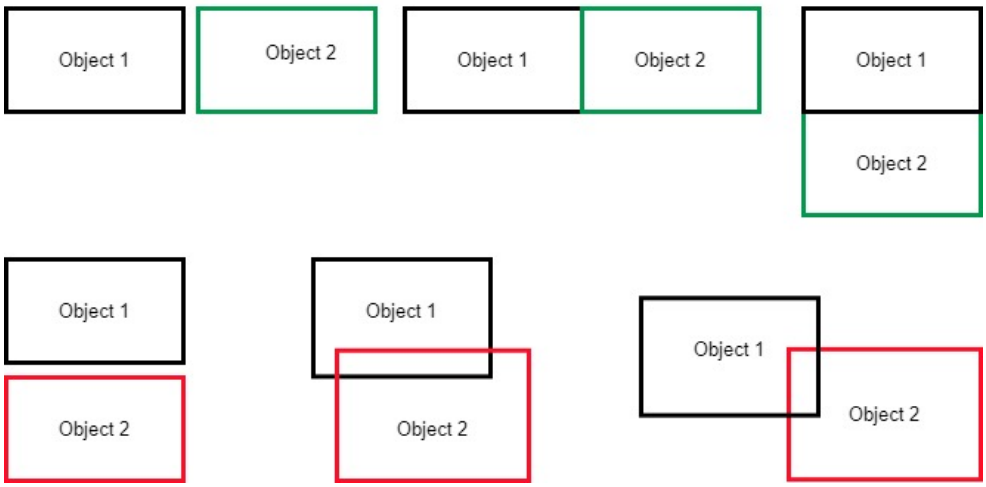



Figure 4-20 The accepted boundaries for the objects are marked green and the non-ideal scenarios are marked in red

As the overlapping can occur on either side of the object, the first step in overlap removal is to identify if the candidate bounding box is on the left, right, bottom, or top of the selected bounding box. The two overlapping boxes are assigned half the area from the overlap region and are moved away from one another. The corresponding minimum coordinates (x and y) of the overlapping pair are also adjusted such that the dimensions of the objects do not increase; **Figure 4-21**.

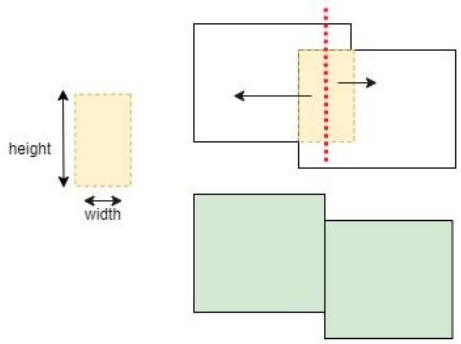



Figure 4-21 Figure depicting how the overlap region is shared between the two adjacent objects; overlap portion (yellow), final segments (green)

## 4.2. Object Geometry

The watershed segmentation result on the range image is selected for extracting object geometry in this study. The watershed segmentation results are mapped onto the point cloud data and the object poses and dimensions are extracted from the point cloud. This is because the point cloud contains more information about a scene when compared to its image counterpart. The lidar point indices that fall onto the pixels during the projection to a range image are tracked to estimate the object parameters.

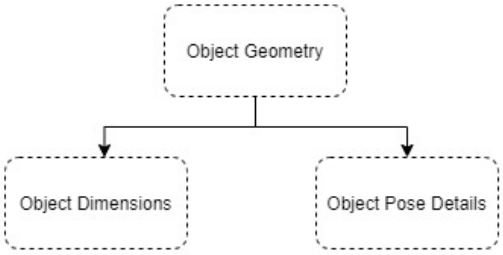



Figure 4-22 The steps in extracting the geometry of the objects

### 4.2.1. Reprojection to 3d point cloud

The lidar points that fall onto each image pixel are tracked by uniquely identifying the points by an index value. Multiple points may land on a single pixel and each point has a corresponding image pixel index onto which it has landed. After the segmentation process, each image pixel is assigned a segment label. The lidar points are now assigned the segment labels their corresponding image pixels have.

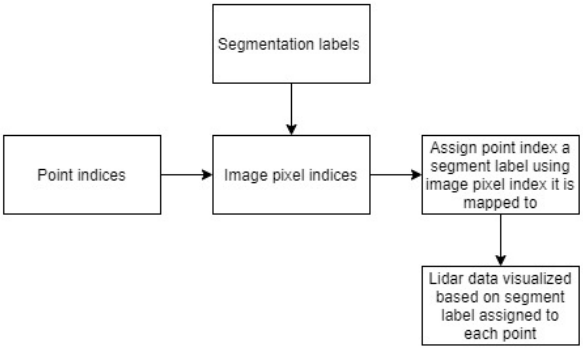



Figure 4-23 Overview of the steps involved in re-projecting the range image segmentation results to point cloud

The re-projection is accomplished by assigning each point in the 3d point cloud an integer label ***N***. A point at ***I(m,n)*** in the segmented range image is labeled with ***N(m,n)***. This is considered a label image. The points that do not fall onto any pixel, such as those with missing data, are assigned ***NaN*** values. The method then partitions the 3d point cloud data into a number of segments using the segment labels assigned from the image.

### 4.2.2. Object Dimensions

The object dimensions are estimated by finding the minimum and maximum values of x and y coordinates from the point cloud. For an object defined by a bounding box, the minimum values of x and y coordinates occur at its bottom-left corner and the maximum coordinate occurs at the top-right corner by the cartesian system it is defined. If the bottom-left corner is given by point ***p(x-minimum,***

***y_minimum)*** and point ***q(x_maximum, y_maximum)***, then the dimensions are computed by using the Euclidean distance -

$$\boldsymbol{width} = \sqrt{\left(\boldsymbol{p_{x_{maximum}}} - \boldsymbol{q_{x_{minimum}}}\right)^2} \qquad \text{Equation (11)}$$

$$\boldsymbol{height} = \sqrt{\left(\boldsymbol{p_{y_{maximum}}} - \boldsymbol{q_{y_{minimum}}}\right)^2} \qquad \text{Equation (12)}$$

#### 4.2.3. Object Pose Details

The 3d pose of the objects is computed by finding the centroid of each segmented object. The 3d lidar points belonging to the same segment are taken and as the results from range image segmentation are refined by fitting rectangular bounding boxes, the center of this bounding box gives an approximation for the object center point. For a given $\boldsymbol{n}$ number of points, each point is defined by (x,y,z) coordinates and the centroid is computed by –

$$\boldsymbol{centroid} = ((\boldsymbol{x_1} + \boldsymbol{x_2} + \cdots + \boldsymbol{x_n})/\boldsymbol{n}), (\boldsymbol{y_1} + \boldsymbol{y_2} + \cdots + \boldsymbol{y_n})/\boldsymbol{n}), (\boldsymbol{z_1} + \boldsymbol{z_2} + \cdots + \boldsymbol{z_n})/\boldsymbol{n})$$

$$\boldsymbol{centroid} = \left(\boldsymbol{x_{c,}} \boldsymbol{y_{c,}} \boldsymbol{z_{c,}}\right) \qquad \text{Equation (13)}$$

The next step is to estimate the orientation of the segmented objects. For this, the angle between the surface normal vectors of the local and the global plane are computed. The global plane is the plane that contains all the object points, and the local plane refers to the plane that is fit to the 3d points of an individual segment. The surface normal vectors of both planes are computed and the angle between the two vectors, also known as the axis-angle, is given by –

$$\boldsymbol{a} \cdot \boldsymbol{b} = \| \boldsymbol{a} \| \| \boldsymbol{b} \| \cos \boldsymbol{\alpha} \qquad \text{Equation (14)}$$

(where, a and b are the two vectors; $\boldsymbol{\alpha}$ is the angle between a and b)

In Euler notation, the orientation of an object is defined by roll, pitch and yaw angles. The three angles refer to the rotations about the x-axis, y-axis and z-axis, respectively. In order to transform a vector from one cartesian system to another, the vector is rotated about a rotation matrix ***R.***

A rotation of psi ψ radians about the x-axis is defined by –

$$\boldsymbol{R_X(\psi)} = \begin{bmatrix} \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\cos \psi} & \boldsymbol{-\sin \Psi} \\ \boldsymbol{0} & \boldsymbol{\sin \psi} & \boldsymbol{\cos \Psi} \end{bmatrix} \qquad \text{Equation (15)}$$

A rotation about the y-axis is given by theta θ radians -

$$\boldsymbol{R_y(\theta)} = \begin{bmatrix} \boldsymbol{\cos \theta} & \boldsymbol{0} & \boldsymbol{\sin \theta} \\ \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} \\ \boldsymbol{-\sin \theta} & \boldsymbol{0} & \boldsymbol{\cos \theta} \end{bmatrix} \qquad \text{Equation (16)}$$

A rotation of phi $\phi$ radians about the z-axis is given by -

$$\boldsymbol{R_z(\phi)} = \begin{bmatrix} \boldsymbol{cos\,\phi} & \boldsymbol{-sin\,\phi} & \boldsymbol{0} \\ \boldsymbol{sin\,\phi} & \boldsymbol{cos\,\phi} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} \end{bmatrix} \qquad \text{Equation (17)}$$

The angles $(\psi, \theta, \phi)$ are the Euler angles. The general rotation matrix **R** is given by -

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{R_{11}} & \boldsymbol{R_{12}} & \boldsymbol{R_{13}} \\ \boldsymbol{R_{21}} & \boldsymbol{R_{22}} & \boldsymbol{R_{23}} \\ \boldsymbol{R_{31}} & \boldsymbol{R_{32}} & \boldsymbol{R_{33}} \end{bmatrix} \qquad \text{Equation (18)}$$

$$\boldsymbol{R} = \boldsymbol{R_z(\phi) R_y(\theta) R_x(\psi)} \qquad \text{Equation (19)}$$

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{\cos\theta\ cos\phi} & \boldsymbol{\sin\psi \sin\theta \cos\phi - \cos\psi \sin\phi} & \boldsymbol{\cos\psi \sin\theta \cos\phi + \sin\psi\ sin\phi} \\ \boldsymbol{\cos\theta\ sin\ \phi} & \boldsymbol{\sin\psi \sin\theta \sin\phi + \cos\psi \cos\phi} & \boldsymbol{\cos\psi \sin\theta \sin\phi - \sin\psi\ cos\ \phi} \\ \boldsymbol{-sin\theta} & \boldsymbol{\sin\psi \cos\theta} & \boldsymbol{\cos\psi \cos\theta} \end{bmatrix} \qquad \text{Eq(20)}$$

With the above rotation matrix **R** from equation (20), the angles $(\psi,\ \theta,\ \phi)$ can be computed by equating the elements in the **R** matrix with the corresponding elements from equation (18) (Slabaugh, 1999).

In conclusion, for each segmented object, the parameter estimation step provides the following information –

1. Main dimensions, i.e: width and height
2. 3d position w.r.t the sensor positions
3. 3d orientation with respect to the global plane formed by all the points representing the scene

# 5. RESULTS

This chapter elaborates the results obtained in each step of the methodology. First, we look at the object segmentation on the range image (section 5.1), followed by a comparison between the two segmentation methods (section 5.2). Finally, dimensions and pose estimates are examined (section 5.3).

Note that the results from section 5.1 are taken as input for section 5.3.

## 5.1. Object Segmentation on Range Image

This section covers the results of watershed segmentation on the range image and the post-processing steps involved. The section goes on to visualizing and analyzing the results at each step of the process. It displays the step by step results of an ideal case.

By experimenting, the suitable kernel size to be used for the projection is a footprint size of 5x5 pixels. To better understand the effects of the kernel (k) size in the projection-based method, segmentation results on the same are examined. As the size of k increases, the performance of the algorithm increases, but the computational speed reduces.

The improvement from k=5 (Table 5-1 image (e)) with 1650x1450 resolution to k=6 (f) with 1800x1600 resolution is not much as of that from k=3 (b) with 1400x1200 to k=5 at 1650x1450 (e) resolution. This suggests that a region with k=5 has essentially included the significant neighboring points. Increasing the resolution from 1400x1200 to 1650x1450 improves the results, but the computational load is relatively higher. However, when the resolution is increased from 1600tox1400 (d) to 1650x1450 (e), the performance reduces. This is because the footprint used for the projection is kept the same, resulting in a smaller receptive field. Thus, for effective results, the kernel size and the resolution are increased with respect to one another. The results of watershed segmentation are visualized by drawing contour lines where the watershed lines are built over the different range image projected using varying footprint sizes. The effect of different footprints and image resolutions is summarized in Table 5-1

Table 5-1 Effects of varying footprint size on the segmentation results

| Image (Figure 5-1) | Footprint size – k (in pixels) | Image Size (in pixels) | Detected number of objects by watershed (ground truth = 53 ) | Runtime (in seconds) |
|---|---|---|---|---|
| (a) | 3x3 | 1200x1000 | 42 | 7.66 |
| (b) | 3x3 | 1400x1200 | 51 | 7.94 |
| (c) | 4x4 | 1400x1200 | 55 | 13.7 |
| (d) | 5x5 | 1600x1400 | 63 | 19.9 |
| (e) | 5x5 | 1650x1450 | 58 | 21.9 |
| (f) | 6x6 | 1800x1600 | 58 | 25.2 |

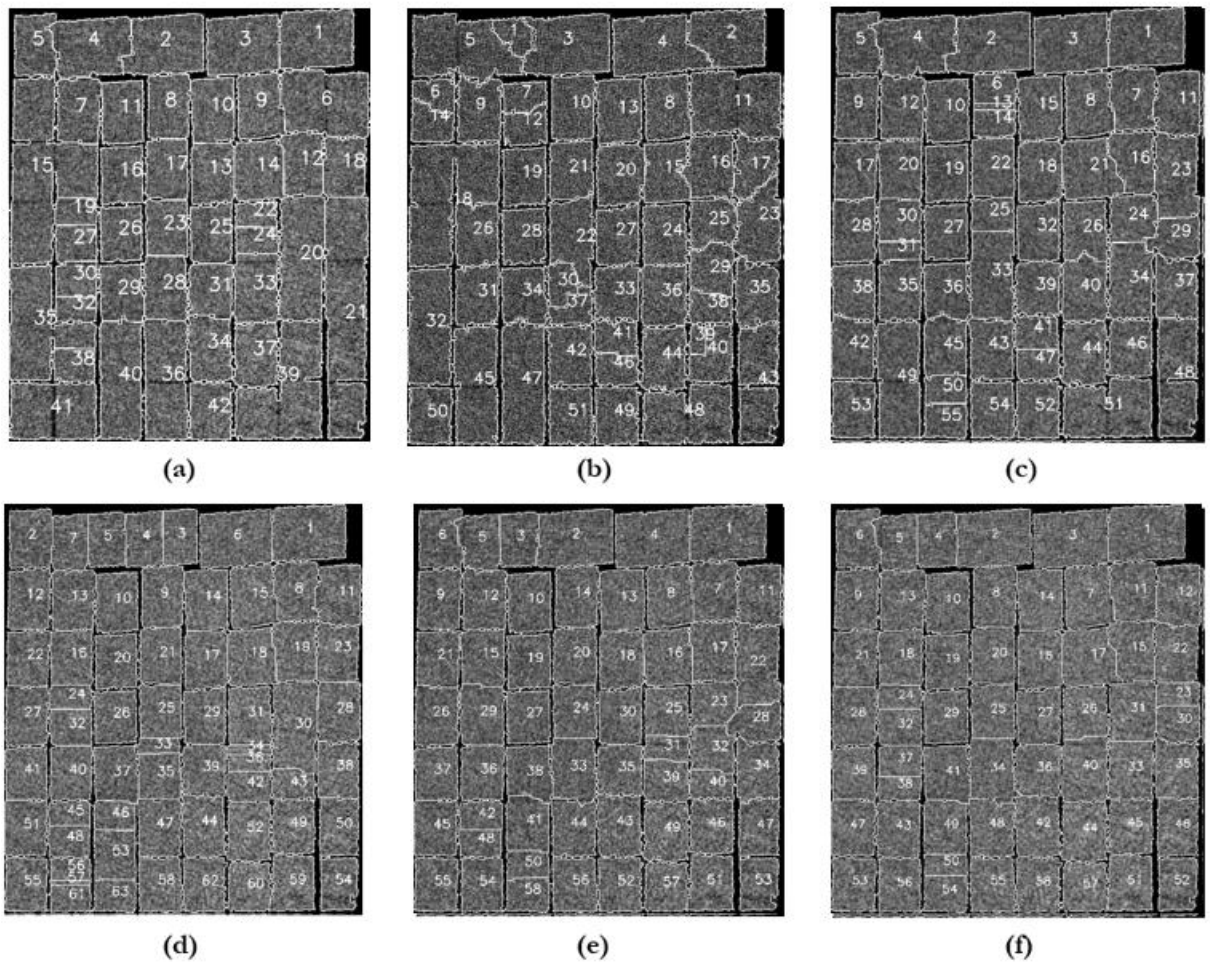



Figure 5-1 Visual representation of segmentation results of varying footprint and image resolutions on the range image; refer Table 5-1

As we are more interested in demarcating the background and the foreground pixels, the grayscale image is subjected to binary thresholding. **Figure 5-2 (a)** shows the result of applying a gaussian smoothening filter still has some noisy areas. A combination of morphological filters can improve the results from the previous step. A series of dilation and erosion steps are performed on this image using a square-shaped structuring element of size 3x3 pixels. The dilation step removes the holes within the object surfaces and increases the boundaries of the box objects. The erosion enhances the dis-joint between adjacent objects and reduces the boundaries of the object grown during dilation; **Figure 5-2 (b)**. The distance transform **(Figure 5-2 (c))** labels are obtained by using the binary image from the result of the morphological operations.

The inverse of the obtained distance transform function **(Figure 5-2 (d))** identifies the minima regions during the watershed segmentation. The water level rises from the pixel regions marked as peaks and the results of the watershed lines built to separate the two regions are visualized by generating contour lines shown in **Figure 5-2 (f**). Each separate region is identified individually by assigning a unique label and the resulting image is visualized in **Figure 5-2 (e)**.

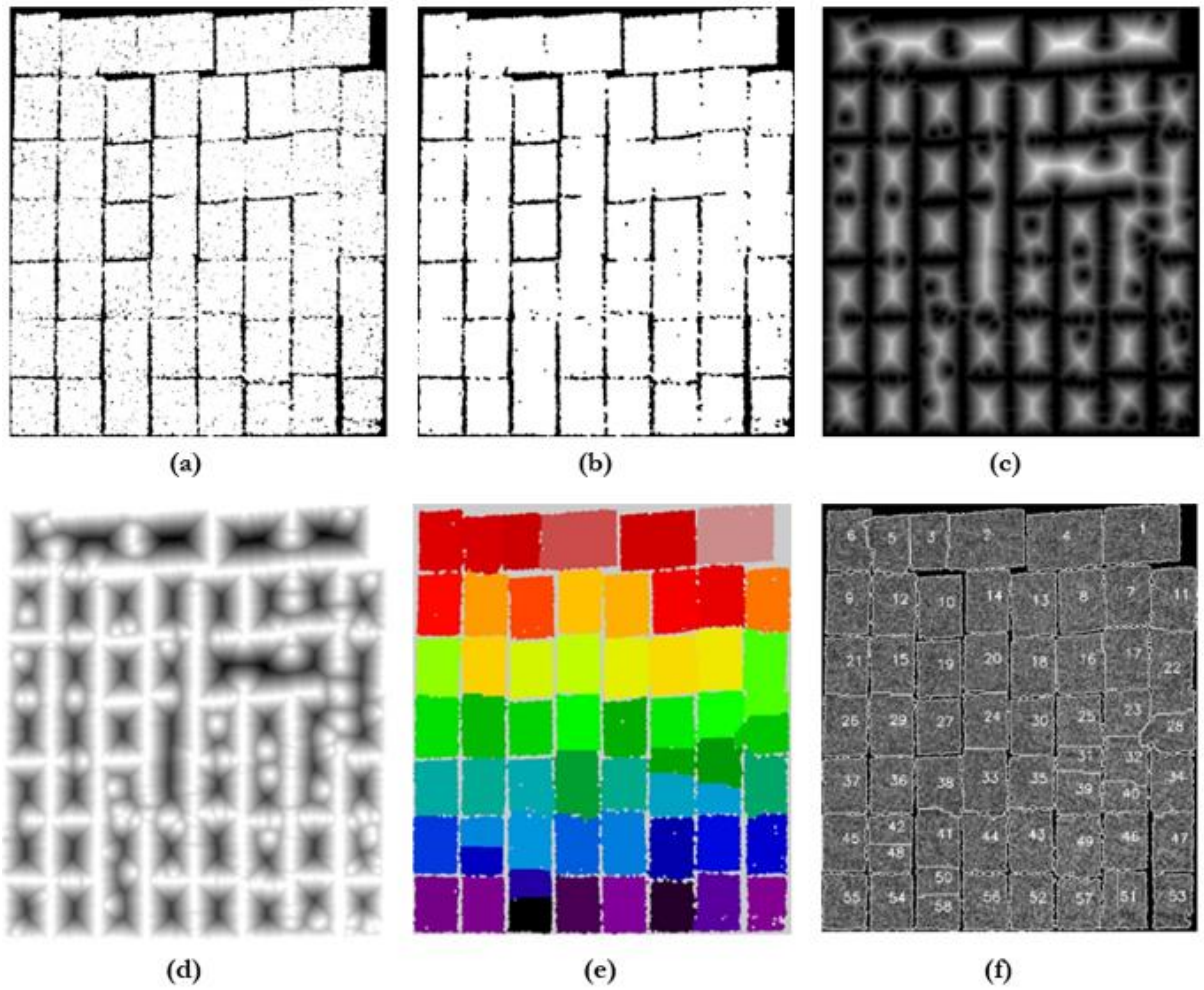



Figure 5-2 Figure showing results of (a) Gaussian smoothing; (b) Morphological operations; (c) Distance transform function; (d) Inverse distance transform; (e) Unique labels to each individual region; (f) Contour lines drawn to separate two adjacent regions with unique segment label on the grayscale image

The following are the results of each of the steps involved in the post-processing phase –

1. Minimum bounding rectangular boxes are fitted to each of the generated contours that uniquely bound the individual segments. This step refines and produces more compact structures for the box-type objects. **Figure 5-3(a)**
2. The over-segmented box objects and the segments identified with having small areas compared to the other segments are highlighted in red. **Figure 5-3(b)**
3. The resulting bounding boxes of successful merging of the identified smaller bounding boxes from step 2. The too-small ones are discarded by identifying their areas. **Figure 5-3(c)**
4. The results of step 3 are replaced in step 1. **Figure 5-3(d)**
5. The final adjusted bounding boxes that are free of overlap, visualized in green over the grayscale image, with final segment labels. **Figure 5-3(e)**
6. The manually generated ground truth labels for the displayed grayscale image. **Figure 5-3(f)**

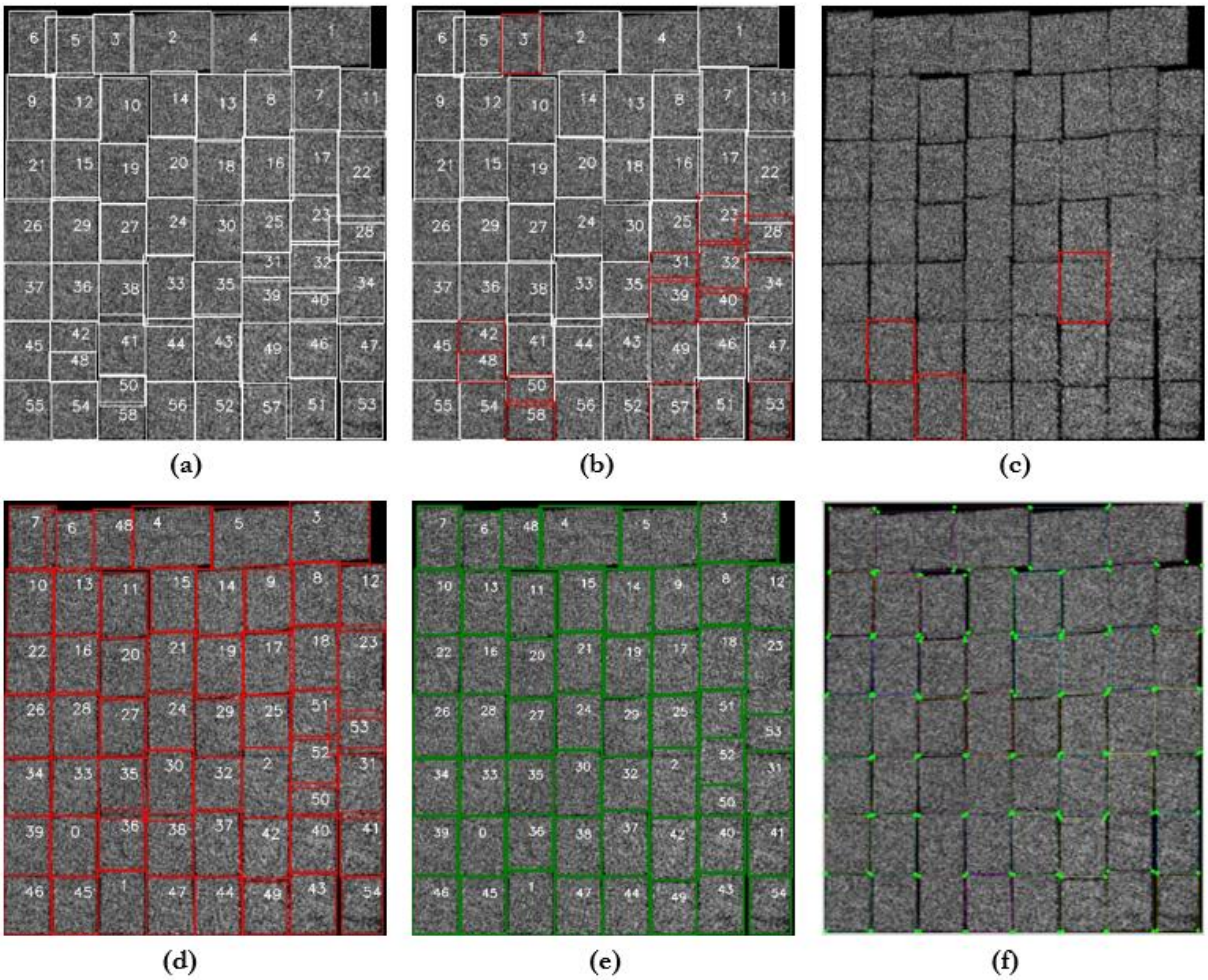



Figure 5-3 Figure showing results of (a) bounding boxes fitted over generated contours; (b)identifying small segments (red) and ideal segments (white); (c) neighboring smaller segments merged (red); (d) merging step combined with ideal segments; (e) bounding boxes that are pruned for overlap; (f) ground truth labels generated manually

The above are the results from one ideal case where the number of segments produced by the algorithm after a series of post-processing steps is close to the ground truth values. The results are evaluated for their accuracy by computing its precision, recall and f1-score; **Table 5-2**. The evaluation of instance segmentation models is complex. However, similar methods to object detection are employed, with IoU masks being used instead of the actual bounding boxes. To evaluate the prediction of the bounding boxes generated over the range image, each of the bounding box pairs (the one that fits the object exactly and the one that is obtained from the algorithm) are compared. Based on this, the following are defined –

1. True positive – when the actual and the resulting bounding box from the algorithm has an IoU score that exceeds a pre-defined threshold value (the value is set at 0.80)
2. False positive – when the bounding box from the algorithm has no corresponding bounding box from ground truth
3. False negative – when the ground truth bounding box has no associated predicted bounding box (IoU scores lesser than the set threshold, in this case)

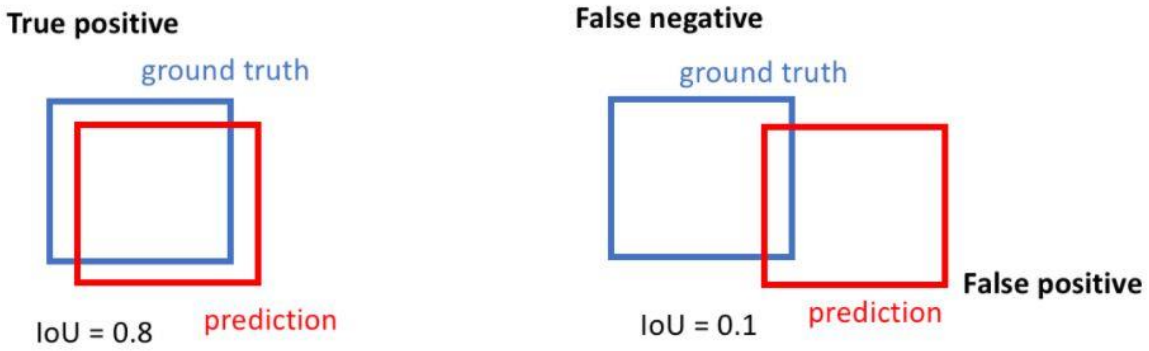



Figure 5-4 Metrics for computing F1 score

$$\text{Precision} = \text{true positives} / (\text{true positive} + \text{false positive}) \quad \text{Equation (21)}$$

$$\text{Recall} = \text{true positives} / (\text{true positive} + \text{false negatives}) \quad \text{Equation (22)}$$

$$\text{F1-score} = 2 * [(\text{precision} * \text{recall}) / (\text{precision} + \text{recall})] \quad \text{Equation (23)}$$

Table 5-2 Evaluation of watershed segmentation results

| Ground Truth | Watershed segmentation | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Results | | | | | Quantitative evaluation | | |
| | Initial results | True positives | False positives | False negatives | Post-processing | Precision | Recall | F1-score |
| 53 | 58 | 46 | 5 | 7 | 55 | 0.90 | 0.87 | 0.88 |

## 5.2. Comparison of Segmentation results

This section discusses the results of the point cloud segmentation by segment growing and compares the generated results with that of the watershed segmentation qualitatively. The results of segment growing is displayed in **Figure 5-5 (a)**. A combination of over and under-segmentation can be seen, and many white points remain unclassified as they do not belong to any of the segments. These are points that do not have similar nearby points to grow into a segment. In order to get smooth and more defined segments from the results of segment growing, the isolated points not belonging to any of the segments are merged with the segment label that is most occurring within a defined neighborhood. For this purpose, majority filtering is applied. The results are smooth segments but still deviate from ideal segmentation results.

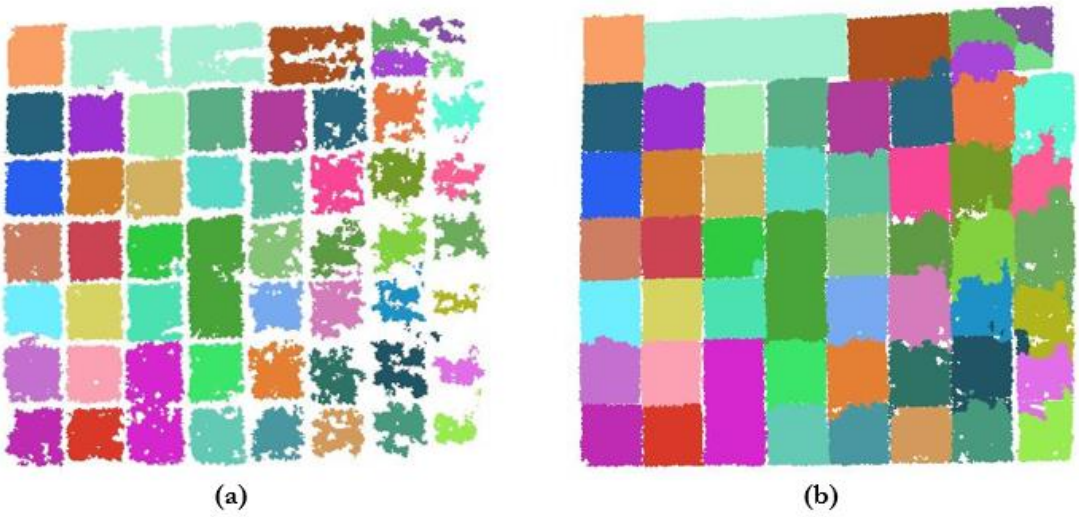



Figure 5-5 Point cloud segmentation results – (a) Segment growing; (b) Majority filtering

The bounding box coordinates obtained from the range image for each of the segmented objects of interest are re-projected into the 3d point cloud. This is done by tracking the pixel indices and the corresponding laser point indices as discussed in sub-section 4.2.1. After the segmentation process, the pixels are re-labeled based on the objects' segment labels. These segment labels are then transferred back to the lidar points. The result is visualized using Cloud Compare software, where the 3d points are colored based on the segment labels.

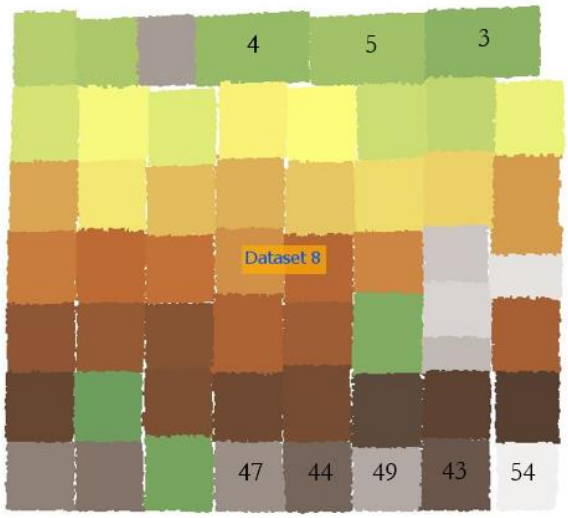



Figure 5-6 Results of watershed segmentation on the range image projected back to the point cloud data; some segments are annotated with their segment labels for reference in section 5.3

The watershed segmentation identifies most of the box objects precisely. In the case of segment growing on the point cloud directly, the results contain some over-segmented and under-segmented objects. In the scope of this research, the post-processing on direct point cloud segmentation has not been explored.

**Figure 5-7** shows the result of the segmentation of both methods. It is noticed that the different datasets used in this study were captured by different sensor geometry. The dataset shown in **Figure 5-7** is scanned from the top-left portion and hence the objects on the far end (last column) are the most affected using the direct point cloud segmentation.

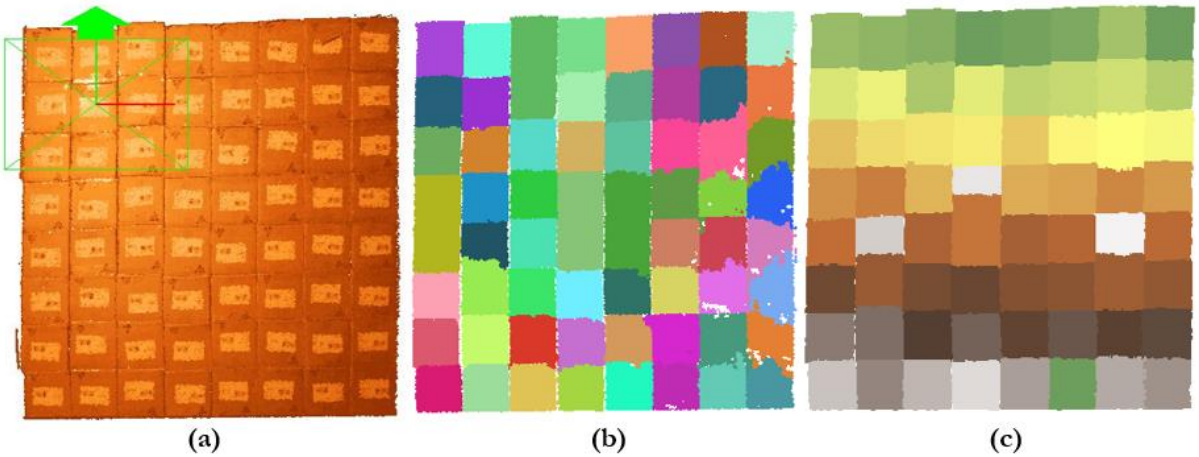



Figure 5-7 Figure illustrating (a) 3d point cloud with origin point marked; (b) segment growing results; (c)watershed results

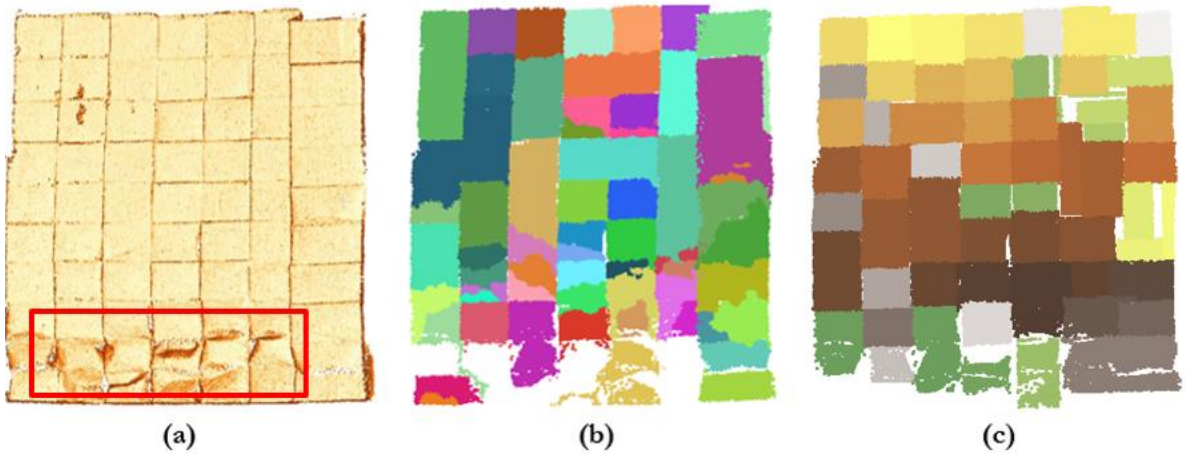



Figure 5-8 Figure illustrating (a) 3d point cloud of a cargo container having box-type objects and sack-type objects (red) colored based on normal vector; (b) segment growing results; (c) watershed results

Another interesting case is of the mixed type of objects – where the cargo container contains box objects and sack-like objects; **Figure 5-8 (a)**. Since the method proposed in this study works on differentiating the face of the object from its edge by identifying the differences in its local normal vector, it does not work well on objects with curved surfaces (i.e: gunny bags or sacks). The varying point normal can be visualized in the figure below and the results of both segmentation methods are displayed. The last two rows that are having the sacks are not well segmented. Also, the box object on the far-right column suffers the most. This is because this column of objects is not in line with the objects from the other columns. The positioning of the sensor is such that it scans this container from the top-left corner. Thus, the sides of the objects in the last column are also scanned. This causes trouble when the point values are mapped to the pixel of the image. The segment growing does not detect most sack-like objects, and the segmentation on the box-type objects is also poor.

The number of segmented box objects from both the segmentation methods is summarized in **Table 5-3**.

Table 5-3 Results of both segmentation methods with ground truth - results that are close to ground truth are highlighted in green for both the methods. Some datasets use different parameter values and are highlighted in light orange. Dataset 9, with a combination of box objects and sacks, has poor results and is highlighted in light red. Datasets 2 and 5 have better results using the projection-based image method, and change in parameter values used on same dataset affects the results (purple)

| Data set | Ground Truth | Detected number of objects | | | | | |
|---|---|---|---|---|---|---|---|
| | | Watershed Segmentation | | | | Segment Growing | |
| | | Parameter value for point clustering | Initial results | Final results | Over and under segmentation | Results | Over and under segmentation |
| **1** | 24 | 70 | 16 | 16 | Poor results | 26 | 3; 1 |
| **2** | 40 | 70 | 40 | 38 | 0; 2 | 34 | 0; 6 |
| **3** | 64 | 70 | 66 | 64 | Ideal results | 60 | 0; 4 |
| **4** | 24 | 70 | 51 | 51 | Poor results | 28 | 4; 0 |
| **4** | 24 | 115 | 31 | 30 | 7; 1 | - | - |
| **5** | 40 | 70 | 41 | 39 | 1; 2 | 42 | 6; 4 |
| **6** | 61 | 70 | 64 | 61 | 2; 2 | 60 | 3; 3 |
| **7** | 53 | 70 | 51 | 49 | 7; 2 | 52 | 6; 5 |
| **7** | 24 | 55 | 58 | 58 | 6; 1 | - | - |
| **8** | 53 | 70 | 58 | 55 | 2; 0 | 54 | 2; 3 |
| **9** | 58 boxes + sacks | 70 | 67 | 60 | Difficult to interpret | 59 | Difficult to interpret |
| **10** | 24 | 70 | 35 | 35 | Poor results | 24 | 1;1 |

The over and under segmentations have been reduced compared with the direct point cloud segmentation method in dataset 2 and dataset 5. Image-based method works the best on objects of almost uniform size as in dataset 3. However, the resulting accuracy is low when the dataset has a combination of object sizes. By varying the parameter values involved in the process for every dataset, the accuracy is better achieved for each of them. Hence, the differences in the measuring geometry contribute to a decrease in the accuracy of the segmentation results while keeping the same parameter values across the datasets.

## 5.3. Dimensions and Pose Estimates

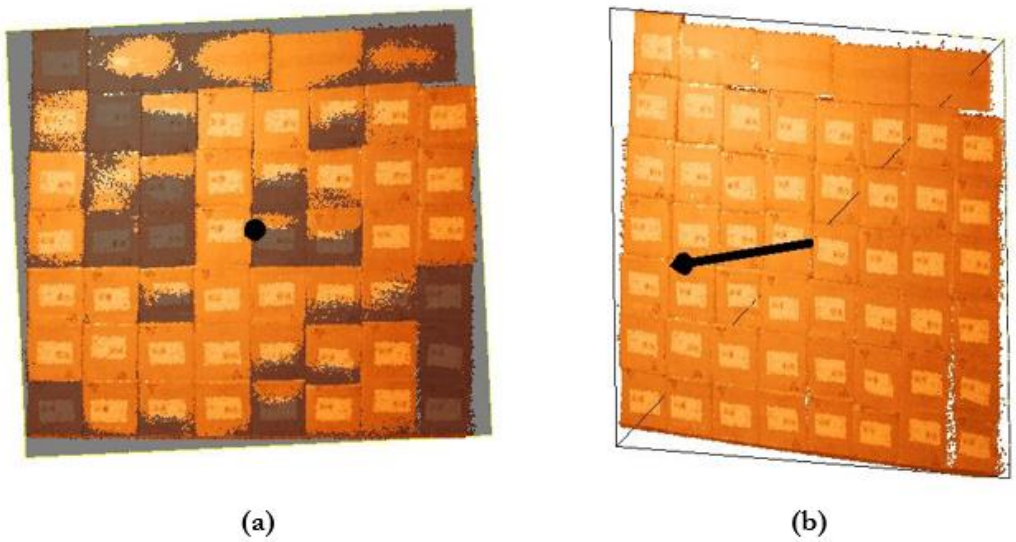
For each of the segmented objects, its dimensions and pose parameters are obtained. **Table 5-4** displays the dimensions for the dataset discussed in section 5.1.

The 3d position of the segmented objects denote the center of mass of each segment and the positions are obtained relative to the (0,0,0) coordinates on the stacked box-pile. To understand the object positions for the box-picking operation, each segmented box-object orientation is found with respect to an arbitrary position. Initially, the points were transformed into a base where the front view of the box-pile was entirely visible. The orientation of the visible surface is represented by the unit normal to its plane, the

segmented object points are then used to fit a plane to each individual object. Since we view each object's front face when looking forward to the stacked objects, alpha (α) is defined as the angle between the surface normal vector of the plane consisting of all object points and the surface normal vector of each segmented object plane. The global plane generated with all the points representing the scene and its respective normal vector is shown in **Figure 5-9**.

Table 5-4 Dimensions of the segmented objects

| **Box object ID (Figure 5-6)** | **Object Dimensions (in cm)** | |
|---|---|---|
| | **Object width** | **Object height** |
| 4 | 47.8 | 31.1 |
| 5 | 48.2 | 30.0 |
| 3 | 48.9 | 30.8 |
| … | … | … |
| … | … | … |
| 47 | 29.6 | 29.2 |
| 44 | 29.2 | 29.3 |
| 49 | 28.9 | 26.8 |
| 43 | 29.6 | 30.4 |
| 54 | 25.0 | 28.2 |



(a) (b)

Figure 5-9 Global plane generated by fitting all the points representing the scene – (a) front view of point cloud with the generated plane visible in grey color and laser points in orange; (b) side view of the point cloud with normal vector to the generated plane pointing outwards (black arrow)

The resulting values for the angles are given in **Table 5-5**. From the obtained results, the box objects are all aligned neatly with not much deviation. It is thus easier to position the robotic arm to do the picking operation.

Table 5-5 Pose details of the segmented objects

| **Box object ID (Figure 5-6)** | **Object centre (in cm)** | | | **Angle α between the two planes (in degrees)** | **Euler angles (in degrees)** | | |
|---|---|---|---|---|---|---|---|
| | **X** | **Y** | **Z** | | **Yaw angle** | **Pitch angle** | **Roll angle** |
| 4 | 0.58 | -0.03 | -2.37 | 5.4 | -5.3 | 1.1 | -0.1 |
| 5 | 1.081 | -0.03 | -2.37 | 4.5 | -3.3 | 3.0 | -0.1 |
| 3 | 1.56 | 0.00 | -2.39 | 6.2 | -5.7 | 2.5 | -0.2 |
| … | … | … | … | … | … | … | … |
| … | … | … | … | … | … | … | … |
| 47 | 0.59 | -1.85 | -2.20 | 2.6 | -1.0 | 2.4 | -0.0 |
| 44 | 0.88 | -1.84 | -2.22 | 2.6 | -1.8 | 1.9 | -0.0 |
| 49 | 1.18 | -1.85 | -2.21 | 3.4 | -3.0 | 1.6 | -0.0 |
| 43 | 1.47 | -1.83 | -2.22 | 3.2 | -2.4 | 2.0 | -0.0 |
| 54 | 1.77 | -1.83 | -2.26 | 2.5 | -1.9 | 1.6 | -0.0 |

# 6. DISCUSSION

This research started with an aim to make maximal use of the single-shot lidar scans and ended using the projection-based method to segment the objects of interest and subsequently find the pose parameters along with their dimensions from the point cloud data. The downside of aiming for maximal accuracy is the computational complexity. The implementation of the segmentation on an image is computationally less heavy due to its reduced-dimensionality; however, the projection of point cloud to range image takes higher time. The results are then re-projected to the 3d point cloud to extract finer details for the segmented objects.

## 6.1. Object Segmentation

As the cargo container is also scanned, some points belong to the back wall of the container as well. These points are present between two adjacent objects and cause problems in understanding the object edges. The scanning geometry affects the objects placed away from the scanner, and the varying point density is then partially tackled by downsampling the point cloud. However, downsampling the data contributes to loss of information and a trade-off between loss of information and achieving close to uniform density is made. The varying point density in the data is further handled by increasing the footprint size during the point cloud projection onto an image. It maps each point to the number of pixels defined by the footprint size and as the footprint size increases, the computational speed reduces; **Table 5-1**.

The measurement geometry of the datasets differs as they are scanned using different scanner positions. Hence, the values of the parameters used during the application of watershed segmentation for some of the datasets differ. Datasets 2 and 5 are a clear case of measuring geometry affecting the results of segmentation results; **Appendix I**. Although the contents of the container in both remain the same, the positioning of the sensor differs in both cases. Therefore, by altering the values of the parameters used in processing the image, the results differ. Due to the different positions from which the cargo containers are scanned, the ideal results that could be obtained for each dataset would depend on specific values set for the parameters involved. Also, when the container is being unloaded, several objects are unloaded after the first scan, revealing objects that are now farther away from the scanner. This impacts the point density and, therefore, the parameter value set on the point clustering varies for different datasets.

In the post-processing steps discussed in 4.1.6, the threshold value set for merging two adjacent box objects when the segmented region is too small is computed statistically. In some cases, the candidates to be merged could belong to two different segments that belong to box-objects on different rows or columns; **Appendix II**. The smaller candidate segment could be incorrectly merged with either of the two adjacent smaller segments. The threshold also impacts the identification of smaller and ideal segments. When the cargo container has objects of the same dimensions, the segmentation works well. However, in a case with at least four different sizes of objects, the threshold values set for identifying the smaller objects and the threshold set on merge condition do not work ideally; **Appendix III.**

## 6.2. Comparison of the Segmentation methods

While running the segmentation process on the point cloud directly, the non-uniform structure and the varying point density limit the accuracy of the applied method. The results of the same show a direct relationship between the two. Since the scanner system is placed at a distance away from the top-left corner of the cargo container - the box objects on the left side are segmented better than the box objects

that are placed on the right-hand side (bottom-right in particular – which suffers the most due to high sparsity of points). A projection-based method that projects the points onto an image plane alleviates the problem of varying laser footprint. In the case of datasets 4 and 10, the point cloud segmentation performs better when compared to the image method. The two datasets contain objects that belong to different planes/rows and hence the segment growing is stopped when the candidate points are far apart.

The k-NN method is employed for the segment growing method for seed point selection. This employs a linear search solution and has a running time of $O(nd)$, where the number of points is $n$ and the dimensionality of the data is $d$. Thus, the computational time for the segmentation method to work directly on the point cloud data is higher. On the other hand, the watershed segmentation applied on the range image resulting from projecting the point cloud data onto an image plane takes much less time. This is because the method works on a reduced dimension of the data. However, the projection onto the image (section 5.1) and its re-projection for estimating the pose parameters are time complex.

### 6.3. Dimensions and Pose Estimates of the Segmented Objects

The 3d position of the objects is found by identifying the center of mass of each object; however, these positions are found relatively. They are with respect to the global center (0,0,0) of the dataset. When the coordinates of the scanning system are used, the positions with respect to the sensor coordinates are obtained.

The removal of overlap between the two adjacent box objects is removed by pruning the bounding boxes of the objects equally. This step does not necessarily adjust the object boundaries to exactly coincide with the actual boundaries. Thus, this can introduce a discrepancy between the computed and the real object dimensions. The objects' estimated dimensions and pose details are affected by the accuracy of segmentation results. In a case where an object is detected too much (such that it combines some portions from two actual objects and represents it as one), the computed 3d pose and the orientation would be less accurate.

# 7. CONCLUSION AND SCOPE FOR FUTURE WORK

## 7.1. Conclusion

On the problem of achieving maximal accuracy from one-shot point clouds, a projection-based point cloud segmentation method that exploits laser point footprints to form a range image, representing the 3d data in an organized format, is explored. The first objective of this research is to identify a suitable point cloud attribute that distinctly identifies the data points as belonging to the object surface and edges. The normal vectors of the points belonging to the surface of the box object have a higher magnitude. They are more aligned towards the global z-axis, while the normal vectors of those points belonging to the edges of these box objects are oriented away from the global z-axis. This helps the demarcation of the object surface and edge points. Following the selection of this attribute, the data is segmented by employing a direct point cloud segmentation method. The segment growing technique is used for the segmentation task and it identifies at least half the number of objects in the scene correctly. Applying a segmenting method on the point cloud directly produces results with a combination of over and under-segmentation. Applying post-processing steps to eliminate both problems is quite tedious while working on the point cloud data directly. The computational effort is higher due to high dense point data. Thus, an alternate projection-based technique is explored.

The problems caused by the varying point density are mitigated by projecting the point cloud data into a range image, increasing the footprint of laser points upon projection. The range image is an organized format with rectangular grids with pixel values representing the local normal vector oriented in the z-axis direction. The watershed segmentation method is used for segmenting the range image. This method works well for objects that have thin boundaries existing between them. Although the segmentation results are not perfect by employing the image-based method, the other option is to have an over-segmented result when compared with a combination of over and under-segmentation. The over-segmentation is removed by using statistical methods that fit most datasets, giving results with good accuracy. The final segmentation results are re-projected to the point cloud to extract the dimensions and pose details of each segmented object with six degrees of freedom. The computational time is high only during the projection onto the image plane and the re-projection of the segmentation labels back to the point cloud data. The rest of the steps are not time complex as they are all on a reduced dimensionality of the point cloud data.

The research is aimed at finding if the segmentation method implemented can enable the detection of objects from single-shot scans and subsequently extract the object parameters. For this research, the results of the watershed segmentation on the range image are considered to extract the object parameters further to access them in an industrial setting. As ground truth data is not available for all the datasets, a qualitative check was made visually and relatively between two adjacent objects. The accuracy of the segmentation results is assessed qualitatively by visual inspection and quantitatively assessed on one dataset by manually generating ground truth using an open-sourced toolbox. The proposed methodology can segment the 3d point cloud data and extract each objects' pose and dimensions with some limitations. The comparison drawn between the two formats of the data helps to understand the strengths and weaknesses pertaining to each of them.

The framework of this research can be made more efficient by exploring a tracking-based algorithm. Each sweep made by the scanning system can be used to update the segmentation result from the previous step. A more refined result can be obtained at each step of the process in this manner.

## 7.2. Research Questions: Answered

The research questions formulated in section 1.2.2 are answered in this sub-section –

1. What combination of attribute(s) is the most suitable to differentiate the foreground objects from the background?
   The most suitable attribute to distinguish objects that are touching each other is the point normal vector aligned along the z-axis direction.

2. What method can be used to distinguish every object?
   Upon analyzing the results of two segmentation techniques, a projection-based method is well suited for similarly shaped objects placed on a planar region. The projection onto an evenly spaced grid format alleviates the problems of processing the point cloud data with non-uniform point density.

3. How can the problem of varying point density be addressed?
   The problems caused by the varying point density are handled using downsampling and further by increasing the laser footprints during the point cloud projection to an image.

4. What are the total number of objects and their dimensions?
   The watershed segmentation method applied on the range image segments the data well. The segmentation results from the image give an estimate for the number of objects present in the scene; **Table 5-3**. For finer pose and geometry, the segmentation labels are re-projected to the point cloud data from where the 3-dimensional pose details and dimensions are extracted; **Table 5-4**, **Table 5-5**.

5. How are the objects of interest oriented in 3d space?
   From the orientation angles displayed in **Table 5-5**, the objects are indeed in neat piles. The robotic system can easily pick up the objects of interest without much angular tilts involved.

6. Does the segmentation work well to aid in recognizing the individual instances?
   Yes, the segmentation works ideally to find the individual objects present in the scene with segmentation accuracy ranging between 0.80 – 0.90; **Table 5-2**.

## 7.3. Scope for future work

The recommendations and scope for future work from this study are –

- Future research can use a tracking method that updates the results of segmentation from the previous step, as the unloading of the cargo container proceeds. In this manner, the results are updated and refined for every scan the system makes.
- Reference data can be captured from the scene using a TLS platform for result evaluation. Furthermore, using ground truth for object position and orientation can quantitatively evaluate the obtained pose details and dimensions.
- A deformable projection-based segmentation method, which allows for flexibility in the kernels used rather than fixed kernels during projection, can be studied (Thomas et al., 2019).
- Post-processing techniques on the direct point cloud segmentation method can be explored further. It would be interesting to combine results from different parameter values to introduce a certainty of segmentation

# LIST OF REFERENCES


Aldoma, A., Tombari, F., Prankl, J., Richtsfeld, A., Di Stefano, L., & Vincze, M. (2013). Multimodal cue integration through Hypotheses Verification for RGB-D object recognition and 6DOF pose estimation. *Proceedings - IEEE International Conference on Robotics and Automation*, 2104–2111. https://doi.org/10.1109/ICRA.2013.6630859

Arnold, E., Al-Jarrah, O. Y., Dianati, M., Fallah, S., Oxtoby, D., & Mouzakitis, A. (2019). A Survey on 3D Object Detection Methods for Autonomous Driving Applications. *IEEE Transactions on Intelligent Transportation Systems*, *20*(10), 3782–3795. https://doi.org/10.1109/TITS.2019.2892405

Baccar, M., Gee, L., Gonzalez, R. C., & Abidi, M. A. (1996). Segmentation of Range Images Via Data Fusion and Morphological Watersheds. *PATTERN RECOGNITION*, *29*. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.21.1177

Besl, P. J., & Jain, R. C. (1988). Segmentation Through Variable-Order Surface Fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *10*(2), 167–192. https://doi.org/10.1109/34.3881

Beucher, S. (1979). Use of watersheds in contour detection. *Proceedings of the International Workshop on Image Processing*. Retrieved from https://ci.nii.ac.jp/naid/10008961959

Bia, Z. M., & Wang, L. (2010, October 1). Advances in 3D data acquisition and processing for industrial applications. *Robotics and Computer-Integrated Manufacturing*, Vol. 26, pp. 403–413. https://doi.org/10.1016/j.rcim.2010.03.003

Bonini, M., Prenesti, D., Urru, A., & Echelmeyer, W. (2015). Towards the full automation of distribution centers. *2015 4th IEEE International Conference on Advanced Logistics and Transport, IEEE ICALT 2015*, 47–52. https://doi.org/10.1109/ICAdLT.2015.7136589

Chazette, P., Totems, J., Hespel, L., & Bailly, J. S. (2016). Principle and Physics of the LiDAR Measurement. In *Optical Remote Sensing of Land Surface: Techniques and Methods* (pp. 201–247). https://doi.org/10.1016/B978-1-78548-102-4.50005-3

Choi, C., Taguchi, Y., Tuzel, O., Liu, M. Y., & Ramalingam, S. (2012). Voting-based pose estimation for robotic assembly using a 3D sensor. *Proceedings - IEEE International Conference on Robotics and Automation*, 1724–1731. https://doi.org/10.1109/ICRA.2012.6225371

Czajewski, W., & Kołomyjec, K. (2017). 3D Object Detection and Recognition for Robotic Grasping Based on RGB-D Images and Global Features. *Foundations of Computing and Decision Sciences*, *42*(3), 219–237. https://doi.org/10.1515/fcds-2017-0011

Deschaud, J., & Goulette, F. (2010). A Fast and Accurate Plane Detection Algorithm for Large Noisy Point Clouds Using Filtered Normals and Voxel Growing. *Symposium A Quarterly Journal In Modern Foreign Literatures*, (May). Retrieved from http://campwww.informatik.tu-muenchen.de/3DPVT2010/data/media/e-proceeding/papers/paper111.pdf

Deza, M. M., & Deza, E. (2009). *Encyclopedia of Distances*. https://doi.org/10.1007/978-3-642-00234-2

Dieterle, T., Particke, F., Patino-Studencki, L., & Thielecke, J. (2017). Sensor data fusion of LIDAR with stereo RGB-D camera for object tracking. *Proceedings of IEEE Sensors*, *2017-December*, 1–3. https://doi.org/10.1109/ICSENS.2017.8234267

Djelouah, A., Franco, J.-S., Boyer, E., Le Clerc, F., & Perez, P. (2015). Sparse Multi-View Consistency for Object Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(9), 1890–1903. https://doi.org/10.1109/TPAMI.2014.2385704

Doliotis, P., McMurrough, C. D., Criswell, A., Middleton, M. B., & Rajan, S. T. (2016). A 3D perception-based robotic manipulation system for automated truck unloading. *IEEE International Conference on Automation Science and Engineering*, *2016-November*, 262–267. https://doi.org/10.1109/COASE.2016.7743416

Douillard, B., Underwood, J., Kuntz, N., Vlaskine, V., Quadros, A., Morton, P., & Frenkel, A. (2011). On the segmentation of 3D lidar point clouds. *Proceedings - IEEE International Conference on Robotics and Automation*, 2798–2805. https://doi.org/10.1109/ICRA.2011.5979818

Echelmeyer, W., Kirchheim, A., Lilienthal, A. L., Akbiyik, H., & Bonini, M. (2011). Performance Indicators for Robotics Systems in Logistics Applications. *IROS Workshop on Metrics and Methodologies for Autonomous Robot Teams in Logistics (MMART-LOG)*, (July 2014). Retrieved from http://kaspar.informatik.uni-freiburg.de/~mmartlog/pdfs/papers/echelmeyer_et_al_mmartlog11.pdf

Elich, C., Engelmann, F., Kontogianni, T., & Leibe, B. (2019). 3D-BEVIS: Bird's-Eye-View Instance Segmentation. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and*

*Lecture Notes in Bioinformatics)*, *11824 LNCS*, 48–61. https://doi.org/10.1007/978-3-030-33676-9_4

Elseberg, J., Borrmann, D., & Nüchter, A. (2011). Efficient processing of large 3D point clouds. *2011 23rd International Symposium on Information, Communication and Automation Technologies, ICAT 2011*. https://doi.org/10.1109/ICAT.2011.6102102

Fabbri, R., F Costa, L. DA, Torelli, J. C., Bruno, O. M., & Torelli, J. C. (2008). 2 2D Euclidean Distance Transform Algorithms: A Comparative Survey. *ACM Computing Surveys*, *40*(1). https://doi.org/10.1145/1322432.1322434

Fernandez-Diaz, J. C., Glennie, C. L., Carter, W. E., Shrestha, R. L., Sartori, M. P., Singhania, A., … Overstreet, B. T. (2014). Early results of simultaneous terrain and shallow water bathymetry mapping using a single-wavelength airborne LiDAR sensor. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *7*(2), 623–635. https://doi.org/10.1109/JSTARS.2013.2265255

Goh, T. Y., Basah, S. N., Yazid, H., Aziz Safar, M. J., & Ahmad Saad, F. S. (2018). Performance analysis of image thresholding: Otsu technique. *Measurement: Journal of the International Measurement Confederation*, *114*, 298–307. https://doi.org/10.1016/j.measurement.2017.09.052

Hernández, J., & Marcotegui, B. (2009). Point cloud segmentation towards urban ground modeling. *2009 Joint Urban Remote Sensing Event*. https://doi.org/10.1109/URS.2009.5137562

Hoffman, R., & Jain, A. K. (1978). Segmentation and Classification of Range Images. *October*, *75*(10), 736–743. Retrieved from https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=4767955

Ibrahim, Y., Nagy, B., & Benedek, C. (2019). Cnn-based watershed marker extraction for brick segmentation in masonry walls. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11662 LNCS*, 332–344. https://doi.org/10.1007/978-3-030-27202-9_30

Jakovljevic, Z., Puzovic, R., & Pajic, M. (2015). Recognition of Planar Segments in Point Cloud Based on Wavelet Transform. *IEEE Transactions on Industrial Informatics*, *11*(2), 342–352. https://doi.org/10.1109/TII.2015.2389195

Jamil, N., Sembok, T. M. T., & Bakar, Z. A. (2008). Noise removal and enhancement of binary images using morphological operations. *Proceedings - International Symposium on Information Technology 2008, ITSim*, *3*. https://doi.org/10.1109/ITSIM.2008.4631954

Johnson, B., & Xie, Z. (2011). Unsupervised image segmentation evaluation and refinement using a multi-scale approach. *ISPRS Journal of Photogrammetry and Remote Sensing*, *66*(4), 473–483. https://doi.org/10.1016/j.isprsjprs.2011.02.006

Johnson, G. S., Lee, J., Burns, C. A., & Mark, W. R. (2005). The irregular Z-buffer. *ACM Transactions on Graphics*, *24*(4), 1462–1482. https://doi.org/10.1145/1095878.1095889

Jolliffe, I. T., & Cadima, J. (2021). *Principal component analysis: a review and recent developments*. https://doi.org/10.1098/rsta.2015.0202

Kim, K., Kim, J., Kang, S., Kim, J., & Lee, J. (2012). Vision-based bin picking system for industrial robotics applications. *2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence, URAI 2012*, 515–516. https://doi.org/10.1109/URAI.2012.6463057

Kirchheim, A., Burwinkel, M., & Echelmeyer, W. (2008). Automatic unloading of heavy sacks from containers. *Proceedings of the IEEE International Conference on Automation and Logistics, ICAL 2008*, 946–951. https://doi.org/10.1109/ICAL.2008.4636286

Křemen, T., Koska, B., & Pospíšil, J. (2006). Verification of laser scanning systems quality. *XXIII FIG International Congress*, 16 (on CD-ROM). Retrieved from https://www.fig.net/resources/proceedings/fig_proceedings/fig2006/papers/ts24/ts24_04_kremen_etal_0452.pdf

Kuo, H. Y., Su, H. R., Lai, S. H., & Wu, C. C. (2014). 3D object detection and pose estimation from depth image for robotic bin picking. *IEEE International Conference on Automation Science and Engineering*, *2014-January*, 1264–1269. https://doi.org/10.1109/CoASE.2014.6899489

Li, H., Zhou, X., Chen, Y., Zhang, Q., Zhao, D., & Qian, D. (2019). Comparison of 3D object detection based on LiDAR point cloud. *Proceedings of 2019 IEEE 8th Data Driven Control and Learning Systems Conference, DDCLS 2019*, 678–685. https://doi.org/10.1109/DDCLS.2019.8908931

Li, W., Guo, Q., Jakubowski, M. K., & Kelly, M. (2012). A new method for segmenting individual trees from the lidar point cloud. *Photogrammetric Engineering and Remote Sensing*, *78*(1), 75–84. https://doi.org/10.14358/PERS.78.1.75

Liu, D., Arai, S., Miao, J., Kinugawa, J., Wang, Z., & Kosuge, K. (2018). Point Pair Feature-Based Pose Estimation with Multiple Edge Appearance Models (PPF-MEAM) for Robotic Bin Picking. *Sensors*, *18*(8), 2719. https://doi.org/10.3390/s18082719

Liu, Y., Bian, L., Meng, Y., Wang, H., Zhang, S., Yang, Y., … Wang, B. (2012). Discrepancy measures for selecting optimal combination of parameter values in object-based image analysis. *ISPRS Journal of Photogrammetry and Remote Sensing*, *68*(1), 144–156. https://doi.org/10.1016/j.isprsjprs.2012.01.007

Meyer, F., & Beucher, S. (1990). Morphological segmentation. *Journal of Visual Communication and Image Representation*, *1*(1), 21–46. https://doi.org/10.1016/1047-3203(90)90014-M

Nguyen, A., & Le, B. (2013). 3D point cloud segmentation: A survey. *IEEE Conference on Robotics, Automation and Mechatronics, RAM - Proceedings*, 225–230. https://doi.org/10.1109/RAM.2013.6758588

Nieto, M., Senderos, O., & Otaegui, O. (2021). Boosting AI applications: Labeling format for complex datasets. *SoftwareX*, *13*, 100653. https://doi.org/10.1016/j.softx.2020.100653

Ning, X., Zhang, X., Wang, Y., & Jaeger, M. (2009). Segmentation of architecture shape information from 3D point cloud. *Proceedings - VRCAI 2009: 8th International Conference on Virtual Reality Continuum and Its Applications in Industry*, 127–132. https://doi.org/10.1145/1670252.1670280

Parvati, K., Prakasa Rao, B. S., & Mariya Das, M. (2008). Image segmentation using gray-scale morphology and marker-controlled watershed transformation. *Discrete Dynamics in Nature and Society*, *2008*. https://doi.org/10.1155/2008/384346

Rabbani, T., van den Heuvel, F. a, & Vosselman, G. (2006). (impo)(Fashuai exper+Sudan recom)Segmentation of point clouds using smoothness constraint. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences - Commission V Symposium "Image Engineering and Vision Metrology,"* *36*(5), 248–253. Retrieved from http://www.isprs.org/proceedings/XXXVI/part5/paper/RABB_639.pdf

Rudorfer, M. (2016). *Evaluation of Point Pair Feature Matching for Object Recognition and Pose Estimation in 3D Scenes*. Retrieved from https://www.semanticscholar.org/paper/Evaluation-of-Point-Pair-Feature-Matching-for-and-Rudorfer/6f470b5e54c30c3fbfc4b5afa31939f2f4b93a52

Saleh, Z., Zhang, K., Calvo-Zaragoza, J., Vigliensoni, G., & Fujinaga, I. (2018). Pixel.js: Web-Based Pixel Classification Correction Platform for Ground Truth Creation. *Proceedings of the International Conference on Document Analysis and Recognition, ICDAR*, *2*, 39–40. https://doi.org/10.1109/ICDAR.2017.267

Shen, C. H., Huang, S. S., Fu, H., & Hu, S. M. (2011). Adaptive Partitioning of Urban Facades. *ACM Transactions on Graphics*, *30*(6), 1–10. https://doi.org/10.1145/2070781.2024218

Sithole, G. (2008). *DETECTION OF BRICKS IN A MASONRY WALL*. Retrieved from https://www.isprs.org/proceedings/XXXVII/congress/5_pdf/99.pdf

Slabaugh, G. G. (1999). Computing Euler angles from a rotation matrix. *Denoted as TRTA Implementation from Httpwww Starfireresearch Comservicesjava3dsamplecodeFlorinE Ulers Html*, *6*(2000), 1–6. Retrieved from http://gregslabaugh.name/publications/euler.pdf

Song, J.-H., Han, S.-H., Yu, K., & Kim, Y.-I. (2002). *ASSESSING THE POSSIBILITY OF LAND-COVER CLASSIFICATION USING LIDAR INTENSITY DATA*. Retrieved from https://www.isprs.org/PROCEEDINGS/XXXIV/part3/papers/paper128.pdf

Soudarissanane, S., Lindenbergh, R., Menenti, M., & Teunissen, P. (2011). Scanning geometry: Influencing factor on the quality of terrestrial laser scanning points. *ISPRS Journal of Photogrammetry and Remote Sensing*, *66*(4), 389–399. https://doi.org/10.1016/j.isprsjprs.2011.01.005

Soulard, C. E., & Bogle, R. (2011). Using terrestrial light detection and ranging (lidar) technology for land-surface analysis in the Southwest. In *Fact Sheet*. https://doi.org/10.3133/FS20113017

Stoyanov, T., Vaskevicius, N., Mueller, C. A., Fromm, T., Krug, R., Tincani, V., … Echelmeyer, W. (2016). No More Heavy Lifting: Robotic Solutions to the Container Unloading Problem. *IEEE Robotics and Automation Magazine*, *23*(4), 94–106. https://doi.org/10.1109/MRA.2016.2535098

Thomas, H., Qi, C. R., Deschaud, J.-E., Marcotegui, B., Goulette, F., & Guibas, L. J. (2019). KPConv: Flexible and Deformable Convolution for Point Clouds. *Proceedings of the IEEE International Conference on Computer Vision*, *2019-October*, 6410–6419. Retrieved from http://arxiv.org/abs/1904.08889

Tóvári, D., & Pfeifer, N. (2005). Segmentation based robust interpolation - A newapproach to laser data filtering. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *36*, 79–84. Retrieved from https://www.isprs.org/proceedings/xxxvi/3-W19/papers/079.pdf

Vaskevicius, N., Pathak, K., & Birk, A. (2017). Recognition and Localization of Sacks for Autonomous Container Unloading by Fitting Superquadrics in Noisy, Partial Views from a Low-cost RGBD Sensor. *Journal of Intelligent and Robotic Systems: Theory and Applications*, *88*(1), 57–71. https://doi.org/10.1007/s10846-017-0540-7

Vo, A. V., Truong-Hong, L., Laefer, D. F., & Bertolotto, M. (2015). Octree-based region growing for

point cloud segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing*, *104*, 88–100. https://doi.org/10.1016/j.isprsjprs.2015.01.011

Vosselman, G., & Maas, H. G. (2010). *Airborne and terrestrial laser scanning*. Retrieved from https://research.utwente.nl/en/publications/airborne-and-terrestrial-laser-scanning-2

Vosselman, George. (2010). *POINT CLOUD SEGMENTATION FOR URBAN SCENE CLASSIFICATION*. https://doi.org/10.5194/isprsarchives-XL-7-W2-257-2013

Woo, H., Kang, E., Wang, S., & Lee, K. H. (2002). A new segmentation method for point cloud data. *International Journal of Machine Tools and Manufacture*, *42*(2), 167–178. https://doi.org/10.1016/S0890-6955(01)00120-1

Xiao, D., Shan, F., Li, Z., Le, B. T., Liu, X., & Li, X. (2019). A target detection model based on improved tiny-yolov3 under the environment of mining truck. *IEEE Access*, *7*, 123757–123764. https://doi.org/10.1109/ACCESS.2019.2928603

Yang, G., Mentasti, S., Bersani, M., Wang, Y., Braghin, F., & Cheli, F. (2020). LiDAR point-cloud processing based on projection methods: A comparison. *2020 AEIT International Conference of Electrical and Electronic Technologies for Automotive, AEIT AUTOMOTIVE 2020*. https://doi.org/10.23919/aeitautomotive50086.2020.9307387

Yang, Jinrong, Wu, S., Gou, L., Yu, H., Lin, C., Wang, J., … Li, X. (2021). *SCD: A Stacked Carton Dataset for Detection and Segmentation*. Retrieved from http://arxiv.org/abs/2102.12808

Yang, Juntao, Kang, Z., Cheng, S., Yang, Z., & Akwensi, P. H. (2020). An Individual Tree Segmentation Method Based on Watershed Algorithm and Three-Dimensional Spatial Distribution Analysis from Airborne LiDAR Point Clouds. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *13*, 1055–1067. https://doi.org/10.1109/JSTARS.2020.2979369

Ye, M., Wang, X., Yang, R., Ren, L., & Pollefeys, M. (2011). Accurate 3D pose estimation from a single depth image. *Proceedings of the IEEE International Conference on Computer Vision*, 731–738. https://doi.org/10.1109/ICCV.2011.6126310

Zhang, Xiaodong, Jia, F., Luo, S., Liu, G., & Hu, Q. (2014). A marker-based watershed method for X-ray image segmentation. *Computer Methods and Programs in Biomedicine*, *113*(3), 894–903. https://doi.org/10.1016/j.cmpb.2013.12.025

Zhang, Xueliang, Feng, X., Xiao, P., He, G., & Zhu, L. (2015). Segmentation quality evaluation using region-based precision and recall measures for remote sensing images. *ISPRS Journal of Photogrammetry and Remote Sensing*, *102*, 73–84. https://doi.org/10.1016/j.isprsjprs.2015.01.009

# APPENDIX I

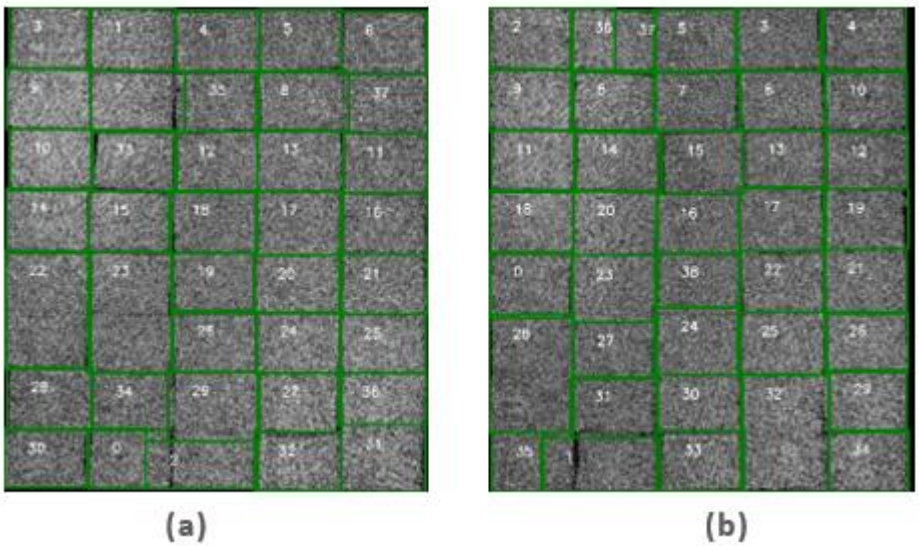



Figure 0-7-1 (a) Dataset 2 and (b) Dataset 5 with varying results upon using the same parameter values

# APPENDIX II

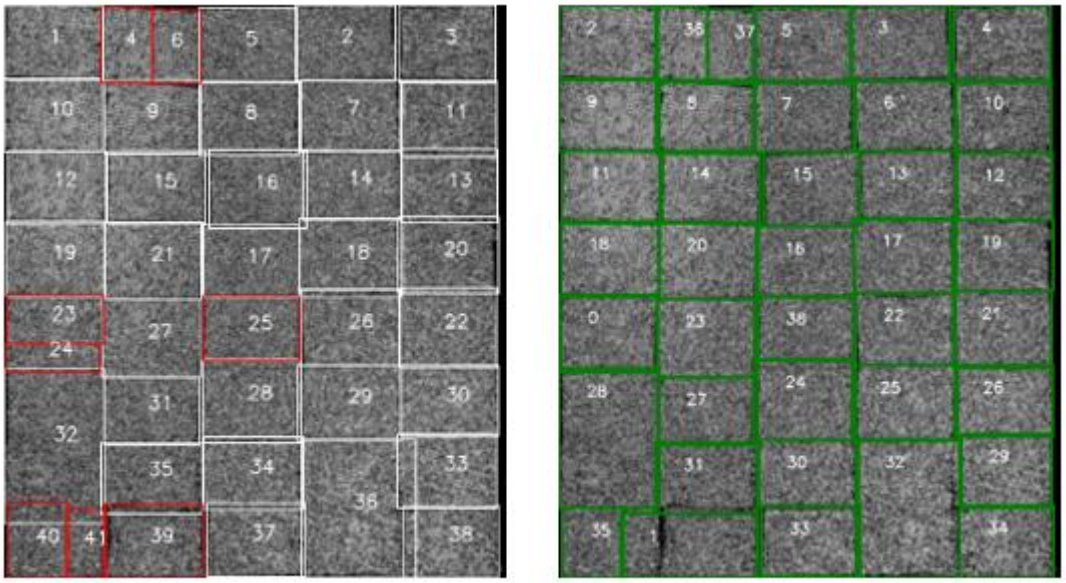



Figure 0-7-2 Segments labeled 40 and 41 are the ideal candidates for a merging (a); Segments 41 and 39 (a) are merged resulting in segment 1 (b)

# APPENDIX III

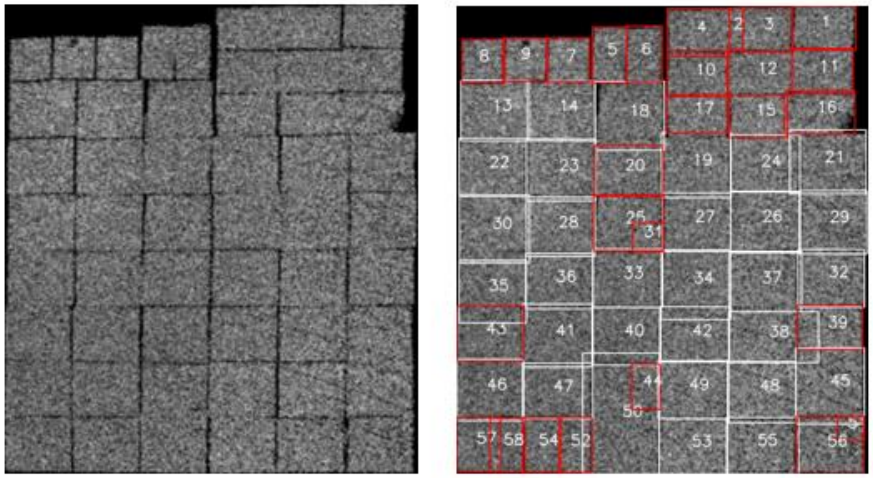



Figure 0-7-3 Dataset 7 (a) with at least four different sizes of objects; the threshold set to separate the smaller boxes from the ideal ones fails in such a case (b), smaller segments identified in red