



EÖTVÖS LORÁND UNIVERSITY

FACULTY OF INFORMATICS

DEPT. OF ALGORITHMS AND THEIR APPLICATIONS

Box Dimension Estimation Through Sensor Fusion for a Parcel Unpacking Robot

Author:

Hanif Mustafa Budiyanto

Computer Science for Autonomous System

Supervisors:

Dr. Hajder Levente

Associate Professor

Kovács Bandó

Engineer

Budapest, 2023

Contents

1	Abstract	3
2	Introduction	4
2.1	Contributions	5
3	Related Works	6
3.1	Sensors	6
3.1.1	Depth Camera	6
3.1.2	LiDAR	7
3.2	3D reconstruction	8
3.3	6 DoF Extrinsic Calibration	9
4	Proposed Methods	11
4.1	Experiment Setup	11
4.2	Camera-Robot Calibration	14
4.3	2D LiDAR-Camera Point Cloud Fusion	16
4.4	Box Dimension Estimation	21
4.4.1	Depth Camera-based Measurements	21
4.4.2	2D LiDAR-based Measurements	24
5	Results and Analysis	27
5.1	2D LiDAR-Camera Fusion	27
5.2	Multiple Depth-camera Measurement	29
5.2.1	Multiple Plane Detection	29
5.2.2	Key Points Extraction	30
5.2.3	Dimension Measurement	32
5.3	Multiple 2D LiDAR Measurement	34
5.3.1	Box Side Line Segmentation	34
5.3.2	Dimension Measurement	35

5.4	Combining Measurements	37
5.5	Box Unpacking	39
6	Conclusion	40
6.1	Future Work	41
	Acknowledgements	42
	Bibliography	43
	List of Figures	48
	List of Tables	50
	List of Algorithms	51

Chapter 1

Abstract

The manual unpacking of parcel boxes, known for its repetitive and labor-intensive nature, could greatly benefit from the introduction of a fully automated unpacking robot. In this paper, we evaluate and compare two distinct methods to obtain the key points and volumetric dimensions of a rectangular box, which are crucial for the accurate operation of the unpacking robot. The first method relies on two 2D LiDAR installed both vertically and horizontally to capture the essential sides of the box. The second method leverages the structured-light stereo depth camera which takes depth images from multiple perspectives to get the complete point cloud model of the rectangular container. Furthermore, we combine both approaches to acquire more accurate measurements. Both of these methods benefit from the RANSAC-based algorithm to segment the side of the box. The unpacking process is subsequently performed by moving the robotic arm equipped with a sharp-edge tool to cut open the box. Additionally, we propose our method to fuse the 2D LiDAR points with the depth camera point cloud by using a custom calibration target.

Chapter 2

Introduction

Automatic box unpacking robots present a unique challenge in industrial automation, where the precise handling of rectangular containers is needed to optimize logistics and manufacturing processes. A system that relies only on a single sensor, such as a robotic arm with limited sensing capabilities, is more likely to encounter difficulties in acquiring a complete model of various boxes with differing sizes and imperfections. By combining data from different sensors, such as multiple cameras and LiDAR, sensor fusion systems extend the spatial coverage, and hence, overall understanding of the environment.

To fuse the information obtained from multiple sensors, sensor calibration needs to be carried out to estimate the intrinsic (e.g. focal length in cameras) and extrinsic (i.e. position and orientation) parameters of the sensors. This process is essential to transform all the sensor information to be expressed in the common frame of reference. The procedure for sensor calibration depends on the type of sensor, however, some of the generalized steps include the use of an identifiable calibration object or setting up the sensor and other equipment in a controlled environment.

While various studies have explored the modeling of the box with sensor fusion [1, 2], the development of automated systems specifically focused on the unpacking process is an area that has yet to receive extensive attention.

2.1 Contributions

This paper will discuss our three main contributions. First, we introduce a custom calibration object based on three mutually orthogonal colored planes which function as a pose estimation and camera-to-lidar calibration object. Next, we present our approach to fusing the LiDAR sensor and point cloud information from the depth camera using the aforementioned calibration target. Third, we describe the method to extract key points and dimension information of rectangular boxes by using the fused data. Additionally, the measurement results will be tested by performing box cutting using a robotic arm equipped with a sharp-edge tool. The demo of the box-cutting is available at this link: [video link](#).

Based on our evaluation of measurement and real-world testing, both the calibration procedure and unpacking process show promising results despite the use of such accessible sensors. Therefore, we anticipate our study to set the groundwork for future research to improve and refine the ongoing effort.

Chapter 3

Related Works

3.1 Sensors

3.1.1 Depth Camera

Depth cameras provide 3D or distance information from the camera to an object in addition to the color or intensity information. The common underlying technology behind the depth camera separates into structured light and Time of Flight (ToF) cameras. Depth cameras usually consist of two identical cameras that capture images from slightly different perspectives, which allows the camera to measure distances on the scene.

Traditional stereo cameras operate by comparing two images from different perspectives to perform triangulation to acquire the 3D location of its corresponding pixel. The challenge with this approach is to find the corresponding points between one image and the other. While researchers have conducted numerous studies on feature matching to identify pixel correspondences [3, 4, 5], this approach tends to fall short especially when large parts of the image lack distinctive features.

One approach to tackle this problem is to actively illuminate structured light onto the scene to be captured by the camera. The earlier version of the structured light depth camera uses a stripe pattern projected onto the scene to calculate the 3D location of every pixel on the image [6]. With this approach, only one camera is needed to acquire the 3D information of the scene, provided the illumination plane of the light is known.



Figure 3.1: Depth Camera D435i [7]

- . Camera components from left to right: first infrared camera, structured light projector, second infrared camera, RGB camera.

In the more modern depth cameras, such as shown in Figure 3.1, the structured light might not be projected as a striped pattern. Instead, a points pattern is projected where point correspondences can easily be acquired from the two infrared cameras. This scheme enables the camera to function with and without structured light [8]. In an environment where light is dim, an infrared structured light camera is able to benefit from the clear pattern projected onto the scene. However, in the outdoor environment when the infrared light is less effective, the system can still estimate the depth by performing features matching techniques.

3.1.2 LiDAR

Light detection and ranging (LiDAR) is a remote sensing technique to measure the distance from a sensor to a point in 3D space by using light. The two common techniques for LiDAR to measure the distance are ToF (Time of flight) and FMCW (Frequency Modulated Continuous Wave) with the former dominating most of the LiDAR types currently in mass production.

Time of Flight LiDAR works by measuring the time taken by the transmitted laser to travel from the transmitter until it is received back to the receiver from the object reflection. FMCW LiDAR takes a different approach by measuring the phase difference between the emitted and backscattered signals to determine its distance from an object.

LiDAR can also be categorized based on the spatial information captured by the sensor. For example, 2D LiDAR records distances in a single 2D plane. Hence, producing a 2D map representation of the environment. 3D LiDAR, on the other hand, captures distances in multiple planes. It produced a 3D structure of points commonly referred to as point clouds. In addition to the range measurements, LiDAR might

produce reflectance information for each point. Examples of laser scan recorded by 2D and 3D LiDAR is depicted in Figure 3.2

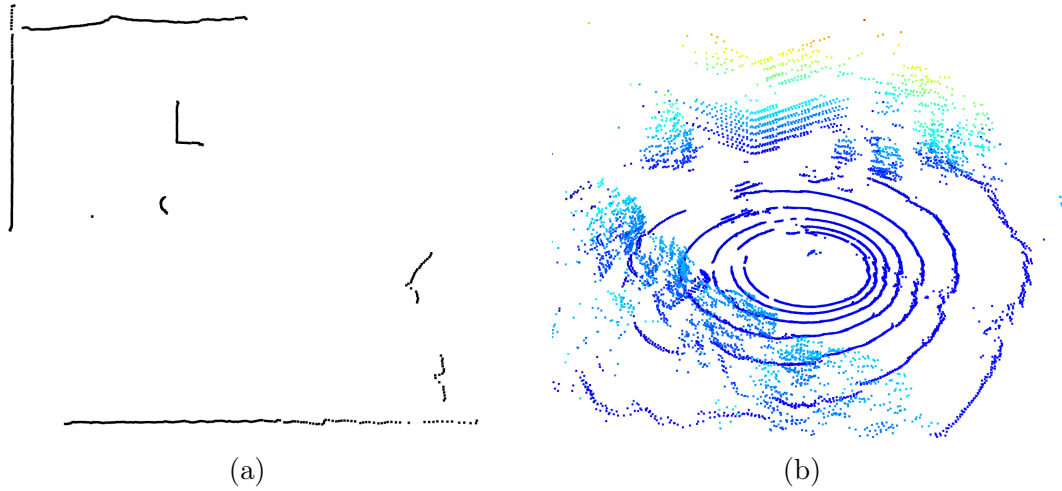


Figure 3.2: An example of (a) 2D LiDAR points and (b) 3D LiDAR point cloud.

Through the advancement of manufacturing technologies, LiDAR has become a more accessible and popular choice in numerous surveying applications. Some of the applications in 3D LiDAR can be found in environmental mapping [9], vehicle detection [10, 11], and building reconstruction [12, 13].

3.2 3D reconstruction

One of the common approaches to constructing a 3D model from images is the structure from motion (SfM) [14]. This method deals with a stationary scene in which the objective is to recover its 3D structure from a set of projective measurements. The SfM method generally involves three main steps. First, feature extraction and matching from multiple 2D images. Second, motion estimation for the camera. Third, 3D reconstruction by using the estimated features and motion; this final step is usually performed by minimizing the reprojection error. Although this method has shown great results, it suffers from the common problem when the scene contains low textured, specular and reflective regions which result in incomplete 3D reconstruction. The recent deep learning-based 3D reconstruction techniques such as MVSNet [15] tried to tackle this issue. MVSNet introduces an end-to-end deep learning architecture approach for depth map inference by taking a reference image and multiple source images as input.

The ability to reconstruct 3D structures in real time is generally more crucial in robotics. Its importance is even more pronounced in mobile robots since their motion is usually affected by the newest information in the environment. Reconstructing 3D structures in a mobile robot usually involves tracking its motion while traversing the environment. The common approach to achieve this is to mount the camera on the mobile robot while tracking its changes in location and orientation. This tracking process is usually referred to as visual odometry [16]. SLAM (Simultaneous Localization and Mapping)[17, 18, 19], for example, is the superset of visual odometry which not only localizes its position but also maps the environment around it. Another approach to tracking the motion of robots is to rely on the hardware odometry such as encoder wheels and robotic arm joints odometry system [20].

3.3 6 DoF Extrinsic Calibration

Sensor calibration in machine vision can be divided into two: intrinsic and extrinsic calibration. Intrinsic calibration is the process of determining the internal parameters (i.e. focal length, principal point, and lens distortion coefficients) of the camera while extrinsic calibration deals with the estimation of the position and orientation of the camera with respect to the world or other entities. In robotics, this process is an essential requirement to fuse multiple sensor data (e.g. camera, LiDAR, odometer).

Camera and LiDAR are the two sensors that have been broadly used in the fields of robotics, autonomous vehicles, and industrial automation. Fusing the information from these two sensors is currently an open research and there have been many contributions to this topic. Determining the extrinsic calibration between the camera and LiDAR can be performed by directly finding the feature correspondences between the two sensors. This approach, however, assumes that the point correspondences between the camera and LiDAR can be known. Since LiDAR illumination is not visible in a normal RGB camera, this method usually relies on infrared cameras.

Rather than relying on point correspondence, scientists have conducted research to find the extrinsic parameters between the camera and LiDAR by using certain constrain derived from lines and depth edges. Some of the 3D features that are commonly used are points at corners edges [21, 22], and lines [23, 24] in 3D. In the

points-based feature method, extrinsic calibration is performed by minimizing the distance between these points' features either in 3D or 2D images (after projection).

Due to the limited information 2D LiDAR provides, acquiring the its extrinsic calibration information with the camera is generally more challenging than 3D LiDAR. The 2D LiDAR provides a single laser measurement which result in 2D plane scan. Compared to 3D LiDAR, the limited data from 2D LiDAR make it harder to find feature correspondences between the two sensor.

Chapter 4

Proposed Methods

In this chapter, we begin by examining the experiment setups including the sensors and robotic arm system employed for the box unpacking (i.e. cutting). Additionally, the software architecture developed in each computer will also be discussed. Next, we will describe the camera-robot calibration method using our custom calibration target which is useful for reconstructing the 3D point cloud from multiple-depth camera information. Subsequently, our method to fuse the 2D LiDAR points into the camera's point cloud will be discussed. Finally, we will focus on the two methods to estimate box corners and dimensions: one relying on multiple cameras and the other on two 2D LiDAR.

4.1 Experiment Setup

The architecture diagram involving the sensors and other hardware is illustrated in Figure 4.1. The software implemented in each hardware can also be seen to better understand their roles. Each computer runs the Robot Operating System (ROS) framework on top of a Linux-based operating system for exchanging information through messages and services (e.g. sensor data and commands). In ROS, each running process called node can exchange messages with the other nodes that are in the same or different machine (in the same network). In this project, this feature is utilized to exchange sensor information including camera images and 2D LiDAR scans. Moreover, a trigger request can also be sent between nodes, for instance, to control the robot's movement.

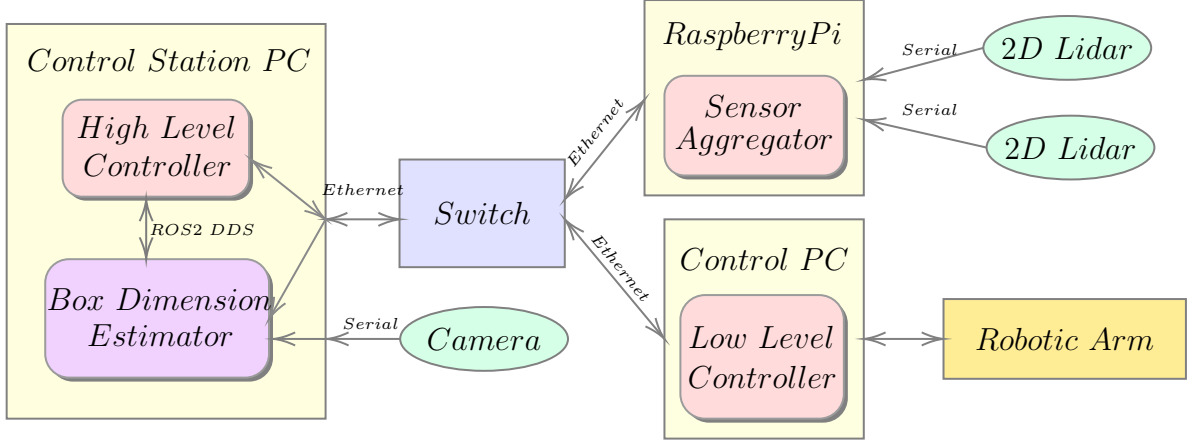


Figure 4.1: Architecture diagram of the automatic box unpacking system.

The RaspberryPi task is to gather the data from two 2D LiDARs where the measured scans are then broadcasted to the control station PC. The reason for this relay is more of a practical purpose since we avoid many sensors connecting to the control station which is a significant distance from the robot operation space. For the 2D LiDAR, we use two RPLidar A1 that are installed vertically and horizontally with respect to the ground (see Figure 4.2). These installations ensure that the vertical 2D LiDAR is able to capture the top and the side of the box that is placed on the conveyor. Similarly, the horizontal 2D LiDAR should be able to capture two sides of the box: the one facing the robotic arm and the other facing the 2D LiDAR itself.

In this experiment, we use an Intel real sense D435 depth camera which connects directly to the control station PC. Note that the camera is mounted onto the robotic arm and hence moves around with it. With this setup, the robotic arm can be

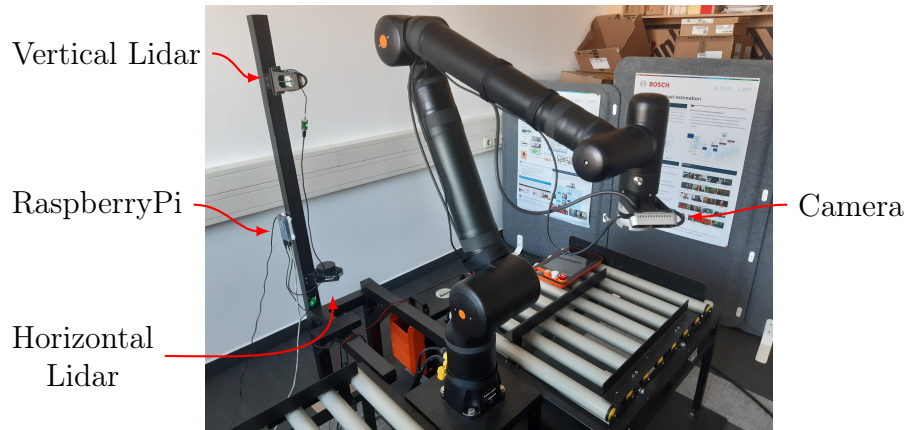


Figure 4.2: Hardware setup for the experiment.

moved around the box to gather depth maps from multiple angles in which later the point cloud information can be merged. The robotic arm is equipped with a highly accurate odometry system in its joints, hence camera pose can be acquired during the operation which is essential information to fuse the point cloud.

The control station PC retrieves the data from the 2D LiDAR and camera sensor through ROS. Then, a running node in the control station takes the acquired sensor data to estimate box corner locations and dimensions. The estimated box model is then passed to the high-level controller node before it sends a command to the robotic arm to perform the box cutting. The robotic arm also has its own computer which manages the odometry data acquisition and control for the robotic arm.

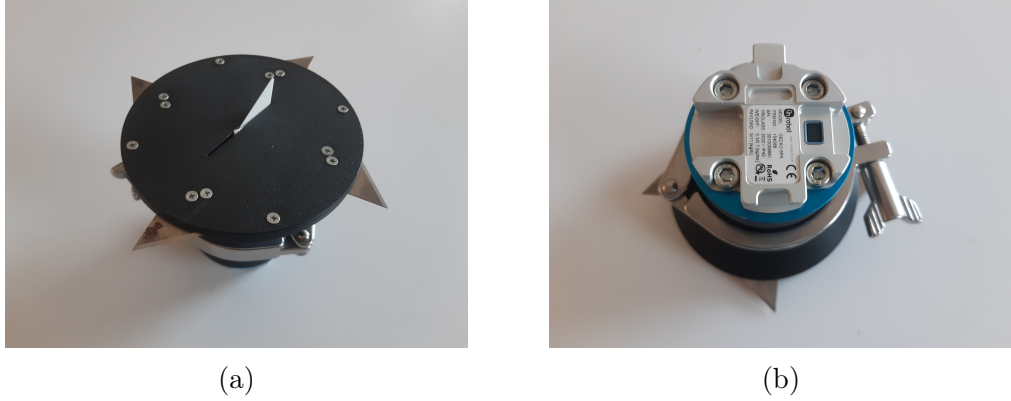


Figure 4.3: (a) The sharp edge tool from the bottom view (b) the sharp edge tool from the top view (mounting section to the robotic arm).

The experiment is performed by placing the box on the conveyor. The robotic arm is then moved to capture the four box corners from a predetermined distance. Since the object (i.e. box) does not move, this scheme emulates the use of four depth cameras from multiple views. Once the data is acquired, the box dimension estimator algorithm is run to calculate the box dimension and corner location which then be used to move the robotic arm to cut open the box. The sharp edge tool mounted at the robotic arm can be seen in Figure 4.3. Meanwhile, the box unpacking using the 2D LiDAR data shares a similar procedure except the box measurement is performed by using 2D LiDAR instead of a depth camera.

4.2 Camera-Robot Calibration

The final objective of the calibration process is to acquire the 6-degree of Freedom (DoF) transformation between the camera and the base of the robotic arm. Since the base of the robotic arm is stationary, we can refer to this coordinate frame as the world coordinate frame. To perform this calibration, we introduce a colored plane calibration object in which the transformation can be obtained when detected by the depth camera. This device can later be utilized to obtain the transformation between the camera and the world given the calibration object is placed in a known location and orientation in the world coordinate. The calibration object can be seen in Figure 4.4.

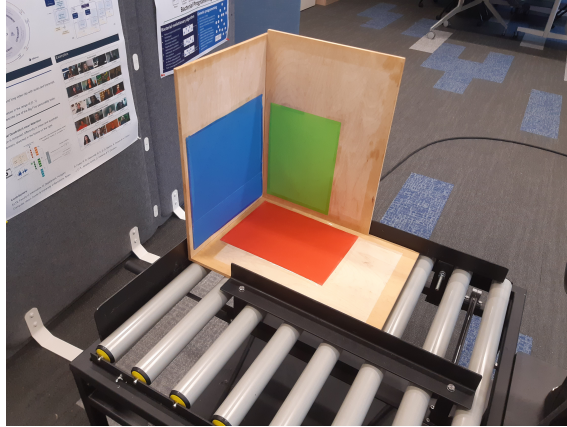


Figure 4.4: The calibration target placed on the conveyor where the box will be unpacked.

The first step in estimating the pose of the calibration box is to detect the planes of the object. The robust fitting method can be applied to deal with such a problem, especially when the data contain outliers. An early example of this algorithm is RANSAC [25] which works iteratively until a sufficiently good model is found based on the number of inliers. This method is also extended to find multiple instances (e.g. finding multiple planes) known as sequential RANSAC [26, 27]. There are numerous other robust fitting methods that deal with multiple instances that can be applied for multiple plane detection [28, 29, 30, 31, 32]. In this study, we applied sequential RANSAC to detect multiple planes since it is conceptually simple and straightforward to implement. Once the planes are found, the three planes can be represented as Equation 4.1.

$$\begin{aligned}
a_1x + b_1y + c_1z + d_1 &= 0 \\
a_2x + b_2y + c_2z + d_2 &= 0 \\
a_3x + b_3y + c_3z + d_3 &= 0
\end{aligned} \tag{4.1}$$

The location of the calibration object can be calculated by finding the intersection point of the three planes. This intersection point can simply be found by solving the linear system of equations of the three detected planes as shown in equation 4.2.

$$\begin{aligned}
AX &= B \\
\begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} &= - \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}
\end{aligned} \tag{4.2}$$

To estimate the rotation of the calibration object, we first need to calculate the intersection vector between every combination of the three detected planes. Note that since the plane is colored, we can assign an axis to a certain intersection vector. For example, the x-axis can be defined along the intersection between the red and green planes. The y-axis and z-axis can then be chosen accordingly.

For example, given the normal vector of the planes $\mathbf{N}_1 = (a_1, b_1, c_1)$ and $\mathbf{N}_2 = (a_2, b_2, c_2)$, the intersection vector between the planes can be calculated by using $\mathbf{V} = \mathbf{N}_1 \times \mathbf{N}_2$. The calculated vector $\mathbf{V}$ is orthogonal to both normal vectors, therefore it lies along the line of intersection.

Performing this operation to every intersection of the three detected planes yields the three vectors that represent the coordinate system of the calibration target. It is important to take care that the calculated vectors might not be orthogonal to one another. Hence, it is essential to orthogonalize the vectors. One common approach for orthogonalization is the Gram-Schmidt process. Equation 4.3 shows the alternative form of the Gram-Schmidt process which combines orthogonalization and normalization. Given a linearly independent vector $\mathbf{x}_1, \mathbf{x}_2$, and $\mathbf{x}_3$ the orthonormal vector $\mathbf{w}_1, \mathbf{w}_2$, and $\mathbf{w}_3$ can be calculated.

$$\begin{aligned}
\mathbf{v}_1 &= \mathbf{x}_1, \quad \mathbf{w}_1 = \frac{\mathbf{v}_1}{\|\mathbf{v}_1\|} \\
\mathbf{v}_2 &= \mathbf{x}_2 - \langle \mathbf{x}_2, \mathbf{w}_1 \rangle \mathbf{w}_1, \quad \mathbf{w}_2 = \frac{\mathbf{v}_2}{\|\mathbf{v}_2\|}, \\
\mathbf{v}_3 &= \mathbf{x}_3 - \langle \mathbf{x}_3, \mathbf{w}_1 \rangle \mathbf{w}_1 - \langle \mathbf{x}_3, \mathbf{w}_2 \rangle \mathbf{w}_2, \quad \mathbf{w}_3 = \frac{\mathbf{v}_3}{\|\mathbf{v}_3\|},
\end{aligned} \tag{4.3}$$

Following the orthogonalization and normalization, the three vectors can be arranged in a row-wise fashion as Equation 4.4 to obtain the rotation matrix.

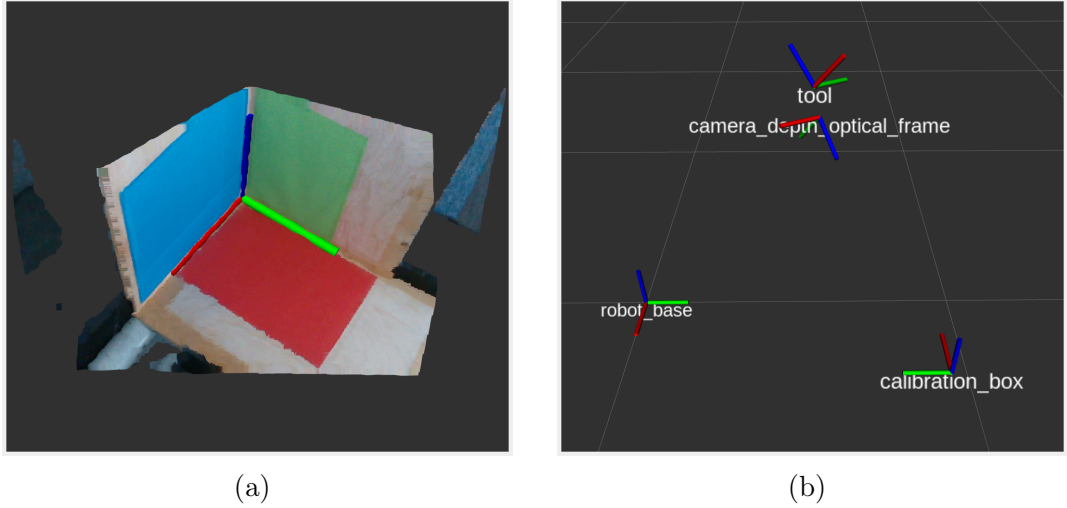


Figure 4.5: (a) The estimated pose of the calibration object with respect to the depth camera (b) The visualization of the rigid body configuration after the calibration box is estimated.

$$R = \begin{bmatrix} \mathbf{w}_{1x} & \mathbf{w}_{1y} & \mathbf{w}_{1z} \\ \mathbf{w}_{2x} & \mathbf{w}_{2y} & \mathbf{w}_{2z} \\ \mathbf{w}_{3x} & \mathbf{w}_{3y} & \mathbf{w}_{3z} \end{bmatrix} \quad (4.4)$$

With the rotation and translation information of the calibration target acquired, the coordinate system of the calibration object can be visualized as shown in Figure 4.5.

Thus far, we are able to estimate the calibration target's pose with respect to the camera. Since we already know the transformation between the calibration device and the world, the camera pose in the world coordinate can be estimated.

This calibration scheme assumes the pose information of the calibration target in the world coordinate is known. However, this information might not be known in some other instances. In this example, we benefit from the robotic arm odometry system to measure the location of the calibration target by aligning the robotic arm to the target location to gain its pose. Since the transformation between the robotic arm and the world is also known through the robot odometry, the location of the calibration box in world coordinates can be retrieved.

4.3 2D LiDAR-Camera Point Cloud Fusion

The laser-based method is known to yield a more accurate and longer-range 3D measurement compared to an active stereo camera [33]. However, LiDAR usually

cannot obtain a denser depth map compared to the depth camera. Therefore, incorporating a LiDAR into the depth camera system might help in measuring the critical space at the scene where the depth camera fails to reconstruct with minimal noise.

In this section, the method to fuse the 2D LiDAR data with the depth camera point cloud will be discussed. To merge the data points from the 2D LiDAR with the depth camera point cloud, we estimate the illumination plane of the 2D LiDAR on the scene. Since the depth camera is equipped with infrared cameras, the illumination line of the 2D LiDAR is visible on the captured image.

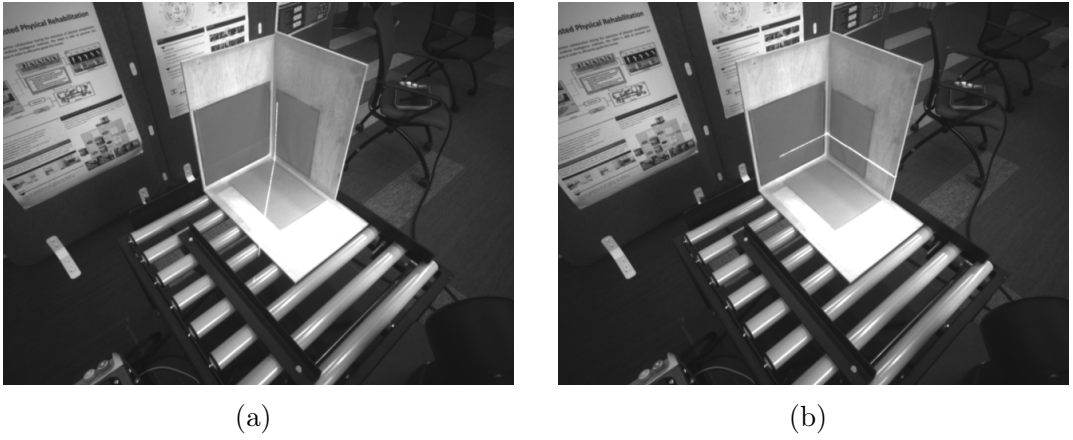


Figure 4.6: (a) Vertical and (b) horizontal 2D LiDAR illumination captured by infrared camera.

To model the 2D LiDAR illumination plane, a calibration object is placed on the scene where it can be captured by the 2D LiDAR and the depth camera as well. Figure 4.6 shows the illumination line of the 2D LiDAR on the calibration box.

By taking any two points captured by the 2D LiDAR illumination line on the infrared image, we can then back-project those points to get the ray vectors. Let K be the intrinsic matrix of the depth camera and R, t consecutively the rotation and translation component which transform the world coordinate to the camera. The projection matrix can then be acquired by Equation 4.5

$$P = K[R \mid t] \quad (4.5)$$

or in the invertible form shown in Equation 4.6

$$\tilde{P} = \begin{bmatrix} K & 0 \\ 0^T & 1 \end{bmatrix} \begin{bmatrix} R & t \\ 0^T & 1 \end{bmatrix} = \tilde{K}E \quad (4.6)$$

The calculated camera matrix then can map the 3D point in world coordinate $\bar{\mathbf{p}}_w = (x_w, y_w, z_w, 1)$ to the 2D pixel $\mathbf{x}_s = (x_s, y_s, 1, d)$. The $\tilde{P}$ then can be inverted and applied to the 2D point on the image to gain the back-projected ray vector that passes through the camera focal point.

By back-projecting the two-pixel locations where the 2D LiDAR line is identified, the plane that goes through the focal point and both rays can be calculated. First, the plane normal can be found by $\mathbf{n} = \mathbf{v}_1 \times \mathbf{v}_2$ where $\mathbf{v}_1$ and $\mathbf{v}_2$ are the two ray vectors. The point on the plane can then be determined as the location of the camera's focal point in the world coordinate. By repeating this process from another perspective, we can have two planes that intersect in the 2D LiDAR illumination line on the box. This plane intersection creates a 3D line which represents the illumination line of the 2D LiDAR.

In order to estimate the 2D LiDAR illumination plane, we need at least three separate lines in 3D. This can be achieved by repeating the aforementioned steps multiple times with different planar objects (e.g. boxes with different sizes) or by using one object in which three non-parallel planes are illuminated by the 2D LiDAR.

To fit a plane into these multiple 3D lines we can minimize the sum of squares of the acute angle between the normal vector of the fitted plane and the 3D line representing the illumination of the 2D LiDAR on multiple boxes. Given the vector of 3D lines $\mathbf{u}_i$, their angles θ_i with the normal vector $\mathbf{v}$ can be calculated as Equation 4.7.

$$\theta_i = \arccos \left(\frac{\mathbf{u}_i \cdot \mathbf{v}}{\|\mathbf{u}_i\| \|\mathbf{v}\|} \right) \quad (4.7)$$

And then ensure that these are acute angles by Equation 4.8

$$\alpha_i = \begin{cases} \pi - \theta_i & \text{if } \theta_i > \frac{\pi}{2} \\ -\pi - \theta_i & \text{if } \theta_i < -\frac{\pi}{2} \\ \theta_i & \text{otherwise} \end{cases} \quad (4.8)$$

The estimated plane can then be calculated by minimizing J in Equation 4.9

$$J = \sum_{i=1}^N (\alpha_i)^2 \quad (4.9)$$

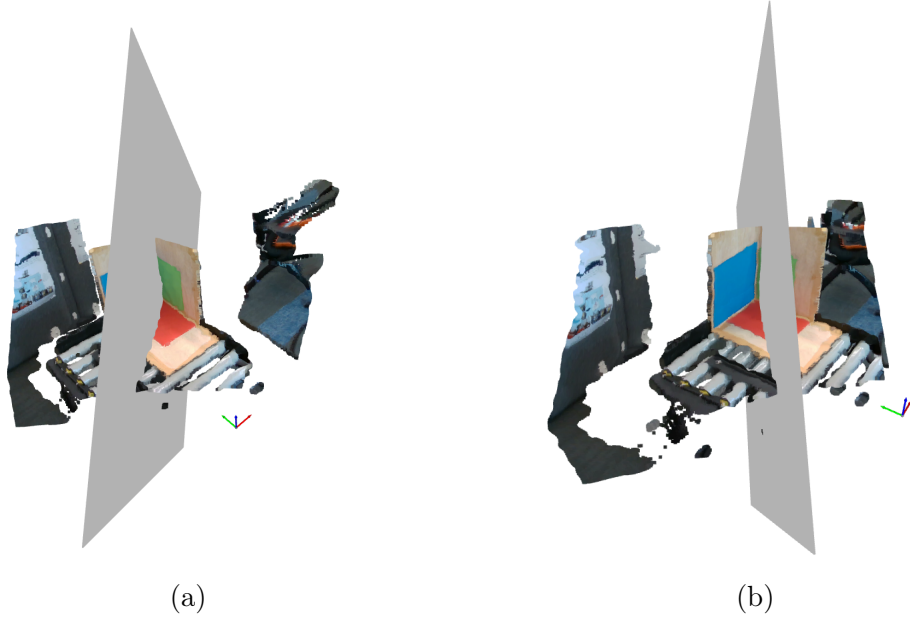


Figure 4.7: (a) and (b) shows the estimated illumination plane of the vertically installed 2D LiDAR visualized along the point cloud from the depth camera (shown in two different views).

The point on this plane can then be defined as the centroid of the points passed through by the 3D illumination lines we fit the plane into. The visualization of this fitted plane, which represents the 2D LiDAR illumination plane, is shown in Figure 4.7.

Once the illumination plane of the 2D LiDAR is known, we need to find a transformation that maps the 2D LiDAR points into the 3D point cloud reconstructed by the camera. First, we sample the points on the depth camera point cloud around the illumination plane. To do this, measure the distance between every point in the point cloud to the plane and include the points in which the distance is below a certain threshold.

Next, the sampled points are projected onto the illumination plane in 3D. For each sampled point x , the vector $\mathbf{v}$ is calculated by $\mathbf{v} = x - p$, where p is a point on the plane. Next, the distance d along the normal plane is computed from a point to the plane by taking the dot product of $\mathbf{v}$ and the plane normal as $d = \mathbf{v} \cdot \mathbf{n}$. Finally, the projected point x' can be computed by $x' = p - d \cdot \mathbf{n}$.

Thus far, the sampled points are in the same plane, however, they are still defined in 3D. To transform the sampled points to 2D we need to define three basis vectors which mutually orthogonal on the illumination plane. First, we take a random point on the plane which acts as the origin. The z-axis can then be defined by the normal

of the plane, then the x-axis can be defined randomly from point a to any point on the plane. The y-axis is calculated by taking the cross product of the normal vector and vector along the x-axis. Given the normalized basis vectors $\hat{\mathbf{u}}$, $\hat{\mathbf{v}}$, and $\hat{\mathbf{n}}$, we obtain the basis points as shown in Equation 4.10

$$a = a, \quad \tilde{\mathbf{u}} = a + \hat{\mathbf{u}}, \quad \tilde{\mathbf{v}} = a + \hat{\mathbf{v}}, \quad \tilde{\mathbf{n}} = a + \hat{\mathbf{n}} \quad (4.10)$$

With a as the point of origin on the plane, now we can map these four basis points into $(0, 0, 0)$, $(1, 0, 0)$, $(0, 1, 0)$, $(0, 0, 1)$, representing the target origin, x, y, and z axis consecutively. This mapping can be achieved by affine transformation whose relation can be described in Equation 4.11.

$$M \begin{bmatrix} a_x & \tilde{\mathbf{u}}_x & \tilde{\mathbf{v}}_x & \tilde{\mathbf{n}}_x \\ a_y & \tilde{\mathbf{u}}_y & \tilde{\mathbf{v}}_y & \tilde{\mathbf{n}}_y \\ a_z & \tilde{\mathbf{u}}_z & \tilde{\mathbf{v}}_z & \tilde{\mathbf{n}}_z \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \quad (4.11)$$

Finding the matrix M which maps the original basis points to the new target coordinate system can be carried out by 4.12

$$\begin{aligned} M \cdot S &= D \\ M \cdot S \cdot S^{-1} &= D \cdot S^{-1} \\ M &= D \cdot S^{-1} \end{aligned} \quad (4.12)$$

Applying the matrix M to the sampled point cloud (which is already projected to the plane) will yield a 3D point with zero on the z component. We can then discard the z-axis and perceive the transformed points as 2D.



Figure 4.8: (a) Sampled point cloud from the depth camera after transformed into 2D. (b) 2D LiDAR points before alignment with the sampled points.

The next step is to align the 2D LiDAR points and the sampled depth-camera point cloud in 2D. Our approach aims to align the two lines and their intersection. The image of the two data points is visualized in Figure 4.8. The translation can be found by aligning the intersection point of the line in both 2D LiDAR and depth camera points. Meanwhile, the rotation can be acquired by aligning the two vectors representing the lines as the new coordinate system.

After the 2D LiDAR points are aligned with the depth-camera point cloud, the 2D LiDAR points can be transformed back into 3D by applying M^{-1} which we have acquired in 4.12.

4.4 Box Dimension Estimation

4.4.1 Depth Camera-based Measurements

Point cloud acquisition

To capture all sides of the box, the depth images have to be taken from multiple perspectives. Since the depth camera is mounted on the robotic arm, this can be achieved by moving the camera around the four corners of the box while capturing its depth map. The four resulting depth images were subsequently converted into point clouds and merged together by using the camera pose information to produce a complete model of the box. The pose of the camera with respect to the world and the robotic arm is essential information that can be acquired by following the camera-robot calibration method (4.2).

Plane detection

The plane detection algorithm runs sequential RANSAC into the merged point cloud and yields exactly five planes (as shown in 4.13) representing the visible sides of the box.

$$\begin{aligned}
 p_1 : a_1x + b_1y + c_1z + d_1 &= 0 \\
 &\vdots \\
 p_5 : a_5x + b_5y + c_5z + d_5 &= 0
 \end{aligned} \tag{4.13}$$

The sequential RANSAC algorithm starts by taking the minimum number of samples (i.e. three for a plane) from a set of data points to fit a plane. The plane's performance is then evaluated by counting the number of inliers based on a predefined threshold. For plane fitting, inliers are determined by the distance of the point to the plane. If the number of inliers is higher than the previous model, it would be considered the best model (the inlier number of the previous model is zero upon initialization). This process is repeated for a number of iterations to find the best model by the highest number of inliers.

Algorithm 1 Multi-plane Detection

Function SEQRANSAC($\mathcal{P}, t, n, \varepsilon$)

Data: A set of input points $\mathcal{P}$, number of iterations t , inlier threshold ε , number of plane to detect n

Result: List of refined planes

```

allRefinedPlanes  $\leftarrow \{\}$ 
for  $j \leftarrow 1$  to  $n$  do
    bestPlane  $\leftarrow$  null bestInliers  $\leftarrow \{\}$ 
    for  $i \leftarrow 1$  to  $t$  do
        randomSample  $\leftarrow$  sample minimally from  $\mathcal{P}$ .
        currentPlane  $\leftarrow$  estimate a plane using randomSample.
        inliers  $\leftarrow$  find plane inliers in  $\mathcal{P}$  using currentPlane and  $\varepsilon$ .
        if  $|inliers| > |bestInliers|$  then
            bestPlane  $\leftarrow$  currentPlane bestInliers  $\leftarrow$  inliers
    refinedPlane  $\leftarrow$  fit a plane to bestInliers
    allRefinedPlanes  $\leftarrow$  allRefinedPlanes  $\cup \{refinedPlane\}$ 
return allRefinedPlanes

```

The best plane is finally refined by fitting a new line into the inlier points by using PCA (Principal Component Analysis). In this step, we can calculate the covariance matrix of the 3D points in which its eigenvalue and eigenvector can be computed. The normal of the fitted plane can be acquired by taking the eigenvector corresponding to the smallest eigenvalue. It is also important to take care that the data needs to be centered first to remove the translation effect hence the PCA focuses on data distribution relative to its center.

It is important to note that since the bottom side of the box is not visible, the calibration process is needed to obtain the base plane equation. This process is done by placing a calibration box at the extreme edge of the conveyor and running the plane detection algorithm to model the bottom side of the calibration box as shown



Figure 4.9: (a) The original point cloud before the planes are segmented. (b) the result of five segmented planes.

in Figure 4.4. This scheme enables us to obtain all six planes of a box. An example of the five detected planes (without the bottom plane) is shown in Figure 4.9.

Planes matching

Let S be the set of all the six detected planes (4.14).

$$S = \{p_1, p_2, p_3, p_4, p_5, p_6\} \quad (4.14)$$

The objective is to match the planes that are considerably parallel to one another. This process is carried out by calculating the acute angle of every combination plane and matching the plane if the acute angle is less than a certain threshold. The resulting matches indicate that the plane pairs are opposing each other and hence do not intersect with one another. The example results of sets of matches are shown in 4.15.

$$\begin{aligned} X_1 &= \{p_1, p_2\} \\ X_2 &= \{p_3, p_4\} \\ X_3 &= \{p_5, p_6\} \end{aligned} \quad (4.15)$$

In the example above, p_1 is parallel with p_2 and so is p_3 with p_4 . Matching the opposite plane together is an essential step in computing the corners since parallel planes are considered to not intersect with one another.

Calculating corners

Getting the 3D coordinate of the box's corners is a matter of solving the linear system of equations of three intersecting planes. This can be achieved by solving

the combination of three planes from three different sets X_1, X_2, X_3 . Hence all the possible combinations of the example 4.15 are shown in 4.16:

$$\begin{aligned} &\{p_1, p_3, p_5\}, \{p_1, p_3, p_6\}, \{p_1, p_4, p_5\}, \{p_1, p_4, p_6\}, \\ &\{p_2, p_3, p_5\}, \{p_2, p_3, p_6\}, \{p_2, p_4, p_5\}, \{p_2, p_4, p_6\} \end{aligned} \quad (4.16)$$

Solving the linear system of plane equations for every combination set will yield the eight points that indicate the eight corners of the box. For example, solving for x , y and z for p_1 , p_3 , and p_5 yield one corner location of the box.

4.4.2 2D LiDAR-based Measurements

In this section, we present a procedure to perform an automatic measurement of box size (i.e. height, width, and length) by using the input of two 2D LiDAR data. Each 2D LiDAR is configured in such a way that they are able to capture the two specific surfaces of the box. Specifically, one 2D LiDAR is placed in a horizontal position to capture the two sides of the box, and the other is positioned vertically to record the top and one other side of the box.

Points Filtering by ROI

Filtering the points acquired by the 2D LiDAR is carried out to obtain the relevant points where the box can be found. This process involves defining the Region of Interest (ROI) which describes the potential space where the box may be located. Since the 2D LiDAR points are given in polar coordinate form, we can define the ROI based on the 2D LiDAR rotation angle. By eliminating the points outside of the ROI, the number of outliers is reduced which makes the subsequent process (i.e. multi-line detection) easier.

Multi-line Detection

Multiple line detection is performed as the step to find the relevant lines representing the side of the box. In this algorithm, the detected lines are the proposed lines which may or may not indicate the box sides. The number of detected lines is user-defined, however, these proposed lines will be rejected to find the relevant line pair in the subsequent step.

After the 2D LiDAR data is filtered based on the scanning angle, sequential RANSAC is applied to find the most significant lines. The complete algorithm which returns all the detected lines is presented in Algorithm 2. The multi-line detection algorithm is similar to the multi-plane detection (4.4.1) with some additional stopping conditions. Supposedly, the RANSAC iteration terminates after the number of iterations t is fulfilled. However, early stopping conditions can be triggered based on the minimum inliers τ or minimum points θ . If the detected lines do not contain a sufficient amount of inliers τ the iteration is stopped early since the estimated line following this iteration most likely contains only outliers. Likewise, when the number of data points left is less than θ the remaining points are presumed to contain only outliers. Similar to the multi-plane detection algorithm (4.4.1), line refinement fits the line using PCA to the detected inlier points except it is applied in 2D.

Algorithm 2 Multi-line Detection

Function SEQRANSAC($\mathcal{P}, t, \varepsilon, n, \tau, \theta$)

Data: A set of input points $\mathcal{P}$, number of iterations t , inlier threshold ε , max number of models to detect n , minimum inliers τ , minimum points θ

Result: List of refined lines

```

allRefinedLines  $\leftarrow \{\}$ 
for  $j \leftarrow 1$  to  $n$  do
    bestLine  $\leftarrow$  null  bestInliers  $\leftarrow \{\}$ 
    for  $i \leftarrow 1$  to  $t$  do
        randomSample  $\leftarrow$  sample minimally from  $\mathcal{P}$ .
        currentLine  $\leftarrow$  estimate a line using randomSample.
        inliers  $\leftarrow$  find line inliers in  $\mathcal{P}$  using currentLine and  $\varepsilon$ .
        if  $|inliers| > |bestInliers|$  then
            bestLine  $\leftarrow$  currentLine  bestInliers  $\leftarrow$  inliers
    if  $|bestInliers| < \tau$  then
        break
    refinedLine  $\leftarrow$  fit a line to bestInliers
    allRefinedLines  $\leftarrow$  allRefinedLines  $\cup \{refinedLine\}$ 
     $\mathcal{P} \leftarrow \mathcal{P} \setminus bestInliers$ 
    if  $|\mathcal{P}| < \theta$  then
        break
return allRefinedLines

```

2D LiDAR scan data contains fewer points and is more vulnerable to detecting the wrong model compared to 3D LiDAR. Although only two lines need to be identified from each 2D LiDAR, detecting only two lines is not an effective strategy since the detected lines can also be outliers. Hence, in this step, multiple-line detection

needs to be carried out followed by subsequent filtering to choose the relevant line pair.

Lines Rejection

Given several detected lines, the irrelevant lines (i.e. lines that do not correspond to the box's plane) will be rejected by testing every combination of lines for their orthogonality and connectivity. The more orthogonal and "connected" the lines are, the more likely they correspond to the relevant sides of the box.

To find the two corresponding sides of the box, we run through this test for every line intersection to examine the orthogonality and connectivity. For example, the intersection of two lines is considered to be false positive when the acute angle is less than 85 degrees. In the case of connectivity, two intersecting lines are said to be more connected when there are more inlier points found around the intersection point.

While the size of the box can be arbitrary, one of the limitations of this implementation is the farthest corner of the box is placed in a fixed position. This allows the region where the box can be detected predictable. Therefore, the region of interest where the box is located can be defined.

Dimension Measurement

Calculating the size of the box is performed by measuring the distance of the detected inlier points of two sides of the box to their corresponding opposite side. However, since the opposite side of the box is not visible by the 2D LiDAR, a calibration box is utilized which represents these opposing sides. This method is feasible since the location of one of the box corners is fixed.

Note that the process discussed above is shown only for one 2D LiDAR. Since multiple 2D LiDARs are implemented, this process is similar for each 2D LiDAR when measuring two dimensions of the box.

Chapter 5

Results and Analysis

5.1 2D LiDAR-Camera Fusion

2D LiDAR-Camera fusion aims to merge the data points from multiple 2D LiDAR and depth cameras in a frame of reference. However, before fusing these sensors, the 3D point cloud of the scene needs to be reconstructed by using multiple depth cameras.

The point cloud information is reconstructed by fusing data captured from four different angles which simulates the use of four-depth cameras. By focusing on a rectangular box as the object of interest, the depth camera captures the box from all four corners. We adjust the camera distance to the target to evaluate its effect on depth noise and measurement error during the box dimension estimation. The distance chosen between the camera and the box corner is 200 mm and 300 mm.

The choice of 200 mm distance comes from the operational limitation of the

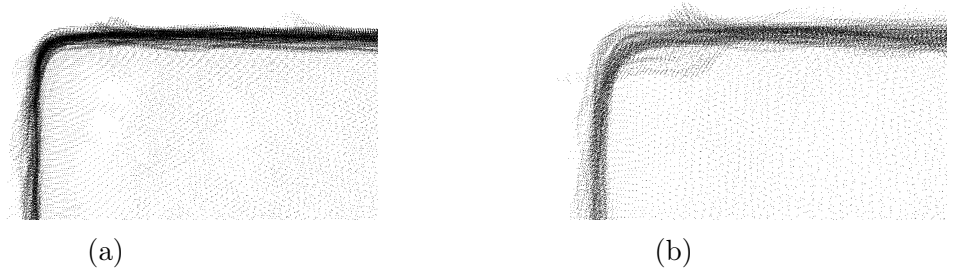


Figure 5.1: Box corner captured from (a) 200 mm and (b) 300 mm distance from the depth camera.

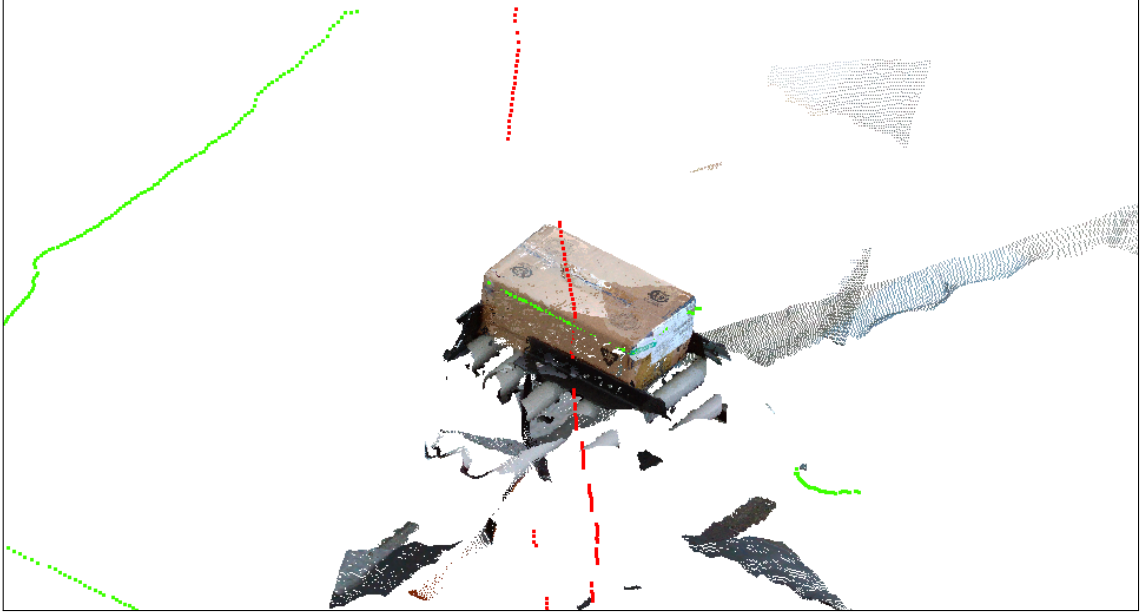


Figure 5.2: The result of point cloud fusion combining four depth camera information and two 2D LiDAR. The green and red points are recorded by two separate 2D LiDAR.

depth camera (Intel D435i), as this represents its lowest feasible range. On the other hand, the 300 mm distance is chosen due to a practical consideration. Since the camera is mounted on the robotic arm, it cannot extend to the 300 mm span from the farthest corner of the box. It is also important to notice that the camera is looking down 45 degrees to the rectangular box.

As shown in figure 5.1, it is expected that the point cloud captured from a distance farther from the object tends to develop more depth noise on the fused point cloud. The depth map is estimated by comparing point correspondences between two views. Between these points, interpolation is performed which introduces depth noise. The inability of the camera to find point correspondences can therefore introduce noise and missing data to the reconstructed point cloud.

To fuse the depth camera and 2D LiDAR information, we sample the point cloud constructed by the depth camera along the illumination plane of the 2D LiDAR. The steps to acquire the illumination plane are discussed in the 2D LiDAR-Camera calibration method (4.3). Take into consideration that when constructing basis vectors to map the sampled 3D points into 2D, the axes directions are crucial. We need to make sure that the transformed point cloud in 2D is not mirrored. Failing to do so will cause the projected point cloud to not be able to be aligned with the original 2D LiDAR points.

Figure 5.2 shows the result of the fusion process between the depth camera and 2D LiDAR with the camera distance of 200 mm to the object and the 2D LiDAR installed vertically and horizontally with respect to the box.

5.2 Multiple Depth-camera Measurement

5.2.1 Multiple Plane Detection

By applying the sequential RANSAC plane detection algorithm to the filtered point cloud yields the five segmented planes in a regular box. Which from visual inspection seems to result in the correct planes representing all sides of the box (Figure 5.3b). However, applying the algorithm to the thinner box (i.e. pizza box) might cause a problem where all the five planes found are parallel. This is because the sides of the boxes are too thin and the number of points is too small to be considered as a plane.

One way to solve this is to separate the detection between the top plane and the other sides of the box. More specifically, during the RANSAC random sampling, consider only the planes that are parallel to the base of the box to find the top box. Once the top box is found, the same process is applied except this time only the

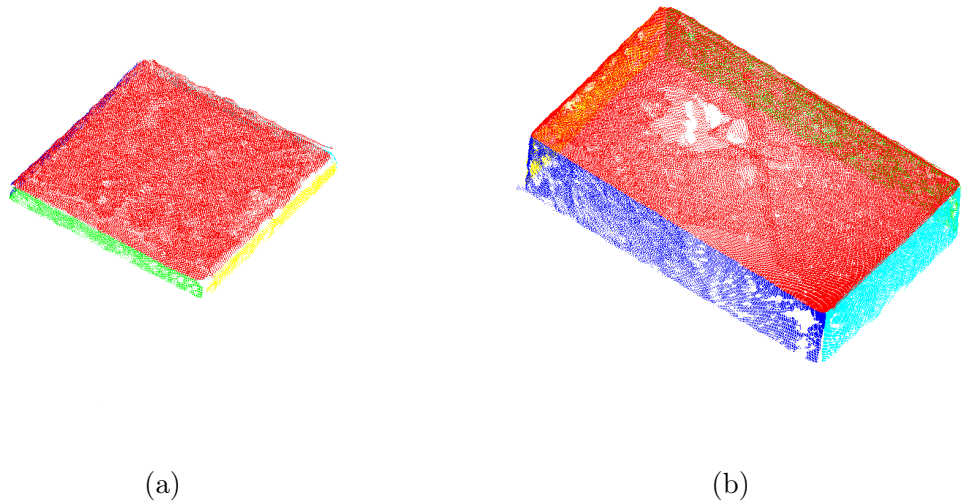


Figure 5.3: (a) The modified sequential RANSAC plane detection applied to the thin box (b) Sequential RANSAC plane detection applied to the regular sized box.

plane orthogonal to the box base plane is considered. The resulting modification of the sequential RANSAC applied to the filtered point cloud is shown in 5.3a.

The presence of an object around the box, especially the ones directly in contact with it, might cause a significant error in the measurement. This is because the objects around the box edges can alter the estimated plane where the wrong object points are included as inliers. We found that this noise is more pronounced at the bottom side of the box since we require a clamp at the conveyor to fix the box in place. Since the fused point cloud is defined in the common coordinate frame, we can easily define the ROI (Region of Interest) where the box can be found around the conveyor. This filtering process is done to minimize the amount of outliers in the point cloud since the box is the only object that we are interested in.

The parameters used for the RANSAC-based plane detection are chosen based on trial and error of multiple boxes which yield the most optimal parameters. Although a theoretical approach can be applied such as determining the number of iterations based on the optimal model fitting probability, our empirical results show that a fixed number of iterations is preferred. This is because the inlier and outlier threshold is unknown which results in a less stable detection. In this experiment, iteration is performed 800 times per plane with point clouds ranging from around 14000 to 30000 data points.

In the case of when the reconstructed point cloud is noisy as the example shown in Figure 5.1b, the inlier threshold for the RANSAC might need to be increased. This is to avoid multiple fitting of parallel planes on the same side of the box. Our experiment identifies the inlier based on a 2.5 mm geometric distance from a point to the plane and a 3.5 mm distance for the camera located farther from the object.

5.2.2 Key Points Extraction

The corner location of the box is extracted by calculating the intersection of three mutually orthogonal planes of the box. From manual visual inspections, the extracted key points aligned to the original point cloud have shown accurate corner locations. The corner points located at the top plane of the box are later used for trajectory planning of the robotic arm to cut open the box. The estimated box corner locations can be seen in Figure 5.4.

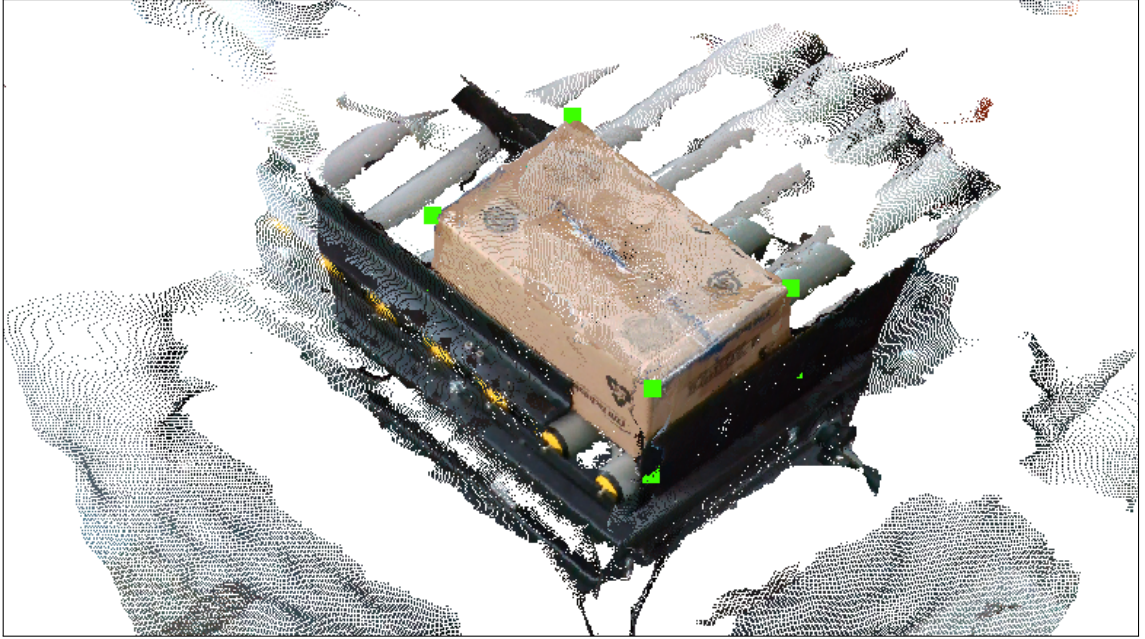


Figure 5.4: The green points exhibit the estimated location of the box corners. Note that the point cloud is shown in full (all data points constructed by four cameras), however, the key points are extracted from the point cloud under a certain ROI.

The accuracy of the corner locations is highly dependent on the plane detection of each side of the box. Hence, for the smaller-sized boxes, as long as the planes can be segmented accurately, the corner location of the box can also be estimated. Figure 5.5 shows the estimated corner points on the thinner and smaller box.

Although it might seem that the corner points at the bottom of the box do not accurately represent the edges, the view from the side of the box shows that the green points at the bottom are parallel with the conveyor plane which the box is



Figure 5.5: Corners extraction in relatively (a) thin and (b) small boxes.

in contact with. Although practically we only need the top corner locations for box unpacking, all the corner points are equally important to extract the dimension of the box.

5.2.3 Dimension Measurement

As discussed in 2D LiDAR-Camera Fusion (5.1), depth noise from the stereo camera tends to increase as the distance to the object grows. In this measurement, we present two setups of distance viewpoints to evaluate the effect of the object distance from the multi-camera system. The two different setup places each camera at 200 mm and 300 mm distance from the top corner of the box (see Experimental Setup).

To derive the dimension of the box from the detected key points, firstly, the average of four corner points on each side of the box is calculated. The resulting average points are exhibited in Figure 5.6a. The distance is then calculated by taking the length for each opposing average point (Figure 5.6b).

The result of the measurement by using the aforementioned parameter is shown both in Table 5.1 and Table 5.2 for 200 mm and 300 mm measurement distance consecutively. It can be seen in the table that the maximum error for any dimension

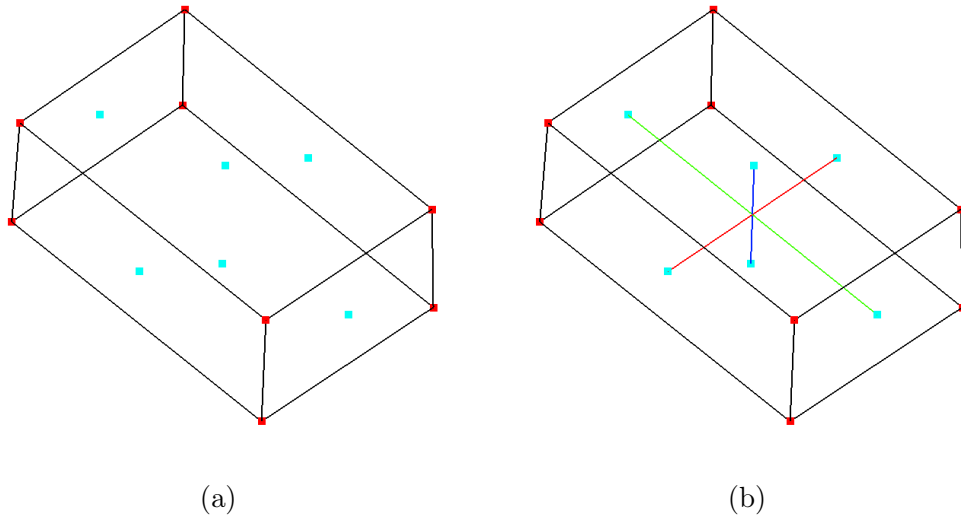


Figure 5.6: (a) Points shown in cyan are the average points of each side of the box's corner points (red points) (b) The dimensions of the box are calculated by measuring the distance between every opposing averaged point (x, y, and z dimensions of the box are measured consecutively by the length of red, green, and blue lines.)

Box	Dim.	Ground Truth	Estimation	Abs. Error	Rel. Error
Box 1	W	240 mm	247 mm	7 mm	2.92%
	L	389 mm	394 mm	5 mm	1.29%
	H	153 mm	154 mm	1 mm	0.65%
Box 2	W	193 mm	197 mm	4 mm	2.07%
	L	261 mm	266 mm	5 mm	1.92%
	H	196 mm	194 mm	2 mm	1.02%
Box 3	W	398 mm	399 mm	1 mm	0.25%
	L	394 mm	395 mm	1 mm	0.25%
	H	184 mm	184 mm	0 mm	0.0%
Box 4	W	372 mm	380 mm	8 mm	2.15%
	L	368 mm	378 mm	10 mm	2.72%
	H	307 mm	304 mm	3 mm	0.98%
Box 5	W	144 mm	147 mm	3 mm	2.08%
	L	197 mm	202 mm	5 mm	2.54%
	H	203 mm	209 mm	6 mm	2.96%
Mean				4.07 mm	1.59%
Standard Deviation				2.89 mm	1.01%

Table 5.1: Box dimension estimation for 200 mm distance between camera and object (measured from the closest box corner.)

Box	Dim.	Ground Truth	Estimation	Abs. Error	Rel. Error
Box 1	W	240 mm	249 mm	9 mm	3.75%
	L	389 mm	396 mm	7 mm	1.8%
	H	153 mm	155 mm	2 mm	1.31%
Box 2	W	193 mm	198 mm	5 mm	2.59%
	L	261 mm	268 mm	7 mm	2.68%
	H	196 mm	195 mm	1 mm	0.51%
Box 3	W	398 mm	399 mm	1 mm	0.25%
	L	394 mm	397 mm	3 mm	0.76%
	H	184 mm	185 mm	1 mm	0.54%
Box 4	W	372 mm	382 mm	10 mm	2.69%
	L	368 mm	379 mm	11 mm	2.99%
	H	307 mm	307 mm	0 mm	0.0%
Box 5	W	144 mm	148 mm	4 mm	2.78%
	L	197 mm	205 mm	8 mm	4.06%
	H	203 mm	209 mm	6 mm	2.96%
Mean				5.00 mm	1.98%
Standard Deviation				3.60 mm	1.32%

Table 5.2: Box dimension estimation for 300 mm distance between camera and object (measured from the closest box corner.)

is less than 10 mm. Note that the maximum size of the measured box is under 400 mm in length. These measurements (i.e. ground truth) are measured by hand which is deemed to be accurate. Meanwhile, the relative error shows the ratio of the absolute error and the true value.

From both measurements, we can conclude that the camera closer to the object performs better in the measurement with lower Mean Absolute Error (MAE) and lower standard deviation (less variability). However, the difference might be considered to be minute in certain applications. Nevertheless, the constructed point cloud in greater observation distance is noisier which may indicate the need for adjustment in the plane detection parameter (i.e. inlier threshold).

5.3 Multiple 2D LiDAR Measurement

5.3.1 Box Side Line Segmentation

As shown in Figure 4.6, the two 2D LiDAR illumination lines can be observed at the surface of the calibration target. This gives a nice visual representation of where the 2D LiDAR illuminates the object during the measurement. The raw measurement of a box from vertical 2D LiDAR can also be seen in Figure 5.7a. Before performing line segmentation on these 2D LiDAR points, the recorded data can be filtered around a certain ROI to include only the points with the box as presented in Figure 5.7b.

Following our discussion in the 2D LiDAR-based measurement method (4.4.2), the valid sides of the box can be obtained by testing the detected line pair based on their connectivity and orthogonality. The visualization of these attributes can be seen in Figure 5.8 in which the green circles indicate the intersection of all the proposed lines. The two lines' segment points are said to be connected when a number of points can be found inside the intersection circle. Take care that the number of proposed lines is the user-defined parameter, however, only one line pair will be chosen based on their orthogonality and connectivity. Setting the radius points parameter is essential to get the correct lines that represent the side of the box. The radius has to be big enough to capture the points around the valid line intersection, but not too big that the invalid line intersection starts to include the valid line points.

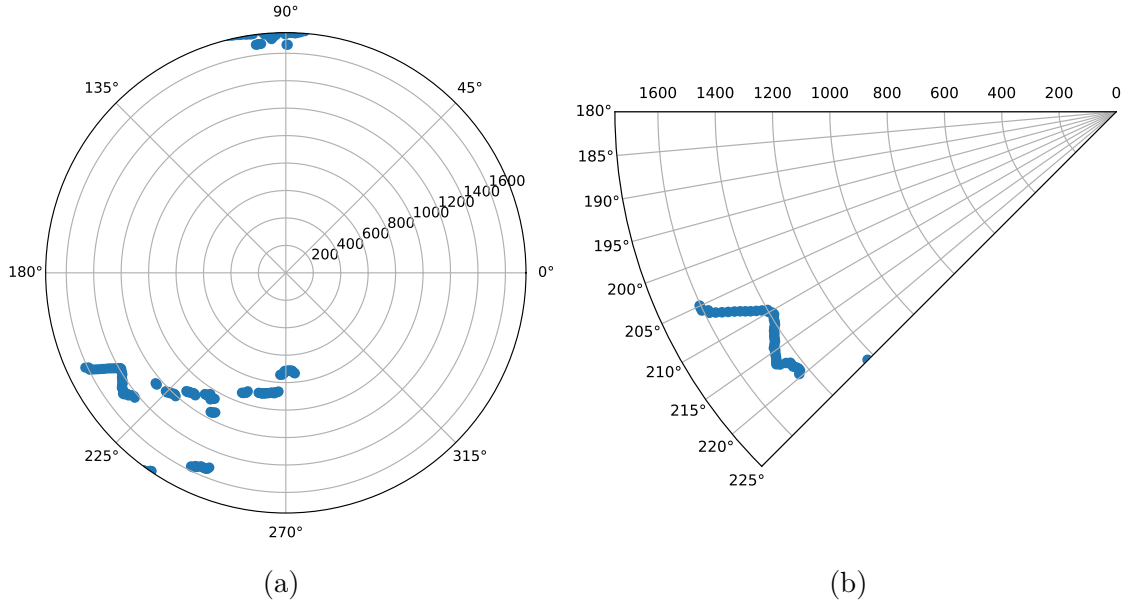


Figure 5.7: (a) The original raw data of the vertically positioned 2D LiDAR scanning 360 degrees upright across the room. (b) The filtered 2D LiDAR data focuses only on the ROI where the box can be found.

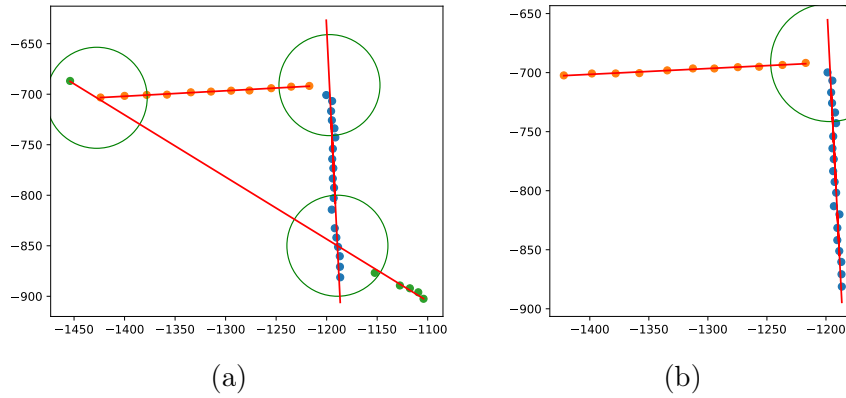


Figure 5.8: (a) The green circles show the radius where the inliers are counted to measure the connectivity of intersecting lines. (b) The selected valid line pair after the invalid line is rejected.

Setting the maximum number of proposed lines is also essential. Since we are interested in looking for a pair of valid lines, a number of proposed lines higher than two is recommended. In this case, we detect three lines and reject one consequently. Which results in a more robust valid line pair detection.

5.3.2 Dimension Measurement

After acquiring the valid line pair that represents the 2D LiDAR points on the side of the box, the next goal is to calculate the box dimension. Since we do not

have the information on the opposite side of the line pair, a plane that acts as a virtual plane should be recorded to substitute the other sides of the box that are not reached by the 2D LiDAR points. The merged data of the virtual line pair and the original line pair is shown in Figure 5.9.

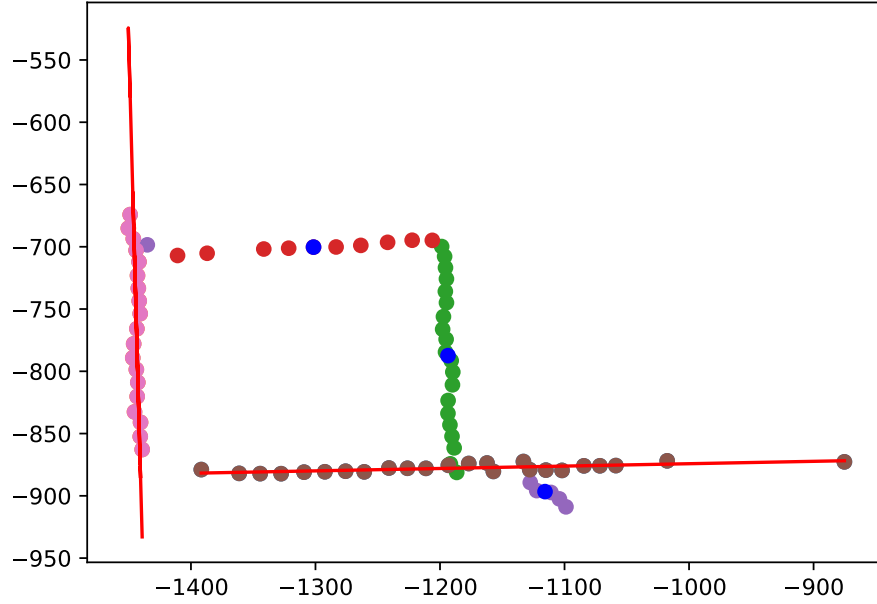


Figure 5.9: The red and green points are the detected top and side of the box consecutively while the purple and brown points are the calibration box. Notice that the blue points are the geometric median of each detected line.

Referring to Figure 5.9, the blue points are the calculated median points of each proposed line inliers. Be aware that the blue point at the bottom (the median of purple points) will be rejected, given that its line does not satisfy the orthogonality and connectivity criteria.

Finally, to calculate the dimension of each side of the box, the distance is taken from each median point to the opposite virtual line. Keep in mind that the result above is shown for the vertical 2D LiDAR, however, an identical method can be applied to the horizontal 2D LiDAR by finding the box line pair and consecutively calculating the distance to the calibration line.

Box	Dim.	Ground Truth	Estimation	Abs. Error	Rel. Error
Box 1	W	240 mm	237 mm	3 mm	1.25%
	L	389 mm	387 mm	2 mm	0.51%
	H	153 mm	142 mm	11 mm	7.19%
Box 2	W	193 mm	187 mm	6 mm	3.11%
	L	261 mm	261 mm	0 mm	0.0%
	H	196 mm	180 mm	16 mm	8.16%
Box 3	W	398 mm	385 mm	13 mm	3.27%
	L	394 mm	389 mm	5 mm	1.27%
	H	184 mm	181 mm	3 mm	1.63%
Box 4	W	372 mm	370 mm	2 mm	0.54%
	L	368 mm	371 mm	3 mm	0.82%
	H	307 mm	299 mm	8 mm	2.61%
Box 5	W	144 mm	140 mm	4 mm	2.78%
	L	197 mm	191 mm	6 mm	3.05%
	H	203 mm	201 mm	2 mm	0.99%
Average				5.60 mm	2.48%
Standard Deviation				4.56 mm	2.37%

Table 5.3: Box measurement using two 2D LiDAR.

Table 5.3 shows the measured dimension for each side of the box (width, length, height). Note that the tested box sizes are identical to the test during the camera-based measurement. From the table, it can be seen that the 2D LiDAR-based measurement tends to be less stable with such large errors throughout the measurements. For example, the maximum error of the 2D LiDAR-based method is 16 mm compared to 10 mm in camera-based measurement. Other large errors are also spotted in box 1 and box 2 which surpass the maximum error of camera-based measurement.

5.4 Combining Measurements

The measurement from multiple 2D LiDAR and depth cameras can be combined in an attempt to gain an even more accurate measurement of the box. This technique is usually referred to as high-level sensor fusion when the estimated measurement is combined during the late stage of the pipeline. The combined results can simply be acquired by taking the average of both measurements. Table 5.4 shows the averaged measurements of all box dimensions.

Since all the boxes measured throughout the experiments stay consistent, comparison can be performed for all measurement methods. Conclusively, combining

Box	Dim.	GT	Combined Est.	Abs. Error	Rel. Error
Box 1	W	240 mm	242.0 mm	2.0 mm	0.83%
Box 1	L	389 mm	390.5 mm	1.5 mm	0.39%
Box 1	H	153 mm	148.0 mm	5.0 mm	3.27%
Box 2	W	193 mm	192.0 mm	1.0 mm	0.52%
Box 2	L	261 mm	263.5 mm	2.5 mm	0.96%
Box 2	H	196 mm	187.0 mm	9.0 mm	4.59%
Box 3	W	398 mm	392.0 mm	6.0 mm	1.51%
Box 3	L	394 mm	392.0 mm	2.0 mm	0.51%
Box 3	H	184 mm	182.5 mm	1.5 mm	0.81%
Box 4	W	372 mm	375.0 mm	3.0 mm	1.81%
Box 4	L	368 mm	374.5 mm	6.5 mm	1.77%
Box 4	H	307 mm	301.5 mm	5.5 mm	1.79%
Box 5	W	144 mm	143.5 mm	0.5 mm	0.35%
Box 5	L	197 mm	196.5 mm	0.5 mm	0.25%
Box 5	H	203 mm	205.0 mm	2.0 mm	0.98%
Mean				3.23 mm	1.29%
Standard Deviation				2.55 mm	1.20%

Table 5.4: Box measurement using the combination of 2D LiDAR and depth camera.

Method	MAE	MRE	Std. AE	Std. RE
Multi-2D LiDAR	5.60 mm	2.48%	4.56 mm	2.37%
Multi Depth Camera (200)	4.07 mm	1.59%	2.89 mm	1.01%
Multi Depth Camera (300)	5.00 mm	1.98%	3.60 mm	1.32%
2D LiDAR-Depth Cam.	3.23 mm	1.29%	2.55 mm	1.20%

Table 5.5: Final comparison of the four measurement methods.

both methods yields the lowest error (MAE and MRE) compared to relying only on the 2D LiDAR or depth camera methods. However, the standard deviation of relative error is higher than the multi-depth-camera method at the 200 mm distance. This shows that although the estimates are close to the actual values on average there is a significant amount of variability in predictions. As discussed in the 2D LiDAR dimension measurement (5.3.2), the LiDAR-based method performs poorly in some instances. This is most likely the cause of higher variability in the measurement. The comparison of all methods can be seen in 5.5.

5.5 Box Unpacking

To unpack the box, a robotic arm is equipped with a sharp edge tool to slice through the top side of the cardboard material. With this unpacking scheme, we can generate the path for the robotic arm to move from one corner of the box to the other (with a specified offset applied from each corner) while making sure that the sharp blade sufficiently penetrates the box. An example of the box that has already been cut can be seen in 5.10.

Based on our test results, the sharp edge tool consistently demonstrated the ability to scratch the box's surface in a significant portion of trials. However, variations in cutting depth were observed, ranging from insufficient to excessive penetration into the box surface. We observe that numerous factors can cause this issue.



Figure 5.10: The box that is already unpacked by the robotic arm.

One of the factors is due to the imperfection of the box itself. The path planning of the box cutting relies on the assumption of a perfectly rectangular box. However, this might not always be true since the surface of the box could slightly be convex or concave.

It is also worth noting that the margin for error during the arm movement of the cutting is very limited. The sharp metal part of the blade is only 2 cm long, demanding precision in determining the ideal depth of cutting. This makes an error as small as a 1 cm distance from the box to result in inadequate cutting depth.

Chapter 6

Conclusion

This research aims to provide a complete procedure for sensor fusion and its subsequent application. First, a new calibration procedure is shown to merge multiple depth cameras and 2D LiDAR information. Subsequently, we tested the fused 3D structure by performing measurements on various boxes. Finally, the conclusion is drawn of whether the system is accurate and reliable for an automatic box unpacking application.

Based on our measurement and practical usage, sensor fusion between the depth camera and 2D LiDAR has shown a satisfactory result. Segmenting lines in the recorded 2D LiDAR points do not show any major error after the measurement is compared with the ground truth of box dimensions. Likewise, plane detection on the reconstructed point cloud shows considerably high accuracy with a dimension error of less than 1 cm for boxes larger than 10 cm in each dimension. Ultimately, the combined approach has shown the best performance among the two working independently from one another.

Despite the high accuracy of box measurement, the problem of box cutting such as slicing is not sufficiently deep or too deep into the cardboard is found from time to time during the test. We also encounter cases where the path of the cut is not parallel with the box surface. However, a handful of successful box cutting has also been observed and we expect future research to improve the box unpacking performance.

6.1 Future Work

We expect future research to implement the state-of-the-art multi-model fitting algorithm both for plane and line detection instead of the sequential RANSAC. Any other robust fitting method could also be explored which improves the speed since our current implementation is considered to be time-consuming.

Another box modeling technique can also be explored. Currently, we assume that boxes are perfectly rectangular hence detecting more than the corner points of the box might provide more comprehensive information to cut the box.

Finally, solving the box unpacking problem goes beyond estimating an accurate model of the box. It is a combination of accurate robotics control, suitable hardware, and precise measurement. While accurate modeling of the box is important, we also need to take care to build a cutting system that can accommodate errors. We also recommend exploring other robot end effectors that might improve the box-cutting performance.

Acknowledgements

This work is conducted within the framework of academic-industry research collaboration with Robert Bosch Kft. I would like to express my gratitude for their generous support to the Department of Artificial Intelligence (AI) of Eötvös Loránd University. Much appreciation goes to my primary supervisor Hajder Levente for his ideas, guidance, and support throughout the research. I would also like to express gratitude to my second supervisor Kovács Bandó, especially for his involvement in the hardware implementation. Special thanks are extended to Sallai Martin for providing assistance with the robotics arm control throughout the research.

Bibliography

- [1] Tao Peng et al. “Dimension Measurement and Key Point Detection of Boxes through Laser-Triangulation and Deep Learning-Based Techniques”. In: *Applied Sciences* 10.1 (2020). ISSN: 2076-3417. DOI: 10.3390/app10010026. URL: <https://www.mdpi.com/2076-3417/10/1/26>.
- [2] Napol Ladplee et al. “Volumetric Measurement of Rectangular Parcel Box Using LiDAR Depth Camera for Dimensioning and 3D Bin Packing Applications”. In: *2022 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*. 2022, pp. 1–4. DOI: 10.1109/ICCE-Asia57006.2022.9954650.
- [3] David G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. In: *International Journal of Computer Vision* 60.2 (Nov. 2004), 91–110. ISSN: 0920-5691. DOI: 10.1023/b:visi.0000029664.99615.94. URL: <http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94>.
- [4] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. “SURF: Speeded Up Robust Features”. In: *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2006, 404–417. ISBN: 9783540338338. DOI: 10.1007/11744023_32. URL: http://dx.doi.org/10.1007/11744023_32.
- [5] Ethan Rublee et al. “ORB: An efficient alternative to SIFT or SURF”. In: *2011 International Conference on Computer Vision*. IEEE, Nov. 2011. DOI: 10.1109/iccv.2011.6126544. URL: <http://dx.doi.org/10.1109/ICCV.2011.6126544>.
- [6] Changsoo Je, Sang Wook Lee, and Rae-Hong Park. “High-Contrast Color-Stripe Pattern for Rapid Structured-Light Range Imaging”. In: (2015). DOI: 10.48550/ARXIV.1508.04981. URL: <https://arxiv.org/abs/1508.04981>.

- [7] *Depth Camera D435i*. <https://www.intelrealsense.com/depth-camera-d435i/>. Last accessed: 2023-12-02.
- [8] *Projectors for D400 Series Depth Cameras*. <https://dev.intelrealsense.com/docs/projectors>. Last accessed: 2023-12-02.
- [9] Mingkang Li et al. “Environment Mapping and Vehicle Localization with a High-Resolution Radar Prototype”. In: Aug. 2019.
- [10] Xianjian Jin et al. “Robust LiDAR-Based Vehicle Detection for On-Road Autonomous Driving”. In: *Remote Sensing* 15.12 (June 2023), p. 3160. ISSN: 2072-4292. DOI: 10.3390/rs15123160. URL: <http://dx.doi.org/10.3390/rs15123160>.
- [11] Bo Li, Tianlei Zhang, and Tian Xia. *Vehicle Detection from 3D Lidar Using Fully Convolutional Network*. 2016. DOI: 10.48550/ARXIV.1608.07916. URL: <https://arxiv.org/abs/1608.07916>.
- [12] Martin Kada, Laurence McKinley, and Zellescher Weg. “3D BUILDING RECONSTRUCTION FROM LIDAR BASED ON A CELL DECOMPOSITION APPROACH”. In: 2009. URL: <https://api.semanticscholar.org/CorpusID:8959736>.
- [13] Carla Nardinocchi et al. “Building Reconstruction and Visualization from LIDAR Data”. In: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* (2003), pp. 151–156. URL: <https://api.semanticscholar.org/CorpusID:16652539>.
- [14] Onur Ozyesil et al. *A Survey of Structure from Motion*. 2017. DOI: 10.48550/ARXIV.1701.08493. URL: <https://arxiv.org/abs/1701.08493>.
- [15] Yao Yao et al. *MVSNet: Depth Inference for Unstructured Multi-view Stereo*. 2018. DOI: 10.48550/ARXIV.1804.02505. URL: <https://arxiv.org/abs/1804.02505>.
- [16] Mohammad O. A. Aqel et al. “Review of visual odometry: types, approaches, challenges, and applications”. In: *SpringerPlus* 5.1 (Oct. 2016). ISSN: 2193-1801. DOI: 10.1186/s40064-016-3573-7. URL: <http://dx.doi.org/10.1186/s40064-016-3573-7>.

- [17] Raul Mur-Artal, J. M. M. Montiel, and Juan D. Tardos. “ORB-SLAM: A Versatile and Accurate Monocular SLAM System”. In: *IEEE Transactions on Robotics* 31.5 (Oct. 2015), 1147–1163. ISSN: 1941-0468. DOI: 10.1109/tro.2015.2463671. URL: <http://dx.doi.org/10.1109/TR0.2015.2463671>.
- [18] Jakob Engel, Jörg Stückler, and Daniel Cremers. “Large-scale direct SLAM with stereo cameras”. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015, pp. 1935–1942. DOI: 10.1109/IROS.2015.7353631.
- [19] T. Voegtlin and P.F. Dominey. “Learning high-degree sequences in a linear network”. In: *IJCNN’01. International Joint Conference on Neural Networks. Proceedings (Cat. No.01CH37222)*. Vol. 2. 2001, 940–944 vol.2. DOI: 10.1109/IJCNN.2001.939486.
- [20] Mordechai Ben-Ari and Francesco Mondada. “Robotic Motion and Odometry”. In: *Elements of Robotics*. Springer International Publishing, Oct. 2017, 63–93. ISBN: 9783319625331. DOI: 10.1007/978-3-319-62533-1_5. URL: http://dx.doi.org/10.1007/978-3-319-62533-1_5.
- [21] Qilong Zhang and R. Pless. “Extrinsic calibration of a camera and laser range finder (improves camera calibration)”. In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*. Vol. 3. 2004, 2301–2306 vol.3. DOI: 10.1109/IROS.2004.1389752.
- [22] S. Wasielewski and O. Strauss. “Calibration of a multi-sensor system laser rangefinder/camera”. In: *Proceedings of the Intelligent Vehicles ’95. Symposium*. 1995, pp. 472–477. DOI: 10.1109/IVS.1995.528327.
- [23] Ganhua Li et al. “An Algorithm for Extrinsic Parameters Calibration of a Camera and a Laser Range Finder Using Line Features”. In: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2007, pp. 3854–3859. DOI: 10.1109/IROS.2007.4399041.
- [24] Zixuan Bai, Guang Jiang, and Ailing Xu. “LiDAR-Camera Calibration Using Line Correspondences”. In: *Sensors* 20.21 (2020). ISSN: 1424-8220. DOI: 10.3390/s20216319. URL: <https://www.mdpi.com/1424-8220/20/21/6319>.

- [25] Martin A. Fischler and Robert C. Bolles. “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography”. In: *Communications of the ACM* 24.6 (June 1981), 381–395. ISSN: 1557-7317. DOI: 10.1145/358669.358692. URL: <http://dx.doi.org/10.1145/358669.358692>.
- [26] E. Vincent and R. Laganier. “Detecting planar homographies in an image pair”. In: *ISPA 2001. Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis. In conjunction with 23rd International Conference on Information Technology Interfaces (IEEE Cat. 2001)*, pp. 182–187. DOI: 10.1109/ISPA.2001.938625.
- [27] Yasushi Kanazawa and Hiroshi Kawakami. “Detection of Planar Regions with Uncalibrated Stereo using Distributions of Feature Points”. In: (Jan. 2004). DOI: 10.5244/C.18.27.
- [28] Hossam Isack and Yuri Boykov. “Energy-Based Geometric Multi-model Fitting”. In: *International Journal of Computer Vision* 97.2 (July 2011), 123–147. ISSN: 1573-1405. DOI: 10.1007/s11263-011-0474-7. URL: <http://dx.doi.org/10.1007/s11263-011-0474-7>.
- [29] Daniel Barath and Jiri Matas. *Multi-Class Model Fitting by Energy Minimization and Mode-Seeking*. 2017. DOI: 10.48550/ARXIV.1706.00827. URL: <https://arxiv.org/abs/1706.00827>.
- [30] Daniel Barath and Jiri Matas. *Progressive-X: Efficient, Anytime, Multi-Model Fitting Algorithm*. 2019. DOI: 10.48550/ARXIV.1906.02290. URL: <https://arxiv.org/abs/1906.02290>.
- [31] Luca Magri and Andrea Fusiello. “Robust Multiple Model Fitting with Preference Analysis and Low-rank Approximation”. In: *Proceedings of the British Machine Vision Conference 2015. BMVC 2015*. British Machine Vision Association, 2015. DOI: 10.5244/c.29.20. URL: <http://dx.doi.org/10.5244/C.29.20>.
- [32] Paul Amayo et al. *Geometric Multi-Model Fitting with a Convex Relaxation Algorithm*. 2017. DOI: 10.48550/ARXIV.1706.01553. URL: <https://arxiv.org/abs/1706.01553>.

- [33] Tsun-Hsuan Wang et al. *3D LiDAR and Stereo Fusion using Stereo Matching Network with Conditional Cost Volume Normalization*. 2019. DOI: 10.48550/ARXIV.1904.02917. URL: <https://arxiv.org/abs/1904.02917>.

List of Figures

3.1	Depth Camera D435i [7]	7
3.2	An example of (a) 2D LiDAR points and (b) 3D LiDAR point cloud.	8
4.1	Architecture diagram of the automatic box unpacking system.	12
4.2	Hardware setup for the experiment.	12
4.3	(a) The sharp edge tool from the bottom view (b) the sharp edge tool from the top view (mounting section to the robotic arm).	13
4.4	The calibration target placed on the conveyor where the box will be unpacked.	14
4.5	(a) The estimated pose of the calibration object with respect to the depth camera (b) The visualization of the rigid body configuration after the calibration box is estimated.	16
4.6	(a) Vertical and (b) horizontal 2D LiDAR illumination captured by infrared camera.	17
4.7	(a) and (b) shows the estimated illumination plane of the vertically installed 2D LiDAR visualized along the point cloud from the depth camera (shown in two different views).	19
4.8	(a) Sampled point cloud from the depth camera after transformed into 2D. (b) 2D LiDAR points before alignment with the sampled points.	20
4.9	(a) The original point cloud before the planes are segmented. (b) the result of five segmented planes.	23
5.1	Box corner captured from (a) 200 mm and (b) 300 mm distance from the depth camera.	27
5.2	The result of point cloud fusion combining four depth camera information and two 2D LiDAR. The green and red points are recorded by two separate 2D LiDAR.	28

5.3	(a) The modified sequential RANSAC plane detection applied to the thin box (b) Sequential RANSAC plane detection applied to the regular sized box.	29
5.4	The green points exhibit the estimated location of the box corners. Note that the point cloud is shown in full (all data points constructed by four cameras), however, the key points are extracted from the point cloud under a certain ROI.	31
5.5	Corners extraction in relatively (a) thin and (b) small boxes.	31
5.6	(a) Points shown in cyan are the average points of each side of the box's corner points (red points) (b) The dimensions of the box are calculated by measuring the distance between every opposing averaged point (x, y, and z dimensions of the box are measured consecutively by the length of red, green, and blue lines.)	32
5.7	(a) The original raw data of the vertically positioned 2D LiDAR scanning 360 degrees upright across the room. (b) The filtered 2D LiDAR data focuses only on the ROI where the box can be found.	35
5.8	(a) The green circles show the radius where the inliers are counted to measure the connectivity of intersecting lines. (b) The selected valid line pair after the invalid line is rejected.	35
5.9	The red and green points are the detected top and side of the box consecutively while the purple and brown points are the calibration box. Notice that the blue points are the geometric median of each detected line.	36
5.10	The box that is already unpacked by the robotic arm.	39

List of Tables

5.1	Box dimension estimation for 200 mm distance between camera and object (measured from the closest box corner.)	33
5.2	Box dimension estimation for 300 mm distance between camera and object (measured from the closest box corner.)	33
5.3	Box measurement using two 2D LiDAR.	37
5.4	Box measurement using the combination of 2D LiDAR and depth camera.	38
5.5	Final comparison of the four measurement methods.	38

List of Algorithms

1	Multi-plane Detection	22
2	Multi-line Detection	25