



Development of Human Following Mobile Robot System with 3D LiDAR and Online Human Classifier

Thesis submitted to the undergraduate degree program in G30 Automotive Mechanical Engineering, Department of Mechanical and Aerospace Engineering and G30 Undergraduate faculty of Nagoya University in partial fulfillment of the requirements for the degree of Bachelor of Engineering.

**Le Nhat Quang
081851013**

Supervised by Prof. Naoki Akai and Prof. Susumu Hara

Hara laboratory of Control Systems Engineering

July 2022

Contents

1	Introduction	5
1.1	Research Background	5
1.2	Research purpose	6
2	Related research	7
3	Proposed System for Human Following Robot	10
3.1	Pipeline	10
3.2	Pipeline breakdown and details	11
3.2.1	LiDAR point cloud pre-processing step	11
3.2.2	Adaptive Euclidean clustering	13
3.2.3	Online learning for SVM human classification	15
3.2.4	PID for Robot controller	19
4	Experimental Setup and Results	21
4.1	Experiments Platform	21
4.1.1	Experiments Design	21
4.1.2	Experiments Setup and Simulations	24
4.2	Experiments Results	27
4.2.1	Clustering result	27
4.2.2	Online training for SVM result	28
4.2.3	Pipeline testing result	30
5	Conclusions and Future Work	34
5.1	Summary and Conclusion	34
5.2	Future Work	36
6	Acknowledgements	37
7	References	38

List of Tables

3.1	All features for human classification	17
4.1	3D VLP-16 LiDAR properties [28]	22

List of Figures

1.1 Application of robot in real life[3, 4]	5
2.1 A Pipeline of the ROS-based Human Following Robot system using camera[10]	7
2.2 Shins Human detection and following using LRS[12]	8
2.3 Human detection and tracking using Online learning with 3D LiDAR [15] . .	9
3.1 A Pipeline of the ROS-based Human Following Robot system	10
3.2 RANSAC plane distance estimation [18]	12
3.3 Example of resultant inliers (purple) point support best planar model (ground) [19]	12
3.4 Example of Euclidean object clustering [19]	13
3.5 Different d^* lead to different clustering results [15, 22]	14
3.6 Shape of human point clouds with respect to distance from LiDAR [15, 22] .	15
3.7 Online learning framework [22]	16
3.8 Positive and Negative clusters	18
3.9 Human target distance and direction of mobile robot [12]	19
4.1 VLP-16 LiDAR vertical field of view	21
4.2 Design of VLP-16 LiDAR onboard Icart-mini robot	22
4.3 Final experiments design in real life	23
4.4 Corridor 3D Map	24
4.5 Human walking simulation using waypoint generator	25
4.6 Experiment setup in 3 cases	26
4.7 Corridor clustering result	27
4.8 Testing hall clustering result	28
4.9 Online training result in corridor environment	29
4.10 Pipeline testing in Gazebo environment	30
4.11 HFR 2D path in Gazebo simulated environment	30
4.12 Pipeline testing in corridor environment	31
4.13 2D path in Nagoya University Engineering Building 2 4th floor corridor . .	31
4.14 Pipeline Test in aerospace hall environment	32
4.15 2D Path in Nagoya University ‘s Aerospace Testing hall	32

Abbreviations and nomenclature

Nomenclature

LiDAR Light Detection and Ranging

HFR Human Following Robot

ROS Robot Operating System

RANSAC Random Sample Consensus

P Initial point cloud data

p_i Point i in point cloud data P

d_t Distance threshold for RANSAC

P^* Point cloud data $P^* \subset P$ after being filtered by RANSAC

ROI Region of Interest

C List of point cloud clusters from P

d^* Clustering threshold

r Detected range of LiDAR

θ LiDAR vertical angle resolution

SVM Support Vector Machine

EKF Extended Kalman Filter

C – SVC C-Support Vector Classification

RBF Gaussian Radial Basic Function

TP True Positive

TN True Negative

FP False Positive

FN False Negative

PID Proportional, Integral and Derivative Control

Rviz ROS Visualisation

Chapter 1

Introduction

1.1 Research Background

The development of robots now plays a very important role in many different fields, such as civil, medical, educational and so on [1, 2]. Society is moving towards the fact that humans can live with robots, with robots being more and more intelligent to be able to assist people in work as well as in daily life. The intelligence of the robot is the determining factor of the robot's ability. As robots get smarter, handling tasks will become easier and smoother. Thus, the research of artificial intelligence is now very developed.



Figure 1.1: Application of robot in real life[3, 4]

Examples of applications of robots today are various, notably the application of robots to transport goods in factories, robot vacuums, or receptionist robots with recognition technology to assist people at airports or public areas. In addition, there are some robot appli-

cations that follow humans to assist with specific tasks such as carrying luggage or goods (see Fig. 1.1). These systems generally use sensors to recognize information about the external environment, combined with machine learning algorithms to increase accuracy in object recognition and tracking[5].A people following system by a robot is composed of two main modules: people identification and tracking and control to follow a recognized person.

1.2 Research purpose

In this research, we focus on identification and tracking of people. For previous studies, most used 2D LiDAR sensors or cameras. For cameras, there are now many algorithms to identify and track people, but for complex environments, there are still limitations such as poor visibility in weather conditions, and especially the use of cameras [6]. The field of vision is narrow, making it simple to lose the item while following and challenging to catch the right thing in time. As for the use of 2D LiDAR, it allows identification in a wide range, can track 360 degrees, but following people in complex environments is limited because the features from 2D LiDAR are very few, easily confused with other objects.

In this study, we use 3D VLP16 LiDAR . Because 3D LiDAR has a wide scanning range and 360 degrees like 2D LiDAR and can capture more features like a camera. The contribution of that project is to propose a pipeline for the robot's human tracking using 3D LiDAR.This pipeline includes two main processing parts that are user identification online learning and control using PID controller. In addition, the pipeline will be added with constraints and filters to increase the accuracy of identifying and following people. Moreover, using online learning in training will boost the learning time and data since the model is continuously updated during the run, thereby improving accuracy and reducing data collection time from users.

The rest of this thesis is organized as follows. Chapter 2 summarizes related works. Chapter 3 presents the Human Following Robot Proposed System. Chapter 4 details all conducted experiments and describes their results. Chapter 5 concludes this study and states ideas for future work.

Chapter 2

Related research

Regarding human-following robots, there has been research on this issue for a long time, outstanding Schlegel *et al.* [7] used human color and contour information to conduct tracking. In the study, the characteristic information was based on the color, edges, and texture of the clothes for tracking. For regular RGB images, Li *et al.* [8] used image segmentation based on shape and color. The exploitation of image color spaces is also studied, including the use of HSV (Hue, Saturation, Value) and D spaces in the RGB-D space of Ren *et al.* [9]. Based on those two parameters, the study has determined the skeleton of the subject to be monitored. With the development of machine learning, and deep learning today. Detecting people with a camera has become a lot easier. B.X.Chen *et al.* [10] proposed a pipe for following human robot based ROS using camera and CNN to detect and tracking people (see Fig. 2.1).

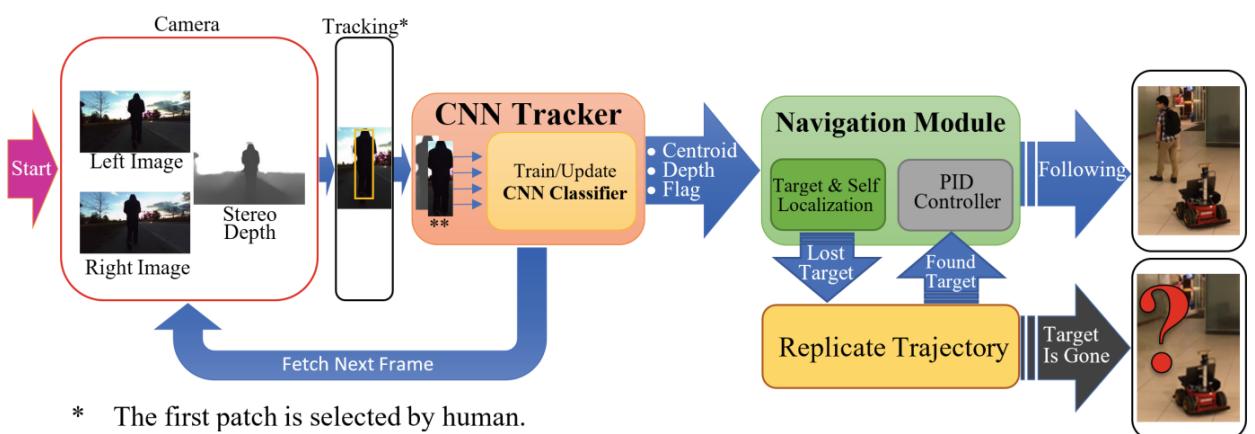


Figure 2.1: A Pipeline of the ROS-based Human Following Robot system using camera[10]

In Redhwan.A and Mun-Taek.Ci's study[11], the team used SSD (Single Shot Detector) to optimize human detection, combined with extracting color features in HSV space to track objects. Research using LiDAR to determine the position of the robot as well as the position of the person in the map, as well as to identify obstacles from which it is possible to create a path from the robot position to the human position for the robot to follow.

For the use of cameras in human-following robots, the disadvantage is that the camera has a narrow field of view, the distance for good quality is low, and it may be affected in complex environments or bad weather. Therefore, many studies have begun to use LiDAR as an alternative to the camera. Kawarazaki *et al.* [12] proposed a method using LiDAR to detect human shins and obstacles around robots (see Fig. 2.2). By using geometric constraints, the study was able to determine where the human shin is, thereby determining the human's position relative to the robot. The quality of the algorithm is at a good level in a simple environment. The main disadvantage of this method is that the information obtained by 2D LiDAR is very little, so it is easy for the algorithm to confuse the human leg with other objects such as the legs of a table, chair or column. So to improve accuracy, people started using 3D LiDAR to replace 2D LiDAR .

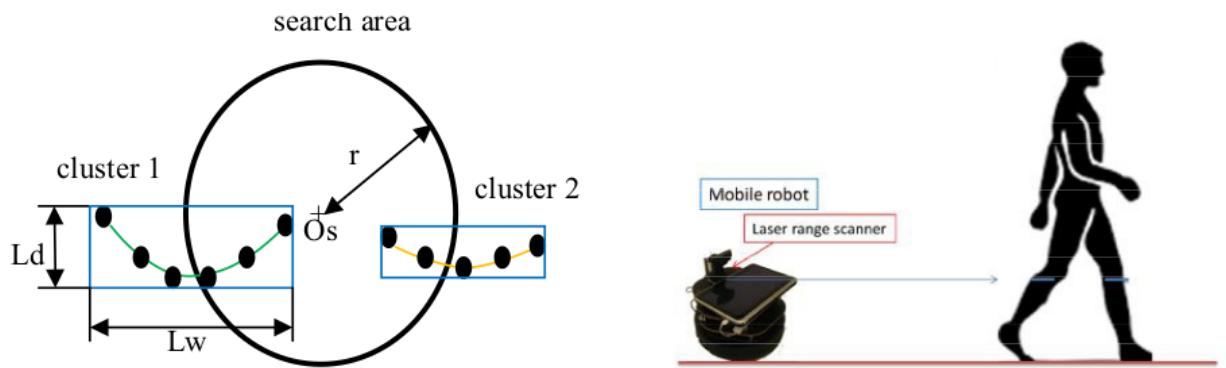


Figure 2.2: Shins Human detection and following using LRS[12]

With 3D LiDAR , the number of point clouds that can scan objects will be much larger than with 2D LiDAR , so one can extract more features from those point clouds. One of the earliest studies on human recognition using 3D LiDAR was by Luis *et al.* [13]. Research using Support Vector Machine (SVM) along with seven features. The downside of the study is that it can only detect people standing, not moving, and over short distances. Then in 2011, Kiyosumi Kidono *et al.* [14]proposed two new features, the Slice feature and the

Distribution of Reflection Intensities. This proposal has somewhat solved the problem of distance and accuracy of detecting people, but there are still disadvantages such as posture or movement of people (see Fig. 2.3). To solve those problems, Zhi Yan *et al.* [15] has inherited previous studies and added a tracker that helps the algorithm to detect and track moving objects. Removing some features helps to speed up computation. And especially, adding an online learning algorithm has solved the problem of accuracy in detecting people, as well as improving the algorithm when it is possible to continuously update new parameters to increase detection ability with training time.

In light of the benefits and drawbacks of the earlier research approaches, we would like to provide a whole pipeline for the HFR system. The pipeline will contain model changes, user detection and tracking for online learning, input preprocessing (point-cloud), and user detection and tracking for input. We utilize PID as the control method to direct the robot to the person's designated position in the previous phase.

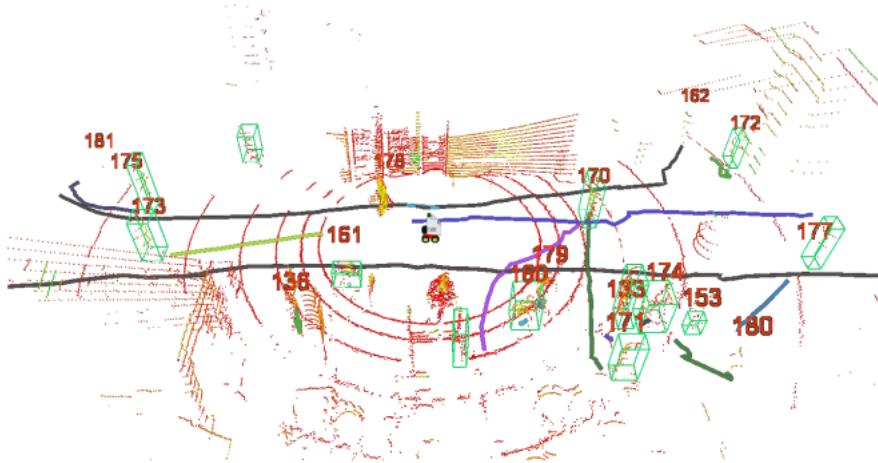


Figure 2.3: Human detection and tracking using Online learning with 3D LiDAR [15]

Chapter 3

Proposed System for Human Following Robot

3.1 Pipeline

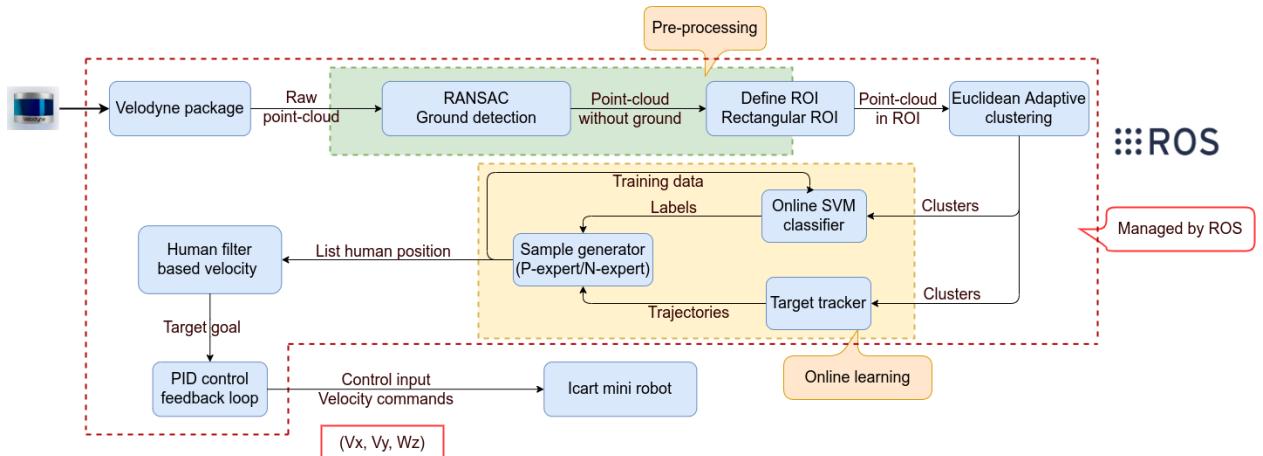


Figure 3.1: A Pipeline of the ROS-based Human Following Robot system

This section describes the proposed pipeline, which is the heart of this thesis. This pipeline takes point cloud obtained from a 3D LiDAR onboard an mobile robot as an input and outputs the final velocity commands for the mobile robot to follow the identified human effectively in an indoor environment. This pipeline is illustrated as a block diagram in Figure 3.1 above. The whole process is utterly managed and coded in ROS (Robot Operating System) framework [16]. In principle, the pipeline can be split into 4 main stages,

which are pre-processing step, clustering step, classification step and control step. The pre-processing step contains RANSAC ground extraction and ROI(Region of Interest) definition, which is presented in section 3.2.1. The clustering step uses Euclidean Adaptive clustering method and is depicted in section 3.2.2. Then, section 3.2.3 describes online learning for SVM(Support Vector Machine) which is the human classification step. Finally, the control phase is discussed in section 3.2.4, which comprises 2 smaller modules: human filter-based velocity and PID control feedback loop. The whole pipeline is written in C++ and maintained in author's Github repository.

3.2 Pipeline breakdown and details

3.2.1 LiDAR point cloud pre-processing step

In this pre-processing step, the ground plane is extracted and removed from the raw point cloud of Velodyne LiDAR using RANSAC algorithm. RANSAC [17] stands for Random Sample Consensus, which is considered to be a congruous technique to detect and filter out any inliers point cloud in any set of point clouds using some pre-defined models such as plane model or circle model. RANSAC algorithm used to detect the plane model is shown in the Algorithm 1. First, the algorithm will select any 3 points in the space randomly to form a plane. Then, the we evaluate the plane model coefficients a, b, c, d that contains those previous selected points.

Algorithm 1 RANSAC(Random Sample Consensus) ground plane detection algorithm [17]

Output: Point cloud data P^* of the plane model

Input: Point cloud data P , where $p_i = [x, y, z] \in P$;

- 1: Randomly select three non-collinear unique points $\{p_i, p_j, p_k\}$ from P ;
 - 2: Compute the model coefficients from the three points ($ax + by + cz + d = 0$);
 - 3: Compute the distances from all $p \in P^* \subset P$ to the plane model (a, b, c, d) ;
 - 4: Count the number of points $p^* \in P$ whose distance d to the plane model falls between $0 \leq |d| \leq |d_t|$, where d_t represents a user specified threshold.
-

Next, distance from all remaining points is calculated to that plane model, as visualised in Figure 3.2. A distance threshold will be set manually and the total number of points whose distances fall within this range will be recorded in each iteration. This process will be looped until the maximum number of points is reached. For a set of 3D point clouds, especially point clouds captured in indoor environment such as corridor or in the building,

there is a large number of points that will belong to the ground. If those point clouds are extracted from the set, the next analysis step will be much faster since the computational time is greatly reduced. Also, filtering out ground point clouds can make the classification step for all points above the ground plane easier. An example of RANSAC application to extract ground point clouds is shown in Figure 3.3. When applying RANSAC, it is assumed that the ground in the experimental environment is completely flat and not rough.

$$Distance = \frac{ax_4 + by_4 + cz_4 + d}{\sqrt{a^2 + b^2 + c^2}}$$

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

Figure 3.2: RANSAC plane distance estimation [18]

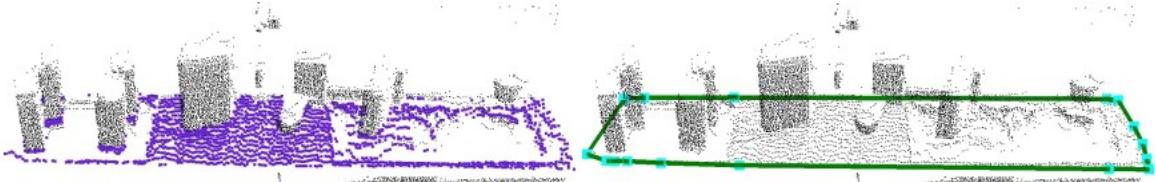


Figure 3.3: Example of resultant inliers (purple) point support best planar model (ground) [19]

After performing ground segmentation, we need one more small step to accomplish this pre-processing step. Point cloud obtained from LiDAR scan, after being removed ground points, will be simplified by creating a region of interest(ROI). To put it differently, we will trim these remaining point cloud by making a box around the mobile robot and get rid of all points outside this boundary box[20]. Robot attached with LiDAR sensor will be the center of this box and in this research, the dimensions of this box is 10 x 10 x 2m because we want to observe full body of human.

3.2.2 Adaptive Euclidean clustering

In this step, point cloud that is above the ground and inside a boundary box will be clustered into multiple separated objects, like the example in Figure 3.4. This clustering method is known as Euclidean object clustering technique [19, 20] and its pseudocode is shown in Algorithm 2. This method is also regarded as prevalent 3D point cloud analysis technique [21]. This algorithm takes point cloud data P as an input and outputs an array of point cloud clusters C . These clusters can contain both human and non-human objects that will be fed to the next classification stage. To implement this algorithm, first kd-tree representation [19] will be created. Array of clusters C and small set of points Q will be first initialised, then we will jump into the main loop of Euclidean clustering method. Point p_i will be added to Q in sequence and in each iteration, another loop is made to search for any point p_i^k within a sphere of radius r . This radius is smaller than an assigned threshold d^* . This nested loop will compile until all suitable points are added to Q and Q is added to C . This algorithm will terminate if all point cloud have been clustered.

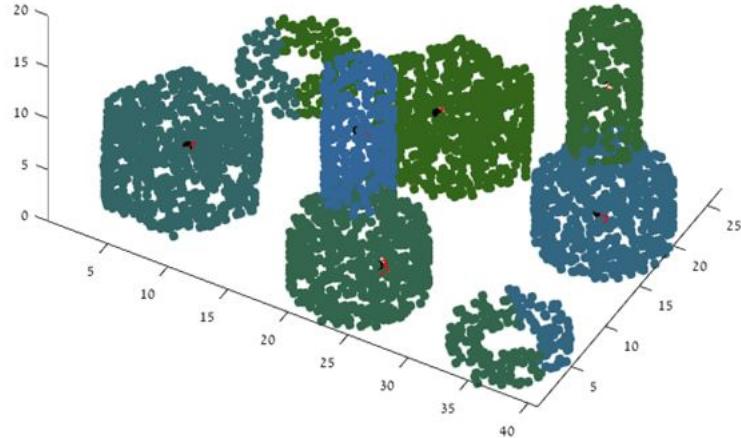


Figure 3.4: Example of Euclidean object clustering [19]

However, although Euclidean algorithm is fast and robust, it has one large disadvantage that should be taken into account. The hindrance of this method lies in the threshold d^* parameter. In fact, the accuracy of the clustering result depends predominantly on the the threshold value. As can be observed in Figure 3.5, if d^* is too large, point cloud obtained from some objects can be clustered together. On the contrary, small d^* leads to the consequence that single target can be divided into multiple clusters. Also, according to [15], the

Algorithm 2 Euclidean clustering algorithm [19, 20]

Output: List of point cloud clusters C

Input: Point cloud data P , where $p_i = [x, y, z] \in P$;

- 1: Create a Kd-tree representation for P ;
 - 2: Setup an empty list of clusters C , and a queue of points to be checked Q ;
 - 3: **for** every point $p_i \in P$ **do**
 - 4: Add p_i to the current queue Q ;
 - 5: **for** every point $p_i \in Q$ **do**
 - 6: Search for the set p_i^k of point neighbours of p_i s.t a sphere with radius r, d^*
 - 7: For every neighbour $p_i^k \in p_i^k$ check if the point has already been processed, else add it to Q ;
 - 8: **end for**
 - 9: When the list of all points in Q has been processed, add Q to the list of clusters C , and $Q \leftarrow \emptyset$
 - 10: **end for**
 - 11: Return the algorithm terminates when all point $\in P$ have been processed;
-

human form generated by LiDAR scan varies enormously with respect to the distance between human and LiDAR sensor, as shown in Figure 3.6 because of LiDAR vertical angular resolution. Thus, we will tune this threshold value using the scan range of LiDAR [15] using this equation below:

$$d^* = 2r \tan \frac{\theta}{2} \quad (3.1)$$

In equation 3.1, r denotes the detected range of LiDAR and θ is the LiDAR vertical angle resolution respectively.

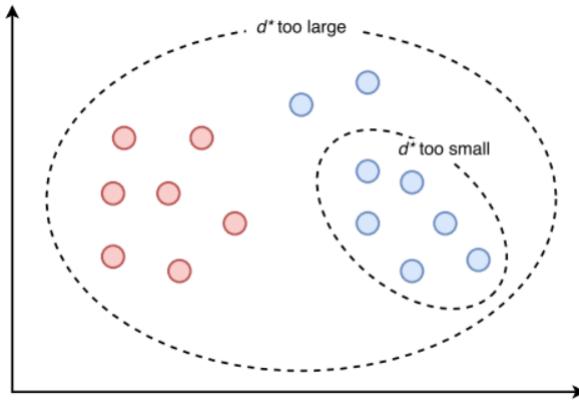


Figure 3.5: Different d^* lead to different clustering results [15, 22]

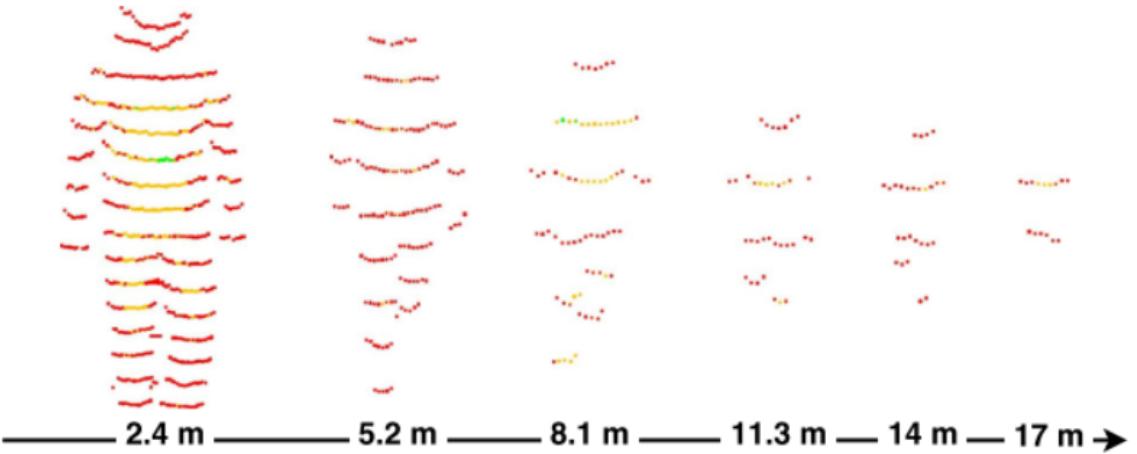


Figure 3.6: Shape of human point clouds with respect to distance from LiDAR [15, 22]

Accordingly for different detection range of LiDAR, an appropriate threshold value will be determined to make the final clustering result more precise compared to the traditional Euclidean method. The result of this clustering step will be shown in section 4.2.1 in Chapter 4 below.

3.2.3 Online learning for SVM human classification

The online learning for SVM human classification method which is the most crucial component of the entire system will be described and analyzed in this section. Figure 3.7 shows the framework of this learning framework. In short, online learning process can be broken into 3 parts: the multi-target tracker, human SVM classifier and P-N sample generator [15].

Each of these part has its own task to complete this online process. The tracker component estimates the position of each cluster in real time and yields all cluster's trajectories. Concurrently, the classifier will do its work - classifies whether the cluster is human or non-human based on the model trained by SVM method. However, the model is not static, it will be updated continuously with the support from the sample generator. This sample generator will be the judge in this scenario. In specific, it will correct positive and negative samples by employing clusters' trajectories information obtained from the tracker and produces new training data for SVM algorithm. The classifier will be retrained again until some terminating conditions are satisfied.

The procedure briefly described above is eminently propitious when handling datasets which are intricate to gather such as human 3D point cloud dataset. Human often has var-

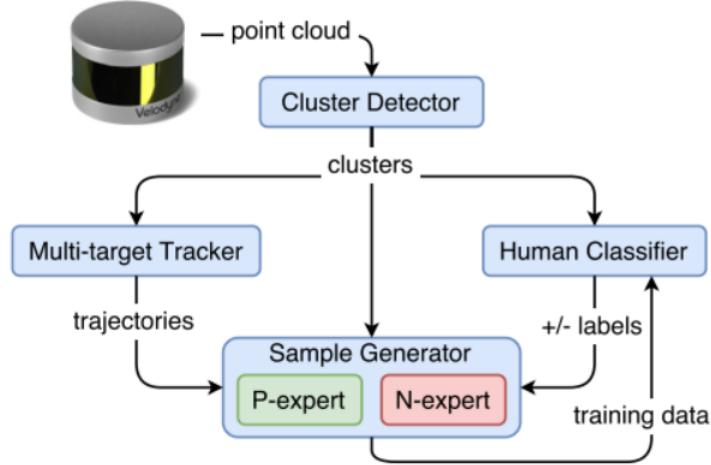


Figure 3.7: Online learning framework [22]

ious poses and different characteristics such as height, body thickness and so on. For that reason, collecting human datasets for training is considered as tough work and normally requires much efforts to make good datasets. However, since some features such as point cloud's reflection intensity are completely distinct, using open-source dataset which is already available on the Internet, namely KITTI is not appropriate. Thus, batch-incremental training [15, 22] as in Figure 3.7 is effective because new human data is collected online when conducting the experiment.

EKF (Extended Kalman Filter) and NN(Nearest neighbor) data association [15, 22, 23] are utilised for the cluster tracking process. Although the data obtained is three-dimensional, only 2D data (i.e x and y coordinates) is used for the tracking. So in this tracking algorithm, human is supposed to move on a flat plane which should be correct in most indoor environment. To use EKF, 2 models including velocity model and obseravtion model will be built. The method is much similar to the traditional Kalman filter, the only difference here is to first linearise all models. Once the cluster is detected and observed, new cluster states are updated using the polar obseravtion model. NN data association will be exploited to operate several perceived clusters simultaneously so that all EKFs are updated in synchronicity [15, 22].

LIBSVM [24] is used to implement the human classifier part of the framework. For this classification task, each human training data has one label and 62 features. The class label can be either 1 (human) or -1(non-human) and all features using to train the human SVM model are shown in Table 3.1 below. The purpose of SVM is to generate a model that can

predict the label of input data, to classify whether this cluster is human or not using only cluster's features information. We add one more feature f_3 or cluster volume, compared to 61 features in [15].

Feature	Description	Dimension
f_1	Number of points in cluster	1
f_2	Minimum cluster distance from LiDAR	1
f_3	Cluster volume	1
f_4	3D covariance matrix of cluster	6
f_5	Normalized moment of inertia tensor	6
f_6	Slice feature of the cluster	20
f_7	Reflection intensity's distribution (mean, std dev)	27

Table 3.1: All features for human classification

In this step, C-SVC (C-Support Vector Classification) of LIBSVM is used. The mathematical concept of C-SVC is constructed as in [24, 25]. For $x_i \in \mathbb{R}^n$ and output label vector $y_i \in \{1, -1\}$, C-SVC optimizes this problem:

$$\begin{aligned} & \min_{\alpha} \quad \frac{1}{2} \alpha^T Q \alpha - e^T \alpha \\ & \text{subject to} \quad y^T \alpha = 0 \\ & \text{with} \quad 0 \leq \alpha_i \leq C, i = \{1, 2, \dots, 62\} \end{aligned} \tag{3.2}$$

In equation 3.2, e is vector of all 1 and Q is 62x62 matrix with $Q_{ij} = y_i y_j K(x_i, x_j)$. K is defined as kernel function with $K(x_i, y_j) = \phi(x_i)^T \phi(x_j)$ [24, 25]. There are several kernel functions has been defined such as linear, polynomoiial, radial basis function (RBF) and sigmoid. Among of them, [25] proposed that RBF kernel should be used. Thus, in this research, the following kernel function will be used:

$$K(x_i, y_j) = e^{-\gamma \|x_i - y_j\|^2} \tag{3.3}$$

where C and γ are 2 parameters which are tuned beforehand.

The sample generator in Figure 3.7 consists P-expert and N-expert. The relation between testing and real human data is shown in Figure 3.8 below.

From Figure 3.8, real human clusters are defined as the sum of TP (True Positive) samples and FN (False Negative) while non-human clusters are the total of FP(False Positive) and TN(True Negative). Evidently, FN and FP are only 2 cases that must be reduced if we want

		Human	
		+	-
Test	+	True Positive (TP)	False Positive (FP)
	-	False Negative (FN)	True Negative (TN)
Real Human = TP + FN		Non - Human = FP + TN	

Figure 3.8: Positive and Negative clusters

better classification accuracy. That is where the role of the sample generator takes place. In specific, all clusters which are categorised as negative samples are fed to the P-expert of the generator, then this P-expert will assess FN samples and renovate them to positive samples. Similarly, all clusters which are considered as postive samples in the initial training are fed to the N-expert and only FP will be transform into negative samples. This new augmented training data will be exploited to retrain the whole human classifier [15]. The way P-expert and N-expert filters out FN and FP samples respectively is genuinely simple. P-expert relies on human-like trajectory whereas N-expert is governed by approximate static objects. In short, there will be some variances conditions that have to be satisfied to be considered whether is is human-like or static samples [15]. The training result will be shown in section 4.2.2.

3.2.4 PID for Robot controller

The online learning stage above will yield human cluster's centered position and this data will be used as input for the PID controller. To successfully follow a human, errors between the state of robot and human must be diminished to the fullest extent. In actual experiment, z-coordinate of both human and robot is not essential for the following step. We will only consider offset angle and distance between the robot and human [10, 11], as illustrated in Figure 3.9.

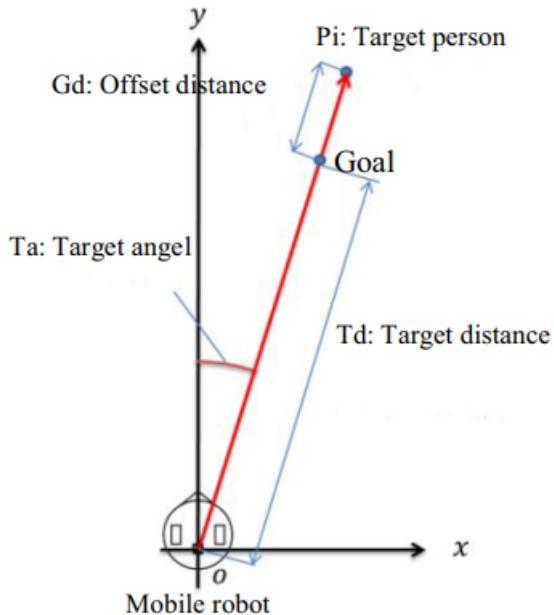


Figure 3.9: Human target distance and direction of mobile robot [12]

General offset angle and distance can be determined as in Equation 3.4. It is worth noting out that in our case, output human cluster's position is with respect to the LiDAR sensor's reference frame. In section 4.1.1, one will see that LiDAR and mobile robot is designed to have same z-axis and same x and y direction. Thus, in 2D format, sensor's frame coincides with robot's frame and x_r and y_r can be set to zero for simplicity. This equation can also be used in global coordinate map with tf library [26] in ROS.

$$d = \sqrt{(x_h - x_r)^2 + (y_h - y_r)^2} \quad (3.4)$$

$$\theta = \tan^{-1} \frac{y_h - y_r}{x_h - x_r}$$

For the following state, PID equation can be used as follows:

$$\begin{pmatrix} v_t \\ \omega_t \end{pmatrix} = \begin{pmatrix} \dot{d}(t) \\ \dot{\theta}(t) \end{pmatrix} = K_p \begin{pmatrix} e_d(t) \\ e_\theta(t) \end{pmatrix} + K_i \int_0^T \begin{pmatrix} e_d(t) \\ e_\theta(t) \end{pmatrix} dt + K_d \frac{d}{dt} \begin{pmatrix} e_d(t) \\ e_\theta(t) \end{pmatrix} \quad (3.5)$$

with distance error $e_d = (d - D)$ and angle error $e_\theta = \theta - \Theta$. They can also be called as state differences , where D and Θ are target distance and angle respectively. In this following step, D is set to be 1m and Θ is around 5° . 3 PID parameters K_p , K_i and K_d are set differently for each testing environment, namely corridor and testing hall (See Chapter 4). These values are tuned by trial and error until reliable result is attained.

However,to ensure the robot follows the exact initial person, the position of the human between 2 consecutive detections will be constrained to be no larger than a maximum distance. We denote r_t as the distance between 2 consecutive recognition and human-filter based velocity is set as follows:

$$\begin{aligned} r_t &= \sqrt{(x_h^t - x_h^{t-1})^2 + (y_h^t - y_h^{t-1})^2} \\ r_t &\leq v_h^{max} \Delta t \end{aligned} \quad (3.6)$$

After setting constraints as in Equation 3.6, target goal or target human's position is fed to the PID feedback loop as in Equation 3.5 above. This completes the whole process of the proposed pipeline. Final pipeline result is presented in section 4.2.3.

Chapter 4

Experimental Setup and Results

4.1 Experiments Platform

4.1.1 Experiments Design

Icart-mini [27] and VLP-16 LiDAR [28] are the robot and sensor used in this investigation. A laptop will be mounted on the robot, it connects to the LiDAR, does processing and calculations, and issues orders to the robot. Icart-mini is a T-frog robot that has a simple design and can be moved both indoors and outdoors. This robot template is available for free. The supplier has offered a variety of packages to help users create apps for Icart-mini. Being the smallest in the Velodyne product range, the VLP-16 LiDAR scanner, was developed at a low cost with mass production in mind, notably for robots, autonomous cars, and industrial automation.

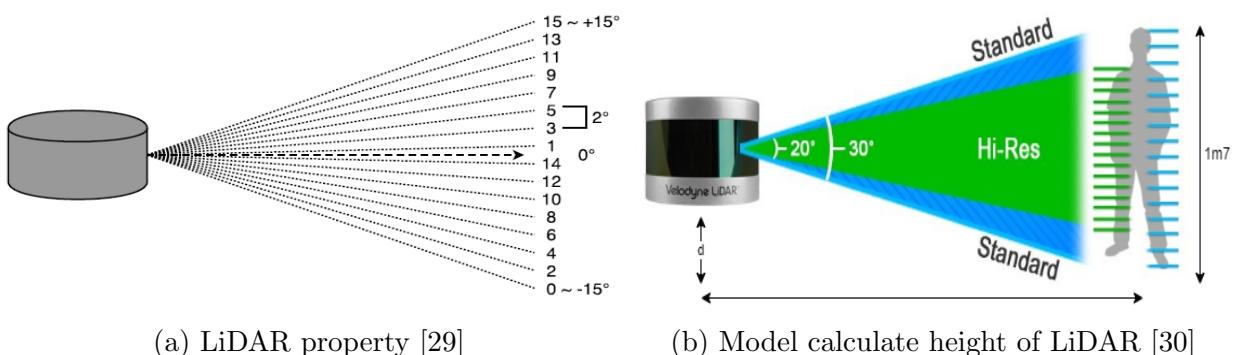


Figure 4.1: VLP-16 LiDAR vertical field of view

Item	Specifications
Scanning rate	10 scans/s
Horizontal field of view	360°
Horizontal angular resolution	0.23°
Vertical field of view	26.8°
Vertical angular resolution	2° (16 lines)
Detection range	40m for pavement, 120m for cars and foliage
Range accuracy	0.02m
Wavelength of laser beam	905nm

Table 4.1: 3D VLP-16 LiDAR properties [28]

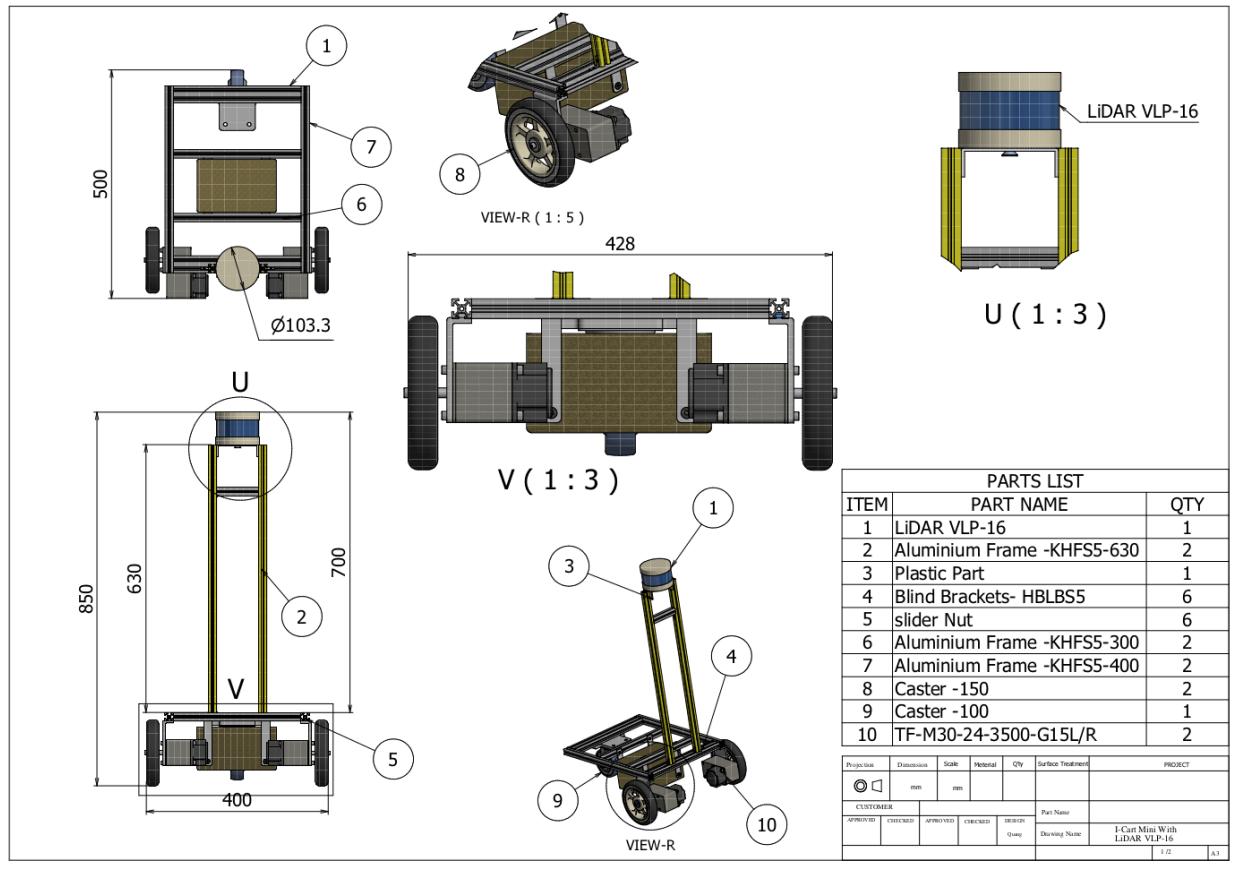


Figure 4.2: Design of VLP-16 LiDAR onboard Icart-mini robot

Figure 4.1a and Table 4.1 show the scan properties of VLP-16. From these data, one knows that VLP-16 has a total of 16 scans. Each scanning ray is aligned at an angle of 2 degrees. The total vertical field of view is 30 degrees. The scan range that is considered

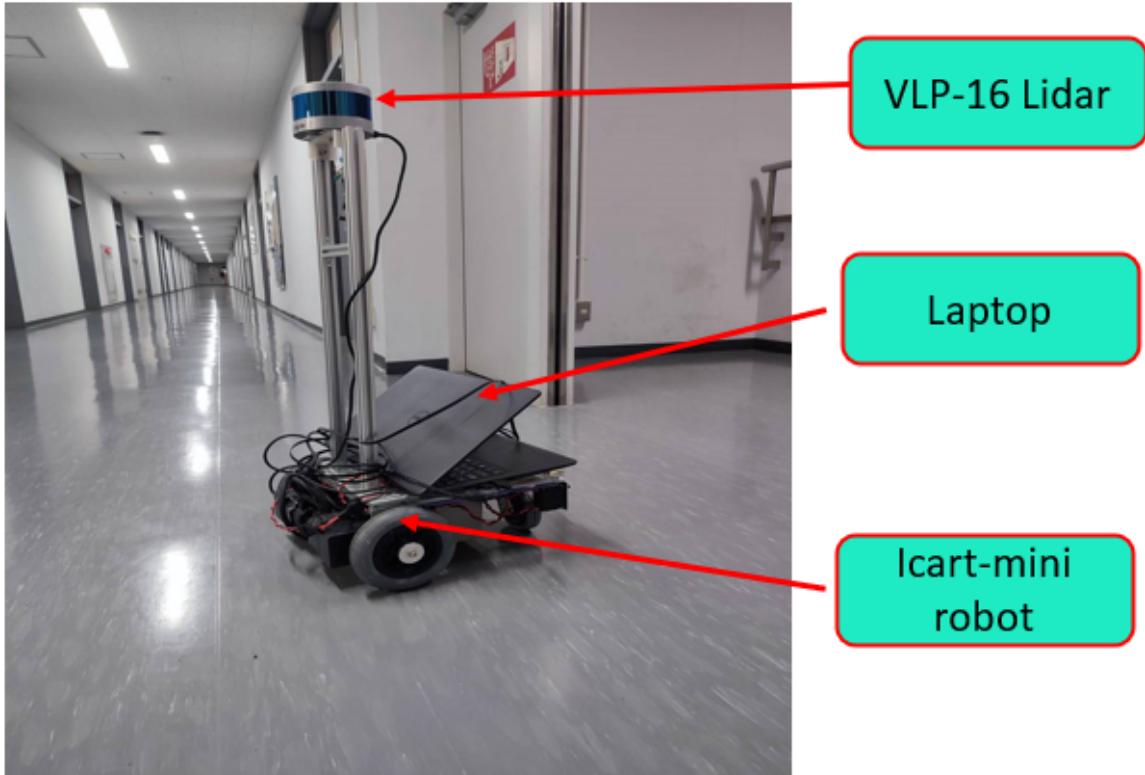


Figure 4.3: Final experiments design in real life

to be the most detailed will have a scan range of 20 degrees and a standard of 30 degrees. Consequently, in order to obtain the most complete and accurate information about people, we need to place the LiDAR at a certain distance from the ground. The height calculation model is shown as Figure 4.1b below. Assume the minimum distance for the robot to recognize people is at least 2m from the robot. The average height of a person is 170cm. From the model, we can calculate that the height of the LiDAR will be approximately 80cm from the ground. After calculating the required height for the LiDAR, we proceed to fabricate and assemble the necessary parts to mount the LiDAR on the robot. The robot model equipped with LiDAR is shown in Figure 4.2. We used aluminum rods buying from *Misumi* to connect Icart-mini and VLP-16. The height from the ground to the top of the LiDAR is 85cm. Figure 4.3 displays the full robot model with all of its components. A laptop with Ubuntu Operating System will be the processor of the robot. In other words, this laptop will be put on top of Icart-mini and acts as the mind of the robot. In this laptop, ROS Noetic 20.04 was installed to run the pipeline.

4.1.2 Experiments Setup and Simulations

In this study, we experimented with the system of robot following people in three environments: simulation (Gazebo environment), corridor and hall environment. To greatly increase comparability, and make the research project reusable for other researchers, which indicate that the experiment can be completely recreated without the necessity to rearrange the real life experiment, we decided to build all models in Gazebo simulated environment .With the Gazebo environment, it is used to train the model, test the PID algorithm and the tracking algorithm.

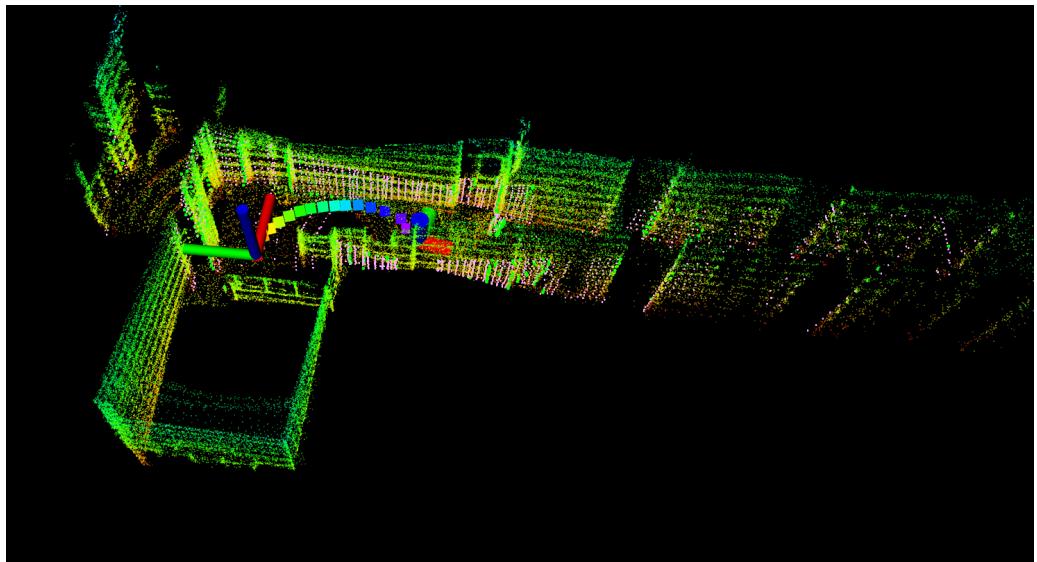


Figure 4.4: Corridor 3D Map

First, for the Gazebo simulation environment, we created an environment that was structured like a hallway, which we later used for experiments. In order to create the corridor environment as close to reality, we use the ALOAM algorithm [31, 32] to scan the corridor area map as illustrated in Figure 4.4. As we showed in the previous workshop, ALOAM is a fairly accurate map scanning algorithm and gives fast results. After creating the corridor map, we extracted the images of the map and imported them into Gazebo software to create the environment in 1:1 size. In addition, we also added a human object environment to train the robot as well as test the algorithm. This object will be defined with a certain and repetitive motion trajectory in the environment. Simulated walking human using waypoint generator is indicated in Figure 4.5.

For experiments conducted in the real environment which includes the corridor and the

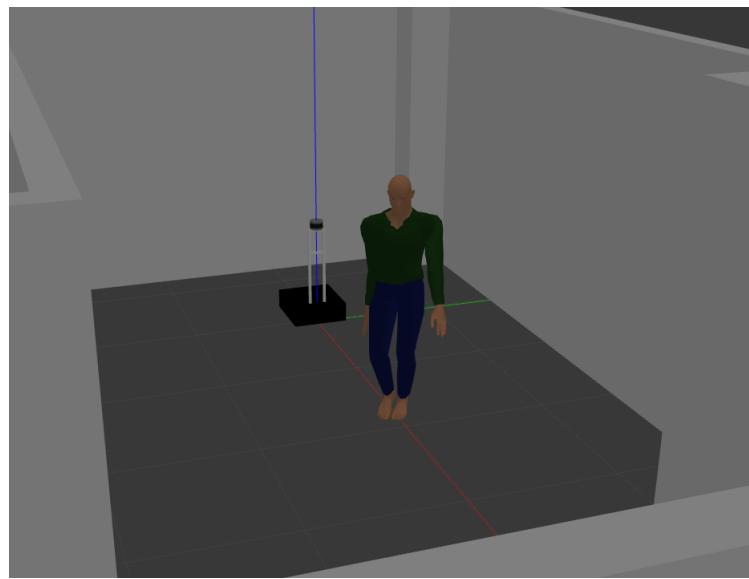


Figure 4.5: Human walking simulation using waypoint generator

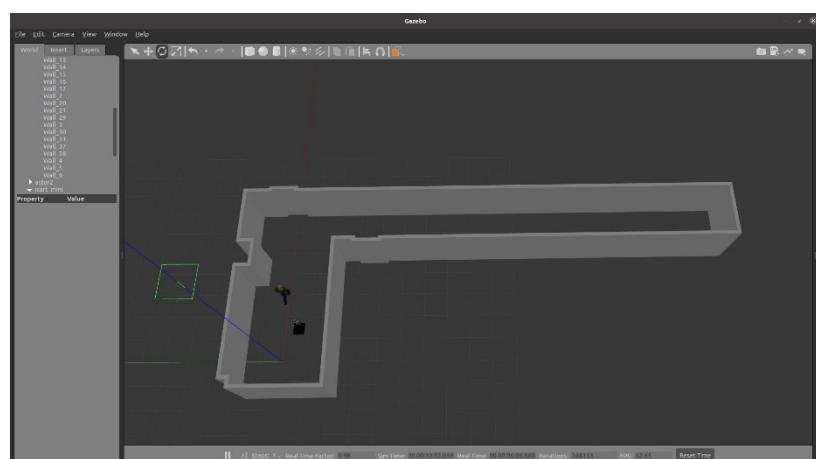
aerospace testing hall, robot as in Figure 4.3 is placed with one participant such as the author himself or author's friend from laboratory. All experiments are illustrated in Figure 4.6. More specifically, Figure 4.6a shows the experiment in the corridor, Figure 4.6b shows the experiment in the aerospace testing hall and Figure 4.6c shows the experiment in the simulated Gazebo environment respectively.



(a) Corridor environment



(b) Testing hall environment



(c) Gazebo environment

Figure 4.6: Experiment setup in 3 cases

4.2 Experiments Results

4.2.1 Clustering result

This section shows the outcome of point cloud clustering whose method was described in section 3.2.2. There are more objects in the aerospace testing hall than in the corridor so the efficiency of the Adaptive clustering technique is evident in the case of hall environment. For point cloud sets of the same size as a standard human, we can observe that they are clustered flawlessly without any errors whereas bigger objects in the hall (Figure 4.8) are sometimes being divided into smaller objects. This is reasonable because their dimensions belong to various LiDAR's horizontal detected range. Walls in the corridor also behave in the similar way, as shown in Figure 4.7.

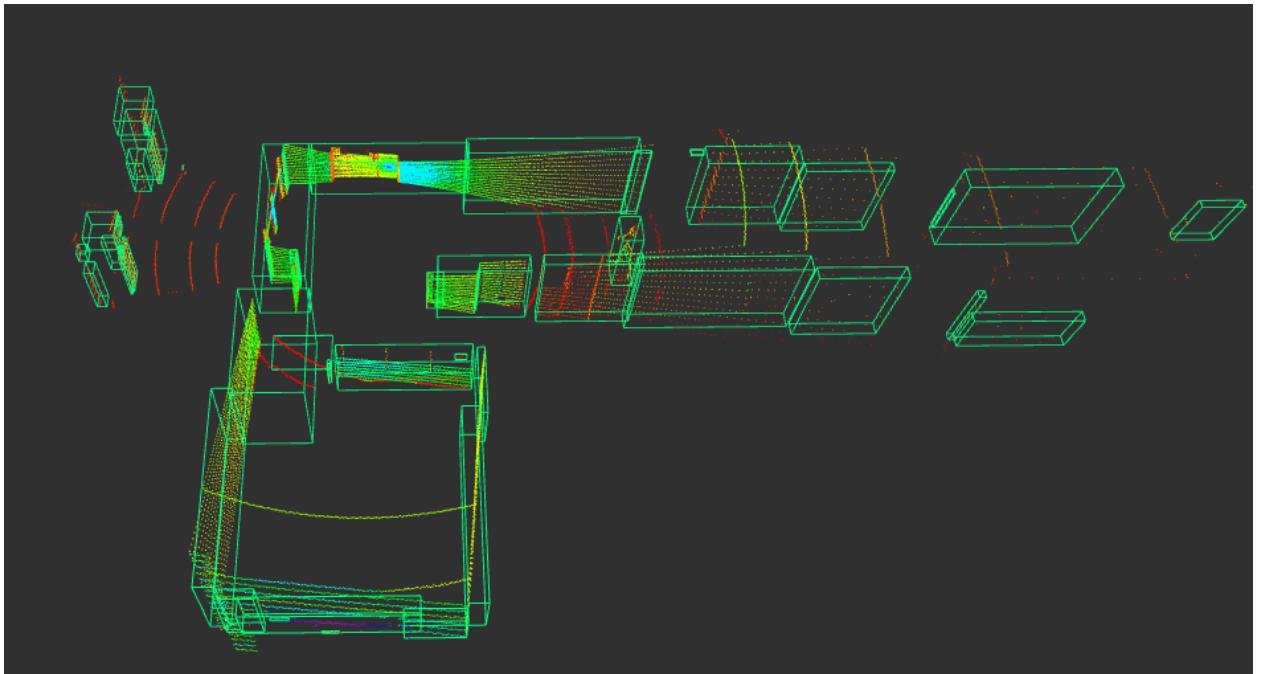


Figure 4.7: Corridor clustering result

Some objects which are outside the window and not inside the corridor are even clustered because of the wide detection range of VLP-16 LiDAR. Those objects are split into multiple smaller clusters because they are too far away from the sensor, however, it is not a main issue of the clustering method. This is because in this research the distance between human and robot is constrained to be not larger than a maximum value (see section 3.2.4) to ensure that human is always clustered into only one cluster and also the inaccuracy of the classification

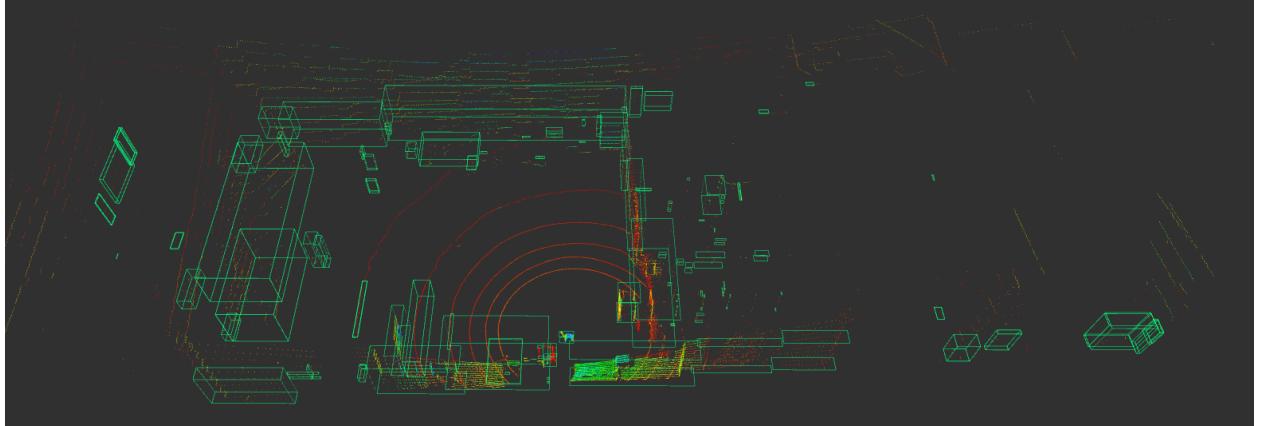


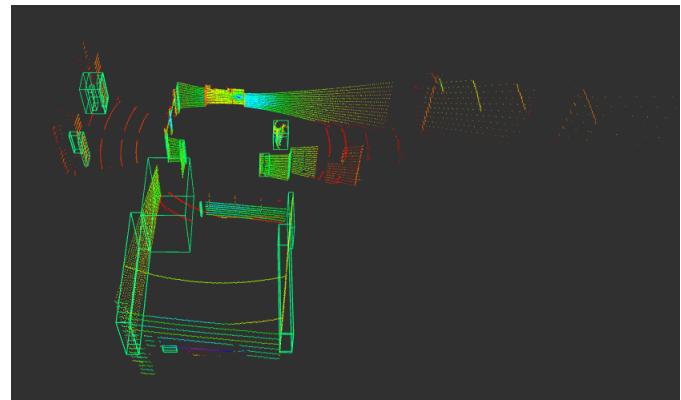
Figure 4.8: Testing hall clustering result

will reduce greatly. Hence, as long as human is clustered correctly, this cluster will be treated as positive samples for the next online training step. To sum up ,it is appropriate to assert that the performance of the adaptive Euclidean clustering in this HFR experiment is acceptable.

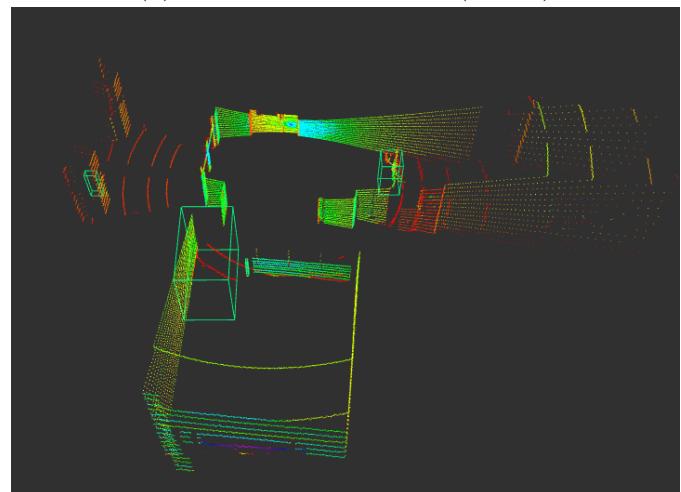
4.2.2 Online training for SVM result

This section displays the consequence of online training whose method was detailed in section 3.2.3. This training can either be done in two ways: by directly training with robots or by recording point cloud into *bag* file and then train later. Both ways will still generate the same result as in Figure 4.9. The result of the online training result can be analysed as follows. First, the SVM human classifier is trained with a small initial training dataset and shows a poor classification result as in Figure 4.9a. After that, the P-N generator will correct wrong samples as described in section 3.2.3 and retrain the whole classifier for better classification result. Then in round 3, we can observe from Figure 4.9b that now there are only 2 clusters are being classified incorrectly as human. When the final training round (round 5) is reached, there are no more mis-classified clusters and human cluster is correctly classified, as indicated in Figure 4.9c.

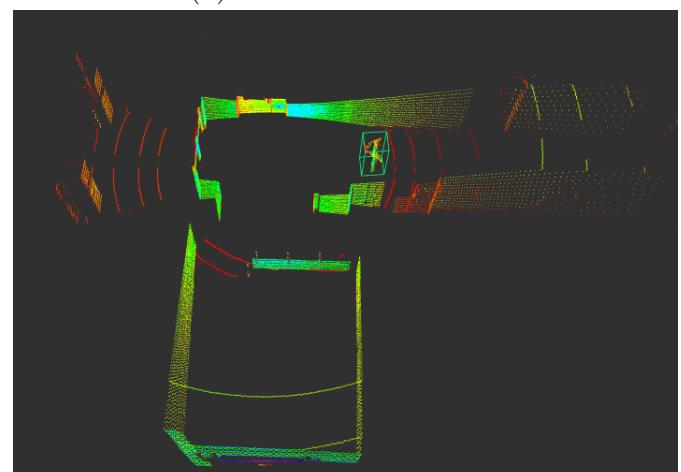
It is worth noting that the result of the online human classifier is mainly affected by the decision of the P-expert and N-expert of the generator in the online learning framework. As long as all assumptions and conditions such as human-like trajectory and static objects mentioned in section 3.2.3 are still satisfied, online human classifier will perform effectively as in Figure 4.9.



(a) Online train round 1 (initial)



(b) Online train round 3



(c) Online train round 5

Figure 4.9: Online training result in corridor environment

4.2.3 Pipeline testing result

This section demonstrates the testing result of the whole pipeline whose image was shown in section 3.1.

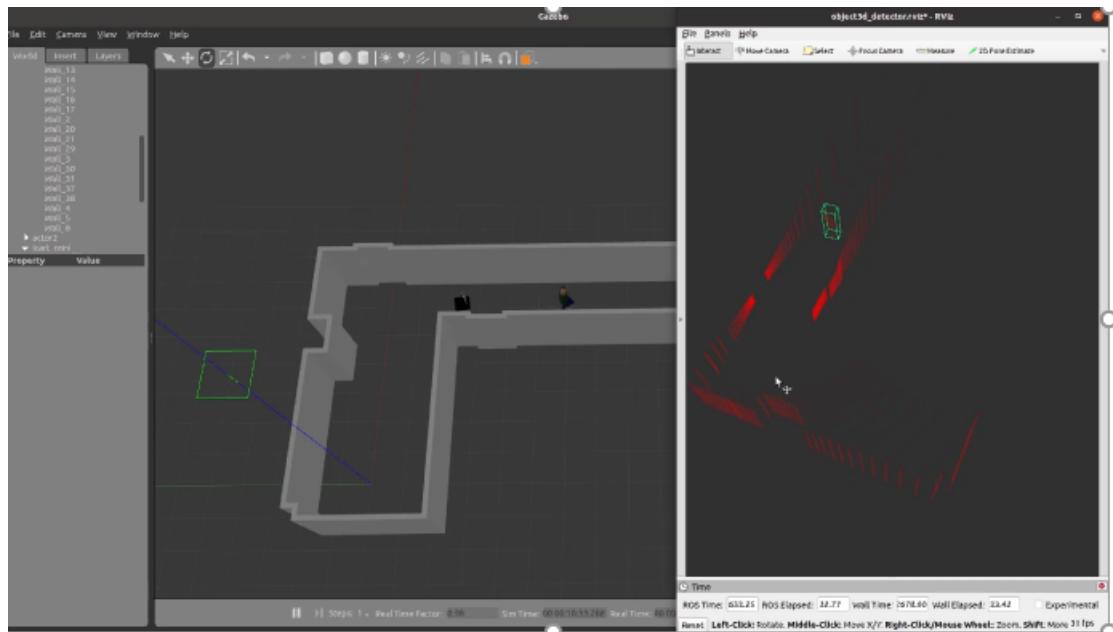


Figure 4.10: Pipeline testing in Gazebo environment

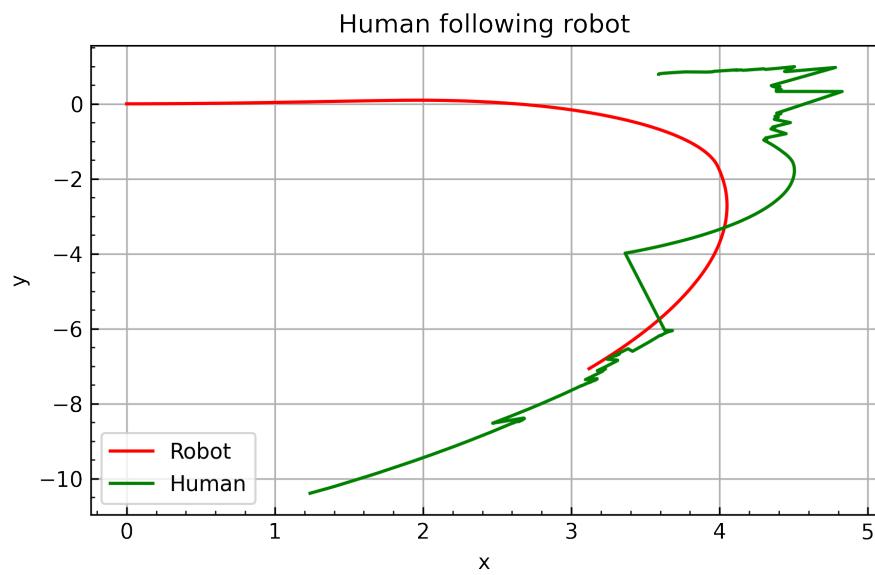
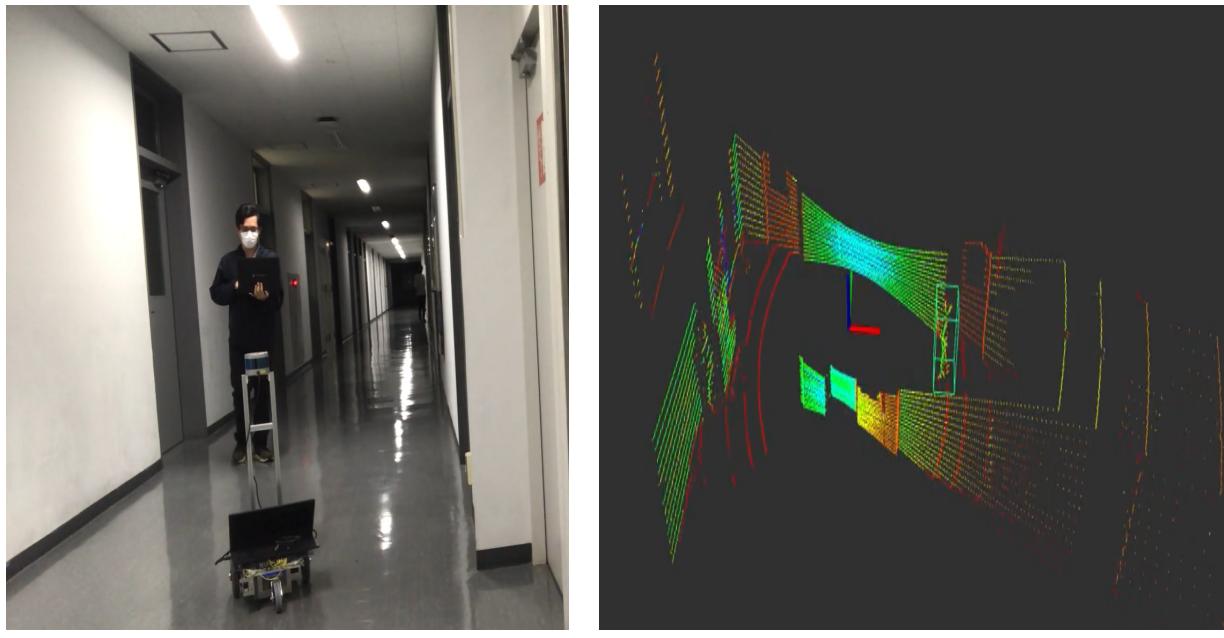


Figure 4.11: HFR 2D path in Gazebo simulated environment



(a) HFR in real corridor

(b) Corresponding time in Rviz

Figure 4.12: Pipeline testing in corridor environment

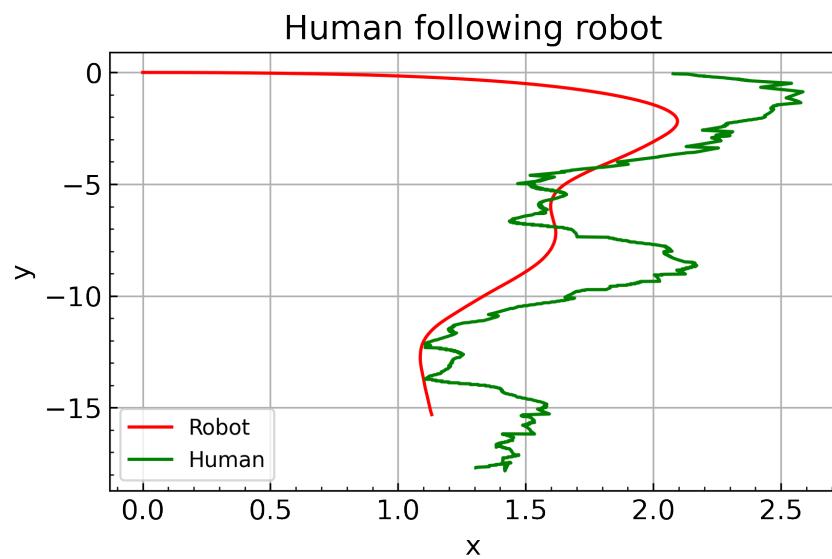
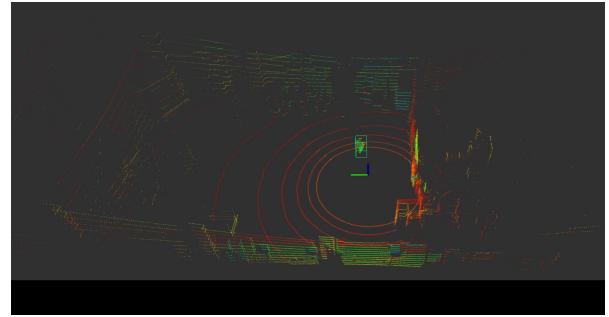


Figure 4.13: 2D path in Nagoya University Engineering Building 2 4th floor corridor



(a) HFR in Testing hall



(b) Corresponding time in Rviz

Figure 4.14: Pipeline Test in aerospace hall environment

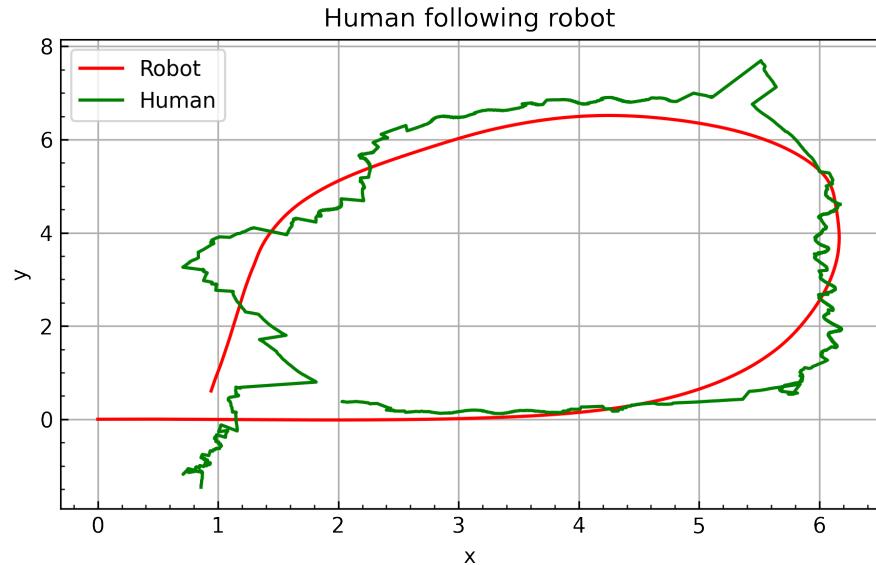


Figure 4.15: 2D Path in Nagoya University 's Aerospace Testing hall

Figure 4.10 and 4.11 show the pipeline testing result in the Gazebo simulated environment. Figure 4.12 and Figure 4.13 illustrate the result in the corridor while Figure 4.14 and Figure 4.15 show the result in the testing hall respectively. In particular, each pair of these Figures first provides the following result in each environment with its corresponding time in Rviz (ROS Visualiation) in the first figure and manifests the path of both robot and human in the 2D plane in the second figure. Rviz image describes the human detection part of the pipeline by surrounding human body with a green box in each frame. The 2D plot describes the tracking and following part of the pipeline with green path for human and red path for robot. In Gazebo simulation case, the robot can detect the simulated walking human and follow human's trajectory in the simulated corridor. In this simulation, we only made the human walk in a straight path then turn right at the corner and then go straight again. The robot follows him and creates a red curve as in Figure 4.11. We can observe that the detection is not good in the simulated case with many lost target moments expressed by a green large zigzag and squiggly line. This is because the LiDAR's intensity simlation in Gazebo is not similar as in real life and in SVM classification, it requires 27 dimensions of LiDAR reflection intensity (see Table 3.1) to train the human model. In fact, simulating LiDAR intensity is still complicated and demands more research work in the future. However, despite that, the inaccuracy of human detection algorithm in the simulated case does not affect the following process because in the PID step we have added the human based velocity filter so that even the robot loses human target or misclassifies human in some frames, the robot can still re-find human in next frame and follow human smoothly. This happens in both other cases, in the corridor and in the testing hall. In the case of the corridor, the human even tried to change the direction of movement continuously, from left to right then from right to left and so on. The robot followed him smoothly and generated a red curve that is almost identical in shape to the human's green trajectory, as indicated in Figure 4.13. In the hall case, the robot even detected and followed the human and made a close loop path as in Figure 4.15. This confirms that the pipeline works well in both corridor case and testing hall case. The robot can still find the person even when the robot misclassifies other objects during the following process. We also notice that the robot's route is a red smooth curve while human's path is a rough green curve in all 3 cases. The reason for that is because the robot's path is directly calculated by its wheel encoder whereas evaluating human's position in each frame took time to operate and transfer to the global coordinate. In all 3 test cases, we make the assumption that there are not any obstacles that can prevent the robot from following human. This is a drawback of this pipeline and will be discussed in Chapter 5.

Chapter 5

Conclusions and Future Work

5.1 Summary and Conclusion

Making robots that follow people is one of the most exciting uses of robotics. Long-term development can be done from there to provide other uses. At the moment, there are few studies on employing robots to track a single object, and these studies have mixed results. The bulk of problems with monitoring people or objects nowadays, in particular, use cameras to collect data. It is simple to lose track of things due to the camera's still constrained information-collection range and limitations in many complex environmental situations. I decided to choose HFR system development as my thesis topic as a result of these factors. My study for a graduation project specifically researches for the employment of a 3D LiDAR user-following robot system.

There are several challenges in the study process because there have not been many studies targeting at utilizing 3D LiDAR to identify and track individuals. In contrast to utilizing a camera, 3D LiDAR finds fewer characteristics in RGB or RGB-D pictures, making it more difficult to identify humans. We had trouble locating and choosing characteristics that can be employed in the robot system that follows people comparing to using camera in the development of the point-cloud user recognition algorithm since not all characteristics that are calculated using collected point-clouds are useful. Because that feature does not yet exude quiet, some algorithm will lengthen calculation times and some will produce noise during the detection. The outcomes of training and testing are also poor since there is no environment and training data. We have not been able to test in scenarios with more complexity. Even if there are still many obstacles to overcome, we continue to work on finishing the research

since it is a promising and novel area that requires further study in order to be effectively applied to human life.

The HFR system with 3D LiDAR is proposed in the thesis, with the 3 key components being SVM (online learning) for human tracking and detection, human filter based velocity as well as PID control for guiding the robot to the user's position. We based on earlier research for the human identification and tracking section, which uses online learning and a tracker to be able to spot individuals moving. In order to increase the accuracy of persons detection, we also introduced a new feature which is the volume feature to the SVM model.

We have added RANSAC, which removes the point-cloud as a plane, and a boundary filter, which is the ROI around the robot, before the human identification section in order to increase the processing performance for the online learning in a certain time period. In specific, only everything within 10 meters of the robot should be taken into account and we remove any extraneous objects from the ROI area. Both actions can save processing time and eliminate noise. We tested the HFR pipeline in 3 different environments (Gazebo environment, university's corridor, and testing hall) and this pipeline performed well enough in all testing cases.

Our study still has some limitations because of the limited research time and the lack of expertise and understanding in robotics. Robots can only follow people at the moment and they are also unable to go around barriers. In other words, robots can only follow the human position without avoiding obstacles and if the environment has multiple people, especially when people are standing close to each other as groups, the robot can not distinguish which person to follow. The robot's movement is still fairly sluggish, and it does not react strongly to objects that move quickly, particularly in areas with plenty of people and items arranged side by side. If they are too close together, single items in online learning are difficult to distinguish.

5.2 Future Work

Since the HFR system employing 3D LiDAR still has numerous limitations, more study into this area may be beneficial. By integrating 3D LiDAR with the camera, for instance, it is easier to recognize several individuals standing near to one another or people standing next to another item. LiDAR offers precise 3D geometry, whereas cameras record additional contextual data [33]. In short, we want to fuse other sensors to add more necessary features and better deal with challenging situations such as groups mentioning in section 5.1.

With regards to the issue of navigation and control, the research still only uses a basic PID algorithm to move the robot to a person's position. To avoid colliding with, harming, or hurting the surrounding environment or the robot itself, the robot must be able to avoid obstacles in the fully autonomous problem. As a result, the integration of algorithms such as A*, D*, RRT, and others [34] should be researched in order to improve and enhance the HFR system in addition to obstacle avoidance. In other words, research works require combining obstacles avoidance to make the robot follow the human more efficiently.

In the realm of control, PID is still a reliable and straightforward control algorithm. The study's findings, however, show that PID is still not particularly adept at tracking individuals. The future research can utilize MPC instead of PID to follow the path from the robot to the location of the human without the robot following straight to the position of the person as in the current study since MPC has numerous notable benefits over PID [35].

Chapter 6

Acknowledgements

In the process of writing this thesis, I have received a great deal of support and assistance. I would first like to thank my supervisors, Professor Naoki Akai and Professor Susumu Hara, whose expertise was invaluable in the formulating of the research topic and methodology in particular. Professor Hara, thank you for giving me a valuable opportunity to study and research in your lab as an undergraduate researcher. Professor Akai, thank you for introducing me to ROS, guiding me in my first tentative steps in the Linux world and have helped develop a stronger desire to continue to pursue research in the robotics field.

I would love to thank my family in Vietnam, especially my father - Quan Le, my mother - Thinh Tran and my sister - Ha Le, for always supporting me whenever I need throughout this journey. I would love to thank my senpai - Truong Phan, for his continuous guidance and support during my 4 years at Nagoya University. I would also love to thank my kouhai - Thanh Nguyen for helping me filming my experiments. Finally, I would like to thank all lab colleagues and seniors in the laboratory, especially the ARIC team with Arashi-san, Yasui-san and Kane-san.

Chapter 7

References

- [1] Saiidi Uptin. *We tested out this hotel robot (and it wasn't totally competent)*. URL: <https://www.cnbc.com/2018/02/21/we-tested-out-this-hotel-robot-in-singaore.html>. (accessed: 15.6.2022).
- [2] Tillman Maggie. *Real-life robots that will make you think the future is now*. URL: <https://www.pocket-lint.com/gadgets/news/134820-real-life-robots-that-will-make-you-think-the-future-is-now>. (accessed: 10.6.2022).
- [3] Da-sol Kim. *E-mart unveils autonomous shopping cart Eli for test run*. URL: <https://m.koreaherald.com/view.php?ud=20180417000718>. (accessed: 15.6.2022).
- [4] Wright Chris. *How Cleaning Robots are Supporting Airports During COVID-19*. URL: <https://www.aviationpros.com/airports/airport-technology/article/21159420/how-cleaning-robots-are-supporting-airports-during-covid19>. (accessed: 15.6.2022).
- [5] Moltzau Alex. *AI, Sensors and Robotics*. URL: <https://towardsdatascience.com/ai-sensors-and-robotics-882caae34df9?gi=59e5ca98650>. (accessed: 10.6.2022).
- [6] Autocrypt. *Camera, Radar and LiDAR: A Comparison of the Three Types of Sensors and Their Limitations*. URL: <https://autocrypt.io/camera-radar-lidar-comparison-three-types-of-sensors/>. (accessed: 10.6.2022).
- [7] Schlegel Christian et al. “Vision Based Person Tracking with a Mobile Robot”. In: *Proceedings of British Machine Vision Conference*. 1998. DOI: 10.5244/C.12.42.

- [8] Haojie Li et al. “Object detection based on color and shape features for service robot in semi-structured indoor environment”. In: *International Journal of Intelligent Robotics and Applications*. 2019. doi: <https://doi.org/10.1007/s41315-019-00113-3>.
- [9] Qimin Ren et al. “Real-time target tracking system for person-following robot”. In: *2016 35th Chinese Control Conference (CCC)*. 2016, pp. 6160–6165. doi: 10.1109/ChiCC.2016.7554324.
- [10] Bao Xin Chen, Raghavender Sahdev, and John K. Tsotsos. “Integrating Stereo Vision with a CNN Tracker for a Person-Following Robot”. In: *Computer Vision Systems*. Ed. by Ming Liu, Haoyao Chen, and Markus Vincze. Cham: Springer International Publishing, 2017, pp. 300–313. ISBN: 978-3-319-68345-4. doi: 10.1007/978-3-319-68345-4_27.
- [11] Redhwani Algabri and Mun-Taek Choi. “Deep-Learning-Based Indoor Human Following of Mobile Robot Using Color Feature”. In: *Sensors* 20.9 (2020). ISSN: 1424-8220. doi: 10.3390/s20092699. URL: <https://www.mdpi.com/1424-8220/20/9/2699>.
- [12] Noriyuki Kawarazaki, Lucas Tetsuya Kuwae, and Tadashi Yoshidome. “Development of Human Following Mobile Robot System Using Laser Range Scanner”. In: *Procedia Computer Science* 76 (2015). 2015 IEEE International Symposium on Robotics and Intelligent Sensors (IEEE IRIS2015), pp. 455–460. ISSN: 1877-0509. doi: <https://doi.org/10.1016/j.procs.2015.12.310>. URL: <https://www.sciencedirect.com/science/article/pii/S1877050915038119>.
- [13] L. E. Navarro-Serment, C. Mertz, and M. Hebert. “Pedestrian detection and tracking using three-dimensional ladar data”. In: *Proceedings of the 7th Conference on Field and Service Robotics (FSR)*. 2009, pp. 103–112.
- [14] Kiyosumi Kidono et al. “Pedestrian recognition using high-definition LIDAR”. In: *2011 IEEE Intelligent Vehicles Symposium (IV)*. 2011, pp. 405–410. doi: 10.1109/IVS.2011.5940433.
- [15] Zhi Yan, Tom Duckett, and Nicola Bellotto. “Online learning for human classification in 3D LiDAR-based tracking”. In: *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vancouver, Canada, Sept. 2017, pp. 864–871.
- [16] Morgan Quigley et al. “ROS: an open-source Robot Operating System”. In: *ICRA Workshop on Open Source Software*. 2009.

- [17] M. Fischler and R. Bolles. “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography”. In: *Communications of the ACM*. 1981, vol.24, no.6.
- [18] RaJ AJith. *3D RANSAC Algorithm for Lidar PCD Segmentation*. URL: https://medium.com/@ajithraj_gangadharan/3d-ransac-algorithm-for-lidar-pcd-segmentation-315d2a51351. (accessed: 15.6.2022).
- [19] R. B. Rusu. “Semantic 3D object maps for everyday manipulation in human living environments”. In: Computer Science department, Technische Universitaet Muenchen, Germany, 2009.
- [20] Jack N.C. Hayton et al. “CNN-based Human Detection Using a 3D LiDAR onboard a UAV”. In: *2020 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. 2020, pp. 312–318. DOI: [10.1109/ICARSC49921.2020.9096075](https://doi.org/10.1109/ICARSC49921.2020.9096075).
- [21] Liu Shan et al. *3D Point Cloud Analysis*. 2021. DOI: <https://doi.org/10.1007/978-3-030-89180-0>.
- [22] Zhi Yan, Tom Duckett, and Nicola Bellotto. “Online learning for 3D LiDAR-based human detection: experimental analysis of point cloud clustering and classification methods”. In: *Auton Robot*. 2020, pp. 147–164.
- [23] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. America: MIT Press, 2005.
- [24] Hsu Chih-Wei, Chang Chih-Chung, and Lin Chih-Jen. “LIBSVM: A library for support vector machines”. In: *ACM Transactions on Intelligent Systems and Technology*. 2011, pp. 1–27.
- [25] Hsu Chih-Wei, Chang Chih-Chung, and Lin Chih-Jen. “A Practical Guide to Support Vector Classification”. In: Taipei, Taiwan, May 2016.
- [26] Tully Foote. “tf: The transform library”. In: *2013 IEEE Conference on Technologies for Practical Robot Applications (TePRA)*. 2013, pp. 1–6. DOI: [10.1109/TePRA.2013.6556373](https://doi.org/10.1109/TePRA.2013.6556373).
- [27] *I-Cart mini*. URL: https://t-frog.com/products/icart_mini/. (accessed: 20.11.2021).
- [28] *Velodyne Lidar*. URL: <https://velodynelidar.com/products/puck/>. (accessed: 20.11.2021).

- [29] M Okunsky and N Nesterova. “Velodyne LIDAR method for sensor data decoding”. In: *IOP Conference Series: Materials Science and Engineering* 516 (Apr. 2019), p. 012018. DOI: 10.1088/1757-899X/516/1/012018.
- [30] URL: <https://robu.in/product/velodyne-puck-hi-res-lidar-sensor/>. (accessed: 30.7.2022).
- [31] T. Shan and B. Englot. “LeGO-LOAM: Lightweight and GroundOptimized Lidar Odometry and Mapping on Variable Terrain”. In: *Proc. of International Conference on Intelligent Robots and Systems (IROS)* (2018).
- [32] Ji Zhang and Sanjiv Singh. “LOAM : Lidar Odometry and Mapping in real-time”. In: *Robotics: Science and Systems Conference (RSS)* (Jan. 2014), pp. 109–111.
- [33] Huazan Zhong et al. “A survey of LiDAR and camera fusion enhancement”. In: *Procedia Computer Science* 183 (2021). Proceedings of the 10th International Conference of Information and Communication Technology, pp. 579–588. ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2021.02.100>. URL: <https://www.sciencedirect.com/science/article/pii/S1877050921005767>.
- [34] Márcia M. Costa and Manuel F. Silva. “A Survey on Path Planning Algorithms for Mobile Robots”. In: *2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. 2019, pp. 1–7. DOI: 10.1109/ICARSC.2019.8733623.
- [35] Innovations Carnot. *Model Predictive Control vs. Proportional, Integral, Derivative Control*. URL: <https://www.carnot-innovations.com/post/model-predictive-control-vs-proportional-integral-derivative-control>. (accessed: 30.7.2022).