CAPSTONE PROJECT 1

Planning Document

PRJ3213 CAPSTONE PROJECT 1

by

Adam Lo Jen Khai

21007364

BCs Computer Science

Supervisor: Dr Steven Eu Kok Seng

Semester: April 2023

Date: 11 September, 2023

Department of Computing and Information Systems

School of Engineering and Technology

Sunway University

Table of Contents

Contents

CHAPTER 1 - INTRODUCTION

Humanity has been fascinated by robots for thousands of years; they are depicted in Greek and Egyptian mythology. The desire to 'make' something living or lifelike out of inanimate materials is at the foundation of this fascination. The tale of the golem, an anthropomorphic living creature made (often) out of clay or mud, is among the most well-known examples. Movie robots like the Terminator, for instance, could be credited for sparking this evolution. A highly developed artificial intelligence device with (almost) humanoid qualities, The Terminator (2004), has the ability to shift into human doppelgangers that resemble human personalities, but with considerably greater (and mostly bad) abilities. Until recently, there has been a trend in popular media and science communication to combine many technical breakthroughs into one concept, namely robotics, including robots, AI, and autonomous systems like self-driving cars [30].

The last few of years have seen a noticeable increase in robotic technology. A few years ago, such advancements were still something out of science fiction for some. However, in this rapidly changing world, a robot like "A Human Following Robot" has become necessary in order to interact and coexist with humans. Robot needs an approach that allows it to see the person and respond appropriately in order to complete this task accurately. The robot must be capable of following a person in overcrowded spaces, vibrant settings, and both indoor and outdoor environments.

In order to recognise and locate the target, human following robots are typically outfitted with a variety of diverse sensors, such as a light detection and ranging sensor, laser rangefinder finder (LFR), radio frequency identification module (RFID), thermal imaging sensors, infrared (IR) sensing modules, wireless transmitter/receiver, camera, etc. To find and follow the target, all of the sensors and components must collaborate.

This research paper would describe a technique for a robot that can follow humans by identifying with the help of a camera. Using various sensors and modules, the robot control unit makes a well-informed decision based on the data gathered from the mentioned sensors and modules, detecting, and tracking the specific target while avoiding obstacles and avoiding collision with the target.

1.1 Problem Scenario

There are two example problem scenarios that shows the importance of human following robot in the library. The story starts when Aric has a hard time carrying books due to his physical ability, he has a fascination towards books with different journals causing him to have a wide selection of books. Another scenario is when Jenny a librarian has difficulty moving the tray of returned books to place it back on to the shelf as the it is heavy and not easily moved. Both scenarios it shows us that visitors tend to have difficulty carrying loads of books, these problems can be solved by a human following robot that is able to carry loads of books and able to follow the host as they move around the library. The robot would be equipped with state-of-the-art navigation and perception capabilities coupled with autonomous human tracking capabilities to assist visitors to carry books.

1.2 Challenges

As the capabilities of human tracking robots is able to assist humans in carrying books and human tracking capabilities. There are some challenges presents, among them are problems with human detection, tracking state and searching state.

Challenge 1: Human Detection
There are a few challenges faces by robot for human detection. One of it is detecting variation of host appearance. As there are a high number of students present in the library it is possible for people have almost similar feature such as height and cloth colour making It is more difficult to find the uniqueness of identifying the person, therefore it is harder to detect. There are also issues of false positive, meaning that the robot might detect and follow the wrong person halfway because of incorrect identification. There are also issues towards the environment where there are different background, lighting and situations caused by other student's movement causing confusion for the robot to identify the correct target.

Challenge 2: Tracking State
Some challenges might occur when tracking the target while navigating the library. When the target is blocked from the point of view of the robot by other objects, robot would have a hard time continuously tracking the target as it has to handle occlusion and target recovery of the actual person as it reappears. Another challenge occurs when the host suddenly changes speed and direction. The target sudden drastic change in movement forcing the robot to adjust quickly and predicting the potential direction of the target. Robot tracking the target also needs to deal with temporary disappearance. This would happen when the target enters a new room or turns a conner, reidentification function is needed to accurately retrack the target.

Challenge 3: Last Observed Position and Tracking State
The third challenge that robots will face is during Last Observed Position (LOP) and searching state to find back the target. When a target has unpredictable movements, the last observed position would be hard to predict the exact location. The robot must be able to intelligently acquire back the target by using intelligence strategy by predicting their possible whereabouts. Another challenge is environmental obstacles, the robot must know how to effectively avoid obstacles and navigate effectively around the obstacles by identifying the objects around it such as furniture and resume following the target. The biggest challenge for the last observed position would be an optimal search pattern. The efficiency of searching algorithms would be crucial in minimizing the search area, time and effort needed to find the missing target. What an effective search pattern needs are factors such as probability of the person's presence in different places, size of the environment, and potential hiding place.

1.3 Objectives

To counter the challenges above to produce a functioning human following robot, a few objectives must be created to resolve the challenges.

Objective 1: Develop a Robust Human Detection System
The first objective is to develop a robust human detection system that can correctly identify the target based on their characteristics such as body height and color of their shirt. This means that we should implement computer vision algorithms or other deep learning models to analyze the vision input of the camera. The robot should be able to distinguish between a human and a random background object. Another important factor for identifying target is lighting conditions and background variables through optimization of system's performance. The objective of this point is to achieve a high detection rate with the lowest false positives and ensuring correct identification of targets.

Objective 2: Implement a Reliable Tracking Mechanism
The second objective to ensure the robot functions correctly is by implementing a reliable tracking mechanism. This objective allows the robot to follow the target person continuously and reliably as they move. This includes implementing and designing a tracking algorithm that is able to make use of sensor data, for example camera feeds or dept information to calculate the proximity distance of the person's location at real time. To handle oscillation and temporary disappearance when the target disappears, the tracking algorithm must be sufficiently robust. To handle the tracking challenges, techniques such as Kalman filter and particle filter can be implemented. The main objective is to accurately track the target person while ensuring and maintaining a consistent following behavior.

Objective 3: Develop a Re-identification for Lost Targets
On top of reliable tracking mechanisms there should be a develop a re-identification mechanism to help robot to find target whenever the tracking mechanism fails, and they become temporarily lost or tracking disrupted. This point revolves around using the last observed position (LOP) of the target and utilizing a search strategy to relocate them. When initiating the search algorithm, the robot should be able to autonomously navigate through the library to efficiently relocate the target. The above objective attempts to make sure that the robot can effortlessly resume following the target person even after brief interruptions or occlusions and recover their position.

1.4 Expected Outcome:

The human following robot would be present at the entrance of the of the library. The robot identifies the visitor as potential candidate then visitor could activate the robot. The robot registers the visitor as the host to follow. After confirmation from the robot, the robot would move towards the host's direction. As the host move the robot would follow them using robot's human tracking capabilities. Host can seamlessly move around as the robot would still continue navigating through the library at the same time not losing sight of host. When host suddenly is out of sight, lost person tracking capabilities would be activated to find back the host. As host wants to load the book on the robot, host would command to stop, the robot that it has to be loaded and would not move for book loading. In the end when the host wants to go out to the counter, the books would be unloaded to the counter for confirmation of books borrowed.

This scenario demonstrates the capabilities of human following robot to accurately identify and track individuals, navigate through the library efficiently, and offer assistance in carrying books.

1.5 Project Scope:

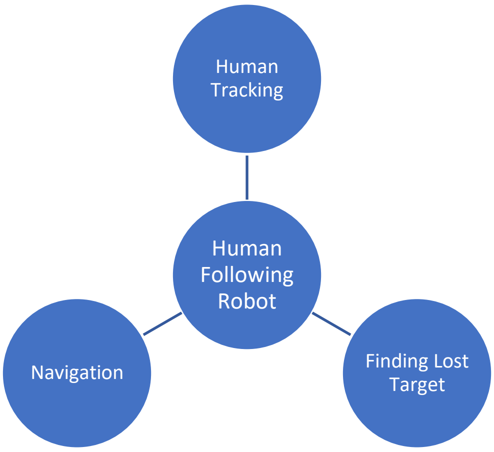
The project scope contains a few key components. The first scope is hardware design and assembly. Student must first develop the physical structure of the robot, selecting appropriate motors, wheels, and sensors, and assembling all components into a functional robot chassis. Second scope is human detection and tracking. Student would implement computer vision algorithms, to detect and track the designated human target in real-time. The third scope is autonomous navigation. Student must develop intelligent navigation algorithms to enable the robot to autonomously move throughout the library while avoiding obstacles.

1.6 Proposed Timeline

The project aims to develop a human-following robot and is divided into several key activities and tasks. The planning phase spans two weeks and that talks about creating a detailed project plan, defining objectives, and engaging with stakeholders. The research phase takes five weeks, during which a detailed online search for literature review is conducted to gather existing knowledge on human-following robots. The methodology phase, lasting three weeks, focuses on determining the system's functionality, defining information flow, and selecting appropriate tools and software. Documentation, scheduled for one week, involves modifying the work plan and Gantt Chart to reflect project progress and updates. A week is allocated for the project assessment for documentation, during which risks, and potential outcomes are evaluated. Finally, the planning document is reviewed and finalized before submission.

CHAPTER 2 – Literature Review

There are three parts to the literature review:



Three essential components drive their functioning: human tracking, navigation, and finding lost target. In this essay, student will compare these components to identify the best solution for the best performance for human-following robot. By understanding their strengths and limitations, student can understand how each part of the components work in different environments and compare to find the best suiting tools. The component methods selected must also fulfill all the objectives that student set out in the introduction. Each component comparison will consist of the sensor used and algorithms used by each author. Our project focuses more towards finding the best algorithms for each robot component.

2.1 Human Tracking

2.1.1 Sensor

For literature paper, the GRB-D camera that they use is ASUS Xtion Pro Live, which combines color (RGB) and depth information. This enables our system to perceive the environment in both 2D and 3D, allowing for a better understanding of the environment and perceives people in it [3]. The RGB-D sensor used in our study was the Orbbec Astra camera, which provides synchronized color and depth images [2]. The literature employed an Orbbec Astra RGB-D camera to track the movement of individuals [4]. The effective approach for estimating the position and orientation of a mobile robot utilizing a passive RFID system is presented in this research. Our strategy aims to precisely pinpoint the robot's location and orientation using the IC tag placement information. We can precisely predict the robot's location and pose by using the suggested algorithm, considering the incidence angle to IC tags [5].

Sensor	Advantages	Disadvantages	Citation
Stereo vision system	- Able to perceive depth in 3D perception	- Poor visibility - lighting variation	[1], [6]
RGB-D	- Can capture both RGB and dept information simultaneously	- limited range and sensitive to reflective/transparent surface	[2], [3], [4]

	Robustness in various lighting conditions		
RFID	- Not require any sensors having low computational load	- Users have to bring around the IC tag No obstacle avoidance functions	[5]

2.1.2 Algorithms

This study introduces a novel method that combines stereo vision-based human detection with modified Kalman filter-based human tracking. The technique makes use of reconstructed 3D object properties and features taken from 2D stereo images to detect humans in a robot's environment. To track individuals, a modified Kalman filter is used to recursively predicted and update estimations of a person's 3D coordinates in the coordinate system of the robot's camera [1]. This research introduces a new framework that combines state-machine control with deep learning approaches to create a repeatable and dependable human-following robot. A resilient to occlusion SSD detector is used to detect and track people. By employing an HS-histogram, which is resistant to changes in lighting, to extract the colour feature from a video series, the target subject is recognised. Using SLAM and a LIDAR sensor to detect obstacles, the robot securely follows the target to its destination. The construction of an effective and repeatable robotic state-machine control, the incorporation of a reliable vision algorithm for handling occlusions and illumination variations, and the creation of a reliable human-following system tested in practical indoor settings are all contributions of this work [2]. For a following mobile robot built on an embedded ARM platform, we suggest a low-computation technique. The Depth of Interest (DOI) algorithm has been enhanced such that it can be used to create mobile robot applications that follow people. The background is removed using the HDOI technique, which eliminates the influence of the surroundings on the CAM-shift method. The use of a virtual spring model with a safe and active region improves the safety and usability of the robot when it is trailing a person. According to experimental findings, the suggested method processes data more quickly than earlier methods like HOG and Stereo Vision with EKF [3]. Mobile robots designed to follow a specific person in the workspace require the ability to predict a person's trajectory and recover the target person if they move out of the robot's camera field of view. This paper presents an extended work of an online learning framework for trajectory prediction and recovery, integrated with a deep learning-based person-following system. The framework utilizes a single shot multibox detector deep neural network for real-time person detection and tracking. It estimates the real-world positions of persons using a point cloud and identifies the target person based on clothes colour extracted using the hue-saturation-value model. The framework enables the robot to generate robot trajectories that follow the target while avoiding obstacles and learn online the target trajectory prediction based on the previous path of the target person [4]. The paper presents a robust system for human detection and tracking in videos, particularly in indoor environments. The method involves obtaining the human's disparity image from a stereo camera and performing human detection using the Hu moment feature extraction method, which provides invariance to translation, rotation, and proportion changes. Human tracking is achieved using the Extended Kalman Filter (EKF). The result of experiments demonstrates improved robustness in human detection and tracking using the proposed method [6]. This research focuses on the detection of human objects and tracking to overcome challenges in difficult conditions. The proposed model utilizes a Cluster segmentation approach for human object detection. The input video is divided into frames, followed by cluster segmentation and feature extraction based on the Histogram of Gradient (HoG). Classification is performed using the Support Vector Machine (SVM) algorithm, and each object activity is detected based on the classification result. The proposed model achieves a detection accuracy of up to 89.59% for each object [7]. The effectiveness of human detection models on edge computing, including PedNet, multiped, SSD MobileNet V1, SSD MobileNet V2, and SSD Inception V2, is assessed in this research. The survey examines the accuracy and computation times of several approaches for real-time applications and gives an overview of them. According to experimental findings, in video datasets with various conditions, the SSD MobileNet V2 model obtains the highest accuracy with the quickest processing time when compared to other models [8]. This study contrasts CNN with the widely used HOG-SVM technique for person detection and suggests using the hybrid KPF filter. The comparison takes into account full occlusion circumstances and filter performances. The findings demonstrate that CNN consistently detects more and performs better in noisy and totally occluded videos. HOG-SVM produces better outcomes in non-occlusion instances. But when it comes to practical applications, CNN delivers higher-quality results. Due to the predominance of linear movement, the Kalman filter (KF) outperforms other filters in comparative tests. When utilising HOG-SVM in circumstances with total occlusion, the suggested KPF filter delivers results that are comparable to those of KF while also being more accurate than KF. However, KF employing CNN offers substantially higher accuracy when all frames are considered. The suggested KPF filter has an advantage over the particle filter (PF) since it combines linear and nonlinear features. However, it is important to remember that KPF requires more time, particularly in non-occlusion circumstances [9]. Robots that track people primarily seek for and pursue their intended human targets. Using Histogram of Oriented Gradient (HOG) features and a Support Vector Machine (SVM) classifier, the system first detects humans. Before computing the colour histogram, background colour information is subtracted using a straightforward foreground extractor. The target is then found using statistical comparisons between colour histograms. Following the determination of the colour similarity, a Kalman filter-based predictor is used to predict the target's location. The HOG-SVM-based human identification system exhibits good detection efficacy, but the execution time is still long, making it difficult for real-time systems to use. A block matching method is added to effectively shorten the execution time for tracking in order to remedy this [10]. Accurate identification of pills is crucial for ensuring the safe administration of drugs to patients. This study compares three mainstream object detection models, namely Faster R-CNN, Single Shot Multi-Box Detector (SSD), and You Only Look Once v3 (YOLO v3), for pill identification and evaluates their performance. The study shows that YOLO v3 offers advantages in terms of detection speed while maintaining a certain Mean Average Precision (MAP), making it suitable for real-time pill identification in a hospital pharmacy environment [11]. This study focuses on human recognition in several colour spaces, including RGB, YCbCr, HSV, and grayscale, employing match methods of local characteristics, such as SIFT and SURF. According to the study, when compared to other colour spaces, SIFT and SURF with HSV had the highest recall values [12]. For the project, three different algorithms, namely YOLO, SSD, and Faster RCNN, will be employed for the detection of a tennis ball. A comparison is conducted among these algorithms to determine their performance. The findings indicate that SSD is a more efficient and accurate algorithm with faster computation speed for the specific task of detecting tennis ball tosses. This system can be further developed to measure various other parameters in addition to the tosses of the tennis ball. It allows for the tracking of the entire trajectory of the ball throughout the game and enables a comprehensive analysis [13]. The study reveals that while SIFT performs well in most situations and is the fastest algorithm, ORB is the fastest overall. Only in noisy photos does SIFT surpass ORB and SURF [14]. In this study, a highly accurate and robust algorithm called K-SSD is proposed to estimate the location and angle of targets, which is particularly efficient for autonomous navigation. The SSD model is introduced to increase object location accuracy and speed, while the Kalman filter increases K-SSD's resilience. In spite of whether or not the SSD prediction results are noisy, experimental results demonstrate that the K-SSD proposed in this study achieves excellent accuracy in estimating the position and angle of the experimental platform [15].

Method	Advantages	Disadvantages	Citation
KF	- Cost effective and good performance in linear environments - Faster processing time - Works well in linear real world linear environment	- Struggle in non-linear environment	[1], [9]
Hu moment + EKF	- Works well in non-linear dynamics. -Able to make decisions based on both measurement and predictions	- Struggle to accurately differentiate between humans and other objects	[6]
SSD + HSV + Lidar	- Robust system with reliable human detection (able to differentiate target) and repeatable robotic control (wont freeze)	- Detect people with same shirt colour	[2], [4]
HDOI + CAM-shift	- Low computational requirements	- General human detection as it does not have a specific target person. - Cannot deal with complex environments such as occlusions.	[3]
Histogram Of Classification (HOG) + Support Vector Machine (SVM)	- Excellent feature representation - Unaffected by geometric anomalies - SVM high accuracy tracking target - Works well in non-occluded environment	- High computational complexity - Slower processing time - Low accuracy and low object variation detection in crowded environments because lack of contextual clues	[7]
SSD MobileNet V2	- Highest accuracy with fastest computation time	- More computational resources are required as it is a more complex system.	[8]
CNN	- High detection accuracy - Effective in real world environments (noisy and occluded environments)	- Slower processing time - Require large amount of training data - Higher complexity system	[9]

KPF	- More accurate than KF in complete occlusion	- More time consuming	[9]
KF + CNN	- More accurate results compared to (KF + HOG + SVM)	- Least time consuming compared to (KPF + CNN) and (KF + HOG + SVM)	[9]
HOG + SVM + BMA	- 10 times faster than (HOG+SVM) but retains same performance - Good detection performance	- Sensitive to light changes and object rotation	[10]
YOLO v3	- Highest detection speed	- Lowest accuracy compared to Faster R-CNN and SSD	[11]
HSV	- Highest recall value - Robust in various lighting conditions	- Need additional computations to convert image	[12], [2], [4], [10]
SSD	- Good balance between speed and accuracy - Able to classify and detect various types of objects -less computational speed	- Need to train data - Struggle to detect heavily occluded objects	[13]
K-SSD	- Robust detection of SSD - Enhanced tracking and accuracy from the help of Kalman filter	- Add complexity to the algorithm - Sensitive to occlusion	[15]
SURF	- Offers a good balance between speed and performance. - Works well in distorted environments and is less sensitive to abrupt object changes. - Provides moderate matching performance, but faster than SIFT by a large margin.	- Available matching points vary significantly. - Moderate matching performance compared to SIFT	[14], [16]
SIFT	- Way better accuracy than SURF and ORB	- Has a slower speed than SURF by a large margin.	[14], [16]
ORB	- Fastest compared to SIFT by a large margin and SUFT. - Good performance in noisy image	- Lower performance compared to SIFT by large margin - Slightly worse performance compared to SURF	[14], [16]

2.2 Navigation

2.2.1 Sensor

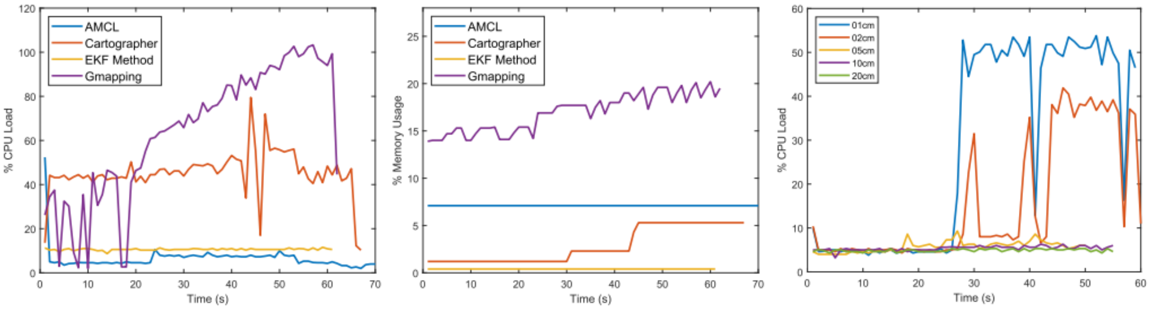
Laser scanners are widely used for distance measurement due to their accuracy and speed. They are commonly employed in SLAM (Simultaneous Localization and Mapping) implementation for mobile robots. However, one major drawback of LiDAR is its relatively high price. Despite this, LiDAR is favored for applications such as obstacle avoidance and SLAM-based navigation control in mobile robots [17]. The LiDAR SLAM system is configured with a 2.5GHz CPU and 6GB RAM for the hardware system. The software system includes Ubuntu v14.04 and the Robot Operating System (ROS) in the indigo version. GPU is not utilized for parallel computing. The sensor setup consists of a 16-line Velodyne LiDAR and an IMU380ZA-200 IMU sensor [18]. The platform is equipped with an onboard computer Jetson TX1, bumpers, sonars, Troyka IMU module, Hokuyo urg-04lx-ug01 2D LiDAR, Velodyne VLP-16 3D LiDAR, and wheel encoders for wheel odometry calculations. The robot's onboard computer runs Ubuntu 16.04 with ROS Kinetic [20]. To enhance the implementation of the algorithm, our robot is equipped with an rplidar A1, an IMU sensor, and odometry sensors to provide additional information feedback [22]. The distance sensor utilized is a low-cost 360-degree laser scanning radar known as the RPLIDAR A1. It is a 2D laser radar (LiDAR) solution developed by SLAMTEC Corporation, offering sensor resolution down to the millimeter level [23]. The mobile robot uses Rplidar A2[19].

Sensor	Advantages	Disadvantages	Citation
Lidar	- Accurate and fast distance measurement	- Relatively higher cost	[17], [18], [19], [23]
IMU	- Provides motion sensing and orientation	- May suffer from drift	[18], [22]
Odometer	- Measures total travel distance	- Slips or inaccuracies might occur	[22]

2.2.2 Algorithms

The research introduces a mobile robot navigation control system that blends real-time obstacle avoidance with laser SLAM localization. Cartographer SLAM is used in the LiDAR SLAM localization system within the ROS software architecture, and adaptive Monte Carlo localization is used on the robot. It is suggested to use an integrated navigation system that combines SLAM with obstacle avoidance to let the robot navigate to its intended location while avoiding unanticipated obstacles. The goal-seeking controller is integrated with the obstacle avoidance controller using a safety-weight parameter. The robot's successful localization, navigation, and obstacle avoidance abilities are demonstrated by experimental findings [17]. Due to vehicle restrictions, we set the high speed for evaluations at various speeds at 0.8 m/s. In both low-speed and high-speed instances, the trials showed that LOAM consistently obtained localization errors below 10 cm, demonstrating its robustness. However, it was discovered that AMCL's localization accuracy was less accurate than Cartographer's. We think this is because AMCL needs a predetermined map from another mapping method, such Gmapping or Cartographer, which can lead to poorer localization performance. The testing revealed that Cartographer used less memory than AMCL in terms of memory utilisation. This observation was made, albeit, in a short amount of time.

Cartographer SLAM may steadily use more memory as time goes on and new maps are created, eventually surpassing AMCL in memory use. The outcomes of the experiment show that: 1) The IMU helps to correct orientation estimation mistakes imposed on the wheel odometry, leading to better pose estimation. 2) When creating maps at the same resolution, both Gmapping and Cartographer exhibit good localization performance in small areas, with Gmapping consuming slightly more RAM than Cartographer. 3) Despite having low CPU loads and requiring less memory, wheel odometry and AMCL's localization accuracy is worse to that of Gmapping and Cartographer. 4) LOAM functions well in both small and large test fields [18]. The paper proposes a four-wheel drive adaptive robot mapping and navigation system based on ROS. The Karto SLAM algorithm is selected to construct a 2D map after comparing the mapping effects of various 2D laser SLAM algorithms. The comparison shows little difference between the maps generated by Gmapping and Karto SLAM, while the Hector SLAM algorithm exhibits relatively poorer mapping performance due to limited lidar update frequency, accuracy, and the absence of loop closure detection. The Karto SLAM algorithm has lower requirements for odometer accuracy and radar frequency. Compared to filtering methods, Karto SLAM adopts graph optimization with front-end matching and loop closure detection, resulting in higher robustness and better mapping performance in various environments [19]. The presented tests indicate that Google Cartographer consistently produces maps with the smallest error compared to the precise ground truth provided by the FARO laser tracker. This algorithm demonstrates robustness to various types of mobile robot movements. Gmapping, although slightly behind Cartographer in map quality, still generates reasonably good 2D maps even without loop closure. Gmapping and Cartographer both utilize odometry for localization correction and map refinement. On the other hand, Hector SLAM relies solely on LIDAR data and lacks an explicit option for loop closure, resulting in less accurate results. Thus, the research concludes that Google Cartographer is one of the best algorithms for generating 2D maps using LIDAR on a mobile robot [20]. The cartographer outperforms the other two SLAM approaches in every case, showcasing its ability to adapt to swift movements and direction changes. The Cartographer's mean square error (RMSE) was as low as 0.017m, with the greatest RMSE being around 0.35m, which is within an acceptable range for dangerous indoor conditions. However, the Cartographer’s CPU performance optimization is inadequate. The second-best mapping results are provided by Hector SLAM, but it has trouble with abrupt changes and extended corridors. The best CPU utilization among the three solutions is demonstrated by Hector SLAM, which is a key benefit. Gmapping was unable to finish mapping in the office scenario since there was no trustworthy odometry source available. The ideal conditions for each technique vary depending on the environment. Based on the experimental study, Cartographer is deemed a solid choice for producing 2D maps using low-cost LiDAR in odometry-free handheld systems, leading to its adoption as the primary technology for the next stage of the research, a 2D mapping system with multiple agents [21]. The mapping effectiveness of Hector SLAM is deemed unsatisfactory because it does not utilize the odometry data from the robot. Consequently, when the vehicle spins too fast, Hector SLAM fails to recognize the rotation, resulting in unmatched laser data and the phenomenon of "ghosting" in the map. Hector SLAM solely relies on 2D LIDAR data and lacks explicit loop closure detection, leading to less accurate results [22]. According to the experimental results, both Gmapping SLAM and Hector SLAM algorithms are capable of real-time robot positioning and indoor map creation. Since Gmapping SLAM relies on the robot's odometer information, which limits the robot's movement speed. Hector SLAM, on the other hand, requires a high laser scanning frequency but does not require odometer information. The decision between Gmapping SLAM and Hector SLAM depends on the particulars of the experimental setting. Each technique has advantages and limitations [23].

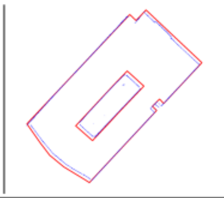
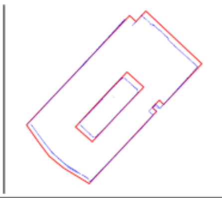
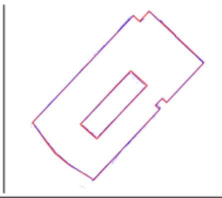
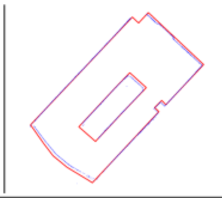
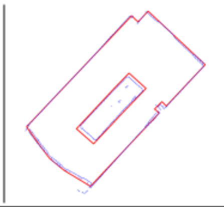
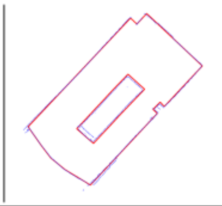
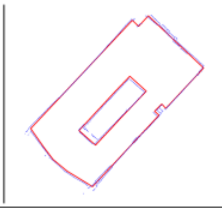
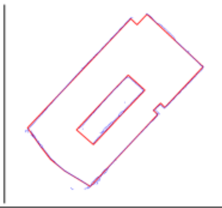
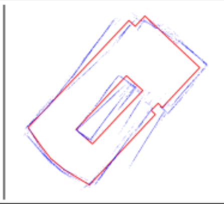


(a) (b) (c)

Fig. 15. Running efficiency. (a) and (b) show the CPU load and memory usage when running different SLAM methods on the same dataset. (c) shows the CPU load when running AMCL on different minimal update distances.

Picture shows performance chart of four SLAM algorithms [18].

TABLE I: Map comparison constructed from SLAM algorithms (in blue colors) and ground truth (in red color)

SLAM method	Slow	Fast/Smooth	Fast/Sharp	No loop closure
Gmapping				
Cartographer				
Hector SLAM				

Picture shows map comparison of four SLAM algorithms [20].

TABLE I

COMPARISON OF DIFFERENT LIDAR SLAMS. NOTE THAT, THE ALGORITHMS ARE CLASSIFIED INTO EKF, PF, EM AND GRAPH

Method	Open Source	3D Pose	Loop Closure	Real Time	fuse IMU or Visual	Algorithm	Map Type	Matching Method	Localization Performance	CPU Load
Gmapping [33]	√			√		PF	Grid Map	Scan-to-Map	Good	High
CoreSLAM [34]	√		√	√		PF	Grid Map	Scan-to-Map	Poor	Low
KartoSLAM [50]	√		√	√		Graph	Grid Map	Scan-to-Scan & Scan-to-Map	Good	Low
LagoSLAM [31]	√		√	√		Graph	Grid Map	Scan-to-Scan	Medium	Medium
HectorSLAM [30]	√	√	√	√	IMU	EKF	Grid Map	Scan-to-Scan	Good	Low
LOAM [51]	√	√		√	IMU		Pointcloud Map	Scan-to-Scan & Scan-to-Map	Excellent	Medium
V-LOAM [52]		√	√	√	Visual		Pointcloud Map	Scan-to-Scan & Scan-to-Map	Excellent	
Cartographer [39]	√	√	√	√	IMU	Graph	Grid Map	Scan-to-Map	Good	High
IMLS-SLAM [41]		√					Pointcloud Map	Scan-to-Scan & Scan-to-Map	Excellent	
CPFG-SLAM [36]		√		√		EM	Grid Map	Scan-to-Map	Good	
LIMO-SLAM [44]	√	√		√	Visual		Pointcloud Map		Good	
STEAM-L [45]		√		√			Pointcloud Map	Scan-to-Scan	Good	
SuMa [35]		√	√	√		Graph	Pointcloud Map	Scan-to-Map	Medium	

Picture shows comparison table of all LIDAR SLAMs [18].

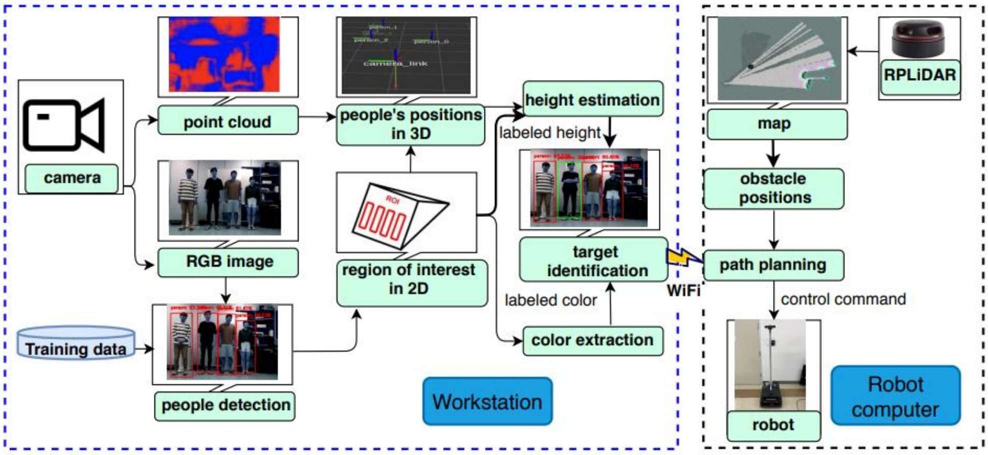
Method	Advantages	Disadvantages	Citation
Cartographer SLAM + adaptive Monte Carlo localization	- Accurate mapping - Robustness	- High computational resources - High complexity to operate	[17]
Wheel Odemetry	- Low memory load and usage	- Less accurate localization by a large margin	[18]
EKF	- Better localization accuracy than wheel Odemetry	- Moderate CPU load and memory - Lower performance compared to ACML, LOAM, Gmapping and Cartographer.	[18]
AMCL	- Low memory and CPU load usage	- Cannot perform in high speed - Lower accuracy than Gmapping and Cartographer	[18]
LOAM	- Can operate in high and low speed conditions - Can operate well in small and large area - Works well with L-shape environment	- Does not perform best accuracy in any situation	[18]
Gmapping	- Good localization in small area especially straight line	- More memory usage and lower accuracy compared to Cartographer	[18], [19], [20], [21],

	-Better accuracy than LOAM and ACML and EKF - Reasonably good outcome without loop closure	- Requires an odometer and high odometry accuracy for optimal performance leads to more computational load - No loopback closure	[22], [23]
Cartographer	- Good localization in small area and start of with less memory load - Lower memory consumption than AMCL	- Would accumulate more memory for new constructed maps - Requires accurate odometry and sensor data	[18], [20], [21]
Karto SLAM	-Relatively low requirements on odometer accuracy and lidar frequency -Utilizes graph optimization for better mapping results -has loop closure function	- Lower real time performance compared to Cartography and Gmapping - limited loop closure handling	[19], [22]
Hector SLAM	- Capable of handling high lidar frequency data, which can provide more detailed and accurate maps - Robust enough for different types of environments -can operate without odometer - Best CPU utilization	- Relatively poor mapping performance in the dataset - Less accurate compared to Gmapping and Cartography - Lack of loopback closure means sensitive to sudden direction changes	[19], [20], [21], [22], [23]

2.3 Finding Lost Target

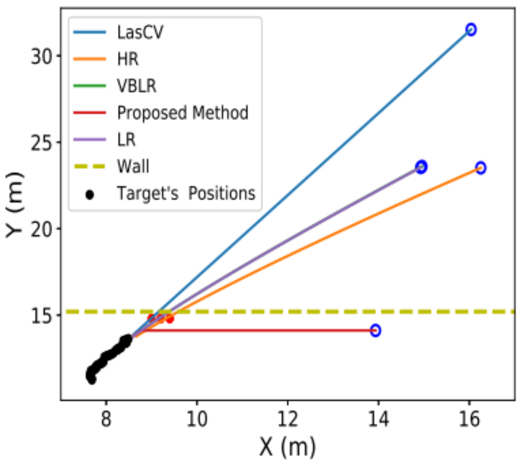
2.3.1 Algorithms

The previous assumption was that a robot could locate a person within the camera's field of view (FoV) when it reaches the person's last observed position (LOP). However, since people move along various trajectories over time, there are instances where robots may struggle to reacquire and track a person. To address this, the robot randomly rotates and scans the environment to detect the desired target. If the target is found again, the robot enters a tracking state. Lest say that the desired human cannot be found, for example when the person moves away and disappears around a corner, the robot either stops after a certain time or allows the users to stop it. In some cases, it is reasonable to assume that the robot may not be able to detect the desired human [2].



Picture diagram shows proposed human following system [2].

This paper presents an extended version of an online learning framework for predicting and recovering trajectories, integrated with a deep learning-based person-following system. The key improvements compared to the previous paper include faster recovery time and a significantly higher success rate when the robot loses track of the target. In our previous work, recovery was primarily based on random search, which resulted in longer recovery times or even failure. In the method section, we describe how we enhanced the state transition control, including recovery, by utilizing online trajectory prediction based on the target's past trajectory [4].



Picture states the prediction results linear regression performance [4].

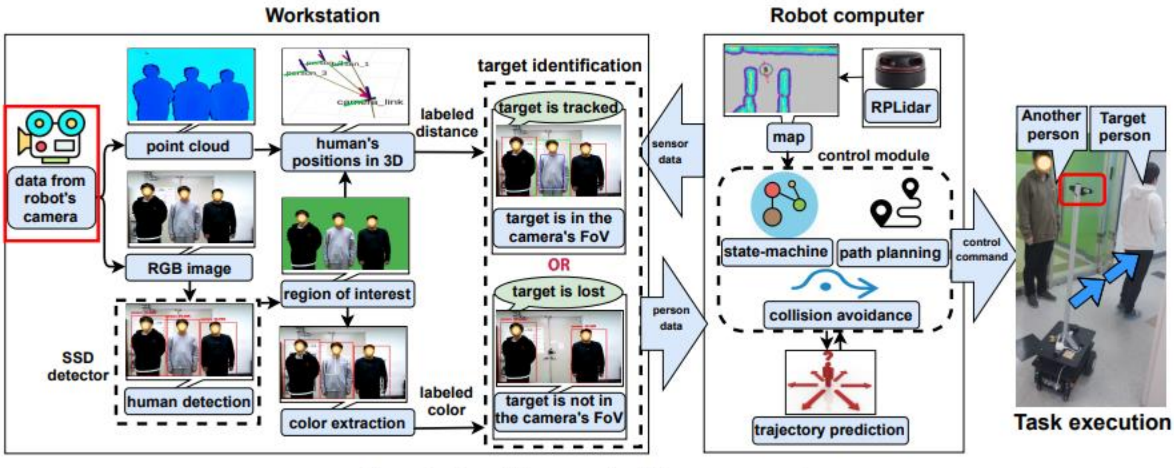
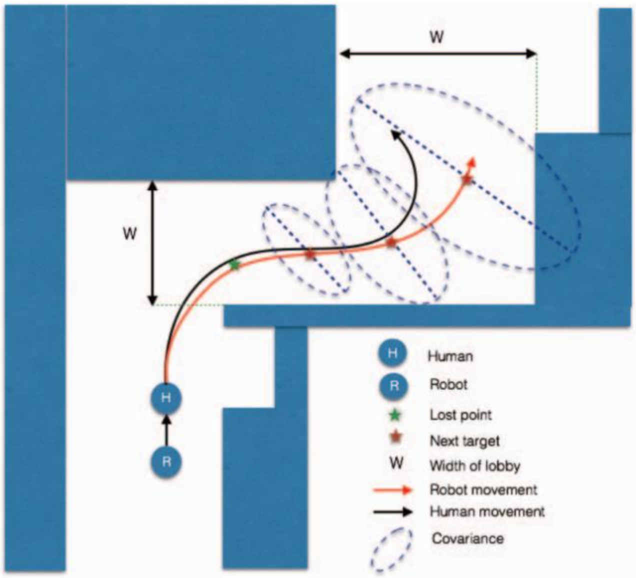


Figure 1. Overall framework of the recovery system.

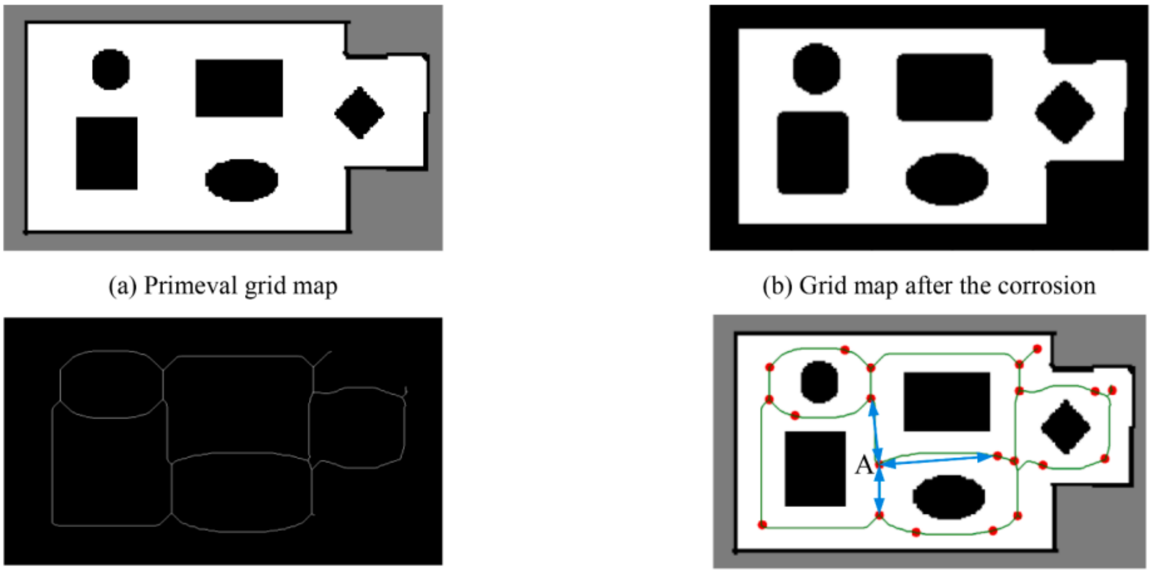
Picture diagram shows the flow of commands for human following robot for target recovery [4].

In cases where the target goes missing, this research suggests a dependable recovery procedure for a person-following robot. Probabilistic methods, such the Kalman Filter, are used to estimate the target's prospective positions by using past data, including the target's previous positions before it was lost. In order to help the robot, choose a search path for relocating the human target, map data is also used. The map also includes obstacle avoidance to help the robot navigate without incident and get to its intended human target. Results from experiments show that the suggested recovery mechanism performs better than expected [25].



Picture above shows method of recovery [25].

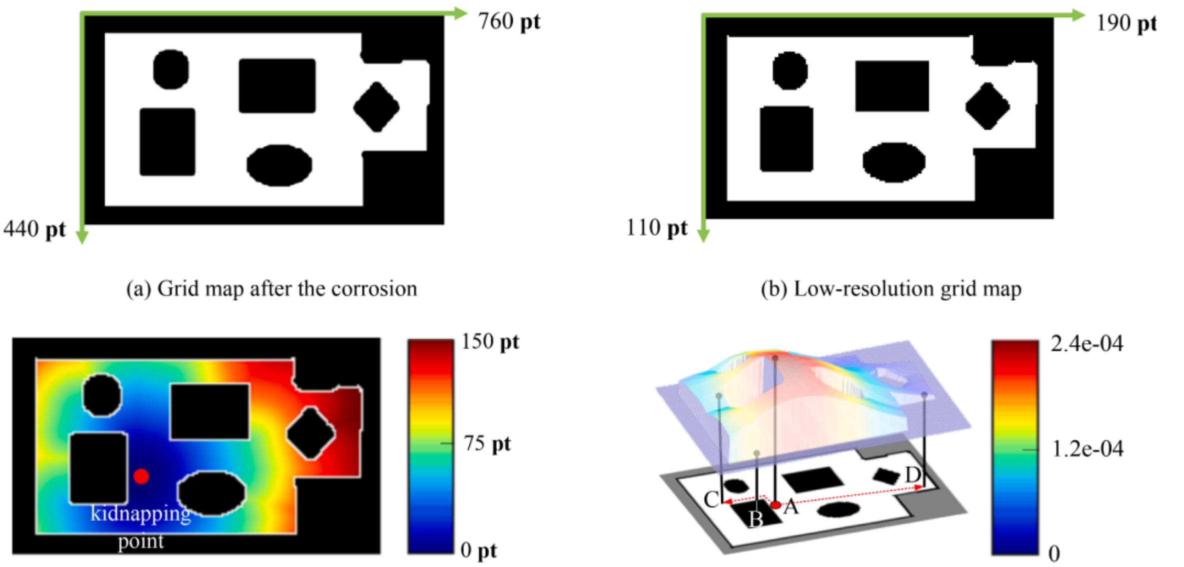
This research provides a successful re-localization method that draws inspiration from the finding a missing person (SFMP) technique with the goal of enhancing recovery efficiency and success rate. The suggested SFMP-based re-localization ensures the presence of particles at the robot's re-appearance site by optimizing the distribution and quantity of randomly generated particles, strengthening the resilience of Monte Carlo Localization (MCL). The method contains a key search space selection to pinpoint the area where the robot is likely to appear and a cutting-edge moving distance map that shows the displacement from the location of the kidnapping, which helps to focus the search area based on the potential robot movement distance. In order to change the number of random particles and dynamically control particle count, a time-varying long-short term method is also implemented. Real-world tests confirm the usefulness of the suggested approach [24].



(a) Primeval grid map
(b) Grid map after the corrosion
(c) Skeleton-based topological map
(d) Pivotal search space selection

Fig. 3. Pivotal search space selecting process.

Picture shows step to generate pivotal search space [24].



(a) Grid map after the corrosion
(b) Low-resolution grid map
(c) MDM
(d) MDM-based Gaussian distribution

Fig. 4. Diagram of the MDM-based Gaussian distribution.

Pictures show gaussian distribution to show range of likelihood of how far the target has gone [24].

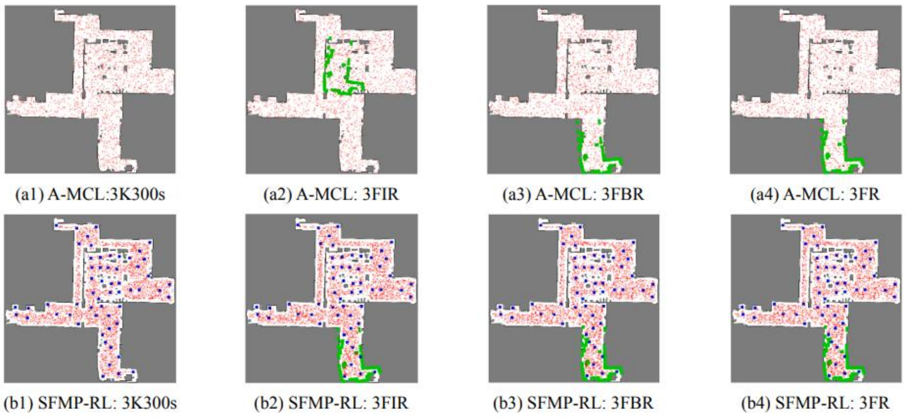
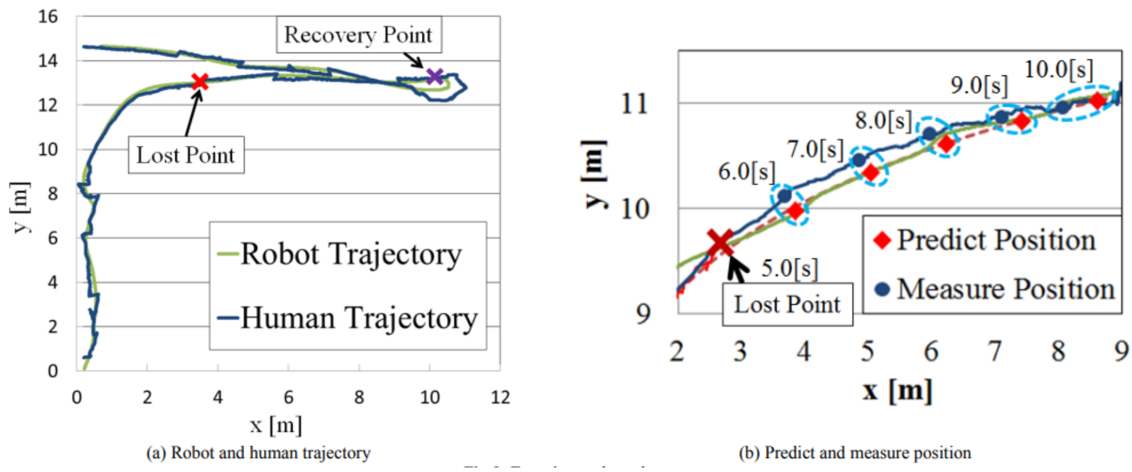


Fig. 13. The partial evolution process of the particle set in localization iteration for the re-localization at point F in localization recovery case 3. The green dots are the LiDAR measurement from the currently estimated robot pose; the red dots are the particles and the blue dots are the pivotal search space; see Table 3 for abbreviations.

Picture above shows pivotal search location indicated by the blue dot meaning the predicted position target would appear [24].

This review of the literature introduces a robot that can follow people and keep track of them even if they get lost around a corner. The authors suggest a human trajectory model and a method of prediction for predicting human loss at corners. To validate the suggested model and its application in the ensuing robot, experiments are carried out on the trajectory as a logarithmic function. With distance deviations between projected and measured positions of less than 0.65 meters, the trial results show that the robot is capable of tracking and following the target person even after losing sight of them at bends. As a result, the authors accomplish what they set out to do [26].



Picture shows method able to recover the target after losing sight [26].

In this study, hidden Markov models (HMMs) are used to predict autonomous behaviour. It deals with the issue of online categorization, which is to choose the most likely behaviour out of a group of behaviours that are mutually incompatible, and it broadens the scope of the method to cover circumstances including incomplete and potentially dependent behaviours. The similarity between actual and ideal observations for each behaviour is calculated using event-based observation models. Two robots in a static environment are used in simulation studies to show how well behaviour can be predicted. Most of the time, the correct behaviour is determined before 25% of its execution, and by the time 40% of the behaviour is complete,

all models have the highest likelihood for the real behaviour. The results highlight how well the technique performs in real-time behaviour classification and prediction tasks [27].
This study looks at the localization techniques used by multiple authors to place mobile robots. It examines automated map-building approaches, probabilistic map-based localization, and RFID localization techniques for mobile robot localization. For mobile robots operating in unknown surroundings, the SLAM model is regarded as an efficient and frequently used localization strategy. Extending the use of SLAM with probabilistic localization techniques like the Extended Kalman Filter (EKF) increases localization and orientation accuracy while lowering positioning mistakes. The EKF approach excels at finding accurate solutions with faster convergence rates in low noise situations. The RFID system has shown effectiveness in robot tracking in contexts with constrained space, such as indoor settings. Robots can estimate almost optimal pathways with the use of evolutionary approaches, which increases the robustness of these systems. In particular for indoor contexts, the combination of RFID systems and SLAM methods seems to offer higher localisation accuracy [28].

If a person-following robot is to be effective, it must be able to estimate the target's trajectory and predict its possible location when it abruptly vanishes. The study recommends using a regression model based on prior data to predict a person's likely trajectory. For the purpose of forecasting individual trajectories, a Support Vector Machine Regression (SVR) is used. SVR forecasts offer a good approximation, whereas polynomial regression predictions deviate from the actual trajectory when turning right or left. The extrapolation process is rife with errors and could produce forecasts that have no real value, hence, it is crucial to understand the limitations of prediction algorithms. Extrapolation methods often diverge from the actual track, therefore estimates should only be taken into account for a specific time frame [29].

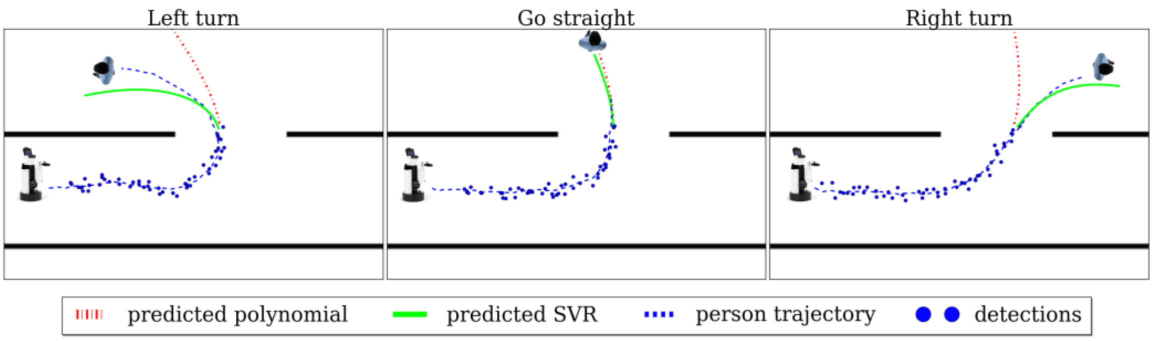


Fig. 4. Trajectory prediction experiments

Picture shows the effectiveness of the algorithm for turning corner situation [29].

Method	Advantages	Disadvantages	Citation
Random search	- Simplicity provides low computer requirements.	- Inefficient searching as there is lack of optimization and potential prolonged search	[2]
LR (Linear Regression)	- Higher recovery rate and faster recovery time. - Increase tracking accuracy	- Prediction rate would decrease in more complicated environment such as u turns	[4]

KF	- Provides an effective and efficient method of estimation with noisy and incomplete environment - Computationally efficient as it can handle large amount of data in real time	- Cannot accurately predict target when sharp turns - Requires initial estimate position and covariance	[25]
(Strategy of finding a missing person) SFMP-(re-localization) RL	- High accuracy and efficiency rate of recovery - Robustness in handling disturbances - Higher success rate compared to A-MCL and BNB-GL.	- Highly reliant on the accuracy of minimum distance map - Higher computational complexity	[24]
Logarithmic function	- Accurately predicting trajectory of person - Flexible to adapt into different corner turning scenarios	- Usability only applicable to turning corner - Sensitive to parameter values	[26]
Hidden Markov Models	- Good for modeling sequential data - Provides probabilistic inference - Identify wide range of behaviors	- Model complexity and training data - Limited prediction representation power - High computational load	[27]
EKF-SLAM + RFID	- Good localization accuracy - Simplest and easiest - Beneficial in places with limited visibility	- Limited range - Frequency interference	[28]
SVR + RBF	- Better trajectory prediction than polynomial regression - Non-linear prediction	- Higher computational load as it depends on training sample	[29]

2.4 Research gap

After weeks long of searching for different research paper regarding human following robot one part that student find has lack of information is regarding recovery search there are many algorithms that cover trajectory predictions to predict target when lost, but there are lack of information specifically searching for lost target. For example, there is one paper that talks about the strategy of finding a missing person where a radius is created to predict the lost target position based on AMCL and pivotal map points. The student is unable to find more models that can find targets in a more global level using map and algorithm model, instead of just predicting at a local level. Let’s say the target is lost for long duration and local level cannot predict the position of target it will just activate random search to go random places and find the target. The rest of the research section such as detection and navigation student don’t find any complications finding information as it is already a mature subject and able to fulfill our requirements.

CHAPTER 3 - Methodology

3.1 Functionality provided by the system

My project objective of this human following robot is to follow the human while carrying a load of books at the same time tracking the human and find the human target when lost.

1. Human Following
The robot is capable of autonomously tracking and following a designated human target. It uses sensors and algorithms to detect and maintain a safe distance from the human while moving along with them.

2. Tracking Human Trajectory
The robot employs tracking mechanism to continuously monitor the human targets location and movement. The moment the human is out of sight of the robot, it will attempt to acquire and follow the user again.

3. Autonomous Navigation
The robot can navigate its environment autonomously to avoid obstacle and ensure smooth movement while following the human.

3.2 Tools Selection

3.2.1 Hardware

The Reeman Big Dog Robot Chassis is a highly versatile and powerful platform design. Most importantly it provides various applications, offering complete perception, cognition, positioning, and navigation capabilities. It is equipped with numerous features and functionalities they make it suitable for developing autonomous robots.

The first features that made us decide to choose this robot is rich interface and strong expansibility, the chassis provides an open SDK with API interfaces enabling users not needing to code everything from scratch. The second feature is incremental mapping function, it uses a positioning and navigation system with a laser detection distance of 25-

meter Blue Shark LIDAR which is preinstalled into the robot. It allows the robot to build environment incrementally allowing detect and build map in real time. The third feature is automatically avoiding obstacle and detour. It is equipped with laser SLAM and 3D camera fusion technology which offers advanced environment perception capabilities. The fourth feature is super loading capability, the Big Dog Chassis can carry up to 100 kilograms of weight without damaging it with the sturdy sheet metal structure. Lastly it has remote navigation deployment as it has a cloud remote deployment capability which allows indoor navigation map through remove controlling the robot.

The Reeman Big Dog chassis offers an open bottom interface where it allows users to perform secondary deployment and customize the robot to meet their specific application requirements. For our case we would install the camera and new processor on the robot. Robot is equipped with a high-capacity lithium iron phosphate battery, providing ample power for longer operating hours of tasks. By integrating laser SLAM and V-SLAM multi sensor fusion algorithm to create a detailed map of large areas. Thus, making it suitable for efficient navigation in a large unknown environment.

The camera that is proposed is Intel RealSense RGB-D camera.

Intel RealSense is a family of depth-sensing cameras developed by Intel Corporation. These cameras are designed to provide both RGB (colour) and depth information, enabling a wide range of applications in computer vision, robotics, 3D scanning, virtual reality, and more. The RealSense cameras utilize advanced sensing technology to capture depth information and provide a 3D perception of the world. They combine RGB image and depth map data to offer synchronized and aligned RGB-D imaging, allowing seamless integration of colour and depth information for a more comprehensive understanding of the environment. With different models offering varying depth ranges and resolutions, RealSense cameras cater to various applications, from close-range interactions to long-distance depth perception. Intel provides Software Development Kits (SDKs) with APIs and tools for easy integration, making RealSense cameras popular choices for developers, researchers, and robotics enthusiasts seeking accurate 3D sensing capabilities to enhance their projects in robotics, computer vision, and beyond.

3.2.2 Software

Now the robot consist of all the hardware presents a robust and versatile platform to meet wide range of requirements. The powerful hardware foundation from Reeman BigDog chasis provides an excellent base for building robots with diverse capabilities. However, to unlock the full potatial of the Reeman Big Dog chassis, an equally compatible and adaptive software framework is essential.

In this case a ROS Kinetic Kame emerges as a pivotal software component as it is most compatible towards Reeman robot compared to newer versions of ROS. The robot might have been designed and tested with ROS kinetic Kame that is why using newer version might introduce compatibility issues or require additional effort to ensure all hardware components are well supported. ROS Kinetic Kame is compatible with Ubuntu 16.04, which is widely used for Linux distributions. Its compatibility with ROS ensures a stable and familiar environment for developers and users. ROS provides a rich ecosystem of packages and APIs that cater a wide range of robotics platform and use cases. For this robot, developers take advantage of existing ROS packages that includes various aspects of robotics such as perception, localization, planning and control. The move_base package is a popular choice for navigation and path planning. It enables the robot to navigate autonomously while avoiding obstacle and reaching target location efficiently by combining global and local path planning. For localization, acml (Adaptive Monti Carlo Localization) package can be used to find approximate robot's pose position in a known map by using its sensor measurements. For SLAM, pakages like gmapping and Cartographer can be sued to build and update the map environment at the same time estimating its approximate location. The Navigation Stack, a group of ROS packages which provides high level navigation capabilities such as obstacle avoidance, localization, and global path planning. For robot manipulation task, MoveIt is used to coordinate all the necessary components and algorithm for motion planning and control. In order to incorporate vision data into our Reeman robot, the system needs a perception package. Image_proc provides better input image, pcl_ros enables 3D processing in point cloud and OpenCV for image processing and extract information from image. For the camera, Inter RealSense SDK 2.0 is a package provided by intel that enables ROS users to communicate with the camera. This package acts as a bridge between camera hardware with the ROS environment.

3.3 Proposed method

3.3.1 Hypothesis

The explanation below shows the decision made on the algorithm for each function:

1. The algorithm for Human Detection is SSD
Research on chapter 2 found out that the SSD has the best balance between speed and accuracy. Due to the limited computational power (Hog + SVM) could be a suitable candidate but unfortunately in a library setting where there are a lot of human's targets and many people moving around this algorithm is not feasible. YOLO is also considered but comparison shows that it has lower accuracy when detection and most importantly it has higher computational complexity compared to SSD. SSD has the highest accuracy with the fastest computational time compared to HOG+ SVM algorithm. Previous chapter also compared to YOLO and Faster R-CNN where it outperforms both of them in terms of accuracy, even though YOLO shows a better processing time, the performance is marginally poorer and also computational load is higher than SSD, Faster R-CNN also shown to have slower speed. According to experimental findings, in video datasets with various conditions, the SSD MobileNet V2 model obtains the highest accuracy with the quickest processing time when compared to other models [8].

2. The algorithm for tracking is Kalman filter
Chapter 2 summarizes that tracking is more suitable to be done by Kalman filter as majority of the movement is more suited for linear movement and more accurate tracking performance compared to other techniques. Another literature also suggests that the combination of KF with convolutional network (CNN) called K-SSD able to improve speed and accuracy of tracking target for autonomous navigation. Another consideration of EKF but it is customed to track object in non-linear movement, which is not what our environment, EKF is more time consuming than KF. The rationale behind choosing algorithm has to do with library noisy environment where occlusion happened frequently and robot most of the time operates between rows of bookshelf where turning or corners is common.

3. Algorithm for Identification Target is Hue-Saturation Value (HSV)
There are a few identification techniques that have already been looked into grayscale, HSV, YCbCr, and RGB. It was found that HSV has the highest target recovery rate and quickest to identify the target, followed by RGB.

4. Algorithm for Robot Navigation is AMCL for localization and Cartographer for mapping
After extensive research, the methods that prove sufficient can be narrowed down to a few, mainly Gmapping, Cartographer, Karto SLAM, and Hector SLAM. Gmapping and Karto and Hector SLAM prove to have the lowest memory requirements best making it suitable for projects where there are limited computational power. The literature states that Hector slam generally generates poor mapping performance as there is no loop closure function, as it uses high frequency from the lidar without the use of odometer [19]. Karto SLAM limited loop closure handling leading to less accurate results and lower real time performance compared to Cartography and Gmapping. Cartographer proves to be the best in terms of balance between performance and accuracy as it has best memory utilization and lower memory consumption especially when used for mapping.

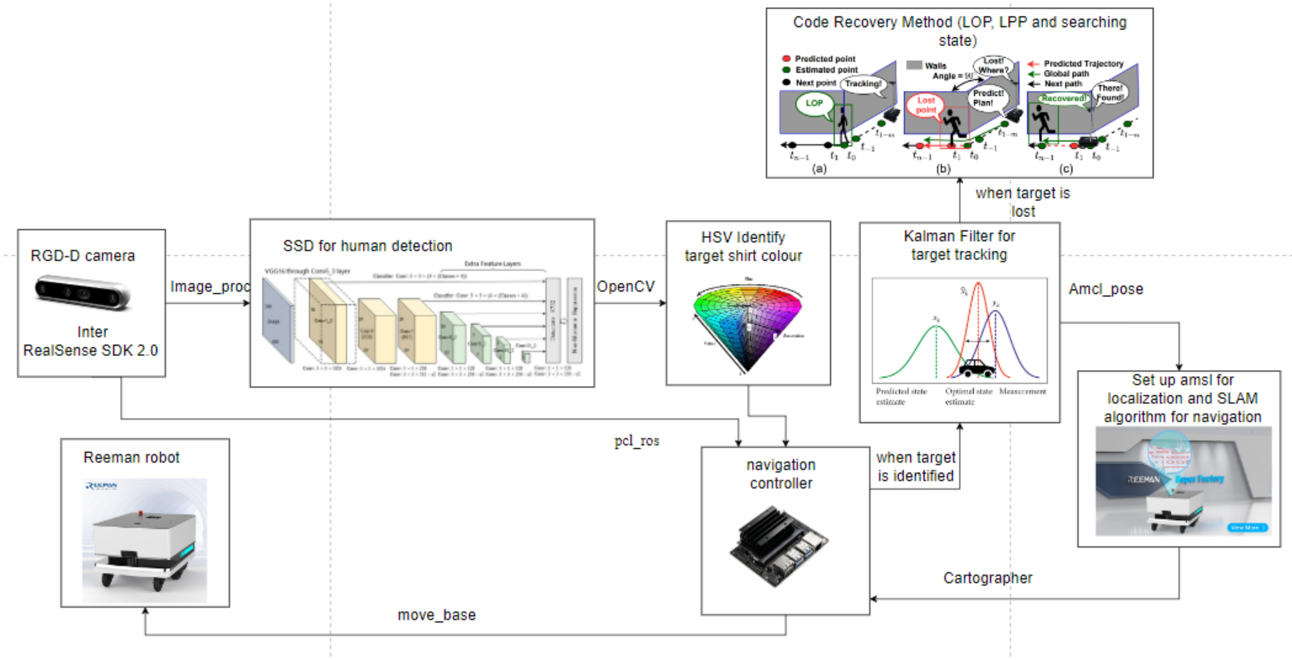
A theory that not a lot of researcher's approach is combining two SLAM algorithms together (AMCL and Cartographer) which utilizes the strengths of both algorithms. Students have also chosen AMCL as it is specifically suited for localization as it required low memory, but it is not suitable for building maps for unknown environments. While the best solution is Cartographer as it is the best algorithm overall. Literature states that integrating both can prove efficient and effective as the processor does not need to activate Cartographer all the time [17]. Instead, Cartographer is only used when creating a new map and AMCL is used for navigation control most of the time for localization and obstacle avoidance. Most importantly, ensuring the smoothness of performance is the top priority as it would prove inefficient as latency causes the robot performance to decrease.

5. Recovery Target using LOP, LPP, searching state.

There are many interesting methods proposed by each author. An author suggests that SLAM with RFID can help recover target, but it is not advised to bring an RFID receiver tag each time they use the robot. Another author suggests Strategy Finding a Missing Person, this way is effectively finding the missing target at a global level by knowing the distance or how far the kidnapper has already gone using map radius and base on that information guess the most likely area the target will appear. Another method talks about finding the last observed position of the robot, then based on the last position, we predict the last predicted position then find the target. If the target is still not found random search would be activated where the robot rotates to find the target.

A theory that the student wants to discover is combining multiple findings together. Through multiple literature results, the trajectory prediction of the target by using Kalman Filter is more effective for library environment. Instead of linear regression from the literature example [4], student utilized the technique used in from the example using LOP, LPP and Searching State but replace the tracking algorithm from linear regression to Kalman Filter. This is because the school library is a more dynamic environment therefore Kalman Filter able to handle escalation form crowded and unpredictable environment at the same time able to predict the turning of corners more effectively compared to Linear Regression.

3.3.2 Proposed framework



Picture diagram above shows proposed method by student.

Here is the explanation of each step of methodology:

1. Detect human

For the first part robot need to identify that this object is a person then only robot can know if it's a human to detect. The RGB-D camera would first acquire every frame of the image. The system would draw out the region of interest of all the human detected based on SSD. SSD is a popular and efficient object detection algorithm that accurately detect and localize objects such as humans in real time. When a Reeman robot captures the video or frame image, the SSD algorithm would process it to detect and localize various objects within the frame. The algorithm predicts the bounding box coordinates and class probabilities for each detected object. The image_proc package is use for transferring the image for processing and open_cv is use for processing the image and insert the ROI.

2. Detect target

After gathering all the ROI, it is time to find the main human target that the robot is going to track. For the correct identification of human target, the shirt colour would determine the valid target using Hue, Saturation, Value (HSV). HSV is commonly used for image processing for effective colour-based segmentation. By identifying the shirt colour using HSV, the robot can mark the person and track them based on colour cue.

3. Trajectory prediction

After identifying the human target, a method is needed to be used for the robot to constantly track the human. The method chosen for tracking a human is Kalman Filter. The functionality of the Kalman Filter is designed to minimize the chances of losing sight of the human target from the robot's field of view (FOV). In case the robot loses sight of the human, the Kalman

Filter's trajectory prediction helps the robot to estimate the target's position and find them again efficiently. The implementation of all models was carried out using Scikit-learn, a robust machine learning library. By collecting a series of (X, Y) coordinates of the target person's position and their corresponding time stamps, the Kalman Filter can predict the best fit trajectory that represents the person's movement. This trajectory information guides the robot's movement, enabling it to maintain a proximity to the person as they move.

4. Navigation

The next step that the robot needs is localization, to navigate and avoid obstacle. It equipes the robot with the ability to determine its precise position and orientation for their environment. The robot can use information from LIDAR sensor and SLAM generated map to accurately assess its location from its surrounding information. For robust mapping and localization in the library, our Reeman robot is equipped with SLAM (Simultaneous Localization and Mapping) technology. The Reeman robot is equipped with Blue Shark Lidar sensor to perform SLAM detection. The Blue Shark Lidar utilizes laser beam to generate a detailed 360 degree, 2D or 3D representation of the surrounding. The Lidar data with other sensor input enables Reeman robot to crate and update an accurate map of the library environment surrounding and localize its position with the environment. With the accurate map and localization information, the robot can perform obstacle avoidance effectively. It uses the map data acquired from lidar to detect obstacle and plan a safe trajectory and ensuring collision free navigation.

5. Recover Lost target

Recovery method would be activated when the target cannot be detection form the FOV of the robot. To find back the target when lost, the robot needs a recovery state function to be implemented. The recovery state consists of Last Observed Position, Last Predicted Position, and Searching State to handle temporary loss of the human target. Lop and LPP would be used to navigate the robot back to the target’s last known position or predict its future position.

6. Maintain safe distance

The system continuously tracks the human target while repeating target recovery process to adapt to changes by the human target movement. The robot would adjust its movement based on the trajectory of the human based on the updated human position. It ensures that the robot is at a safe distance to not lose sight of the target from field of view of camera using the package ROS move_base. Pcl_ros is used for generating a point could data which is send to processor able to determine the position of the target in 3D to determine if the labelled target distance is safe.

Chapter 4 – Work Plan

4.1 Capstone 1 Gantt Chart

Activities/ Task	Duration	Week													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
1.Introduction															
- Crete a detailed project plan	Two Weeks														
- Define project Objective and Challenges.	Three Days														
- Engage with stake holders	Two Day														
2.Research															
- Literature Review	Five Weeks														
3.Methodology															
- Decide which tools to use and selection of software for system	Ten Days														
- Create a proposed framework	Ten Days														
4.Documentation															
- Modify the work plan and Gantt Chart	One Week														
5.Project Assessment															
- Determine the project risk and the project outcome	One Week														
Touch Up and Submit Planning Document															

This picture indicates milestone.

In the Introduction section, this is where the project's initial preparations are addressed. For first task student involves creating a detailed project plan, which will serve as a roadmap for the entire semester. This task is expected to take two weeks to ensure thorough planning and scheduling. Next, the project's objectives and challenges are defined within three days. The student must identify the project's scope and limitations, for example focusing on the challenges related to tracking a person throughout their time in the library. Student engagement is another crucial step, which takes two days to communicate with mentors and clarify the project's feasibility and scope.

The Research section is allocated the most time, lasting five weeks. During this period, students will conduct an extensive literature review on the topic of human following robots. By gathering a diverse range of research papers, students will gain a comprehensive understanding of the subject matter, ensuring that they explore various theories and concepts related to the project's objectives.

The Methodology section, taking three weeks, student would be involved in applying the knowledge gained from the research phase. Students will decide on the tools and software to be used in the project, choosing those that align with the objectives. Following the tool selection, a proposed framework will be developed which will illustrating how all the components will interact and work together to create the human following robot. This stage is crucial as it determines the architecture of the robot and how it will function in real-world scenarios.

In the Documentation section, students will modify the work plan and Gantt Chart to reflect any changes or adjustments made during the project. This step is important as it maintains a clear and updated overview of the project's progress for supervisor and student to refer at, ensuring that the student stays on track with the project timeline.

Finally, the Project Assessment section which last one week, involves student evaluating the project's risks and potential outcomes. They will assess the challenges faced, potential issues, and the overall feasibility of the robot's implementation. This assessment will help identify areas of improvement and ensure the project's success.

4.2 Capstone 2 Gantt chart

Activities/ Task	Duration	Week													
		15	16	17	18	19	20	21	22	23	24	25	26	27	28
1. Hardware and Software set up															
- Set up ROS on laptop and Reeman Big Dog Chassis	One Week														
- Assemble all the components on the Reeman robot	One Week														
- Set up, train data and test human detection SSD	One Week														
- Implement colour based identification HSV	One Week														
- Set up trajectory prediction algorithm	One Week														
- Implement SLAM navigation using LIDAR	One Week														
- Implement obstacle avoidance algorithm	One Week														
- Implement recovery module (Last Observed Position, Last Predicted Position, and Searching State)	One Week														
2. Testing and feedback															
- Testing the robot after all integrations	One Week														
- Fine tuning for performance optimization	One Week														
- Real world implementation and evaluation	One Week														
3. Documentation															

- Documentation and final report writing	One Week														
- Edit capstone 1 report	One Week														

 This picture indicates milestone.

The project will start with setting up the necessary hardware and software components which will take around one week. This includes configuring the Robot Operating System (ROS) on a laptop and integrating it with the Reeman Big Dog Chassis. All the robot's physical components, such as wheels, sensors, and cameras, will be assembled on the Reeman robot to create a functional platform.

The robot's core functionality will rely on human detection student would install Single Shot Multibox Detector (SSD) algorithm which takes up one week. Additionally, a colour-based identification system using HSV (Hue, Saturation, Value) will be implemented to further enhance the accuracy of target detection.

On the next week a trajectory prediction algorithm will be incorporated to ensure smooth and efficient tracking. This enables the robot to anticipate the human target's movements. The implementation of SLAM (Simultaneous Localization and Mapping) navigation using LIDAR (Light Detection and Ranging) will provide accurate localization and mapping of the environment.

Continuously for the next week student would also code the obstacle avoidance algorithm to safely navigate through dynamic and crowded environments while avoiding collisions. In order to combat missing target problem, a recovery module will be integrated which includes strategies such as Last Observed Position, Last Predicted Position, and Searching State, enabling the robot to find and re-identify the target in case of temporary disruptions or occlusions.

Testing and feedback will play a critical role in the development process which takes around three weeks. The robot will undergo rigorous testing after all integrations to verify its performance. Fine-tuning will be carried out to optimize the robot's behaviour and response in various scenarios. Real-world implementation and evaluation will be conducted to assess the robot's capabilities in practical environments.

Throughout the project from week 5 to week 14, documentation will be maintained by the student. Which will including progress reports, technical details, and results. A final report will be prepared, summarizing the project's development process, challenges faced, and solutions implemented. By the end of the project, a fully functional Ground Human Following Robot will be showcased, capable of autonomously tracking and following designated human targets in real-world environments.

4.3 Project Risk

Type of Risk	Description
Algorithm not compatible	For the recovery algorithm, student use the Kalman filter to replace the linear regression for tracking target, in theory it is executable, but the formula and implementation is different. Trail and errors needed to be done to text out the theory, worst case scenario a simple Kalman filter can be implemented to predict turning corners.
Processor Overload	Due to the high load of computational power needed because of various algorithms running at the background SSD for human detection which will take up large chunk of processing power and memory not to mention both SLAM algorithms that are used for tracking. Trial and error needed to be done to ensure that the process run smoothly without overloading the processor.
Tracking algorithm efficiency	Based on research student conclude that Kalman filter is the best option but in practice real world environment of library the result would be different.
SLAM algorithm efficiency	On paper the navigation algorithms show promising result but in real world implementation might have a different result. Student also unsure if the processor can handle both algorithms AMCL and Cartography at once.
Time Constraint	Due to the large amount of work must be done, time might be an issue as problems or error might arise that needs time to address not to mention the complicated algorithm that needed to be implemented.

4.4 Project Outcome

The project outcome is the successful development of a Human Following Robot capable of accurately identifying and tracking humans in a library setting. The robot utilizes various sensors and modules, such as a camera and other perception sensors, to detect and recognize human targets based on their characteristics like body height and clothing colour.

What do we expect from the human following robot to do:

1. Human Detection:
The robot can identify between object and human and among the human identified, robot use a method to identify the target through their shirt colour to identify the target.

2. Reliable Tracking Mechanism:
After the target is locked, a reliable tracking mechanism must be in place to ensure that the robot can successfully follow the human whichever direction they more. Ensuring that the robot will handle occlusion and other external environut interruptions in the library. S reliable tracking mechanism must be in place to reduce the chance of needing to activate re-identification of lost target, thus reducing the chance of recovery.

3. Re-Identification for Lost Targets:
As the human suddenly moves quicky or turn a steep comer, the robot would find a way to re-identify the targets whereabouts again. This is done though LOP, LPP and searching state using trajectory prediction algorithm.

Chapter 5 – Conclusion

After going through the while project of CP1 student realize the importance of a preparation of report writing and research. Without concrete understanding of the scope, student would face a hard time building a project, an inconsistent plan would cause the project to fail in the end because of unclear instructions and direction.

In this initial project planning student is gathered all the requirements that are necessary for it to complete the task which the robot is set out to do. Thought the objectives he gathers all the components that are required for the human following robot. Through literature revie student manage to do a comparison between all methods that can be found online, giving ideas and various implementations of the project. In the end methodology helps put all the pieces together by constructing a proposed method for the continuation of the project implementation on CP2.

There are a few worries that student suggested one of it is unable to implement the formula for recovery method because of algorithm compatibility risk. This issue can be addressed through more research on the implementation of the formula and continuous testing.

To conclude this report of project planning for the human following robot, it provides a good base of understanding of the project to execute the next step ahead. Student can refer to this document as he continues CP2.

Reference List

1. E. Petrović, A. Leu, D. Ristić-Durrant, and V. Nikolić, “Stereo Vision-Based Human Tracking for Robotic Follower,” International Journal of Advanced Robotic Systems, vol. 10, no. 5, p. 230, Jan. 2013, doi: 10.5772/56124.
2. R. Algabri and M.-T. Choi, “Deep-Learning-Based Indoor Human Following of Mobile Robot Using Color Feature,” Sensors, vol. 20, no. 9, p. 2699, May 2020, doi: 10.3390/s20092699.
3. M. Do and C. H. Lin, Embedded human-following mobile-robot with an RGB-D camera. 2015. doi: 10.1109/mva.2015.7153253.
4. R. Algabri and M.-T. Choi, “Target Recovery for Robust Deep Learning-Based Person Following in Mobile Robots: Online Trajectory Prediction,” Applied Sciences, vol. 11, no. 9, p. 4165, May 2021, doi: 10.3390/app11094165.
5. S. Park and S. Hashimoto, "Autonomous Mobile Robot Navigation Using Passive RFID in Indoor Environment," in IEEE Transactions on Industrial Electronics, vol. 56, no. 7, pp. 2366-2373, July 2009, doi: 10.1109/TIE.2009.2013690.
6. S. Jia, L. Zhao, X. Li, W. Cui and J. Sheng, "Autonomous robot human detecting and tracking based on stereo vision," 2011 IEEE International Conference on Mechatronics and Automation, Beijing, China, 2011, pp. 640-645, doi: 10.1109/ICMA.2011.5985736.
7. K. Seemanthini and S. S. Manjunath, “Human Detection and Tracking using HOG for Action Recognition,” Procedia Computer Science, vol. 132, pp. 1317–1326, Jan. 2018, doi: 10.1016/j.procs.2018.05.048.
8. W. Rahmaniar and A. Hernawan, “Real-Time Human Detection Using Deep Learning on Embedded Platforms: A Review,” Journal of Robotics and Control (JRC), vol. 2, no. 6, Jan. 2021, doi: 10.18196/jrc.26123.
9. M. F. Aslan, A. Durdu, K. Sabanci, and M. A. Mutluer, “CNN and HOG based comparison study for complete occlusion handling in human tracking,” Measurement, vol. 158, p. 107704, Jul. 2020, doi: 10.1016/j.measurement.2020.107704.
10. T.-H. Tsai and C. Y. Yao, A Real-time Tracking Algorithm for Human Following Mobile Robot. 2018. doi: 10.1109/isocc.2018.8649982.
11. L. Tan, T. Huangfu, L. Wu, and W. Chen, “Comparison of YOLO v3, Faster R-CNN, and SSD for Real-Time Pill Identification,” Research Square (Research Square), Jul. 2021, doi: 10.21203/rs.3.rs-668895/v1.
12. O. Biglari, R. Ahsan, and M. Rahi, “Human Detection Using Surf And Sift Feature Extraction Methods In Different Color Spaces,” The Journal of Mathematics and Computer Science, vol. 11, no. 02, pp. 111–122, Jul. 2014, doi: 10.22436/jmcs.011.02.04.
13. R. Deepa, E. Tamilselvan, E. Abrar, and S. Sampath, Comparison of Yolo, SSD, Faster RCNN for Real Time Tennis Ball Tracking for Action Decision Networks. 2019. doi: 10.1109/icacce46606.2019.9079965.
14. E. Karami, S. Prasad, and M. S. Shehata, “Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images,” ResearchGate, Nov. 2015, [Online]. Available: https://www.researchgate.net/publication/318194587_Image_Matching_Using_SIFT_SURF_BRIEF_and_ORB_Performance_Comparison_for_Distorted_Images
15. W. Lin, X. Ren, J. Hu, Y. He, Z. Li, and M. Tong, “Fast, robust and accurate posture detection algorithm based on Kalman filter and SSD for AGV,” Neurocomputing, vol. 316, pp. 306–312, Nov. 2018, doi: 10.1016/j.neucom.2018.08.006.
16. M. Bansal, M. Kumar, and M. Kumar, “2D object recognition: a comparative analysis of SIFT, SURF and ORB feature descriptors,” Multimedia Tools and Applications, vol. 80, no. 12, pp. 18839–18857, Feb. 2021, doi: 10.1007/s11042-021-10646-0.

17. K.-T. Song et al., Navigation Control Design of a Mobile Robot by Integrating Obstacle Avoidance and LiDAR SLAM. 2018. doi: 10.1109/smc.2018.00317.
18. Q. Zou, Q. Sun, L. Q. Chen, B. Nie, and Q. Li, "A Comparative Analysis of LiDAR SLAM-Based Indoor Navigation for Autonomous Vehicles," IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 7, pp. 6907–6921, Jul. 2022, doi: 10.1109/tits.2021.3063477.
19. J. Zhao, S. Liu, and J. Li, "Research and Implementation of Autonomous Navigation for Mobile Robots Based on SLAM Algorithm under ROS," Sensors, vol. 22, no. 11, p. 4172, May 2022, doi: 10.3390/s22114172.
20. R. Yagfarov, M. Ivanou, and I. Afanasyev, Map Comparison of Lidar-based 2D SLAM Algorithms Using Precise Ground Truth. 2018. doi: 10.1109/icarcv.2018.8581131.
21. Q. H. Nguyen, P. Johnson, and D. Latham, "Performance Evaluation of ROS-Based SLAM Algorithms for Handheld Indoor Mapping and Tracking Systems," IEEE Sensors Journal, vol. 23, no. 1, pp. 706–714, Jan. 2023, doi: 10.1109/jsen.2022.3224224.
22. Z. Xuexi, L. Guokun, F. Genping, X. Dongliang, and L. Shiliu, SLAM Algorithm Analysis of Mobile Robot Based on Lidar. 2019. doi: 10.23919/chicc.2019.8866200.
23. Y.-B. Cheng and G. Wang, Mobile robot navigation based on lidar. 2018. doi: 10.1109/ccdc.2018.8407319.
24. J. Meng, S. Wang, Y. Xie, L. Jiang, G. Li, and C. Liu, "Efficient re-localization of mobile robot using strategy of finding a missing person," Measurement, vol. 176, p. 109212, May 2021, doi: 10.1016/j.measurement.2021.109212.
25. M. D. Hoang, S.-S. Yun, and J. Choi, The reliable recovery mechanism for person-following robot in case of missing target. 2017. doi: 10.1109/urai.2017.7992828.
26. M. Ota, H. Hisahara, Y. Ishii, T. Ogitsu, H. Takemura, and H. Mizoguchi, Recovery function for human following robot losing target. 2013. doi: 10.1109/iecon.2013.6699818.
27. A. Hamlet and C. D. Crane, "Robotic behavior prediction using hidden Markov models," ResearchGate, Dec. 2014, [Online]. Available: https://www.researchgate.net/publication/269040508_Robotic_Behavior_Prediction_Using_Hidden_Markov_Models
28. P. K. Panigrahi and S. K. Bisoy, "Localization strategies for autonomous mobile robots: A review," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 8, pp. 6019–6039, Sep. 2022, doi: 10.1016/j.jksuci.2021.02.015.
29. M. Kim, "An Architecture for Person-Following using Active Target Search," arXiv.org, Sep. 24, 2018. https://arxiv.org/abs/1809.08793#:~:text=Toward%20this%20end%2C%20an%20active,conditions%20such%20as%20crowded%20environments.
30. E. Dan Zhang and E. Bin Wei, "Human–Robot Interaction: Control, Analysis, and Design," Newcastle-upon-Tyne, UK: Cambridge Scholars Publishing, 2020. [Online]. Available: https://research.ebsco.com/linkprocessor/plink?id=dce89277-271f-367a-9ffd-5e6e6b548ae1.