VISION BASED REAL TIME OBSTACLE DETECTION AND HUMAN MOTION TRACKING SYSTEM FOR MOVEMENT OF INDOOR AUTONOMOUS MOBILE ROBOT

by

Enes Gönül

Submitted to Graduate School of Natural and Applied Sciences
in Partial Fulfillment of the Requirements
for the Degree of Master of Science in
Electrical and Electronics Engineering

Yeditepe University
2019

VISION BASED REAL TIME OBSTACLE DETECTION AND HUMAN MOTION TRACKING SYSTEM FOR MOVEMENT OF INDOOR AUTONOMOUS MOBILE ROBOT

APPROVED BY:

Prof. Dr. Cem Ünsalan ....
(Thesis Supervisor)
(Marmara University)

Prof. Dr. Duygun Erol Barkana ....
(Yeditepe University)

Assoc. Prof. Dr. Engin Maşazade ....
(Marmara University)

DATE OF APPROVAL: ..../..../2019

ACKNOWLEDGEMENTS

I would like to thank all the people who supported my educational life that I needed to complete my thesis. I would like to special thank Cem Ünsalan, who is my thesis advisor. He has always contributed to me being a better engineer with his visionary approach and recommendations. Also, his wonderful academic career has inspired me. It's my best chance to work with him.

Of course, each member of my family is the inspiration of my entire life and my only motivation in setting new goals. I am ready to do everything I can to be worthy of the love and respect in your eyes. I'm glad you're there and it's good that you're my family, I'm very, very in love with them.

I would like to dedicate this thesis work to my grandparents, my life idols. You are very special people for me, you are all my own legends. Your loves, your personalities, your beings and your longings will be with me throughout my life. Osman Nuri Gönül, Necati Fidan, Fatma Gönül, Melek Fidan ...

ABSTRACT

VISION BASED REAL TIME OBSTACLE DETECTION AND HUMAN MOTION TRACKING SYSTEM FOR MOVEMENT OF INDOOR AUTONOMOUS MOBILE ROBOT

During the collection of goods in the warehouse, the person performing the work is expected to collect both the correct goods and do it quickly. That's why he needs to carry the vehicle he collects and follow up the orders correctly. As in this project, systems such as Automated Guided Vehicle and Autonomous Mobile Robot prevent time and power consuming tasks such as pushing and pulling. The aim of this thesis is to develop software and hardware that will enable people collecting goods to follow their vehicles without pushing and pulling. This system will work as a fully embedded system with processor and cards to be positioned on the vehicle. The system, which will be formed with hardware based on vision based sensors and algorithms, will be able to follow the person in front of it by processing and understanding the data received and to continue the movement without accident by detecting obstacles. RGB camera and LIDAR sensor data are processed by the processing board and able to detect obstacle and tracking the identified person.

ÖZET

İÇ MEKAN OTONOM MOBİL ROBOTLARIN HAREKETİ İÇİN GERÇEK ZAMANLI GÖRÜNTÜ TABANLI ENGELDEN KAÇINMA ve İNSAN TAKİP SİSTEMİ

Depodaki malların toplanması sırasında, işi yapan kişinin hem doğru malları toplaması hem de hızlıca yapması beklenir. Bu nedenle topladığı aracı doğru bir şekilde taşıması ve siparişleri doğru bir şekilde takip etmesi gerekiyor. Bu projede olduğu gibi, Otomatik Kılavuzlu Araç ve Otonom Mobil Robot sistemleri, itme ve çekme gibi zaman ve güç tüketen işleri önler. Bu tezin amacı, mal toplayan kişilerin araçlarını itmeden ve çekmeden takip etmelerini sağlayacak yazılım ve donanım geliştirmektir. Bu sistem, araçta konumlandırılacak işlemci ve kartlarla tamamen gömülü bir sistem olarak çalışacaktır. Görüntüleme tabanlı algılayıcı ve algoritmalara dayalı donanım ile oluşturulacak olan sistem, alınan verileri işleyerek ve anlayarak önündeki kişiyi takip edebilecek ve engelleri tespit ederek kazasız hareket etmeye devam edebilecektir. RGB kamera ve LIDAR sensör verileri, işlemci kartı ve yazılımı tarafından işlenir ve robotun tespit edilen kişiyi engelleri de tespit edip kaçınarak takip etmesi sağlanır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS/ABBREVIATIONS

2D	Two dimensional
3D	Three dimensional
AGV	Automated guided vehicle
AH	Ampere hour
AI	Artificial intelligence
AMR	Autonomous mobile robot
API	Application programming interface
CAN	Controller area network
CMOS	Complementary metal oxide semiconductor
CNN	Convolutional neural network
COT	Cotangent
CPU	Central processing unit
DPTAM	Dense piecewise planar tracking and tracking
DRAM	dynamic random access memory
DSO	Direct sparse odometry
DSP	Digital signal processing
DWA	Dynamic window approach
ECO	Efficient convolution operator
FIFO	First in first out data channel
FOV	Field of view
fps	frame per second
GB	Giga byte
GB/s	Giga byte per second
GPIO	General-Purpose input/output
GPU	Graphical processing unit
I2C	Inter-Integrated circuit
IMU	Inertial measurement unit
IO	Input-Output

init	Initialize signal
LIDAR	Light detection and ranging
MB	Mega byte
MHz	Mega hertz
ORB	Oriented fast and rotated brief
OS	Operating system
PCB	Printed circuit board
PCL	Point cloud library
RCNN	Region-based convolutional neural network
Re-ID	Reidentification
RGB	Red green blue
ROS	Robotic operating system
RPM	Revolutions per minute
RTT	Rapidly-exploring random tree
SDK	Software development kit
SLAM	Simultaneous localization and mapping
SONAR	Sound navigation and ranging
SPI	Serial peripheral interface
SSD	Single shot detector
UAV	Unmanned aerial vehicle
USB	Universal serial bus
UV	Ultraviolet
V	Volt
W	Watt
Wi-Fi	Wireless fidelity

1. INTRODUCTION

In the light of technological developments, the integration of automation and robotic processes is rapidly increasing in many sectors and working areas. The use of robots is also increasing in places with indoor use, such as factories and warehouses. These robots are used in direct production lines as well as in many transport and storage processes.

The subject of this thesis is a Autonomous Mobile Robot intended to be used to transport goods in warehouses. In order to provide driving safety, decreasing labor cost and time savings, many autonomous robots began to navigate in the warehouses. Product and order picking robots, inventory robots, unmanned areal robots and self driving forklifts are being used mostly by the warehouses shown in Figure 1.1.

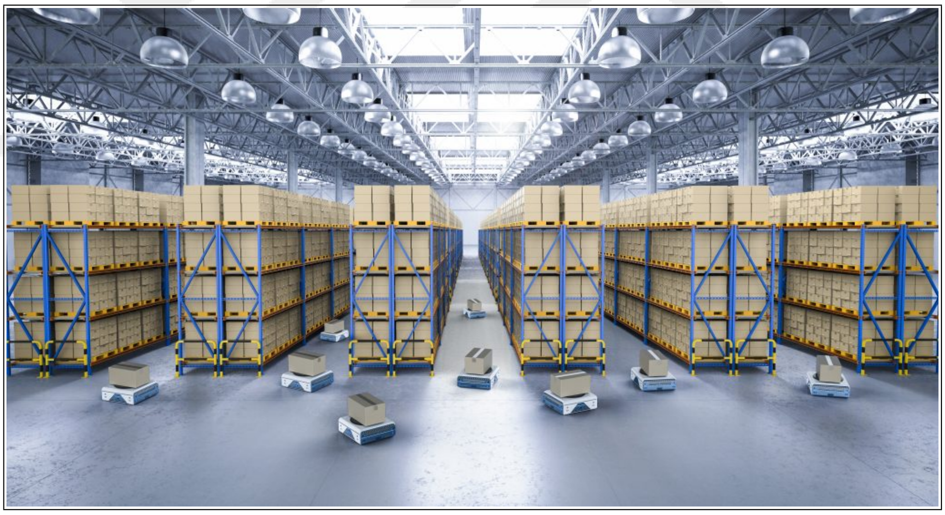


Figure 1.1. AMR in warehouse

In most of the logistics and warehousing processes still using manual paper based picking with human based push and pull vehicles. Autonomous Mobile Robots (AMRs) and Automated Guided Vehicles (AGVs) are started to using and decreasing the lots of unnecessary walking and pushing. AGVs and AMRs are not to be needed to remote control by the human or computer, they are autonomous and have on board computing systems. AGVs are the vehicle can follow to the constant lines or routes by its sensors but AMRs have the advanced sensors and processing big data and meaning them by the way they can navigate them for any routes flexibility and guide by artificial intelligence and machine learning methods.

The robot which is designed and used for this thesis is the AMR how navigate itself and follow the identified people in the logistic warehouses. Autonomous driving method of this robot is based on computer vision. AMRs have different types of vision hardware and software varieties for navigation. Our proposed method is the hybrid system of 2D camera and LIDAR sensor and the detection and navigation algorithm on these hardware structure.

The thesis organization flow is designed as follows. Literature reviews and searches are listed and mentioned according to the parts of the project in Chapter 2. In Chapter 3, all hardware and software are detailed and also general structure of the robot. In Chapter 4, we provide the detailed explanation of the mathematical methods in 3 subsections as simultaneous localization and mapping, human detecting and human tracking, and autonomous mobile robot navigation. The experiment result and implementation of all the elements of the system are detailed and also discussions are about the state of the art and the proposed work in the autonomous mobile robot searches are matched with the our system in Chapter 5. Finally in Chapter 6, we summarize our proposed prototype system and make inferences about the usage experience and mentioned possible future works to improve and develop.

2. LITERATURE REVIEW

In recent years in line of the developments of the unmanned vehicle there have been many varieties as areas of use both military and civilian scenarios. Remote controlled vehicles and autonomous vehicles are classified as unmanned vehicles. These types of vehicles can be examined in many sub-categories in terms of the variety of used environment, control modes and autonomous driving algorithms.

2.1. AGV AND AMR

Autonomous vehicles and robots became more useful and popular for many industries and markets areas. These autonomous devices are divided into two main parts as AGV (Automated Guided Vehicle) and AMR (Autonomous Mobile Robot). According to the their abilities and the working principles they are differed from each others. AGVs are only be moved through its guided that is actually a line or magnetic differences but AMRs have the ability of definition of their enviroment and could be moved because of their hardware and software facilities itself.

AGVs and AMRs are an intelligent robot controlled by central system, working autonomously without the need of manpower, used in storage, transportation, intelligent production and wide application areas of the industry. The first AGV in 1953, Arthur Barrett Jr. It was developed by an attractive and called it Guide-O-Matic. A rail system has been developed for the vehicle following the ground cables. The first guidance system came about when the sensors under the tractor were looking for a magnetic field. The magnetic field was created by current passing through a cable or a series of cables on the ground. With Guide-O-Matic, Barrett Electronics of Northbrook has become a major logistics and distribution center equipment provider.

Today, technology continues to evolve. The cable on the floor, the magnetic tape is still used for suitable situations. However, today many systems, laying cables, are designed without magnetic tape. The vehicles are now equipped with computers to communicate, direct and manage the system and communicate instantly with the center. First class sensors are used to identify the obstacle and, where necessary, the operator. Thus, in case of emergency (obstacle, stop, impact), the vehicle can be controlled centrally.

Auto-guiding tools can be tracked in many ways. In some applications, aluminum tape is laid on the floor according to the way the robot will follow. A certain number of inductive sensors are placed under the vehicle and these sensors detect the aluminum line and guide the vehicle [1]. White tape is drawn on a dark background and the vehicle is able to process the image with a front-mounted camera to follow the line [2]. The Baoguo and Chunxi tool uses virtual line tracking, which eliminates the need for tapes or other types of guides. With sonar sensors, the vehicle detects obstacles and creates a new virtual path to follow [3]. Norhashim, Mohamad and Noorfadzli used a single infrared sensor to move the vehicle and monitor a ground-based line. Instead of using multiple sensors, he used two different color-tone lines on the white background surface to determine the position of the vehicle relative to the line [4]. As another method, it is possible to find the way of working environment by removing the digital map and loading it into the vehicle or by making a digital map with the machine learning method in semi-autonomous mode. Obstacle recognition and safe motion can be gained with laser and lidar sensors. The patents of the automatic guided tools that we have reached as a result of the literature surveys were also examined. A highly innovative approach to order picking is the Kiva warehouse automation system purchased in 2012 by e-commerce giant Amazon. Kiva aims to increase the collection efficiency by moving the shelves of a warehouse with an autonomous vehicle which is shown in Figure 2.1.

Allows the order selector to remain at a point when Kiva moves. With intelligent control software, Kiva vehicles can significantly reduce the collection cycle time by tracking barcode labels on the warehouse floor [5]. FIFI is the pilot of the order picking tool developed by BAR Automation and KIT. This intelligent self-propelled tool can track the order picking process using vision guidance and respond to human movements (such as a rocking hand) that instructs the order picker to follow. FIFI is shown in Figure 2.2 is also able to autonomously transport loads to the specified cycle points.

Introduced by Kuka Robot Group in 2017, KMP 1500 which is shown in Figure 2.3. AGV

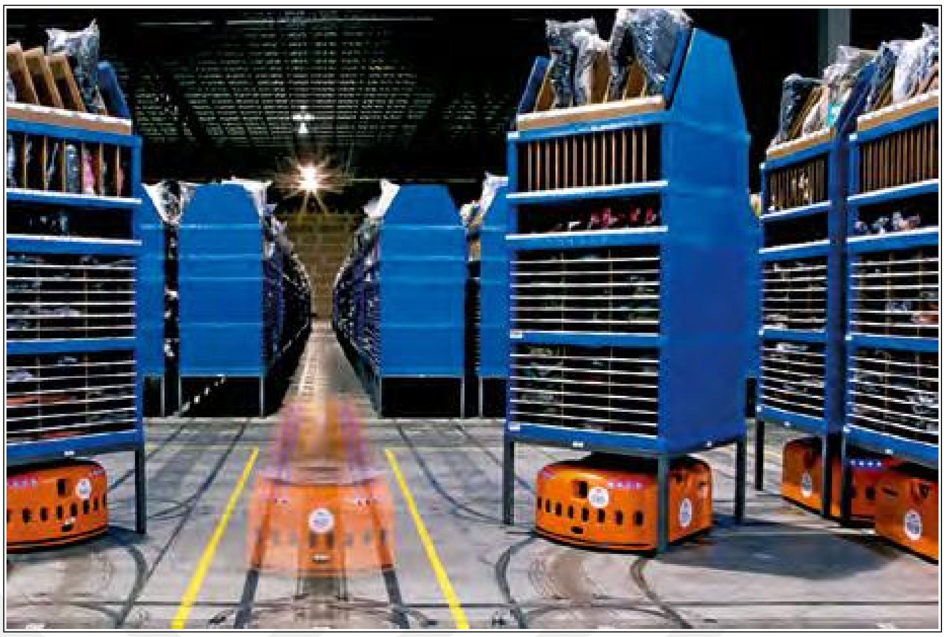


Figure 2.1. Kiva robots in warehouse

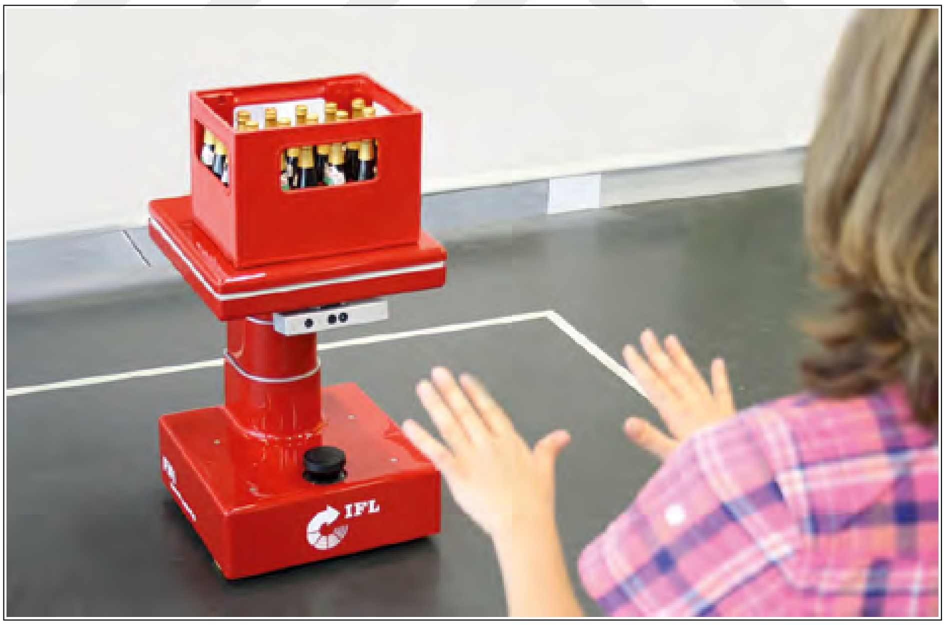


Figure 2.2. FIFI robot in warehouse

model is able to carry out its versatile movement accurately and stably up to +/- 1 mm thanks to its mechanic tire mechanics. Optionally, the product's advanced logistics and warehouse management algorithms improve warehouse and freight handling efficiency as much as possible. 360 LIDAR sensors ensure safe and accurate driving.

Figure 2.3. Kuka robot in warehouse

2.2. SIMULTANEOUS LOCALIZATION AND MAPPING

SLAM stands for simultaneous localization and mapping. This method must be used for the autonomous vehicle and robots to sense and understand to the enviroment of the system. Localization and mapping are the important needed for the robot. Systems must be know the drive area and their environments. Especially autonomous indoor vehicles and robots are using this method to navigate themselves. Various types of SLAM have been developed according to the using hardware and system requirements. According to the used algorithm the autonomous mobile robot systems could use much various sensors and systems like SONAR, LIDAR, altimeter, Inertial Motion Unit, RGB and RGB-Depth cameras. Developers and system designers mostly began to use SLAM methods based on signal processing and computer vision algorithms. SLAM method and algorithms must be fast, precises and robust in real time usage as autonomous mobile robots. Usage of all sensors and high mathematical models has surely provide perfect solutions but they became high cost and high computational loaded systems.

Since the SLAM coming most popular problems and solution for the robot in real-time, various types of SLAM methods have been developed like MonoSLAM and Parallel Tracking and Mapping PTAM, the number of monocular SLAM-related methods has developed via included of Dense Tracking and Mapping DTAM, Large-Scale Direct Monocular SLAM LSD-SLAM Oriented FAST and Rotated BRIEF ORB-SLAM, Dense Piecewise Planar Tracking and Tracking DPTTAM, Elastic Fusion, Google Cartographer, Direct Sparse Odometry DSO [1]. To increased the result successful match of monocular SLAM performed, we use in this thesis work Google Cartographer as SLAM methods and algorithms because of the system requirements and elements [2,3].

2.3. HUMAN DETECTION AND HUMAN TRACKING

In recent years technological developments, human-robot and robot-robot interaction and collaboration has become a big part of the searching field compared to other field of area. Among various methods of a human-robot and robot-robot interaction is necessary for the successful implementation of various methods and equipment. Autonomous vehicles allow people to follow and move in a predetermined area to provide navigation in such situations. In the last few years, they have increased their application areas and numbers considerably thanks to the tasks they serve and their increasing capabilities. A lot of development and research are being done by using various sensors, hardware and software for the detection, recognition and tracking of humans by robots. These sensors are being used in the area of mobile robotics can be categorized in two main areas; vision based sensors and laser based sensors. At using vision sensors with a camera facilitates tracking humans through extracting features like color based information of them. Mostly and recently some human features get by the testing process and trying to learnt by the machine learning algorithm. However, it has a limit that human has to face a robot in order to get skin color of the human face, because skin color is a distinctive color of human so that vision sensors mainly use it to perceive human. For this reason, some identification features get most important data to get needed accuracy.

In laser sensors, more accurate range information could be get relatively by using laser range finders compared to other sensors. Moreover, another advantage is that laser range finder sensors are not affected by brightness or darkness of surroundings. The quantities of the used laser sensors are proportional to get increasing the accuracy of the range and enviroment detection, this could be changed by the criteria and thresholds of the aims of the project. However, the cost would increase if large numbers of sensors are implemented. The works design a modeling of human walking patterns and tracks humans stochastically through Kalman Filter, a particle filter and other filters [4-6].

2.4. AUTONOMOUS MOBILE ROBOT NAVIGATION

Autonomous Mobile Robots need to find answers; Where I am, What is the next move and How could I get there? SLAM is the answer of the location. SLAM also provide to AMR map to show its enviroment could be get the next move. These step need to clean road by detecting the obstacle. In fact, in order for the robot to perform the motion trajectory given by the path planning system, it certainly needs path planning to know how the robot can reach its position and navigation. Robot navigation is considered from different perspectives and is divided into three main approaches: geometric navigation, anemic navigation and topological navigation. Although according to the their definition and their methods three differ in, same questions have been focused to solve by them. From the beginning, developers have focused on creating metric maps and moving on the map using metric road planners. The best known algorithm for creating metric maps is the simultaneous localization and mapping (SLAM) proposed by Bailey and Durrant-Whyte [2]. Wang et al. as proposed by Bailey and Durrant-Whyte [2]. Wang et al. use SLAM and Rapidly-exploring Random Tree (RTT) planning with Monte Carlo localization to drive a wheelchair in indoor environments. Pfrunder et al. use SLAM and occupancy grids to navigate in heterogeneous environments [3,4]. However, as larger maps are considered, it is computationally expensive to keep metric maps and other authors have focused on topological representations. Fernndez-Madrigal et al. design a hierarchical topological model to drive a wheelchair and perform reactive navigation and path-planned navigation. Ko et al. works with topological maps where nodes are bags of visual words and a Bayesian framework is used for localization [5,6].

3. HARDWARE AND SOFTWARE

In this chapter, the general system elements with its all software and hardware components are mentioned and launched.

3.1. GENERAL SYSTEM STRUCTURE

This section is about the physical structure of the AMR. This robot is designed for the indoor usage in warehouses for picking operations. The Robot needs to run at human speed 1 m/s and is intended to carry 300 kg load. Robot follows the picker person so it should have safety and human friendly physical structure. At the same time, the size of the robot had to be decided according to the warehouse standards in order to move freely between the corridors. Robot is designed with a length of 100 cm and a width of 60 cm. These sizes provide the comfortable driving in the warehouse. Sizes of the robot are in the shown in Figure 3.1.

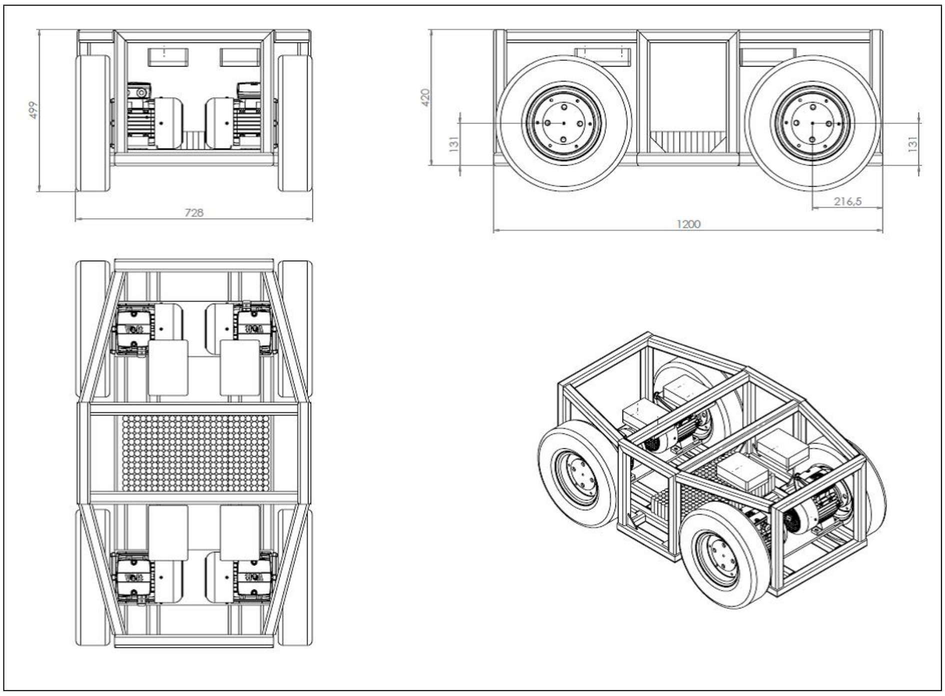


Figure 3.1. Technical sizes drawing of the AMR

The project envisages giving different speed to the right and left wheels instead of steering system to steer the vehicle. In this way, the method of ensuring the rotation of the vehicle and the rotation of the wheels by applying differential speed is referred to as id skid steering in the literature. In this project, the skid steering method will bring mobility to see the adequacy of the real prototype without waiting for the construction of a simple design a proof-of-concept vehicle has been produced. At the same time, this reduced prototype will allow the development of algorithms such as sensor processing, positioning, mapping and autonomous driving to be developed without waiting for the actual prototype to be ready. The chassis design of the reduced prototype vehicle is produced from pieces cut from aluminum profile in dimensions of 120 cm x 85 cm. The rail structure in the profile is used for mounting the motors. In this way, the distance between the wheels and the placement of the engines, which are important in skid steering, are made adjustable. 4 wheels with 44 cm diameter are provided for connection to the motors. These wheels are connected to the shafts of the motors by the coupling structure detailed in mechanical design drawing which is shown in Figure 3.2. Lithium battery use at this stage The use of a reduced prototype was not preferred because it would allow additional time for the original design and whether the cell tensions were safe during operation. Instead, 4x12 V 7 Ah lead acid batteries are connected in series and the battery group is ready and 48 V nominal voltage is obtained which will form the DC bus voltage of the driver cards. The cabling design required for communication between the cards and battery power connections is designed and made to be carried through aluminum profiles. A fuse and switch are placed between the battery group and the motor driver cards. A body cut from the wood is placed on the chassis to accommodate the battery group, to place the sensors at the desired height and depth and to place a laptop on the vehicle in experimental studies.

After the vehicle became ready, experimental studies were started. Here, the ability of the vehicle to make straight forward, straight backward, right and left rotation movements was examined. In experimental studies, it was observed that the vehicle can perform straight forward or straight back movements easily. However, the return movements were not successful. For example, if 10 RPM speed reference is given to the right wheels and 20 RPM speed reference to the left wheels, it is observed that all engines rotate at 10 RPM speed or 20 RPM speed and therefore the angle of the vehicle does not change. Similarly, when the +10 RPM and -10 RPM speed reference was given to the right and left wheel groups, the

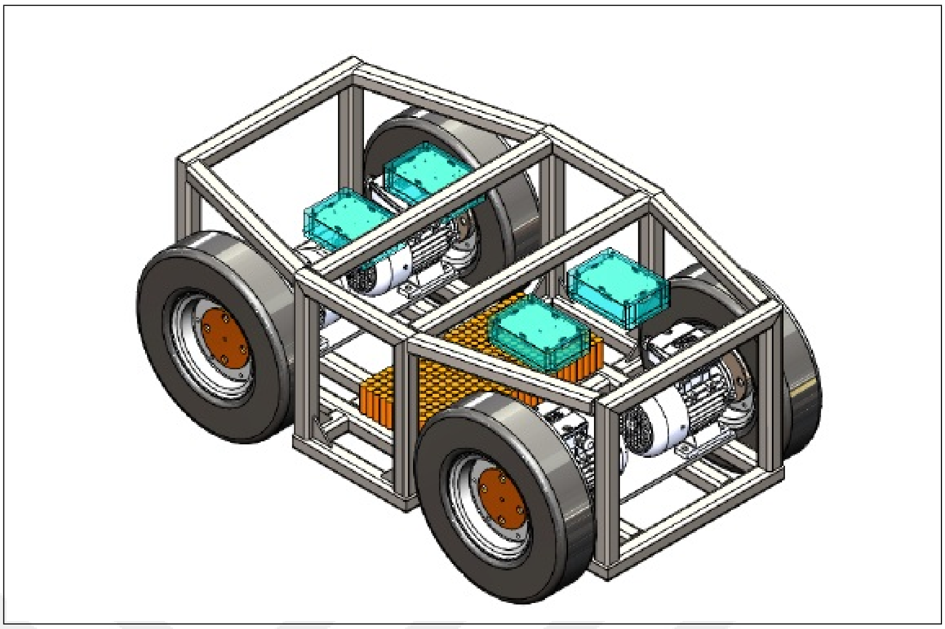


Figure 3.2. Vehicle prototype 3D drawing

vehicle did not move. The main reason for this occurrence is that the wheel must be able to overcome the friction with the ground in order to provide skid steering rotation. It has been recognized that the pneumatic rubber wheels provided have very high adhesion. This is due to the low degree of hardness of the rubber material used and the large surface contact with the ground due to the pneumatic structure of the wheel. In addition, the width of the wheel above 10 cm causes the friction surface to be much larger than desired. In accordance with the information obtained from this experiment, it was decided that the wheel, which would be suitable for use in skid steering method, should be of a filling structure, made of high hardness material and having a small width. New wheels with the specified specifications and a diameter of 30cm are supplied. The diameter of the wheel is reduced to increase the force acting on the vehicle in order to make the rotation movements smoothly.

In addition, since skid steering method reduces the distance between the wheels on the same side of the vehicle, the engines are positioned closer to each other on the aluminum profile since this will increase the turning performance. The image of the rapid prototype vehicle with the new wheels is as shown in Figure 3.3.

After the changes, the maneuverability of the vehicle was tested again. During these tests, it

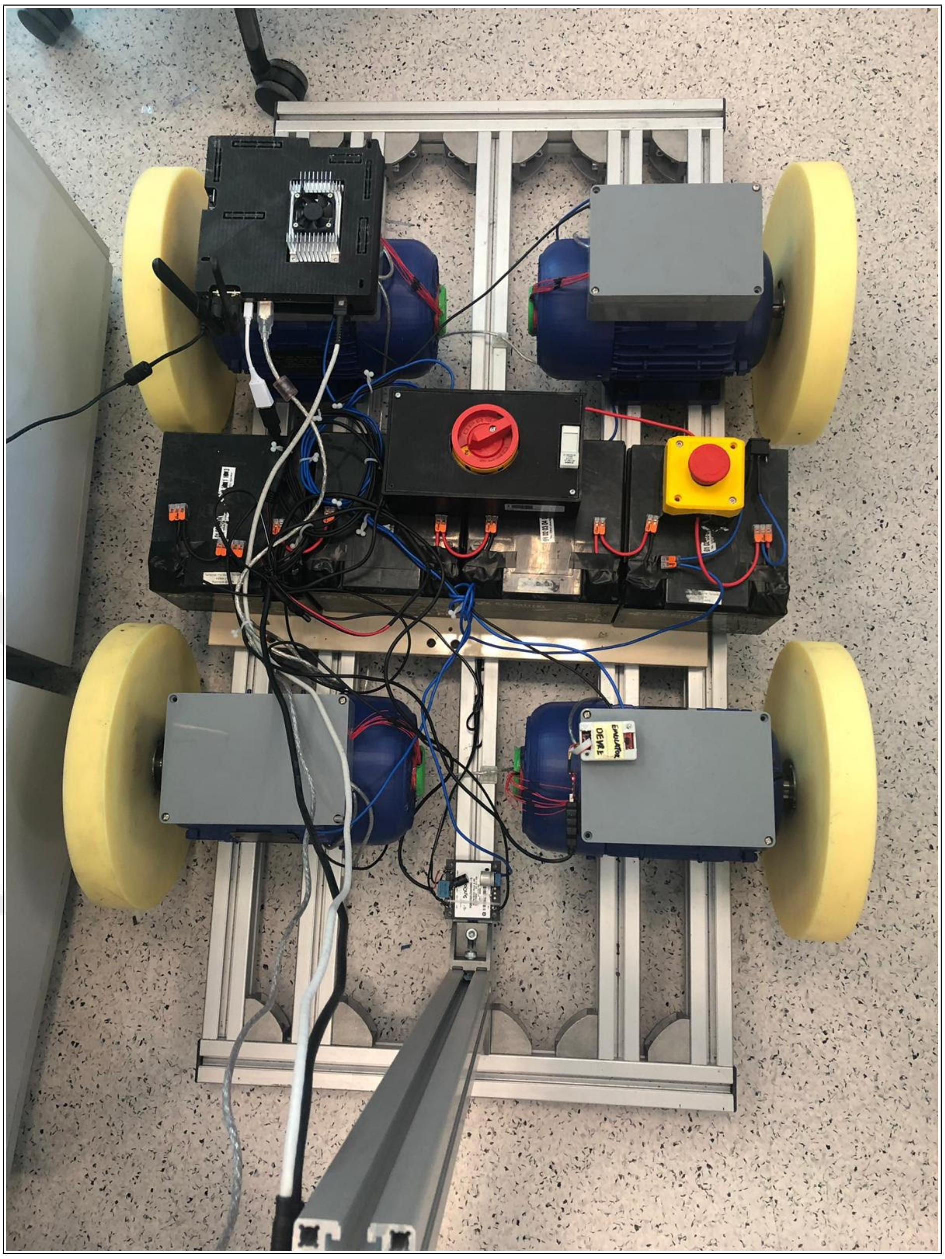


Figure 3.3. Vehicle prototype product

was observed that all the movements, including the turns, could be performed successfully, and when the differential speed command was sent, the right and left wheels rotated at the given speed references and thus the vehicle was able to perform the rotation maneuvers. During the experimental studies, data such as speed and current of all motors were obtained from driver master driver board using FreeMASTER software. The graphs obtained from the motors during operation are shown in the Figure 3.4. Here, it is seen that all motors follow the same speed reference in straight motions and right and left double wheels (Green and orange graphs indicate right, red and blue graphs indicate left wheel speeds) depending on the rotation diameter. The graph is also obtained from the motors power characteristics during operation are shown in the Figure 3.5.

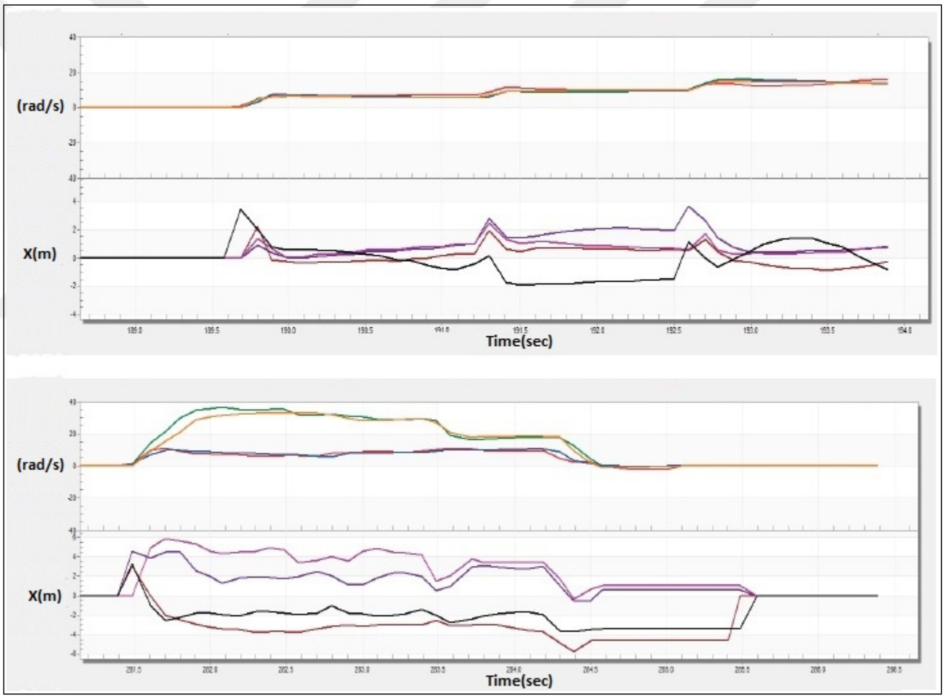


Figure 3.4. Motor velocity and angle outputs

Because of the interaction and working principle of this robot how is collaborative with human and working areas the health and safety rules are the important requirements of the design parameters. Both hardware and software are applied and adapted all the safety rules.The robot is designed to be used indoors. Prototype tests and applications will be carried out in the warehouses determined in line with operational needs. The robot will be equipped with special navigation systems to monitor the operator and to avoid obstacles. In

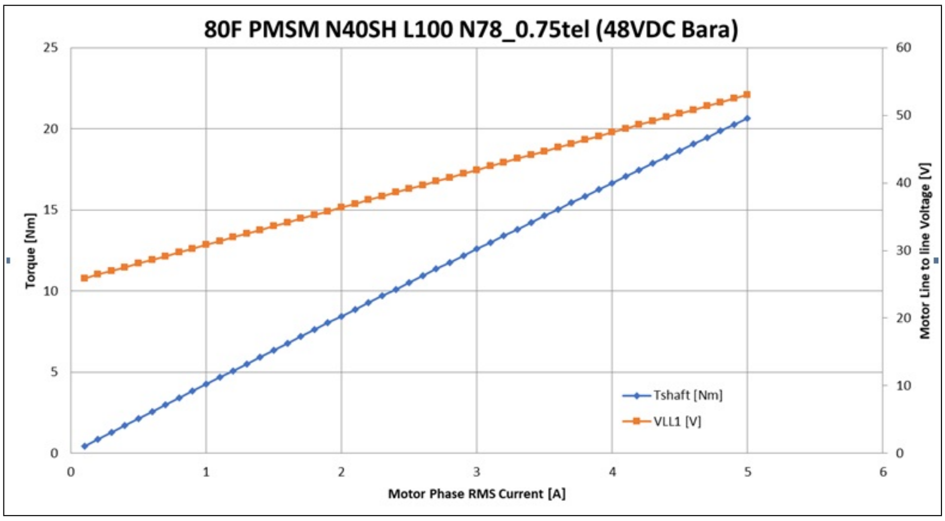


Figure 3.5. Motor testing outputs

the first phase precaution area, the robot will either stop or circulate around the obstacles. In the second phase braking area, the vehicle will be braked suddenly to limit the impact of the obstacle. The brake areas are shown in the Figure 3.6 with the red and blue colours. The brake area will vary depending on the vehicle speed. If an obstacle is encountered in the brake area, the vehicle will enter an emergency stop and stop autonomous activities.

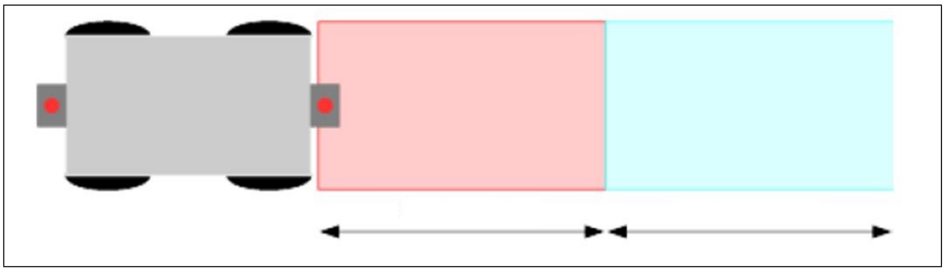


Figure 3.6. Breaking areas of the robot on follow mode

3.2. HARDWARE

Robots components and physical feature are mentioned in the previous section. In this section, our development hardware components are to be detailed. Firstly the main board is Nvidia Jetson TX2 Developer Kit which is the brain of the system and responsible all the robot run and computer vision parts of the robot. After that the sensors which are the data collected from the environment for the robot's navigation such as RGB camera and

Lidar scanner. Sensor, camera and control unit requirements for autonomous driving were determined the system was designed in accordance with the requirements. As in the shown in Figure 3.7.

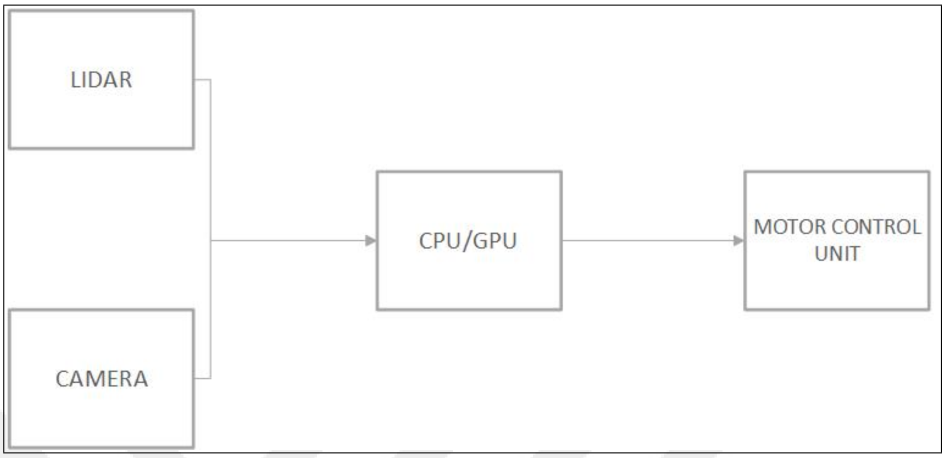


Figure 3.7. Scheme of the sensors and processing unit

Benewake CE-30-C point cloud LIDAR, and a See3CAM CU30 low light USB camera were installed on the vehicle. It was decided to use Nvidia Jetson TX2 development board for processing camera and LIDAR sensor.

3.2.1. Embedded Computer Processing Board

System requirements and the processing load could be handle with the high power processing unit board. Nvidia Jetson TX2 is selected for this reasons as the main board of the system which is shown in Figure 3.8. Nvidia Jetson TX2 is the fastest, most power-efficient embedded AI computing device. This 7.5 W supercomputer on a module brings true AI computing at the edge. It's built around an Nvidia Pascal-family GPU and loaded with 8 GB of memory and 59.7 GB/s of memory bandwidth. It features a variety of standard hardware interfaces that make it easy to integrate it into a wide range of products and form factors. It has powerful CPU as Dual-Core Nvidia Denver 2 64-Bit CPU Quad-Core ARM Cortex-A57 MPCore. Also system needs powerful GPU for the computer vision processes and this card has also high performance GPU as 256-core Nvidia Pascal GPU architecture with 256 Nvidia CUDA cores 100. This Module has also 8 GB L128 bit DDR4 Memory, 32 GB eMMC 5.1 Flash Storage and provide connectivity to 802.11ac Wi-Fi and Bluetooth-Enabled Devices.10/100/1000BASE-T Ethernet module is included on this board.As interface for the environmental components has input and output modules such as; USB 3.0 Type A, HDMI output, GPIOs, I2C, I2S, SPI, CAN connectivity. Nvidia Jetson TX2 is also provide to run on needed operating system and libraries which are open source and the using for computer vision processes. All these software are compatible with this board and also hardware such as camera and LIDAR.

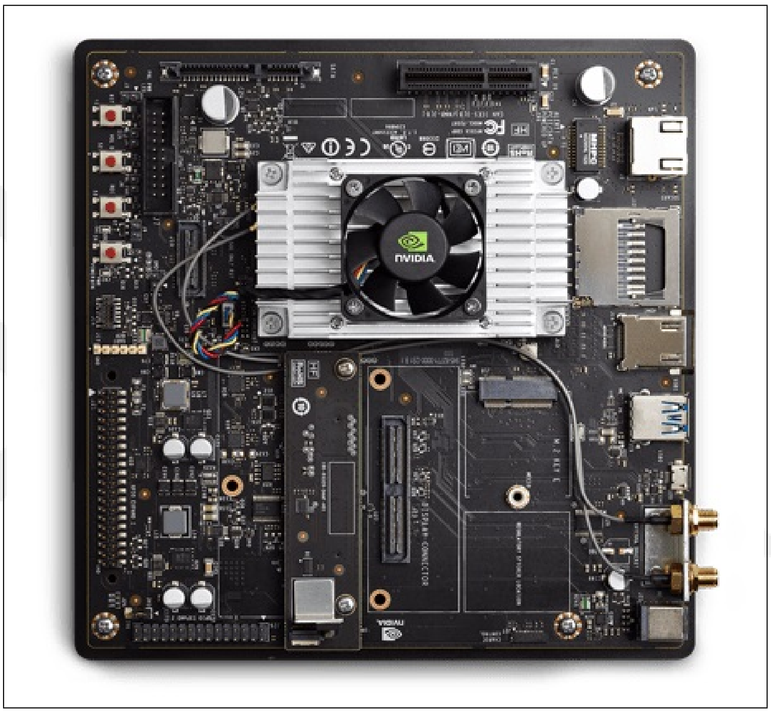


Figure 3.8. Nvidia Jetson TX2

3.2.2. LIDAR Scanner Sensor

Autonomous Mobile Robot needs the simultaneous localization and maps and the real time environmental checking to detect human and obstacle. Many sensors and sensor fusion methods could be used for this purpose. In this thesis work decision is made by the system requirements and low cost needed. Vision system of the robot is the hybrid of the RGB camera and LIDAR sensor. Therefore high level 360 degrees scanner was not chosen because of the cost and high computational load. According to the simulations and computations the low cost but precise Lidar which is launched by Benewake modeled CE30. Benewake CE30-C is a solid-state LIDAR with large FOV which is shown in Figure 3.9. It could simultaneous output grey and depth information within 132 horizontal FOV and 9 vertical FOV. Meanwhile no mechanical rotating component brings higher reliability and stability. Its operational range is between the 0.1 m and 4 m with 320*24 pixel resolution at 20 fps. This Lidar sensor is able to 3D scanning with Point Cloud Data. It is a mass product sensor and its sizes 79 mm x 50 mm and its weight is 219 g.

Figure 3.9. Benewake CE30-C LIDAR sensor

3.2.3. RGB Camera

Our Robot which is designed for the indoor usage in warehouses as a picker robot which follows the relevant person. Actually the systems are only have LIDAR sensor can follow the any movements or any moving objects. Providing o the just selected person following is needed more complicated algorithm or more and high cost sensors fusion. Tracking and following to the identified person are important rules of the system. This vision system has a LIDAR sensor for the obstacle detection and 3D mapping, but a RGB camera is needed to identify specific person to follow. This camera could be handle low light efficiency because of the using area of the robot. Warehouses have more low lighted area commonly. The health and safety rules are most important thing in the human-robot interacted working area so the camera could be streaming needed fps and resolution. Various types of cameras are designed for the diff rent aims and for the different working conditions and their field of view angles are changing as ; omnidirectional cameras have 360 FOV, panoramic cameras around 180 FOV and cameras with wide-angle or fish-eye lenses. Image distortion is the main reason of the dividing of the types of the hardware design. In these cases, standard perspective no longer applies and straight lines no longer appear straight in the image. These requirements are provided by the See3CAM-80 USB camera module shown in Figure 3.10.

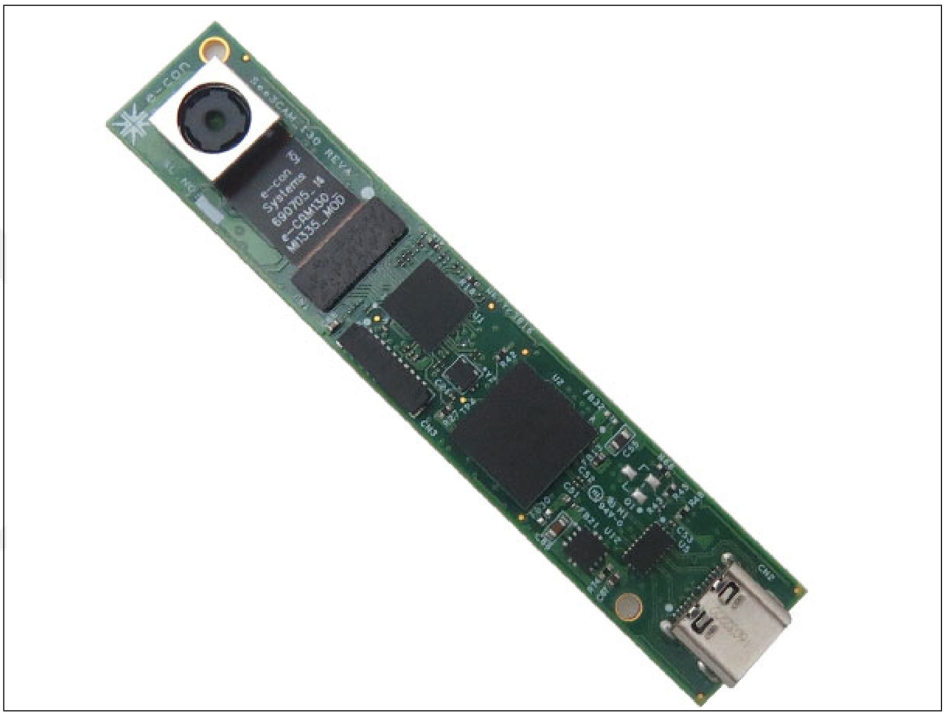


Figure 3.10. See3Cam 8 MP USB camera module

This is a high performance 8MP AF UVC USB camera module based on OV8825 CMOS Image sensor from OmniVision Inc and is compatible with USB 3.0 SuperSpeed interface on the target platform. It is a plug and play setup and usable all operating systems. It could be streaming video in 1080p resolution at 30 fps.

3.3. SOFTWARE

Autonomous Mobile Robot and many complicated robot have more than one software at the same time. For this reason, systems are run as multi processing processors and operating systems which is provide this structure. The main board which is selected for our AMR is Nvidia Jetson TX2 could handle these type operating system version. In this thesis, we constructed the software system which is shown in Figure 3.11.

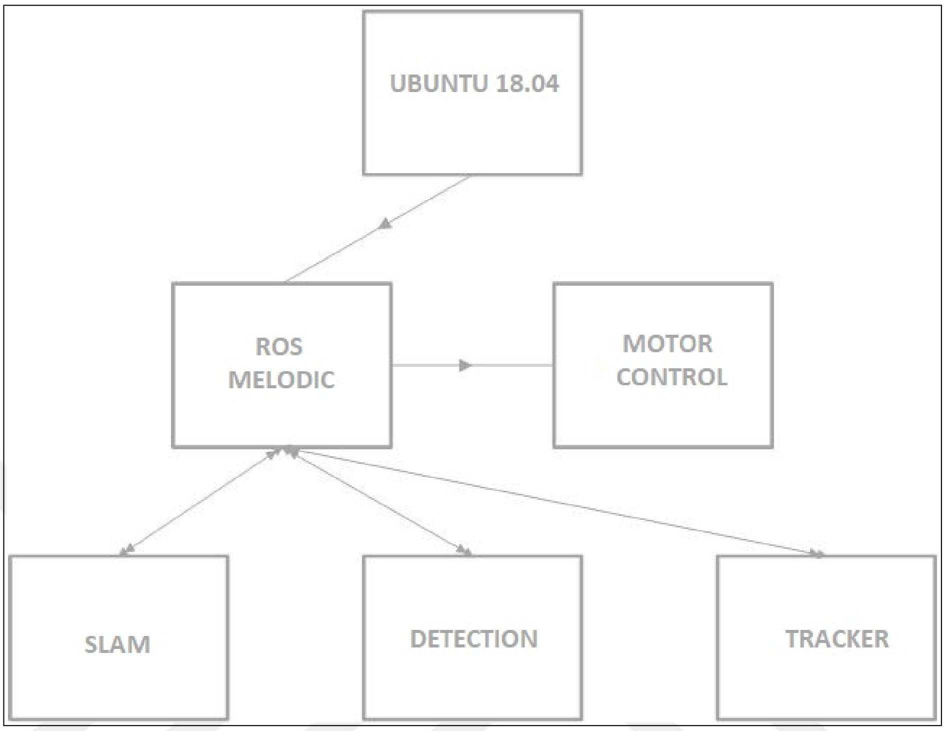


Figure 3.11. Scheme of the software

Ubuntu is the operating system of the our robot. Robotic Operation System is running on the OS actually as a framework to run all hardware, libraries and communicate all parts each other. The other libraries, algorithm and input/output units are running on the operating system. Operating system, robotic operating system, SLAM, tracking and detecting algorithms are mentioned in the subsections of this section.

3.3.1. Operating Systems

The main operating system of the designed robot is Ubuntu 18.04 version. Ubuntu is a complete Linux operating system. Ubuntu is available and usable as freely. Their developers communities and the possibility of professional technical support are more large and easy. The Ubuntu community is built on the ideas enshrined in the Ubuntu Manifesto: that software should be available free of charge, that software tools should be usable by people in their local language and despite any disabilities, and that people should have the freedom to customize and alter their software in whatever way they see fit. Ubuntu is suitable for both desktop and server use so Nvidia Jetson TX2 is also compatible for the usage of Ubuntu operating systems. Ubuntu includes thousands of pieces of software, starting with the Linux kernel version 4.15 and GNOME 3.28, and covering every standard desktop application from word processing and spreadsheet applications to internet access applications, web server software, email software, programming languages and tools and of course several games.

Robotic Operating System (ROS) is also used in this robot, it is actually a framework and set of tools that provide functionality of an operating system on a heterogeneous computer cluster. ROS provides functionality for hardware abstraction, device drivers, communication between processes over multiple machines, tools for testing and visualization. Robotic operating system is the system that provides connection way to a network of different processes (nodes) with a central hub or central system unit. Nodes can be run on multiple devices, and they can be able to connect to that hub in different ways. Alternative approaches like MRPT, CARMEN, LCM, Player, Microsoft RDS and others provide some of those features, but not all. Most of the time, the design downfalls are language support limitations, unoptimized communication between processes or the lack of support for various devices which is arguably the hardest problem to fix. Therefore, ROS is decided to use for our AMR because of the suitable for all our software and hardware. ROS has many distribution version which is developed by the developers and groups. According to the our system requirements and elements of the system ROS Melodic distribution is selected to run on the system.

3.3.2. Simultaneous Localization and Mapping

Autonomous vehicle and robot need to sense their enviroment. This could be solved by fusion of methods and systems. As in the mentioned in the literature search section, there are various alternatives. Elements and the requirements of the systems are important roles when the selection are made. Therefore Google Cartographer is decided as the SLAM and 3D Mapping tool for this robot. Cartographer is a system that provides real-time simultaneous localization and mapping in 2D and 3D across multiple platforms and sensor configurations. That system high level scheme is shown in Figure 3.12. SLAM algorithms combine data from various sensors (e.g. LIDAR, IMU and cameras) to simultaneously compute the position of the sensor and a map of the sensors surroundings. SLAM is an essential component of autonomous platforms such as self driving cars, automated forklifts and robots in warehouses, robotic vacuum cleaners, and UAVs. Cartographer builds globally consistent maps in real-time across a broad range of sensor configurations common in academia and industry. Because of the Google Cartographer's focus is on the LIDAR SLAM, it is selected as the platform for this purpose.

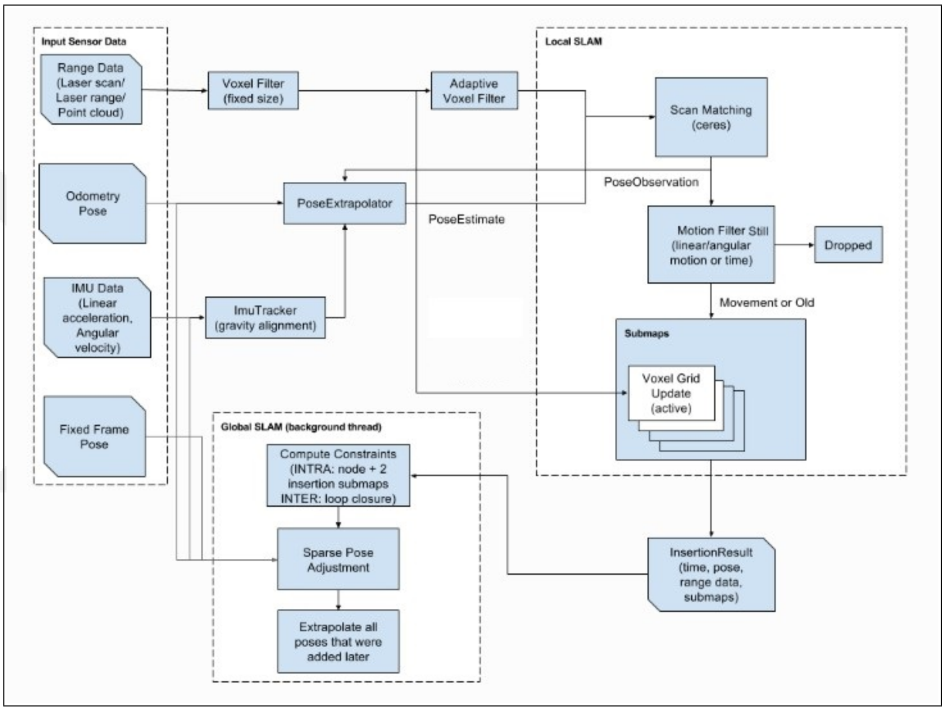

Input Sensor Data
Range Data (Laser scan/ Laser range/ Point cloud)
Odometry Pose
IMU Data (Linear acceleration, Angular velocity)
Fixed Frame Pose
Voxel Filter (fixed size)
Adaptive Voxel Filter
PoseExtrapolator
PoseEstimate
PoseObservation
ImuTracker (gravity alignment)
Local SLAM
Scan Matching (ceres)
Motion Filter Still (linear/angular motion or time)
Dropped
Movement or Old
Submaps
Voxel Grid Update (active)
Global SLAM (background thread)
Compute Constraints (INTRA: node + 2 insertion submaps INTER: loop closure)
Sparse Pose Adjustment
Extrapolate all poses that were added later
InsertionResult (time, pose, range data, submaps)

Figure 3.12. High level system overview of Google Cartographer

3.3.3. Human Detecting and Human Tracking

Systems need to detect and track the identified person continuously. Therefore systems always make detecting and tracking. There are various detection and tracking algorithms and models, according to the our systems requirements and capacities optimized methods are selected.

Firstly, system must be detects the person who will be navigator of the robot. Human recognition algorithm is required for this purpose. Open-ReID is selected as the detector of this project. Open Re-ID is a lightweight library of person re-identification for research purpose. It aims to provide a uniform interface for different datasets, a full set of models and evaluation metrics, as well as examples to reproduce (near) state-of-the-art results. Open-ReID is mainly based on PyTorch. Open-ReID is structed in three parts as in shown in Figure 3.13.

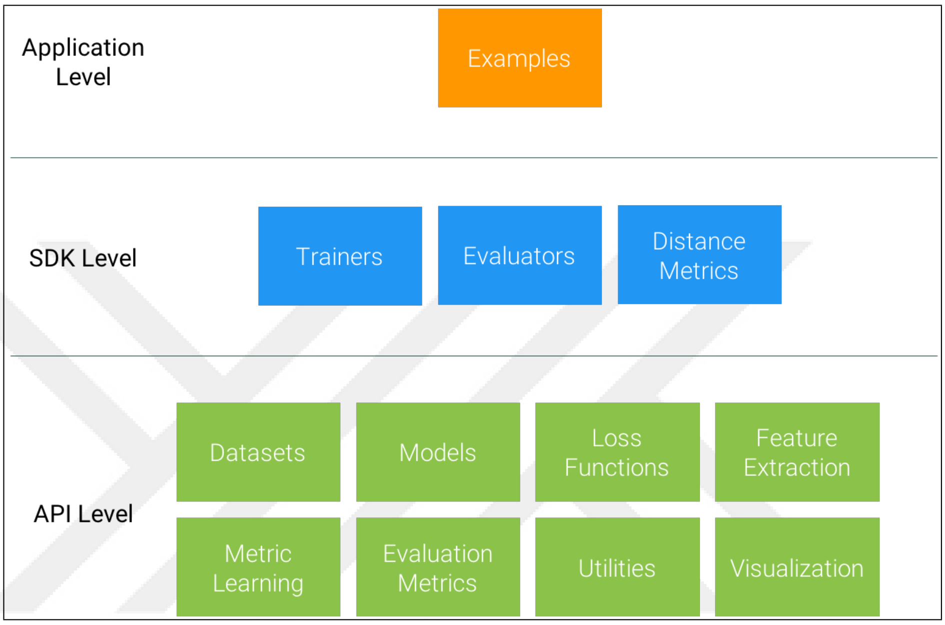


Figure 3.13. Structures of the Open-ReID library

These structures allow all level of work by this library. API, SDK and Application levels can be run for various project and solution types. Moreover, this library could be run as multi data parallel training on multiple GPUs. By default, the program will use all the GPUs also with the GPU of the Nvidia Jetson TX2 board. Person who operates the robot meet the algorithm and Open-ReID detects and define the specific person by getting features.

After this process, system always run to find the defined person in the field of view of the robot. Actually, system must be run continuously and follow the person so required to tracking algorithm. This algorithm could be run real time and must be detect frame by frame or adopted with the working frequency of the system. Complexity should be lower for a based embedded system robot. Recent years Discriminative Correlation Filter is used for these purpose works but it has not enough light mathematical and algorithmic load for the systems. ECO Tracker which is developed for the real time high accuracy required system which is stand for Efficient Convolution Operators for trackers.

3.3.4. Visual Based Motors Control

In this project, the AMR have four motor four tires and differences control cards. They are feed every 100ms period and refresh the route and velocity. The visual based system give the required information to the motor control unit. This process is continued to provide robot movement correctly and continuously. The data comes from the detection and tracking algorithms are translated to the motor control unit parameter format which are angle and velocity. This message is the navigation parameter off the robot. This motor controlling message and communication between the computer vision unit and the motor control unit is shown in Figure 3.14.

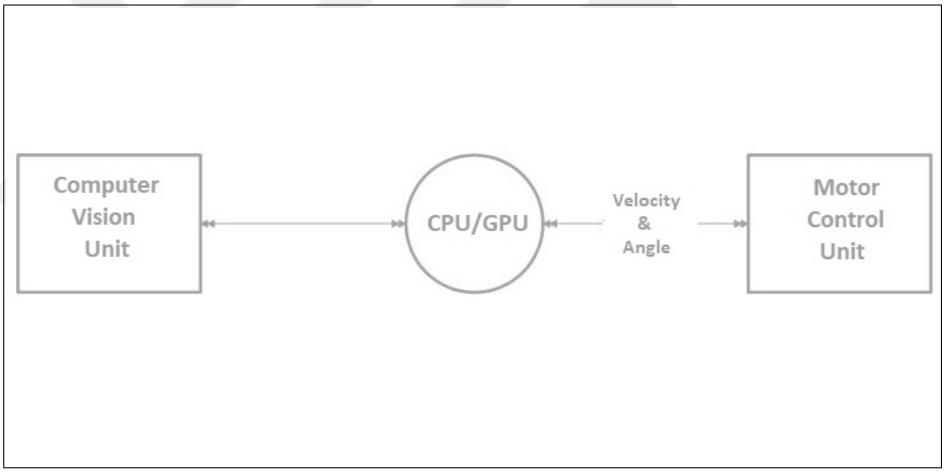


Figure 3.14. Structures of vision unit and motor control unit

Simulation-based studies were carried out in order to enable image processing, autonomous driving, mapping and advanced positioning as in the shown in Figure 3.15.

It was decided to use Robot Operating System (ROS) for the card, which will develop autonomous driving and sensor processing algorithms. The advantage of this system is that it allows the development of the algorithm using various analysis and simulation tools in Linux based environments. These algorithms developed in computer environment will be directly transferable to the control card on the physical prototype production.

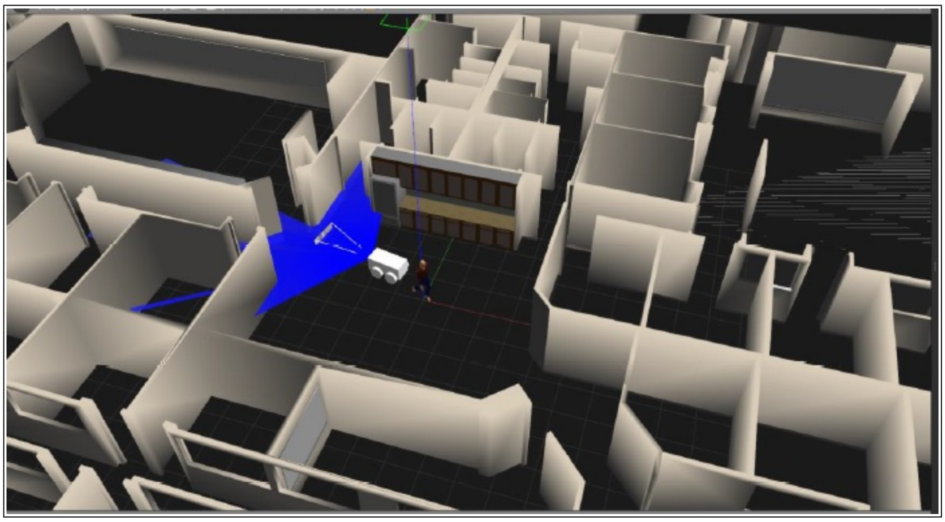


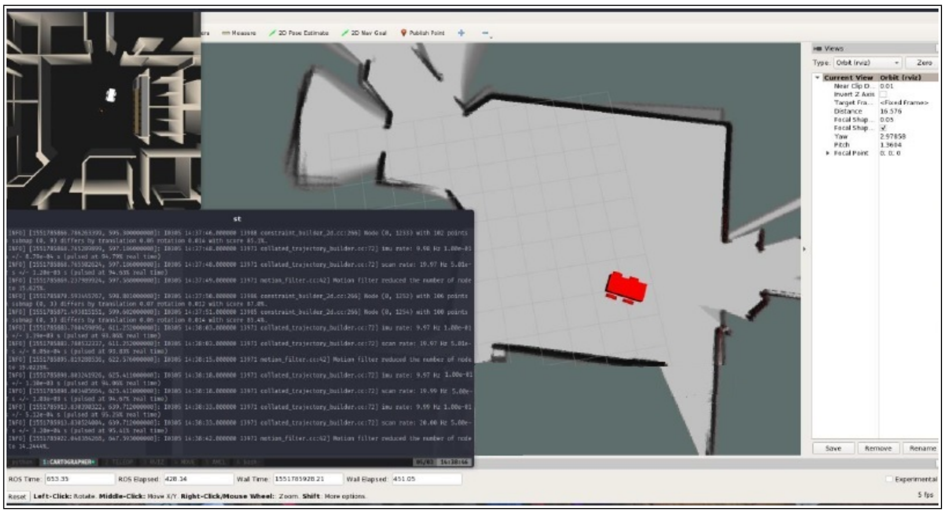
Figure 3.15. Gazebo simulation of navigation

4. MATHEMATICAL METHODS

Our real time computer vision based autonomous mobile robot system has four main implementation and developing steps. These were listed as follows; simultaneous localization and mapping, human detecting and human tracking, autonomous mobile robot navigation and vision based motor control. In order for autonomous mobile robot to continue its movement accurately and safely, these steps must be ensured to be fully and synchronized. Realization of SLAM, it is ensured that it follows the environment and location instantly and clean route alternatives are produced for the robot. In the meantime, the route that the robot will follow according to human movement is determined with human detection and human tracking algorithms. At the same time, this image processing-based output must be transmitted to the motor driver boards in a correct message packet as could be meant by the motor driver unit. By this section, the mathematical model behind the steps used and the methods of implementation will be explained.

4.1. SIMULTANEOUS LOCALIZATION AND MAPPING

Autonomous mobile robots AGVs need to navigate and move in unknown area they need localization and mapping also need to detect obstacle detection. We selected the Google Cartographer Library for this SLAM enviroment. Gazebo software was used to test and simulate the algorithms that will be developed by using ROS. A design was created to represent the vehicle with its size and weight in the Gazebo software. Modeling of the sensors to be used after the simple case and force generating wheels of the vehicle were put into the simulation. A simulation model was created to represent this sensor in the Gazebo environment by considering the values, such as resolution, angle of view and noise in the information booklet of the LIDAR sensor to be used. This model was installed in the vehicle. Similarly, a camera sensor with similar features was placed in the simulation for the camera to be positioned on the vehicle. A human model was also added to the simulation environment. A camera was placed to provide a top view in the simulation environment. The data of this camera is now displayed in a mini window. By using this window, the human in the simulation is allowed to go to the specified point which is shown in Figure 4.1.



Type: Orbit (rviz)

Current View	Orbit (rviz)
Near Clip D...	0.01
Invert Z Axis	□
Target Fra...	<Fixed Frame>
Distance	16.576
Focal Shap...	0.05
Focal Shap...	☑
Yaw	2.97858
Pitch	1.3604
Focal Point	0; 0; 0

```
INFO] [1551785866.786263399, 595.300000000]: I0305 14:37:46.000000 13988 constraint_builder_2d.cc:266] Node (0, 1233) with 102 points
 submap (0, 3) differs by translation 0.06 rotation 0.014 with score 85.3%.
INFO] [1551785868.765289099, 597.186000000]: I0305 14:37:48.000000 13971 collated_trajectory_builder.cc:72] imu rate: 9.98 Hz 1.00e-01
 +/- 8.70e-04 s (pulsed at 94.79% real time)
INFO] [1551785868.765582624, 597.186000000]: I0305 14:37:48.000000 13971 collated_trajectory_builder.cc:72] scan rate: 19.97 Hz 5.01e-
 s +/- 1.20e-03 s (pulsed at 94.63% real time)
INFO] [1551785869.237989924, 597.580000000]: I0305 14:37:49.000000 13971 motion_filter.cc:42] Motion filter reduced the number of node
 to 15.625%.
INFO] [1551785870.593445707, 598.801000000]: I0305 14:37:50.000000 13986 constraint_builder_2d.cc:266] Node (0, 1252) with 106 points
 submap (0, 3) differs by translation 0.07 rotation 0.012 with score 87.0%.
INFO] [1551785871.493815151, 599.602000000]: I0305 14:37:51.000000 13985 constraint_builder_2d.cc:266] Node (0, 1254) with 100 points
 submap (0, 3) differs by translation 0.05 rotation 0.014 with score 85.4%.
INFO] [1551785883.788459096, 611.252000000]: I0305 14:38:03.000000 13971 collated_trajectory_builder.cc:72] imu rate: 9.97 Hz 1.00e-01
 +/- 1.39e-03 s (pulsed at 93.06% real time)
INFO] [1551785883.788532237, 611.252000000]: I0305 14:38:03.000000 13971 collated_trajectory_builder.cc:72] scan rate: 19.97 Hz 5.01e-
 s +/- 8.05e-04 s (pulsed at 93.83% real time)
INFO] [1551785895.819288536, 622.576000000]: I0305 14:38:15.000000 13971 motion_filter.cc:42] Motion filter reduced the number of node
 to 15.0235%.
INFO] [1551785898.003241926, 625.411000000]: I0305 14:38:18.000000 13971 collated_trajectory_builder.cc:72] imu rate: 9.97 Hz 1.00e-01
 +/- 1.30e-03 s (pulsed at 94.06% real time)
INFO] [1551785898.003485004, 625.411000000]: I0305 14:38:18.000000 13971 collated_trajectory_builder.cc:72] scan rate: 19.99 Hz 5.00e-
 s +/- 1.03e-03 s (pulsed at 94.07% real time)
INFO] [1551785913.838398322, 639.712000000]: I0305 14:38:33.000000 13971 collated_trajectory_builder.cc:72] imu rate: 9.99 Hz 1.00e-01
 +/- 5.12e-04 s (pulsed at 95.25% real time)
INFO] [1551785913.838524004, 639.712000000]: I0305 14:38:33.000000 13971 collated_trajectory_builder.cc:72] scan rate: 20.00 Hz 5.00e-
 s +/- 3.20e-04 s (pulsed at 95.41% real time)
INFO] [1551785922.048384288, 647.593000000]: I0305 14:38:42.000000 13971 motion_filter.cc:42] Motion filter reduced the number of node
 to 14.2444%.
```

1:CARTOGRAPHER*

ROS Time: 653.35 ROS Elapsed: 428.34 Wall Time: 1551785928.21 Wall Elapsed: 451.05

Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options.

Figure 4.1. Navigation test on Gazebo simulation

Sensor fusion and assistance of the sensors to improve SLAM accuracy more robustly system. As in this proposed method for this autonomous mobile robot the Simultaneous Localization and Mapping (SLAM) algorithm was mapped using LIDAR and camera sensor information placed on the vehicle. Google Cartographer library is allowed to get localization and mapping by the data that get from 2D cameras. After the enviroment of the robot is defined the slam algorithm is using the LIDAR data and processed to detect the 3D location of the element of the enviroment which are detected in the 3D space. These matching are realized the 3D mapping for the autonomous mobile robot. The vision system of this autonomous mobile robot detects the points in the enviroment through both camera and LIDAR. In z space there is a third space occurs when these tow sensor data points are matched. That is the combination of the cameras feature and LIDAR detected as obstacle and measured points. Because of the fuse of points and feature that comes from the both LIDAR and camera, their error outputs are also be merged. These method is usually use for the stereo image cameras or system. These errors are usually represented as re-projection error which is shown in Figure 4.2.

At the mono cameras are represented as the consecutive two frame. In a real world this distance between the two pints have some error and it is called as re-projection error. A transformation and rotation matrices are calculated by the 3D location of the points in both frames. According to the camera parameters this difference could be calculated as the error

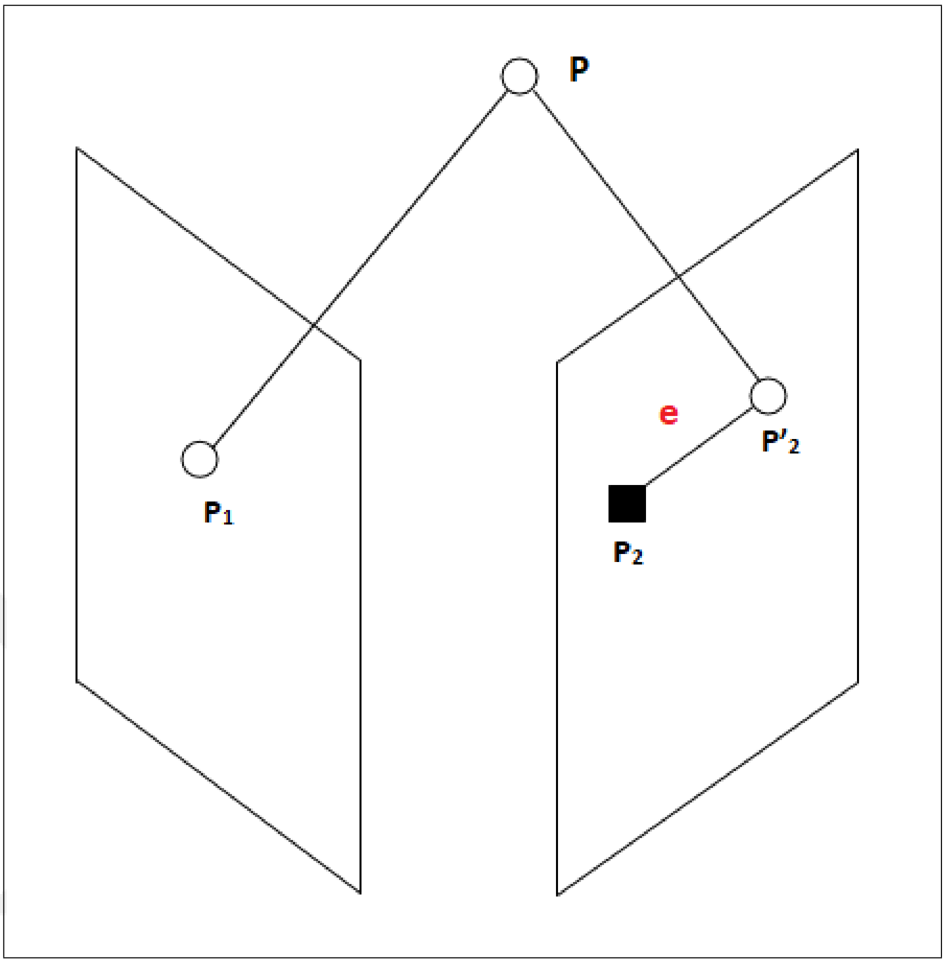


Figure 4.2. Re-projection error estimate

of the re-projection. These calculation is done for all matched points and the sum of them to reach total error. Before the cameras parameter using it is calibrated to get more true feature extraction and imply distance on this output.

Actually as in the visual SLAM error, also LIDAR SLAM error occurs according to the consequence when the feature points matches. These fusion of the sensor based feature points matches, the system could be get the SLAM points to get the localization and mapping elements. These multi way data getting, meaning and the matches of them is shown in Figure 4.3.

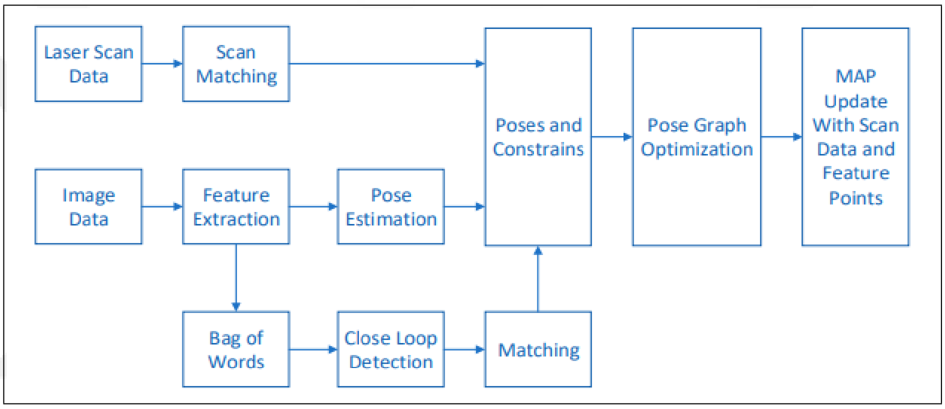


Figure 4.3. Fusion of camera and LIDAR data for SLAM

Google Cartographer detects and create the SLAM parameter by this way. As in our method camera data provide to get feature points to define enviroment and LIDAR data is measure the all environmental obstacle and the detected wall distance in the depth direction. These information are combined and matched to get 3D space and to exhibit on truly 3D map by using the Point Cloud Library (PCL). This architecture and mathematical background of the system is mentioned below according to the this article [9].

Submap development is the iterative process of continuously aligning scanning and submap coordinate sets, also known as frames. H = $\{h_k\}_{k=1,...,K}$, $h_k \in R^2$ with the origin of the scan at $0 \in R^2$ was written as the scanning points details in Equation (4.1). The position ξ of the scanning frame in the submap frame is defined as the transition T_ξ, that stiffly converts scanning points from the scanning frame into the submap frame Equation (4.2).

$$T_\xi p = \underbrace{\begin{pmatrix} \cos\xi_\theta & -\sin\xi_\theta \\ \sin\xi_\theta & \cos\xi_\theta \end{pmatrix}}_{R_\xi} p + \underbrace{\begin{pmatrix} \xi_x \\ \xi_y \end{pmatrix}}_{t_\xi} . \tag{4.1}$$

$$\text{odds}(p) = \frac{p}{1-p,} \tag{4.2}$$

A number of matrix points for hits and a disjoint set for fails are determined when a sample is to be put into the chance grid. The closest grid point are unsorted into the target range for each attack. The grid point is attached associated with each pixel for each miss that intersects one of the rays between the origin of the scan and each scan stage, excluding grid points that are already in the hit range.

A chance p_{hit} or p_{miss} is allocated to each previously unobserved grid point if it is in one of these sets. Whether grid point x has been reached already, the probability of hits and misses has to be changed and equivalently as misses.

$$M_{\text{new}}(x) = \text{clamp}(\text{odds}^{-1}(\text{odds}(M_{\text{old}}(x)) \cdot \text{odds}(p_{\text{hit}}))) \tag{4.3}$$

$$\underset{\Xi^m,\Xi^s}{\text{argmin}} \frac{1}{2} \sum_{ij} \rho(E^2(\xi_i^m, \xi_j^s, \sum_{ij}, \xi_{ij})) \tag{4.4}$$

$$E^2(\xi_i^m, \xi_j^s, \sum_{ij}, \xi_{ij}) = e(\xi_i^m, \xi_j^s, \xi_{ij})^T \sum_{ij}^{-1} e(\xi_i^m, \xi_j^s, \xi_{ij}) \tag{4.5}$$

The residual E is determined for such a limit as in Equation (4.6) and Equation (4.7):

$$e(\xi_i^m, \xi_j^s, \xi_{ij}) = \xi_{ij} - \begin{pmatrix} R^{-1}_{\xi_i^m}(t_{\xi_i^m} - t_{\xi_j^s}) \\ \xi^m_{i;\theta} - \xi^s_{j;\theta} \end{pmatrix} \tag{4.6}$$

$$d_{\text{max}} = \underset{k=1,\ldots,K}{\text{max}} \parallel h_k \parallel \tag{4.7}$$

Through carefully selecting stage measurements, performance is improved. They select the δ_θ angular step size so that the scanning points at the maximum d_{max} range do not move more than r, the width of one pixel. They derive from the concept of cosines,

$$\delta_\theta = \arccos(1 - \frac{r^2}{2d^2_{max}}) \tag{4.8}$$

$$\omega_x = \lceil \frac{W_x}{r} \rceil, \omega_y = \lceil \frac{W_y}{r} \rceil, \omega_\theta = \lceil \frac{W_\theta}{\delta_\theta} \rceil \tag{4.9}$$

An integral number of steps are measured that span the linear and angular search window sizes, e.g., $W_x = W_y = 7m$ and $W_\theta = 30^o$

$$\overline{W} = \{-w_x, ..., w_x\} \times \{-w_y, ..., w_y\} \times \{-w_\theta, ..., w_\theta\} \tag{4.10}$$

It results in a finite range of W creating a search window around an approximation ξ_0 in the middle of it,

$$W = \{\xi_0 + (rj_x, rj_y, \delta_\theta j_\theta) : (j_x, j_y, j_\theta) \in \overline{W}\} \tag{4.11}$$

$$\overline{\overline{W}}_c = (\{(j_x, j_y) \in Z^2 : \left. \begin{array}{c} c_x \le j_x < c_x + 2^{c_h} \\ c_y \le j_y < c_y + 2^{c_h} \end{array} \right\} \times \{c_\theta\}) \tag{4.12}$$

Branching rule: A tuple of integer $c = (c_x, c_y, c_\theta, c_h) \in Z^4$ defines increasing node in the chain. Height c_h nodes merge translations of up to $2^{c_h} \times 2^{c_h}$, but reflect a different rotation:

$$\overline{W}_c = \overline{\overline{W}}_c \cap \overline{W} \tag{4.13}$$

$$\overline{W}_{0,x} = \{-\omega_x + 2^{h_0} j_x : j_x \in Z, 0 \le 2^{h_0} j_x \le 2\omega_x\} \tag{4.14}$$

$$\overline{W}_{0,y} = \{-\omega_y + 2^{h_0} j_y : j_y \in Z, 0 \le 2^{h_0} j_y \le 2\omega_y\} \tag{4.15}$$

$$\overline{W}_{0,\theta} = \{j_\theta \in Z, -\omega_\theta \le j_\theta \le \omega_\theta\} \tag{4.16}$$

$$C_0 = \overline{W}_{0,x} \times \overline{W}_{0,y} \times \overline{W}_{0,\theta} \times \{h_0\} \tag{4.17}$$

Leaf nodes are $c_h = 0$ in height and suit feasible solutions $W \ni \xi_c = \xi_0 + (rc_x, rc_y, \delta_\theta c_\theta)$.

The root node, containing all possible alternatives, does not exist directly and branches to a group of initial nodes C_0 at a defined height h_0 that fills the quest range.

$$C_c = \Big(\big(\{c_x, c_x + 2^{c_h - 1}\} \times \{c_y, c_y + 2^{c_h - 1}\} \times c_\theta \big) \cap \overline{W} \Big) \times \{c_h - 1\} \tag{4.18}$$

Link into up to four height children $c_h - 1$ at a given node c with $c_h > 1$,

$$\begin{aligned} score(c) &= \sum_{k=1}^{K} \max_{j\in\overline{\overline{W}}_c} M_{\text{nearest}}(T_{\xi_j} h_k) \\ &\geq \sum_{k=1}^{K} \max_{j\in\overline{W}_c} M_{\text{nearest}}(T_{\xi_j} h_k) \\ &\geq \max_{j\in\overline{W}_c} \sum_{k=1}^{K} M_{\text{nearest}}(T_{\xi_j} h_k) \end{aligned} \tag{4.19}$$

Computing upper limits: The majority of the branch and bound methodology is an effective way to calculate upper bounds at internal nodes, both in terms of numerical energy and bound consistency. We use;

$$score(c) = \sum_{k=1}^{K} M_{\text{precomp}}^{c_h}(T_{\xi_c} h_k) \tag{4.20}$$

Pre-computed grids are used $M_{\text{precomp}}^{c_h}$ to be able to calculate the actual effectively. Calculating one grid per potential height c_h helps us to measure the score in the amount of scan points with a linear effort. Remember that we also determine the limits over $\overline{\overline{W}}_c$ that can be greater than $\overline{W}_c$ near the limit of our search space to be able to do this.

$$M_{\text{precomp}}^{h}(x, y) = \max_{\substack{x'\in[x,x+r(2^h-1)] \\ y'\in[y,y+r(2^h-1)]}} M_{\text{nearest}}(x', y') \tag{4.21}$$

with ξ_c as before for the leaf nodes. Notice that M_{precomp}^{h} has the same pixel arrangement as $M_{nearest}$, but the maximal values of the $2^h \times 2^h$ pixel box starting there are contained in each row.

4.2. HUMAN DETECTING AND HUMAN TRACKING

This thesis is about the autonomous mobile robot which is follow the identified person to navigate the robot. For this aim the brain of the robot firstly detect the human and then by taking the needed feature and match with the trained dataset that identify the detected person as the special one who know as navigator. Detecting and tracking are the main parts of this aim, moreover these computations should be correctly and real time working as least delay and failed.

Once the person being monitored is not detected continuously, once detected. Because the detection algorithm will be followed by The more important the criterion is, the more accurately it works. A value The criterion is how easily the Jetson TX2 platform is implemented on the GPU. That's why, it's right, fast enough, and SSD algorithm and inception supported by the TensorRT library model selected which is shown in Figure 4.4.

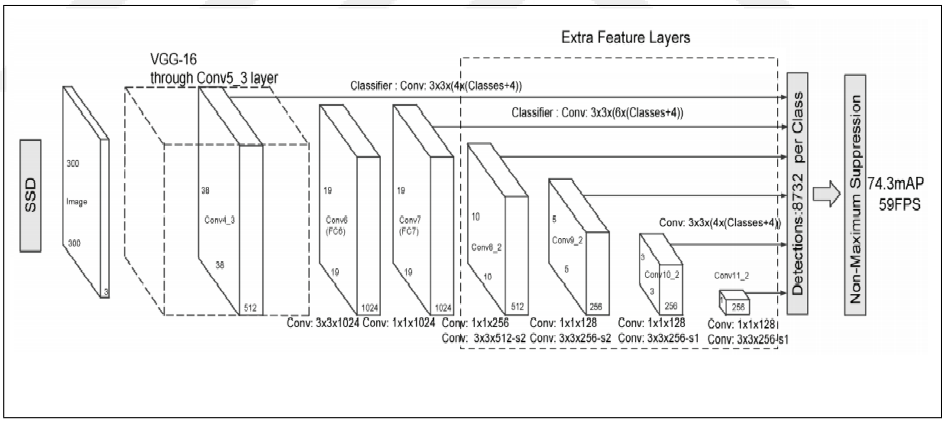


Figure 4.4. SSD detector principle

This detector algorithm is, in one step, a neural network that recognizes objects in the image. There are also different model variants such as RCNN. RCNN works differently. First find the regions that can be objects in the picture and then work on these regions. In other words, the location of the objects is calculated in two steps. It works more slowly and gives more accurate results. The reason it gives more accurate results is the background in an image rather than an object. This is much simpler for the neural network to accurately detect these regions as negative than to detect objects positively. Since this creates an imbalance, the neural network is trained in a different way. 2-stage detectors reduce this imbalance. Recently, faster and more accurate models have been proposed due to both single stage. One of them is RetinaNet. The RetinaNet focuses more on difficult examples by ”punishing” easily guessed samples. In this way, it prevents slipping to the easy samples. SSD algorithm is sufficient for our application, this algorithm is selected. There are too many algorithms to track objects. Mathematical methods include optical flow, kalman filter, boosting, mile, kcf, tld, medianflow, mosse and csrt. At the beginning of the project we used a tracker algorithm called ECO tracker. This algorithm showed very good results at the IEEE Computer Vision and Pattern Recognition conference. Despite the good results we obtained from this tracking algorithm, it was not found enough so we had to switch to pysot trackera. The most important reason for this tracker. This is because it can run on the GPU. Since this algorithm will work continuously and follow the person, it is very important to have an algorithm that is implemented in such a way that it can work in GPU as fast as the detector works. Since the vehicle recognizes a particular person, if the person followed is lost, it must be recognized and resumed the next time it is returned to the screen. Because detection recognizes all people on the screen, the tracker follows the area assigned to it. Detected people must be filtered and given to the tracker. In Computer Vision, this is called re identification, in short, Reid. This issue although there is too much research on the literature, it has not been as successful as detection and tracking. This problem can be solved in general by identifying people in a dataset, but also decoding a person and then comparing it with this feature vector. Since our aim is the latter, a neural network should be selected accordingly. The loss of the simply created neural network is low, the difference between the feature vector of two difference between feature vector is highly optimized. Two such neural networks have been tested. Both give similar results, but the last selected model is selected because it can work in the GPU.

The model we started to use separates the human into six regions and takes out the features for these six regions. The name of the model is PCB stands for Part-based Convolutional Baseline and is shown in the Figure 4.5. Structure can be seen. Of course, this will not work in our case and it would not make sense to train an artificial neural network for every human being, but there is no need to do such a thing because of the structure of the model. By removing this last layer, we achieved successful results using the 2048x6 feature vector calculated in the previous layer.

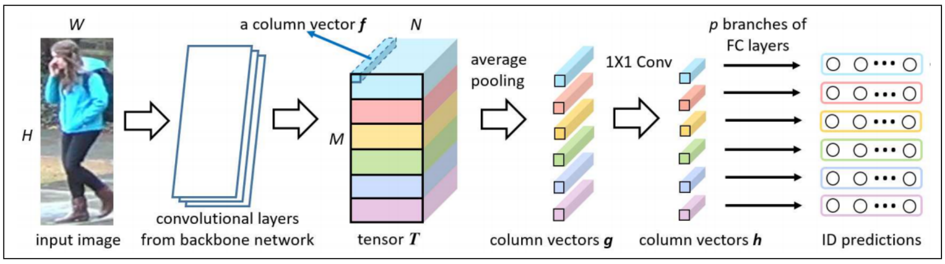


Figure 4.5. Part-based convolutional baseline

This mathematical models of the human detection and continuously tracking method of re-identification is mentioned below as the references of the articles [10,11].

$$\underset{f_1,f_2,...f_D,g}{\operatorname{argmin}} \sum_{i=1}^{D}\sum_{j=1}^{N_i} L\Big(f_i(g(x_i^{(j)}), y_i^{(j)}\Big) \tag{4.22}$$

A straightforward solution when combining all the D domains together is to use a multi-task objective method, i.e. to use D softmax classifiers $f_1, f_2, ..., f_D$ and a common extractor g features that reduce the number of D domains.

where L is the softmax loss function corresponding to the crossroads between the expected vector of likelihood and the ground reality.

$$\underset{f,g}{\operatorname{argmin}} \sum_{i=1}^{D}\sum_{j=1}^{N_i} L\Big((f \circ g)(x_i^{(j)}), y\prime_i^{(j)}\Big) \tag{4.23}$$

A single-task learning paradigm allows the network to identify users from all contexts at the same time compared to the multi-task formulation. Field prejudices (e.g., context noise, illumination, etc.) as well as personal appearance and characteristics are depicted in the function depictions. If the data distributions of two domains vary a lot, through analyzing only the domain preferences, it would be easy to separate the individuals of the two domains. Nevertheless, if these assumptions are not overly important, the network will need to learn to make judgments regarding discriminate person-related features. Thus the single-task objective fits better to our setting and is chosen for this work.

$$s_i = L(g(x)_{\setminus i}) - L(g(x)) \tag{4.24}$$

Because of the pre-trained CNN model using the mixed dataset, we classify the neurons are successful for each domain. We describe the effect of a single neuron on this dataset for each domain sample as the benefit of the loss function when we extract the neuron. In specific, let $g(x) \in Rd$ denote an image x's d-dimensional CNN function vector. The impact score of the i-th $(i \in \{1, 2, ..., d\})$ neuron on this image sample is defined as in Equation (4.24).

Where $g(x)_{\setminus i}$ is the vector function after the i-th answer is set to zero. For each domain D, we then take s_i's assumption of achieving the average effect value $\bar{s}_i = E_{x \in D}[s_i]$ over all its samples.

$$s_i \approx -\frac{\partial L}{\partial g(x)_i} g(x)_i + \frac{1}{2} \frac{\partial^2 L}{\partial g(x)_i^2} g(x)_i^2 \tag{4.25}$$

We empirically research the accuracy of this approximation and find that for higher-level layers near to the feature of failure, it is more precise.

$$m_i = \begin{cases} 1 & \text{if } s_i > 0 \\ 0 & \text{if } s_i \leq 0 \end{cases} \tag{4.26}$$

We continue to train the CNN model after collecting all the $\bar{s}_i$, but with these effect scores as feedback for lowering various neurons for different domains during the training process. We create a binary mask m for the neurons according to their impact scores s for all the samples belonging to a particular domain, and then element wisely multiply m with the neuron responses. Two methods on how to generate the m mask are proposed.

The first is deterministic, which discards all non-positive impact ratings of neurons:

$$p(m_i = 1) = \frac{1}{1 + e^{-s_i/T}} \tag{4.27}$$

The other is stochastic, where m_i is likely to be taken from a Bernoulli process.

Here we use the sigmoid feature to map an effect score to $(0, 1)$, and T is the temperature that regulates how the s scores will significantly affect the probabilities. When $T \longrightarrow 0$, it is equivalent to the deterministic scheme; when $T \longrightarrow \infty$, it falls back to the standard dropout with a ratio of 0.5.

We apply the Domain Guided Dropout to the fc7 neurons and resume the training process. The networks learning rate policy is changed to decay polynomial from 0.01 with the power parameter set to 0.5. The whole network is trained for 10 more epochs.

The neurons will also be eliminated during the evaluation stage for the deterministic system if their effect is no greater than zero. Although we hold all the neuron responses for the stochastic process, we scale the i-th one with $1/(1 + e^{-s_i/T})$.

4.3. AUTONOMOUS MOBILE ROBOT NAVIGATION

The navigation could be described as the process accomplished by a mobile robotic platform, for determining the reasonable and safe path between a starting and a target point.

Robots have been navigating using vision since the days of the Stanford Cart [13]. Mostly autonomous robots use sensors and embedded systems to reach successful result and product. Sensors are commonly LIDAR, SONAR, UV or range finders as non cameras based vision sensor. Moreover type of the cameras are the most using sensor for these purpose to identify and detect the enviroment and objects. In its simplest form, obstacle detection is the process of distinguishing an obstacle from the floor. It is not necessary to understand what is seen in order to avoid obstacles - simply distinguishing between the floor

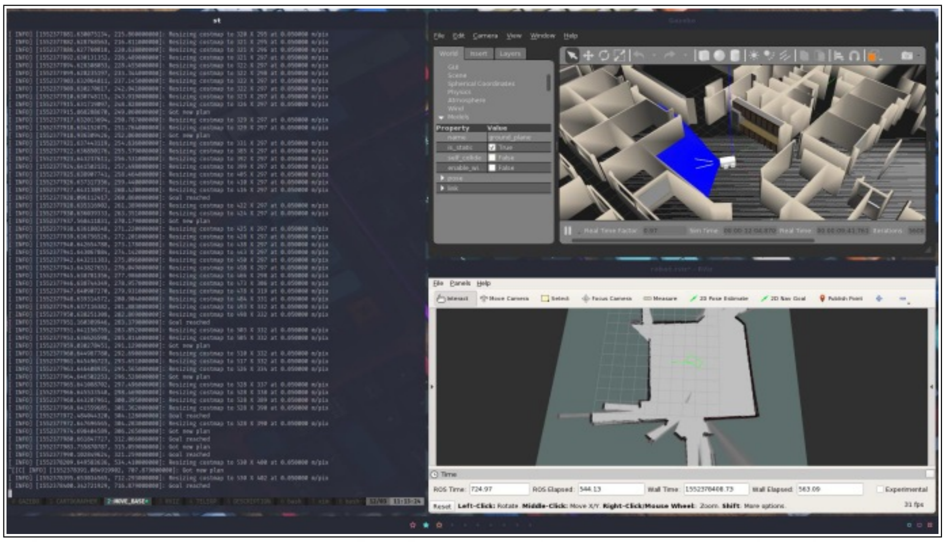


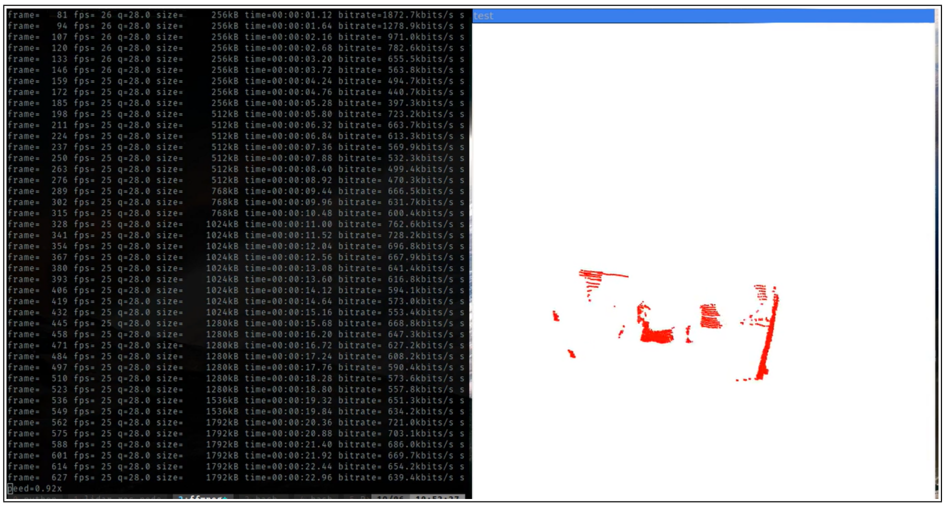
Figure 4.6. Mathematical model structure

and all other objects (even moving humans) is sufficient [12] shown in Figure 4.6.

Dynamic Window Approach (DWA) method was used for the driving algorithm. In this method, the approach used to determine the route to be used to reach the specified destination is as follows; discrete sample of the vehicle's control plane, for each sampled speed, simulate the current state of the vehicle and determine if the sampled speed is applied for a short time score all possible results and rank them. When creating the score ruler, take measures such as proximity to target, proximity to other objects, global course and proximity to speed. Ignore invalid routes (intersecting another object), send the speed vector to the vehicle that will generate the highest score, the vision of the robot is shown in Figure 4.7.

Repeated repetition of the DWA algorithm makes it possible to move the vehicle to the optimum route. The following illustrations show the location destination indicated by the green arrow on the map and the vehicle following this destination.

In order to develop the autonomous driving algorithms of the vehicle as close as possible to the real environment, a tank was created in Gazebo environment. The location of the vehicle in this warehouse environment was made possible by the use of algorithms developed to follow a known person.



```
frame=   81 fps= 26 q=28.0 size=     256kB time=00:00:01.12 bitrate=1872.7kbits/s s
frame=   94 fps= 26 q=28.0 size=     256kB time=00:00:01.64 bitrate=1278.9kbits/s s
frame=  107 fps= 26 q=28.0 size=     256kB time=00:00:02.16 bitrate= 971.0kbits/s s
frame=  120 fps= 26 q=28.0 size=     256kB time=00:00:02.68 bitrate= 782.6kbits/s s
frame=  133 fps= 26 q=28.0 size=     256kB time=00:00:03.20 bitrate= 655.5kbits/s s
frame=  146 fps= 26 q=28.0 size=     256kB time=00:00:03.72 bitrate= 563.8kbits/s s
frame=  159 fps= 25 q=28.0 size=     256kB time=00:00:04.24 bitrate= 494.7kbits/s s
frame=  172 fps= 25 q=28.0 size=     256kB time=00:00:04.76 bitrate= 440.7kbits/s s
frame=  185 fps= 25 q=28.0 size=     256kB time=00:00:05.28 bitrate= 397.3kbits/s s
frame=  198 fps= 25 q=28.0 size=     512kB time=00:00:05.80 bitrate= 723.2kbits/s s
frame=  211 fps= 25 q=28.0 size=     512kB time=00:00:06.32 bitrate= 663.7kbits/s s
frame=  224 fps= 25 q=28.0 size=     512kB time=00:00:06.84 bitrate= 613.3kbits/s s
frame=  237 fps= 25 q=28.0 size=     512kB time=00:00:07.36 bitrate= 569.9kbits/s s
frame=  250 fps= 25 q=28.0 size=     512kB time=00:00:07.88 bitrate= 532.3kbits/s s
frame=  263 fps= 25 q=28.0 size=     512kB time=00:00:08.40 bitrate= 499.4kbits/s s
frame=  276 fps= 25 q=28.0 size=     512kB time=00:00:08.92 bitrate= 470.3kbits/s s
frame=  289 fps= 25 q=28.0 size=     768kB time=00:00:09.44 bitrate= 666.5kbits/s s
frame=  302 fps= 25 q=28.0 size=     768kB time=00:00:09.96 bitrate= 631.7kbits/s s
frame=  315 fps= 25 q=28.0 size=     768kB time=00:00:10.48 bitrate= 600.4kbits/s s
frame=  328 fps= 25 q=28.0 size=    1024kB time=00:00:11.00 bitrate= 762.6kbits/s s
frame=  341 fps= 25 q=28.0 size=    1024kB time=00:00:11.52 bitrate= 728.2kbits/s s
frame=  354 fps= 25 q=28.0 size=    1024kB time=00:00:12.04 bitrate= 696.8kbits/s s
frame=  367 fps= 25 q=28.0 size=    1024kB time=00:00:12.56 bitrate= 667.9kbits/s s
frame=  380 fps= 25 q=28.0 size=    1024kB time=00:00:13.08 bitrate= 641.4kbits/s s
frame=  393 fps= 25 q=28.0 size=    1024kB time=00:00:13.60 bitrate= 616.8kbits/s s
frame=  406 fps= 25 q=28.0 size=    1024kB time=00:00:14.12 bitrate= 594.1kbits/s s
frame=  419 fps= 25 q=28.0 size=    1024kB time=00:00:14.64 bitrate= 573.0kbits/s s
frame=  432 fps= 25 q=28.0 size=    1024kB time=00:00:15.16 bitrate= 553.4kbits/s s
frame=  445 fps= 25 q=28.0 size=    1280kB time=00:00:15.68 bitrate= 668.8kbits/s s
frame=  458 fps= 25 q=28.0 size=    1280kB time=00:00:16.20 bitrate= 647.3kbits/s s
frame=  471 fps= 25 q=28.0 size=    1280kB time=00:00:16.72 bitrate= 627.2kbits/s s
frame=  484 fps= 25 q=28.0 size=    1280kB time=00:00:17.24 bitrate= 608.2kbits/s s
frame=  497 fps= 25 q=28.0 size=    1280kB time=00:00:17.76 bitrate= 590.4kbits/s s
frame=  510 fps= 25 q=28.0 size=    1280kB time=00:00:18.28 bitrate= 573.6kbits/s s
frame=  523 fps= 25 q=28.0 size=    1280kB time=00:00:18.80 bitrate= 557.8kbits/s s
frame=  536 fps= 25 q=28.0 size=    1536kB time=00:00:19.32 bitrate= 651.3kbits/s s
frame=  549 fps= 25 q=28.0 size=    1536kB time=00:00:19.84 bitrate= 634.2kbits/s s
frame=  562 fps= 25 q=28.0 size=    1792kB time=00:00:20.36 bitrate= 721.0kbits/s s
frame=  575 fps= 25 q=28.0 size=    1792kB time=00:00:20.88 bitrate= 703.1kbits/s s
frame=  588 fps= 25 q=28.0 size=    1792kB time=00:00:21.40 bitrate= 686.0kbits/s s
frame=  601 fps= 25 q=28.0 size=    1792kB time=00:00:21.92 bitrate= 669.7kbits/s s
frame=  614 fps= 25 q=28.0 size=    1792kB time=00:00:22.44 bitrate= 654.2kbits/s s
frame=  627 fps= 25 q=28.0 size=    1792kB time=00:00:22.96 bitrate= 639.4kbits/s s
peed=0.92x
```

Figure 4.7. 3D vision of the robot

The working principle of the route planning algorithm is important for the smooth and safe movement of the robot. The route that is planned to move is copied to the control space. Simulation is made for each movement and speed sampled. Each model is rated and the highest rated route and speed is selected. This step is repeated each time a motion command is received. These steps are shown in Figure 4.8.

The background mathematical details of the this method is mentioned below as from the references article of the algorithms [12,13].

$$x(t_n) = x(t_0) + \int_{t_0}^{t_n} v(t) \cdot \cos\theta(t)dt \tag{4.28}$$

Let $x(t)$ and $y(t)$ in some global coordinate system denote the coordinate of the robot at the time t and let $\theta(t)$ define the orientation of the robot (heading direction). The triplet $< x, y, \theta >$ identifies the robot's cinematic setup. The operation of a synchro-drive robot is limited in such a way that the translation speed v often contributes to the robot's steering path θ which is a non-holonomic limitation. Let $x(t_0)$ and $x(t_n)$, at the moment (t_0) and (t_n) respectively, represent the robot's x-coordinates. Let $v(t)$ denote the robot's translation speed at time t and $\omega(t)$ its rotational speed. Then it is possible to express $x(t_n)$ and $y(t_n)$

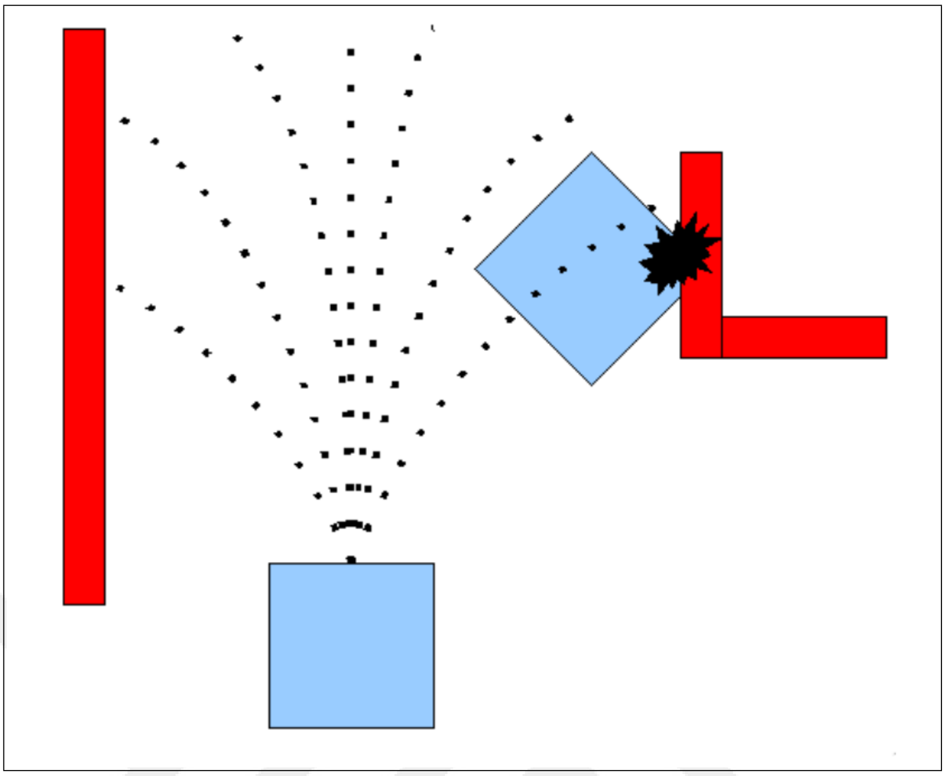


Figure 4.8. Robot navigation and clean road planning simulation

as $x(t_0)$ $\upsilon(t)$ and $\theta(t)$:

$$y(t_n) = y(t_0) + \int_{t_0}^{t_n} \upsilon(t) \cdot \text{sin}\theta(t)dt \tag{4.29}$$

Equation (4.28) and Equation (4.29) depend on the velocities of the robot, which usually cannot be adjust directly.

$$x(t_n) = x(t_0) + \int_{t_0}^{t_n} \left(\upsilon(t_0) + \int_{t_0}^{t} \dot{\upsilon}(\hat{t})d\hat{t} \right) \cdot \cos \left(\theta(t_0) + \int_{t_0}^{t} \left(\omega(t_0) + \int_{t_0}^{\hat{t}} \dot{\omega}(\tilde{t})d\tilde{t} \right) d\tilde{t} \right) dt \tag{4.30}$$

Instead, the velocity $\upsilon(t)$ depends on the initial transnational velocity $\upsilon(t_0)$ at t_0, and the transnational acceleration $\dot{\upsilon}(t)$ in the time interval $t \in [t_0, t]$. Similarly, the $\theta(t)$ orientation is a function of the initial $\theta(t_0)$ orientation, the initial $\omega(t_0)$ velocity at t_0, and the $\dot{\omega}(t)$ rotational acceleration with $t \in [t, t_0]$. Replacing $\upsilon(t)$ and $\theta(t)$ with the corresponding initial kinematic and dynamic configuration $v(t_0), \theta(t_0), \omega(t_0)$, and the accelerations $\dot{v}(t)$, $\dot{\omega}(t)$ gives the expression (note that the $y(t_n)$ derivation is equivalent,

so we only define the x-coordinate derivation Equation (4.30):

$$x(t_n) = x(t_0) + \sum_{i=1}^{n-1} \int_{t_i}^{t_i+1} (v(t_i) + \dot{v}_i \cdot \Delta_t^i) \cdot \cos \left(\theta(t_i) + \omega(t_i) \cdot \Delta_t^i + \frac{1}{2} \dot{\omega}_i \cdot (\Delta_t^i)^2 \right) dt \quad (4.31)$$

We are now going to simplify Equation (4.30) to derive a more functional model by approximating the velocities of the robot with a constant value within the time interval $[t, t_{i+l}]$. The resulting movement equation, Equation (4.31), converges to Equation (4.32) as the time interval length goes to zero. The trajectory of a robot can be approximated by piecewise circular arcs, as we will see under this theory. This representation is well suited to generate real-time motion control, as the discussion of the dynamic window method will be defined in the next section.

If the time intervals $[t_i, t_{i+1}$ are sufficiently small, the term $v(t_i) + \dot{v} \cdot \Delta_t^i$ may be approximated by an arbitrary translation velocity $v_i \in [v(t_i), v(t_{i+1})]$ due to the smoothness of the timing of the robot.

Similarly, the term $\theta(t_i) + \omega(t_i)\Delta_t^i + \frac{1}{2}\dot{w}_i(\Delta_t^i)^2$ may be approximated by an arbitrary velocity of $\theta(t_i) + \omega_i \cdot \Delta_t^i$, where $\omega_i \in [\omega(t_i), \omega(t_{i+1})]$. It results in the following movement Equation (4.32),

$$x(t_n) = x(t_0) + \sum_{i=1}^{n-1} \int_{t_i}^{t_i+1} v_i \cdot \cos \left(\theta(t_i) + \omega_i \cdot (\hat{t} - t_i) \right) d\hat{t} \quad (4.32)$$

which can be reduced by solving the integral in in Equation (4.33):

$$x(t_n) = x(t_0) + \sum_{i=1}^{n-1} \left(F_x^i(t_{i+1}) \right) \quad (4.33)$$

where in Equation (4.34),

$$F_x^i(t) = \begin{cases} \frac{\upsilon_i}{\omega_i}\Bigg(\text{sin}\theta(t_i) - \text{sin}\Big(\theta(t_i) + \omega_i\cdot)(t - t_i)\Big)\Bigg), \omega_i \neq 0 \\ \upsilon_i\text{cos}\Big(\theta(t_i)\Big)\cdot t, \omega_i = 0 \end{cases} \tag{4.34}$$

The related y-coordinate equations are in Equation (4.35) and Equation (4.36):

$$y(t_n) = y(t_0) + \sum_{i=1}^{n-1}\Big(F_y^i(t_{i+1})\Big) \tag{4.35}$$

$$F_y^i(t) = \begin{cases} \frac{\upsilon_i}{\omega_i}\Bigg(\text{cos}\theta(t_i) - \text{cos}\Big(\theta(t_i) + \omega_i\cdot)(t - t_i)\Big)\Bigg), \omega_i \neq 0 \\ \upsilon_i\text{sin}\Big(\theta(t_i)\Big)\cdot t, \omega_i = 0 \end{cases} \tag{4.36}$$

Remember that if $\omega_i = 0$, a straight line is formed by the device.

By comparison, if $\omega_i \neq 0$, the direction of the robot represents a sphere, as can be seen by taking into account as in Equation (4.37) and Equation (4.38),

$$M_x^i = -\frac{\upsilon_i}{\omega_i}\cdot\text{sin}\theta(t_i) \tag{4.37}$$

$$M_y^i = \frac{\upsilon_i}{\omega_i}\cdot\text{cos}\theta(t_i) \tag{4.38}$$

for which the arrangement is as follows in Equation (4.39):

$$(F_x^i - M_x^i)^2 + (F_y^i - M_y^i)^2 = (\frac{\upsilon_i}{\omega_i})^2 \tag{4.39}$$

This indicates that the i-th direction is a M_i loop (M_x^i, M_y^i), with radius $M_r^i = \frac{v_i}{\omega_i}$.

Therefore, we can estimate a robot's trajectory by a series of circular and straight line arcs by using piecewise constant velocities.

$$V_a = \left\{ v, \omega | v \leq \sqrt{2 \cdot \text{dist(v}, \omega) \cdot \dot{v}_b} \wedge \omega \leq \sqrt{2 \cdot \text{dist(v}, \omega) \cdot \dot{\omega}_b} \right\} \tag{4.40}$$

Restrictions on rotational and translational speeds are enforced by barriers in the robot's closer setting. The overall acceptable speed on a curvature, for example, depends on the distance from this curvature to the next barrier.

Assume that the word $dist(v, \omega)$ for a velocity (v, ω) describes the distance to the nearest obstacle at the corresponding curvature (in the next section we explain how to measure this distance given circular trajectories). If the robot may halt before it hits this barrier, a velocity is called admissible. Let $\dot{v}_b$ and $\dot{\omega}_b$ be the break-up accelerations.

The set V_a of permissible speeds is then described as Equation (4.40).

Therefore V_a is the series of speeds (v, ω) that helps the robot to stop without an object colliding.

$$V_d = \left\{ (v, \omega) | v \varepsilon [v_a - \dot{v} \cdot t, v_a + \dot{v} \cdot t] \wedge \omega \varepsilon [\omega_a - \dot{\omega} \cdot t, \omega_a + \dot{\omega} \cdot t] \right\} \tag{4.41}$$

The cumulative quest range is restricted to the elastic frame, which includes only the speeds that can be achieved within the next time period, in order to take into consideration the minimal accelerations performed by the motors. Let t be the time period during which $\dot{v}$ and $\dot{w}$ accelerations are applied and let (v_a, ω_a) be the actual velocity. Then the dynamic window V_d is defined as Equation (4.41).

The complex window is based around the actual speed and its extensions are reliant on the accelerations that can be performed. It is not possible to reach all curvatures outside the complex range within the next time span and is therefore not known to avoid obstacles.

4.4. DECIDING THE DIRECTION OF THE AUTONOMOUS MOBILE ROBOT

Deciding of the simultaneous direction of the robot is related to the human tracker outputs and navigation outputs. This means that robot is needed to the enviroment condition and clean road which is mentioned in the previous section. This command is getting from the navigation system of the robot. This is the safely driving side of the movement of autonomous mobile robot. The other direction decider of this system is the human tracker system. Because of the robot is needed to guide a person who is followed by the autonomous mobile robot. Human tracker algorithm continuously run and track the identified person frame by frame. ECO Tracker which is used in this thesis as mentioned in previous sections that track the identified person and draw a rectangular line of the area that person tracking. These rectangular areas are signed by the frame and the translation of these rectangular area are used for definition of the movement by the differentiation of them. Deciding the direction of the robot which would follow the detected and identified person so we need the tracking algorithms and models to be follow up the right person and decide the direction of the next move of the robot. This models and mathematical background of the system is mentioned below which is references of the related article [10].

$$J_d\{x^d\}(t) = \sum_{n=0}^{N_d-1} x^d[n] b_d \left(t - \frac{T}{N_d} n \right) \tag{4.42}$$

b_d is a $T > 0$ interpolation kernel. The $J_d\{x^d\}$ result is therefore an interpolated layer of attributes, provided as a continuous T-periodic variable. $J\{x\}$ is used to mean the whole function of interpolated diagram, where $J\{x\}(t) \in R^D$ is positioned as in Equation (4.42).

A continuous T-periodic multi-channel convolution filter $f = (f^1 ... f^D)$ is equipped in the

C-COT formulation to forecast the target's $S_f\{x\}(t)$ detection scores as in Equation (4.43),

$$S_f\{x\} = f * J\{x\} = \sum_{d=1}^{D} f^d * J_d\{x^d\} \tag{4.43}$$

The scores are set the region which is in the related image region $t \in [0, T]$ of the $x \in X$ function. The conversion of single-channel T-periodic functions is described in Equation (4.44) as $f * g(t) = \frac{1}{T}\int_0^T f(t-\tau)g(\tau)d\tau$. The multi-channel convolution $f * J\{x\}$ is obtained by summing all channels results as described in Equation (4.45). Filters were learnt by using the minimizing for the following target,

$$E(f) = \sum_{j=1}^{M} \alpha_j \parallel S_f\{x_j\} - y_j \parallel_{L^2}^2 + \sum_{d=1}^{D} \parallel \omega f^d \parallel_{L^2}^2 \tag{4.44}$$

Test x_j's labelled detection scores $y_j(t)$ are set to a replicated Gaussian function on a regular basis. The data term consists of the classification error weighted by the $L^2 - \text{norm}\|g\|_{L^2}^2 = \frac{1}{T}\int_0^T |g(t)|^2 \text{d}t$ where $\alpha_j \geq 0$ is the weight of sample x_j.

Regularization requires a spatial penalty $\omega(t)$ to minimize the disadvantages of the periodic assumption and thus allow spatial spatial assistance Equation (4.46).

$$E(f) = \sum_{j=1}^{M} \alpha_j \parallel \widehat{S_f\{x_j\}} - \hat{y}_j \parallel_{\ell^2}^2 + \sum_{d=1}^{D} \parallel \hat{\omega} * \hat{f}^d \parallel_{\ell^2}^2 \tag{4.45}$$

The hat $\hat{g}$ of the T-periodic equation g corresponds to the coefficients of the Fourier sequence. Fourier coefficients of the detection scores Equation (4.47) are given by the formula $\widehat{S_f\{x\}} = \sum_{d=1}^{D} \hat{f}^d X^d \hat{b}_d$, where X^d is the Discrete Fourier Transform (DFT) of x^d.

$$(A^H \mathit{\Gamma} A + W^H W)\hat{f} = A^H \mathit{\Gamma} \hat{y} \tag{4.46}$$

Below, in $\hat{f}$ and $\hat{y}$, respectively, f^d and y_j are vectorizations of Fourier coefficients. Matrix A has a sparse form, with diagonal blocks comprising $X_j^d[k]\hat{b}_d[k]$ components. In comparison, Γ is a diagonal weight vector of α_j and W is a kernel $\hat{w}[k]$ convolution matrix.

$$S_{Pf}\{x\} = Pf * J\{x\} = \sum_{c,d} p_{d,c} f^c * J_d\{x^d\} = f * P^T J\{x\} \tag{4.47}$$

Then you can compose the new multi-channel filter as the Pf matrix-vector component. We get the operator of factorized convolution Equation (4.48),

$$E(f,P) = \left\| \hat{z}^T P \hat{f} - \hat{y} \right\|_{\ell^2}^2 + \sum_{c=1}^{C} \left\| \hat{\omega} * \hat{f}^c \right\|_{\ell^2}^2 + \lambda \|P\|_F^2 \tag{4.48}$$

Here, as a regularization, we have introduced the Frobenius norm of P, regulated by the weight parameter λ.

$$\hat{z}^T (P_i + \Delta P)(\hat{f}_i + \Delta \hat{f}) \approx \hat{z}^T P_i \hat{f}_{i,\Delta} + \hat{z}^T \Delta P \hat{f}_i = \hat{z}^T P_i \hat{f}_{i,\Delta} + (\hat{f}_i \otimes \hat{z})^T \mathrm{vec}(\Delta P) \tag{4.49}$$

Set $\hat{f}_{i,\Delta} = \hat{f}_i + \Delta \hat{f}$. In the last equality, the Kronecker product $\otimes$ is used to obtain a vectorization of the matrix step ΔP.

$$\tilde{E}(\hat{f}_{i,\Delta}, \Delta P) = \left\| \hat{z}^T P_i \hat{f}_{i,\Delta} + (\hat{f}_i \otimes \hat{z})^T \mathrm{vec}(\Delta P) - \hat{y} \right\|_{\ell^2}^2 + \sum_{c=1}^{C} \left\| \hat{\omega} * \hat{f}_{i,\Delta}^c \right\|_{\ell^2}^2 + \mu \|P_i + \Delta P\|_F^2 \tag{4.50}$$

The Gauss-Newton subproblem at iteration i is derived by substituting the first-order approximation Equation (4.51) into Equation (4.50),

$$E(f) = E\left\{ \|S_f\{x\} - y\|_{L^2}^2 \right\} + \sum_{d=1}^{D} \left\| \omega f^d \right\|_{L^2}^2 \tag{4.51}$$

Our method is based on the sample function maps x's joint probability distribution $p(x, y)$ and the related expected performance scores y. The logical aim, given $p(x, y)$, is to find the filter that minimizes the predicted error of correlation. This is obtained by replacing with in Equation (4.51).

For the mutual sample distribution $p(x, y)$, assumption E is evaluated. Remember that the initial loss is obtained by estimating the distribution of the sample as $p(x, y) = \sum_{j=1}^{M} \alpha_j \delta_{x_j,y_j}(x, y)$, where δ_{x_j,y_j} refers the Dirac impulse at the training sample x_j, y_j.

Then, propose instead to approximate a compact sample distribution model $p(x, y)$ leading to a more accurate approximation of the expected Equation (4.52).

$$\pi_n = \pi_k + \pi_l, \mu_n = \frac{\pi_k \mu_k + \pi_l \mu_l}{\pi_k + \pi_l} \tag{4.52}$$

To update the GMM, we use Declercq and Piater's simplified version of the online algorithm. With a new x_j model, we initialize a new m variable for $\pi_m = \gamma$ and $\mu_m = x_j$ first. If the part number approaches the L limit, we can simplify the GMM. If its weight π_l is below a level, we discard a part. Otherwise, the two closest components k and l are merged into a common feature n.

The required distance comparisons $\|\mu_k - \mu_l\|$ are efficiently computed in the Fourier domain using Parsevals formula.

Ultimately, it approximates the expected loss Equation (4.53) as,

$$E(f) = \sum_{l=1}^{L} \pi_l \|S_f\{\mu_l\} - y_0\|_{L^2}^2 + \sum_{d=1}^{D} \|\omega f^d\|_{L^2}^2 \tag{4.53}$$

It is remembered that in the Gaussian implies μ_l and the previous weights π_l substitute x_j and α_j, respectively. Therefore, the same training technique can be implemented.

According to the frequency of the motor control card is tuned as 10 Hz. That means in every 100 ms this difference between the distance of the rectangular centers is used for the next 100 ms movement of the autonomous mobile robot. This human detector rectangular area and the difference between the consecutive frames are shown in Figure 4.9.

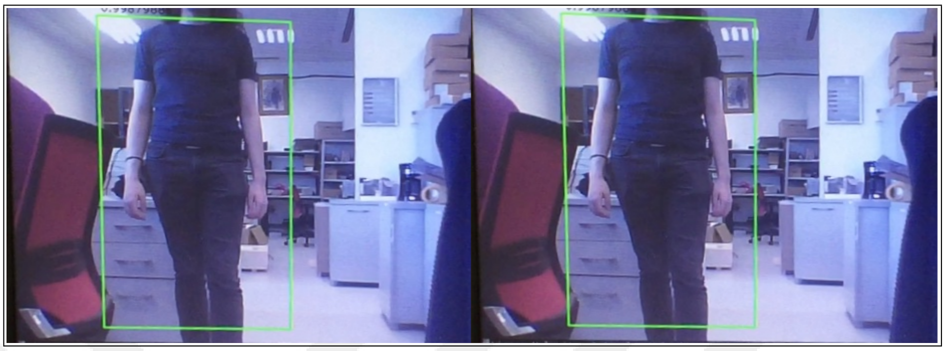


Figure 4.9. Decision of the direction by the human tracker

5. RESULTS AND DISCUSSION

5.1. TEST RESULTS

In this chapter, we will present the test results of our system. Whole tests are made. Firstly we show the results part by part and the end of the test section ended whole AMR system tests are mentioned.

5.1.1. Simultaneous Localization and Mapping Test Results

System needs to localization and mapping to navigate safety and correctly. Firstly, gazebo simulation tests and design are made as shown in Figure 5.1.

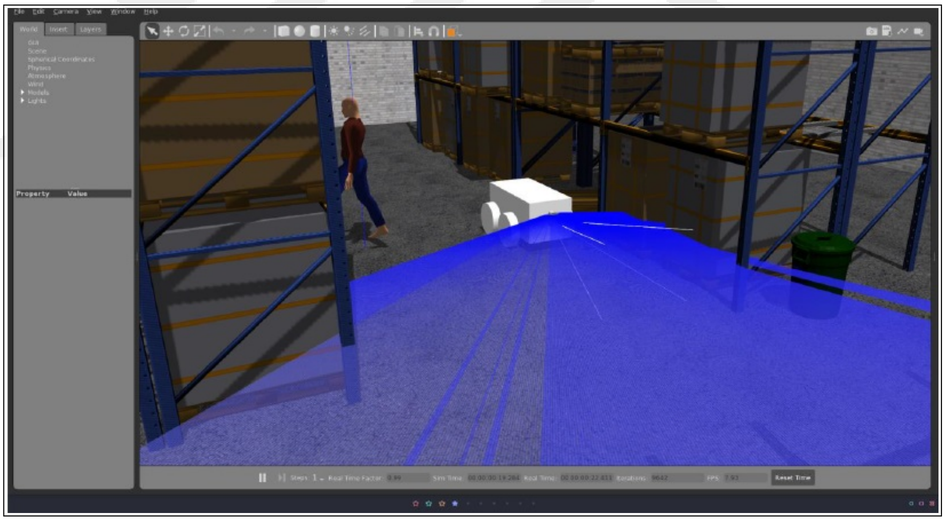
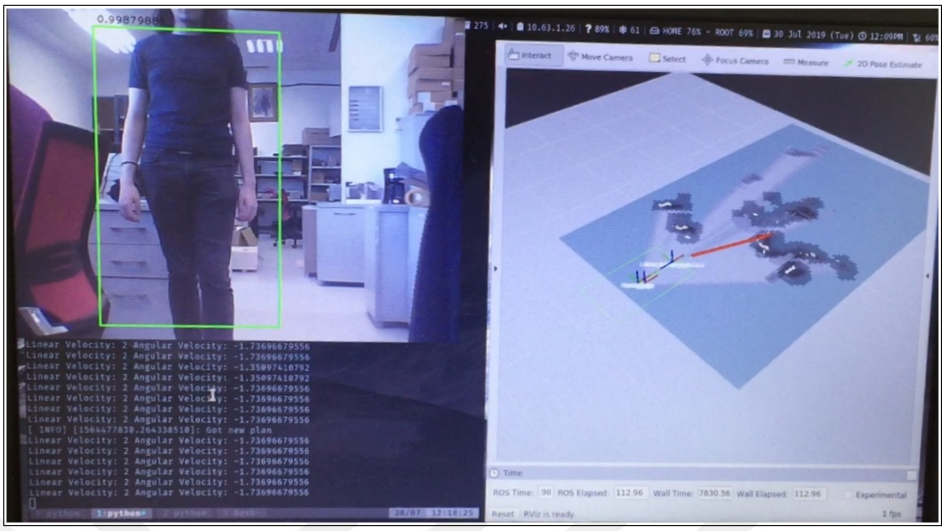


Figure 5.1. SLAM Gazebo simulation

5.1.2. Obstacle and Human Detection Test Results

After that the real time tests are made and match by the real world environment and motion. These result is satisfied for the autonomous mobile robot motion by the laboratory tests shown in Figure 5.2.

Human detection is made correctly and according to the navigator vector output, the direction of the robot where the next move direction is shown in the right side of the figure.



```
Linear Velocity: 2 Angular Velocity: -1.73696679556
Linear Velocity: 2 Angular Velocity: -1.73696679556
Linear Velocity: 2 Angular Velocity: -1.35097410792
Linear Velocity: 2 Angular Velocity: -1.35097410792
Linear Velocity: 2 Angular Velocity: -1.73696679556
Linear Velocity: 2 Angular Velocity: -1.73696679556
Linear Velocity: 2 Angular Velocity: -1.73696679556
Linear Velocity: 2 Angular Velocity: -1.73696679556
[ INFO] [1564477830.264338510]: Got new plan
Linear Velocity: 2 Angular Velocity: -1.73696679556
Linear Velocity: 2 Angular Velocity: -1.73696679556
Linear Velocity: 2 Angular Velocity: -1.73696679556
Linear Velocity: 2 Angular Velocity: -1.73696679556
Linear Velocity: 2 Angular Velocity: -1.73696679556
Linear Velocity: 2 Angular Velocity: -1.73696679556
[]
 0 python   1:python*   2 python   3 bash-                30/07  12:10:25
```

Figure 5.2. Human detection and clean road

The second test of this part is the distance tests of the vision module. Test 1, Test 2 and Test 3 was made for the measurement of the distance accuracy of the algorithms that decide the real world estimation of the environments. The distances of the detected object, wall or human that are been seen by the vision algorithms must be tested and confirmed by the experiments. The distance accuracy tests were made and the specific area that signed are measured in the real world space and then match by the results that are coming from computer vision module of the robot. The points which is assigned as a test points are shown in the Figure 5.3 like A,B,C and D points. Then these distances matched are shown in the Table 5.1.

5.1.3. Person Identification and Human Tracking Test Results

Human detection algorithm detect the person who will navigate the robot by getting needed feature extraction and identification. Then robot can follow identified person. After the feature extraction and identify special person, human tracker algorithm is started to track identify person. This algorithm is giving a similarity score why the verify the following on the true person. For this reason we tested the tracker algorithm after the detection and identification parts of the system is completed, some object and person and also identified person are entered the vision perspective and observe the similarity score by these

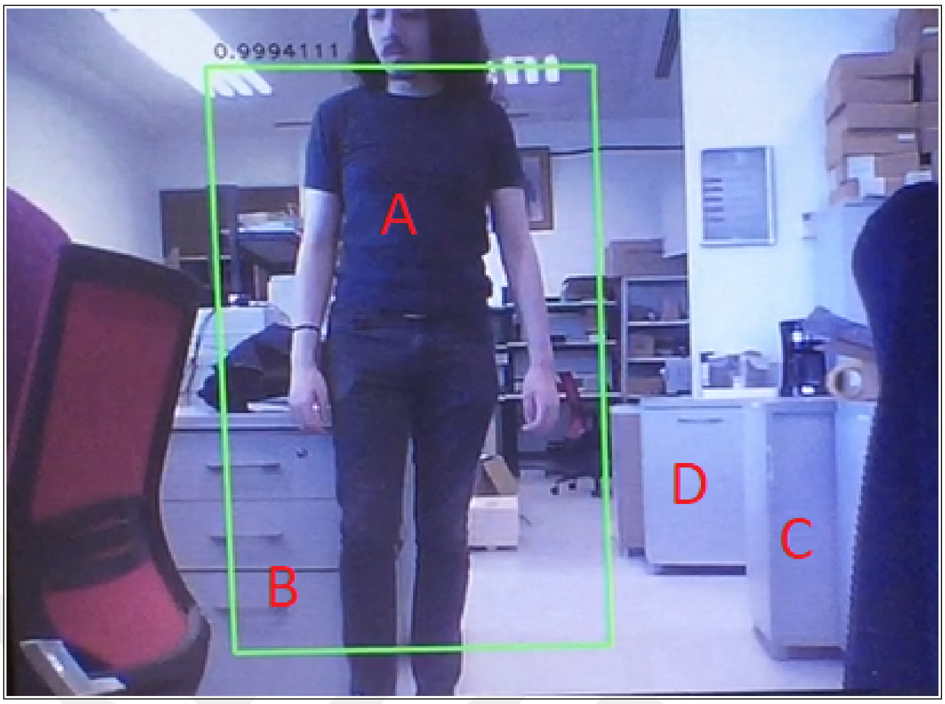


Figure 5.3. Distance measurement test points

Table 5.1. Distance accuracy test

	Test 1 (%)	**Test 2 (%)**	**Test 3 (%)**
Point A	99,7	99,7	99,7
Point B	99,8	99,8	99,5
Point C	99,7	99,2	99,6
Point D	99,8	99,8	99,6
Point E	86,8	89,5	87,1
Average	97,1	97,6	97,1

experiments that are the results shown in Table 5.2.

Test 1, Test 2 and Test 3 were performed to confirm the tracking of the identified human. A user defined in the warehouse environment includes tests on other objects interfering while the robot is moving, and on the fact that people do not mislead the tracking mechanism of the system. In accordance with the attributes gathered from the person defined here, it was determined that the threshold of the similarity rates that our algorithm seeks in every scene should be. When the similarity rates were examined, the percentage of similarity rates assigned to other objects from the defined person was ensured to follow the system towards the right and unfaithful way.

According to the these tests and get the high accurate results, the basis of autonomous driving and tracking was realized.

Table 5.2. Person identification test

	Test 1 (Similarity Score (%))	**Test 2 (%)**	**Test 3 (%)**
Identified Person	99,9	99,9	99,9
Unknown Person	61,7	35,4	23,8
Chair	12,5	29,7	25,1

The results of this experiment were also useful for us to determine the threshold value of the human tracker algorithm. After the identification tests and the decide the threshold of the tracking algorithm, then trial of the trackers success is done as in the shown in Figure 5.4.

5.1.4. Autonomous Driving Test Results

All the parts of the system as software and hardware tests are made and verified correctly. This autonomous mobile robot can be moved correctly safely and as follower on the identified person in real time working by the robot on embedded system shown in Figure 5.4.

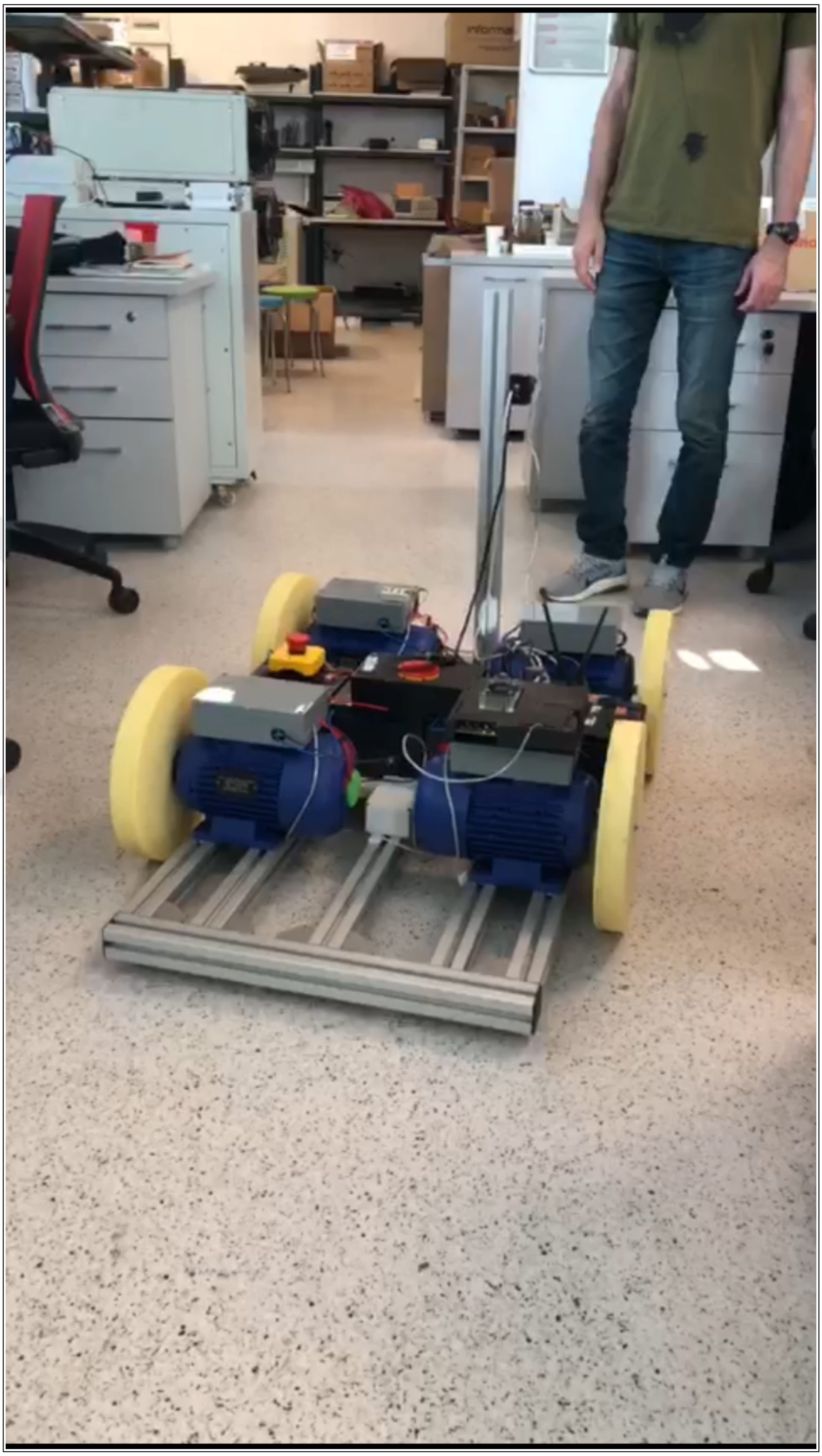


Figure 5.4. Human following mode

5.2. DISCUSSION

This autonomous mobile robot, which is the subject and the output of this thesis work, is a robot that will be used to make the process of collecting goods efficient by following the operator designed for use in storage areas as we mentioned before. These systems are called AGV and AMR. They have different varieties and solutions, as mentioned earlier in the introduction. In this thesis, apart from similar robotic solutions, software and hardware solutions that will improve human tracking and recognition are tried to be produced and thus, it is aimed to be more useful for operations. Unlike traditional structures, in order to establish SLAM and to follow and recognize the person correctly; It is aimed to increase the accuracy by using RGB camera with LIDAR. High success has been achieved in both mapping and recognition tracking algorithms, which can be seen in tests. In addition to using a powerful embedded card, fast algorithms and mathematical models were needed to have a real-time tracking and decision-making mechanism. Therefore, the algorithms which is more fast and have same accuracy were selected as mentioned in the 3rd section [16]. In this way, the robot has been provided to make the senses with the desired accuracy and speed.

SLAM is a tool it needs for autonomous vehicles to realize where it is and where it wants to go. However, when this robot works with human tracking, what it needs is never the whole map. We chose the SLAM type and hardware we use according to realize this goal. The robot creates a 3D environment only in the area of action and movement, and continues on its way by perceiving obstacles and people with SLAM Google Cartographer which is selected to use in this thesis work. Processing speed and competence of the robot are increased with using this SLAM algorithm and hardware, a low cost robot can be designed using less expensive sensors [3].

Unlike similar robots and similar vehicles, studies have been made to follow the targeted person in this robot and to realize the ability to act only with his movements. It was possible to introduce the person to the robot by extracting personalized attributes to the problem of the fact that the human recognition algorithm is not competent and that another person cannot follow successfully when it enters the image of the robot. After this determination, in addition to the human recognition algorithms, our software was supported by algorithms that can identify the person and it was provided to follow only the specified person. Eco tracker algorithm is used to solve this identification step [11]. Except for the person determined by the tests made, it has been successfully determined that he has followed only the identified person by eliminating the people entering the robot usage and imaging area.

As AMR was working in interaction with people, he needed to provide superior security measures as well as his speed and correct recognition structure. It is necessary to detect sudden obstacles by detecting and stop its movement within the determined follow-up distance. In order to meet this requirement, the area defined as the safe zone and set as the minimum distance has to maintain the follow-up distance in order to gain the time required for it to stop even when it has the maximum speed and load when any obstacle is detected. At this point, as stated in the third section, the follow-up distance is defined and the outputs from the DWA algorithms [15,17]. The parameters of the decided clean and optimized road are transferred to the engine management unit and it is ensured that it performs its movement safely.

6. CONCLUSIONS

An autonomous mobile robot designed to be used as a collection vehicle in warehouse areas was designed. For the image processing part of the robot, intensive studies have been conducted in the thesis and successful results have been tried. Today, human-t-robot interaction and collaborations increase their importance and popularity day by day. For this reason, both the smooth operation of the robots and working in harmony with the environment is a must. Robots designed especially for environments in which they can be in direct interaction with human beings such as warehouse and factory and in contact with them should have security measures made in terms of both hardware and software. In this thesis, such conditions as well as the requirements of the system had to be included in the problem. This was taken into account when designing and developing the entire system. A number of improvements have been made while developing the robot's imaging system and therefore the intelligence to allow movement. The robot first had to be able to detect its environment and have instant mapping and location information. SLAM was used and successfully implemented. The next step for the robot, which can scan and make sense of its environment, is to recognize the human being and follow them and identify them in the system. Object recognition, obstacle recognition and identification algorithms are used for this subject. When these conditions work correctly, the data expressed in the imaging center generate output for the robot movement and ensure its movement. This embedded system robot, which is targeted to work for desire hour, is also designed to draw to optimum values in terms of energy and calculation loads. The other clause which must be considered is the robust and real time system design. For this purpose optimum processing unit and sensors are selected according to the capabilities and costs.

To discussion of this proposed autonomous mobile robot system wit the state of the art and the requirements of the warehouse and manufacture areas. As a result of the studies and tests performed on similar robots and vehicles , a number of improvements have been made by taking into consideration the examinations and user comments made on the AGVs and AMRs used to assist in the collection of goods in the warehouses and in the similar areas. The first and perhaps one of the most important advantages is undoubtedly the cost.

Compared to robots with similar tasks, pricing is less than a fifth. In this way, a prototype has been designed with a high probability of finding most customer market share in the market. Hardware and software improvements have been made to achieve these small costs. It has been developed to meet all the needs of the system with cheaper and simpler sensors.

As a future this robot is easily developable to its more complicated and succeed version. Ultimately, this tool is a prototype, and it needs to go through several certification and safety tests before it can be turned into a real product. It also has many points that can be improved in terms of mechanical design and aesthetic design. Before attempting to enter the productive period, many more attempts should be made on hardware choices. This computer imaging system developed by using Camera and Lidar sensor together should be tested with different sensor fusions and even with a single camera or depth camera. In this way, the system can be implemented with less labor and cost. At the same time, the system can be supported by inertial motion unit (IMU) sensors to improve the accuracy of autonomous driving. After these steps have been carried out, the system can be developed in a fully autonomous way to produce robots that work just like it, and then a colony can be established to work together.

REFERENCES

1. Ibragimov ZI, Afanasyev IM. Comparison of ROS-based visual SLAM methods in homogeneous indoor environment. *Workshop on Positioning, Navigation and Communications (WPNC)* on; 2017: IEEE.

2. Rashid MZA, Shah HNM, Aras MSM, Kamaruddin MN, Kassim AM, Jaaafar HI. Metal line detection: A new sensory system for line following mobile robot. *Journal of Theoretical and Applied Information Technology*. 2014; 64(3): 756-764.

3. Hess W, Kohler D, Rapp H, Andor D. Real-time loop closure in 2D LIDAR SLAM. *IEEE International Conference on Robotics and Automation (ICRA)* on; 2016: IEEE.

4. Ismail AH, Ramli HR, Ahmad MH, Marhaban MH. Vision-based system for line following mobile robot. *IEEE Symposium on Industrial Electronics and Applications* on; 2009: IEEE.

5. Baoguo L, Chunxi Z. Adaptive fuzzy control for mobile robot obstacle avoidance based on virtual line path tracking. *IEEE International Conference on Robotics and Biomimetics* on; 2006: IEEE.

6. Norhashim, MA, Mohamad FM, Noorfadzli AR. Single infra-red sensor technique for line-tracking autonomous mobile vehicle. *IEEE 7th International Colloquium on Signal Processing and its Applications* on; 2011: IEEE.

7. Karlsson N, Bernardo di E, Ostrowski J, Goncalves L, Pirjanian P, Munich ME. The vSLAM algorithm for robust localization and mapping. *IEEE International Conference on Robotics and Automation* on; 2005: IEEE.

8. Kokert J, Hoflinger F, Reindl LM. Indoor localization system based on galvanometer-laser-scanning for numerous mobile tags (GaLocate). *International Conference on Indoor Positioning and Indoor Navigation (IPIN)* on; 2012: IEEE.

9. Toledo-Moreo R, Gruyer D, Lambert A. A theoretical analysis of the extended Kalman filter for data fusion in vehicular positioning. *11th International Conference on ITS Telecommunications* on; 2011: IEEE.

10. Dong XM, Yuan K. Landmark design and real-time recognition based on fisheye image for robot navigation. *2nd International Conference on Computer Engineering and Technology* on; 2010: IEEE.

11. Danelljan M, Bhat G, Khan F, Felsberg M. Eco: Efficient convolution operators for tracking. *IEEE Conference on Computer Vision and Pattern Recognition* on; 2017.

12. Xiao T, Li H, Ouyang W, Wang X. Learning deep feature representations with domain guided dropout for person re-identification. *IEEE Conference on Computer Vision and Pattern Recognition* on; 2016.

13. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C, Berg A. Ssd: single shot multibox detector. *European Conference on Computer Vision* on; 2016: Springer.

14. Fox D, Burgard W, Thrun S. The dynamic window approach to collision avoidance. *IEEE Robotics and Automation Magazine.*1997;4(1):23-33.

15. Liu W, Ruixin Y, Guangrui W, Lei S. Local path planning algorithm for blind-guiding robot based on improved DWA algorithm. *Chinese Control And Decision Conference (CCDC)* on; 2019: IEEE.

16. Zhang Z, Saligrama V. Zero-Shot Learning via Joint Latent Similarity Embedding. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* on; 2016.

17. Kojima J, Kushita N, Murakami N, Kuwana A, Kobayashi H. DWA Algorithm for Band-Pass DAC with Ternary Unit Cells. *IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)* on; 2018.