Follow Me Function Development For Service Robot

By

Khoo Lin Yu

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfilment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (Hons)

COMPUTER SCIENCE

Faculty of Information and Communication Technology
(Kampar Campus)

MAY 2019

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: ____

____

____

Academic Session: _____________

I ____

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

_________________________ (Author's signature)

_________________________ (Supervisor's signature)

Address:

____

____

____

_________________________ Supervisor's name

Date: _____________________

Date: _____________________

Follow Me Function Development For Service Robot

By

Khoo Lin Yu

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfilment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (Hons)

COMPUTER SCIENCE

Faculty of Information and Communication Technology
(Kampar Campus)

MAY 2019

DECLARATION OF ORIGINALITY

I declare that this report entitled "**Follow Me Function Development For Service Robot**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : __________________________

Name : __________________________

Date : __________________________

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors, Mr.Teoh Shen Khang who has given me this bright opportunity to engage in this human-following function development robotic project. It is my first step to establish a career in robotic development field. A million thanks to you.

Finally, I must say thanks to my friends, my parents and my family for their love, support and continuous encouragement throughout the course.

ABSTRACT

Human-following robot is one of the most popular research areas in recent robotics study. The aim of develop this human-following robot is to study on how the robot can perform owner detection and tracking so that it can follow behind the target person closely and able to track back once it lose the sight of the target. This will be a great ideas that can helps the elderly in performing various activities with other function implement and meanwhile the robot can follow behind them. Object detect and recognition, distance measurement from a camera to an object in real time using depth image data, and robot control will be implemented in this project. In this project, a marking logo is used to be trained as the target to be detected using SIFT feature extraction and detection. Then, the detected target will be bound with a bounding box and its centroid will be identified. The centroid is used to locate the detected target in the depth image in order to obtain the distance. Once the distance is collected, robot control will be implemented based on the distance value. Software such as OpenCV Python and hardware such as TurtleBot3 and a Creative Senz3D camera is used to implement this project.

TABLE OF CONTENTS

TITLE PAGE **I**

DECLARATION OF ORIGINALITY **II**

ACKNOWLEDGEMENTS **III**

ABSTRACT **IV**

TABLE OF CONTENTS **V**

LIST OF FIGURES VI

LIST OF TABLES VII

LIST OF ABBREVIATIONS VIII

CHAPTER 1 INTRODUCTION **1**

1.1 Problem Statement 1

1.2 Background and motivation 1

1.3 Project Objectives 4

CHAPTER 2 LITERATURE REVIEW **6**

2.1 Learning in Robotics 6

2.1.1 ML for Logo Detection 7

2.1.2 Scale-Invariant Feature Transform 8

2.2 Human Following Robots 12

2.3 People detection and tracking 14

2.3.1 Histograms of Oriented Gradients (HOG) for People Detector 14

2.3.2 People Detection using RGB-D data 16

2.4 Obstacles Avoidance 18

2.5 People following 20

CHAPTER 3 SYSTEM DESIGN **22**

3.1 Turtlebot3 Component Layer 22

3.2 System Block Diagram 23

CHAPTER 4 Proposed Method and Approach **26**

4.1 Design Specifications 26

4.1.1 Prototyping Methodology 26
4.1.2 Hardware 27
4.1.3 Software 30

CHAPTER 5 IMPLEMENTATION AND TESTING 34

CHAPTER 6 CONCLUSION 41

BIBLIOGRAPHY 44

APPENDIX - A POSTER A-1
A.1 Poster A-1

APPENDIX - B TURNITIN RESUT B-1
B.1 Plagiarism Check Result B-1

LIST OF FIGURES

Figure Number	Title	Page
Figure 1.1	Robot industry market projections through 2035	2
Figure 1.2	Statistic of Japan’s elderly population since 1950 to 2016	3
Figure 1.3	Follow-me robot system flow chart	4
Figure 2.1	Machine Learning Algorithm	6
Figure 2.2	Main steps of recognition pipeline	7
Figure 2.3	SIFT feature matching	8
Figure 2.4	Lowe’s Pyramid Scheme	9
Figure 2.5	Key Point Localization	9
Figure 2.6	Eliminating the Edge Response	10
Figure 2.7	Keypoint localization with orientation	11
Figure 2.8	Lowe’s Keypoint Descriptor	11
Figure 2.9	Overview system of stereo vision based CNN tracker	12
Figure 2.10	Input or output data and the main operations detection and tracking modules block diagram	15
Figure 2.11	The people detection qualitative results using the RGB-D data and combination of HOD detector	17
Figure 2.12	The rectangular window which presents the possible speed (v,w) for the dynamic window approach.	18
Figure 2.13	The concept of DGPF method	19
Figure 2.14	Organization of the people-following algorithm	20
Figure 2.15	Overview of the initialization	20
Figure 2.16	Illustration of Depth Segmentation	21
Figure 3.1	Component Layer of Turtlebot3 Burger	22
Figure 3.2	Block Diagram of SIFT	23
Figure 3.3	System flowchart	24
Figure 3.4	ROS Block Diagram	25
Figure 4.1	Layout of Prototyping Methodology	26
Figure 4.2	TurtleBot3 Burger	27

Figure 4.3	Creative Senz3D	29
Figure 4.4	ROS communication illustration using topics and services method	30
Figure 4.5	The overview function of OpenCV	32
Figure 4.6	Overview of hardware and software requirements	33
Figure 4.7	Example of SIFT in OpenCV Python	33
Figure 5.1	Detected marking logo with boundary box	34
Figure 5.2	Detected target with bounding box and center point	34
Figure 5.3	The distance of the center point	35
Figure 5.4	Depth image	35
Figure 5.5	Depth image and distance captured by Senz3D	36
Figure 5.6	Depth image and distance captured by Senz3D	36
Figure 5.7	SIFT code (read in the marking logo and perform feature extraction)	37
Figure 5.8	Feature extraction	38
Figure 5.9	Matched features	38
Figure 5.10	SIFT code (bounding box and center point)	39
Figure 5.11	Subscribe RGB and Depth image	40
Figure 5.12	Robot move	40
Figure 6.1	Undetected target	42
Figure 6.2	Failed to capture distance	42

LIST OF TABLES

Table Number	Title	Page
Table 4.1	The specification of the TurtleBot3	28

LIST OF ABBREVIATIONS

CMOS	Complementary Metal–Oxide–Semiconductor
CNN	Convolutional Neural Network
DGPF	Dynamic Goal Potential Fields
DOG	Difference of Gaussians
HOD	Histogram of Oriented Depths
HOG	Histogram of Oriented Gradients
IR	Infra-red
ML	Machine Learning
OpenCV	Open Source Computer Vision Library
PID	Proportional Integral Derivative
RGB-D	Red, Green, Blue, and Depth
RGBSD	RGB Stereo Depth
ROS	Robot Operating System
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous localization and mapping (SLAM)

Chapter 1: Introduction

1.1 Problem Statement

As human getting older, the ability to respond starting to reduce, most of the elderly will find difficulties in performing some simple task effectively as compared to younger adults. Hence, it is important for having an assisted living which can assist them in daily activities.

An intelligence human-following robot will be a great idea which not only will follow tightly behind the elderly to keep an eyes on them but also helping them in various daily activities by implementing some other extra functionality. For instance, grocery carrying which the robot not only help the elderly to carry the grocery stuff but also will follow closely behind the elderly back to home. Moreover, a human-following robot can also accompany the elderly to anywhere such as performing simple walking exercise which not only help in maintaining the health of the elderly but also reduce the feeling of loneliness.

1.2 Background and motivation

Robotics technology has been evolved in a rapid pace across the years. In recent years, there is an increasing use of service robots all around the world that can assist human in daily activities and is capable to perform various tasks and applications which make human everyday life more easier. One common human-robot interaction that has been studied quite frequently that is the robotic human-following.The human-following robot is able to perform owner detection and tracking and thus it can follow behind them anywhere and thus provides a better interaction between them.

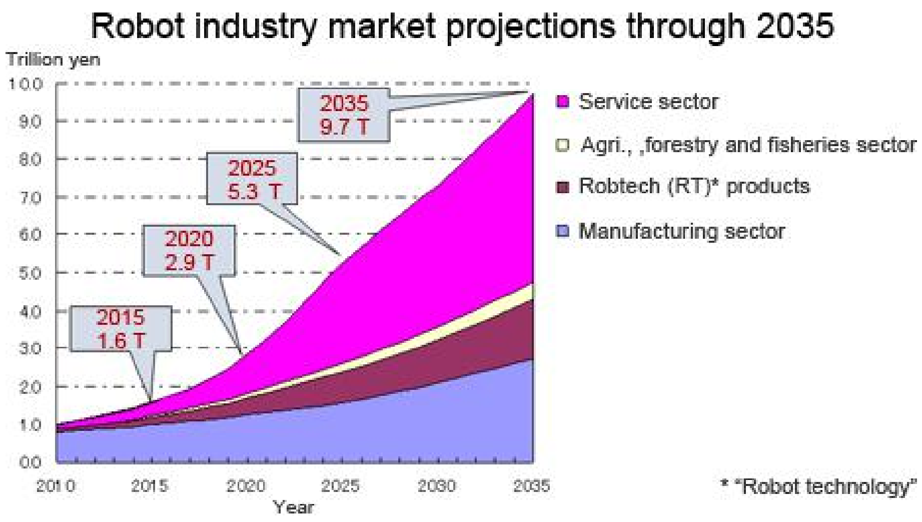


Figure 1.1 Robot industry market projections through 2035

According to report of (Ministry of Economics and Trade), service robots has become the dominance in robot industry market as shown in figure above where the service robots have overtaken the leadership. Service robots are usually designed for more sophisticated tasks compare to other devices, so that the robot can perform the communication task between human-robot interaction to aid in human daily activities.

Japan is the one of the most rapid ageing societies with citizens who age 65 and above are accounting for more than one quarter of the whole population which is 127 million. According to research, the ageing society in Japan will face a expected shortfall of 370,000 health professional by 2025. Thus, Japan government would like to increase the community acceptance towards the robotic technology which can help to fill up the gap in nursing workforce. According to the estimation of Ministry of Economy, Trade, and Industry of the Japan, it was predicted that the market growth for nursing-care robots will be growing to 20 fold starting from 2015 to 2025. In point of fact, Japan government is spending one third of the budget on care robots development (Deena 2017).

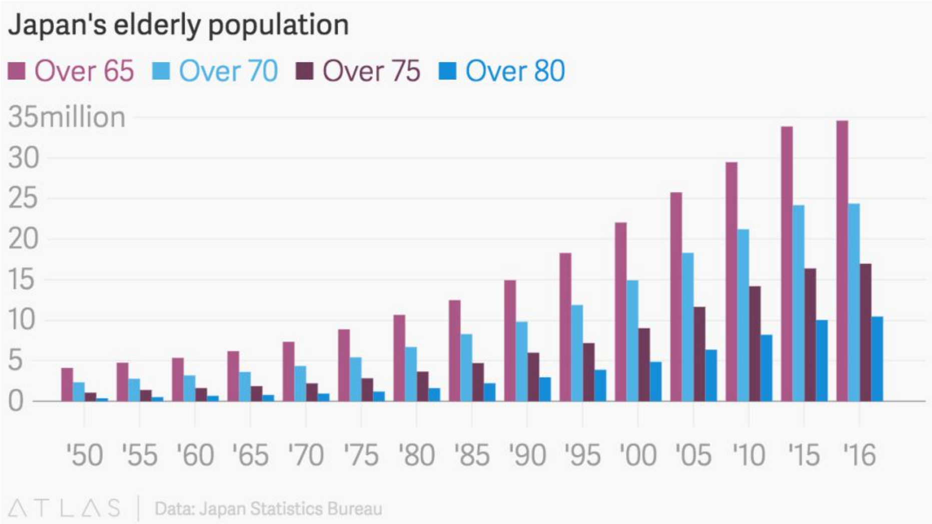


Figure 1.2 Statistic of Japan's elderly population since 1950 to 2016.

One of the example of nursing care service robotic devices in Japan worked by Hirukawa's research is an electric-boosted mobility which help a person to hold onto it when walking around. The sensor will detect if the user is going up the hill and thus activate a booster function. While an automatic brake will kick in in order to reduce falls if the user going down the hill (Hurst 2018). This will be a great help for the elderly in Japan.

According to research (Hamilton, 2013), there are almost 90% of people all around the world over the age of 65 would like to stay in their own house alone for a long period of time which might also become a problem for the elderly in having an society isolation problem. Moreover, loneliness is one of a likely killer that would leads to mental disease which might be deathly if serious. Thus, in order to overcome the loneliness of an elderly and encourage a more healthy living to the elderly, having an intelligent elderly-companion robot is a great solution where the robot can follow the elderly and accompany them at any time anywhere and this will also help the elderly in reducing the feeling of loneliness. Moreover, it is also dangerous not only for the elderly but also a person living alone. So, if the elderly have a robot which can follow them around may also guarantee their safety at the same time by implementing some security functionalities.

Besides, for the elderly, living assisting is important to them to access in some important activities such as medication. Some elderly may also need help with the daily activities such as of eating, bathing, and dressing up. It will be a great solution if the robot can follow closely behind the elderly to everywhere in the house automatically through visual processing and performs the tasks such as helping in eating and medication.

1.3 Project Objectives

The main objective in this project is to develop a people-following service robot which can detect and follow behind the target owner.

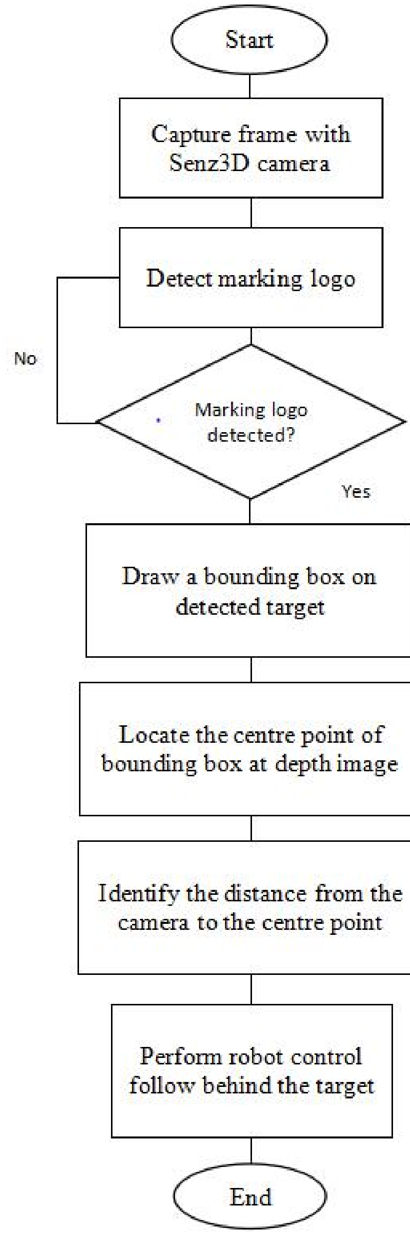


Figure 1.3 Follow-me robot system flow chart

First, a marking logo is designed and a machine learning is implemented to detect and recognize the marking logo which is used to recognize as the target owner. The robot will just have to follow behind the detected marking logo in order to keep track on the owner. Here, Scale Invariant Feature Transform which also known as SIFT was implemented to detect the features and keypoints from an input trained picture (the marking logo) and perform matching to the query image capture with live camera in real time.

There is a bounding box which shows that the marking logo is detected and is recognized as it move around. Besides, the center point of the bounding box is also identified and a circle is drawn to represent it. This center point is used to identify the distance between the camera to the target (marking logo). A Creative Senz3D camera is used to capture the depth image and locate the centroid of the bounding box that detected the target. Once the location of the centroid is detected and captured in the depth frame, depth data will return the value of the distance to the center point in meters. Lastly, robot control is implemented to move the robot following behind the target.

Chapter 2: Literature Review

2.1 Learning in Robotics

Machine learning is a major part of the research in Robotics and is considered as a study field in artificial intelligence which allows the computers to learn form data and making data-driven predictions rather than just following explicit codes. Machine learning algorithms are often being used in Robotics in order to tackle with learning tasks where huge quantities of datasets will be trained to allow the robots to effectively teach themselves accordingly (Mosavi & Varkonyi 2017). To be simple, ML algorithm learns a model from the training samples and that particular one algorithm is fits for different problems.

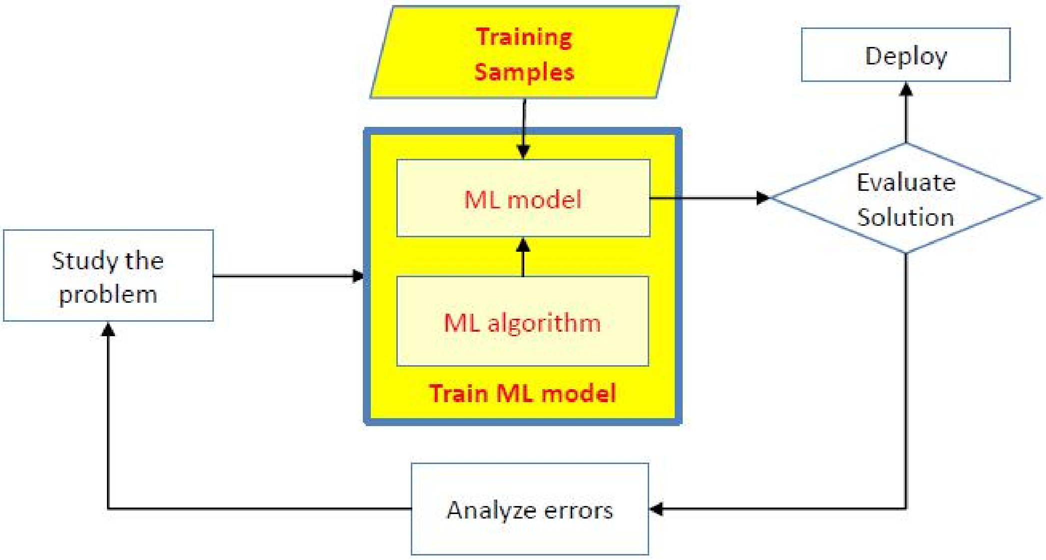


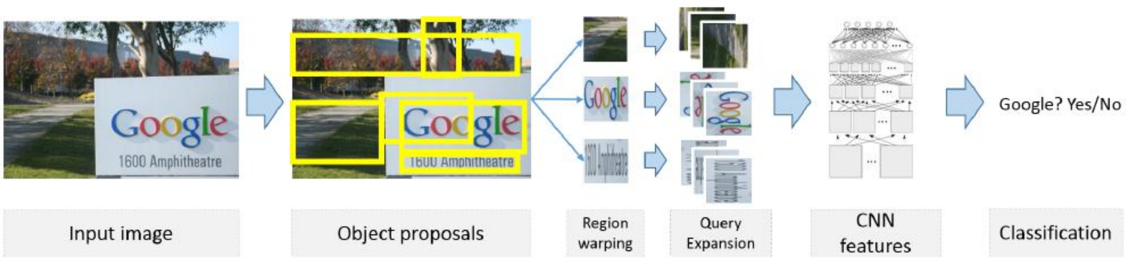
Figure 2.1 Machine Leaning Algorithm

In addition, there are some learning modes which can be tackled for the ML. First, supervised learning techniques where the examples of data is given as an input-output pair and it a function will be learned that joins from input to output where input will be the set of features which used to describe the samples and the output is the attribute interested to predict. Second, unsupervised learning where output value do not have to accessed and only simply have a collection of input examples. The underlying patterns of the data will be learned. Next, in reinforcement learning, there is no direct access to the correct output for an input, but it learns a behaviour based on an environment feedback and receives rewards of taking a particular action (Lison 2015).

Nowadays, machine learning techniques application in robotics has become a popular topics where the learning techniques can range from inductive learning algorithms, rote learning, and analogical reasoning to Explanation Based Leaning and also Case Based Leaning. Besides, there are also many robotics which can perform on neural networks, reinforcement learning and also genetic algorithms.

2.1.1 ML for Logo Detection

(Bianco et al. 2015) has proposed a logo recognition using CNN features. A Convolutional Neural Networks take an input image that undergone some minor image preprocessing. The input image will processed with a different layers cascade of transformation and result in a final prediction of the image class. With a given input image, Bianco extract the regions that are called object proposals which are region that are most likely to have an object. Different regions aspect ratios that can be used in recognizing the object under various geometric transformation is first extracted and then are warped to a common size and was then processed for query expansion to increase recall. Lastly, a pre-trained CNN will act as the extractor of the feature and a linear SVM used for the classification and logo recognition by using the extracted features as the input. Figure below shows the major steps of the recognition pipeline.



Outline of the recognition pipeline: (1) Candidate objects regions are extracted from the image. (2) Regions are warped to a common size and multiplied through Query Expansion. (3) CNN features are computed over each region. (4) Classification is performed using linear SVMs.

Figure 2.2 Main steps of recognition pipeline

2.1.2 Scale-Invariant Feature Transform

(Lowe 2004) proposed a new algorithm naming Scale Invariant Feature Transform where the features from an input images can be extracted and detect. SIFT is able to perform matching among different object view. Figure 2.3 shows that the content of the image is being transformed into coordinates of local feature which are unvarying to different imaging parameter such as scaling, rotation, translation and etc.

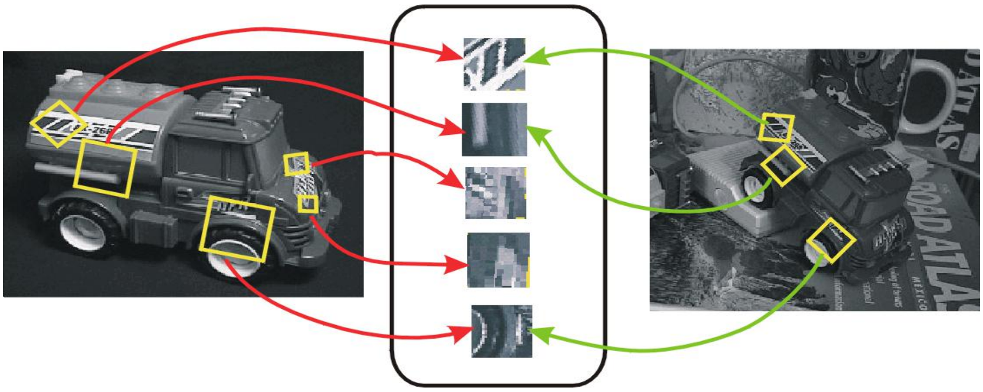


Figure 2.3 SIFT feature matching

There are four steps to be involved in SIFT algorithm. First step, scale-space extreme detection and the second step, keypoint localization. The third step which is orientation assignment and lastly keypoint descriptor. Keypoints here refer to the position of the detected feature. Descriptor represent an array which contains numbers that is used to describe that feature. In the first step of scale-space extreme detection, various scales and coordination of image are searched over. Difference of Gaussians is used to determine the potential keypoint which is scale invariant and orientation. By using two different σ such as σ and κσ, the difference of Gaussian blurring an image to obtain the DOG.The DOG process is done for different image octaves in Gaussian Pyramid as shown in figure 2.4.

In each and every octave, the starting image with Gaussians is continuously convolved to obtain scale space images set. In order to produce the DOG, the adjacent Gaussians are then subtracted. Subsequently each octave, by factor of 2, down-sampled the Gaussian image to create an image which is quarter size to continue the next level.

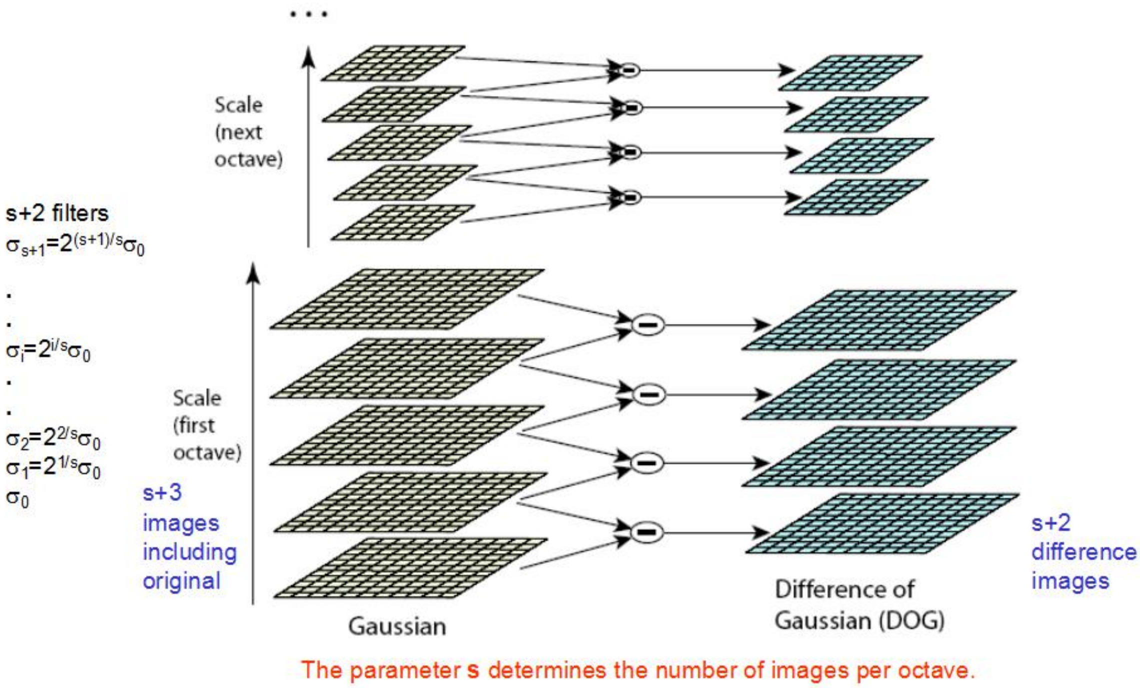


Figure 2.4 Lowe's Pyramid Scheme

Once the DOG is retrieved, the images will look for the local extrema over scale and space. Figure below shows that the DOG is detected by computing the pixel of an images to its surrounding neighbours as well as pixels in next scale and previous scale. If it is a local extrema means that it is also a potential keypoint, the keypoint that is best represent in that scale (OpenCV 2018). Figure below shows that maxima and minima of DOG in scale space are detected by comparing X to its surrounding neighbour at current and adjacent scales. Each of the point is computed each in the scales below and above 9 neighbours and also to its respective 8 neighbours in the current image.

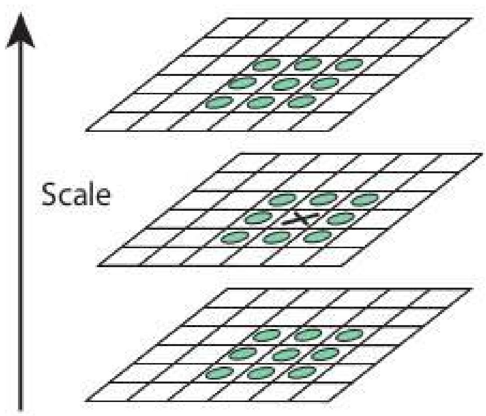


Figure 2.5 Key Point Localization

Keypoint localization states that a clear model will be fit to identify the location and scale at every prospect location. Based on the measure of stability, the keypoint will be chosen. Thus, once the candidate of the keypoint if determined, a detailed fit will be performed to the closest data to identify the scale, location and also the principal curvatures ratio. Initially, the keypoints were identified at the central sample point's location and scale. As DOG has a higher response for edges, to improve the interpolation accuracy, a 3D quadratic function will be fitted and Hessian matrix is used to remove the edge responses. Figure below shows the equation used to eliminate the edge response as mentioned by Lowe in the paper:

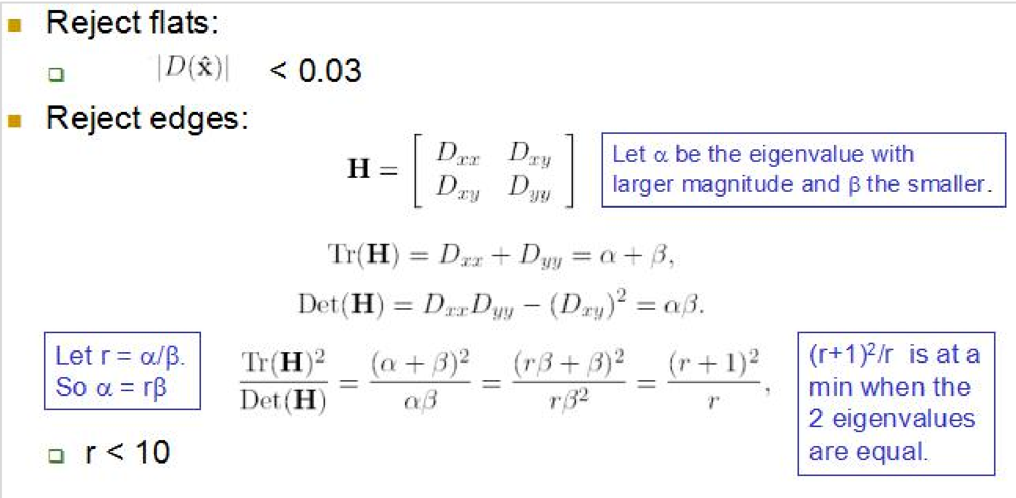


Figure 2.6 Eliminating the Edge Response

Orientation assignment compute for the best orientation for every region of the keypoint. Each keypoint will be assigned to an orientation to obtained invariance to the rotation of the image. Here, a histogram of local gradient directions is created at chosen scale. The highest peak and any peak which is 80% and above will be taken care to measure the orientation. This generates the keypoints with different directions but with the similar scale and also locations.

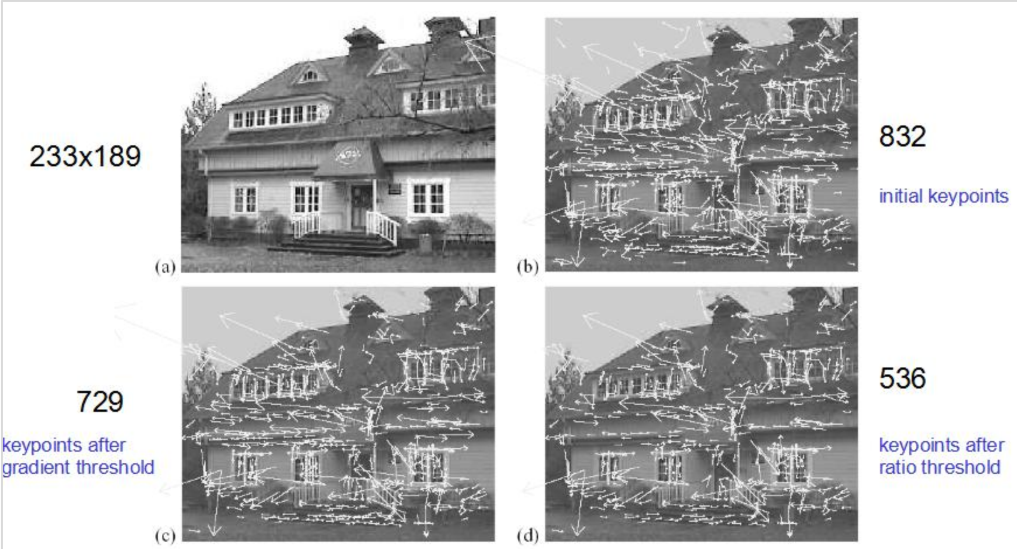


Figure 2.7 Keypoint localization with orientation

Next, keypoint descriptor will be created to compute the keypoint extracted by the SIFT detector and descriptor. For keypoint descriptor, it uses the gradients of the local image at the chosen scale and rotation to define every area of the keypoint (Lowe 2004). Here, take the 16x16 neighbourhood around the keypoint and divided them into 16 sub-blocks of 4x4 size. This create a 8 bin orientation histogram for all sub-block. Thus, a total of 128 bin values are now available.

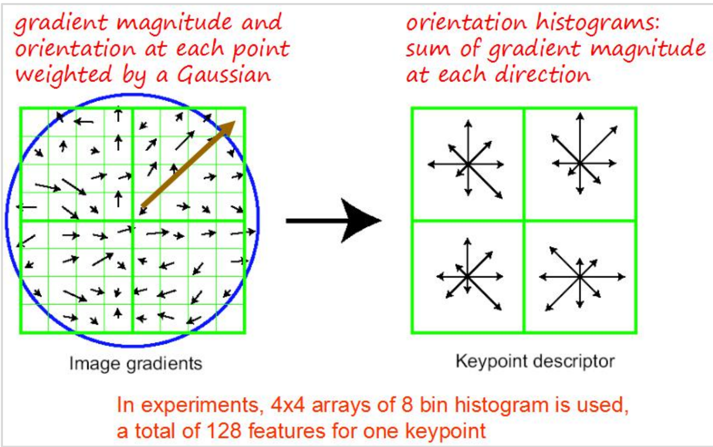


Figure 2.8 Lowe's Keypoint Descriptor

To form keypoint descriptor, it is represented as vector. Besides, in order to obtained robustness against illumination and rotation changes, various measures will also be taken. Lastly, by determining the nearest neighbours, the keypoints between the two images will be matched. In short, to match and recognize an image using SIFT, the SIFT features are initially extract from an input image as references and is stored into database. Then, by computing each and every feature from the database, new image will be matched.

2.2 Human Following Robots

A following-human robot is a robot that can follow behind the human by tracking on the human with the aids of camera. The robot may first detect the target person's face in capture image and using the face image data in determining its movement. Options of the movement such as move to front, stop, move to back, turn right or turn to left is essential to assist the robot in identifying the correct position while following behind the target person. For instance, if the target person or the owner face was detected small in the capture image, the robot will move closer to the owner, otherwise the robot move backwards. And if the face detected on the left or right hand side of the frame, the robot will turning around itself either left or right. Moreover, the robot know how to avoid obstacles along the way while following behind the owner itself.

Since 1998, person following robots has always been a researched where (Schlegel et al. 1998) has proposed the method using colour and contour information of tracking target which also known as contour-based object detection.

Another approach which is similar using colour based tracking was proposed by (Tarokh & Ferrari 2013) using H-S histogram in a HSV space. However, the appearances changes and occlusions could not handle fitly by these proposed approach .

(Chen, Sahdev & Tsotsos 2017) proposed a CNN tracker stereo vision for the people following robot. In their proposed method, CNN models with RGBSD (RGB Stereo Depth) Images is used.

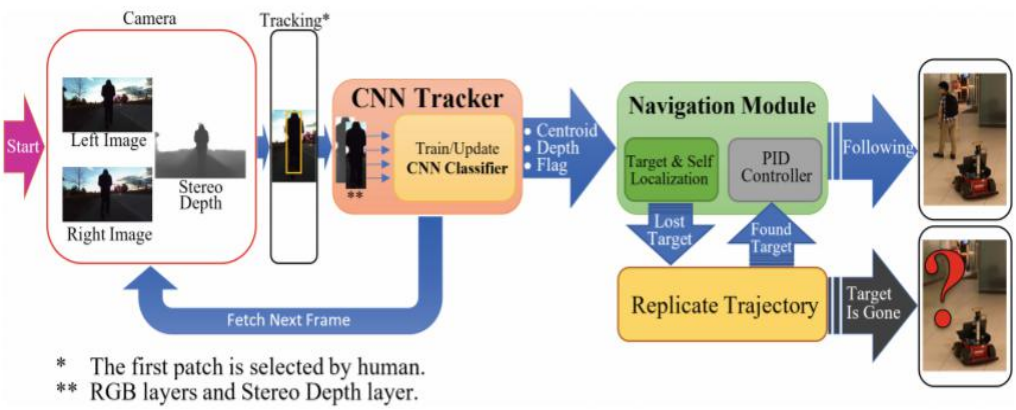


Figure 2.9 Overview system of stereo vision based CNN tracker

Figure 2.2.1 shows the overview system of the approach. The computed stereo depth images together with the RGB image will act as the inputs to CNN and the CNN will return the centroid and the depth of the target being tracked will be the output which will be used for navigation module in order to follow the target and replicated required path.Chen, Shdev and Tsotsos used a PID controller for robot following behaviour in the situation when the target person is detected in the image using equations:

$$v = K_p * (d - D) + K_i * \int_T (d - D)dt;$$

$$\omega = K_p^{'} * (x - X\_mid) + K_i^{'} * \int_T (x - X\_mid)dt;$$

where D is the pre-defined distance that is used to maintained the distance between both robot and detected target. In the proposed approach, if the robot lost the sight of the target, it will replicated the target's local patch and continue look for target.

2.3 People detection and tracking

In the robot's environment, people detection is an important challenges where human interaction are involved in finding people. In order to allow the robot to follow a people closely, people detection and tracking is very useful as it will help the robot to recognize and track on the owner by distinguishing people from the other objects or people in an environment.

2.3.1 Histograms of Oriented Gradients (HOG) for People Detector

Histograms Of Gradients approach is a feature descriptor that used for image analysis. First, the image is divided into few regions and then the regions will be further divided into cells. And for each cell, the gradient angle and magnitude will be used to compute a weighted histogram for a region. The result will then be used as a descriptor.

(Dalal & Triggs 2005) proposed a human detector method using descriptor of a HOG to detect people where the method take conceive on a static size detection window that has been subdivided into cells in uniform grid. The gradient orientations over the pixels for every cell are then compared and stored in a dimension histogram. Without knowing the fine position of the cell, a local gradients distribution characterize the local appearance and shape. The blocks, which is the adjacent cells groups used to normalize the contrast locally. Chaining every block histograms, the descriptor is built and is then used for linear SVM training. Then, the image will be scrolled over by the detection window in various scales for people detection. For each position and scale, with the learned SVM the descriptor of HOG will be computed and classified .

(Munaro, Basso, & Menegatti 2012) has proposed a people detector using sub-clustering and HOG (Histogram of Oriented Gradients) and multiple human tracking algorithm using a RGB-Depth sensor. Munaro utilized a 3D sub-clustering method that gives people detection result efficiently when there are multiple people are very close to each other. Figure below shows the overview of the system proposed by them for the detection and tracking modules.

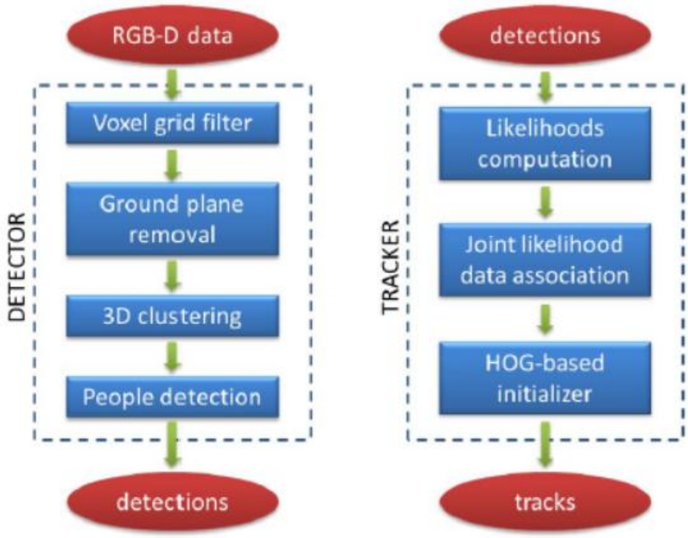


Figure 2.10 Input or output data and the main operations detection and tracking modules block diagram

The point cloud data filtered by a detection module will process the RGB-D data initially and take away the ground and the 3D clustering of the remaining points will be performed. The people detection of HOG based algorithm is used by Munaro to the RGB images of the resulting clusters to maintain only those cluster which belongs to human class. The display result of the detection set will passed to tracking module. Detection-track is performed by the tracking algorithm that associate as a joint likelihood maximization which composed by colour appearance, motion and human detection confidence these three terms. A people classifier for each target is learnt online for the evaluation of colour appearance by using the features that has been pull from the target's colour histogram as the negative examples and also other detections.

For the people detection algorithms used by (Munaro, Basso, & Menegatti 2012), the classifier use the appearance of the colour which is based on Adaboost. In order to have a more finer classification, motion model is used. A probability for a detected person to be the same than another person detected will be the result. However, there is a disadvantage in using this method whereby the people detector is required for every frame thus increase the processor's cost.

2.3.2 People Detection using RGB-D data

A revolution in image algorithm is made by the invention of a RGB-Depth sensors. The RGB-D sensor is a camera that make use a RGB image and also depth image. The sensor give the depth value information of every pixel in the image. The Microsoft Kinect RGB-D sensor consists of an infra-red (IR) camera and projector, together with a standard colour camera. The structured IR light principle will followed by the sensor to calculate the depth. RGB-D image is a combination of an image with the RGB and its related depth image where an image channel in which each and every pixel is related to a distance between the plan of image and the equivalent object in the RGB image. Kinect camera is able to be used to capture the RGB-D image.

Queva, 2013 has proposed a people detection method by implementing a RGB-D sensor in a project making a robot follow a person. The tracking algorithm will take the detector result by using depth segmentation and a region increasing algorithm. According to Queva, this method allows the robot to follow closely to the target person that walks slowly in-front of him.

(Spinello & Arras 2011) has proposed an algorithm of detection of people for RGB-D data which includes both HOG and HOD descriptors. In Spinello and Arras's approach, they were inspired by the Histogram of Oriented Gradient (HOG) method together with the Kinect RGB-D sensor depth characteristics. They developed a dense depth people detection which is robust and is later known as Histogram of Oriented Depths (HOD). Based on a trained scale-to-depth regression and an integral images new usage, Spinello and Arras carry out an informed scale-space operation for HOD .

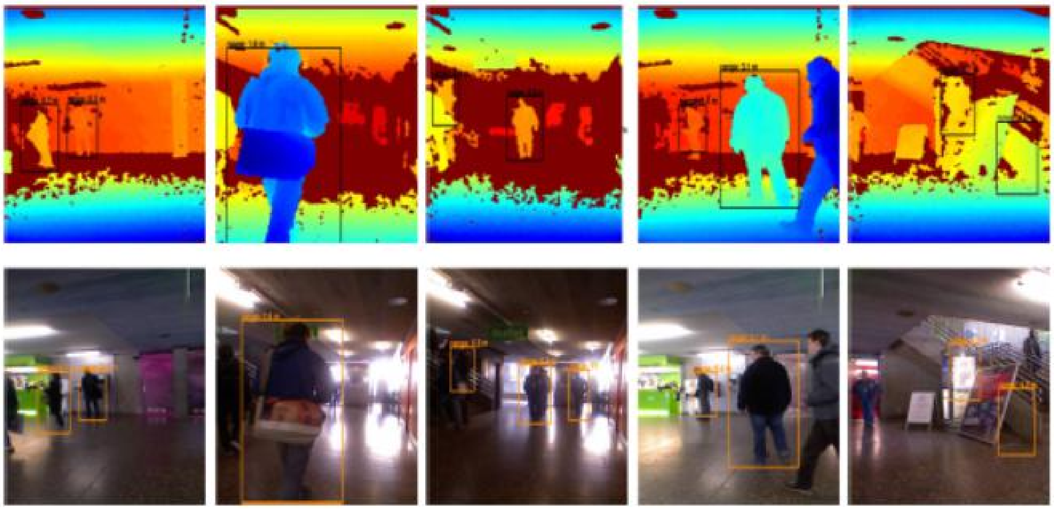


Figure 2.11 The people detection qualitative results using the RGB-D data and combination of HOD detector

The detected people in the figure above are at several ranges and in contrasting visual and also depth cluster. When the data are challenging in both sensor modalities, a false negatives will be occurred. While a false positive will be discover when the visual and depth clutter occur at the same time. In the third picture, although there is no available depth data, the detector is still able to detect a person. The method do not count on background learning or ground plane assumption.

As the summary for people detecting and tracking method, people detector is used in the most of the existing method in order to track on the target person. The sensor gives an image at every frame, and the target people will detected by the people detector initially. After that, people tracking algorithm will be used to identify person detected and associated to the person in last frame. The disadvantage of using this method is that the robot will required a high CPU performance which will take more CPU consumptions to run the algorithms and the execution cost is very high.

2.4 Obstacle avoidance

To avoid collisions with the obstacle, obstacle avoidance is implement while following behind the target person by using local algorithms and sensors to avoid the obstacles when the robot is moving around.

In the earlier 1990, one of the most simplest obstacle avoidance algorithm which is known as the bug algorithm was proposed by (Lumelsky & Skewis 1990). This bug algorithm has the view to follow the contour of each of the obstacle the robot encounter along its path. Another technique called the vector field histogram was then introduced by the (Borenstein & Koren 1991) that creates a local map of the robot environment and then produce a polar histogram that indicate the probability of obstacle finding in that direction.

Another method named Dynamic Window Approach which involves the robot kinematic. Using a velocity-space that represents every possible tuples (v,w) where v represent as the the linear velocity and w represent the angular velocity. With a current given speed of the robot, the algorithm will choose a dynamic window that include every points in the velocity space where the robot can reach within the following period. Then, the algorithm will decrease the window, maintaining the speeds to make sure there is no collision with an obstacle. In order to keep a minimal distance away from the obstacles, a new motion direction will be selected and the robot will move towards the target (Queva 2013).

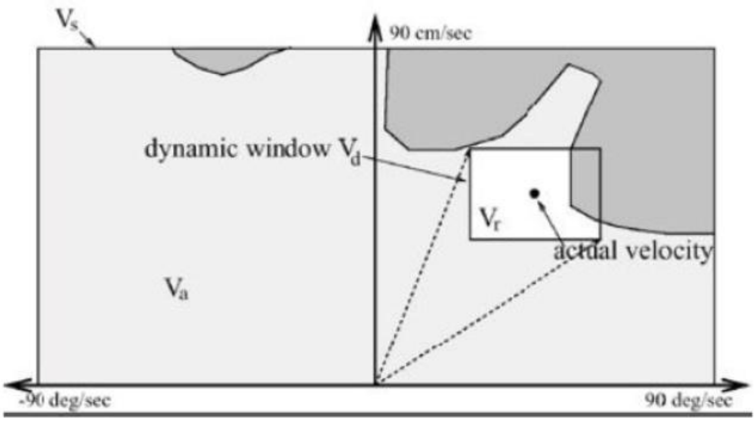


Figure 2.12 The rectangular window which presents the possible speed (v,w) for the dynamic window approach.

One of the obstacle avoidance algorithm that proposed by (Chen & Cheng 2006) is known as the dynamic goal potential fields (DGPF), an obstacle avoidance algorithm which can understand the dynamic goal position issues that take the path adjustment corresponding to the destination changes during obstacle avoidance. The DGPF algorithm use the actual and the configuration of goal together with the sensor data. The robotic platform will then predict the path by running a basic potential fields algorithm. If there is no changes for the goal configuration, the robot will then follow according to the path in avoiding obstacles. If the goal configuration move to another new location, where there is vast changes from the previous location, the algorithm will then choose some points in the path that has been predicted randomly and run the basic potential fields approach. This method will compute many different path and the cost path with the lowest cost will be chosen according to the Euclidean distance and the new path will be used by the robot to move on to new goal configuration.

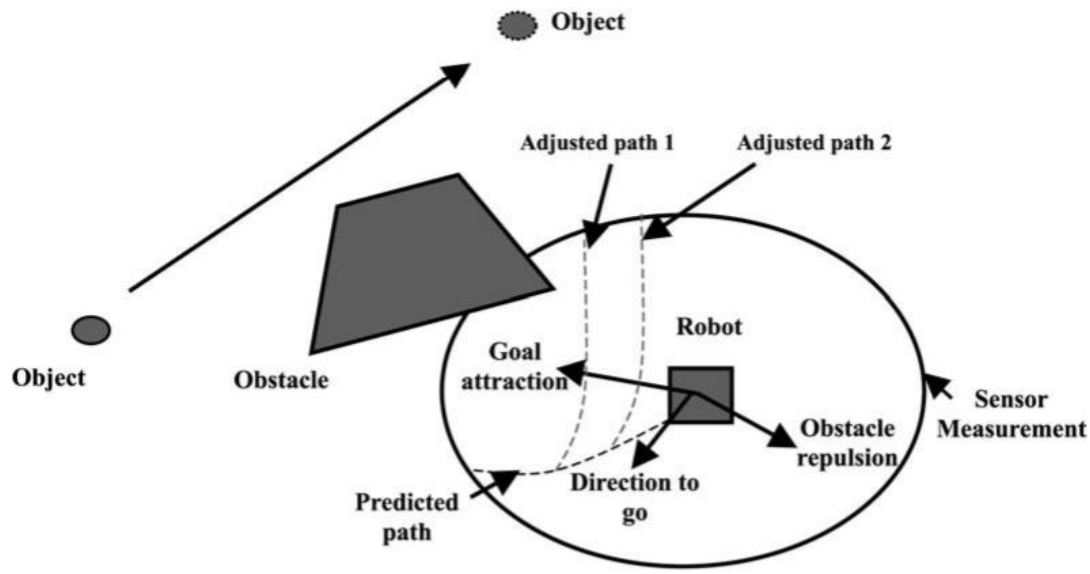


Figure 2.13 The concept of DGPF method

One of the advantages using DGPF is that the requirement of the obstacle avoidance while tracking the object that is moving in real time can be fulfilled.

2.5 People following

The algorithm for people following proposed by (Queva 2013) is shown as figure below:

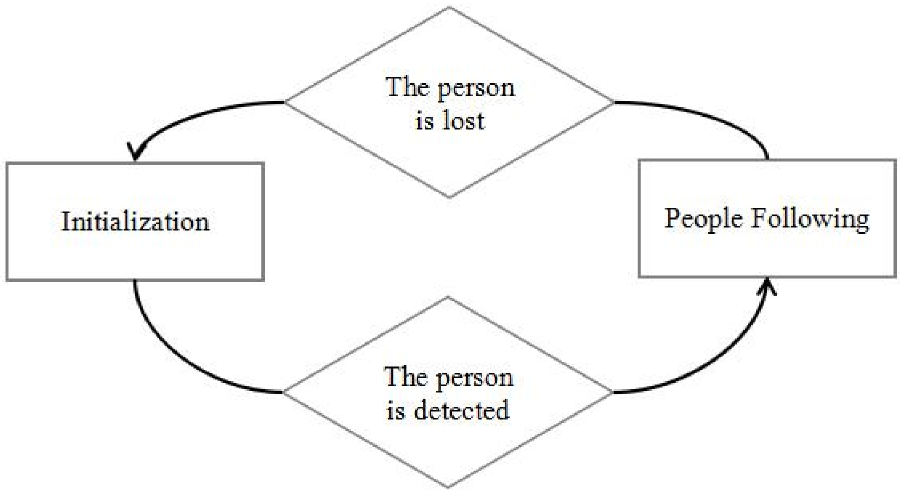


Figure 2.14 Organization of the people-following algorithm

People detector will first used in the initialization step to detect and look for the people. Then, without the people detector, the main loop will continue runs and tries to trace on the target. If the target person is lost of sight and unable to detect, the algorithm will return back to initialization step. If the person is not lost of sight, the program will keep looping.

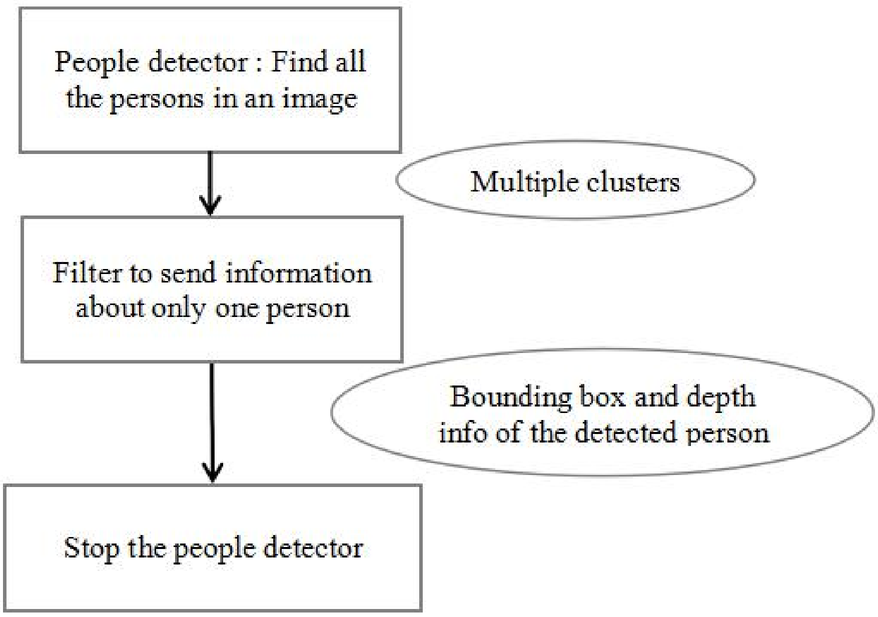


Figure 2.15 Overview of the initialization

(Queva 2013) also proposed depth image segmentation for people following method. Depth image segmentation is a step which expect the target person the robot follow will not move greater than delta millimetres. The assumption is made reasonable as if the target person followed by the robot is moving at the same speed while the robot itself is not moving. The distance between pixels' depth value of the person in new image and depth value which has been obtained in the previous frame will be measured. The distance measured should not be greater than the delta. As both the minimal and maximal depths information of the person contained in the previous frame, each pixel's depth value in the depth image is computed with those values:

$$(depthMin - delta) < depthPixel < (depthMax + delta)$$

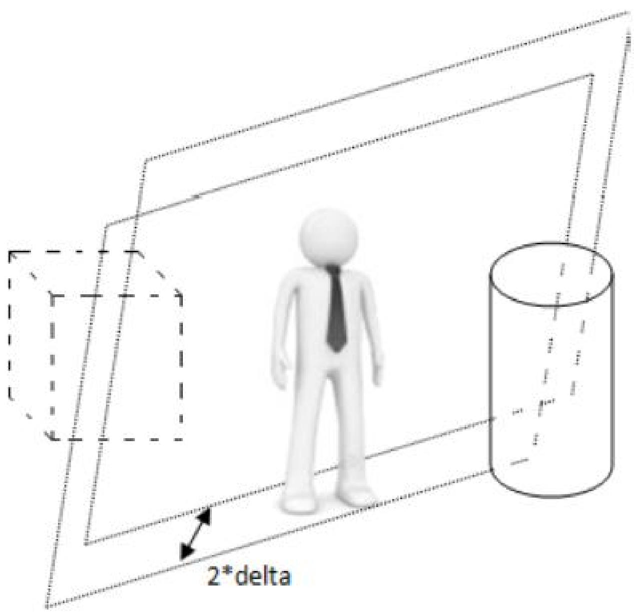


Figure 2.16 Illustration of Depth Segmentation

Figure above show the illustration of the depth segmentation. After the segmentation is done, the cube pixels will be removed and left only the person and some of the cylinder pixels.

Chapter 3: System Design

3.1 Turtlebot3 Component Layer

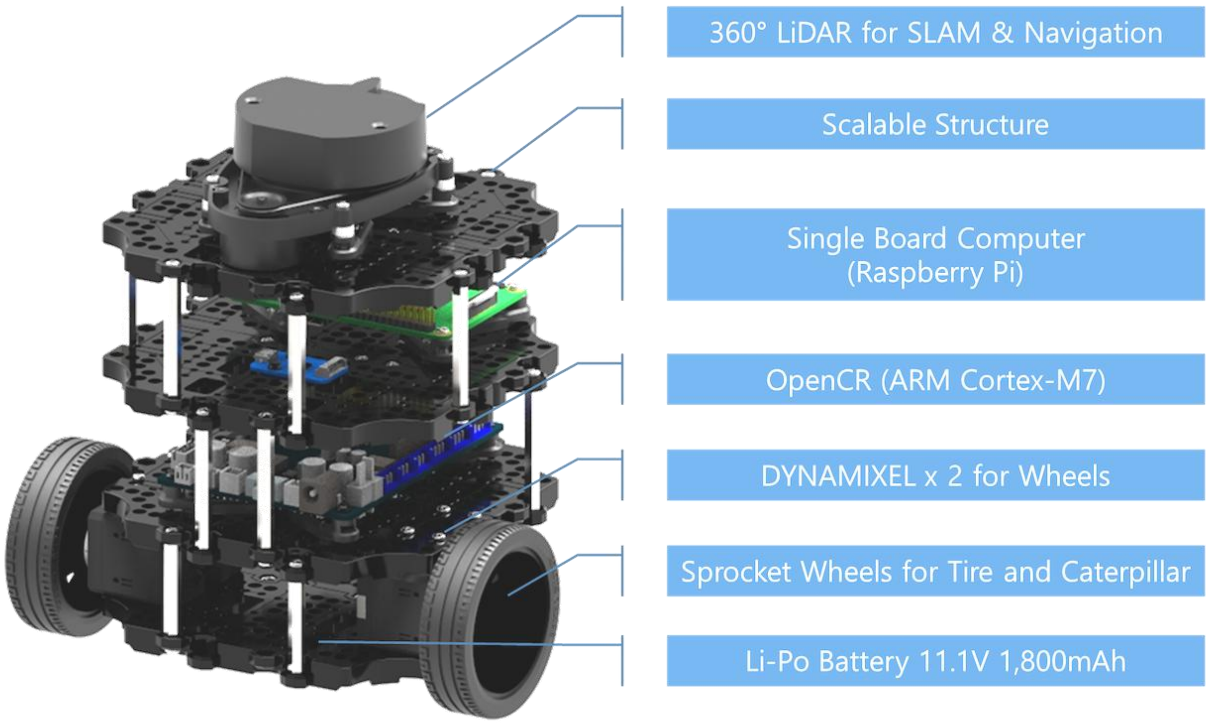


Figure 3.1 Component Layer of Turtlebot3 Burger

In this project, Turtlebot3 Burger will be used. Figure above shows the component layer of Burger.

3.2 System Block Diagram

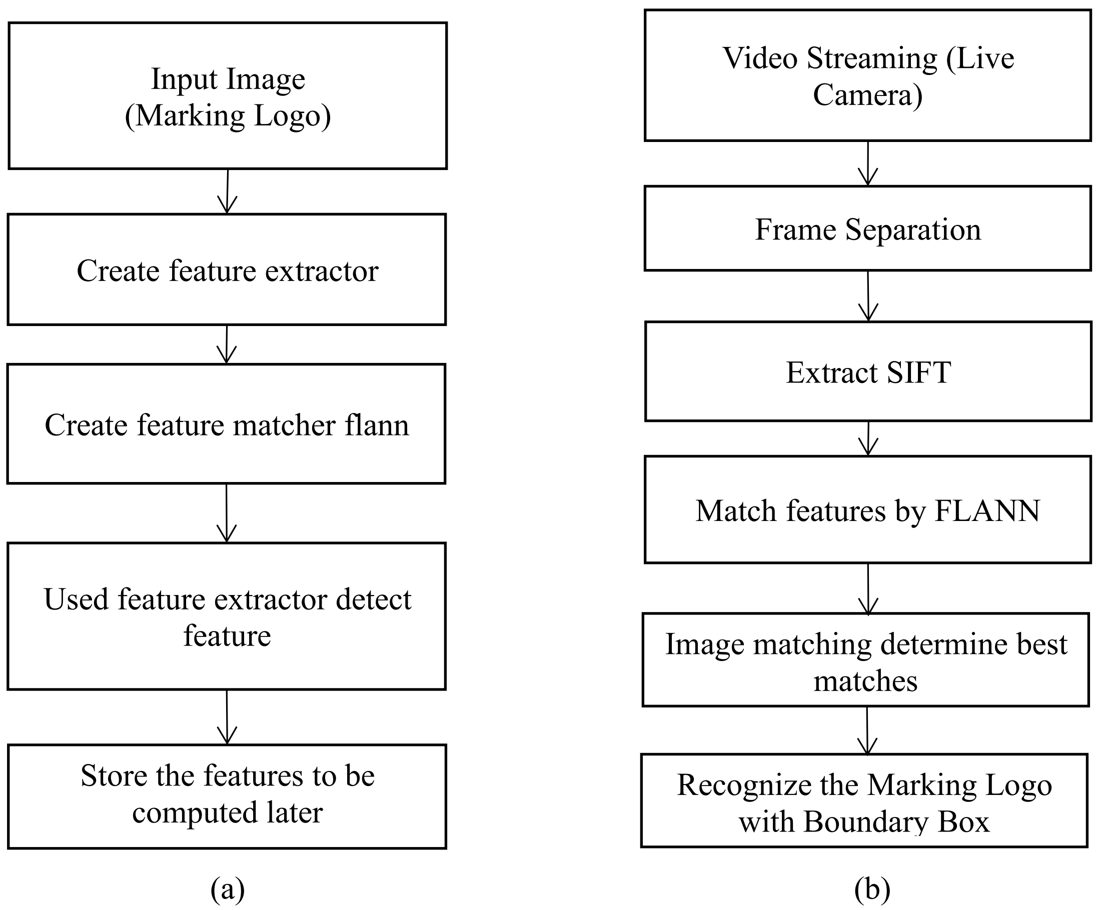


Figure 3.2 Block Diagram of SIFT

Diagram(a) shows the block diagram overview of machine learning for marking logo detection and recognition using SIFT. First, the designed marking logo which is also the training image is read by the programme. Then, create SIFT feature extractor to detect feature by using sift.detectAndCompute() which provided in SIFT library in OpenCV and initialised the feature matcher flann which will be used later to match the features. The detected features or keypoints of the marking logo will be stored. The stored features will be used to compute with the detected marking logo later in real time.

Diagram(b) shows the detection and recognition of the marking logo in real time using live web-came. Once the camera is opened, frame is captured from the camera. The capture frame will be stored as query image and again sift.detectAndCompute() method is used to extract the features from the query image. After that, FLANN feature matcher will match feature in both image (the trained marking logo image and the query image of captured frame), stored the matches results and compared to look for good match. Here, FLANN is also known as Fast Library for Approximate Nearest Neighbours. FLANN consists an algorithms collection that optimized for higher dimensional features and also for fast nearest neighbour search in bigger datasets . A bounding box will be drawn on the query image showing the result of the detected image.

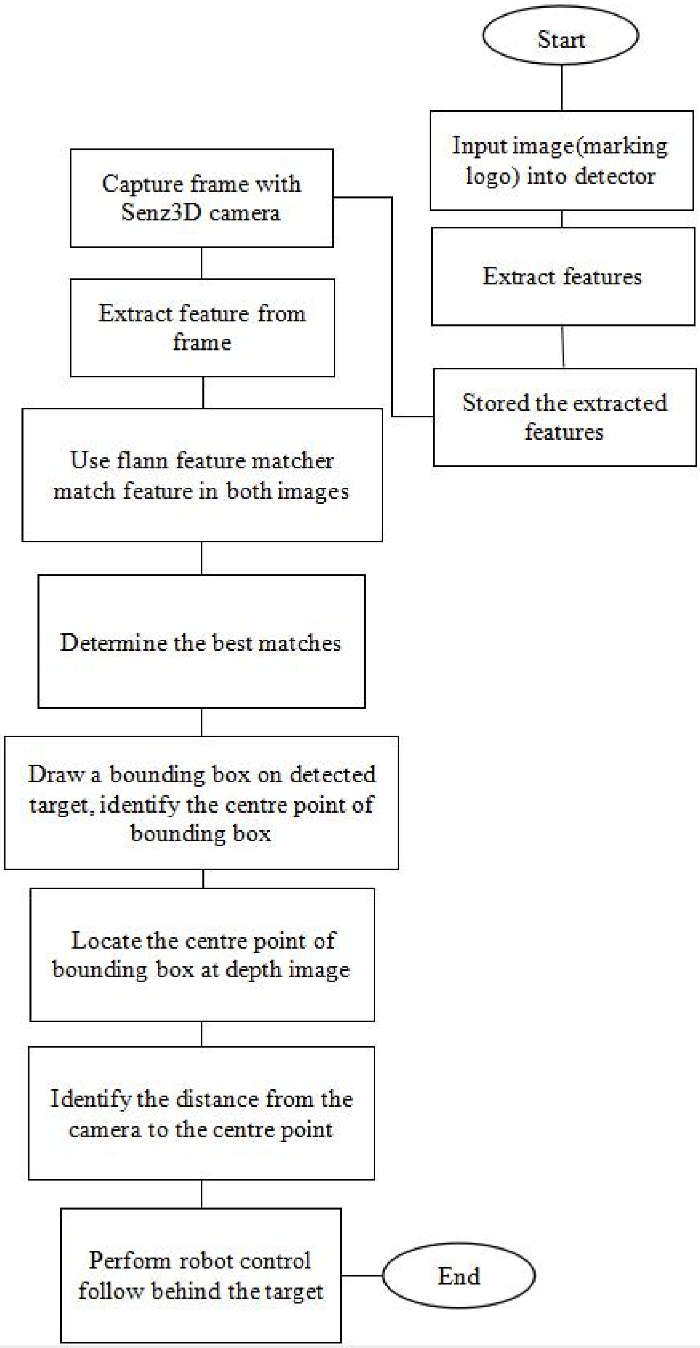


Figure 3.3 System flowchart

The overall system flow of this project is shown in figure 3.3. First, the designed marking logo is loaded to be trained. SIFT is used to for the marking logo detection. The marking logo image is first input into the detector created for SIFT. The detector algorithm then detect the features and keypoints in the image. Then the extracted features will be stored into variables which will be used for later computation. Next, Creative Senz3D camera is opened to capture the frame in real time. Again, the feature will be extracted from the captured frame by SIFT feature extractor. After that, FLANN feature matcher will determine the best matches by matching the featured detected in both images and stored the matches results in another variable.

Once the best match is founded, a rectangle bounding box is drawn around the detected marking logo and the centroid (the center point x and y) of the bounding box is calculated. The centroid is represented by a single circle. The coordination of the detected center point is passed to the depth image and is used to identify the distance from the camera to the center point using Creative Senz3D camera. The distance value is used to move the robot when the distance values is captured. The robot will move forward based on the distance capture. The robot will move toward the target and stop in front where the target stopped by and if the target move again and detected by the robot, then the robot continue to move.

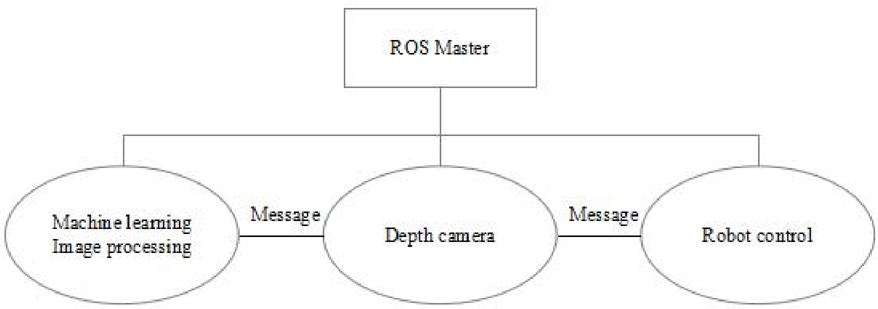


Figure 3.4 ROS Block Diagram

Chapter 4: Proposed Method and Approach

4.1 Design Specifications

4.1.1 Prototyping Methodology

Prototyping methodology will be used in this project. This is used where a preliminary design will be built, tested and then improved as necessary until the final version of the project is implemented when the robot is well constructed, programmed and tested. This project is aim to develop a human-following robot that can detect and follow behind the target person.

Initially, existing and similar project about human-following robot were reviewed and different machine learning regarding object detection method were analysed. After that, a preliminary design of the ML block diagram is constructed and implemented resulting in the marking logo detection and recognition using SIFT. Next, distance measurement between the robot and the marking logo is implemented by using the 3D camera. And lastly, robot control is programmed and constructed to allow the robot to follow the target as desired.

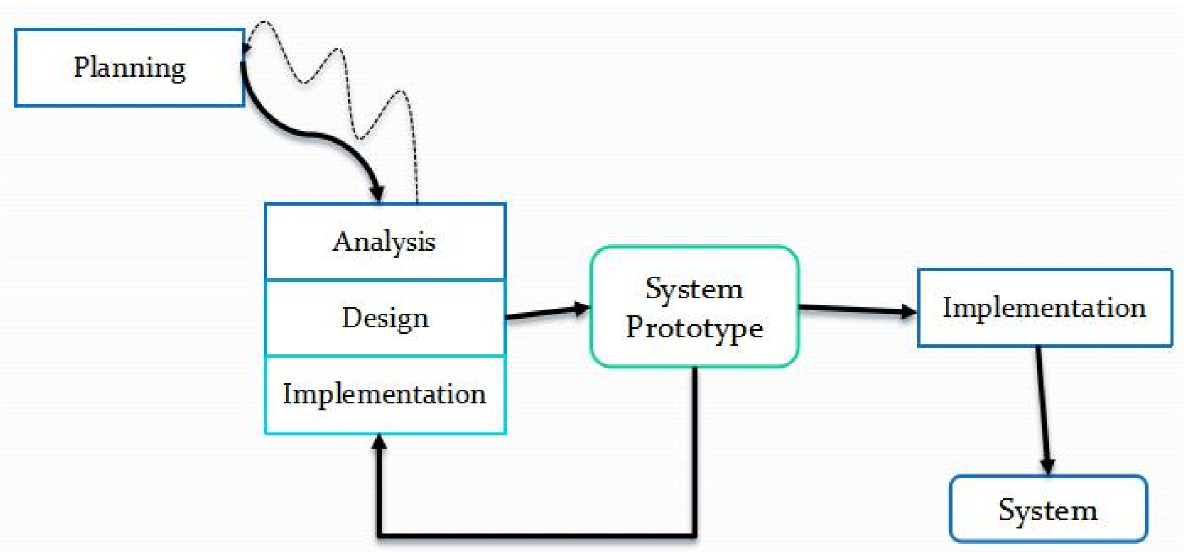


Figure 4.1 Layout of Prototyping Methodology

4.1.2 Hardware

TurtleBot3

The hardware that will be using in this project is, the TurtleBot3 Burger. TurtleBot3 is the most favourite open source robot used in the world especially for education and also in research because it has the most affordable cost and has a smaller size which allow user to carry it to anywhere.

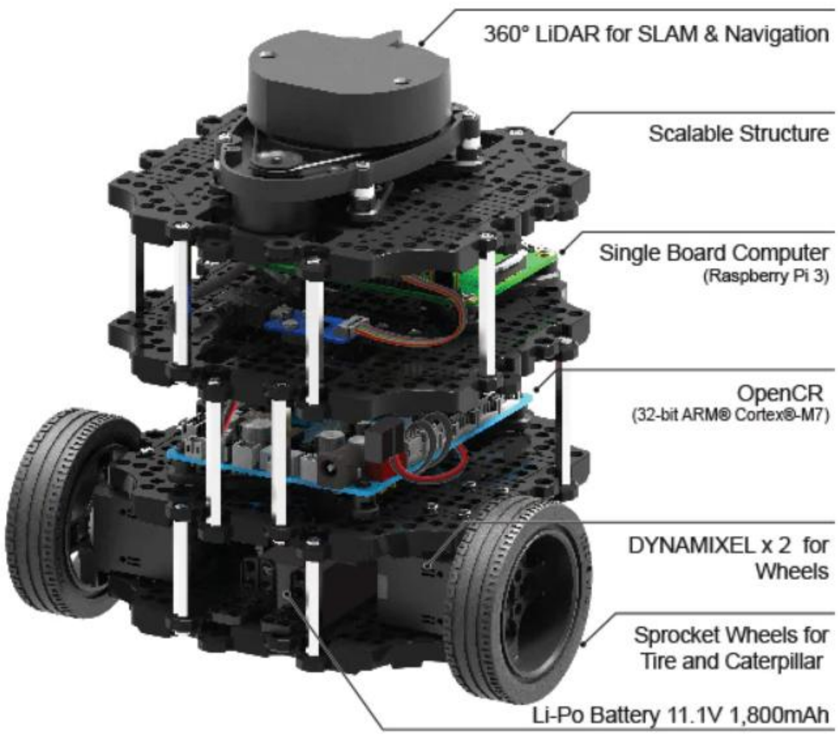


Figure 4.2 TurtleBot3 Burger

TurtleBot3 is a ROS based mobile robot. The core technology of TutleBot3 is SLAM, Navigation and Manipulation which make it more desirable for home service robot. SLAM algorithm can run in TurtleBot to build a map that can move around in an indoor situation. By using laptop, joypad or Android-based smart phone, TurtleBot can be distant controlled. Moreover, TurtleBot can follow the target as they walking around. It also can be used as a mobile manipulator that is able to manipulate an object attaching on a OpenManipulator.

Items	Waffle Pi	Burger
Maximum Translational Velocity	0.26m/s	0.22m/s
Maximum Rotational Velocity	1.82rad/s (104.27 deg/s)	2.84rad/s (162.72 deg/s)
Maximum Payload	30kgs	2.84rad/s (162.72 deg/s)15kgs
Size (L x W x H)	281mm × 306mm × 141mm	138mm × 178mm × 192mm
Weight (+ SBC + Battery + Sensors)	1.8kgs	1kg
Operating Time	About 2hr	About 2hr 30m
Charging Time	About 2hr 30m	About 2hr 30m
DYNAMIXEL	XM430-W210-T	XL430-W250-T
SBC	Raspberry Pi 3	Raspberry Pi 3
Sensor	Raspberry Pi Camera 360°LiDAR 3-Axis gyroscope 3-Axis accelerometer 3-Axis magnetometer	360°LiDAR 3-Axis gyroscope 3-Axis accelerometer 3-Axis magnetometer

Raspberry Pi Model B	Raspberry Pi Camera Module V2	360°LiDAR
		
▪ CPU : Quad Core 1.2GHz ▪ 1GB RAM ▪ Wireless LAN and BLE on board ▪ External Ports	▪ Size : 25 × 24 × 9 mm ▪ Still Resolution : 8Megapixels ▪ Video modes : Max 1080p30 ▪ Sensor : Sony IMX219	▪ Detection distance : 120mm ~ 3,500mm ▪ Angular Range : 360° ▪ Angular Resolution : 1°

Table 4.1 The specification of the TurtleBot

Creative Senz 3D (DS325)

Creative Senz3D provides access to the Creative Senz3D RGB-Depth camera. It also provides information about the distance to the objects and also RGB images which is much similar to the Kinect. The main differences between Kinect and Senz3D is the method used to determine the distance to the objects. As Kinect uses a projection system projecting a dot pattern which is later processed accordingly, Senz3D uses 'time-of-flight' to determine the time taken for light to travel and reflect from objects within its field of view. Thus, edges are sharper than can be seen in Kinect but at a lower resolution. In Senz3D, 320x240 is the maximum depth image size. Besides, unlike Kinect which uses power cable, Senz3D only requires a single USB cable and it is much smaller form factor. Senz3D has a much shorter range than the Kinect which is rated for 0.5ft to 3.25ft. However, it allows registration for a closer objects whereas Kinect has about 2ft blind spot for objects that is too near. Senz3D is very useful as night camera too as the IR image in Senz3D is higher bit depth allowing more scene to be viewed (RoboRealm, 2019).

Figure 4.3 Creative Senz3D

4.1.3 Software

ROS

The Robot Operating System is a written robot software that has a flexible structure which can be executed on Unix-based platforms. It provides the collection of libraries, tools, and conventions which will assist user in structuring the robot application in order to make the task simple enough to create a robust and complicated behaviour of the robot over a wide variety of robotic platforms (Conley 2011).

In the general concepts of ROS system, it begins with a ROS Master that manage communication of the others pieces of ROS nodes between every programs. The software parts of ROS are structured as packages. Every package is the directory that include more than one executables known as nodes. The node communicate with other nodes through both topics and services method (Queva 2013). When topics method is used, the node will direct a message by publishing the message to a specific topic. The other node will subscribe to this topic if it is concerned by the message. Both publisher and subscriber does not realise the existence of each other. Only the Master is responsible for the message transaction (Robotics 2017).

In other hand, the services method are used for modelling the request and reply. As a pair of message structures, one node will act as the request and the other act as the reply. A node provides the service while the other node which can also known as client will sends request and waits for the reply.

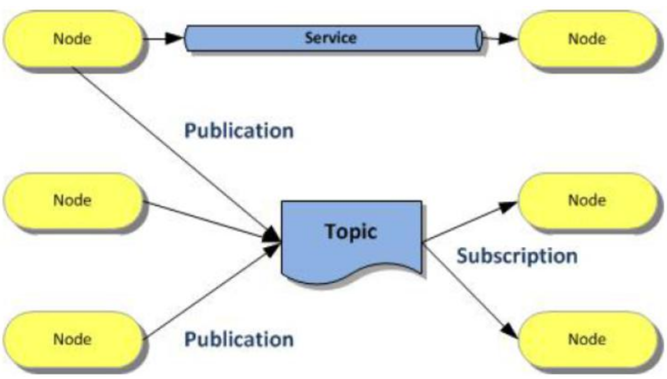


Figure 4.4 ROS communication illustration using topics and services method.

The advantages of using ROS such that all the nodes are running at once. Besides, it can communicate in asynchronous or synchronous way. Besides, URDF files is used to define the robot structure in ROS which describes the 3D model of the robot , all location of the sensor, and also describe how to move each of the non-static parts. A new description file must be created repeatedly. Based on the robot file that uses the exact mobile base, the file was done. Then, all the other sensors will be added into the description file. Next, the part for the implementation of the robot is reduces to the major algorithms as described in the project such as people detection and following, and obstacles avoidance.

Python Programming Language

In this project, Python is used for the programming language. It is a high-level programming language which supports both structured and also object-oriented programming. Besides, through extensions there are many other paradigms that can be supported including the design by contract and logic programming. Python features dynamic late binding that binds names of method and variable in the program execution. Python provides assist for functional programming such as list comprehensions, generator expressions, and dictionaries and sets. Python contains two modules in the standard library which are itertools and functools.

OpenCV

Open Source Computer Vision is a software library of machine learning which mostly focus at computer vision in real-time . The library provides simple language to be used for image processing. OpenCV not only allows common infrastructure for computer vision applications, but it also able to speed up the machine perception used in commercial products. In OpenCV library, there are 2500 algorithm and above that are optimized including both state-of-the-art and classic computer vision comprehensive set and also machine learning algorithms .

OpenCV has functions like faces detection and recognition, object recognition, human behaviours classification in real time videos, camera movement tracking, tracking of moving object, and et al. Besides, OpenCV was designed to be cross-platform and it provides interfaces such as C++, Python, Java and MATLAB and also able to run on Windows, Android, Linux and Mac OS.

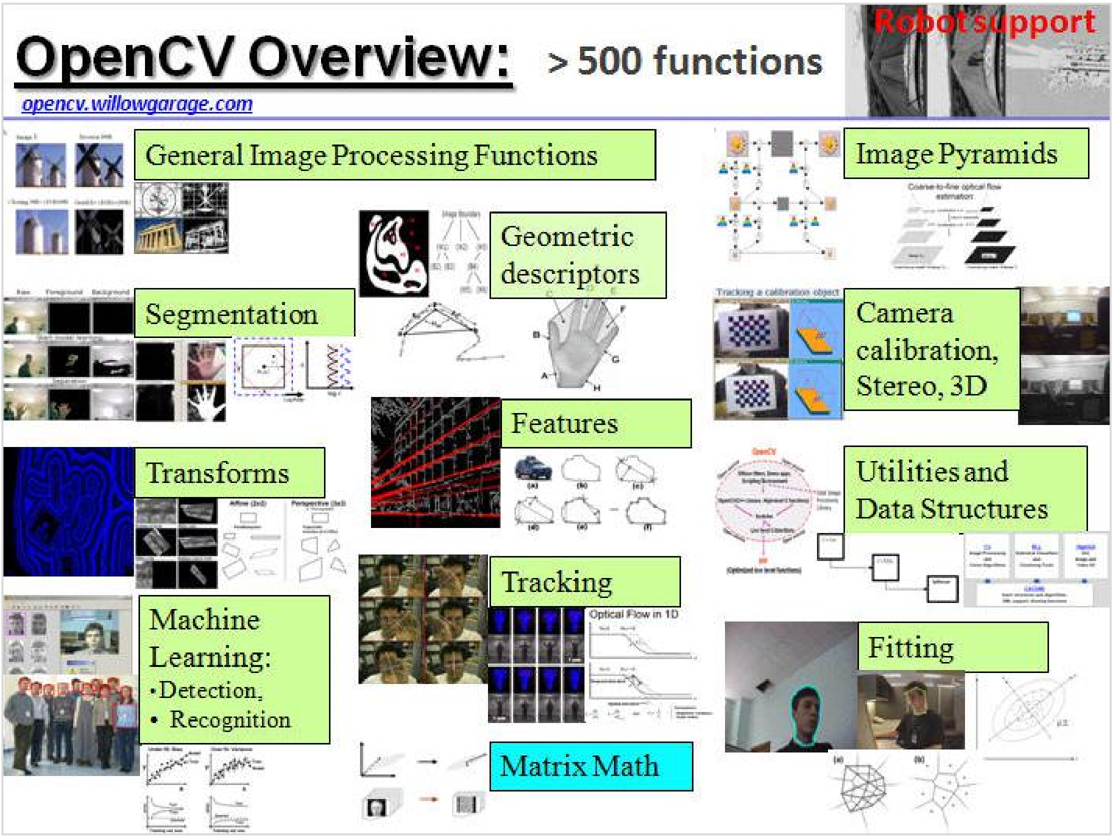


Figure 4.5 The overview function of OpenCV

Figure 4.6 Overview of hardware and software requirements

SIFT in OpenCV

```
import numpy as np
import cv2 as cv

img = cv.imread('home.jpg')
gray= cv.cvtColor(img,cv.COLOR_BGR2GRAY)

sift = cv.xfeatures2d.SIFT_create()
kp = sift.detect(gray,None)

img=cv.drawKeypoints(gray,kp,img)

cv.imwrite('sift_keypoints.jpg',img)
```

```
sift = cv.xfeatures2d.SIFT_create()
kp, des = sift.detectAndCompute(gray,None)
```

Figure 4.7 Example of SIFT in OpenCV Python

First, initialized the feature extractor SIFT such as sift = cv.xfeatures2d.SIFT_create(). to finds the image keypoint. sift.detectAndCompute() method can directly find the keypoints and descriptor in one single step.

Chapter 5 Implementation and Testing

In this project, detection and recognition of marking logo has been achieved using SIFT method as shown in (Figure 5.1). The marking logo can be detected and recognized. A bounding box will be drawn when it is detected as it move around. Besides, the center of the bounding box is also detected and is represented by a circle. This center point is used to capture the distance from the camera in the depth image using Senz3D camera.

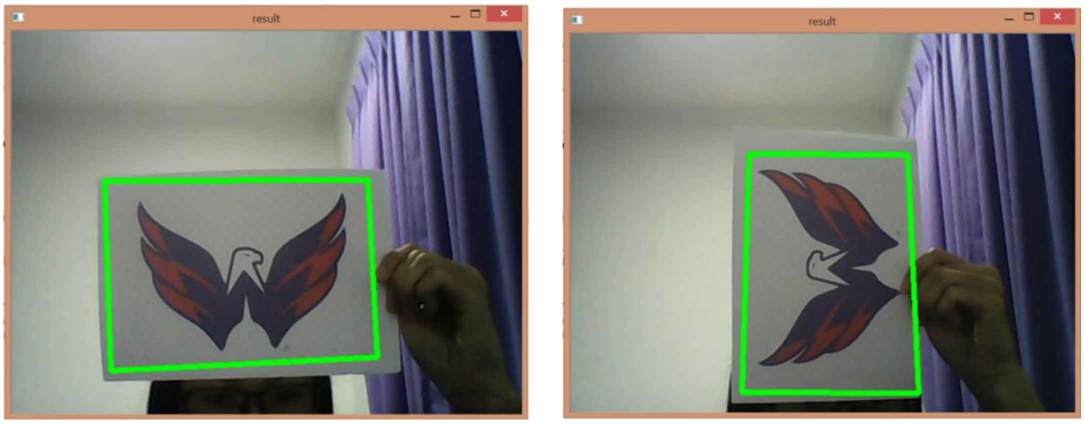


Figure 5.1 Detected marking logo with boundary box

The result shows that SIFT keeps the invariance on scale and rotation change. The bounding box shows that the marking logo is detected and recognized as it move around.

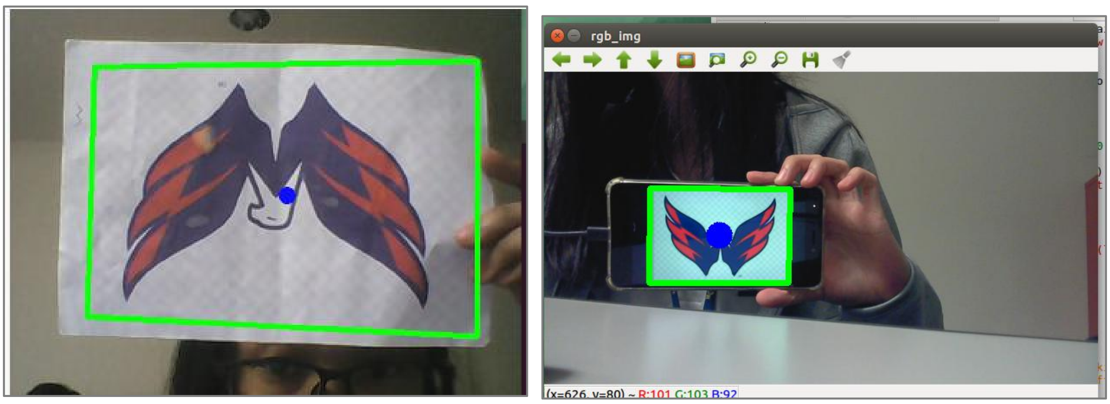


Figure 5.2 Detected target with bounding box and center point

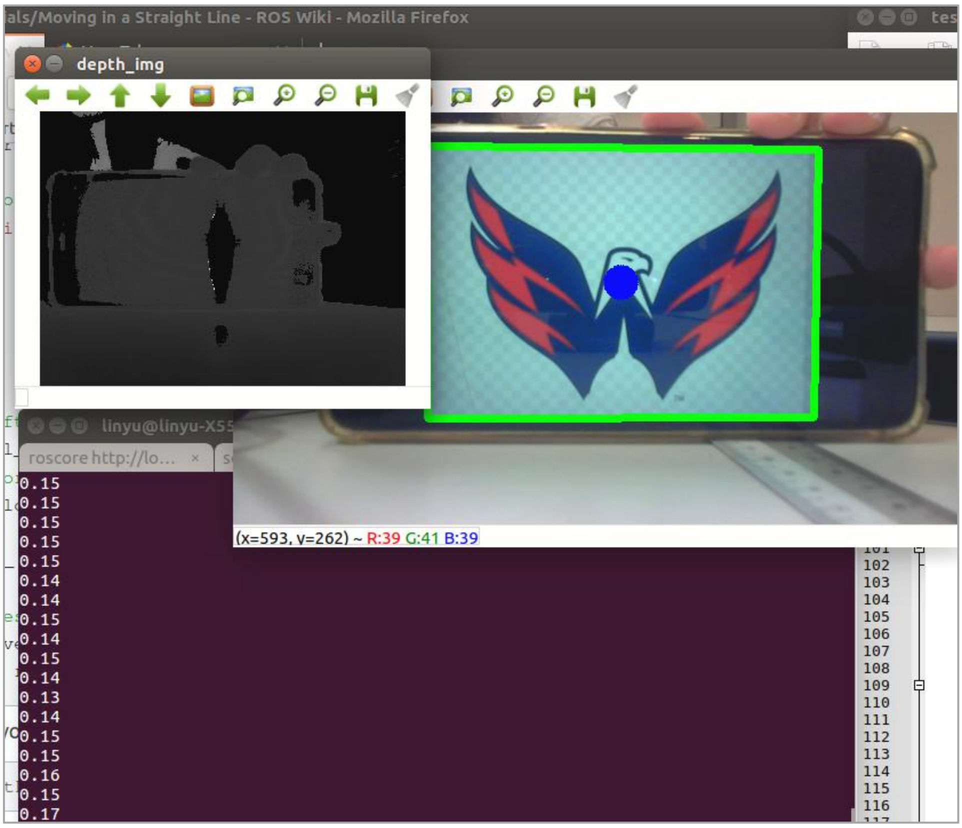


Figure 5.3 The distance of the center point

Figure above shows the detected marking logo and the depth frame. The distance from the camera to the center point is capture and is shown on the terminal which is around 0.15 meter (15centimeter).

Figure 5.4 Depth image

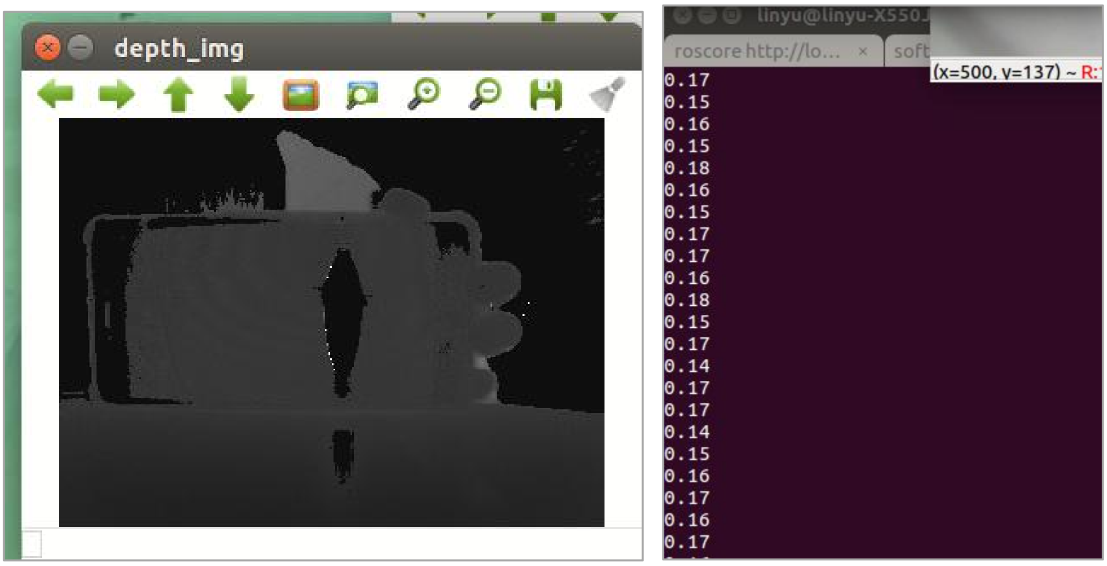


Figure 5.5 Depth image and distance captured by Senz3D

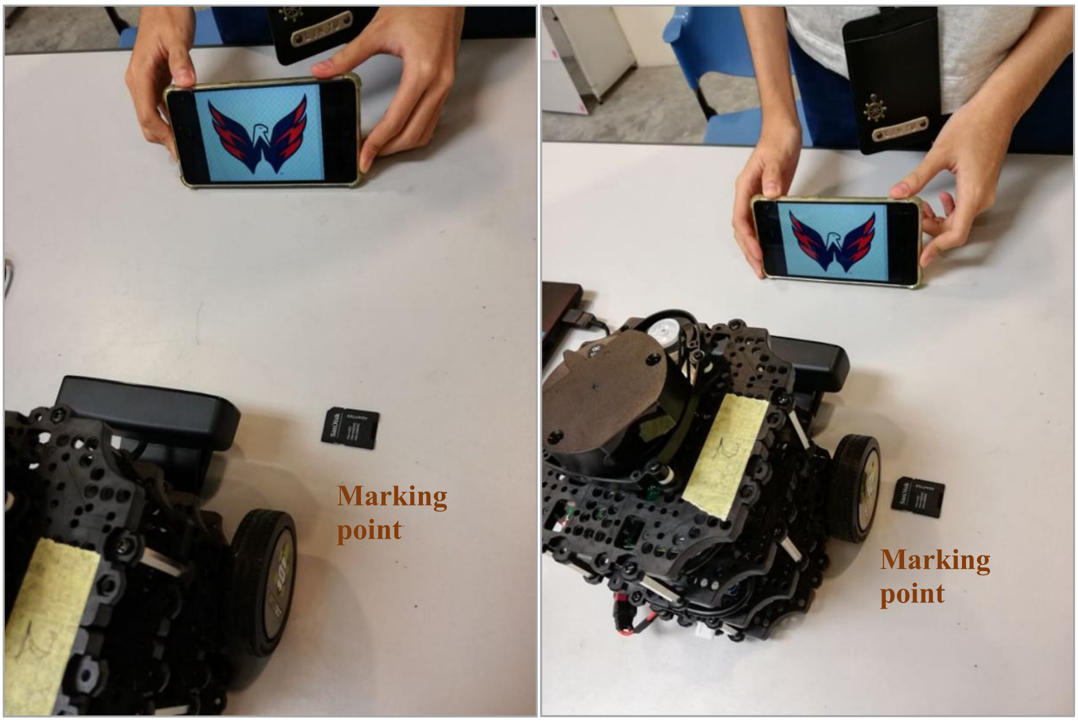


Figure 5.6 Robot move to the detected target

Figure 5.6 shows that the robot is moving as it pass across the marking point heading towards the marking logo.

```
#Load image, convert to jpg
img = cv2.imread('C:/Users/PC/Desktop/markinglogo.png',1)

# percent of original size
scale_percent = 30
width = int(img.shape[1] * scale_percent / 100)
height = int(img.shape[0] * scale_percent / 100)
dim = (width, height)

## resize image
resized = cv2.resize(img, dim, interpolation = cv2.INTER_AREA)

# set threshold
MIN_MATCH_COUNT = 30

# initialized the feature extractor SIFT and feature matcher flann
sift = cv2.xfeatures2d.SIFT_create()
FLANN_INDEX_KDITREE = 0
flannParam = dict(algorithm=FLANN_INDEX_KDITREE, tree=5)
flann = cv2.FlannBasedMatcher(flannParam, {})

# used feature extractor detect feature & store in
# trainKP - list of key points / features coordinates
# trainDesc - list of desc of corresponding key points
trainKP, trainDesc =sift.detectAndCompute(resized, None)
```

Figure 5.7 SIFT code (read in the marking logo and perform feature extraction)

Figure 5.7 shows that the training image which is the marking logo is loaded into the program. Next the loaded image is resized to ensure that the features that is going to extract later is available for better detection. After that, a threshold is set for the minimum matching count. This will be used to compared the matching features between both the trained image and query image to obtain the good matches for detection. Then, uses the feature extractor created earlier to extract and stored the detected features .

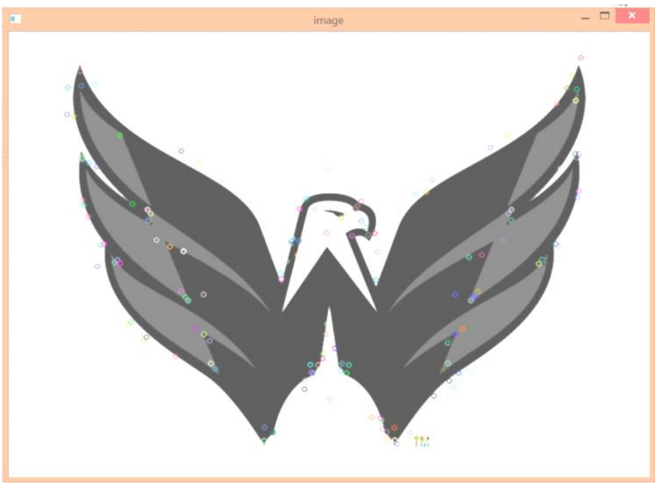


Figure 5.8 Feature extraction

Figure above shows the the features detected by SIFT using sift.detect() function. The small circles which indicates the location of the keypoints is drawn using cv2.drawKeypoint() function.

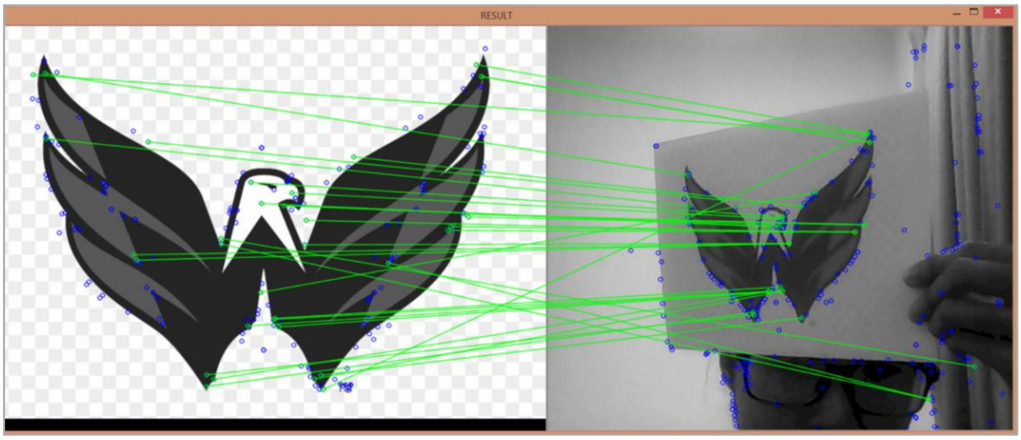


Figure 5.9 Matched features

Figure above draw out the good matches between the original image and the target image from a live video camera.

```
if(len(goodMatch)>MIN_MATCH_COUNT):
    tp=[]
    qp=[]

    for m in goodMatch:
        tp.append(trainKP[m.trainIdx].pt)
        qp.append(queryKP[m.queryIdx].pt)
    tp,qp=np.float32((tp,qp))
    H,status=cv2.findHomography(tp,qp,cv2.RANSAC,3.0)
    h,w, _ = resized.shape
    trainBorder=np.float32([[[0,0],[0,h-1],[w-1,h-1],[w-1,0]]])
    queryBorder=cv2.perspectiveTransform(trainBorder,H)
    #print("width:{0}, height:{1} ".format(w, h))
    #print("box", np.int32(queryBorder))
    # draw borders in query image
    cv2.polylines(cv_image,[np.int32(queryBorder)],True,(0,255,0),5)
    print("A")
    for c in queryBorder:
        M = cv2.moments(c)
        cX = int(M["m10"] / M["m00"])
        cY = int(M["m01"] / M["m00"])
        cv2.circle(cv_image,(cX,cY),7,(255,0,0),15)
        print("Centre: ", cY,cX)
        print(cv_image1[int(cX/4)][int(cY/3)])
```

Figure 5.10 SIFT code (bounding box and center point)

Figure 5.10 shows that when the good matches is detected, a bounding box and its center point will be drawn on the detected target. The bounding box shows that the marking logo is detected and is able to detect along with the moving target in real time. The center point is determined by computing the image moment for the bounding box region. Image moments usually used to characterized an object shape in an image. Some basic statistical properties of the object shape is captured by the moment such as the orientation of an object, the object area, the center (x,y)-coordinates of the object and etc. The coordinate of the centroid will later detected in depth frame for distance measure as the depth stream provide a number of depth frames where pixels in every frame holds the distance information in meters.

```
def display():

    rgb_img = message_filters.Subscriber('/softkinetic_camera/rgb/image_color', Image)
    depth_img = message_filters.Subscriber('/softkinetic_camera/depth/image_raw', Image)

    ts = message_filters.ApproximateTimeSynchronizer([rgb_img, depth_img], 10, 0.5)
    ts.registerCallback(image)
```

Figure 5.11 Subscribe RGB and Depth image

Figure 5.11 shows that in order to use the Senz3D camera to capture both RGB and depth images at the same time, it is necessary to subscribe to both RGB and Depth image with a Synchronizer.

```
while True:
    if robotmove == 1 :
        #while not rospy.is_shutdown():
        t0 = rospy.Time.now().to_sec()s
        current_distance = 0
        print("A")

        while(current_distance < distance):
          print("B")
          velocity_publisher.publish(vel_msg)
          t1=rospy.Time.now().to_sec()
          current_distance= speed*(t1-t0)
      vel_msg.linear.x = 0
      velocity_publisher.publish(vel_msg)
```

Figure 5.12 Robot move

Figure shows the codes to move the turtlebot. The code will keep looping to move the turtlebot in an specified distance. The distance that is used to computed with the current distance is the distance of the center point which obtained from the depth frame.

Chapter 6 Conclusion

The motivation in proposing this human-following robot development project aim to provide living assistance for the elderly as a service robot. As the robot can follow closely behind the elderly, it can perform various activities by implementing other serviceable function such as grocery carrying, aids in eating and medication mechanics, and at the same time accompany the elderly to anywhere and also ensuring their safety since the robot can keep track on them.

The proposed solutions in this project is first by detecting and recognizing the marking logo which will be used to recognize the target. After successfully tracking on the marking logo, distance measurement from the camera to the target object will be implement using depth image data from a 3D camera so that the robot can move behind the target within a range of distance and not lose the sight of the target.

In this project, the machine learning which used to train the designed marking logo for the detection is SIFT. SIFT first extracts the features of the trained image and later compute these extracted features with the query image captured in real time and look for the best matches and perform matching. The detected marking logo will then drawn with a bounding box and the centroid of the bounding box is also identified and represented with a circle. The centroid is used to locate the target at the depth image to capture the distance value from the camera to the detected target. And lastly, robot control is implemented to move the robot to the detected marking logo based on the captured distance.

Discussion

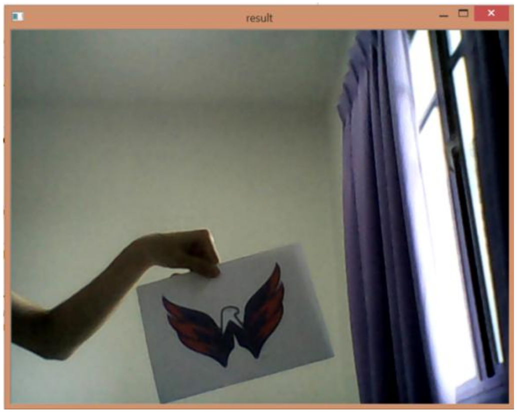


Figure 6.1 Undetected target

In this project, there is a few problem encountered. First, as the distance of the marking logo is getting far away from the web camera, it start to lost the detection. Besides, as Senz3D has a lower resolution where 320x240 is the maximum depth image size. Thus, the RGB frame had to be scale down to obtain more accurate depth value. Besides, the depth image that captured by Senz3D camera has some limitations. From the figure below, a lighter colour such as gray and white is the region that can be detected in depth value while the black colour will be the region that cannot be detected by depth value. Thus, if centroid (x,y) of the bounding box that detect the target overlapped with the black region, the value of the distance will not be obtained as it can not be detected.

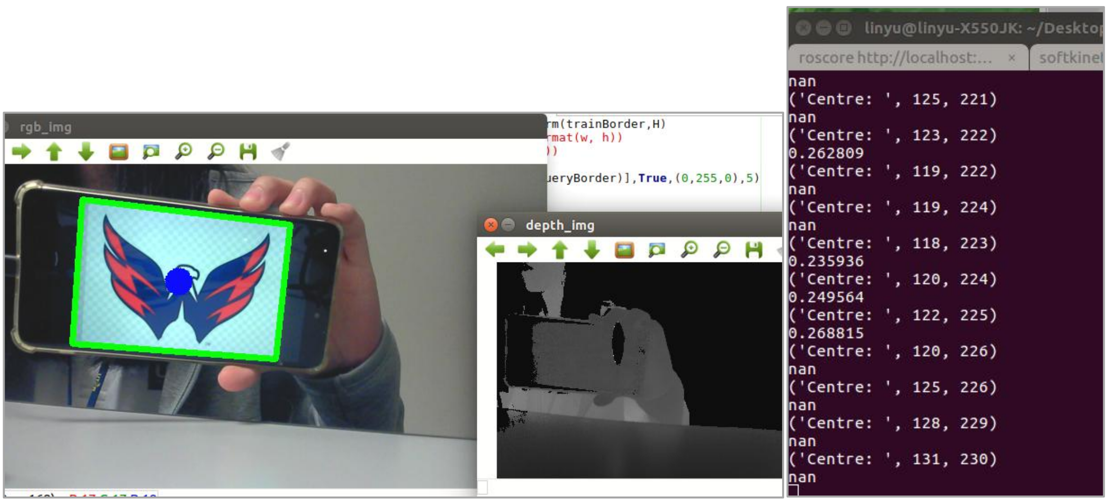


Figure 6.2 Failed to capture distance

Figure 6.2 shows that sometimes the distance cannot be detected as the center point of the bounding box has overlapped with the black region that cannot be detected by depth camera.

Future Work

There are various experiments and tests have been left for future work which can be carry out to improve the project. In this project, instead using a marking logo to detect the target person, a huge datasets and a better algorithm such as convolutional neural network can be used to trained to detect the target person based on the height, weight, body shape, shirts of the day, and many more other features. Here, convolutional neural network is a method in machine learning modelled after the structure of the brain where the neurons will learn to convert input signals (such as the image of a person) into respective output signals (e.g. with a labelled “target”) and forms the automated recognition basis. By doing this, the robot not only can be used to detect the target more accurately and it still can recognised on the same target person even when there are other person who wears the same shirt (as the robot can use other features to make comparison and detect the correct target). Moreover, a better depth camera can be used to capture both RGB and depth images from real time as Creative Senz3D has its own limitations where it only can capture the depth data within 1 meter. For examples, Kinect camera can be tried in future work as it has a higher depth resolution of 640x480 as compared to Senz3D which only have 320x240. In addition, the performance of moving the robot can be improved to allow the robot to be able to turn around to look for the target by turning right and left smoothly. Besides, obstacles avoidance can also have a better implementation such that the robot learn to avoid the obstacles itself while following behind the target.

Bibliography

Bianco, S., Buzzelli, M., Mazzini, D. and Schettini, R., 2015, September. Logo recognition using CNN features. In International Conference on Image Analysis and Processing (pp. 438-448). Springer, Cham.

Chen, B.X., Sahdev, R. and Tsotsos, J.K., 2017, July. Integrating stereo vision with a CNN tracker for a person-following robot. In *International Conference on Computer Vision Systems* (pp. 300-313). Springer, Cham.

Conley, K., 2011. ROS introduction. Internet: http://www.ros. org/wiki/ROS/Introduction.

Dalal, N., & Triggs, B. 2005, June, Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (Vol. 1, pp. 886-893). IEEE.

Hurst, D., 2018. Japan lays groundwork for boom in robot carers. The Guardian, 6.

Haggag, H., Hossny, M., Filippidis, D., Creighton, D., Nahavandi, S. and Puri, V., 2013, December. Measuring depth accuracy in RGBD cameras. In Signal Processing and Communication Systems (ICSPCS), 2013 7th International Conference on (pp. 1-7). IEEE.

Lison, P., 2015. An introduction to machine learning.

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60(2), pp.91-110.

Ministry of Economics and Trade (METI). (2010). 2035nen ni muketa robotto sangyou no shourai shijou yosoku (Market forecast of robot industry in 2035)

Mosavi, A. and Varkonyi, A., 2017. Learning in robotics. Int. J. Comput. Appl, 157(1), pp.0975-8887.

Munaro, M., Basso, F. and Menegatti, E., 2012, October. Tracking people within groups with RGB-D data. In Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on (pp. 2101-2107). IEEE.

OpenCV. 2018, Introduction to SIFT (Scale-Invariant Feature Transform), Available from <https://docs.opencv.org/3.4.3/da/df5/tutorial_py_sift_intro.html> [28 March 2019]

Queva, C., 2013. Human following behaviour for an autonomous mobile robot.

Robert, C., & Ryan, W. (2011). How Does The Xbox Kinect Work. Available from:<https://www.jameco.com/jameco/workshop/howitworks/xboxkinect.html>

Robotics, C., 2017. ROS 101: Intro To The Robot Operating System| Robohub. Robohub. org. Np.

RoboRealm. 2019, Creative Senz3D, Available from <http://www.roborealm.com/help/Creative_Senz3D.php> [12 August 2019]

Schlegel, C., Illmann, J., Jaberg, H., Schuster, M. and Wörz, R., 1998, August. Vision Based Person Tracking with a Mobile Robot. In *BMVC* (pp. 1-10).

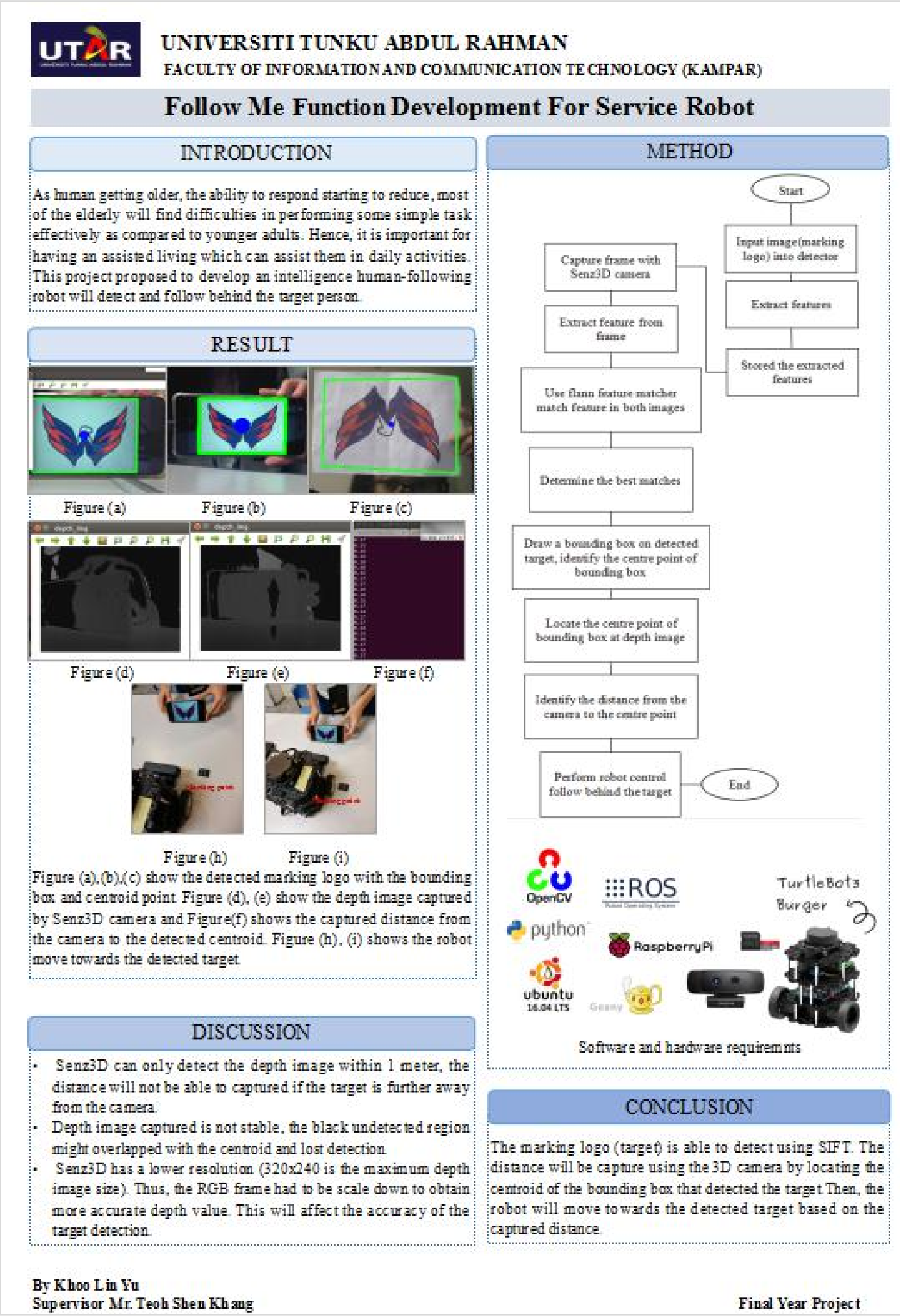
Spinello, L. and Arras, K.O., 2011, September. People detection in RGB-D data. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on* (pp. 3838-3843). IEEE.

Takemura, H., Ito, K. and Mizoguchi, H., 2007, December. Person following mobile robot under varying illumination based on distance and colour information. In *Robotics and Biometrics, 2007. ROBIO 2007. IEEE International Conference on* (pp. 1500-1505). IEEE.

Tarokh, M. and Ferrari, P., 2003. Case study: Robotic person following using fuzzy control and image segmentation. *Journal of Robotic Systems*, *20*(9), pp.557-568.

urgen Kreuziger, J., 1992. APPLICATION OF MACHINE LEARNING TO ROBOTICS {AN ANALYSIS. In Proc. of the 2nd Int. Conf. on Automation, Robotics and Computer Vision, Singapore, pages RO {15.4. 1 {RO {15.4. 5.

A.1 Poster



UNIVERSITI TUNKU ABDUL RAHMAN
FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY (KAMPAR)

Follow Me Function Development For Service Robot

INTRODUCTION

As human getting older, the ability to respond starting to reduce, most of the elderly will find difficulties in performing some simple task effectively as compared to younger adults. Hence, it is important for having an assisted living which can assist them in daily activities. This project proposed to develop an intelligence human-following robot will detect and follow behind the target person.

RESULT

Figure (a) Figure (b) Figure (c)

Figure (d) Figure (e) Figure (f)

Figure (h) Figure (i)

Figure (a),(b),(c) show the detected marking logo with the bounding box and centroid point. Figure (d), (e) show the depth image captured by Senz3D camera and Figure(f) shows the captured distance from the camera to the detected centroid. Figure (h), (i) shows the robot move towards the detected target.

DISCUSSION

- Senz3D can only detect the depth image within 1 meter, the distance will not be able to captured if the target is further away from the camera.
- Depth image captured is not stable, the black undetected region might overlapped with the centroid and lost detection.
- Senz3D has a lower resolution (320x240 is the maximum depth image size). Thus, the RGB frame had to be scale down to obtain more accurate depth value. This will affect the accuracy of the target detection.

METHOD

Software and hardware requirements

CONCLUSION

The marking logo (target) is able to detect using SIFT. The distance will be capture using the 3D camera by locating the centroid of the bounding box that detected the target Then, the robot will move towards the detected target based on the captured distance.

By Khoo Lin Yu
Supervisor Mr. Teoh Shen Khang

Final Year Project

B.1 Plagiarism Check Result

FYP2

ORIGINALITY REPORT

4%	0%	1%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to University of Southern California Student Paper	<1%
2	Submitted to Universiti Teknologi MARA Student Paper	<1%
3	Submitted to Rochester Institute of Technology Student Paper	<1%
4	Submitted to KYUNG HEE UNIVERSITY Student Paper	<1%
5	Submitted to Universiti Tunku Abdul Rahman Student Paper	<1%
6	Submitted to University of Sheffield Student Paper	<1%
7	Submitted to Universiti Tenaga Nasional Student Paper	<1%
8	Luciano Spinello, Kai O. Arras. "People detection in RGB-D data", 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011 Publication	<1%

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	
ID Number(s)	
Programme / Course	
Title of Final Year Project	

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index:____________% **Similarity by source** Internet Sources :__________________% Publications : __________________% Student Papers : __________________%	
Number of individual sources listed of more than 3% similarity: _________	
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

______________________________ ______________________________

Signature of Supervisor Signature of Co-Supervisor

Name: ____________________________ Name: ____________________________

Date: _____________________________ Date: _____________________________

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	
Student Name	
Supervisor Name	

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Front Cover
	Signed Report Status Declaration Form
	Title Page
	Signed form of the Declaration of Originality
	Acknowledgement
	Abstract
	Table of Contents
	List of Figures (if applicable)
	List of Tables (if applicable)
	List of Symbols (if applicable)
	List of Abbreviations (if applicable)
	Chapters / Content
	Bibliography (or References)
	All references in bibliography are cited in the thesis, especially in the chapter of literature review
	Appendices (if applicable)
	Poster
	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report. ______________________ (Signature of Student) Date:	Supervisor verification. Report with incorrect format can get 5 mark (1 grade) reduction. ______________________ (Signature of Supervisor) Date: