Visvesvaraya Technological University, Belagavi.

PROJECT REPORT

on

"Robot Movement Automation with Computer Vision"

Project Report submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Engineering

in

Electronics and Communication Engineering

For the academic year 2019-20

Submitted by

USN	Name
1CR16EC004	Adarsh U
1CR16EC024	B Karthik
1CR16EC033	Dabir Hasan Rizvi
1CR16EC037	Deepak Kamath K

Under the guidance of

Dr. R Elumalai

Head of the Department

Department of ECE

CMRIT, Bengaluru

CELEBRATING 25 YEARS

CMR INSTITUTE OF TECHNOLOGY

Department of Electronics and Communication Engineering

CMR Institute of Technology, Bengaluru – 560 037

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CELEBRATING 25 YEARS
CMR INSTITUTE OF TECHNOLOGY

CERTIFICATE

This is to Certify that the dissertation work **"Robot Movement Automation with Computer Vision"** carried out by Students Adarsh U, B Karthik, Dabir Hasan Rizvi, Deepak Kamath K, USNs: 1CR16EC004, 1CR16EC024, 1CR16EC033, 1CR16EC037, bonafide students of **CMRIT** in partial fulfillment for the award of **Bachelor of Engineering** in **Electronics and Communication Engineering** of the **Visvesvaraya Technological University, Belagavi,** during the academic year **2019-20**. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said degree.

Signature of Guide	Signature of HOD	Signature of Principal
Dr. R. Elumalai Head of the Department, Dept. of ECE., CMRIT, Bengaluru.	Dr. R. Elumalai Head of the Department, Dept. of ECE., CMRIT, Bengaluru.	Dr. Sanjay Jain Principal, CMRIT, Bengaluru.

External Viva

Name of Examiners	Signature & date
1.	
2	

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose consistent guidance and encouragement crowned our efforts with success.

We consider it our privilege to express the gratitude to all those who guided in the completion of the project.

We express our gratitude to Principal, **Dr. Sanjay Jain,** for having provided us the golden opportunity to undertake this project work in their esteemed organization.

We sincerely thank Dr. R. Elumalai**,** HOD**,** Department of Electronics and Communication Engineering, CMR Institute of Technology for the immense support given to us.

We express our gratitude to our project guide Dr. R. Elumalai**,** HOD, Department of Electronics, for his support, guidance and suggestions throughout the project work.

Last but not the least, heartful thanks to our parents and friends for their support.

Above all, We thank the Lord Almighty for His grace on us to succeed in this endeavor.

Table of Contents

List of Figures

ABSTRACT

In recent times, we have seen an immense increase in the amount of research, development and investment put in the field of Autonomous Vehicle Technology or in common terms - Self-Driving Vehicles. While modern autonomous vehicles commonly use Machine and Deep Learning using data from sensors to understand the environment and move autonomously, this project demonstrates the use of a digital image processing algorithm(SIFT) to enable a robot to track specific objects in an environment and follow them. While the title of the paper shows more emphasis on the movement of the robot, it is actually the efficient implementation of the image tracking algorithm using computer vision that is the most important outcome of this project. This paper discuses two use cases in detail and also concludes by specifying future potential use cases.

Chapter 1

INTRODUCTION

An autonomous vehicle is capable of sensing its environment and operating without human involvement. Autonomous vehicles rely on sensors, actuators, complex algorithms, machine learning systems, and powerful processors to execute software. There are several reasons as to why Autonomous Vehicles are required and how they can help revolutionize the modern world. Studies show that around 90% of all vehicular accidents are caused by human error leading to a huge number of unnecessary deaths worldwide. Autonomous Vehicles also bring about automation and comfort to a commuter. Majority of current research and development is focused on self-driving cars and this brings forth the need for industry specific robots that work autonomously or with very little human interference/commands.

Computer Vision is another field of research that is being used quite extensively to solve problems in several fields. Computer Vision is a field of digital image processing that focuses on replicating parts of the complexity of the human vision system and enabling computers to identify and process objects in images and videos in the same way that humans do using image information from cameras and other devices. Computer Vision enables self-driving cars to make sense of their surroundings. Cameras capture video from different angles around the car and feed it to computer vision software, which then processes the images in real-time to find the extremities of roads, read traffic signs, detect other cars, objects and pedestrians.

This project aims at the development of one such autonomous vehicle with interesting use cases mentioned later in the report. The exploitation of very efficient object feature tracking image processing algorithms, use of modern technologies like Augmented Reality in the user-interface and IoT based communication in the working environment makes this paper an impressive state-of-the-art contribution towards science, technology and engineering.

Chapter 2

LITERATURE REVIEW

In order to understand the research and development in autonomous driving and computer vision in the last years, it is important to conduct a literature survey thereby understanding the background of this project and the potential of it.

2.1 A brief history of Autonomous Vehicles

A paper by the name “Algorithms applied in Autonomous Vehicle Systems” [1] discusses the development paths and key algorithms of autonomous vehicles. The first autonomous vehicle concept was realized in 1995 when a vehicle was able to move for about a 1000 km without human assistance using a road network map, landmark of static objects on a map and dimensions of the road/ path. Fast-forwarding to the 21st century, Tesla launched a not fully autonomous car in the year 2014. Research is still on with a intense competition from several manufacturers such as Ford.

2.2 Relevant work

This section covers an example of an implemented digital image processing based robot and some algorithms currently in use for image analysis, path planning, information storage and decision systems.

In the paper by the name "Line Following Robot Using Image Processing" [2], a robot was developed that had the functionality to follow a line detected by a camera mounted on it. A camera is used to obtain image of the track and then converted into a bitmap image. Least Square Method is used to follow the predefined path. Turns were calculated using the slopes of the line captured within the scope of the camera. Thus, by using image processing the line following robot is guided along the desired path.

Some examples of sensor based image processing algorithms are - Vanishing Point method for finding a road in the desert which consists of calculating dominant orientations in image segments and estimating the position of point of convergence between two parallel lines because two parallel lines that form a lane(road) meet at one point in a 2D image captured by a camera, modern methods of analysis using Machine Learning to detect and classify objects captured by a camera, RADAR based estimation of speed, orientation and range of objects.

Specifically talking about path finding, algorithms such as A* and Djikstra's algorithm have been used by autonomous vehicles to find paths based on travel time, distance and fuel consumption.

Autonomous Vehicles combine various techniques of perceiving the environment using sensors, analytical algorithms as well as AI or ML which generally produces better results. But replacing all image processing algorithms with ML does not seem to be too viable as it requires a lot of data and storage infrastructure. The choice of algorithms depend on-computational power, type of sensor,quality of sensors and level of automation that the vehicle aims to achieve. Several researchers have implemented object detection using algorithms known as Speeded-Up Robust Features(SURF) and Scale-Invariant Feature Transform(SIFT) which are specifically used to identify edges and features in an image. In this project, SIFT was used to serve our use case efficiently.

Chapter 3

OBJECTIVE AND MOTIVATION

The development of the robot has been motivated by identifying two use cases which will be described in this chapter. The following sections will cover the details about the functionality of the robot and the work-flow.

3.1 Objective

The basic objective of this project is to develop a robot that efficiently follows any object(image) as selected by the user in an environment as long as the robot and the object are in the range of the same network connection(Wi-Fi).

The system proposed must be able to quickly identify feature points of the selected object that the robot has to follow.

Besides identifying feature points, the robot has to be able to follow the object even if the objects orientation and scale changes.

3.2 Motivation

An environment such as a warehouse or a factory consists of a lots of tasks that involve humans driving industry specific vehicles inside the environment such as forklifts, hand trucks, order pickers etc. These vehicles have to be driven by humans to specific locations such as racks, storage facilities within the environment which may also include hazardous environments potentially risking the life of the operators. This is what the project aims to eliminate. Eliminating the requirement of a driver also brings about comfort if the same work can be done remotely by passing on the task to an autonomous vehicle. Comfort and safety is also the main motivation behind autonomous cars on the road or open environments and this project also serves as an added functionality to existing autonomous cars.

Chapter 4

METHODOLOGY

The basic working principle of this project is checking if the robot is within certain bounds of the object and if it is not, the robot receives a message to move into the bounds as long as the robot and the object are in the same network range.

4.1 Work-flow of the system

This section describes the step by step work-flow of what the project aims at achieving within an enclosed environment like a warehouse or factory as well as an open environment such as on the road.

4.1.1 Closed environment work-flow

With respect to closed environment such as a warehouse, an object may refer to a specific rack in the warehouse that the vehicle must approach in order to perform any task like picking or dropping which are tasks that can also be automated but are not in the scope of this particular project. In this particular use case, cameras are not mounted on the robot but fixed in front of the location that the robot has to move to.

The work-flow is as follows:

- A machine operator is given the responsibility of dropping a certain item at a specific location for example a rack which will have a high detailed image attached to it for feature points identification. A feature point refers to details in the image that make it very unique and easy identifiable. This image on the rack can also be referred to as a “Target Image” .
- This rack and the target image on it will be in the scope of a camera in front of it.
- The target image will be selected by the machine operator on a hand held device with a screen that shows the feed or output that is being captured by the camera. In this project, a mobile phone will be used to serve the purpose of both, the camera as well as the hand held device used to choose the target image.
- The algorithm will identify all the feature points on that particular selected image using an image processing algorithm which will be explained in the later chapters.
- Communicating through Wi-Fi, the robot will start its movement towards the selected target image and stop when it reaches its bounds.

4.1.2 Open environment work-flow

In this use case, the camera will be mounted on a vehicle like a car in an open-environment(a road). The target image can refer to any object on the road that the vehicle has to follow. It can be another moving vehicle, a stationary vehicle, road signs along the path or even as simple as a tree on the sidewalk. This use case with the proposed system is unique and efficient as the image processing algorithm used is extremely fast in terms of extracting feature points irrespective of the orientation of the target image/object. This makes it a very ideal system to follow another moving vehicle.

The work-flow is as follows:

- The person sitting in the car uses his hand held device(mobile phone) connected to the camera to select an object/image in front of it.
- The feature points of the object are extracted using the algorithm and the vehicle starts moving towards that object and stops when it is in the objects bounds.

In both these cases, the robot or the vehicle will stop moving if the view of the camera is blocked, camera is moved to focus on something else or if the feature points are not identifiable due to any other reason. The benefit of identifying several feature points on an object is that some feature points will always look the same if the orientation of the object is changed and the movement of the vehicle will not be disturbed in such a case. A detailed understanding of target images, feature points and the algorithm behind their identification will be explained in the following chapters.

The work-flow of the system is shown below.

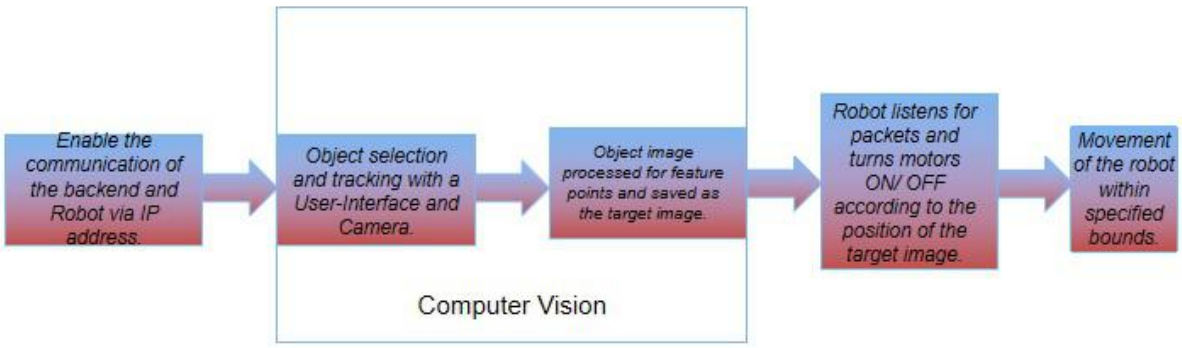

Figure 1 : Work-flow

Chapter 5

TECHNICAL DETAILS AND IMPLEMENTATION

Tracking an environment and depicting virtual images on it by identifying features of the environment and continuously tracking these features is the principle behind a technology like Augmented Reality(AR). In this project we have used the same algorithm used by AR applications to detect and identify features of objects in the real world. Since the motivation and work-flow of the proposed system has been established in the previous chapters, the deep understanding of the architecture, algorithms and other technical details is discussed in the sections of this chapter.

5.1 Image Tracking

A specific challenge in tracking moving objects as in the case of tracking moving vehicles is the fact that when the object is selected, the algorithm takes a while to identify its feature points and by the time it actually starts tracking it, the object might have moved further away or changed its orientation by taking a turn. This is the reason behind using a scale independent edge detecting algorithm called Scale-Invariant Feature Transform(SIFT). Several rotation-invariant algorithms like Harris are not good enough because they are not scale-invariant.

5.1.1 Scale-invariance

A simple explanation of scale invariance is given by Tony Lindeberg's "Scale-space Theory: A basic tool for analyzing structures at different scales" [3]- An inherent property of objects in the world is that they only exist as meaningful entities over certain ranges of scale. A simple example is the concept of a branch of a tree, which makes sense only at a scale from, say, a few centimeters to at most a few meters. It is meaningless to discuss the tree concept at the nanometer or the kilometer level. At those scales it is more relevant to talk about the molecules that form the leaves of the tree, or the forest in which the tree grows. Similarly, it is only meaningful to talk about a cloud over a certain range of coarse scales. At finer scales it is more appropriate to consider the individual droplets, which in turn consist of water molecules, which consist of atoms, which consist of protons and electrons etc.

The scale of an object in an image is roughly its diameter in the image. It is denoted by σ, which is measured in pixels. With respect to this, scale-invariance enables us to identify an object at different scales.

A corner may not be a corner if the image is scaled. What this means is that a corner in a small image within a small window is flat when it is zoomed in the same window. This is clearly depicted in the image below.

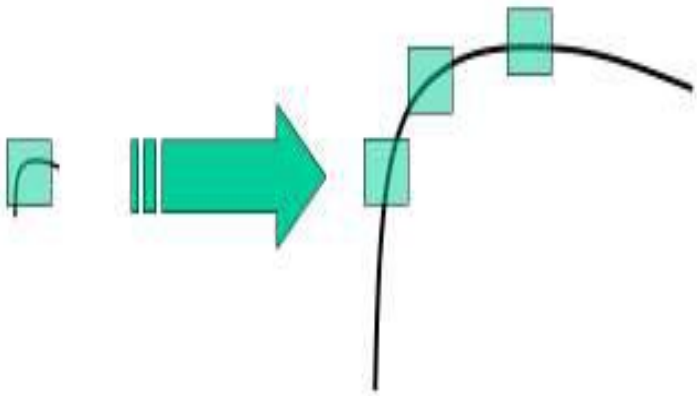

Figure 2 : A depiction of edge detection for the same images of different scales.

In simple words, the same window cannot be used to identify keypoints with different scales. When zoomed, an image or edge requires more windows as show above. This is the reason an algorithm that is only rotation-invariant(Harris) cannot be used for the system proposed by this paper. Hence, the SIFT algorithm is a suitable scale-invariant algorithm that can be used.

5.2 Scale-Invariant Feature Transform(SIFT) Algorithm

The scale-invariant feature transform (SIFT) is a feature detection algorithm in computer vision to detect and describe local features in images. This algorithm was proposed by David G. Lowe in his paper titled “Distinctive Image Features from Scale-Invariant Keypoints” [4] where he described the algorithm in steps as follows:

1. Scale-space Extrema Detection

2. Keypoint Localization

3. Orientation Assignment

4. Keypoint Descriptor

5. Keypoint Matching

The subsections will cover each of these steps in detail.

5.2.1 Scale-space extrema detection

From Figure 2, it is obvious that we can't use the same window to detect keypoints with different scale. There is no problem with a small corner. But to detect larger corners we need larger windows. For this, scale-space filtering is used. In it, Laplacian of Gaussian(LoG) is found for the image with various σ values. LoG acts as a blob detector which detects blobs in various sizes due to change in σ. In short, σ acts as a scaling parameter. For eg, in the above image, Gaussian kernel with low σ gives high value for small corner while Gaussian kernel with high σ fits well for larger corner. So, we can find the local maxima across the scale and space which gives us a list of (x, y, σ) values which means there is a potential keypoint at (x,y) at σ scale.

LoG is a little costly, so SIFT algorithm uses Difference of Gaussians(DoG) which is an approximation of LoG. Difference of Gaussian is obtained as the Difference of Gaussian(DoG) blurring of an image with two different σ, let it be σ and $k\sigma$. This process is done for different octaves of the image in Gaussian Pyramid shown in the figure below.

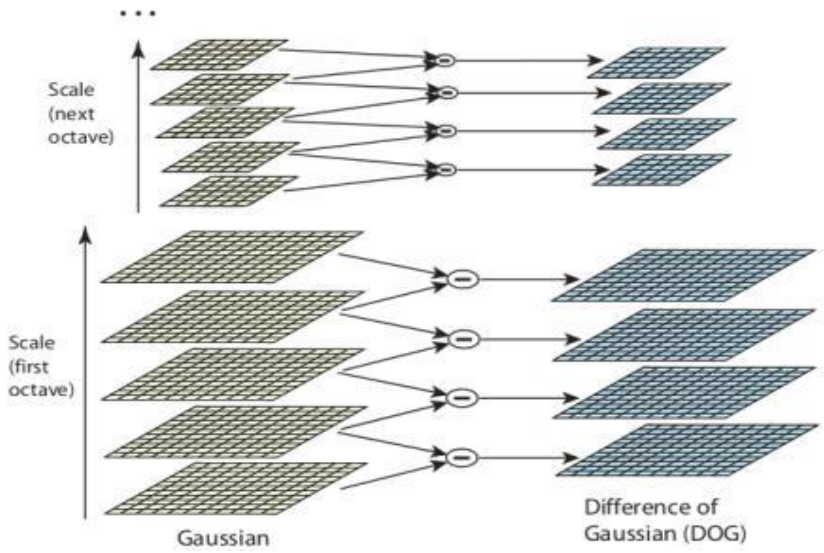

Figure 3 : Difference of Gaussian(DoG)

Once this DoG are found, images are searched for local extrema over scale and space. For eg, one pixel in an image is compared with its 8 neighbours as well as 9 pixels in next scale and 9 pixels in previous scales. If it is a local extrema, it is a potential keypoint.

It basically means that keypoint is best represented in that scale. It is shown in Figure 4.

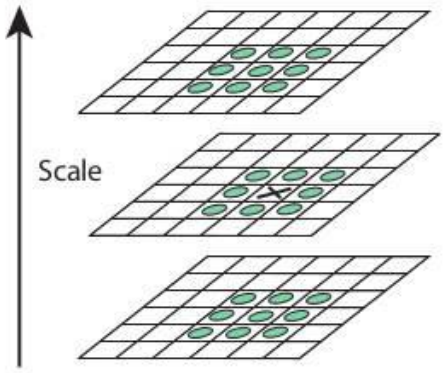

Figure 4 : Finding the local extrema over scale and space

Regarding different parameters, the paper gives some empirical data which can be summarized as, number of octaves=4, number of scale levels=5, initial $\sigma = 1.6, k = \sqrt{2}$ etc as optimal values.

5.2.2 Keypoint Localization

Once potential keypoints locations are found, they have to be refined to get more accurate results. The Taylor series expansion of scale space is used to get a more accurate location of extrema, and if the intensity at this extrema is less than a threshold value (0.03 as per the paper), it is rejected. DoG has higher response for edges, so edges also need to be removed. For this, a concept similar to Harris corner detector is used. A 2x2 Hessian matrix (H) is used to compute the pricipal curvature. In the Harris corner detector that for edges, one eigen value is larger than the other. If this ratio is greater than a threshold(10 according to the paper), the keypoint is discarded. So it eliminates any low-contrast keypoints and edge keypoints and what remains is strong interest points.

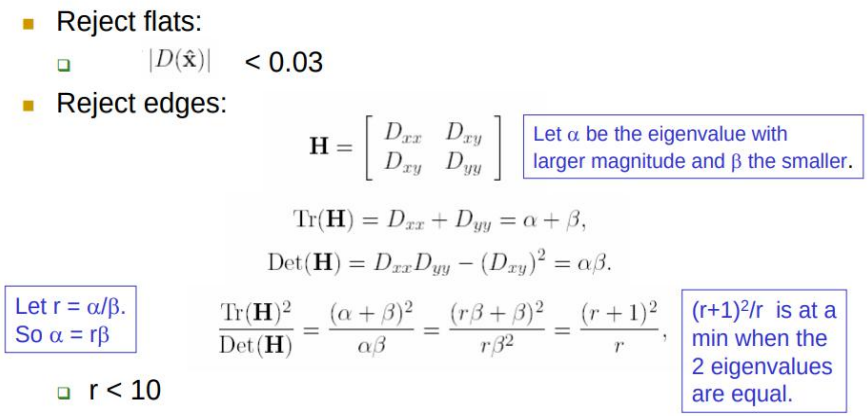

Figure 5 : Rejection of keypoints based on a threshold

5.2.3 Orientation Assignment

Now an orientation is assigned to each keypoint to achieve invariance to image rotation. A neighborhood is taken around the keypoint location depending on the scale, and the gradient magnitude and direction is calculated in that region.

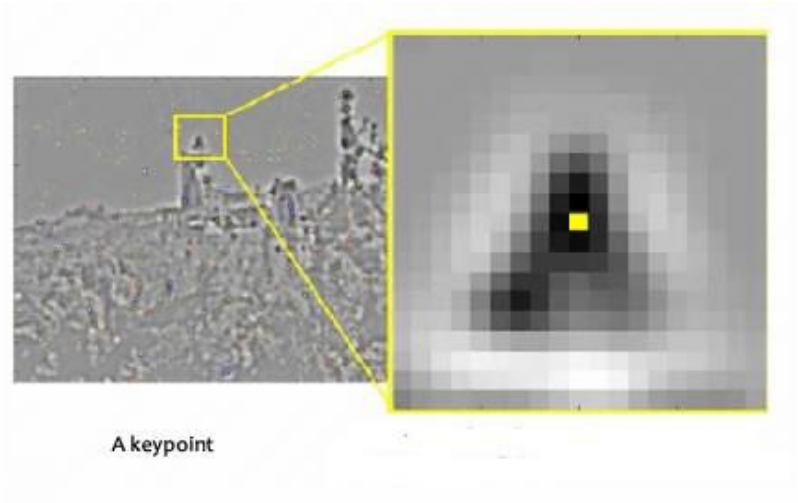

Figure 6 : A sample keypoint and its neighborhood pixels

An orientation histogram with 36 bins covering 360 degrees is created as shown in the figure. It is weighted by gradient magnitude and Gaussian-weighted circular window with σ equal to 1.5 times the scale of keypoint. The highest peak in the histogram is taken and any peak above 80% of it is also considered to calculate the orientation. It creates keypoints with same location and scale, but different directions. It contributes to the stability of matching.

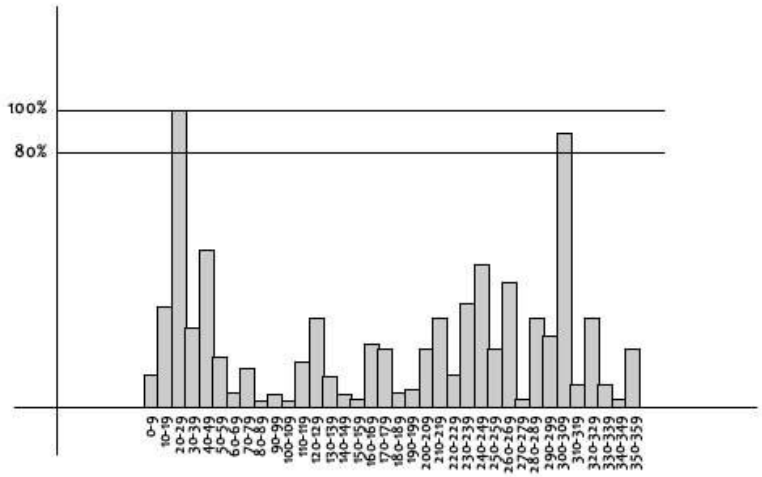

Figure 7 : Orientation Histogram

5.2.4 Keypoint Descriptor

Now the keypoint descriptor is created. A 16x16 neighbourhood around the keypoint is taken. It is divided into 16 sub-blocks of 4x4 size as shown in Figure 8.

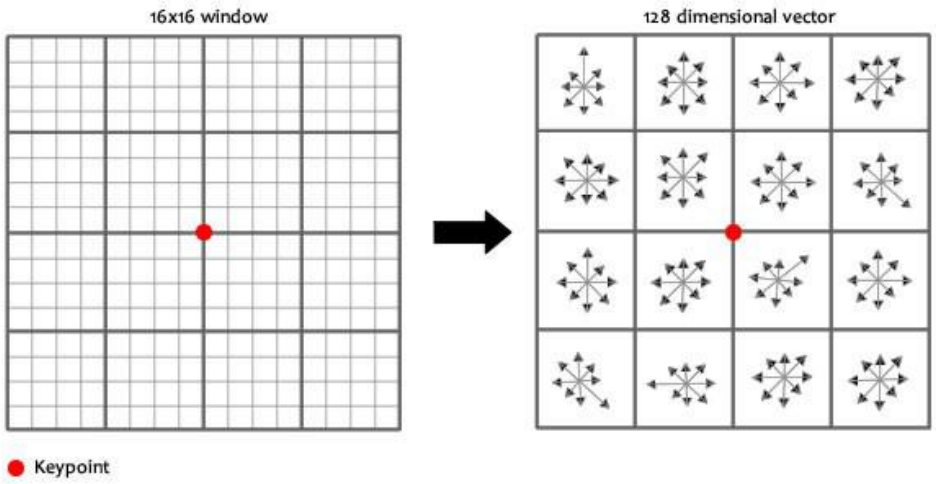

Figure 8 : SIFT Descriptor

For each sub-block, 8 bin orientation histogram is created as shown in the figure below.

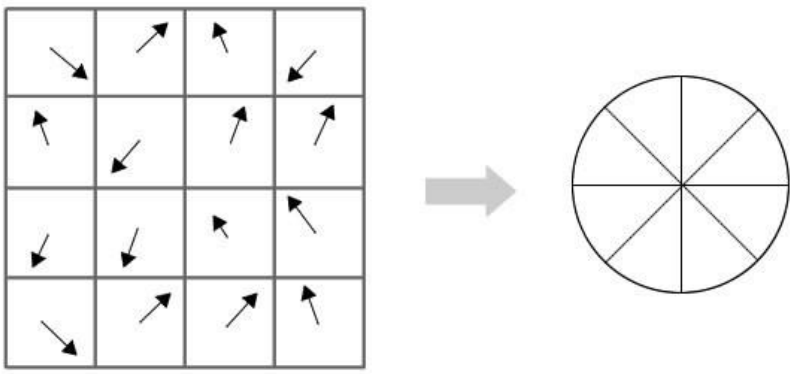

Figure 9 : An example 8 bin orientation histogram

So 4 X 4 descriptors over 16 X 16 sample array were used in practice. 4 X 4 X 8 directions give 128 bin values. It is represented as a feature vector to form keypoint descriptor. This feature vector introduces a few complications. We need to get rid of them before finalizing the fingerprint.

Rotation dependence-The feature vector uses gradient orientations. Clearly, if you rotate the image, everything changes. All gradient orientations also change. To achieve rotation

independence, the keypoint's rotation is subtracted from each orientation. Thus each gradient orientation is relative to the keypoint's orientation.

Illumination dependence-If we threshold numbers that are big, we can achieve illumination independence. So, any number (of the 128) greater than 0.2 is changed to 0.2. This resultant feature vector is normalized again.

5.2.5 Keypoint Matching

Keypoints between two images are matched by identifying their nearest neighbors. But in some cases, the second closest-match may be very near to the first. It may happen due to noise or some other reasons. In that case, ratio of closest-distance to second-closest distance is taken. If it is greater than 0.8, they are rejected. It eliminates around 90% of false matches while discards only 5% correct matches, as per the paper.

5.3 Implementation of SIFT

This section demonstrates the practical implementation of SIFT on a sample picture of a human face with the result of each step discussed in the previous section.

The image on the left is used to to find feature points and then tested on the rotated image on the right.

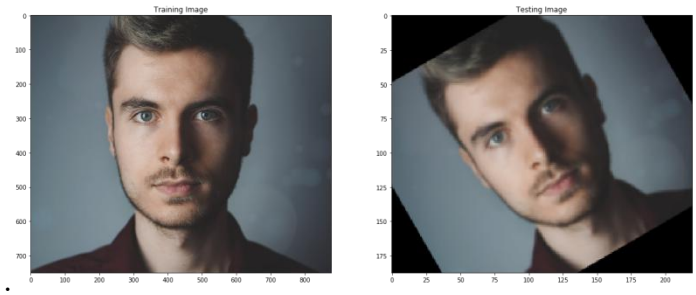

Figure 10 : Sample image in different orientations

Then the keypoints are detected and the descriptor is created.

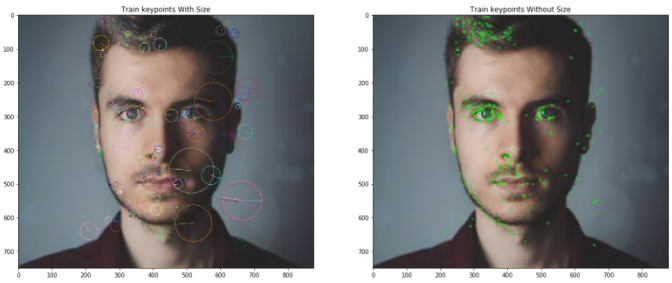

Figure 11 : Identified Keypoints and Descriptor

This is finally followed by matching the keypoints with the scale varied, rotated image (top right corner of the image) as shown in the following figure.

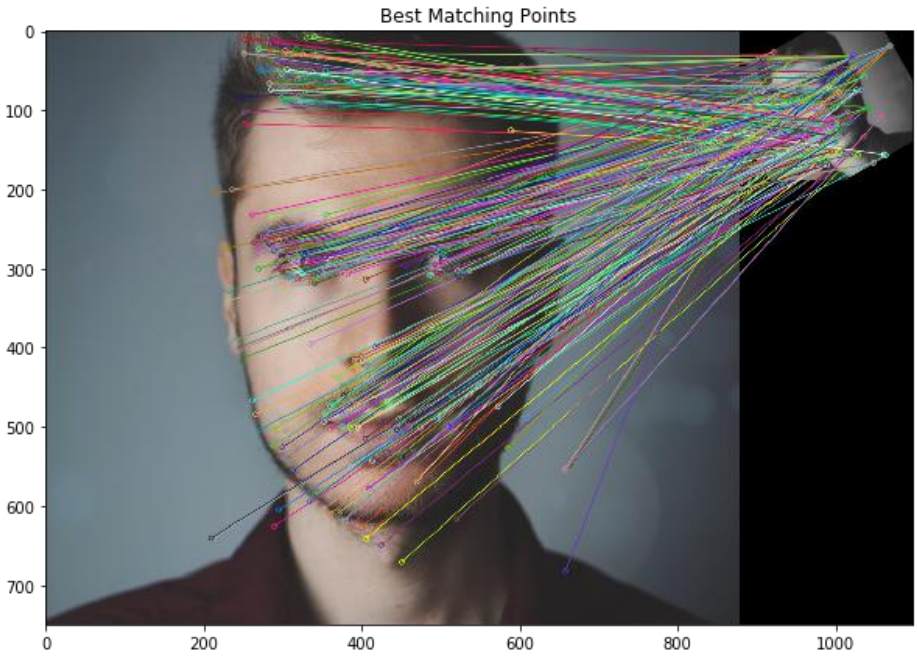

Figure 12 : Matching keypoints.

Chapter 6

HARDWARE

The hardware used consists of all the parts required by a basic Arduino based 4 wheel drive robot.

The hardware parts used and their specifications are:

1. Robot Chassis with mounting holes for modules, wheels, connecting wires and battery holder

Figure 13 : 4WD Robot Chassis

2. L298N Motor Driver Module

The L298N Based Motor Driver Module is a high power motor driver perfect for driving DC Motors and Stepper Motors. It uses the popular L298 motor driver IC and has the on-board 5V regulator which it can supply to an external circuit. It can control up to 4 DC motors, or 2 DC motors with directional and speed control

This motor driver is perfect for driving DC and Stepper motors for micro mouse, line-following robots, robot arms, etc. Its specifications are shown in the table below:

Driver Model	*L298N 2A*
Driver Chip	*Double H Bridge L298N*
Motor Supply Voltage(Maximum)	*46V*
Motor Supply Current(Maximum)	*2A*
Ligocal Voltage	*5V*
Driver Voltage	*5-35V*
Driver Current	*2A*
Logical Current	*0-36mA*
Maximum Power (W)	*25W*
Length (mm)	*44 mm*
Width (mm)	*44 mm*
Height (mm)	*28 mm*
Weight (Kg)	*25 gm*

The pins are shown in the following figure.

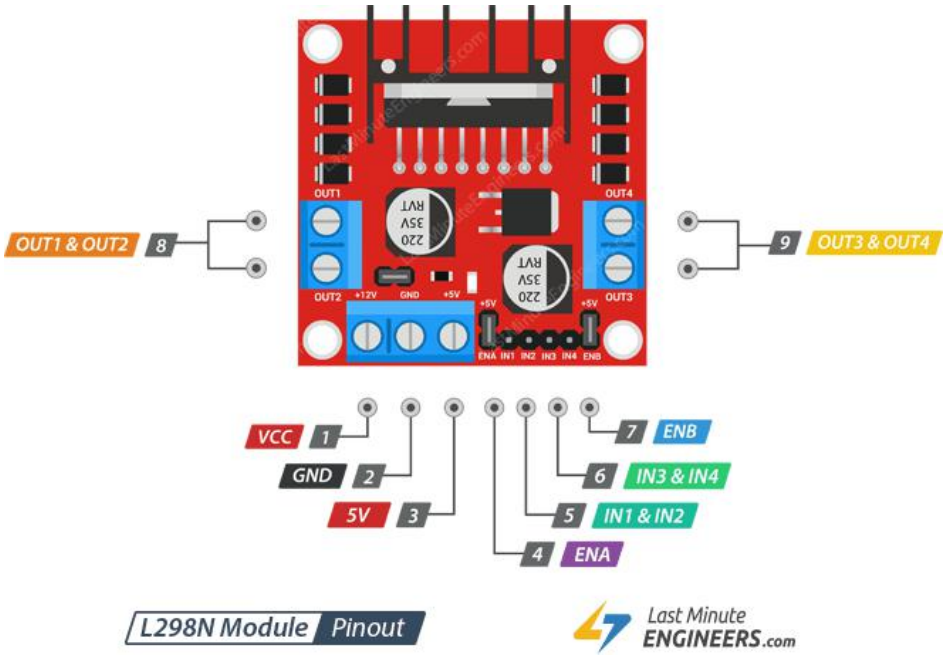

Figure 14 : L298N motor driver and its pins

Connections for the 4 Wheel Drive Robot:

- Out1: Motor A,B lead out
- Out2: Motor A,B lead out
- Out3: Motor C,D lead out
- Out4: Motor C,D lead out
- GND: Ground
- 5v: 5v input
- EnA: Enables PWM signal for Motor A,B
- In1: Enable Motor A,B(Clockwise)
- In2: Enable Motor A,B(Anticlockwise)
- In3: Enable Motor C,D(Clockwise)
- In4: Enable Motor C,D(Anticlockwise)
- EnB: Enables PWM signal for Motor C,D

Note: A,B represent motors on one side and C,D represent motors on the other side of the robot. The image shows the representation for a 2WD and not a 4WD robot.

3. WiFi Module-ESP8266 Node MCU

The NodeMCU (Node MicroController Unit) is an open-source software and hardware development environment built around an inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266, designed and manufactured by Espressif Systems, contains the crucial elements of a computer: CPU, RAM, networking (WiFi), and even a modern operating system and SDK. That makes it an excellent choice for the Internet of Things (IoT) projects of all kinds. The NodeMCU offers a variety of development environments, including compatibility with the Arduino IDE (Integrated Development Environment). The ESP8266 NodeMCU has total 30 pins that interface it to the outside world. The connections are as follows:

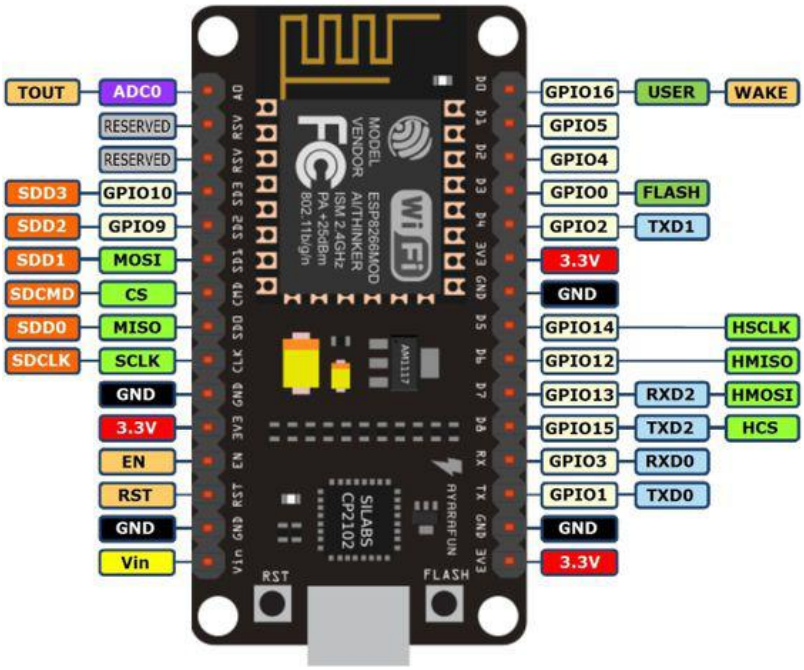

Figure 15 : ESP8266 Node MCU

4. DC Motors for wheels

Figure 16 : DC Motors

Specifications:

- Operating Voltage 3V - 12V DC
- RPM Approximately 150 RPM
- No load current 40 - 80 mA
- Output Torque 3.5 kgf.cm

The circuit connections between the robot, NodeMCU and the motor driver are as shown in the following figure

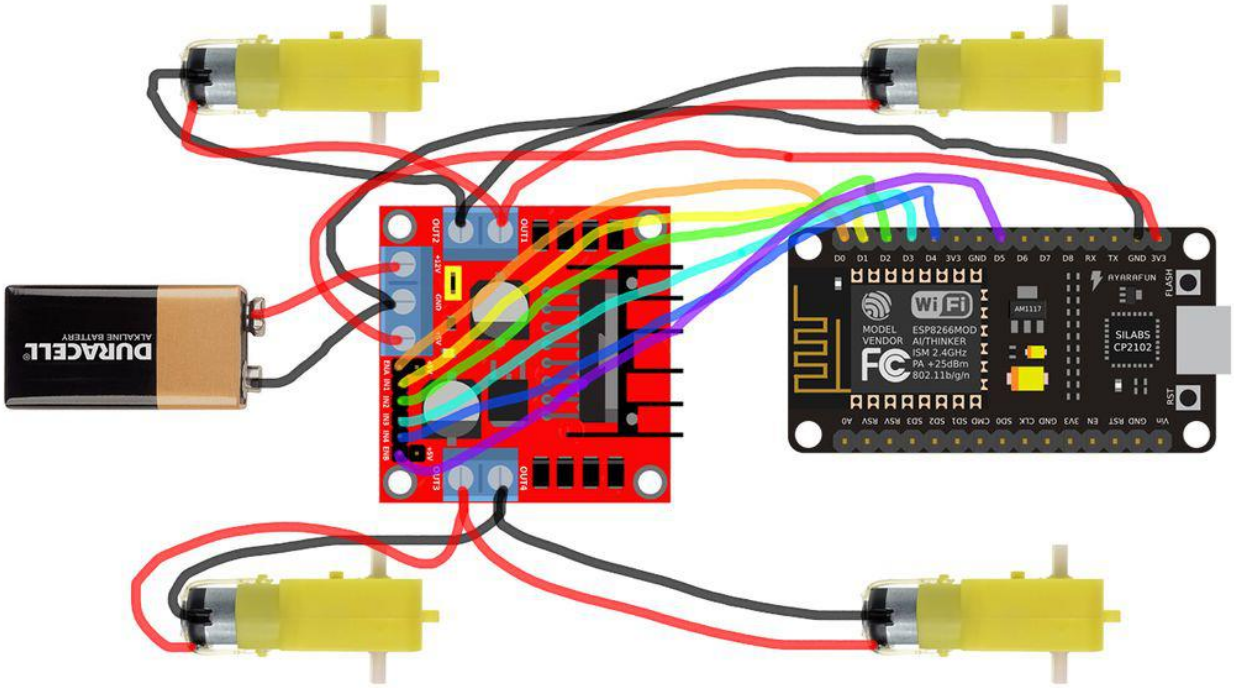

Figure 17 : Circuit connections of the robot

Chapter 7

SOFTWARE

The very heart of the project that is the image tracking algorithm lies in the software application(an Android based application) developed as the user interface to select the object that has to be tracked in the environment.

The software libraries and development kits used are:

1. OpenCV

OpenCV is a library of programming functions mainly aimed at real-time computer vision. This library has dedicated functions and modules to implement several image processing algorithms including the SIFT algorithm. All the example output images depicted in this report were implemented on OpenCV using inbuilt functions and sample images for all steps associated with the SIFT algorithm.

2. Unity 3D with Vuforia SDK

Unity3D is a commercially available multiplatform game engine used for the production of 2D and 3D video games as well as non-game interactive simulations and visualizations.

Vuforia is an Augmented Reality(AR) SDK used to build AR applications. Since this project involves image feature detection similar to AR, Vuforia SDK with Unity3D was used to build the user interface application.

3. Arduino IDE

The NodeMCU and the application must be configured to run on the same network or IP Address for the robot to interact with the application. We use the Arduino platform to enable the communication between the robot and the motors.

Chapter 8

RESULTS

The main objective of this project was to implement a very efficient object tracking robot and this requires a very robust and efficient object tracking image processing algorithm. SIFT was chosen for this purpose and an application was developed to interface the robot and the back-end image processing and feature extraction.

As expected the SIFT algorithm was very efficient in tracking objects kept in multiple orientations and scale as shown in Figure 20 and 21 respectively. If the object has enough feature points, the application mentions that the image is of high quality and begins tracking it.

A green coloured boundary was included to represent the object that is being tracked. Low quality images and images in poor lighting conditions were not able to be tracked by the application. The robot was able to follow the target image tracked by the application communicating via the NodeMCU intern communicating with the motor driver.

The following results depict the tracking of an object in this case a book, in different orientations and scales:

Detecting a High-Quality Frame with enough feature points

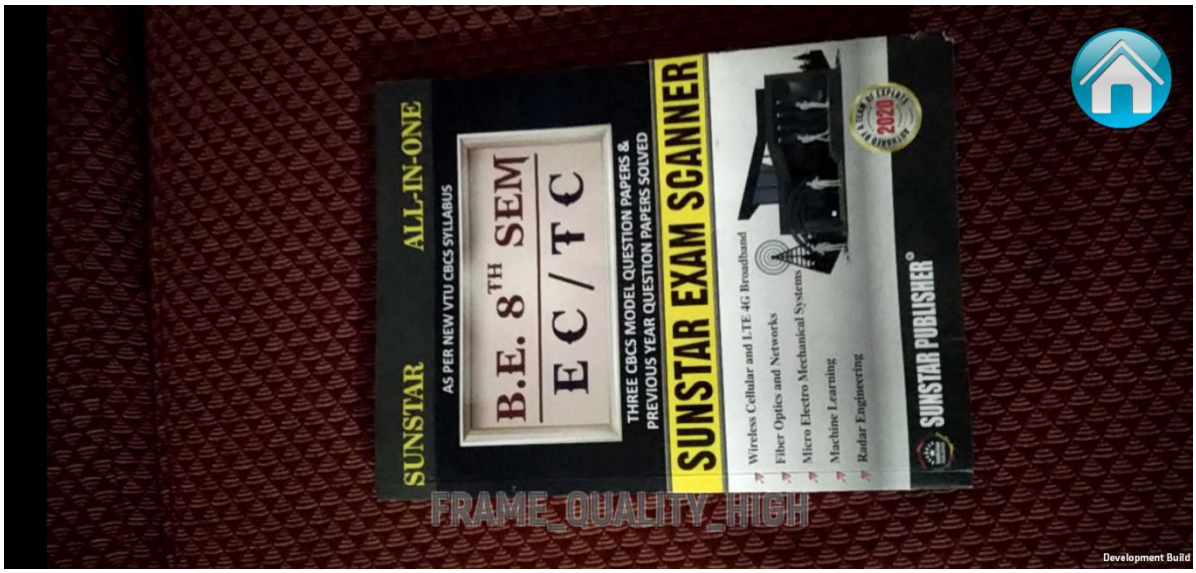

Figure 18 : A High Quality Frame is detected

Once the feature points are identified and stored in the database, the object starts to be tracked.

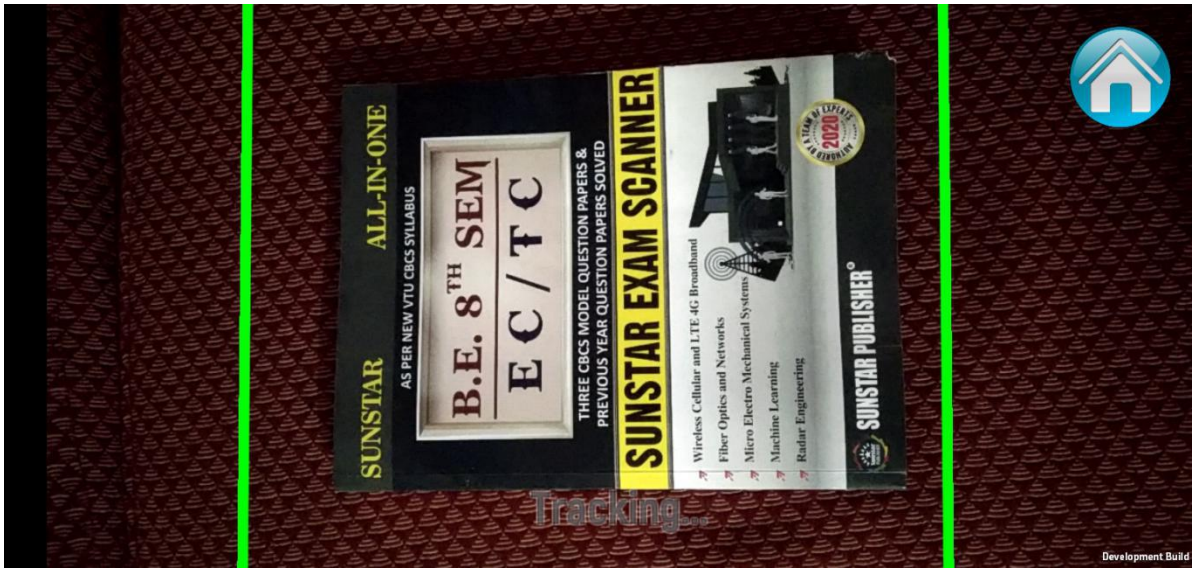

Figure 19 : The object is now being tracked represented in a green boundary

If the object is moved away, the object still continues to be tracked hence satisfying scale-invariance

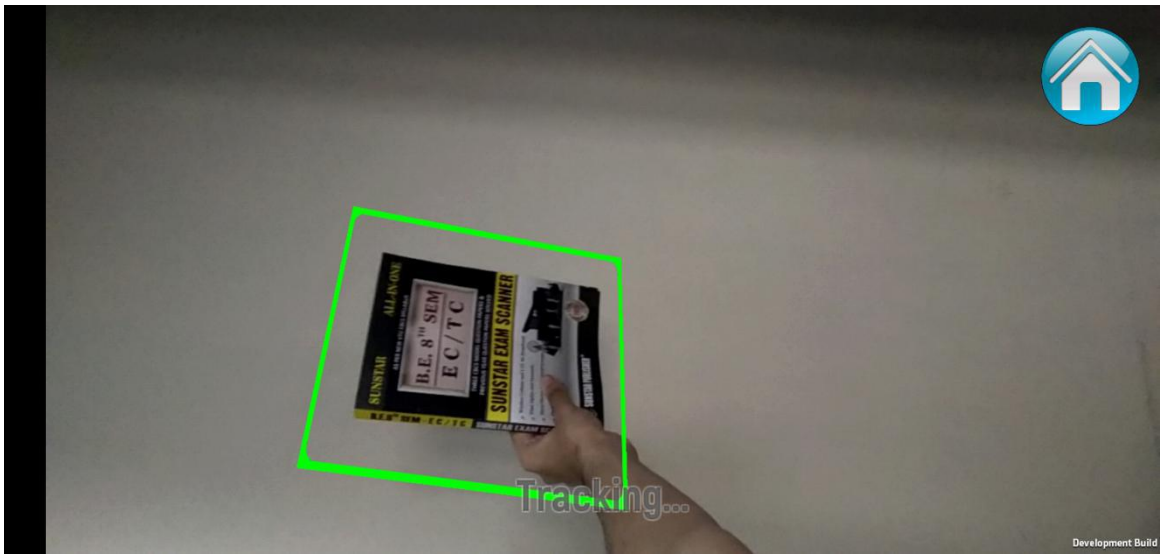

Figure 20 : Tracking a scale changed image

When the object was held in different angles, the object was still being tracked which means that it was orientation-invariant.

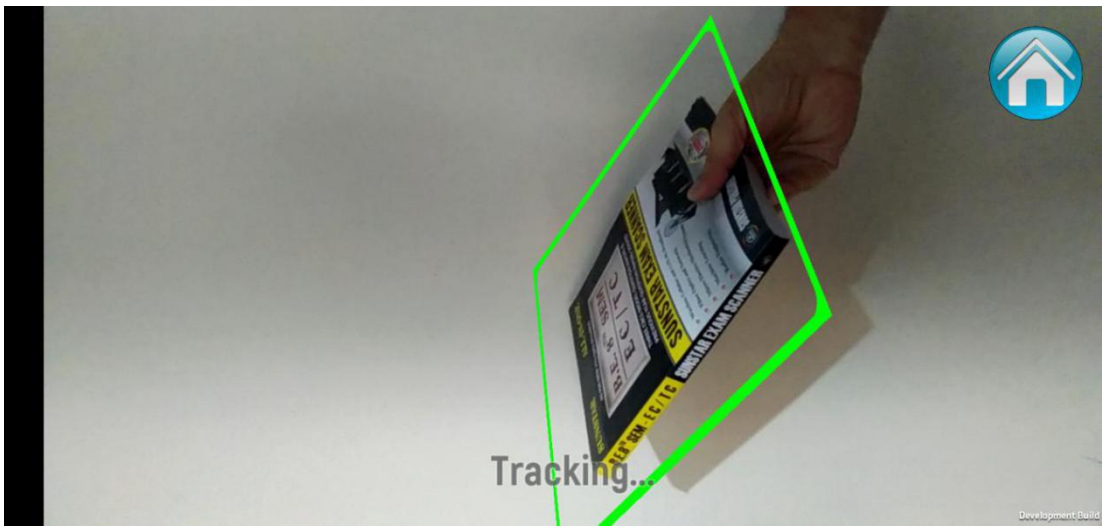

Figure 21 : Tracking an orientation changed object

The complete object following robot is shown below. Evidently, on running the SIFT application on the shown target image/object, several feature points were detected (shown in yellow) thus making the object following operation of the robot very efficient. For the second use case, similar accurate results were obtained ie., the robot was able to approach a remote target object.

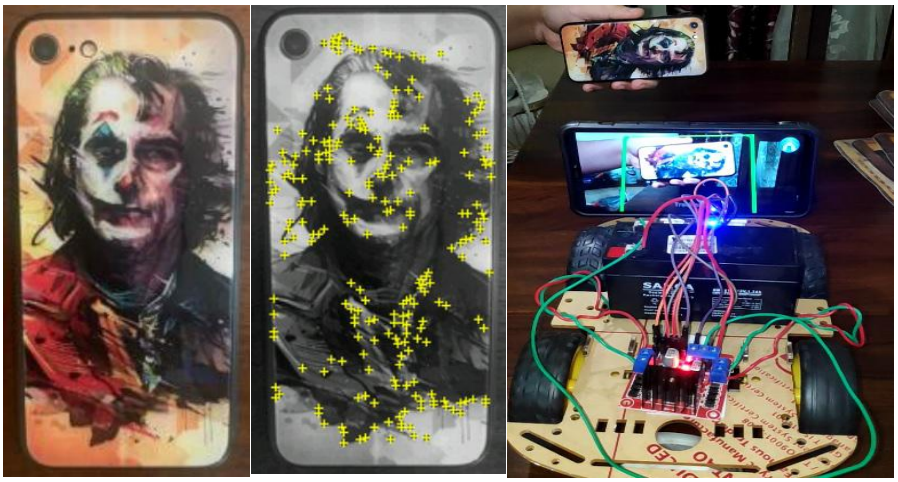

Figure 22 : The complete solution

The robot was able to successfully track and follow the object but objects in low light could not be tracked with the same accuracy. The movement of the robot was smooth with good battery levels but became a little poor with decrease in the battery levels.

Chapter 9

CONCLUSIONS AND SCOPE FOR FUTURE WORK

As per the aim of this project, we were able to identify two applications of image following robots- Industrial robots and Autonomous Cars. Several methodologies were initially identified to go ahead with this project and after thorough research and survey of other research papers and projects, we narrowed down our choices to the usage of a very efficient feature extraction algorithm known as SIFT. As shown in the results and implementation section of this paper, a lot of keypoints were identified in very less time and the robot was able to follow them even when the object was rotated and scaled down but not all object images can yield such good results and good target images must be images with enhanced local contrast, a lot of details and good environmental lighting.

The problem of poor lightning can be solved using illumination techniques based on digital image processing and the same goes with improving contrast and details with techniques such as contrast histogram flattening, image sharpening etc. Another drawback of this implementation is the fact that robot will stop moving if the camera is blocked or no longer in the scope of the object due to any disturbance in the environment or a technical error in the camera. This can be solved by saving the approximate location of the object based on signal strength but this only works in the case of static objects. Similarly if the robot is blocked by an obstacle, it must be able to take a turn as required to avoid collision using ultra sonic sensors for instance or image processing itself. These are the short term goals that have been planned as the next step to improve the functionality of this project

While only two applications and use cases were described in detail in this report, the very same technology can be used in military applications such as guiding a missile to follow an object based on computer vision rather than using heat signature and coordinate systems. Another military application could be in the usage of spy devices designed to follow people or objects. We can conclude that this project has a lot of potential in not just such future applications but also has scope to display better results owing to improved research in this field and greater computation capabilities of modern world technologies such as Computer Vision, Machine Learning, IoT etc.

REFERENCES

Bugała, Michał. (2018). Algorithms applied in Autonomous Vehicle Systems.

J. Sarwade, S. Shetty, A. Bhavsar, M. Mergu and A. Talekar, "Line Following Robot Using Image Processing," 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2019, pp. 1174-1179, doi: 10.1109/ICCMC.2019.8819826.

Lindeberg, Tony. (1994). Scale-Space Theory: A Basic Tool for Analysing Structures at Different Scales. Journal of Applied Statistics. 21. 224-270. 10.1080/757582976.

Lowe, David. (2004). Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision. 60. 91-. 10.1023/B:VISI.0000029664.99615.94.