Self-Driving Autonomous Car using Raspberry Pi

Raj Shirolkar
Department of Computer Engineering
Marathwada Mitramandal's College of Engineering
Pune, India

Rohan Datar
Department of Computer Engineering
Marathwada Mitramandal's College of Engineering
Pune, India

Anushka Dhongade
Department of Computer Engineering
Marathwada Mitramandal's College of Engineering
Pune, India

Gayatri Behere
Department of Computer Engineering
Marathwada Mitramandal's College of Engineering
Pune, India

Abstract— **In the past decade or two, self-driving cars have been drawing a considerable attention for various applications in military, transportation and industrial production. Here we present a remote controlled car which can drive itself to the selected destination using the voice commands as input.**
This car will be able to provide real time obstacle detection and path planning in a dynamic environment with the help of a Raspberry-Pi controller. The car uses Ultrasonic sensors for obstacle detection, a Pi-camera for obstacle identification and a microphone for voice control. The car will contain a cognitive map based model for localization and mapping, which will contain source and destination data. The proposed system is very inexpensive and efficient as compared to the other available systems.

Keywords— *Ultrasonic sensor, localization, obstacle detection, Optimization*.

I. INTRODUCTION

With the birth of voice assistants, it is becoming increasingly convenient to get information using just our voice. We plan to take it a step further and make use of our voice to drive ourselves around. We plan to develop a miniature version of an autonomous self-driving car system which will be able to accept voice input and drive itself to the destination taking the shortest available path by avoiding all the obstacles on its way. To drive this system, we have used a Raspberry-Pi B+ which can handle computationally expensive tasks like image processing and dynamic selection of best path depending on path length and number of obstacles present on the way.

Ultrasonic sensors are employed for Obstacle Detection. A 2-D cognitive map based model represented merely using sparse matrix ensures faster mapping. We have also used a Pi camera for lane detection and obstacle recognition. All the detected obstacles encountered on the way will be dynamically added to the database with their location. For storing all the data related to the map we have used SQLite database running on Raspberry-pi. We have provided web-based controls deployed using Flask web framework and the Raspberry Pi's built in Wi-Fi modem.

There are numerous papers on self-driving cars which use web-based controls, but they lack any other secondary mode of input. Here, along with these web-based controls, an onboard microphone is provided which will take voice input as commands for the car.

II. RELATED WORK

The concept of self-driving cars has been around for a while now. Different controller combinations like Arduino and Raspberry-pi have been used to distribute the work load. This is because different real-time tasks like image processing distance calculation, localization, decision making etc. are needed to be done continuously with precision and accuracy, hence having a threat of hardware failure [1][8]. For localization and mapping purpose, a dynamic feature extraction technique for mapping of point cloud for generation of 3D cognitive maps has been used with the help of scene recognition and alignment without any initial pose estimation requirement. It surpasses the typical approach of localization through a particle filter [2]. The 2D Neural network based map learning techniques have also proven to be effective for path planning. In this approach different images of the environment are trained with their position as the label. The car then clicks a picture of the environment dynamically through a camera module and finds its current location. But this technique gives high time and space complexity which is not affordable [4]. Work has been done in path planning using trajectory generation and evaluation techniques which have proven to be useful on curved roads and mountain passes to obtain an optimal solution provided the environmental information is available beforehand [3]. Exploration of car designs which work on solar power has also been done. It solves the problem of continuous energy supply required by the car and it's in-built hardware. The overtaking mechanism depending on the measurements obtained from the environment, along with its mathematical computation has been introduced by Luiz G. Hafemann, Robert Sabourin, Luiz S. Oliveria [6]. PSO based search algorithm to find single source multiple destination shortest paths using an particle encoding decoding scheme for exploitation of sub optimality features of the shortest path is provided [7]. Ultrasonic distance sensor and a buzzer is used in building a system to execute obstacle detection and alerting. The system detects the obstacle within a certain distance threshold. This system can be integrated in cars to avoid collision[5].

III. PROPOSED SYSTEM

The overall work can be divided into following major categories. The proposed methodology can be shown briefly using fig 1
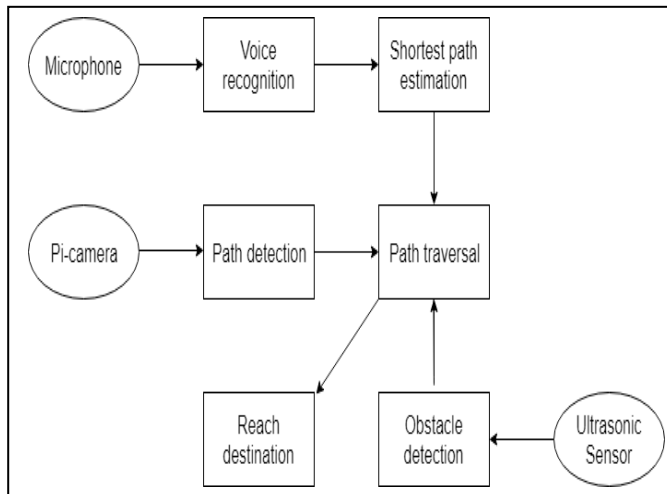

Fig 1 : System Architecture

A. Voice Recognition for input module:

In our system destination input is provided by using voice commands through microphone. This input needs to be converted from speech to text. In speech-to-text conversion, the basic function is to convert speech in physical sound to an electrical signal. Then this signal is converted to digital data. This audio data is transcribed to text using several speech recognition services. In the proposed model, we have used SpeechRecognition package which acts as a wrapper for various speech APIs.

In our model Google Web Speech API is used. This API recognizes 120 languages which makes it flexible for the global user base. The Recognizer class is used for recognizing the speech. This Recognizer instance has various methods for recognizing speech from audio input using different APIs. The method requires audio_data argument which must be an instance of the class AudioData. Here, as we are providing the facility to provide speech input using a microphone, this input will be used as an argument to our recognize method.

To access Microphone for the input, PyAudio library is required. An instance of Microphone class is used to access the microphone. Now the microphone input is captured using Recognizer class. The audio source is used as an argument to and it records input from source until silence is detected.

B. Lane detection module

1) Preprocessing

a) Cropping the Region of Interest

Image processing is a computationally heavy task to be handled on a Raspberry-Pi. In order to reduce this task we crop the Region Of Interest from the original image and utilize it for further processing. We need to detect the parallel lanes on the road between which our car is moving. The parallel lanes converge towards the center of the image and also the road is located only in the lower half of the image

captured through the pi-cam. Therefore a triangle having it's vertices at the two bottom corners and the center of the image will be suitable for cropping process.

b) Converting into GrayScale

For the process of edge detection, we do not care about the colors in the image but just the difference in the intensities of the pixels. Hence in order to remove the extra information regarding the colors and assigning a single intensity value to each pixel we convert the RGB image into a grayscale image.

2) Detecting Edges

The edges in the image are nothing but the areas in the image where the intensities of the values change rapidly. Hence finding the mismatch between a pixel and all of its neighbors for all such pixels is necessary. For detecting these areas, we have used the Canny Edge Detection algorithm. It detects areas of the gray scale image that have a strong gradient. A minimum threshold can also be set for the gradient in order to indicate the strength of the edge detected.
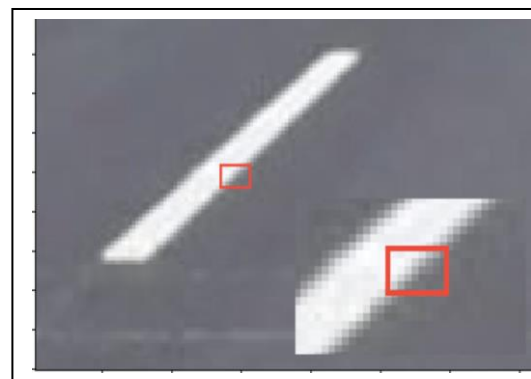

Fig 2: Edge Detection

3) Generating lines using Hough Transform

In Hough transform all the points and lines in an image are converted into a different dimensional space known as the Hough space. Here all the points from the image space represent a line from the Hough space and all the lines from the image represent a point in the Hough space. Hence for every edge pixel detected, subsequent line is drawn in the Hough space. The intersection point of the majority of these lines is the selected as a single line in the original image space which actually defines the lane detected on the road.
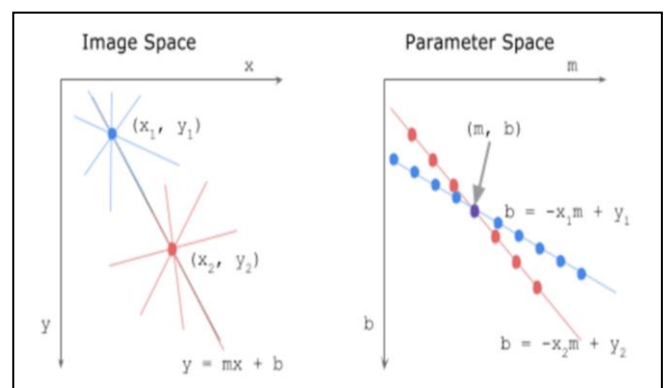

Fig 3: Hough Transform

C. Obstacle detection and identification module

Whenever the car stops, an image is captured from the pi-cam. This image is then passed on to a pre-trained model to identify the objects in the image and particularly in front of the car. This model is already trained on the ImageNet-1000 dataset using a Neural Network. We have used the python's ImageAI OjectDetection library. This pre-trained model is configured and loaded into the Detector class. At runtime only the image has to be supplied to the algorithm which gives all the objects detected from the image along with its accuracy.

As the raspberry-pi is not suitable for such computationally heavy tasks, we have implemented this module on a remote server to increase the processing speed and get quick results at real-time giving the accuracy of 99.83%.

D. Web based controls and video streaming

The controls of the car can be accessed via a web app deployed on the Raspberry Pi using Flask. Flask is a micro framework known for its light weight nature and limited but useful functionalities thus making it perfect to deploy on a computationally weak device like Raspberry Pi. The web app consists of a live video stream generated directly from the Pi camera and basic controls like voice input and manual start stop buttons for additional functionality. The web app can be accessed from authorized devices on the same network.

IV. CONCLUSION

This paper presents a system for a self-driving car which is able to reach a given destination specified through voice commands or web based controls. Also, the system efficiently detects and avoids any obstacles on its way. The proposed system has a generic model which can be deployed on any vehicle irrespective of its size. This low cost alternative to current generation self-driving modules will pave the way for a bright future for self-driving cars and thus will have a wider reach.

REFERENCES

[1] Mohammad Rubaiyat, Tanveer Hussain "Design of an IoT based autonomous vehicle with aid of computer vision", International Conference on Electrical, Computer and Communication Engineering, IEEE 2017.

[2] Ren C. Luo, Vincent W.S and Chung-Kai Hsieh," 3D Point Cloud Based Indoor Mobile Robot in 6-DoF Pose Localization Using Fast Scene Recognition and Alignment Approach, International Conference on Multi-sensor Fusion and Integration for Intelligent Systems" ,IEEE ,2017.

[3] Chaocheng Li, Jun Wang, Xiaonian Wang and Yihuan Zhang, "A model based path planning algorithm for self-driving cars in dynamic environment," *2015 Chinese Automation Congress (CAC)*, Wuhan, 2015.

[4] S. Chen, J. Shang, S. Zhang and N. Zheng, "Cognitive map-based model: Toward a developmental framework for self-driving cars," 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, 2017.

[5] A. W. Mohemmed, Nirod Chandra Sahoo and Tan Kim Geok, "A new particle swarm optimization based algorithm for solving shortest-paths tree problem," *2007 IEEE Congress on Evolutionary Computation*, Singapore, 2007.

[6] Mayur Bhangale, Gaurav Dabhade, Akshay Khairnar, Mamta Bhagat, "Self-driving car to demonstrate real time obstacles & object detection", International Research Journal of Engineering and Technology, 2016.

[7] Deepak G, Harish Kumar N, Nagaraj J, "An IoT based Obstacle Detection and alerting System in vehicles using Ultra Sonic sensors", International Research Journal of Engineering and Technology, 2017.

[8] Gurjashan Singh Pannu, Mohammad Dawud Ansari, Pritha Gupta, "Design and implementation of Autonomous car using Raspberry-pi", IEEE, 2015.