University of Colorado
Boulder

EXERCISE 5

BY

SIDDHANT JAJOO

UNDER THE GUIDANCE OF:

PROFESSOR SAM SIEWERT,
UNIVERSITY OF COLORADO BOULDER

7/26/2019

BOARD USED: NVIDIA JETSON NANO

TABLE OF CONTENTS

QUESTION 1 2
SUMMARY 2
QUESTION 2 3
CODE DESCRIPTION 3
QUESTION 3 4
QUESTION 4 5
BLOCK DIAGRAM 5
DESCRIPTION 5
REFERENCES 6

QUESTION 1

SUMMARY

The main technological challenge in the development of Stanley was to build a highly reliable system, capable of driving through unrehearsed off-road terrain. One of the major goals and requirement was to vary the speed level of Stanley as per the terrain and its surrounding. This was achieved using Machine vision and its application to detect drivable paths, detecting objects in its path to maneuver it, real time collision detection and stable vehicle control. In order to achieve this, algorithms were developed using machine vision and machine learning. Laser and vision systems were used for this purpose. These were placed at the roof of the vehicle where the GPS signals were best received and visibility was the highest. The laser system was used for terrain mapping i.e detecting drivable and non-drivable terrain to make stops beforehand if required. The surface was assigned with three possible values: occupied, free and unknown. Applying this classification to the laser data directly yields inaccurate results. This is resolved by using a probabilistic test. This method eliminates the error and generates a drivable map. Now since the lasers are used to detect short and medium range obstacles, this is sufficient to detect obstacles at a speed of 25 mph, but for higher speeds it would require to cover a lot of ground in order to detect drivable paths. For this reasons, color camera is used for computer vision terrain analysis. This can classify upto 70 metres of range in front of Stanley. Since the road appearance is affected by a number of factors, it is difficult to classify the road in drivable and non-drivable regions by the color camera all by itself. Therefore an adaptive algorithm is implemented which interprets the images as per the environment conditions. The drivable region is found out by projecting the area classified by the laser on to the camera vision. A point to note here is that the camera vision should not be used for steering control. It is not as accurate as the laser control. However, the camera vision can be used for velocity control. The vision system works as an early warning system for non-drivable regions. i.e If it sees a non-drivable region beyond a range of 40 m, it slows down the vehicle to 25 mph and increases to 40 mph if it identifies a drivable region. Thus machine vision is also used in velocity control of the vehicle.

Some other applications of machine vision are finding out the road boundary to keep it at the center of the road with the help of probabilistic low pass filter such as Kalman Filters(KF), terrain ruggedness which helps to keep the velocity in check by identifying the type of terrain. Computer vision also plays an important role in velocity and steering control.

As far as the question to replace LIDAR by multiple vision cameras, this can be done but would require extensive improvements in the algorithms to detect obstacles and path. The driving conditions vary a lot in different environments. LIDAR is invariant to changes in the environment thus yielding good positive results as opposed to computer vision cameras. Color, shapes and sizes of different objects make it difficult to detect a certain area solely on the basis of computer vision cameras. This is the sole reason why cameras were not used to control the steering in Stanley. A good option prevalent at this time is to use the camera and LIDAR simultaneously, thus yielding better results but with improved algorithms, machine vision may be used in future.

QUESTION 2

CODE DESCRIPTION

The first step in the code is to read the image from the video. The image obtained is converted into grayscale. The graysclae image is then fed to the detect_display_vehicles function in order to detect the vehicle using Haar Transform and drawing rectangular boxes around the same. The image is then displayed on the screen. For detecting lanes, the grayscale image obtained is then applied a Gaussian filter to eliminate noise. Canny edge detection is then applied in order to obtain edges in the Image. HoughLinesP is then applied in order to obtain lane lines.

Alternative Improved Implementation:

The first step in the code is to read the image from the video. In order to reduce the computations in the code, the image is halved by 2. The top of the image which includes the sky is cut off, thus keeping only the required part of the image i.e the part consisting of lanes and vehicles. Furthermore, the resolution of the image is also halved by 2. Now for vehicle detection the latest image obtained after halving the resolution is converted into grayscale and fed to the detect_display_vehicles function in order to detect the vehicle using Haar Transform and drawing rectangular boxes around the same. The image is then displayed on the screen.

For detecting lanes, the obtained image is first converted into grayscale, then the grayscale image is darkened by increasing the contrast. For this purpose, equalizehist function was used. The Grayscale image obtained is then applied a Gaussian filter to eliminate noise. Canny edge detection is then applied in order to obtain edges in the Image. The rest of the edges in the image except the lanes are eliminated by choosing a region of interest and masking the rest of the image. HoughLinesP is then applied in order to obtain lane lines.

The above algorithm has much more scope to improve. Filters can be created in order to detect white and yellow lines in the image. This will increase the accuracy of lane detection. An algorithm to distinguish the left and right lane on the basis of their slope is yet to be written which would lead us to provide the lane departure warning system. The FPS can be improved using multithreading techniques or by using CUDA.

All the false detects would be removed if Region of interest is implemented and all the unnecessary parts of the frames are removed.

The entire frame sequence is recorded in mp4 format.

QUESTION 3

RELEVANCE

There is a huge scope and significant progress has been made in machine vision applications. In order to explore them, the best application of machine vision would be to design an autonomous driving vehicle or atleast some parts of it. To that end, I have thought about a project which includes some of its basic features if not all of it. The project has a lot of future scope.

The project name is "Self-Driving Auto-Pilot" with the following features:

- Lane Following with a feedback system maintaining a pointer at the steering wheel.
- Vehicle Detection with messages to decelerate when required.
- Pedestrian Detection.
- Sign Detection.

CHALLENGE, APPLICATION AND LEARNING

Lane detection algorithm will make use of canny edge detection and Hough lines to mark the lanes. This concept has been taught in this course and a small assignment has also been completed. Thus, I would like to use this concept to create a lane departure warning system by correctly detecting lane lines on both the sides of the vehicle. If possible a feedback pointer to steering wheel would also be incorporated to indicate the turn in degrees.

A moving car needs to keep a map of the vehicles and pedestrians in front and the back of the car. Thus, vehicle detection and pedestrian detection would also be implemented to stop or decelerate the car at any point of time.

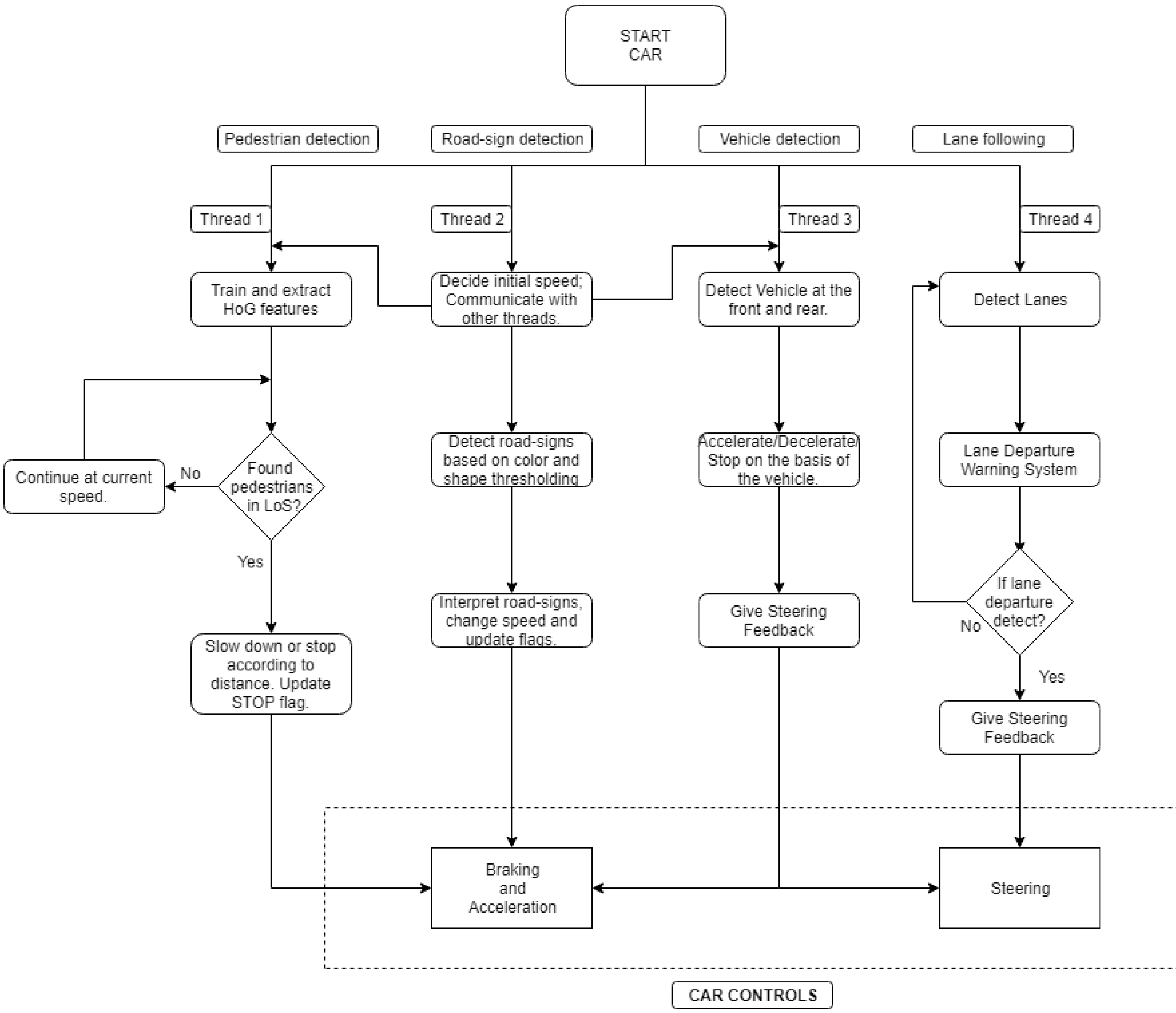
In addition to this, Sign Detection will also be employed to check for any stop signs or speed limit signs to obey the traffic rules. Haar transform would be used for this particular objective.

All these applications can be extended to autopilot features in an autonomous vehicle.

Challenges would include to improve FPS for real time detection, make the lane departure system to work even at turning points, detecting curved lanes and having steering feedback.

QUESTION 4

BLOCK DIAGRAM



DESCRIPTION

The lane detection module will detect lanes and trigger a warning if it crosses a lane thus enabling the lane departure warning system with steering feedback. Vehicle Detection would detect the vehicles at the front in order to determine if it should decelerate or not based on the distance of the vehicle in front. Pedestrian Detection would detect any pedestrians on road and if present any, would trigger the vehicle to stop, slow down or continue at the current speed. The same is applied for road sign detection, if it detects a stop sign, the vehicle must stop. Following and reacting to speed signs can also be implemented if time permits.

Minimum:

1. Detection edges using canny and Hough transform.

2. Use of Histogram of Gradience.
3. Using Haar Transform to detect vehicles and road signs.

Target:

1. Lane Detection with steering feedback.
2. Proper Vehicle Detection without noise.
3. Pedestrian Detection without noise.
4. Road Sign Detection and Deceleration messages.

Optimal:

All the above target requirement with multithreading or using GPUs in order to increase FPS. Any methods such as applying correct filter to images, reducing resolution of images, cropping the image converting to grayscale or binary image in order to reduce computations and thus increase frame rate.

Future scope if time permits:

1. Use of Google Map APIs to navigate.
2. Accelerating and Braking.

The Project would be done in a group of two.

Team Members: Siddhant Jajoo and Sarthak Jain.

REFERENCES

- https://opencv.org/