Detecting Lane Lines for Self-Driving Cars

Madhusudhan D Just now · 4 min read

> *This Project is a concept demonstration for self driving cars to identify the Lane lines as part Udacity Self-Driving Cars Nanodegree.*

Description and Goal of the Project

When we drive, we use our eyes to decide where to go. The lines on the road that show us where the lanes are act as our constant reference for where to steer the vehicle. Naturally, one of the first things we would like to do in developing a self-driving car is to automatically detect lane lines using an algorithm. Such an Algorithm will analyse all the images coming from the vehicle camera and tries to annotate those images by marking the lane lines.

The goal of this project, the first of Term 1 of the Udacity Self Driving Car Engineer Nanodegree, is to develop a lane finding Pipeline based on Python and OpenCV. OpenCV means "Open-Source Computer Vision", which is a package that has many useful tools for analyzing images. The Pipeline take a single image as input and writes out the same image with marked lane lines.

Description of the Pipeline

The Pipeline consists of the following steps executed in series:

1. Read in a image and convert it into Grayscale

Pic 1 : Test image read and converted to Gray Scale

2. Apply Gaussian smoothing to blur the Grayscale-image. This in turn will reduces image noise and detail. This step enhances the quality of Edge-Detection algorithms

Pic 2 : Gaussian blurred gray scale image

3. Use canny edge detection algorithm to detect edges.By smoothing in previous steps, the number of edges detected are considerably reduced

Pic 3 : Image after canny edge Detection

4. Apply hough transform to transform canny edges into lines in hough space and pick only those lines in pre-defined region of interest. The Region of interest here is rectangular region from the base of the car which will include the lane lines

Pic 4 : Hough lines detected in Region of Interest

5. Calculate the lane line and draw line over the image. Calculating the lane line from the lines delivered by hough transform is probably the most critical part of this project. Here the lines segregated into left and right lanes based on their slope. Each of the line are then tested by extending to base of the image using the slope and checking if they fall into emperically determined X Values. Use the x and y points of the 3 longest three lines to linearly fit to a line. The interpolated line image is then merged to initial image using addWeighted of the CV library.

Pic 5 : Interpolated line drawn on original image

The parameters used for canny edge detection, hough transform and draw lines function have to tuned to arrive at the best solution. This however have to be done iteratively on the given test data. The values used by my Pipeline is available in my source code.

Analysis of Results

The Pipeline identifies the left and right lines of the lane in which the car is travelling and then plots it on the image successfully with different colors. Outputs of the pipeline for two different camera videos are shown here :

There are however potential shortcomings and scope for improvement

1. The lane line, especially dotted line, could vanish from the frame because in some frames, the dotted lane is barely visible and hence cannot be isolated. This leads to flickering of lane lines in the video.
2. The lane line direction is not always point to driving direction. This is attributed to the fact that linear interpolation is used for approximating lane. Hence curving lane lines cannot be captured.

These shortcoming can be overcome by making a architectural change for the Pipeline. The pipeline should support using information from previous images to approximate the lines when lane lines are not visible and cannot be isolated. Another change could be the use of high order interpolation to approximate the lane curves

Lane Detection Python Computer Vision Self Driving Cars Pipeline