

Reflections : Project 1, Finding Lane Lines on the Road

1. Pipeline

- 1- Convert the original color image with 3 channels to grayscale image with 1 channel
- 2- Convert grayscale image to Gaussian filtered image to smooth out noise
- 3- Convert Gaussian filtered image to image with edges using Canny edge detection
- 4- Convert image with Canny edges to image with edges only in region of interest on black background :
  - a. Define mask (numpy array) with same dimensions as Canny image, with black background
  - b. Apply bitwise-and operation to Canny image and mask
- 5 – Find lines in Canny edged image using Hough Transform, stored as numpy array
- 6 – Draw Hough lines on image with black background:
  - a. Define mask (numpy array) with same dimensions as original image (with 3 channels), with black background
  - b. Draw lines on mask
- 7 – Perform weighted addition operation on original image array and array representing Hough lines on image with black background
- 8 – Return resulting image array

2. Potential Shortcomings

If there are any objects in the same lane, such as tree branches, lumber, rain, snow, trash, road imperfections, shadows, hanging tree branches, animals, passing motorcycles, these will be interpreted as lines that will incorrectly influence the lane line determination.

3. Possible Improvements to pipeline

The potential shortcoming, of objects in the lane, has already been partly addressed in the current pipeline by:

Only accepting candidate lines of a certain pixel length (a parameter in the Hough Line Transform), and also only accepting lines of at least a certain slope. This is able to select out the dashboard in the challenge video, and a large number of road imperfections and shadows.

A big possible improvement would be to recognize other possible objects in the lane, using machine learning, so that the objects are not considered in the lane line calculations.