

Project Writeup

Anirudh AV (anirudh.av93@gmail.com)

1. Pipeline

1.1. *Converting to a grayscale image*

The image is initially converted to a grayscale image which will later be fed into the canny edge detection algorithm. This is done using the `cvtColor` function in `opencv`.

1.2. *Canny edge detection*

Canny edge detection is performed using the Canny function in `openCV` which basically computes the gradient at every pixel which when applied on the grayscale image efficiently indicates the presence of an edge.

1.3. *Region of Interest*

I know that the lane lines will appear at a particular part of the screen (given a fixed position of the camera). I can use this knowledge to apply a mask to the image and narrow down my region of interest to a certain portion of the screen. This will improve my efficiency in processing time and also rule out many possible fake edges.

1.4. *Hough Transform*

Now I have the information about the presence of edges contained in pixels, I now apply a hough transform to the image. Since the output of the canny function is an image of dots containing information about edges of any shape, I can now fit a line to these dots find lines that pass through these edges. I can use the `min_no_of_lines` and `max_line_gap` as tuning knobs in my hough transform to efficiently compute only the required edges in my hough image.

1.5. *DrawLines*

After the implementation of the above steps, I have the desired lines that should represent the lane lines in the image. I need to represent it on the given image. I first overlay the lines using the `addWeighted` helper function. Now I extrapolate the lines that have been identified by the hough transform in order to get a solid line on the image. This is done by first segregating the points from the hough transform into left and right (based on the sign of their slope). Then I fit a line to the point to figure out the slope and intercept which I use to calculate the 'y' value for a given 'x' value. Once done for the 4 endpoints of the region of interest I have the final lines that I display on the image.

1.6. *Potential improvements*

The lines drawn on the image flicker for the solid yellow lines video. The parameters can be more systematically tuned to achieve better results.