Lane Line Estimation

1. Pipeline Description

The input to the system is an rgb frame. It is assumed that the lane markings in the frame are either yellow or white. Below are the steps in the pipeline.

1. Color thresholding
    a. Convert the image from rgb space to hsv space
    b. Define hsv ranges for yellow and white through trial and error
        i. Use hue and value range for yellow detection
        ii. Use saturation and value range for white detection
    c. Set all non-yellow and non-white pixels in the image to black
2. Grayscale
    a. Convert frame to grayscale
3. Smoothing
    a. Run a gaussian blur over the image with a standard deviation of 1 via a kernel of dimensions of 5x5
4. Edge Detection
    a. Run the canny edge detector over the image
    b. Algorithm parameters were selected through trial and error
5. Region of Interest Selection
    a. Establish a symmetric ROI in the shape of a trapezoid
        i. The ROI was selected by overlaying several ROIs and seeing which one best covers the ego vehicle's lanes
    b. Set all pixels outside the ROI to black
6. Line Segment Detection
    a. Use the hough algorithm to extract edges from lines
    b. The parameters to the algorithm were selected through trial and error
7. Line Segment Classification
    a. Lines found in the previous step are classified as either a left lane line or a right lane line based on the slope of the line.
        i. Lines with a positive slope are said to belong to the left lane
        ii. Lines with a negative slope are said to belong to the right lane
        iii. Lines with an almost 0 slope are discarded
8. Line Fitting
    a. All line segments belonging to a single lane are aggregated into a single line segment
        i. This is done by decomposing the lines into points, running the RANSAC algorithm to remove outliers and finally fitting a linear regression model to inliers
        ii. A line is determined from the bottom of the frame to the uppermost inlier
9. Line Drawing
    a. The lines produced from the previous step are drawn on the original image

2. Potential Shortcomings

- The system may not function as expected on faint lane lines
    - The edge detection step may miss faint lines meaning that the line detection
- The system will not function as expected on roads with non-white and non-yellow road lines
    - The color thresholding step will result in an all black image
- The system will not function as expected if the lane lines are not directly ahead of the vehicle
    - The ROI assumes the lane lines are directly ahead of the ego vehicle
    - If the vehicle is driving directly over a lane line, it might not detect the other lane line if it is outside the ROI
- The system will not function as expected on non-straight drives
    - The line fitting algorithm fits a line to a set of points. If the set of points trace out a curve, a line will not fit well
    - One possible solution is to use a higher order polynomial in the curve fitting step.
- The system will not work very well on roads with a lot of snow that may occlude the lane markings
- The system likely will not detect a lane line bisected by a speed bump

3. Potential Improvements

- Fitted Line Low Pass Filter
    - In reality, the lane lines on a road do not change very much in very short amounts of time. We can require that the lane lines in the n+1th frame be similar to the lane lines in the nth frame
    - This may be done by running the lines found in step 8 through a low pass filter before displaying them in step 9.
    - This will attenuate sudden and large changes in the generated line estimates
- Fit a higher order polynomial to the data in step 8. This will allow the system to detect curved lane lines
- The speed at which the system processes each frame should be improved
    - Currently, it takes about 30 minutes to process a 30 second video. This is not usable for real time applications
    - This can be done by running a speed profiler on my code and seeing where the bottlenecks are and optimizing those
    - Converting the system to a faster language like C++ can also help