writeup_template.md

Finding Lane Lines on the Road

Writeup Template

You can use this file as a template for your writeup if you want to submit it as a markdown file. But feel free to use some other method and submit a pdf if you prefer.

Finding Lane Lines on the Road

The goals / steps of this project are the following:

- Make a pipeline that finds lane lines on the road
- Reflect on your work in a written report

Reflection

1. Describe your pipeline. As part of the description, explain how you modified the draw_lines() function.

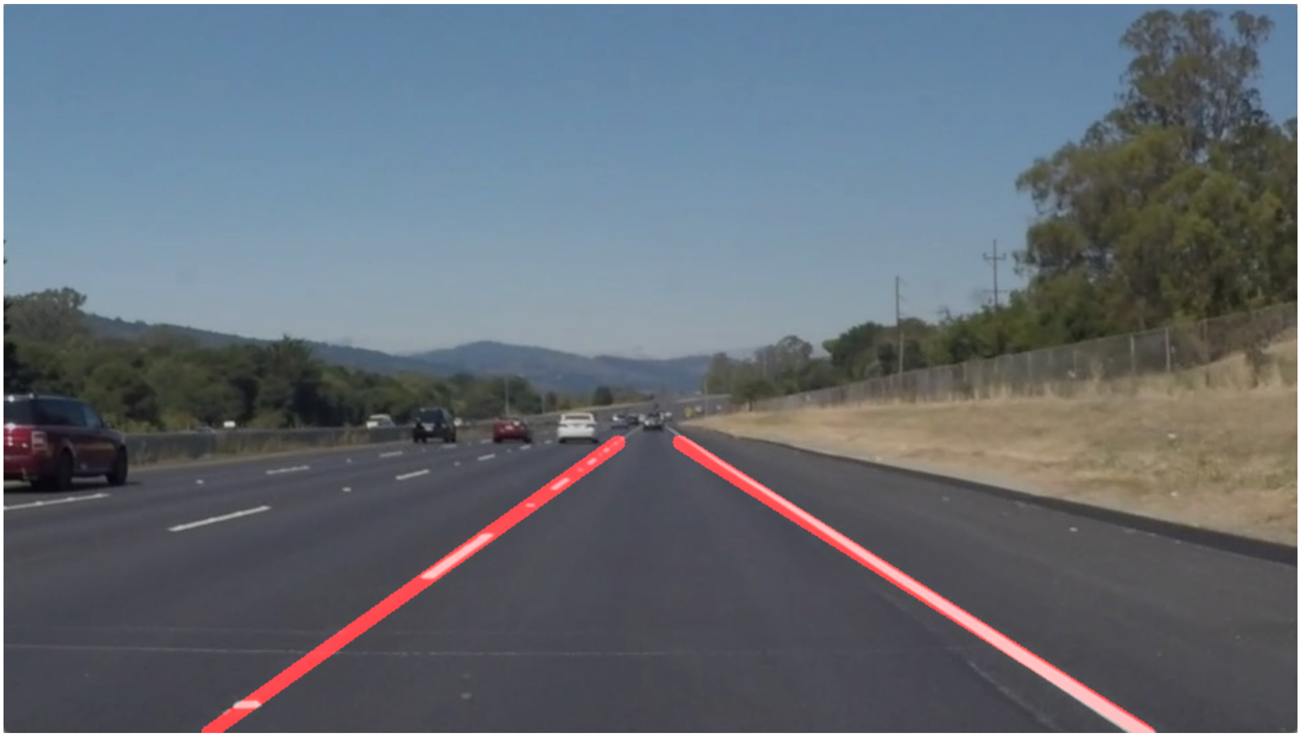
My pipeline consisted of the following steps:

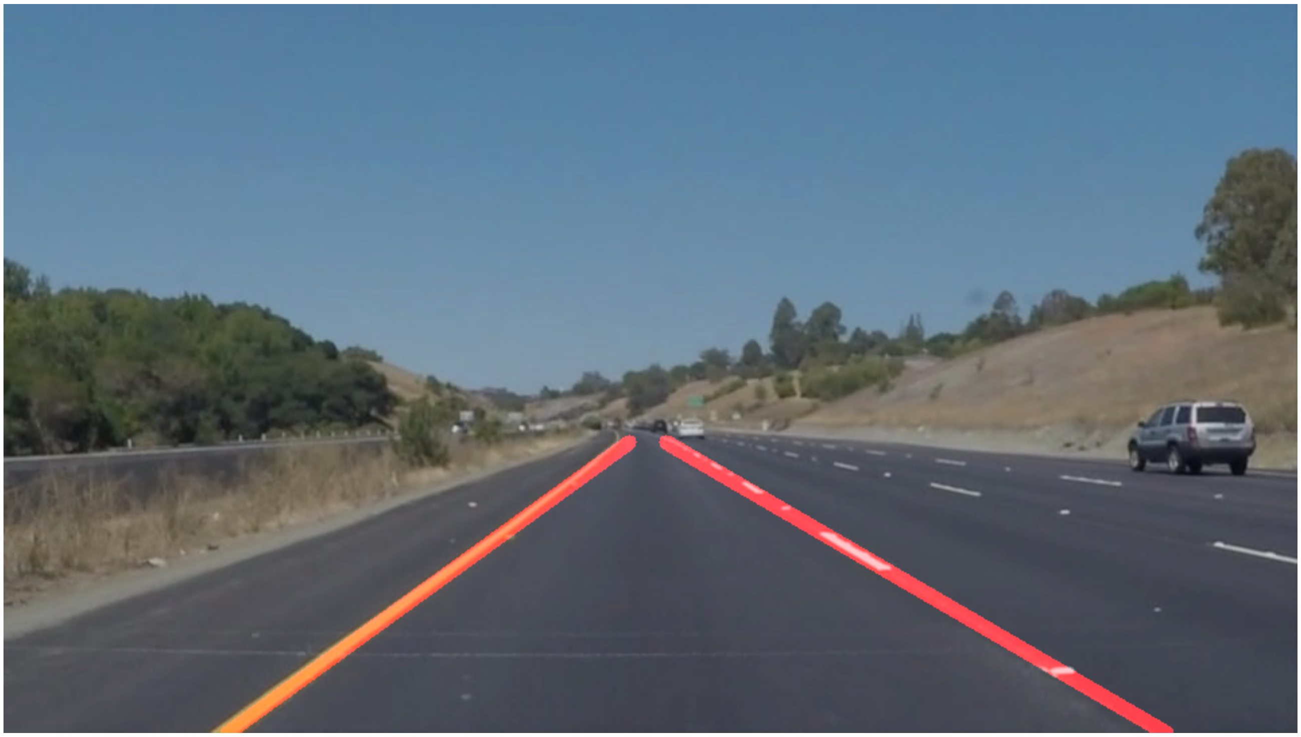
- First, I converted the images to HSV due to its advantage over detecting colors and ignoring shadows or higher lumosity areas
- I used this advantage to create 2 masks, for the two types of lines I was trying to detect: yellow and white
- I also converted the images to grayscale and I applied the 2 masks to the grayscale image
- Afterwards, I applied the Gaussian smoothing to the masked grayscale image
- I then applied the Canny filter to get the edges
- I tried to create a farily dynamic polygon and get the region of interest from the edges image
- Lastly, I applied the Hough transform to get the lines from the region of interest

In order to draw a single line on the left and right lanes, I modified the draw_lines() function by doing the following:

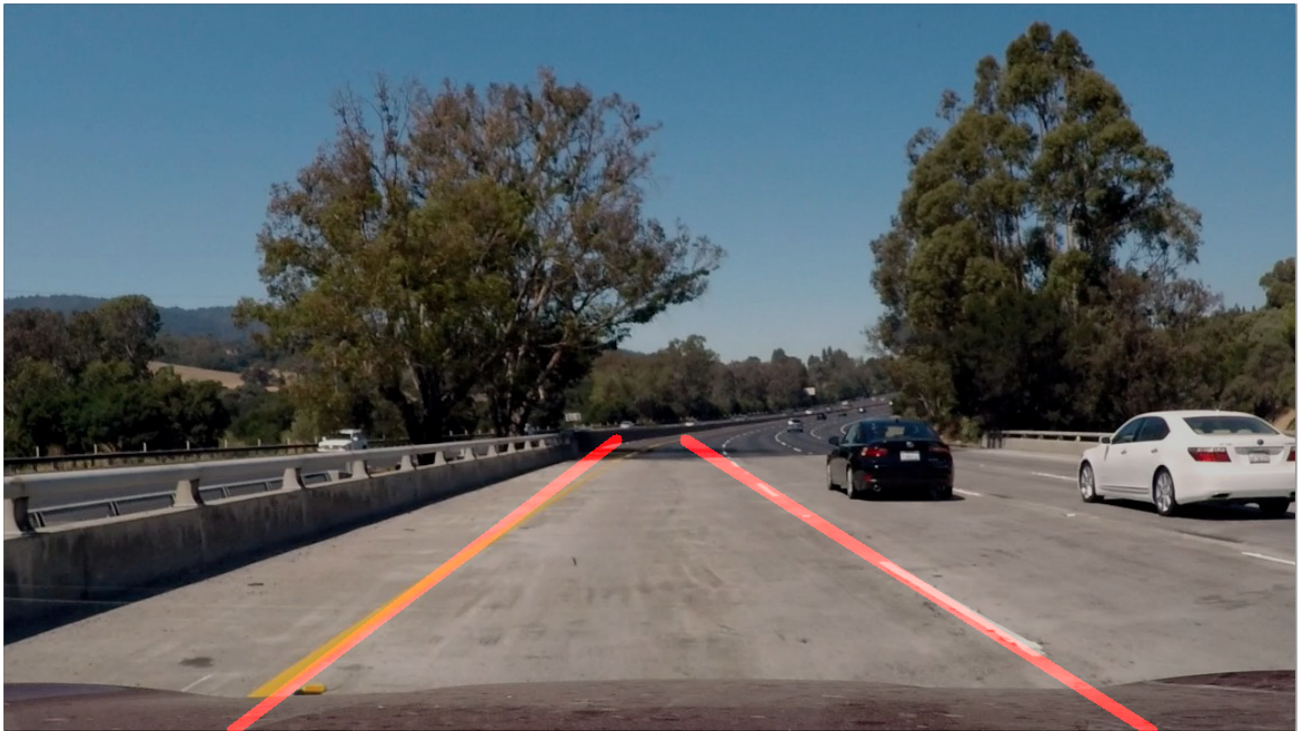
- Group the lines (returned by the Hough transform) in left or right lines, based on their slope
- Average each group (left and right) on the slopes and also on the x-intercept values
    - Use a weighted average, with the weights being the length of the lines
- Having the averaged slopes and x-intercepts, reconstruct the line coordinates

Examples of my pipeline processing images:





Here is a screenshot I took from the Challenge video, that my pipeline had a lot of trouble with initially:

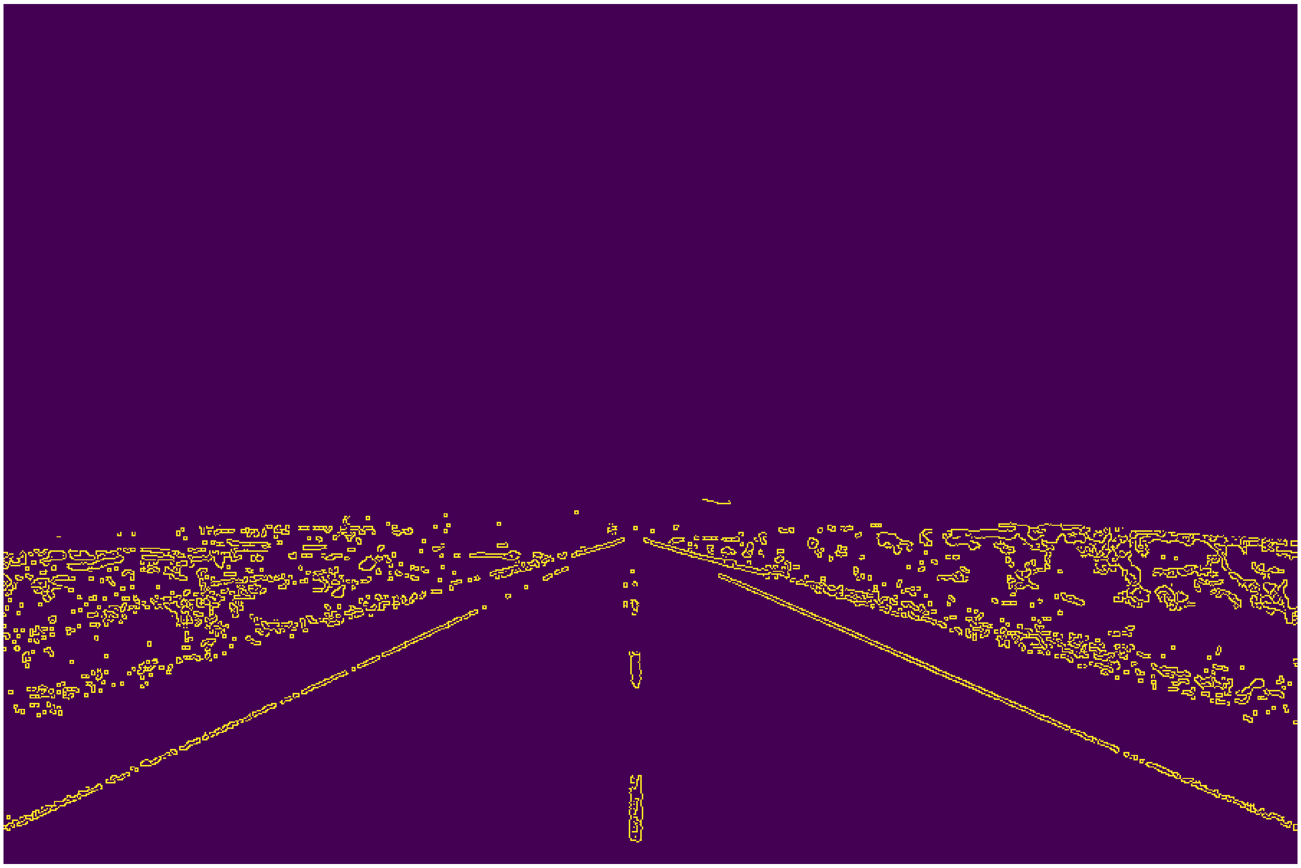


2. Identify potential shortcomings with your current pipeline

I spent a lot of time just playing around with the Hough transform parameters and I don't know if this is the right way of doing it - by that I mean that I feel that my parameters are not dynamic such that the pipeline would be able to handle any type of images.

For example, since I spent a lot of time trying to make the pipeline perfect (since I loved this project haha), I tried running the following dummy road wallpaper through the pipeline:

Only the middle line was barely detected, and I tried to understand which part of the pipeline was failing. I noticed, not surprisingly, that it was the Hough transform. I checked the RoI image and it looks very decent, with nice edges around the lines:



I also ran the same image through a version of my pipeline without the HSV yellow and white masks, and that's when I noticed the huge impact that they can have - since the image's road has a lot of white areas, just using grayscale + Canny creates tons of fake edges on the road (which can probably be filtered out through Hough, but my goal was to prepare everything before running the Hough transform such that I don't have to make the magic happen through those parameters).

3. Suggest possible improvements to your pipeline

I tried improving the stabilization of the lines, as they deviate a lot from frame to frame. I am not 100% sure how to properly achieve this (I see that the P1_example.mp4 video has pretty stable lines), but I tried using some global state of the previous left and right lines that I drew, and average the current ones with the previous ones. This kind of worked, but not totally, solidWhiteRight I decided to leave it out of the solution.

Also I would like to improve the "hard coding" of the Hough transform parameters, of course :)