

Reflection

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

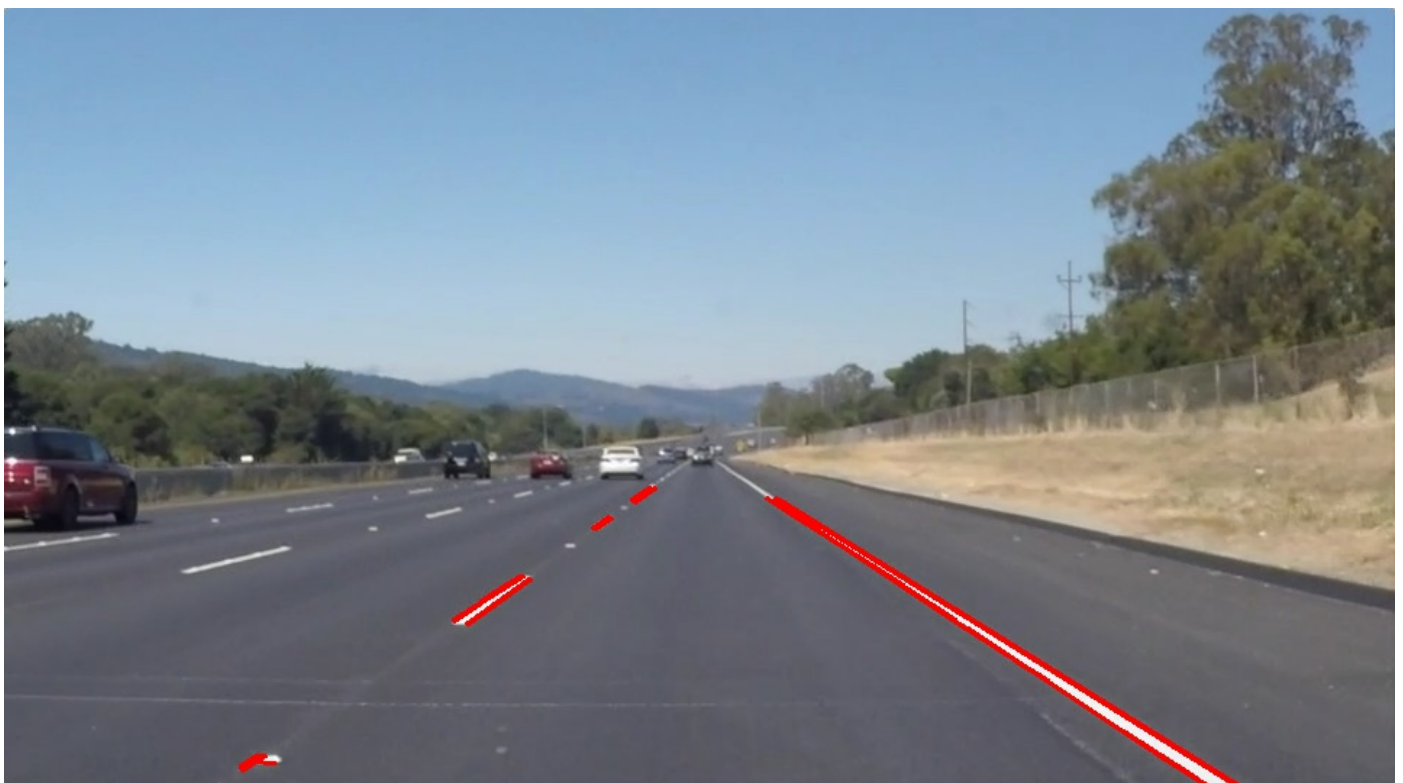
My pipeline consisted of 6 steps. First, I use gaussian kernel to clean the image. Second, I converted the images to grayscale. Third I use Canny filter to detect the lines in image. Then I use mask to extract the lines of the region I want from the images and use hough transformation to find the lines which satisfied the conditions. Then I draw the lines on the image. In order to draw good lines on the images, I have changed the `draw_lines` function in several steps:

First, I use linear regression to fit the lines. So the lines from hough transformation look good at the first glance, however when we look a little deeper, we will see that the whole line is made by many small lines which means if we use the average of the direction of the lines coordinate extracted from the hough transformation we may get wrong answers especially when we set some parameters like `threshold` and `min_line_len` for the lines. But using linear regression method can easily meet our demand because we are going to find the line direction which fits all the data.

Second, even if the lines from linear regression fit the images well, however when I put the algorithm on the video I find that there are several lines on images of the video that do not fit well. So I am trying to use history information which is the slopes of the lines in previous images to help adjust the current lines. I set the threshold to ten which means the slope of each current line is the average of the slopes of ten images.

Here are some images which show the changes of the output of the pipeline with different `draw_line` functions.

The first image is the output of the pipeline without change of the `draw_line` function and the second image is the output of the pipeline with changed `draw_line` function:





2. Identify potential shortcomings with your current pipeline

One potential shortcoming would be what would happen when the car turns on the road because the function I use to fit the lane line is a linear function which means it can only fit the straight lines well but not the curves. Another shortcoming could be the pipeline implementation with only lane line detection is not enough because there may be some obstacle on the center of the lane which can not be detected by the algorithm right now.

3. Suggest possible improvements to your pipeline

A possible improvement would be to use polynomial functions to fit the line which can be found in `numpy.polyfit`. Another potential improvement could be to use the mask to detect the obstacle on the lane.