PROJECT #1 Finding Lane Lines on the Road

0, Outline

1, My lane finding pipeline
2, Shortcomings
3, Ideas

1,The Pipe Line

Firstly, I organized the full pipeline following the course. It is:

I, Convert the image to grayscale
II, Using a gaussian filter (size equals 5 by 5) smooth the image
III, Using a canny filter to find the edge points in the image
IV, Set a triangle region select the edge points at the bottom-center of the image
V, Using hough transform function to detect lines from the edge points
VI, Drawing lines on the image

After tuning the parameters in gaussian/canny/hough function, the test images showed a good result. But when I using this function into the challenging video, It was totally not work. Then I watched the video carefully, I found the pipeline didn't work because of the horizontal edges. So I write my own filter instead of canny filter. Using a long flat filter to detect only the vertical edges. This function list as below:

```
def myfilter(gray_img,ratio):
    kernel = np.ones((1,6))
    kernel[:,-3:] = -1
    dst = cv2.filter2D(gray_img,cv2.CV_32F,kernel)
    dst = np.fabs(dst)
    dst = dst/np.max(dst)
    dst[dst>ratio] = 1
    dst[dst<=ratio] = 0
    result = np.zeros((gray_img.shape), dtype=np.uint8)
    result[dst==1]=255
    return result
```

The line in image could not able to be strictly vertical, the parameter ratio controls how much slope could we want to keep. The result of the filter like this:

Finally, I change the draw_line function to draw the extrapolation. Using the idea from the function note, we need the slope and the position of the lane. So I write a combine_lines function to make all lines grouped by its slope. The function is :

```
def find(slopes,slope):
    i = 0
    if len(slopes)>0:
        for sp_info in slopes:
            if np.fabs(sp_info[0]-slope)<0.15:
                return i
            i = i+1
    return -1

def combine_lines(lines):
    slopes = []
    if lines is not None:
        for line in lines:
            for x1,y1,x2,y2 in line:
                slope = (y2-y1)/(x2-x1)
                i = find(slopes,slope)
                #print(slope,':',i)
                if i>=0:
                    #print(slopes[i])
                    #slopes = [ [line_1_slope, number, x, y], [line_2_slope, number, x, y] ]
                    slopes[i][0] = (slopes[i][0]*slopes[i][1]+slope)/(slopes[i][1]+1.0)
                    slopes[i][1] = slopes[i][1]+1
                    slopes[i][2] = (slopes[i][2]*(slopes[i][1]-1)+(x1+x2)/2)/(slopes[i][1])
                    slopes[i][3] = (slopes[i][3]*(slopes[i][1]-1)+(y1+y2)/2)/(slopes[i][1])
                else:
                    slopes.append([slope,1,(x1+x2)/2,(y1+y2)/2])

    return slopes
```

For every line, I computed its slope, if this slope already recorded(difference less than 0.15), recompute the recorded slope by adding the weight sum of the current line, and change the position point of this group of line accordingly.

Then the final results achieved.

2,Shortcomings

The full pipeline before worked on both test video and challenging video. But Its result is far from perfect.

First, the lane is unstable. Because of the harsh parameter setting, my own filter give less points or hough line missing, result in line vanish at particular frames of the video. So the result lane will blink sometime.

Second, for the challenging video, a part of the road has higher brightness, which lead to my filter lose its efficacy. The result showed at that very period, no lane was detected.

3,ideas

For the two shortcomings, their are solutions.
First, fine tune the parameters. I know their are spaces to improve. But it's also obvious that these test video is too sample. The current parameter-settings might be the best, but I think it is robust. So the better idea could be a better pipeline, with more steps, like combine with another filter.
Second, distinguish the brightness situation. Ideally, for the selected region, if its average value has a rapidly change to a different level, changing the parameter accordingly.

The above are just ideas, for I've spent too much time on preparing the development environment, I felled behind the class. So let these ideas remain ideas now.