Finding Lane Lines on the Road

Sorry for my poor English writing, please feel free to ask me (381082014@qq.com) if you have any question or any detail that I didn't make a good explanation in this document.

If you could give me any feedback on my solution or writing errors, e.g., incorrect word or gramma, I would be very aprreciated.

---

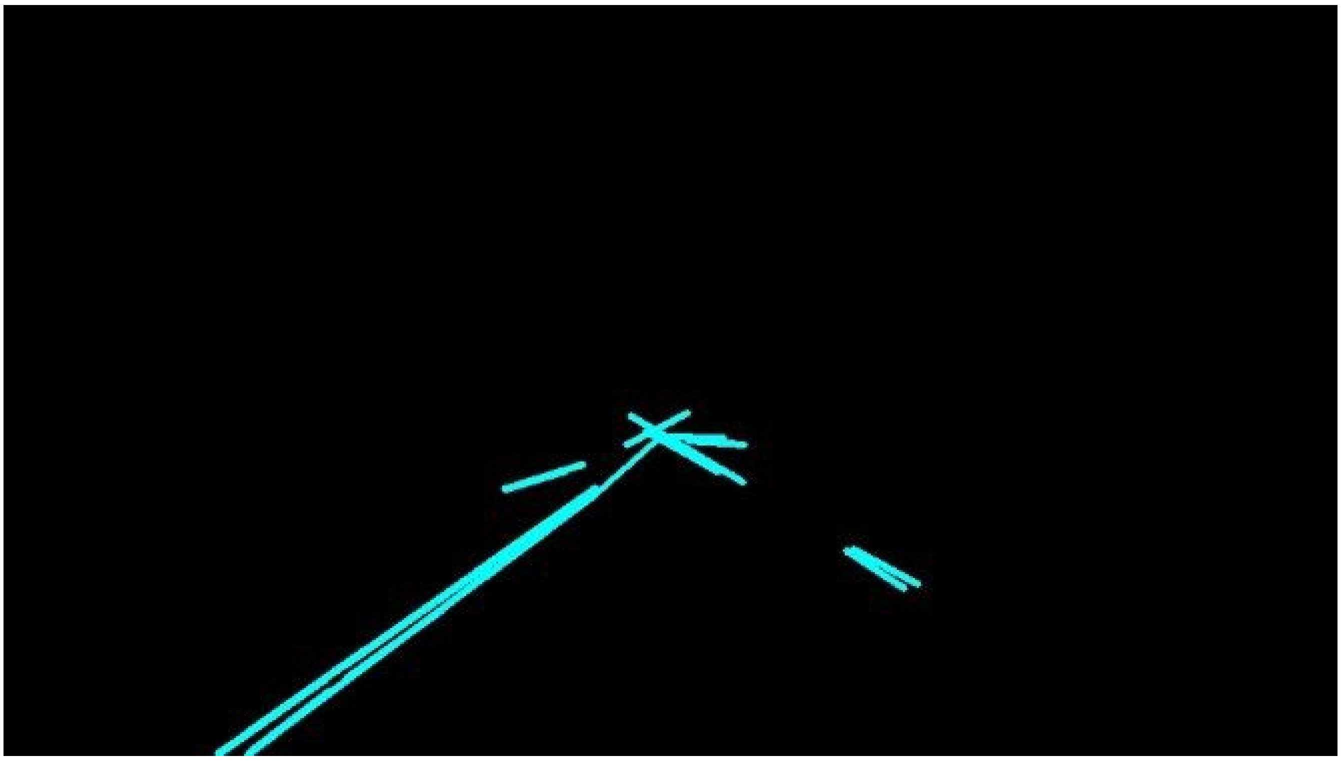
1. Processing Pipeline

An example code of pipeline

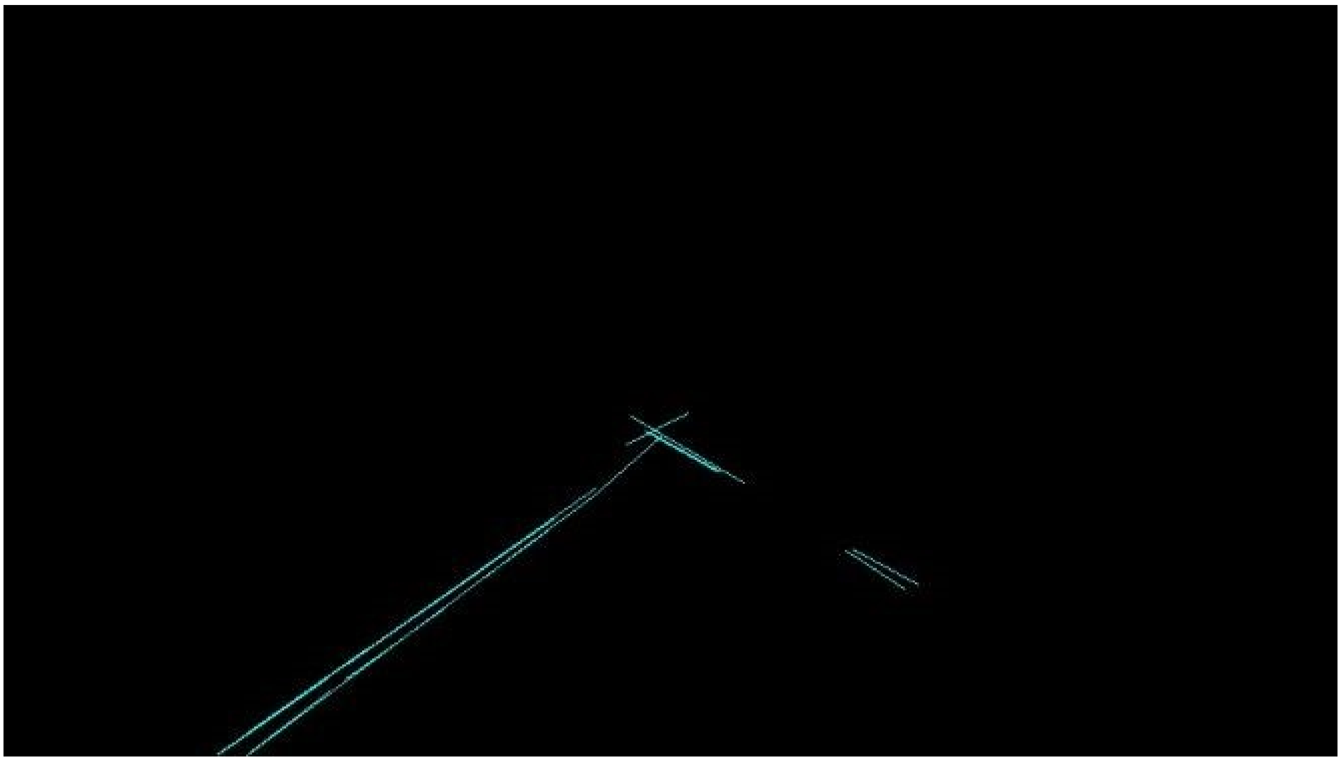
```
roi_ltrt_coeff = [0.469, 0.537, 0.510, 0.537]

img = mpimg.imread('test_images/solidYellowLeft.jpg')

img_gray = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)

mask = FindLanes.getROIMask(img_gray.shape, roi_ltrt_coeff=roi_ltrt_coeff)

edges = FindLanes.detectEdges(img_gray)
mask_edges = cv2.bitwise_and(edges, mask)

lines = FindLanes.detectLines(mask_edges)

lines, angles = FindLanes.filterLines(lines)
lane_l, lane_r = FindLanes.merge2Lanes(lines, angles, img.shape[0],
y_start=0.6*image.shape[0])

lane_img = drawLanes(img, [lane_l, lane_r])
```

My pipeline consists of 3 steps.

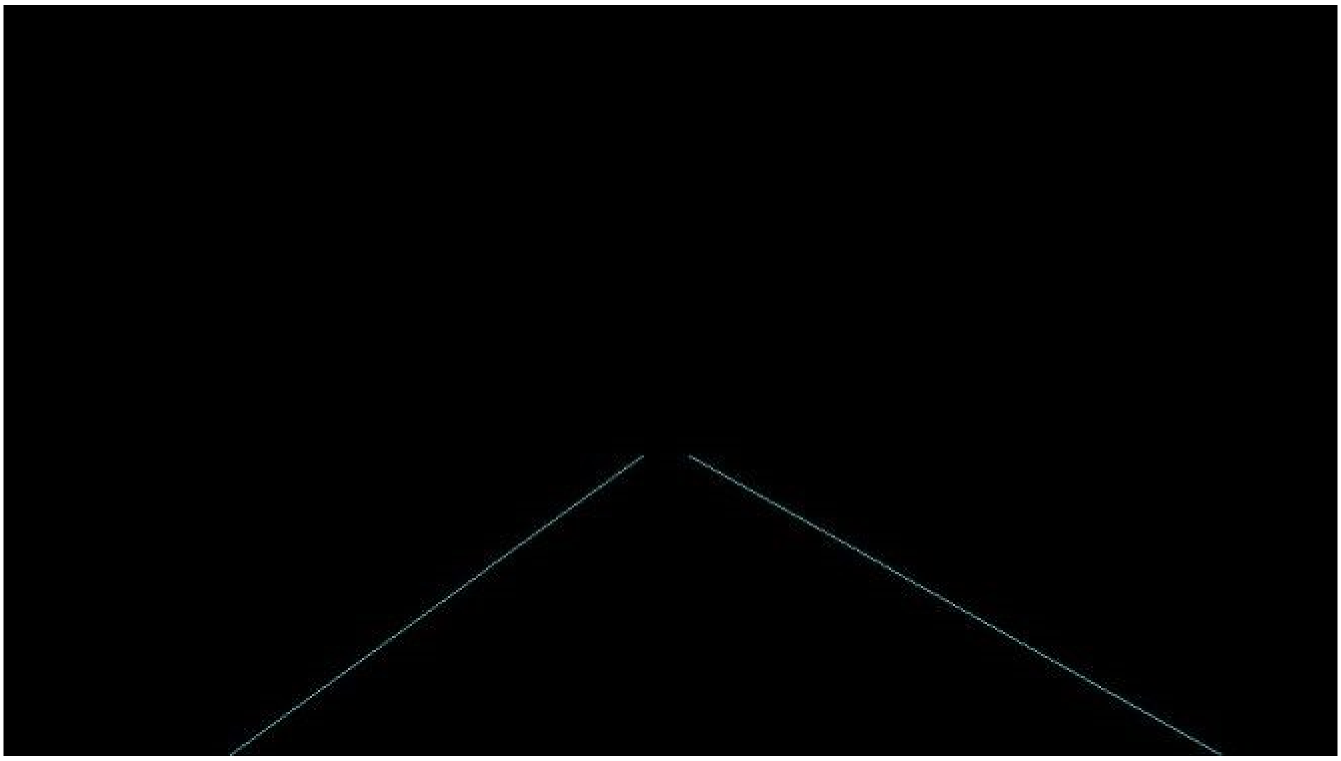
First, I convert the images to grayscale, then I set an ROI to the image in where to detect lines using Canny and Hough Transform as what we did in the course exercsie. You can see the result from the below image.



Second, I drop out lines with inappropriate angles. To do that, I only keep those lines between 25 and 155 degrees. The result becomes better although some noise lines are still remaining.



At last, in order to remove incorrect lines missed by angle filter, I developed the `remove_outlier` function. The function partion these lines into left and right clusters in angle-intersection space, where "angle" is the angle between the line and the bottom of the image and "intersection" is the point where the line intersects with the bottom of the image. In angle-intersection space, `remove_outlier` finds best center point with minimum variance for each group which may drop out several furthest points according to the parameters of `min_num` (minimum number of a group) and `min_std` (minimum stand deviation). Once I get clustered points I can calculate the mean angle and mean intersection point of the merged line that is the lane we are looking for.



2. Shortcomings

One potential shortcoming would be what would happen when there are many noise lines with good angles. These lines may escape from the angle filter and would lead to incorrect average points in angle-intersection space.

Another shortcoming could be the relations between continuous frames haven't been considered. Intuitively, the lanes of current frame should have very little difference with the lanes in previous frame. If no appropriate line could be detected on current frame due to environment rapidly changing or other reasons, previous lane detection result may be helpful.

3. Future Work

A possible improvement would be to add Kalman Filter or similar filter to get more accurate, stable result.

Another potential improvement could be to detect curved lines instead of straight lines