Assignment 3 Report

1. Problem Description and brief solution

Lane Detection is a common but useful task in image processing and computer vision. It can be utilized in auto-driving. Hence the critical issue for lane detection recognizes the correct lane on the road. I solve problems by following the main steps, and the detailed steps will introduce in part2:

(1) Generate color mask: yellow and white lane mask

(2) Canny Edge Detection

(3) Determine the Region of Interest Selection

(4) Detecting Lines by Hough Transform

2. Detailed Solution

(1) Add Gaussion filter

This a preprocess step, we adding a Gaussion blur. It will reduce noise and detail in image.

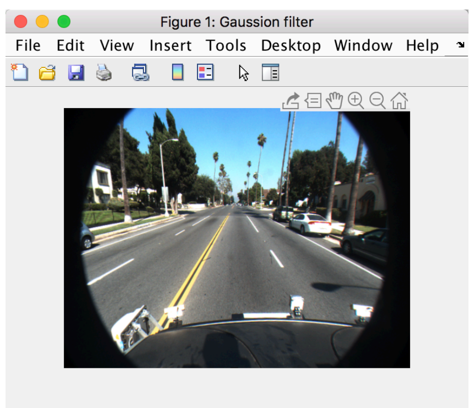

Fig1. After add guassion blur

(2) Generate color mask: yellow and white mask

For yellow mask: we set a specific color range for three channels and set the pixel to 1 if its color value in the specified range and set the pixel to 0 if its color value not in this range. This color range can be a representation of white items in the image.

For white mask: we convert image to grayscale and set the pixel values to 1 if it is in a specific color range(e.g. [120,255]) and set it to 0 if it is out of the range. This color range can be a representation of white items in the image.

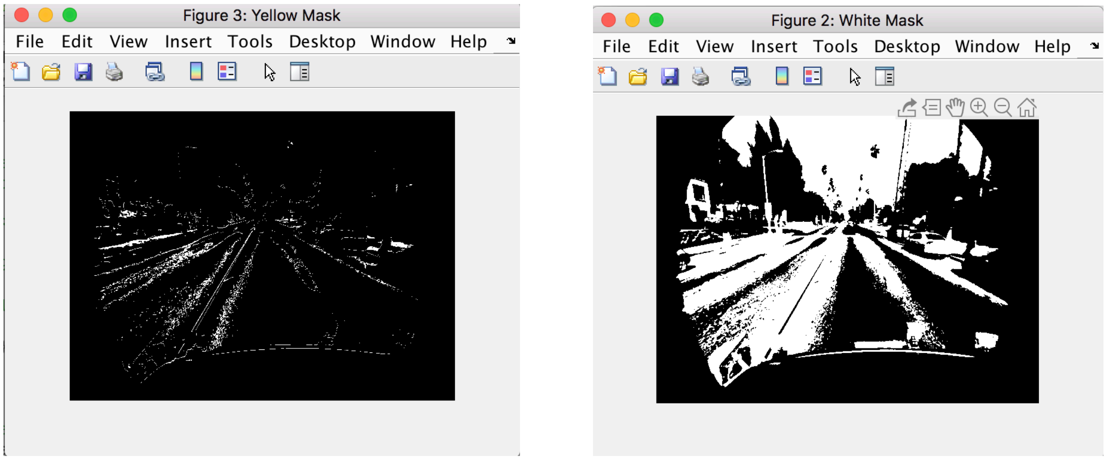

Fig2. Yellow and White mask

(3) Canny Edge Detection

Detecting candidate edges in the yellow and white mask.

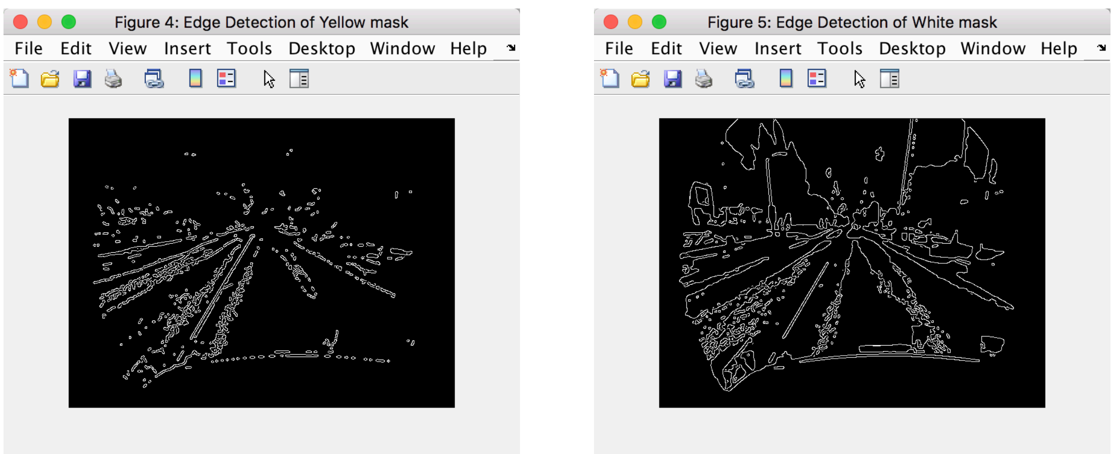

Fig3. Canny Edge detection on Yellow and White mask

(4) Determine the Region of Interest

I set two global ROI for the project. ROI for yellow and white mask will be different.

Before deciding the final ROI values, I use ginput() function to get a coarse region coordinates. Those coordinates will be the global ROI coordinates and applied in all images of two sets.

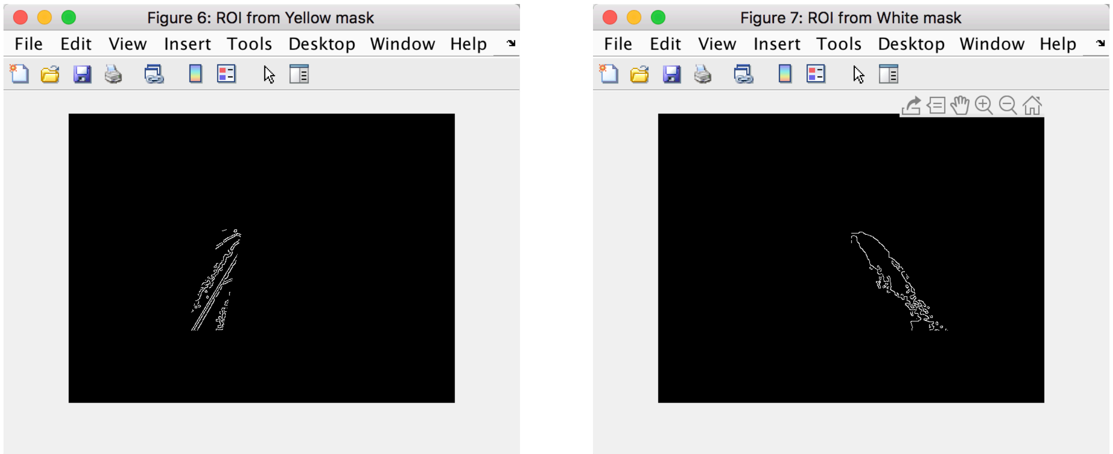

Fig4. ROI for Yellow and White mask

(5) Detecting Lines by Hough Transform

We use Hough transform to detect lanes. The line can be represented as rho = x*cos(theta) + y*sin(theta)

rho is the distance from the origin to the line along a vector perpendicular to the line. ***theta*** is the angle between the x-axis and this vector. We convert degrees to radians as the final result. Hough Transform function will generate a series pair of rho and theta as candidates.

After we compute the Hough transform, we can use the houghpeaks function to find hough peak values in candidate rho and theta. These peaks represent the most potential lines in the input image.

The blue cubes in the following figures are hough peaks.

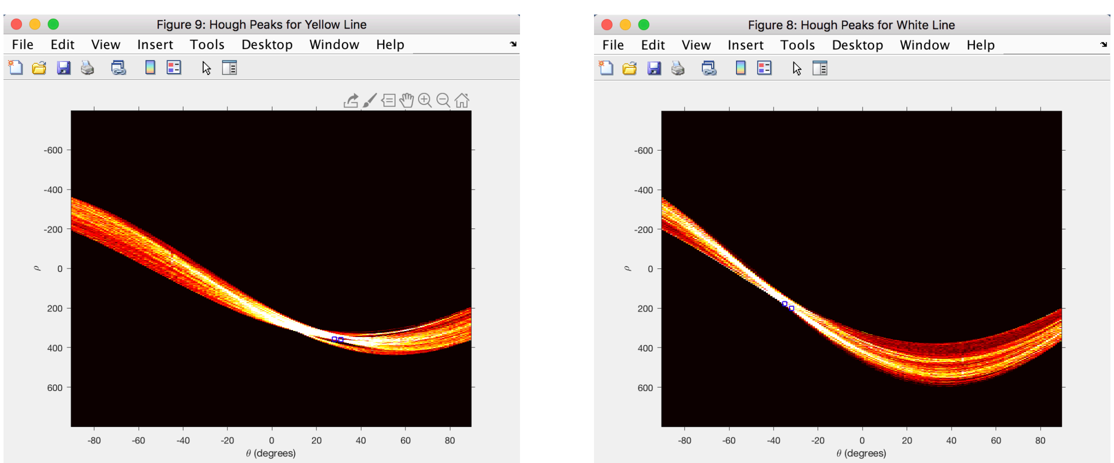

Fig5. Hough Peaks

(6) result

Fig6. Final lane detection result

3. Result and analysis

This is the best result I got is a total 1323 points for two sets.

```
Set 2 - frame 231.  Distance [18, 15, 14, 29]. Point [4], Total Points [1323]
```

4. Some difficulties in my experiments

1. Generating mask: In my experiments, I found the yellow lane is easier to create, and the generated mask is more accurate. For the white mask, it is more challenging to create a precise mask for the white lane. As my analysis, the reason is white value is more common in the entire image. Hence the noise for the white lane is much more than the yellow lane

2. ROI: It’s difficult to find a good global ROI for all images. For our test dataset, I can find a global ROI for set1 and set2. But for other test sets, ROI used in set1 and set may not fit all task sets. How to find a reliable ROI can be a critical issue for lane detection. This selection will affect the final result.