Eduardo Ribeiro de Campos - Udacity / SDCND - march/2019.

Finding Lane Lines on the Road

This report will explain the pipeline process to find lane lines in images.

The next image will be used as start point and it will support the explanation :

1. Describing the pipeline:.

My pipeline consisted of 6 steps:

Step 1: I converted the images to grayscale.

Step 2: I have applied a gaussian blur in the gray image (in result from step 1).

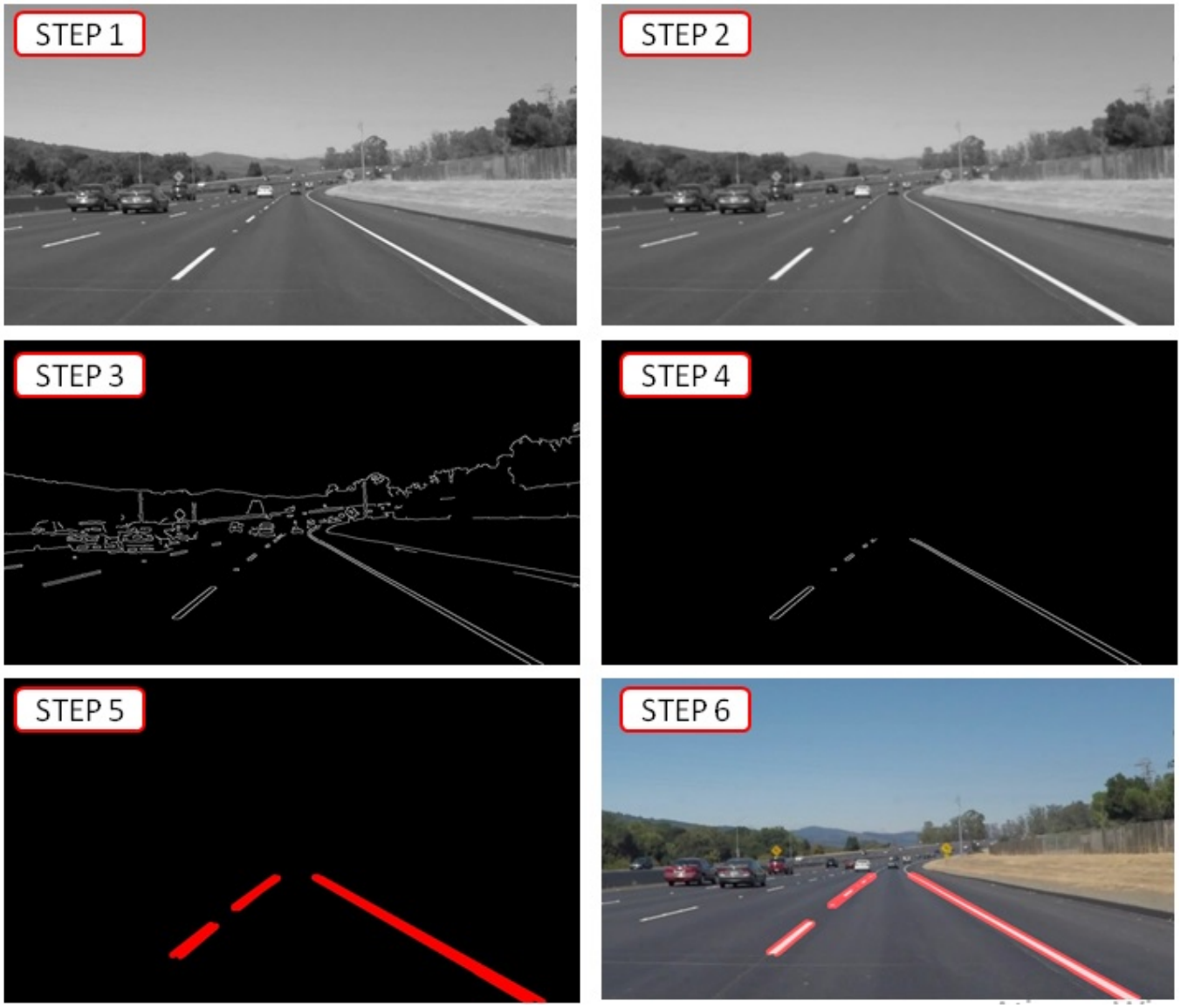
Step 3: It was used the tool canny edges detection (in result from step 2).

Step 4: I defined a mask area to keep only the lane lines (in result from step 3).

Step 5: It was created a empty picture and it was added lines using the picture obtained on step 4 as reference.

Step 6: And finally it was overlaid the result from step 5 to original picture.

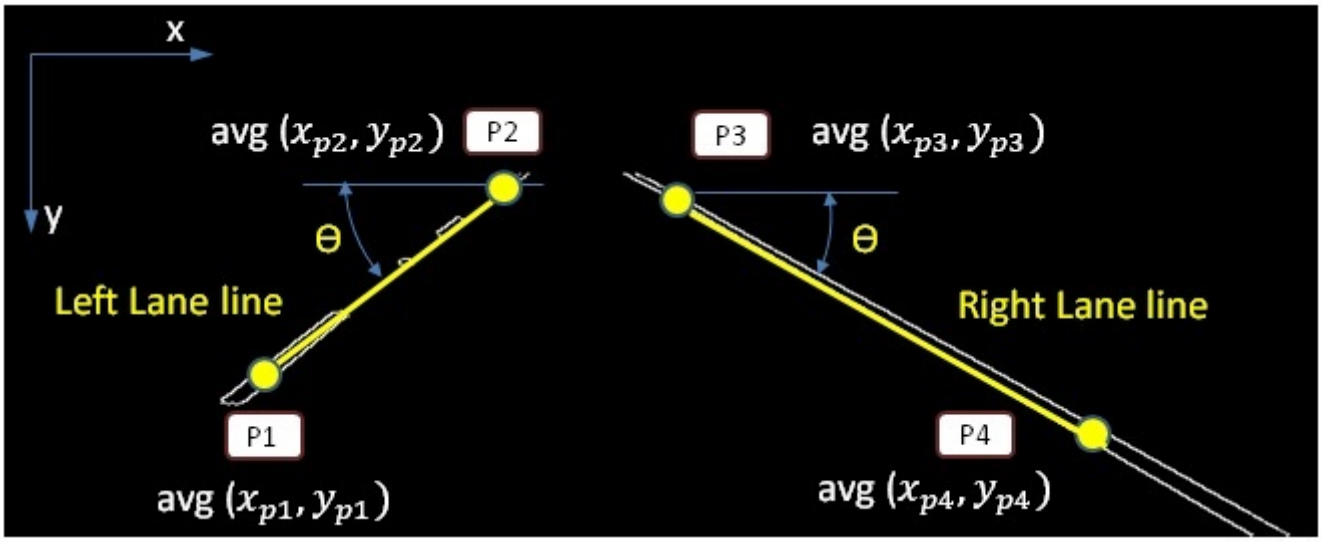
In the next picture is possible to see the result of each step.



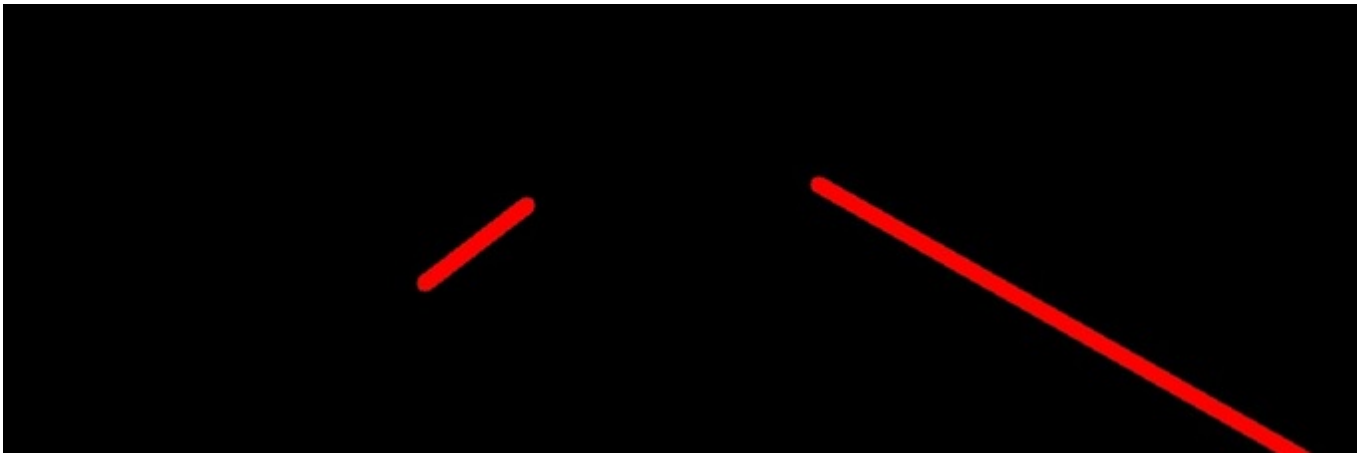
In order to draw a single line on the left and right lanes, I modified the draw_lines() function by a new a function named "draw_Extended_lines".

The strategie was use the for loop to added the coordinate points for each line inside a list, and after calculate de average for each coordinate point to create the lines. To sort the information between left line and right line the Slope of the line was used

The schematic picture below show the startegie:



The result of this strategie coul see in the next picture:



To accomplish the goals of this project, it was required extend the lines along the lane proposing new limits for both sides (left and right).

Here , using an analytic geometry concept it was created a matrix to define the equation for each line. Where:

$$\begin{bmatrix} X & Y & 1 \\ x1 & y1 & 1 \\ x2 & y2 & 1 \end{bmatrix} = 0 \qquad \begin{cases} A = y1 - y2 \\ B = x2 - x1 \\ C = (x1.y2) - (y1.x2) \end{cases} \qquad AX + BY + C = 0$$

To figure out the lower point. it was fixed the Y coordinate with the limite of the image size, And it was calculated the X coordinate. The equation below was used for both lines:

$$X = \frac{-B.\,Y_{image\ size} - C}{A}$$

The upper point for both lines is the intersection point of them. where the variables X is isolated and the equations are combined. (L = left line and R = right line)

X coordinate for upper point:

$$X_{upper} = \frac{-C_R.\,B_L + C_L.\,B_R}{-A_L.\,B_R + A_R.\,B_L}$$

Y coordinate for upper point:

starting from the known coordinate X_{upper},using the equation fron line 1, it was found the Y_{upper}

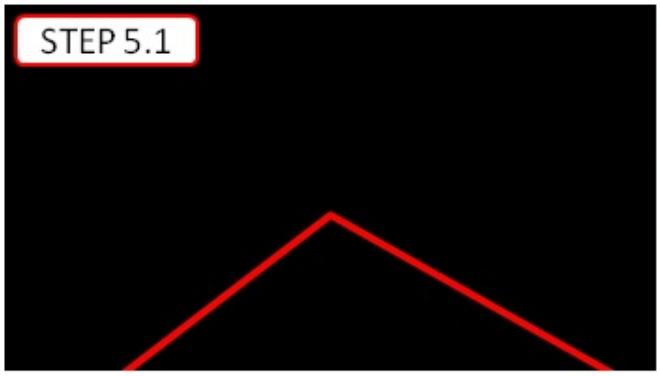
$$Y_{upper} = \frac{-A.\,X_{upper} - C}{B}$$

.

So after the function "draw_Extended_lines" already created , it was necessary create a new function named "hough_Extended_lines" where the new criteria to create lines was applied.

For the official Pipeline the Steps 5 and 6 was replaced by the new one.

The next picture show the final result.



general information about the parameter and some variables:**

Gaussian Blur Kernel: Gaussian Blur Kernel: It was used 5x5 - others values was set, but without any gain.

low_threshold / high_threshold: Used (50/150), low values show a inconsistent results . High values was not good because sometimes, the algorithm was not able tho recognize the lines. Both observation was made when the algorithm run with the videos.

Vertices for Masked edges: Coordinates created in order to isolate the lane lines.

rho,theta,threshold,min_line_len,max_line_gap: the values were combined to generate a solid result regard the lane line identification during the time of the video test.

2. Identify potential shortcomings with your current pipeline

The main Shortcoming from the pipeline was identified with the videos. Sometimes the algorithm is not able to recognize the points and the line disappear, if you set the parameter to fix the error the lines identification turn unstable.

Another shortcoming it was identified in the challenge propoused at the end of the project. This algorithm is able to work only in the straight condiction and the car must be in the middle of the lane. If the car is unsymmetrical in the lane, due the masked region one of the lines could not be identified.

3. Suggest possible improvements to your pipeline

A possible improvement would be eliminate the masked area and use filters to consider lines inside a range of angle. Another improvement that I hope could solve the Challenge propoused is work with more than 2