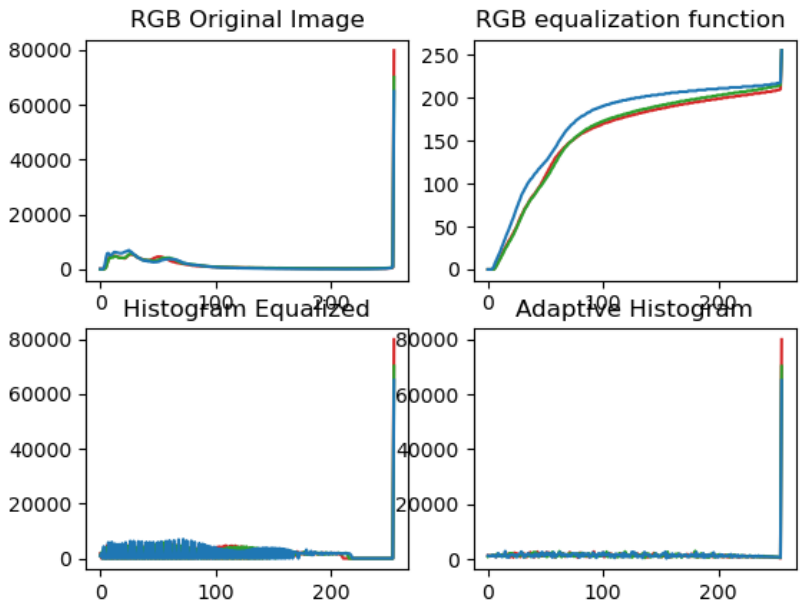
ENPM 673 Project2

Sahruday Patti

April 2022

1. **Problem 1: Histogram equalization**



The given images/images are Highly Saturated and have a smaller number of Dark Values and a large number of light Values with a very few values between these values. In order to do Image Processing we would ideally like all the regions in the image to be high contrast.

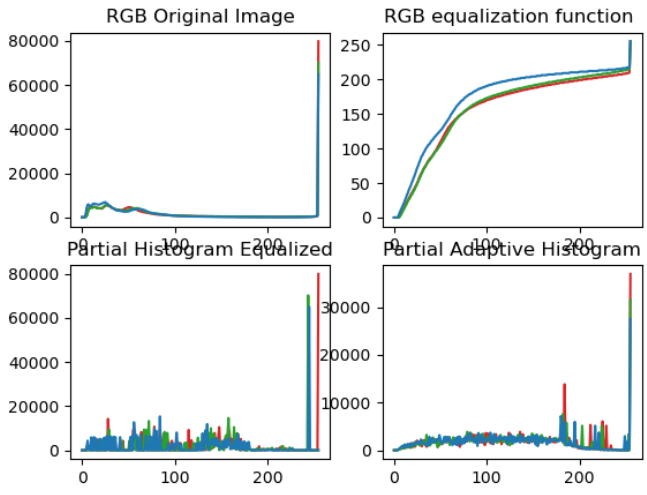
Histogram Equalization is a method through which we can simultaneously brighten some dark values and darken some light values, while still using the full extent of the available dynamic range. That is, we would boost the values in the middle range to a higher value which in turn utilizes the full dynamic range. As our image is an over saturated image, there is a loss of information so we want to spread these intensity values to the whole dynamic range to extract information hence Histogram Equalization can

be performed.

As seen in the above graph, after performing the Histogram equalization, still most of the values are high intensities because there are very few pixels in between high and low in the original image but we amplified these values by a certain amount. In the original image the dark values have a intensities like a steep curve and fall off but we have distributed this values evenly over a certain range. By this we made use of the full Dynamic range. But the resultant image does not have contrast. To Enhance the contrast we can partially compensate for the histogram unevenness,by a linear blend between the cumulative distribution function and original function. this can been seen in the below image which is 70 percent equalized with 30 percent retained from the original image.

A potential drawback of Utilizing the full Dynamic range is that we could see colors which are little bit off as we enhance the intensities in each channel this might twist the colors as we can see in the below image where color intensities are present in Histogram image but not in orginal image.Another drawback is that noise in dark regions can be amplified and become more visible.

While global histogram equalization can be useful, for some images it might be preferable to apply different kinds of equalization in different regions. Adaptive Histogram Equalization performs Histogram Equalization over a window thus considers the intensities only in the window. This gives us High contrast but the difference between the boundaries of objects in the image is accentuated as we can see in the image below.



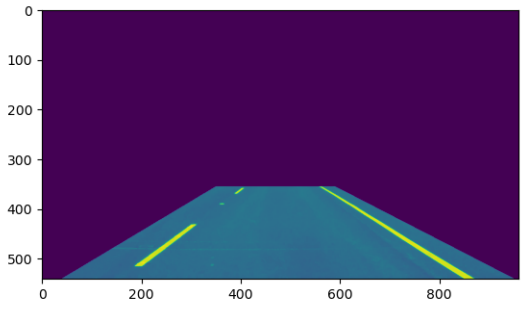
Partial Histogram Equalization

Histogram Equalized Image

Adaptive Histogram Equalized Image

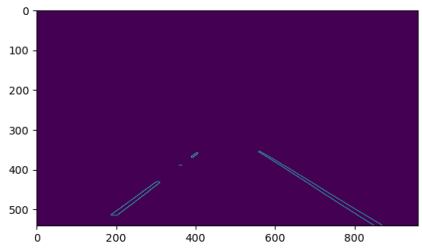
Problem 2: Straight Lane Detection

The frame is converted to gray scale first and a trapezoid is defined with coordinates on the frame such that it isolates the region of interest which is in our case the lanes on the road. This is achieved by masking the image with the polygon and filling the rest of the area with zeros so that only the road and the lanes are in the image. This masked image is then used for detection of edges.



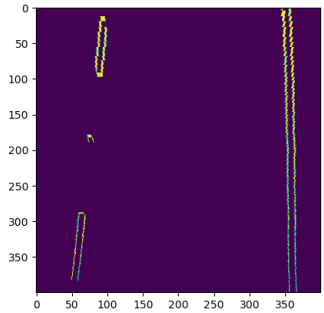
Region of Interest

Now, in the region of interest we detect the edges and threshold the image to be binary. The edge image looks like this:



Canny edge image

Now, to get a Birds Eye view i performed the Perspective transformation.The curves of the lanes are more visible from the Birds eye view.



Warped image

Once i got the warped image, i have calculated the sum of intensities column wise on the warped image. Now there are two ways to detect the Lanes based on this sum of values of the coloumn. One is to just sum the sum of coloumn values(that is all the intensities) till the mid point in the image and classify as left lane and repeat this from mid point to end of image as right lane. Now, Dashed lines intensities sum will obviously be less than solid line(Assuming we perform image operations properly and we are able to detect the lanes in the image). So if the left lane hist sum is greater or lesser than the right lane and classify them as solid line or dashed line.

Hough Lines: In an edge image, there are so many lines. which are not useful or noisy lines or incomplete lines. So to detect a robust lines in the image we make use of hough space. The hough space is a parameter space. so for a line of form $y = mx + c$ the parameter space is b,m. When we represent the line in hough space we get a point. Thus a point in the line corresponds to a point in the hough space. so how do we detect the lines? say there are two points in the image x1,y1 and x2,y2, these two points correspond to two lines in the hough space. the point of intersection of these lines will be a line joining those two points. This is how hough space can be helpful to detect lines in the image. But using this slope intercept representation of line,if there are vertical lines, the slope of the line will be infinity. To mitigate this we make use of polar representation of line. In polar representation a line is represented by a Perpendicular distance r from the origin, and theta, the angle made by the perpendicular with the origin. Now, we know that dot product of any point on the line with the sine and cosine is equal to r. so the representation becomes $r = x\cos\theta + y\sin\theta$. So now the point in image space becomes a curve in hough space. Also, in this representation if r can be both positive and negative then the theta goes from 0 to 180. Now the Hough lines algorithm is that, for every edge pixel x,y in the image, theta is 0 to 180, we calculate the r value by substituting in the line equation. here r can be both positive and negative so for every theta a value of r is calculated and this value is incremented by 1. say r range is from -100 to 100, and we get a value -99 45 times for the edge pixel x,y, the value of -99 will be 45. In this way we take the voting of the r value for a range in theta and we make use of the r where max number occurred. Now, the line is given by the r and theta in hough space. This way we calculate for every pixel and we take the point that is the intersection of all these lines as our line parameters. This line parameters will be a line that is joining the all the points that contributed more to this parameters.

In the pipeline, i used a more robust houghline algorithm called probabilistic hough lines. This algorithm gave me lines which are more useful.

I have detected lines in the left half and right half separately and from above histogram sum i have made these lines as solid line or dashed line.

Output

Output of flipped image

Generalization : As long as we can detect the lanes edges effectively, this lane detection should work as we are summing the intensity values of a binary image. If we are detecting the solid line in patches then this algorithm will fail.

Problem 3: Predict Turn

Converted each frame of the given RGB image to different color representations like HLS, LAB, HSV, etc. and separated each channel of these images and checked which of these channels give us better representation of edges in the image through out the video sequence. To find the edges i have used different types of edges detection like Sobel and canny. Canny gave me better results with fine tuning of the lower and upper threshold. The sobel operator gave me a lot of noise and didn't work with my method of histogram based edge detection hence dropped it.

Note : Although S channel of the HLS image gives me the edges in the whole Video, I have also used a gray image and performed the lane detection. The results of both of these images are explained further in the report.

Threshold Edge image of Gray channel

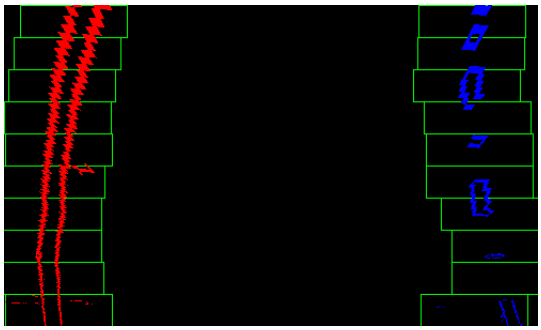
Threshold Edge image of S channel of HLS image

Once the threshold-ed image is obtained, i have manually saw the frames and detected the area of interest where the lanes lie. Then to get a Birds Eye view i performed the Perspective transformation. The Birds eye view helps us to eliminate all the other objects in the image and gives us the useful information. Also the curves of the lanes are more visible from the Birds eye view.

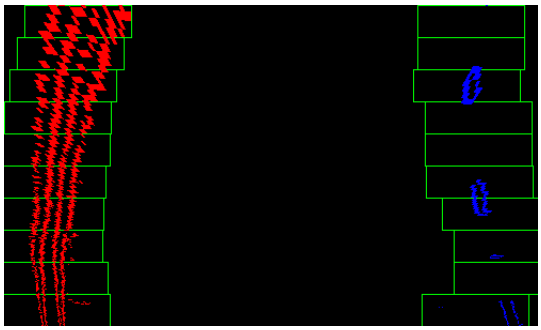
Warped Image of Gray - Birds Eye view

Once i got the warped image, i have calculated the sum of intensities column wise on the warped image. Now there are two ways to detect the Lanes based on this sum of values of the coloumn. One is to just sum the sum of coloumn values(that is all the intensities) till the mid point in the image and classify as left lane and repeat this from mid point to end of image as right lane. Now, Dashed lines intensities sum will obviously be less than solid line(Assuming we perform image operations properly and we are able to detect the lanes in the image). So if the left lane hist is greater or lesser than the right and classify them as solid line and dashed line. This method was used in the previous problem where the lanes are straight and we could draw a line directly.

Here, i have used a slightly different approach. I wanted to see how the lane is curved along from bottom of the image to the top. So i have taken a coordinates of the maximum values along the columns. max values are the lanes of our image. Now i have used a sliding window along this from bottom to top. each window is shifted to left or right depending on the mean of previous window.



sliding window Image of Gray



Sliding window Image of Schannel

Now, the schannel images is able to give me the pixels through out the video sequence. so, Now by using Sliding window i have pixels coordinates of lanes through out the video sequence. In order to fit a curve which is parabola we need a 3rd order Polynomial. So with the obtained pixel coordinates i have used cv2.polyfit function to fit a 3rd order polynomial. Now i get the coefficients of the 3rd order polynomial as return. I have used this values to fill the area between the two lanes. i have used equation of form $x = ay^2 + by + c$ which is a parabola. now the a,b,c are obtained by polyfit and we substitute all y values that is 0 to 720 and get the corresponding x location. this gives us all the x coordinates that lie on our curve in the image. Now, used cv2.fillpoly to fill the area between these lanes.Below is the image of the final result. the radius of curvature and turn prediction is explained below.

Sliding window Image of Schannel

But the sliding window technique fails when we couldn't detect the edges in the images or are lost due to lightness. So when this happens, i have used previously fit polynomials as reference and fitted a polynomial. I have used Ransac, Weighted average, average of the previous fitted polynomial coefficients. I have used a custom built ransac function to do this, so sometimes even ransac fails because of the prob˙outlier is becoming zero. so when this happens in ransac i have used a weigted average of the coefficients and performed a fit. The results are not that great compared to the schannel image. The below is the best i could predict in the video sequence. But if i hard code the frame number and directly shift to weigted average i get the better results. so i have discarded the previous 10 values of coefficients when i am performing the weigted avg.

Ransac and weigted avg final image

Now for calculating the radius of curvature, i have assumed that the whole image covers 3 meters in x direction and 30 meters in y direction. I have tried using the mean point in the image line to predict the radius of curvature but results were not good when i was trying to predict the turn. i also tried to use the max point in the curve to predict the turn but still it was not useful. Hence, converted the image coordinates to real world distances and used them to calculate the curvature.

Now once we found the left and right curvatures, to predict the turn we can see that if left curvature is greater than the right then we go right and vice versa.

In the same way to predicting the turn, i have used center of the image and the center of the lanes to see how far we are away from the center.

Generalization : I have used the same analogy as the previous, so as long as the lanes are detected the algorithms works perfectly for a curve lane as well. But if we do not determine the lane edges then use of weighted average and ransac failed to give a perfect output and hence the pipeline will not generalize.

The link to videos can be found here:

`https://drive.google.com/drive/folders/1BeEjC39HH8tttE7O_VKSNtCIZMq3zz4G?usp=sharing`