Main zip: https://drive.google.com/file/d/1VT5_Pihcd7G6d4X5_Cw-ceokUGLtJblY/view?usp=sharing

Problem 5 (15 points) What are some interesting design choices you considered and executed in creating different functions in your lane detection module? E.g. which color spaces did you use? how did you determine the source points and destination points for perspective transform?

Firstly, we optimized the parameters for the Gaussian blur with sizes 3, 5, and 7, where size 5 resulted in the best image. The Kernel size in the Sobel filter was also explored with odd sizes from 1-19 where size 1 was optimal. Additionally, we also performed color thresholding on the L channel of the HLS image. The color spaces used in our lane detection module uses the HLS color space is used for color thresholding *cv2.cvtColor(img, cv2.COLOR_BGR2HLS)*. HLS is more suited to varying lighting conditions than the regular RGB color space and this is important given the existence of trees and obstacles causing shadows on the lanes. The perspective transform used predefined source points and destination points that were determined visually by manually picking four distinct points (A, B, C, D) on the image. The quality of our 'Bird's-eye-view' image was optimized by trial and error with the chosen points A-D, where the image aims to capture as much as the road as possible with minimal stretching or tearing. Figures 1-3 display a binary representation of the input to the camera for the ROS bags.

Figure 1: ROS bag 11 simulation - binary

Figure 2: ROS bag 56 simulation - binary

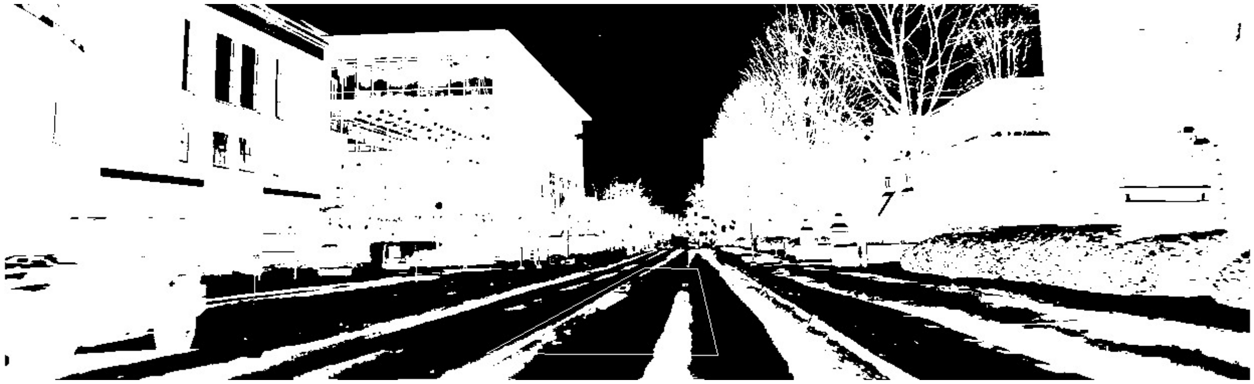

Figure 3: ROS bag 484 simulation - binary

Problem 6 (25 points) In order to detect the lanes in both scenarios, you will need to modify some parameters. Please list and compare all parameters you have modified and explain why altering them is helpful?

Figure 4: Bird's eye view

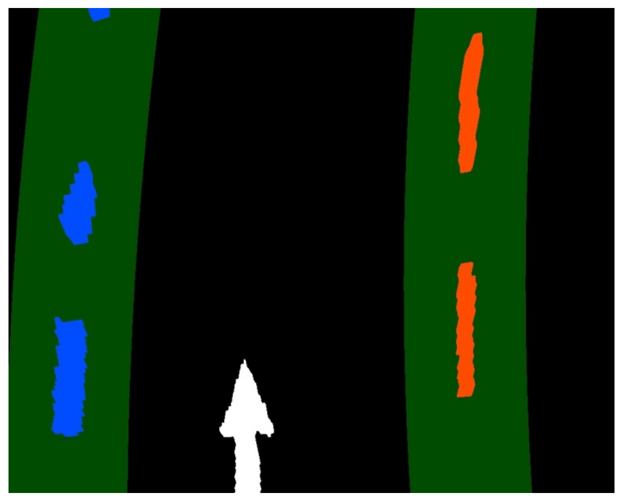

Figure 5: Bird's eye view with lane detection

The minimum and maximum gradient threshold parameters were selected by observing the output after applying the Sobel operator. The modified parameters implemented in the lane detection module help highlight the lanes clearly and suppress noise. The *color_thresh* function uses the HLS color space with a chosen threshold for the V channel. This parameter ensures the lane markers are as bright as possible and contrast well with the road for effective lane detection. For color threshold, the values in L channel for the lanes were plotted and were used in the threshold.

The source points (A, B, C, D) had to be changed between the ROS bag scenario and GEM scenario as the ROS bag simulation did not have continuous, clear markings on the road due to shadows and other obstacles relative to the GEM model. Figures 1-3 display a binary representation of the input to the camera for the ROS bags and Figure 6 shows the binary representation for the GEM simulation. The change in parameters is evident as the width of the lane detection region is expanded, this is because when the width of the lane expands when the car turns and hence the wider box accommodates for this increase.

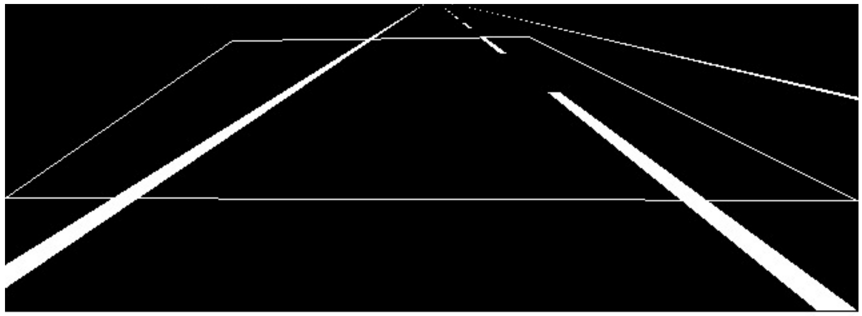

Figure 6: GEM simulation - binary

Problem 7 (30 points) Record 2 short videos of Rviz window and Gazebo to show that your code works for both scenarios. You can either use screen recording software or smart phone to record.

https://drive.google.com/drive/folders/1cDEr-sLh9aeNCJyxbyoIjhFFrcWwGmfw?usp=share_link

Problem 8 (20 points) One of the provided rosbags (0484_sync.bag) is recorded in snowfall condition. Your lane detector might encounter difficulties when trying to fit the lane in this specific case. If your lane detector works, please report what techniques you used to accomplish that. If not, try to find the possible reasons and explain them. (Note that you will not be evaluated by whether your model fits the lane in this case; instead we will evaluate based on the reasoning you provide.)

Snowfall can cover road lanes or even cause the lane detection module to mistake a streak of snow for a lane which could harm the effectiveness of the detection system. Our lane detection model is optimized for this environment as we inverted the threshold values. In other scenarios, we conducted a threshold on the white pixels, in the snow scenario we conducted threshold on the black pixels since the road is primarily white and track marks are primarily black. The threshold-based module may not work well due to the reflective nature of snow as it can reflect light and caused an unwanted increase in brightness. The variable light conditions and reduced visibility induced by snow can cause the algorithm to find it difficult to distinguish colors in low light and manage edge detection, this can introduce more noise into the image.

*Refer to snow video simulation using link provided in 'Problem 7'.