# MP 1 Writeup

**Team Members:**

Ritvik Goradia (Goradia3)

Aumkar Renavikar (aar8)

Devul Nahar (danahar2)

Toby Liang (tobyzl2)

**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?**

For color thresholding, we spent a lot of time playing around with different channels in the HSL field. Initially we focused on tuning the saturation and the hue channels as instructed in the MP document. Despite our best efforts to tune these parameters,our resulting images still contained a lot of noise. We noticed that since we were aiming to detect the white dotted lines, the luminance channel might be helpful to detect the color white without using the hue or saturation channels. We were able to eliminate most of the noise from the picture by adjusting our bird’s eye view transform to display only the dashed lines of the road.

Using the gradient field, we were able to obtain outlines of lanes in black-and-white images, which allowed detection of edges of objects such as dotted white lines in the center of the street and the solid yellow lines on the outside. Through this, we were able to also detect arrows in the rosbag-0011, but since we are dealing with objects with edges that resemble lines instead of arrows, we were able to keep our predictions to the dotted white lines in that scenario.

Since we were only working with luminescence, detecting yellow lines was much harder due to its close proximity to green’s hue. We were able to get around this by ORing the results from gradient thresholding (which easily detected the yellow lines). Finally we worked on optimizing the making the perspective transform such that most of the noise was eliminated in the final combined image passed in as an argument for lane detection.

Another design choice that we considered was the source points and destination points of the perspective transform. Since we wanted to project the source image into the same dimensions of the full image, we set the destination points as the corners of the full image. We specified these points in counter-clockwise order starting from the top-left point. To select the source image points, we used the Pixlr software to determine the specific coordinates of the dotted white lines and the solid yellow lines from the input images, which was very useful to produce close-cut cropped images for perspective transform. We ensured to choose points that excluded the horizon to create the birds-eye effect through perspective transform. We also adjusted the left and right edges such that only the lanes closest to the cars are shown to reduce the noise from other objects such as trees, other cars and other lanes are avoided.

**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?**

**Rosbag:**

Gradient Thresholds: (125, 150)

Color Thresholds: (180, 255)

Perspective Source TopLeft: (425, 225)

Perspective Source BotLeft: (400, 350)

Perspective Source BotRight: (775, 350)

Perspective Source TopRight: (750, 240)

(Destination points are top left, bottom right, bottom left, and top right of the image)

**Gazebo:**

Gradient Thresholds: (25, 100)

Color Thresholds: (100, 255)

Perspective Source TopLeft: (150, 275)

Perspective Source BotLeft: (75, 400)

Perspective Source BotRight: (500, 400)

Perspective Source TopRight: (500, 275)

(Destination points are top left, bottom right, bottom left, and top right of the image)

For the gradient and color thresholds, altering them between these two scenarios is useful since the Rosbag input images contain more shadows, colors, and noise while the Gazebo input images only have a couple of different colors (roads, lanes, grass, sky) and do not include any shadows. Further, the color scheme between these two scenarios are different in terms of the color of the roads and lanes. Thus, we needed to tune the gradient and color thresholds for Rosbag and Gazebo independently, which are listed as follows:

**Comparing Color Threshold:**

In the Rosbag tests, the main complication was detecting the lane as opposed to the surrounding sky and shadows. Since the grass in the images was quite dull in color, there was no need to filter it out using the h-channel. The luminescence threshold for the Rosbag tests was chosen to be quite high because most of the video was dull except for the white lane markings. In Gazebo, however, the colors in the video are quite bright and we also have a yellow lane we need to detect. Thus, a saturation threshold is used to isolate the lanes and a hue threshold is used to exclude the surrounding grass from our detection.

**Comparing Gradient Threshold:**

The gradient thresholds for the 2 tests are quite different because of the effects of the surroundings on the videos. In the Rosbag tests, the images/lines are quite dull and at some points are affected by shadows. As opposed to this, the Gazebo test, being a simulation, has very sharp edges and no shadows. These clear images mean we can have a much broader threshold range.

**Comparing Perspective Transforms:**

We also realized that we would need to change the perspective transform between the two scenarios since the size of the images, width of the roads, level of the horizon, and camera position would change between the two inputs. Further, the car is driving on the left side of the road in Rosbag but on the right side in Gazebo, which yields to fluctuations in the lane detection algorithm. Because of this, we designed our source points separately for the two but still kept the destination points to be the corners of the images such that the projection resulted in the whole image.

Comparing the two perspective transforms, we centered our source points more towards the left of the image for Gazebo as the car was driving in the left lane and we wanted to exclude points on the right lane that could confuse the line fit model. For Rosbag, we did the opposite in that we placed more focus on points more towards the right side.

**Problem 7 (25 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 a smartphone to record.**

**Rosbag**

https://drive.google.com/file/d/1sMLo3ZWIztIfmCCt9eVHnnsS7h0Z4DUp/view?usp=sharing

https://drive.google.com/file/d/1blVUAIpQaznVAOaFGYBQRA0RSb6FL8Ir/view?usp=sharing

https://drive.google.com/file/d/1gFSBSDDyW8QAGaXzp3Zf9i9qS9naNEpm/view?usp=sharing

**Gazebo**

https://drive.google.com/file/d/13qtynPC\_0fdUpQi0CJ\_CU9qh9C40eO2h/view?usp=sharing

**Problem 8 (10 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.)**

While our lane detection algorithm is able to detect the road, it has trouble detecting the exact boundaries of the lane since the snow causes the line fit algorithm to consider the side of the road which is filled with snow. This likely occurs because the color of the snow is the same color as the lanes in the other rosbag files, so the thresholding values we used previously would not filter out the snow and cause extra nonzero pixels during line fit. One approach to fix this would be to have different thresholds for just the snowfall scenario due to snow’s color being the same as the white line markings. If we were to make this change, we could reduce the upper bound of the color threshold, or experiment with removing the color filter entirely and using only the gradient filter to detect the lanes.

Gazebo Code: https://drive.google.com/drive/folders/1vhsM3n2YcUU8kiJCbg32\_1InyT7RoCw4?usp=sharing

Rosbag Code: https://drive.google.com/drive/folders/1nwWcWYfrSl7zwsW\_k5XYQwdYjSMJDNh2?usp=sharing