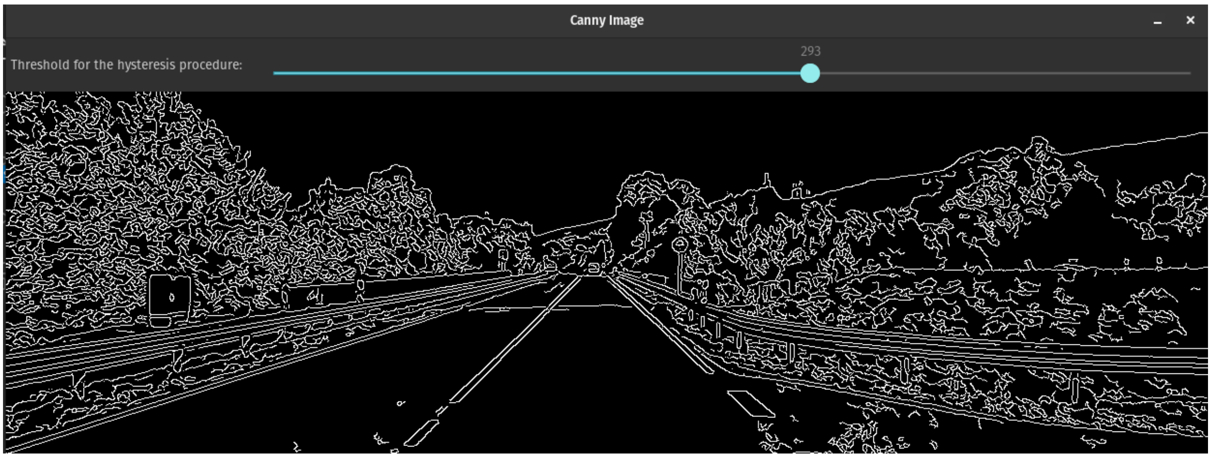
Computer Vision - Report for lab 4

Task 1

In this task I have created a trackbar in order to change the threshold value inside the Canny function. To achieve this I have used the functions createTrackbar in order to create the trackbar and setTrackbarPos in order to align the bar's position with respect to the chosen threshold value. I also have used a structure called MyValues, since an external function cannyAlgorithm is called every time the user changes the threshold value using the trackbar. I did this since the function wasn't able to access the threshold values properly. In this function, we first apply a GaussianBlur function with a kernel 5x5 and then we apply the function Canny to the image with the user's desired threshold. In the end, the filtered image is shown in a window. This is the final result:

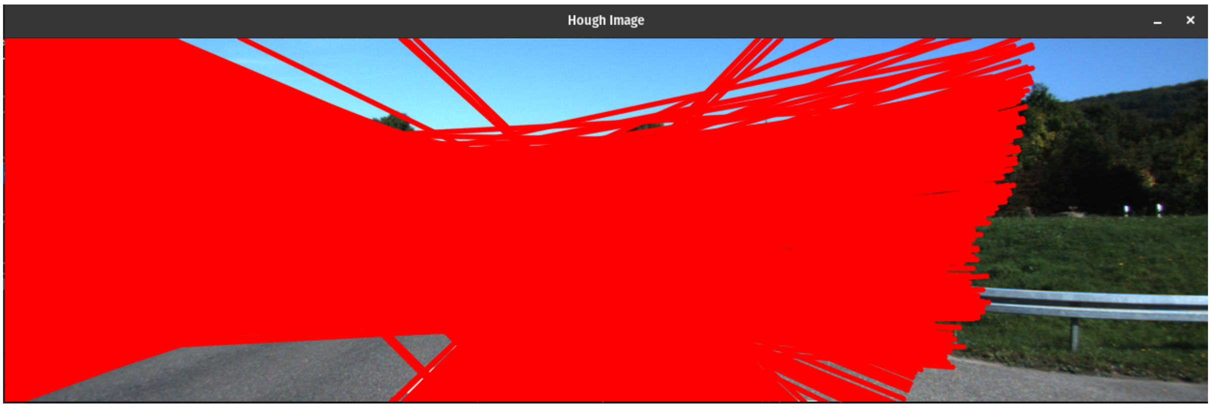


Task 2

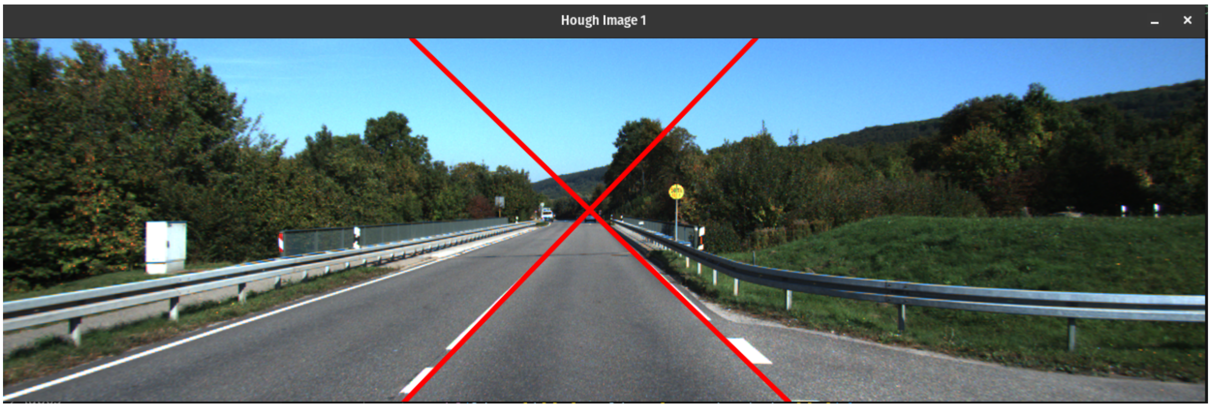
We could use the Hough Transform. The purpose of the technique is to find imperfect instances of objects within a certain class of shapes by a voting procedure. Let's consider line detection: the Hough transform converts each point in the image into a line into parameter space and intersection of these lines in this parameter space corresponds to a line in the original image. Thanks to this, the algorithm can detect different shapes like lines and circles.

Task 3

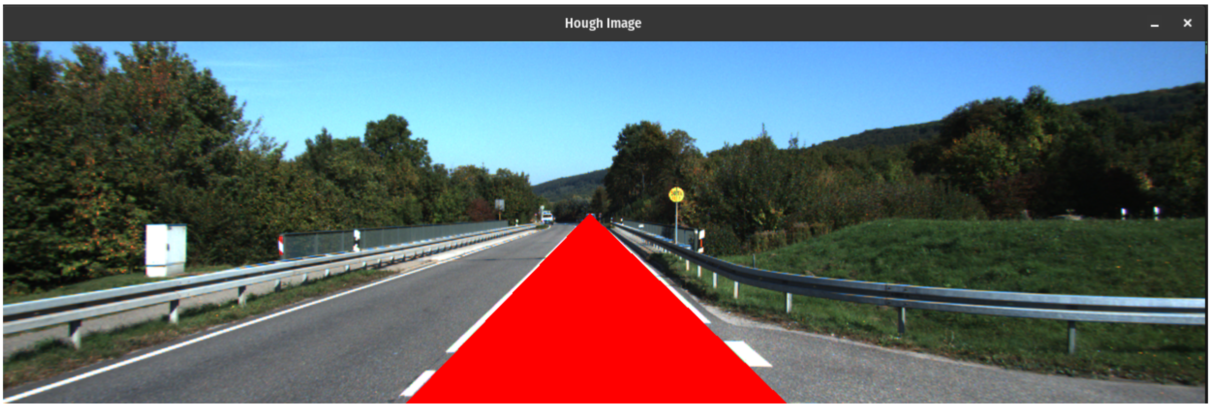
In this task we apply the HoughLines() function to the image in order to detect the road lines. This function takes in input the image processed by the Canny() function and it is able to detect the lines in it.



After tuning a bit the parameters, I realized that the road lines were the ones respectively with maximum and minimum rho values. So, I selected the two lines and printed them.

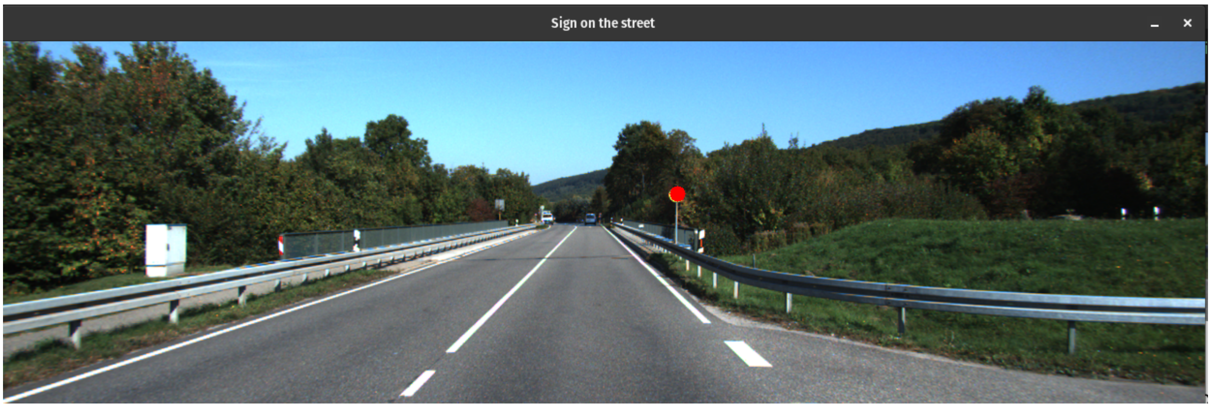


After that, I computed the angular coefficients and constants of the two lines and I colored the pixel between them. This is the final result:



Task 4

In this task we are still applying the Hough Transform, but for a different shape: circles. In fact, we apply the *HoughCircles()* function to detect the sign along the road. Before applying the function we blurred the image in order to help the identification of the circles. After tuning the parameters a bit, I have found the right circle positioned above the road sign and I have colored the pixels inside it. This is the final result:


Sign on the street