Lab 4 - Edge and line detection

Computer Vision

Alberto Pasqualetto, 2121718

14 April 2024

The original base image is shown in figure 1.

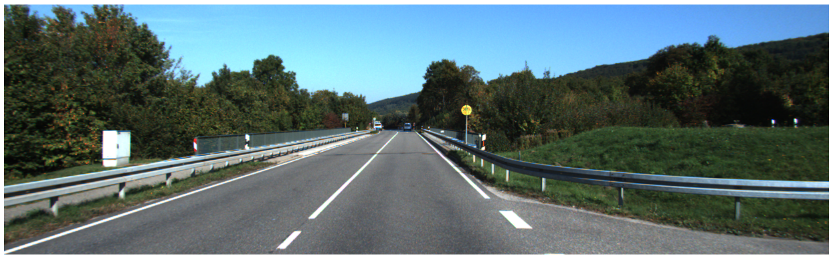
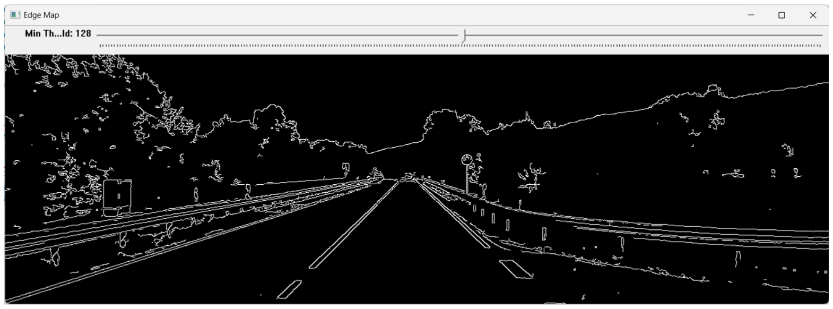


Figure 1: Original image

Task 1

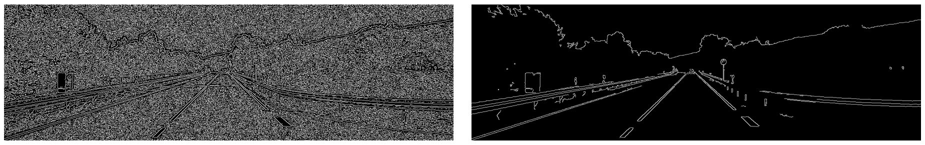
In task 1 the aim is to use the OpenCV's Canny filter in order to detect edges in an image and test it using different values of its threshold selected by a trackbar directly in the image's window as seen in image 2. The trackbar selects the lower threshold, instead the higher threshold is set to be 3 times the lower one as suggested by the OpenCV documentation.

The results of Canny filter with different thresholds are shown in figure 3: raising thresholds, less edges, the main ones, are detected.



Min Th...ld: 128

Figure 2: Window with trackbar for Canny min threshold selection



(a) Thresholds 1 and 3

(b) Thresholds 255 and 765

Figure 3: Canny filter results with different thresholds

Task 2

In task 2 it is required to detect the white lines on the road without using the Hough transform.

In order to do that first of all a mask for white color is created using the function `cv::inRange` and a very limited range. Starting from the mask, 2 points for each white line (the middle and the right ones) on the image are found by looking for the first white pixel starting from the middle of the image going left (for left line) and right (for right line) at 2 different heights (4/6 and 5/6 of the image height, selected empirically).

Finally the equation of the line is calculated using the 2 points found for each line and both lines are drawn on the image in red as shown in figure 4.

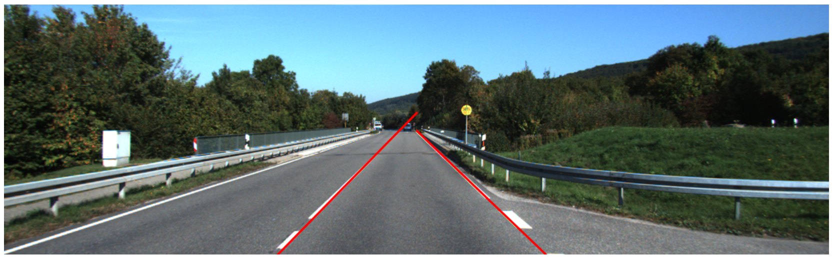


Figure 4: Lines detected without Hough transform

Task 3

In task 3 it is required to detect the white lines on the road (middle and right one) using the Hough transform and then color in red the road between the lines.

In order to accomplish this task, the following steps are taken:

1. The image is converted to grayscale.
2. The Canny filter is applied to detect edges using empirically chosen thresholds (100 and 255).
3. Hough transform is applied to detect lines in the image using the OpenCV function `cv::HoughLines` using the empirical threshold 150.
4. Keep only one line that have a slope near $\frac{1}{4}\pi$ and one line with slope near $\frac{3}{4}\pi$ to filter out horizontal and vertical lines and remain with no more than 2 lines. (Selected slopes are decided by visualizing the original image and confirmed by testing different values).
5. Draw the lines over the image of Canny filter like in figure 5.
6. Find the perspective vanishing point which is the intersection of the 2 lines at the top of the road.
7. Find a point at the bottom of each line by picking the red pixels drawn in figure 5.

   In order to ensure that the lines' pixels are present and exactly red (RGB(255, 0, 0)) at the last row of the image, the lines were drawn in the previous step using `lineType=cv::LINE_8`, without antialiasing.
8. Draw a red-filled polygon which connects the 3 points found above (realistically the points are more than 3 since those found from the bottom of lines can be more than one each since lines can be made of more than one pixel in each row, but this is not a problem for polygon drawing).

The result is shown in figure 6.

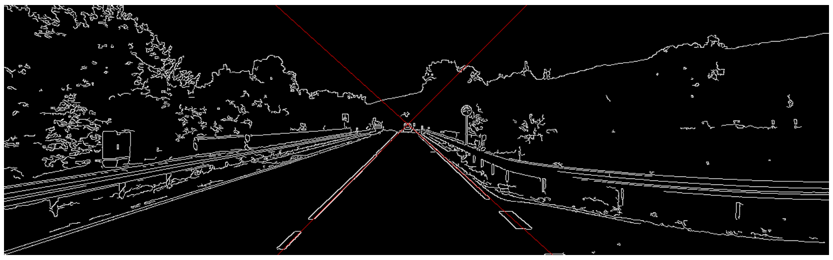


Figure 5: Lines detected with Hough transform

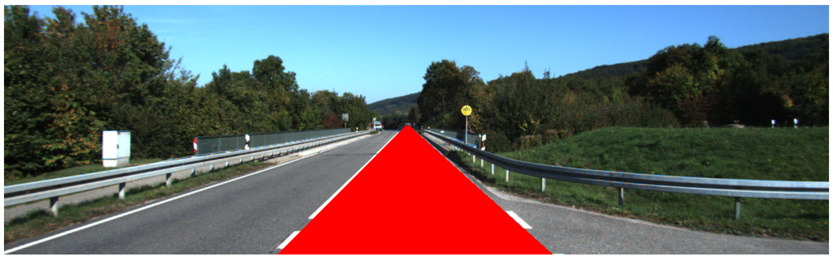


Figure 6: Road colored in red between the lines detected with Hough transform

Task 4

This task requires to detect the road sign using the Hough circular transform. First of all a gaussian blur is applied on the image to reduce noise and result in less false circles detected. Then the Hough circular transform is applied through the `cv::HoughCircles()` function which is called with the following parameters: `dp=2, method=cv::HOUGH_GRADIENT, minDist=10, param1=200, param2=30, minRadius=7, maxRadius=12`. Parameters were found empirically by testing different values. Specifically the targeted radius of the sign is about 8 pixels, so the min and max radius are set to 7 and 12 respectively in order to pick it and exclude the other circles. Results can be observed in figure 7.

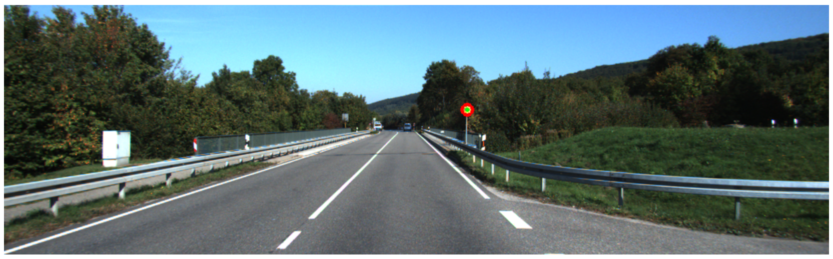


Figure 7: Road sign detected with Hough circular transform