COMPUTER VISION 2023 - LAB 3

P. Zanuttigh, G. Rizzoli, D. Shenaj

Topics: Edge detection and Hough transform
Goal: Segment the street lanes and the round street signs from the provided images

Write a *C++* program exploiting *openCV* that does the following tasks:

1. Loads an image (you can use some of the provided ones or try with your own images)

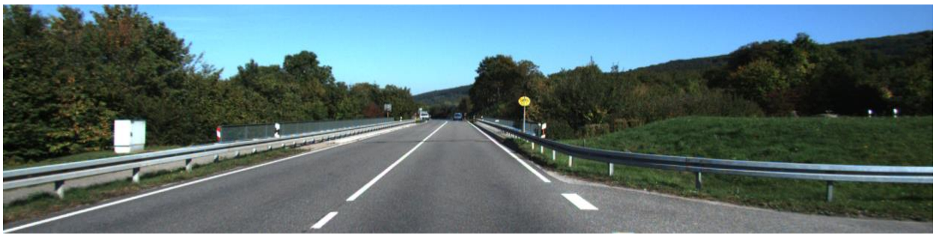
2. Colors in red the street lane (or the two lines marking the street lane) using the information extracted using the standard Hough Transform. This is the suggested pipeline:
   a. Generate the edge map with the Canny algorithm. You can use the `cv::Canny()` function. *Verify the edge detector output before going to step b.*
   b. Use the image generated in 2.a as input for the standard Hough transform (`cv::HoughLines()` function). You can use your knowledge on the image to set the parameters in order to get only the desired lines (e.g., enforce the orientations of the lines to be close to the desired ones). Alternatively, you can add some trackbars and exploit them to interactively find the optimal parameters. The returned lines should reflect the boundaries of the street or of the street lane to color. Notice that the optimal parameters depend on the image, *for this homework you can hand-tune them differently for each image*.
   c. Plot in red the two lines over the image *or (optional) color in red all the pixels between the lines (you can do manually with a for loop or use the `cv::fillConvexPoly()` function).*

3. Colors in green the pixels belonging to the round street signs. The function to compute the circles is `cv::HoughCircles().` Notice that this function includes also the edge detection step, differently from the `HoughLines` one. Again, you have to tune the parameters manually using the same suggestions in point 2.

4. *(Optional step or you can do in place of 3.)* Try to detect the crosswalks (zebra crossings). You can use a modified version of the code for step 2 that tries to locate a set of lines close together and roughly parallel.

Example of the results:

INPUT:



SAMPLE RESULT: