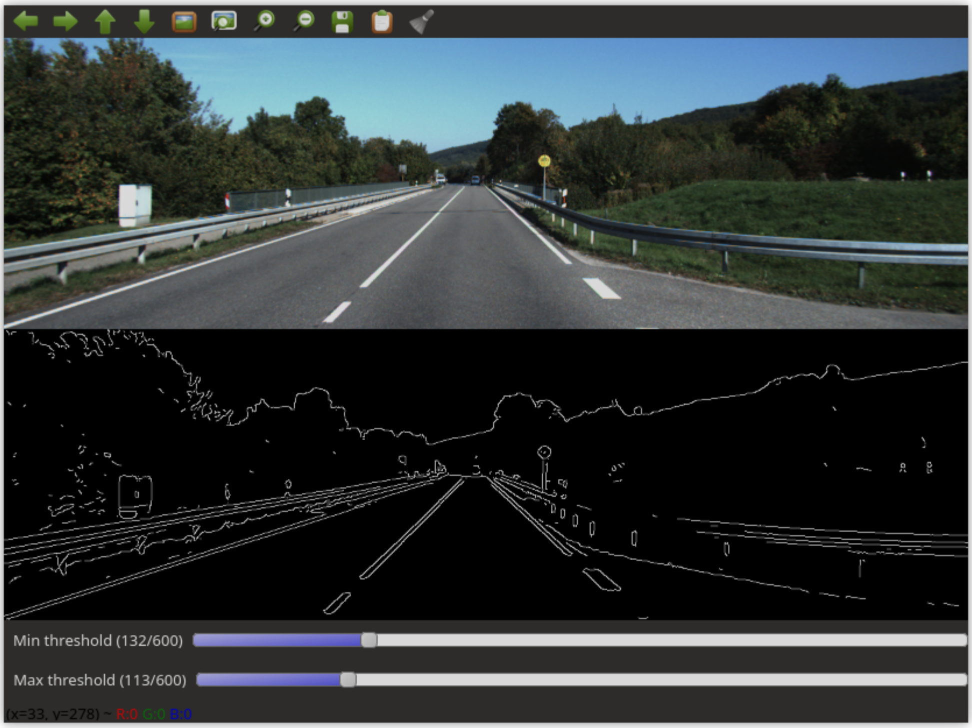
Laboratory report (Edge detection and Hough Transform) - Luca Dolci 1234008

The execution is divided in three steps:

1. Lines detection: this is performed in three sub steps:

   (a) The first thing to do is to apply a blur (in this case a Gaussian Blur, fixed parameters). This actually helps lot the edges detection, because it filters out lots of little and not weak edges, like the ones on the trees and some on the asphalt. The result is that it is possible to keep the paramters of Canny low.

   (b) The second thing to do is to apply Canny edges detection:



Min threshold (132/600)

Max threshold (113/600)

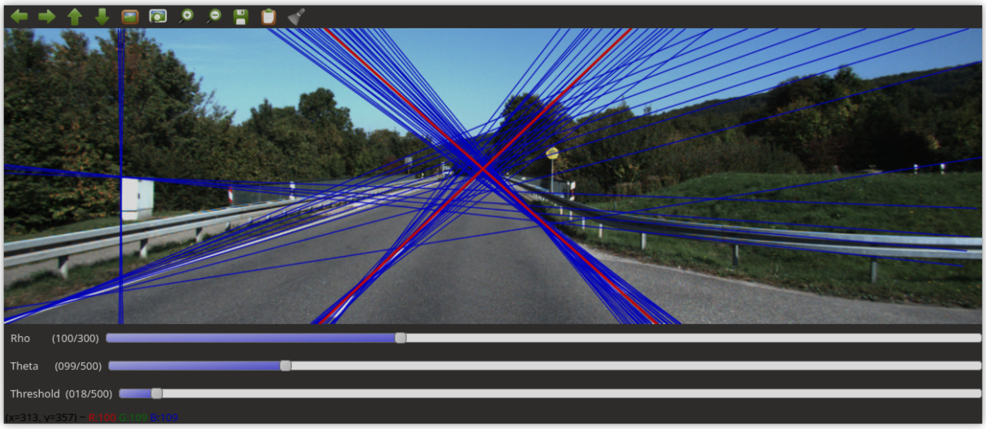
(x=33, y=278) ~ R:0 G:0 B:0

It is possible to tune two paramters:

- *Min Threshold*: this acts as a sort of erosion on the edges: when you raise this parameter you cancel out weak part of edges connected with stronger ones
- *Max Threshold*: this acts like a real threshold on edges because it deletes entire edges based on their strength, so if an edge became non so strong (by setting an higher threshold) also the weak parts of that edge (which falls between the two thresholds) are deleted

Implementation don't check if $Min\ Threshold \leq Max\ Threshold$ because it is possible to obtain interesting results. Usually the higher threshold is three times the lower one.

(c) Last step is to apply Hough Lines Transform on the Canny image:

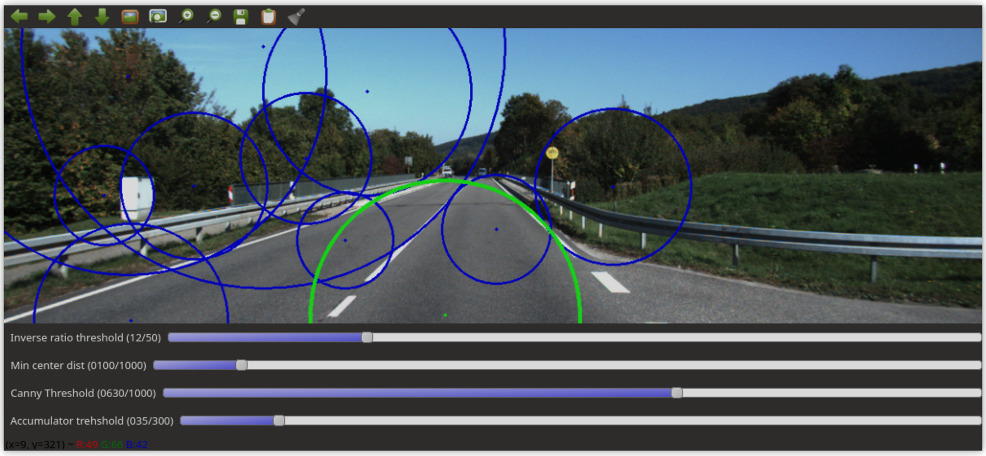


Rho (100/300)
Theta (099/500)
Threshold (018/500)
(x=313, y=357) ~ R:100 G:109 B:109

The blue lines are detected lines, the red ones is the ones with more score. It is possible to tune three paramters:

- *Rho*: resolution of ρ in pixel. Usually it is left as 1.0, but with small changes it is possible to select the desidered lines.
- *Theta*: resolution of θ in degree. Usually it is left as 1 degree but, as *Rho*, it is possible to select different lines by small changes (like switch between the guard rail and the asphalt lines).
- *Threshold*

2. Circles detection: performed in one shot because the OpenCV method also contain the Canny edges detection:



Inverse ratio threshold (12/50)
Min center dist (0100/1000)
Canny Threshold (0630/1000)
Accumulator trehshold (035/300)
(x=9, y=321) ~ R:49 G:66 B:42

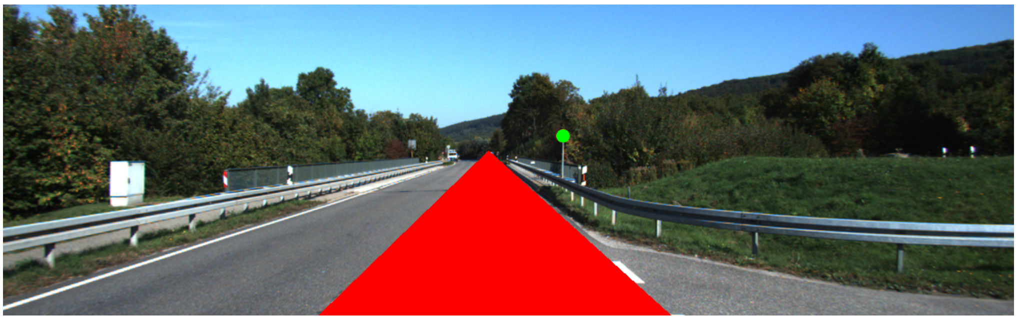
The tunable parameters are four:

- *Inverse ratio threshold*: since the selected method for detection is `HOUGH_GRADIENT` this parameter should remain in range between 1.0 and 1.5

- *Minimum dist between centers*: minimum distance between circles centes such that they are both detected: userful to filter out lots of small circles
- *Canny threshold*: max threshold during Canny edges detection, the min threshold is computed as half of this one. This parameter is very important since, fixed the previous two ones, it makes possible to perform a good detection. (As a Canny does previously with lines detection.) The idea is to find a good threhsold in step 1a and report the value here.
- *Accumulator threshold*

With some combination of values (in particular small value on the last three parametrs) it is possible to detect thousand of circles, creating computational issues.

3. The last part is about retrive the information of lines and circles and draw on screen the detected part. The final result is this one:



Note: all the parameters are pre-tuned in order to perform the better result.