Computer Engineering Master Degree

Computer Vision

Report about Lab4

Raffaele Di Nardo Di Maio 1204879

May 11, 2020

1 Goal of the experience

The goal of this lab experience was to detect street and a street signal in an image, highlighting them respectively with red and green colors. The extraction of them was done by using Canny edge detector and Hough transforms and tuning their parameters.

2 Organization of code

The code of this lab is organized in ten main files:

- **Lab4.cpp** and its header file **Lab4.h**
  they are associated to the *main()* in which an instance of *StreetDetector()* is created and its function **detect()** is called;
- **StreetDetector.cpp** and its header file **StreetDetector.h**
  these are used to create the class *StreetDetector*, which its main function called **detect()**, that creates all the windows for Canny detector and Hough transforms, creating objects of the classes that implement them. In this function, the main flow of the program is managed and this is organized in two main phases:
  1. **Detection of street**
     on screen there are two windows in the same time (Canny edge detection and Hough lines transform windows) and updating the first one, the second one is affected also by these changes. When one key, between 'q','ESC' and '\r', is pressed by user, the last image in the Hough line window is used for the next part;
  2. **Detection of signal**
     Using previously computed image, a new window is created for Hough Circle Transform;
- **CannyDetector.cpp** and its header file **CannyDetector.h**
  these are used to create the class *CannyDetector*;, that performs Canny edge detection
- **HoughLinesDetector.cpp** and its header file **HoughLinesDetector.h**
  these are used to create the class *HoughLinesDetector*;, that performs Hough transform detection for lines and highlights in red the triangle in the image, obtained by two most significant detected lines if they intersect;
- **HoughCirclesDetector.cpp** and its header file **HoughCirclesDetector.h**
  these are used to create the class *HoughCirclesDetector*;, that performs Hough transform detection for circles and highlights in green the most significant circle in the image, between all detected circles;

3 Command line parameters

The program needs to have, as command line argument, the specification of the path of the image, including also the name of the image, on which you want to detect street and signal.

4 Experimental results

In the following part there are the resulting images used in my workflow, to detect the circle and the image:

1. **Evaluation of parameters of Canny detection and Hough lines transform**
   Before applying the Canny Detection, to reduce hysteresis thresholds values, I apply a Blur filter of size 9x9. After this smoothing step, I apply Canny Detector and after Hough Lines transform. The most significant parameters of these detectors are respectively the first threshold and the angle resolution.

2. **Evaluation of parameters of Hogh circles transform**
   The most significant parameters of Hough Circles transform are max radius, if fixed min radius to 1, and the second method parameter. By changing second method parameter, the range of maximum radius, that highlights the signal, changes.

In Table 1, there are the parameters I used to obtain good detection of street and signal. In Figure 1, there is the image obtained by the program, using specified parameters at construction of objects, with detected street and signal.

Canny Detector	$threshold_1$	$[0.0, 157.0]$
	$threshold_2$	200
Hough Lines Detector	*distance resolution*	1
	angle resolution	$\frac{37\cdot\pi}{150}$
	accumulator threshold	[0,14]
Hough Circles Detector	*minimum distance*	69.0
	parameter 1	30.0
	parameter 2	23.0
	minimum radius	1
	maximum radius	$[10 - 21]$

Table 1: Estimated parameters of detectors that highlights signal and street.

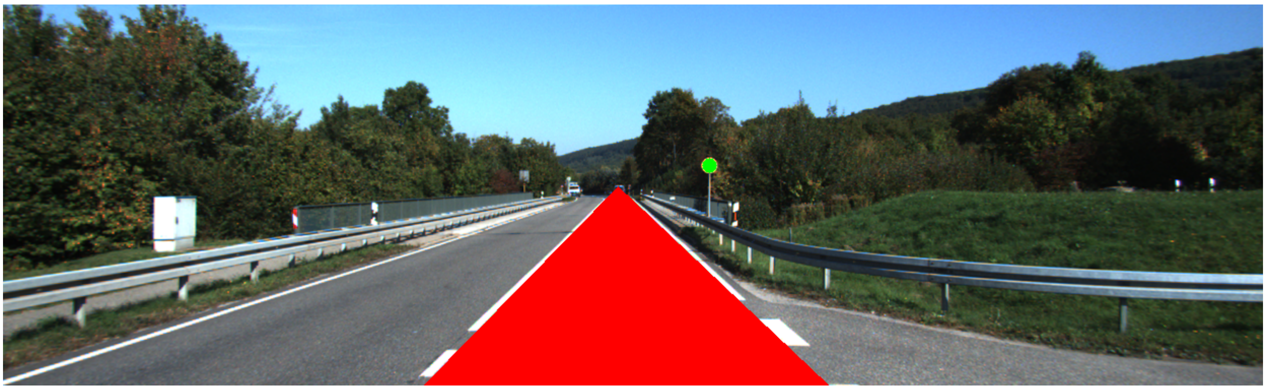

Figure 1: Image with highlighted street in red and signal in green.