

# Assignment 3

## Lines detection

Due date: April 30<sup>th</sup>, 2017

Suppose to be part of the engineering team of a car racing team. To increase your chances of winning the next race, you want to analyze the performance of your opponent's cars. In particular, you are interested to plot the speed/time curve during the acceleration from 0 to the maximum car speed.

After some YouTube browsing, you discovered that one of your opponents have a camera mounted on the cockpit showing the car speed dial during one of their tests:



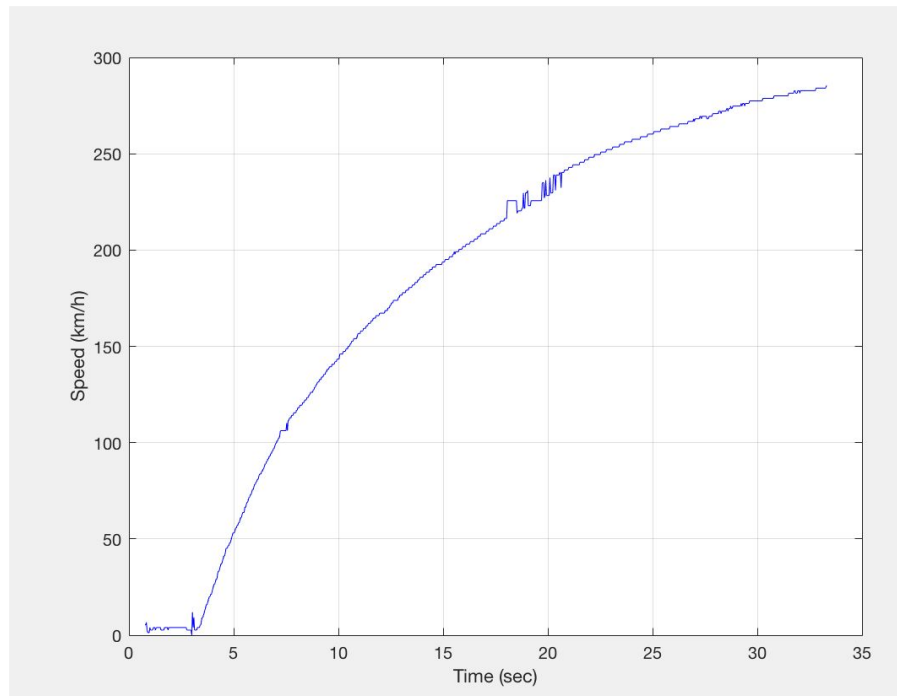
The goal is to implement a C++ program using the OpenCV library that loads the video, processes all the frames and recovers the speed at each frame from the left dial.

In particular, the program workflow is the following:

1. Load the video named "carspeed.mp4"
2. For each frame:
  - a. Isolate the pixels belonging to the left dial
  - b. Use the hough transform to recover the line parameters of the dial
  - c. Output the current frame, the time elapsed (video is at 30 frames per second), the line angle, and the car speed (the function relating the line angle with the car speed should be manually/empirically defined)

After that, produce a plot (with Matlab or Excel) showing the camera speed over time.

An example of the produced plot is the following:



For debug purposes, it's suggested to show each video frame during the processing and draw the detected line when available. Optionally, a debug video showing the detected line can be saved as output.mp4

### Notes:

There are no particular constraints on the number of function/classes/cpp files produced as long as no additional libraries are used except for OpenCV.

Comment your code whenever possible. Since no additional report is required, source comments are a good way to clarify what your code is supposed to do. Small bugs may not hinder your assignment as long as I can understand what is going on.

A clean organization of the program code is considered a plus and promotes the recycling of code for your final assignment.

Use all the filters and techniques you need to obtain the best line estimation possible. Comment each operation and parameter used.

## Additional requisites:

The following OpenCV functions cannot be used:

- `cv::HoughLines`

## How to submit

Make sure that your program compiles and works on a system with Ubuntu 16.04 Xenial or newer (use lab computers as reference). Then:

1. Remove both the dist and build subdirectories. You must not submit any executable or object files.
2. Add a README file if appropriate (example you want to clarify some of the choices you made in the assignment or some additional features you implemented). It is also a good idea to write your name and surname in the README
3. Compress the whole project directory to a zip or tar.gz file
4. Name the project package you submit as: `<name>_<surname>_assignment3.zip`
5. Submit via moodle

Please remember that the assignment has to be done individually. You are not allowed to group into teams and submit the same source code for more than one person.

For any question feel free to mail me at [filippo.bergamasco@unive.it](mailto:filippo.bergamasco@unive.it).

Filippo Bergamasco  
Computer Vision 2016/2017