Assignment 1 Lane Detection Using Classical Computer Vision Methods

January 10, 2020

Objective

To learn and apply classical computer vision techniques in application of automated driving.

Udacity's Self-Driving Car Nanodegree program has publicly released a problem that perfectly addresses this objective. The task is to identify the left and right lane marking in a vehicle's front camera view. Once identified, lines are drawn on the camera image to visualize what the algorithm believes to be lane lines.

For the assignment, it is recommended to follow the setup instructions i.e. using Anaconda and virtual environment however you are free to set up your own custom environment as long as deliverables are met.

Resources and Instructions

Environment Setup:

See the README at https://github.com/udacity/CarND-Term1-Starter-Kit

Assignment:

See the README at https://github.com/udacity/CarND-LaneLines-P1

Clone/download the repository in a location of your choice and follow the instructions in the README file.

As part of the assignment, you will develop the lane detection pipeline sequentially in a Jupyter notebook. The notebook explains all the development steps, code templates, and solution ideas. Simply follow the instructions in the README and the Jupyter notebook.

Deliverable

HTML output: In the Jupyter notebook, go to File > Download as > HTML (.html)
Submit a ZIP file containing the HTML output, write-up, and sample output videos with lines identified.

Please follow the naming convention of your zip file: a1_<user_id>.zip

Due Date

11:59 pm, Thursday 23, 2020
No late submissions will be accepted. There will be no extensions.

Marking Scheme

Assignments are marked on a 0 – 5 scale.

Please be concise in your answers. Disregard the project rubric linked in the assignment README.

0 – No submit / No answers / Irrelevant solutions
1 – Solutions are mostly incorrect, solution shows no understanding of material
2 – Solutions are mostly incorrect, solution is justified in writeup showing false understanding
3 – Solutions are somewhat correct, solution is justified in writeup showing moderate understanding
4 – Solutions are mostly correct, solution is well justified in writeup showing understanding
5 – Solutions are correct, solution is well justified in writeup showing deep understanding

Policies

Collaboration

You can discuss the problem with peers, but you must design and implement your own solution independently.

Use of online resources

You may consult online resources for inspiration, but you must develop your own code.