ENGI8814/9805- Computer Vision
Winter 2020- Memorial University

Lab 1: Hough Transform

Due date: February 10, 2020

The goal of this assignment is to write a code to use Hough transform to detect checkerboard in an image. Show me your code during the lab and if you cannot finish it in the lab then finish at come and show me your results before the deadline.

Prelab work:

- Familiar yourself with opencv functions for Canny edge detection and Hough transform
- Familiar yourself with opencv and python functions to draw points, lines, plot.
- Familiar yourself with opencv read and write images
- Read Szeliski section 2.1.1

Procedure

1. Download the zip file called lab1 and extract it to your home directory.

   From windows start menu, open anaconda command prompt and install opencv library using running the following command: pip install opencv-contrib-python
2. Open spyder for anaconda 3 as the python editor and open the sample code in Lab1.py.
3. Read the implementation of the code and understand what each part does and then run the code
4. As you have seen in different test images that results are not great and sometimes even the detected angles are not correct (try image test4.jpg). You can remove some of the false results by increasing the threshold in HoughLines function. Feel free to come up with a different idea to improve the results.
5. Find intersections of horizontal and vertical lines.
6. Extend the code to make able to run on a video. It should process every 30th frame and display the results for that frame then pause. To read video, you can use cv2.VideoCapture function.

Instructions:

Submit your codes and results as a zip file through the drop box folder for lab 1. You can upload several versions, however, the most recent submission will be the only one marked.

General guidelines for programming assignments:

The marking of the implemented functions is based on the following general scheme:

- Correctness and logic 80%: the program logic should be correct and code is compiled without errors.
- Style 10%: codes should have comments and properly formatted.
- Optimization and simplicity 10%: codes should have reasonable implementation and should not be too complicated.