Programming Assignment 2
CAP5415: Computer Vision

Fall 2019

General Instructions:

- Proper coding standards must be followed by submitting well-documented code. Code with no comments will not receive full credit.
- While any programming language is permitted, Python is preferred as the programming language. You may use any in-built IDE as a working environment.
- Please submit a report with necessary explanations and the output that you have obtained. Submit the code as well.
- While you are allowed to use any library for trivial tasks such as reading and saving an image, plotting of graphs, etc., the specified mathematical operations must be of your own effort.
- You are required to cite any materials that might have assisted you in completing this assignment.
- You are required to specify every mathematical step used in your implementation in the report.
- Making use of word or latex for report writing is encouraged. Handwritten reports that are illegible will **not** be graded. No specific format is prescribed.
- Please follow below file structure:
    - <UCFID>.zip (please zip <UCFID> folder)
        1. report.pdf
        2. code/
        3. data/
- **Please note**:
    - Not following the file structure may attract a deduction of 10% of the total points for this assignment (it implicitly means that the submitted code that doesn't run on its own will attract a deduction).
    - Missing graphs and output images in your report will attract a deduction of 1 point.
- **Late Policy:** 10% of the total points per day.

Questions:

Edge Detection:

By making use of the image given (image2.jpg), implement the Sobel Edge Detector. You may refer slides 21-24 of Lecture 4-Edges. Please be aware that the implementation of Sobel Edge Detection must be of your own effort.

Specifically, you are required to implement the following steps:

a) Write a function that implement the convolution of a 3x3 operator (or kernel) with an image.

b) Using the code from a) convolve the given image (image2.jpg) with the Sobel operator for both x and y direction as indicated in the lecture slide 22.

c) Display your output for both x and y component (call it x_comp.jpg and y_comp.jpg, respectively).

d) Compute the edge response by combining the x and y component. The magnitude at each pixel can be obtained by making use of the equation given in slide 24 of Lecture 4 (Edges).

e) Apply Thresholding as suggested in slide 24 of Lecture 4 to get the final edge-map.

Corner Detection:

Implement the Harris Corner Detector and show its operation using the given input images (image_hcd1.jpg and image_hcd2.jpg),. You may make use of slides 10-28 of Lecture 5-Features. You are expected to implement Harris Corner Detection on your own by making use of both the images.

Below steps are to be followed:

a) Implement the Gaussian Filter **G** to apply smoothing to the images.

b) Make use of edge detector to find image derivatives along x and y direction.

c) Compute the matrix **M** at candidate detection points as suggested in slide 19. Please note that the **W(x, y)** is just a smoothing function.

d) Implement the measure of cornerness **R** as stated in slide 21 and apply thresholding as given in slide 22, and state your observations in your report.

e) Call the output image as hcd_output.jpg.