COMP 425/COMP6341 Computer Vision (Winter 2023)

Homework 1

Due Date: Feb 14 at 3pm Montreal time

Note

- You are encouraged to meet with other students to discuss the homework, but all write-ups and codes must be done on your own. Do not take notes from those meetings. You should know how to work out the solutions by yourself.
- Please acknowledge other students with whom you discussed the problems and what resources (other than the instructor/TAs, lecture notes, and textbook) you used to help you solve the problem. This won't affect your grade.
- For programming questions, please make sure your code is clearly commented on and easy to read. If the marker cannot understand your code and it does not run correctly, you might not be able to get any partial marks.
- Follow the instructions exactly. We reserve the right to refuse to grade the homework or deduct marks if the instructions are not followed.
- Use your grace days wisely.
- Make sure your code is sufficiently optimized. In particular, do not use unnecessary loops. You do not need any additional loops for questions in this homework.
- Some of the questions require setting certain parameters. You should play with these parameters to find suitable values.
- **DO NOT import any additional Python package in your code. The goal of this homework is to ask you to implement certain algorithms from scratch**.

Part 1 (Image Filtering)

1. (2 points) Implement the missing code in the function "filter2d" in "utils.py". This function performs 2D image filtering (NOT convolution). You can assume the input image is grayscale (i.e. having only one channel), and the height/width of the filter is an odd number
2. (4 points) Implement the functions "partial_x" and "partial_y" in "utils.py". Given an image, these two functions calculate the gradient image along the x and y directions respectively. You should use a 3*3 derivative filter in each function**. In your report, write down the filter you used in each of these two functions.**

Part 2 (Edge)

3. (6 points) Fill in the missing code in "edge.py" to implement part of an edge detector (see the comments in this file for specific steps). Here you will use the functions in "utils.py" to find the x gradient, y gradient, and the gradient magnitude on each pixel. Apply your code on the image "iguana.png" and visualize these 3 gradient images (gradient image on x direction, gradient image on y direction, gradient magnitude). **Put these 3 figures in your report.**

Part 3 (Corner)

4. (6 points) Fill in the missing code in "corner.py" to implement the Harris corner detector and apply it on the image "building.jpg". Note that the Harris detector involves 3 major steps: (1) computing the "corner response" map on each pixel; (2) thresholding on the response; (3) non-maximum suppression (NMS). You can use the function peak_local_max from skimage (already imported) to perform NMS by selecting the peak local maximum on the response map. There are certain choices you can play with (e.g. whether to use Gaussian weighted derivatives) in your implementation.

Generate 3 figures. The first two figures visualize the response map after (1) and (2), respectively. The 3rd figure visualizes the detected corners after non-maximum suppression by putting an "X" at the corresponding coordinates in the image. **Put these 3 figures in your report**.

Part 4 (Image Subsampling)

5. (4 points) Fill in the missing code in "downsample.py". This code consecutively downsizes the input image ("paint.png") to a factor of 2 (i.e. reducing image size from 200*200 to 100*100) for 5 levels. This code implements both the naïve downsampling (with aliasing) and the one without aliasing. Put the results of the naïve downsampling as the 1st row (i.e. 5 subplots on the 1st row), and the results from anti-aliasing downsampling as the 2nd row (i.e. 5 subplots on the 2nd row). **Put the final visualization figure in your report**.

How to Submit

Please create a document in PDF format (name it as hw1.pdf) containing the following:

- Your name, student ID, email address
- Relevant plots/figures requested in this homework. Make sure you organize and label the plots/figures appropriately, so that it is easy for the makers to find the relevant plots/figures for a given question.

Put your codes (including data files and others that are provided as part of the assignment) in a folder named lastname_firstname (replace with your last and first names). Create a ZIP file (name it lastname_firstname.zip, again, replace with your last and first names accordingly) containing the top-level directory.

For example, if I were to submit the homework, I would create a ZIP file named "wang_yang.zip". After running "unzip wang_yang.zip", I should get the following:

wang_yang/

wang_yang/utils.py

wang_yang/edge.py

wang_yang/...... (other files)

If I run "python edge.py" in the "wang_yang" folder, I should get the plots in the report from the relevant question.

Go to this course in Moodle, then click "HW1 submission" (under "29 January - 4 February"module). Submit the following two files: 1) the PDF document; 2) the zip file containing your codes (see above). The submission server is configured to only accept PDF and ZIP files. If you try to submit other file format, the submission server will not accept it. So make sure you know how to create PDF and ZIP files ahead of time.

Note that you can make multiple submissions. But we will only consider the last submission. We will use the time stamp recorded on Moodle to calculate the late days you have used for this assignment.