UNIVERSITY OF LIVERPOOL

COMP338 – Computer Vision – Assignment 1

- This assignment is worth 15% of the total mark for COMP338
- Students will do the assignment **individually**.

Submission Instructions

- Send all solutions as a single PDF document containing your answers, results, and discussion of the results. Attach the source code for the programming problems as separate files. (One PDF doc, one source code file {python or Jupyter Notebook (Ipython)})
- Each student will make a single submission to the Canvas system.
- **The deadline for this assignment 14/11/2024, 5:00pm**
- Penalties for late submission apply in accordance with departmental policy as set out in the student handbook, which can be found at
  http://intranet.csc.liv.ac.uk/student/msc-handbook.pdf
  and the University Code of Practice on Assessment, found at
  https://www.liverpool.ac.uk/media/livacuk/tqsd/code-of-practice-on-assessment/code_of_practice_on_assessment.pdf

Task 1. (50 marks) Canny Edge Detection

OpenCV provides a function canny() to get the edge detection result with an image (you can use any grey image). Please do the following:

1. (25 marks) Reimplement the canny operation **without** using the built-in canny() function (with some explanations of the code).
2. (10 marks) Test and visualize your implementation results. (with different filters, different thresholds and others)
3. (15 marks) Discuss the difference between your implementation, your results compared with the OpenCV implementation. (Compare the numerical results and the running time and others.)

Note:

- It is acceptable if the implementations do not match exactly; you will need to analyze the differences between your implementation and the Canny method. Including your own reflections in the report will result in additional bonus points. However, it is mandatory that you reimplement the function based on your understanding.

Task 2. (50 marks) Feature Extraction

In **Lecture 11** and **Lab 04 - SIFT & Feature Matching**, we have discussed the SIFT feature. In practice, there are several other feature extraction methods such as SURF or ORB. In this task, we will do extra reading, implementation, and compare SIFT vs. SURF vs. ORB.

Papers to read:

- Bay et al., **SURF: Speeded Up Robust Features**, ECCV 2006
- Rublee et al., **ORB: An efficient alternative to SIFT or SURF**, ICCV 2011.

Good tutorials:

- https://docs.opencv.org/4.x/df/dd2/tutorial_py_surf_intro.html
- https://docs.opencv.org/4.x/d1/d89/tutorial_py_orb.html

1. (20 marks) Read the SURF and ORB papers and tutorials, summarize your understanding. Compare the differences among SIFT vs. SURF vs. ORB.
2. (10 marks) Given two images (victoria.jpg and victoria2.jpg – both available on Canvas), call OpenCV functions to extract ORB keypoints. You can use the built-in functions from OpenCV. Visualize the detected keypoints.
3. (20 marks) Given two images (victoria.jpg and victoria2.jpg), extract the descriptors using SIFT and ORB. Perform keypoint matching using Brute-Force Matcher. From the results, which method do you think perform the best? Justify your answer.

Note:

- You can also choose the images yourself, as long as they are of the same subject taken from different perspectives. You may directly use greyscale images.
- Including your own reflections in the report will result in additional bonus points. However, it is mandatory that you reimplement the function based on your understanding.