

## ECE661: Homework 4

Fall 2020

Due Date: Sept 28,2020

Turn in typed solutions via BrightSpace. Additional instructions can be found at BrightSpace.

### 1 Introduction

In Homeworks 2 and 3, you manually picked pixel coordinates to estimate transformations between two images and for metric rectification. In this homework, you will learn to implement an automated approach for interest point detection and correspondence search for a given pair of images of the same scene.

For the detection part, you will implement your own *Harris Corner Detection* algorithm and test the SIFT or SURF implementation that are available in OpenCV. You will then use these features obtained from your own Harris corner detector or OpenCV's SURF or SIFT algorithm, to establish the correspondences between the two views. You will implement your own functions for computing the SSD (Sum of Squared Differences) and the NCC (Normalized Cross Correlation) as the feature similarity measures.

Consider replacing inefficient nested 'for' loops with Numpy vectorized operations. Note that you also have one theoretical question for this homework.

### 2 Theory Question

What is the theoretical reason for why the LoG of an image can be computed as a DoG. Also explain in your own words why computing the LoG of an image as a DoG is computationally much more efficient for the same value of  $\sigma$ .

### 3 Programming Tasks

#### 3.1 Task 1

For this task you are given three image pairs per scene, first image from each pair is shown in Fig. 1. The complexity of each pair varies from clouds

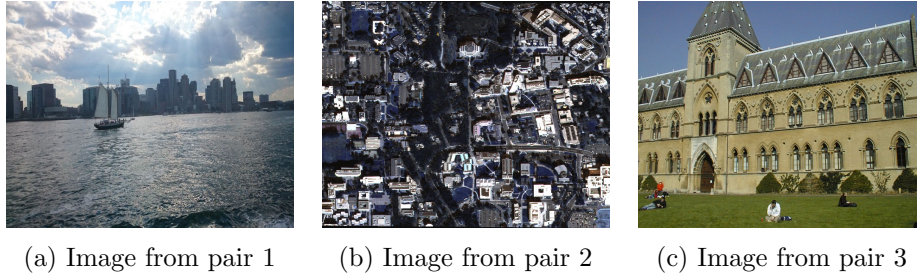


Figure 1: Input pairs for Task1 (only first image is shown)

and water noise in Fig. 1a, multi-date satellite views of 1 sq. km region in UCSD campus in Fig. 1b, to building facade in Fig. 1c.

Compare the quality of the correspondences obtained from the following two sub-tasks and state your observations.

### 3.1.1 Harris Corner Detector

1. Extract interest points using your own Harris Corner detector implementation. You can refer to the last section of Lecture 9 and the sample solutions to the previous years homework for understanding the steps involved.
2. Use NCC and SSD metrics to establish correspondences between the two sets of interest points of the image pairs.
3. Apply the Harris corner detector for at least 4 different scales. Your implementation should allow for any suitable scale as input.

### 3.1.2 SURF or SIFT

Use the OpenCV implementation of either SIFT or SURF algorithm to find interest points and establish correspondences between images. In this case you can directly compare the feature vectors of interest points to establish correspondences.

## 3.2 Task 2

Show results for Tasks 3.1.1 and 3.1.2 on at least two image pairs taken with your own camera.

### 3.3 Notes

1. For efficiency purpose, it's recommended to replace nested 'for' loops by vectorized operations.
2. Draw lines to indicate the selected correspondences.
3. You can use the OpenCV SURF or SIFT functions for the second task [3.1.2](#).

## 4 Submission Instructions

Include a typed report explaining how did you solve the given programming tasks.

1. Turn in a zipped file, it should include (a) a typed self-contained pdf report with source code and results and (b) source code files (.py, .cpp,.c). Rename your .zip file as hw4\_<First Name><Last Name>.zip and follow the same file naming convention for your pdf report too.
2. Your pdf must include a description of
  - Your answer to the theoretical question in Sec. 2
  - A description of your implementation of the Harris corner detector and an overview of SIFT or SURF algorithm.
  - A description of how you used the NCC and SSD metrics to establish correspondences with relevant equations.
  - The input and output images for each task. **Clearly show the output correspondences by drawing lines between the interest points.**
  - Your observations on the interest points detected at various scales, using the Harris Corner Detector.
  - Your observations on the output quality and performance of each approach (Harris vs SIFT or SURF) and the NCC and SSD similarity measures.
  - The parameters that you chose for best feature extraction and matching.
  - Your source code. Make sure that your source code files are adequately commented and cleaned up.

3. The sample solutions from previous years are for reference only. **Your code and final report must be your own work.**