ECE661 Fall 2024: Homework 4

TA: Rahul Deshmukh (deshmuk5@purdue.edu)

Due Date: Midnight, 23 Sep 2024

Late submissions will be accepted with penalty: -10 points per-late-day, up to 5 days.

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 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 do the following:

1. Implement your own *Harris Corner Detection* algorithm.
2. Test the SIFT or SURF implementation that are available in OpenCV.
3. Test the CNN-based SuperPoint [1] interest point detector.

And to establish the point-to-point correspondences between the two views, you will do the following:

1. Implement your own functions for computing the SSD (Sum of Squared Differences) and the NCC (Normalized Cross Correlation) as the feature similarity measures.
2. Test the GNN-based SuperGlue [2] feature matching network.

Lastly, note that you also have one theoretical question for this homework.

2 Deep Learning Based Interest Point Detection and Matching

In this section, we would like to give a brief overview of the neural network based interest point detection and matching pipeline.

For interest point detection, we will be using the fully convolutional **SuperPoint**. SuperPoint is comprised of a shared encoder and two decoder heads – a interest point decoder and a descriptor decoder. SuperPoint determines locations of interest points with the output of the interest point decoder, which represents a probability map of "point-ness". The corresponding feature descriptors are given by the descriptor decoder, which outputs a dense map of fixed length descriptor vectors.

After interest point detection, we subsequently use **SuperGlue** to establish the optimal correspondences between the two sets of interest points from the two views. SuperGlue consists of two major components: an attentional Graph Neural Network (GNN) and an Optimal Matching Layer. First, the GNN jointly aggregates the interest point locations and descriptors from the two views into more powerful descriptors that can be used for matching. What is interesting is that such aggregation is done using repeated self- and cross-attention layers to take both intra- and inter-image context into account, respectively. Subsequently, with the context-aware descriptors, SuperGlue solves a linear assignment problem to find the optimal matchings between the two sets of interest points.

You are strongly encouraged to read the SuperPoint [1] and SuperGlue [2] papers yourself for more details. Also, note that you will be performing only inference with pretrained weights in this homework so GPU is not required to produce results within reasonable time.

3 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 σ.

4 Programming Tasks

(a) First image pair: Maheshwar Temple, Indore, India (b) Second image pair: Hovde Hall, Purdue University

Figure 1: Input pairs for Task1

4.1 Task 1

For this task you are given an image pair per scene as shown in Fig. 1. Compare the quality of the correspondences obtained from the following three sub-tasks and state your observations.

4.1.1 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.

4.1.2 SURF or SIFT

Use the OpenCV implementation of either SIFT or SURF algorithm and descriptor matcher 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.

4.2 SuperPoint and SuperGlue

1. Download the `superglue_ece661.zip` package.

2. Follow the instructions given in `ece661_instructions.txt` to set up the necessary modules and your SuperGlue Conda environment.

3. Familiarize yourself with the provided SuperPoint+SuperGlue wrapper and helper functions in `superglue_ece661.py`.

4. Modify and run the demo script `run_ece661.sh` to generate and plot the matchings on the input pairs.

4.3 Task 2

Show results for Section 4.1 and Section 4.2 on at least two image pairs taken with your own camera.

4.4 Notes

1. For efficiency purpose, it's recommended to replace nested 'for' loops by vectorized operations.

2. Draw lines to indicate the selected correspondences. Try to randomize the colors of the lines.

5 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 **(only .py files are accepted — convert your .ipynb files to .py)**. Rename your .zip file as hw4_<First Name><Last Name>.zip and follow the same file naming convention for your pdf report too.

   There should be two items in your submission - (1) Homework PDF (2) ZIP file containing source code (*.py files) and text files (if any).

2. Your pdf must include a description of

   - Your answer to the theoretical question in Section 3.
   - 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.
   - Your observations on the output quality and performance of the SuperPoint+SuperGlue pipeline. How do the results compare to the "classical approaches" in the previous bullet?
   - 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.**

Homework File Size

For your homework submissions, please ensure that your reports (PDFs) are **under 10Mb**.

Some ways to reduce your report sizes are:

- Downsample your input/output image when including in the reports.

- When including plots in your reports, save them as PDFs instead of PNG/JPEG. PDFs are vector formats which are lightweight and do not pixelate when zooming into the plot.

Strike a balance between the file size and image quality displayed in your report. This will help your TA in easy distribution of homeworks for grading without maxing out the disk space. Moreover, this will help you when you upload your homeworks to your individual git repos. You don't want to upload 40 Mb PDF to your git repo!

Finally, do not include the images in your ZIP files for your homework submissions. Ideally, the ZIP file should be under 1Mb because it only contains ASCII (*.py /*.txt) files.

6 Further Reading (not for evaluation)

Now that you have used SuperPoint and SuperGlue models, you must have noticed that the deep-learning models can extract larger number of matching correspondences. Furthermore, when the two images are captured with wide differences in viewing angles or illumination, the classical feature matching pipeline (i.e using Harris+NCC/SSD or SIFT/SURF) fails to find matching points. This is where trainable networks significantly outperform the classical methods. This area of research is known as *Image Matching* and one of the top Computer Vision conference, CVPR, has been hosting the Image Matching Workshop [3] for the past 5 years (2019-2024).

There are many newer deep-learning models focusing on keypoint detection/description/matching. Some of the notable and interesting models are DISK[4], CAPS[5], DualRC-Net [6], LoFTR[7], and LightGlue [8]. If you are interested in diving deeper into this research area, you should read these papers.

In addition to the deep-learning models, the datasets used to train the above models are also equally important. Some of the most popular datasets are MegaDepth[9], ScanNet [10], Aachen Day and Night [11], and In-loc [12].

To get familiar with these models and datasets, you should visit their git repos. The git repos will have interesting demos which should quickly convey the utility of these models/datasets.

If you want to play with some of these models you should visit the kornia library. The library provides you quick access to these models with a single function call.

References

[1] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-Supervised Interest Point Detection and Description," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2

[2] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning Feature Matching

with Graph Neural Networks," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[3] Y. Jin, D. Mishkin, A. Mishchuk, J. Matas, P. Fua, K. M. Yi, and E. Trulls, "Image Matching across Wide Baselines: From Paper to Practice," 2020. 5

[4] M. Tyszkiewicz, P. Fua, and E. Trulls, "DISK: Learning local features with policy gradient," 2020. 5

[5] Q. Wang, X. Zhou, B. Hariharan, and N. Snavely, "Learning feature descriptors using camera pose supervision," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 5

[6] X. Li, K. Han, S. Li, and V. Prisacariu, "Dual-Resolution Correspondence Networks," in *Proceedings of Conf. on Neural Information Processing Systems (NeurIPS)*, 2020. 5

[7] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou, "LoFTR: Detector-free local feature matching with transformers," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 5

[8] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys, "LightGlue: Local Feature Matching at Light Speed," in *Proceedings of Intl. Conf. on Computer Vision (ICCV)*, 2023. 5

[9] Z. Li and N. Snavely, "MegaDepth: Learning Single-View Depth Prediction from Internet Photos," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5

[10] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D Reconstructions of Indoor Scenes," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5

[11] T. Sattler, T. Weyand, B. Leibe, and L. Kobbelt, "Image Retrieval for Image-Based Localization Revisited," in *Proceedings of British Machine Vision Conference (BMVC)*, 2012. 5

[12] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii, "InLoc: Indoor Visual Localization with Dense Matching and View Synthesis," in *Proceedings of IEEE Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5