AIN 433 - Introduction to Computer Vision Laboratory

HACETTEPE UNIVERSITY
Department of Computer Engineering

Programming Assignment 3
Fall 2023
Assoc. Prof. Dr. Ali Seydi KEÇELİ
R.A. Görkem AKYILDIZ

Due Date: 13.12.2023 - Wednesday (23:00:00) **Programming Language: Python 3**

Panorama Image Stitching with Keypoint Descriptors

In this assignment, you will merge sub-images provided by using keypoint description methods (SIFT and SURF) and obtain a final panorama image that including all scenes in the sub-images. First of all, you will extract and obtain the multiple keypoints from an sub-images by using an keypoint description method. Then you will compare and match these keypoints to merge give sub-images as a one panorama image (See Figure 1). As a dataset, you will use a subset of Adobe panorama dataset [1].

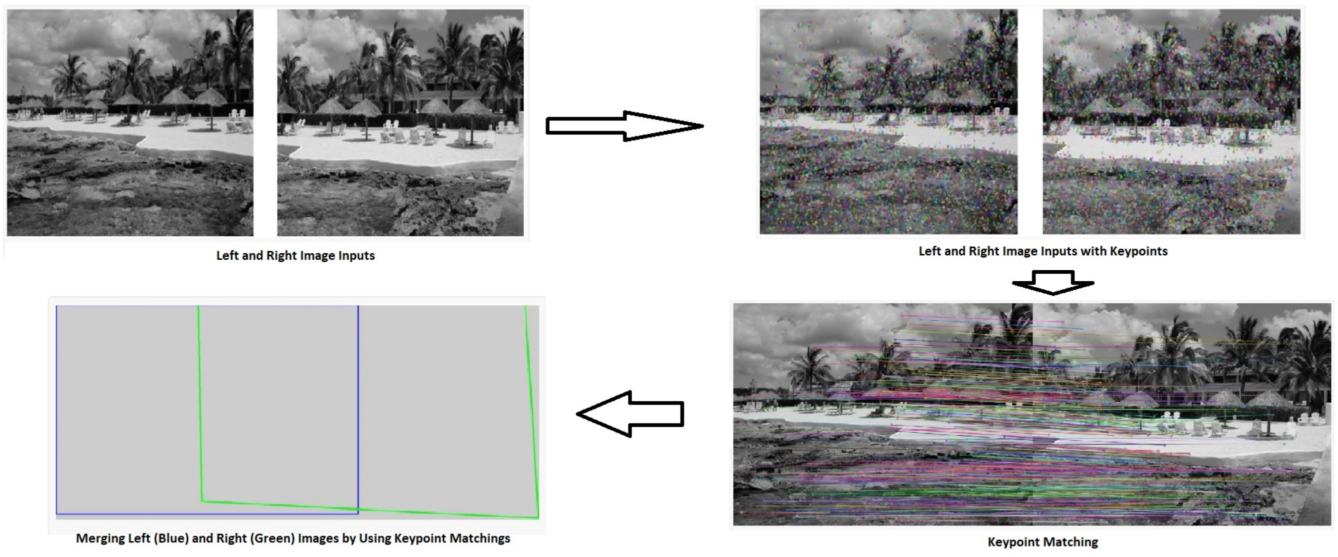
Left and Right Image Inputs
Left and Right Image Inputs with Keypoints
Keypoint Matching
Merging Left (Blue) and Right (Green) Images by Using Keypoint Matchings

Figure 1: Visual of Procedure to Be Used for Merging Images into Panorama

Keypoint Descriptors

Firstly, in the images, key points should be found to determine how to merge the multiple images into single one (Such as from which locations and as how much rotated). To detect keypoints in the image in a robust way (rotation and scale invariant) SIFT and SURF methods have been developed (See Figure 2).

Dataset

Dataset [1] includes various different scenes consisting of sub-images groups.

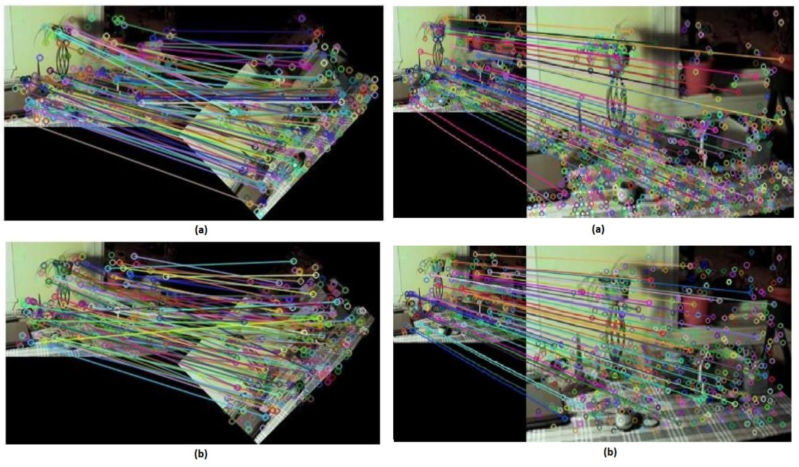

Figure 2: Matching keypoints original image with rotated version (**left**) and scaled version (**right**) with SIFT (**a**) and SURF (**b**)

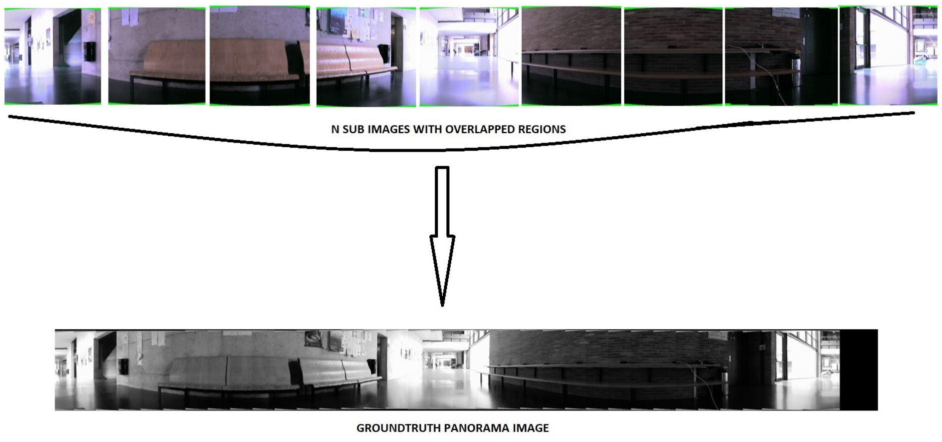
N SUB IMAGES WITH OVERLAPPED REGIONS
GROUNDTRUTH PANORAMA IMAGE

Figure 3: Sample sub-images (**Top**) and their corresponding ground-truth panorama picture (**Bottom**)

Implementation Details

1. **Feature Extraction (10 Points):** Firstly you are expected to the extract keypoints in the sub-images by an keypoint extraction method (SIFT or SURF).

2. **Feature Matching (10 Points):** Then you are expected to code a matching function (for example this can be based k-nearest neighbor method) to match extracted keypoints between pairs of sub-images. (For example, Pair 1: (SubImage 1 - SubImage 2), Pair 2: (SubImage 2 - SubImage 3), Pair N: (SubImage N-1 - SubImage N)) You can use libraries for this part.

3. **Finding Homography (30 Points):** Then you should calculate a Homography Matrix for each pair of sub-images (**by using RANSAC method**) and **you must implement this part by your own.**

4. **Merging by Transformation (30 Points):** Merge sub-images into single panorama by applying transformation operations on sub-images by using the Homography Matrix. **You must also implement this part by your own.**

5. You should pay attention to code readability such as comments, function/variable names and your code quality: 1) no hard-coding 2) no repeated code 3) cleanly separate and organize your code 4) use consistent style, indentation.

6. Your code should process all images in folders that are in folder named "dataset" folder by folder and save the result of them as "result.png" to corresponding folder. Moreover your code must save intermediate steps (that have been requested for report) to the disk after its run with an appropriate name.

What should you write in the report?

In your report, firstly, you are expected to write a detailed explanation (coding details, about your calculations and implementation details) for the four main steps:

- Feature Extraction,
- Feature Matching,
- Finding Homography,
- Merging by Transformation.

Secondly, you are expected to plot your results by the format below for each panorama in the dataset:

- Plots showing feature points for each ordered pair of sub-image,
- Plots showing feature point matching lines for each ordered pair of sub-image,
- Your constructed panorama image,
- Table for runtime of the description methods (SIFT and SURF).

Finally, you are expected to comment about your results (Explanation about your intermediate results and commenting about your program's performance by analyzing visually your constructed panorama. Note that you may use any existing tool to create your own ground truth for the panorama images **only for the comparison purposes**.).

What to Hand In

Your submission must contain following:

- README.md *(Text file containing the details about your implementation, how to run your code, which libraries have to installed, the organization of your code, functions etc. The template must be followed will be shared.)*
- src/ *(directory containing all your code)*
- Report.pdf

File hierarchy is as follows and must be zipped before submitted (Not .rar, only not compressed .zip files because the submit system just supports .zip files).

```
- b<StudentID>.zip
     - README.md
     - Report.pdf
     - <src>
        - *.*
```

Note that you MUST exactly score ONE from the submit system, otherwise you will have 20% of point deduction even if your hierarchy is correct. (For MacOS users, you can check the Piazza post for the script that zips the folder without any extra content which causes getting zero from submit)

Grading

The assignment will be graded out of 100; 80 for the code part and 20 for the report part.

The score for the report will be multiplied by your code score which is divided by maximum score that can be taken from the code part. For example, say that some scored 60 for code part and 16 for report part, his/her final score will be calculated as follows: 60+(60/80)*16=72

You MUST use justify -iki yana yasla in Turkish- page alignment and passive voice at your report, otherwise 20% of your score will be deducted FOR EACH VIOLATION for the relevant part of your report. Note that also your report's alignment must be well designed, otherwise you may also face with some point deductions for bad alignments/designs

You MUST write comments to your code as necessary and also your code MUST be readable.

Academic Integrity

All work on assignments must be done individually unless stated otherwise. You are encouraged to discuss with your classmates about the given assignments, but these discussions should be carried out in an abstract way. That is, discussions related to a particular solution to a specific problem (either in actual code or in the pseudocode) will not be tolerated. In short, turning in someone else's work, in whole or in part, as your own will be considered as a violation of academic integrity. Please note that the former condition also holds for the material found on the web as everything on the web has been written by someone else.

References

[1] Adobe open source datasets. `https://sourceforge.net/adobe/adobedatasets/home/Home` ˙ (Last access: 29.11.2023).

[2] Berkeley image stitching. `https://inst.eecs.berkeley.edu/~cs194-26/fa18/upload/files/proj6B/cs194-26-aeh/website` ˙ (Last access: 29.11.2023).