BBM 418 - Computer Vision Laboratory

HACETTEPE UNIVERSITY

Department of Computer Engineering

Programming Assignment-2
Spring 2019-2020
Doç. Dr. Nazlı İkizler Cinbiş

Due Date: 17.04.20 23:59

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 [1].

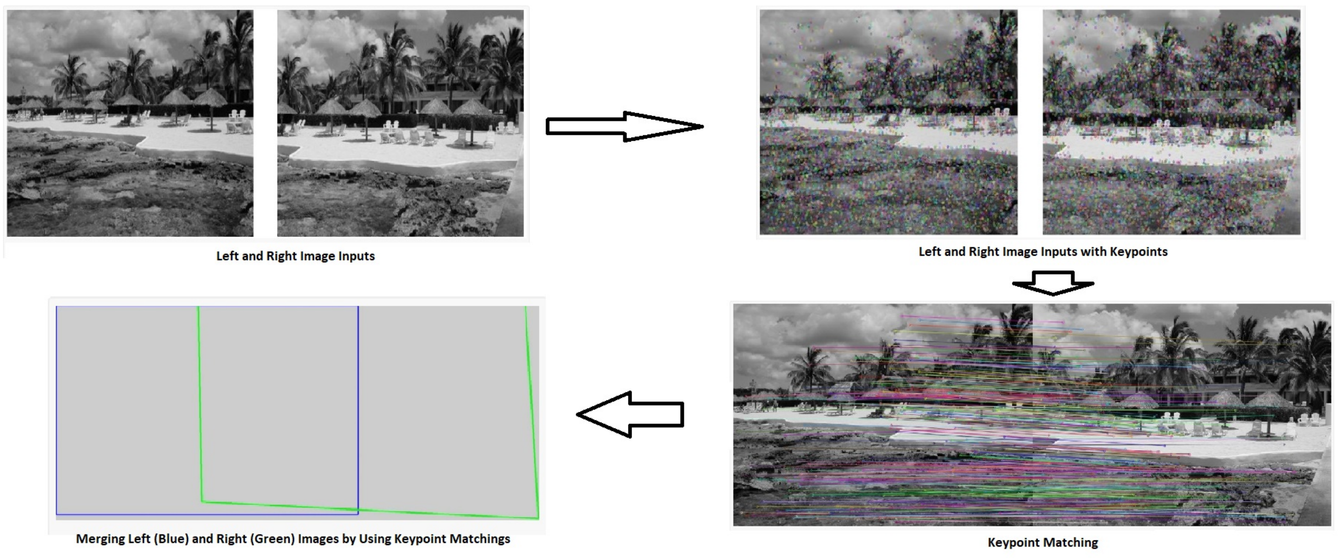
Left and Right Image Inputs
Left and Right Image Inputs with Keypoints
Keypoint Matching
Merging Left (Blue) and Right (Green) Images by Using Keypoint Matchings

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 and their corresponding ground-truth panorama image.(See Figure 3).

The Implementation Details

1. **Feature Extraction (10 Points):** Firstly you are expected to extract keypoints in the sub-images by an keypoint extraction method (SIFT or SURF). You can use libraries for this part.
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 ordered pairs of sub-images. (For

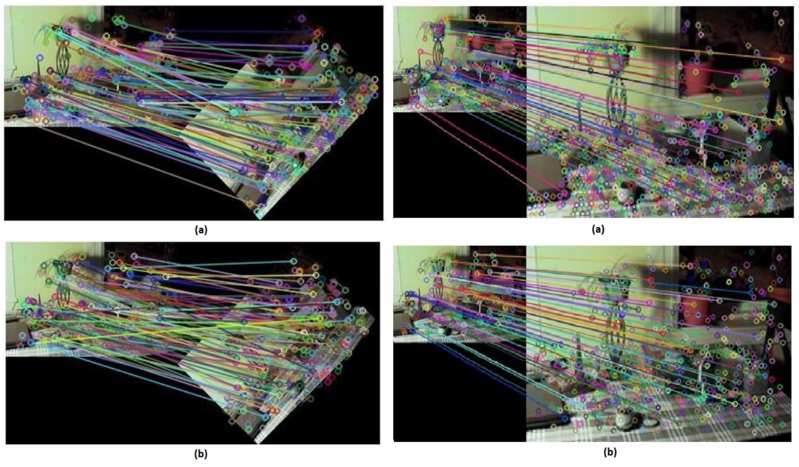

Figure 2: Matching keypoints original image with rotated version (**left**) and scaled version (**right**) with SIFT (**a**) and SURF (**b**)

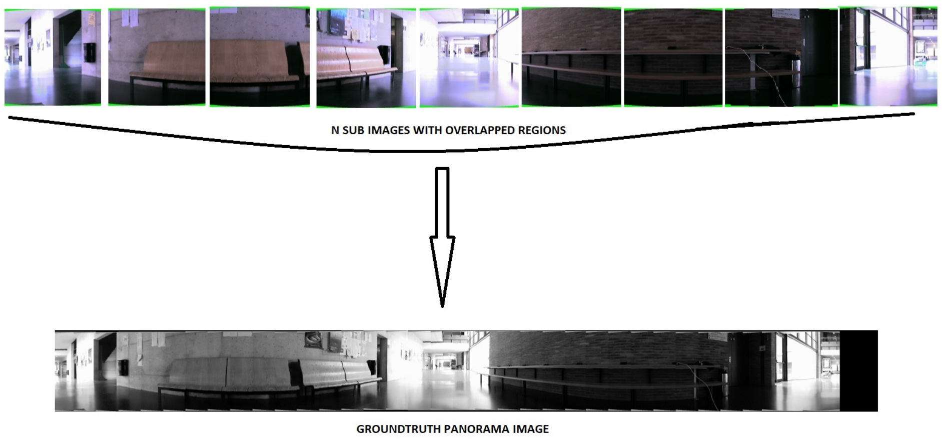
N SUB IMAGES WITH OVERLAPPED REGIONS
GROUNDTRUTH PANORAMA IMAGE

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

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 read all images from a folder named "dataset" and write results to the console as the same format specified in the "The Report" section.

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:

(a) Feature Extraction

(b) Feature Matching
(c) Finding Homography
(d) Merging by Transformation

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

(a) Plots showing feature points for each ordered pair of sub-image
(b) Plots showing feature point matching lines for each ordered pair of sub-image
(c) Your constructed panorama image
(d) Corresponding ground-truth panorama image

Finally, you are expected to comment about your results (Explanation about your intermediate results and commenting about your program's performance by comparing visually your constructed panorama and ground-truth panorama)

What to Hand In

Your submission format will be:

- README.txt *(give a text file containing the details about your implementation, how to run your code, the organization of your code, functions etc.)*
- code/ *(directory containing all your code)*
- report.pdf

Archieve this folder as **b<studentNumber>.zip** and submit to `https://classroom.github.com/a/Uv3MRC7-`.

Grading

The assignment will be graded out of 100:

- CODE: 0 (no implementation), 20 (an extremely incomplete implementation), 40 (an incomplete implementation), 60 (a partially correct implementation), 80 (a correct implementation) and REPORT: 20

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] This reference is hidden.