



Due Date: 23:59 pm on Wednesday, April 26th, 2022

## Image Panorama Stitching

In this assignment, you will merge sub-images provided by using keypoint description methods (SIFT,SURF and ORB) and obtain a final panorama image that including all scenes in the sub-images. You will also compare the methods (SIFT,SURF and ORB) with respect to run time and accuracy. First of all, you will extract and obtain the multiple keypoints from 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 subset of HPatches dataset from the given link [1].



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

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,SURF and ORB methods have been developed (See Figure 2). subset of HPatches dataset includes various scenes consisting of sub-images groups and the estimated ground truth homography with respect to the reference.(See Figure 3).

## The Implementation Details

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

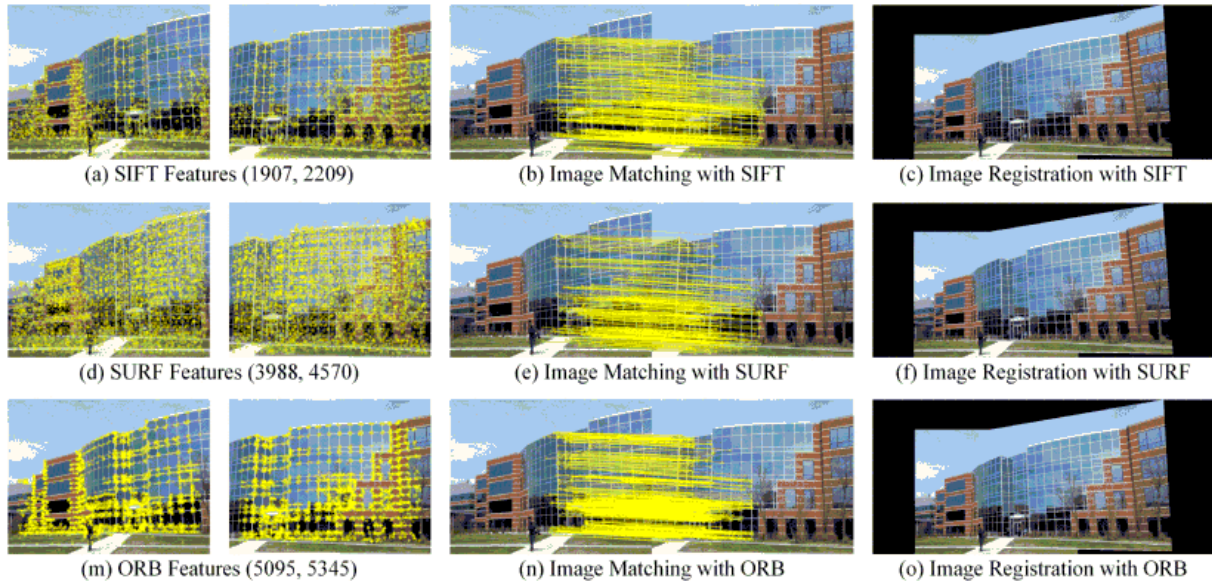


Figure 2: Matching keypoints original images with SIFT, SURF and ORB features



Figure 3: Sample sub-images [1] from the dataset

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. You should use Python 3 for the assignment.
7. Your code should read all images from a subset of HPatches dataset and write results to the console as the same format specified in the "What should you write in the report?" section.

## 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 and visual comparison of the description methods (SIFT, SURF and ORB).

Finally, you are expected to comment about your results (Explanation about your intermediate results and commenting about your program's performance of your constructed panorama visually).

## 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

Archive this folder as **b<studentNumber>.zip** and submit to <https://submit.cs.hacettepe.edu.tr>.

## Grading

The assignment will be graded out of 100:

- 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] <https://github.com/hpatches/hpatches-dataset>

[2] S. A. K. Tareen and Z. Saleem, "A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK," 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), pp. 1-10, 2018.