CS 484, Spring 2022
Homework Assignment 2: Local Features

Due: 8 April, 2022

1 Introduction

Image stitching is the process of combining multiple photographic images with overlapping fields of view to produce panoramic views. Using image stitching, the view is enlarged and the amount of information increases with the number of images that are stitched. It finds applications in a variety of image analysis tasks including:

- ‘Image Stabilization’ feature in camcorders that use frame-rate image alignment;
- High-resolution photomosaics in digital maps and satellite photos;
- Medical imaging;
- Multiple image super-resolution;
- Etc.

For a description of image stitching see the following reference:

- Matthew Brown and David G. Lowe. 2007. Automatic Panoramic Image Stitching using Invariant Features. Int. J. Comput. Vision 74, 1 (August 2007), 59-73.

2 Background

As part of this assignment, you would need to estimate a geometric transformation between a pair of images. For this purpose, it will be assumed that an ’affine’ transformation can characterise such a transformation. In addition, in order to minimise the effects of possible mismatches between pairs of images, the Random sample consensus (RANSAC) algorithm shall be used. A description of the ’affine’ transformation may be found for example in:

- L. G. Shapiro and G. C. Stockman, Computer Vision, Prentice Hall, 2001, Chapter 11.

The background material for the RANSAC algorithm may be found in:

- Martin A. Fischler and Robert C. Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography.

Note: You need NOT implement the affine transformation or the RANSAC algorithm yourself and are allowed to use available libraries for this purpose.

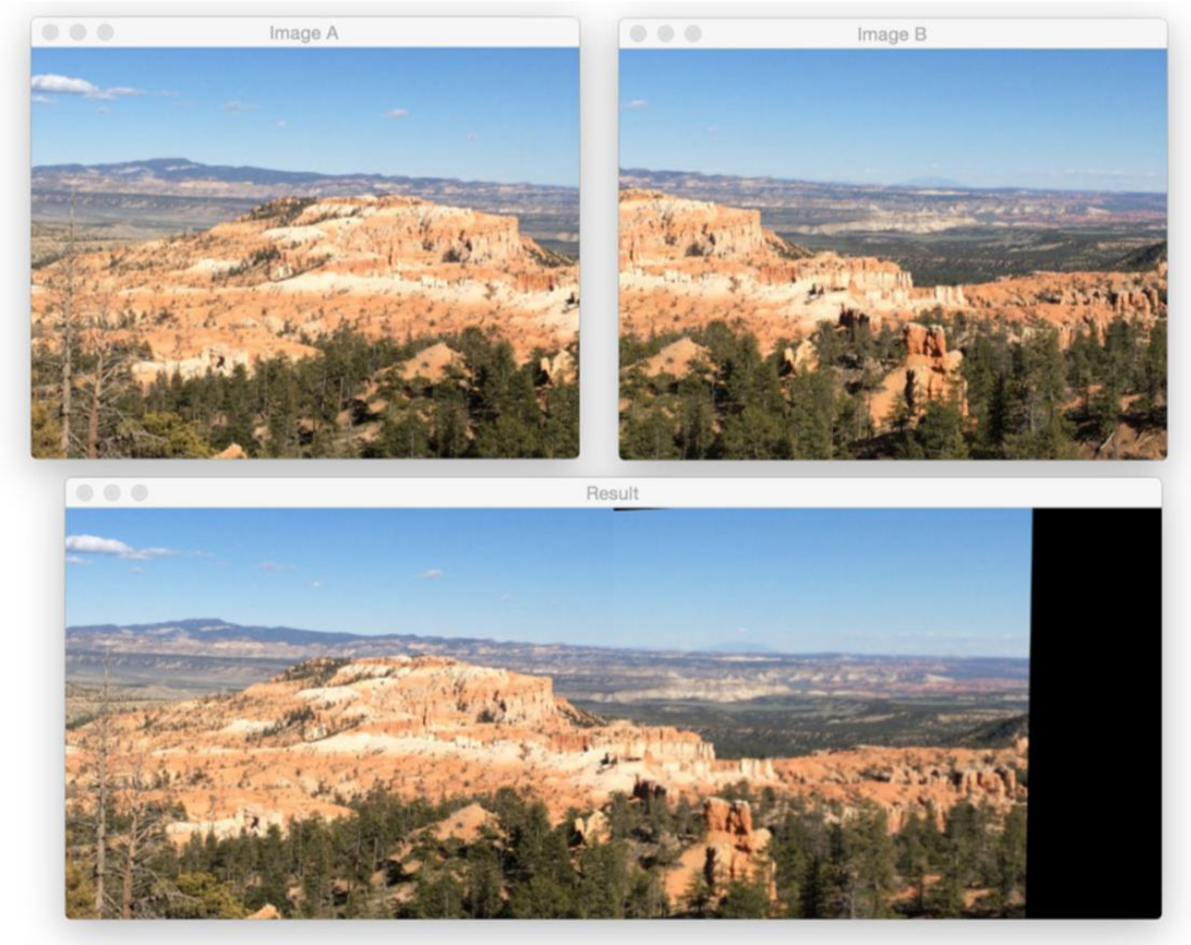

Figure 1: Example panoramic view generation. Image A and B (top row) are stitched together to produce the result (bottom row).

3 Approach

The approach may be summarized in terms of the following steps.

3.1 Preparing data

Download the image stitching data for this assignment from the course home page. It includes two sets of images. The first set (fishbowl) contains 13 images while the second set (goldengate) is comprised of 6 images. Images in each set are to be stitched together; i.e. one panoramic view for the first set and another panoramic view for the second set are to be produced in this assignment.

3.2 Detecting local features

The first processing step is to run the detector of the Scale-Invariant Feature Transform (SIFT) on each image to obtain a set of interest points. The output of this step for each image is a set of keypoints represented with four numbers: x and y location, scale (for which the point is a local maximum), and orientation (the dominant orientation in the neighborhood of the point). Each image can contain a different number of interest points.

Note: You can use the VLFeat open source library (http://www.vlfeat.org/) for an implementation of the SIFT detector or other available libraries you are comfortable with.

3.3 Describing local features

We will use two separate descriptors for each point:

- Gradient-based descriptor: compute the gradient-based descriptor of each local feature by using the descriptor step of the SIFT algorithm. Each point will have a SIFT descriptor of length 128.

- Raw pixel-based descriptor: compute a greyscale histogram descriptor for each local feature. You should use the location, scale and orientation values computed in the previous step to form a scaled and rotated square around each point. Then, the raw pixel-based descriptor will be computed as the histogram of the greyscale values of the pixels within the square neighbourhood. You may use a 256−bin histogram.

Note: You can use the VLFeat open source library (http://www.vlfeat.org/) for an implementation of the SIFT descriptor but the raw pixel-based descriptor step MUST BE your own implementation.

3.4 Feature matching

The next step is finding the similar keypoints between a pair of images. For this purpose, you would need to

- Consider a distance measure between feature vectors, the simplest one being the Euclidean distance; However, other alternatives do also exist as discussed during the lectures.

- Consider a threshold for similarity/dissimilarity above/below which a pair of keypoints from two images shall be considered as a match. You may need to examine a number of different thresholds to find the best possible working threshold.

- As an alternative to the previous step, you may consider a minimum number (m) of matched keypoints to exit between an image pair and then consider different values for m for best possible overall results.

- Find and store the coordinates of matching keypoints from a pair of images. Note that some keypoints from one image would not correspond to any points in another image. Ideally, only the overlapping areas from a pair of image should include matched pairs of keypoints. This assumption might be used as an initial cue to look for the optimal threshold for similarity/ dissimilarity or minimum number of matched keypoints (m).

3.5 Image registration

After finding the corresponding points in the reference image and the target image, the next step is to estimate the geometric relationship between the two input images, also referred to as image registration/alignment. Alignment is necessary to transform an image to match the view point of another image it is being composted with. Alignment in simple terms is a change in the coordinates system so that it adopts a new coordinate system which outputs image matching the required viewpoint. The types of transformations considered for an image to go through in this assignment are those of translation, rotation, scaling and shear which are collectively referred to as an affine transformation. A widely used approach for estimating an affine relationship is RANdom SAmple Consensus (RANSAC). The RANSAC algorithm is a method to estimate a model in the presence of outliers. In simple terms, the RANSAC algorithm finds a transformation which is less affected by mismatched keypoints.

Note: This step of the assignment can be performed using available libraries such as those in Matlab. However, you may need to tune some parameters for the RANSAC algorithm manually to obtain the best possible results.

3.6 Blending

The final step is to blend a pair of images stitched together. The purpose of this step is to create a soft transition along the borders where a transition occurs from one image into another in a panoramic view. Although more sophisticated methods exist, for this step, you are required to examine the following two options:

- averaging the pixel values in the overlapped regions from two or more images;
- weighted averaging of overlapping pixel values where the weights vary linearly according to the distance of a pixel in the overlapped region to the centre of either one of the images.

Note: You MUST implement this step yourself.

3.7 Discussion

You must discuss your implementation choices at each step and how these choices affect the final result in a report. The discussion should include example outputs for each step, such as local features detected for at least one image, example gradient-based and raw pixel-based descriptors (shown as bar plots) for several points from these images. You must discuss how different descriptors affect the final results, and discuss which images are easier or more difficult to be stitched together and why.

3.8 Software

You must provide well-documented code that implements all steps described above. For the steps that use external libraries, you must provide a README file that explains which libraries are needed and how they can be obtained and installed.

You must have a specific entry point (e.g. a function or script) to your solution to this homework assignment with the inputs listed below:

- A text file that lists a set of image names to be stitched together;
- An option for choosing a particular local feature descriptor (one of two options).

The code for this entry point must be clearly described and documented in your solution. You are free to use any data structures and programming languages in your implementation. Note that we are going to test your code by using both the provided image set as well as a separate set of images.

Submit:

1. **A report (pdf file)** that contains the information requested above. You are also expected to provide a discussion of the results (e.g. which steps were easy and which were more difficult, what was possible and what was not).
2. **Well-documented code** for all steps that you implemented yourself. The specific entry point must be clearly indicated and documented.
3. Citations to all external resources that you used in your solution

Make sure that all files you submit include your name and student ID.

Notes:

This assignment is due by 23:59 on Friday, April 8, 2022. You should submit your solutions as a **single archive file** that contains your **code** and **report** (a pdf file that contains the resulting plots, descriptions of how you obtained them, and discussion of the results) through moodle. You can contact the TA (navid.ghamari@bilkent.edu.tr) for your questions.