Panoramic Photo Construction

Seokchan Ahn
Department of Computer Science
University of California, Irvine
seokchaa@uci.edu

Abstract

The goal of this project to replicate the image stitching technique which is widely used for constructing a panoramic photo or implementing a vehicle surround view system. This project focus on automatically constructing a panoramic photo given two pieces of images that shares the same area. The process of image stitching consists of four steps: local invariant descriptor extraction, feature matching, homography matrix estimation, and warping transformation. I followed each step using the algorithm implementation in OpenCV 4.5.5 and visualized the process and results.

1 Project Overview

Panoramic image stitching has several commercial applications not only the landscape photography and generating maps from satellite photos, but also 360 degrees panoramic images for VR(Chen, 1995). The technique that fully automate the image stitching is well known (Brown and Lowe, 2007), and this project follows the steps presented in the paper. Though using a neural network model is a big trend and there are also several works with such approach (Yan et al., 2016), neural network models often suffer from various kinds of transformation in input images, especially when they do not exist in the training data. Thus, it is common practice to augment training images by flipping or rotation (Sokolić et al., 2017). In contrast, the invariant feature based approach by Brown and Lowe (2007) enables reliable matching robust to rotation, zoom and illumination change in the input images without requiring such models and can be run very fast.

The image stitching process consists of four steps. When a pair of images is given, we first extract local invariant descriptor, often dubbed as keypoints. Keypoints are the points on the edges or corners that help distinguishing the shape of the objects in the image. Then we guess the same areas in the two images by matching the keypoints in them. Next step is to find a homography matrix, which is used for warping an image. If we have enough number of matched keypoints, we can find a matrix that match those points by rotating and stretching one of the images. Finally, warp the second image to stitch the two images.

2 Data Sets

I took a photo of the UCI campus and cutted the photo into several pieces that have overlapping area between them. Then, I applied transformations on the pieces such as rotation and warping. Figure 3 shows the setting for one of the experiments. The photo on the left is the original image and the two pieces on the right are came from the same original photo.

3 Algorithms

3.1 SIFT

Scale-invariant feature transform (Lowe, 2004) is a method for extracting distinctive invariant features from images that enables to perform a reliable matching between different views of an object or scene. The SIFT features are useful because they are invariant to image scale and rotation and highly distinctive.

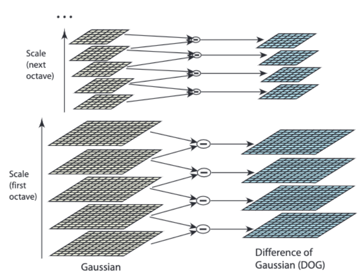

Figure 1: For each octave of scale space build an image pyramid with the same size by repeatedly convolve with Gaussians. Difference-of-Gaussian (DoG) images are generated by subtracting adjacent Gaussian images. The process repeated after down-sampling the Gaussian image by a factor of 2.

The first step is to extract scale-invariant features using a method proposed in Lowe (1999). As shown in Figure 1, it finds scale-invariant feature candidates by building multiple image pyramids of the images convolved with different size of Gaussians and finding the extreme points in the DoGs. The next step is to make the extracted keypoints rotation-invariant by assigning the orientation attribute for each keypoint. Figure 2 shows the process of creating a keypoint descriptor. It takes a small window around a keypoint and make a histogram of image gradients for each subregions. With this, we can match keypoints that have different directions.

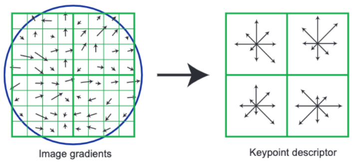

Figure 2: A keypoint descriptor generated from image gradients.

3.2 RANSAC

Random sample concensus (Fischler and Bolles, 1981) is a model fitting algorithm that is robust to significant percentage of outlier data. Our goal of using RANSAC is to find a best homography matrix from the feature matches we found in the previous step. As we can observe in the Figure 3, there can be features that are repeateadly observed in different positions in a photo such as windows or other patterns in the image. Since such features are not one-to-one correspondence, the matched pairs are inevitably noisy and we need a very robust algorithm.

RANSAC first select a few sample data and find parameters that fit best for the sample data. Then it count the number of data points near the model function and remember the parameter if the number was larger than a specific threshold. It repeats this process several times and return the best model in the history as a result.

3.3 Experimental Design

3.3.1 Data

As described in Section 2, I cut a single original photo into two pieces with overlaps and make three different versions of the right piece by rotating and warping using a graphic editor program. The images used are shown in Figure 3.

3.3.2 Image Stitcher Implementation

The implemented image stitcher takes two images as input and return the stitched image and visualization of matched features.

Figure 3: The original image (top) and its splitted pieces used in the test.

4 Results

The first experiment is to find a simple parallel translation (Figure 4). Next, to test the rotation-invariant attribute of the SIFT features, I exper-

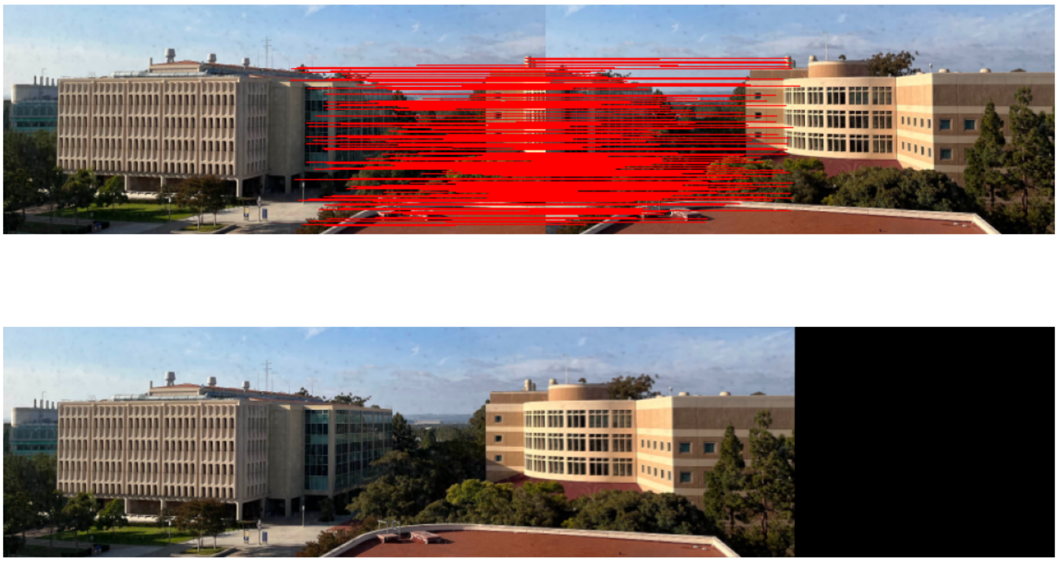

Figure 4: Experiment 1: Parallel translation.

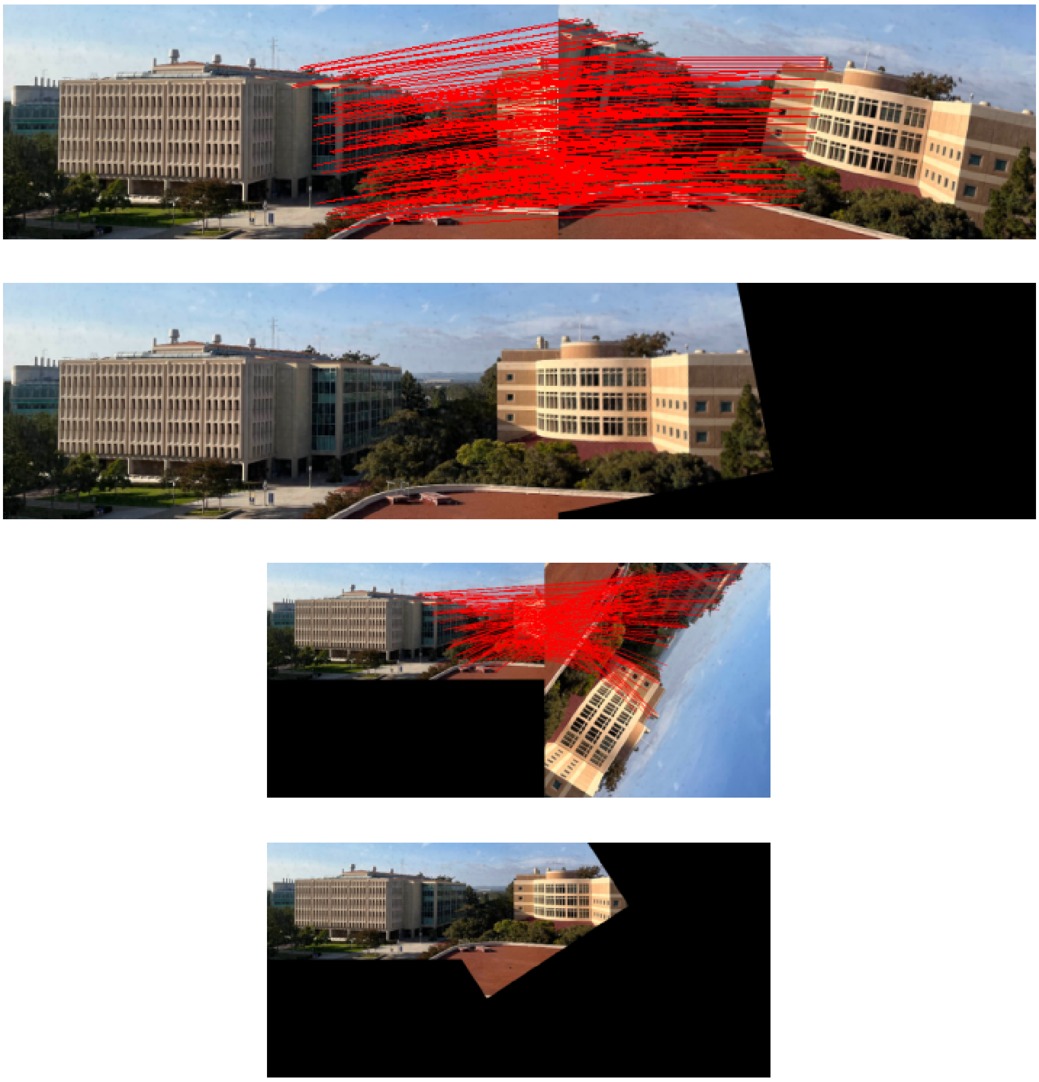

Figure 5: Experiment 2: Rotational transform + Parallel translation. Works fine even with the extreme rotation.

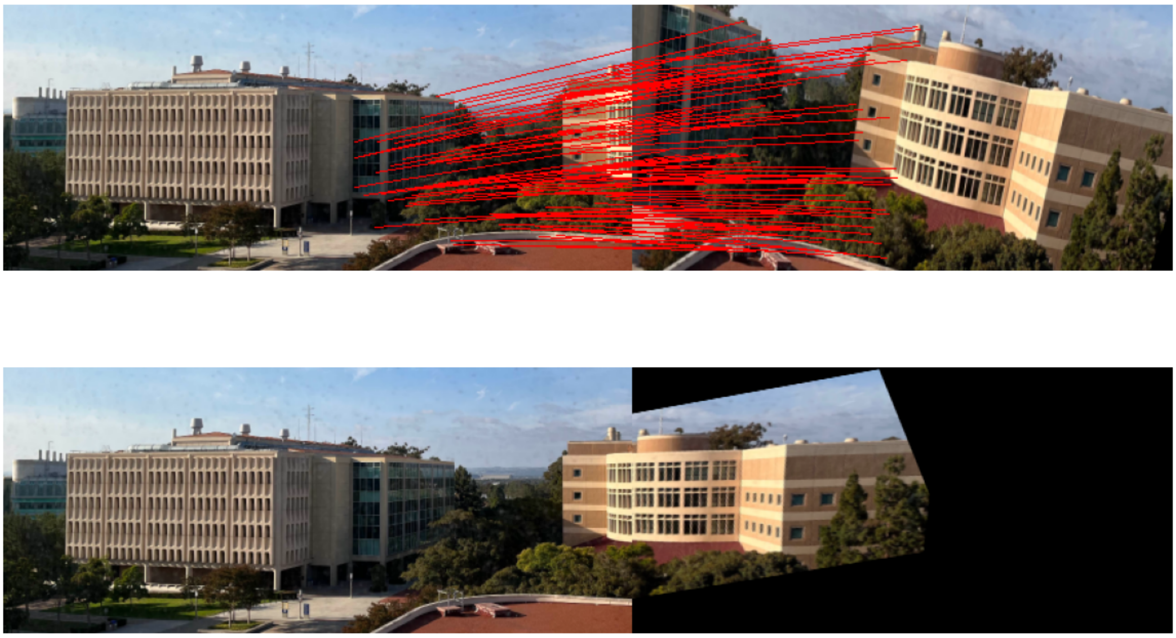

Figure 6: Experiment 3: Perspective transformation (Warp).

imented stitching a slightly rotated image and a extremely rotated image(Figure 5). Finally, I tested with the warped image to see whether the algorithm can conduct a complex transformation (Figure 6).

5 Assessment and Evaluation

For the simple first experiment, as expected, the stitcher found the horizontal feature matches and perfectly matched the images. For the next experiment that tests the rotation-invariant feature matching, the stitcher worked perfectly even with the rotation of greater than π degrees in radian. This proves the SIFT algorithm can compare the features regardless of its direction on the image. For the last experiment, it worked fine, but was not perfect. I was able to observe the step on the right edge of the left image where the ivory and brown wall meet. The error was not significant, but this amount of error might be critical in some settings such as stitching the satellite photographs for a map. One of the possible reason for this would be the lack of data points feed to the RANSAC algorithm. However, in case we cannot find enough number of matches, we can think about another approach that finds a homography matrix that makes the stitched image as smooth as possible. The assumption under this idea is the continuity on the edge of the image. This might be valid in most cases and we would be able to get a better perspective transformation with that idea.

References

Matthew Brown and David G Lowe. 2007. Automatic panoramic image stitching using invariant features. *International journal of computer vision*, 74(1):59–73.

Shenchang Eric Chen. 1995. Quicktime vr: An image-based approach to virtual environment navigation. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 29–38.

Martin A Fischler and Robert C Bolles. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.

David G Lowe. 1999. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee.

David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

Jure Sokolić, Raja Giryes, Guillermo Sapiro, and Miguel RD Rodrigues. 2017. Robust large margin deep neural networks. *IEEE Transactions on Signal Processing*, 65(16):4265–4280.

Min Yan, Qian Yin, and Ping Guo. 2016. Image stitching with single-hidden layer feedforward neural networks. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 4162–4169. IEEE.

6 Appendix

In the file *ImageStitcher.ipynb*, I implemented a python named Stitcher and visualized the results using Jupyter Notebook. I used the algorithms implemented in *opencv-python* 4.5.5. The main functions are as below:

- **stitch(imageA, imageB, ratio, reproj_thresh)**: The API for stitching two input images. Takes threshold values for Lowe's ratio test and reproject threshold.

- **detect_keypoints_and_features(image)**: Detects keypoints using Difference of Gaussian (DoG) algorithm and extracts scale invariant features using Scale-invariant feature transform (SIFT) algorithm. Uses implementation of OpenCV internally.

- **match_features(featuresA, featuresB, ratio)**: Match features in the two images using K-nearest neighbors and filter outliers based on the given ratio parameter.

- **estimate_homography_matrix(kpsA, kpsB, matches, reproj_thresh)**: Estimate a homography matrix that transforms image B, matching the keypoints the best, using the RANSAC algorithm. Uses the OpenCV RANSAC implementation internally.

- **visualize_matches(self, imageA, imageB, kpsA, kpsB, matches, status)**: Visualize feature matching. Juxtapose two images in order and draw red lines between matched features.