Panorama Stitching

Srijan Mallick

May 2021

In this project, we shall perform image stitching using Python and OpenCV. Given a pair of images that share some common region, our goal is to “stitch” them and create a panoramic image scene. The steps to achieve that goal are as follows:

- Keypoint detection
- Local invariant descriptors (SIFT)
- Feature matching
- Homography estimation
- Perspective warping

1. **Keypoint Detection**
In our code, we have used the function *cv2.ORB_create()* to create a descriptor object, and then we have extracted keypoints by calling the method *detectAndCompute()*. We have visualised the keypoints in the image by the function *cv2.drawKeypoints()*.

2. **Feature Matching**
After extracting keypoints from both images,we would like to compare the 2 sets of features and stick with the pairs that show more similarity. For this, we have used a BruteForce (BF) Matcher object by calling the function *cv2.BFMatcher*, in which the first parameter is the distance metric and the second one is the crossCheck boolean.

3. **Homography Estimation**
Homography is a 3x3 matrix, essentially, a linear transformation that will relate the two images. Since we have obtained feature points from the two images, we can pass them into the *cv2.findHomography()* function to get a perspective transformation.

4. **Perspective Warping**
Once an accurate homography has been calculated, the idea is to transform one of the images so that both images merge as one. This is done using the $cv2.warpPerspective()$ function in OpenCV.

An Example

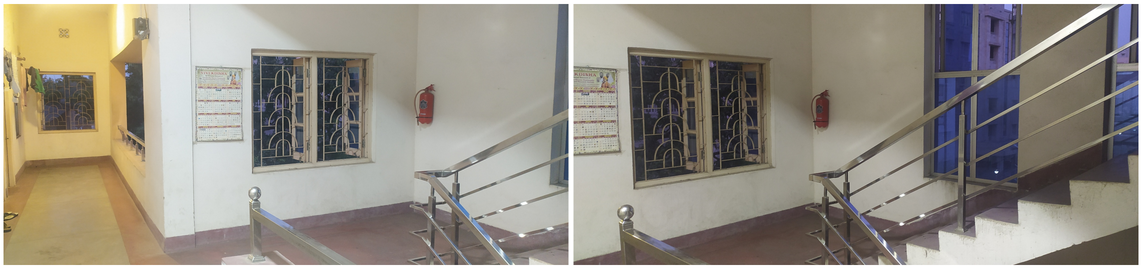

Figure 1: The two images to be stitched

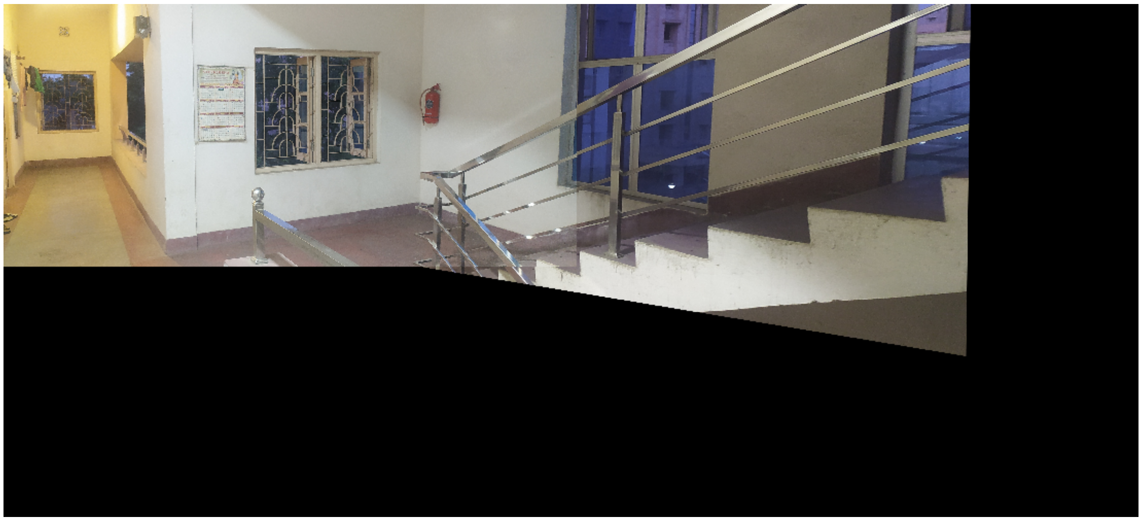

Figure 2: The stitched panorama

1 Reference

- Silva, T., *"Image Panorama Stitching with OpenCV"*