COMPUTER VISION

ASSIGNMENT 1 - IMAGE STITCHING APPLICATION

Amit K. Jadhav

Supervisor - Stylios Asteriadis / Mirela Popa

Department of Data Science and Knowledge Engineering
Maastricht University

May 9, 2022

1 Introduction

The technique of stitching two images to form a Panoramic image has been explored quite a bit by many individuals in the past years. This can serve as a tool to observe the changes in one part of a natural habitat across different points in time. Here I am listing the steps taken to implement the image stitching to create a panorama image. The recent advances in computer vision make it all the more relevant to apply such techniques to many useful applications. Image stitching application to create meaningful panorama images is one such use case.

Demonstrated here is an application which is capable of taking in two images with different affinities i.e. different scales, orientations and translations and stitching them to form one panorama image using the union of the information contained in both the images.

2 Methodology

1. The images were read and converted from BGR channel to RGB channel. Then resultant images were converted from RGB Channel to single channel Grayscale images.

2. The gray scale images were fed to Harris Corner Detector to examine corners.A corner can be recognized as the junction of 2 edges where any edge detector such as a the Sobel operator is applied to detect edges i.e. find the x and y gradient values for each pixel in the grayscale image. After this, the corner strength is computed using the individual magnitude and the mutual ratio of Eigen values $\lambda 1$ and $\lambda 2$ (the values defining the strength of change of gradient in each edge direction) is computed around a n*n pixel window for every pixel named Harris value.

3. Upon examining the presence of Corners, a separate approach was followed to also detect corners using SIFT detectors and the DOG scale invariant 128 feature arrays were computed for each of the features detected in this step. This was done since corners described as SIFT features are rotation / translation and scale invariant unlike harris corners.

4. The features described for all the key points (corner points) in the first image were matched for correspondence with all the features described for the key points in the second image. The match distance was computed using BFMatch function part of the opencv-python library. The bi-directional aspect of this correspondence was examined by checking that the correspondence given by this function was direction neutral or commutative (i.e. A $\longrightarrow$ B had the same distance as b $\longrightarrow$ A). It is worth noting that the BFmatch gives the unsorted results and just the correspondence with the distance between corresponding points along with the query image index and train image index which is then used to extract the x,y coordinates of the points in the respective pixel space.

5. The x,y coordinates of the features included as a part of the considered feature matches were then separated out and fed to two RANSAC implementations separately.

6. The first RANSAC implementation was for achieving the outlier invariant line fit for finding equation of x,y coordinates of corners in second image to x coordinates (defined as U coordinates in the code) in first image thereby giving solution (a,b,c) to the equation

$$a * X + b * Y + c = U \tag{1}$$

. The second RANSAC implementation was respectively for finding equation of x,y coordinates of corners in second image to y coordinates (defined as V coordinates in the code) in first image thereby giving solution (a,b,c) to the equation

$$d * X + e * Y + f = V \tag{2}$$

7. The two RANSAC implementations provided the first two rows of the affine transformation matrix and the assumption of restricting this image stitching exercise to no projective transformation led to the last row of affine transformation matrix being [001]
8. The resulting 3 x 3 affine matrix was used to warp the second image onto the first image and create a stitched image using the warpPerspective function in the opencv-python library.
9. Finally any rectangular black region resulting from warping the two images was cropped away and the final panorama image was returned.

3 Experiments

1. DIFFERENT INPUT IMAGES: I have attempted to stitch the left to right unrotated/rotated images and also top to bottom unrotated/rotated images. Also I have unsuccessfully tried to implement the before after images, which give inconsistent and high inaccurate stitching results when the input images have a visible projective variation between the first and second input images. I have attached the screenshots of my results below for left to right stitch on Maastricht Town skyline, one random city skyline, Maastricht town skyline top to bottom stitch and Ukraine Building Before and After images demonstrating destruction.
2. HARRIS CORNERS: Different window sizes (3/4/5) and different thresholds of Eigen ratio (0.3/0.4/0.5/0.6) were experimented to get high or low number of corners and to achieve a clear separation between detected corners versus edges and low gradient areas. As the threshold is reduced, more and more key points with even lower cornerness values are also output by the Harris Corner Detector.
3. SIFT FEATURES DISTANCE MATCHING: This nearest match checking between key points of two images was earlier tried using euclidean distance formula of computing the Square Root of Sum of Squares of differences in each of the 128 descriptors. This comparison was done for

all key-points of first image with all key-points of second image. However due to the ease of feeding the DMatch objects to match-line drawing methods the BFMatch function part of the opencv-python library was used. The manual numpy based calculation of distance between keypoints is retained as commented section in the code (to demonstrate the computational thought process).

4. RANSAC: Different combinations of proportion of correspondence matches i.e. 50%, 75% and 90% respectively of the feature matches were extracted from the total feature matches obtained from SIFT and fed to RANSAC. It was observed that RANSAC works well with about 75% of the matches in some experimented use case with non rotated images and 90% of the matches with rotated images.

4 Results

The results for image stitching have been demonstrated below. Additionally I have attached a short two minute .mp4 video demonstration showing the outlier removal effect of RANSAC by ploting the 3d graphs of Pre-Ransac results between corespondence versus Post-Ransac results.

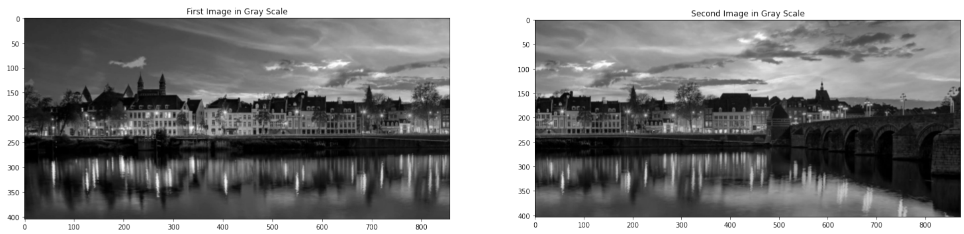

Figure 1: Grayscale Images Left - Right stitching - Maastricht

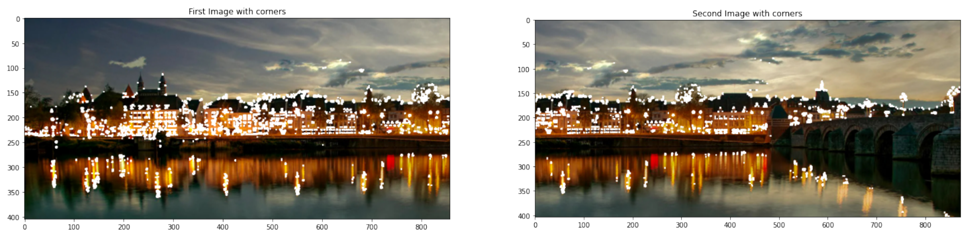

Figure 2: Harris Corner Images Left - Right stitching - Maastricht

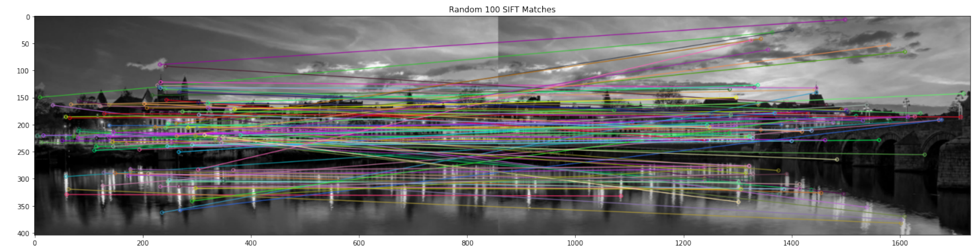

Figure 3: SIFT Features with Correspondence for Left - Right stitching - Maastricht

Figure 4: Stitched Image for Left - Right stitching - Maastricht - with uncropped black region

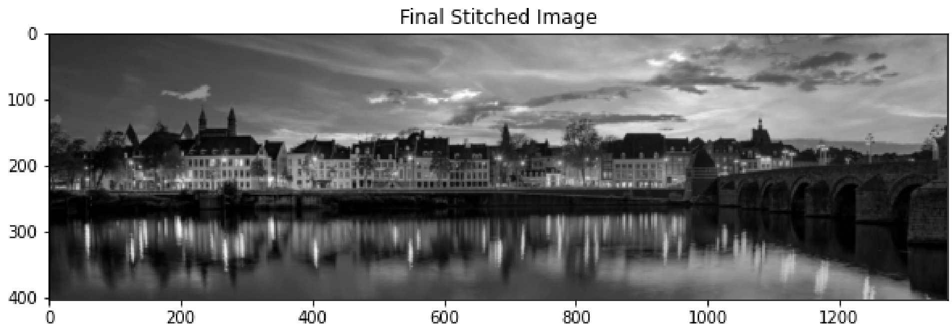

Figure 5: Final Stitched Image for Left - Right stitching - Maastricht - black region cropped

Figure 6: Grayscale Images Top-Bottom stitching - Maastricht

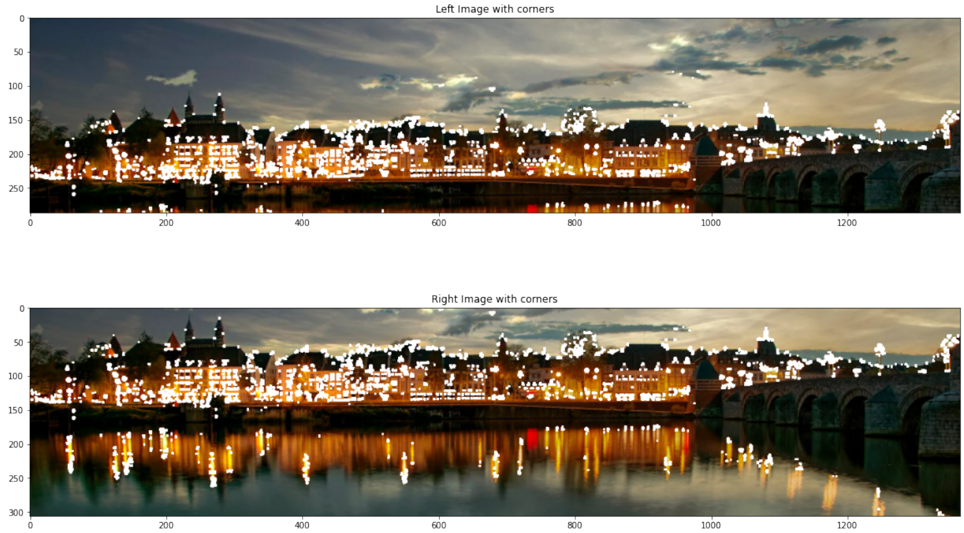

Figure 7: Harris Corner Images Top-Bottom stitching - Maastricht

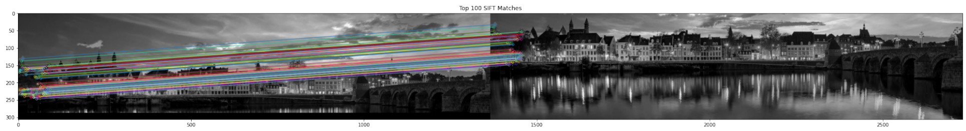

Figure 8: SIFT Features with Correspondence for Top-Bottom stitching - Maastricht

Figure 9: Stitched Image for Top-Bottom stitching - Maastricht - with uncropped black region

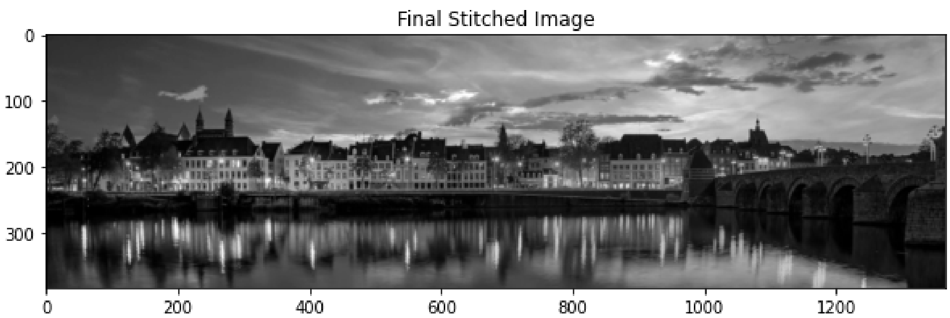

Figure 10: Final Stitched Image for Top-Bottom stitching - Maastricht - black region cropped

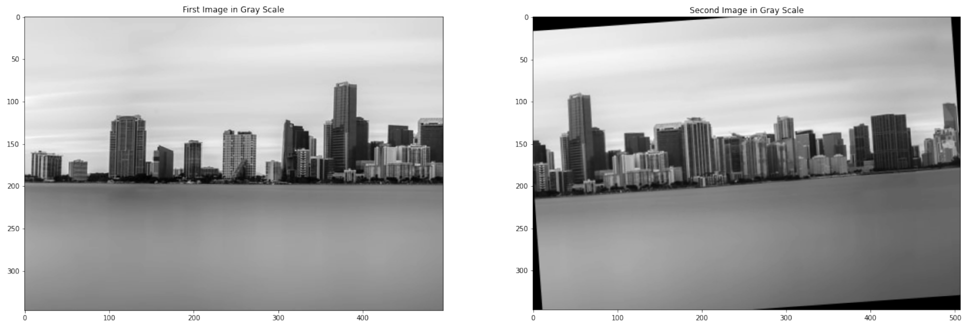

Figure 11: Grayscale Images Left-Right stitching with Rotation

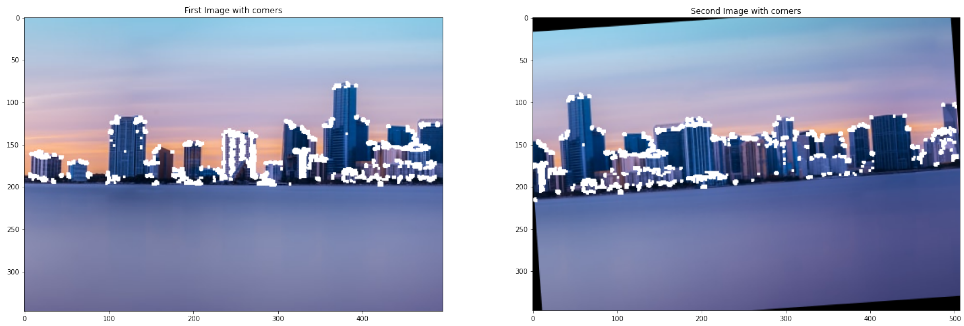

Figure 12: Harris Corner Images Left - Right stitching with Rotation

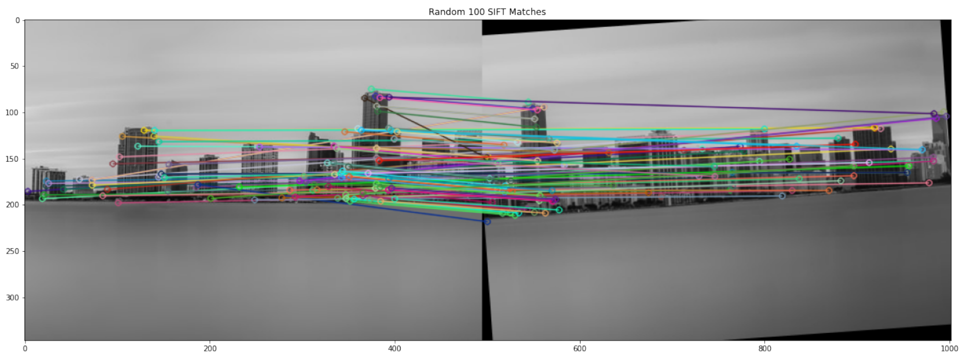

Figure 13: SIFT Features with Correspondence for Left - Right stitching with Rotation

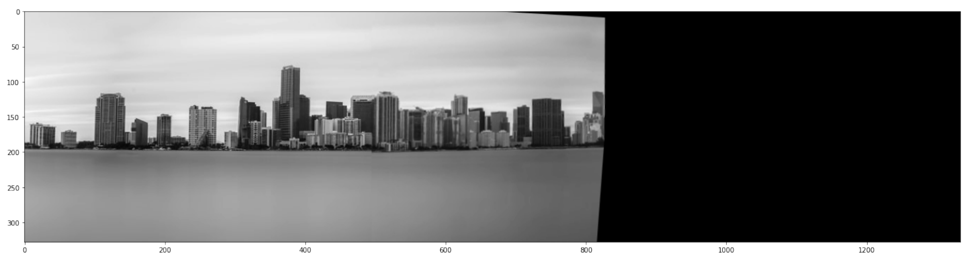

Figure 14: Stitched Image for Left - Right stitching with Rotation - with uncropped black region

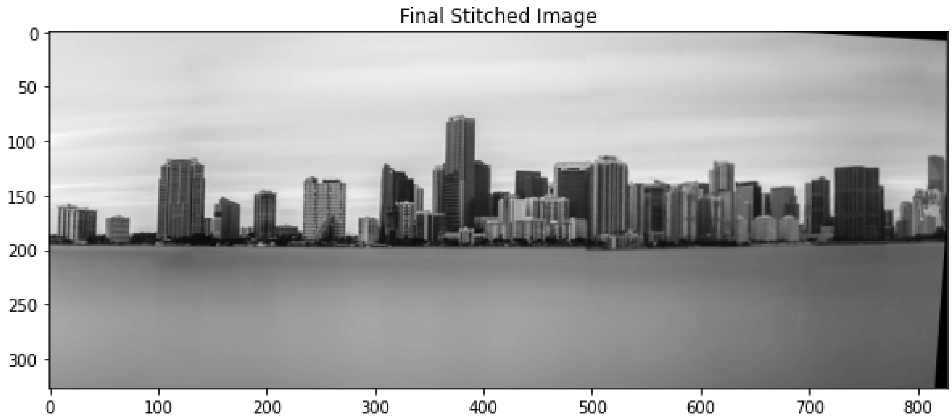

Figure 15: Final Stitched Image for Left - Right stitching with Rotation - black region cropped

Figure 16: Grayscale Images Before After stitching - Ukraine

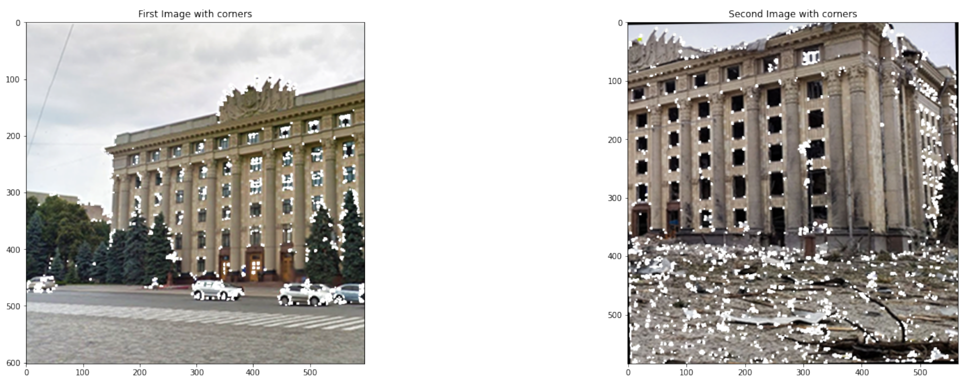

Figure 17: Harris Corner Images Before After stitching - Ukraine

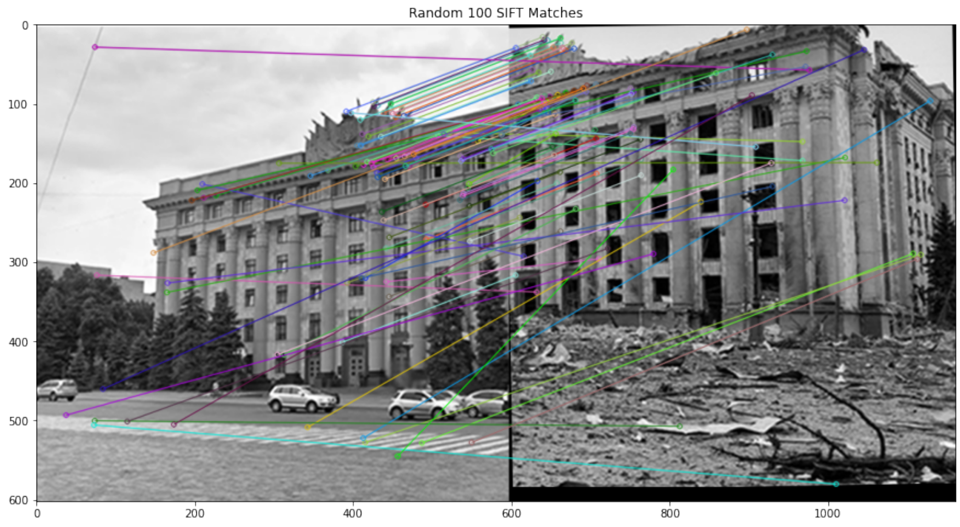

Figure 18: SIFT Features with Correspondence for Before After stitching - Ukraine

Figure 19: Stitched Image for Before After stitching - Ukraine - with uncropped black region

Figure 20: Final Stitched Image for Before After stitching - Ukraine - black region cropped