Image Stritching

Authors: Yehuda Neumann, Yakir Arie, Mark Gurin

IDs: 305539066, 205387491, 320755499
Department of Computer Science
Computer Vision and Image Processing
Ariel University

Advisor: Dr. Gil Ben-Artzi

October, 2019

Introduction To Image Stitching

The problem: From series of photos that were made from static rotating camera, we have to create one image that will represent the space, where the original made images transformed to the other image's plane. To achieve that task a couple of setting should be done. before diving into the result and the process let's figure out what are the difficulties that we will have to deal with. Stitching errors are a common problem in panoramic photography, where the photo stitching software made a mistake in lining up the images when it stitched them together. After stitching your panorama you might see some places where objects have been distorted, misaligned or cut in half during the panorama stitching process, like in this example:

Figure 1: Panorama stitching errors caused by distortion and lack of overlap.

Figure 2: Series of photos

Figure 3: Final result of panorama

Approach and Method

The main problem is to find the best Homography matrix that will move each pixel to the right place s.t the set of images will looks like one complete image without distortions. to find the Homography and achieve image stitching we have to follow the basic steps:

- Detect keypoints in each image
- Match the most common points of the two sequenced images using RANSAC
- Find the Homography matrix
- Transform one image to the other's image plane
- Wrap the two images

Keypoints and Features

Using OpenCV we detect the features of the images by using the SIFT algorithm (Scale Invariant Feature Transform). As we seen in the lectures there is another approaches to detect features in image. One of them is Harris corner detection. Harris corner detection is bad approach because it is not scale invariant.
The following 3 images explaining the reason of Harris corner failure. If the corner is scaled up

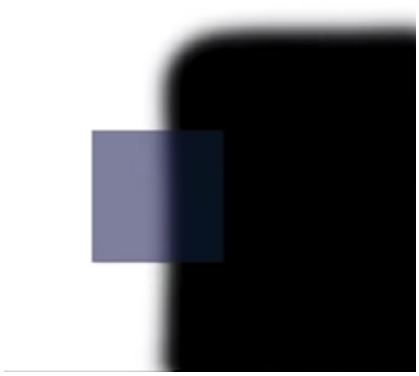

Figure 4: Filter window on edge

Figure 5: Filter window on corner

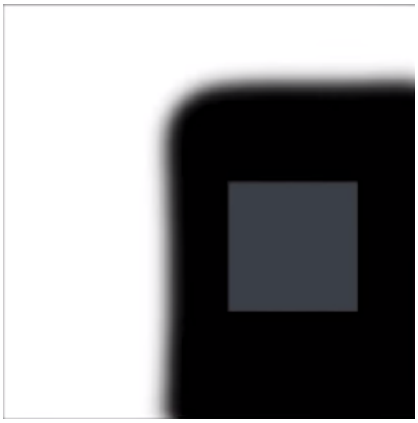

Figure 6: Filter window can not recognize the corner any more

and the filter window (light blue color) is on the corner's side or on the x,y edge then we can still recognize that that is a corner, but if the filter window falls inside the corner figure, then we lose the indication of the corner.

RANSAC

Random Sample Consensus, is allowing us effectively remove outliers and determine the strongest linear model of the dataset.

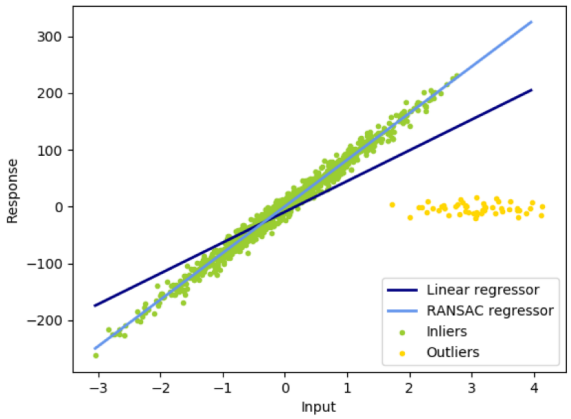

Figure 7: RANSAC model vs. Linear regressor

As we can see in the example above, there is two lines (models). One for the Linear regression and one for the RANSAC. Because of the outliers (yellow dot group) the Linear model has wrong estimation, on the other hand RANSAC model show's the most estimated model by ignoring the false data (outliers). The model runs on iteration over the dataset, on each iteration it selects randomly 4 dots, and calculate the distances from them to the line.

After that the model checks how mush data is placed near (by some delta) or on that line. The model with maximum number of dataset that placed on the same line returned. After running RANSAC we render two images that show's the inliers and the outliers of the images.

Homography

After using the RANSAC, we determine the inliers that best fit to the model. Now we want to fine the matrix that will transform one plane pixel to another plane, we want transform image2 to the plane of image1. When an image is transformed to another image's plane, it ensures that the epilines in both images are parallel to this image plane. This is a useful property that can be harnessed to reduce the number of searches while matching keypoints in 2 images, since a given point in an image will lie on the corresponding epiline in the other image (observed from a different point in space).
Homography can be mathematically expressed as

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Our Test and Results

We implemented the RANSAC from scratch, used the OpenCV feature detection with the help of SIFT, finally found the best Homogrphy and at the end made the warping. Here is our image set, the RANSAC inliers, outliers and the final panorama result.

Backyard

Figure 8: backyard1

Figure 9: backyard2

Figure 10: backyard3

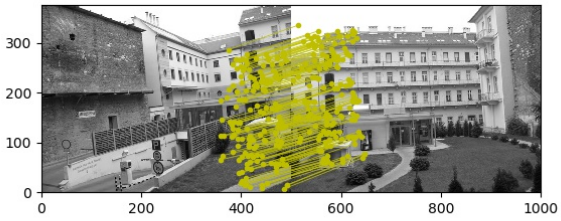

Figure 11: Inliers backyard1 and backyard2

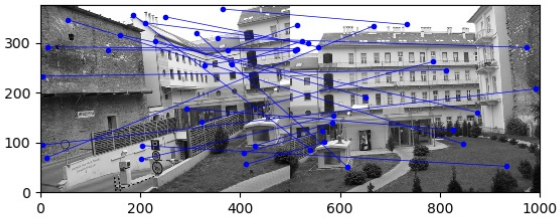

Figure 12: Outliers backyard1 and backyard2

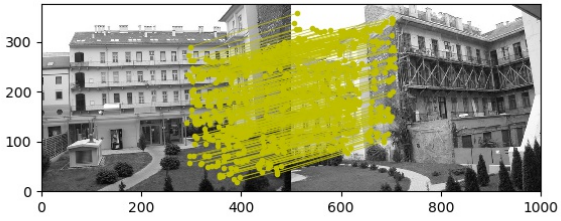

Figure 13: Inliers backyard2 and backyard3

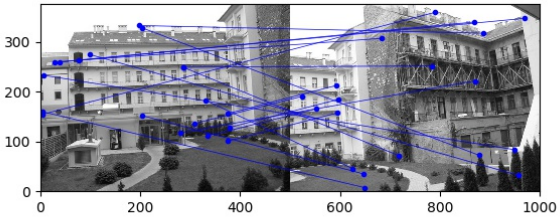

Figure 14: Outliers backyard2 and backyard3

Office

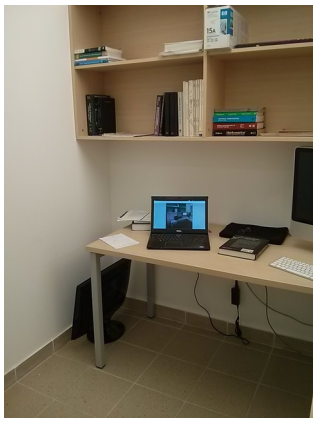

Figure 15: office-1

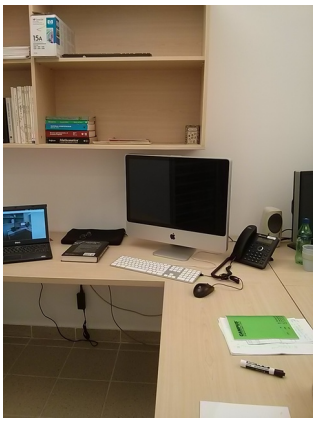

Figure 16: office-2

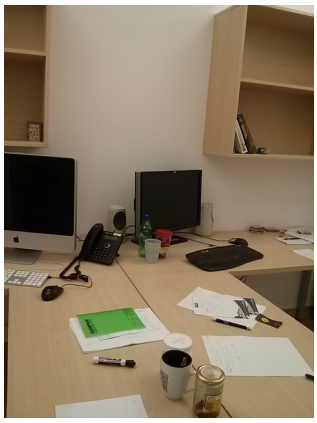

Figure 17: office-3

Figure 18: office-4

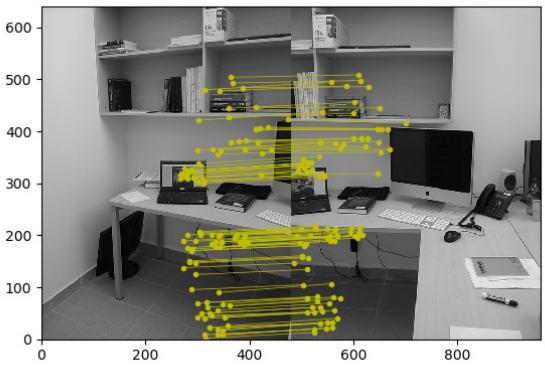

Figure 19: Inliers office1 and office2

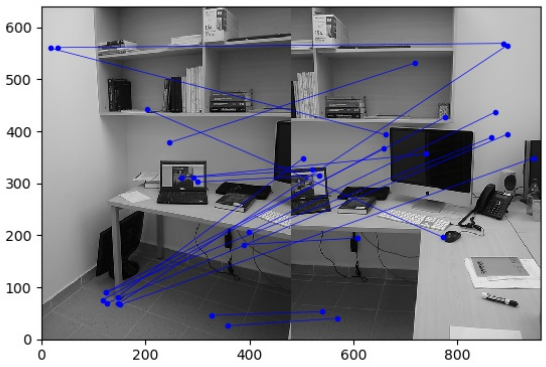

Figure 20: Outliers office1 and office2

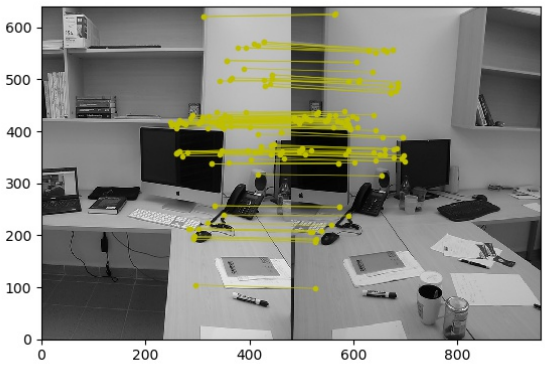

Figure 21: Inliers office2 and office3

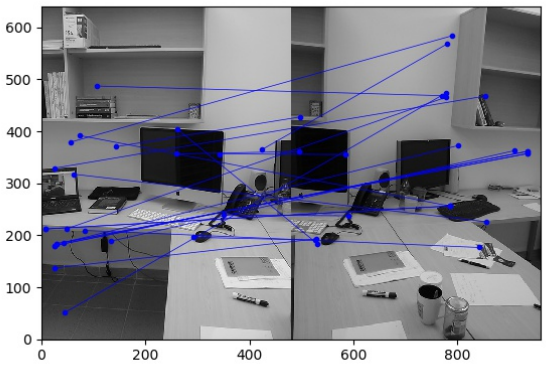

Figure 22: Outliers office2 and office3

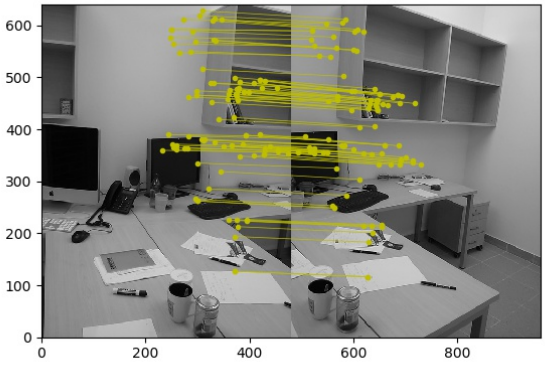

Figure 23: Inliers office3 and office4

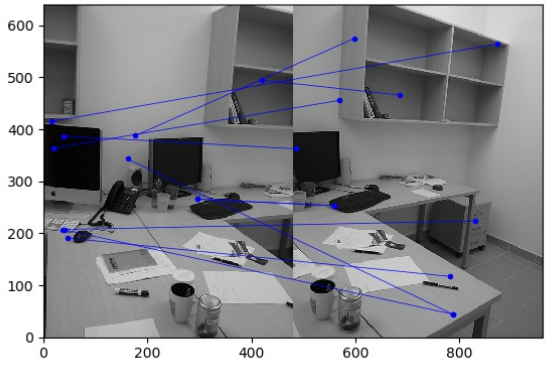

Figure 24: Outliers office3 and office4

Panorama

Figure 25: Panorama Backyard

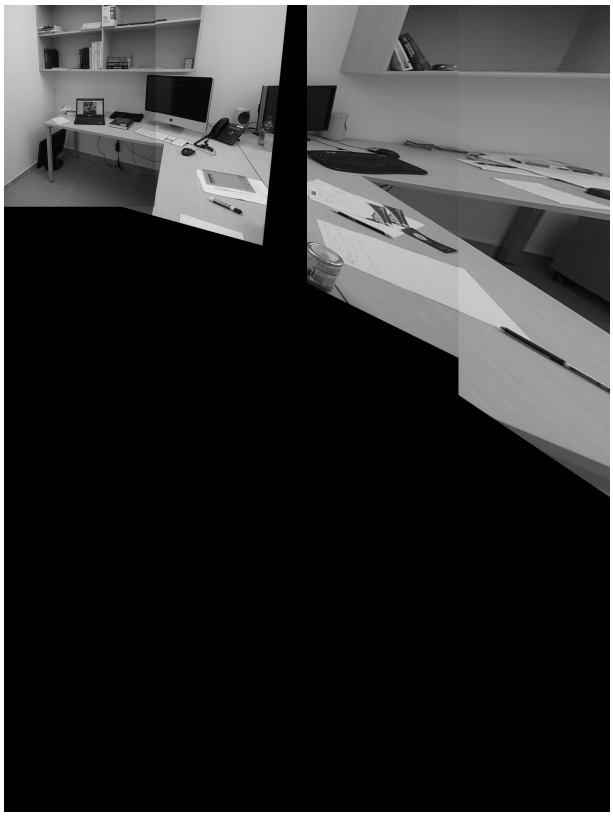

Figure 26: Panorama Office