Computer Vision: Image Alignment and Stitching

Sylke Gosen, Victor Zuanazzi, Simon Passenheim, David Speck

March 13, 2019

1 Introduction

For this assignment, we will be taking a look at how images can be aligned by matching SIFT key points in both images and finding the affine transformation that most accurately transforms these key points from their positions in one image to the position of their match in the other image.

Next, we examined how this alignment algorithm can be used to do image stitching, that is combining multiple pictures of the same scene into one bigger picture. The results we got for that are surprisingly good which is in line with the fact that this type of alignment is still the state of the art when it comes to image stitching for panorama photography.

2 Image Alignment

Question 1.2

Figure 1: Random subset of matching SIFT keypoints with lines drawn in between.

We can see in figure 1 that most pairs are fairly accurate, but there are definitely some errors.

Question 1.3

To do the custom warp we first transformed the four corners of the image to find out how big the resulting image should be. Next, we created a black image with the required dimensions.

Then there are two ways of transforming colors from the original image to the transformed image. The most naive way is to simply transform every pixel index from the original image and plot its color in the closest (using rounding) pixel in the transformed image. This has the downside that some pixels could map to the same location in the transformed image and there is no guarantee that every pixel in the transformed image will end up with a color, causing black artifacts.

The other way of doing this, which solves the problem of black artifacts, is mapping pixels from the transformed image back to the original image and picking the color from the closest (using rounding again) pixel or black if it lies outside

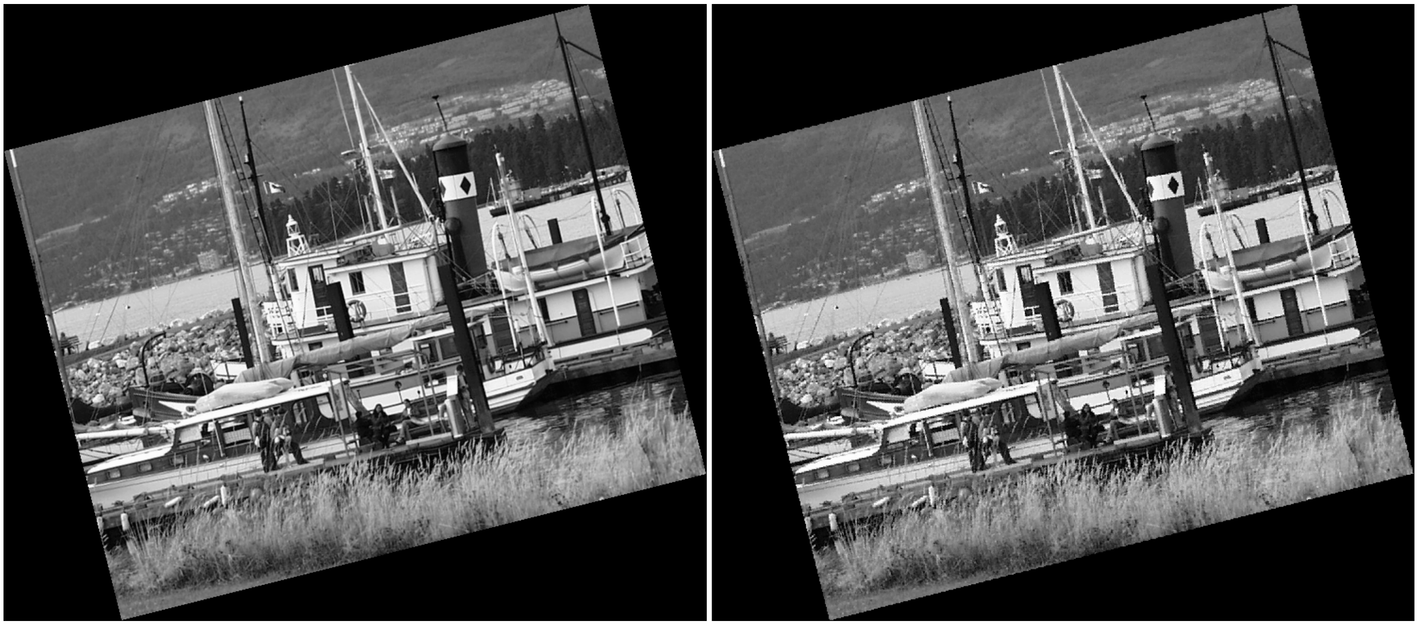

Figure 2: Image 1 transformed using MATLAB imwarp function (left) and custom warp function (right).

the original image. This ensures that all pixels will have a color, but can cause duplicates of original pixel colors. This roughly approximates a linear interpolation and is a lot easier to implement.

It is hard to spot in images at this scale, but we can see in figure 2 that the built-in imwarp function results in a slightly smoother image.

Question 2.1 The linear system we are trying to solve has 3 unknowns. An angle and a translation in the x and y direction. This means we also need 3 matches to find a proper solution.

Question 2.2 This depends on the images, matches that are found and the definition of good. We decided to define good as more than 900 inliers out of 947. If we then repeat the RANSAC algorithm 100 times with 10 random matches we get the following histogram for the required number of trials needed to get a good transformation.

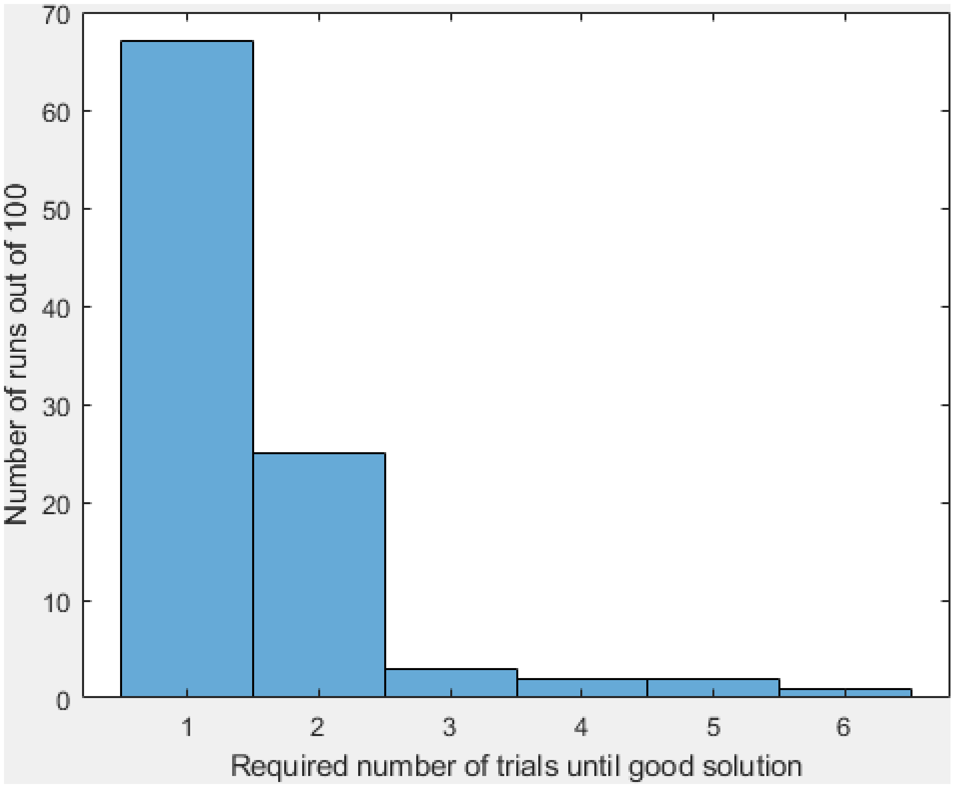

Figure 3: Number of trials required to reach a good (> 900 inliers) transformation.

We can see that in the vast majority (%92) of the cases the RANSAC algorithm finds a good transformation in 1 or 2 trials and it always found a good transformation within 6 trials (at least for these 100 runs). This is a testament to the quality of the matches that are found for this image pair.

3 Image Stitching

In this task, two images are stitched together. The original images can be seen in figure 4, where the left image is on the left side and the right image on the right side. The procedure to stitch two images works as follows:

1. Convert both images to grayscale
2. Find key matching points between both images
3. Find with RANSAC rotation and translation of the right image in terms of left image with the matching points determined in the previous step
4. Express corners of right image in the coordinate system of the left image and extend the frame of the left image such that the corners of the right image fit into the extended image
5. Use Matlab inbuilt function `imwarp` to warp the right image and shift it to the place such that its coordinates are now given in terms of the left image
6. Overlay both images with a mask

The result of the stitching procedure can be seen in figure 6 and the intermediate images of step 4. and 5. can be seen in figure 5. The right image loses a bit of quality and becomes blurrier due to the warping. However, both images flow quite fluently into each other and there are hardly strong cuts visible.

Figure 4: Left and right original image

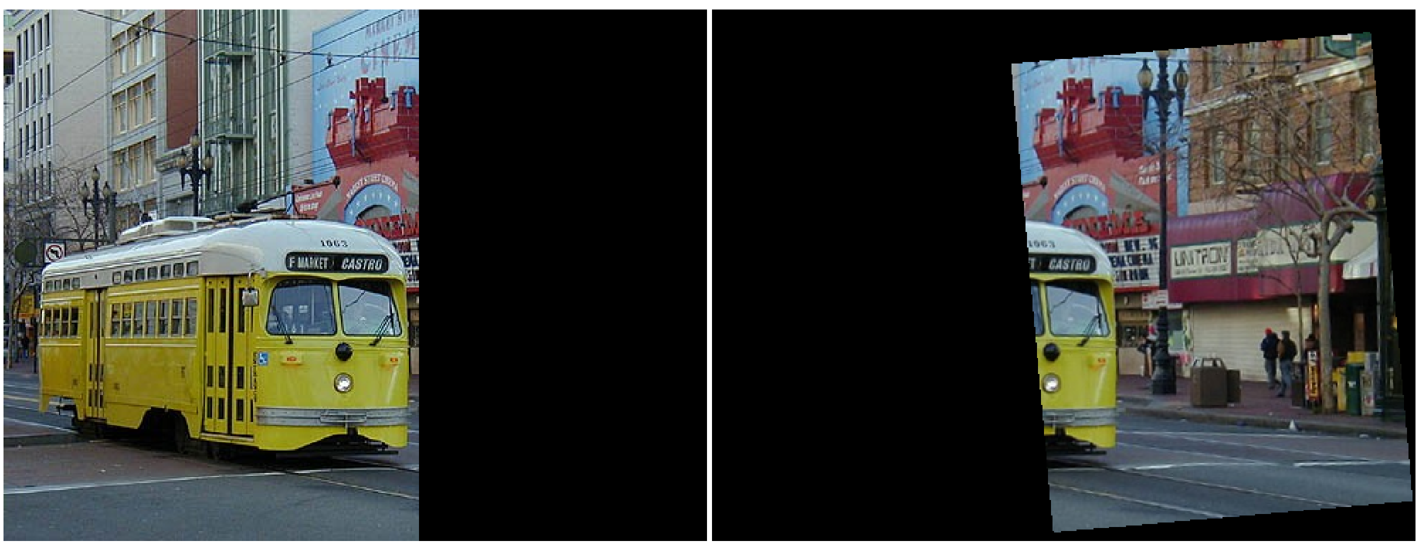

Figure 5: Intermediate result for stitching procedure. On the left side is the extended image, such that the corners of the warped right image fit into the original left image. On the right vice versa.

Figure 6: Stitched image

4 Conclusion

For image alignment using matches between SIFT key points, we found that at least 3 of these matches are required to find a proper transformation because there are 3 unknowns and that the required transformation can be approximated by solving the linear system for a small number of matches, repeating that several times and picking the transformation that transforms most of the matched key points to within 10 pixels of their counterpart. This procedure is known as RANSAC. If the found matches are good such as was the case for the image pair we used, RANSAC can reliably find a good transformation with only 2 repeats.

Using the same technique to match key points we proceed to apply it to stitching two images in order to form a panorama. The procedure used finds the key points, then uses them to find the rotation and translation matrix. With those parameters, the images are rotated and placed against each other in such a way that the key points match. By last a mask is used to produce the panorama image. By human eye, it is hard to find where one image ends and the other starts.