UNIVERSITY OF AMSTERDAM

Assignment 4 Report

Luca Simonetto - 11413522
Heng Lin - 11392533
Computer Vision 1

March 12, 2017

Image Alignment

For this task we implemented an algorithm that manages to detect and align the interest points of two images, by calculating the optimal transformation between the two. After having detected all the interest points, the RANSAC algorithm is run, in order to search and improve the values of the transformation matrix A, that defines rotation and scaling, and the translation matrix b, indicating rigid movement of the pixels.

In order to generate the pixels values for the new image, a number of steps are taken:

- first, the boundaries of the new image are calculated, by computing the transformed coordinated of the four original corners and by determining the maximum and minimum values for x and y.
- the pseudoinverse of the matrix A is computed, as our method maps each pixel of the new image to one pixel of the original one: in order to accomplish this, the coordinate of the originating pixel is calculated by using the following formula:

$$\begin{bmatrix} x \\ y \end{bmatrix} = round\left(A^{\dagger}\left(\begin{bmatrix} x' \\ y' \end{bmatrix} - t\right)\right)$$

  As the transformation generates two decimal numbers, their value is rounded to the nearest whole number.
- each pixel coordinate of the new image is considered, the coordinate of the original pixel is determined and the correct color value is associated.

Having all the necessary data, we generated a new image using this matrices, in order to compare the result with the matlab built in functions and the given result image. The input images are shown in Figure 1 and Figure 2, while the results are shown in Figure 6 and 4

We can see that the results are quite comparable, indicating a successful implementation of the algorithm. In order to determine the quality of the output, we checked the number of inliers that the algorithm manages to correctly transform to the second image (correct is defined as being inside a 10x10 pixel window around the correct location) and we compared them to the total number of interest points available: our algorithm manages to correctly transform on average 96% of the total points. We also checked the number of iterations needed by the algorithm and the results were quite surprising: using 3 random points for each iteration, the

Figure 1: Input image 1

Figure 2: Input image 2

Figure 3: Algorithm output

Figure 4: Matlab (built in functions)

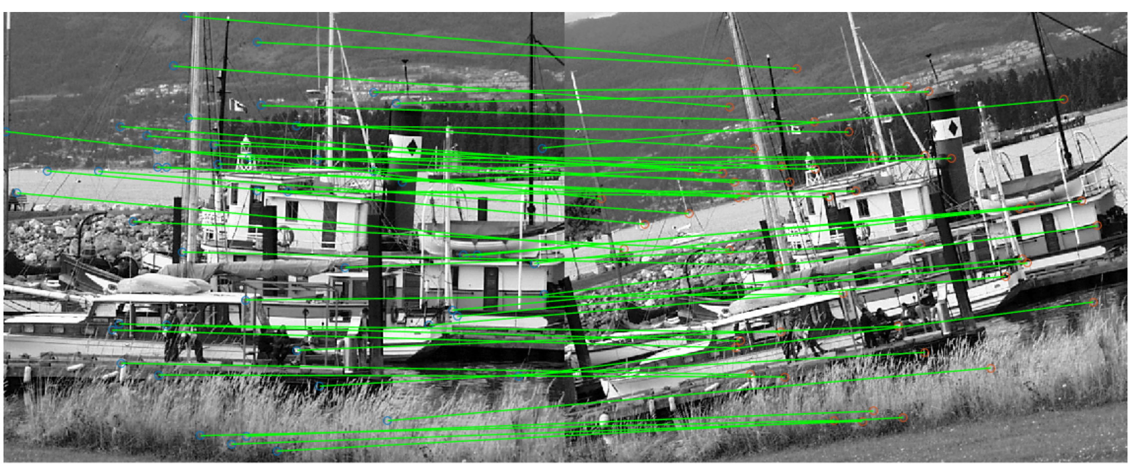

Figure 5: Matches of 50 random points of interest

above mentioned score is, on average, reached at the first iteration, meaning that the basis on which it has been built are robust.

In order to understand the functioning of the described method, Figure 5 shows the transformations made to a random subset of 50 points, that are connected to their calculated positions in the second image

Figures 7 and 11 show the results for the execution of the algorithm working from the second image to the first one.

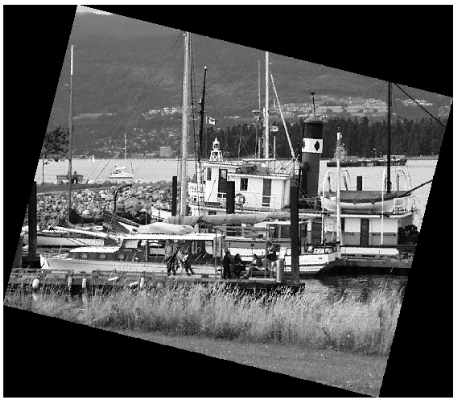

Figure 6: Algorithm output

Figure 7: Matlab (built in functions)

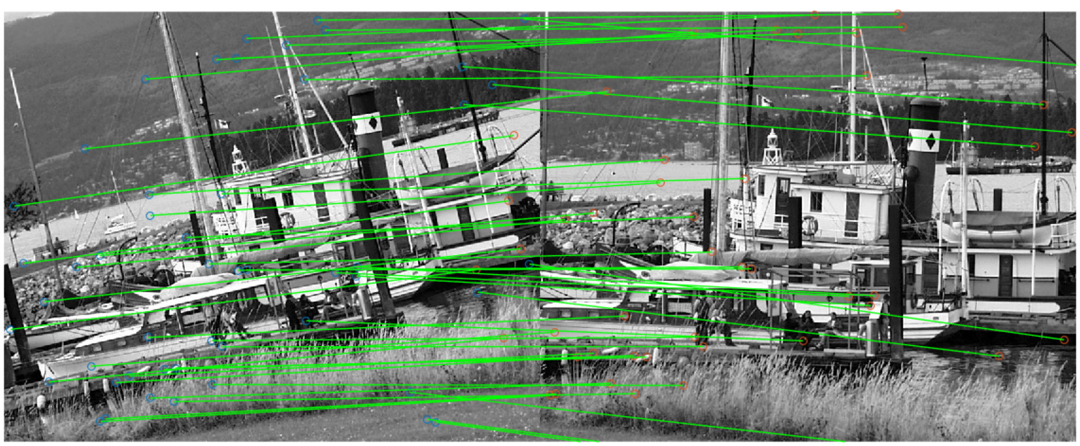

Figure 8: Matches of 50 random points of interest

Image Stitching

This task uses the above algorithm in order to stitch together two images of the same subject, creating a panorama image that combines the two. Very similarly to the image alignment problem, this task first computes the interest points of the images and then applies the RUNSAC algorithm in order to find the best values for A and t. With these matrices a new image is created by interpolation of the original pixels.

Figure 9 and 10 show the two input images used for this task: this time it can be easily noticed how the main subject is cropped in the second image, but containing enough information to be stitched to the first one.

Figure 9: Left input image

Figure 10: Right input image

Figure 11: Stitched images