Assignment 4: Computer Vision 1 - Assignment 4
Image Alignment and Stitching

Annelore Franke 2141469
Arash Parnia 11431482

March 13, 2017

Image Alignment

VLFeat: feature points and feature matching

The first part of this 4th assignment consists of *image alignment*, for which we wrote a function that takes 2 images as input and computes the affine transformation between them. In order to perform the first few steps of the *image alignment*, VLFeat had to be installed, to able to use the sift functions `vl_sift` and `vl_ubcmatch`.

Features of both images were gotten with the `vl_sift` function. The feature points of the first boat image can be seen in figure 1. The same goes for the second boat image and its feature points, which can be seen in figure 2. After that, the `vl_ubcmatch` function was used in order to match the features, which resulted in the form of matches of two sets of SIFT descriptors. In figure 3 you can see the result of the feature matching of the two boat images.

Figure 1: boat1.pgm with its matching points

Figure 2: boat2.pgm with its matching points

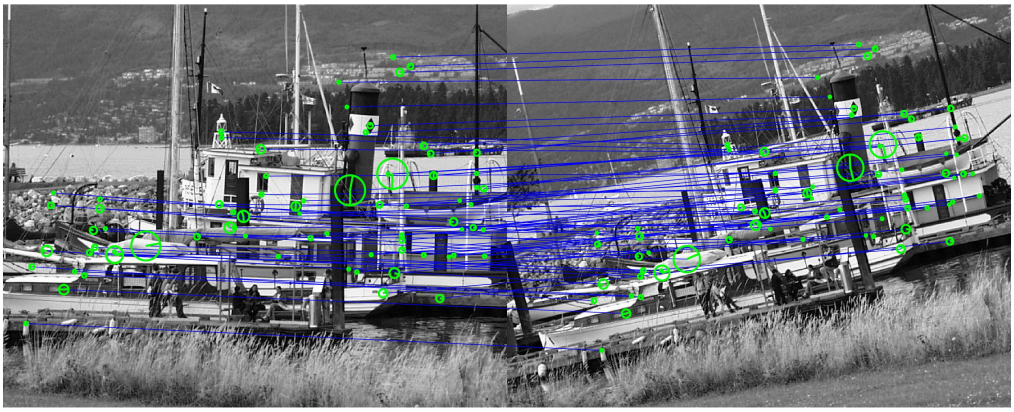

Figure 3: Matched features of both images boat1.pgm and boat2.pgm

RANSAC

After these first few steps, which were performed using the VLFeat functions, the final stage had to be carried out using the RANSAC fitting algorithm. We started off with picking random matches

which gave us the index in f1 and f2. These f1 and f2 features, obtained through the `vl_sift` function, gave us the index for sift features. From those indexes we found the x and y of the first and second image.

Matrix A was made for the first image, matrix B was made for the second image. In order to get matrix X, the equation was solved using `pinv` of matches $a*b$. Matrix X will have the transformation parameters *m1, m2, m3, m4, t1 and t2*, where *m1* to *m4* are the affine rotations and *t1, t2* are the translation points. We then went through all the pixels of the first image, in order to get the x and y and be able to create matrix A, to then multiply by the matrix X. Now that the x and y points are obtained, we check if a point is near the matching point of the second image. This is then done for every single feature point. The more feature points we have, the better matrix X is, to approximate the transformation. After that, the best matrix X is returned.

Now, from the matrix X, we use *m1, m2, m3 and m4* to create an affine transform matrix and use it to rotate the image to make the first image look like the second image. Calculating the matrix X we have to use at least 3 matches (P=3) from each image and concatenate those together in order to get a good approximation of matrix X.

The results of the RANSAC fitting algorithm can be seen below. The best transformations that were found are shown in figures 4 and 5. In figure 4 you can see the transformation image from `boat1.pgm` to `boat2.pgm`. In figure 5 you can see the transformation image from `boat2.pgm` to `boat1.pgm`.

Figure 4: Transformation from boat1.pgm to boat2.pgm

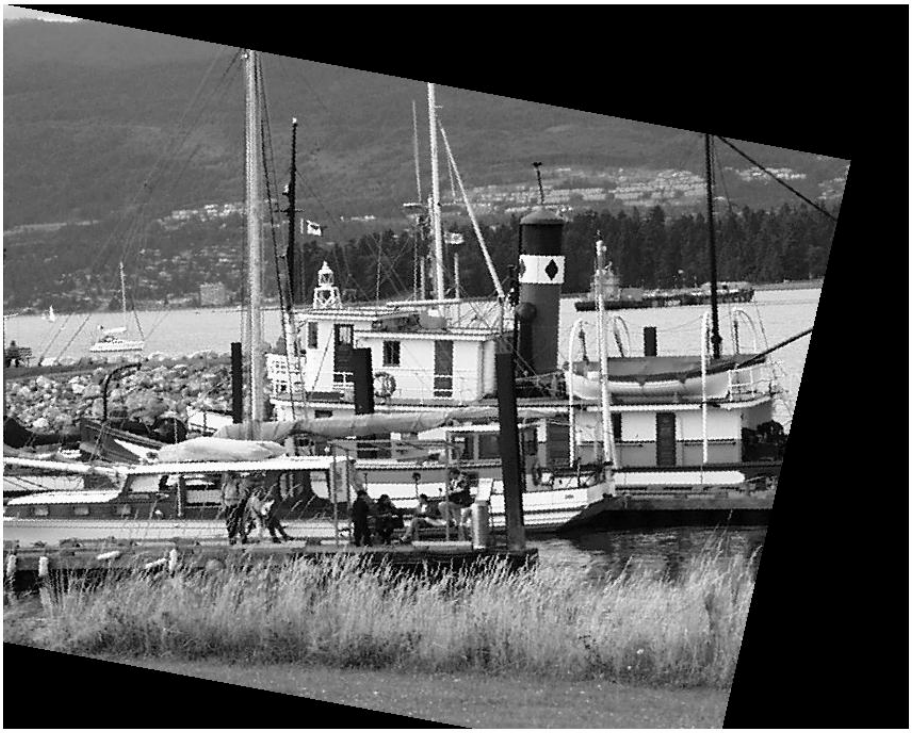

Figure 5: Transformation from boat2.pgm to boat1.pgm

How many iterations on average are needed to find good transformation parameters?

On average, we can say that 7 iterations has the highest amount of matches (947).

Image Stitching

The second part of this 4th assignment consists of *image stitching*, for which we wrote a function that takes two images as input and stitches both images together. First, the best transformation between the two images has to found. Second, the size of the stitched image has to be estimated. Lastly, the first image and the second transformed image need to be combined in one image.

In order to perform the image stitching, we have taken the feature points of the first image and matched it to the feature points of the second image. After the matching, the transformation matrix was made, using RANSAC from the first part of this 4th assignment. The affine transformation matrix was then used to transform the second image (`right.jpg`). The matching feature points have been overlapped in order to create a new stitched image. In figure 6 you can see the result of the image stitching, the combined image from `left.jpg` and transformed `right.jpg`.

Figure 6: Combined image from left image and transformed right image