COMPUTER VISION 1 ASSIGNMENT 4

MAURITS BLEEKER (10694439) & JÖRG SANDER (10881530)

13 March 2016

1 QUESTION 1 – RANSAC IMAGE ALIGNMENT

1.1 How many matches do we need to solve an affine transformation which can be formulated as shown in Figure 1?

A 2D affine transformation contains six unknown variables e.g. $m1, m2, m3, m4, t1, t2$ as is noted in the description of assignment 4. A *point* in 2D is characterized by two variables e.g. $(x1, y1)$. Therefore if one specifies 3 points in the original image with corresponding matches in the second image the six linear equations can be solved. Hence we need 6 points (3 matchpoints) to solve for the affine transformation.

1.2 Compare the results of *your own* affine transformation with the results obtained with the build-in MATLAB functions *imtransform* and *maketform*

Figure 1 shows the result of the affine transformation from img1 to img2 (and vice versa). We found 947 matchpoints from which the *optimal transformation* could transform 904 relatively [1] accurate. A qualitative comparison (with the bare eyes) of the MATLAB transformation [2] with our own developed transformation (both using the optimal parameters) reveals good results. Both transformed images match the original second image. The total size of both transformations (MATLAB and our *custom* one) is larger than the original image due to the skewness of the affine transformation. Differences between the transformations are only visible if one zooms into the images so that each pixels becomes visible. We think that these differences can be presumably explained by a difference in the *nearest-neighbor interpolation*. The same results can be observed for the transformation from img2 to img1 which are shown in figure 2.

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

If we define "good transformation parameters" in the sense of *the number of inliers that could be correctly transformed within a radius of 10 pixels in the target image* than we can even find good parameters with 50 iterations although the

1 within a radius of 10 pixels
2 we used the build-in function *maketform* and *imtransform*

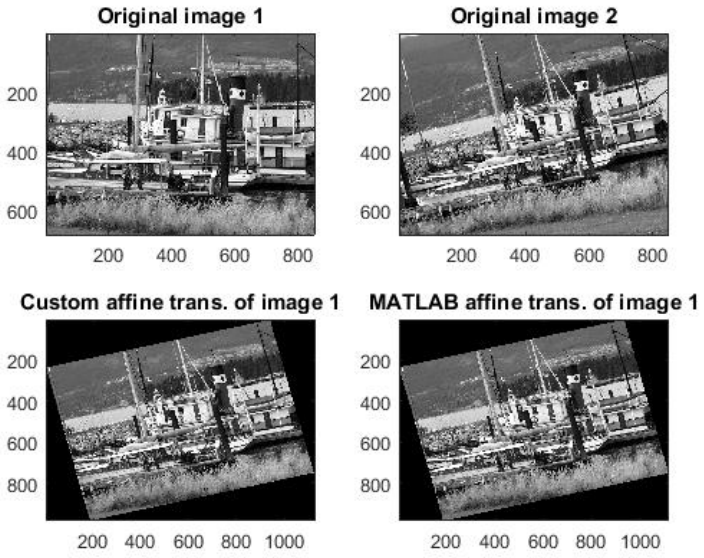

Figure 1: Results for transformation from img1 to img2 with *optimal parameters determined by RANSAC*

probability that we find them is low. The search for the optimal parameters is getting relatively stable around 200 to 300 iterations.

Figure 3 on the following page shows 50 randomly selected match points from the set of *inlier match points* determined by our RANSAC algorithm implementation for the two *boat* images (MATLAB function *plotMatchingPoints*).

2 IMAGE STITCHING

For this question we developed a function that takes two images as input. The first thing we do is calculating the transition matrix between input image one and two.

The result of the images stitching function is not always guaranteed to be correct due to the probabilistic nature of the RANSAC algorithm which can provide a suboptimal transformation matrix. As mentioned above, when setting the number of iterations to at least 200-300 hundred results in a high probability of finding an optimal set of transformation parameters.

After calculating the transformation between images one and two, we transform the second image by means of the optimal transformation parameters determined by RANSAC.

Based on the transformation matrix we can calculate the (minimal) size of the new image. We use the translation parameters t1 and t2 from the transformation matrix to determine the stitching positions for the second image. t1 is the x offset for images 2 in the 'new images'. t2 if the offset for the y direction.

Figure 4 shows the result of the image stitching function for *left.jpg* and *right.jpg*.

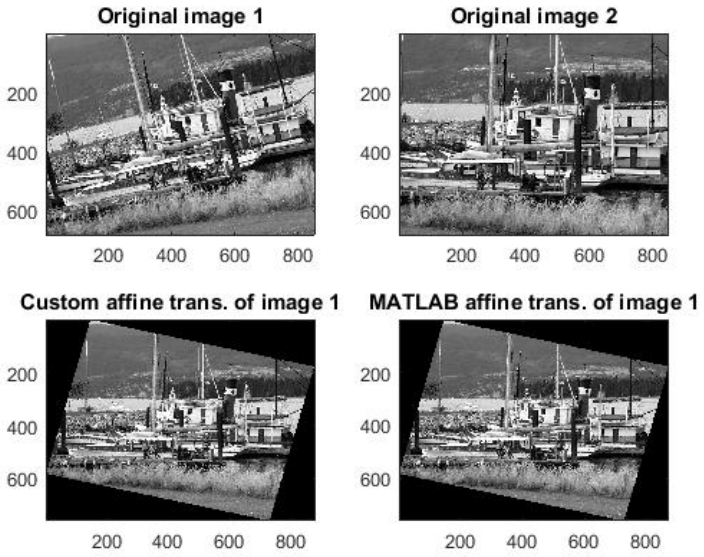

Figure 2: Results for transformation from img2 to img1 with *optimal parameters determined by RANSAC*

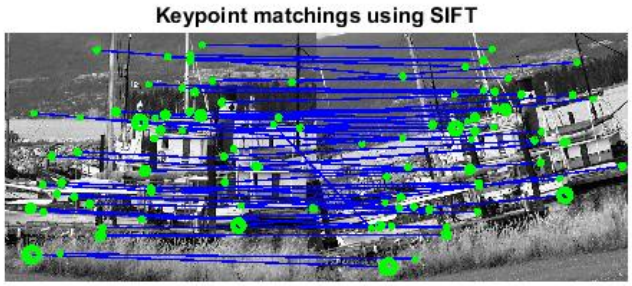

Figure 3: Keypoint matchings using SIFT implementation of Andrea Valdadi

Figure 4: resulting segmented images of left.jpg and right.jpg for our image stitching function