

EE838 Assignment 7

Jonas Nikula
20176392

1 Implementation steps

First we have to find the interest points. I do this using the built-in Computer Vision Toolbox, and its "detectSURFFeatures" function. For putative feature correspondences I created a simple SSD algorithm.

After we have obtained the interest points and putative correspondences, we can proceed to use the RANSAC algorithm to try and find the inlier matches. First we randomly select 4 correspondences. We check them for collinearity; If they are collinear, we randomly select another 4 correspondences, until we find 4 correspondences that aren't collinear.

We calculate a homography from the correspondences using the normalized DLT method. We then count how many of the correspondences conform to this homography. The conformance is calculated by transforming a point from a correspondence pair by the homography, and calculating the distance to the actual second point, and vice versa, and summing the distance. This is also called a symmetric transfer error. For the inlier calculation I use a distance threshold of 450. We also calculate the standard deviation of distances.

We then compare the number of inliers to the previous best; If we have more inliers, the current homography and number of inliers is recorded as the best. In the special case where the inlier count is the same as the previous best but the standard deviation of distances is better, ie. smaller, we record that as the new best result. At the end of the cycle we calculate a new N, which is the number of iterations to calculate.

After the RANSAC algorithm has finished we should have a homography that performs decently. However, we also use the recorded inliers to calculate the optimal homography using the Levenberg-Marquadt algorithm to minimize the error.

2 Results

The results varied quite a bit, which I guess is somewhat expected due to the iterative nature of RANSAC. Still, there was considerable variance between runs.

The parameter with the biggest influence on results was the distance threshold. Smaller thresholds usually come up with the best results, but they were also far more unstable. Larger thresholds were more stable, but never gave any really good results.

Choosing the threshold also seems to be dependent on the quality of the feature extractor; Features from a bad extractor have more outliers, and thus it's preferable to have a smaller threshold. And vice versa with a good extractor. Unfortunately, it also seems that the threshold needed to be selected according to the picture pair, which is quite undesirable behaviour.

Table 1 contains the resulting homographies for both image pairs.

Table 1: The best homographies for both image pairs, obtained by the RANSAC algorithm

Table 2: H1 image pair best homography		
5.22E-01	-1.49E-01	6.06E+01
-5.49E-02	4.98E-01	1.07E+02
-9.04E-05	-1.65E-04	7.53E-01

Table 3: H2 image pair best homography		
3.28E-01	-4.11E-03	2.39E+02
2.63E-02	2.65E-01	1.11E+02
1.11E-04	-3.94E-06	4.52E-01

2.1 Visualization

2.1.1 Image pair H1 visualization



Figure 1: Found interest points

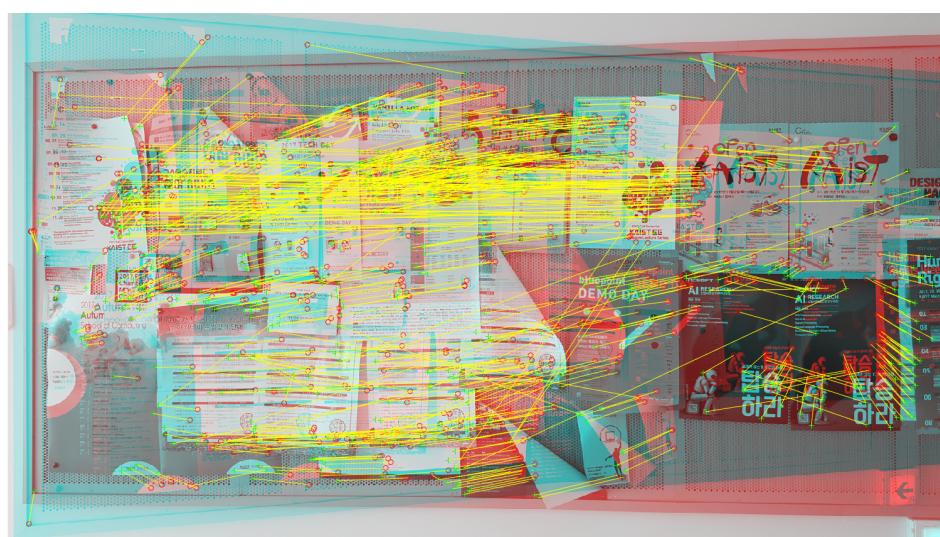


Figure 2: Matches between interest points

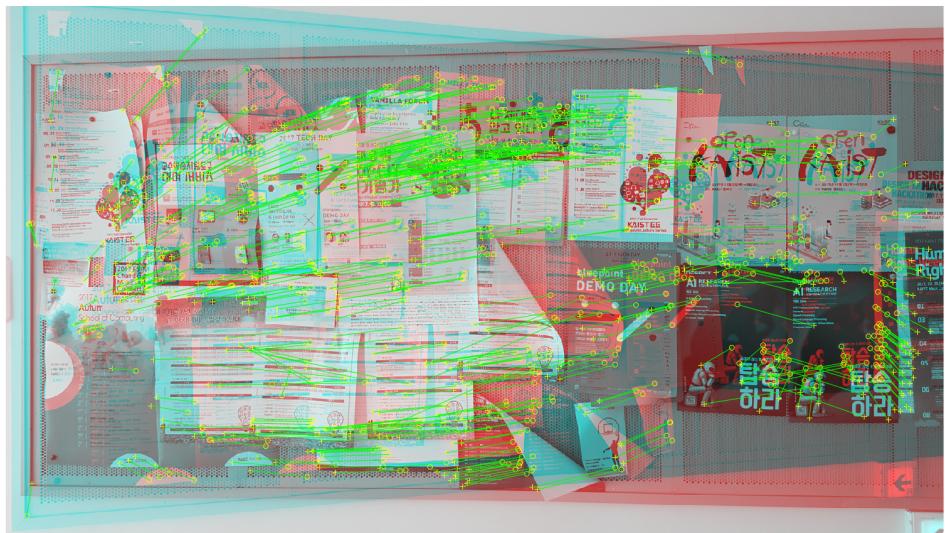


Figure 3: Inliers from the best fitting homography obtained by RANSAC



Figure 4: Outliers from the best fitting homography obtained by RANSAC

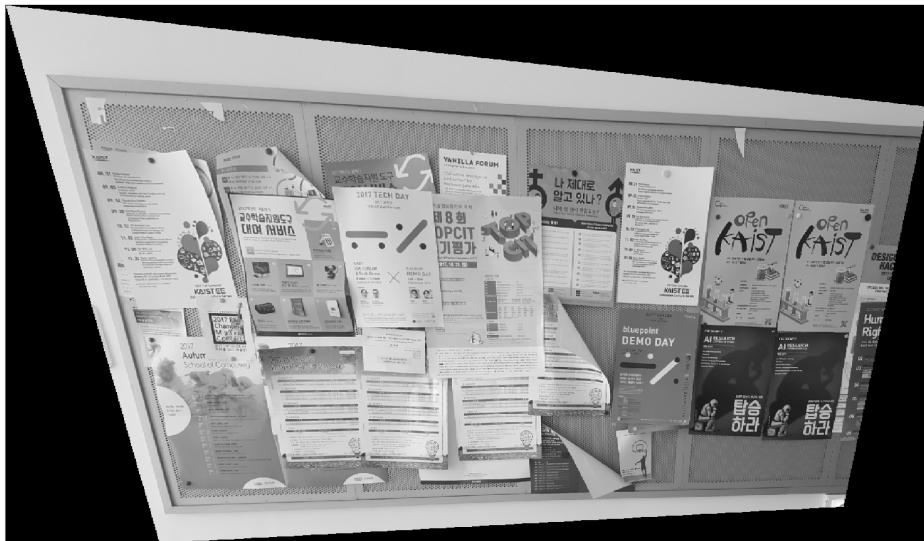


Figure 5: The second image transformed by the homography obtained with RANSAC



Figure 6: The second image transformed by the homography obtained from the inliers using the Levenberg-Marquadt fitting algorithm

2.1.2 Image pair H2 visualization



Figure 7: Found interest points



Figure 8: Matches between interest points

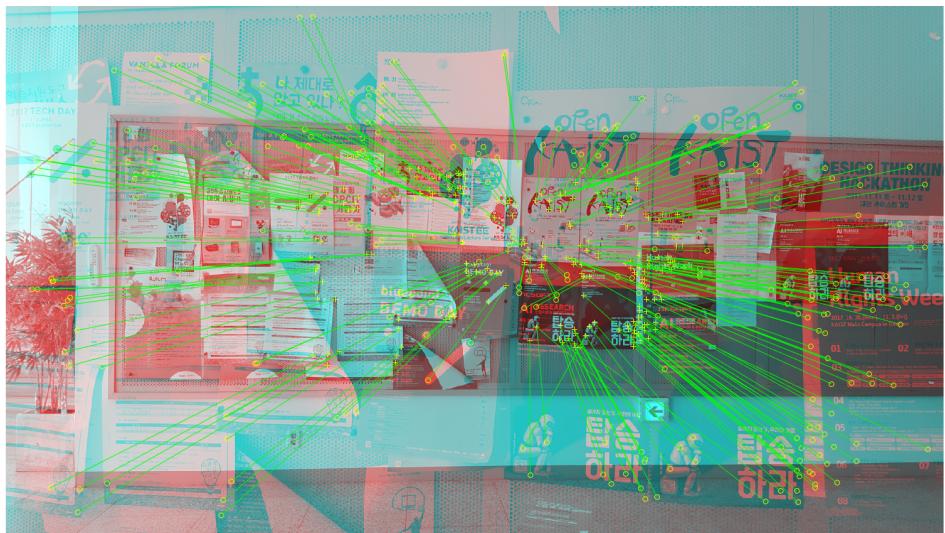


Figure 9: Inliers from the best fitting homography obtained by RANSAC

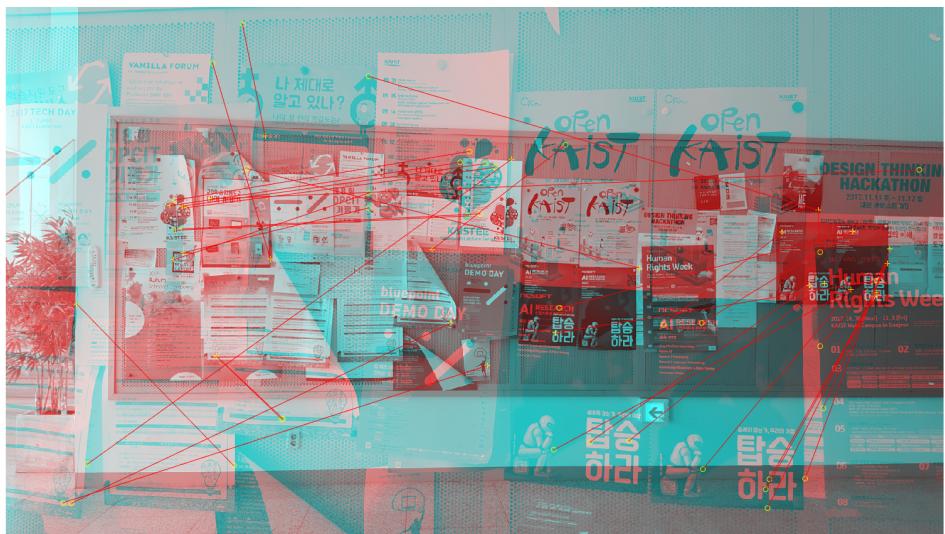


Figure 10: Outliers from the best fitting homography obtained by RANSAC



Figure 11: The second image transformed by the homography obtained with RANSAC



Figure 12: The second image transformed by the homography obtained from the inliers using the Levenberg-Marquadt fitting algorithm