

Robust Estimation-Automatic computation of H

Computing homography (H) can be done using normalized DLT. In order to compute the DLT we use few selected random points (4 points for homography). We compute the H and check whether other points meet the threshold according to the model found in DLT. We keep updating the best possible model using RANSAC algorithm.

A. Describe the method of each step

- '*harrisCorner.m*': Computing the interest points using Harris corner detection algorithm. Using the image gradient (sobel operator) we can compute the R value for cornerness of every pixel. We extract only the positive big value of R and merge the value using maxima suppression window size of 9.
- '*nccMatching.m*': first compute the normalized cross correlation of every possible feature point matches with patch size of 15.

$$NCC(i, j) = \frac{\sum_W (I_1(\tilde{\mathbf{X}}) - \bar{I}_1)(I_2(\tilde{\mathbf{X}}') - \bar{I}_2)}{\sqrt{\sum_{W(\mathbf{x}_i)} (I_1(\tilde{\mathbf{X}}) - \bar{I}_1)^2 \sum_{W(\mathbf{x}'_j)} (I_2(\tilde{\mathbf{X}}') - \bar{I}_2)^2}},$$

By comparing every value of the NCC we give the appropriate match for feature matching. This was computed similarly to bipartite graph matching

- '*RANSAC_H.m*': first we extract 4 points to compute H using DLT. For accurate result, we normalize the value of the points using similarity Transform.

$$s = \frac{\sqrt{2}}{\frac{1}{n} \sum_i \sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2}}.$$

$$t_x = -s\bar{x} \text{ and } t_y = -s\bar{y}.$$

Using those normalized points we then compute H using DLT and then we unnormalize the H using

$$H = T_2^{-1} \hat{H} T_1$$

Check the inlier using H for every matching point and compute the standard deviation.

If the computed inlier number are bigger than the previous largest inlier number update to the new inlier number and inlier points. If the number of inlier is the same as previous inlier numbers then compare with standard deviation and update if the current standard deviation is smaller.

From the inlier number given, recompute the number of iteration of RANSAC.

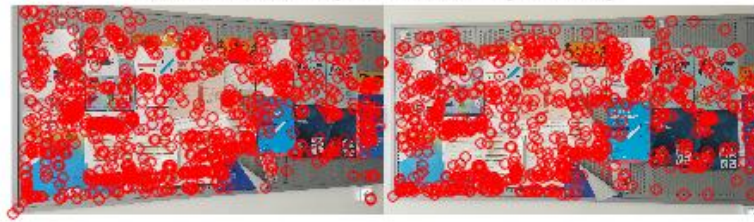
If iteration was reached its limit, stop and **optimize** H homography matrix using all the inlier points. Then using the optimized H, recompute the inliers.

- '*deformImg.m*': using all the coordinate points compute the new point position. Then using matlab function '*griddata*' we deform the image as the other image.

B. Visualize the results of each step

- Interest points

features in board image 1 and board image 2 montage

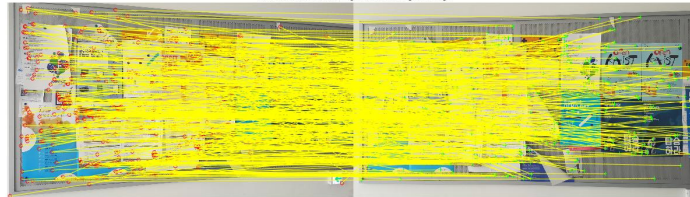


features in board image 3 and board image 4 montage

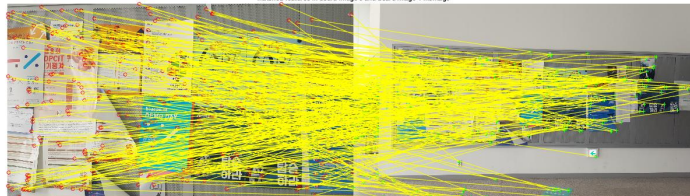


b. Correspondence

matched features in board image 1 and board image 2 montage

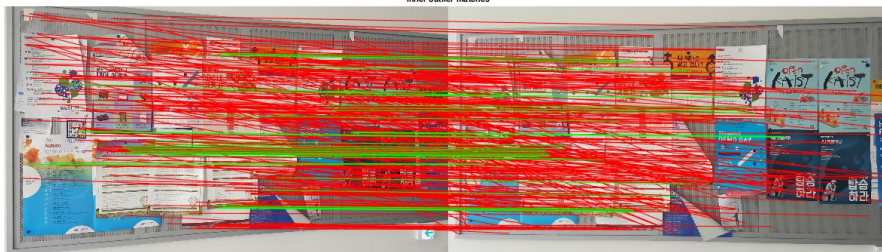


matched features in board image 3 and board image 4 montage



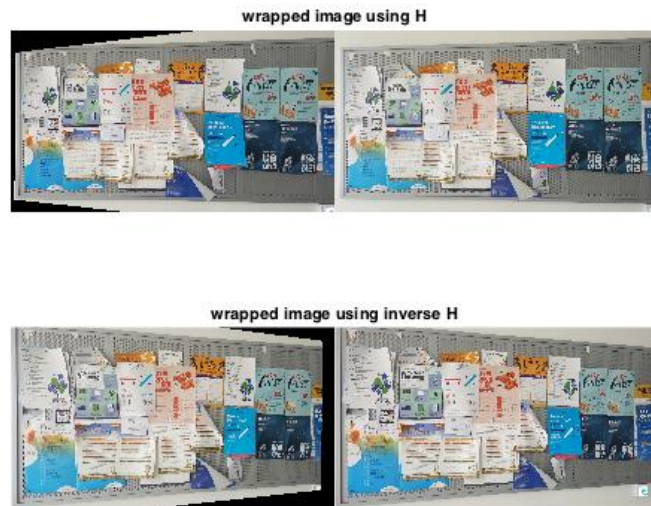
c. RANSAC inlier outlier

inlier outlier matches



*inliers is in green and outlier is in red

d. Image transformed by best H obtained by every inliers



C. Write the best H for each pair

$H = \begin{bmatrix} 0.470751046703718 & -0.00963731320413023 & 21.6336257748270 \\ -0.0734376511710604 & 0.557208161295464 & 84.5166248749717 \\ -0.000136011137209080 & -8.56674440518727e-06 & 0.697188145543299 \end{bmatrix}$

The other second pair H wasn't able to find

D. (BONUS) Analyze the difference in results as each method changes

a. Interest Points

i. SIFT vs Harris Corner: feature extracted numbers can be controlled by the value of the input, but Harris corner features only contain corner and edge information while SIFT extracted more unique features

b. Matching

i. NCC vs SIFT

1. Using NCC did not scale invariant which results in bad results in second image pair. SIFT is scale invariant thus likely to match feature well in scale difference.

E. My Results

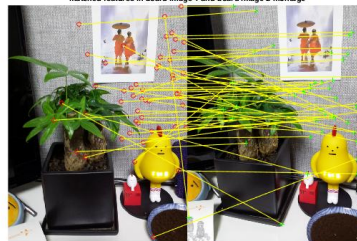
features in board image 1 and board image 2 montage



features in board image 3 and board image 4 montage



matched features in board image 1 and board image 2 montage



matched features in board image 3 and board image 4 montage



