School of Electrical Engineering and Computer Science (EECS)

KTH, Stockholm

[EQ2425] Analysis and Search of Visual Data - AY 2019/2020
Project #1 Report

Image Features and Matching

Federico Favia
Mayank Gulati

Table of contents

Introduction

In this project, the open source Matlab library VLFeat [5] for SIFT features is used. The purpose of the project is to compare the results of SIFT [2] and SURF [3], two methods to find features in images. On the other hand, SURF is implemented through the already available Computer Vision Toolbox [6]. After we have exercised and understood the functions of the library, we started with the first task: testing repeatability for robustness of keypoints with respect to change in rotation and scaling to original image.
As our second task, we had to match keypoints in different images (reference and target images) possessing similar corresponding keypoints for every keypoint in the reference image. We calculated its Euclidean distance with target image in "feature" (descriptor) space and found out the best matches based on various criteria discussed in detail in the following section.

Robustness of Keypoint Detector

An approach to test robustness of keypoint detector and descriptor is evaluating the repeatability measure on the same image modified in several ways. The repeatability is defined as follows:

$$R = \frac{\#matches\ between\ original\ and\ modified\ image}{\#keypoints\ detected\ in\ original\ image}$$

where the numerator increases when every time a match is found, so when a predicted keypoint $[x', y']$ in the modified image (basically it is a transformation on the key points' coordinates from the reference image) satisfies this constraint with the detected keypoint $[x_0, y_0]$ in the modified one:

$$|x_0 - x'| \leqq 2 \ and\ |y_0 - y'| \leqq 2$$

Since the geometric transformations made on the reference image are rotation and scaling, we first thought about rotating and scaling the detected keypoints in the reference and then comparing with the ones in the target. Our main doubt was about the repeatability evaluation over different axis of rotation. Indeed, from the course slides (Fig. 1) [1], we knew that repeatability was compared with respect viewpoint angle of the image (meaningful transformation), while for the report we had to compare with respect a simple 2D rotation of the image from 0 to 360°. Moreover, it was difficult finding a function in Matlab for rotating the viewpoint angle, so we opted for the simple rotation as requested. Then we tried a different method, making use of already implemented matching functions in descriptors space, to compute repeatability. The obtained plots are similar in shape to the final ones because at the end similar keypoints that are close in the location space (the task of the project) would also be having a high probability of being closer in descriptor vectors. However, we followed the instructions, went back to the given strategy, and compute repeatability from a point of view of spatial information. The following plot is also based on scale and/or orientation while our final result is based only on searching for the coordinates of predicted keypoints in the spatial neighbourhood with coordinates of detected keypoints of the target image.

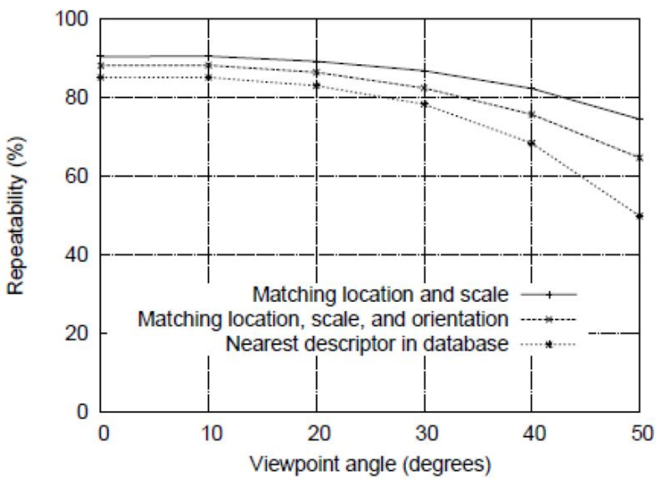

Fig. 1 - SIFT Repeatability versus viewpoint angle

Thresholds of SIFT and SURF (a)

SIFT and SURF keypoint detectors are applied to the image *obj1_5.jpg*, in order to obtain a few hundreds of keypoints. With the default setup, SIFT detects around seventeen thousand keypoints (with default Peak Threshold = 0, Edge Threshold = 10), while SURF finds around six thousand keypoints (with default Metric Threshold = 1000). Since not all of them are significant, we decided to restrict the number of keypoints to a few hundred (we opted for a fixed number of 250). For instance, the most significant are the ones corresponding to the KTH logo (a unique feature of the image) and to some certain corners of windows and roofs.

To do so, for SIFT the Peak Threshold is set at 8, while the edge threshold at 1.8. Increasing the Peak Threshold (keeping bigger peaks of the DoG scale space in terms of absolute value) and decreasing the Edge Threshold (keeping only the peaks of the DoG scale space whose curvature is bigger), we finally obtained fewer keypoints, ensuring we cover the most visually significant keypoints.

For SURF we discovered the possibility of setting a Metric threshold lately, therefore we used the function *selectStrongest(250)*, which anyway for that image it is equivalent to setting a Metric Threshold approximately equal to 8.03×10^3 (minimum Metric of the strongest 250 keypoints). The results are shown below (Fig. 2 for SIFT and Fig. 3 for SURF):

Fig. 2 - SIFT best 250 keypoints

Fig. 3 - SURF best 250 keypoints

From the images, we could say that for both detectors, the biggest number of keypoints come from the KTH logo, the corners of windows and the corner of the roof and chimney. SIFT detector also detects a big scale keypoint for the KTH logo, one smaller for the flag, and some spots on the wall bricks while all these features are not detected completely by SURF detector.

Repeatability versus rotation angle (b)

In the following (Fig. 4) repeatability versus rotation angle with increments of 15 degrees, from 0 to 360, is plotted for the two keypoints detection methods respectively. Both the graphs are periodic because of the intrinsic periodicity of rotation and for 360 degrees rotation repeatability is equal to 1 because it is comparing the image with itself. SURF achieves overall better performance, even though not with higher margin. In particular, the angles where SURF has valleys (where it performs poorly) are different from the angles where SIFT has valleys. On the other hand, the peaks are reached at the same angles (90°, 180° and 270°) and in these SURF performs better (repeatability approximately 1).

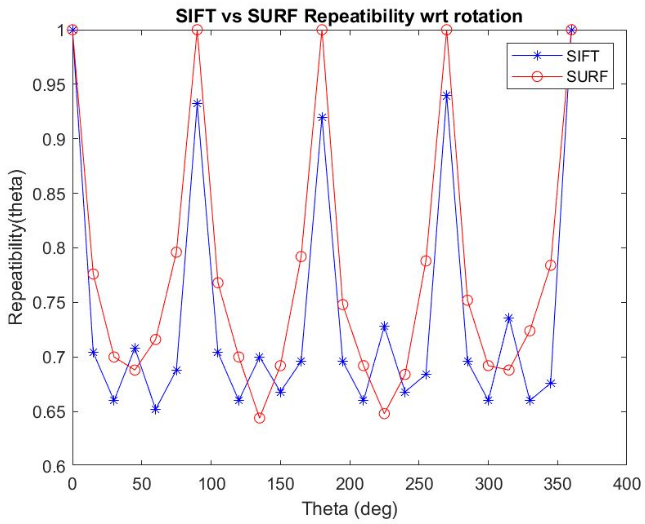

Fig. 4 - SIFT vs SURF - Repeatability versus rotation angle

Repeatability versus scaling factor (c)

In the following (Fig. 5), repeatability versus scaling factor (m^0, m^1, … m^8, where $m = 1.2$) is plotted for the two keypoints detectors respectively.

If we compare between SIFT and SURF in terms of repeatability with respect to scaling, SIFT performs extremely better than SURF. Due to the trade-off between performance and speed SURF is faster and at the same time less effective and robust in finding keypoints in scaled image if compared with SIFT, also mentioned in [4].

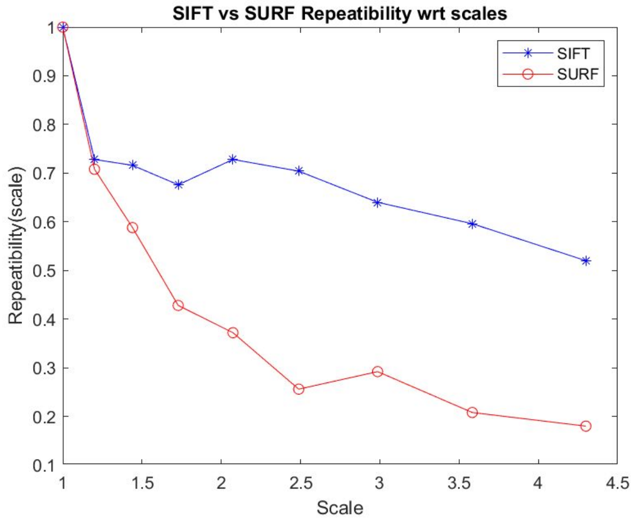

Fig. 5 - SIFT vs SURF - Repeatability versus scaling factor

Image Feature Matching

SIFT keypoints in two images (a)

By using specified Peak Threshold, Edge Threshold and Metric Threshold we calculated few hundred keypoints (250) in two images, the reference *obj1_5.jpg* and the target *obj1_t1.jpg* (same object but photographed in a different viewpoint and light) in SIFT (Fig. 6) and SURF algorithm respectively. For SIFT the keypoints on the two images (a) are shown below:

Fig. 6 - SIFT Keypoints comparison between the reference and target

The keypoints are then matched on the basis of calculating the Euclidean Distance

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

in the descriptor space between keypoints of the reference image and target image, following one algorithm at a time. We used three strategies to compare the images namely fixed threshold (b), nearest neighbour (c) and nearest neighbour distance ratio (d-e).

SIFT Matching - Fixed Threshold (b)

For fixed threshold the algorithm compares in the 128-descriptor space of SIFT for each reference keypoint with every target keypoint. If the euclidean distance between them is below a certain fixed threshold, then a match is found. After computing the minimum (50.52), maximum (141.57), mean (99.04) and median values (98.39) of the euclidean distance metric we started tuning the parameter of the threshold, deciding for an optimal value equal to 62.5 (Fig. 7). Out of a total of 6 found matches, 5 are correct with respect to the ground truth which is the optimal case and for suboptimal (Fig. 8) the threshold distance is 64 with 5 correct matches out of 10 matches. The results are shown below:

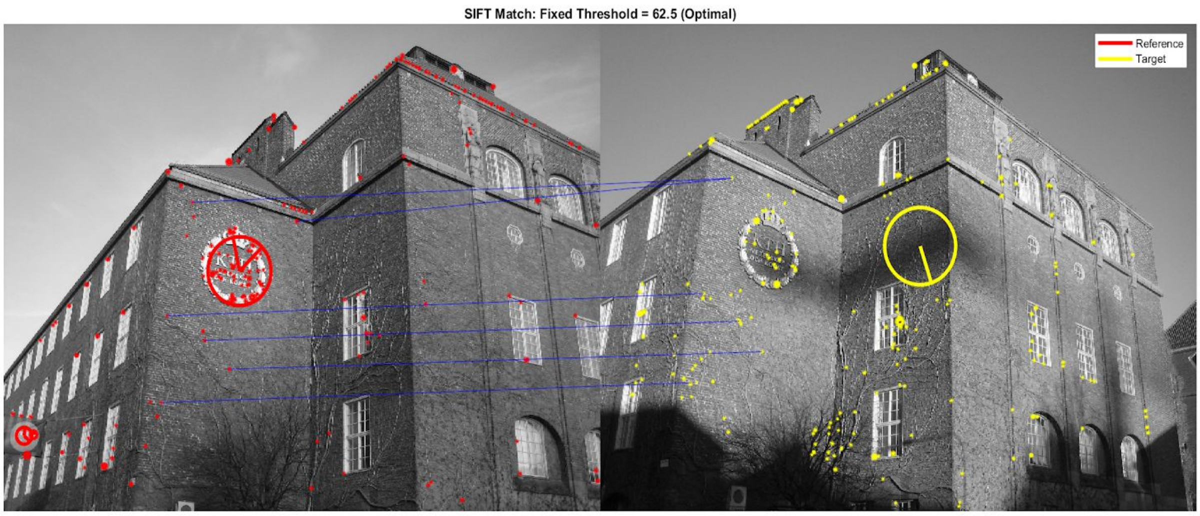

Fig. 7 - SIFT Match Fixed Threshold (optimal case)

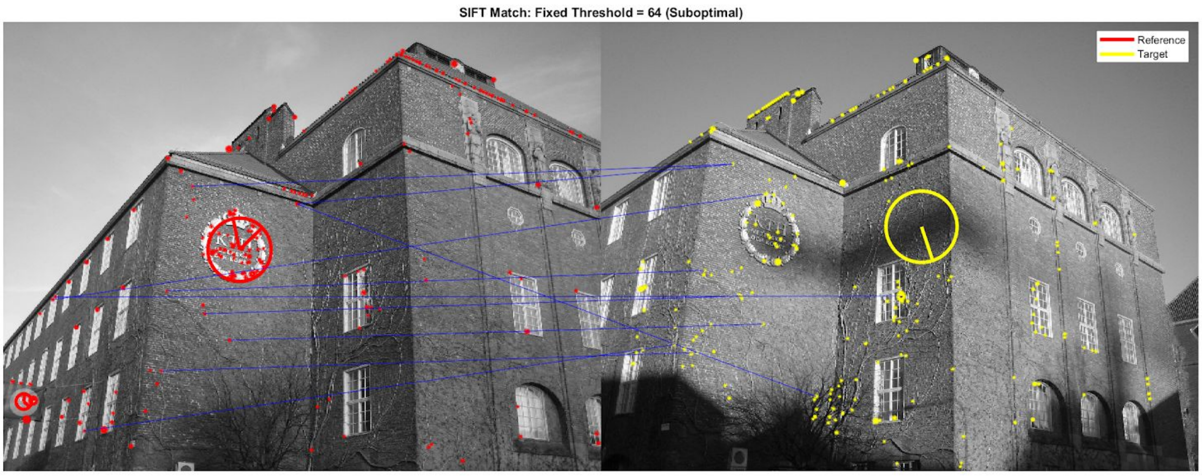

Fig. 8 - SIFT Match Fixed Threshold (suboptimal case)

With these chosen thresholds, we are not satisfied by the result due to the missing of KTH logo match, a unique feature present in both images.

SIFT Matching - Nearest Neighbour (c)

In nearest neighbour strategy we looked for the best match of points possessing least between them in descriptor space. That results in one to one mapping of nearest neighbours seen in Fig. 9. As we obtained 250 matches in the reference image we matches these points with best possible match in target image based on closest euclidean distance which is not always correct. So out of 250 matches more than half are correctly matched especially it considered the KTH logo.

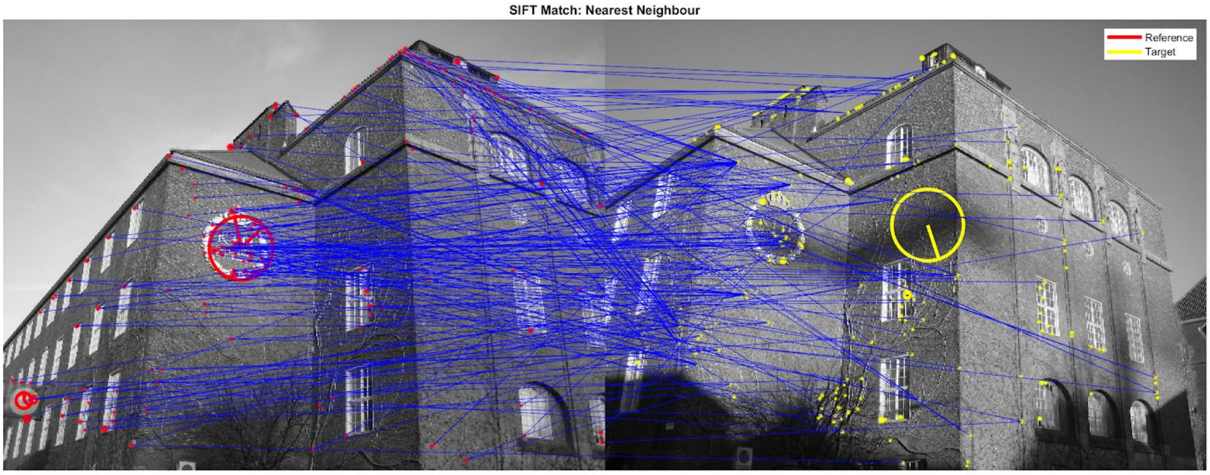

Fig. 9 - SIFT Match Nearest Neighbour

SIFT Matching - Nearest Neighbour Distance Ratio (d)

For nearest neighbor ratio technique for every keypoint in reference we calculated nearest and second nearest in the target image. Using the ratio of nearest and second nearest as 0.89 we filtered the 17 perfect matches for optimal case (Fig. 10) while 0.91 reduced 90 percent of false matches for suboptimal case (Fig. 11) with again 17 correct matches.

$$NNR = \frac{\textit{distance of nearest neighbour}}{\textit{distance of second nearest neighbour}}$$

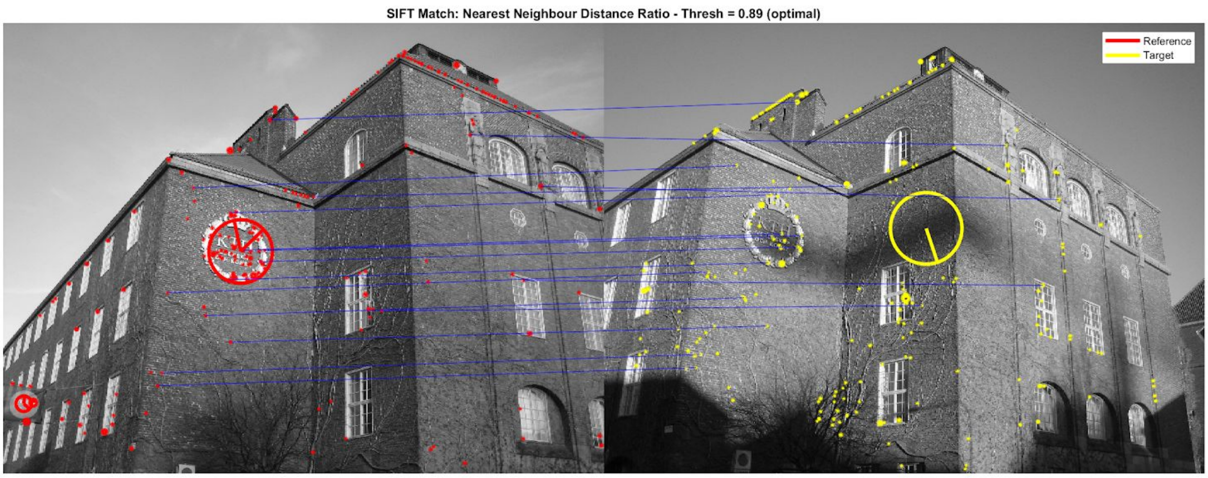

Fig. 10 - SIFT Match Nearest Neighbour Distance Ratio (optimal case)

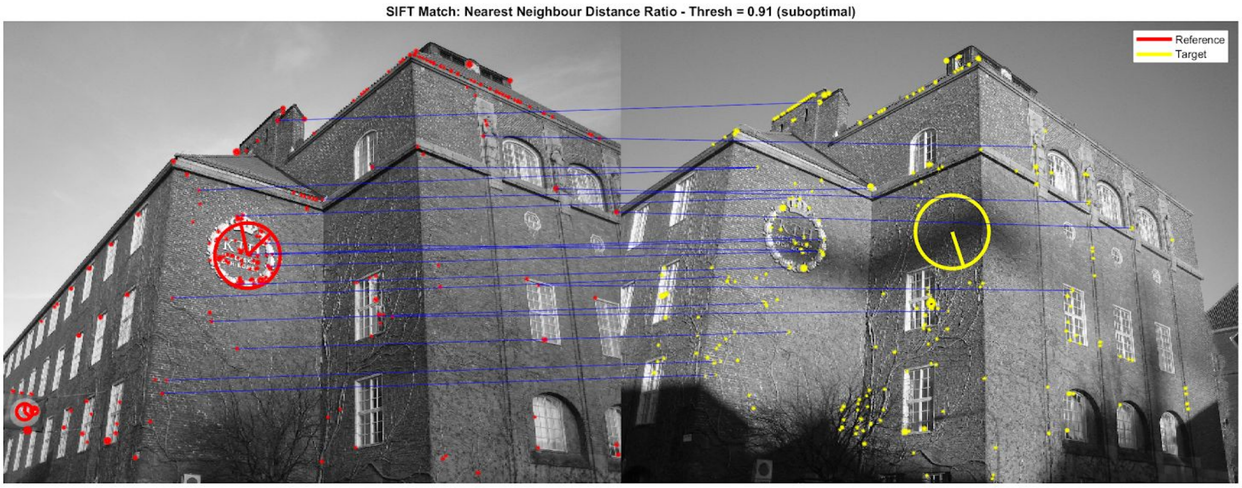

Fig. 11 - SIFT Match Nearest Neighbour Distance Ratio (suboptimal case)

Even though the optimal threshold is different (slightly higher) from the usual one found in literature (0.8), the obtained result is satisfying because of KTH logo match and it is also better performance than nearest neighbour, as it should be.

SURF Matching - Nearest Neighbour Distance Ratio (e)
For SURF only Nearest Neighbour Distance ratio matching algorithm in 64-descriptor space is implemented with an optimal threshold of 0.85 (Fig. 12) and a suboptimal of 0.87 (Fig. 13), after a trial-and-error method. The results are better than the SIFT in terms of speed and number of correct matches (especially KTH logo). In addition, some of the errors are made on windows corners, that even if they are different, will have similar descriptors. Results are shown below:

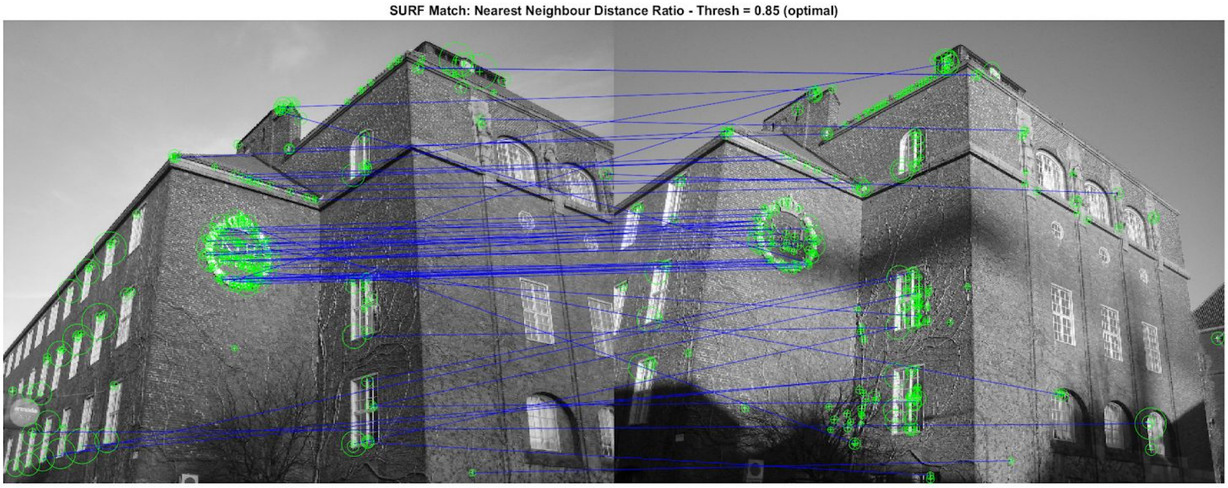

Fig. 12 - SURF Match Nearest Neighbour Distance Ratio (optimal case)

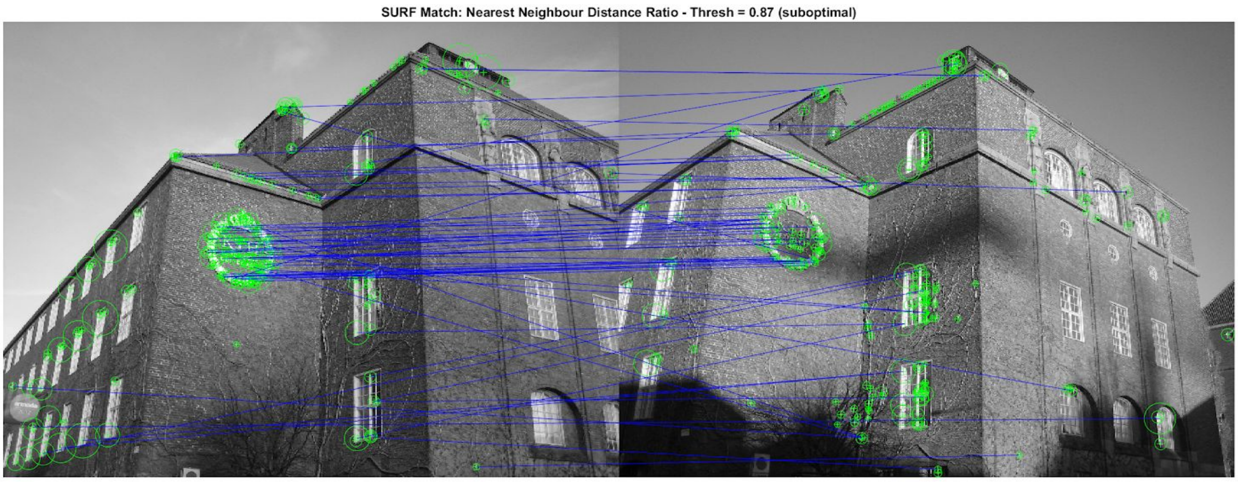

Fig. 13 - SURF Match Nearest Neighbour Distance Ratio (suboptimal case)

Conclusions

In conclusion, for keypoint detection SIFT performs better in scaling on the other hand SURF is better in case of rotation of target images with faster speed. For keypoints matching, out of three SIFT keypoint matching strategies nearest neighbour ratio finds more correct matches followed by nearest neighbour and fixed threshold. While in general if we compare SIFT and SURF nearest neighbour ratios, SURF outperforms SIFT with higher number of accurate matches. Possible improvements for matching can be done, for example adjusting the peak and edge thresholds to obtain more keypoints. Also a different distance measurement metric [7] can be used to search for nearest neighbour in the descriptor space: it is said [8] that using Chi Square distance or Hellinger distance may improve results. We tried with the first one, but because of time constraints we did not find good results.

References

[1] M. Fleri, Analysis and Search of Visual Data [EQ2425], Course Slides, KTH, 2019.

[2] D. Lowe, “Distinctive image features from scale-invariant keypoints,” Int. J. Comput. Vis., vol. 60, no. 2, pp. 91–110, 2004.

[3] H. Bat, T. Tuytelaars, L. Van Gool, SURF: Speeded Up Robust Features, European Conference on Computer Vision, 2006.

[4] D. Mistry, A. Banerjee, Comparison of Feature Detection and Matching Approaches: SIFT and SURF, GRD Journals, 2017.

[5] http://www.vlfeat.org/index.html

[6] https://it.mathworks.com/help/vision/ref/detectsurffeatures.html

[7] https://it.mathworks.com/help/stats/classification-using-nearest-neighbors.html#btap7nm.

[8] https://stackoverflow.com/questions/4357352/euclidean-distance-in-sift