Lab 6 - Image matching

Aurora A. Pedersen
INP9087774 - Computer Vision

May 8, 2023

Task 1

i)

In OpenCV, the features available are Harris corners, Shi-Tomasi corners, SIFT, SURF, FAST, BRIEF and ORB. Since it is preferable to have scale invariant features since the features in the images appear at different scales, I chose to stick with the SIFT-based and SIFT-like features. I found it interesting to test SIFT against its sped-up alternative SURF, and the un-patented version, ORB, to see how they all compare.

ii)

To match the features, I chose to use the Brute Force matching from OpenCV with kNN matching. I figured that the BF approach would give better results than FLANN since FLANN only provides an approximate of the nearest neighbors and thus will not be the most accurate option. Since there are no given constraints in terms of computation time, Brute Force seemed like the better option.

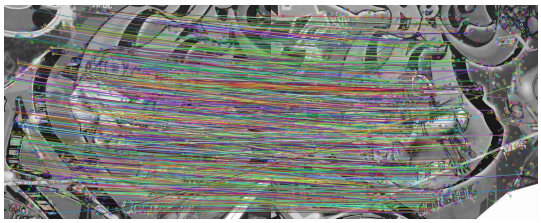
To check if an image has been subject to a strong transformation, I though that finding the affine transformation between the matched features would provide a good indication of the fact. Searching through the OpenCV documentation revealed that findHomography would be the appropriate approach to the problem, which conseptually is quite similar since it gives the perspective transformation between the matched feature vectors. How to use this homography matrix was another challenge during the lab work. First, I though that looking at the magnitude of the matrix eigenvalues would be a good idea. A simpler approach came to mind though, which was to look at the value of the determinant, where a value close to 0 (a singular value) would indicate that a string transformation has been applied.

One of the bigger challenges was to set an appropriate threshold for the Lowe's ratio test to discard false matches, and to set an appropriate threshold for the required amount of matches to conclude that two images are similar. In figure 3, we can see the same motive, but quite strongly transformed. With the tuning of 0.8 for the Lowe threshold, and 25 as the number of matches needed to conclude that the objects are similar, the results on this image were quite good, as it was reported by the program that the images are similar, but strongly transformed. The drawback to this tuning is the result seen in figure 2, where the images are not similar, yet there are a lot of matches, indicating that we do not reject enough matches. By tuning the parameters in a more conservative way, such as letting less matches though the Lowe's ratio test, there will be less false positives, but consequently also more false negatives. This is depicted in figure 5, where the dissimilar images in 5b give almost no matches, and the same happens in the strongly transformed image in 5a. This highlights the fact that there is a trade-off in terms of accuracy, and one will need to define in advance if false positives or false negatives are the least desirable, and tune the parameters based on that.

Yet, we can see that with the more *easy* images, such as in figures 1 and 4, there are few mismatched keypoints, and the program also reports that the images are similar without a strong transform.

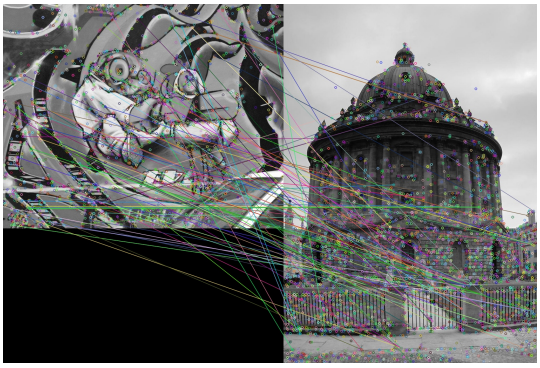
(a) SIFT matching result

(b) SURF matching result

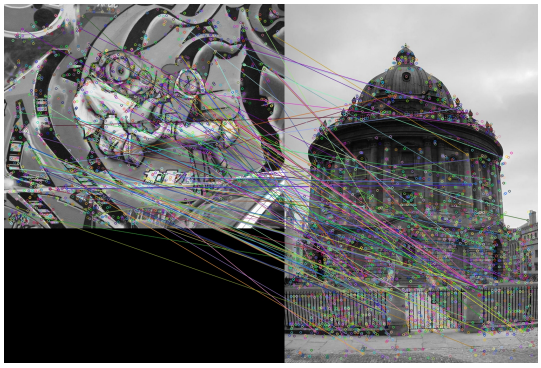


(c) ORB matching result

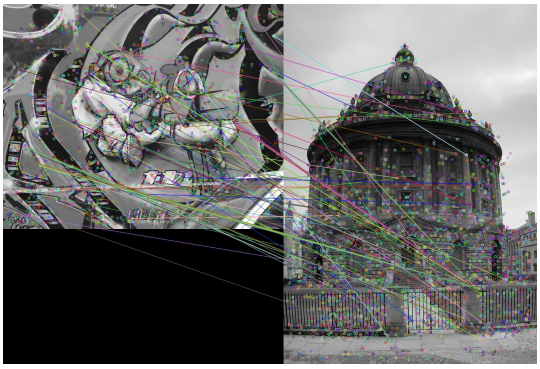
Figure 1: Matching between img1.png and img3.png.



(a) SIFT matching result



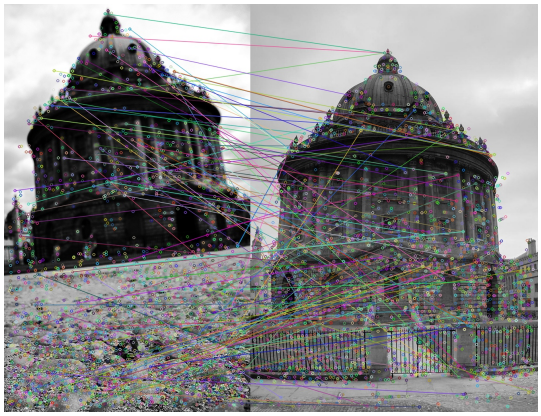
(b) SURF matching result



(c) ORB matching result

Figure 2: Matching between img1.png and bodleian_000167.jpg.

(a) SIFT matching result

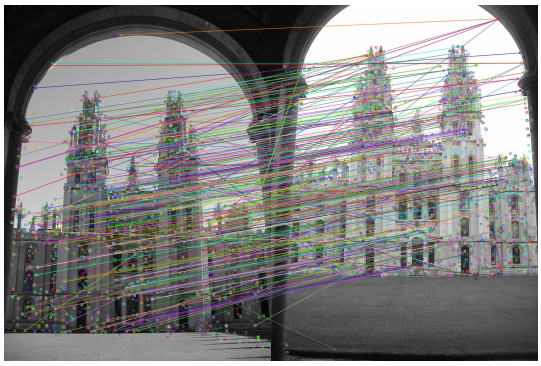


(b) SURF matching result

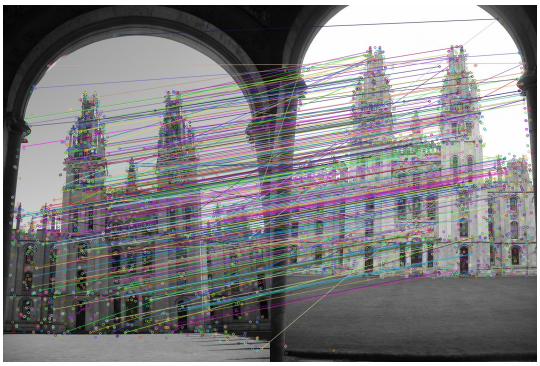
(c) ORB matching result

Figure 3: Matching between bodleian_000192.jpg and bodleian_000167.jpg.

(a) SIFT matching result



(b) SURF matching result



(c) ORB matching result

Figure 4: Matching between all_souls_000002.jpg and all_souls_000006.jpg.

(a) SIFT matching result.

(b) SIFT matching result.

Figure 5: Matching with Lowe ratio of 0.6.