UNIVERSITY OF COPENHAGEN

COMPUTER SCIENCE

ADVANCED TOPICS IN IMAGE ANALYSIS

Assignment 3

Author:
Casper BRESDAHL

Teacher:
Sune DARKNER

November 4, 2021

1 Assignment 3

1.1 Design of experiments

For this assignment a subset of the 'HPatches' data set has been chosen, namely the first 10 series with illumination changes and the 10 fist series with view point changes. Three image descriptors, namely SIFT, ORB and SURF will then be evaluated on the aforementioned data. A consideration for the experiments have been, that an image descriptor both chooses a location in an image which it finds fitting, and it describes this location. That is, we both get key points and descriptors for each found feature in the image. To evaluate the key points we use *mean localization error*, to evaluate descriptors we use *nearest neighbour mean average precision*, and to evaluate the image descriptor as a whole, we use *homography estimation*. Before going into details with these error measures, we will first look at how we classify true positives, false positives and false negative detections.

CHANGE

For each pair of images considered, we apply the image descriptor to both images, and the detections made in *image 1* are considered ground truth whereas the detections made in *image 2* are considered candidate points. If there is a perspective change between the two images, the ground truths are warped into the perspective of the candidate points. To detect false positives, we go through each detection in our candidate points, and each candidate point which does not have a similar ground truth is considered a false positive. To detect false negatives we go through each ground truth and if it does not have a similar candidate point it is a false negative. A true positive can be measured as either each candidate point which has a similar ground truth, or each ground truth which has a similar candidate point. This 'similar' / ϵ term is measured as the L2 norm, and can be increased / shrunk in order to create precision / recall curves. We will compute precision and recall by starting with $\epsilon = 0$ and increase it. As we might have more ground truths than candidate points and vice versa, the recall and precision is rescaled to always go from 0 to 1. To ensure correct curves, 1 - recall will be used to compute precision / recall curves. An example of a precision / recall curve can be seen in Figure 1. Each ground truth or candidate point can only be matched once, that is, if we have two candidate points around a ground truth, only one of them will be considered a true positive whereas the other will be considered a false positive.

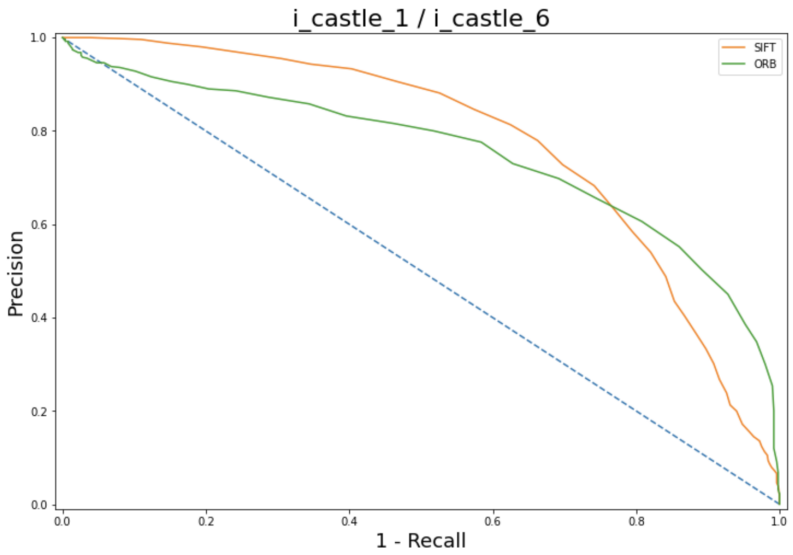


Figure 1: Precision / recall curve for two images in the castle series.

1.2 Error measures

We can now discuss the definition of each error measure.

Mean localization error is defined in accordance to [1] where we only use true positive detections and find the average distance between their nearest neighbours. This measure tells us about how close the key points are to each other in the two images, that is, if the image descriptor finds the same points in the two images. The error ranges between 0 and ϵ, where ϵ is the maximum distance between the two key points there can be for them to be considered true positives. As it is unclear which epsilon is used in [1], I have chosen to report the mean localization error for a range of epsilon values. To avoid rewarding image descriptors which find 0 features, the maximum error (ϵ) is used for a run where 0 features are found.

Nearest neighbour mean average precision is defined in accordance to [1], and takes all descriptors of an image pair and finds their nearest neighbour. Then creates a precision / recall curve and measures the area under the curve for this image pair. Then the average area under the curve is estimated across all image pairs. This measure estimates how well the image descriptor detects true positives without introducing false positives. The measure ranges between 0 and 1. As we are comparing the same scene, the descriptions made should describe the same image patches, and thus we do not perform any perspective warping for this measure. As it is not clear from [1] which range of ϵ values are used, I have chosen to use 100 evenly spaced ϵ values between 100 - 800 as the distance between all found descriptors in the manually inspected images seem to be matched in this range. To make this measure computational feasible, only the first 1000 descriptors found are used. These are likely in the same part of the image, but sampling 1000 descriptors across the image might result in a lot of descriptors which are not measured at the same key points.

Homography estimation is defined in accordance to [1] and takes the four corners of *image 1* and first measures the average distance between each corresponding corner transformed by the true homography matrix and the estimated homography matrix found by taking nearest neighbour matches found by the image descriptor. If the average distance is less than some ϵ then it is reported as a correct homography estimation, otherwise it is reported as a wrong estimation. The average number of correct estimations over all image pairs is then reported as the *homography estimation* measure. This measure evaluates the image descriptor as a whole by taking points matched in the two images to estimate a homography to warp one image into the perspective of the other. The result ranges between 0 and 1. As SIFT returns *a lot* of features, a Lowe's ratio at 0.7 is used to filter the number of features used to estimate the homography.

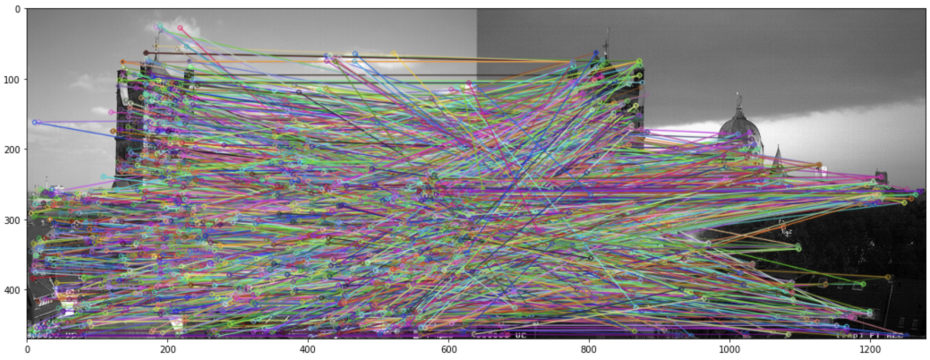
1.3 Results

To get an idea of how many features each image descriptor produces, we will begin this section by taking a look at three image pairs where each image descriptor have been run. The found features can be seen in Figure 2, and have not been constrained in any way, and there are both true positives along false positives.

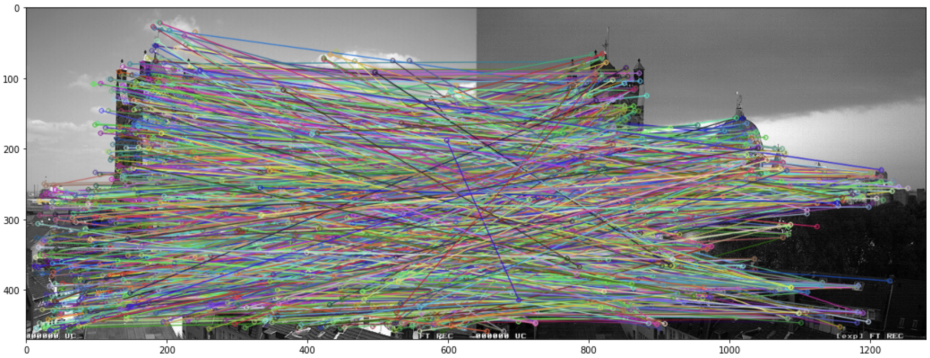
	Homography estimation			Mean localization error			NN mean average precision
	eps =1	eps = 3	eps = 5	eps = 1	eps = 3	eps = 5	
SIFT	0.34	0.61	0.71	0.56	1.46	2.41	0.72
ORB	0.05	0.21	0.31	0.54	1.54	2.41	0.73
SURF	0.16	0.38	0.47	0.56	1.48	2.27	0.50

Table 1: Results of the experiments.

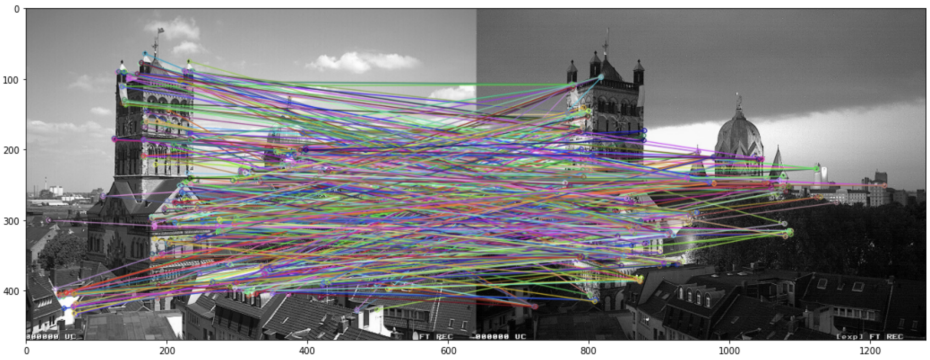
As seen, SIFT finds about as many features as SURF whereas ORB finds a considerable amount less.



(a) The detected SIFT features.



(b) The detected SURF features.



(c) The detected ORB features.

Figure 2: Each image descriptor run on the same image pair to give an indication of how many features each finds.

In Table 1 we see the results of the experiments conducted. When comparing SIFT and ORB we see their ability to estimate correct homographies varies greatly. ORB estimates as many correctly homographies when epsilon is 5 as SIFT does when epsilon is 1, that is, SIFT is much more accurate when estimating homographies. SURF lies in between the two, performing closer to ORB than SIFT. However, when we compare the mean localization error SIFT, SURF and ORB obtain similar errors. This means their ability to choose the same places in the image pairs are equivalent. However, looking at the actual errors, we see the error is about halfway between the minimum and maximum possible which means we are fairly far away from the actual location we were looking for. When it comes to nearest neighbour mean average precision SIFT and ORB performs similar and fairly well, whereas SURF performs quite poorly.

1.4 Discussion of results

A reason for the big difference in homography estimation performance between SIFT and ORB might be how many feature points each find and are able to match. SIFT does find a considerable amount more features than ORB does, and because SIFT have more points, and point matches, it is able to estimate the homography better. RANSAC is used to eliminate outliers, which means even though SIFT might find more false positive matches, some are eliminated. It is likely SURF performs better than ORB for the same reason, however, because SURF produces worse descriptors (seen from nearest neighbour mean average precision) than SIFT, it is worse at estimating the homographies than SIFT.

The fact the mean localization error seems to be fairly high for all image descriptors might be due some aliasing or other approximation of the new location after moving the ground truth point locations into the coordinate system of the candidate points. If the new point location is off by 1 or 2 pixels it will make a quite big error in the mean localization error.

We see ORB and SIFT performs fairly well on nearest neighbour mean average precision, which is likely because they both are scale and rotation invariant, however, SURF is not rotation invariant, and this might be the reason it performs a lot worse.

1.5 Conclusion

In conclusion we have seen SIFT and SURF detects roughly the same amount of features where ORB detects considerable less features. All three image descriptors perform somewhat bad when it comes to mean localization error, but this is likely due to aliasing or approximation error when transforming the images. When it comes to nearest neighbour mean average precision we see SIFT and ORB performs similar and fairly well, however SURF performs quite bad. This might be due to SIFT and ORB being both scale and rotation invariant, whereas SURF is only scale invariant. Lastly we have seen SIFT performs a lot better than both SURF and ORB when it comes to homography estimations, which is likely because SIFT both detects a lot of features and describes them well.

If we are to determine which of the three is better, it would depend on the task at hand. Although it is clear SIFT performs the best in the these experiments, it is also the slowest of the three. If the task requires fast feature matching it would not be possible to use SIFT, and we would need to trade performance for speed.

2 References

[1] DeTone, D., et. al. 'SuperPoint Self-Supervised Interest Point Detection and Description'. Unknown publisher. Apr 2018.