EQ2425 Analysis and Search of Visual Data
Image Features and Matching

Project 1

Aleix Espuña Fontcuberta
aleixef@kth.se

Ioannis Athanasiadis
iath@kth.se

September 17, 2011

Summary

In this project we considered the problem of analysing the robustness of two feature extraction algorithms (SIFT and SURF). We analyzed their robustness, with respect to rotation and scale, by extracting keypoints on transformed versions of the same input image. We concluded that the SIFT algorithm is the most robust one, although it is also the slowest one. Finally, we experimented with three different feature matching algorithms: fixed threshold, distance ratio and nearest neighbor. We obtained the best performance with the distance ratio algorithm, with a threshold set to 0.7.

1 Introduction

In image processing, there is particular interest in algorithms that are capable of extracting salient points from images, termed as keypoints, in a robust manner. Robustness, in the context of salient point extraction, refers to identifying identical keypoints under various transformations. In this project, we experimented with the SIFT [1] and SURF [2] algorithms which are known to be robust under rotation and scale transformations. Additionally, these two algorithms are capable of generating feature representations for each of the detected keypoints. The extracted feature representations allow for keypoints matching across different frames depicting the same scene. In the context of this project, we experimented with three feature matching algorithms named fixed threshold, distance ratio and nearest neighbor (NN).

2 Problem Description

2.1 SIFT and SURF Robustness

Both SIFT and SURF algorithms are given an image and return a set of pixel coordinates (keypoints). In order to analyze their robustness under image transformations, we will extract keypoints from an original image and from a transformation of it. Then we will predict the ideal keypoint locations on the modified image. By comparing the predicted keypoints with the keypoints on the transformed image we get a measure of the algorithm robustness. The measure is realized by the repeatability metric as explained in the Project's description.

Moreover, we also considered the problem of feature matching across different views. The process of feature matching is carried out by detecting keypoints and thereafter extracting their representation in the form of $1 \times D$ vector. The term D refers to the length of the feature representation and is 128 bins long for the SIFT algorithm while in the case of SURF is 64. Generally, larger descriptors provide richer representations and consequently better feature matching capabilities in the expense of speed. The extracted representations are called descriptors and they are indicative of the feature they describe, these descriptors aim at representing the local neighborhood of each feature in a way that allows matching features even in the presence of different capture condition e.g. scale and rotation. Admittedly, matching features across different frames can get quite challenging due to occlusions, repetitive patterns and wide baselines etc. In the context of this project, we employed different matching algorithms to match features from the query image obj1_5.JPG to the database image obj1_t1.JPG. Due to not having been provided with the intrinsic parameters of the camera neither the extrinsic parameters corresponding to these two frames, we are restricted to evaluating the matching performance visually.

3 Results

3.1 Robustness of keypoint detector

In Figure 1 we display an example of the keypoints extracted by SIFT and SURF. Considering float images in [0, 1] interval, for the SIFT algorithm we used an edge threshold of 5 and a peak threshold of 0.047. For the SURF algorithm we set the metric threshold to 6000. The results in Figure 1 indicate that most of the keypoints are located in edges, straight lines and also in corners (which are stable edge points in SIFT). Additionally, it is evident that the KTH logotype is quite salient and thus many of the detected keypoints refer to that specific region.

When it comes to the robustness under image rotations, Figure 2 depicts the behavior of the two algorithms. We computed the repeatability for each rotation of the original image. The rotation angles used, range from 0 to 360 degrees with steps of 15. The results show that SURF produces the highest peaks but also the smallest minimas (below 0.7). On the other side, SIFT shows a robust performance where all values are above 0.8. Additionally, we also observed that in both algorithms, the repeatability minimized for degrees that are the furthest apart from being multiples of 90 degrees. We believe this is the case due to bilinear interpolation introducing more evident artifacts for rotations that are further apart from being flipped version of the initial image. What happens to the algorithm's robustness if we change the resolution (scale) of the image instead? We applied 9 different scale changes to the original image, with scale factors 1.2^n from $n = 0$ to $n = 8$. Figure 3 shows that the SIFT curve is above the SURF curve for all the scale factors applied. Based on both these figures, we concluded that the SIFT keypoint detection algorithm is more robust compared to the SURF one with respect to scale and rotation.

(a) (b)

Figure 1: Keypoints on the "1 5.JPG" image as detected by the SIFT and the SURF algorithms. (a) SIFT PeakThresh: 0.047 and EdgeThresh: 5 (b) SURF MetricThreshold: 6000.

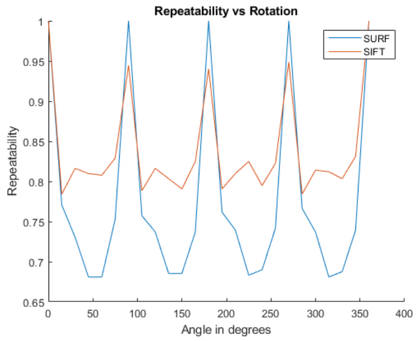


Figure 2: Comparison between the SIFT repeatability and the SURF repeatability when the original image suffers rotations from 0 to 360 degrees in steps of 15.

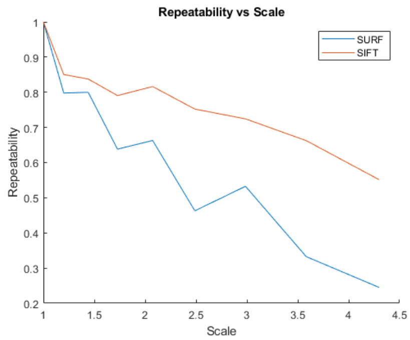


Figure 3: Comparison between the SIFT repeatability and the SURF repeatability when the original image suffers scale changes.

3.2 Image Feature Matching

3.2.1 SIFT keypoint extraction

In Figure 4, we plot the keypoints detected by the SIFT algorithm with the same thresholds used in the previous section. The setup described above, results in 468 keypoints for the query and 394 for the database images.

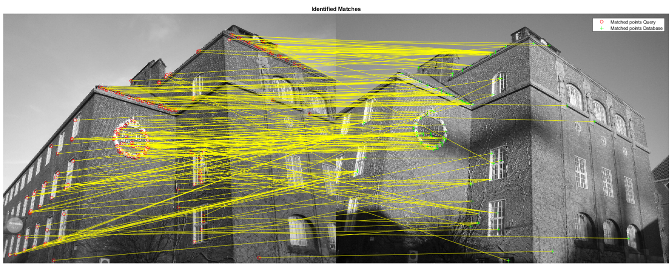
(a) (b)

Figure 4: SIFT keypoints detection (a) Query Image (b) Database Image

3.2.2 SIFT keypoint matching-Fixed Threshold

For this task we consider two feature points to match if the Euclidean distance between their descriptors is lower than a pre-defined threshold. Based on the qualitative results provided in Figure 5, we identify the threshold of 0.32 to be the ideal one. More specifically, the threshold of 0.5 seems to be producing way too many incorrect matches, for instance, it matches feature points corresponding to the front roof of the building to features corresponding to the side one. Finally, we observed that matching features corresponding to windows is challenging for all thresholds, due to the repetitive patterns.

(a)



(b)

Figure 5: Fix-threshold matching (a) Threshold : 0.32 (c) Threshold : 0.5

3.2.3 SIFT keypoint matching-NN matching

In this task, we simply matched each feature of the query image to its closest one of the database image. The feature matching output is provided in the figure bellow. Based on that, we can see that the Nearest Neighbor (NN) matching forces unrelated features to be matched just because it happened to be their closest one. Build upon this, we consider this approach to not be quite effective.

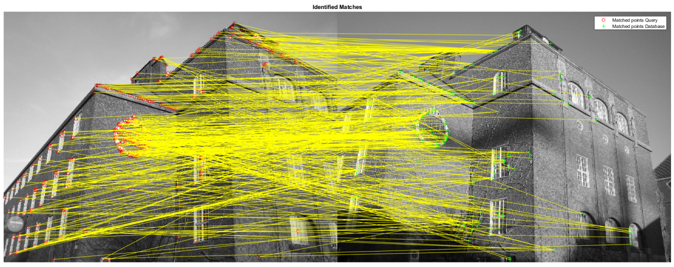
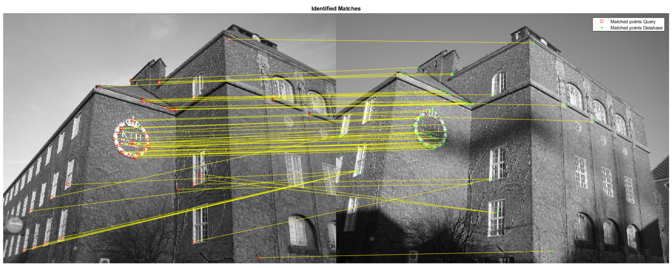


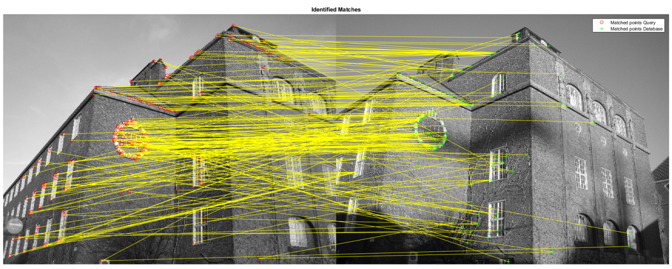
Figure 6: NN matching

3.2.4 SIFT keypoint matching-NN distance ratio matching

In here, we enhance the naive NN matching approach by also considering the 2nd closest match. More specifically for each query feature, we construct a ratio-based threshold through dividing the distance to its closest database feature by the distance to its second closest one. Based on that, candidate matches having a ratio close to 1 are considered ambiguous and thus are not considered to be matches. On the other hand the closer the distance ratio is to 0, the more confident we are about the match. In Figure 7 we provide the feature matching output for different distance-ratio thresholds. Based on that, we consider the distance-ratio threshold of 0.7 to be the best. On the other had, we identified the threshold of 0.95 to be suboptimal. Naturally, setting the ratio-distance theshold to 1 results in the NN matching algorithm of the previous section.



(a)

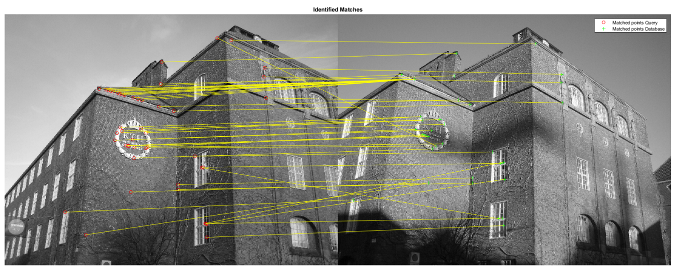


(b)

Figure 7: Distance-ratio matching (a) Threshold : 0.7 (b) Threshold : 0.95

3.2.5 SURF keypoint matching-NN distance ratio matching

In the context of this task, we apply the NN distance-ratio matching approach on the keypoints as detected and described by the SURF algorithm. More specifically, we used a MetricThreshold of 6000, which results in 445 feature points being detected on the query image and 491 on the database one. Finally, we report the feature matching performance of the SURF algorithm for distance ratio of 0.7 in Figure 8. Based on this comparison, we consider the SIFT algorithms to produce slightly better matches.



(a)

Figure 8: Distance-ratio matching (a) SURF Threshold : 0.7

4 Conclusions

Summing up, Figures 2 and 3 clearly indicate that SIFT is more robust than SURF. However, we noticed while running the experiments that indeed SURF is much faster than SIFT. Finally, based on the figures provided in section 3.2, we conclude that the distance ratio threshold performs the best when it comes to correctly identifying matching features across different frames. On the other hand, we demonstrated that the naive NN feature matching approach performs the worst. The lacking performance of NN is mainly due to matching unrelated features just because their feature representation happened to be closer compared to all the others.

Appendix

Who Did What

Both of us coded separately the results for section 3.1. We followed this approach in order to ensure that our repeatability was indeed calculated correctly. The part of the report corresponding to keypoints robustness, was mostly written by Aleix. When it comes to feature matching implementation as well as the part of the report corresponding to that, it was done by Ioannis. Finally, we together discussed the results and wrote the conclusion section.

References

[1] Lowe, David G, *Object recognition from local scale-invariant features. Proceedings of the seventh IEEE international conference on computer vision. Vol. 2. Ieee, 1999*

[2] Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool, *Surf: Speeded up robust features. European conference on computer vision. Springer, Berlin, Heidelberg, 2006.*