EQ2425 Analysis and Search of Visual Data
Project 1 Report: Image Features and Matching

Jingxuan Mao jmao@kth.se
Yuqi Zheng yuqizh@kth.se
Aarati Medehal medehal@kth.se

September 18, 2022

Summary

The following Project focuses on:

- Extracting key-point detectors and descriptors from images.
- Evaluating the robustness of key-point detectors and descriptors by measuring the 'repeatability'.
- Implementing Feature Matching Algorithms, namely *Fixed Threshold Matching Algorithm*, *Nearest Neighbour Matching Algorithm*, *Nearest Neighbour Distance Ratio Matching Algorithm*

Note that the following project has been implemented on Python 3.7 using the OpenCV library to perform feature extraction and related operations on the provided images.

1 Image Features

The two feature detection algorithm that have been used are the SIFT(Scale Invariant Feature Transform) and the SURF(Speeded up Robust Features).

SIFT has 4 basic steps:

1. Scale Space is constructed to ensure that the features are scale-independent using Difference of Gaussian(DoG).
2. Key point Localisation is performed by identifying the suitable features (removing the low contrast keypoints).
3. Keypoints are ensured to be invariant to rotation by creating a histogram for magnitude and orientation.
4. A unique descriptor is assigned to each keypoint which enables feature matching.

SURF too works in a very similar fashion:

1. Scale Space is constructed using box filters to approximate the DoG.
2. To perform keypoint localisation, a Hessian matrix based BLOB detector is used.
3. Orientation assignment is performed using wavelet responses.

4. In order to get the feature descriptor, the wavelet responses are used for representation of each subregion in the image.

Repeatability is a term used to represent the ability to detect the same features after going through some sort of change in scale, alignment, transformation, viewpoint etc.

The working of the algorithms to extract the features, and their robustness have been depicted and discussed thoroughly in the following subsection.

1.1 Results and Discussions

1.1.1 a.The first question required us to apply both the SIFT and SURF key-point detectors on the provided image and display the key-points superimposed on the original image after adjusting the peak and edge threshold of the SIFT detector, and the strongest feature threshold on the SURF detector.

(a) SIFT keypoints superimposed after adjusting peak and edge thresholds

(b) SURF keypoints superimposed after adjusting the strongest feature threshold

Figure 1: SIFT and SURF keypoints after respective threshold adjustments

In the SIFT algorithm, the key-points are refined by eliminating those that are found to be of low contrast or at an edge, making them unstable. The **Peak Threshold** is defined as the minimum amount of contrast which would make it accept a key-point. The **Edge Threshold** is the edge rejection threshold, i.e, rejects peaks of the scale space where the curvature is too small.

The values of the threshold have been identified by using visual perception aided by trial-and-error. (Figure 1(a))

For the given image, the best set is (Peak,Edge) = (0.18,5).

The **strongest feature threshold** in SURF is a scalar(non-negative). The default value of this threshold is 1000, and by decreasing the value, we can return more blobs.(Figure 1(b))

For the given image, the (strongest_feature_threshold) = (800).

On observing the detected results, we feel that more number of keypoints are detected at the region where the text *vetenskap och konst* is written.

1.1.2 b. Next, we are required to plot 'Repeatability versus Rotation Angle' in increments of 15 degrees.

The following observations can be made (Figure 2):

- When a rectangular image is rotating, the computer will add zero-value pixels around it to make up a larger rectangular image which it can pro-

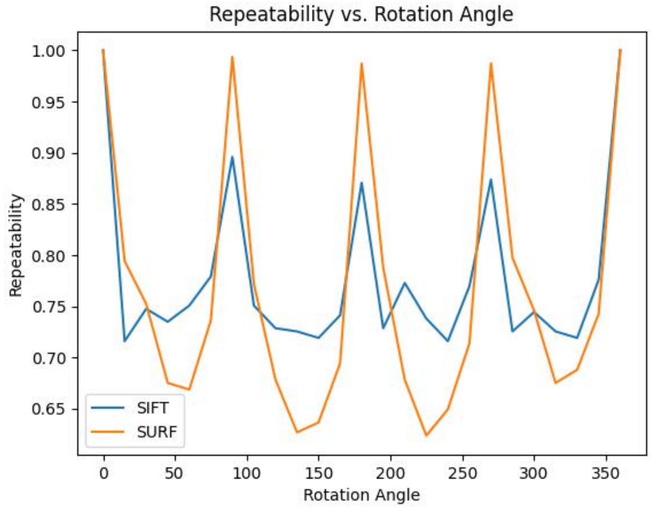


Figure 2: Plot of Repeatability versus Rotation Angle

cess. The image will not be modified by the computer for every 90 degrees of rotation. Therefore, the repeatability peaks occur every 90 degrees.

- In SURF: We notice that the repeatability is quite excellent (≈ 1), at every 90 degrees interval, but there is a steep decline as it changes angle. This shows that SURF is not very robust to rotation.
- In SIFT: We notice that though the repeatability is not as great at SURF at every 90 degrees, it is still significantly high. Additionally, there is no steep decline as observed earlier, making **SIFT more robust to rotation**(i.e, rotation-invariant) and thus producing better repeatability.

1.1.3 c. In the final part, Plot Repeatability versus Scaling Factors, and comment on the robustness

The function *cv.resize* has been used to modify the scale of the image, with scaling factors in powers of m(=1.2)
From the plot (Figure 3),

- In SIFT: We observe that there is a much higher repeatability that is obtained. It is worthy to notice that the repeatability decreases more slowly and almost remained as the scaling factor increases, hence making it more robust.
- In SURF, there is a clear lower repeatability as the scaling factor increases, as compared to SIFT. So it can be concluded that SURF actually performs worse than SIFT both in the scenarios of rotation and scaling. The best merit of it is fast and effective considering computation.

2 Image Feature Matching

Feature Matching is simply the process where the computed descriptors are used to find the respective features between two similar looking images and enable matching between the two images.

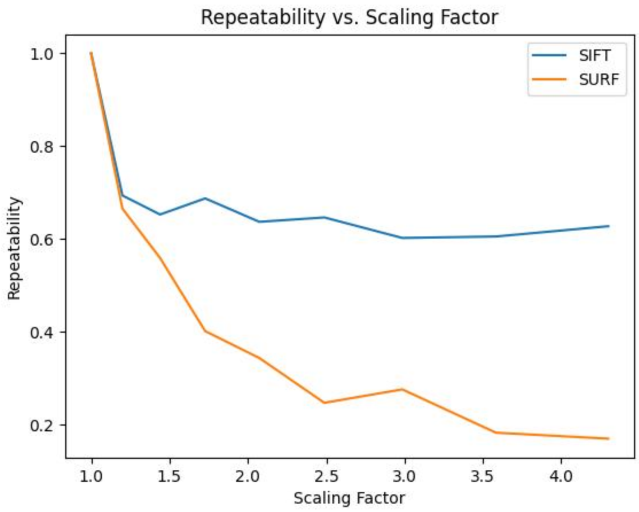


Figure 3: Plot of Repeatability versus Scaling Factor

In this section, we first detected the SIFT keypoints of the database image and the query image respectively. Then, we implemented the fixed threshold, nearest neighbor and nearest neighbor distance ratio matching algorithms to match the features of the two images. Finally, we utilized the nearest neighbor distance ratio algorithm to match the SURF features of the two images.

2.1 Results and Discussions

2.1.1 Superimposing SIFT feature keypoints

The query image and the database image look similar. We extracted 317 features of the query image and 402 features of the database image with the same SIFT keypoint detector, as shown in Figure 4(a) and Figure 4(b).

(a) Query Image (b) Database Image

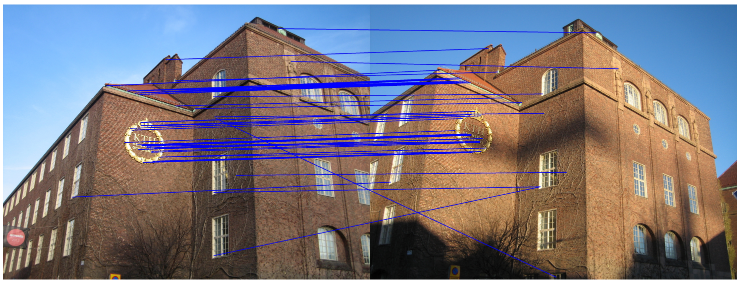
Figure 4: Detected SIFT Keypoints of Images

2.1.2 'Fixed Threshold' Matching Algorithm

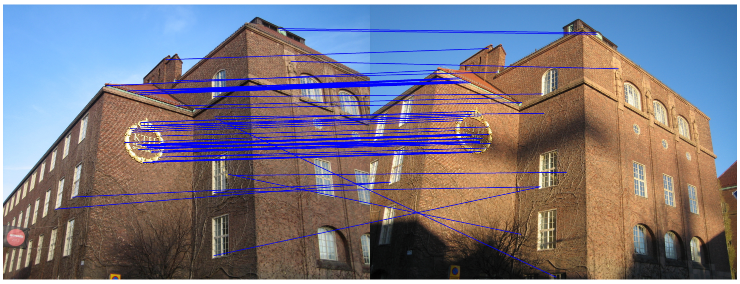
The 'Fixed Threshold Matching' Algorithm selects the matches that occur within the threshold, these matches do not have to occur one-one.

After different trials in adjusting the threshold, the best match occurred at the threshold of 200, while a suboptimal result occurred when threshold was

210, as shown in Figure 5(a) and Figure 5(b). When the threshold is small, some matches are not included. While as the threshold becomes larger, some non-matching keypoints are considered as true matches.



(a) Optimal Result: Threshold=200



(b) Suboptimal Result: Threshold=210

Figure 5: Fixed Threshold Matching Algorithm

2.1.3 'Nearest Neighbour' Matching Algorithm

It is also termed as the Brute Force/Violent Match Method. For each keypoint of the query image, the Nearest Neighbor Matching Algorithm calculate the distances between its descriptor and all descriptors of the database image. Then, these distances are sorted and the keypoints with the shortest descriptor distance will be selected as the match. In this case, we still need to make sure that the descriptor distance of the match we select is within the threshold. The distance metric is calculated using the Euclidean Distance NORM_L2. The threshold chosen for this algorithm after numerous iterations is 208.

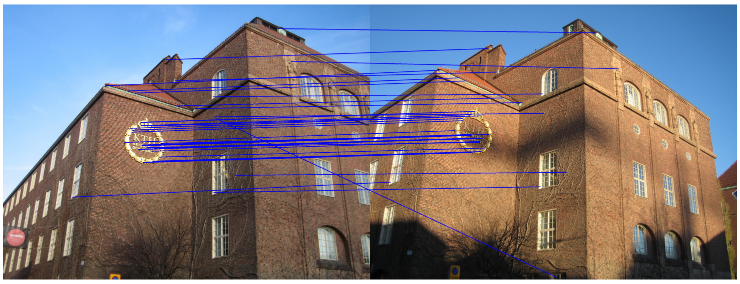


Figure 6: Nearest Neighbour Matching Algorithm

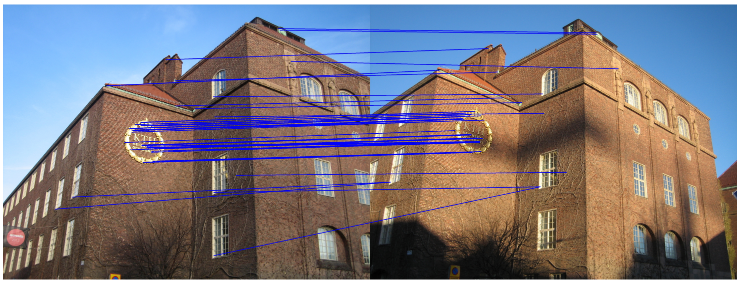
One disadvantage of this method is the occurrence that some keypoints might not ever get their correct match due to an unidentified keypoint in the reference image or background clutter. In order to solve this issue, the Nearest Neighbour Distance Ratio Algorithm is used.

2.1.4 'Nearest Neighbour Distance Ratio' Matching Algorithm: SIFT and SURF

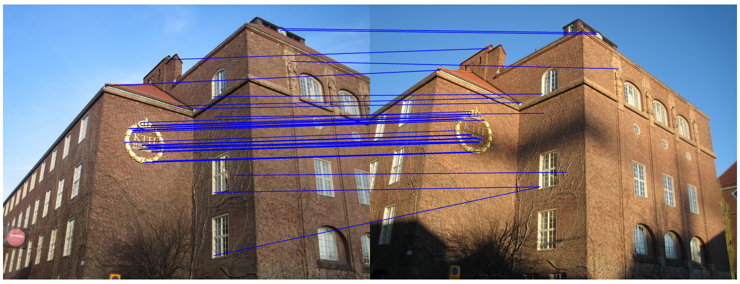
This next algorithm is also very similar to the previous Brute Force Method. Additionally, the distance of the nearest neighbour is compared to the second nearest neighbour distance. Therefore, matching occurs via distance ratio threshold. The *knnMatch* method in OpenCV is used by choosing the value of $k = 2$ to enable finding the second nearest neighbour.

By definition, $\frac{DistancetoNearestNeighbour}{DistancetoSecondNearestNeighbour} \leq 0.8$ results in good separation. Via this additional threshold measure, 90% of false matches are eliminated while losing less than 5% of correct matches.

On implementing the algorithm, the ratio threshold of 0.75 provided the best result and a threshold of 0.70 yielded a suboptimal matching, as shown in Figure 7(a) and Figure 7(b). And the results of NNDR matching with the SURF keypoints are shown in Figure 8(a) and Figure 8(b).

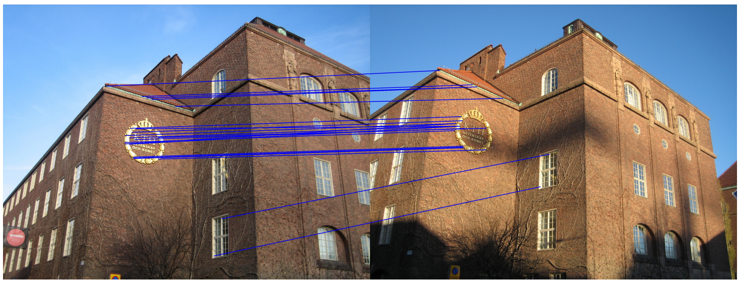


(a) Optimal Result: Threshold=0.75



(b) Suboptimal Result: Threshold=0.70

Figure 7: Nearest Neighbour Distance Ratio Matching Algorithm: SIFT



(a) Optimal Result: Threshold=0.72

(b) Suboptimal Result: Threshold=0.75

Figure 8: Nearest Neighbour Distance Ratio Matching Algorithm: SURF

We can see that the performance of this algorithm is relatively better compared to the previous algorithms. Both NNDR matching algorithms with SIFT and SURF features could yield good results, but in this case, SIFT provided relatively more and more accurate matches than SURF. This may be the result of the fact that the descriptor of each SIFT keypoint contains 128 elements while the descriptor of each SURF keypoint only contains 64 elements. Moreover, their specific algorithms such as constructing scale space and assigning orientation are all different, which are suitable for different scenarios. In addition, we noticed that the NNDR matching algorithm with SURF took less time than SIFT.

3 Conclusions

This project thus helps to provide a clear picture on how feature matching in images occurs, the different methods involved in helping to extract the keypoints and the algorithms that enable feature matching through the descriptors extracted. The project also facilitates to understand how the keypoints matching occurs when the image undergoes different transformations in size and rotations.

In general, both for rotation and scaling, the SIFT keypoint detector shows better robustness than the SURF detector. However, the SURF detector requires less time and computational resources. The Fixed Threshold, Nearest Neighbor, and Nearest Neighbor Distance Ratio matching algorithms can all achieve good keypoints matching. The NNDR matching algorithm with SIFT performs the best overall. In the future study, we need to select appropriate algorithms for different application scenarios according to their features.

Appendix

Who Did What

In the first Project, Aarati Medehal joined the group at a later stage due to administrative issues. Hence, was mainly involved in ensuring to understand and execute the code, as well in writing the report.

Jingxuan Mao and Yuqi Zheng were involved in the major part of the coding aspect in equal amounts along with helping to write the report.