Report Template for EQ2425 Analysis and Search of Visual Data

EQ2425, Project 1

Yuzhou Liu yuzhoul@kth.se

Yage Hao yage@kth.se

September 16, 2021

Summary

In this project, our first task is to apply SIFT and SURF keypoint detectors on image "obj1 5.JPG" and test the detectors' robustness against some simple transformation such as rotation and scaling.

Our second task is to do image feature matching between the database image "obj1 5.JPG" and the test one "obj1 t5.JPG" using three different algorithms: "fixed threshold" matching algorithm, "nearest neighbor" matching algorithm and "nearest neighbor distance ratio" matching algorithm.

1 Introduction

1.1

==========
==========
==========
==========
==========
==========
==========
==========

1.2 Image Feature Matching

The key to match a modified image to our database image is to compare the distance between keypoints in the descriptor space. Intuitively, we choose to compute **Euclidean distance** which in Euclidean plane is defined as:

$$d(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \tag{1}$$

where point x has coordinate (x_1, x_2) and point y has coordinate (y_1, y_2).

To compare images using the obtained distance, we applied three different strategies:

- **Fixed Threshold**: Set a threshold. For distance less than the threshold, we record the keypoint pair as matching.
- **Nearest Neighbor**: For each keypoint in the reference image, find the corresponding keypoint which has shortest distance in the target image and mark such pairs as matching points.

- **Nearest Neighbor Distance Ratio**: We set a threshold. And for each keypoint in the reference image, we compute the value of distance ratio as:

$$distance\_ratio = \frac{distance\_to\_nearest\_neighbor}{distance\_to\_second\_nearest\_neighbor} \tag{2}$$

  Then we filter and get the matching pairs whose distance ratio is less then the ratio threshold.

2 Problem Description

2.1 Robustness of Keypoint Detector

==========
==========
==========
==========
==========
==========
==========

2.2 Image Feature Matching

Apart from the original image's SIFT and SURF features, we need to extract the SIFT and SURF features of the test image. In MatLab, SURF features detection requires the Computer Vision Toolbox, which can be installed in the software. And SIFT features detection requires an open-source VLFeat library. To setup VLFeat for temporary use, run *setup.m* first.

Then, to extract SIFT features for both reference image and test image, run *q3_sift.m*. This will generate 128 dimensional keypoint descriptor, and intuitively we plot images with the keypoints marked.

Now, we can do matching. With background knowledge provided in part1.2, we can implement three matching algorithms based on the following pseudo code:

- **Pseudo code for Fixed Threshold Algorithm**

```
DEFINE threshold
FOR point x in reference image:
    FOR point y in test image:
        compute d(x,y)
        IF d(x,y) < threshold:
            match x,y
```

- **Pseudo code for Nearest Neighbor Algorithm**

```
FOR point x in reference image:
    For point y in test image:
        compute d(x,y)
    get minimum d(x,y)
    match x,y
```

- **Pseudo code for Nearest Neighbor Distance Ratio Algorithm**

```
DEFINE threshold
FOR point x in reference image:
    For point y in test image:
        compute d(x,y)
    get distance_ratio
    IF distance_ratio < threshold:
        match x,y
```

To get the matching results with the above three algorithms applied, run *q3_fixedthreshold.m*, *q3_nn.m* or *q3_nndr.m* respectively.

In addition, to get the matching result on SURF features with Nearest Neighbor Distance Ratio Matching, run *q3_surf_nndr.m*.

3 Results

3.1 Robustness of Keypoint Detector

==========
==========
==========
==========
==========
==========
==========
==========

3.2 Image Feature Matching

3.2.1 (a) SIFT Features

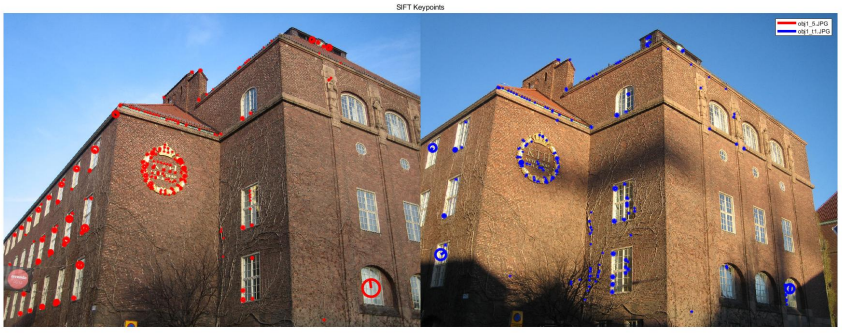

Figure 1: SIFT feature keypoints on reference image (left) and test image (right) with $peak\_threshold = 14$, $edge\_threshold = 7$

By adjusting the peak threshold and the edge threshold, we obtained 318 keypoints on reference image and 237 keypoints on test image as shown in Figure 1. For the following parts, we applied matching based on these SIFT feature keypoints.

3.2.2 (b) Fixed Threshold Matching

By adjusting the threshold, we noticed that as threshold increasing, the number of matching pairs increases. As we set threshold to 60 (Figure 2), 60 matches

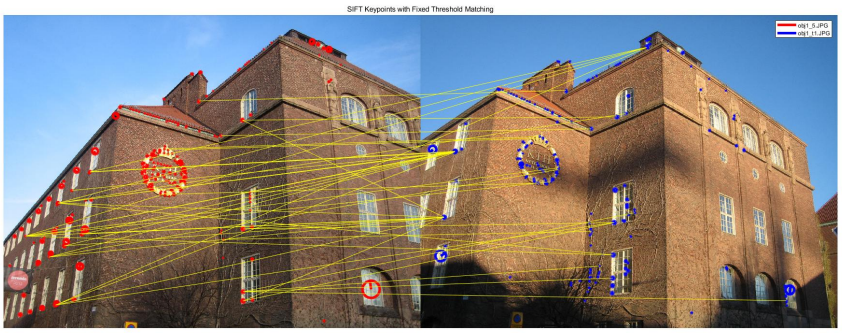

Figure 2: Fixed Threshold Matching on SIFT feature keypoints with $threshold = 60$

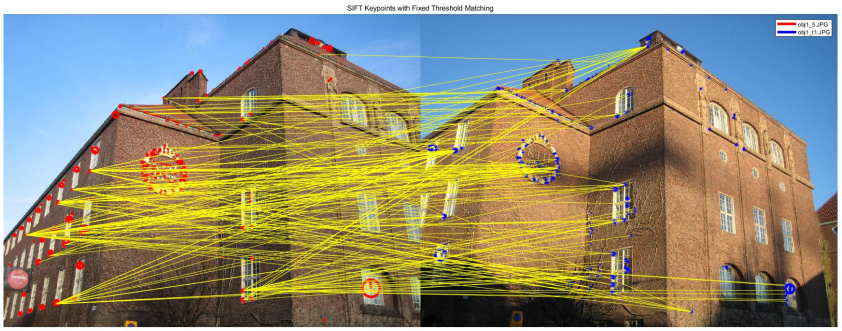

Figure 3: Fixed Threshold Matching on SIFT feature keypoints with $threshold = 65$

are formed, while there are 235 matches with threshold setting to 65 (Figure 3). The larger threshold will provide more matches, but probably include more wrong matches with one-to-many relationships.

3.2.3 (c) Nearest Neighbor Matching

Figure 4: Nearest Neighbor Matching on SIFT feature keypoints

As shown in Figure 4, every point of the reference's 318 keypoints is matched to its nearest keypoint on test image. For those points that appear in both images, the Nearest Neighbor matching provides a reliable result. However,

for those that only appear once, the Nearest Neighbor algorithm gives a wrong match.

3.2.4 (d) Nearest Neighbor Distance Ratio Matching

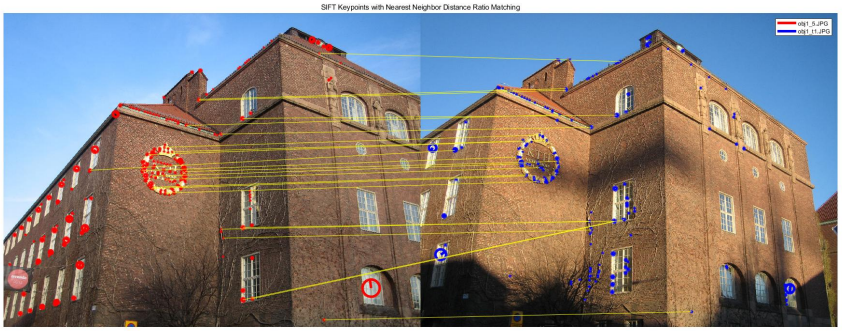

Figure 5: Nearest Neighbor Distance Ratio Matching on SIFT feature keypoints with $threshold = 0.8$

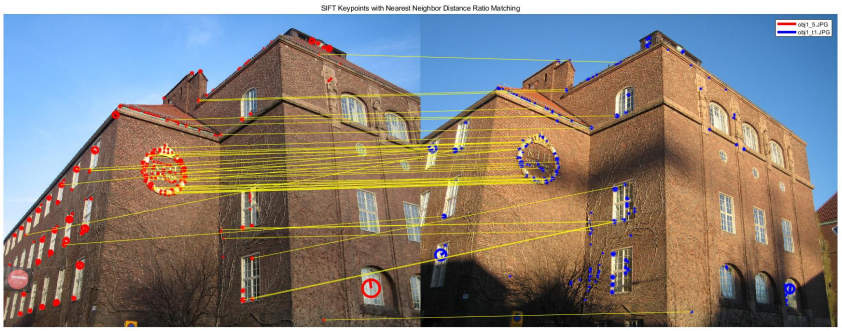

Figure 6: Nearest Neighbor Distance Ratio Matching on SIFT feature keypoints with $threshold = 0.85$

For the Nearest Neighbor Distance Ratio Matching, we first tried the empirical ratio threshold of 0.8 and only obtained 28 matched pairs (Figure 5), though there are only a few wrong matches. We then tried to increase the ratio threshold to 0.85, and the visualized result (Figure 6) is much satisfying since more feature keypoints, like the ones at the edge of the building and on the KTH logo, are paired.

3.2.5 (e) SURF Features with Nearest Neighbor Distance Ratio Matching

Similar to part 3.2.4, we applied Nearest Neighbor Distance Ratio Matching on SURF feature keypoints, set ratio threshold to 0.8 and obtained Figure 7. To make sure the comparison are on a similar level, we took the 250 strongest keypoints from SURF features, which is close to the number of SIFT feature keypoints in test image. Figure 7 shows 68 matches, which is higher than 28 matches in Figure 5. Through the visual examination, we noticed the Nearest Neighbor Distance Matching on SURF keypoints performed well both on correctness and efficiency. And we also find that the wrong matches are mostly happened on those keypoints that lie on the edge of windows.

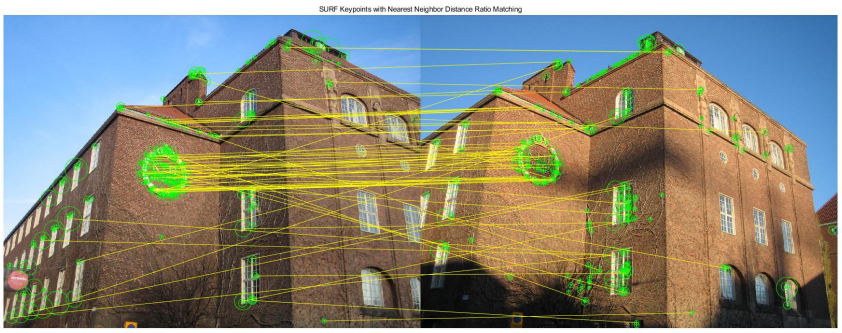

Figure 7: Nearest Neighbor Distance Ratio Matching on SURF feature keypoints with $threshold = 0.8$

4 Conclusions

==========
==========
==========
==========
==========
==========
==========
==========

Appendix

Who Did What

Yage Hao is responsible for the "Image Feature Matching". In detail, she implemented question 3's code and wrote the report's 'Summary', '1.2', '2.2' and '3.2' parts.

==========
==========
==========
==========
==========
==========
==========
==========

References

[1] Markus Flierl, *EQ2425 Analysis and Search of Visual Data*, Lecture Slides, 2021

[2] Wikipedia, *Euclidean distance*, Available: https://en.wikipedia.org/wiki/Euclidean_distance, Last edited: August 15, 2021