

COMPUTER VISION

Assignment 3

Keio University



1 Keypoint Detectors and Descriptors

Figure 1a and 1b were taken from [3].



(a) Original picture



(b) Warped picture

The keypoints and their descriptors were extracted from both images with the ORB algorithm [2].



(a) Keypoints in the original picture



(b) Keypoints in the warped picture

The keypoints were matched with their nearest neighbours as shown in figure 3



Figure 3: Matches between keypoints

Some keypoints are ambiguous. They have no clear nearest neighbour and end up matched with multiple keypoints. The result was thus improved with Nearest Neighbour Distance Ration as shown in figure 4. This method ensures that only keypoints with clear nearest neighbours are part of the matching.



Figure 4: Improved matches between keypoints

2 Code

The previous results were obtained with the following code, which was adapted from [1].

```
1 import cv2 as cv
2
3 # Read both images
4 img1 = cv.imread("img1.png", cv.IMREAD_GRAYSCALE)
5 img2 = cv.imread("img2.png", cv.IMREAD_GRAYSCALE)
6
7 # Initiate an ORB detector
8 orb = cv.ORB.create()
9
10 # Find the keypoints and descriptors within each image
11 kp1, des1 = orb.detectAndCompute(img1, None)
12 kp2, des2 = orb.detectAndCompute(img2, None)
13
14 # Draw keypoint location without size or orientation
15 img3 = cv.drawKeypoints(img1, kp1, None, color=(255, 0, 0), flags=0)
16 img4 = cv.drawKeypoints(img2, kp2, None, color=(0, 255, 0), flags=0)
17
18 # Save resulting images
19 cv.imwrite("img1_kp.png", img3)
20 cv.imwrite("img2_kp.png", img4)
21
22 # Create BFMatcher object
23 matcher = cv.BFMatcher()
24
25 # Match Keypoints with their nearest neighbor
26 matches = matcher.knnMatch(des1, des2, k=1)
27 img5 = cv.drawMatchesKnn(img1, kp1, img2, kp2, matches, None,
28                         flags=cv.DrawMatchesFlags_NOT_DRAW_SINGLE_POINTS)
29
30 # Use NNDR to improve result
31 matches = matcher.knnMatch(des1, des2, k=2)
32 good = []
33 for m, n in matches:
34     if m.distance < 0.75*n.distance:
35         good.append([m])
36
37 img6 = cv.drawMatchesKnn(img1, kp1, img2, kp2, good, None,
38                         flags=cv.DrawMatchesFlags_NOT_DRAW_SINGLE_POINTS)
39
40
41 cv.imwrite("matches1.png", img5)
42 cv.imwrite("matches2.png", img6)
```

References

- [1] OpenCV. *Feature Matching*. URL: https://docs.opencv.org/3.4/dc/dc3/tutorial_py_matcher.html (visited on 05/02/2022).
- [2] OpenCV. *ORB (Oriented FAST and Rotated BRIEF)*. URL: https://docs.opencv.org/3.4/d1/d89/tutorial_py_orb.html (visited on 05/02/2022).
- [3] University of Oxford. *Affine Covariant Features*. URL: <https://www.robots.ox.ac.uk/~vgg/research/affine/> (visited on 05/02/2022).