

COMPUTER VISION 2023

LAB 3

TABLE OF CONTENTS

Objective.....	2
SIFT	2
Matching.....	2
Reference.....	3

OBJECTIVE

1. SIFT main steps
2. Match 2 images based on SIFT.
3. Hands On

SIFT

STEPS

Here is a recap from the lecture on the main steps of SIFT:

1. Constructing a scale space.
2. Laplacian of Gaussian approximation (DOG).
3. Key-point localization.
4. Eliminate edges and low contrast regions.
5. Orientation Assignment.
6. Key-point description.

SIFT IN OPENCV

Initially, create the sift object to be used later for detection/description the key-points of the image:

```
sift = cv2.xfeatures2d.SIFT_create()
```

Then pass a grayscale image and detect the key-points

```
kp = sift.detect(gray, None)
```

or you can detect and describe these key-points as well (useful for matching):

```
kp, dest = sift.detectAndCompute(gray, None)
```

MATCHING

To match features resultant from 2 images, we need them as descriptive as possible. For the sift, we need to compute the descriptors using the `detectAndCompute` method. Then the matches are calculated using brute force algorithm:

```
bf = cv2.BFMatcher(cv2.NORM_HAMMING, crossCheck=True)
```

where the first and second parameters are the distance measurement (default: NormL2), and cross check

The next step is to match the descriptors of the 2 images which returns the distance

```
matches = bf.match(des1,des2)
```

The result of the previous line is a list of DMatch objects. This DMatch object has following attributes:

- DMatch.distance - Distance between descriptors. The lower, the better it is.
- DMatch.trainIdx - Index of the descriptor in train descriptors
- DMatch.queryIdx - Index of the descriptor in query descriptors
- DMatch.imgIdx - Index of the train image.

REFERENCE

[1] https://docs.opencv.org/3.3.0/dc/dc3/tutorial_py_matcher.html