

## Laboratory – Local Features / Keypoints

### 1 Introduction

The goal of this laboratory is to get first experience in the work with local features in the task of object detection. This time, sample images and a visualization class but no python skeleton will be given. You are asked to follow the referenced OpenCV tutorials using the given sample images.

SIFT Features were protected by a patent until 2020 and are now free for commercial use.

### 2 Tasks

- a) Have a look at the topics in the “Feature Detection and Description” of the OpenCV-Python Tutorials in order to get an overview: [https://docs.opencv.org/4.6.0/db/d27/tutorial\\_py\\_table\\_of\\_contents\\_feature2d.html](https://docs.opencv.org/4.6.0/db/d27/tutorial_py_table_of_contents_feature2d.html)
- b) Read the “Introduction to SIFT (Scale-Invariant Feature Transform)” ([https://docs.opencv.org/4.6.0/da/df5/tutorial\\_py\\_sift\\_intro.html](https://docs.opencv.org/4.6.0/da/df5/tutorial_py_sift_intro.html)) and use the function `sift.detectAndCompute()` to create both an array of keypoints and descriptors of both given files `stapleRemover.jpg` and `clutteredDesk.jpg`. Display the Keypoints with the function `cv.drawKeypoints`
- c) Find out how many components the query and the train descriptor vectors have by exploring the object `refFeatures`.
- d) Replace the keypoint visualization with the given class `SiftVisualizer()`. You can right-click on a keypoint to see a graphical visualization of its descriptor.
- e) Read the section “Feature Matching” ([https://docs.opencv.org/4.6.0/dc/dc3/tutorial\\_py\\_matcher.html](https://docs.opencv.org/4.6.0/dc/dc3/tutorial_py_matcher.html)) and use the Brute Force or the FLANN matcher followed by the ratio test to find matches. Display the matches with the OpenCV function `cv.drawMatches()`
- f) What metric does the Class `BFMatcher_create()` use to compare the feature vectors? What other metrics are supported?
- g) Find out by the documentation what the flag `crossCheck` means and try it out!
- h) Read the section “Feature Matching + Homography to find Objects” ([https://docs.opencv.org/4.6.0/d1/de0/tutorial\\_py\\_feature\\_homography.html](https://docs.opencv.org/4.6.0/d1/de0/tutorial_py_feature_homography.html))

- i) Why is the method `findHomography()` powerful but still in some cases cannot reject all the outliers?
- j) What kind of objects are the return value `H`, and `mask`?
- k) How can `H` be used to transform image points from the reference to the query image?
- l) Explain in your own words what makes the strength of this object recognition method.