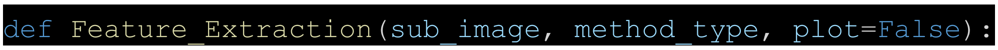
HACETTEPE UNIVERSITY
COMPUTER ENGINEERING DEPARTMENT

BBM418 COMPUTER VISION LABORATORY
PROGRAMMING ASSIGNMENT 2

Student Name: Hasan Malkoç
Student ID: 2200356826

The Implementation Details

1. Feature Extraction Method



```
def Feature_Extraction(sub_image, method_type, plot=False):
```

takes an input image and applies one of the three feature extraction methods - SURF, ORB, or SIFT - based on the input parameter method_type. The function also allows you to plot the resulting key points on the original image if the plot parameter is set to True.

Some examples below: (more in main.ipynb)

ORB

SURF

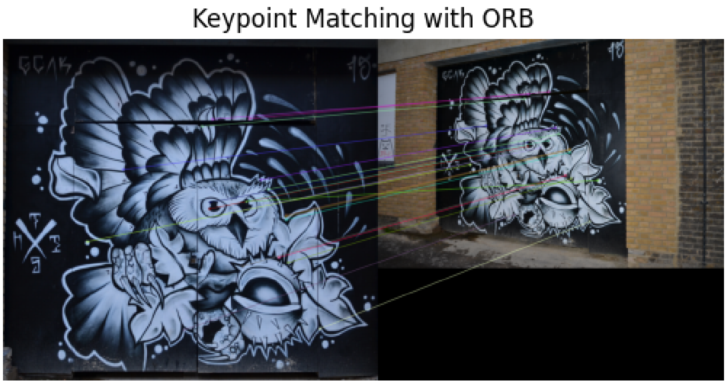
SURF

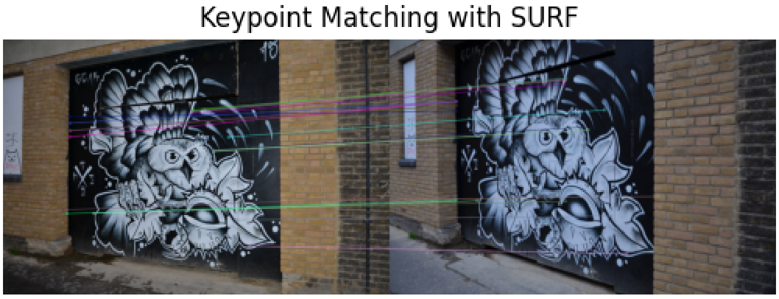
2. Feature Matching Method

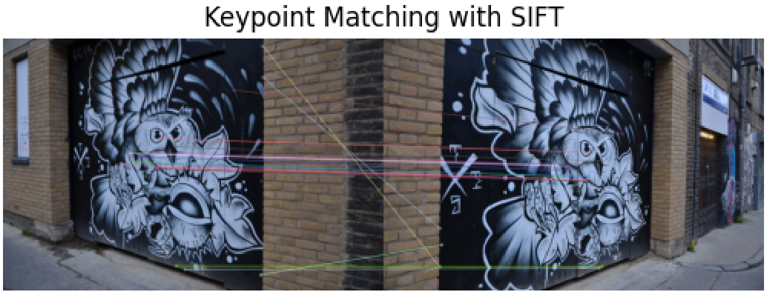
```
def match_keypoints(kp1, desc1, kp2, desc2, img1, img2,
method_type, ratio_threshold=0.75, plot=False):
```

Takes the keypoints and descriptors of two images and performs feature matching between them. It returns an array of common points found in both images. Additionally, the function can display a plot of the matched keypoints if the optional parameter plot is set to True.

Some examples below: (more in main.ipynb)







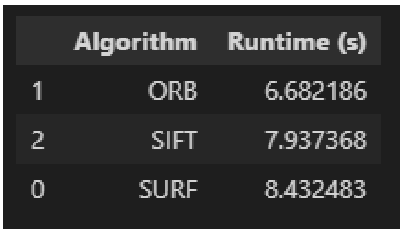
3. Finding Homography and Ransac Method

```
def homography(matches): #returns a homography matrix for given
matches between two images (by using the method in last part)
```

```
def ransac(matches, n=30, threshold=2, max_iterations=4000):
```

Implements the RANSAC (RANdom SAmple Consensus) algorithm to robustly estimate the homography matrix between two images given a set of matched features. RANSAC is particularly useful for handling outliers in the matched features, leading to more accurate results.

4. Table for runtime



	Algorithm	Runtime (s)
1	ORB	6.682186
2	SIFT	7.937368
0	SURF	8.432483