

Lab2 Readme

Ziyi Hu

November 25, 2019

1. Submission

After unzipping the zip file I have submitted, you will find two files and a folder right there, `find_matches.m`, `lab2.m` and `training_images` folder.

- The `training_images` folder contains 7 pairs of images from the one you have given us to test our code. (Otherwise the size of the zip file will exceed the limit.)
- The `find_matches.m` file is the exact one you have required in this lab. It includes two functions, `find_matches` and `cloest`, `cloest` is called in the `find_matches` function.
- The `lab2.m` file is the entrance I have used to execute my code. It loads all those image pairs from `training_images` folder and calls the `find_matches` function from `find_matches.m` to find the corresponding loctions as required. (For pos1, I have selected the strongest 25 points after using `detectHarrisFeatures` function as the interest point detector to test my work.)

2. Method

- In this task, I use seven different interest point detectors respectively on every pair of images. These detect functions are all official MATLAB functions such as `detectBRISKFeatures`, `detectHarrisFeatures`, `detectSURFFeatures`, etc.
- Then extract and match features using different official MATLAB functions for different detect functions, after that it will generate seven pairs of matched points, merge them so there is only one pair left and delete duplicate elements. I use multiple algorithms to help me find more matched points.
- Next, remove outliers from the pair of matched points. (Also official MATLAB function.)

- Last, as `pos1` is provided by you, it could be different from the matched points in `img1` I find. For every point in `pos1`, if there is a same point among matched points in `img1` then its matched point in `img2` should be the point in `pos2` corresponds to the one in `pos1`. If the point does not exist among matched points in `img1`, find the closest point among them and figure out the term between the closest point and its matched point using simple liner relation, then try the same term on this point in `pos1` to find out its matched point as the corresponding one in `pos2`. We can assume that neighbouring points should have similar corresponding relations in this task.
- Note: `closest` is a function that find the closest point in a given point array to a given point.