

[My Workspace](#)[CS-6210-O01 FALL14](#)[CS-6210-O01 SPR15](#)[CS-6250-O01 SUM15](#)[CS-6475-O01](#)

ASSIGNMENTS

Engineering

[Home](#)
[Syllabus](#)
[Announcements](#)

[Resources](#)

[Assignments](#)

[Gradebook](#)

[Email Archive](#)

[Roster](#)

[Site Info](#)

[Section Info](#)

[Peer Feedback](#)

[Schedule](#)

[Tests & Quizzes](#)

[Piazza](#)

[Help](#)

Assignment #7: Feature Detection - Returned

Title Assignment #7: Feature Detection
Student Tran, Ngoc T
Submitted Date Oct 12, 2015 1:14 am
Grade **98.0 (max 100.0)**

Instructions

Assignment 7: Feature Detection and Matching

Introduction

In this homework assignment, we will focus on putting to practice the code we have seen in lecture for feature detection and matching. This assignment will have two parts. The first part of the assignment will involve us capturing five different images, and the second part will involve us trying to do feature detection between the template and the remaining four images. We would like to stress that we want you to take these images with your camera, do not take them from the web.

Before we get started, download the assignment files under Resources > Assignments > Assignment 7 that contains the files you will need.

Once you have extracted the file above, you are ready to get started.

In the zipped file you can see the sample images which are provided for you in the images/source/sample directory. Every time you run the code, it will look inside each folder under images/source/, and attempt to find a folder that has images with filenames that contain the characters 'template'. Once it finds a folder that contains two images with those characters in the name, it will apply a feature detection and matching algorithm to them, and save the output to images/output/.

As with previous assignments, running `assignment7_test.py` directly will apply a unit test to your code and print out helpful feedback. It will also output the image result if your code is working. You can use this to debug your functions.

Part 1: Taking five images with the same subject.

First, find a subject. A stuffed animal, or small object (things with more texture will work better) that you wish to use. Take a picture of the object, and crop it so it is just the object.

1. **original image** : This will be your template. If you were 'finding Waldo', this is your picture of just Waldo (usually best to remove some background noise). In this occasion, I decided to play Finding Buzz, so this is my example template:



2. **sample:** This will be a sample image with your subject. Make sure your subject is standing very similar to how your temp. This will be your proof of concept before we begin to make changes to the image. This is my example sample image:



3. **lighting:** Take a picture of a scene with your subject, but make sure the lighting has changed so your object is visually different (it can be a bit darker or brighter, or if you have colored lights play with that). This example shows a Buzz in a darker setting.



4. **rotation:** Take a picture of a scene with your subject, but this time rotate the subject. Put it on its side, or in a silly way. Make sure the object be rotated. This example shows a Buzz lying upside down. Note: You may not just rotate one of your other images computationally, we want you to capture a new image with your object rotated.



5. **scale:** Take a picture of a scene with your subject, but this time drastically change the size of your object. It can be a close-up or far distance (granted if you go too far and your object is too small SIFT will probably not work). This example shows a closeup of the plush Buzz Bee.



Note: You'll probably take multiple pictures for each of the above categories to find the image that gives you the best results for feature detection, continue reading the entire assignment before you start this part.

Take a look at the images/source/ folder. I have included the above images for testing, each in their own folder. You should replace these with your own images since you will need to output your results for the PDF.

Part 2: Feature Detection and Matching

findMatchesBetweenImages

For this assignment you will complete one function. We strongly suggest that you look at the lectures for feature detection and matching and implement them in code. In order to tackle the code as the implementation is fairly straightforward to what we have done in lecture.

The code documentation in assignment7.py further elaborates on what you have to do. Note that regardless of which version of SIFT you are using, to create an instance of SIFT, all you have to do is call SIFT(), since the provided code takes care of automatically importing the necessary libraries.

appropriate method for you.

We also include some code that is commented out to quickly output your result. You may run this or simply run the code that is part of assignment7_test.py which we heavily recommend you use to test your function.

Note: We do not expect your feature detection to work perfectly, it should have some matches that line up correctly (if they are) and some matches that are erroneous. Depending on your algorithm and parameters of choice, some pictures may have much worse than others. Try experimenting with different objects and backgrounds in your pictures as well as different parameters for the detection and matching.

The Writeup

This is what we want you to do for the PDF.

1. Demonstrate your five input images (the template, and the sample, lighting, scale, rotation images)
2. Explain how you implemented your function using OpenCV's libraries.

Then, for each of the four images (sample, lighting, scale, rotation), explain the following:

- How many features does your algorithm get correctly? Why do you think it gets some of the errors it gets?
- Did you try taking the picture multiple times to try and improve results? What did you do to make this work?
- Include the output result for each of the four images (sample, lighting, scale, rotation) that shows the matches.

What to turn in:

Please turn in the following files. **Keep the size limit of each file to 6MB.** If you need to compress your PDF before submission, use <http://smallpdf.com/compress-pdf>:

- **assignment7.py** - Your code.
- **assignment7.pdf** - See above for the writeup, don't forget to include your input and output images in the PDF!

Submitted Attachments



[assignment7.py](#) (7 KB; Oct 10, 2015 4:17 pm)



[assignment7.pdf](#) (2 MB; Oct 12, 2015 1:11 am)

Additional instructor's comments about your submission

Programming: 30/30

Writeup: 68/70

Explained function implementation: 20/20

Input Images: 10/10

Sample: 10/10

- Included output image: 2/2

- Mentioned # of features matched correctly: 2/2
- Gave reasons for errors: 2/2
- Mentioned about multiple attempts to improve results: 4/4
-

Lighting: 10/10

- Included output image: 2/2
- Mentioned # of features matched correctly: 2/2
- Gave reasons for errors: 2/2
- Mentioned about multiple attempts to improve results: 4/4

Scale: 10/10

- Included output image: 2/2
- Mentioned # of features matched correctly: 2/2
- Gave reasons for errors: 2/2
- Mentioned about multiple attempts to improve results: 4/4

Rotation: 8/10

- Included output image: 2/2
- Mentioned # of features matched correctly: Since the matches shifted, approximately how many were actually correct? 0/2
- Gave reasons for errors: 2/2
- Mentioned about multiple attempts to improve results: 4/4

