Solution and Answers on Homework 3

Solution Explanation:

In this Homework we were tasked to implement a script to detect scale-invariant features.

Like in the previous Homework we were given a code template for the main process as well as the detector Class with its methods.

The process already has a predefined procedure on how to process an input image.

The Feature Detector Class with its methods was already provided as well. In the functions we were given on what the functionality was and which Arguments and Returns to use.

The program starts of by reading the image, creating a Feature Detector object and then creating the difference-of-Gaussian(DOG) response stack. The DOG-stack was built by first gray scaling the image, convert it into float and then normalizing. Then we used that initial image to build an array of (number of blured) blurred images to substract from each other to build the DOG-response stack.

The exact process is best described in the script.

After building the DOG-response stack we passed it along to find the features. For that we used the maximum_filter/minimum_filter from scipy to compute the non-maxima-suppression on the DOG stack over a 3x3x3 window. With that we applied the threshold and the np.argwhere() function to get the coordinates of the features.

Those features then get passed along to the draw_features() function, which loops through all the detected features (their coordinates and what level they were found) and draws circles around the features accordingly.

Note the comments in the code and the code itself for further clarification and the corresponding return types.

Qualitative Evaluation Questions:

- Q: What is the effect of changing the thresh parameter? Show and analyze example results where thresh is increased and decreased from the default.

  - A: On the left you can see the sunflower input with a low threshold value (0.025) and on the right you can see the sunflower input with a high threshold value (0.1). As expected, the low threshold allows many “small” maxima to make it through as a feature. Which means we are not very strict of what makes it a blob feature.

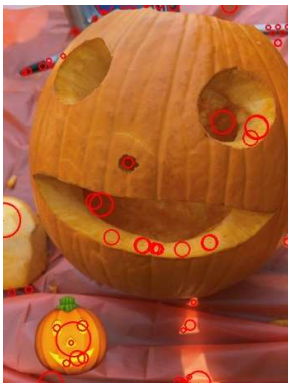
Whereas increasing the threshold value gets rid of the “smaller” maxima, meaning we are stricter with what makes a feature a feature.

- Q: What is the effect of increasing the sigma parameter? Show and analyze example results where sigma is increased from the default.

  - A: On the left you can see the sunflower input with a high sigma value (4) and on the right you can see the sunflower input with an even higher sigma value (10). As the sigma value “spreads” out the gaussian distribution, the Bandpass filters get bigger in size as well. This can be observed in the two pictures above. As the features detected grow with increasing the sigma value. Basically, with sigma you determine the smallest possible size for a feature.

- Q: Using an image of your choice (from the internet or captured by you on your phone), first hypothesize where you think features will be found, and then compare to what the detector finds.

  - A: Above is a picture taken of my pumpkin after my attempt to carve it. As it has a few “blobs” in it as well as the pumpkin emoji in the bottom I expect to maybe detect the eyes of the real pumpkin. But I at least expect it to detect the emoji pumpkins eyes and the nose of the pumpkin.



- Yeah well… At least the nose got detected and the eyes of the emoji pumpkins. Probably because my subjective definition of round is not round in the real picture. With playing around with the thresh and sigma value I could get a bit closer to my desired detections.