EECS 442 Computer Vision: HW3

Term: Fall 2017
Instructor: Jason J. Corso, EECS, University of Michigan
Due Date: 10/23 23:59 Eastern Time

Constraints: This assignment may be discussed with other students in the course but must be written independently. Over-the-shoulder Matlab coding is strictly prohibited. **Web/Google-searching for background material is not permitted. Everything you need to solve these equations is presented in the course notes and background materials, which have been provided already.**

Goals: Deepen understanding of feature scale selection and feature descriptor.

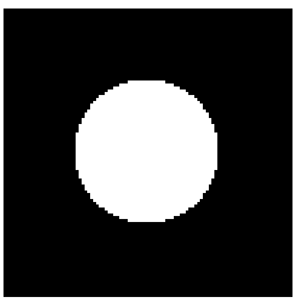
Data: You need to download the supplemental data to complete this assignment. All paths in the assignment assume the data is off the local directory. The download has data and code; it is named `hw3.zip`.

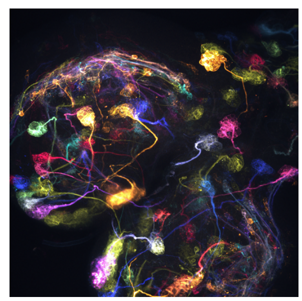
Problem 1 (40): DoG Blob Detection and Scale Selection

We discussed scale-space ideas and the relationship between the DoG and blob detection. We will implement these in Matlab and analyze their behavior. The matlab code provided goes through a sequence of steps. You will implement three pieces of the algorithm and analyze what is happening on three sample images.

The file `notes.m` runs through everything. Probably you only want to use this as reference.

The three images that are included are a synthetic image of a circle, a field of sunflowers, and a *brainbow* image of neurons in a Drosophila brain. These are depicted below, left-to-right. The fly-brain is very interesting and a relevant problem for which detecting neuron nuclei is an important problem. This image has been provided by Dr. Dawen Cai in the Department of Cell and Developmental Biology `http://www.cai-lab.org`. Each step will run on each image; you are asked to describe what you find and you should do that for each image.





1. (10) Step 1 is to create the Difference of Gaussian Scale Space. `step1.m` is the driver here and it will load/create the images and then call the `DoGScaleSpace.m` function, which you need to implement. Follow the instructions in the file and refer to the course notes for how to compute the Difference of Gaussian pyramid. Visualization code is included.

   Include visualized scale spaces for the sunflower and the circle image in your pdf. Describe the scale space and anything you find interesting about it (less than five sentences).

2. (10) Step 2 is to extract the local extrema of the resulting Difference of Gaussian Scale Space. `step2.m` is the driver here and it will call the `findSSExtrema.m`. This function looks through $3 \times 3 \times 3$ windows in space and scale to find local extrema (when the center point is larger or smaller than every other point in the window). To ensure you gain the experience of working in a scale space, you may not use the Matlab `imregionalmax` function. Visualization code is included here (called in `step2.m`) to visualize the detected extrema.

   Provide a visualization of at least the circle-image extrema detections in the scale space. Discuss what you see; why are so many extrema detected?

3. (10) Step 3 is to filter these detected local extrema to reduce the detected set down to a more usable size. `step3.m` is the driver here and it will call the `filterBlobs.m` function, which you need to fill out. This function needs to filter the blobs in two ways. First, it needs to filter out blobs that have a DoG response magnitude smaller than a certain threshold (`DoGtau`), whose value has been provided for fairness. Second, it needs to filter out regions that do not resemble blob regions. If you carefully inspect the extrema points from the question above, you will find that extrema in the 2D DoG scale space are present in many places other than blobs. You need to **invent a method** for filtering out as many as these extraneous, non-blob-like, extrema as possible. You will not be graded on the perfectness of the final method, but you are expected to leverage the methods you know already in the course to accomplish this. You should yield filtering results similar in performance to my result below.

Explain your filtering idea, the implementation and the results in your pdf report; include the blob visualizations on the circle and the sunflower image. Submission of original code is NOT required.

$\rightarrow$

4. (10) Consider the blob visualization on the drosophila image. Include the result blob image here. Discuss the detections you find; the false positives; the missing regions. Consider the structure of the image. Describe an algorithm that could not only find the blobs (cell nuclei) but also the long stringy-things connected to them (cell processes, like axons). Can the scale-selection ideas be generalized to lines?

Problem 2 (15): Matching balloons with SIFT

We have two images of colorful hot air balloons as shown below, and we want to match the balloons in the two image. Assume that we've already got the positions of feature points by using some fancy detectors. To match those feature points in the two images, we need to measure the similarity between points by comparing their feature descriptors. Since there might be variance in scale, rotation or illumination, SIFT is a good choice to help us complete this task.

Recall the three main steps to do SIFT: localize the feature points, assign orientations and calculate descriptors.

To find the scale of the features, you may use `DoGScaleSpace.m` from **Problem 1** and find the extrema when incrementing the sigma. Different from **Problem 1**, you know the exact position of the feature points so just find the scale that gives the strongest response. To get a good approximation, it is suggested to use smaller increments of sigma and more levels. One possible set of the parameters is: `k = 1.1, s1 = 4*k, level = 25`. It is not the optimal combination. You can find one set that works for your program.

It is **optional** to implement dominant orientation assignment. You will get 5 bonus points for implementing that. (But the total score of this assignment will not be greater than the full score as marked.)

You're provided a list of 5 feature points for each image named `points1.mat` and `points2.mat`. The main script is `matching_balloons.m`; it will call the `sift` function, match the points based on feature descriptors and display the matchings. Follow the instructions in the script and fill the missing part in `sift.m`. Note that this is a simplified version of SIFT and you're encouraged to learn more about the original algorithm and add more modules into the function.

Example output:

Submission requirement:

Attach the output images (with matchings) to your pdf report. Also include your code as plain text in your report. Point out what parameters you're using in your program and provide brief explanation if you do the optional task. Submit the code files to Canvas packed as a single zip file. Include all the files needed to run your program (including those provided by the instructor).

Submission Process (updated):

Submit a single pdf with your answers to these problems, include the code verbatim as text in the pdf. Include all plots and discussion in the pdf. Submit the pdf to Gradescope. The entry code of this course is **M5VRZV**.

Pack the original program files into one file and upload your code to Canvas. The problem description will clarify whether you should turn in the code for that problem.

Grading and Evaluation: The credit for each problem in this set is given in parentheses at the stated question (sub-question fraction of points is also given at the sub-questions). Partial credit will be given for both paper and Matlab questions. For Matlab questions, if the code does not run, then limited or no credit will be given.