Computer Vision: Homework 2

CV2022@CSU

1 Instructions

Python Environment We are using Python for this homework. You can find references for the Python standard library here: https://docs.python.org. To make your life easier, we recommend you to install Anaconda for Python (https://www.anaconda.com/download/). This is a Python package manager that includes most of the modules you need. We will make use of the following package: Numpy, SciPy, Matplotlib.

2 Harris Corner Detector

Task 1:

- Complete the function `corner_score()` in `corners.py` which takes as input an image, offset values (u, v), and window size W. The function computes the response $E(u, v)$ for every pixel. (Advice: You can use `np.roll` for offsetting by u and v)
- Plot your output for `grace hopper.png` for $(u, v) = \{(0, 5), (0, -5), (5, 0), (-5, 0)\}$ and window size $(5, 5)$.

Task 2:

- Implement this optimization by completing the function `harris_detector()` in `corners.py`.
- Generate a Harris Corner Detector score for every point in a grayscale version of `grace_hopper.png`, and plot these scores as a heatmap.

3 Blob Detection

3.1 LoG Filter vs. DoG Filter

Task 3:

- In `blob_detection.py`, you are given two LoG filters. You are not required to show that they are LoG, but you are encouraged to know what an LoG filter looks like. Include in your report, the following: the outputs of these two LoG filters and the reasons for their difference.
- Instead of calculating a LoG, we can often approximate it with a simple Difference of Gaussians (DoG). Specifically many systems in practice compute their "Laplacian of Gaussians" is by computing $(I * G_{k\sigma}) - (I * G_{\sigma})$ where G_a denotes a Gaussian filter with a standard deviation of a and $k > 1$ (but is usually close to 1). If we want to compute the LoG for many scales, this can be far *faster* – rather than apply a large filter to get the LoG, one can get it for free by repeatedly blurring the image with little kernels. To help understand this, we provide a three 1D filters with 501 entries in `log1d.npz`. Load these with `data = np.load('log1d.npz')`. There is a LoG filter with $\sigma = 50$ in variable `data['log50']`, and Gaussian filters with $\sigma = 50$ and $\sigma = 53$ in `data['gauss50']` and `data['gauss53']` respectively. You should assume these are representative samples of filters and that things generalize to 2D.

3.2 Single-scale Blob Detection

Task 4: Your first task is to use DoG filters to detect blobs of a single scale.

- Implement the function Gaussian filter in `blob_detection.py` that takes as an input an image and the standard deviation, σ, for a Gaussian filter and returns the Gaussian filtered image. Read in '`polka.png`' as a gray-scale image and find two pairs of σ values for a DoG such that the first set responds highly to the small circles, while the second set only responds highly the large circles. For choosing the appropriate sigma values, recall that radius and standard deviation of a Gaussian are related by the following equation: $r = \sigma\sqrt{2}$.

- Plot the two responses and report the parameters used to obtain each. Comment in your report on the responses in a few lines: how many maxima are you observing? Are there false peaks that are getting high values?

4 Cell Counting

Task 5: Your task here to to apply blob detection to find the number of cells in 4 images of your choices from the images found in the `/cells` folder.

- Find and include in your report a set of parameters for generating the scale space and finding the maxima that allows you to accurately detect the cells in each of those images. Feel free to pre-process the images or the scale space output to improve detection. Include in your report the number of detected cells for each of the images as well.

- Include in your report the visualized blob detection for each of the images and discuss the results obtained as well as any additional steps you took to improve the cell detection and counting. Include those images in your zip file under `cell_detections` as well.

5 Submission

将所有的代码、实验图像和实验报告打包成zip文件(班级-学号-姓名-HW2.zip)通过学校可视化教学平台提交，代码中所有文件路径请使用相对路径(切勿使用绝对路径)