CPEG 585 – Computer Vision

Assignment #4 - Key points for features descriptors

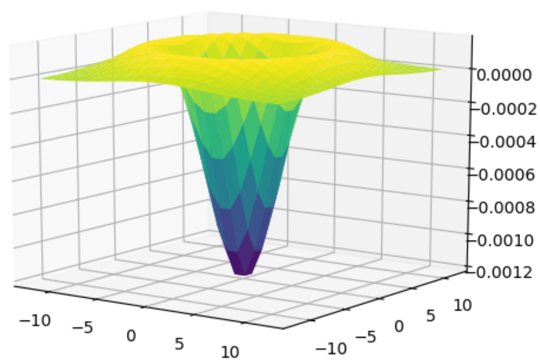
Problem #1: Program the Laplacian of Gaussian (LoG) and test it in some images for different parameters.

As explained in the lecture, the Laplacian is defined as:

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

The LoG starts by applying the Gaussian filtering first to remove the noise and then applies the Laplacian i.e., the sum of the second derivatives in the x and y direction. We can obtain the closed form for the LoG as:

$$LoG(x,y) = -\frac{1}{\pi\sigma^4}\left[1 - \frac{x^2+y^2}{2\sigma^2}\right]e^{-\frac{x^2+y^2}{2\sigma^2}}$$



To detect a blob, the value of sigma must equal the blob radius over $\sqrt{2}$. Note that the kernel size must satisfy the 6σ rule.

Modify your code to detect the blob core kernels of the sunflowers in the given image. What is the value of σ that can provide maximum derivative over the blob regions?

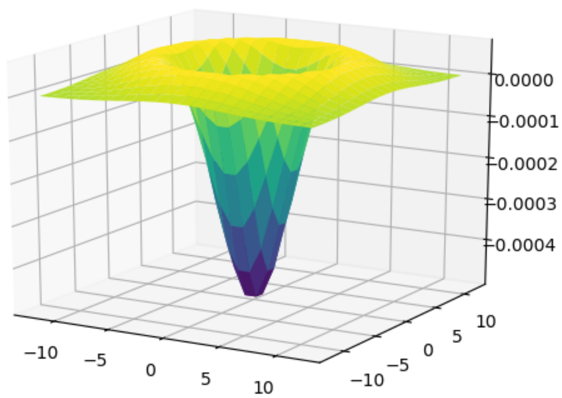
The detected blob should look like this:

Problem #2: Program the difference of Gaussian (DoG) kernels and test it for different values of σ1 and σ2.

Since a Gaussian is a low pass filter whose cut-off frequency depends upon the value of σ, the difference of Gaussian ends up being a band-pass filter, where the band depends upon the value of σ1 and σ2.

$$DoG = \frac{1}{2\pi\sigma_1} e^{-\left(\frac{x^2+y^2}{\sigma_1^2}\right)} - \frac{1}{2\pi\sigma_2} e^{-\left(\frac{x^2+y^2}{\sigma_2^2}\right)} \qquad where\ \sigma_1 > \sigma_2$$

In 2-D, the DoG kernel appears as:



Modify the provided code to detect the blob core kernels of the sunflowers in the given image. Compare the performance of the DoG to the LoG algorithm used in problem 1. Which one is better? And which one is faster?

Problem #3: The features' key points detection techniques are already built in the OpenCV library. To use these features, you have to install the OpenCV python library through the pip install as follows (if you did not install it already):

```
pip3 install opencv-python
```

This installation will include all the open-source algorithms. Unfortunately, the SURF algorithm is a patent technique so it is not included in this package. To use the patent algorithms (like SIFT, SURF,…, etc.) you need to install an older version of the opencv-contrib-python library using this pip command (This command has been tested with Python 3.7. Higher Python versions will give an error as there are other versions compatible with them). If you get an error that the version is not available, please use any 3.4 version that is close to 3.4.2.16:

```
pip3 install opencv-contrib-python==3.4.2.16
```

After installing the required library, you can run the provided code to extract the key points blobs using the Hessian technique used with the SURF algorithm.
Modify the provided code to detect the blob core kernels of the sunflowers in the given image using both SIFT key points (DOG) and SURF key points (Hessian). Compare the performance of the blobs detected using the OpenCV library to the blobs detected in Problems 1, and 2.

Problem #4

a) Using the SURF features extracted from all attached images in the scene folder, build a panorama view of the given scene using the process described in the lectures and presentations. Do not use OpenCV built-in Stitcher system.
b) In a different Python file, modify the code of part (a) by building a Laplacian Pyramid algorithm to improve the blending between the panorama-created images. (bonus question)

For the three problems, add your code, output images, and your answers to the questions, in a PDF file, to one zip file. The zip file has to be submitted to Canvas.

The file name should be lastname_firstname_ID_Assignment_4.zip
(e.g. Jobs_Steave_987654321_Assignmnet_4.zip)