Harris Corner Detector: Compute Image Gradients

Image features such as Harris Corners can serve as a compact image representation useful for task such as image matching, computing image statistics, 3D model estimation and video tracking. In this lab you will build a Harris Corner Detector via the implementation sketch in video Robo_2_4.Otherfeatures_pt2_v1_Good.mp4.

In the last section you wrote a function to filter a grayscale image with a Gaussian filter. In this section you will use your previous solution to smooth the input image prior to processing with your function to compute image gradients. Subsequent sections will require you to: (1) compute the corner score for each image pixel and (2) perform non-maximum suppression and thresholding to isolate the image locations with the strongest corner scores.

Your Script


```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
```

```
img = imread('peppers.png');
img_gray = double(rgb2gray(img));

smooth = gauss_blur(img_gray);
[I_x,I_y] = grad2d(smooth);

function [I_x,I_y] = grad2d(img)
        %% compute image gradients in the x direction
        %% convolve the image with the derivative filter from the lecture
        %% using the conv2 function and the 'same' option

```

▶ Run Script

Previous Assessment: All Tests Passed

Submit

- ✓ **Is the gauss_blur() solution correct?**
- ✓ **Is the solution for the x derivative in grad2d correct?**
- ✓ **Is the solution for the y derivative in grad2d correct?**

Output

```
Code ran without output.
```