

# Harris Corner Detector: Gaussian smoothing

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 this section you will write a function to filter a grayscale image with a Gaussian filter. Subsequent sections will require you to: (1) derive image gradients, (2) compute the corner score for each image pixel and (3) perform non-maximum suppression and thresholding to isolate the image locations with the strongest corner scores.

## Your Script

 Save  Reset  MATLAB Documentation (<https://www.mathworks.com/help/>)

```
1 img = imread('peppers.png');
2 img_gray = im2double(rgb2gray(img));
3
4 smooth = gauss_blur(img_gray);
5
6 figure()
7 subplot(121)
8 imagesc(img_gray)
9 subplot(122)
10 imagesc(smooth)
11
12 function smooth = gauss_blur(img)
13     %% Since the Gaussian filter is separable in x and y we can perform Gaussian smoothing by
14     %% convolving the input image with a 1D Gaussian filter in the x direction then
15     %% convolving the output of this operation with the same 1D Gaussian filter in the y direction.
16
17     %% Gaussian filter of size 5
18     %% the Gaussian function is defined  $f(x) = 1/(\sqrt{2\pi})\sigma \exp(-x.^2/(2\sigma^2))$ 
19     x = -2:2; % -(m-1)/2:(m-1)/2
20     sigma = 1;
21
22     gauss_size = [1 5];
23
24     gauss_filter = 1/(\sqrt{2\pi})\sigma \exp(-x.^2/(2\sigma^2));
25
26     %% using the conv2 function and the 'same' option
27     %% convolve the input image with the Gaussian filter in the x
28     smooth_x = conv2(img, gauss_filter, 'same');
29     %% convolve smooth_x with the transpose of the Gaussian filter
30     smooth = conv2(smooth_x, gauss_filter', 'same');
31 end
```


 Run Script



## Previous Assessment: All Tests Passed

Submit



 Is the gauss\_blur() solution correct?

 Are additional Matlab functions used?

## Output

