
Tutorial 7: Spatial Filtering

Convolution.

Convolution is an operation on two signals in which one signal is multiplied with the other signal at a number of different sample offsets. The values in each product signal are summed to produce each sample in the output signal, which is, therefore, the sum-of-products signal as a function of signal offset. In FIR filtering, one signal is an impulse response of some sort, and this signal will need to be time-reversed before convolution. Mathematically the convolution of two 1D signals can be expressed as:

$$f(x) * g(x) = \sum_{m=0}^{M-1} f(m)g(x-m) \quad \forall x=0,1,2,\dots,M-1$$

In Matlab this operation can be performed by the instruction (and its higher dimensional counterparts):

```
conv, conv2, convn
```

8.1 Use help to understand the syntax and conditions of the convolution operations in MATLAB.

8.2 Load an image, generate a simple kernel:

```
im3 = imread('tire','tif');  
simpKernel=ones(3,3)/9;  
outputImage=conv2(image,simpKernel);
```

(could not be any simpler!) and perform the convolution of the image and the kernel. Notice the dimensions of the output image and the visual effect. Can you easily detect differences on the image? This will depend largely on the image itself. Try now 5x5 or 9x9 kernels and see the difference.

This simple filter is obtaining an average or mean of its surrounding pixels. The average brightness of a region is defined as the sample mean of the pixel brightness within that region.

A bit more complex than the average is the process of edge detection, which consists in detecting abrupt changes in the grey level intensities of the images, and interpret them as edges of certain objects. The edges are determined by local information and can overcome problems of non-uniformity since they are based in the local variation of the pixels. There are many methods in the literature that use different approaches, like parallel differential operators such as Roberts, Sobel and Prewitt gradients or Laplacian operator. The operators respond differently to vertical, horizontal, or diagonal edges, corners, lines or isolated points. An important parameter to consider is noise immunity. Marr and Hildreth propose the Laplacian of Gaussian as the best operator; it is differential and it can be tuned at any scale to detect both sharp and blurry edges.

8.3 Generate now the following kernels and observe the effect on the image (remember to normalise them!):

-1	-1	-1	-1	2	-1	-1	-1	2	2	-1	-1	-1	-1	-1
2	2	2	-1	2	-1	-1	2	-1	-1	2	-1	-1	8	-1
-1	-1	-1	-1	2	-1	2	-1	-1	-1	-1	2	-1	-1	-1
-1	0	1	0	-1	0	0	1	0	0	-1	0	-1	-2	-1
-2	0	2	-1	4	-1	1	4	1	1	0	1	0	0	0
-1	0	1	0	-1	0	0	1	0	0	-1	0	1	2	1

Can you identify any of these kernels? Which ones are low-pass filters? Which ones are high-pass filters? Can you think of some kernels that are not 3x3?

8.4 Generate now two Gaussian kernels of bigger dimensions (9x9 for instance) with different variances, convolute with the image separately and see the effects. Then subtract one from the other to obtain a difference of Gaussians and use it as the convoluting kernel.

8.5 Properties of convolution. Try to verify the following properties of convolution:

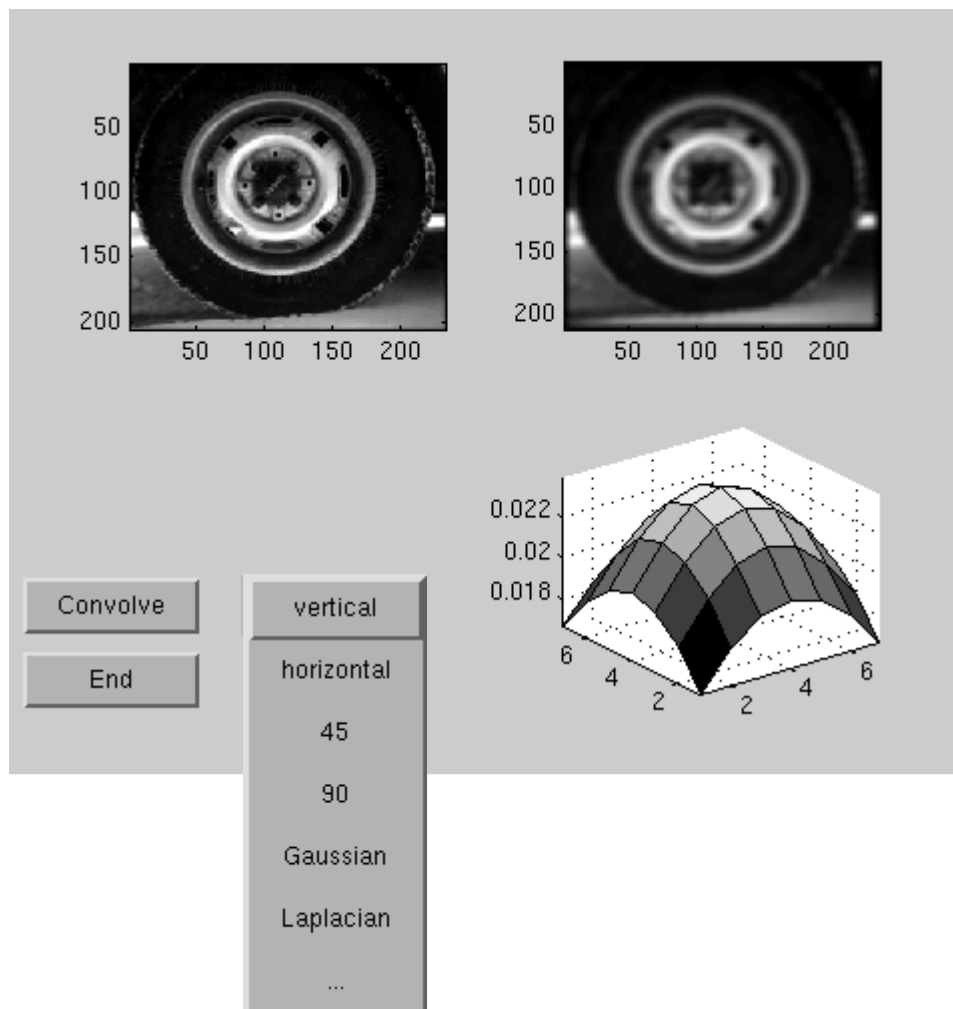
$f(x) * g(x) = g(x) * f(x)$	Commutative
$f(x) * [g(x) * h(x)] = [f(x) * g(x)] * h(x)$	Associative
$f(x) * [g(x) + h(x)] = [f(x) * g(x)] + [f(x) * h(x)]$	Distributive

These properties become important if you think of every operation as a filtering of an image, then you can use a combination of different filters to obtain a particular effect. One example of this would be a band-pass filter that is implemented with two sequential filters, one low-pass, one high-pass. Remember as well that the convolution is a linear operator, so the combination of convolutional filters is linear as well.

8.6 Can you think of a non-linear filter? Would it need a special kernel or a special operation?

Exercises

1 Create a GUI (something like the image below) where you can convolve an image with different filters; high-pass, low pass, band pass, ... and different dimensions; 3x3, 5x5, ...



2 Find out what are the following Matlab commands: uicontrol, uimenu, scrsz, nargin, varargin, findobj.