Foundations of Data Science
Image Filtering and Object Identification

Andrea Gasparini, Edoardo Di Paolo, Cirillo Atalla

October 2020

Contents

1 Image Filtering

1.1 Question 1.d

The effect of applying a filter can be studied by observing its *impulse response*. Executing the following snippet we created a test image (Figure 1) in which only the central pixel has a non-zero value:

```
img_imp = np.zeros([27,27])
img_imp[13, 13] = 1.0
plt.figure(1), plt.imshow(img_imp, cmap='gray')
```

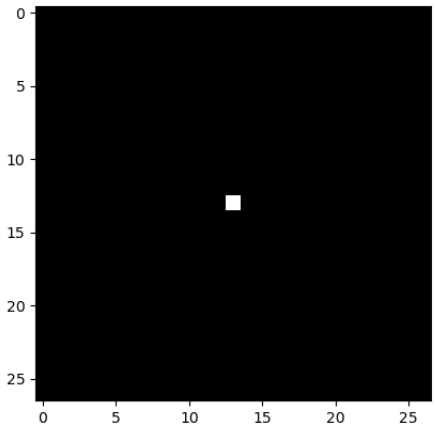

Figure 1: Test image

Executing the following snippet we created 1D Gaussian and Gaussian derivative kernels, `Gx` and `Dx` respectively.

```
sigma = 7.0
[Gx, x] = gauss_module.gauss(sigma)
[Dx, x] = gauss_module.gaussdx(sigma)
```

We applied the following filter combinations:

1. First `Gx`, then $\texttt{Gx}^T$
2. First `Gx`, then $\texttt{Dx}^T$
3. First $\texttt{Dx}^T$, then `Gx`
4. First `Dx`, then $\texttt{Dx}^T$
5. First `Dx`, then $\texttt{Gx}^T$
6. First $\texttt{Gx}^T$, then `Dx`

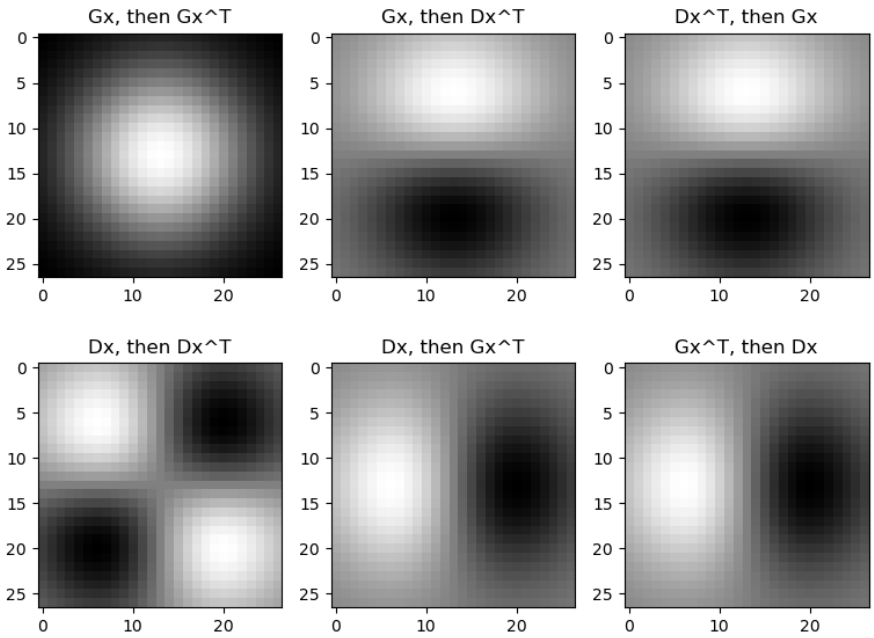

Figure 2: Applying filter combinations

As we can see in Figure 2, the first filter combination is the result of the gaussian filter applied first on the rows and then on the columns. So we compute two 1D convolution instead of one 2D convolution due to the separability of gaussian filter.
The second and third filter combinations are the same, there is no difference in applying `Gx` and then $\texttt{Dx}^T$ or viceversa. We find an edge when we apply the first derivative filter.
In the fourth filter combination there are some edges. We can see the changes from white to black and viceversa.
The fifth and sixth filter combinations are the same. This is the same case of 2nd and 3rd, but with inverted axis; there is no difference in applying `Dx` and then $\texttt{Gx}^T$ or viceversa.

1.2 Question 1.e

We implemented a `gaussderiv` method that takes an input image and generates three copies of it. The first two are smoothed according to a standard deviation σ and derived in the directions x and y respectively; the third is a combination of them, obtained as $f(img_x, img_y) = \sqrt{img_x^2 + img_y^2}$ where img_x and img_y are corresponding pixels of the two filtered images.

The results of applying `gaussderiv`, with $\sigma = 7.0$, to the provided example images (`graf.png` and `gantrycrane.png`) are shown in Figures 5 and 6.

Figure 3: `graf.png`

Figure 4: `gantrycrane.png`

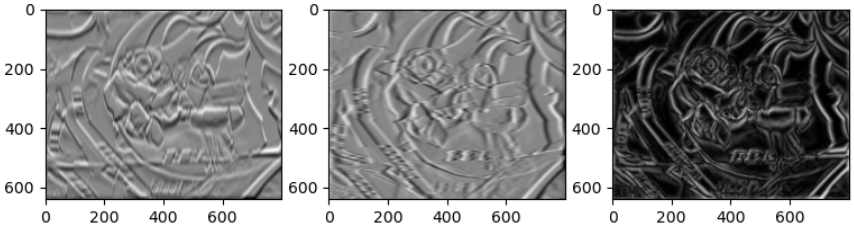

Figure 5: Results of applying `gaussderiv` on `graf.png`

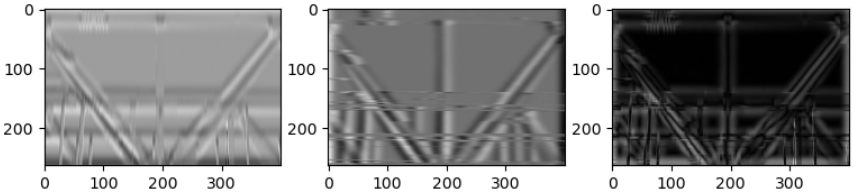

Figure 6: Results of applying `gaussderiv` on `gantrycrane.png`

Smoothing an image is important because with this process we can leave out the noise of the images.

From the left plot in Figures 5 and 6 we can see the edges on the vertical axis, while in the second one we can see the edges on the horizontal axis. The third one is the combination of the previous two.