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')
```

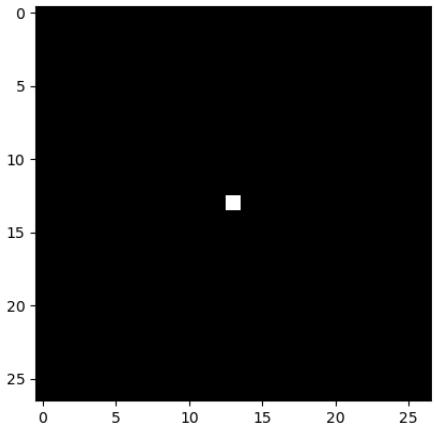

Figure 1: Test image

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`

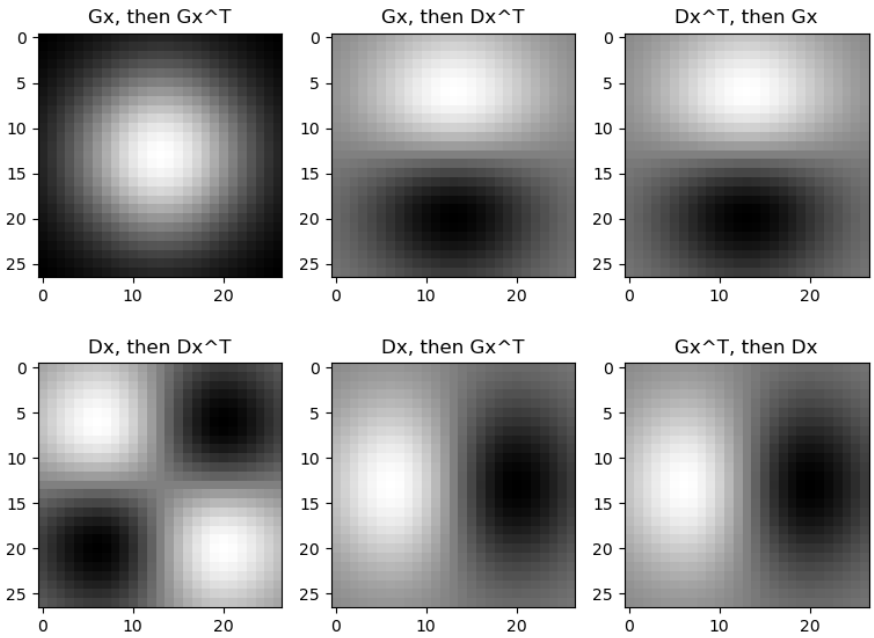

Figure 2: Applying filter combinations

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 image are the same, there are no differences in applying `Gx` and then $\texttt{Dx}^T$ or viceversa. We find an edge when we apply the first derivative filter.
In the fourth image there are some edges. We can see the changes from white to black and viceversa.
The fifth and sixth images are the same. This is the same case of 2nd and 3rd images, but the images are rotated.

1.2 Question 1.e

We implemented a `gaussderiv` method that takes an input image and generates two copies of it, smoothed according to a standard deviation σ and derived in the directions x and y respectively.

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

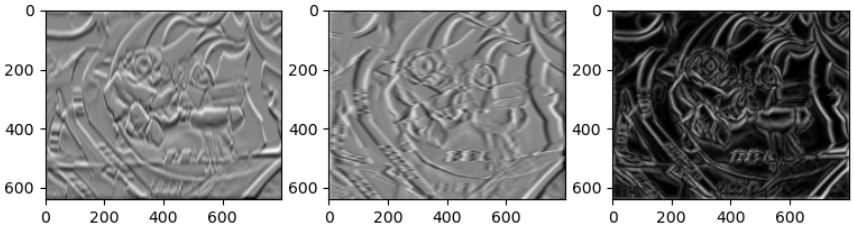

Figure 5: Results of applying `gaussderiv` on `graf.png`

TODO *Comment on the output in your report.*
TODO *Consider also why smoothing an image is important before applying the derivative filter.*

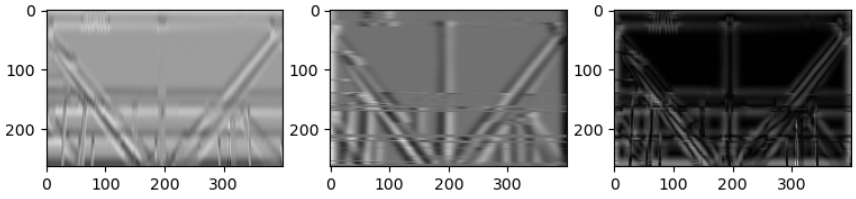

Figure 6: Results of applying `gaussderiv` on `gantrycrane.png`