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')
```
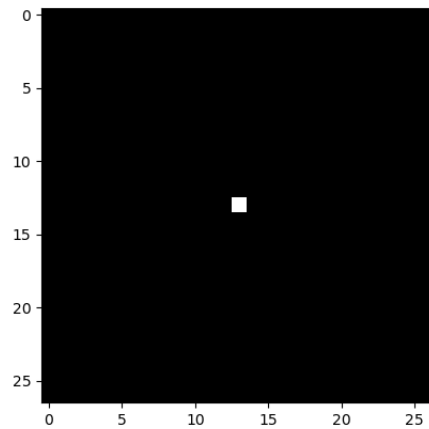

Figure 1: Test Image

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 $\texttt{Dx}^T$

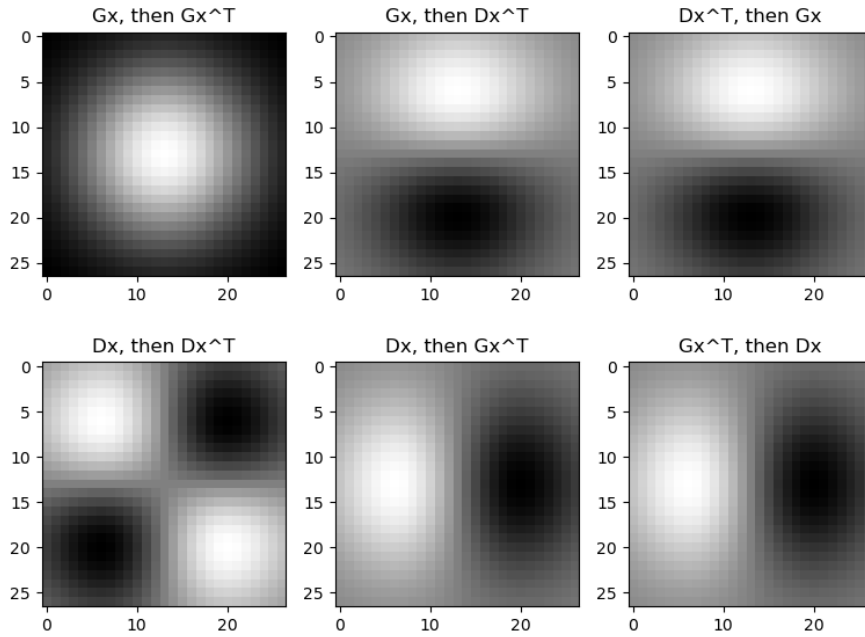5. First `Dx`, then $\texttt{Gx}^T$

6. First $\texttt{Gx}^T$, then `Dx`

Figure 2: Applying filter combinations

As we can see in Figure 2, the first filter combination [...]
TODO *What happens when you apply the following filter combinations?*

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.

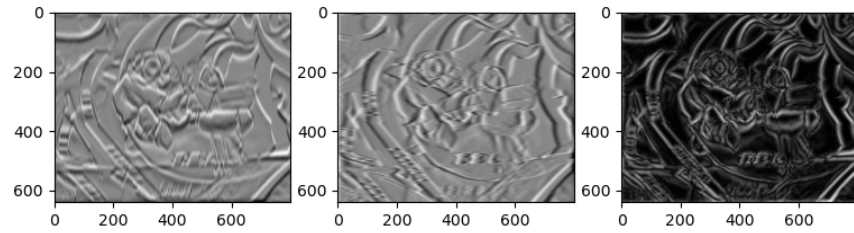Figure 3: graf.png

Figure 4: gantrycrane.png

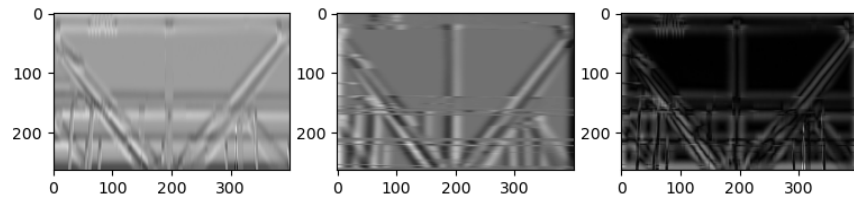Figure 5: Results of applying `gaussderiv` on `graf.png`

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

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