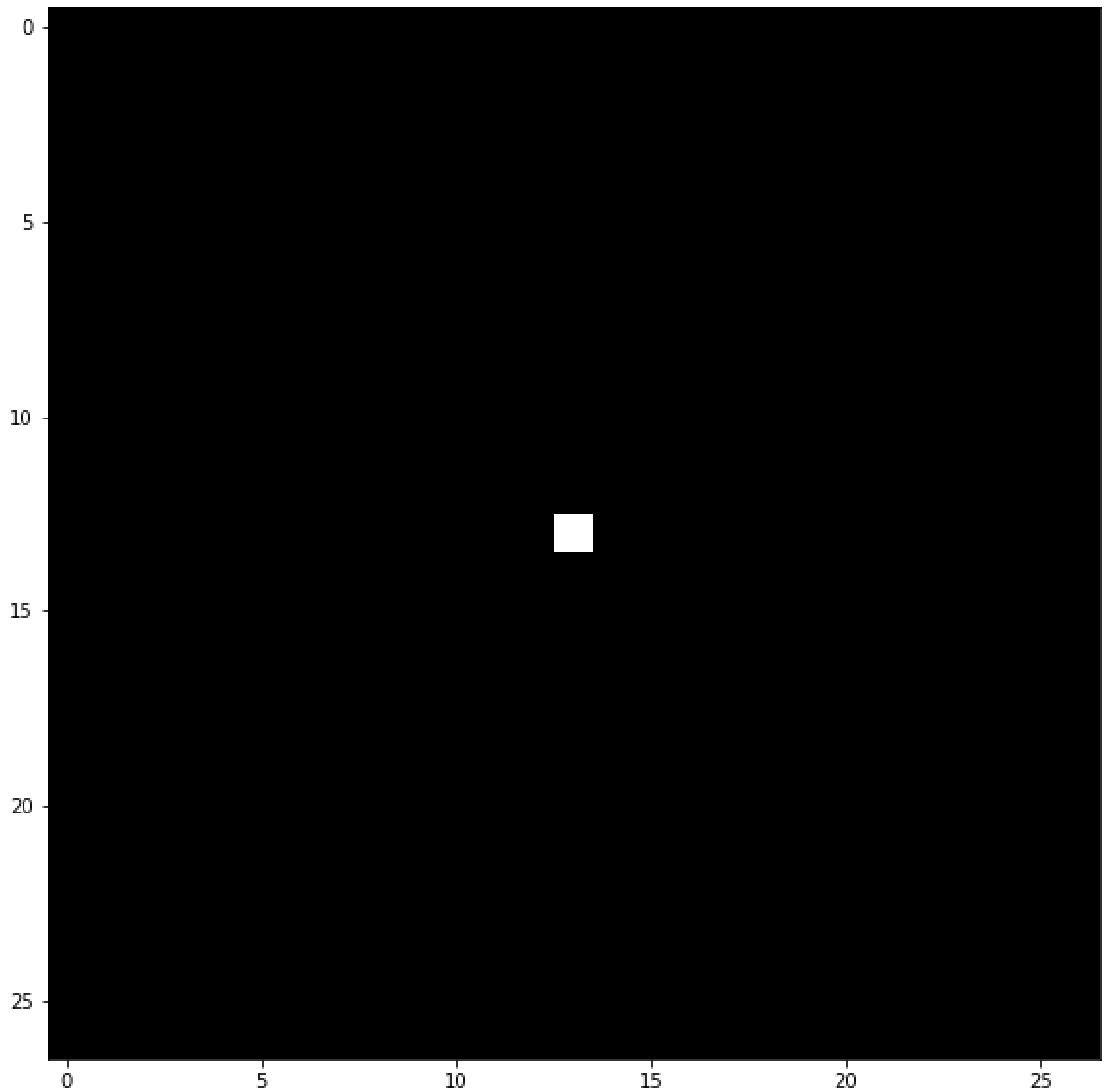
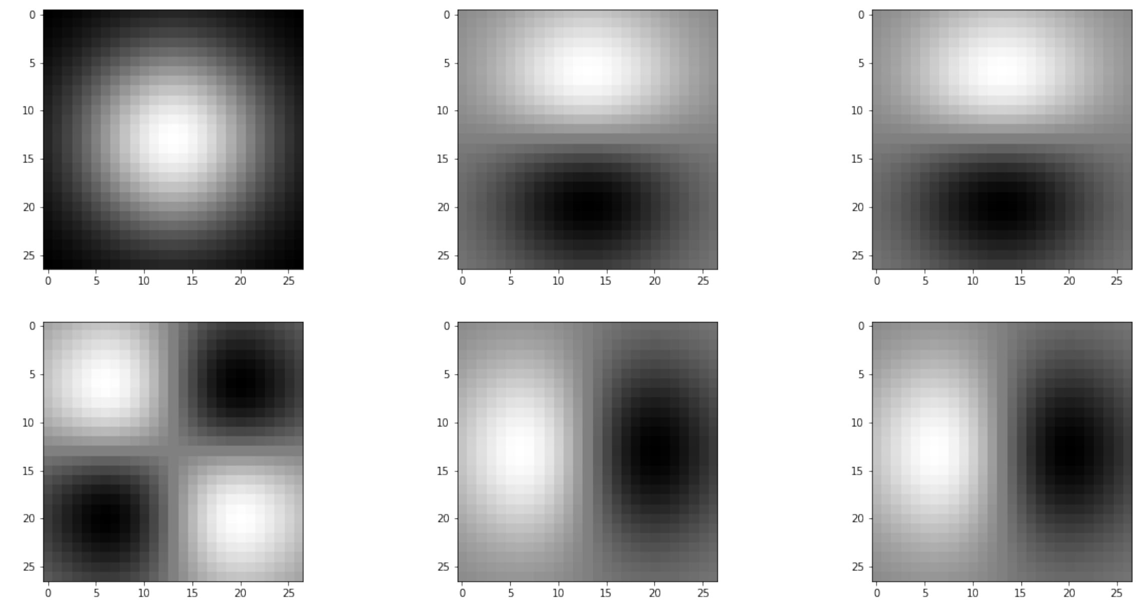
Report Question 1

November 5, 2020

```
In [7]: ## function gaussdx (Question 1.d)
```





REPORT 1-d

Subplot 1

In the first subplot implementation of 1D Gaussian kernel in both x and y direction can be observed (Gx, Gx.T). Applying the horizontal and vertical gaussian filter smoothes the image in both direction. Initial image's single dot point expands in 360 degrees. Since the end points of the gaussian filter have less weight, the center point of the picture has a lighter color while there is darkening towards the edges.

Subplot 2/3/5/6

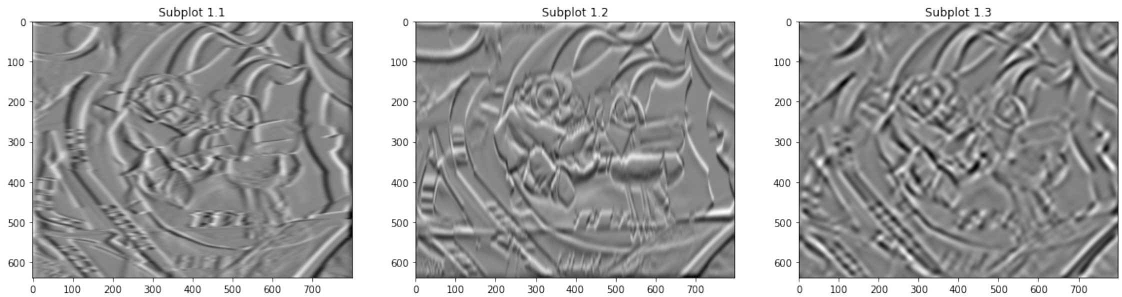
Derivative as well as convolution are linear operations. Therefore, it can be stated that, (Gx)(Dx.T)=(Dx.T)(Gx) => (Subplot2 = Subplot3) (Dx)(Gx.T)=(Gx.T)(Dx) => (Subplot5 = Subplot6) While 1D Gaussian derivative kernel in x-direction(Dx) detects vertical edges(Subplot5/6), the same kernel in y-direction(Dx.T) distinguishes horizantal edges(Subplot2/3). Applying 1D Gaussian kernel opposite direction of the 1D Gaussian derivative kernel smoothes the image to inspect edge detection in the wide picture. (Subplot2/3/5/6)

Subplot 4

In Subplot4 1D Gaussian derivative kernels are applied in both x and y-direction. This implementation aims to detect vertical and horizontal edges. Because derivative amplifies high frequencies, noise becomes problem in the Subplot4.

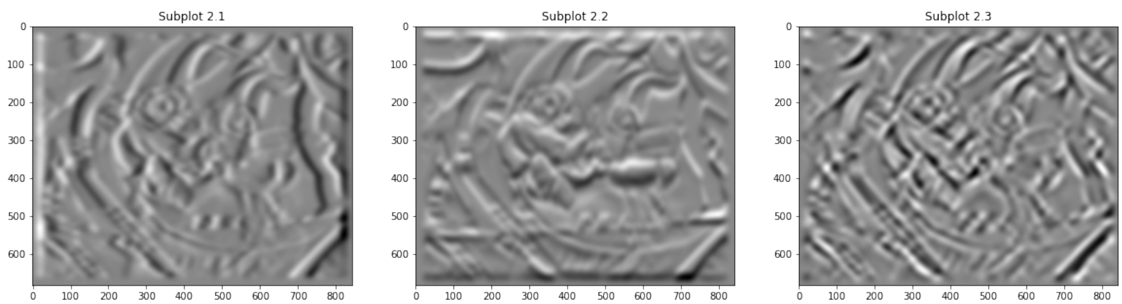
```
In [40]: ## function gaussderiv (Question 1.e)
         ## graf.png image without smoothing

         [imgDx, imgDy] = gaussderiv(img, 7.0, smoothen_image=False)
```



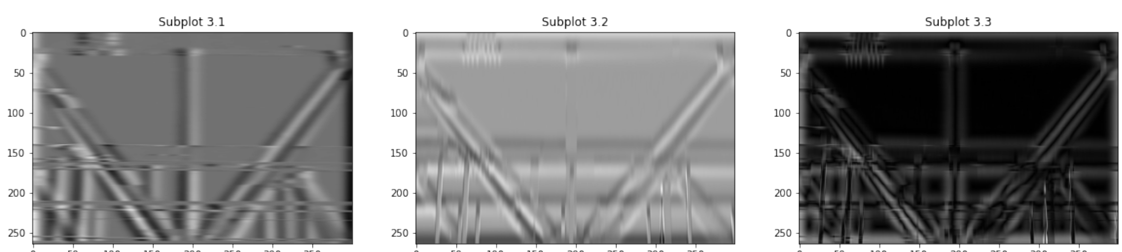
In [41]:
```
## function gaussderiv (Question 1.e)
## graf.png image with smoothing

[imgDx, imgDy] = gaussderiv(img, 7.0, smoothen_image=True)
```



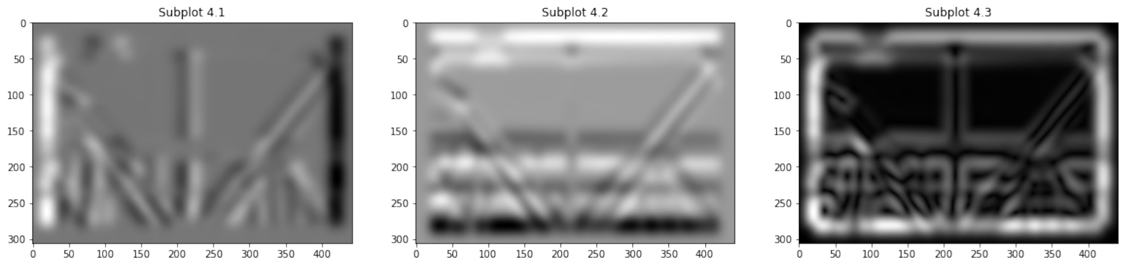
In [42]:
```
## function gaussderiv (Question 1.e)
## gantrycrane.png image without smoothing

[imgDx, imgDy] = gaussderiv(img, 7.0, smoothen_image=False)
```



In [43]:
```
## function gaussderiv (Question 1.e)
## gantrycrane.png image with smoothing

[imgDx, imgDy] = gaussderiv(img, 7.0, smoothen_image=True)
```



REPORT 1-e

While vertical edges can be detected by partial derivative in x-direction(Dx), applying partial derivative in y-direction(Dy) detects horizantal edges. Thus, using derivative kernel Dx in the smoothed image, sharpens vertical edges.(Subplot2.1/4.1) Also, using derivative kernel Dy in the smoothed image, sharpens horizontal edges.(Subplot2.2/4.2)

Diagonal Gradient Detection

Diagonal edge cannot be classfied as neither horizontal nor vertical. It is the combination of the partial derivatives in x and y-direction. Gradient image, detected diagonally, can be calculated with the formulation coded like **imgmag = np.sqrt(imgDxˆ2 + imgDyˆ2)** . It is important to state that imgDx and imgDy are smoothed before one directional edge detection. Edge detection in general perspective, gradient image, can be seen in Subplot 2.3 and Subplot4.3.

Importance of smoothing before applying the derivative filter

In order to reduce noise, before applying the derivative filter, smoothing with gaussian filter is applied.(Subplot 2./4.) Implementing derivative filter without smoothing increases even more frequency.(Subplot 1./3.) In other words, derivative filter results too many oscillations which is interpreted as noise. Thus, before using derivative filter, smoothing creates a better result.