EN2550 Exercise 3 on Spatial Filtering

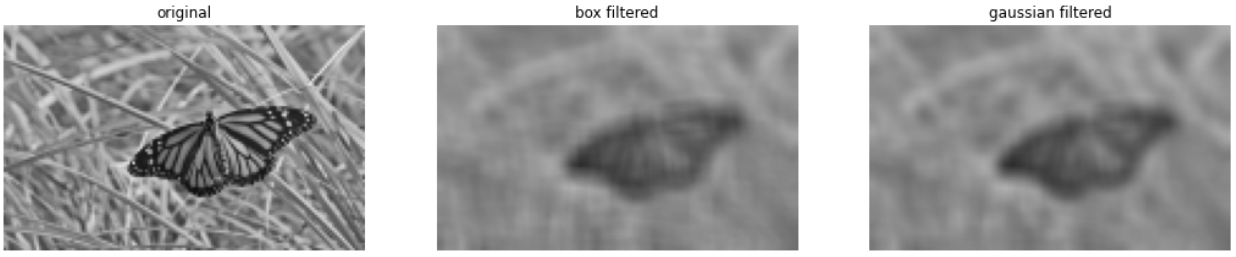
Name - Ekanayake E.M.S.S.N.

Index no - 190164M

In [ ]:
```
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt
```

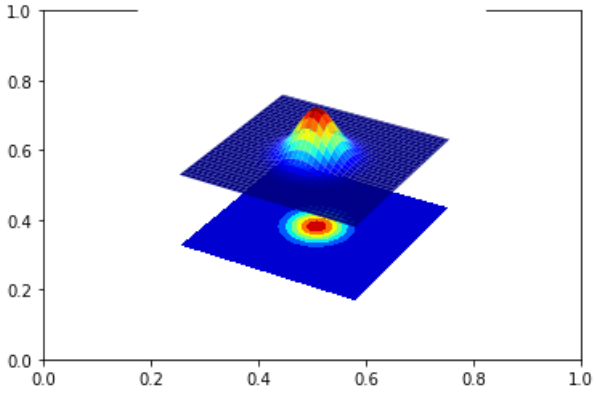
(01)

In [ ]:
```
im = cv.imread(r'butterfly.jpg',cv.IMREAD_REDUCED_GRAYSCALE_8)
assert im is not None

sigma = 4
box_kernal = 1./81*np.ones((9,9),np.float32)

im_avg = cv.filter2D(im, -1,box_kernal)
im_gaus = cv.GaussianBlur(im, (9,9),sigma)

fig, ax = plt.subplots(1,3,figsize=(18,6))
ax[0].imshow(im, cmap='gray',vmin=0,vmax=255)
ax[0].set_title('original')
ax[0].axis('off')
ax[1].imshow(im_avg, cmap='gray',vmin=0,vmax=255)
ax[1].set_title('box filtered')
ax[1].axis('off')
ax[2].imshow(im_gaus, cmap='gray',vmin=0,vmax=255)
ax[2].set_title('gaussian filtered')
ax[2].axis('off')
plt.show()
```



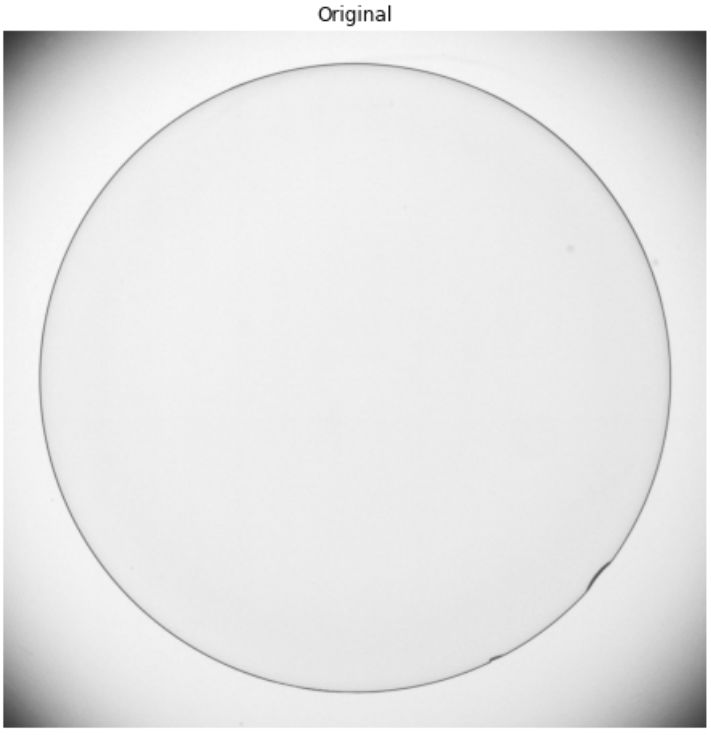
(02)

In [ ]:
```
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import cm

fig, ax = plt.subplots()
ax = fig.add_subplot(111, projection='3d')

step =0.1
sigma = 1.
X = np.arange(-5, 5+step, step)
Y = np.arange(-5,5+step, step)
XX, YY = np.meshgrid(X,Y)
g = np.exp(-(XX**2 + YY**2)/(2*sigma**2))

surf = ax.plot_surface(XX, YY, g, cmap=cm.jet)
cset = ax.contourf(XX, YY, g, zdir='z', offset=np.min(g) -1.5, cmap=cm.jet)
ax.set_zlim(np.min(g) -2, np.max(g))
plt.axis('off')
plt.show()
```

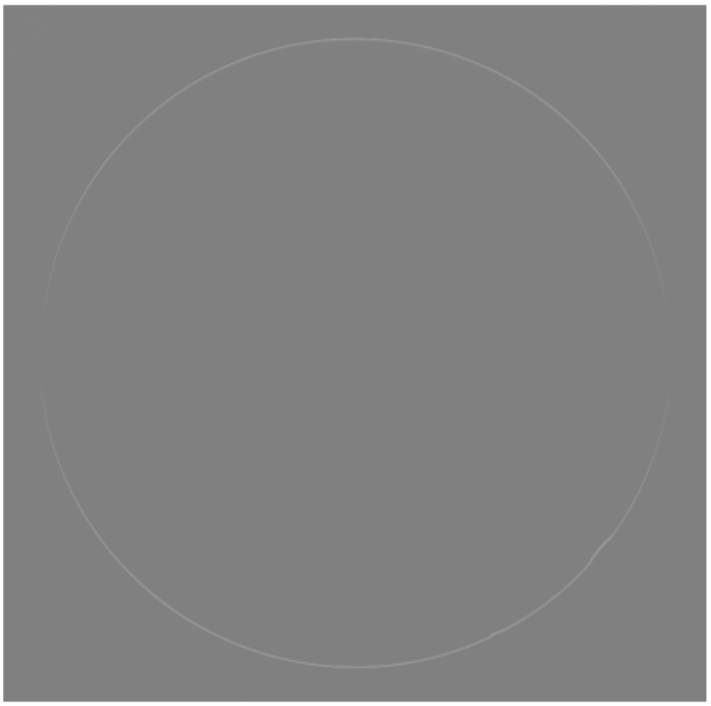


(03)

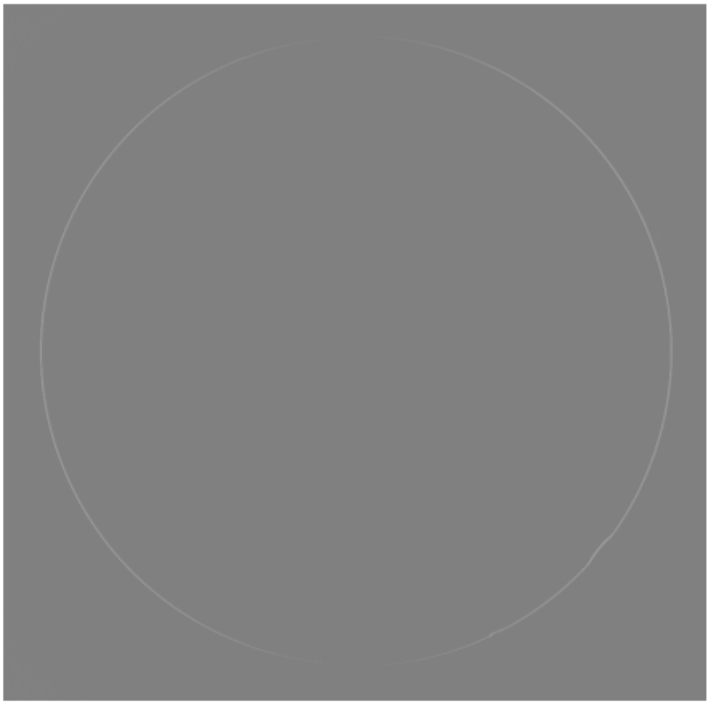
In [ ]:
```
f = cv.imread(r'contact_lens.tif', cv.IMREAD_GRAYSCALE)

fig, ax = plt.subplots(figsize=(8,8))
plt.axis('off')
ax.set_title('Original')
ax.imshow(f,cmap='gray', vmin=-0, vmax=255)
plt.show()

sobel_v = np.array([[-1,-2,-1],[0,0,0],[1,2,1]], dtype=np.float32)
f_x = cv.filter2D(f, -1, sobel_v)
sobel_h = np.array([[-1,0,1],[-2,0,2],[-1,0,1]], dtype=np.float32)
f_y = cv.filter2D(f, -1, sobel_h)
grad_mag = np.sqrt(f_x**2 + f_y**2)

fig, ax = plt.subplots(figsize=(8,8))
plt.axis('off')
ax.set_title('Sobel Vertical $f_x$')
ax.imshow(f_x,cmap='gray', vmin=-1020, vmax=1020)
plt.show()

fig, ax = plt.subplots(figsize=(8,8))
plt.axis('off')
ax.set_title('Sobel Horizontal $f_y$')
ax.imshow(f_y,cmap='gray', vmin=-1020, vmax=1020)
plt.show()

fig, ax = plt.subplots(figsize=(8,8))
plt.axis('off')
ax.set_title('Gradient Magnitude $\sqrt{f_y^2+f_x^2}$')
ax.imshow(grad_mag, cmap='gray')
plt.show()
```

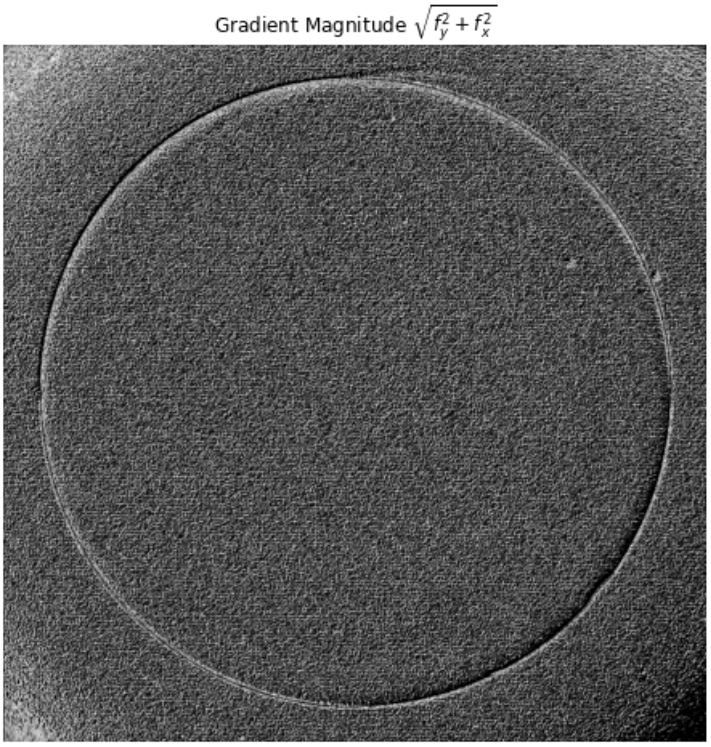


Sobel Vertical f_x



Sobel Horizontal f_y





(04)

In [ ]:
```
f = cv.imread(r'tom.jpg', cv.IMREAD_GRAYSCALE).astype(np.float32)

sigma = 2
gaussian_1d = cv.getGaussianKernel(5, sigma)
f_lp = cv.sepFilter2D(f, -1,gaussian_1d,gaussian_1d)
f_hp = f - f_lp
f_sharpened = cv.addWeighted(f, 1.0, f_hp, 2.0,0)

fig, ax = plt.subplots(1,4, figsize = (18,6))

ax[0].imshow(f, cmap='gray')
ax[0].set_title('Original')
ax[1].imshow(f_lp, cmap='gray')
ax[1].set_title('$f_{lp}$')
ax[2].imshow(f_hp, cmap='gray')
ax[2].set_title('$f_{hp}$')
ax[3].imshow(f_sharpened, cmap='gray')
ax[3].set_title('Sharpened')
for i in (0,3):
    ax[i].axis('off')
plt.show()
```