Filters Sheet - Omar Rashad Salem

Task 1

answers :

a) blurs the image

b) sharpens the image

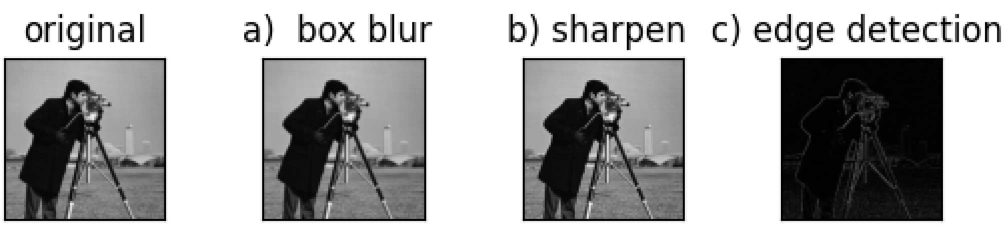
c) detects edges

In [1]:

```
import cv2 ; import numpy as np ; import matplotlib.pyplot as plt
org = cv2.imread( "man.png" , cv2.IMREAD_GRAYSCALE)

# a)  box blur kernel 3x3
a_image = org.copy()
a_kernel = np.full((3,3) , 1/9 , dtype= float )

a_image = cv2.filter2D ( a_image , -1 , a_kernel ) # -1 is to ouput same source image pixel depth


# b) sharpen kernel 3x3
b_image = org.copy()
b_kernel = np.array([[0 , -1 , 0],
                     [-1 , 5 , -1],
                     [ 0 , -1 , 0]])
b_image = cv2.filter2D( b_image , -1 , b_kernel)


#c) edge detection kernel 3x3
c_image = org.copy()
c_kernel = np.array([[-1 , -1 , -1],
                      [-1 , 8 , -1],
                     [-1 , -1 , -1]])
c_image = cv2.filter2D ( c_image , -1 , c_kernel )

show all images
titles = ['original','a)  box blur ' ,'b) sharpen ' , 'c) edge detection ' ]
images = [org , a_image , b_image , c_image]
for i in range(4) :
 plt.subplot(4,4,i+1) # row colums index
 plt.imshow(images[i],"gray",vmin = 0, vmax = 255)
 plt.title(titles[i])
 plt.xticks([])
 plt.yticks([])

plt.xticks([])
plt.yticks([])

# cv2.waitKey(0)
# cv2.destroyAllWindows
```

Out[1]:

```
([], [])
```



Task 2

answers :

a) it's a Difference filter

b) min = (-255 -(*2255) - (2* 255)) = -1275 => (0 intensity pure black) max = 1275 => (255 intensity pure white pixel)

c) corelation result will be the kernel values in center but fliped 180 degrees (anti c.w) and serrounded with black pixels (zero intensity)

EXTRA : if convolution was done between impulse and the kernel, result will be exact same kernel in center.

d) yes : because convolving flips the kernel before applying the summation formula. so the difference filter result will be sharpend and inverted.

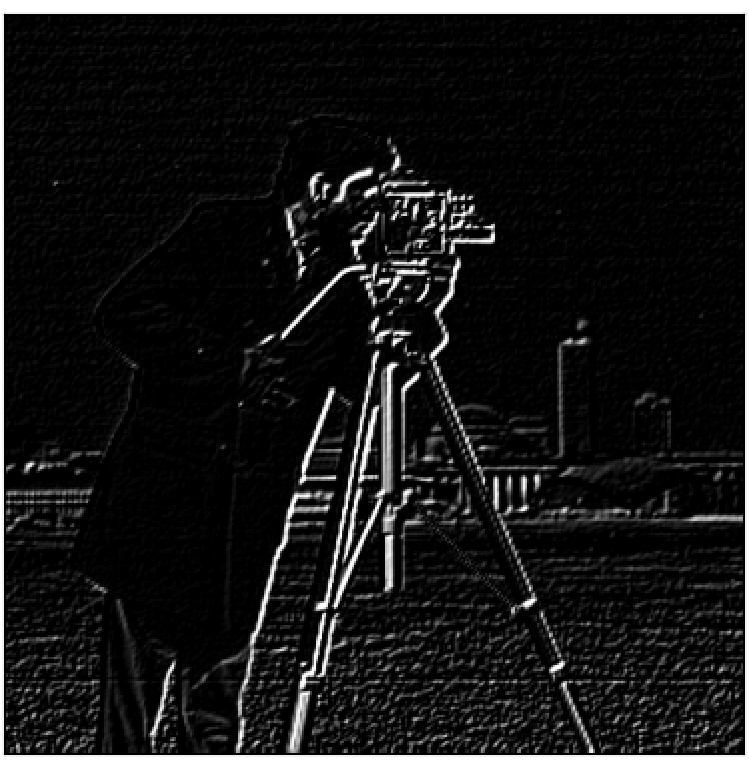
e) not an isotropic filter

In [2]:

```
#TASK 2 : a

import cv2 ; import numpy as np ; import matplotlib.pyplot as plt
from scipy.ndimage import rotate

org = cv2.imread( "man.png" , cv2.IMREAD_GRAYSCALE)
# cv2.imshow("original" , org )

cpy = org.copy()
kernel = np.array ([[-1 , -2 , 0] , [-2 , 0 , 2], [0 , 2 , 1]])

cpy = cv2.filter2D(cpy , -1 , kernel)


plt.imshow(cpy,"gray")

plt.xticks([])
plt.yticks([])
# cv2.waitKey(0)
```

Out[2]:

```
([], [])
```



In [3]:

```
# TASK 2 : C

import cv2 ; import numpy as np ; import matplotlib.pyplot as plot
from scipy.ndimage import rotate

#make impulse signal
org = np.zeros((510,510) , dtype = np.uint8)
org[255 , 255] = 255

cv2.imshow("original" , org )

cpy = org.copy()
kernel = np.array ([[-1 , -2 , 0] , [-2 , 0 , 2], [0 , 2 , 1]])

#convolve impule with 180 deg rotated kernel == correlation
kernel = rotate(kernel , 180 )
cpy = cv2.filter2D(cpy , -1 , kernel)




cv2.imshow( "Task 2 - c " , cpy )
cv2.waitKey(0)
```

Out[3]:

```
-1
```

In [1]:

```
#TASK 2 : d

import cv2 ; import numpy as np ; import matplotlib.pyplot as plt
from scipy.ndimage import rotate

org = cv2.imread( "man.png" , cv2.IMREAD_GRAYSCALE)

cpy = org.copy()
kernel = np.array ([[-1 , -2 , 0] , [-2 , 0 , 2], [0 , 2 , 1]])

cpy = cv2.filter2D(cpy , -1 , kernel)



#convolve impulse with 180 deg rotated kernel == correlation
kernel = rotate(kernel , 180 )
cpy2 = cv2.filter2D(cpy , -1 , kernel)

cv2.imshow("conv" , cpy)
cv2.imshow( 'core' , cpy2)

# titles = ['original',' convolve' , 'correlate ']
# images = [org , cpy ,cpy2]
# for i in range(3) :
#  plt.subplot(4,4,i+1) # row colums index
#  plt.imshow(images[i] , 'gray')
#  plt.title(titles[i])
#  plt.xticks([])
#  plt.yticks([])
```

```
The Kernel crashed while executing code in the the current cell or a previous cell. Please review the code i
n the cell(s) to identify a possible cause of the failure. Click <a href='https://aka.ms/vscodeJupyterKernel
Crash'>here</a> for more info. View Jupyter <a href='command:jupyter.viewOutput'>log</a> for further detail
s.
```

Task 3

answer : the box filter (uniform weight smoothing filter)

Task 4

answer : because it removes the intensity spikes in pixels.

END OF SHEET 4