2019

Digital Image Processing

CE-38-B

MISHAAL SAFEER

SAMEEN ABBAS

FOHA KHALID

MUHAMMAD HASHIR SHOAIB

Task 1

Q:

Create a distance map of 480 rows and 640 columns as shown below, The image has 0 at the corners and 255 at the middle.

P (i, j) = 255 – (r / c * 255)

Where: c = distance between location (0, 0) and center of image.
r = distance between pixel location and center of image.

Algorithm:

```
def circularFilter(row, col):
    ciPic = np.zeros((row, col))
    rowCentre = row / 2
    colCentre = col / 2
    c = m.sqrt((0 - rowCentre) * (0 - rowCentre) + (0 - colCentre) * (0 - colCentre))
    for i in range(0, row):
        for j in range(0, col):
            r = m.sqrt((i - rowCentre) * (i - rowCentre) + (j - colCentre) * (j - 
colCentre))
            ciPic[i][j] = int(255 - (r / c * 255))
    return ciPic


cv.imwrite("circleFilter.png", circularFilter(480, 640))
```

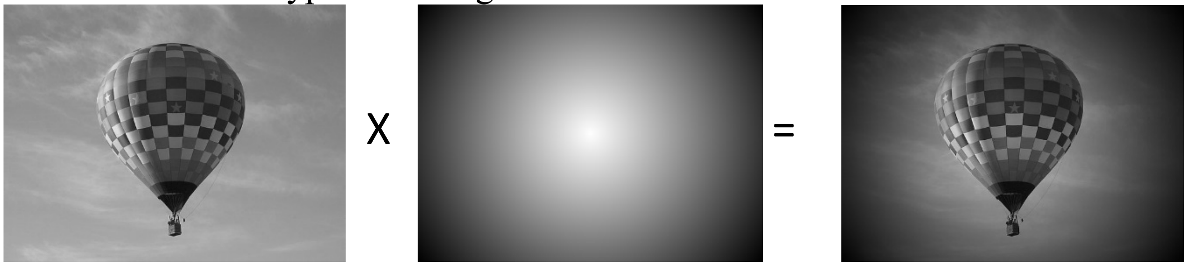
Output:

Figure 1: circleFilter.png

Tack 2

Q:

Multiply the image obtained in Task 1 with image of the balloon. Take necessary measures for data types and ranges.



Algorithm:

```
def mapRange(s):
    a1, a2, b1, b2 = 0, 200 * 267, 0, 255
    return b1 + ((s - a1) * (b2 - b1) / (a2 - a1))


balArray = cv.imread("./Picture1.png", 0)
brow, bcol = np.shape(balArray)
temparray = circularFilter(brow, bcol)
new1array = np.zeros((brow, bcol))
for i in range(0, brow):
    for j in range(0, bcol):
        new1array[i][j] = int(mapRange(temparray[i][j] * balArray[i][j]))
cv.imwrite("filteredPic.png", new1array)
print(new1array)
```

Output:

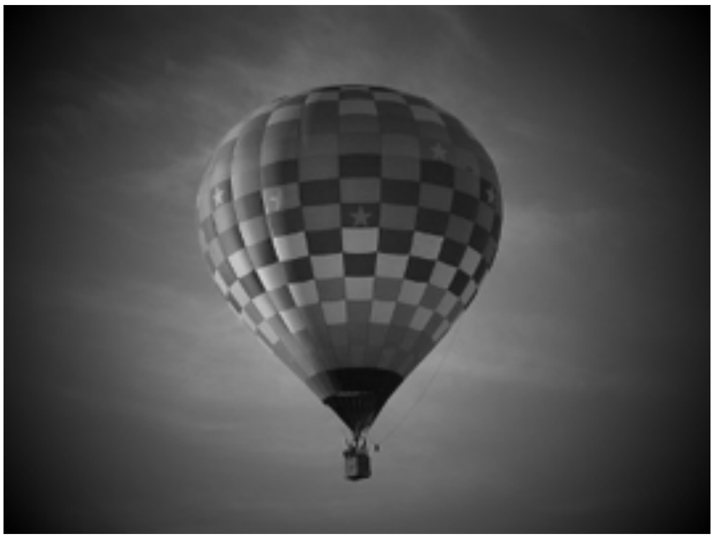
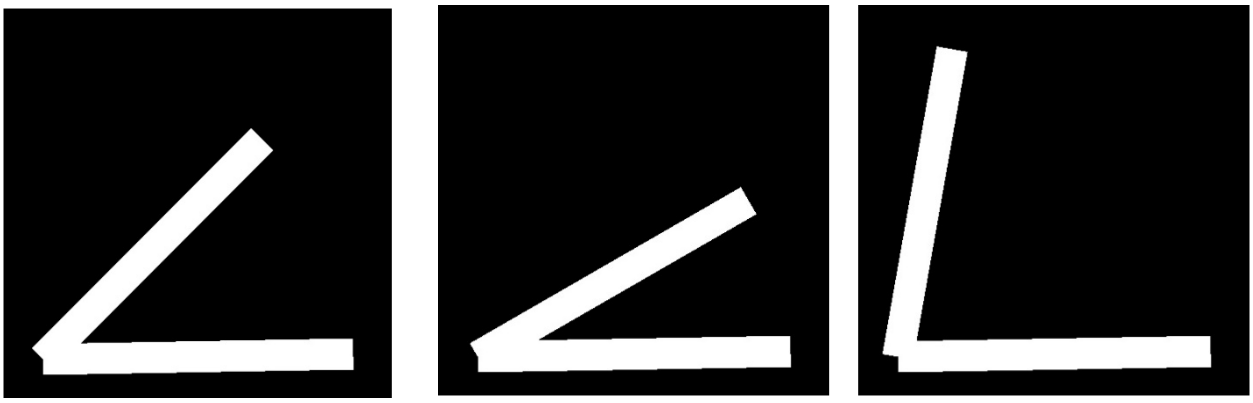


Figure 2: filteredPic.png

Task 3

Q:

Write a code that take theta from the user and draw a rectangle in that direction as shown. The size of the image should be 500 x 500, origin of the angle is at location (i, j) = (450, 50) and, length and width of rectangle is equal to 400 and 20 pixels respectively. Description is on the next page.



Algorithm:

```
angleImage = np.zeros((500, 500))


def makeRectangleinWhite(angleinRadian, Image):
    x, y = [450, 50]
    t = angleinRadian
    l, w = [400, 20]
    ax = int(x + (w * m.cos(t)))
    ay = int(y + (w * m.sin(t)))
    bx = int(x - (w * m.cos(t)))
    by = int(y - (w * m.sin(t)))
    px = int(x - (l * m.sin(t)))
    py = int(y + (l * m.cos(t)))
    cx = int(px + (w * m.cos(t)))
    cy = int(py + (w * m.sin(t)))
    dx = int(px - (w * m.cos(t)))
    dy = int(py - (w * m.sin(t)))
    for i in range(0, 500):
        for j in range(0, 500):
            if j < ((i-ax)/m.tan(-t))+ay and j > ((i-bx)/m.tan(-t))+by and j > ((i-
ax)*m.tan(t))+ay and j < ((i-cx)*m.tan(t))+cy:
                Image[i][j] = 255
    return Image


angleImage = makeRectangleinWhite(0.57, angleImage)
angleImage = makeRectangleinWhite(0.0001, angleImage)
cv.imwrite("angleImage.png", angleImage)
```

Output:

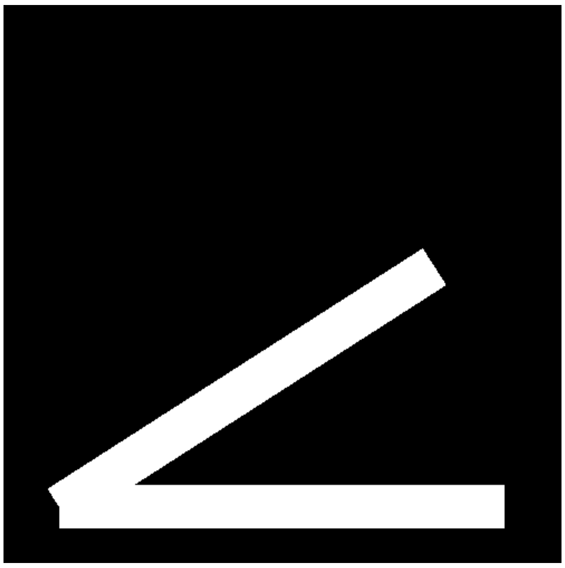


Figure 3: angleImage.png