

Homework 2 (Due:3/16)

(A) Given a grayscale image I ,

Step 1: Use the dithering matrix D_2 to generate an array D of image size by repeating D_2

$$D_2 = \begin{bmatrix} 0 & 128 & 32 & 160 \\ 192 & 64 & 224 & 96 \\ 48 & 176 & 16 & 144 \\ 240 & 112 & 208 & 80 \end{bmatrix}$$

D

D_2	D_2	D_2	D_2
D_2	D_2	D_2	D_2
D_2	D_2	D_2	D_2
D_2	D_2	D_2	D_2

Step 2: Threshold image I by

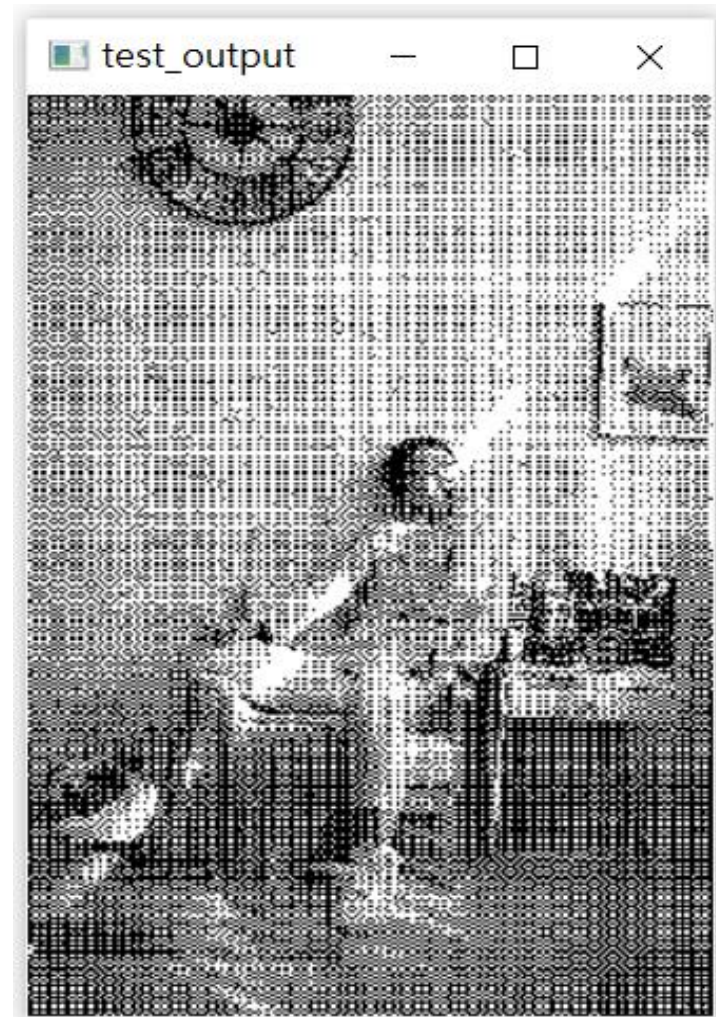
$$I'(i, j) = \begin{cases} 255 & \text{if } I(i, j) > D(i, j) \\ 0 & \text{if } I(i, j) \leq D(i, j) \end{cases}$$

Step 3: Show images I and I'

input



output



Source code

#讀入並顯示影像

```
imgA = cv2.imread('test_input.jpg')
imgA = cv2.cvtColor(imgA,cv2.COLOR_BGR2GRAY)
cv2.namedWindow('test_input', cv2.WINDOW_NORMAL)
cv2.imshow('test_input', imgA)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

#讀取影像大小

```
heightA, widthA = imgA.shape
print("影像大小為 :",imgA.shape)
```

#製作DA

```
dA = np.array([[0,128,32,160],
               [192,64,224,96],
               [48,176,16,144],
               [240,112,208,80]])
hA=int(round(heightA/4))
wA=int(round(widthA/4))
DA = np.tile(dA,(hA+1, wA+1))
```

Source code

```
#Threshold imgA
for i in range(heightA):
    for j in range(widthA):
        if (imgA[i][j])>DA[i][j]:
            imgA[i][j]=255
        else:
            imgA[i][j]=0
cv2.imwrite('test_output_A.jpg', imgA)
print("test_output_A.jpg is SAVED~")

#顯示輸出
cv2.namedWindow('test_output_A',
cv2.WINDOW_NORMAL)
cv2.imshow('test_output_A', imgA)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

(B) Extend to $n = 4$ gray values

1. $255 / 3 = 85$

2. $Q(i, j) = [I(i, j) / 85]$

3. $D_1 = \begin{bmatrix} 0 & 56 \\ 84 & 28 \end{bmatrix} \Rightarrow_{\text{extend}} D$

4. $I'(i, j) = Q(i, j) + \begin{cases} 1 & \text{if } I(i, j) - 85Q(i, j) > D(i, j) \\ 0 & \text{if } I(i, j) - 85Q(i, j) \leq D(i, j) \end{cases}$

5. Scale values of I' so that its values are in $[0, 255]$ for displaying

input



output



Source code

#讀入並顯示影像

```
imgB = cv2.imread('test_output_A.jpg')
imgB = cv2.cvtColor(imgB,cv2.COLOR_BGR2GRAY)
cv2.namedWindow('test_input',
cv2.WINDOW_NORMAL)
cv2.imshow('test_input', imgB)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

#讀取影像大小

```
heightB, widthB = imgB.shape
print("影像大小為 :",imgB.shape)
```

#製作DB

```
dB = np.array([[0,56],
               [84,28]])
hB=int(ceil(heightB/2))
wB=int(ceil(widthB/2))
DB = np.tile(dB,(hB, wB))
```

Source code

```
#Threshold imgB
Q=imgB/3
for i in range(heightB):
    for j in range(widthB):
        if ((imgB[i][j])-85*(Q[i][j])) > DB[i][j]:
            imgB[i][j]=round(Q[i][j])+1
        else:
            imgB[i][j]=round(Q[i][j])
cv2.imwrite('test_output_B.jpg', imgB)
print("test_output_B.jpg is SAVED~")

#顯示輸出
cv2.namedWindow('test_output_B',
cv2.WINDOW_NORMAL)
cv2.imshow('test_output_B', imgB)
cv2.waitKey(0)
cv2.destroyAllWindows()
```


Comments

In B part, I don't understand why we use $\text{img}[i][j] - 85 * Q[i][j] > D[i][j]$ instead of $\text{imgB}[i][j] - 85 * Q[i][j] > DB[i][j]$ when $85 * Q[i][j] = \text{img}[i][j]$