

Spatial Image Enhancement

609410162 彭郁翔

Data due:10/28

Data handed in:10/20

Technical description

在此之前我先將圖檔從.bmp 檔轉為.jpg 檔，原因會在下面的 Discussion 說明。
利用以下三種方法來增強影像：

(1) power-law (gamma) transformation

此方法我使用 python 來撰寫，從基本公式得知 $s = c * r^{\gamma}$ ，其中 c 和 r 為 positive constants，並藉由調整 γ 參數大小來決定把影像變亮或變暗，而我是將影像調暗。

Ex:

```
10 for gamma in [0.1,0.5,1.0,1.5,2.0]: #可調整參數來決定要變暗或變亮
11     gamma_transformed1 = np.array(255*(img1 / 255) ** gamma, dtype = 'uint8') # s = c * r ^ gamma
12     gamma_transformed2 = np.array(255*(img2 / 255) ** gamma, dtype = 'uint8')
13     gamma_transformed3 = np.array(255*(img3 / 255) ** gamma, dtype = 'uint8')
```

(2) histogram equalization

此方法我使用 python 來撰寫，根據方程式 $g_{(i,j)} = (L-1) \sum_{n=0}^{f(i,j)} p(n)$ 來做，
設 f 為原始灰階影像， g 為 histogram equalization 後的影像， L 為灰階度，此設為 256，因為影像畫素強度範圍是 0~255， $p(n)$ 是影像 f 中強度是 n 佔的比例。

```
6 def histoequ(img, L=256):
7
8     #計算histogram
9     histo_gram = np.bincount(img.flatten(), minlength=L)
10    '''print(histo_gram)''' #test
11
12    #描繪出function
13    uniform_hist = (L - 1)*(np.cumsum(histo_gram)/(img.size * 1.0))
14    uniform_hist = uniform_hist.astype('uint8')
15    '''print(uniform_hist)''' #test
16
17    #設定pixel的強度
18    h = img.shape[0]
19    w = img.shape[1]
20    uniform_gray = np.zeros(img.shape, dtype='uint8')
21    for i in range(h):
22        for j in range(w):
23            uniform_gray[i,j] = uniform_hist[img[i,j]]
24
25    return uniform_gray
```

(3) image sharpening using the Laplacian

此方法我使用 matlab 來撰寫，根據定義及以下矩陣表示：

0	1	0
1	-4	1
0	1	0
1	1	1
1	-8	1
1	1	1

$$g(x,y) = f(x,y) + c[\Delta^2 f(x,y)]$$

Fig 2

f(x,y) is the input image

g(x,y) is the sharpened image

此 c 我是設定為 -1

這裡我先分配 3 張跟原圖 size 一樣的零矩陣，設定 mask1 或 mask2 來和原本的 image 矩陣做相乘。

```

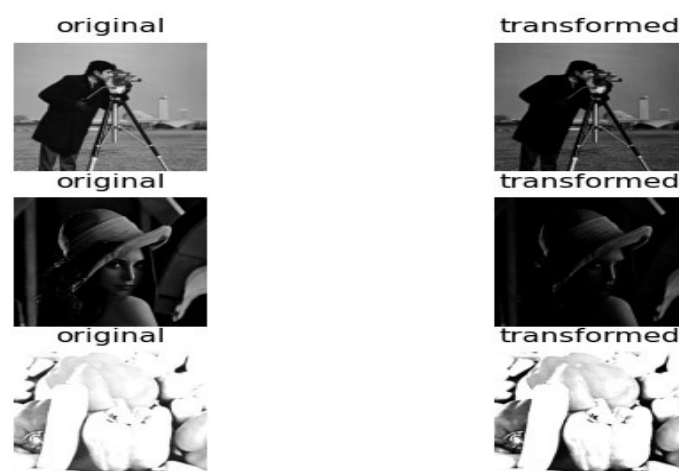
12 %把transformed過的image先設定為零矩陣
13 - img1_t = zeros(size(img1));
14 - img2_t = zeros(size(img2));
15 - img3_t = zeros(size(img3));
16
17 %設定Masks
18 %mask1 = [0 1 0; 1 -4 1; 0 1 0];
19 - mask2 = [1 1 1; 1 -8 1; 1 1 1];
20
21 - img1 = padarray(img1, [1, 1]);
22 - img1 = double(img1);
23 - img2 = padarray(img2, [1, 1]);
24 - img2 = double(img2);
25 - img3 = padarray(img3, [1, 1]);
26 - img3 = double(img3);
27
28 %利用Laplacian equation
29 - for i=1:size(img1,1)-2
30 -     for j=1:size(img1,2)-2
31 -         img1_t(i, j, 1) = sum(sum(mask2.* img1(i:i+2, j:j+2)));
32 -         img1_t(i, j, 2) = sum(sum(mask2.* img1(i:i+2, j:j+2)));
33 -         img1_t(i, j, 3) = sum(sum(mask2.* img1(i:i+2, j:j+2)));
34 -     end
35 - end

```

Experimental results

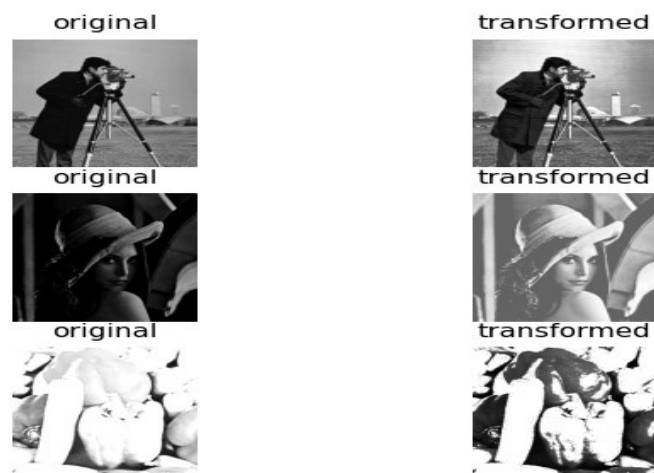
(1) power-law (gamma) transformation

由此可看出經過轉換後和原圖相比變暗了。

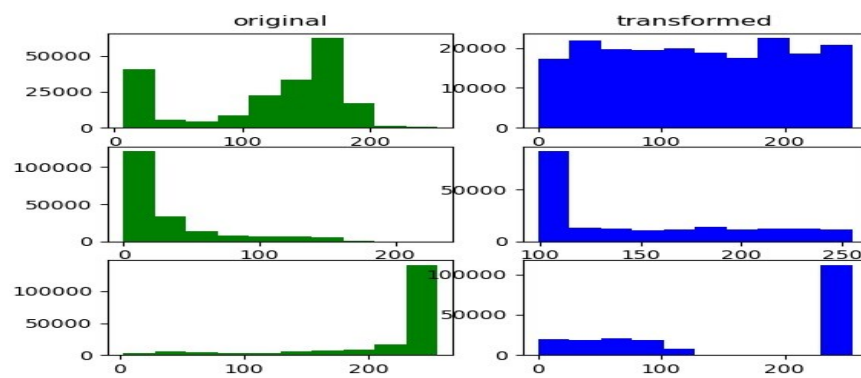


(2) histogram equalization

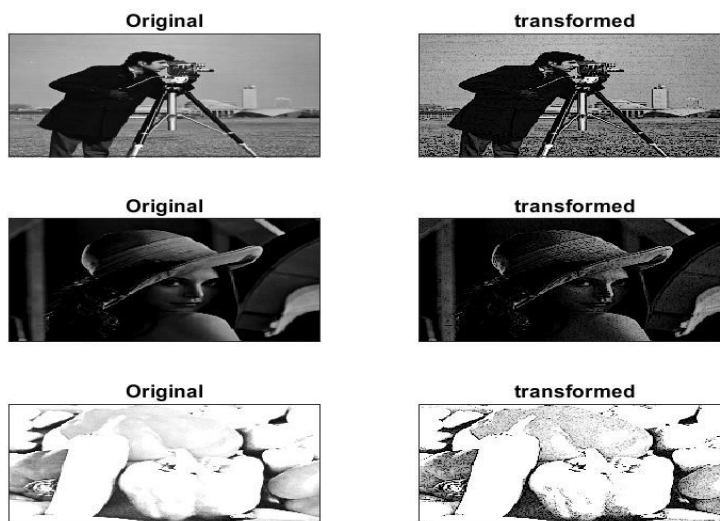
由此可看出經過轉換後和原圖相比變亮了。



此為 histogram

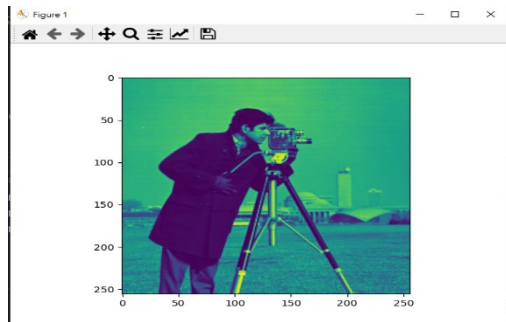


(3) image sharpening using the Laplacian.



Discussions

遇到問題:我用 python 或 matlab 讀取.bmp 檔時，會產生以下圖片:



導致變成不是原本的灰階圖，還是原本的圖片就不是灰階圖???

解決方法: 因為題目上也沒規定，所以我就把.bmp 檔轉為 .jpg 檔就 ok 了。

References and Appendix

<https://www.imageprocessing.com/2013/07/image-sharpening-using-second-order.html>