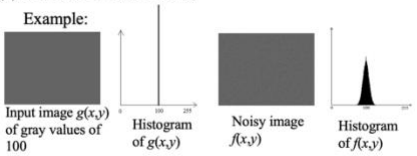
41047011S 資工 114 趙安庭

(1) Assignment statement

Homework 7 (Due: 5/2)

(1) Create an image $g(x,y)$ whose pixels all have the same gray value of 100. Show the image $g(x,y)$.
(2) Generate Gaussian noise $n(x,y)$, with $\mu = 0$, $\sigma^2 = 25$, using the algorithm shown in the next page.
Show the noisy image $f(x,y) = g(x,y) + n(x,y)$.
(3) Display the histogram $h(i)$ of $f(x,y)$.
(4) Comment on your results.

Example:



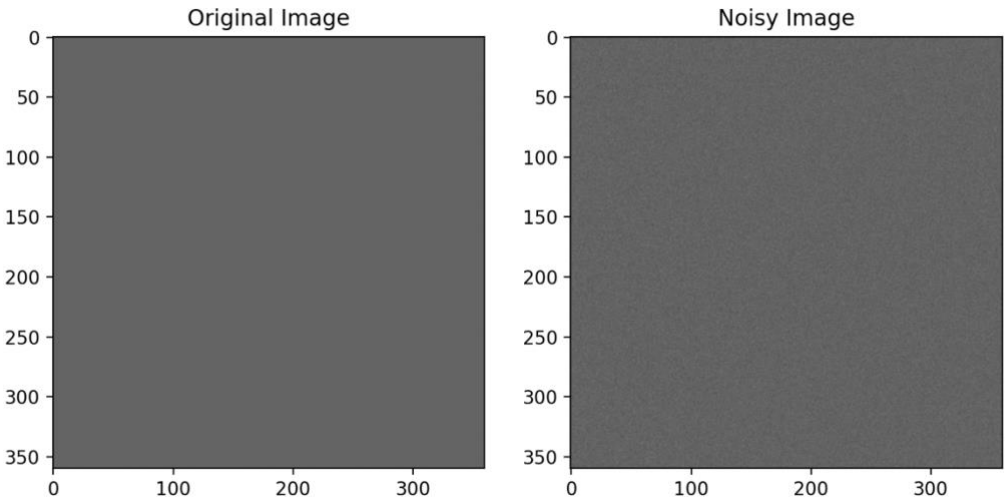
Input image $g(x,y)$ of gray values of 100 | Histogram of $g(x,y)$ | Noisy image $f(x,y)$ | Histogram of $f(x,y)$

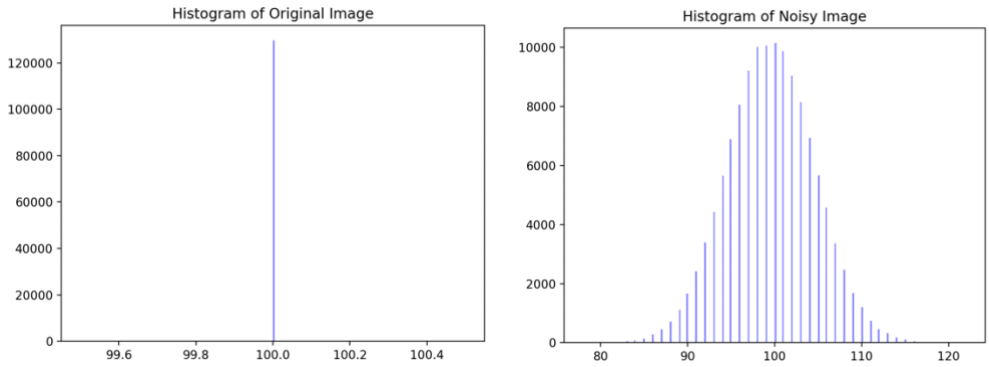
Algorithm : Generation of zero mean Gaussian noise

1. Suppose an image has gray-level range $[0, G-1]$. Select $\sigma > 0$;
2. For each pair of horizontally neighboring pixels (x,y), $(x,y+1)$ generate a pair of uniform random numbers r, ϕ in the range $[0,1]$.
3. Calculate $z_1 = \sigma\cos(2\pi\phi)\sqrt{-2\ln r}$, $z_2 = \sigma\sin(2\pi\phi)\sqrt{-2\ln r}$.
4. Set $f'(x,y) = g(x,y) + z_1$ and $f'(x,y+1) = g(x,y+1) + z_2$, where g is the input image.
5. Set
$$f(x,y) = \begin{cases} 0 & \text{if } f'(x,y) < 0, \\ G-1 & \text{if } f'(x,y) > G-1, \\ f'(x,y) & \text{otherwise,} \end{cases}$$
$$f(x,y+1) = \begin{cases} 0 & \text{if } f'(x,y+1) < 0, \\ G-1 & \text{if } f'(x,y+1) > G-1, \\ f'(x,y+1) & \text{otherwise.} \end{cases}$$
6. Go to 3 until all pixels have been scanned.

(2)

(a) Input/output images





(b) Source code

hw07.py

```python
import sys
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import skimage.io as io
import skimage.transform as trans

import random
import math

side = 360
img = np.zeros([side, side])
img = np.full((side, side), 100)
img2 = img.copy()

mu = 0
var = 5
Pi = math.pi
e = np.e

for x in range(0, side):
    for y in range(0, side, 2):
        g1 = img[x][y]
        g2 = img[x][y+1]
        r = random.random()
        q = random.random()
        z1 = var * math.cos(2*Pi*q) * ( (-2 * math.log(r) ) **0.5 )
        z2 = var * math.sin(2*Pi*q) * ( (-2 * math.log(r) ) **0.5 )
        ff1 = g1 + z1
        ff2 = g2 + z2
        f1 = ff1
        f2 = ff2
```

```
            if(ff1<0):
                f1 = 0
            elif(ff1>255):
                f1 = 255
            if(ff2<0):
                f2 = 0
            elif(ff2>255):
                f2 = 255
            img2[x][y] = f1
            img2[x][y+1] = f2

    plt.subplot(1, 2, 1)
    plt.imshow(img, cmap='gray', vmin=0, vmax=255)
    plt.title("Original Image")

    plt.subplot(1, 2, 2)
    plt.imshow(img2, cmap='gray', vmin=0, vmax=255)
    plt.title("Noisy Image")

    plt.show()

    # histogram

    plt.figure()
    plt.hist(img.flatten(), bins=256, color='blue', alpha=0.5)
    plt.title('Histogram of Original Image')

    plt.figure()
    plt.hist(img2.flatten(), bins=256, color='blue', alpha=0.5)
    plt.title('Histogram of Noisy Image')
    plt.show()
```

(c) Comments

這次作業因爲老師有給予步驟指引，實作時能明確知道要做什麼、怎麼做。原本我生成 img 時，pl.show() 顯示出的圖不是全灰的而是全黑的，就算我將 cmap 參數改成各種不同模式都無法解決，甚至有時候顯示出的圖會變紫色。後來發現是因為圖片的像素值範圍與 imshow() 預設的範圍沒有符合，因此我加上了 imshow() 中的 vmin=0 及 vmax=255，設定像素值範圍以後就能正常顯示 img 了。另外，原本我想讓原圖及噪音圖的 histogram 輸出圖的縱軸範

圍相同，縱軸比例相同比較好比較兩張圖片中的像素分布個數，但我發現將縱軸範圍拉太大會讓噪音圖的 histogram 看起來太平緩（如下圖），所以為了方便看出噪音圖的 histogram 分布形狀，我最後沒有特別設定縱軸範圍，讓 python 自行決定最適合該圖表的縱軸範圍。