Image Processing HW8+

0310120 陳家煒 電工系大三

Programing language: Python

OpenCV version (if needed): 3.2.0

Develop environment: Mac terminal

A. Program flow

先建構 1-D FFT function：

```
def fft(f):         # FFT for list type
    N = len(f)
    if N <= 1: return f
    even = fft(f[0::2])
    odd =    fft(f[1::2])
    return [even[k] + exp(-2j*pi*k/N)*odd[k] for k in xrange(N/2)] + \
                [even[k] - exp(-2j*pi*k/N)*odd[k] for k in xrange(N/2)]
```

建構 2-D FFT function：

1. 首先得先 2-D zero padding（為了實作 radix-2 FFT）

```
def pad2(f):        # 2D zero padding for list type
    m, n = len(f), len(f[0])
    M, N = 2 ** int(ceil(log(m, 2))), 2 ** int(ceil(log(n, 2)))
    F = [ [0]*N for _ in xrange(M) ]
    for i in range(0, m):
        for j in range(0, n):
            F[i][j] = f[i][j]
    return F, M, N
```

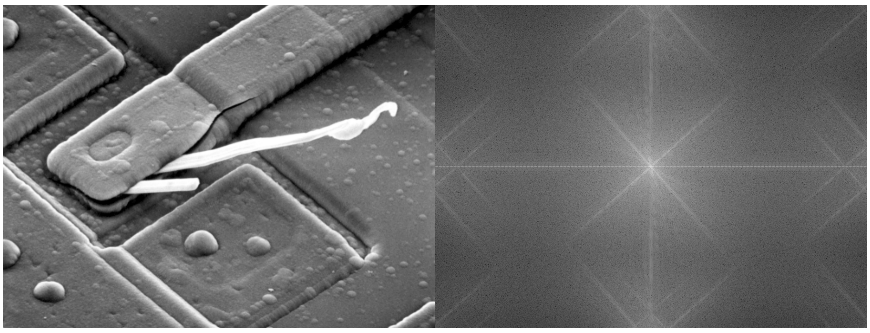
2. 進行二維運算、並使用 log(1+mag)且以灰階 Normalization

```
def FFT2D(image):
    width , height = image.shape
    image = image.tolist()
    image , M , N = pad2(image)             # zero padding for FFT
    image_temp = image                      # initial a temp_2Darray
    ########## execute row FFT ###########
    for i in range(M):
        image_temp[i] = fft(image[i])
    # transfer list to numpy.arr
    image_temp = numpy.asarray(image_temp,dtype = complex)
    image_temp = numpy.transpose(image_temp)          # transpose for further work
    image = image_temp.tolist()                        # transfer numpy.arr to list

    ########## execute column FFT ###########
    for i in range(N):
        image_temp[i] = fft(image[i])
    # transpose for correct position
    image_temp = numpy.transpose(image_temp)
    ########## transfer mag to log(1+mag) and normalize ###########
    magnitude = cv2.sqrt(image_temp.real**2.0 + image_temp.imag**2.0)
    log_spectrum = cv2.log(1.0 + magnitude)
    cv2.normalize(log_spectrum, log_spectrum, 0.0, 1.0, cv2.NORM_MINMAX)

    #######resize the Frequency padding image back to original size #######
    res = cv2.resize(log_spectrum,(height, width), interpolation = cv2.INTER_CUBIC)
    ###### exchange position of Frequency image due to periodic image ######
    fshift = numpy.fft.fftshift(res)

    return fshift
```

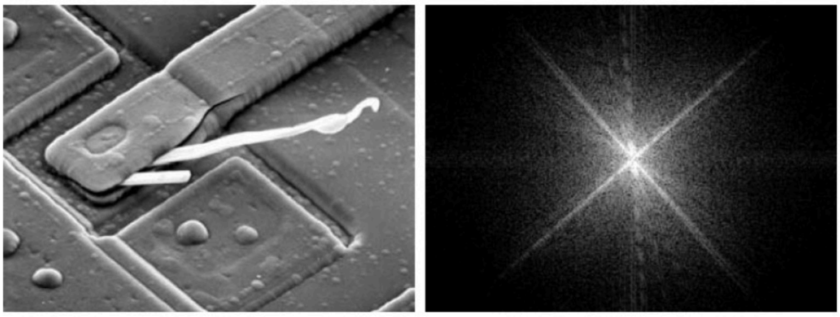
B. Result and Analysis

I. Fig. 4.29

a. Please attach your FT result



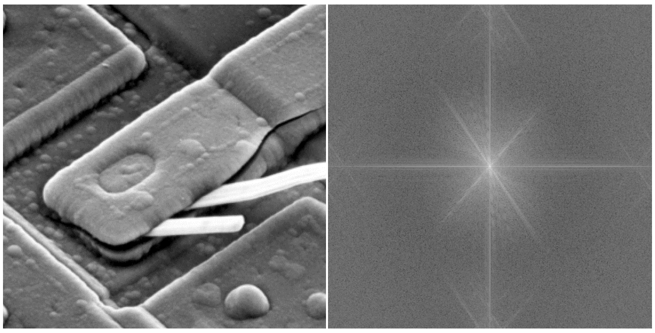
b. Please explain the difference, if any, between your result and the corresponding FT result in Fig. 4.29.



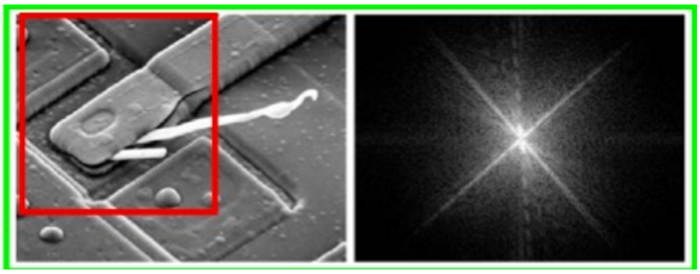
課本上的圖只有截取中間的部分，並且把他 resize 成原圖的大小 (906x678)，另外亮度上的差異可能是課本不只是以 10 為底取 log，而是更大的，又或是跑完 FFT 再做一次 histogram equalization 讓深淺更明顯。

II. Fig. 4.36

a. Please attach your FT result



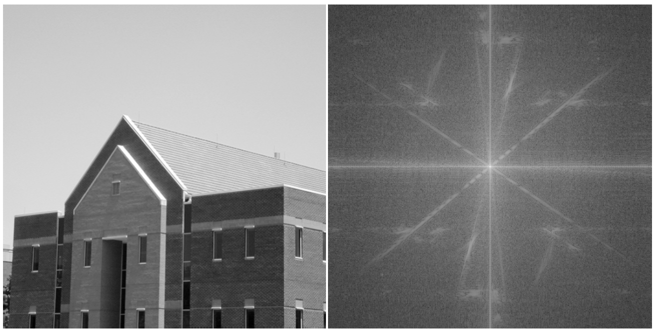
b. Please explain the difference, if any, between your result and the corresponding FT result in Fig. 4.36.



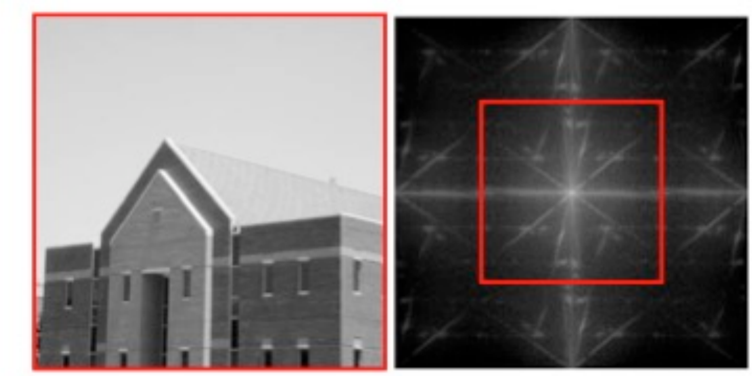
本圖的結果應該要是正方形的，如 a.（右）所示，深淺的問題如我 Q1 解釋的一樣。

III. Fig. 4.38

a. Please attach your FT result



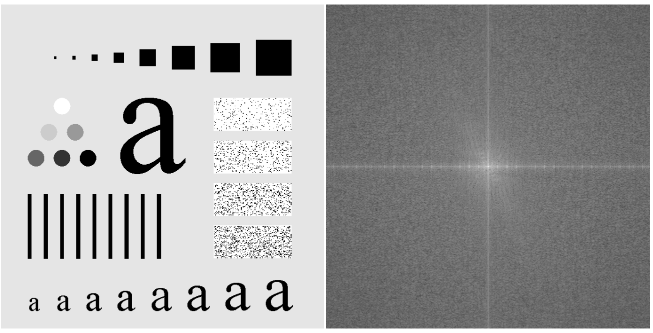
b. Please explain the difference, if any, between your result and the corresponding FT result in Fig. 4.38.



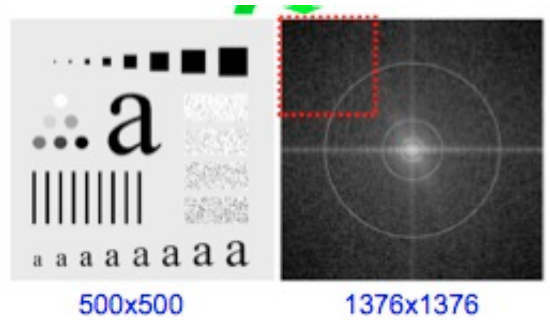
原圖為 600x600，而課本上的 FT 結果應該是給我們看 1200x1200，所以會有週期的效應出現，而我做的是單純一個週期 600x600 的結果。

IV. Fig. 4.41

a. Please attach your FT result



b. Please explain the difference, if any, between your result and the corresponding FT result in Fig. 4.41.



課本上做出來的 FT 圖與我跑出來的差異非常大，但我也用過官方的 fft 跑出來跟我的結果是一樣的，因此我有點懷疑同心圓的部份是不是哪邊出了問題，可能是我的顏色深淺區分不夠，讓結果顯示不明顯。