

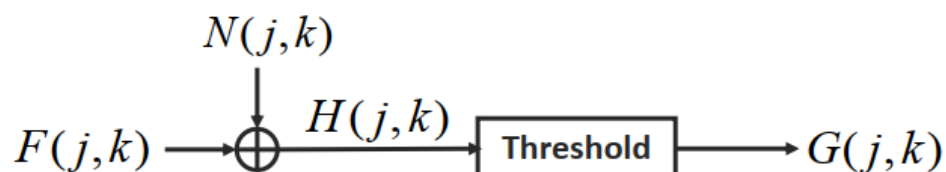
Problem 1 – Digital Halftoning

ORIGINAL



(a) Dithering by I_2

Dithering 的流程如下，主要為兩步驟 (1) Add noise (2) Thresholding。



首先，我以 Normal distribution $N(\mu = 0, \sigma = 5)$ 對原圖加 Gaussian noise，再利用 dither matrix $I_2(i, j)$ ，由下式建立 threshold matrix $T(i, j)$ 。

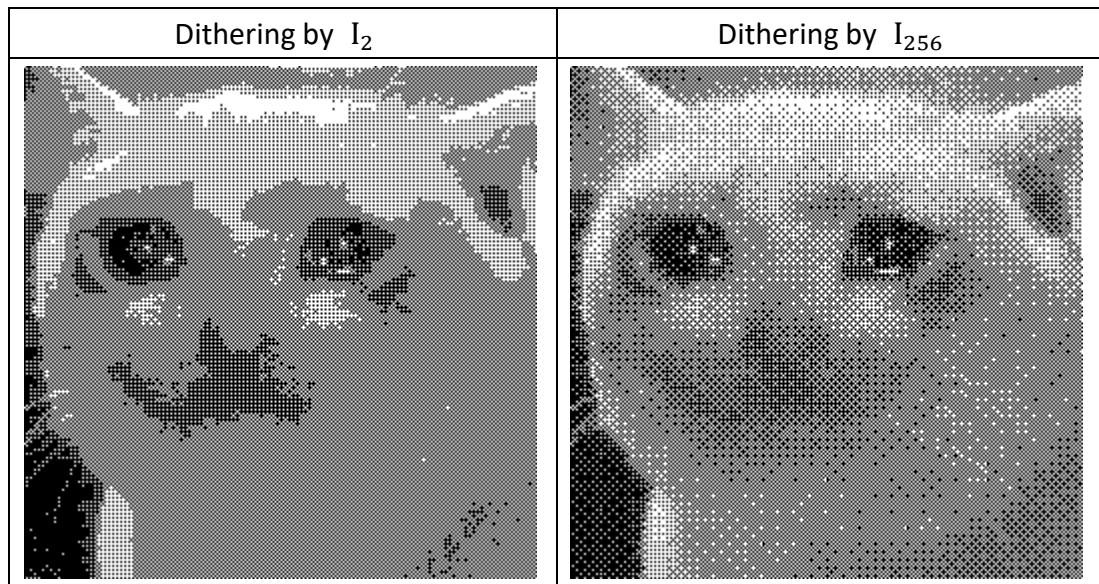
$$I_2(i, j) = \begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix} \quad T(i, j) = 255 \cdot \frac{I(i, j) + 0.5}{N^2}$$

對於原圖中，每個 pixel 座標 (i, j) 對應到 threshold matrix $(i \% 2, j \% 2)$ ，進行 thresholding，得到 half-toned image 如表。

(b) Dithering by I_{256}

將(a)小題使用的 I_2 以以下遞迴方式，擴展為 I_{256} 的 256×256 matrix。並一樣將其轉換為 threshold matrix 後，將原圖每個 pixels 對應到的 threshold matrix 進行 thresholding，得到 half-toned image 結果如表。

$$I_{2n}(i, j) = \begin{bmatrix} 4I_n(i, j) + 1 & 4I_n(i, j) + 2 \\ 4I_n(i, j) + 3 & 4I_n(i, j) + 0 \end{bmatrix}$$

RESULT :

由以上兩種不同大小 dither matrix 做出來的 half-toned image 結果有很大的不同。

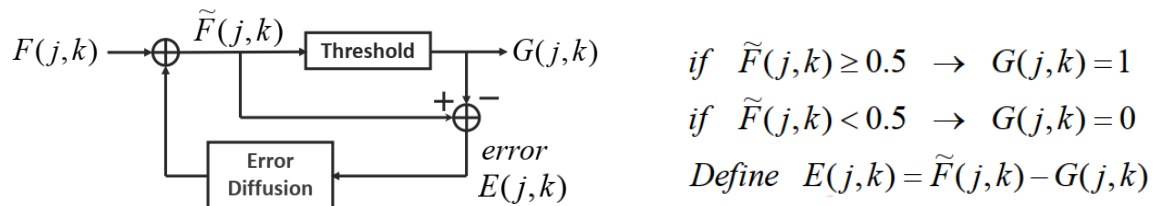
I_2 做出來的結果中，僅以幾個特定的 patterns 填滿，並沒有太多變化，且在深淺交界處沒有像 I_{256} 那樣有融合，雖然有真實呈現原圖深淺差異，但這樣區塊性的劃分還是稍顯呆板，因為在真實照片中，色彩、明暗變化應是連續且相互融合。導致這樣結果是因為整張圖用固定的 2×2 threshold matrix 進行 threshold，因此只會 $2^4 = 16$ 種 patterns，相似深度的地方就會以單一方式呈現。

I_{256} 是由 I_2 以特定規則遞迴運算得到的 dither matrix，打破規律性，使得每個 pixel 位子對應到的 threshold 有較大的變化，所以繪製出來的 half-toned image 就不會區塊分明，而是一些黑、白點以不規則形式相互交織融合出較接近真實影像的效果。

(c) Error diffusion

以 Error diffusion 產生 half-toned image 的作法是：先將原圖 $F(i,j)$ Normalized 到 $[0, 1]$ 產生 $\tilde{F}(i,j)$ ，再以 $\text{threshold} = 0.5$ 對 $\tilde{F}(i,j)$ 進行 thresholding 產生 output $G(i,j)$ 。計算該 output 和 threshold 前 $\tilde{F}(i,j)$ 的差異，定義 error $E(i,j) = \tilde{F}(i,j) - G(i,j)$ ，並以特定擴散方式將該 error 大小分佈在該 (i,j) 點後尚未以 threshold 處理的 \tilde{F} 上，即為 error diffusion。

整個流程的 block diagram 及公式如下：



上述實作擴散方式的 error diffusion 方法為使用 error diffusion masks，有兩種：Floyd Steinberg 及 Jarvis，下表為該兩種 error diffusion masks 及產生的結果。

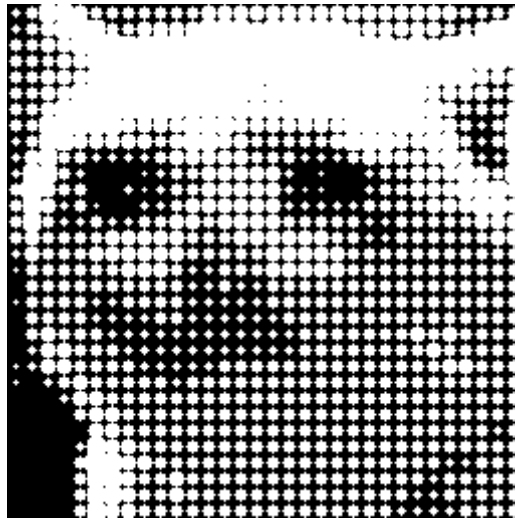
RESULT:

Floyd Steinberg (1975)	Jarvis et al (1976)
<p>error</p>	<p>error</p>

看出以 Jarvis 具有範圍較大的 error diffusion，對於鄰近點、較遠點的 weighting 也比較明顯的差異，因此該 error diffusion mask 可以做到比較接近原圖的 half-toned image。

(d) Dotted halftoning (Manga style)

這題的作法我將原圖劃分為很多 8×8 小區塊，並以該區塊為單位，計算區塊內平均 pixel 強度，以其平均強度 $\times 7.5/255$ 作為半徑在 output image 的該區塊中心位子畫圓，若平均 pixel 強度小於 50，則不畫任何東西，保留黑色，得到結果如下。

RESULT:

Problem 2 – Frequency Domain

ORIGINAL



(a) Fourier Transform

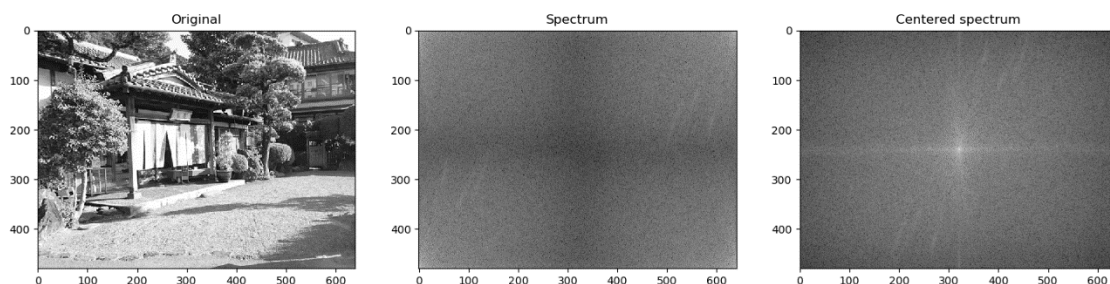
對原圖做 2D Fourier Transform 得到 frequency domain 的 2D spectrum。

$$F(u, v) = \frac{1}{MN} \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} f(j, k) e^{-j2\pi(\frac{uj}{M} + \frac{vk}{N})}$$

$$\text{Fourier Spectrum: } |F(u, v)| = [R^2(u, v) + I^2(u, v)]^{1/2}$$

由共軛性質，我們知道該 spectrum 會有對稱關係，因此將 $F(0, 0)$ 作為 spectrum 新的中心位子，並且根據該中心平移整個 spectrum 得到 centered spectrum。（以 log scale 繪製以利觀察）

$$\mathcal{F}[f(j, k)(-1)^{j+k}] = F(u - M/2, v - N/2)$$

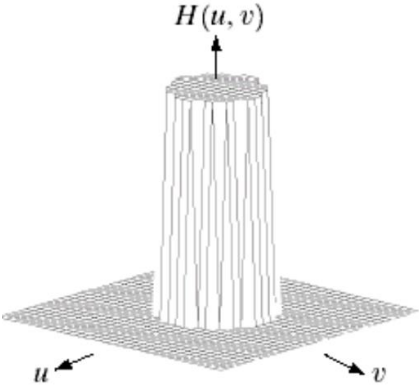
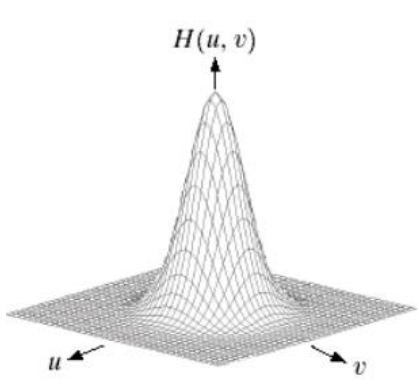
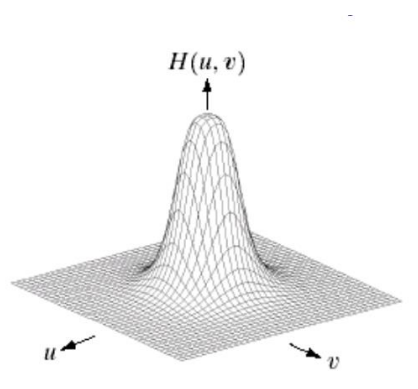
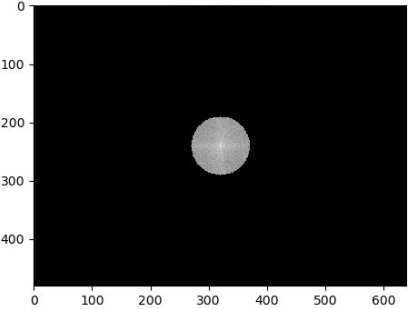
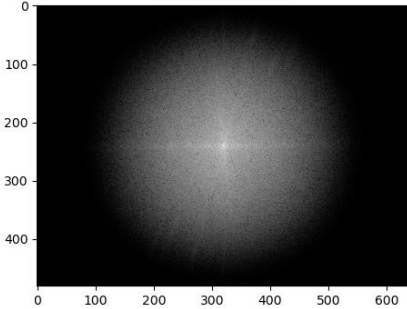
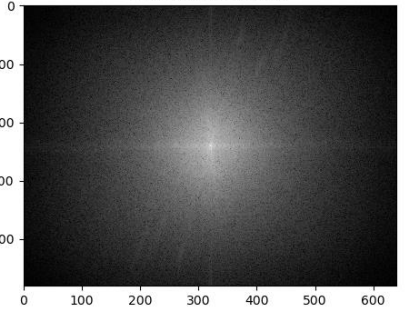
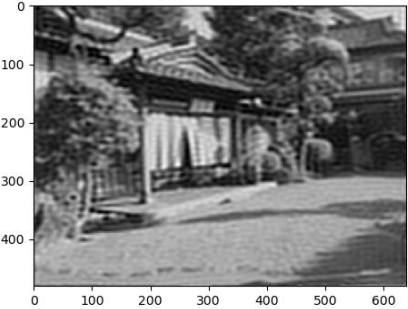
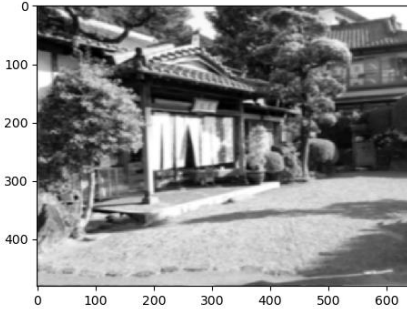
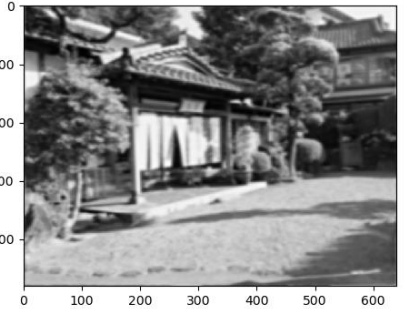


(b) Low Pass Filter

我使用三種 Filters 實作在 frequency domain 上的 low pass filter：(1) Ideal LP
(2) Gaussian LP (3) Butterworth LP。各自的公式及結果如下：

D 為 pixel 距離中心的位子 $D(u, v) = \sqrt{(u-M/2)^2 + (v-N/2)^2}$

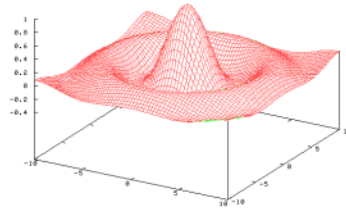
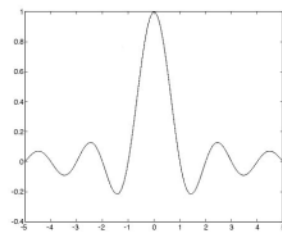
D_0 為 cutoff frequency 的半徑。(下表代 $D_0 = 50$)

Ideal LP	Gaussian LP	Butterworth LP (n=2)
$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) \leq D_0 \\ 0 & \text{if } D(u, v) > D_0 \end{cases}$	$H(u, v) = e^{-D^2(u, v)/2D_0^2}$	$H(u, v) = \frac{1}{1 + [D(u, v)/D_0]^{2n}}$
		
		
		



比較 frequency domain low-pass filter 和 pixel domain low-pass filter 的效果如下表，其中，frequency domain 我使用 Ideal LP filter 做為代表，pixel domain 則使用下面那個 low-pass filter matrix 對所有 pixels 進行 convolution。

$$H = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

為使得兩張模糊化程度相近，我調整 $D_0 = 110$ ，但是仔細看會發現圖中有些鬼影，即為漣漪(ripples)現象。原因是 Ideal LP 所使用 frequency domain 的方波，以 inverse fourier transform 轉換回 spatial domain 後會是一個 $\text{sinh}()$ function，而 $\text{sinh}()$ function 在橫軸上除了我們想要的中央 main lobe 以外，還會有許多小小震盪的 side lobes，如下示意圖，此即為漣漪現象。要避免這個漣漪現象，也可以改用 Gaussian LP filter、或 n 比較小的(≤ 2 為佳) Butterworth LP filter，如上頁表中結果。



RESULT:

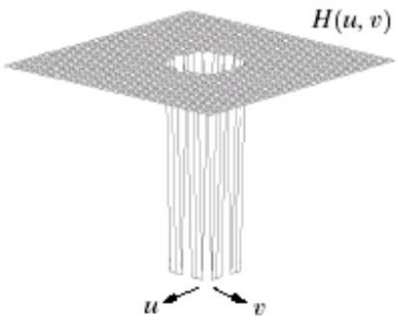
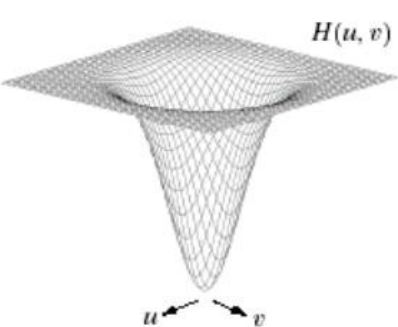
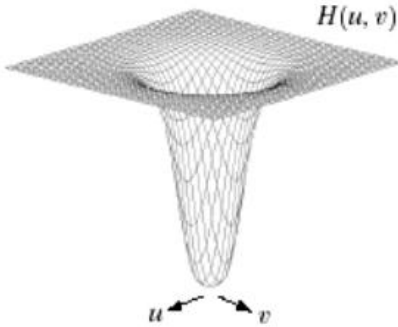
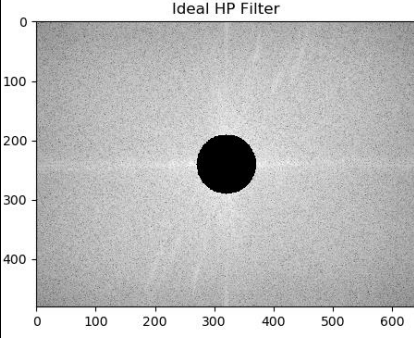
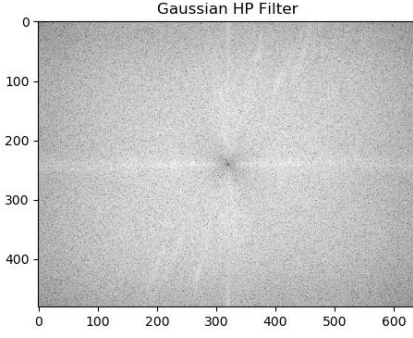
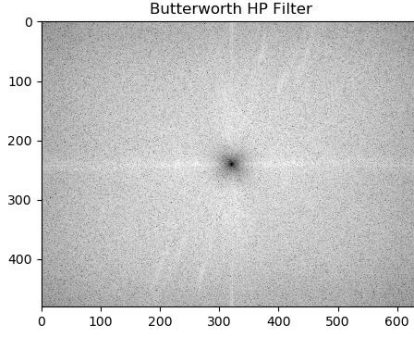
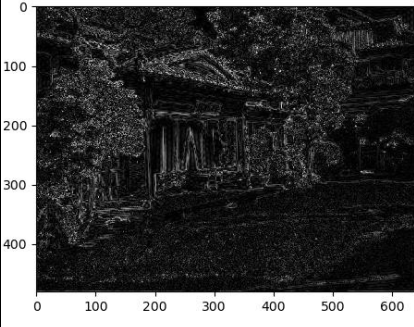
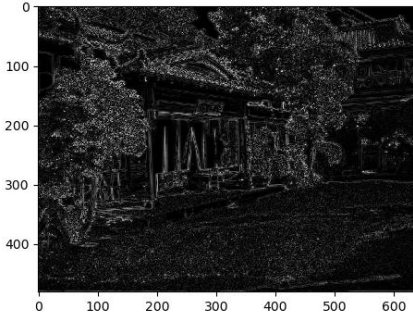
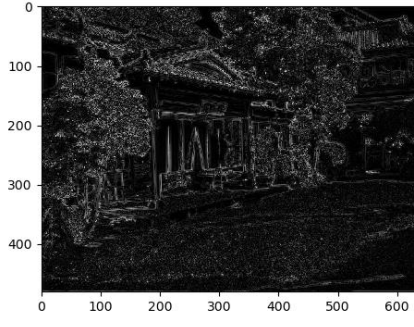
frequency domain low pass(ideal filter)	pixel domain low pass
	

(c) High Pass Filter

同 low pass filter 作法，使用三種 Filters 實作在 frequency domain 上的 high pass filter：(1) Ideal LP (2) Gaussian LP (3) Butterworth LP。各自的公式及結果如下：

D 為 pixel 距離中心的位子 $D(u, v) = \sqrt{(u - M/2)^2 + (v - N/2)^2}$

D_0 為 cutoff frequency 的半徑。(下表代 $D_0 = 50$)



Ideal HP	Gaussian HP	Butterworth HP (n = 2)
$H(u, v) = \begin{cases} 0 & \text{if } D(u, v) \leq D_0 \\ 1 & \text{if } D(u, v) > D_0 \end{cases}$	$H(u, v) = 1 - e^{-D^2(u, v) / 2D_0^2}$	$H(u, v) = \frac{1}{1 + [D_0 / D(u, v)]^{2n}}$
		
		
		

比較 frequency domain high-pass filter 和 pixel domain 的效果如下表，其中，frequency domain 使用 Ideal HP filter(D_0 取 110)做為代表，pixel domain 則使用下面那個 high-pass filter matrix 對所有 pixels 進行 convolution。

$$H = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

High pass filter 保留原圖中 high frequency 的變化，有標記變化不連續處的效果，因此所有 edge 都會突顯出來。然而這裡我認為 pixel domain 的 high pass 比 frequency domain 好，因為以 Ideal HP filter 做 frequency domain 中漣漪的現象太強，其發生原因和(b)小題中 low-pass filter 描述的一樣，使輸出結果有些模糊不清，但如前頁表格下使用沒有漣漪現象的 Gaussian 比較好。

RESULT:

frequency domain high pass(ideal filter)	pixel domain high pass
	

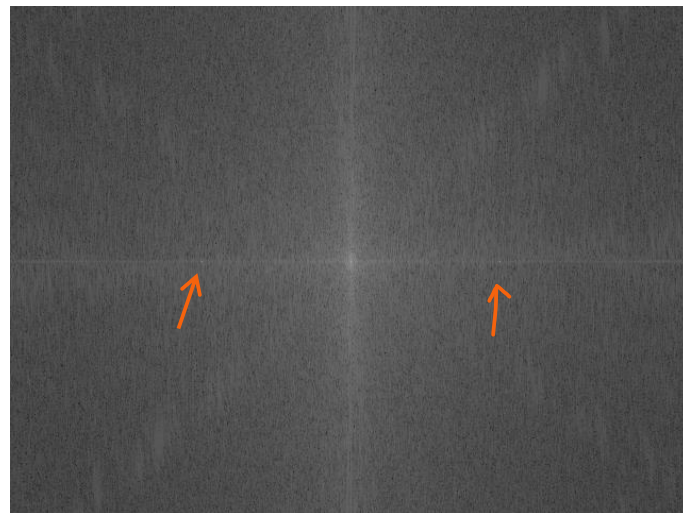
(d)(e) Case Study

ORIGINAL:



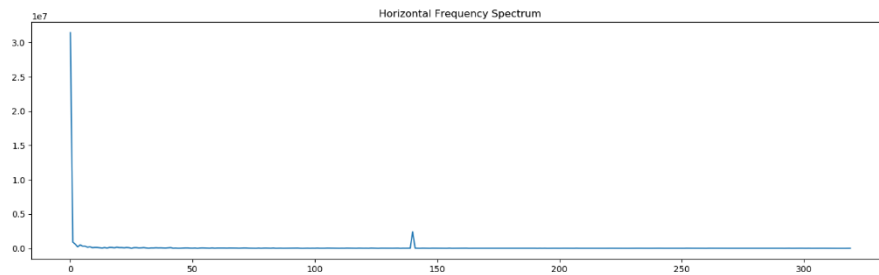
原圖是一張海港的圖，但影像中有一條條細小的垂直線佈滿整張圖，我們的目標是移除該雜訊。由於線條雜訊有週期性，因此我使用 **frequency domain** 的方法來做。

首先，以(a)小題詳述的流程對它做 **Fourier Transform**，結果如下：



觀察 **frequency spectrum** 會發現在橫軸座標上，有兩個明顯亮的白點（兩點對縱軸對稱，這是 **conjugate** 性質導致，事實上他們代表的是同一個頻率的訊號）。而這兩個白點應就是代表原圖中的雜訊，他們以該頻率水平散佈在整張圖中，只要將其移除就可以消掉原圖的雜訊了。

繪製橫軸上原點到最大值 frequency 的 histogram 以找到移除該雜訊的 threshold。如下，可觀察到 140 附近確實有個強度異常高的值。



設定 $\text{threshold} = 500000$ ，避開本身訊號量較強的小頻段區域，其餘部分找到高於 threshold 異常值，設為零。再將 frequency spectrum 轉回 spatial domain 上，結果如下，可以該雜訊確實被消除。

RESULT:

