**影像處理LAB3**

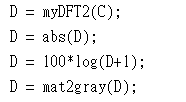
學號:103062129 姓名 :李國緯

**4-1 Two-Dimensional Fast Fourier Transform**

**做法說明**

有了圖片的Size之後，要對原始圖片做padding，

padarray 是內建的function，這邊的長寬是為原圖的兩倍，得(b)。 再來要做center shifting，這裡我實現的方法是，用double for loop，並用一個matrix儲存每個(-1)^(x+y)，最後再跟原始圖片做array multiplication( in matlab .\* )得到圖片(c)。Spatial Domain的處理在此已經完成，將圖片用myDFT2 function做2D的Fourier Transform，轉換到Fourier Domain。因為轉換完後，array type 會是 complex single (double)，為了展示spectrum，先利用abs取得real的部分，然後再用log transform做強化，以利顯示。

此時，圖片的值會超出1，而非0~1。

因此我再用mat2gray，將值重新map到0與1之間。

得到圖片(d)。

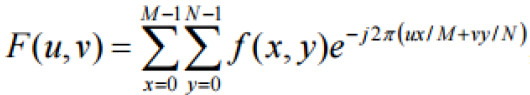
利用4-3的myGLPF，即Gaussian Lowpass Filtering，D0 = 30，得(e)。

將specturm(為myDFT2直接做出來的，不是經過abs與log過後的)，與Gaussian Lopass Filter 做array multiplication(.\*)運算，得(f)。

此時Fourier已經完成，轉回spatial domain，用myIDFT2，再將中心移回得到(g)。

將padding去除之後可得filtered output，為圖(h)。

myDFT2 : 基本的做法就是依照公式解。



我將x (u), y (v) 分開做 exponential 最後再利用matrix multiplication (\*) 將x , input, y 相乘，得到Output。

其中，exp()，是內建的函數，虛數的部分我用1i表示(matlab建議)。

分開算第一個是可以避免看起來複雜的4層for loop，再者可以利用matlab的矩陣相乘，而非傳統的點對點矩陣相乘。

myIDFT2: 幾乎與myDFT2相同，唯二的差別在於exponential 係數少了一個負號，

還有最後除以 (MxN)。

**結果圖片**

|  |  |  |
| --- | --- | --- |
| 4.36(a) Original | **C:\Users\WorkingUse\AppData\Local\Microsoft\Windows\INetCache\Content.Word\4.36(b).tif**  4.36(b) padding | 4.36(c) shift center |
|  | 4.36(d) Spectrum | 4.36(e)Gaussian lowpass Filtering |
| 4.36(f) filtered in fourier domain | 4.36(g) back to spatial domain | 4.36(h) remove padding |

**分析以及討論**

觀察Fourier Transform 我們可以發現運算結果具有週期性，也就是一切的根本，或說frequency。這是個十分常用的方法，美中不足的部分就在於，速度。理論上而言，DFT的時間是O(n^4)，在實作上是O(n^2)加上係數(In matlab)。因此有了所謂的Fast Fourier Transform (FFT)，速度為O(nlgn)。實作的方法理論上為divide- and-conquer (recursive)，將problem size 縮小，然後可以得到O(nlgn)。

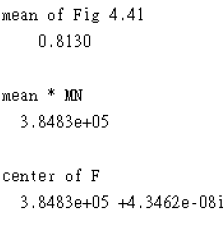
**4-2 Fourier Spectrum and Average Value**

**結果圖片**

|  |  |
| --- | --- |
|  |  |

**分析以及討論**

Spectrum的原點為(0,0)，帶入公式會得到exponetial ^ 0 = 1。因此這個公式就會變成，把原圖的每一點加起來(sum)。而mean = sum / image\_size，所以sum就會和mean差MN倍。Mean乘上image size(MN) = centered image on Fourier domain 的Real part(Spectrum)。



**4-3 Lowpass Filtering**

**結果圖片**

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

由上至下，左至右，分別為，

原圖 ，D0 = 10 ， D0 = 30

D0 = 60 ，D0 = 160， D0 = 460

**分析以及討論**

Lowpass Filter 顧名思義允許低頻的訊號通過，把高於一定頻率的訊號變成零。隨著D0的變大，能通過原始filter的頻率就越多，模糊的效果就會越弱，如上圖顯示。在一般的圖片上做，也就是spatial domain，會是週期性函數函數，實作上十分不容易，但如果是在fourier domain上做，經過公式的幫忙，就會是簡單的array multiplication( .\*)運算即可達成。也因此在實作上才會將圖片先經4-1的步驟，轉至Fourier domain。

GLPF能將圖片變模糊(當D0不大時)，與一般的blur(average filter)差異性在於 :

Average blur 是很粗糙的將圖片blur (平均)，會產生非常多雜訊，看起來甚至會有一格一格的感覺，而GLPF處理blur上的效果較為理想。

**4-4 Highpass Filtering**

**結果圖片**

|  |  |  |
| --- | --- | --- |
|  |  |  |

D0 = 30 D0 = 60 D0 = 160

**分析以及討論**

相反於GLPF，GHPF就是允許高頻率的訊號通過，將低於一定頻率的訊號eliminate。

當D0越大，能通過的頻率就越多，效果越弱。

相反於GLPF的blur，GHPF就是sharp，由上圖可得知，edge被強化了，當D0越大，效果越不明顯。

同樣是銳利化，相似的還有unsharpFiltering(Laplacian)。

一個是直接對image做filter，另一個是在fourier domain做文章。

我認為原理上，unsharp效果不會比GHPF來的好，因為Laplacian會受周遭的影響多，在一些較為複雜的edge上比較難detection。而GHPF是根據頻率上做調整，因此在整體的感覺上會較佳。

但實際的效果上，經測試(月球表面)，肉眼看起來差不多。

****

左圖為GHPF。

右圖為Laplacian Mask。