

Implement Edge Detector

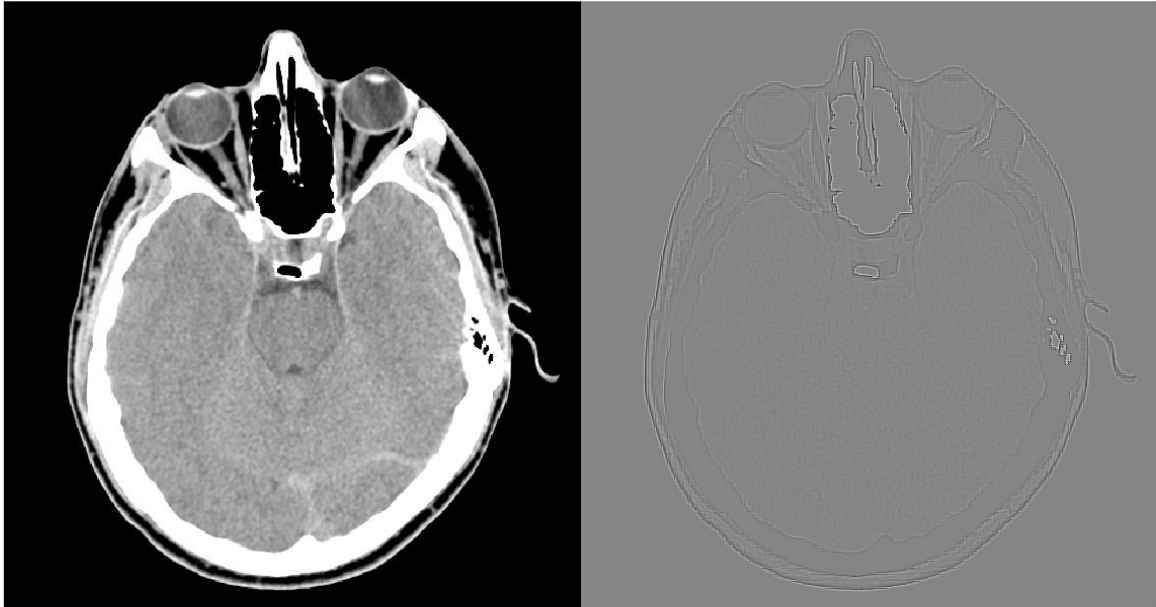
資工四乙
405262518
謝承佑

程式說明

Gradient_magnitude 函式有兩個參數：第一個是要丟入的灰階影像；第二個是在計算矩陣相乘的時候要不要取絕對值，預設為 true，因為後續 Zero_crossing 時會使用到不取絕對值的影像。

該函式使用的 mask 是取圖片的 N, E, S, W 來加總而成，所以運算時間會比一般取水平加垂直的還要久一點。

下圖為 **Gradient_magnitude('headCT.tif', false)** 的結果：



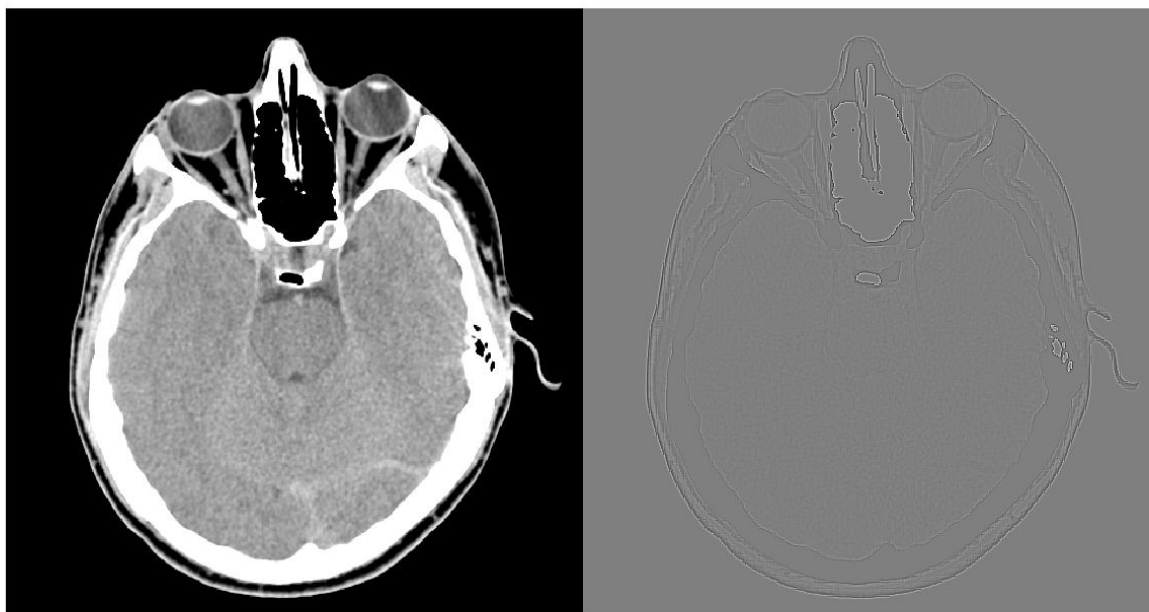
下圖為 **Gradient_magnitude('lena.tif', true)** 的結果：



Laplacian_mask 函式有四個參數：第一個是要丟入的灰階影像；第二個是要以幾鄰居的方式來做，預設為 4 鄰居；第三個也是在取絕對值，預設為 true；第四個是要不要用已經給定的 mask 來做，預設為 “mask”。第四個參數可使用 “normal”，意思是直接算每個 Pixel 的二階導數。

自己在算每個 Pixel 的二階導數 (normal) 出來的結果還比不上直接套用 mask 來得好，不太懂為什麼。

下圖為 **Laplacian_mask('headCT.tif', 8, false)** 的結果：



下圖為 **Laplacian_mask('lena.tif', 4, true)** 的結果：



zero_crossing 函式有三個參數：第一個是要丟入的灰階影像，第二個是想使用的方法，預設為 “Gradient”，其方法就是上面兩個已寫好的函式，第三個則是在運算時的 Threshold，預設為最大數字乘上 0.1。

lena 出來的結果比較差一點，我在想是不是我在寫 zero-crossing 演算法的時候有問題，也有可能是因為 lena 的雜訊太多，因為我這裡取 zero-crossing 之前，是使用上面兩個函式出來結果下去做，所以要是上面兩個函式的 Binary 影像太多黑白相間的情形，在這裏就會變得很多雜訊。

下圖為 **zero_crossing('headCT.tif', "Gradient")** 的結果：



下圖為 **zero_crossing('lena.tif', "Laplacian")** 的結果：



LoG 函式有四個參數，第一個是要丟入的灰階影像；第二個是 filter 的大小，為 $n \times n$ ；第三個是 sigma 的大小，用來帶入公式算出 filter 的值；第四個則是 threshold，預設為 filter 後的影像的最大數字乘上 0.1。

這函式嘗試實作出 Laplacian of Gaussian，先將圖片高斯模糊化，再用 Laplacian mask，最後再取 zero-crossing。用 $\text{sigma} = 1$ 的方式會讓雜訊減少很多，比前面直接用 Laplacian 做 zero-crossing 效果還要好，但是 sigma 和 threshold 的參數很難調整，如果 sigma 取太大，會讓每個點的數值放大太多，就會變更多雜訊，所以如果 sigma 大於等於 2 那麼 threshold 也要調大才能濾得更好。

下圖為 **LoG('lena.tif', 3, 1)** 的結果，效果比直接取 Laplacian 還要好：



下圖為 **LoG('headCT.tif', 3, 2, 100)** 的結果，對於雜訊本來就不是很多的圖，使用 LoG 好像會讓他效果變得比較不佳一點，所以這裡調整了 sigma 和 threshold：

心得

對於這次的作業，其實 MATLAB 上面有很多已經寫好的 Toolbox，而且效果非常好，但我還是把所有東西都自己實作看看，雖然效果很差，也有可能想錯或是寫錯的地方，但是至少有比較了解實際上要做邊界偵測時會遇到的問題和狀況。