

程式設計第三次作業

Due: 2020/4/25 23:00

※注意事項：請依照課程網站內所公告之“作業檔案命名規則與規定”進行作業檔案命名以及繳交作業，未依照規定將斟酌扣分。

本次作業共有一題，為程式撰寫。

影像處理上常利用一個固定的遮罩或稱運算子(operator)對影像進行卷積(convolution)運算，進而將影像邊緣強化、影像銳化或是模糊化等效果。一個簡單的卷積運算如下：

$$\text{假設有一 } 4 \times 4 \text{ 影像矩陣 } A = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix}, \text{ 及 } 3 \times 3 \text{ 運算子 } S = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \text{ 則在 } A \text{ 陣列位置}$$

(2,2)的卷積值：

$$(S * A)(2,2) = (a \times 9) + (b \times 8) + (c \times 7) + (e \times 6) + (f \times 5) + (g \times 4) + (i \times 3) + (j \times 2) + (k \times 1)$$

因此，要取得一張灰階影像經遮罩進行卷積的新影像，必須將遮罩中心放置在影像上的每個位置並分別進行卷積運算得到新的灰階值。然而，在圖形邊界會有遮罩超出邊界的問題(圖 1)，常見的措施有邊界補 0 或是不計算，而本次作業請將邊界補 0 進行運算。

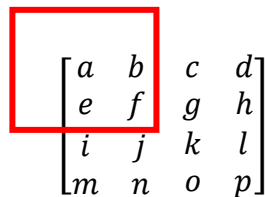


圖 1 遮罩超出邊界示意圖

索伯運算子(Sobel operator)是影像處理中常用的運算子之一，常被用來對影像進行邊緣檢測。此運算子包含兩組 3×3 的矩陣，分別為橫向運算子 S_x 及縱向運算子 S_y ，將之與影像進行平面卷積。其中兩運算子的定義如下：

$$S_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

若將影像 A 進行索伯邊緣檢測則會先分別計算橫向 G_x 及縱向 G_y 的邊緣檢測影像，再將每個像素之橫縱向值結合得到邊緣檢測後的新影像 G ，公式如下：

$$G_x = S_x * A, G_y = S_y * A, G = \sqrt{G_x^2 + G_y^2}$$

- 作業題目：

利用動態配置一個二維 double 矩陣(其中矩陣大小為 $m \times n$ 矩陣， m, n 可能不相等且介於 4 到 10 間，請分別利用亂數產生)，兩矩陣內的元素亦利用亂數產生(在 0 到 255 之間的整數)。請計算並輸出以下內容：

1. 原始矩陣 A
2. 矩陣與 S_x 進行卷積的結果 G_x
3. 矩陣與 S_y 進行卷積的結果 G_y
4. 矩陣進行索伯檢測的結果 G
5. 分別輸出上述 4 個矩陣之最大元素及其下標，下標由 1 開始標記，Ex:最大值在[2][2]的位置，則應顯示為(3,3)，如果數值一樣則必須將所有位置均列出

請在陣列使用完畢後歸還使用空間。**所有程式碼均須在主程式內撰寫，請先勿使用函數！**

※請勿使用標準樣板函式庫與額外的巨集指令※