R07922106 曾俊為 CV_HW4 Report

Binary Morphology

在做出以下 5 個效果前，先將 lena 的原始圖檔讀進來，用 128 當 threshold 將圖片轉為 binary 的圖。再來先將 octogonal 3-5-5-5-3 kernel 和 L-shaped kernel 簡單地用陣列刻出來。

(a) Dilation

這裡實作 dilation 的部分，是使用迴圈一個一個 pixel 做，只要此 pixel 為白點，接下來依序由 35553 kernel 的 element 做出擴張的效果，只要沒有超出圖片的範圍，都視為可擴張的區域。

```
if img_b[i][j] == 255:
    for element in kernel:
        x, y = element
        if i+x >= 0 and i+x <= h-1 and j+y >= 0 and j+y <= w-1:
            res_1[i+x][j+y] = 255
```

(b) Erosion

再來 erosion 的部分，則和 dilation 的作法大同小異，只差在後續演算法的不同，同樣先用迴圈一個一個 pixel 做，接下來對 binary 圖的白點做動作，也是依序由 35553 kernel 的 element 做判定，只要中途發現移動過後，會觸碰到 pixel value 為 0 的點，則直接跳出判定的迴圈，進行下一個 pixel 的判定；但是如果在中途碰到的都是白點且不超出圖片範圍，則予以保留。最後做完可得出上圖的結果。

下圖是演算法的部分:

```
for element in kernel:
    x, y = element
    if i+x < 0 or i+x > h-1 or j+y < 0 or j+y > w-1 or img_b[i+x][j+y] ==
    0:
        res_2[i][j] = 0
        break
    if True :
        res_2[i][j] = 255
```

(c) Opening

再來 opening 的部分則是遵照公式，binary image 對 35553 kernel 先做 erosion 的動作，再做 dilation 的部分，可得出上圖效果

(d)Closing

Closing 的部分，跟上述類似，也是遵照公式，原始的 binary image 對 kernel 先做 dilation 再做 erosion，則可得出上圖結果。

(e)Hit-and-miss transform

而最後 hit-and-miss 的部分，這邊使用的為課本的兩個 L 型 kernel，分別為 J_kernel，K_kernel，接下來遵照公式找出 binary image 對 J_kernel 做 erosion 和 binary image 的補集(也就是黑白顛倒) 對 K_kernel 做 erosion 兩張實驗圖的交集，就是要找出的部分。

下圖是找出補集的部分:

```
img_rev = -img_b + 255
```

下圖為找出交集的部分:

```
img_J = erosion(img_b, J_kernel)
img_K = erosion(img_rev, K_kernel)
for i in range(h):
    for j in range(w):
        if img_J[i][j]  == 255:
            if img_K[i][j] == 255:
                res_ham[i][j] == 255
```

但比較可惜的是，我做不出課程網站上的範例結果，結果為幾乎全黑，也就是找不太到兩張實驗圖交集的部分。