影像處理 - 去除 impluse 雜訊

5105056017 黃凱鴻

簡述

高斯雜訊與 impluse 雜訊是最常見的兩種雜訊，可以透過 mean filter 與 median filter 去除。而這兩種 filter 可以透過簡單的判斷，使得去雜訊的結果更加優秀。

本次作業將實作 Adaptive Median Filter，其原理是透過中間值(in a NxN matrix)與附近極值的比較，決定要輸出原值或中間值。

實作說明

(原始圖片：source.jpg)

增加雜訊

對原圖隨機增加雜訊，其機率為

- 0 : 25%
- 255 : 25%
- 原始值 : 50%

```
## add impluse noise if the source is pure
def addNoise(img,implusePercent=0.25):
    output = Image.new("L",(img.width,img.height))
    outputPixels = output.load()
    pixels = img.load()

    for j in range(img.height):
        for i in range(img.width):
            ran = random.random()
            if ran < implusePercent:
                outputPixels[i,j] = 0
            elif ran < 2*implusePercent:
                outputPixels[i,j] = 255
            else:
                outputPixels[i,j] = pixels[i,j]

    return output
```

於是可以得到一張充滿雜訊的圖片如下

(充滿50%雜訊的圖片：implused.jpg)

實現 AdaptiveMedianFilter 演算法

```
def adaptiveMedianFilter(img,x,y,size=3,maxSize = 7):
    Zmin,Zmed,Zmax,Zxy = getTags(img,x,y,size)
    #print( Zmin,Zmed,Zmax,Zxy,size)

    if Zmin < Zmed < Zmax:
        # level B
        if Zmin < Zxy < Zmax:
            return Zxy
        else:
            return Zmed
    else:
        size +=2

    if size <= maxSize:
        return adaptiveMedianFilter(img,x,y,size)
    else:
        ## med or xy?
        return Zmed
        #return Zxy
```

在教材上分成 a, b 兩個 part，但實作時將其合併。另外在最下面一段，雖然教材是說要回傳 Zxy，但是在 0 or 255 的平坦區，會得到錯誤的值，所以改成 Zmed。

結果比較

(使用本作業演算法得到的結果 : result-adaptive.jpg)

(使用簡易 median filter 的結果 : result-median.jpg)

可發現在大量雜訊下，使用原始的 median filter 依然會存在許多雜訊。然而使用本演算法得到的圖片，無法去除的雜訊已相當稀少。

當然，若是與原圖仔細比較，會發現去除雜訊的圖片，其實也是崎嶇不平。這是因為大量雜訊過大(50%)，若是將雜訊量減少，這種現象也會隨著減輕。