

## DIP Homework Assignment #2

R06922097 資工碩一 鄭雅文

### PROBLEM 1: EDGE DETECTION

(a)

1<sup>st</sup> order edge detection

1. Approximation I– 3 points

計算一階微分：

$$\frac{\partial F}{\partial x}(j, k) \cong F(j, k) - F(j, k - 1) = G_R(j, k)$$

$$\frac{\partial F}{\partial y}(j, k) \cong F(j, k) - F(j + 1, k) = G_C(j, k)$$

$$G(j, k) = \sqrt{G_R^2(j, k) + G_C^2(j, k)}$$

若 $G(j, k) \geq T$ ，則為邊緣點，設為 1。下圖為 $T = 50/T = 60$ 。



2. Approximation II– 4 points

$$G_1(j, k) = F(j, k) - F(j + 1, k + 1)$$

$$G_2(j, k) = F(j, k + 1) - F(j + 1, k)$$

$$G(j, k) = \sqrt{G_1^2(j, k) + G_2^2(j, k)}$$

若 $G(j, k) \geq T$ ，則為邊緣點，設為 1。下圖為 $T = 50/T = 70$ 。



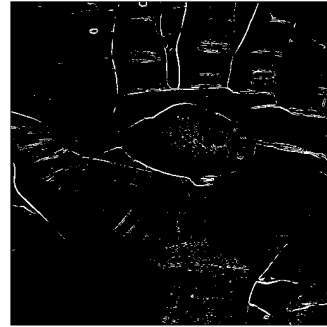
Approximation III– 9points

$$G_R(j, k) = \frac{1}{K+2} [(A_2 + KA_3 + A_4) - (A_0 + KA_7 + A_6)]$$

$$G_C(j, k) = \frac{1}{K+2} [(A_0 + KA_1 + A_2) - (A_6 + KA_5 + A_4)]$$

$$G(j, k) = \sqrt{G_R^2(j, k) + G_C^2(j, k)}$$

若  $G(j, k) \geq T$ ，則為邊緣點，設為 1。其中  $K$  為 2 (Sobel Mask)，下圖為  $T = 50 / T = 70$ 。



**1<sup>st</sup> order edge detection** 偵測手和石頭的邊緣效果還不錯，但是手的下方和石頭上的雜訊也會被當成是邊緣；如果考慮  $F(j, k)$  周圍越多 pixel (9 points) 則雜訊的影響會較小，且邊緣也會越清楚。這邊的 Threshold 取的越大，則輸出的邊緣會越少。

## 2<sup>nd</sup> order edge detection

Step 1. Laplacian matrix

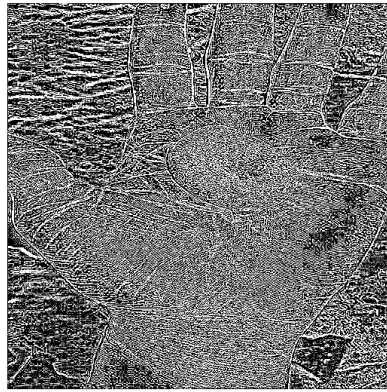
$$\text{Four-neighbor: } H = \frac{1}{4} \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$\text{Eight-neighbor non-separable: } H = \frac{1}{8} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\text{Eight-neighbor separable: } H = \frac{1}{8} \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix}$$

$$\text{LOG: } H = \frac{1}{273} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

由 Four-neighbor 矩陣得下圖：



Step 2. Set up a threshold to separate zero and non-zero to get  $G'$

$$|G(j,k)| \leq T \Rightarrow G'(j,k) = 0$$

由 Four-neighbor 矩陣得左下圖： $(T = 10)$ ，由 LOG 得右下圖 $(T = 100, \text{sigma} = 1)$



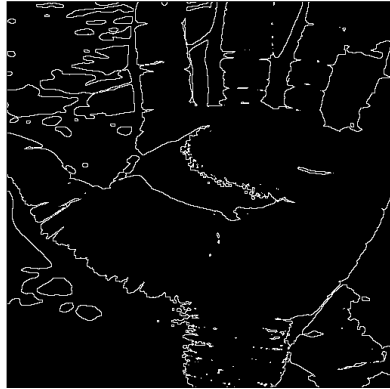
Step 3. For  $G'(j,k)$ , decide whether  $(j,k)$  is a zero-crossing point

若  $G'(j,k) = 0$ ，確認通過該 pixel 的 cross line 是不是異號，若是異號則為 edge。下圖左到右為 Four-neighbor, Eight-neighbor non-separable, Eight-neighbor separable。這三個得出來的效果較不理想，也跟一般 second order edge detection 的結果不太相像，大概是某些地方做錯了，但還沒發現 bug。



LOG 的效果就比較像 second order edge detection 的結果：其中  $\text{sigma}=1, \text{matrix}$

size  $n = \text{ceil}(\sigma * 3) * 2 + 1$ 。下圖左到右為  $T = 100$  及  $T = 150$ 。  $T$  越低細節越少，second order edge detection 對可以過濾掉手上的雜訊，但是線條看起來較扭曲，對背景的邊緣也很敏感。



### Canny edge detection

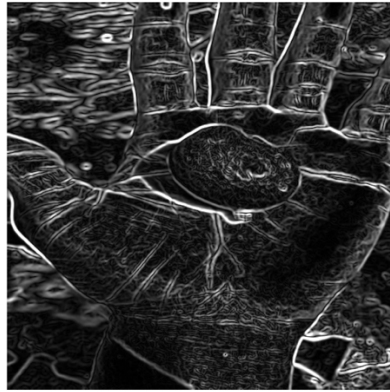
#### Step 1. Noise reduction

用 Gaussian filter  $F_{NR} = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 14 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$  對原圖作模糊。



#### Step 2. Compute gradient magnitude and orientation

用 1<sup>st</sup> order edge detection 效果較好的 Sobel Mask 取 gradient magnitude 及 orientation。其中 magnitude 的圖如下：



### Step 3. Non-maximal suppression

用求出的 orientation 對每個 pixel 法向量上的 neighbors 進行比較，若該 pixel 不是三個 pixel 中的最大值，則設為 0；此步驟可以過濾掉一大部分非邊緣的點。此步驟執行的結果如下：



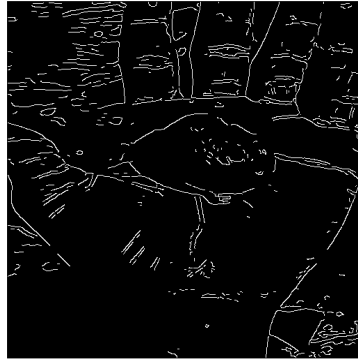
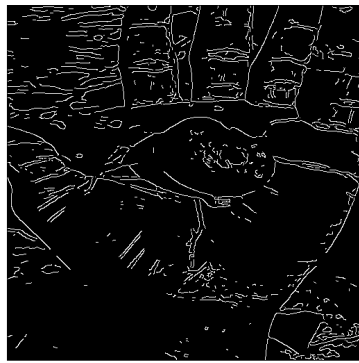
### Step 4. Hysteretic thresholding

### Step 5. Connected component labeling method

設定 $T_H$ 及 $T_L$ ，

$$\begin{cases} G_N(x, y) \geq T_H & \text{Edge Pixel} \\ T_H > G_N(x, y) \geq T_L & \text{Candidate Pixel} \\ G_N(x, y) < T_H & \text{Non-edge Pixel} \end{cases}$$

若為 Candidate Pixel，則檢查此 Pixel 的周圍 8 個點，若有 Edge Pixel，則此點也設為 Edge Pixel。下圖為 $T_H = 100, T_L = 60$ 及 $T_H = 120, T_L = 90$ 。Threshold 設的較小則邊緣的部分越多，Canny edge detection 的雜訊較少，但是手後面地板的地方細節較多。



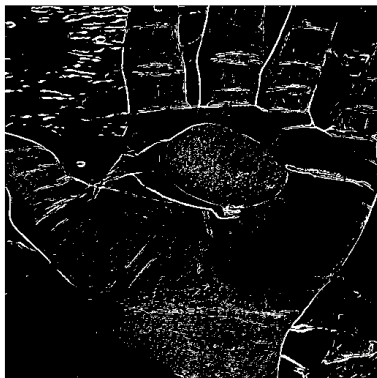
總結以上三種 edge detection 的方法：

|                                      | pros                     | cons                    |
|--------------------------------------|--------------------------|-------------------------|
| 1 <sup>st</sup> order edge detection | 可以寫實的取出線條，<br>取前景線條的效果較好 | 取到的雜訊較多                 |
| 2 <sup>nd</sup> order edge detection | 雜訊較少                     | 容易取到背景線條，且<br>線條看起來較不寫實 |
| Canny edge detection                 | 較寫實且雜訊較少                 | 容易取到背景線條，步<br>驟最繁瑣      |

(b) 套用上一題的三種方法做比較：

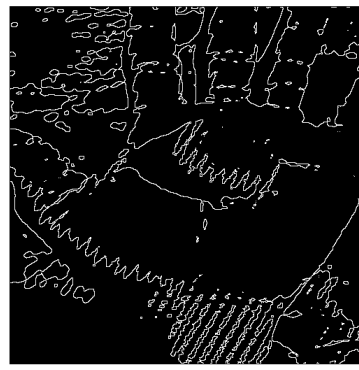
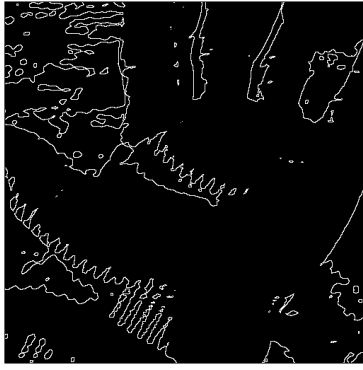
#### 1<sup>st</sup> order edge detection

用 1<sup>st</sup> order edge detection 的結果跟上一題差不多，不管 threshold 怎麼調，都不太受此題的 noise 影響。下圖為  $T = 50$  /  $T = 70$ 。



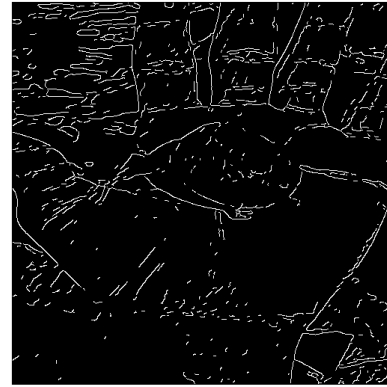
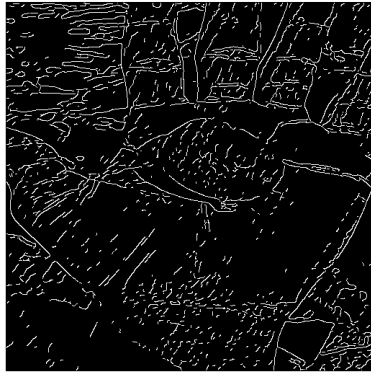
#### 2<sup>nd</sup> order edge detection

2<sup>nd</sup> order edge detection 就看得出來手的下半部有明顯的斜線 noise，下圖左到右為  $T = 100$  及  $T = 150$  (sigma 為 1)， $T$  取得越大，斜線的 Noise 越多。



### Canny edge detection

Canny edge detection 的結果也會被 noise 影響，下圖左到右為  $T_H = 100, T_L = 80$  及  $T_H = 120, T_L = 90$ 。



總結以上，雖然 Canny 的方法在前幾個步驟跟 First order edge detection 差不多，但可能在 Non-maximal suppression 的時候把雜訊的效果放大了，2<sup>nd</sup> order edge detection 也是會放大細節部分的變化，所以雜訊的影響也變大，就像 Canny 和 2<sup>nd</sup> order edge detection 會把背景的邊緣也清楚的顯示一樣。所以若有這種雖然深淺變化不大但全面性的雜訊，適合用 first order edge detection。

### PROBLEM 2: GEOMETRICAL MODIFICATION

(a)

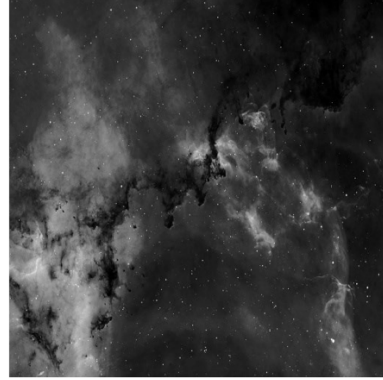
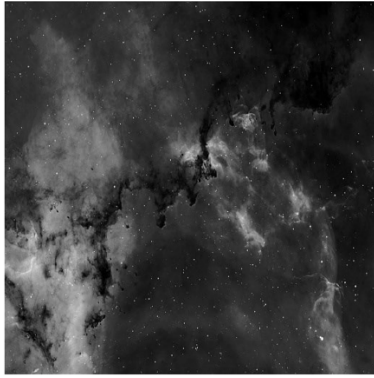
用公式：

$$G(j, k) = \frac{c}{2c - 1} F(j, k) - \frac{1 - c}{2c - 1} F_L(j, k), \text{ where } \frac{3}{5} \leq c \leq \frac{5}{6}$$

其中  $F_L$  為原圖通過 low-pass filters 的結果，這裡我取的 low-pass filter 為

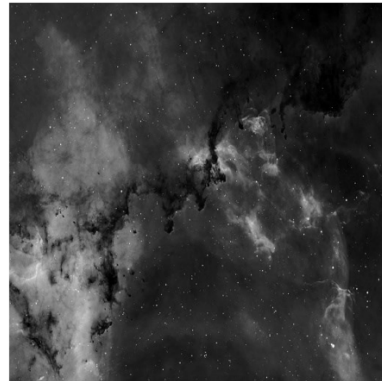
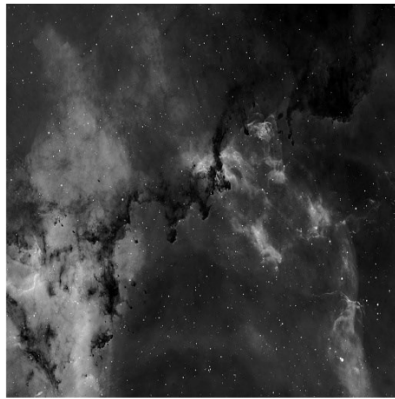
$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

下圖左到右為  $c = \frac{3}{5}$  及原圖



下圖左到右為 $c = \frac{5}{6}$ 及原圖，放大看有經過 **crispening** 的圖都較原圖銳利，

但 $c = \frac{3}{5}$ 和 $c = \frac{5}{6}$ 的結果差不多。



(b) 轉換的公式為：

$$u = ax + b\sin\left(-\frac{2\pi y}{c} + d\right)$$

$$v = ex + f\sin\left(-\frac{2\pi y}{g} + h\right)$$

其中 **a** 和 **e** 是調整左右或上下波形間間距，若 **a, e** 較小，則黑色部分較多，兩個波形較靠近。

**b** 和 **f** 是調整波形的幅度，如果 **b, f** 越大，則波的起伏較大。

**c** 和 **g** 是調整波形的寬度，如果 **c, g** 越大，則波長越大，波越寬。

**d** 和 **h** 是調整波形的位移，向左右、上下沿著波的方向移動波形。

在此題取的參數為 $a = 0.96, b = 15, c = 220, d = 1, e = 0.96, f = 17, g =$

$160, h = 0$ 。結果如下圖，雖然左上角有多出一點黑色的部分，但因為其他幾個邊都能大致上吻合，所以選用這個調整函數。



若轉換後的圖有沒有對應到的 pixel，則把該點的上下左右點做平均。

