

Report DIP HW2

B07902144 彭約博 資工三

Problem 1 EDGE DETECTION

(a)

(1)

▼ approach

使用課堂上教到的水平和垂直梯度計算方法，K=2：

■ Row gradient

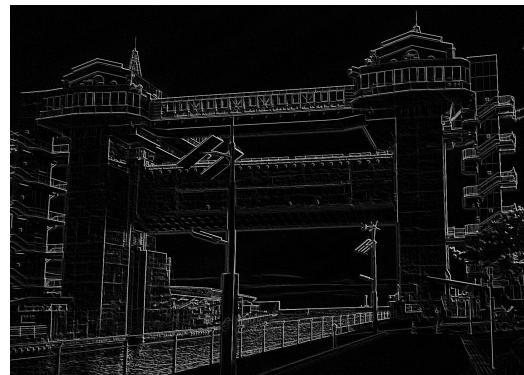
$$G_R(j, k) = \frac{1}{K+2} [(A_2 + KA_3 + A_4) - (A_0 + KA_7 + A_6)]$$

■ Column gradient

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

並將其取幾何平均，得到result1

- sample1.jpg
- result1.jpg



(2)

▼ approach

首先使用low pass filter(5x5):

$$\frac{1}{273} \begin{array}{|c|c|c|c|c|} \hline 1 & 4 & 7 & 4 & 1 \\ \hline 4 & 16 & 26 & 16 & 4 \\ \hline 7 & 26 & 41 & 26 & 7 \\ \hline 4 & 16 & 26 & 16 & 4 \\ \hline 1 & 4 & 7 & 4 & 1 \\ \hline \end{array}$$

再對其使用Laplacian:

$$-\nabla^2 \cong \frac{1}{4} \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix};$$



找出其為0之點，並檢查其是否為某方向的zero cross
若是的話將其標示成edge，得到result2

- sample1.jpg

- result2.jpg



(3)

▼ approach

依照課堂中之方法執行5步驟:

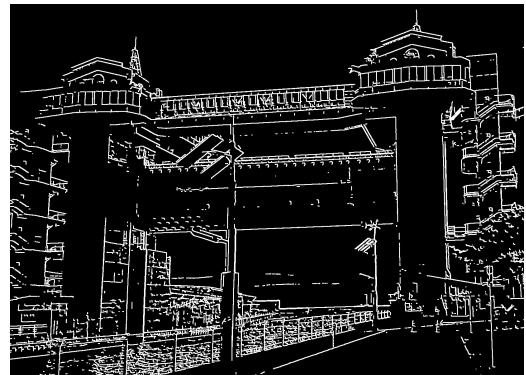
1. 先使用low pass filter進行降噪，filter選用以下matrix

```
G_low_pass = np.array(([2, 4, 5, 4, 2],  
                      [4, 9, 12, 9, 4],  
                      [5, 12, 15, 12, 5],  
                      [4, 9, 12, 9, 4],  
                      [2, 4, 5, 4, 2]), np.float32) * (1/159)
```

2. 使用first order Edge detection
3. 使用上步驟計算出之theta判斷沿著此方向的各一個pixel有沒有值，若有則保留，否則設為0
4. 進行thresholding，這裡設high=35，low=20，並且介於其中之Pixel設為25
5. 最後判斷每個pixel之周圍8個pixel相加之後是否小於255，若是，則將其設為0，若否則設為255，完成連接

- sample1.jpg

- result3.jpg



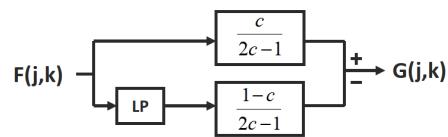
(4)

▼ approach

使用課堂上提到之Unsharp masking方法:

■ Unsharp Masking

- Appropriate combination of all-pass and low-pass filters



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

Low pass filter選用以下的matrix

```
G_low_pass = np.array(([2, 4, 5, 4, 2],  
[4, 9, 12, 9, 4],  
[5, 12, 15, 12, 5],  
[4, 9, 12, 9, 4],  
[2, 4, 5, 4, 2]), np.float32) * (1/159)
```

其中c選擇4/5

- sample1.jpg
- result4.jpg



(5)

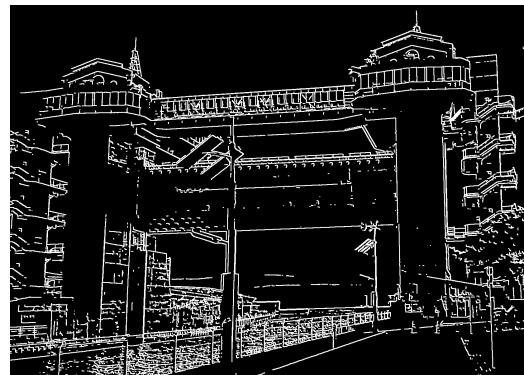
▼ approach

直接使用Canny偵測Edge

- sample1.jpg



- result5.jpg



討論:

可發現使用1階和2階edge detection時，2階會在一些原本1階沒有標記出edge的地方，出現edge，而那確實也是建築物的edge，可發現2階的確是在偵測到2階微分為零的地方後，藉由zero crossing判斷其是否為edge

使用canny藉由connect component 和thresholding後相較前兩者可看清楚更多建築物上的細節

result4藉由Unsharp masking後加強之邊界效果頗好，再對其做Canny後可以發現許多建築物原先無法連接起來的邊界可以成功連成一條線，例如建築物中連結兩建物的通道的水平線等等

(b)

▼ approach

步驟如下：

首先使用2次了HW1中的power law filter加強對比，gamma分別為1.95、1.85

再使用Edge crispening加強輪廓

再使用兩次low pass filter

再使用Canny

最後使用2nd Edge detection

在製作過程中發現：

若僅僅使用Canny或額外的1次low pass filter可能會造成拱門中間有需多白點

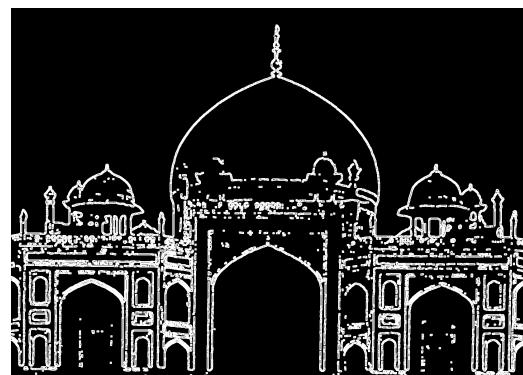
若沒有使用power law filter和Edge crispening等加強對比和輪廓的方法，會使得建築物圓頂處之edge標記的不完全，有許多斷點甚至缺失整條邊界

經許多嘗試之後得到以下的結果

- sample2.jpg



- result6.jpg



Problem 2 GEOMETRICAL MODIFICATION

(a)

▼ approach

- rotating、shifting、scaling都使用課堂中之方法

步驟如下:

先將其對圖之正中心旋轉90度

再將其對-x, -y 方向分別shift 100、300個pixel

再對其x,y 方向都scale 1.8

最後矯正其位置 往x, -y 方向分別shift 50、-100個pixel

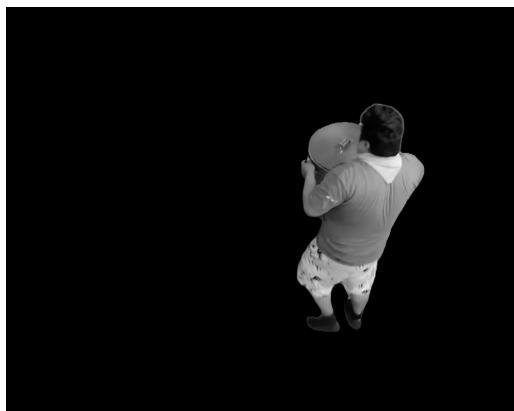
最後由於scale的矩陣會將原本的pixel分配到拉長後的地方，造成中間有些空隙，故使用了幾次以下的filter:

```
K = np.array(([1, 1, 1],  
             [1, 0, 1],  
             [1, 1, 1]), np.float32)/8
```

對圖片卷積後補齊空隙(雖然可能損失一些解析度，但目測砍起來還好)，再調高其亮度

其中也發現使用matrix旋轉圖片時在大部分的角度會使得圖片產生孔洞，上網查詢是因為函數的疊加現象，不過此處因為旋轉到90度沒有出現，故沒有對其特別處裡。

- sample3.jpg



- result7.jpg



(b)

▼ approach

由於觀察到統神的腰部有變細，且似乎黑洞中心處旋轉的最劇烈，故先將其做 pincushion distortion 後再對其旋轉，並且最靠近中心者旋轉最多，往外逐漸減少

首先將其變換成極座標，再對其進行 pincushion distortion，使用以下公式：

$$r_d = r_u(1 - \alpha|r_u|^2)$$

其中 alpha 為其扭曲的程度，此處設為 -0.0001

再對其中心點做旋轉分別，從最中心 +-8 +-16 +-64 +-256 逐步進行旋轉。

- sample5.jpg



- result8.jpg

