

學號:B06902136

系級:資工四

姓名:賴冠毓

### Problem 1: EDGE DETECTION

(a)

(1)

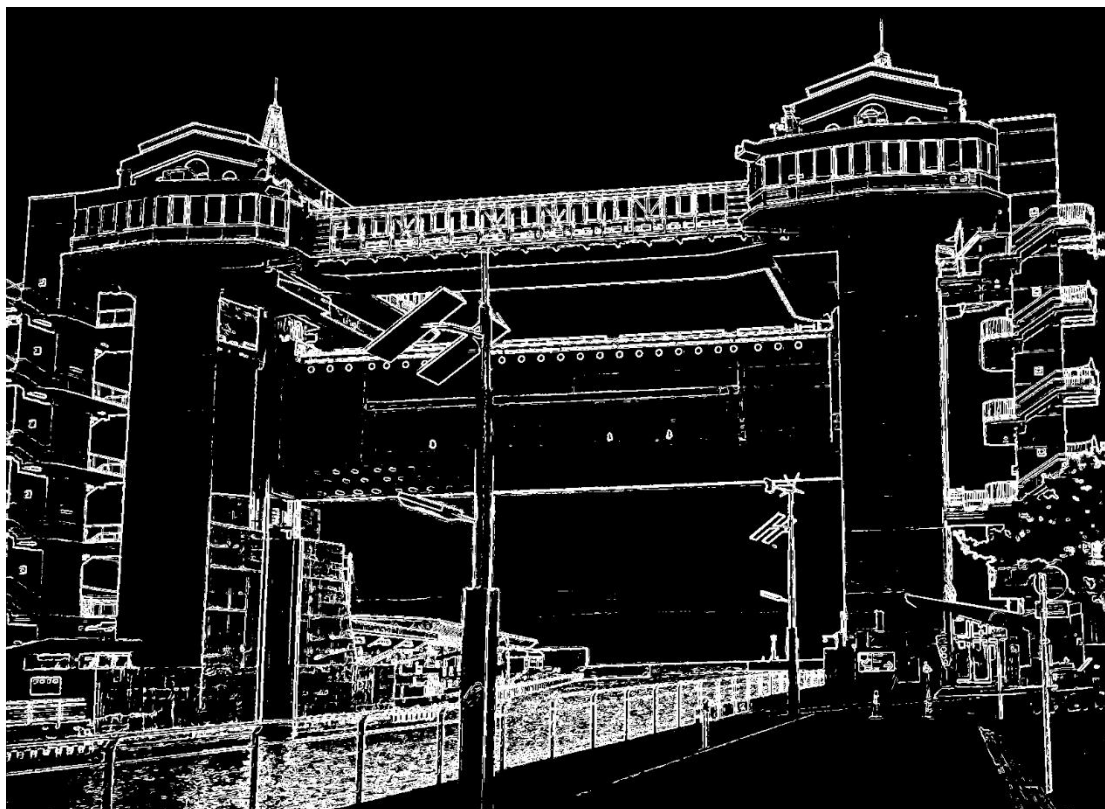
My motivation and approach:

就類似算梯度，用 row filter 跟 column filter 計算出大小，然後跟 threshold 比較，比較大的就是 edge。

Original images:



Output images:



Discussion of results:

filter 的  $b$  調小一點似乎比較亮，threshold 則控制在 20 比較好。

(2)

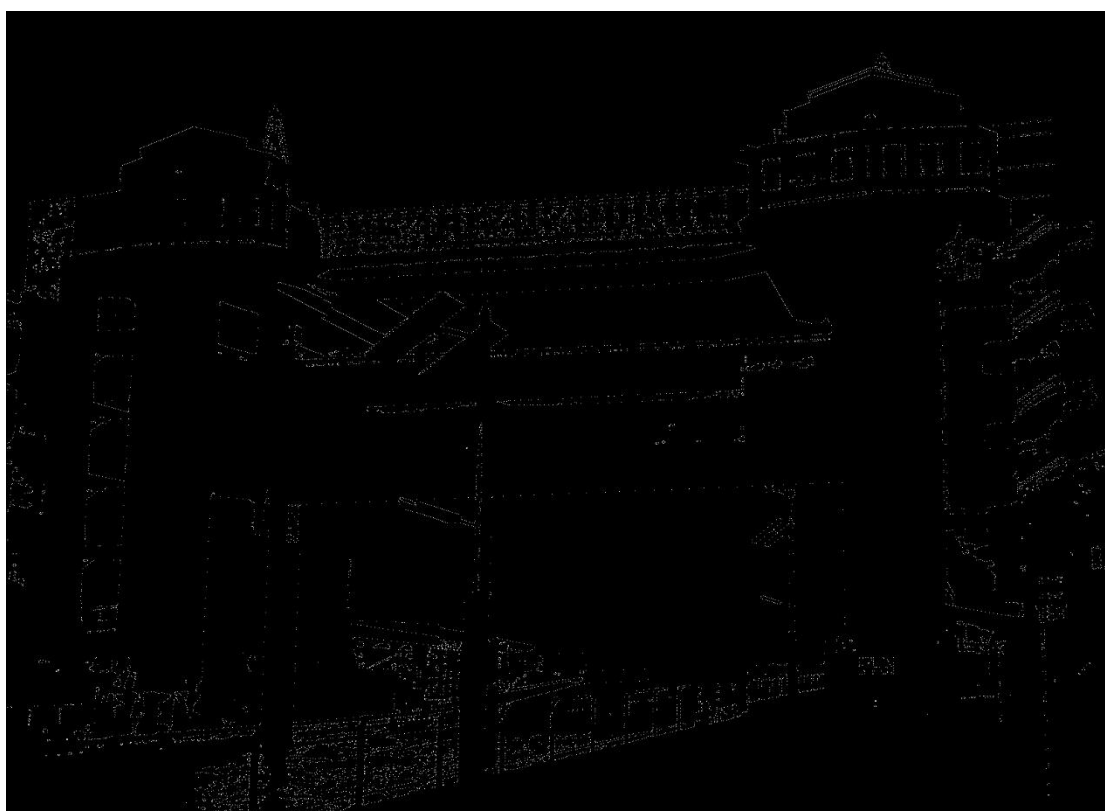
My motivation and approach:

開更大的 kernel ( $5 * 5$ )，然後第一次先用這個 filter 加上 threshold 去過濾，第二次再根據過濾後的值去判斷，如果這個點附近沒有相同的點，那它是 edge，其餘狀況則不是。

Original images:



Output images:



Discussion of results:

high threshold 用 100，low threshold 用 20 比較好。

(3)

My motivation and approach:

就是實作 Canny edge detection 的 5 個步驟。

Noise reduction:

用 Gaussian filter ( $\sigma = 1.4$ )。

Compute gradient magnitude and orientation:

類似 first-order 的方法，用 row filter 跟 column filter，只是最後要算出梯度值跟角度。

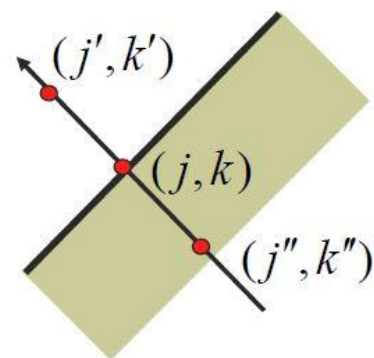
$$G(j, k) = \sqrt{G_R^2(j, k) + G_C^2(j, k)}$$

$$\theta(j, k) = \tan^{-1} \left( \frac{G_C(j, k)}{G_R(j, k)} \right)$$

Non-maximal suppression:

**Search the nearest neighbors  $(j', k')$  and  $(j'', k'')$  along the edge normal**

$$G_N(j, k) = \begin{cases} G(j, k) & \text{if } G(j, k) > G(j', k') \\ & \text{and } G(j, k) > G(j'', k'') \\ 0 & \text{otherwise} \end{cases}$$



Hysteretic thresholding:

**Label each pixels according to two threshold:  $T_H, T_L$**

$G_N(x, y) \geq T_H$       **Edge Pixel**

$T_H > G_N(x, y) \geq T_L$       **Candidate Pixel**

$G_N(x, y) < T_L$       **Non-edge Pixel**

Connected component labeling method:

根據 threshold 比較完的結果給 pixel 值。如果是 candidate pixel 則判斷附近的 pixel，如有一個是 edge pixel 那也把它當 edge，這樣出來的 edge 看起來會比較粗。

Original images:



Output images:



Discussion of results:

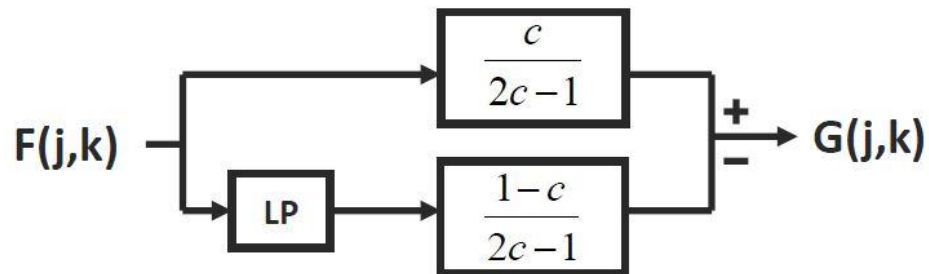
row filter 跟 column filter 的  $b$  跟 first-order 一樣，換其他數字似乎沒甚麼影響。

high threshold 用 30，low threshold 用 10 比較好。

(4)

My motivation and approach:

圖片如下面方法，而 edge map 用 Canny 實作。



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

Original images:



Output images:

image:





edge map:



Discussion of results:

filter 的  $b$  一樣沒什麼差異，但  $c$  比較小的話效果比較明顯。

(5)

Discussion of results:

比較起來 first-order 的亮度最亮，還可能帶有部分雜質。

second-order 則太暗了，edge 相當不明顯。

Canny 則是三者中最好的方法，最符合實際 edge 的情況。

edge crispening 後的圖邊現有更銳利，但是 edge map 看起來沒有差到太多。

(b)

My motivation and approach:

用最好的方法—Canny。

Original images:



Output images:





Discussion of results:

用 Canny 的效果的確是最好的，但是惟需注意，這邊的 threshold 要調低，high threshold(這裡用 10)跟 low threshold(這裡用 5)的差距也要縮小，因為整張圖的太亮，邊緣差距夠不明顯，如果 threshold 設太高 edge map 會幾乎全黑。

## Problem 2: GEOMETRICAL MODIFICATION

(a)

My motivation and approach:

透過變換 x、y 座標，然後不斷地調整參數，達到跟旋轉矩陣加放大一樣的效果。

x 座標 =  $\text{round}(0.6 \times j) + 200$

y 座標 =  $\text{img3\_height} - \text{round}(0.6 \times i) + 50$

Original images:



Output images:



Discussion of results:

看起來統神的火鍋確實打翻了，跟 sample4. jpg 有像。

(b)

My motivation and approach:

首先要先把圖片長寬弄成一樣比較好操作，然後把座標的定在圖中間，然後進行旋轉，距離中心越近，旋轉角度就越大，最後再 resize 回原本大小。

Original images:



Output images:



Discussion of results:

有弄出旋轉扭曲的效果了，但是好像扭轉太多，統神的頭已經在地上了，所以出來的圖片很奇怪= =