Digital Image Processing Homework 2

R13944043 林詩程

Problem 1:

(a) **Motivation and Approach:**

首先利用投影片的方法找到 Row Gradient 和 Column Gradient ，其中 $K = 2$ (Sobel) ，再計算兩種 Gradient 的 Euclidean Norm 作爲新值，最後生成 result1.png 。
result2.png 則是在 result1.png 的基礎上，選擇一個閾值 (Threshold) ，區分 edge point 和 non-edge point ，當閾值越小，被認定爲邊界的點就會越多，也就是雜訊會越多，但是能捕捉到的邊界也會越多，反之亦然，因此需要在雜訊和邊界上做權衡，我這邊選擇 $Threshold = 30$。

Discussion of Results:

從 result1.png 和 sample1.png 的比較可以發現， 肉眼上感覺會是邊界的點大致上都有在 result1.png 被標記出來。
result2.png 中，我首先觀察 CDF ，排除較不合理的範圍 (>100)，並嘗試了幾個比較有機會的閾值，最後選定 Threshold = 10, 30, 50 作爲候選，其中 Threshold = 10 的雜訊太多不考慮，而 Threshold = 30, 50 則各有優劣， Threshold = 30 捕捉到的合理邊界較多，Threshold = 50 的雜訊較少。最後我選擇 Threshold = 30 的理由是 Threshold = 30 多捕捉到拱門左上方磚頭的邊界，而 Threshold = 50 在這部分幾乎消失了。這個理由是基於我覺得爲了多捕捉到更多合理的邊界而增加一點雜訊，比爲了減少雜訊而損失捕捉更多合理邊界的機會來得更好一些。

sample1.png

result1.png

Threshold = 10

result2.png (Threshold = 30)

Threshold = 50

(b) **Motivation and Approach:**

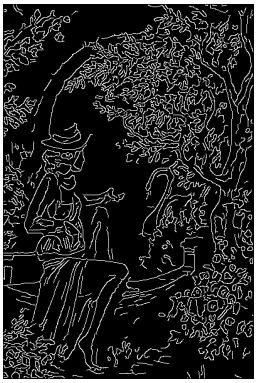
利用投影片的五個步驟進行處理，首先利用 5x5 Gaussian Filter ($\sigma = 1.4$) 平滑原圖。再計算 Gradient Magnitude 和 Orientation，並沿用上一題的 Sobel Mask。接著利用 Non-Maximal Suppression 保留每個方向上值最大的點。之後使用 Hysteretic Thresholding 標記 Edge, Candidate, Non-Edge Pixel。最後將和 Edge 相連的 Candidate 也加入 Edge。

Discussion of Results:

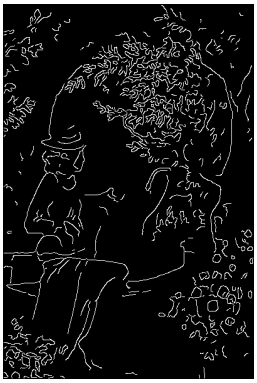
關於選擇 Threshold ，首先經過測試，發現當 Threshold (Low) >100 時，邊界會消失很多，往下嘗試後找出三個候選 Threshold (Low) = 10, 30, 50，雖然 Threshold (Low) = 10 很細緻的捕捉到幾乎所有的邊界，但是太多邊界導致分不清主次，而 Threshold (Low) = 50 則感覺少了一些細節，於是折衷選擇 Threshold (Low) = 30 (Threshold (High) 皆固定爲 Threshold (Low) 的兩倍)。可以發現 Canny Edge Detection 對於邊界偵測的效果很好，幾乎所有標記的部分都是乾淨俐落的線條，而沒有模糊的感覺。

sample1.png

Threshold (Low) = 10



result3.png
(Threshold (Low) = 30)



Threshold (Low) = 50

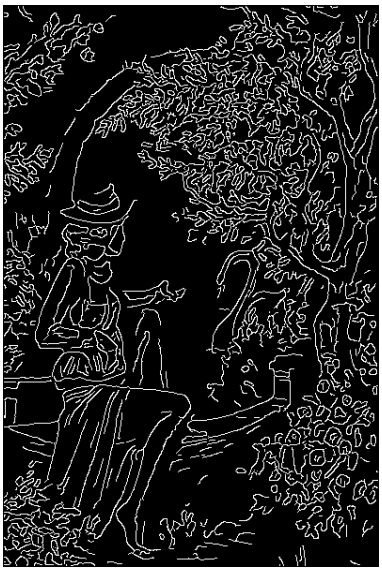
(c) **Motivation and Approach:**

利用投影片的方法處理，首先利用 5x5 Gaussian Low-pass Filter 平滑原圖，再通過 The Gain-Normalized version of the Separable Eight-Neighbor Laplacian 得到 G ，最後選定合適的閾值 ($Threshold = 2$) 進行 Zero-Crossing Detection。
閾值的選定也是利用類似投影片的方法，找出並觀察 G 的直方圖，發現閾值設定在 2 左右可能是不錯的選擇。

Discussion of Results:

比較 result2.png, result3.png, result4.png 可以發現，就線條完整度與俐落度來說， result3.png 有較爲明顯的優勢， result2.png 則看起來似乎有較爲明亮的感覺，可能是邊界附近的雜訊造成的， 而 result4.png 在這張圖的邊界檢測上似乎沒有特別的優勢。就結果而言， Canny Edge Detection 在這張圖的處理上表現看起來是更爲出色的。

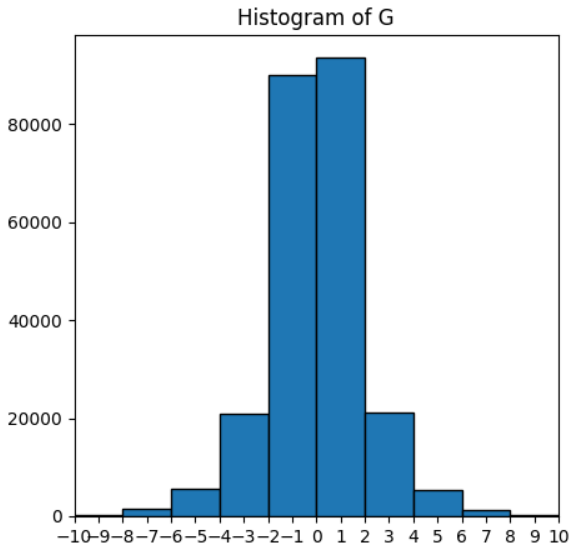
result2.png



result3.png

result4.png

sample1.png



Histogram of G

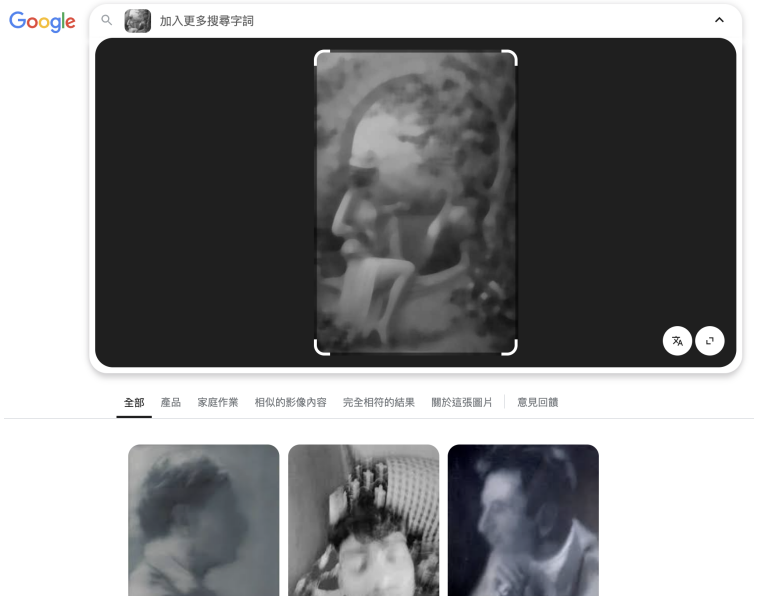
(d) **Motivation and Approach:**

原圖仔細看的話不太像一個老者的側臉，後來發現把圖片拉遠或者是不在原圖上對焦，簡而言之就是變模糊的話，一個老者的側臉就顯現出來了，根據這個想法，我的目標應該就是盡可能的在合理的範圍內模糊整張圖片，這邊我直接引用作業一的 Median Filtering ，原本 Median Filtering 降噪的副作用之一就是會把圖片變模糊，但是在這個副作用在本題變成了一個很好的優勢，經過測試發現在 (Mask Size = 17) 時，會有不錯的效果。

Discussion of Results:

根據結果截圖可以發現這個想法取得了不錯的成效，顯示 Google 判斷圖片的方式可能也跟肉眼有共通之處。此外，經過測試，如果變得太模糊，搜尋出來的結果也是一堆模糊的幻影，因此也不能無止盡的模糊圖片。

resultd.png



screenshot

(e) **Motivation and Approach:**

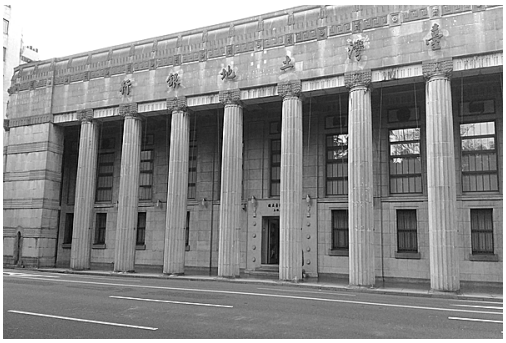
利用投影片的五個步驟進行處理，首先利用 5x5 Gaussian Filter ($\sigma = 1.4$) 平滑原圖。再計算 Gradient Magnitude 和 Orientation，並沿用第一題的 Sobel Mask。接著利用 Non-Maximal Suppression 保留每個方向上值最大的點。之後使用 Hysteretic Thresholding 標記 Edge, Candidate, Non-Edge Pixel。最後將和 Edge 相連的 Candidate 也加入 Edge。

result6.png 透過投影片的公式 $\rho = x\cos(\theta) + y\sin(\theta)$ ，將每個被偵測爲邊界的像素點轉換到 Hough Space 形成曲線並繪製。

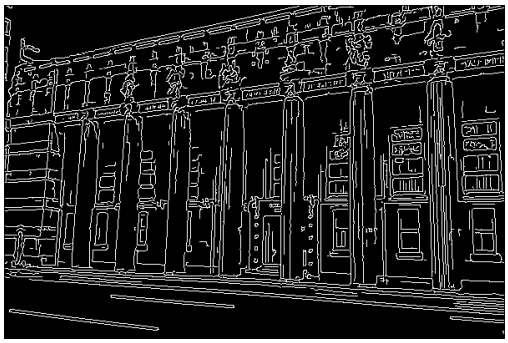
result7.png 將 Hough Space 上的點映射回笛卡爾坐標，並適量加長畫成線段，爲避免太多線段，以 Votes 數 >100 作爲閾值，結果如 result7.png 紅色線段所示。

Discussion of Results:

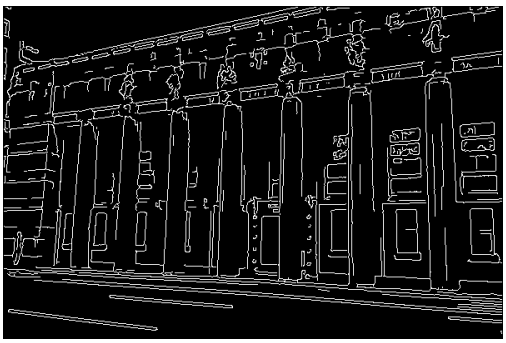
在 result5.png Threshold 的選擇上，首先經過測試，發現當 Threshold (Low) >100 時，邊界會消失很多，往下嘗試後找出三個候選 Threshold (Low) = 30, 50, 70，雖然 Threshold (Low) = 30 捕捉到很多邊界，但是太多邊界可能會對後續數柱子造成一些負面影響，而 Threshold (Low) = 70 則感覺太過簡略，於是折衷選擇 Threshold (Low) = 50 (Threshold (High) 皆固定爲 Threshold (Low) 的兩倍)。
result6.png 中 Votes 越高，表示被轉換後的曲線重疊數目越高，也表示原笛卡爾坐標中的直線是由越多點組成，即更可能是邊界。
Mark and Count the Pillars 將和柱子較爲無關的線段轉成藍色，可以發現加上最左邊的大柱子，總共有八根柱子，而紅色線段把這八根柱子的輪廓大致上都勾勒出來了。



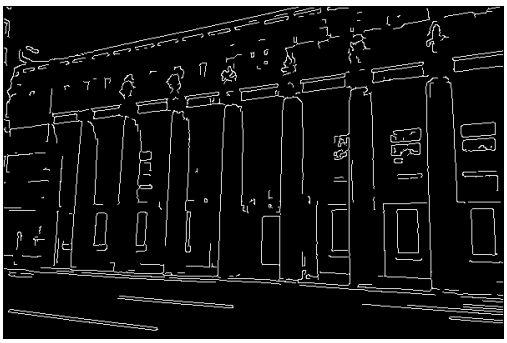
sample2.png



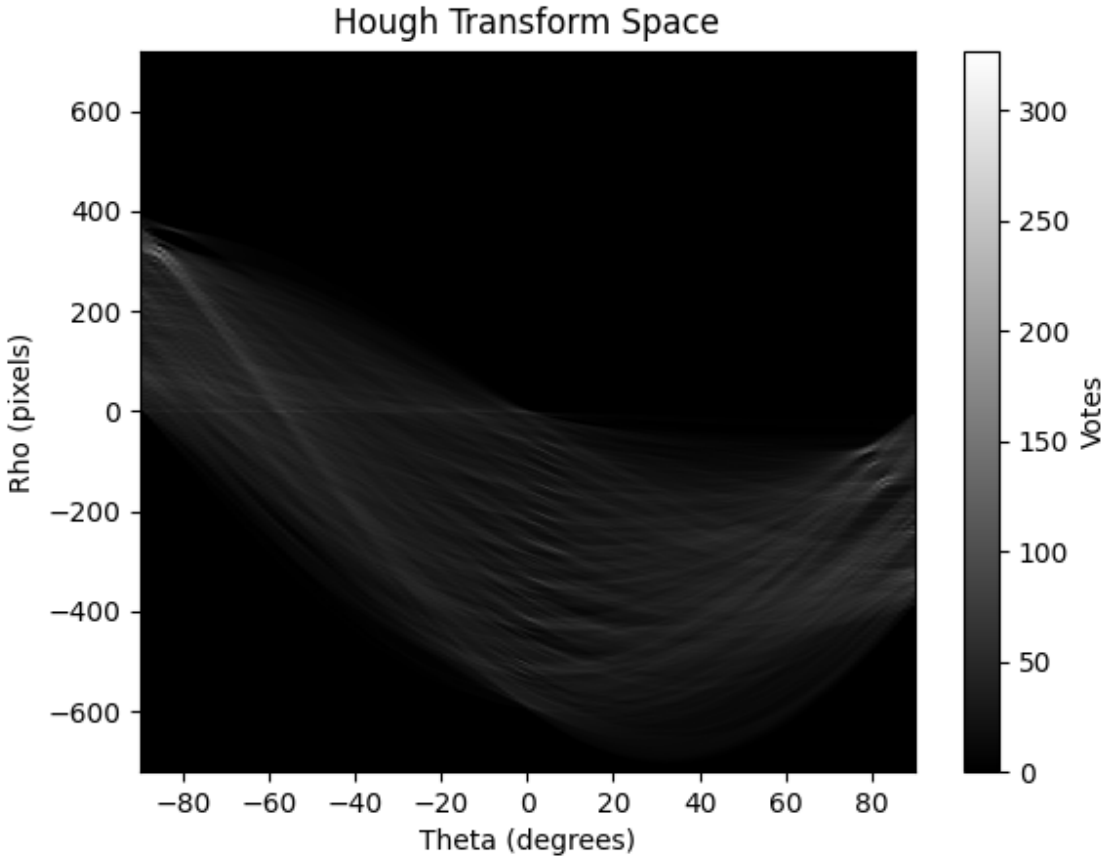
Threshold (Low) = 30



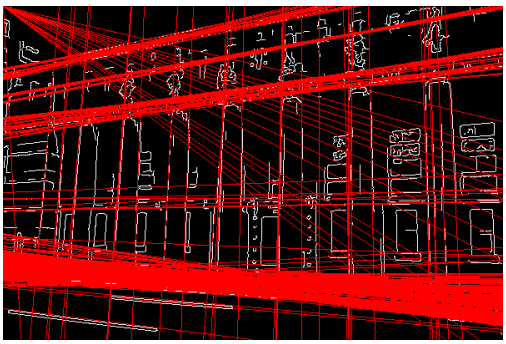
result5.png (Threshold (Low) = 50)



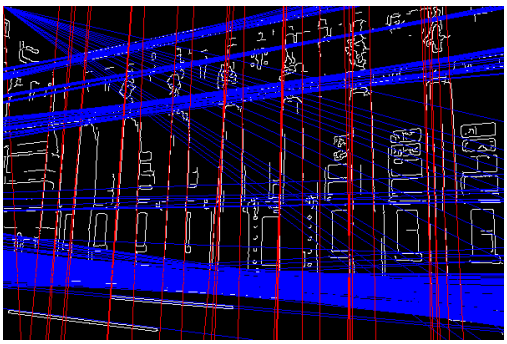
Threshold (Low) = 70



result6.png



result7.png



Mark and Count the Pillars

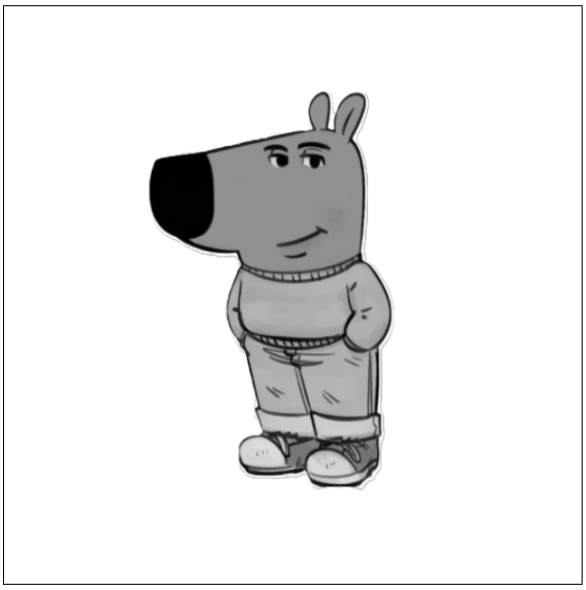
Problem 2:

(a) **Motivation and Approach:**

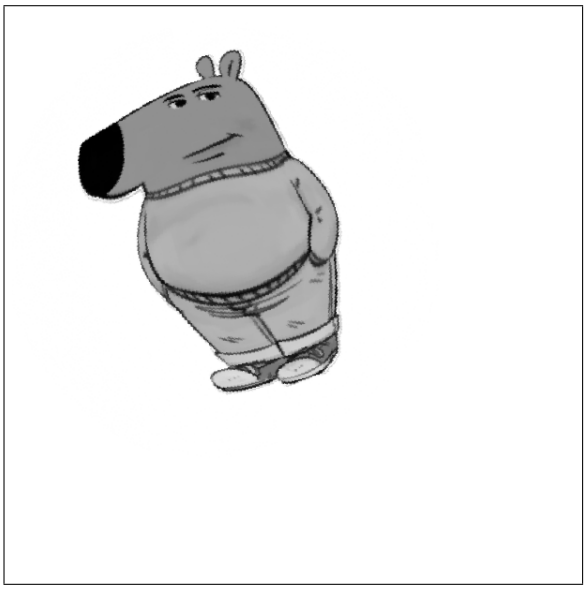
對照 sample3.png 觀察 sample4.png ，發現 Chill Guy 肚子變大，但是其他部位沒有等比例放大，推測可能有經過 Barrel Distortion 。同樣根據肚子，發現兩邊肚子位置不一樣，應該是有向左上位移的跡象。根據頭部則是發現應該有逆時針旋轉的痕跡。最後觀察整體，發現 Chill Guy 好像小隻一點，所以也可能有經過縮小。
經過嘗試發現，首先以大約中心的位置爲中心點，進行 Barrel Distortion ，並將 Chill Guy 縮放 0.8 倍，再逆時針旋轉 15° ，最後往左上移動 $75\sqrt{2}$ 可以看起來接近 sample4.png 。

Discussion of Results:

看起來和 sample4.png 接近，不過仍有一點差距，雖然我不會通靈，但結果看起來尚且可以接受，而且有點落差可以表示不是用截圖的。



sample3.png



result8.png

(b) **Motivation and Approach:**

首先爲整張圖片切割成一個一個網格點以方便計算距離，每個點減去旋轉中心座標使旋轉中心爲 (0, 0) ，將這些距離轉換成極座標， dist 表示每個像素點距離旋轉中心的距離， θ 表示每個像素點相對於旋轉中心的角度（弧度），漩渦效果用這個公式決定：

$$\text{thetas} = \theta + \text{pwr} \cdot e^{-\frac{\text{dist}}{\text{rad}}}$$

pwr 爲強度， rad 爲最大影響距離。這個式子符合直覺，距離旋轉中心越遠的點（ dist 越大），受到的影響越小。最後轉回笛卡爾坐標， X 方向爲 $ctr[0] + dist \cdot cos(thetas)$ ， ctr[0] 爲 X 方向的旋轉中心座標。 Y 方向爲 $ctr[1] + dist \cdot sin(thetas)$ ， ctr[1] 爲 Y 方向的旋轉中心座標。
後續經過嘗試發現，中心設定爲 (200, 200) ， $rad = 40$ ， $pwr = 30$ 可以有不錯的效果。

Discussion of Results:

對比 sample6.png ，可以發現已經相當接近了，雖然仍有一點點差距，不過結果感覺不錯，可能是有些參數設定上和 sample6.png 略有落差，但我不會通靈，所以也不確定問題確切出在哪裡。儘管如此，至少方向上我覺得是正確的。

sample5.png

result9.png