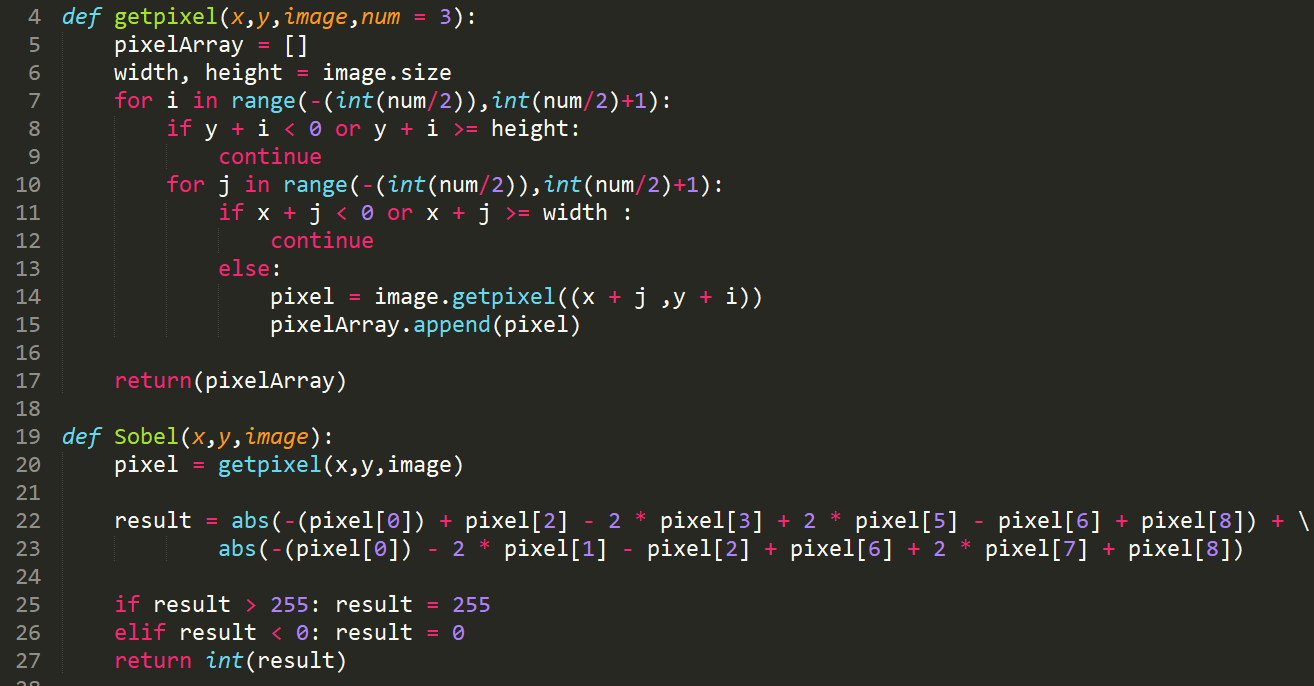
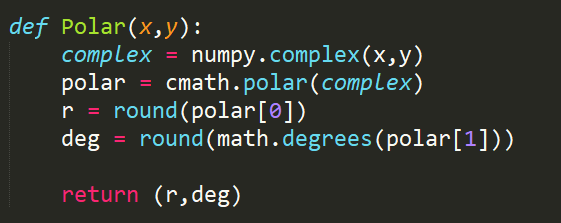
影像處理 作業四

學號:7106056008 姓名:詹英鴻

使用Hough Transform 找出圖中最長直線

獲取3\*3 像素的函式，並運用在一階微分(Sobel)函式的計算上

當今天(x,y)的位置在圖片邊緣，則超出圖片範圍的點忽略不存取



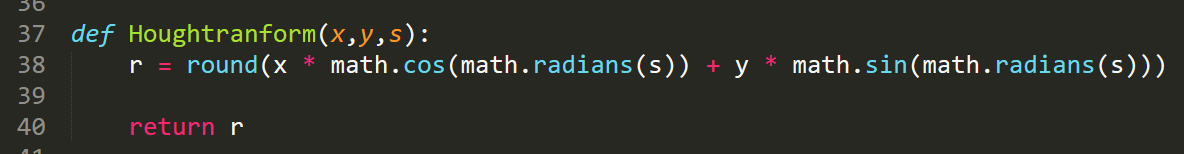
計算將(x,y)轉成極座標的函式

使用numpy.complex()將(x,y)座標轉換成複數表示 x + yj

再代入cmath.polar()算出極座標的r,θ

這邊的θ是弧度必須再經過math.degrees()轉換成度數

最後回傳r,θ



Hough transform 轉換的函式

將x, y, θ 代入 x \* cosθ + y \* sinθ = r的公式

計算出(x,y)在不同θ下r的數值

運用在畫出(x,y)轉換成(r,θ)後的曲線圖

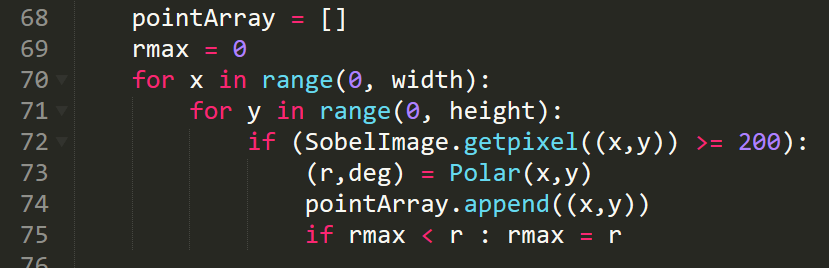
統計出重複最多次的(r,θ)即為圖中最長的直線

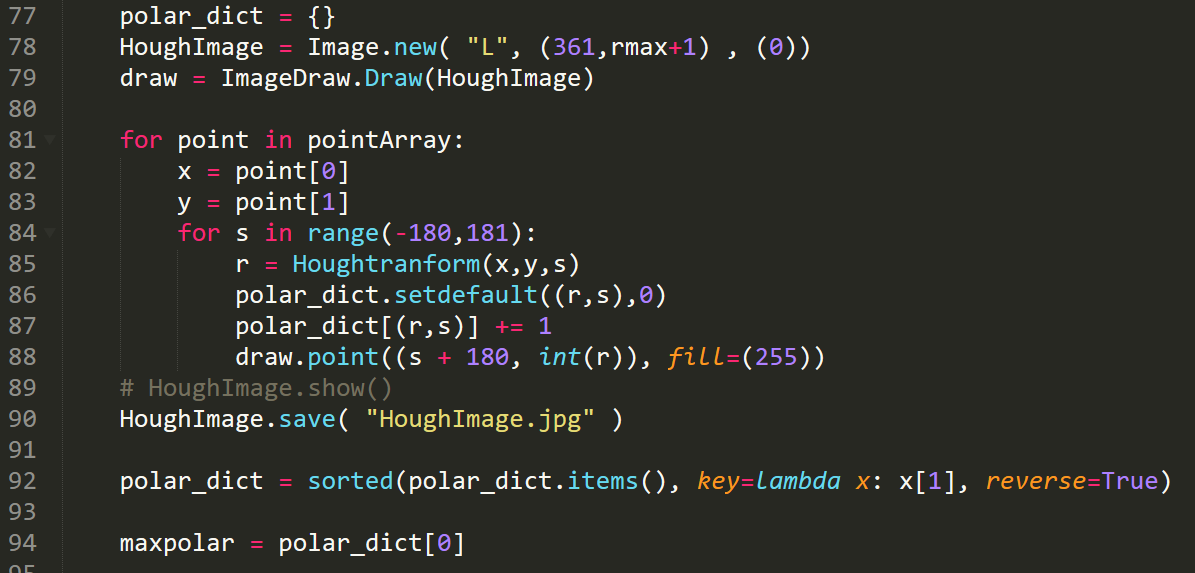
主程式

先將原圖讀入並轉成灰階，因為我們要找出最長的直線，所以再利用Soble函式找出圖中的edge

****

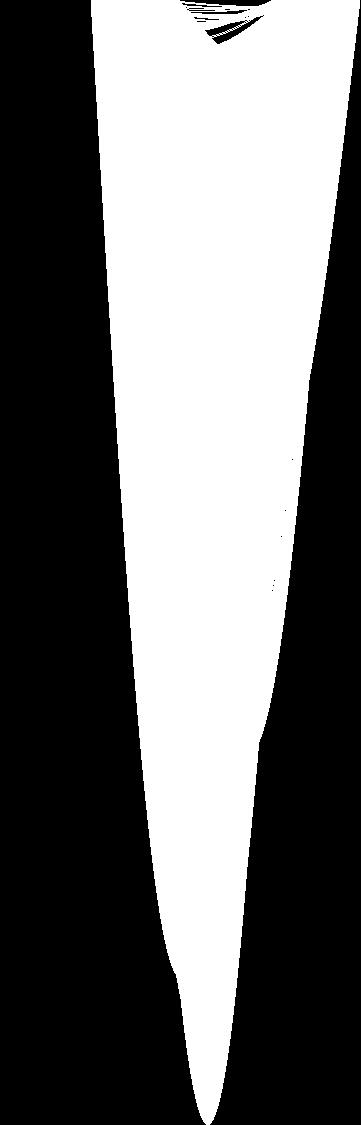
灰階Sobel

將sobel圖中像素值 >= 200都設為edge的位置，並代入Polar函式中，算出最大的r，並將edge的位置存到pointArray中



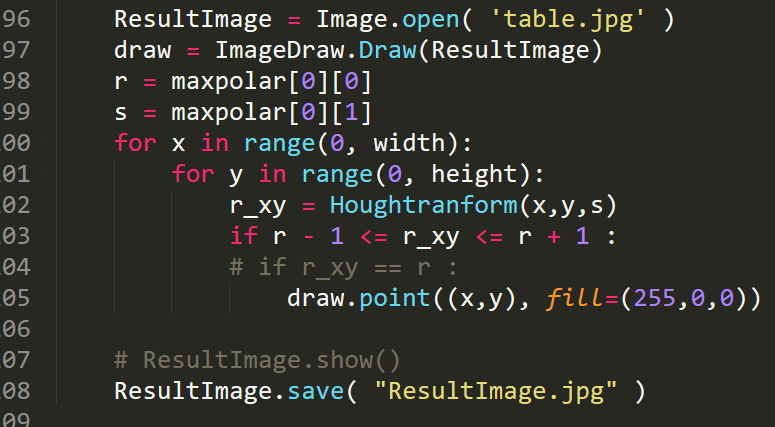
開始畫出(r,θ)曲線圖，x軸為θ(-180 ~ 180)，y軸為r(0 ~ rmax)，將每個edge的(x,y)代入Houghtransform函式，計算出θ在 -180 ~ 180間的每個 r值，並將其一一點在圖上，在polar\_dict中統計(r,θ)的次數，畫完圖後，將polar\_dict做排序，找出次數最多的(r,θ)，即為原圖中的最長直線

-180 180

 0

rmax

Hough transform曲線圖



將最長直線的(r,θ)帶回原圖中的(x,y)，透過Houghtransform函式找出相等的r值即為最長直線的像素位置，將其像素值改為紅色，程式碼中將範圍擴大為r–1 ~ r+1都成立，使畫出來的紅線能夠變寬、明顯



結果圖

結論

做此次作業時並不順利，一開始的想法為使用python中的cmath.polar()函式，將每個edge的(x,y)代入轉換成極座標，直接統計出現最多的(r,θ)，再將其edge的像素位置在原圖上變為紅色，想跳過做(r,θ)曲線圖的步驟，結果完全錯誤，後來依照步驟先做出(r,θ)曲線的圖，在統計出現最多的(r,θ)並畫回原圖中就正確了。

原因出在使用cmath.polar()函式算出來的極座標(r,θ)是代表(x,y)與原點(0,0)的直線距離r和x軸的夾角θ，而在本題實驗中(r,θ)則用來代表距離r的那條直線方程式，但我們不能確定(x,y)在edge圖中的直線就是與原點相連的那條直線，所以我們必須將(x,y)在不同θ下產生出的r值都算出來，也就是算出(x,y)在不同θ產生的直線方程式，最後統計出哪個(r,θ)在edge的點中能夠有最多解(有解為共線)，就是原圖中最長的直線。