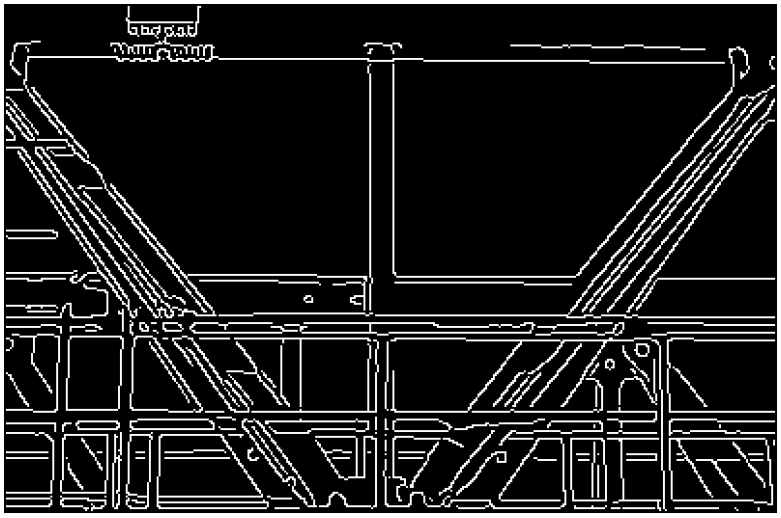
HW6 B063012054 林祐安

Part 1.



↑經內建函數’ edge(image,’Canny’) ’製作之邊緣偵測圖，threshold、sig 預設

(a)

```
function [E, M, A, Gx, Gy, T]=canny(img,sig,TH,TL)
    %先將rgb圖片轉成grayscale
    grayimg=rgb2gray(img);
    %step1:smooth the input image with a Gaussian filter
    %sig是題目要求的輸入參數
    H_1=fspecial('gaussian',[3,3],sig);
    G_img=imfilter(grayimg,H_1);
    %step2:compute the gradient components along x and y with sobel filters
    %原本打算用[1 0 -1;1 0 -1;1 0 -1]以及[-1 -1 -1;0 0 0;1 1 1]
    %但大部分資料都有指名使用sobel
    sobel_x=[1 0 -1;2 0 -2;1 0 -1];
    sobel_y=[-1 -2 -1;0 0 0;1 2 1];
    Gx=filter2(sobel_x,G_img,'same');
    Gy=filter2(sobel_y,G_img,'same');
    %calculate the gradient magnitude after using smoothing(Gaussian) filter
    M=sqrt(Gx.^2+Gy.^2);
    %calculate the gradient angle
    A=180*atan(Gy./Gx)./pi;
    [m,n]=size(A);
    %create matrix E which refers to the edge map after non-maximum suppression
    E=zeros(m,n);
```

```
        for jj=1:m
            for kk=1:n
                %先判斷角度是否有負值，將其限制於0度~180度
                if A(jj,kk)<0
                A(jj,kk)=A(jj,kk)+180;
                end
                %角度歸類
                if A(jj,kk)>=22.5 && A(jj,kk) <67.5
                    A(jj,kk)=45;
                elseif A(jj,kk)>=67.5 && A(jj,kk) <112.5
                    A(jj,kk)=90;
                elseif A(jj,kk)>=112.5 && A(jj,kk) <157.5
                    A(jj,kk)=135;
                else
                    A(jj,kk)=0;
                end

            end
        end
        for jj=2:m-1
            for kk=2:n-1
                if  A(jj,kk)==0
                    %如果歸類後的角度=0 中心值與水平方向最大值相比
                    E(jj,kk) = M(jj,kk) == max([M(jj,kk), M(jj,kk+1), M(jj,kk-1)]);
                elseif A(jj,kk)==45

              %如果歸類後的角度=45 中心值與右下 左上方向最大值相比
              E(jj,kk) = M(jj,kk) == max([M(jj,kk), M(jj+1,kk-1), M(jj-1,kk+1)]);
          elseif A(jj,kk)==135
              %如果歸類後的角度=135 中心值與右上 左下方向最大值相比
              E(jj,kk) = M(jj,kk) == max([M(jj,kk), M(jj+1,kk+1), M(jj-1,kk-1)]);
          else
              %如果歸類後的角度=90 中心值與垂直方向最大值相比
              E(jj,kk) = M(jj,kk) == max([M(jj,kk), M(jj+1,kk), M(jj-1,kk)]);
          end
    end
end
%將原本計算完的梯度強度經非極大值抑制
E=E.*M;
%套用雙閥值
T=zeros(m,n);
for jj=2:m-1
    for kk=2:n-1
        %若該點<TL,該點不為邊緣，最後成品圖為二值化圖，因此設為0
        if E(jj,kk)<TL
            T(jj,kk)=0;
        %若該點>TH,該點為邊緣，最後成品圖為二值化圖，因此設為1
        elseif E(jj,kk)>TH
            T(jj,kk)=1;
        %若不幸處於兩閥值間，判斷周圍八個點是否有點高過TH，若有則為邊界，i.e. 該點設為=1
```

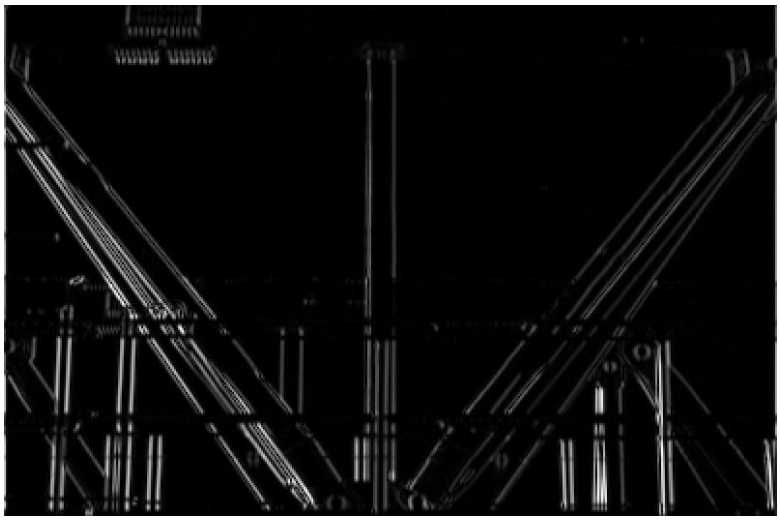
```
            if E(jj,kk)<TL
                T(jj,kk)=0;
            %若該點>TH,該點為邊緣，最後成品圖為二值化圖，因此設為1
            elseif E(jj,kk)>TH
                T(jj,kk)=1;
            %若不幸處於兩閾值間，判斷周圍八個點是否有點高過TH，若有則為邊界，i.e. 該點設為=1
            elseif E(jj,kk+1)>TH || E(jj-1,kk+1)>TH || E(jj-1,kk)>TH || E(jj-1,kk-1)>TH || E(jj,kk-1)>TH || E(jj+1,kk-1)>TH || E(jj+1,kk)>TH || E(jj+1,kk+1)>TH
                T(jj,kk)=1;
            end
        end
    end

end
```

整個函數步驟依序為:　(該處圖片使用 sig=0.5,TH=100,TL=50)

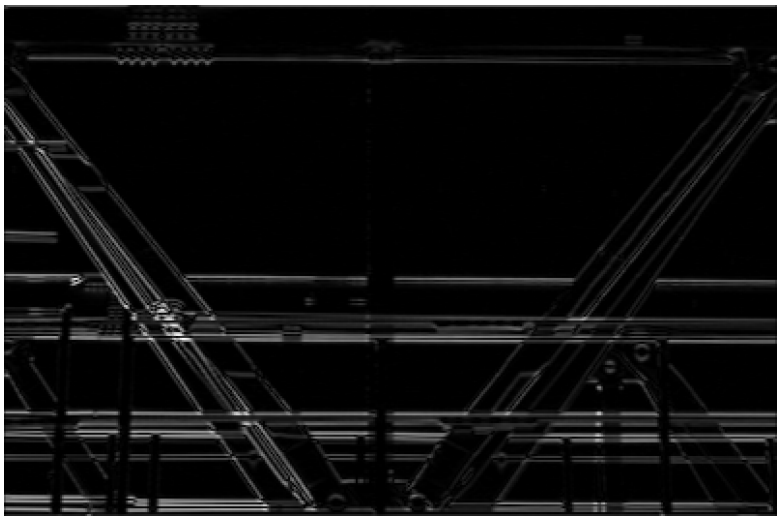
(i)　彩圖轉灰階，並通過高斯低通濾波器

(ii)　經過 sobel 找尋 x,y 方向的梯度 Gx、Gy，顯示的圖片為經過絕對值處理|Gx|、|Gy|

Gx --gradient components along x



Gy --gradient components along y

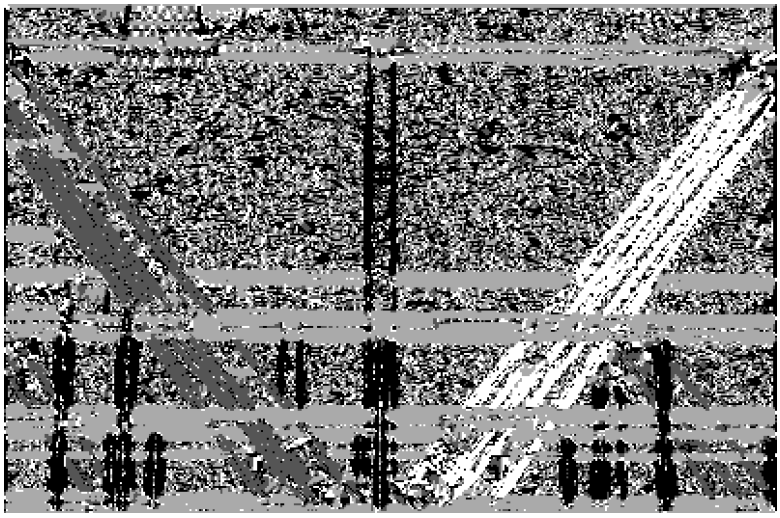


(iii)　計算梯度強度 M 與梯度方向 A

M --smoothed gradient magnitude

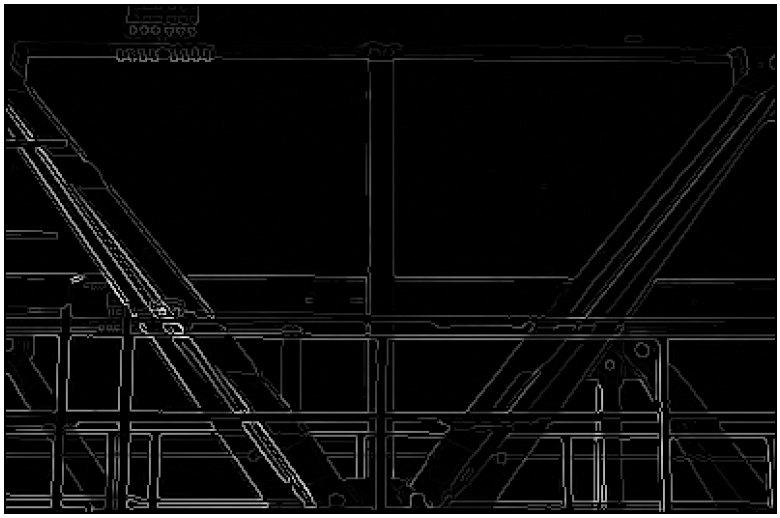
(iv) 歸類梯度方向 A，4 值化:0 45 90 135，而函數輸出的 A 為已分類過角度的矩陣，因此只有四種顏色強度 0 45 90 135

A --gradient angle



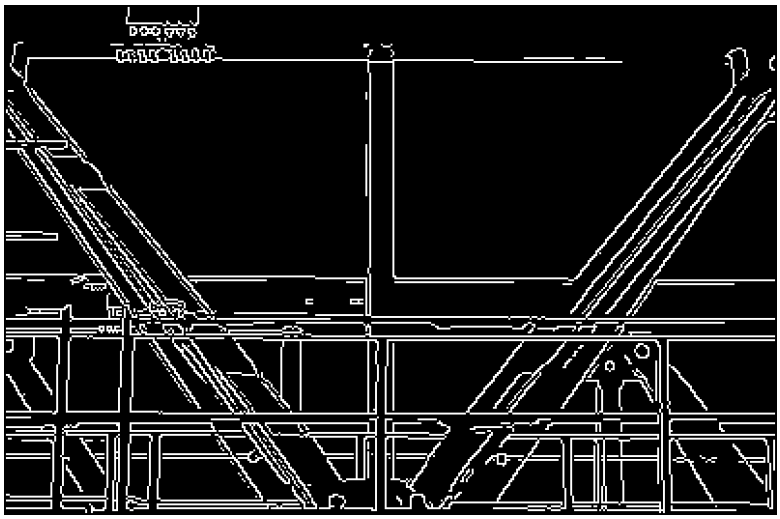
(v) Non-maximum suppression，將梯度強度 M 矩陣依據 Non-maximum suppression 處理成 E 矩陣

E --edge map after non-maximum suppression



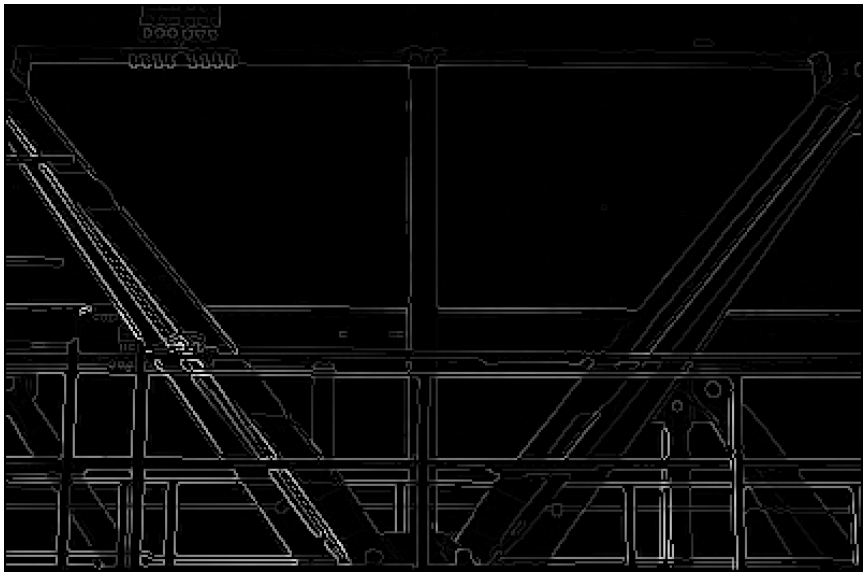
(vi) Bonus 部分，將 E 矩陣經 TH、TL 過濾出邊界，其中 TH 與 TL 是經過前述步驟觀察 E 矩陣內的值而選擇，此為第一次嘗試的數值 TH=100、TL=50

Bonus T --double thresholding on the resulting image

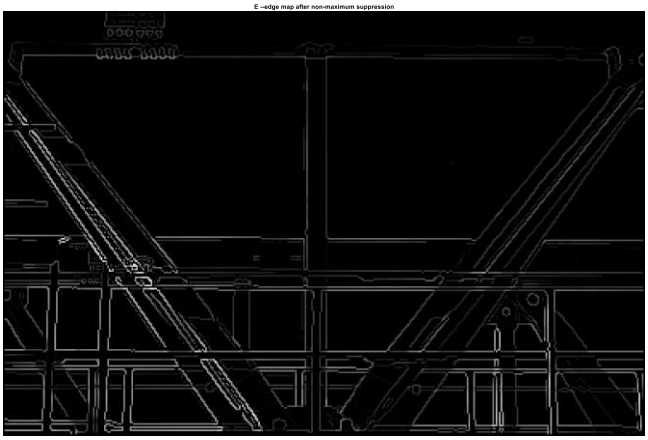


(b)

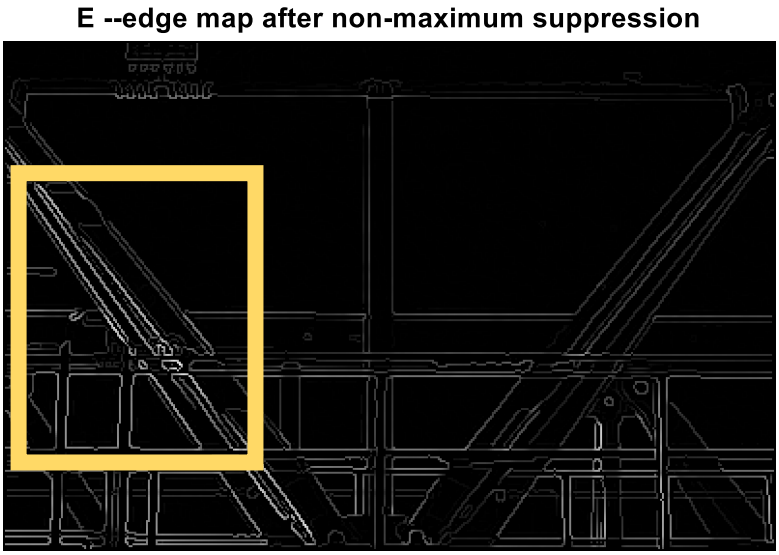
E --edge map after non-maximum suppression



(↑ sig=0.01)



(↑ sig=0.5)



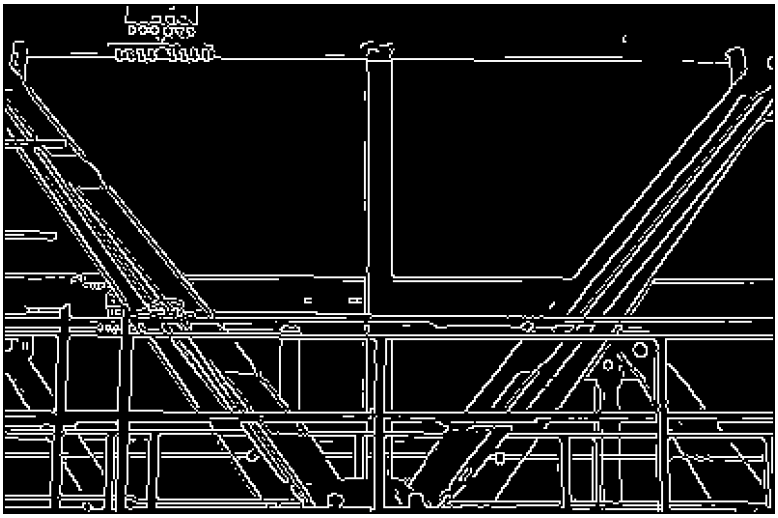
(↑ sig=100)

(sig=0.5) (sig=100)

隨手試了 sig=0.01、0.2、0.5、0.8、100，因圖片眾多僅放上兩組特別極端的做比較，主要著重在左下角吊臂處，在 sig=100 時，比起 sig=0.5、0.1，吊臂內部的線條較少條，放大看其他較細節的部分，也是 sig 小時較多線條。這可能是因為 sig 愈大使得模糊程度愈高，因此對於後續偵測邊界時，模糊的影像較難以尋找邊界。

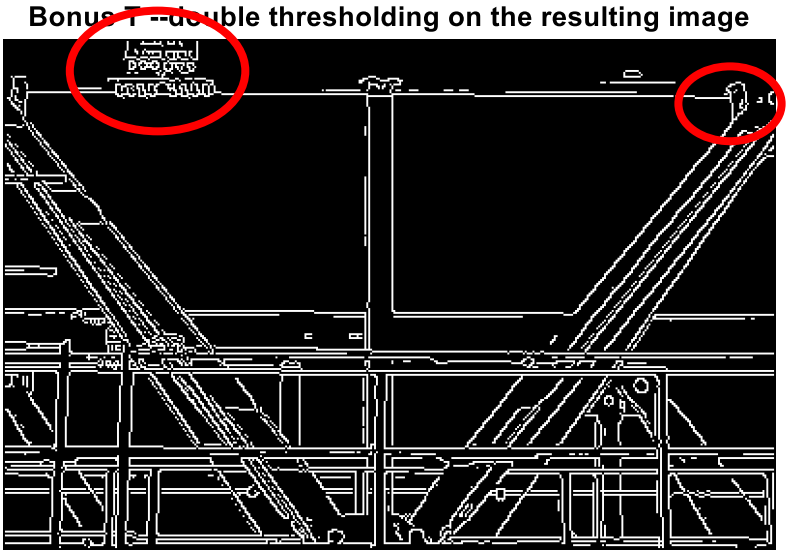
(c)

Bonus T --double thresholding on the resulting image

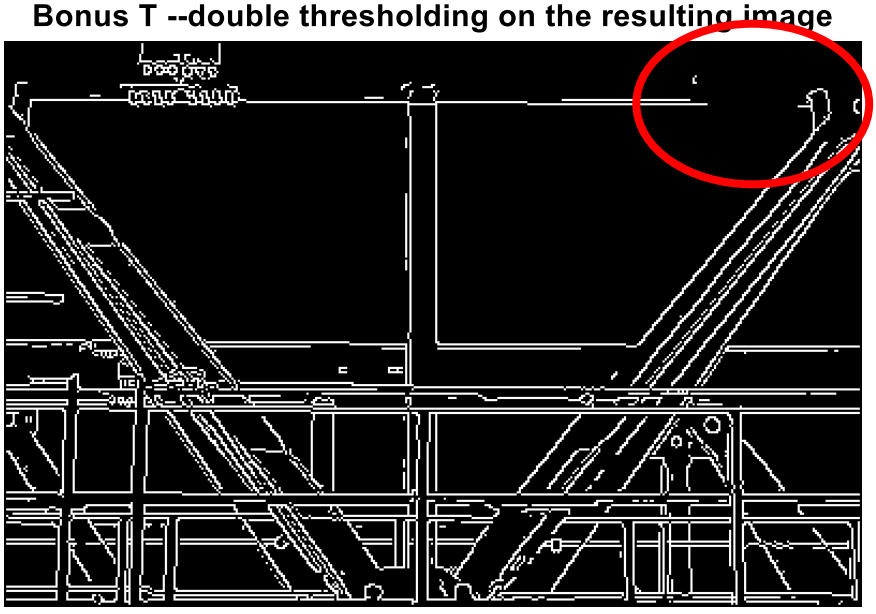


(↑根據(b)，於 TH=100、TL=10 時，sig = 0.15 效果最好)

以下固定 sig=0.15，調整 TH、TL，效果好不好則與用內建函數 edge 所作相比



(↑sig=0.15，TH=80、TL=30 時，左右上角多了奇怪的點點)

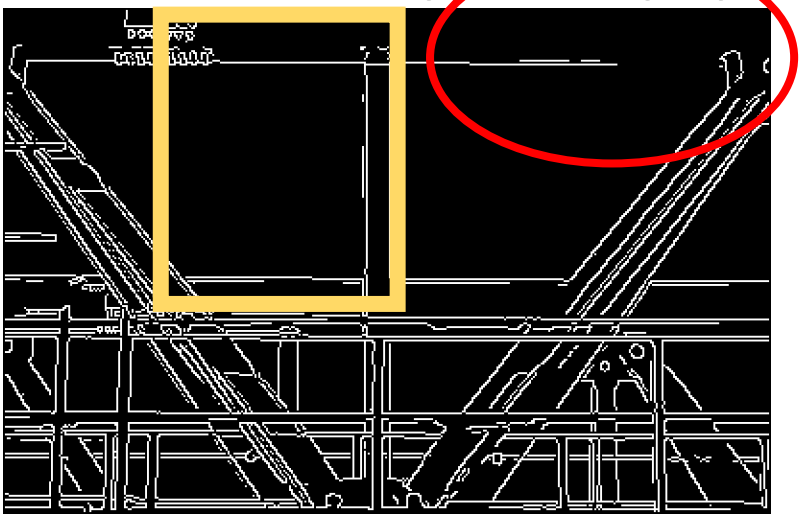


(↑sig=0.1，TH=106、TL=20 時)

消除了原本左右上角多餘的點點，但換成紅圈處無法密合….，為了不犧牲其他更多地方的資訊，選用此組參數。

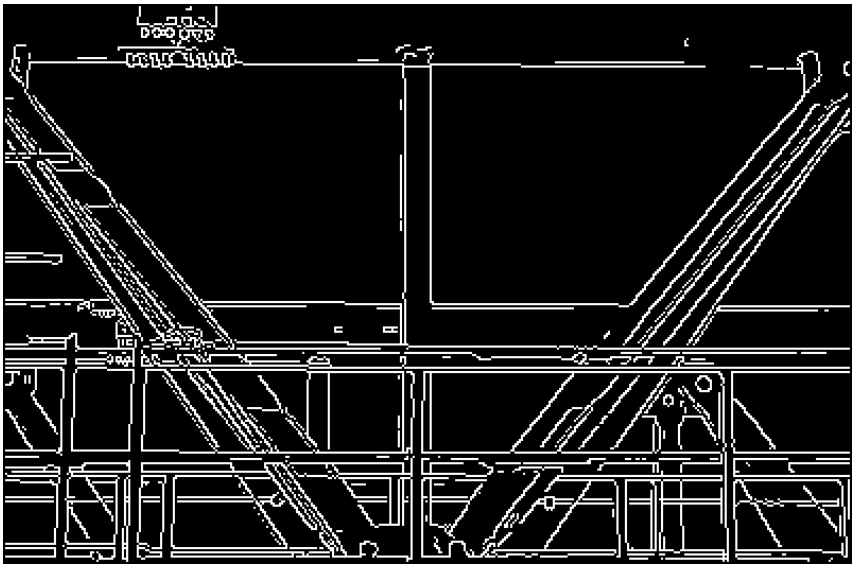
```
figure,imshow(canny_img,[]);
[E, M, A, Gx, Gy, T]=canny(img,0.1,106,20);
figure,imshow(E,[]);
```

Bonus T --double thresholding on the resulting image

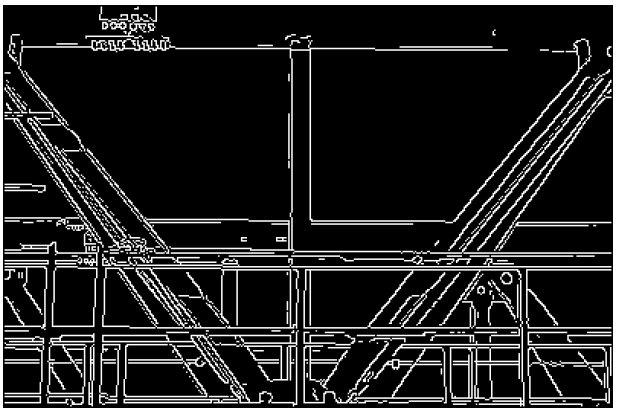


比較大的問題是將圖片使用 copy figure 貼到 word 時，word 會因為圖片縮放大小而損失某些細節，例如黃框處。

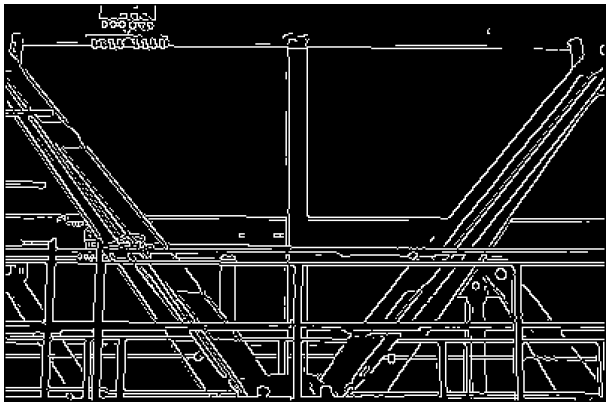
Bonus T --double thresholding on the resulting image



Bonus T --double thresholding on the resulting image



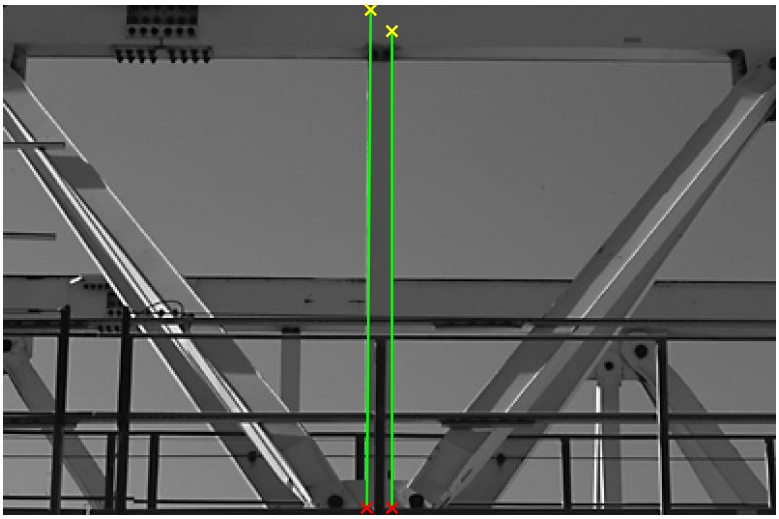
Bonus T --double thresholding on the resulting image



(↑以上三張圖都是同一張圖片 sig=0.15，但我在 word 做了不同縮放，線條的呈現大不相同)

Part2.

sig 採用 1-(b)，sig=0.15



```
%因為中心桿子是垂直線，考慮柱子可能不是完全垂直，需要找theta很接近0的地方
[H,T,R]=hough(E,'THETA',0:0.5:2);
figure
imshow(H,[],'XData',T,'YData',R,'InitialMagnification','fit');
title('\rho\theta-plane');
xlabel('\theta'), ylabel('\rho');
axis on, axis normal, hold on;

%threshold設為0.8*MAX是經過多次測試出來的最佳解
P=houghpeaks(H,5,'threshold',ceil(0.8*max(H(:))));
%找到最大角度
x = T(P(:,2));
%找到最大長度
y = R(P(:,1));
plot(x,y,'s','color','white');

%根據help houghlines資料調整'FillGap' 'MinLength'讓中間的現更連續與更明顯
lines = houghlines(E,T,R,P,'FillGap',40,'MinLength',30);
figure, imshow(grayimg), hold on
max_len = 0;
for k = 1:length(lines)
   xy = [lines(k).point1; lines(k).point2];
   plot(xy(:,1),xy(:,2),'LineWidth',1,'Color','green');

   % Plot beginnings and ends of lines
   plot(xy(1,1),xy(1,2),'x','LineWidth',1,'Color','yellow');
   plot(xy(2,1),xy(2,2),'x','LineWidth',1,'Color','red');

   % Determine the endpoints of the longest line segment
   len = norm(lines(k).point1 - lines(k).point2);
   if ( len > max_len)
      max_len = len;
      xy_long = xy;
   end
end
```