4104I5069 電機三 陳毅軒

HW3.1 Hough Transform

for all θ, $x\cos\theta + y\sin\theta$

(x,y) \ θ	-45°	0°	45°	90°
(2,-3)	3.54	2	-0.71	-3
(0,-1)	0.71	0	-0.71	-1
(2,-1)	2.12	2	0.71	-1
(-1,0)	-0.71	-1	-0.71	0
(-3,2)	-3.54	-3	-0.71	2
(2,2)	0	2	2.83	2

	-3.54	-3	-1	-0.71	0	0.71	2	2.12	2.83	3.54
-45°	1			1	1	1		1		1
0°		1	1		1		3			
45°				4		1			1	
90°		1	2		1		2			

最大值出現在 $x\cos45° + y\sin45° = -0.71$

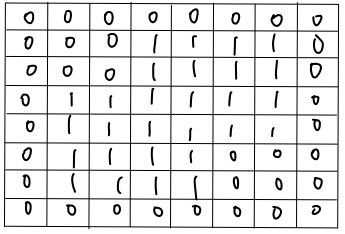
$$\Rightarrow y = -x - \frac{0.71\times 2}{\sqrt{2}} = -x - 1.004$$

$\Rightarrow y = -x - 1$，過 $(0,-1)$，$m = -1$

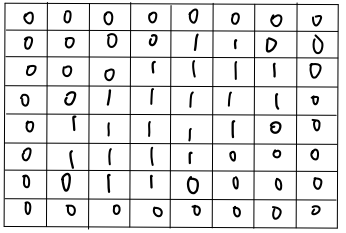
	−3	−2	−1	0	1	2	3
-3	0	0	0	0	0	0	0
-2	0	0	0	0	0	0	0
-1	0	1	0	1	0	1	0
0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0
2	1	0	0	0	0	1	0
3	0	0	0	0	0	0	0

$y = -x - 1$

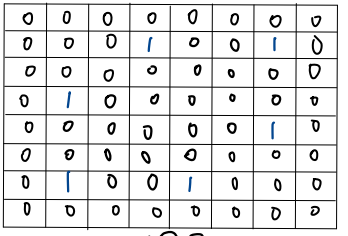
HW3.2 Skeletonization



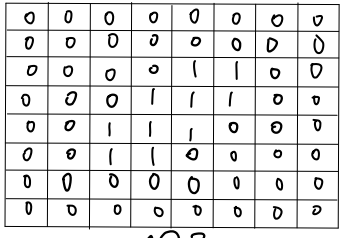
A



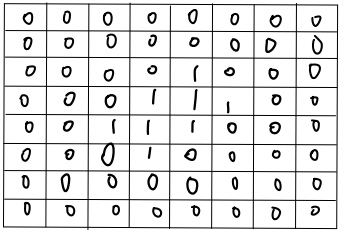
$A \circ B = (A \ominus B) \oplus B$



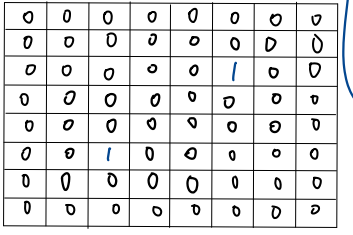
$A \ominus B$



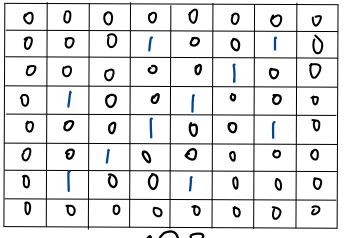
$A \ominus B$



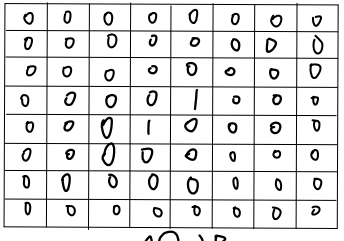
$(A \ominus B) \circ B = (A \ominus 2B) \oplus B$



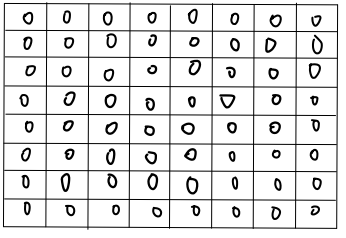
$(A \ominus B) - ((A \ominus B) \circ B)$



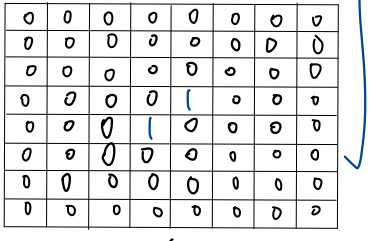
$A \ominus B$



$A \ominus 2B$



$(A \ominus 2B) \circ B = (A \ominus 3B) \oplus B$



$(A \ominus 2B) - ((A \ominus 2B) \circ B)$

H3.3 Hit –or-miss transform

Use the hit or miss transform with appropriate structuring element to find the dot on “I” in the word “Friday” in the image “friday_text.png”.

圖一、friday_text.png原圖

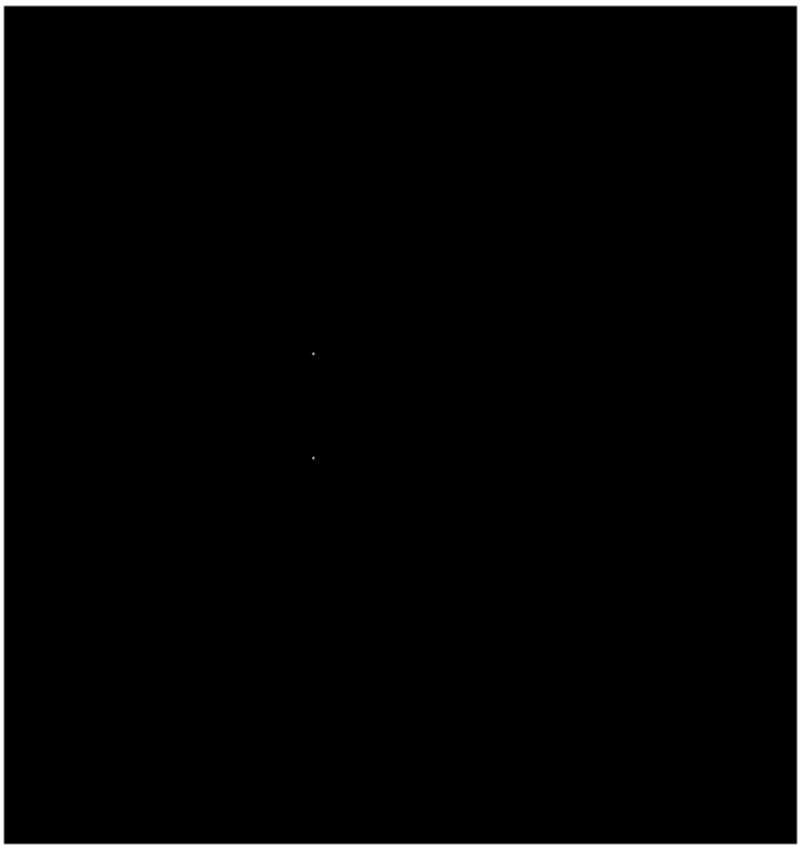
Solution

觀察圖像內的元素，會發現"i"的部分，都是由矩形所組成，因為我不知道"i"上面的點的大小，所以我使用 `for` 迴圈對不同大小的矩形圖像做 `hit-or-miss transform` 。

hit-or-miss transform

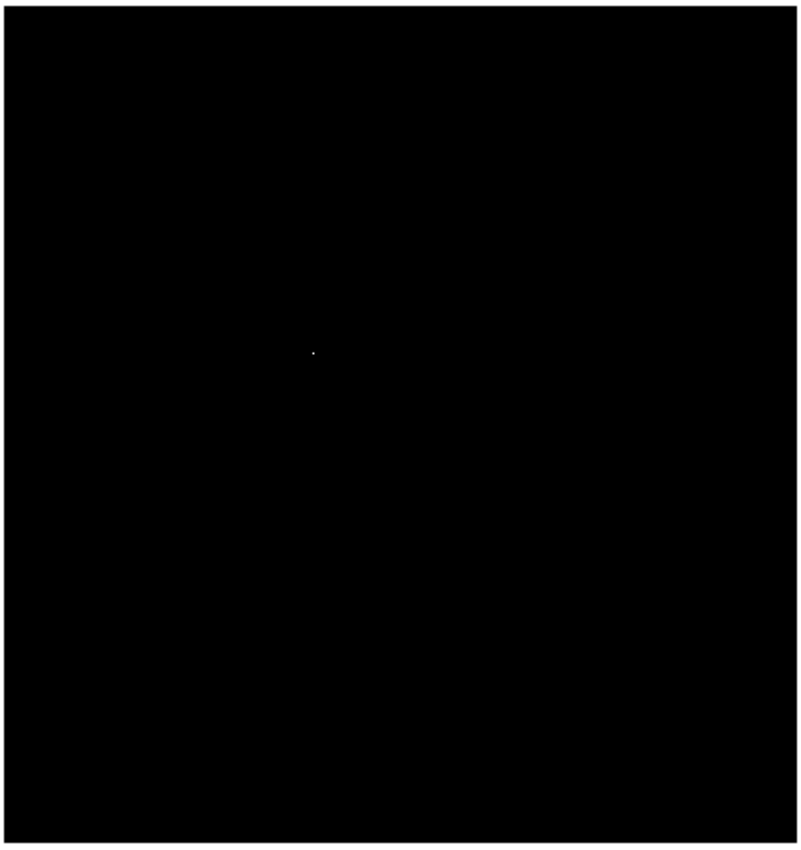
```
hit_or_miss=zeros(len,width);
for xsize=80:-1:10
    for ysize=80:-1:10
        b1=ones(ysize,xsize);
        b2=ones(ysize+2,xsize+2);
        b2(2:ysize+1,2:xsize+1)=~b1;
        tb1=imerode(inputImage,b1);
        tb2=imerode(~inputImage,b2);
        hit_or_miss=hit_or_miss | tb1&tb2;
    end
end
```

如果將大小設定在80* 80 ~ 10* 10的矩形大小， `hit-or-miss transform` 會找到兩個相符的區塊，如圖二所示。



圖二、hit-or-miss transform所找到相符的區域(80* 80 ~ 10* 10)

可以發現找到的兩個點分別是"i"上面的點與下面的矩形。如果將範圍調整到60* 60~10* 10，結果如圖三。



圖三、hit-or-miss transform所找到相符的區域(60* 60 ~ 10* 10)

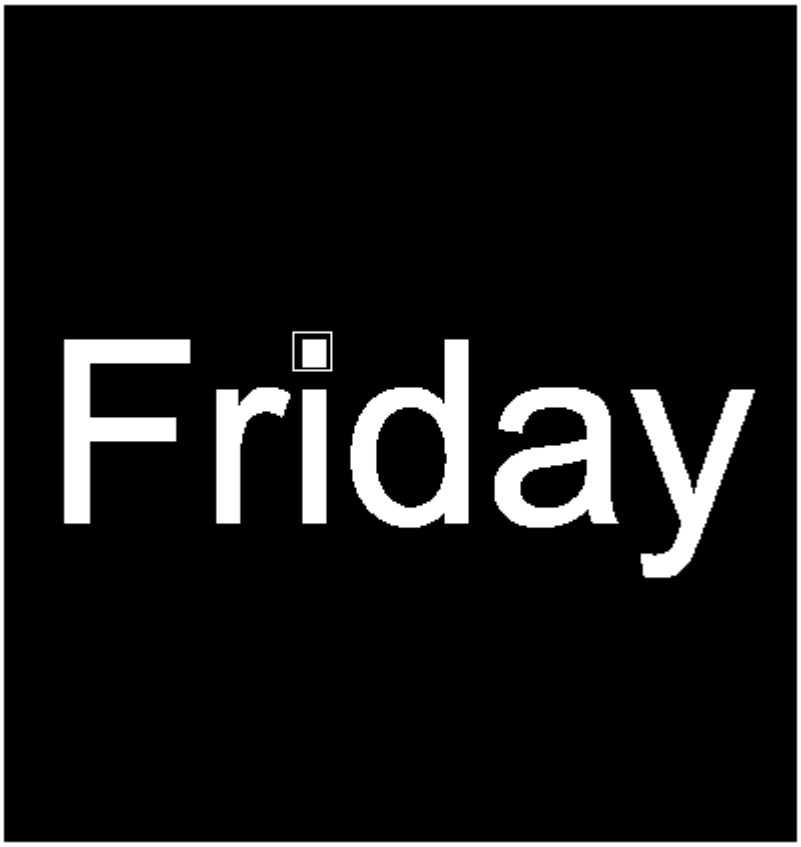
框出"i"上的點

設計一個中空的mask，根據大小罩住所找到對應的點。

```
[x,y]=find(hit_or_miss==1);
frameWidth = 1;
maskxsize=xsize+10;maskysize=ysize+10;
mask = zeros(maskysize, maskxsize);
mask(1:frameWidth, :) = 1; % top
mask(end-frameWidth+1:end, :) = 1; % bottom
mask(:, 1:frameWidth) = 1; % left
mask(:, end-frameWidth+1:end) = 1; % right
copy_image=inputImage;

centerX = x(i);
centerY = y(i);
% 計算mask的最大邊界座標
rowStart = max(ceil(centerX - (maskysize)/2), 1);
rowEnd = min(floor(centerX + (maskysize)/2), size(copy_image, 1));
colStart = max(ceil(centerY - (maskxsize)/2), 1);
colEnd = min(floor(centerY + (maskxsize)/2), size(copy_image, 2));
% 擷取出要框出的部分
targetRegion = copy_image(rowStart:rowStart+maskysize-1
colStart:colStart+maskxsize-1);
%與要擷取的範圍和mask做OR得到覆蓋上mask的圖像
copy_image(rowStart:rowStart+maskysize-1, colStart:colStart+maskxsize-
1)= targetRegion | maskAdjusted;
```

藉由插入mask的方法會得到圖四的結果。

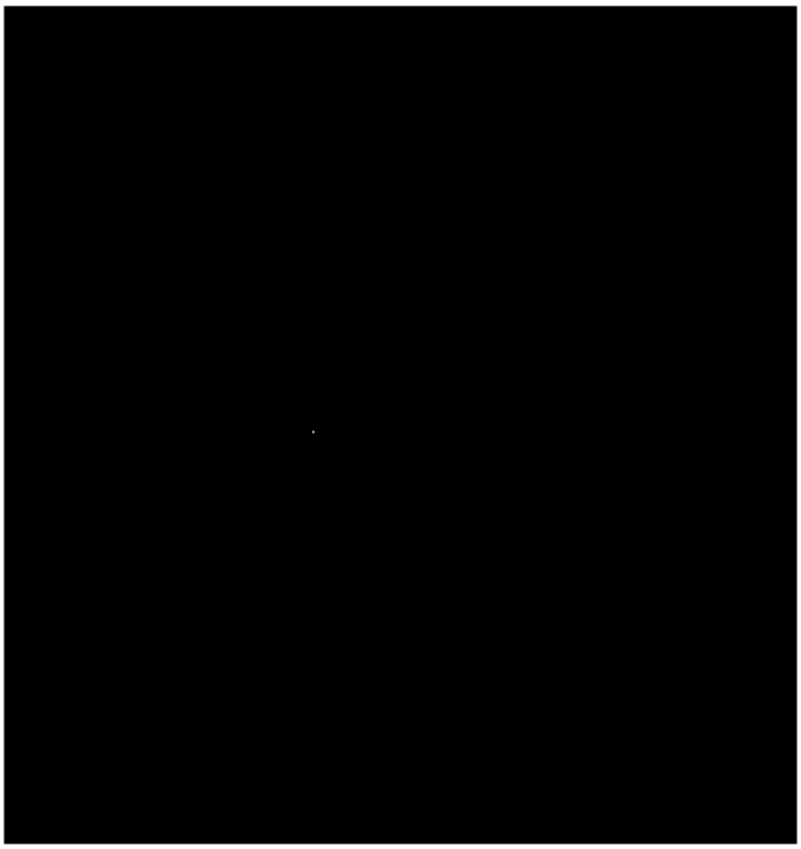


圖四、框出"i"上面點的結果

找出"i"

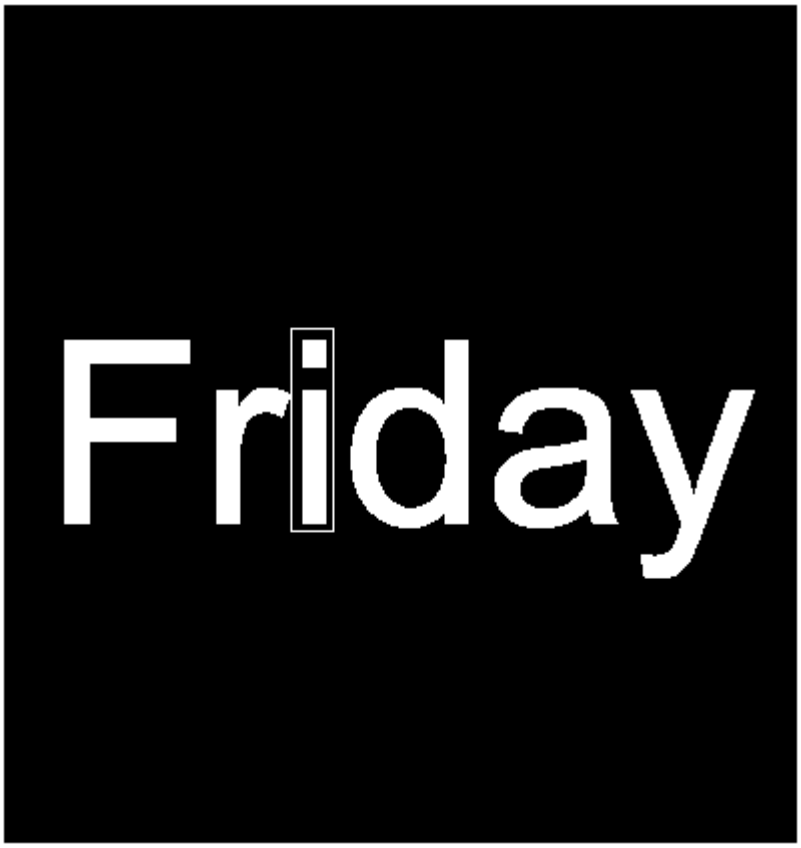
因為"i"本身是一個特定的圖形，所以我將"i"直接描繪出來，做 hit-or-miss transform ，可以得到圖五的結果。

```
xsize=12;ysize=92;
b1=zeros(ysize,xsize);
b1(1:14,:)=1;b1(26:92,:)=1;
b2=ones(ysize+2,xsize+2);
b2(2:ysize+1,2:xsize+1)=~b1;
tb1=imerode(inputImage,b1);
tb2=imerode(~inputImage,b2);
hit_or_miss=tb1&tb2;
```



圖五、hit-or-miss transform直接找"i"

接著在做與上面相同的框線方法，框出"i"。結果如圖六。



圖六、框出"i"

結論

根據第一種方法，找出矩形，可以在不知道圖像大小的情況下做到 `hit-or-miss transform` ，但是當需要框出特定圖形，就需要直接定義圖形的樣子，才能做 `hit-or-miss transform` ，所以要根據所需要找到的特徵，來改變目標的樣式。

未來改善

因為找尋符合大小的矩陣，需要用 `for` 迴圈，所以耗時較久，希望能找到更快收斂的方法，更快速的找到符合目標大小的矩形。以及能找到不用直接畫出"i"的樣子，就能找到"i"的方法。