



## 8. 特徵擷取

### 8.1 點擷取

### 8.2 線擷取

### 8.3 邊擷取

### 8.4 輪廓連結

### 8.5 角擷取



### ❁ 目的

根據影像中的像素灰階不連續性 (discontinuity) 偵測影像中的基本特徵：點 (point)、線 (line)、邊 (edge)、及角 (corner)。

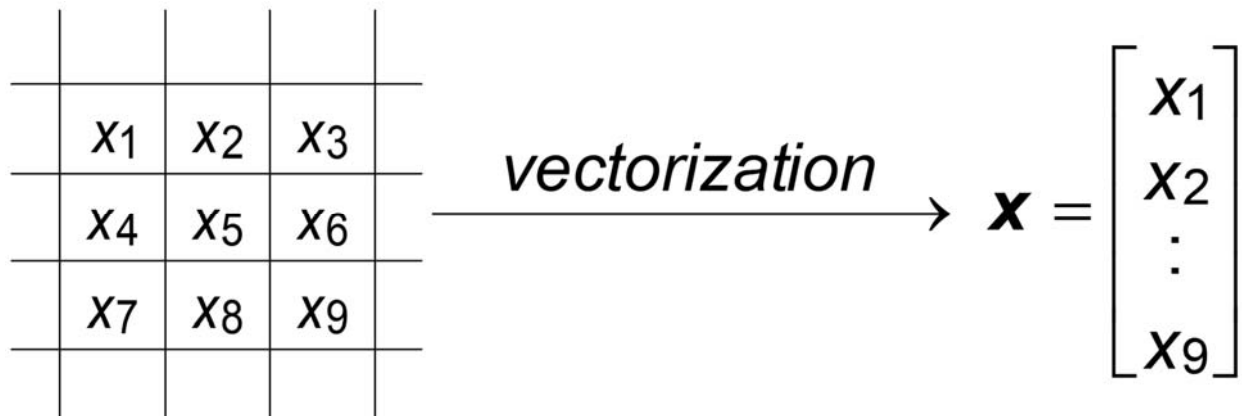
### ❁ 主要的方法是用 遮罩運算 (mask operators)

### ❁ 一個 $3 \times 3$ 遮罩可表示成一個向量

$$\begin{array}{|c|c|c|} \hline W_1 & W_2 & W_3 \\ \hline W_4 & W_5 & W_6 \\ \hline W_7 & W_8 & W_9 \\ \hline \end{array} \xrightarrow{\text{vectorization}} \mathbf{W} = \begin{bmatrix} W_1 \\ W_2 \\ \vdots \\ W_9 \end{bmatrix}$$



一個影像區塊也表示成一個向量，遮罩運算及成的兩個向量的內積 (inner product)



$$\mathbf{w}^T \mathbf{x} = \sum_{i=1}^9 w_i x_i.$$



## 8.1 點擷取

🌸 偵測孤立點 (isolated point) 的遮罩定義為

$$M_p = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

影像做完遮罩運算後，還要用一個準則 (criterion) 根據每一像素的遮罩運算值，來判定哪些像素具有該遮罩所定義的特徵。此準則可以定義為  $|M_p * f| > T_i$ ， $T_i$  是事先定義好的門檻值 (threshold value)， $f$  是影像。



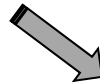
### 範例一

### 影像

0	0	0	0	0
0	0	0	0	0
0	0	100	0	0
0	0	0	0	0
0	0	0	0	0

### 遮罩

-1	-1	-1
-1	8	-1
-1	-1	-1



	-100	-100	-100	
	-100	800	-100	
	-100	-100	-100	



### 範例二

### 影像

0	0	0	0	0
0	0	0	0	0
0	100	100	0	0
0	100	100	0	0
0	0	0	0	0

### 遮罩

-1	-1	-1
-1	8	-1
-1	-1	-1



	-200	-200	-100	
	500	500	-200	
	500	500	-200	



## 8.2 線擷取

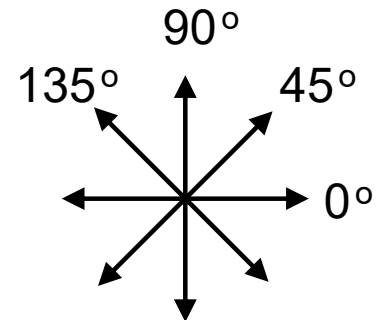
- 線 (line) 有方向性，所以線上點要根據方向來偵測。我們可以只定義  $0^\circ$  (水平)、 $45^\circ$ 、 $90^\circ$  (垂直)、及  $135^\circ$  四個方向來擷取各種方向的線

$$M_{11} = \begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix}$$

$$M_{12} = \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix}$$

$$M_{13} = \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}$$

$$M_{14} = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$



- 準則定義為

$$A > T_1$$

其中  $A = \text{Max} \{ |M_{11} * f|, |M_{12} * f|, |M_{13} * f|, |M_{14} * f| \}$ ， $T_1$  是事先定義好的門檻值。

- 一般化的通式；例如， $M_{11} = c$   
 $c$  是任意常數；  
 其他遮罩依此類推。

$$\begin{bmatrix} -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \\ 1 & 1 & 1 \\ -\frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \end{bmatrix}$$

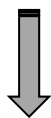
- 問題

霍夫轉換 (Hough transform) 與本節的線擷取有什麼關係或差異？

## ❁ 範例一

## 擷取前

0	0	0	0	0
100	100	100	100	100
0	0	0	0	0
0	0	0	0	0



0	0	0	0	0
100	100	100	100	100
100	100	100	100	100
0	0	0	0	0

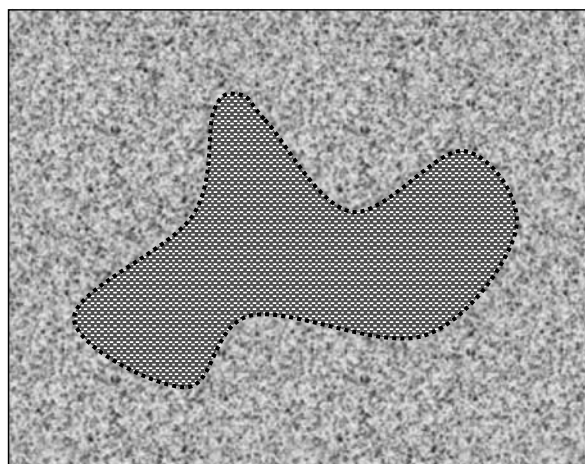
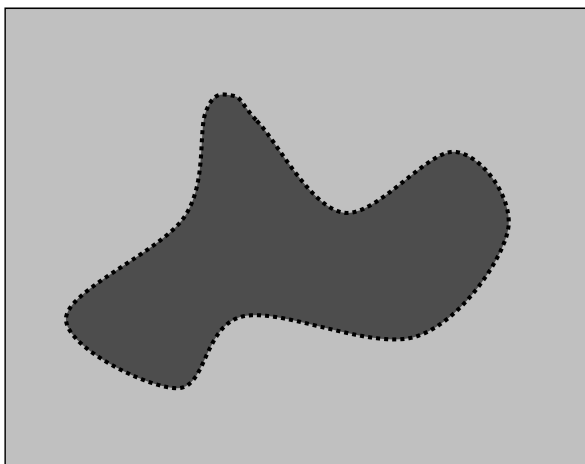
## 擷取後

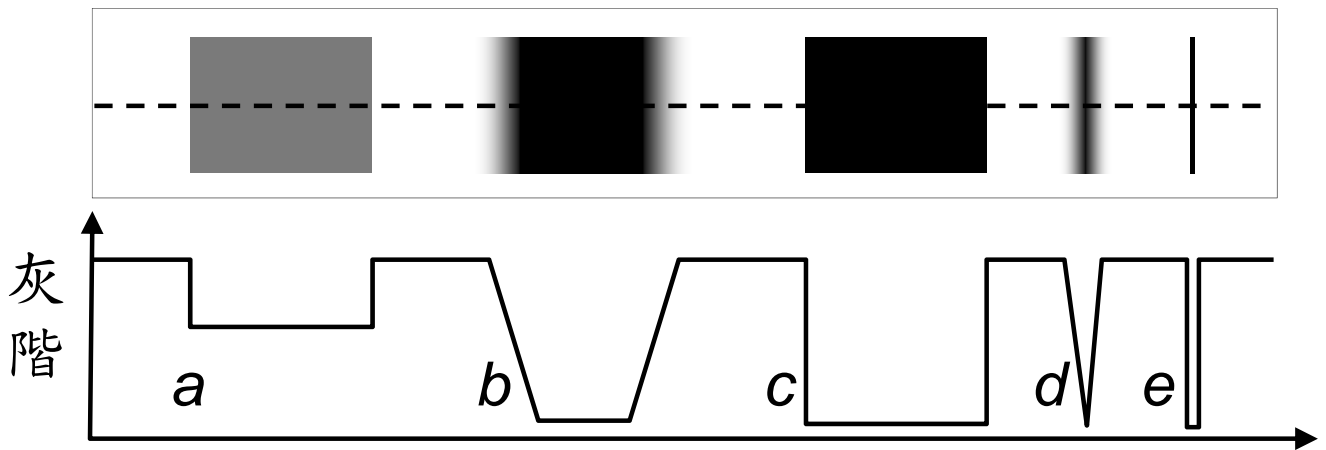
	600	600	600	
	-300	-300	-300	

	300	300	300	
	300	300	300	

## 8.3 邊擷取

- ❁ 在所有特徵中，邊 (edge) 是最重要的。
- ❁ 兩個顯著不同特性區塊之間就稱為邊或邊界。
- ❁ 在所有特徵中，邊是最難精確求得的；原因在於無唯一定義 (模糊) 的問題。





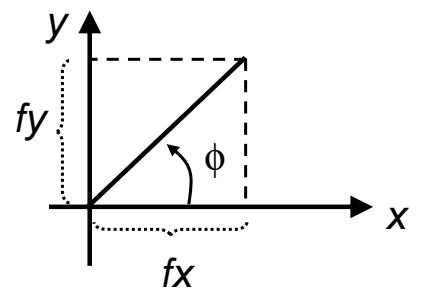
- ✿ 邊是在小區域中灰階變異最大的地方；所以我們大都是用微分 (derivative) 或差分 (differentiation) 來偵測邊。常用的微分有一次微分；例如，梯度 (gradient)，及二次微分；例如，拉普拉斯運算 (Laplacian operator)。



### 8.3.1 一次微分

- ✿ 微分是有方向性的；我們以相鄰灰階在  $x$  及  $y$  方向的微分向量，稱為梯度 (gradient) 表示影像  $f$  在  $(x, y)$  位置的一次微分 (first derivative)，

$$\nabla f = [f_x \ f_y]^T = \left[ \frac{\partial f}{\partial x} \ \frac{\partial f}{\partial y} \right]^T$$



梯度的強度 (magnitude) 定義為

$$|\nabla f| = \sqrt{f_x^2 + f_y^2} \quad \text{或} \quad |\nabla f| \approx |f_x| + |f_y|$$

方向 (direction) 定義為  $\phi(x, y) = \tan^{-1}\left(\frac{f_y}{f_x}\right)$



✿ 準則定義為  $|\nabla f| > T_e$  ,  
其中  $T_e$  是事先定義好的門檻值。

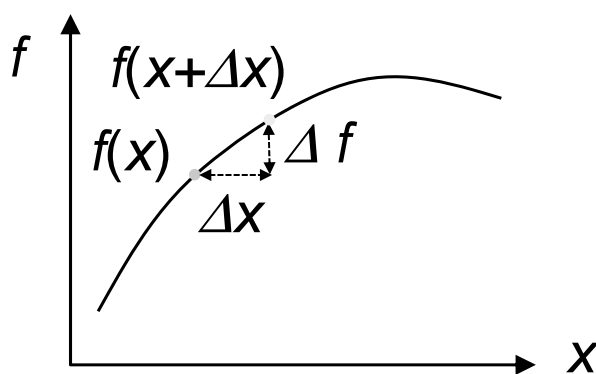
✿ 我們以一次差分 (differentiation) 代替一次微分做離散資料處理。一次差分運算有多種定義

### A. $1 \times 2$ 及 $2 \times 1$ 遮罩

$$f_x = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad f_y = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$



### ✿ Differential of one-dimensional function



$$f_x = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{(x + \Delta x) - (x)} = \lim_{\Delta x \rightarrow 0} \frac{f(x) - f(x - \Delta x)}{(x) - (x - \Delta x)} = \lim_{\Delta x \rightarrow 0} \frac{\Delta f}{\Delta x}$$

in discrete format,  $\Delta x = 1$ ,  $f_x = f(x) - f(x - 1)$



## B. $1 \times 3$ 及 $3 \times 1$ 遮罩

$$f_x = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad f_y = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}$$

## C. 布里威特運算 (Prewitt operator) $3 \times 3$ 遮罩

$$f_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \quad f_y = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$



## G. 梭柏運算 (Sobel operator) $3 \times 3$ 遮罩

$$f_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad f_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

## H. 柯奇運算 (Kirsch operator) $3 \times 3$ 遮罩

$$f_x = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix}, \quad f_y = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$



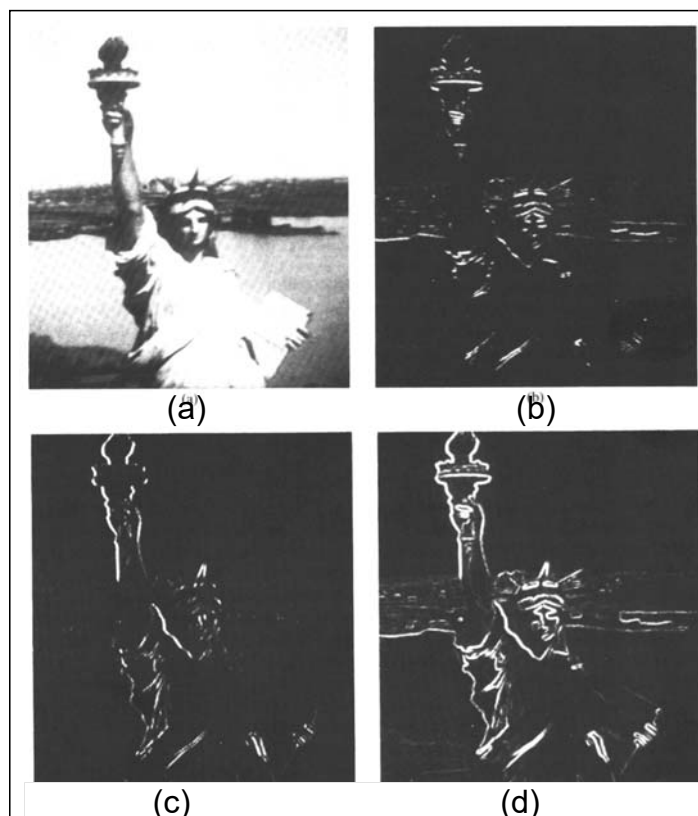


- ❁ 上述布里威特運算及梭柏運算最為一般人使用。這兩個運算有一個很特別的意義，就是這兩個運算在做差分運算前，已先做了一次等加權或不等加權的平均運算，以減少“強化雜訊”。
- ❁ 先做平均(低通濾波)可以消除少許雜訊，以減少做差分(高通濾波)時，將原本不顯著的雜訊給強化出來。梭柏運算與布里威特運算的差別在於做差分前，布里威特運算用等比重的平均法，而梭柏運算改用了類似於高斯分佈的平均法，愈靠近中間點，加權數愈大；所以梭柏運算更能保持影像的邊界特性，比較不會模糊邊界；也因此梭柏運算是最常被使用的邊界偵測運算。



### ❁ 梭柏邊偵測範例

原始影像



$|f_y|$

$|f_x|$

$|f_x| + |f_y|$



## 7.3.2 二次微分

- 二次微分相當於連續兩個一次微分；二次微分沒有方向性。習慣上，我們不將二次微分的結果看成向量，而是直接表示為純量的和 (sum)；也就是兩個方向之“二次微分”和。影像  $f$  在  $(x, y)$  位置的二次微分 (second derivative)

$$D_L f = \Delta f = \nabla \cdot \nabla f = \nabla^2 f = f_{xx} + f_{yy} = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

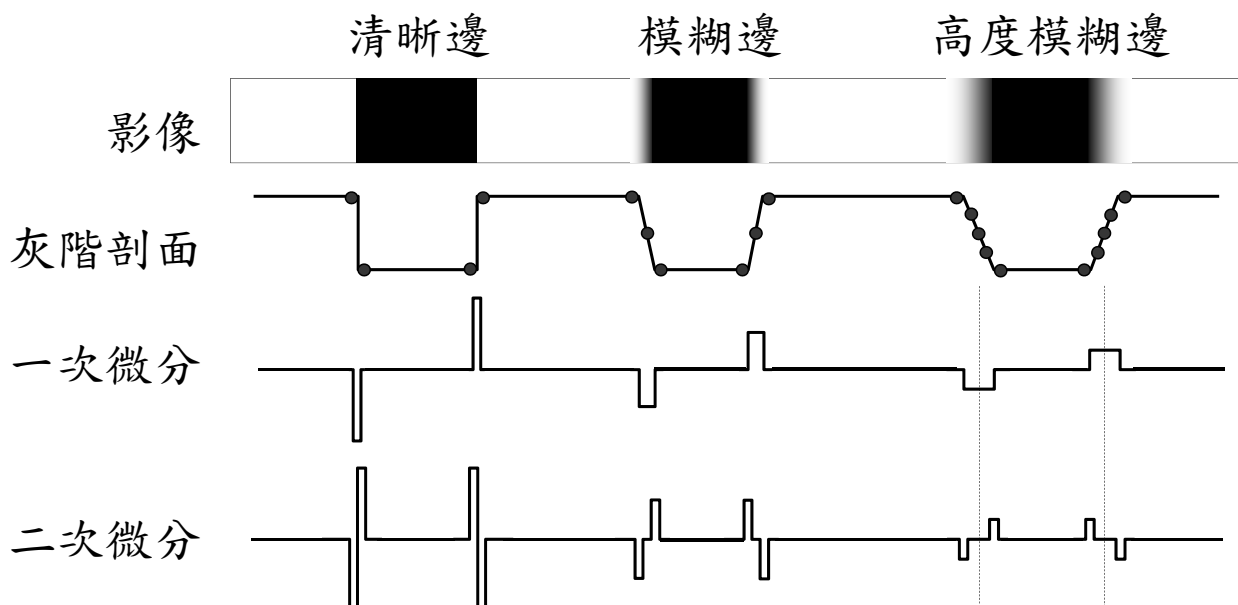
這個運算稱為拉普拉斯運算 (Laplacian operator)

$$D_L = f_{xx} + f_{yy} =$$

0	1	0
1	-4	1
0	1	0



- 為什麼要考慮二次微分？首先比較一次微分與二次微分的差別，再敘述二次微分的使用方式。
- 二次微分會在邊界處產生一個極大值及一個極小值，習慣上我們稱之為**双重邊 (double edge)**。





- ❁ 從極大值跨到極小值或從極小值跨到極大值都會經過 0，而該 0 點剛好就是邊界的最中央位置。所以若是我們在二次微分的結果中尋找跨過 0 的位置，那麼就可以找到最準確的邊界點位置；這個位置就稱為過零點 (zero-crossing)。
- ❁ 另外為了抑制雜訊被強化出來，做二次為先前先做高斯平滑化。二次微分與高斯平滑亦可先結合在一起變成一個運算子，

$$\nabla^2 * (h * f(x, y)) = (\nabla^2 * h) * f(x, y),$$

其中  $h$  為高斯平滑遮罩  $h(x, y) = \exp\left(-\frac{x^2 + y^2}{2\delta^2}\right)$ ,

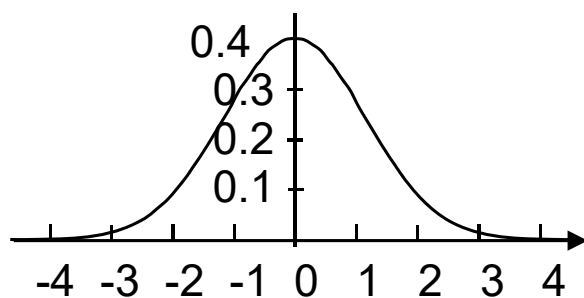
$\delta$  是高斯分佈的標準差 (standard deviation)。



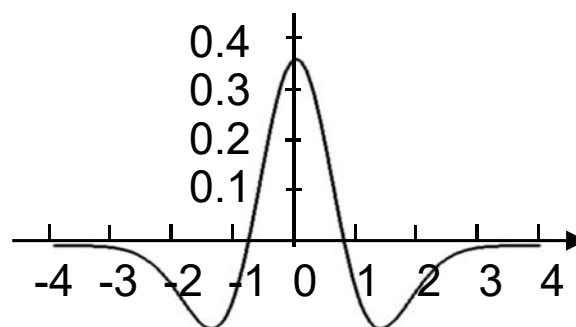
- ❁ 因此二次微分與高斯平滑結合在一起的遮罩變成

$$\nabla^2 h = \left(\frac{r^2 - \delta^2}{\delta^4}\right) \exp\left(-\frac{r^2}{2\delta^2}\right) \quad \text{with } r^2 = x^2 + y^2.$$

- ❁ 下圖是高斯函數與  $\nabla^2 h$  函數的縱切面圖



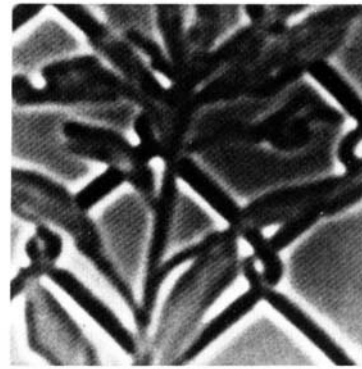
高斯函數



$\nabla^2 h$  函數



## ❁ 二次微分範例

(a)  $f$ .(b)  $\nabla^2 h * f$   
with  $d = 4$ .(c) 以 0 當  
門檻值的二  
值化影像(d) 過零點  
(the zero  
crossings)

### 8.3.3 高斯平滑的邊偵測

❁ 為了抑制雜訊，在微分前常先做平滑化。一個典型的運算是先做高斯平滑化再做一次微分，稱為一次高斯微分 (first derivative of a Gaussian, *FDoG*)；先做高斯平滑化再做二次微分，稱為高斯拉普拉斯 (Laplacian of a Gaussian, *LoG*)。

❁ 另外有一個很有名的高斯平滑邊偵測運算，是影像先各別做兩種不同標準差的高斯平滑化後，再相減；稱為高斯差分 (Difference of Gaussians, *DoG*)。高斯差分 (*DoG*) 的效果幾乎等同於高斯拉普拉斯 (*LoG*)。

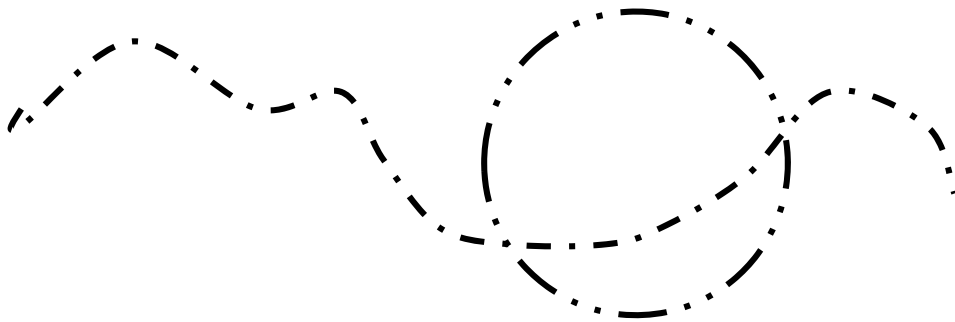


## 8.4 輪廓連結

(Edge linking and boundary detection)

✿ 輪廓連結是探討如何將一個個獨立的像素連結 (link) 成一個比較完整或有意義的圖形；例如，曲線、輪廓、..等。

✿ 連結中最大的問題是斷線 (break) 的問題。



### 8.4.1 以區域特性做連結

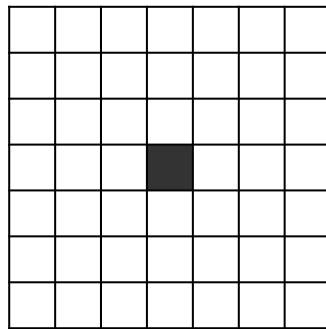
✿ 以區域特性做連結 (edge linking by local analysis) 是比較做完邊偵測之像素的邊強度及方向，若相鄰點的邊強度及方向都相似，即表示這兩像素是同一邊界上的像素，就可以將這兩像素連結。

✿ 若兩鄰近像素為  $(x, y)$  及  $(x', y')$ ，它們各別的梯度為  $\nabla f = [f_x \ f_y]^T$  及  $\nabla f' = [f'_x \ f'_y]^T$ 。

梯度量及梯度方向各別為  $|\nabla f|$ ,  $|\nabla f'|$ ,  $\phi$ , 及  $\phi'$ 。如果  $|\nabla f| - |\nabla f'| \leq T_1$  且  $|\phi - \phi'| \leq T_2$ ，則連結  $(x, y)$ ,  $(x', y')$  像素。



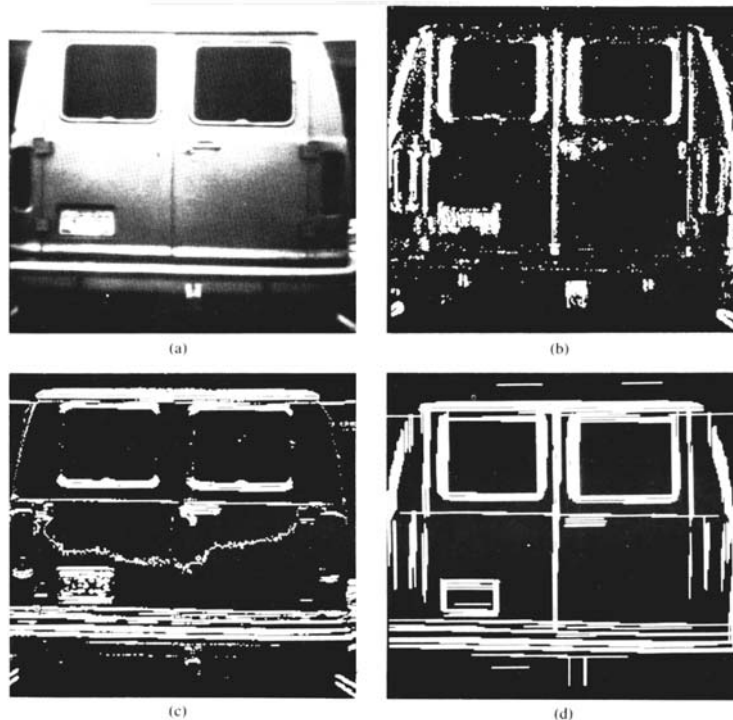
- ❁ 連結的結果受到門檻值  $T_1$  及  $T_2$  的影響；另外影響連結結果的因素還有“鄰近像素”的定義。鄰近像素可以用  $5 \times 5, 7 \times 7, ..$  等範圍來定義；例如，定義  $(x, y)$  為  $n \times n$  區塊的中心，若  $(x', y')$  落在該  $n \times n$  區塊內，則稱此兩像素為鄰近像素。要填補多大的斷線就是要  $n$  來決定的；使用  $n \times n$  區塊可以填補  $(n - 3) / 2$  的斷線。



Using a  $7 \times 7$  mask to link 2-pixel break.



❁ 邊界點連結範例



垂直邊點

水平邊點

連結結果

當相鄰像素的梯度量  $|\nabla f| > 25$  and  $|\nabla f'| > 25$ ，且梯度方向差  $|\phi - \phi'| \leq 15^\circ$ ，則連結這兩像素。



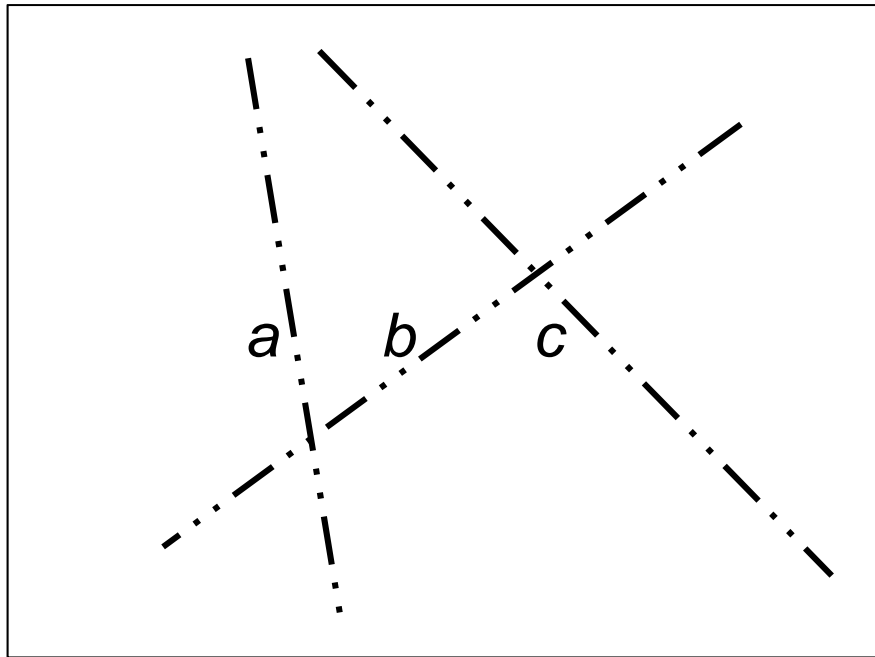
## 8.4.2 以霍夫轉換做連結

❁ 霍夫轉換 (Hough transform) 原本是用來偵測影像中一些圖形的參數；例如，直線的斜率與截距、圓圈的中心座標、橢圓的中心座標及長短軸長度、.. 等。但是霍夫轉換所求出來的參數不是那麼準確；因此我們會想用別的方法求圖形參數。例如用圖形匹配法 (curve fitting) 藉由最小平方誤差估計法估計圖形參數。但是如果影像中有多於一個的圖形，我們要拿哪些點來匹配呢？這時我們將可以先用霍夫轉換，採記名投票的方式，將特徵點分成一群群的，再將一群中的所有特徵點拿出來匹配一個圖形。



### ❁ 演算法

- S1. 將影像做一次微分，根據微分絕對值的大小做二值化。將微分絕對值大於門檻值  $T_1$  的像素定義為邊特徵點，否則就屬於背景點。
- S2. 做霍夫轉換用特徵點投票到各個累積票箱 (accumulator cells) 中。一個像素要累積到一個票箱時，要記錄它的座標  $(x, y)$ 。
- S3. 當所有像素都投完票後，找出票數大於門檻值  $T_2$  的  $n$  個票箱， $n$  個票箱依序執行步驟 S4。
- S4. 將一票箱中之所有票的座標拿出來以最小平方誤差估計法匹配 (fitting) 圖形。



### ❁ 觀念

S1. 用霍夫轉換將邊界點分群。

S2. 將每一群中的所有邊界點拿出來，以最小平方誤差估計法匹配 (fitting) 一直線圖形。



## 8.5 角擷取

❁ 角偵測 (corner detection) 的研究啟程比線、邊的研究稍晚一些，數量也較少；從 1977 年起才開始有較多的研究。以下介紹五種較有名的方法

i. 1988 年 *Harris-Stephens* 角偵測法

ii. 1978 年 *Hessian matrix* 角偵測法

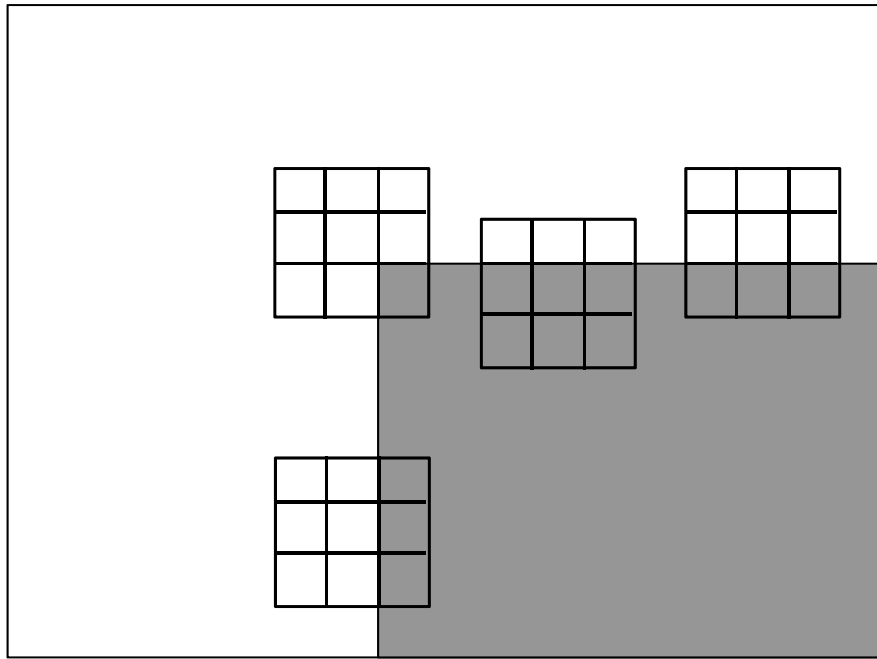
iii. 2004 年 *EoH* 角偵測法

iv. 1997 年 *SUSAN* 角偵測法

v. 2006 年 *FAST* 角偵測器

問題：可以用兩個方向都有邊來判斷有角嗎？





相鄰遮罩  
相減平方和  
計算灰階變異

$$\left( \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} - \begin{array}{|c|c|c|} \hline & & \\ \hline & & \\ \hline & & \\ \hline \end{array} \right)^2 > T$$



❁ Moravec 在1977年提出一個角偵測觀念，認為在各個方向都有灰階變異 (intensity variation) 就是角點 (corner point)。

❁ 演算法

S1. 計算像素  $(x, y)$  在 8 個方向的灰階變異

$$v_{i,j}(x,y) =$$

$$\sum_{a=-1,0,1} \sum_{b=-1,0,1} [f(x+a,y+b) - f(x+a+i,y+b+j)]^2$$

其中  $(i, j) = (1, 0), (1, -1), (0, -1), (-1, -1), (-1, 0), (-1, 1), (0, 1), (1, 1)$  即表示 8 個移動方向。



例如： $a_5$  在右上方的區域灰階變異  $v_{1,1}(a_5)$

$$v_{1,1}(a_5) = \sum_{i=1}^9 [f(a_i) - f(b_i)]^2$$

		$b_1$	$b_2$	$b_3$	
	$a_1$	$b_4$ $a_2$	$b_5$ $a_3$	$b_6$	
	$a_4$	$b_7$ $a_5$	$b_8$ $a_6$	$b_9$	
	$a_7$	$a_8$	$a_9$		



S2. 構建角圖 (cornerness map)  $c(x, y)$ ,

$$c(x, y) = \min \{v_{i,j}(x, y) \text{ for all } (i, j)\}$$

S3. 若  $c(x, y) < T_c$ , 則  $c(x, y) = 0$ .

S4. 若  $c(x, y)$  在以  $(x, y)$  為中心的  $3 \times 3$  範圍內不是最大值, 則  $c(x, y) = 0$ ; 這個處理稱為抑制非最大 (non-maximum suppression)

S5. 所有留下來的非 0 點即為角點 (corner point)

。



## 8.5.1 Harris corner detection

✿ Harris – Stephens 在 1988 年改進 Moravec 角偵測法，定義區域灰階變異為

$$c(x, y) = \sum_{i=1}^m \sum_{j=1}^n [f(i, j) - f(i + \Delta x, j + \Delta y)]^2 \quad (1)$$

其中位移影像  $f(i + \Delta x, j + \Delta y)$  可以用泰勒展開式 (Taylor expansion) 取第一項近似

$$f(i + \Delta x, j + \Delta y) \approx f(i, j) + [f_x(i, j) \ f_y(i, j)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (2)$$

將公式 (2) 代入公式 (1) 中



✿ Taylor series of one-dimensional function  
If  $f$  function is analytic at  $x_0$ , then  $f$  can be expended as a series

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + f''(x_0) \frac{(x - x_0)^2}{2!} + f'''(x_0) \frac{(x - x_0)^3}{3!} + \dots$$

or

$$f(x + \Delta x) = f(x) + f'(x)(\Delta x) + f''(x) \frac{(\Delta x)^2}{2!} + f'''(x) \frac{(\Delta x)^3}{3!} + \dots$$



$$\begin{aligned}
c(x,y) &= \sum_{i=1}^m \sum_{j=1}^n [f(i,j) - f(i+\Delta x, j+\Delta y)]^2 \\
&= \sum_{i=1}^m \sum_{j=1}^n \left( -[f_x(i,j) \quad f_y(i,j)] \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 \\
&= [\Delta x \quad \Delta y] \begin{bmatrix} \sum_{i=1}^m \sum_{j=1}^n f_x^2(i,j) & \sum_{i=1}^m \sum_{j=1}^n (f_x(i,j) f_y(i,j)) \\ \sum_{i=1}^m \sum_{j=1}^n (f_x(i,j) f_y(i,j)) & \sum_{i=1}^m \sum_{j=1}^n f_y^2(i,j) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \\
&= [\Delta x \quad \Delta y] \mathbf{C}_x \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}
\end{aligned}$$

$$(\mathbf{Ax})^2 = \mathbf{x}^T \mathbf{A}^T \mathbf{A} \mathbf{x}$$



其中  $\mathbf{C}_x$  是以像素  $(x, y)$  為中心之遮罩  $(m, n)$  內的所有像素之垂直邊與水平邊強度的共變異數矩陣 (covariance matrix)

$$\mathbf{C}_x = \begin{bmatrix} \sum_{i=1}^m \sum_{j=1}^n f_x^2(i,j) & \sum_{i=1}^m \sum_{j=1}^n (f_x(i,j) f_y(i,j)) \\ \sum_{i=1}^m \sum_{j=1}^n (f_x(i,j) f_y(i,j)) & \sum_{i=1}^m \sum_{j=1}^n f_y^2(i,j) \end{bmatrix}$$

令  $\lambda_1, \lambda_2$  為  $\mathbf{C}_x$  的兩個固有值 (eigenvalue)。共變異數矩陣 ( $\mathbf{A}^T \mathbf{A}$  type) 的固有值一定大於等於 0。此固有值描述兩個垂直方向的邊強度之分佈程度；固有值愈大表示邊強度愈大。



$$\begin{aligned}c(x,y) &= [\Delta x \ \Delta y] \mathbf{C}_x \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \\ &= [\Delta x \ \Delta y] \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} e_{11} & e_{12} \\ e_{21} & e_{22} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \\ &= \lambda_1 \rho_{e_1}^2 + \lambda_2 \rho_{e_2}^2,\end{aligned}$$

其中  $\rho_{e_1}$  及  $\rho_{e_2}$  是  $[\Delta x \ \Delta y]^T$  正交投影到固有向量  $\mathbf{e}_1$  及  $\mathbf{e}_2$  上的投影分量或轉換到固有向量  $\mathbf{e}_1$  及  $\mathbf{e}_2$  座標的分量。

若  $[\Delta x \ \Delta y]^T$  是單位長度，則  $\rho_{e_1}$  及  $\rho_{e_2}$  可以表示為  $\sin \theta$  及  $\cos \theta$ ，

$$c(x,y) = \lambda_1 \sin^2 \theta + \lambda_2 \cos^2 \theta.$$



上述的固有值  $\lambda_1, \lambda_2$  愈大，則區域灰階變異愈顯著；因此我們可以用固有值的大小判定角的顯著性：

- i.  $\lambda_1, \lambda_2$  都很小，則  $(x, y)$  附近沒有任何特徵；
- ii.  $\lambda_1, \lambda_2$  一大一小，則  $(x, y)$  附近只有邊特徵；
- iii.  $\lambda_1, \lambda_2$  都比較大，則  $(x, y)$  附近有角的特徵。



計算固有值比較耗時，因此 Harris & Stephens 制訂一個函數  $R$  代替用固有值判定有無角特徵

$$R = \lambda_1 \lambda_2 - \kappa (\lambda_1 + \lambda_2)^2 = \det(\mathbf{C}_x) - \kappa \text{trace}^2(\mathbf{C}_x)$$

其中  $\det(\mathbf{C}_x)$  是  $\mathbf{C}_x$  矩陣的行列式、 $\text{trace}(\mathbf{C}_x)$  是  $\mathbf{C}_x$  矩陣的對角線元素相加、 $\kappa$  是敏感參數 (sensitivity parameter)，用以控制偵測率及偵測錯誤率；Harris and Stephens 在他們的研究中， $\kappa = 0.04 - 0.15$ 。判斷角點的準則為：

$$\begin{cases} R > 0, & \text{則}(x,y) \text{是角點} \\ R = 0, & \text{則}(x,y) \text{不是特徵點} \\ R < 0, & \text{則}(x,y) \text{是邊界點} \end{cases}$$

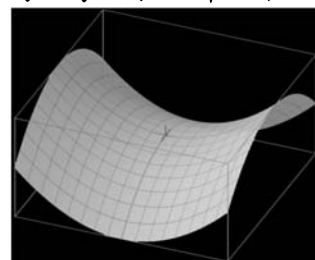


## 8.5.2 Hessian matrix 角偵測法

❁ Beaudet 在 1978 年提出以二次微分的黑森矩陣 (Hessian matrix) 之行列式求影像中的角點或馬鞍點 (saddle point) 方法；黑森矩陣的意義相當於是描述一個區域的灰階曲率 (curvature)。

❁ 以  $(x, y)$  點為中心的  $n \times n$  區域內的所有像素計算

$$\Delta_{\text{Beaudet}} = \left| \begin{bmatrix} \sum f_{xx} & \sum f_{xy} \\ \sum f_{yx} & \sum f_{yy} \end{bmatrix} \right|$$



若  $\Delta_{\text{Beaudet}} > T_b$ ，則  $(x, y)$  是一個角點或馬鞍點。Beaudet 的方法還算穩定，但對於過銳利的角點偵測效果並不好。



### 8.5.3 EoH 角偵測法

- ✿ EoH 是 Eigenvalue of Hessian matrix 的縮寫；是 Lowe 於 2004 年在 SIFT 論文中所提出的方法。Lowe 最後也是導出黑森矩陣，再以黑森矩陣的固有值來進一步篩選徵點。

$$H = \begin{bmatrix} \sum f_{xx} & \sum f_{xy} \\ \sum f_{yx} & \sum f_{yy} \end{bmatrix}$$

- ✿ 令  $\lambda_1, \lambda_2$  是黑森矩陣兩個固有值 (eigenvalue)。該固有值與影像區塊的灰階曲率成正比；但黑森矩陣的固有值不一定大於 0，所以判定方式與 Harris-Stephens 的不一樣。



#### ✿ 定義

$$Tr H = \sum f_{xx} + \sum f_{yy} = \lambda_1 + \lambda_2$$

$$Det H = \sum f_{xx} \sum f_{yy} - (\sum f_{xy})^2 = \lambda_1 \lambda_2.$$

若  $Det H < 0$ ，表示兩個固有值的正負號不同，也就是像素  $(x, y)$  是反曲點不是極點，丟棄。令

$$r = \begin{cases} \frac{\lambda_1}{\lambda_2}, & \text{if } |\lambda_1| \geq |\lambda_2| \\ \frac{\lambda_2}{\lambda_1}, & \text{if } |\lambda_1| < |\lambda_2| \end{cases}, \quad |r| \geq 1$$

$$\text{則 } \frac{(Tr H)^2}{Det H} = \frac{(\lambda_1 + \lambda_2)^2}{\lambda_1 \lambda_2} = \frac{(1+r)^2}{r}$$



$r \geq 1$ ； $r$  愈大，表示兩個固有值差異愈大。

$r$  與  $(1+r)^2/r$  的關係如下

$r =$	1	2	3	4	5	6	..
$(1+r)^2 / r =$	4	4.5	5.3	6.25	7.2	8.17	..

所以我們可以用  $r$  值制定一個門檻值  $T = (1+r)^2/r$

若  $\frac{(\text{Tr} H)^2}{\text{Det} H} < T$ ，則表示兩個個有值相差不大，

兩方向的曲率相近，所以  $(x, y)$  是一個角點。  
在 Lowe 的論文中，使用  $r = 10$  制定門檻值  $T = 12.1$ 。



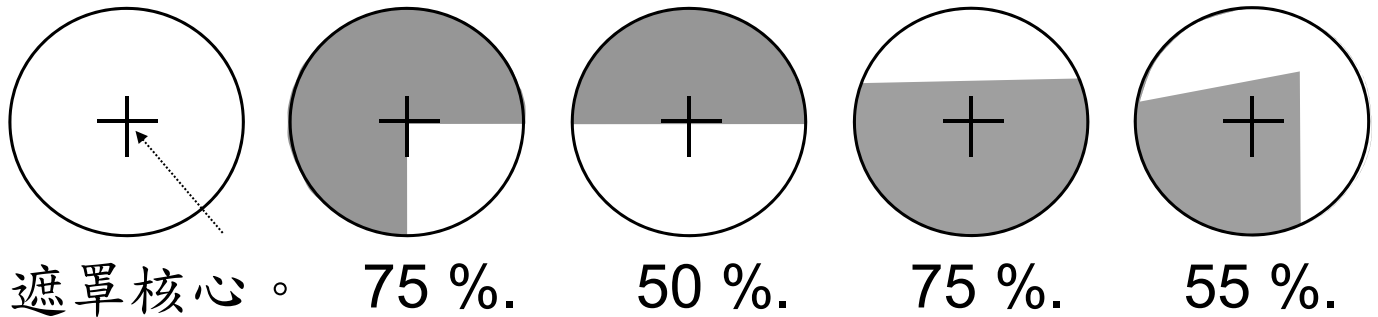
## 8.5.4 SUSAN 角偵測法

- ❁ **SUSAN** (Smallest Univalued Segment Assimilating Nucleus) 角偵測法是 Smith and Brady 在 1997 年所提出的；意思是“與核心相似的最小單一數值分割”。
- ❁ **SUSAN** 使用一個圓形遮罩，中心點 (像素) 稱為核心 (nucleus)。以核心點的灰階與圓形遮罩內的其他像素灰階比較。若某一像素灰階與核心灰階相似 (差不多)，則該像素即設定為 “1”；否則為 “0”。“1” 像素所形成的區域稱為 **USAN**。計算 **USAN** 與圓形遮罩面積的比值，藉以找出邊緣點與角點。

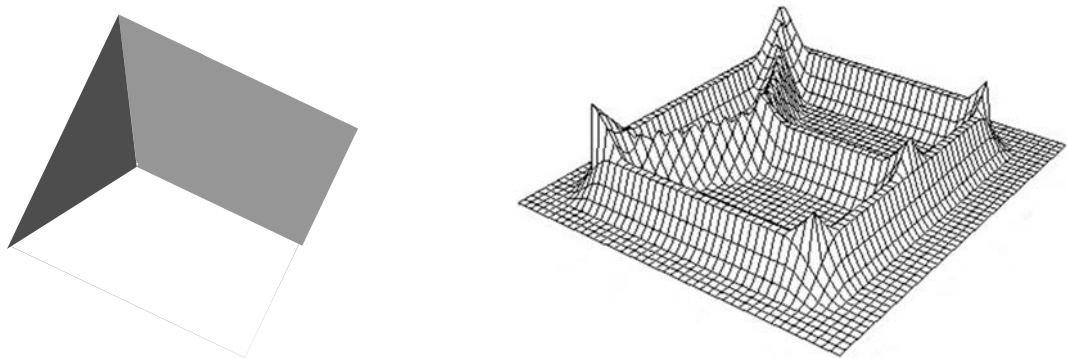




❁ 不同 *USAN* 與圓面積比值的圓形遮罩。

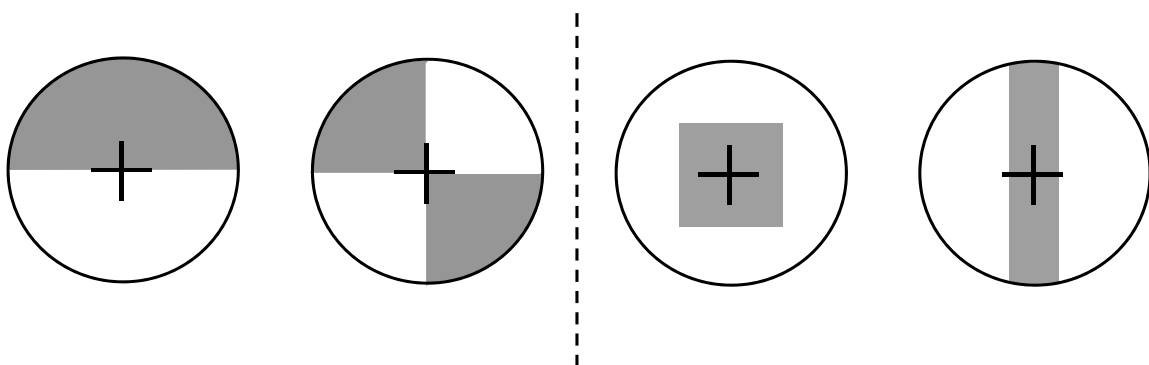


❁ 將每一像素當做核心所求出來的 *USAN* 數值當成高度值繪製成立體圖 (*USAN* 愈小，高度愈高)



❁ *SUSAN* 方法與其他角點偵測法最大不同之處在於不需計算影像像素的梯度 (gradient)，因此執行速度快，且不需做雜訊抑制。

❁ *SUSAN* 角偵測法只分析圓形遮罩內的 *USAN* 區域面積，沒有分析 *USAN* 形狀；所以會誤判某些形狀的角點；例如，以下兩例，兩個 *SUAN* 的 *R* 值都是  $\frac{1}{2}$ ，不能區分。





## 8.5.5 FAST 角偵測器

✿ FAST (Features from Accelerated Segment Test) 是由 Rosten and Drummond 在 2006 年所提出的一種快速角點偵測法。該方法的觀念是以待測點  $p$  為中心定義一個  $7 \times 7$  遮罩，以遮罩內最大圓圈上的 16 像素之灰階判斷該中心點是否為角點，

		16	1	2		
	15				3	
14						4
13			$p$			5
12						6
	11				7	
		10	9	8		



### ✿ 演算法

各別比較  $p$  像素與周圍 16 像素的灰階，定義周圍 16 個像素的狀態；第  $i$  像素的狀態定義為

$$State_i = \begin{cases} 0 \text{ (darker),} & l_i \leq l_p - t \\ 1 \text{ (brighter),} & l_p + t \leq l_i \end{cases}$$

其中  $l_p$  為  $p$  像素的灰階， $l_i$  為第  $i$  像素的灰階， $t$  為一門檻值 (threshold value)。當有連續 9 個或多於 9 個像素為相同狀態，則像素  $p$  像素判定為一角點。



上述連續 9 個像素有相同狀態的準則可以改成“連續 12 個或多於 12 個像素為相同狀態”。9 個像素的準則比較穩定 (reliable)；但比較耗時。判斷 9 個像素的方法稱為 *FAST-9*；12 個像素的方法稱為 *FAST-12*。

### ✿ 加速策略

在 *FAST-9* 方法中，可以先判斷位置 1, 5, 9, 13 像素的狀態；如果這些狀態中，有連續 2 個狀態相同，才執行連續 9 個像素的狀態。在 *FAST-12* 方法中，一樣先判斷位置 1, 5, 9, 13 像素的狀態；如果這些狀態中，有連續 3 個狀態相同，才執行連續 12 個像素的狀態。



✿ 上述方法有可能造成相鄰像素都是角點，因此要抑制非最大角點 (non-maximum suppression)。但是在上述準則中只有判定角點與非角點兩種，並沒有計算是角點的機率值；因此要另外創造一個評估函數 (score function) 以取捨鄰近角點。Rosten and Drummond 所定義的評估函數為

$$ev(p) = \max \left( \sum_{i \in S_{bright}} (|I_i - I_p| - t), \sum_{i \in S_{dark}} (|I_p - I_i| - t) \right)$$

其中  $S_{bright} = \{i \mid I_i \geq I_p + t\}$  and  $S_{dark} = \{i \mid I_i \leq I_p - t\}$ 。對每一個角點  $c$ ，以  $c$  為中心定義一個  $m \times m$  遮罩，在這個遮罩範圍內，若  $c$  的  $ev$  值不是最大的，則  $c$  將被標示將去除。等所有像素都處理過後，再將所有標示去除的像素一次去掉。



## ❁ FAST12, 11, 10, & 9 方法的評論

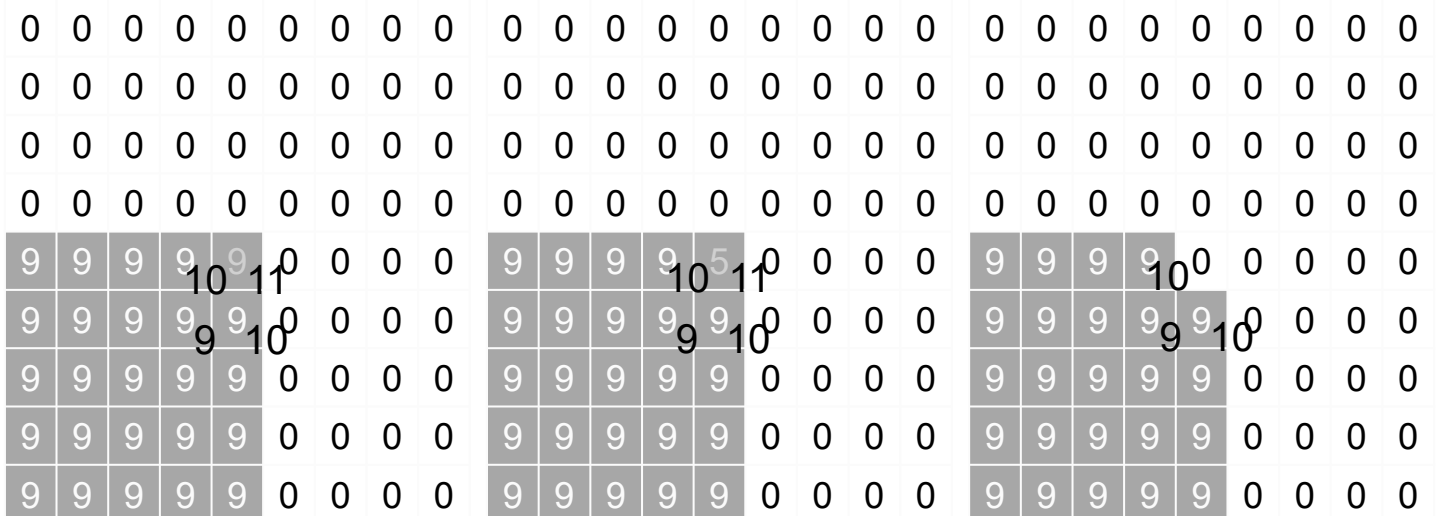
❁ FAST12 能找到的角點，FAST11 一定能找到  
FAST11 能找到的角點，FAST9 一定能找到

❁ 對於完美的角點於言  
FAST12 約可偵測 80 度以內的角點，  
FAST11 約可偵測 90 度以內的角點，  
FAST10 約可偵測 110 度以內的角點，  
FAST9 約可偵測 140 度以內的角點。

❁ 獨立點 (isolated point) 及  
細線端點 (end point of thin line) 也會被偵測出來



❁ 對於不夠完美的角點 (例如，圓角) 於言  
FAST 方法的偵測結果較不穩定



FAST11 找正確的角點

FAST11 找出正確的角點

FAST10 找出不唯一的角



## ❁ 角點偵測法的效能比較

以 *Intel Pentium III 850 MHz* 處理器為準

角點偵測法	花費時間 (ms)
<i>FAST-9</i> (無抑制非最大)	4.34
<i>FAST-9</i> (有抑制非最大)	5.29
<i>FAST-12</i> (無抑制非最大)	4.31
<i>FAST-12</i> (有抑制非最大)	4.60
<i>Harris</i>	166.00
<i>DoG</i>	345.00
<i>SUSAN</i>	27.50

