



第4章 影像輪廓的擷取

4.1 輪廓與影像處理

4.2 研究輪廓的性質

4.3 利用微分擷取影像的輪廓

4.4 利用範本匹配法擷取輪廓

4.5 實際擷取影像輪廓



4.6 影像輪廓之簡化(細線化)處理

4.7 利用拉普拉斯與零點交叉求取輪廓

4.8 直線檢測

4.1 輪廓與影像處理

► 輪廓

- ▶ 字典中的解釋是顯示物體外緣的線條。
- ▶ 但在影像處理的範疇中，似乎應加上構成影像特徵的線條要素，定義才會更清楚一些。
- ▶ 輪廓圖應當具有讓人一看就懂的特點。
- ▶ 影像輪廓的擷取(又可稱為「」或者「」)是電腦影像處理中一項重要的基本技術。



擷取輪廓

邊緣擷取

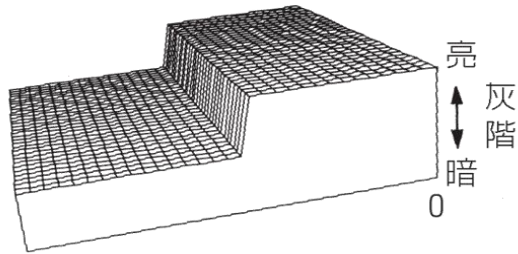
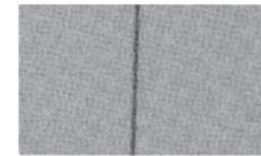
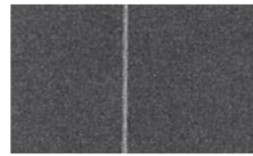
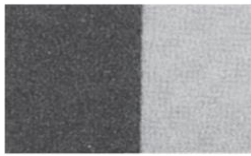
4.2 研究輪廓的性質

- ▶ 影像中物體與物體，或者物體與背景的交界線，形成了所謂的輪廓。
- ▶ 在影像的灰階濃度和色彩發生急劇變化之處，就可以見到輪廓。
- ▶ 自然影像中，當色彩產生變化時，一定也伴隨著相當程度的灰階濃度變化。
- ▶ 因此，在擷取輪廓時，首先應當著重於灰階濃度的變化。

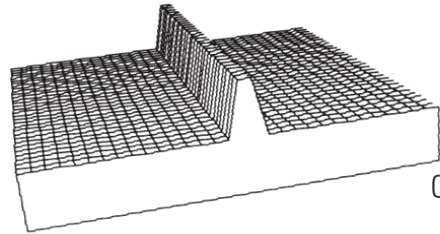


- ▶ 圖4.1(a)是以階梯狀(Step)表現灰階濃度的急劇變化。可以很清楚地看見輪廓外形，稱為邊緣(Edge)。
 - ▶ 它是指在物體與背景的交界處，產生了明顯的階梯狀濃度變化。
- ▶ 圖4.1(b)中，是在線條位置產生了濃度變化，也很清楚地看見輪廓線條(Line)。
- ▶ 圖4.1(c)的灰階濃度呈折線狀變化，但當折線的角度較急陡時，仍能清楚地看見對應的輪廓。
- ▶ 圖4.1(d)雖然也有濃度變化，但因其變化平滑而緩慢，故看不清輪廓在何處。

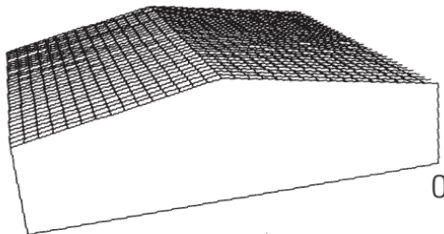
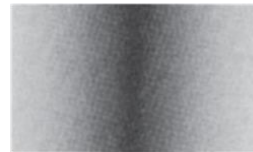
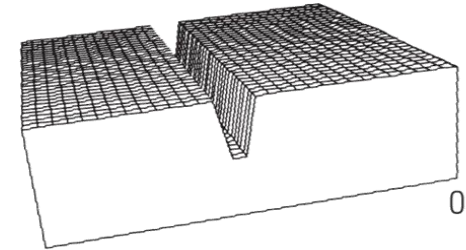




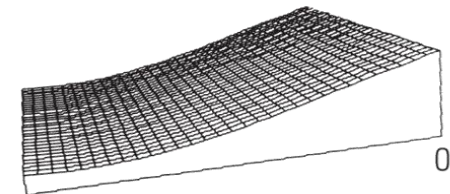
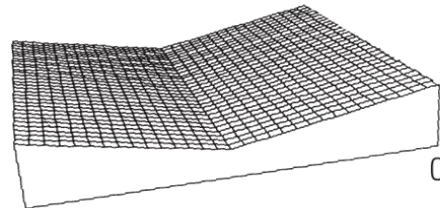
(a)邊緣(edge)



(b)線



(c)折線的變化



(d)緩慢而平滑地變化(看不見輪廓)

圖4.1 輪廓的灰階濃度變化模型(model)

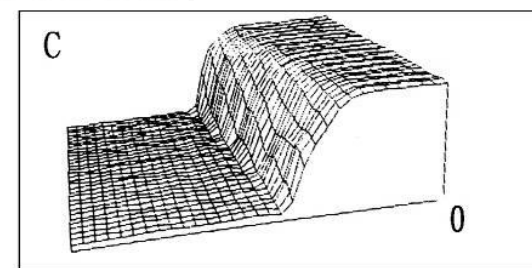
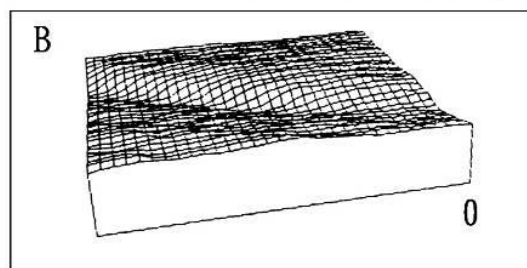
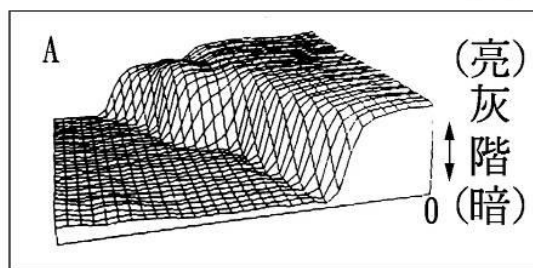
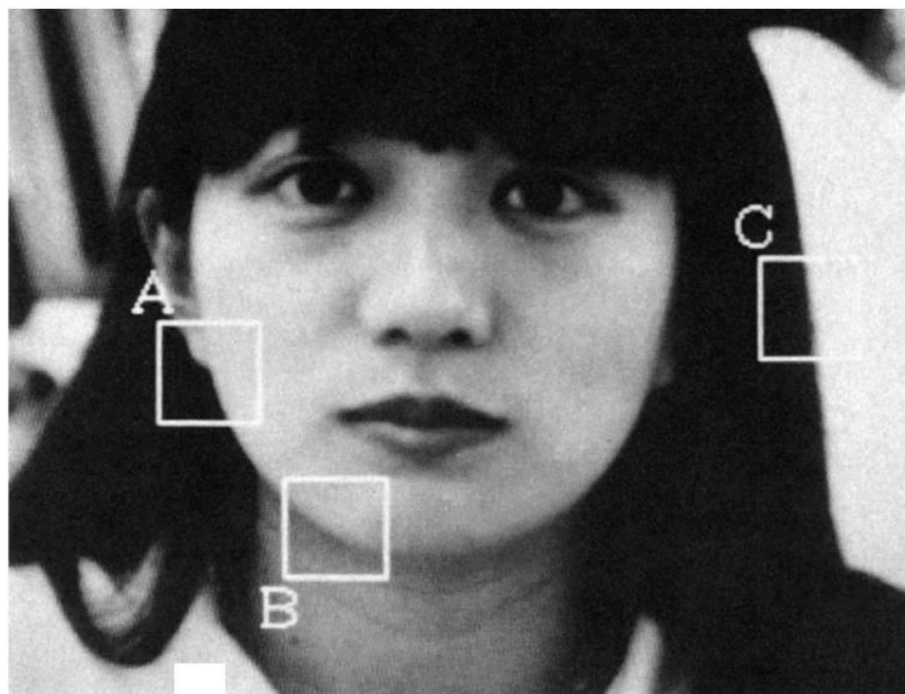


圖4.2 自然影像的灰階濃度變化

4.3 利用微分擷取影像的輪廓

- ▶ 輪廓是濃度急遽變化的部分
 - ▶ 利用影像灰階濃度函數變化部分之微分運算，就能利用來擷取輪廓。
- ▶ 用來表示座標 (x,y) 濃度分配之微分值，也稱**梯度(Gradient, G)**
 - ▶ 可用含有大小與方向之向量(vector)， $G(x,y)=(f_x, f_y)$ 表達。

- 其中， f_x 為 x 方向的微分， f_y 為 y 方向的微分。

- f_x, f_y 在數位影像中，可用

$$x\text{方向的微分 } f_x = f(x+1, y) - f(x, y) \quad (4.1)\text{式}$$

$$y\text{方向的微分 } f_y = f(x, y+1) - f(x, y)$$

加以計算。求得微分值 f_x 、 f_y 之後，可利用以下的公式計算出輪廓值的強度。

【強度】

$$\sqrt{f_x^2 + f_y^2} \quad (4.2)\text{式}$$

$$\text{或是 } |f_x| + |f_y| \quad (\text{其中，} | \quad | \text{代表絕對值}) \quad (4.3)\text{式}$$

【方向】 向量 (f_x, f_y) 的方向



- ▶ 在數位影像中，因為數據(Data)是以一定間隔跳躍式排列，無法用具真正意義之微分作運算。因此，如公式(4.1)，取相鄰圖素之間的差來運算，作近似之微分，稱之差分。
- ▶ 為了進行微分運算，用相鄰圖素之間的運算來表達之係數組合，稱為微分運算子(Operator)。

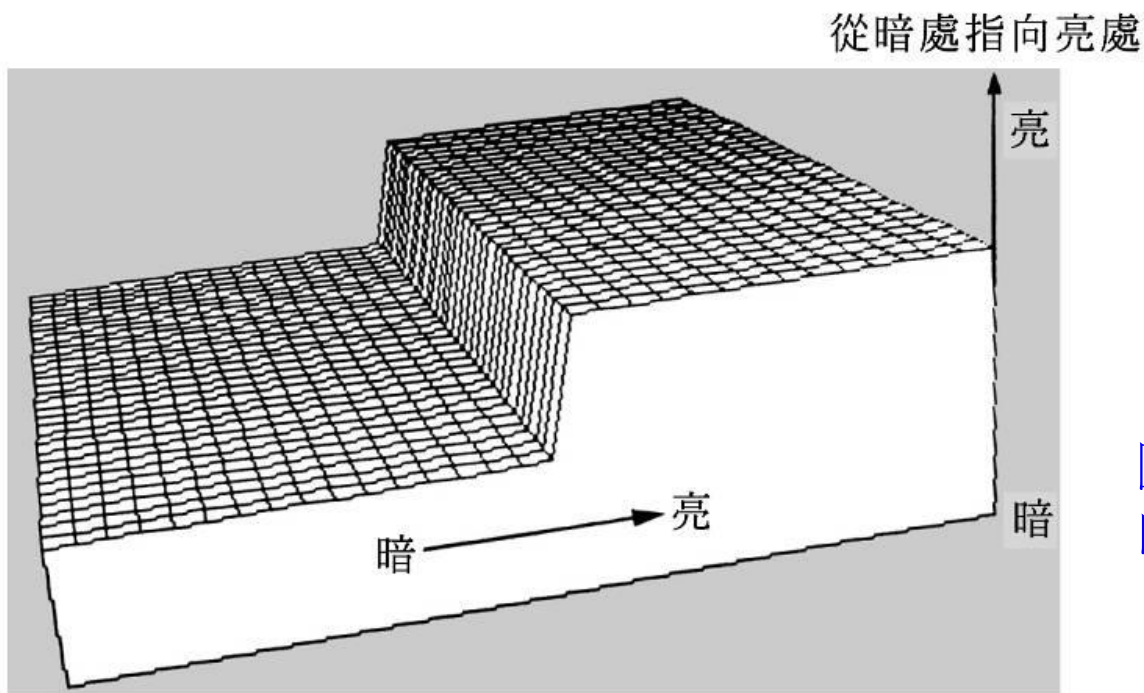


圖4.3 表示邊緣之
向量方向



表4.1 計算梯度變化的微分運算子

運算子名稱	通常的差分 (以式4.1為例)			Roberts運算子			Sobel運算子		
求 f_x 的運算子	0	0	0	0	0	0	-1	0	1
	0	-1	1	0	-1	0	-2	0	2
	0	0	0	0	0	1	-1	0	1
求 f_y 的運算子	0	0	0	0	0	0	-1	-2	-1
	0	-1	0	0	0	-1	0	0	0
	0	1	0	0	1	0	1	2	1

4.4 利用範本匹配法擷取輪廓

- ▶ 所謂範本匹配(Template matching)，是指利用範本(樣板)，作匹配(符合，一致)比對的研究調查方式。
 - ▶ 先準備若干個用來表示輪廓之標準樣式(Pattern)，也就是範本，將影像的一部分拿來與範本一一做比較，再從中選出最相似的一份。



► Prewitt方法

- 對應邊緣(edge)的八個不同的方向，準備8個遮罩(mask)。
- 遮罩如何與實際影像作比較。其比較的方法如同微分運算子之計算過程，我們先將輸入影像的圖素，乘以其周圍圖素的遮罩值，計算其一致的程度後，再進行比較。
- 逐一進行上述的計算。在這當中求得最大值的遮罩方向，就是邊緣的方向，而其計算結果則為該邊緣的強度。
- 其輸出的結果是與輪廓強度對應之灰階濃淡影像。

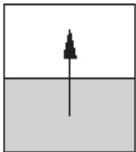
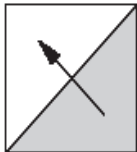
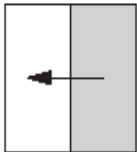
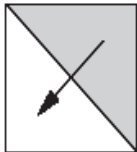
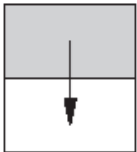
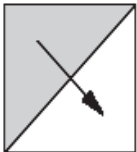
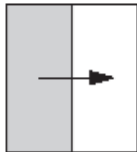
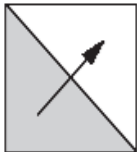
遮罩樣式	a	b	c	d	e	f	g	h
	1 1 1	1 1 1	1 1 -1	1 -1 -1	-1 -1 -1	-1 -1 1	-1 1 1	1 1 1
	1 -2 1	1 -2 -1	1 -2 -1	1 -2 -1	1 -2 1	-1 -2 1	-1 -2 1	-1 -2 1
	-1 -1 -1	1 -1 -1	1 1 -1	1 1 1	1 1 1	1 1 1	-1 1 1	-1 -1 1
對應之邊界								

圖4.4 使用範本匹配之遮罩圖群(Prewitt運算子)



輸入影像 $f(x,y)$ 的灰階濃度

		→ x	
	50	100	100
↓ y	0	80	100
	0	0	100



輸出影像 $g(x,y)$ 的灰階濃度

		→ x	
↓ y		290	

因其中290為最大，故邊緣的種類選h，強度為290

(從所關注像素及周邊像素，計算出與各個遮罩圖的一致程度)

遮罩圖	a	b	c	d	e	f	g	h
計算結果	90	-110	-310	-310	-210	-10	190	290

例如，遮罩a計算如下

$$1 \times 50 + 1 \times 100 + 1 \times 100 + 1 \times 0 + (-2) \times 80 + 1 \times 100 + (-1) \times 0 + (-1) \times 0 + (-1) \times 100 = 90$$

圖4.5 匹配(Matching)的計算範例





(a) 原有影像



(b) 梯度(通常的差分)



(c) 梯度(Roberts)



(d) 梯度(Sobel)



範本匹配(Prewitt)

圖4.6 輪廓擷取的結果



4.5 實際擷取影像輪廓

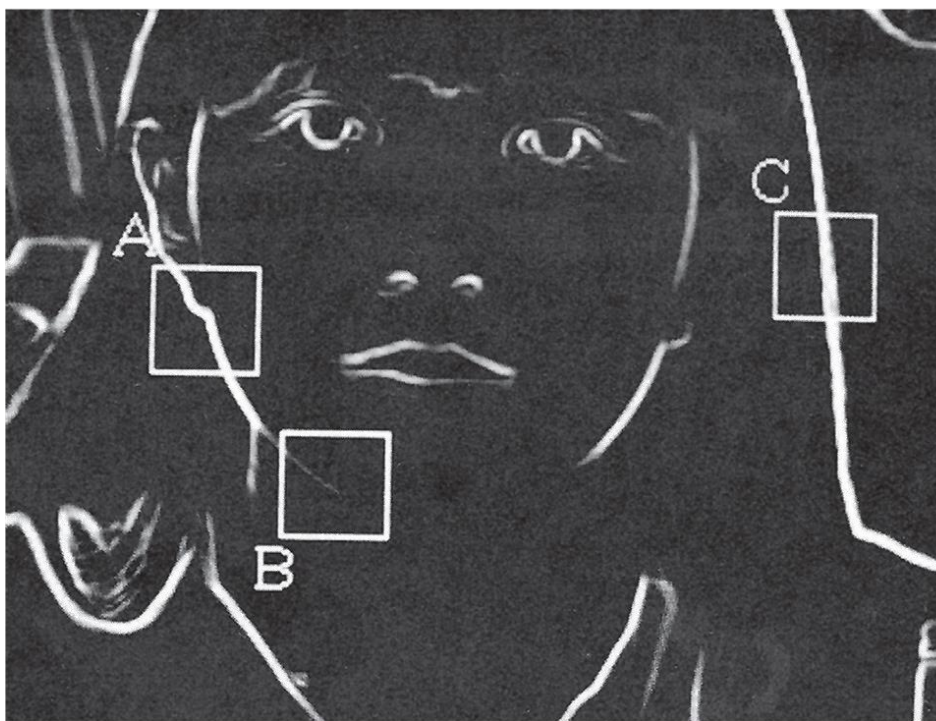
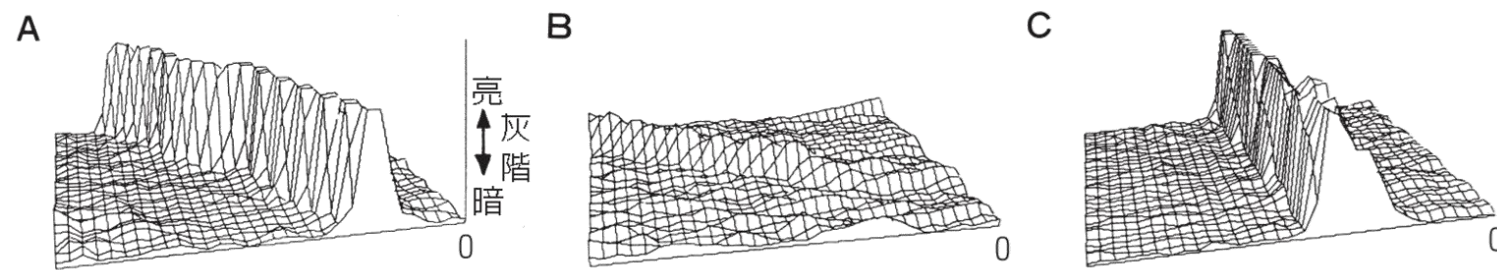


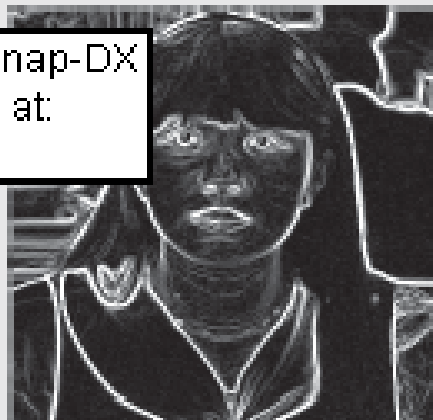
圖4.7
擷取輪廓的灰階變化



4.6 影像輪廓之簡化(細線化)處理

- ▶ 求得「與輪廓強度相對應的灰階影像」，所得到的輪廓線，經二值化後，若要讓線寬只占一個像素，則線條還必須經過細線化處理。
- ▶ 二值化處理，需要對所擷取輪廓的輸出影像，進行臨界值處理。
- ▶ 臨界值越大，輪廓線越模糊；反之，臨界值太小，則不必要的雜訊會增多。
- ▶ 除了採用反覆試驗以確定出較合適的臨界值以外，也可以考慮採用直條圖處理的辦法。

Image captured with HyperSnap-DX
Get a free temporary license at:
<http://www.hyperionics.com>



(a) 輸入的輪廓影像(Robert運算子, amp=5)



(b) 二值化影像(臨界值50)



(c) 二值化影像(臨界值100)



(d) 二值化影像(臨界值150)

圖4.8 二值化的輪廓影像



- ▶ 所謂細線化，就是把粗細不一的輪廓線條整理成寬度為一個像素的處理方式。
- ▶ 利用臨界值處理後的二值影像，將較粗的輪廓線從外側開始削細，直到變為只有一個像素的寬度為止。
- ▶ 細線化處理是在保持其連結性的前提下，逐步削掉可刪除的像素。
- ▶ 細線化是在不改變形狀的情況下，削除所連結的像素。



- 細線化的代表性方法為「Hilditch法」，它是將符合以下六種條件的像素依序削除，直到再也沒有可削除的像素時，細線化的過程就結束了。

條件1 圖形的一部分

條件2 境界像素

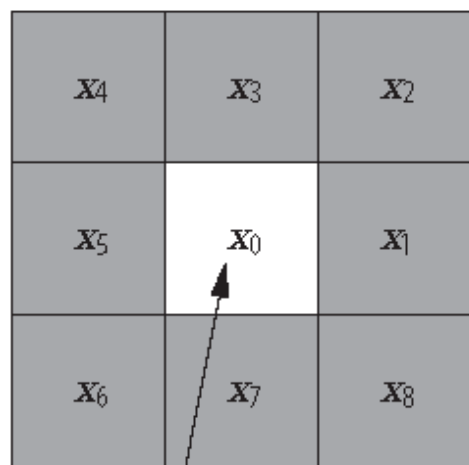
條件3 保留端點

條件4 保留獨立點

條件5 保留連結性

條件6 線寬為2時，只除去單邊





中心像素

圖4.9 像素的鄰近

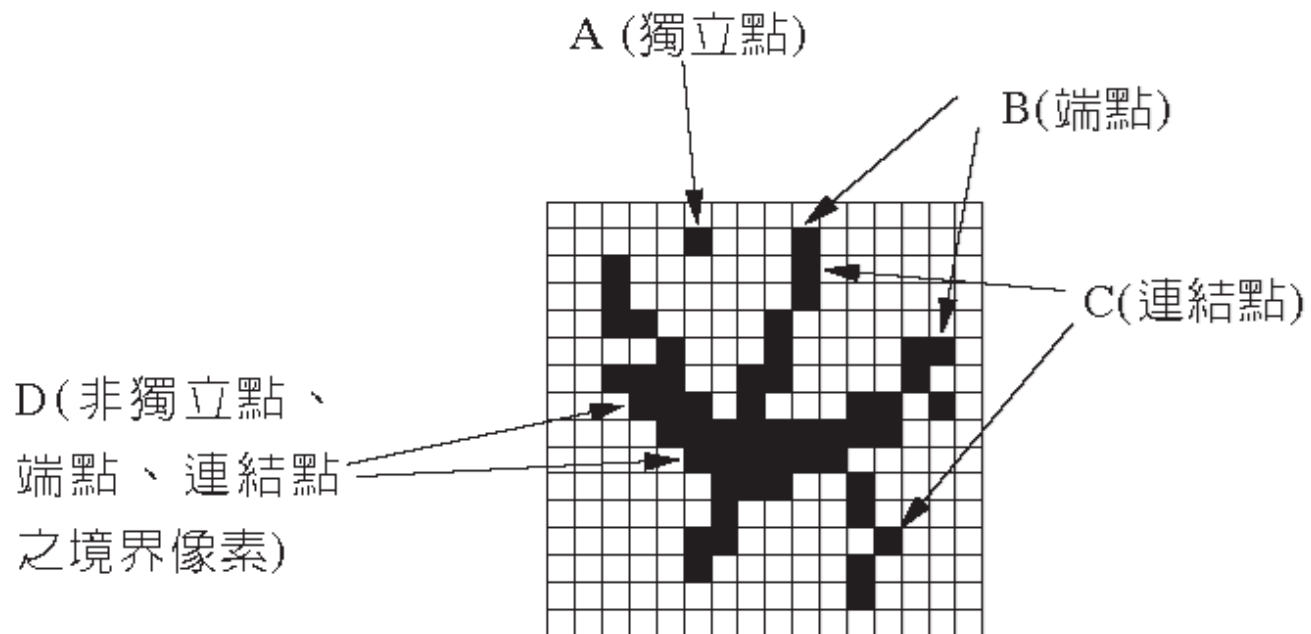


圖4.10 像素的種類





連結數=0



連結數=1



連結數=1



連結數=2



連結數=3



連結數=4

圖4.11 連結數的範例



表4.2 連結數及像素的特徵

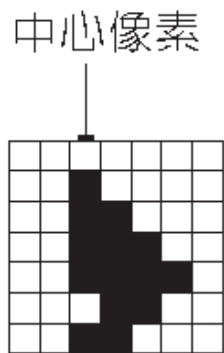
連結數	像素的特徵
0	獨立點或內部像素
1	端點或境界像素
2	連結點
3	分歧點
4	交叉點

► 研究連結狀態的參數(Parameter)，也就是「連結數」。

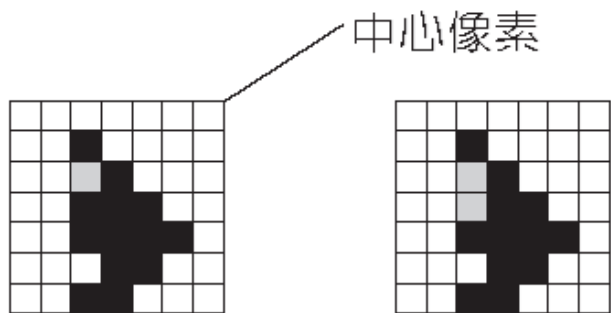
$$N = \sum_{k=1,3,5,7} (\bar{f}(x_k) - \bar{f}(x_k)\bar{f}(x_{k+1})\bar{f}(x_{k+2})) \quad (4.4) \text{式}$$

但 $\bar{f} = 1 - f$, $x_9 = x_1$

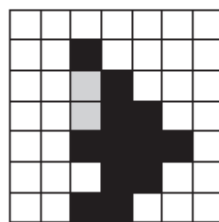




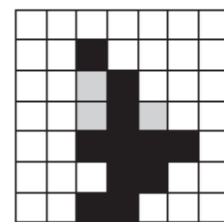
(1)



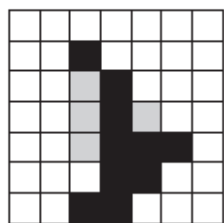
(2)



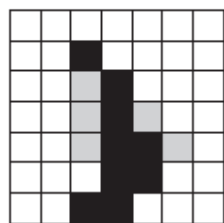
(3)



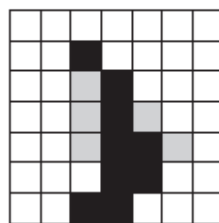
(4)



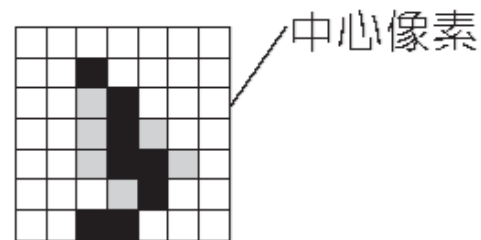
(5)



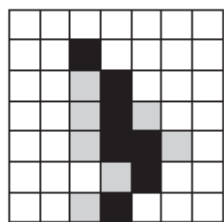
(6)



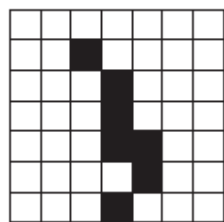
(7)



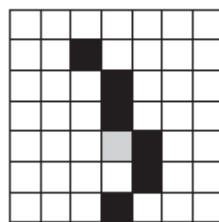
(8)



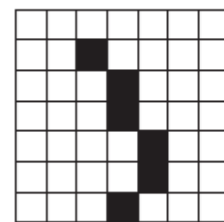
(9)



(10)



(11)



(12)

圖4.12 細線化的步驟



輸入輪廓影像(用Prewitt方法進行臨界值處理)



細線化影像(線條的寬度為1)

圖4.13 細線化



4.7 利用拉普拉斯與零點交叉求取輪廓

► 本節要說明的，則是利用輪廓的強度資訊，直接求得輪廓線條的方法。

► 輪廓的擷取是使用二次微分來計算。右圖為一次微分與二次微分的關係。

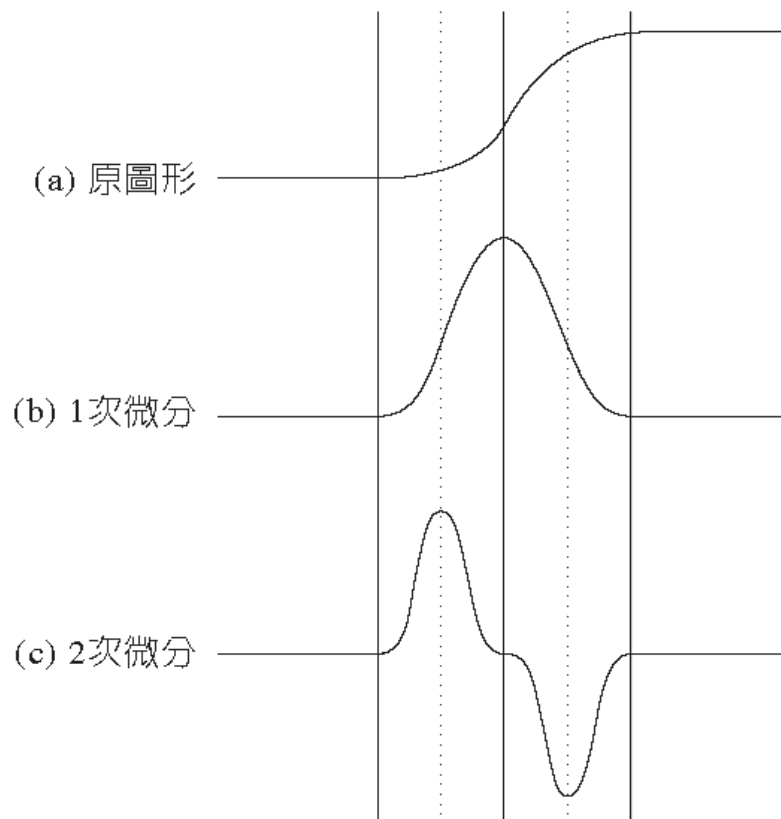


圖4.14 1次微分與2次微分



- ▶ 二次微分的計算方法，可使用相當於二個變數函數的拉普拉斯(laplacian)之濾波器(filter)。
- ▶ 這個濾波器也可稱之為拉普拉斯濾波器。
- ▶ 要求取2個變數函數 $f(x,y)$ 的拉普拉斯值，請使用偏微分，用以下公式表示：

$$L(x,y) = \frac{\partial^2}{\partial x^2}f(x,y) + \frac{\partial^2}{\partial y^2}f(x,y) \quad (4.5) \text{式}$$

- ▶ 如果是要求數位影像的拉普拉斯 $L(x,y)$ ，因為是梯度的再一次微分，所以可用下列簡單公式表示：

$$L(x,y) = 4 \cdot f(x,y) - \{f(x,y-1) + f(x,y+1) + f(x-1,y) + f(x+1,y)\} \quad (4.6) \text{式}$$

- ▶ 拉普拉斯值也可用微分運算子的微分演算加以計算。



- ▶ 相對於梯度有大小與方向性，拉普拉斯則只有大小。
- ▶ 拉普拉斯有很多種類。
- ▶ 拉普拉斯是輸出以0為中心，擷取正負值。
- ▶ 隨著所使用的拉普拉斯種類不同，輸出影像的樣子，或多或少會有點差異。
- ▶ 任一輸出的輪廓線都會「夾著輪廓線條的亮區及暗區」成對出現。

表4.3 拉普拉斯計算用的微分運算子

運算子名稱	拉普拉斯1 ((4.5)式的範例)			拉普拉斯2			拉普拉斯3		
求得 $L(x,y)$ 的運算子	0	-1	0	-1	-1	-1	1	-2	1
	-1	4	-1	-1	8	-1	-2	4	-2
	0	-1	0	-1	-1	-1	1	-2	1

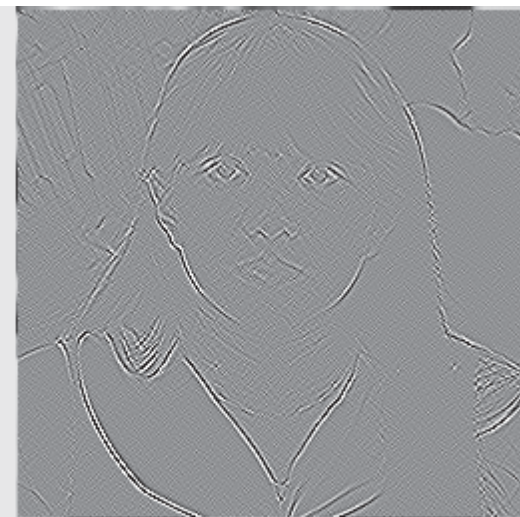




(a) 拉普拉斯1



(b) 拉普拉斯2



(c) 拉普拉斯3

圖4.15 拉普拉斯濾波器的處理範例



- ▶ 微分是計算鄰近像素的差，因此其結果受到雜訊的影響頗大。
- ▶ 使用拉普拉斯求取輪廓線條時，必須充份去除雜訊的影像，才能得到令人滿意的結果。
- ▶ 滿足以下條件的像素，為零點交叉的像素。

●●● — 像素值使用補償值(OFFSET)，鄰近有正負像素。

●●● — 像素值不使用補償值，鄰近有與自己正負相反的像素。



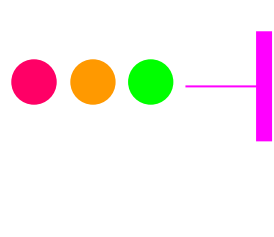
► 輪廓線條的顯示方式：



擷取出多條細的輪廓線條。



在正負波峰(波谷)的交界，求得零點交叉，所以其特徵是擷取的輪廓線條不會斷斷續續。



零點交叉的結果，並不能保證輪廓線條的寬度為一個像素。想求得寬度為一個像素的輪廓線條，必須經由細線化處理。





4.8 直線檢測

- ▶ 邊緣檢測所得到的交界，內含雜訊，而且不連續。
- ▶ Hough變換法，能從鋸齒狀、線條不連續的輪廓線條影像中，檢測出直線。

- ▶ 影像中的直線，可用下列式子表示：

$$y = ax + b \quad (4.7) \text{式}$$

- ▶ a 為斜率， b 為在 y 軸上之切片線段。如果影像中的像素 (x_1, y_1) 是在直線上，將它代入(4.7)式時，可寫成

$$b = -x_1 a + y_1 \quad (4.8) \text{式}$$

- ▶ 在 a - b 空間內，可以想像成斜率為 $-x_1$ ，而 b 切片線段則為 y_1 直線。



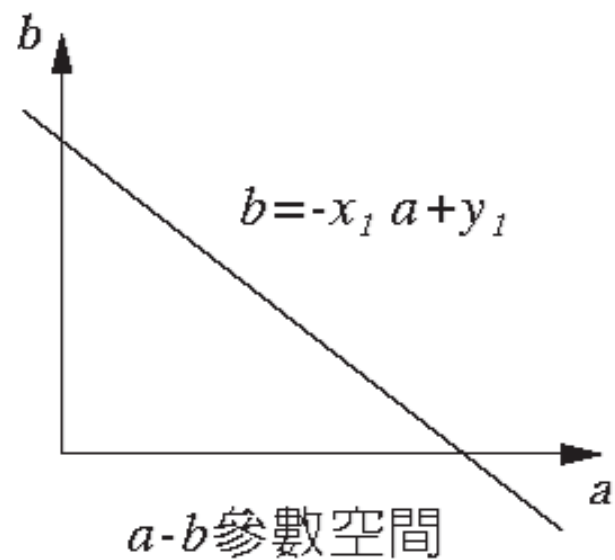
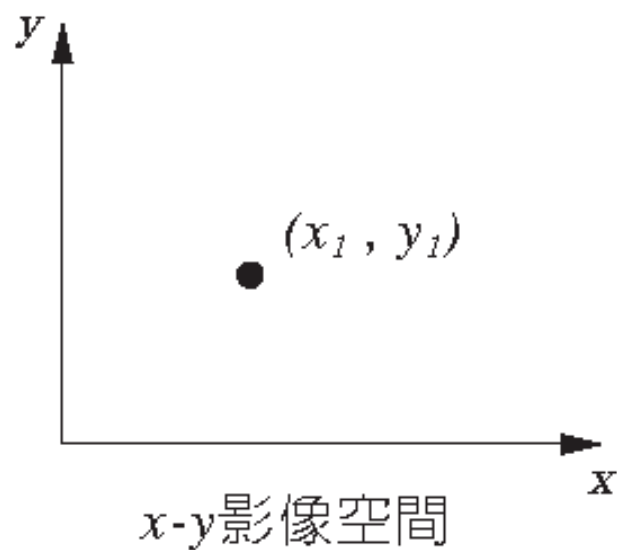
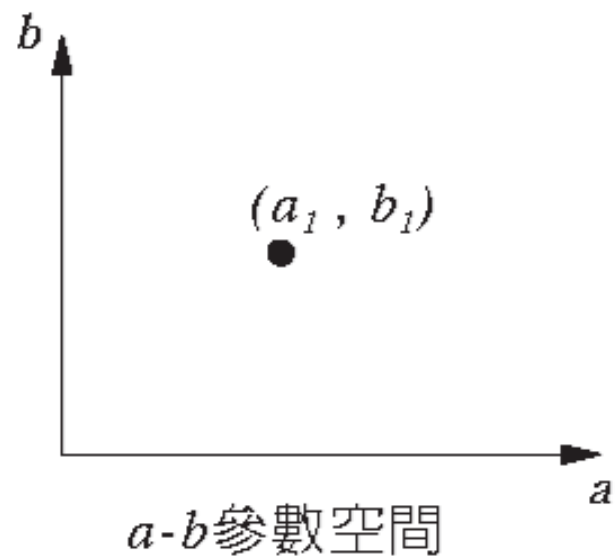
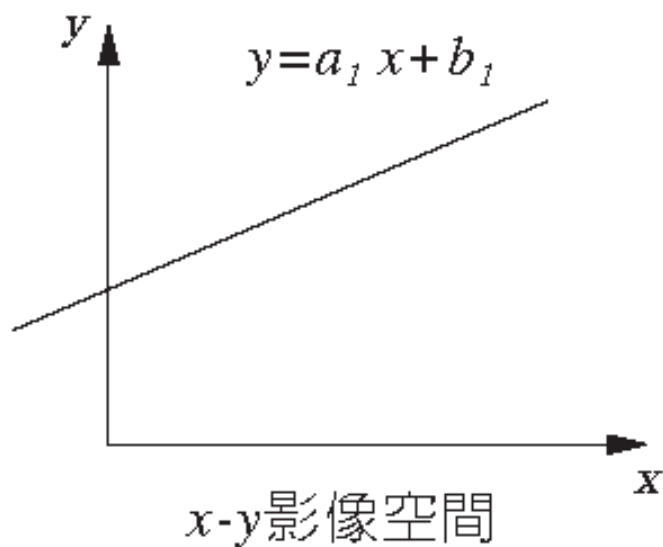


圖4.17 x - y 影像空間與 a - b 參數空間

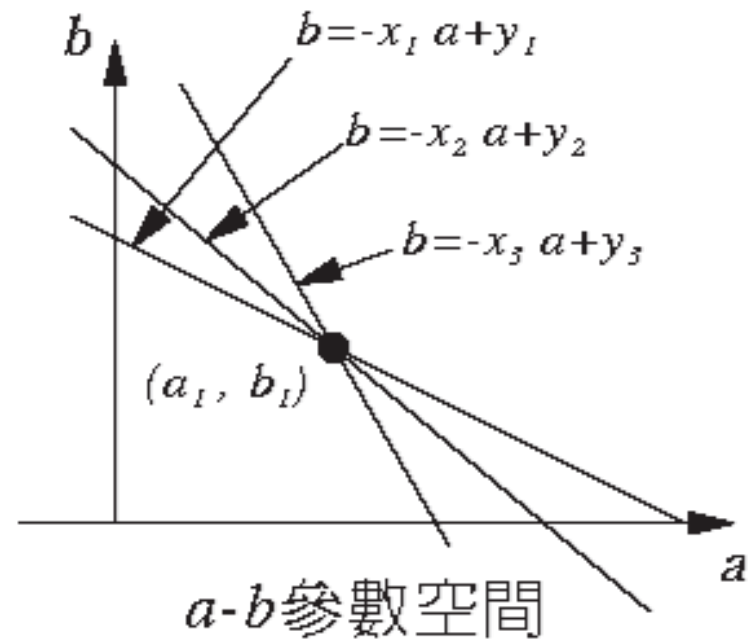
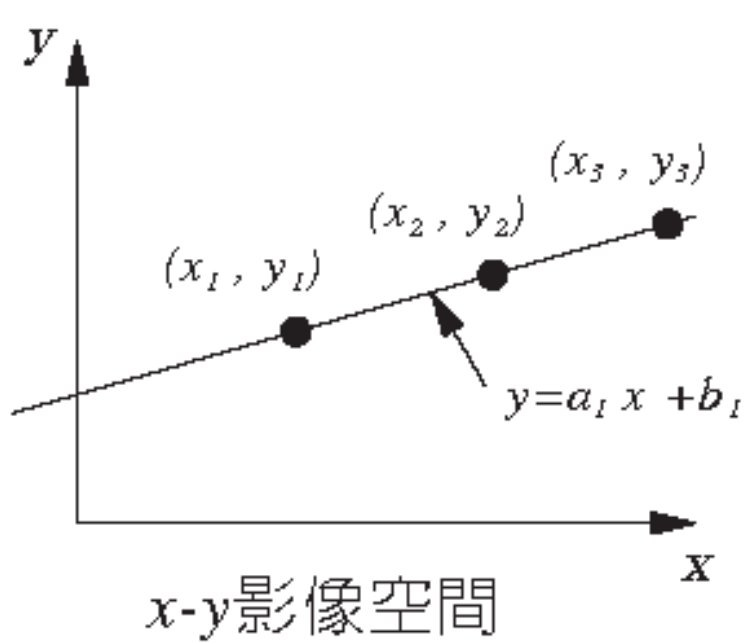


圖4.18 x - y 影像空間上的直線



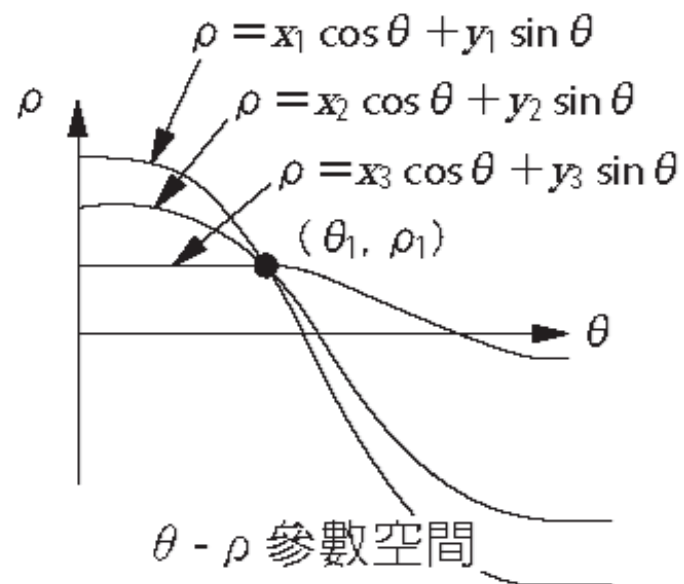
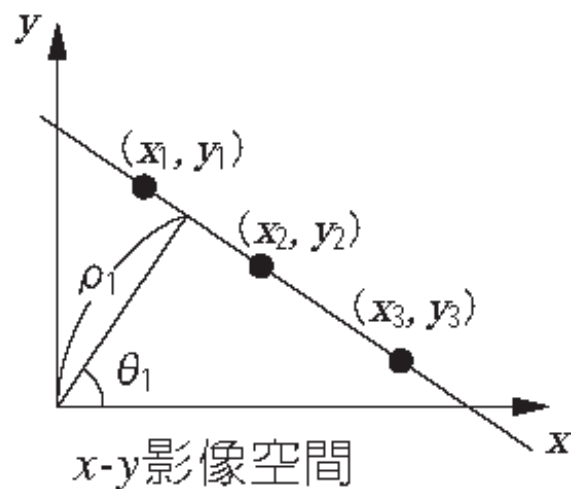
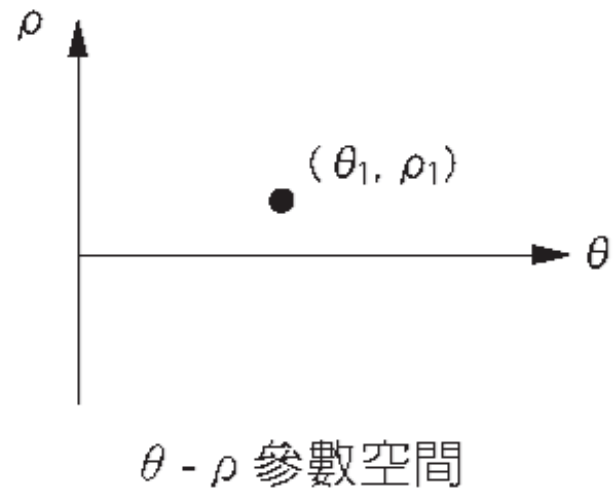
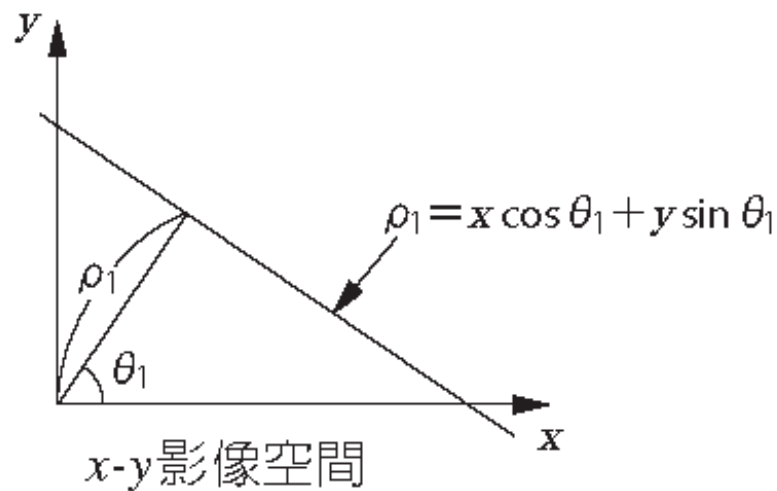
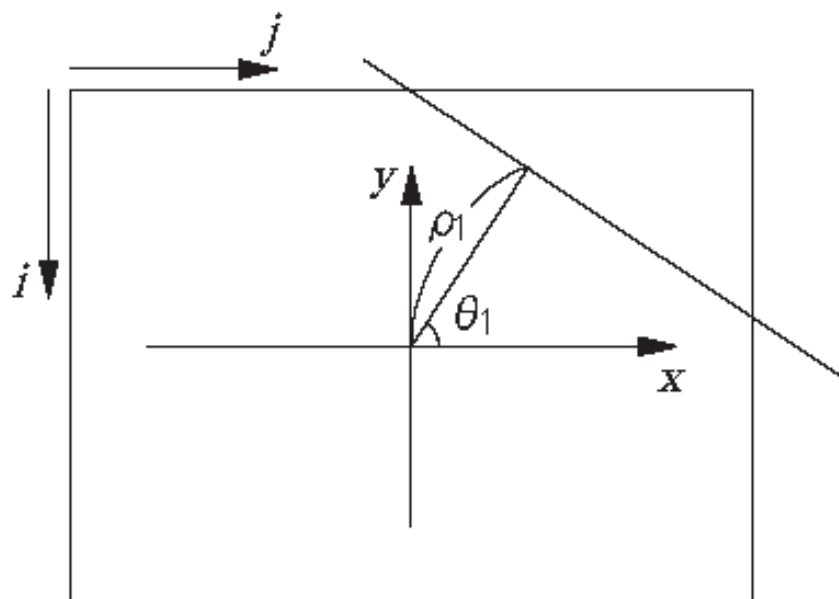
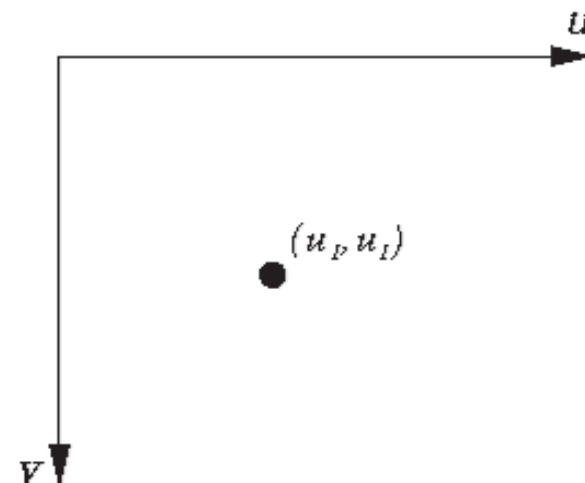


圖4.19 用 θ - ρ 參數空間，表達直線



輸入影像空間



Hough變換影像空間

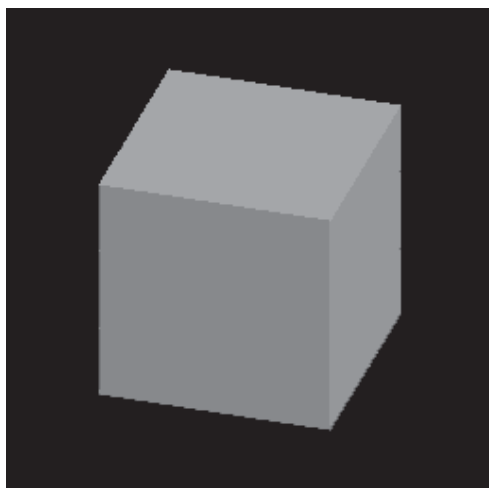
$$\rho_1 = \left(j - \frac{X_SIZE}{2}\right) \cos \theta_1 + \left(\frac{Y_SIZE}{2}\right) \sin \theta_1$$

$$v_1 = \rho_1 \frac{Y_SIZE}{2 \sqrt{\left(\frac{X_SIZE}{2}\right)^2 + \left(\frac{Y_SIZE}{2}\right)^2}} + \frac{Y_SIZE}{2}$$

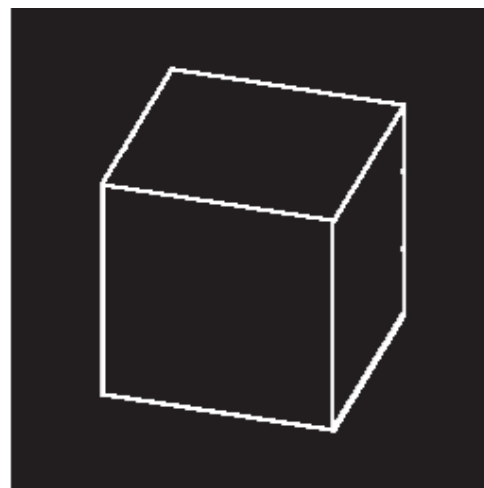
$$u_1 = \frac{\theta_1}{\pi} X_SIZE$$

圖4.20 輸入影像空間及Hough變換影像空間

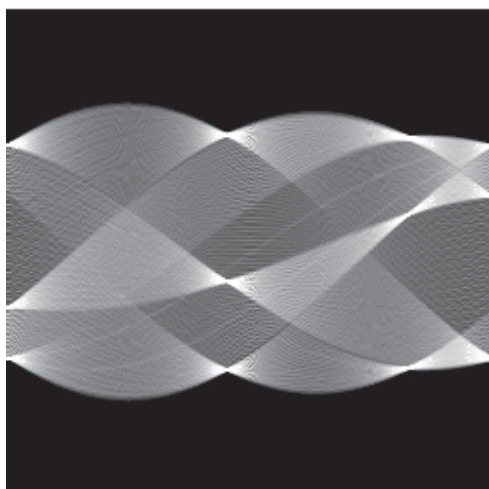




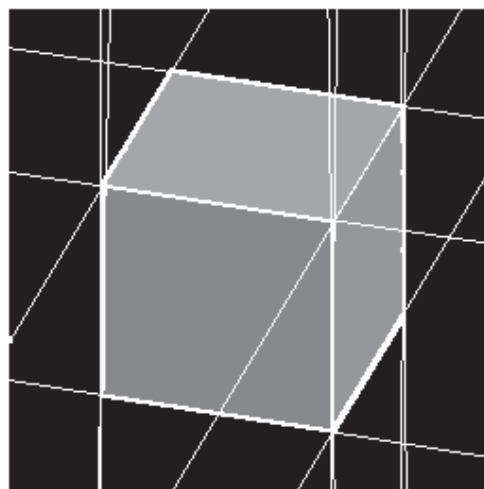
(a) 原影像



(b) 邊緣檢測二值化

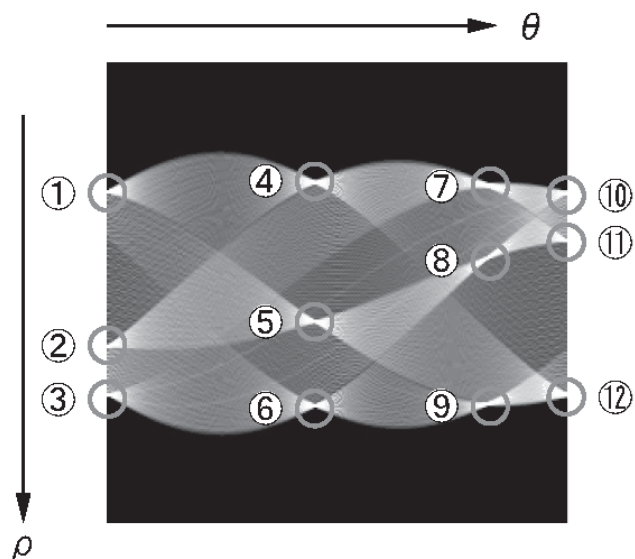


(c) $\theta - \rho$ 參數空間

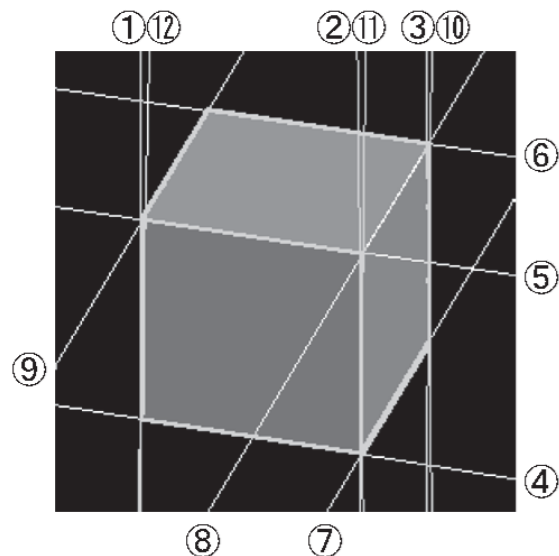


(d) 原影像及檢測之直線

圖4.21 Hough變換法



(a) $\theta - \rho$ 參數空間

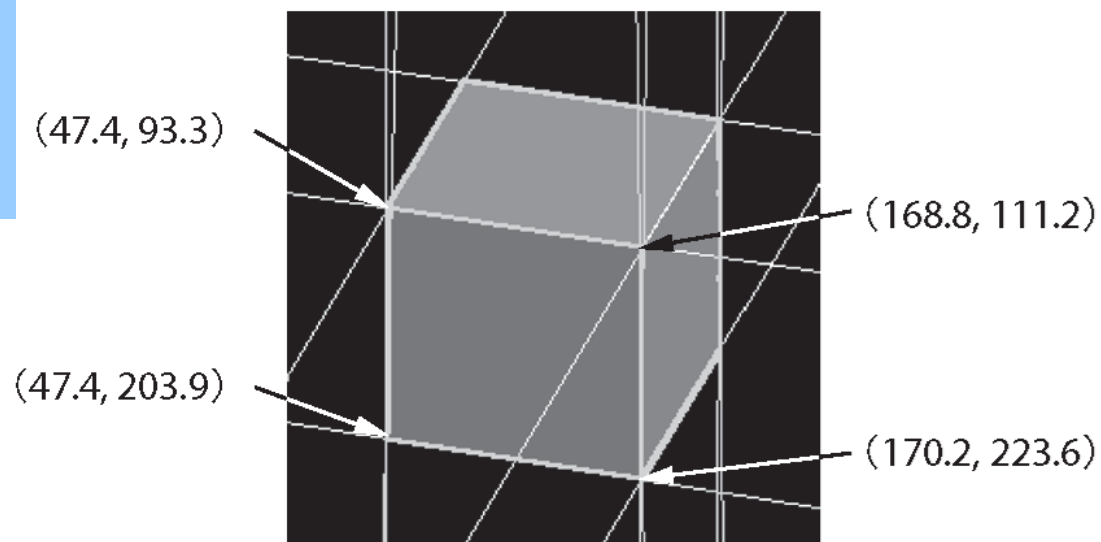


編號	(θ, ρ)	直線方程式
①	(0.0, -80.6)	$-80.6 = x \cos(0.0) + y \sin(0.0)$
②	(0.7, 41.0)	$41.0 = x \cos(0.7) + y \sin(0.7)$
③	(0.7, 79.2)	$79.2 = x \cos(0.7) + y \sin(0.7)$
④	(80.9, -87.7)	$-87.7 = x \cos(80.9) + y \sin(80.9)$
⑤	(81.6, 22.6)	$22.6 = x \cos(81.6) + y \sin(81.6)$
⑥	(81.6, 87.7)	$87.7 = x \cos(81.6) + y \sin(81.6)$
⑦	(149.1, -84.9)	$-84.9 = x \cos(149.1) + y \sin(149.1)$
⑧	(149.1, -28.3)	$-28.3 = x \cos(149.1) + y \sin(149.1)$
⑨	(149.1, 86.3)	$86.3 = x \cos(149.1) + y \sin(149.1)$
⑩	(179.3, -79.2)	$-79.2 = x \cos(179.3) + y \sin(179.3)$
⑪	(179.3, -42.4)	$-42.4 = x \cos(179.3) + y \sin(179.3)$
⑫	(178.6, -79.2)	$79.2 = x \cos(178.6) + y \sin(178.6)$

(b) 所檢測的直線

圖4.22 (θ, ρ) 及直線





(a) 直線的交叉點



(b) 原影像與變換的影像

圖4.23 直線的交叉點與變換的影像

