

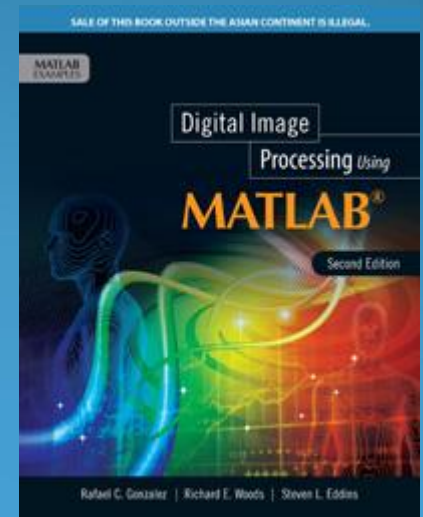
Medical Imaging

Medical Image Processing II

Yi-Li Tseng
FJU-EE
Fall 2017

References:

1. Digital Image Processing Using MATLAB, 2nd ed.: Rafael C. Gonzalez



第二章

強度轉換和 空間濾波

Outline

- 背景
- 強度轉換函數
- 直方圖處理和繪圖函數
- 空間濾波
- 影像處理工具箱標準的空間濾波器
- 使用模糊技巧的強度轉換和空間濾波

強度轉換函數-直方圖處理和繪圖函數

■ 直方圖正規化

$$p(r_k) = \frac{h(r_k)}{n} = \frac{n_k}{n}$$

- 此處 $k = 0, 1, 2, \dots, L-1$ 。以機率原理來說， $p(r_k)$ 被當成強度等級 r_k 出現的機率
- 在工具箱中處理影像直方圖的主要函數是 `imhist`，其基本語法為：

```
h = imhist(f, b)
```

其中 f 是輸入的影像， h 是它的直方圖，而 b 是用來形成直方圖的區間個數(如果 b 不包含在引數中，則用系統設定值 $b=256$)，一個區間只是強度刻度上較細的分隔

強度轉換函數-直方圖處理和繪圖函數

- 例如，假如我們處理uint8 的影像，且 $b=2$ ，則強度刻度被分割成兩個範圍：0 到127 和128 到255，產生的直方圖有兩個值：

$h(1)$ 等於是影像中強度值在 $[0, 127]$ 區間之像素總數

$h(2)$ 等於是影像中強度值在 $[128, 255]$ 區間之像素總數

- 下列式子得到正規化的直方圖

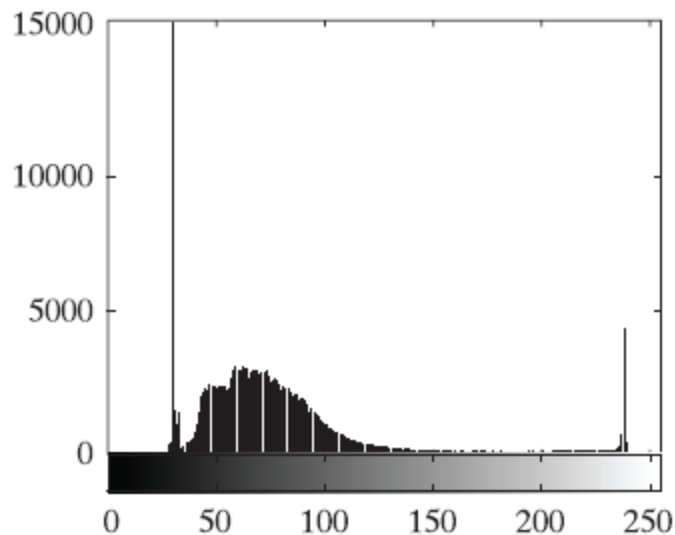
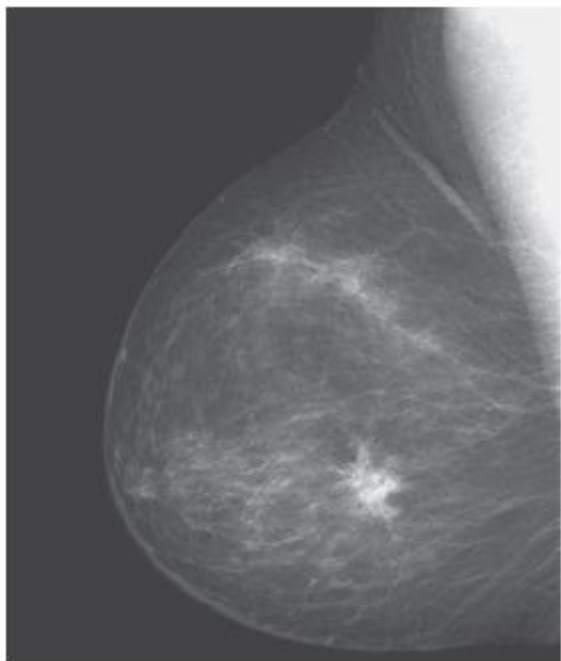
```
p = imhist(f, b)/numel(f)
```

`numel(f)` 函數表示陣列`f` 中元素總數

強度轉換函數-直方圖處理和繪圖函數

■ 範例 2.4 計算和繪製影像

圖2.7 各種繪製直方圖的方法

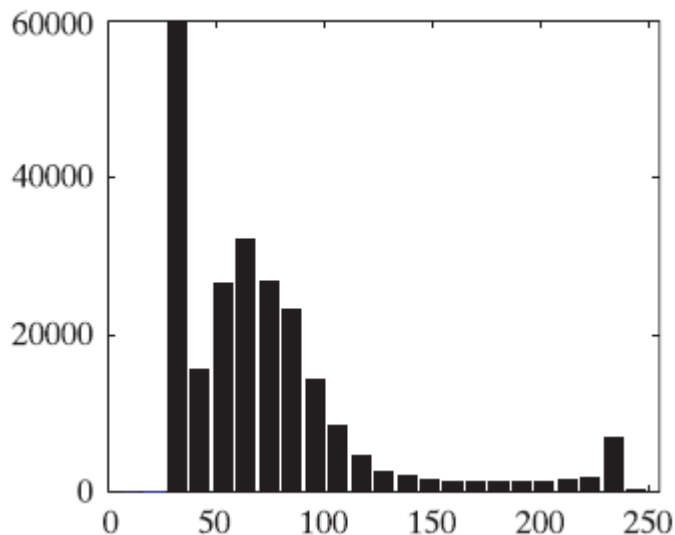
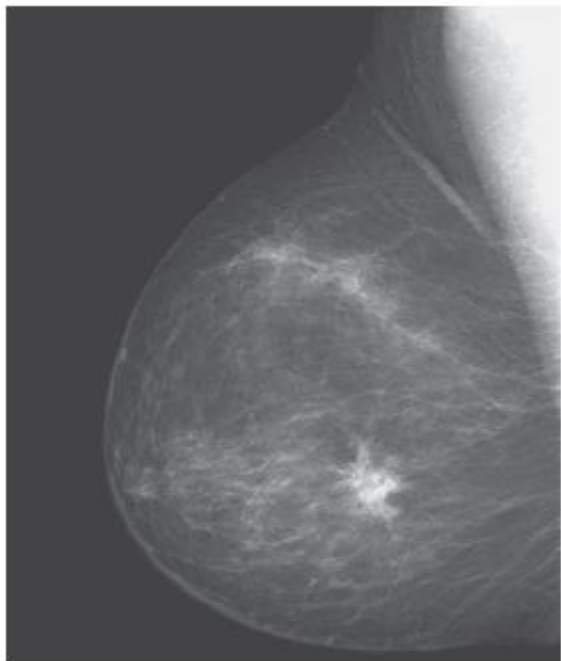


- imhist(工具箱預設顯示的直方圖)

```
>> imhist(f);
```

強度轉換函數-直方圖處理和繪圖函數

圖2.7 各種繪製直方圖的方法

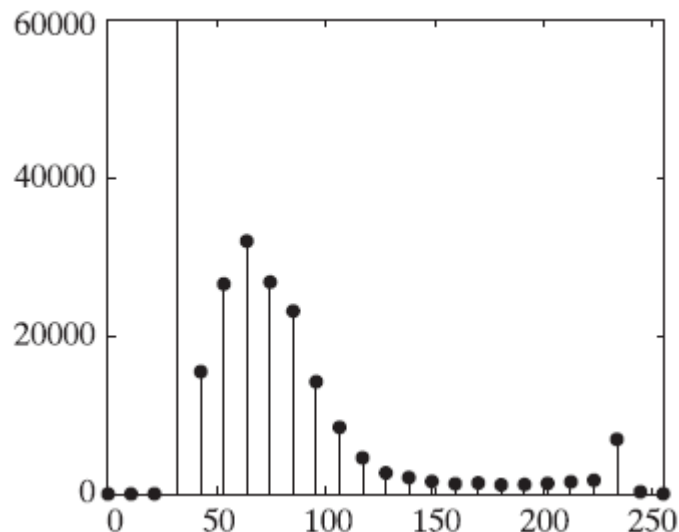
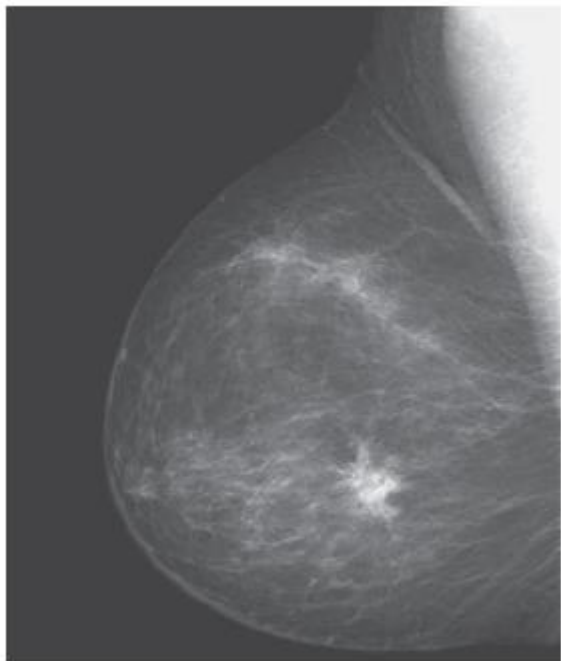


- 長條圖(bar)

```
>> h = imhist(f, 25);  
>> horz = linspace(0, 255, 25);  
>> bar(horz, h)  
>> axis([0 255 0 60000])  
>> set(gca, 'xtick', 0:50:255)  
>> set(gca, 'ytick', 0:20000:60000)
```

強度轉換函數-直方圖處理和繪圖函數

圖2.7 各種繪製直方圖的方法

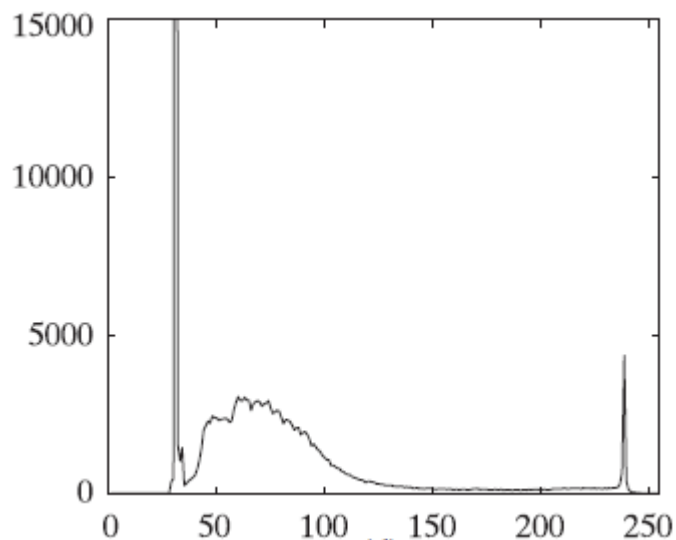
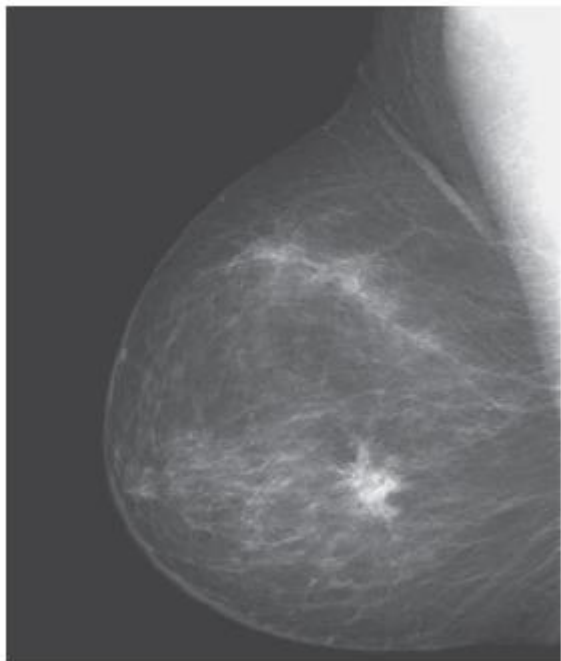


- 針狀圖(stem)

```
>> h = imhist(f, 25);  
>> horz = linspace(0, 255, 25);  
>> stem(horz, h, 'fill')  
>> axis([0 255 0 60000])  
>> set(gca, 'xtick', [0:50:255])  
>> set(gca, 'ytick', [0:20000:60000])
```


強度轉換函數-直方圖處理和繪圖函數

圖2.7 各種繪製直方圖的方法



- plot

```
>> hc = imhist(f);  
>> plot(hc) % 使用預設值。  
>> axis([0 255 0 15000])  
>> set(gca, 'xtick', [0:50:255])  
>> set(gca, 'ytick', [0:2000:15000])
```

強度轉換函數-直方圖處理和繪圖函數

- 表2.1 使用在函數stem和plot的顏色、線和標記的設定

顏色設定		線的設定		標記設定	
符號	顏色	符號	線的型態	符號	標記
k	黑色	-	實線	+	加號
w	白色	--	虛線	o	圓
r	紅色	:	點線	*	星號
g	綠色	-.	虛點線	.	點
b	藍色			x	交叉
c	青色			s	正方形
y	黃色			d	菱形
m	紫紅色			^	上三角
				v	下三角
				>	右三角
				<	左三角
				p	五角星
				h	六角星

強度轉換函數-直方圖處理和繪圖函數

■ 函數處理權fplot，基本語法

```
fplot(fhandle, limits, 'LineSpec')
```

其中fhandle 是一個函數處理權，而limits 指定 x 軸限制在[xmin xmax] 的向量

- 例如繪製雙曲切線函數tanh，在[-2 2] 範圍內使用虛線，可寫成

```
>> fhandle = @tanh;  
>> fplot(fhandle, [-2 2], ':')
```

強度轉換函數-直方圖等化

- 假設強度等級是正規化在 $[0,1]$ 範圍內的連續數值，令 $p_r(r)$ 為影像強度等級的機率密度函數(probability density function，簡稱PDF)，假設我們對輸入的強度等級執行下列轉換來得到輸出的強度等級 s

$$s = T(r) = \int_0^r p_r(w) dw$$

其中 w 是一個仿造(dummy)的積分變數

- 它可以證明輸出的強度等級的機率密度函數是均勻分佈(uniform)，亦即

$$p_s(s) = \begin{cases} 1 & \text{for } 0 \leq s \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

強度轉換函數-直方圖等化

- 此一強度等級等化(equalization)處理的結果是讓影像增加動態範圍，此將提高對比
- 其實轉換函數實際上只不過是累積分佈函數(cumulative distribution function，簡稱CDF)
- 由於離散變數的本質，影像的直方圖不會是均勻隨機數，所以直方圖等化(histogram equalization)就是將直方圖離散量化
- 令 $p_r(r_j)$ ， $j = 0, 1, 2, \dots, L-1$ ，表示與影像強度等級相關的直方圖，正規化直方圖的值近似於影像每個強度等級發生的機率

$$s_k = T(r_k) = \sum_{j=0}^k p_r(r_j) = \sum_{j=0}^k \frac{n_j}{n}$$

$k = 0, 1, 2, \dots, L-1$ ，其中 s_k 是相對應於輸入影像中 r_k 的輸出影像強度值

強度轉換函數-直方圖等化

- 直方圖等化在工具箱中可藉由函數`histeq`來實現，它的語法是

`g = histeq(f, nlev)`

- 其中`f`是輸入的影像，`nlev`是指定給輸入影像強度等級的數目。假設`nlev`等於`L`(輸入影像可能的強度等級總數)，`histeq`會直接實現轉換函數。假設`nlev`小於`L`，則`histeq`會嘗試分配等級

強度轉換函數-直方圖等化

■ 範例 2.5 直方圖等化

```
>> imshow(f); % 圖2.8(a) ◦  
>> figure, imhist(f) % 圖2.8(b) ◦  
>> ylim('auto')  
>> g = histeq(f, 256);  
>> figure, imshow(g) % 圖2.8(c) ◦  
>> figure, imhist(g) % 圖2.8(d) ◦  
>> ylim('auto')
```

強度轉換函數-直方圖等化

圖2.8

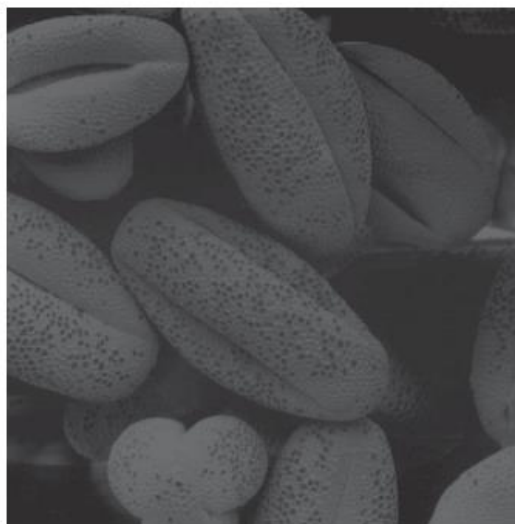
直方圖等化的說明：

(a) 輸入影像

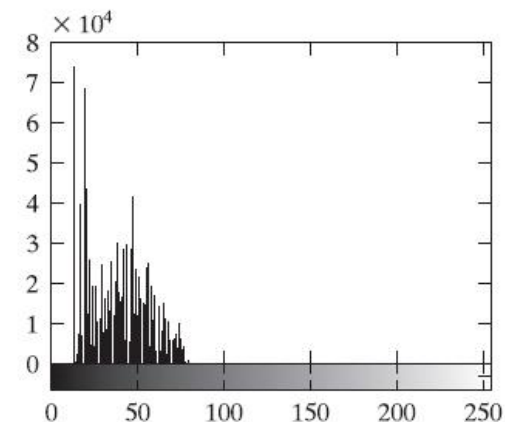
(b) 輸入影像的直方圖

(c) 直方圖等化後的影像

(d) (c) 的直方圖(a) 和 (c)之間的改進很明顯



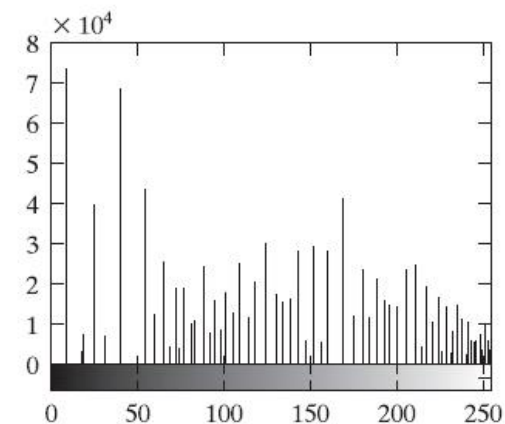
(a)



(b)



(c)



(d)

強度轉換函數-直方圖等化

- 如前敘述，使用在直方圖等化上的轉換函數是正規化直方圖的值的累計總和。我們可以使用 `cumsum` 來得到轉換函數

```
>> hnorm = imhist(f)./numel(f); % 正規化直方圖。  
>> cdf = cumsum(hnorm); % CDF。
```

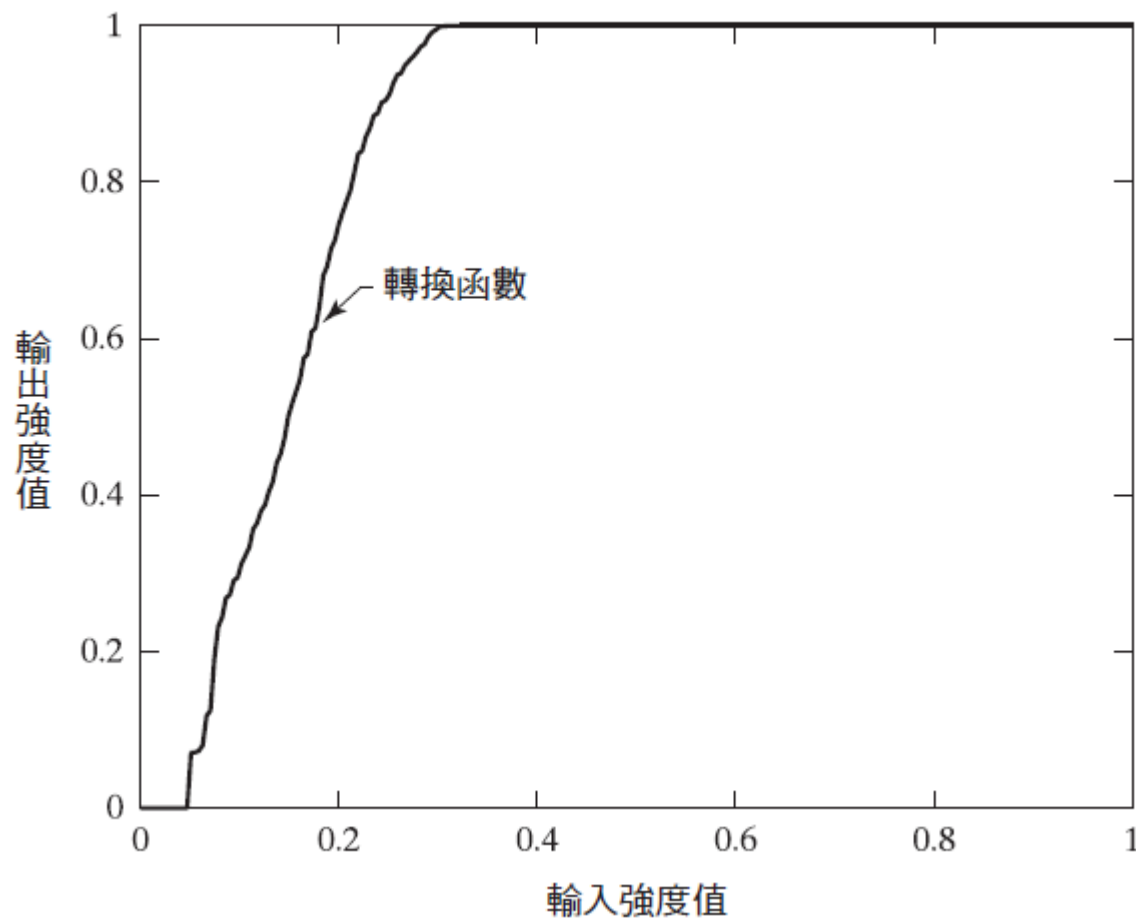
`cdf` 使用下列指令得到圖 2.9：

```
>> x = linspace(0, 1, 256); % [0, 1] 範圍的水平純量。  
>> plot(x, cdf) % 繪製 cdf vs. x。  
>> axis([0 1 0 1]); % 調整、設定和標示。  
>> set(gca, 'xtick', 0:.2:1)  
>> set(gca, 'ytick', 0:.2:1)  
>> xlabel('Input intensity values', 'fontsize', 9)  
>> ylabel('Output intensity values', 'fontsize', 9)
```

強度轉換函數-直方圖等化

圖2.9

圖2.7(a) 的輸入
影像對映到圖
2.7(c) 的輸出影
像的轉換函數



強度轉換函數-直方圖一致(指定)

- 就直方圖等化是依據給定影像的直方圖而言，直方圖等化產生一個自我調整的轉換數。然而，一旦影像的轉換函數已經算出來了，它是不能改變的
- 產生有指定直方圖的影像的方法稱為直方圖一致(histogram matching) 或是直方圖指定(histogram specification)

強度轉換函數-直方圖一致(指定)

- 考慮正規化到 $[0,1]$ 區間的連續強度等級，令 r 和 z 代表輸入和輸出影像的強度等級。輸入等級的機率密度函數為 $p_r(r)$ ，指定的輸出等級機率密度函數為 $p_z(z)$

$$s = T(r) = \int_0^r p_r(w) dw$$

產生強度等級 s 的均勻機率密度函數 $p_s(s)$ 。假設定義有下列性質的變數 z

$$H(z) = \int_0^z p_z(w) dw = s$$

強度轉換函數-直方圖一致(指定)

- 考慮正規化到 $[0,1]$ 區間的連續強度等級，令 r 和 z 代表輸入和輸出影像的強度等級。輸入等級的機率密度函數為 $p_r(r)$ ，指定的輸出等級機率密度函數為 $p_z(z)$

$$s = T(r) = \int_0^r p_r(w) dw$$

產生強度等級 s 的均勻機率密度函數 $p_s(s)$ 。假設定義有下列性質的變數 z

$$H(z) = \int_0^z p_z(w) dw = s$$

$$z = H^{-1}(s) = H^{-1}[T(r)]$$

強度轉換函數-直方圖一致(指定)

- 工具箱使用下列在histeq 的語法來實現直方圖一致：

```
g = histeq(f, hspec)
```

其中f 是輸入影像，hspec 是指定的直方圖(有指定值的列向量)，而g 是輸出影像，它近似的直方圖是hspec

強度轉換函數-直方圖一致(指定)

■ 範例 2.6 直方圖一致

圖2.10 (a) 火星月球Phobos 的影像 (b) 直方圖

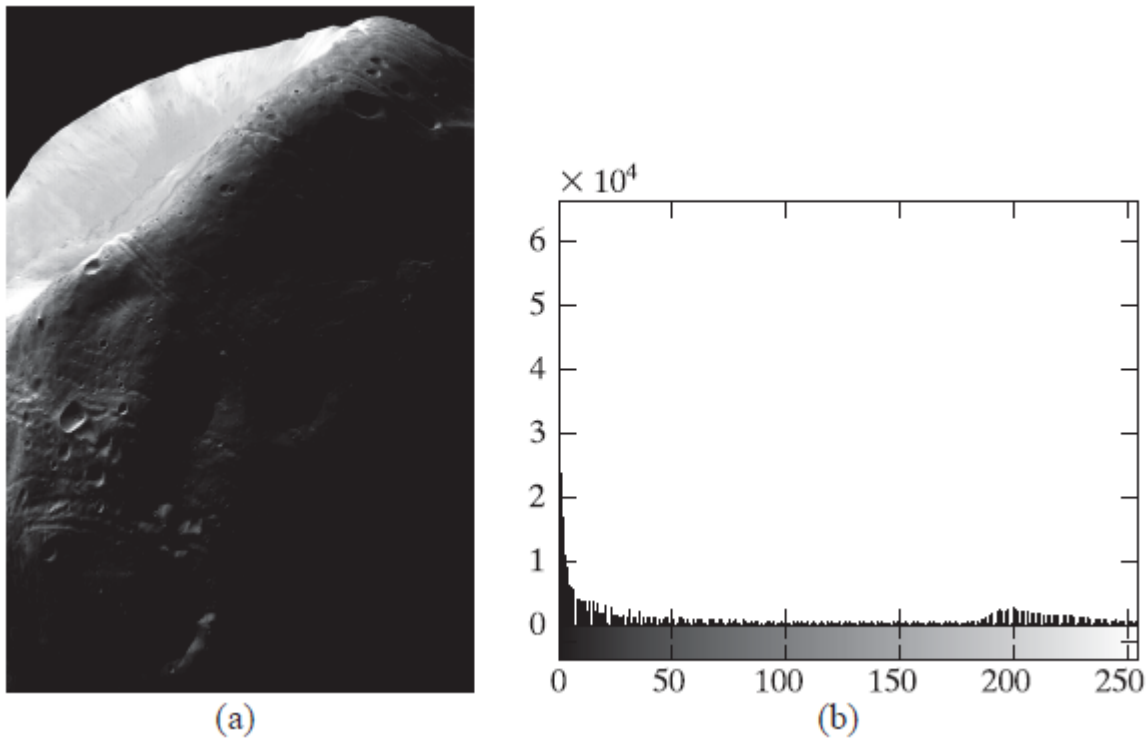
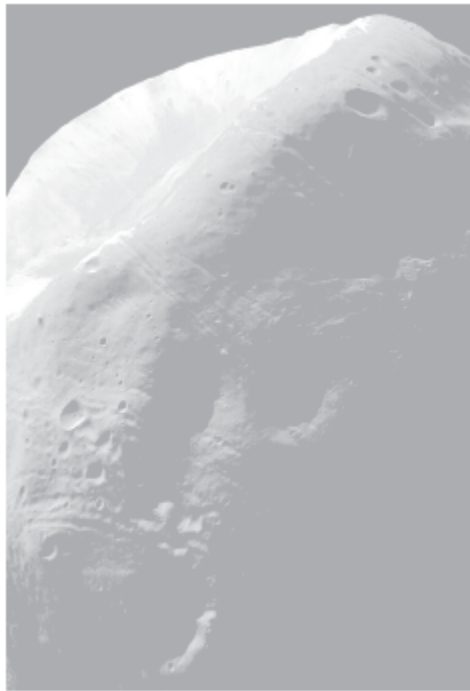


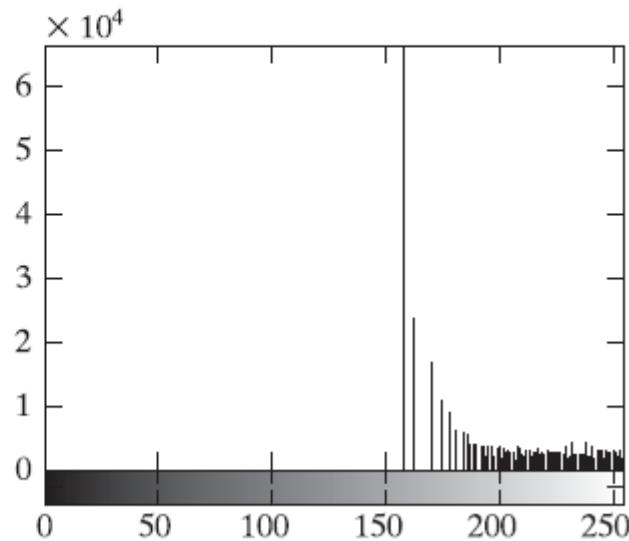
圖2.10(b) 顯示使用imhist(f) 所得到的直方圖

強度轉換函數-直方圖一致(指定)

圖2.10 (c) 直方圖等化後的影像 (d) (c) 的直方圖



(c)



(d)

使用以下的指令得到圖2.10(c) 的結果

```
>> f1 = histeq(f, 256);
```


強度轉換函數-直方圖一致(指定)

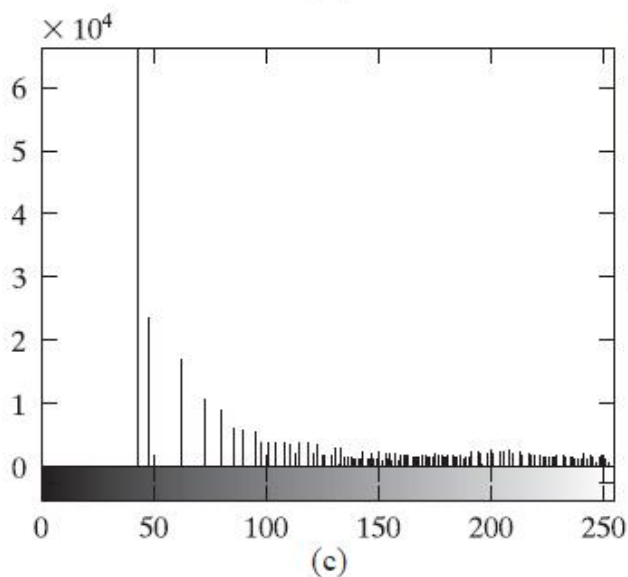
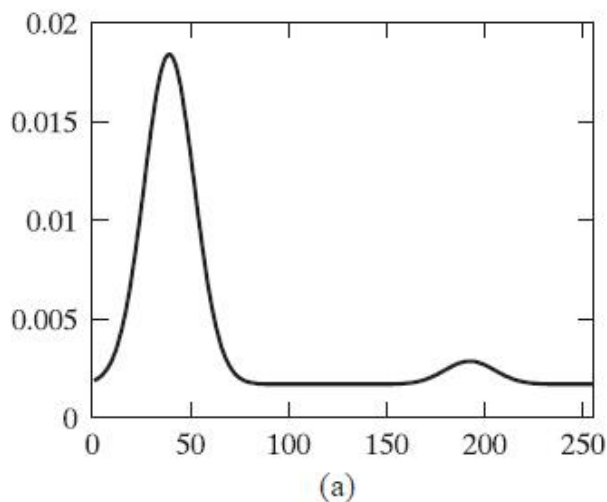
- 圖2.11(a) 是使用程式manualhist 得到的函數，此函數保留原本直方圖的形狀，但在強度等級的黑暗區域中有較平滑的轉移，使用以下的指令可產生指定的直方圖的影像

```
>> g = histeq(f, p);
```

強度轉換函數-直方圖一致(指定)

圖2.11

- (a) 指定的直方圖
- (b) 使用直方圖指定
強化後的結果
- (c) (b)的直方圖



強度轉換函數-函數adapthisteq

- 這是工具箱函數執行所謂的適應對比限制的直方圖等化(contrast-limited adaptive histogram equalization，簡稱CLAHE)
- 此方法是使用直方圖指定，針對影像個別的小區域(稱為瓷磚(tiles))做處理。相鄰的磁磚使用雙線性內插來消除磁磚間的邊緣，尤其在同質強度區域的對比可以被限制以防雜訊擴大
- adapthisteq 的語法為

```
g = adapthisteq(f, param1, val1, param2, val2, ...)
```

其中f是輸入影像，g是輸出影像，成對的param與val列在表2.2

強度轉換函數－函數 `adapthisteq`

表 2.2 使用在函數 `adapthisteq` 的參數和對應的值

參數	值
'NumTiles'	由列數和行數來指定磁磚的個數，它是兩個正值元素的向量 <code>[r c]</code> ， <code>r</code> 和 <code>c</code> 至少是 2，磁磚的個數等於 <code>r*c</code> ，預設值是 <code>[8 8]</code> 。
'ClipLimit'	在 <code>[0, 1]</code> 範圍內的純量，用來指定對比強的限制，值越高對比越高，預設值是 0.01。
'NBins'	用來指定直方圖區間個數的正整數，此直方圖是用在建立一個對比強化的轉換，值越高動態範圍越高，但處理的速度越慢，預設值是 256。
'Range'	指定輸出影像資料範圍的字串： 'original' — 原始影像的範圍， <code>[min(f(:)) max(f(:))]</code> 。 'full' — 使用輸出影像類別完整的範圍，比如， <code>uint8</code> 的資料是 <code>[0 255]</code> ，這也是預設值。
'Distribution'	指定影像磁磚直方圖形態的字串： 'uniform' — 平坦直方圖（預設）。 'rayleigh' — 鐘形直方圖。 'exponential' — 曲線直方圖。 (4.2.2 節有這些分佈的方程式)
'Alpha'	應用到 Rayleigh 和指數型分佈的非負純量，預設值是 0.4。

強度轉換函數-函數adapthisteq

■ 範例 2.7 函數adapthisteq的使用

圖2.12 (a) 與圖2.10(a) 相同的影像；(b) 使用函數adapthisteq 預設值的結果；(c) 使用相同的函數但是參數NumTiles 設定為[25 25]；(d) 使用相同的磁磚個數而且ClipLimit = 0.05



(a)



(b)



(c)



(d)

強度轉換函數-函數adapthisteq

- 圖2.12(a) 與圖2.10(a) 一樣，而圖2.12(b) 是在函數 adapthisteq使用預設設定的結果：

```
>> g1 = adapthisteq(f);
```

- 圖2.12(c) 顯示將磁磚增為[25 25] 的結果：

```
>> g2 = adapthisteq(f, 'NumTiles', [25 25]);
```

- 使用下列指令

```
>> g3 = adapthisteq(f, 'NumTiles', [25 25], 'ClipLimit', 0.05);
```

產生的結果在圖2.12(d)

空間濾波

空間濾波

- 鄰域處理(neighborhood processing) 或空間濾波(spatial filtering)
 - (1) 選擇一個中心點(x, y)
 - (2) 針對該中心點事先定義的鄰域上的像素執行運算
 - (3) 稱該點運算的結果為處理過程的「響應」
 - (4) 對影像上的每一點重覆此程序。對於每一個在輸入影像的像素，移動中心點會產生新的鄰域

空間濾波-線性空間濾波

- 線性濾波(linear filtering) 的觀念來自於在頻率域上信號處理的傅立葉轉直接在影像的像素上執行濾波運算
- 使用線性空間濾波(linear spatial filtering) 來與頻率域濾波(frequency domain filtering) 做區別。
- 線性運算是將鄰域裡的每一個像素乘上一個對應的係數並將結果加總來得到在每一個點 (x, y) 的響應

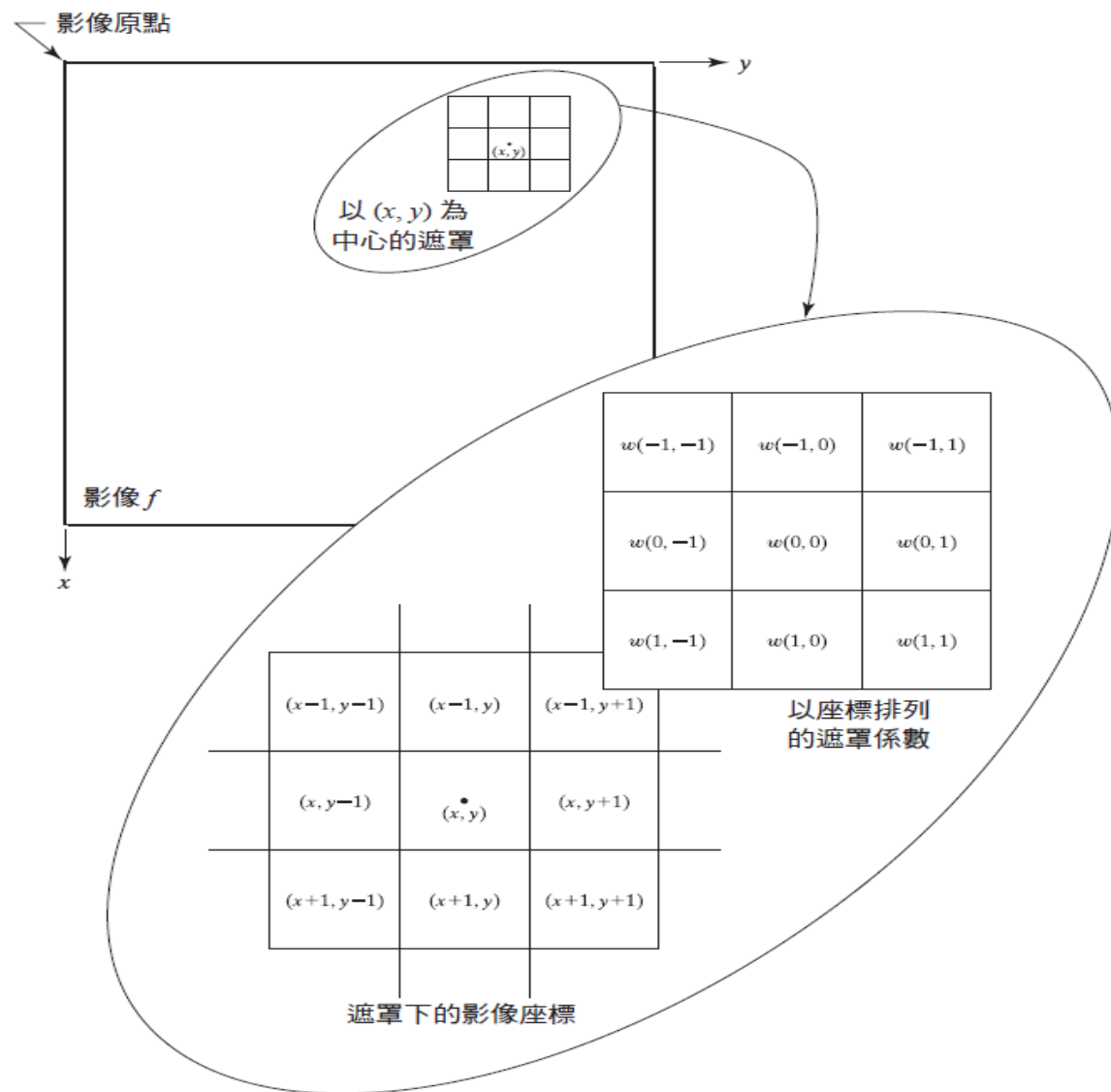
空間濾波-線性空間濾波

- 濾波器(filer)、遮罩(mask)、濾波器遮罩(filter mask)、核心(kernel)、模版(template)或窗(window)
 - 鄰域大小為 $m \times n$ 的矩陣
 - 濾波器在每一個點 (x, y) 的響應是濾波係數與濾波器遮罩所涵蓋區域對應的鄰域像素的乘積再加總
 - 對於一個大小為 $m \times n$ 的遮罩，通常假設 m
 n 是奇數，其中有意義又最小的遮罩是 3×3

空間濾波-線性空間濾波

圖2.13

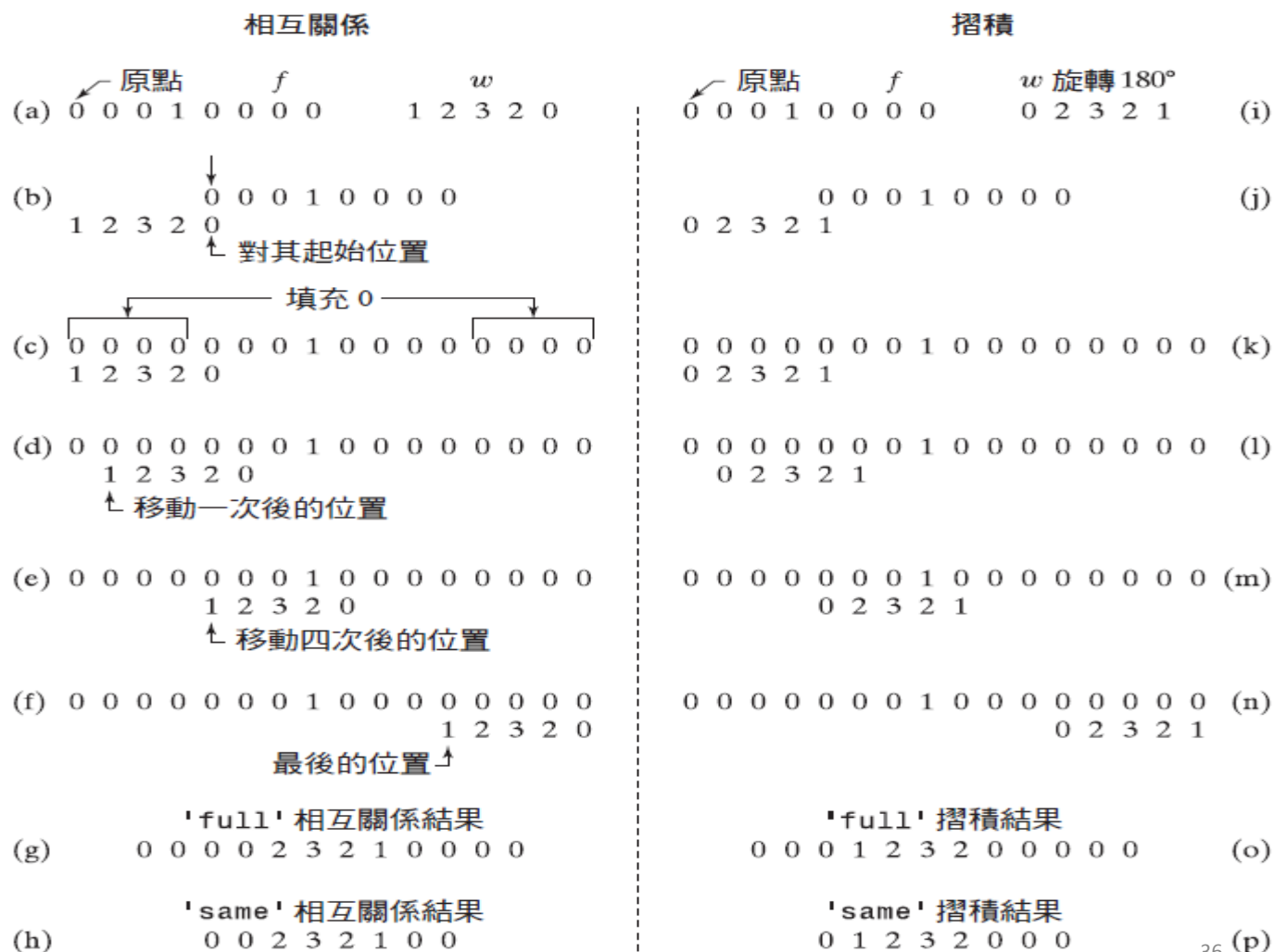
線性濾波器的技巧， 3×3 濾波器遮罩的放大圖與在它下方的影像鄰域



空間濾波-線性空間濾波

■ 相互關係(correlation)與摺積(convolution)

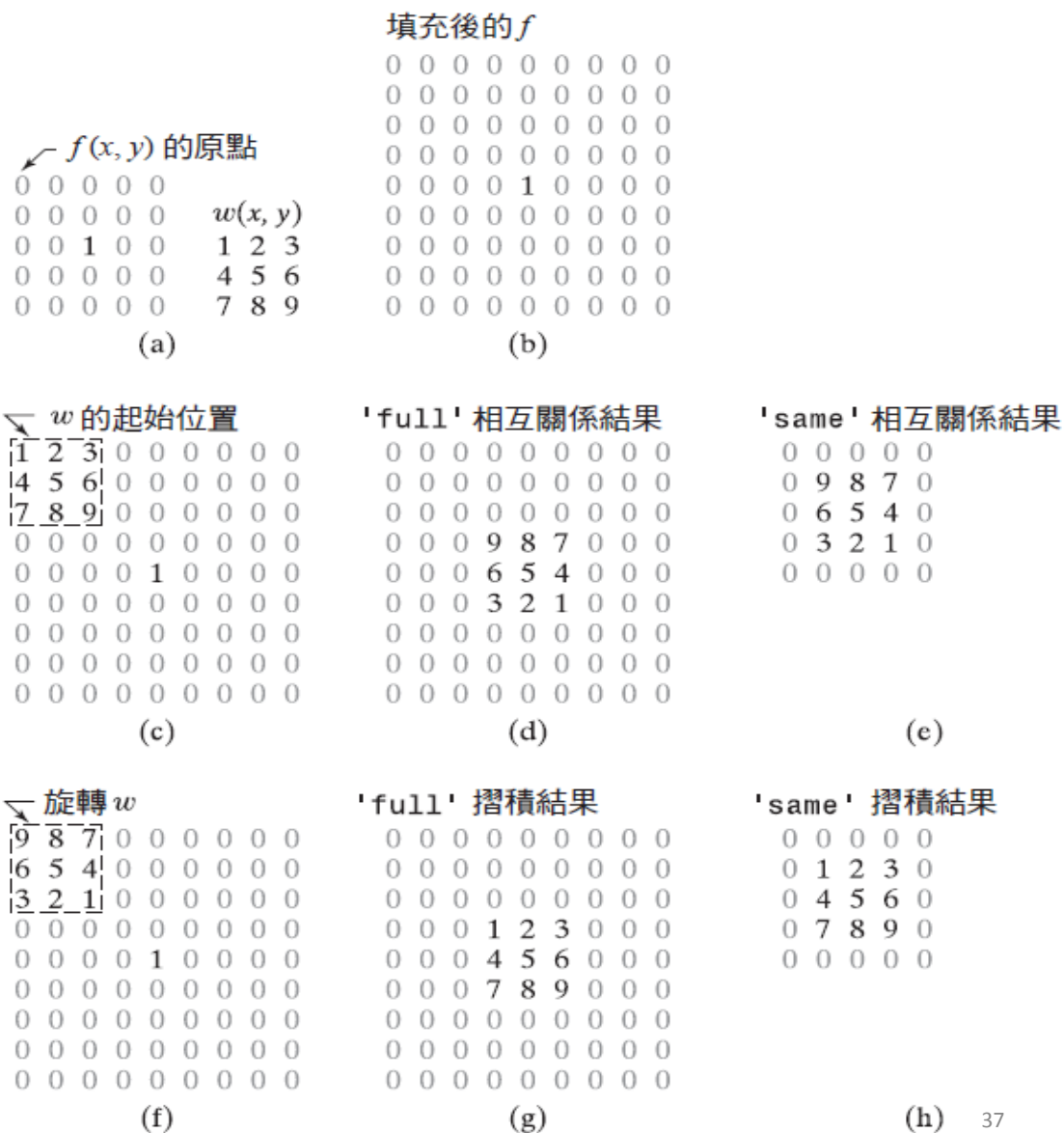
圖2.14
一維相互關係
與摺積
的說明



空間濾波-線性空間濾波

圖2.15

二維相互關係與摺積的說明，用灰色顯示的0是為了方便觀看



空間濾波-線性空間濾波

- 對於一個大小為 $m \times n$ 的濾波器遮罩 $w(x, y)$ 與函數 $f(x, y)$ 的相互關係

$$w(x, y) \star f(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)$$

- 計算變數 x 和 y 所有位移(displacement)的值，所以 w 的所有元素要經過 f 的每一個像素
- 通常 m 和 n 都被假設為奇數

- $w(x, y)$ 與 $f(x, y)$ 的摺積，

$$w(x, y) \star f(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x - s, y - t)$$

- 方程式右邊的負號是將 f 旋轉 180° 。對 f 旋轉和平移而不是對 w 是為了簡化表示式

空間濾波-線性空間濾波

- 工具箱使用函數imfilter 來實現線性空間濾波，此函數有下列的語法：

```
g = imfilter(f, w, filtering_mode, boundary_options, size_options)
```

其中f 是輸入影像，w 是濾波遮罩，g 是濾波後的結果，其它參數參閱表2.3

- imfilter 一般的語法是

```
g = imfilter(f, w, 'replicate')
```

空間濾波-線性空間濾波

表2.3 函數imfilter的選項

選項	描述
濾波模式	
'corr'	使用相互關係執行濾波（見圖 2.14 和 2.15），此為預設值。
'conv'	使用摺積執行濾波（見圖 2.14 和 2.15）。
邊界選項	
P	以 P 填充來延伸輸入影像的邊界（不用引號），此為預設情況，且值為 0。
'replicate'	複製影像外部邊界的值來延伸影像的大小。
'symmetric'	跨過邊界鏡射影像的值來延伸影像的大小。
'circular'	使用一個二維週期函數的一個週期來延伸影像的大小。
大小選項	
'full'	輸出影像大小與延伸後（填充後）的影像相同（見圖 2.14 和 2.15）。
'same'	輸出影像大小與輸入一樣，這可由限制濾波器遮罩中心在原始影像所包含的點內移動（見圖 2.14 和 2.15）來完成。此為預設情況。

空間濾波-線性空間濾波

- 當濾波器既不是先旋轉也不是對稱，而且我們又希望執行摺積，我們有兩種選擇，一是使用語法

```
g = imfilter(f, w, 'conv', 'replicate')
```

另一種方法是使用函數`rot90(w,2)`先將`w`旋轉180°然後使用`imfilter(f,w,'replicate')`，這兩個步驟可以結合成一個：

```
g = imfilter(f, rot90(w, 2), 'replicate')
```

此結果產生影像`g`，其大小和輸入相同

空間濾波-線性空間濾波

■ 範例 2.8 函數imfilter的使用

- 圖2.16(a) 是大小為 512×512 、類別為double 的影像f，考慮 31×31 的濾波器

```
>> w = ones(31);
```

- 圖2.16(b) 顯示執行下列濾波運算的結果

```
>> gd = imfilter(f, w);  
>> imshow(gd, [ ])
```

空間濾波-線性空間濾波

- 在濾波的影像中黑色與與白色之間的邊緣是模糊的，而在影像亮的部份和邊界之間的邊緣也是如此，原因是填充的邊緣是黑色的。我們使用‘replicate’選項來處理此一困擾

```
>> gr = imfilter(f, w, 'replicate');  
>> figure, imshow(gr, [ ])
```

- 使用‘symmetric’選項會有等效的結果(圖2.16(d))

```
>> gs = imfilter(f, w, 'symmetric');  
>> figure, imshow(gs, [ ])
```

- 使用‘circular’選項產生圖2.16(e)的結果

```
>> gc = imfilter(f, w, 'circular');  
>> figure, imshow(gc, [ ])
```

空間濾波-線性空間濾波

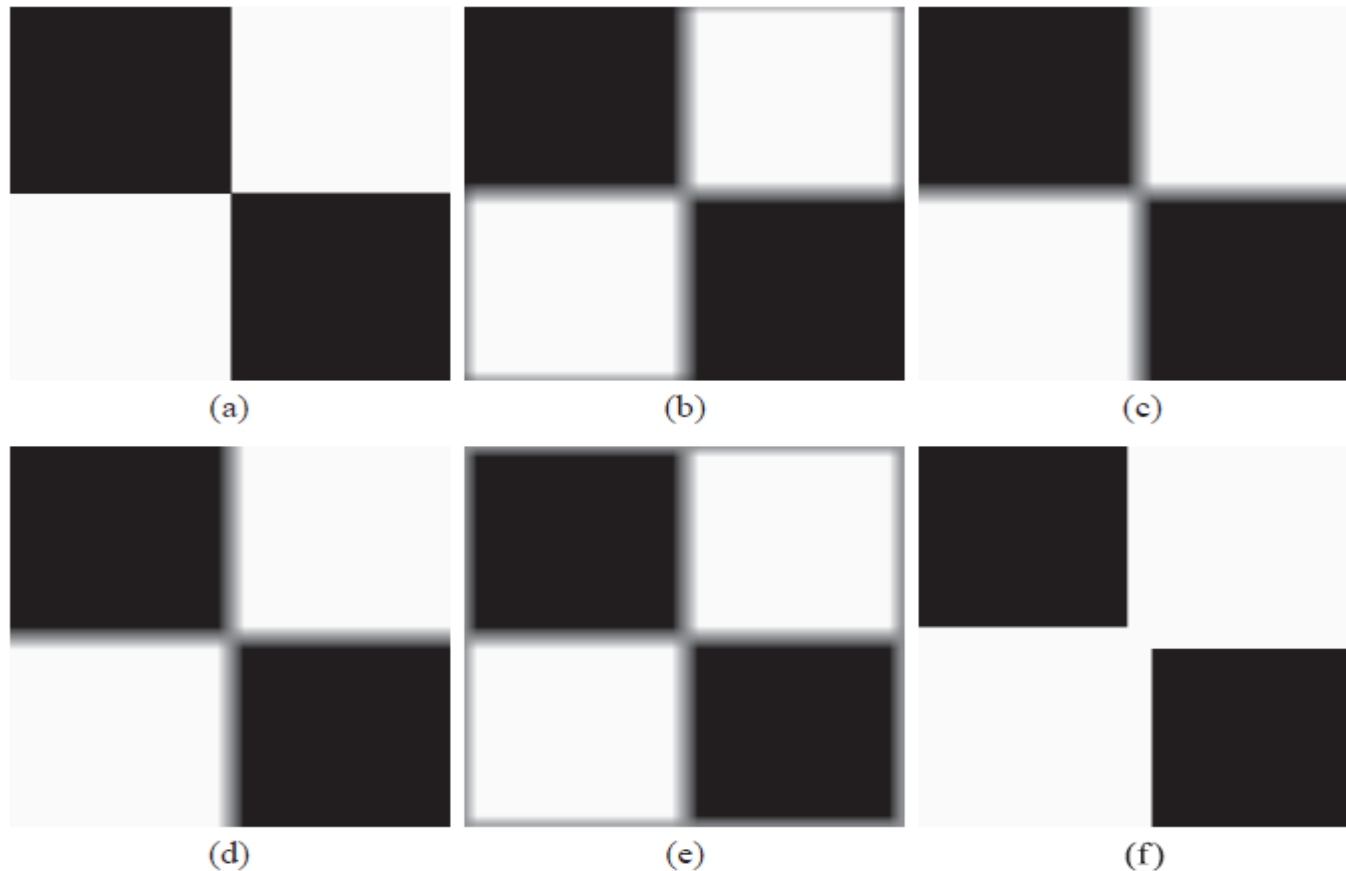


圖 2.16

(a) 原圖；(b) 使用函數 `imfilter` 預設 0 填充的結果；(c) 使用 'replicate' 選項的結果；(d) 使用 'symmetric' 選項的結果；(e) 使用 'circular' 選項的結果；(f) 將原始影像轉為類別 `uint8` 然後使用 'replicate' 的選項濾波，全部使用全為 1 且大小為 31×31 的濾波器

空間濾波-線性空間濾波

- `imfilter` 產生與輸入相同類別的結果可能會導致困擾(圖2.16(f))

```
>> f8 = im2uint8(f);  
>> g8r = imfilter(f8, w, 'replicate');  
>> figure, imshow(g8r, [ ])
```

空間濾波-非線性空間濾波

- 非線性空間濾波也是以鄰域運算為基礎
 - 工具箱提供兩個執行一般非線性濾波的函數：
nlfilter 和 colfilt
 - nlfilter 直接在二維上運算， colfilt 則是將資料以行的形式組建
 - colfilt 需要較多的記憶體，但是執行的速度一般都比nlfilter 快很多
 - 已知一大小為 $M \times N$ 的影像 f ，以及一大小為 $m \times n$ 的鄰域，函數 colfilt 產生一個大小最大為 $mn \times MN$ 的矩陣 A
 - 每一行對應一些像素，它們是以影像某個位置為中心的鄰域所涵蓋的像素

空間濾波-非線性空間濾波

- colfilt 使用零填充來處理所有需要的填充

- 函數colfilt 的語法是

```
g = colfilt(f, [m n], 'sliding', fun)
```

其中 m 和 n 是濾波器的維度，‘sliding’表示在輸入影像 f 中，從一個像素滑動 $m \times n$ 區域到另一個像素， fun 是函數處理權

- 函數 fun 對 A 的每一行個別運算，並回傳一個列向量 v ， v 的第 k 個元素是 fun 對 A 的第 k 行執行運算的結果，因為 A 最多可到 MN 行，所以 v 的最大維度為 $1 \times MN$

影像處理工具箱標準的 空間濾波器

影像處理工具箱標準的空間濾波器-線性空間濾波器

■ 二維線性空間濾波器

■ 函數fspecial產生濾波器遮罩w

```
w = fspecial('type', parameters)
```

其中'type'指定濾波器型式，而parameters進一步定義所指定的濾波器。由fspecial產生的空間濾波器整理在表2.5中

影像處理工具箱標準的空間濾波器-線性空間濾波器

表2.5

由函數

fspecial支援
的空間濾波

類型	語法和參數
'average'	fspecial('average',[r c])，一個大小為 $r \times c$ 的矩形平均濾波，其預設值為 3×3 ，如果是正方形濾波器，就使用一個數字來取代 [r c]。
'disk'	fspecial('disk',r)，一個半徑為 r 的圓形平均濾波器（在大小為 $2r+1$ 的正方形內），預設半徑為 5。
'gaussian'	fspecial('gaussian',[r c],sig)，一個大小為 $r \times c$ 的高斯低通濾波器而且標準差為 sig（正值），預設值為 3×3 和 0.5，如果是正方形濾波器，就使用一個數字來取代 [r c]。
'laplacian'	fspecial('laplacian',alpha)，一個大小為 3×3 的拉普拉斯濾波器，alpha 的範圍在 [0, 1] 之間用來指定形狀，alpha 的預設值為 0.2。
'log'	fspecial('log',[r c],sig)，大小為 $r \times c$ 的高斯拉普拉斯濾波器 (LoG) 而且標準差為 sig（正值）。預設值為 5×5 和 0.5，如果是正方形濾波器，就使用一個數字來取代 [r c]。
'motion'	fspecial('motion',len,theta)，當與一個影像摺積時，以近似 len 個像素的線性運動（照相機相對於影像）輸出一個濾波器，其運動方向為 theta，從水平依逆時針方向以度為單位測量，預設值為 9 和 0，代表在水平方向移動 9 個像素。
'prewitt'	fspecial('prewitt')，輸出一個 3×3 的 Prewitt 遮罩 wv，它近似一個垂直梯度，將此結果轉置得到水平梯度的遮罩：wh=wv'。
'sobel'	fspecial('sobel')，輸出一個 3×3 的 Sobel 遮罩 sv，它近似一個垂直梯度，將此結果轉置得到水平梯度的遮罩：sh=sv'。
'unsharp'	fspecial('unsharp',alpha)，輸出一個 3×3 的 (unsharp) 濾波器，alpha 的範圍在 [0, 1] 之間用來控制形狀，alpha 的預設值為 0.2。

影像處理工具箱標準的空間濾波器-線性空間濾波器

■ 範例 2.10 使用函數imfilter 實現拉普拉斯濾波器

■ 影像 $f(x, y)$ 的拉普拉斯(記作 $\nabla^2 f(x, y)$)定義為

$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2}$$

一般二階導數常用的數位近似是

$$\frac{\partial^2 f(x, y)}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

和

$$\frac{\partial^2 f(x, y)}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

所以

$$\nabla^2 f(x, y) = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] - 4f(x, y)$$

影像處理工具箱標準的空間濾波器-線性空間濾波器

- 將影像的每一個點與下列的空間遮罩執行摺積來實現此表示式：

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

- 數位二階導數的另一種定義是將對角線的元素列入考慮，因此可以使用以下的遮罩來實現

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

- 兩個導數的定義有時候與此處的正負號相反，這將導致此遮罩與前述兩個遮罩差一個負號

影像處理工具箱標準的空間濾波器-線性空間濾波器

■ 拉普拉斯強化

$$g(x, y) = f(x, y) + c[\nabla^2 f(x, y)]$$

其中 $f(x, y)$ 是輸入影像， $g(x, y)$ 是強化的影像，

- 如果遮罩中心的係數是正的，則 c 是1；否則是-1
- 因為拉普拉斯是導數運算子，所以它使影像銳化，但會使固定區域變為零，所以將原本影像加回來以回復灰階色澤

■ 函數`fspecial('laplacian',alpha)` 實現更通用的拉普拉斯遮罩：

$$\begin{array}{ccc} \frac{\alpha}{1+\alpha} & \frac{1-\alpha}{1+\alpha} & \frac{\alpha}{1+\alpha} \\ \frac{1-\alpha}{1+\alpha} & -4 & \frac{1-\alpha}{1+\alpha} \\ \frac{\alpha}{1+\alpha} & \frac{1-\alpha}{1+\alpha} & \frac{\alpha}{1+\alpha} \end{array}$$

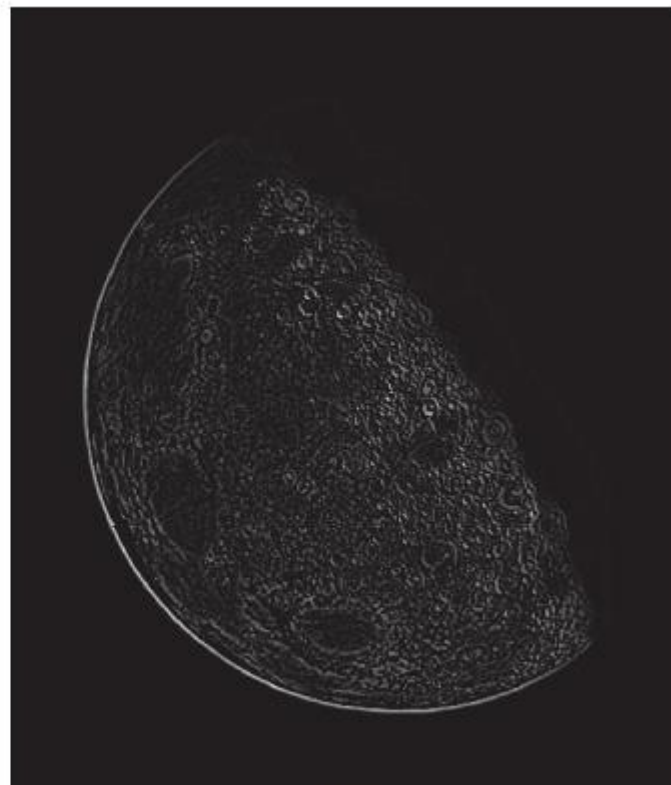
影像處理工具箱標準的空間濾波器-線性空間濾波器

圖2.17

(a) 月球北極的影像；(b) 拉普拉斯使用uint8 格式濾波後的影像(因為uint8 沒有正負號，所以負值在輸出時被設為0)



(a)



(b)

影像處理工具箱標準的空間濾波器-線性空間濾波器

■ 拉普拉斯濾波器

```
>> w = fspecial('laplacian', 0)
```

```
w =
```

```
    0.0000    1.0000    0.0000  
    1.0000   -4.0000    1.0000  
    0.0000    1.0000    0.0000
```

- 此濾波器的類別是double，而且alpha = 0
- 手動方式

```
>> w = [0 1 0; 1 -4, 1; 0 1 0];
```

■ 將w 運用到類別為uint8 的輸入影像f(圖.17(a))：

```
>> g1 = imfilter(f, w, 'replicate');  
>> imshow(g1, [ ])
```

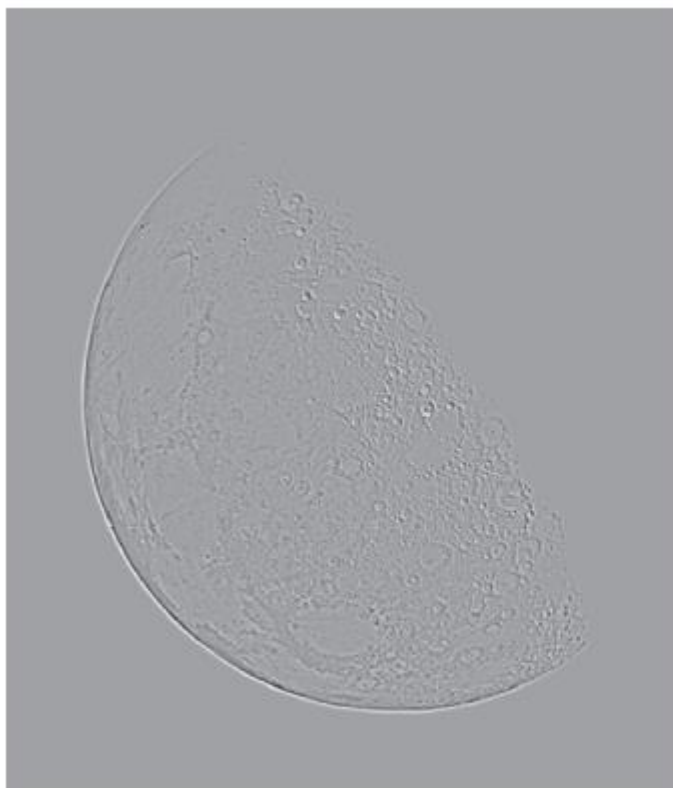
影像處理工具箱標準的空間濾波器-線性空間濾波器

- 圖2.17(b) 是運算結果看起來合理，但有一個問題：所有的像素都是正的
 - 由於濾波器中心的係數是負的，所以通常預期會有正負值的拉普拉斯影像
 - 以imfilter 濾波得到的影像和輸入影像有相同的類別，所以負值被截去。因此在濾波之前先將f 轉為浮點來避免此一困擾(圖2.17(c) 的結果是典型的拉普拉斯影像)

```
>> f2 = tofloat(f);  
>> g2 = imfilter(f2, w, 'replicate');  
>> imshow(g2, [ ])
```


影像處理工具箱標準的空間濾波器-線性空間濾波器

圖2.17 (c) 拉普拉斯使用浮點濾波後的影像；(d) (a) 減掉 (c) 強化的結果(原圖來自NASA)



(c)



(d)

影像處理工具箱標準的空間濾波器-線性空間濾波器

- 將原始影像減去(因為中心係數是負的)拉普拉斯影像來回復失去的灰階色調：

```
>> g = f2 - g2;  
>> imshow(g);
```

圖2.17(d) 的結果比原本影像銳化。

影像處理工具箱標準的空間濾波器-線性空間濾波器

- 範例 2.11 人為指定濾波器及強化技巧的比較
 - 工具箱支援一個中心為-4 的 3×3 拉普拉斯濾波器
 - 更好的銳化是使用一個-8 在中心且四周環繞1 的 3×3 拉普拉斯濾波器

```
>> f = imread('Fig0217(a).tif');  
>> w4 = fspecial('laplacian', 0); % 與範例 2.10 的 w 相同。  
>> w8 = [1 1 1; 1 -8 1; 1 1 1];  
>> f = tofloat(f);  
>> g4 = f - imfilter(f, w4, 'replicate');  
>> g8 = f - imfilter(f, w8, 'replicate');  
>> imshow(f)  
>> figure, imshow(g4)  
>> figure, imshow(g8)
```

影像處理工具箱標準的空間濾波器-線性空間濾波器

圖2.18

(a) 月球北極的影像；
(b) 使用拉普拉斯濾波器
'laplacian'(-4 在中心)強化後的影像；
(c) 使用拉普拉斯
(-8 在中心)濾波器
強化後的影像



(a)



(b)



(c)

影像處理工具箱標準的空間濾波器-非線性空間濾波

- 函數ordfilt2 計算順序統計濾波器(order-statistic filter)(也稱為級別濾波器(rank filter))
 - 此濾波器的響應是依據影像鄰域中像素的排序(級別)，來取代鄰域中心的像素值。
 - 函數ordfilt2 的語法

```
g = ordfilt2(f, order, domain)
```

- 此函數是在domain 範圍內，非零元素所指定的相鄰像素排序後的第order 個元素取代f 的每一個元素來產生輸出影像g
- domain 是由1和0 組成的 $m \times n$ 矩陣，它指定哪些鄰域中的像素位置可以被計算

影像處理工具箱標準的空間濾波器-非線性空間濾波

■ 最小值濾波器(min filter)

- 排序後的第一個樣本(指百分比第0 個)
- 使用下列的語法實現大小為 $m \times n$ 的最小值濾波器(min filter)(級別1)

`g = ordfilt2(f, 1, ones(m, n))`

- 1 表示 mn 個樣本排序後的第一個樣本，而`ones(m,n)` 產生全為1 的 $m \times n$ 矩陣，用來表示鄰域中的所有樣本都要參與計算

■ 最大值濾波器(max filter)

- 百分比第100 個是排序後的最後一個樣本，也就是第 mn 個樣本

`g = ordfilt2(f, m*n, ones(m, n))`

影像處理工具箱標準的空間濾波器-非線性空間濾波

■ 中間值註濾波器(median filter)

■ 百分比第50個樣本

```
g = ordfilt2(f, (m*n + 1)/2, ones(m, n))
```

其中 m 和 n 是奇數

■ 工具箱提供二維中間值濾波器專用的實現

```
g = medfilt2(f, [m n], padopt)
```

- 其中 $[m\ n]$ 定義一個要計算中間值的鄰域，其大小為 $m \times n$
- padopt 指定可能的邊界填充選項：
 - ‘zeros’(預設)、‘symmetric’(其中 f 是跨過邊界作鏡射的對稱延伸)以及‘indexed’(如果 f 的類別為double，則 f 用1填充；否則用0填充)

影像處理工具箱標準的空間濾波器-非線性空間濾波

- 此函數的預設是

```
g = medfilt2(f)
```

它使用 3×3 的鄰域來計算中間值，並以0 來填充輸入影像的邊界

影像處理工具箱標準的空間濾波器-非線性空間濾波

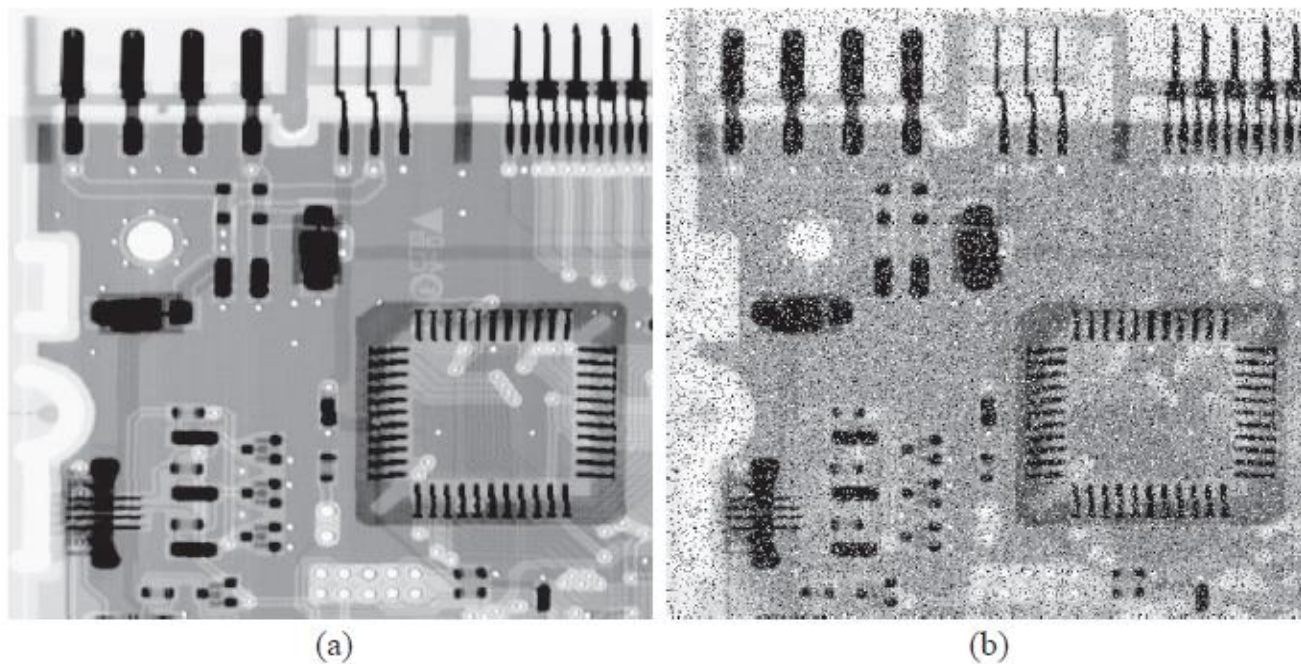
- 範例 2.12 函數medfilt2的中間值濾波
 - 中間值濾波對於減少胡椒鹽雜訊是一個有用的工具
 - 圖2.19(b) 是被胡椒鹽雜訊破壞的同一張影像，其中黑點和白點的發生機率都是 0.2，這張影像是使用函數imnoise 產生的好的銳化是使用一個-8 在中心且四周環繞1 的 3×3 拉普拉斯濾波器

```
>> fn = imnoise(f, 'salt & pepper', 0.2);
```

影像處理工具箱標準的空間濾波器-非線性空間濾波

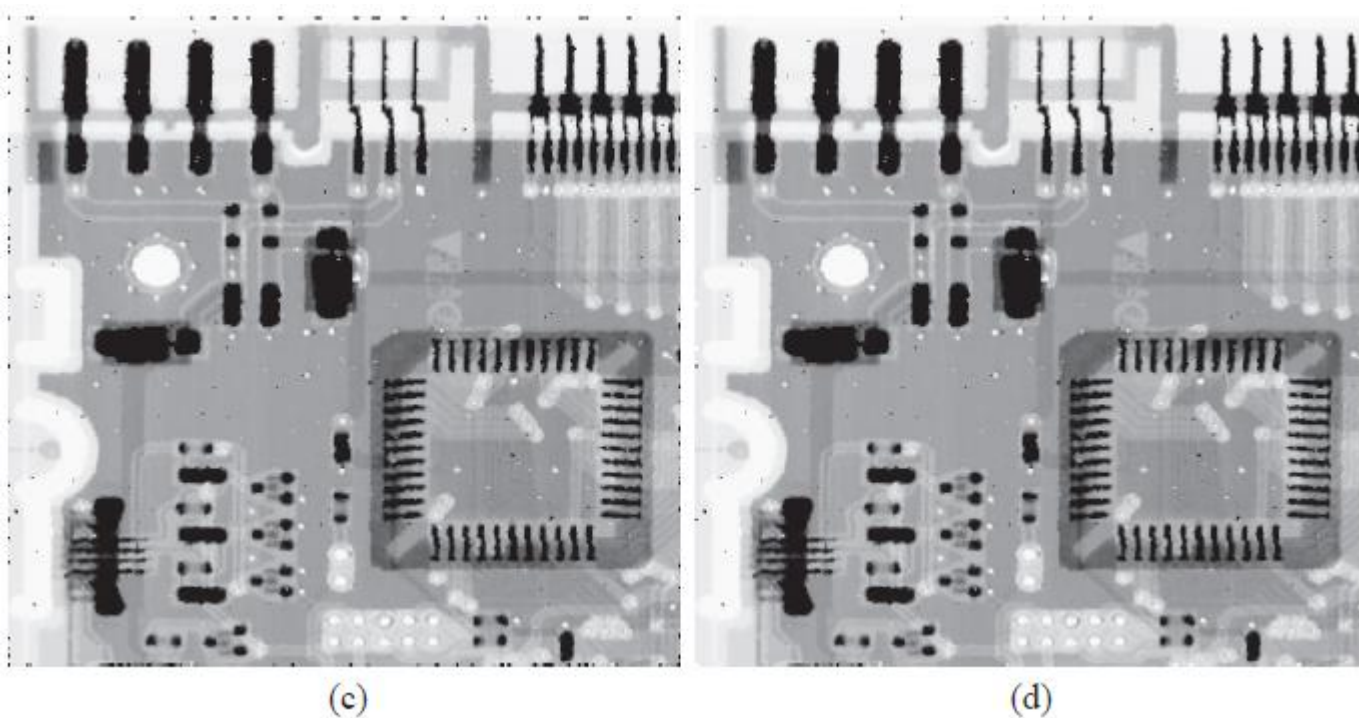
圖2.19

中間值濾波：(a) X 光影像；(b)被胡椒鹽雜訊破壞的影像



影像處理工具箱標準的空間濾波器-非線性空間濾波

圖2.19 (c) 使用medfilt2預設值的中間值濾波的結果；
(d) 使用'symmetric'選項的中間值濾波的結果，注意(d) 與(c) 之間邊框的改進(原圖來自Lixi, Inc.)



影像處理工具箱標準的空間濾波器-非線性空間濾波

- 圖2.19(c) 是針對雜訊影像使用下列陳述作中間值濾波的結果

```
>> gm = medfilt2(fn);
```

- 圖2.19(c)圖2.19(d) 是使用'symmetric'的結果

```
>> gms = medfilt2(fn, 'symmetric');
```