

# 影像處理

## (Image Processing)

真理大學 資訊工程系 吳汶涓老師

### Course 5

#### 鄰域處理



# Outline

5.1 前言

5.2 運算符號

5.3 Matlab中的濾波函數

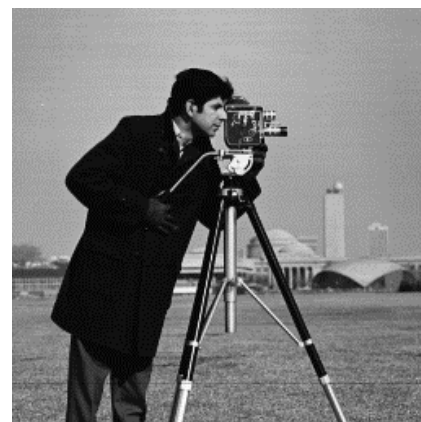
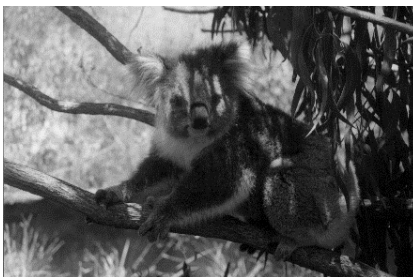
5.4 頻率:低通與高通濾波器

5.5 高斯濾波器

5.6 邊緣銳利化

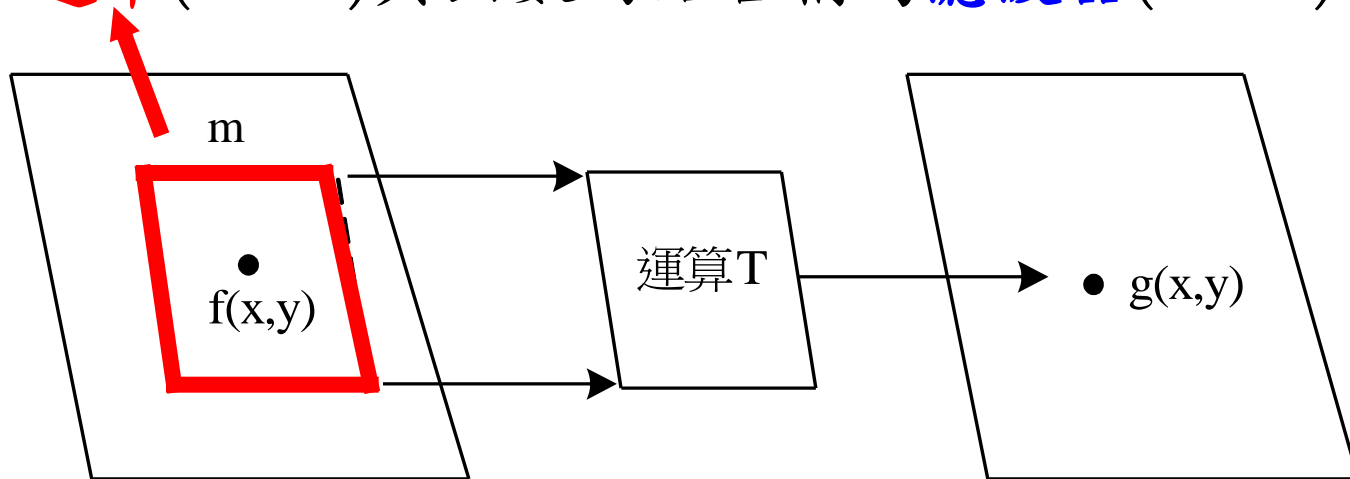
5.7 非線性濾波器

5.8 感興趣的區域處理



## 5.1 前言

- 點處理：其處理結果只受此點的灰度值影響和此點所在的位置無關。
- 鄰域處理：處理的結果會受此點的鄰近點影響
- 遮罩(Mask)與函數的結合稱為濾波器(Filter)。



## 遮罩值 (3×5)

$m(-1, -2)$	$m(-1, -1)$	$m(-1, 0)$	$m(-1, 1)$	$m(-1, 2)$
$m(0, -2)$	$m(0, -1)$	$m(0, 0)$	$m(0, 1)$	$m(0, 2)$
$m(1, -2)$	$m(1, -1)$	$m(1, 0)$	$m(1, 1)$	$m(1, 2)$

## 相對應的像素值

$$\sum_{s=-1}^1 \sum_{t=-2}^2 m(s, t) p(i + s, j + t)$$

$p(i - 1, j - 2)$	$p(i - 1, j - 1)$	$p(i - 1, j)$	$p(i - 1, j + 1)$	$p(i - 1, j + 2)$
$p(i, j - 2)$	$p(i, j - 1)$	$p(i, j)$	$p(i, j + 1)$	$p(i, j + 2)$
$p(i + 1, j - 2)$	$p(i + 1, j - 1)$	$p(i + 1, j)$	$p(i + 1, j + 1)$	$p(i + 1, j + 2)$

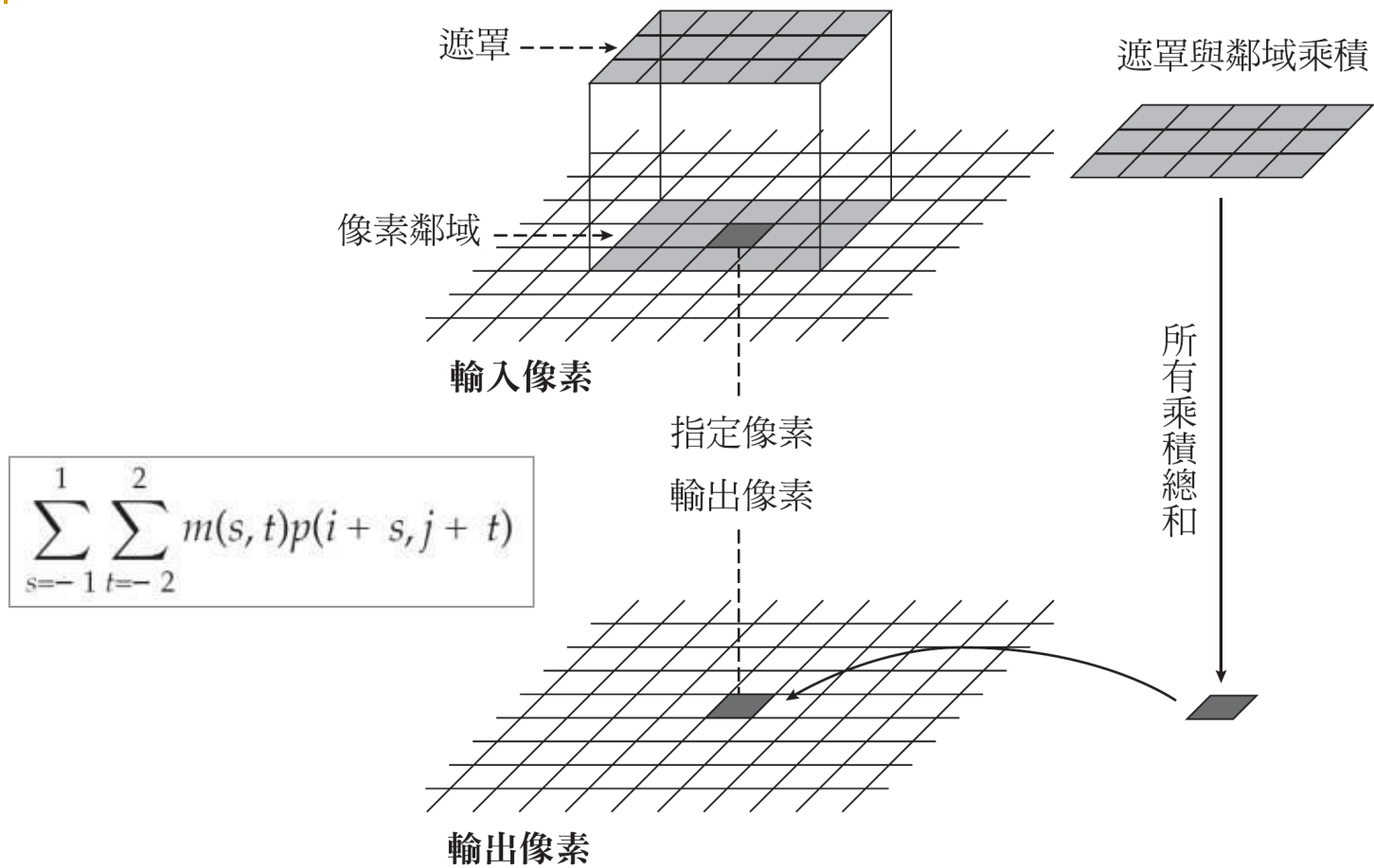


圖 5.2 執行空間濾波

## ■ 範例：3×3的平均濾波器

	a	b	c
	d	e	f
	g	h	i

$$\rightarrow \frac{1}{9}(a + b + c + d + e + f + g + h + i)$$

```
>> x=uint8(10*magic(5))
```

```
x =
```

170	240	10	80	150
230	50	70	140	160
40	60	130	200	220
100	120	190	210	30
110	180	250	20	90

```
>> mean2(x(1:3,2:4))
```

```
ans =
```

```
108.8889
```

這矩陣是影像x套用3×3 平均濾波器的結果

111.1111	108.8889	128.8889
110.0000	130.0000	150.0000
131.1111	151.1111	148.8889

## 5.2 運算符號

	a	b	c
	d	e	f
	g	h	i

- 平均濾波器可用下列矩陣表示：

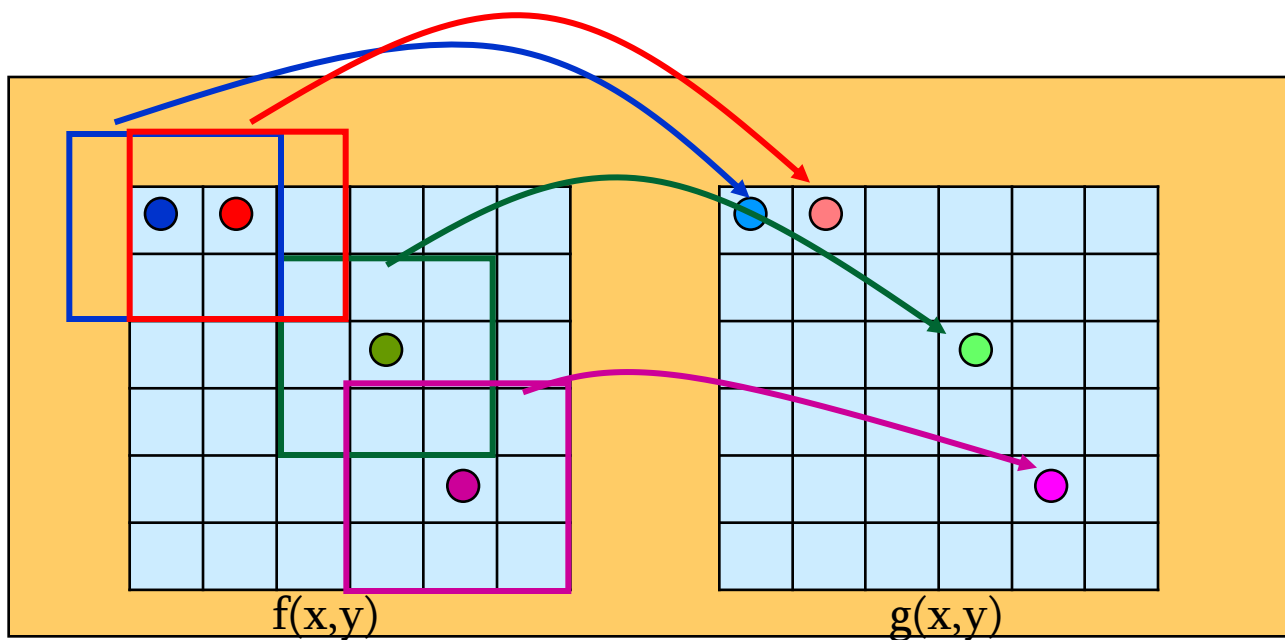
$$\begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix} = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \rightarrow \frac{1}{9}(a + b + c + d + e + f + g + h + i)$$

- 高通濾波器

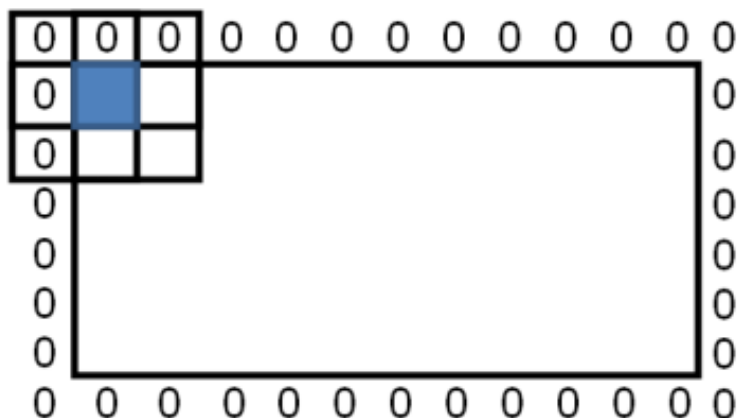
$$\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \rightarrow a - 2b + c - 2d + 4e - 2f + g - 2h + i$$

## □ 遮罩運算

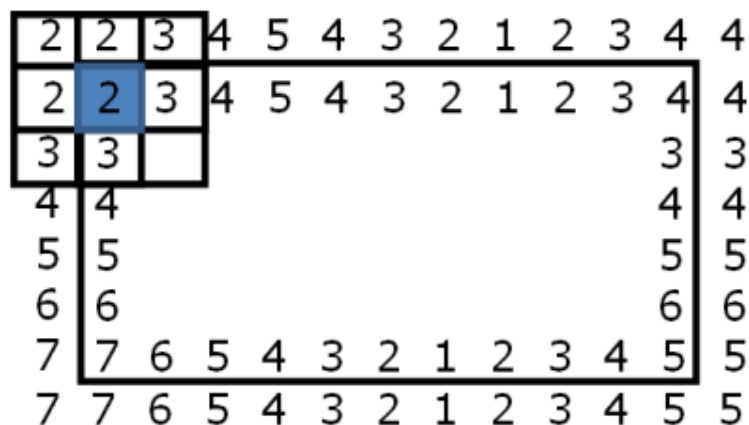
- 鄰近區即為遮罩大小，並以由左往右、由上往下方式滑動
- 每個輸出像素均以相同方式運算
- 超出影像範圍之鄰近區：
  1. 補0
  2. 重複邊緣像素值 (鏡射)
  3. 不考慮會超出的部分 (忽略邊緣)



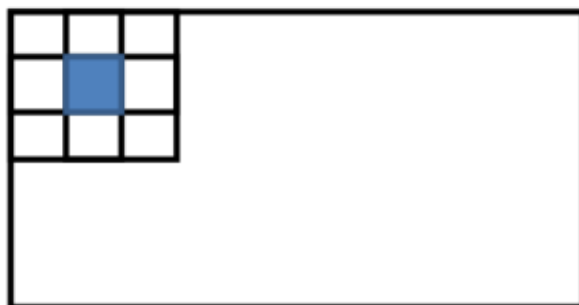




1. 補0



2. 重複邊緣像素值 (鏡射)



3. 不考慮會超出的部分 (忽略邊緣)

## 5.3 Matlab 中的濾波函數

- 使用 filter2 函數執行線性濾波運算：

```
filter2(filter, image, shape)
```

結果會產生一個 double 資料形態的矩陣

```
>> a=ones(3,3)/9  
>> filter2(a,x,'same')
```

- 參數 shape 非必要，用來指定影像邊緣的處理方式
  - 'same' - 補零
  - 'valid' - 忽略邊緣

## ■ 範例：3×3的平均濾波器

```
>> x=uint8(10*magic(5))  
>> a=ones(3,3)/9  
>> filter2(a,x,'valid')  
ans =  
    111.1111    108.8889    128.8889  
    110.0000    130.0000    150.0000  
    131.1111    151.1111    148.8889
```

➔ 濾波器可自己設定，也可使用 **fspecial** 函數產生

```
>> c=imread('cameraman.tif');  
>> f1=fspecial('average', [3,3]); >>f1=fspecial('average', 3);  
>> cf1=filter2(f1,c);  
>> figure, imshow(c), figure, imshow(cf1/255)
```



(a)



(b)



(c)



(d)



(e)

結論：

- 平均濾波器會模糊影像
- 邊緣補 0 會在影像四周產生深色邊線，遮罩大越明顯

**圖 5.4** 平均濾波 (a) 原始影像 (b) 平均濾波 (c)  $9 \times 9$  濾波  
(d)  $25 \times 25$  濾波 (e) 使用鏡射的  $25 \times 25$  濾波

## 5.4 頻率：低通與高通濾波器

- 影像的頻率(frequency)：是灰階值隨著距離而產生變化程度的一種度量。
  - High frequency: 如邊緣、雜訊等
  - Low frequency: 如背景、皮膚紋理等

係數總和為零，  
針對邊緣偵測、  
邊緣增強

$$\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

高通濾波器  
(high-pass filter)

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

低通濾波器  
(low-pass filter)

如：平均濾波器

```
>> c=imread('cameraman.tif');
>> f=fspecial('laplacian')
f =
    0.1667    0.6667    0.1667
    0.6667   -3.3333    0.6667
    0.1667    0.6667    0.1667
>> cf=filter2(f,c);
>> imshow(cf/100)
```

### 高通濾波器 (high-pass filter)

```
>> f1=fspecial('log')
f1 =
    0.0448    0.0468    0.0564    0.0468    0.0448
    0.0468    0.3167    0.7146    0.3167    0.0468
    0.0564    0.7146   -4.9048    0.7146    0.0564
    0.0468    0.3167    0.7146    0.3167    0.0468
    0.0448    0.0468    0.0564    0.0468    0.0448
>> cf1=filter2(f1,c);
>> figure,imshow(cf1/100)
```



(a) Laplacian 濾波



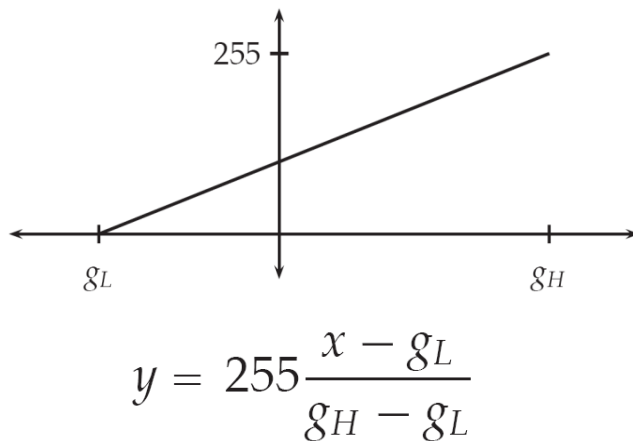
(b) 高斯的 Laplacian (LoG) 濾波

## ■ Why divide by 100?

```
>> imshow(cf/100)
```

### □ 因為處理數值超出0至255的範圍

- 將負值變為正值 (make negative values positive)
- 數值裁剪 (clip values)
- 比例轉換 (scaling transformation)

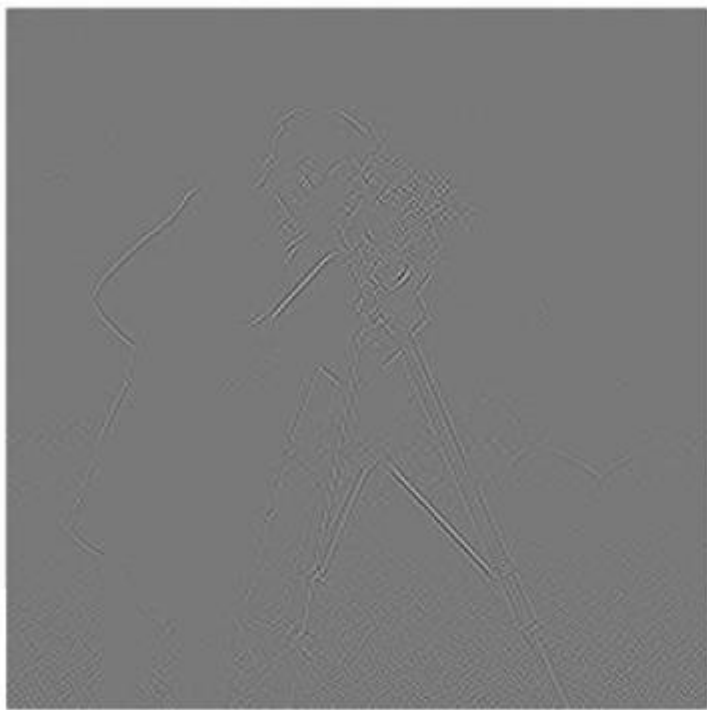


```
>> f2=[1 -2 1;-2 4 -2;1 -2 1];  
>> cf2=filter2(f2,c);  
>> figure,imshow(mat2gray(cf2)); 做法1
```

```
>> maxcf2=max(cf2(:));  
>> mincf2=min(cf2(:)); 做法2  
>> cf2g=(cf2-mincf2)/(maxcf2-mincf2);
```

```
>> figure,imshow(cf2/60) 做法3
```





(a) 使用 `mat2gray`



(b) 除以一个常数

**圖 5.6** 執行高通濾波並顯示結果

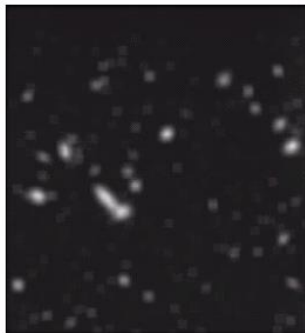


# 應用

## ■ 低通濾波器



Original image



Blurring image  
(15×15 mask)



Binary image

## ■ 高通濾波器



## 5.5 高斯濾波器(Gaussian filter)

- 高斯濾波器是個**低通**濾波器，由高斯機率分佈函數變化而來
  - $\sigma$ 是標準差，值越大，曲線越**平緩**；反之，越**陡峭**
  - 此濾波器具模糊效果

```
>> g=fspecial('gaussian', [5 5], 2);
```

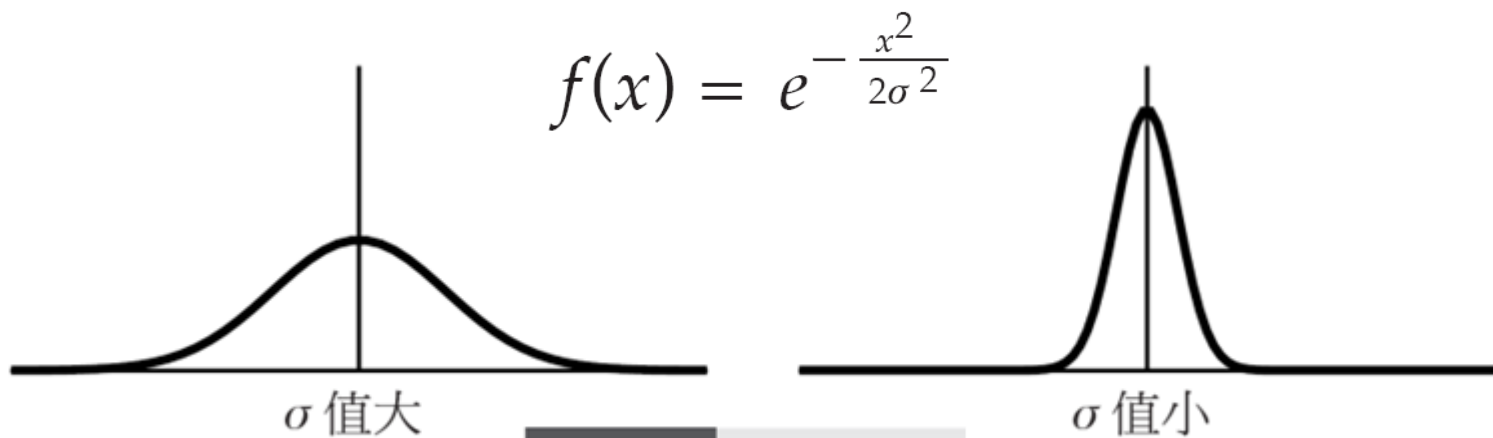


圖 5.7 一維高斯

```
>> a=50;s=3;  
>> g=fspecial('gaussian',[a a],s);  
>> surf(1:a,1:a,g)  
>> s=9;  
>> g2=fspecial('gaussian',[a a],s);  
>> figure,surf(1:a,1:a,g2)
```

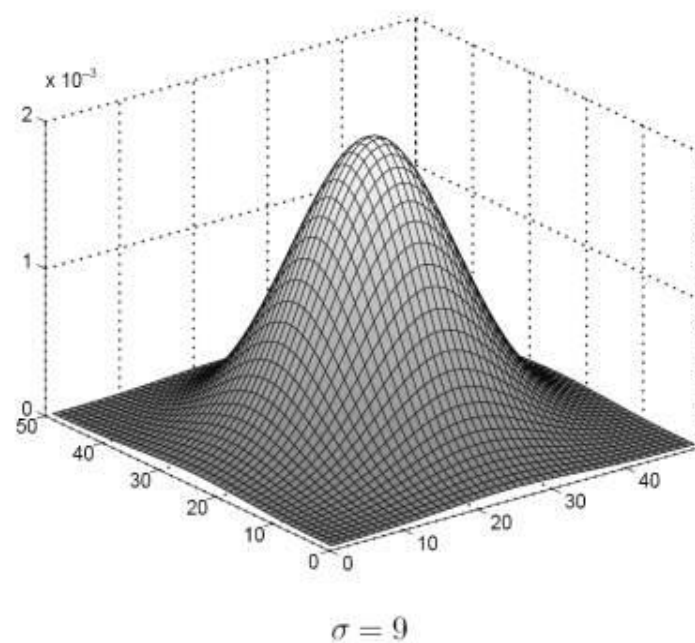
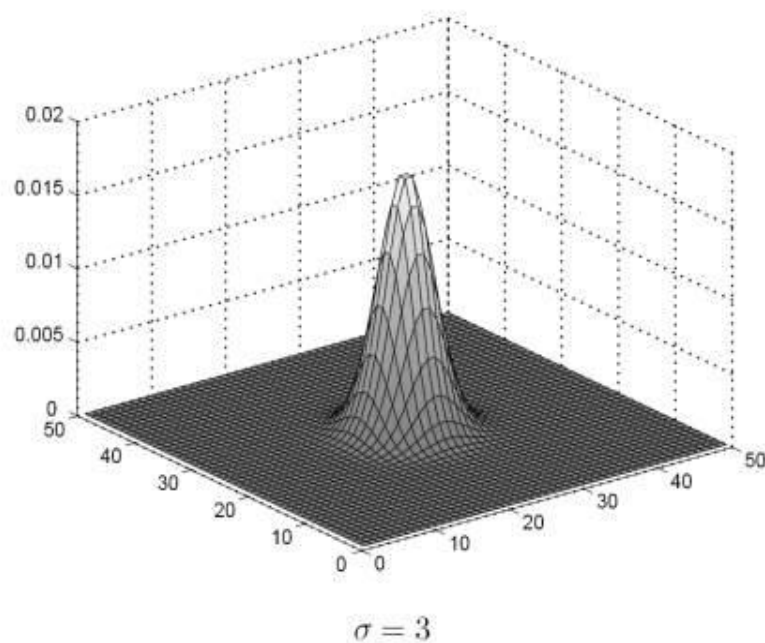


圖 5.8 二維高斯

## ■ 範例： $n \times n$ 的高斯濾波器( $\sigma=2$ or $5$ )

```
>> c=imread('cameraman.tif');  
>> g1=fspecial('gaussian',[5,5]);  
>> g2=fspecial('gaussian',[5,5],2);  
>> g3=fspecial('gaussian',[11,11],1);  
>> g4=fspecial('gaussian',[11,11],5);
```

預設  $\sigma=0.5$

```
>> imshow(filter2(g1,c)/256)  
>> figure,imshow(filter2(g2,c)/256)  
>> figure,imshow(filter2(g3,c)/256)  
>> figure,imshow(filter2(g4,c)/256)
```

標準差越大  
效果越模糊



$5 \times 5, \sigma = 0.5$



$5 \times 5, \sigma = 2$



$11 \times 11, \sigma = 1$



$11 \times 11, \sigma = 5$

## 5.6 邊緣銳利化

- 動作又稱edge enhancement、edge crispening、unsharp masking

```
>> f=fspecial('average');  
>> xf=filter2(f,x);  
>> xu=double(x)-xf/1.5  
>> imshow(xu/70)
```

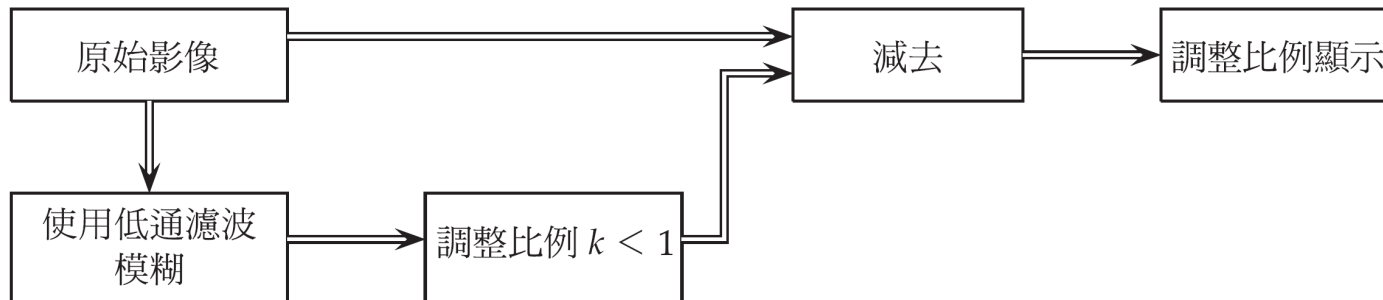


圖 5.10 去銳利化遮罩的流程

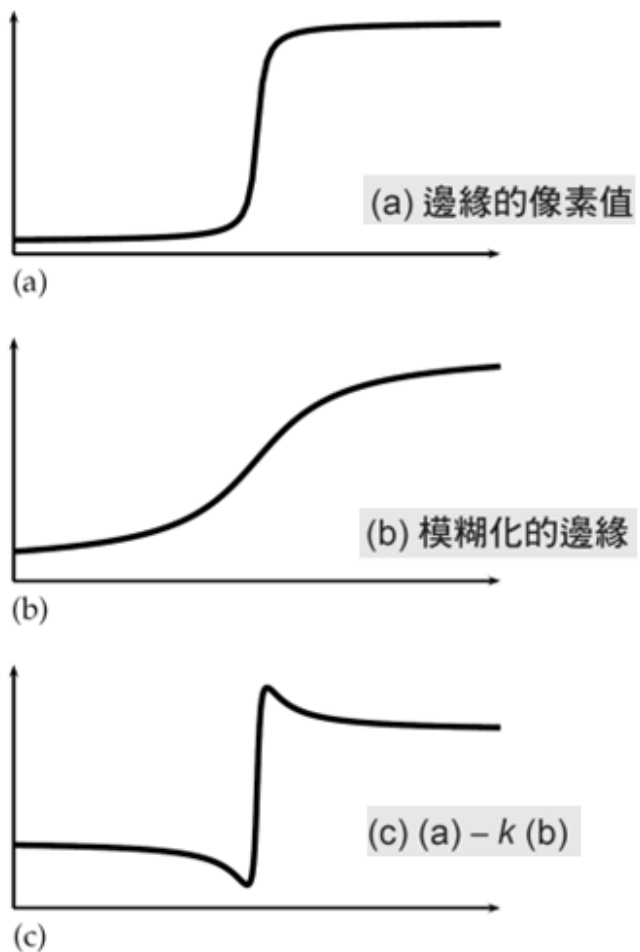


圖 5.12 去銳利化遮罩



(a) 原始影像



(b) 去銳利化遮罩後影像

## ■ 或，使用fspecial函數下unsharp去銳利化參數

```
>> p=imread('pelicans.tif');  
>> u=fspecial('unsharp',0.5);  $\alpha = 0.5$   
>> pu=filter2(u,p);  
>> imshow(p),figure,imshow(pu/255)
```



圖 5.13 (a) 原始影像



(b) 去銳利化遮罩後影像



## 5.7 非線性濾波器

- 非線性濾波器則是對遮罩內灰階值執行非線性函數

- **排序濾波器** (rank-order filters)

- 最大濾波器 (maximum filter)
- 最小濾波器 (minimum filter)
- 中位濾波器 (median filter)

```
>> c=imread('cameraman.tif');  
>> cmax = n1filter(c, [3 3], 'max(x(:))');  
>> cmin = n1filter(c, [3 3], 'min(x(:))');  
  
>> cMAX=ordfilt2(c, 9, ones(3,3));  
>> cMIN=ordfilt2(c, 1, ones(3,3));  
>> cMed=ordfilt2(c, 5, ones(3,3));
```

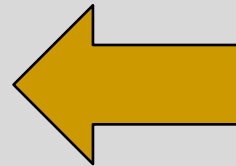






圖 5.15 (a) 執行最大濾波器



(b) 執行最小濾波器

## 5.8 感興趣區域 (ROI)

- 影像中的特定感興趣之區域(regions of interest)

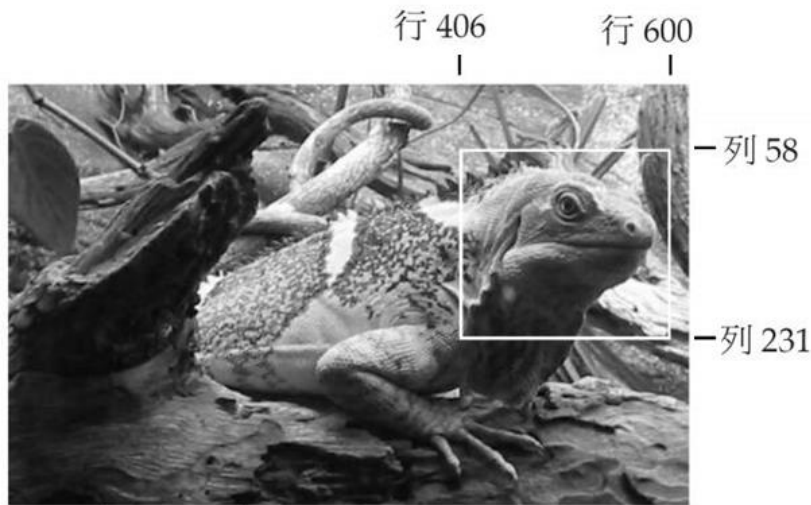


圖 5.16 影像及感興趣的區域



圖 5.17 圖 5.16 中定義之感興趣區域的相對應遮罩

```
>> ig=imread('iguana.tif');  
>> roi=roipoly(ig,[406 600 600 406],[58 58 231 231]);
```

## ■ 範例：針對特定區域作處理

### □ 平均濾波

```
>> a=fspecial('average',[15,15]);  
>> iga=roifilt2(a,ig,roi);  
>> imshow(iga)
```

### □ 去銳利化遮罩

```
>> u=fspecial('unsharp');  
>> igu=roifilt2(u,ig,roi);  
>> figure,imshow(igu)
```

### □ 高斯log濾波

```
>> l=fspecial('log');  
>> igl=roifilt2(l,ig,roi);  
>> figure,imshow(igl)
```



# 總結

## ■ 自動產生濾波器: Create 2-D special filters

□ `fspecial(type,parameters)`

ex.: `F1=fspecial('average', 11)`

FILTER TYPE:

Value	Description
'gaussian'	Gaussian lowpass filter
'sobel'	Sobel horizontal edge-emphasizing filter
'prewitt'	Prewitt horizontal edge-emphasizing filter
'laplacian'	Filter approximating the two-dimensional Laplacian operator
'log'	Laplacian of Gaussian filter
'average'	Averaging filter
'unsharp'	Unsharp contrast enhancement filter

# 練習

- 使用下列指令顯示猴子(mandrill.tiff)的灰階影像，執行下列所有濾波，請選出各個濾波器的效果。

☐ 高通濾波器

☐ 低通濾波器

☐ 偵測垂直方向

☐ 偵測水平方向

☐ 偵測135度

☐ 偵測45度

☐ 邊緣偵測

☐ 模糊濾波器 ←

☐ 全相偵測

$$a = \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

$$b = \begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix}$$

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

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

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

$$f = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

$$g = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

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