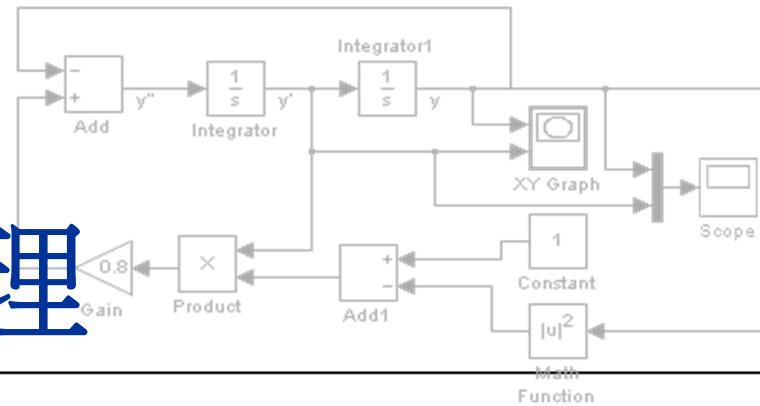


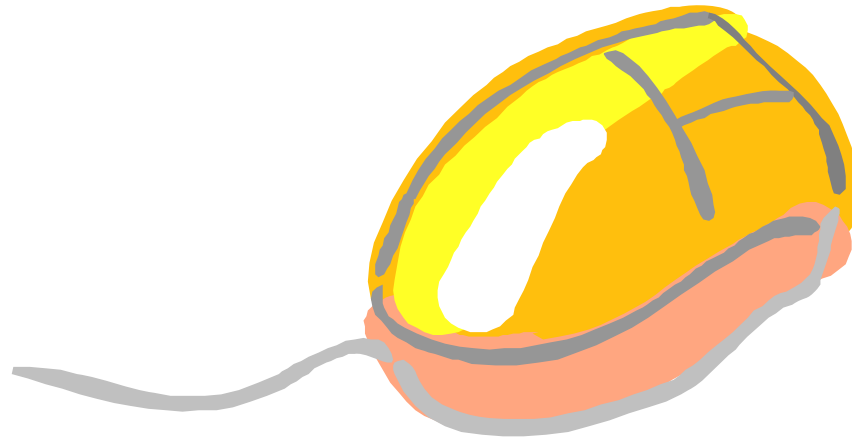
檔案的處理



檔案的觀念與操作的方式

文字檔與二進位檔案的使用方式

檔案指標的操作





簡單的檔案處理

- Matlab支援了四種常用的資料檔案格式：

表 16.1.1 檔案類型與相關資訊

檔案格式	說明	讀取函數	寫入函數
MAT	儲存 Matlab 工作區的變數所產生的檔案	load	save
CSV	Comma-separated value，即以逗號隔開的數據檔案	csvread	csvwrite
DLM	Delimited text，以特定的分隔符號隔開的數據檔案	dlmread	dlmwrite
TAB	Tab-separated text，以 Tab 鍵隔開的數據檔案	dlmread	dlmwrite




寫入與取出工作區內的變數

- Save 與 load 可以存取一般的變數、結構、多質陣列、影像等資料型態：

表 16.1.2 存取工作區內的變數之函數

函 數	說 明
<code>save('filename', 'var₁', 'var₂', ...)</code>	將工作區內的 <code>var₁</code> 、 <code>var₂</code> 、 <code>...</code> 等變數儲檔
<code>load('filename')</code>	從檔案 <code>filename</code> 讀取由 <code>save</code> 函數所存取的變數



```
>> prime_list=primes(1000);  
>> mag=magic(3);  
>> save('my_data','prime_list','mag')  
>> clear all  
>> m=load('my_data','mag')
```

```
m =  
    mag: [3x3 double]
```

```
>> m.mag
```

```
ans =  
     8     1     6  
     3     5     7  
     4     9     2
```



以逗號隔開的數據處理

- 如果數據資料是以逗號隔開，則稱之為**CSV**型態
- `csvread`與`csvwrite`函數，可存取**CSV**型態的資料

表 16.1.3 存取由逗號隔開的數據資料

函 數	說 明
<code>m=csvread('filename')</code>	讀取以逗號為分隔符號的數據資料
<code>csvwrite('filename',m)</code>	將數據資料以 <code>csv</code> 的格式寫到檔案裡



```
>> A=magic(3)
```

```
A =
```

```
     8     1     6
     3     5     7
     4     9     2
```

```
>> csvwrite('csvdata.csv',A)
```

```
>> type csvdata.csv
```

```
8,1,6
```

```
3,5,7
```

```
4,9,2
```

```
>> B=csvread('csvdata.csv')
```

```
B =
```

```
     8     1     6
     3     5     7
     4     9     2
```



以特定符號隔開的數據處理

- 如果數字不是以逗號來分隔，則可利用 `dlmread` 與 `dlmwrite` 函數來存取：

表 16.1.4 處理以特定符號隔開的數據

函 數	說 明
<code>m=dlmread('filename', 'dlm')</code>	讀取以 <code>dlm</code> 為分隔符號的數據資料
<code>dlmwrite('filename', m, 'dlm')</code>	以 <code>dlm</code> 為分隔符號來儲存數據資料 <code>m</code>



```
>> mat=[3 4 12;2 4 19]
```

```
mat =
```

```
     3     4    12
     2     4    19
```

```
>> dlmwrite('dlmfile.dlm',mat,'\t')
```

```
>> dlmread('dlmfile.dlm','\t')
```

```
ans =
```

```
     3     4    12
     2     4    19
```

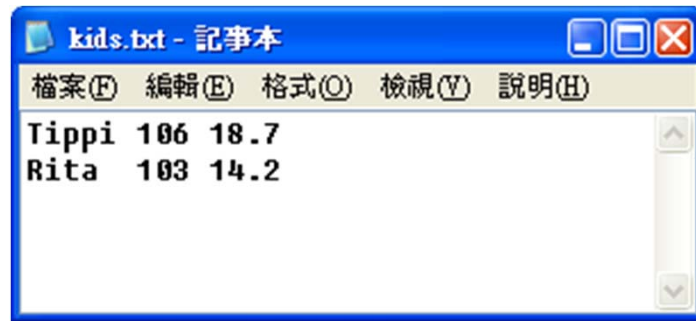



讀取摻雜文字與數據資料的檔案

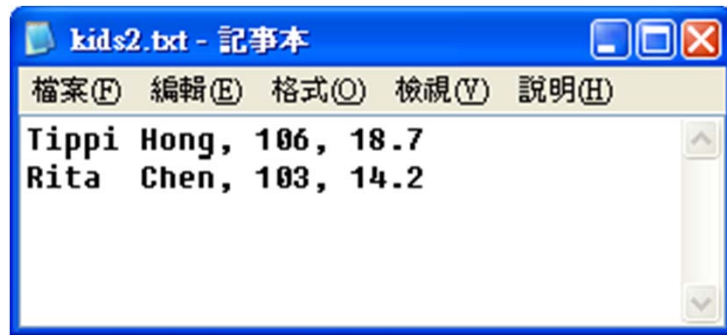
- 如果文字資料檔摻雜有文字與數字，可以利用 `textread` 函數讀取：

表 16.1.5 讀取摻雜文字與數據資料的檔案

函 數	說 明
<code>[a,b,...]=textread('fname','format')</code>	依 <i>format</i> 所記載的格式讀取資料
	<code>%n</code> - 可讀取整數或浮點數
	<code>%d</code> - 讀取含正負號的整數
	<code>%f</code> - 讀取浮點數的數據
	<code>%s</code> - 讀取由空白鍵隔開的字串



```
>> [name height weight]=textread('kids.txt','%s%d%f')
name =
    'Tippi'
    'Rita'
height =
    106
    103
weight =
    18.7000
    14.2000
```



```
>> [n h w]=textread('kids2.txt','%s%d%f','delimiter',' ','')
n =
    'Tippi Hong'
    'Rita Chen'
h =
    106
    103
w =
    18.7000
    14.2000
```



文字檔案的處理

讀取與關閉檔案

- Matlab分別以fopen與fclose二個函數來處理開檔與關檔的動作：

表 16.2.1 開檔與關檔的函數

函 數	說 明
<code>fid=fopen('filename', 'permission')</code>	讀取檔案的內容，並傳回檔案識別碼
<code>fclose(fid)</code>	關閉檔案識別碼為 <i>fid</i> 的檔案

表 16.2.2 檔案存取模式

存取模式	代碼	說 明
讀取資料	r	開啟檔案以供讀取
寫入資料	w	開啟檔案以供寫入
附加於檔案之後	a	開啟一個檔案
讀取與附加	a+	可讀取檔案



寫入與讀取文字檔

- 要把資料數據寫到檔案中，可用 `fprintf` 函數
- 要讀取資料數據，可用 `fscanf` 函數

表 16.2.3 檔案寫入與讀取函數

函 數	說 明
<code>fprintf(fid, 'str', e₁, e₂, ...)</code>	依序將運算式寫入檔案識別碼為 <i>fid</i> 的檔案中
<code>fscanf(fid, 'str')</code>	讀取檔案識別碼為 <i>fid</i> 之檔案裡的資料
<code>fscanf(fid, 'str', n)</code>	一次讀取 <i>n</i> 筆資料
<code>fscanf(fid, 'str', [m, n])</code>	一次讀取 <i>m</i> × <i>n</i> 筆資料

- 
-
- 下面是利用fprintf函數寫入資料的範例：

```
>> fw=fopen('test.txt','w')
```

```
fw =  
    3
```

```
>> fprintf(fw,'%d ',primes(20));
```

```
>> fclose(fw)
```

```
ans =  
    0
```

```
>> type test.txt  
2 3 5 7 11 13 17 19
```

- 
-
- 下面是利用 `scanf` 函數讀取資料的範例：

```
>> fr=fopen('test.txt','r')
```

```
fr =  
    3
```

```
>> fscanf(fr,'%d')
```

```
ans =  
    2  
    3  
    5  
    7  
   11  
   13  
   17  
   19
```

- 
-
- 要測試檔案是否已讀取到末端，可利用`feof`函數
 - 如要從頭讀取，可用`frewind`函數

表 16.2.4 檔案讀取函數

函 數	說 明
<code>frewind(fid)</code>	設定檔案可從頭讀取
<code>feof(fid)</code>	測試檔案是否已讀取到末端。若是，則回應 1，否則回應 0



一行一行讀取檔案

- 如果想一行一行的讀取檔案的內容，可利用 `fgetl` 或 `fgets` 函數：

表 16.2.5 檔案寫入函數

函 數	說 明
<code>fgetl(fid)</code>	從檔案讀取一行字串，但不會讀取換行字元
<code>fgets(fid)</code>	從檔案讀取一行字串，連同換行字元也一併讀入



```
>> fr=fopen('seasons.txt','r');
```

```
>> s1=fgetl(fr)
```

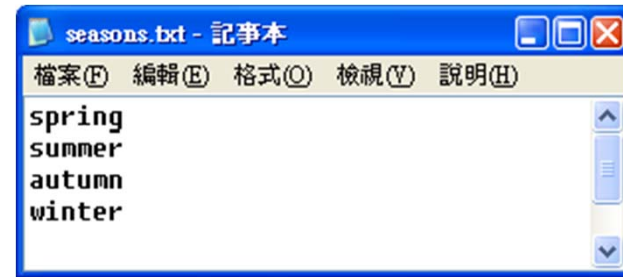
```
s1 =
```

```
Spring
```

```
>> s2=fgets(fr)
```

```
s2 =
```

```
Summer
```



```
>> double(s1)
```

```
ans =
```

```
115 112 114 105 110 103
```

```
>> double(s2)
```

```
ans =
```

```
115 117 109 109 101 114 13 10
```

```
>> fclose(fr);
```



二進位檔案的處理

- 把資料以二進位的格式寫到檔案裡，必須使用 `fwrite` 函數
- 從二進位檔讀取資料時，則是使用 `fread` 函數

表 16.3.1 二進位檔案寫入與讀取函數

函數	說明
<code>A=fread(fid, size, precision)</code>	讀取 <i>size</i> 個 <i>precision</i> 大小的二進位檔案
<code>fwrite(fid, A, precision)</code>	以指定的 <i>precision</i> 大小，將陣列 <i>A</i> 的內容寫入檔案 <i>fid</i> 中

表 16.3.2 可供使用的 precision 字串

precision	說明	所佔位元
'schar'	有號的字元	8 bits
'uchar'	無號的字元	8 bits
'int8'	8 bits 整數	8 bits
'int16'	16 bits 整數	16 bits
'int32'	32 bits 整數	32 bits
'int64'	64 bits 整數	64 bits
'uint8'	8 bits 無號整數	8 bits
'uint16'	16 bits 無號整數	16 bits
'uint32'	32 bits 無號整數	32 bits
'uint64'	64 bits 無號整數	64 bits
'float32'	32 bits 的浮點數	32 bits
'float64'	64 bits 的浮點數	64 bits
'double'	倍精度浮點數	64 bits

讀取與寫入二進位檔的練習

```
>> fid=fopen('data.bin','wb');  
>> cnt=fwrite(fid,magic(5),'int16')
```

```
cnt =  
    25
```

```
>> fclose(fid)
```

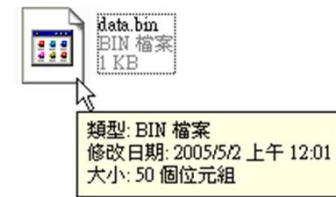
```
ans =  
     0
```

```
>> fr=fopen('data.bin','rb');
```

```
>> dat=fread(fr,3,'int16')
```

```
dat =  
    17  
    23  
     4
```

```
>> fclose(fr);
```





控制檔案的指標位置

- 檔案指標（file indicator）可記錄哪一筆資料要讀取
- 指標指到檔案的最前面，代表檔案尚未被讀取
- 指標指到檔案末端，代表檔案已全數被讀取完畢。

表 16.3.3 更改指標所指向位置的函數

函 數	說 明
<code>frewind(fid)</code>	將指標移到檔案的最開頭
<code>fseek(fid, offset, origin)</code>	設定檔案指標所在的位置
<code>ftell(fid)</code>	取得檔案指標的值

○ origin可以是下列字串之一：

cof : 指標目前的位置 (current position of the opened file)

bof : 檔案起始的位置 (beginning of the file)

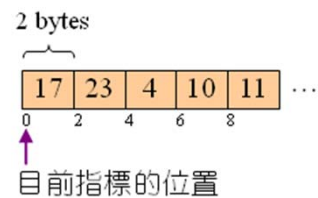
eof : 檔案的結束位置 (end of the file)

下面的例子說明了如何更改指標所指向的位置：

```
>> fr=fopen('data.bin','rb');
```

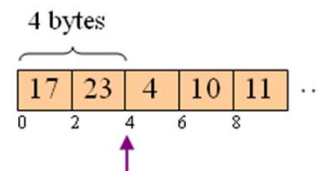
```
>> ftell(fr)
```

```
ans =  
    0
```



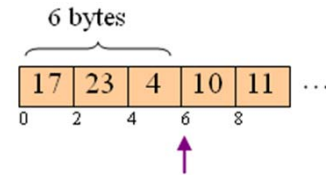
```
>> fseek(fr,4,'bof')
```

```
ans =  
    0
```



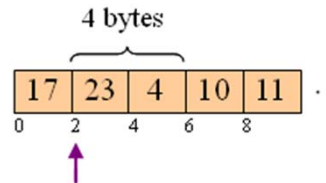


```
>> fread(fr, 1, 'int16')
ans =
     4
```



```
>> ftell(fr)
ans =
     6
```

```
>> fseek(fr, -4, 'cof')
ans =
     0
```

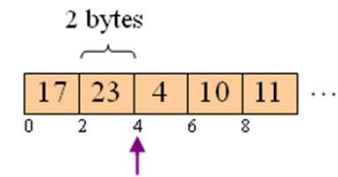


```
>> ftell(fr)
ans =
     2
```

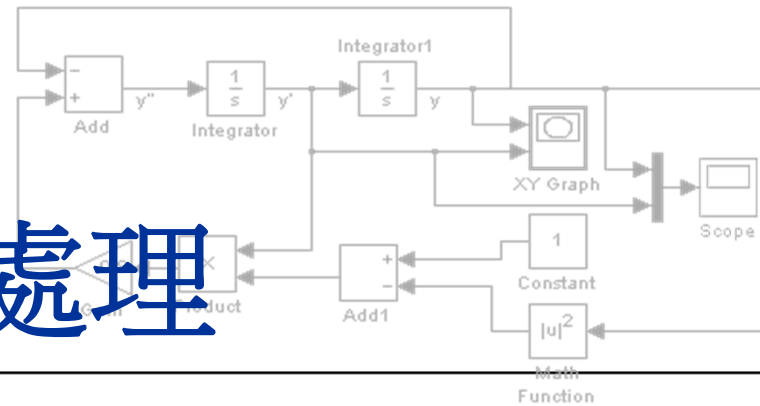


```
>> fread(fr,1,'int16')
ans =
    23
```

```
>> fclose(fr);
```



數位影像處理

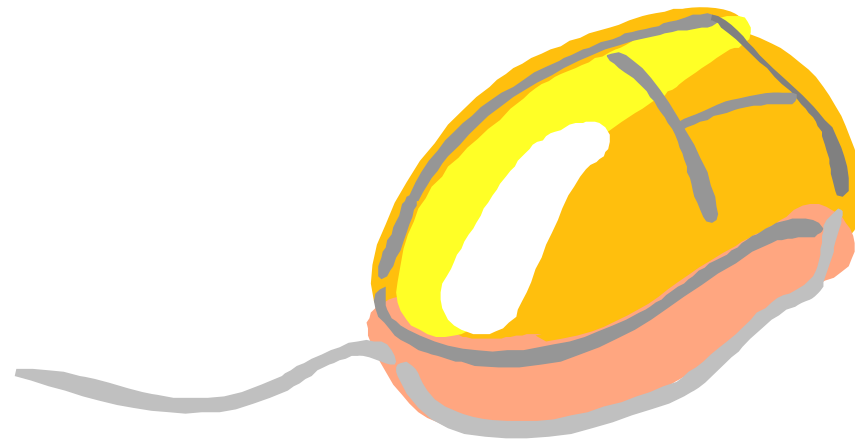


影像的分類

影像檔案的儲存

空間域與頻率域的影像處理

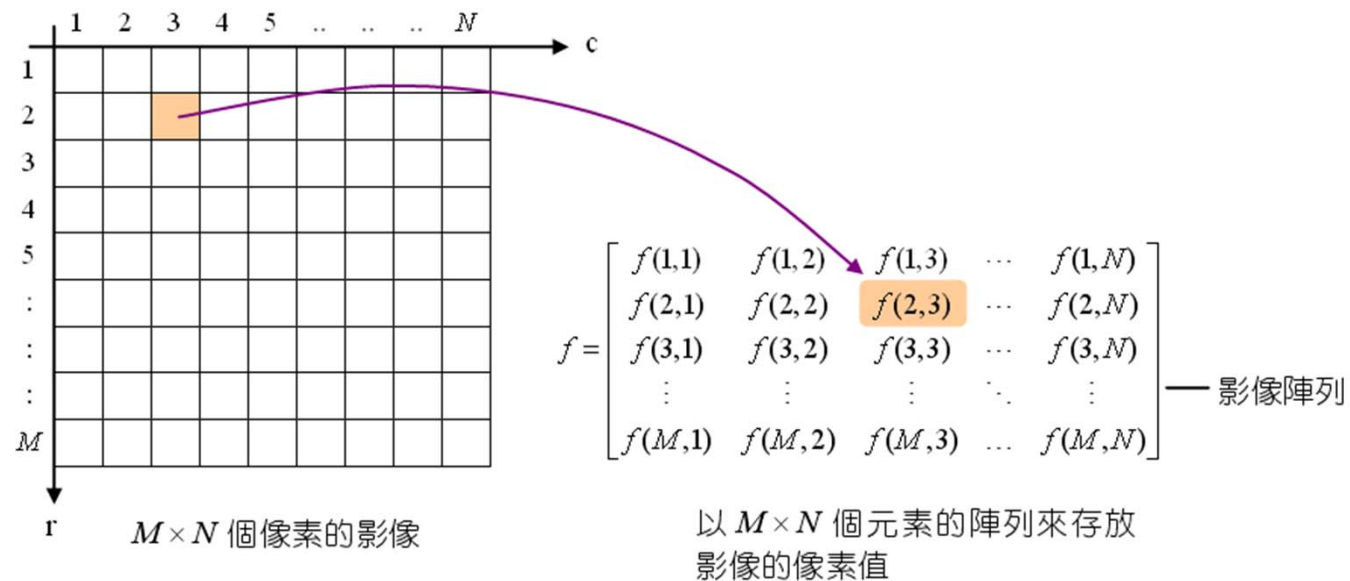
簡單濾波器的設計



影像的基本認識

座標系統

- 陣列與影像之間的關係可以由下圖來表示：





影像的讀取與顯示

- 下表列出了imread與imfinfo函數的用法：

表 20.1.1 影像讀取函數

函數	說明
<code>x=imread(filename)</code>	讀取影像檔案 <i>filename</i>
<code>[xi,map]=imread(filename)</code>	讀取影像檔案 <i>filename</i>
<code>s=imfinfo(filename)</code>	查詢影像檔案的資訊

- 
-
- 要顯示影像，可用`imshow`函數：

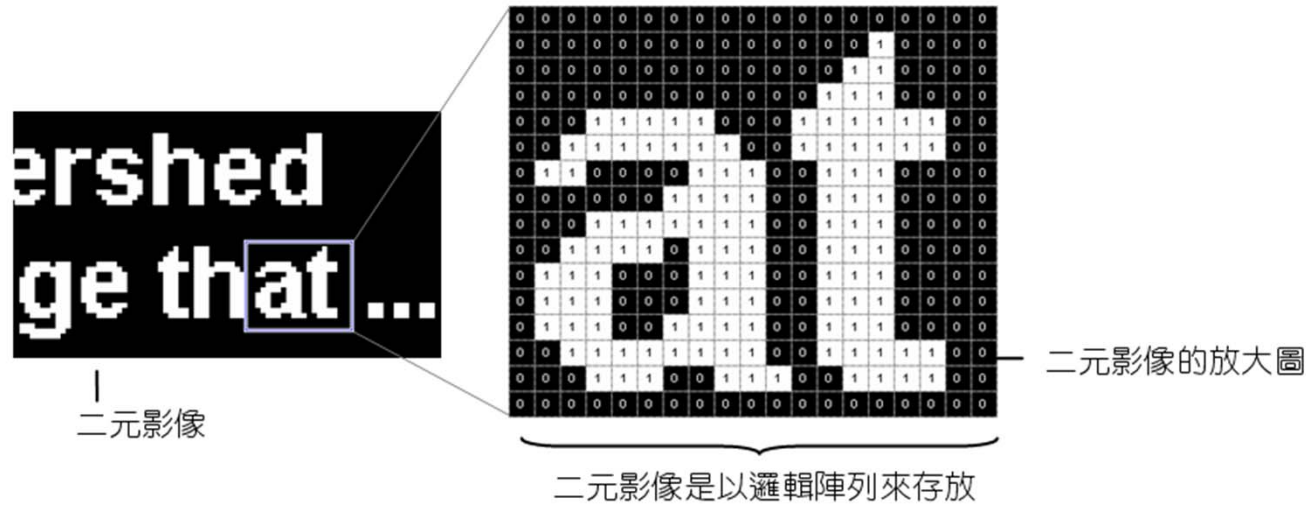
表 20.1.2 影像顯示函數


函數	說明
<code>imshow(x)</code>	顯示影像陣列 x 。
<code>imshow(x, [low high])</code>	顯示影像陣列 x ，並以 low 與 $high$ 來限制顯示的範圍。
<code>imshow(xi, map)</code>	以顏色對應表 map 顯示索引影像 xi
<code>imshow(filename)</code>	顯示影像檔案 $filename$ 所儲存的影像

影像的分類

二元影像

- 下圖是一張二元影像儲存方式的示意圖：





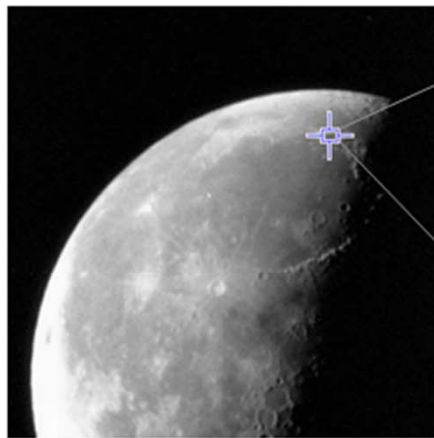
```
>> bw=imread('text.png');  
>> imshow(bw)
```

The term watershed
refers to a ridge that ...

... divides areas
drained by different
river systems.

灰階影像

- 下面是灰階影像的示意圖：



灰階影像

150	148	146	140	143	131	128	120	118	117
144	147	145	145	137	130	127	124	115	115
144	151	143	133	125	127	125	119	118	116
149	145	134	116	109	108	111	112	117	121
149	144	116	96	95	96	99	99	102	108
149	134	101	87	92	94	98	92	88	93
143	118	94	92	96	96	98	89	83	88

灰階影像的放大圖

灰階影像通常是以 `uint8` 或 `double` 型態的陣列來存放


```
>> x=imread('cameraman.tif');  
>> imshow(x); pixval on  
>> impixelinfo → show (x,y) BW/RGB value  
BW:0 or 1, Grey: 0~255, RGB:(210,156,200)  
>> indistline → map中兩點間距
```

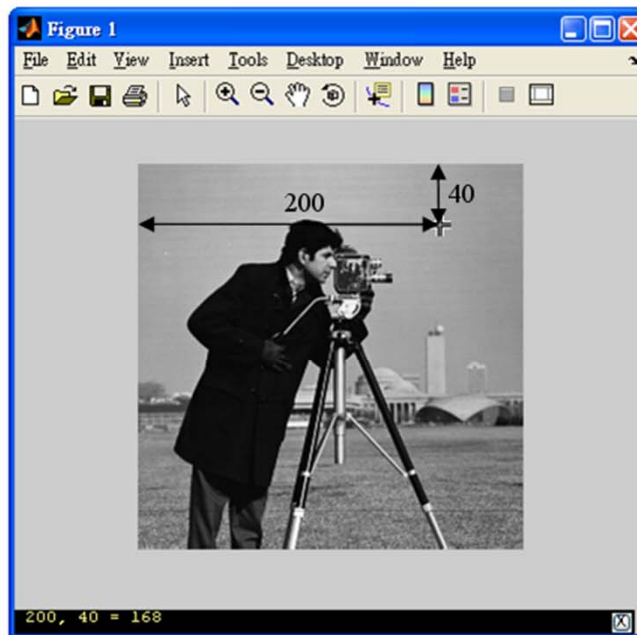


圖 20.2.3

顯示出滑鼠所在位置的座標與灰階值

此處會顯示出滑鼠所在位置的座標與灰階值

x 座標 | y 座標

按此處可關閉座標與灰階值的顯示

```
>> imshow(x,[0 128])
```

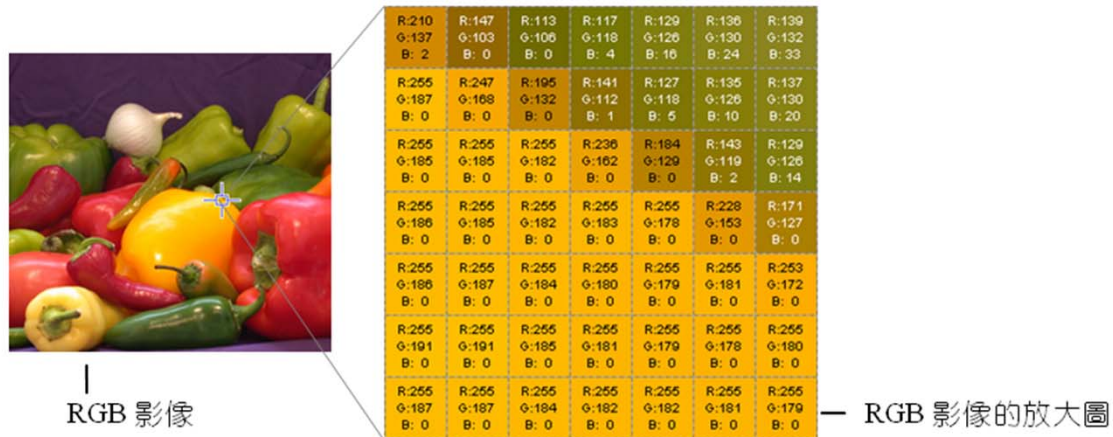


```
>> imshow(x>128)
```



RGB影像

○ 下圖是RGB影像的示意圖：



RGB 影像

RGB 影像的放大圖

210	147	113	117	129	138	139	137	103	106	118	126	130	132	2	0	0	4	16	24	33
255	247	195	141	127	135	137	187	168	132	112	118	126	130	0	0	0	1	5	10	20
255	255	255	236	184	143	129	185	185	182	162	129	119	126	0	0	0	0	0	2	14
255	255	255	255	255	220	171	186	185	182	183	178	153	127	0	0	0	0	0	0	0
255	255	255	255	255	255	253	186	187	184	180	179	181	172	0	0	0	0	0	0	0
255	255	255	255	255	255	255	191	191	185	181	179	178	180	0	0	0	0	0	0	0
255	255	255	255	255	255	255	187	187	184	182	182	181	179	0	0	0	0	0	0	0

紅色 (第一頁)

綠色 (第二頁)

藍色 (第三頁)

RGB 影像是由 $M \times N \times 3$ 的三維陣列來存放

```
>> rgb=imread('peppers.png');  
>> imshow(rgb)
```



索引影像

- 下圖是索引陣列與顏色對應表之間的關係：



<10>	<10>	<10>	<10>	<1>	<10>
R:0.19	R:0.19	R:0.19	R:0.19	R:0.06	R:0.19
G:0.22	G:0.22	G:0.22	G:0.22	G:0.06	G:0.22
B:0.22	B:0.22	B:0.22	B:0.22	B:0.03	B:0.22
<7>	<10>	<1>	<10>	<7>	<1>
R:0.26	R:0.19	R:0.06	R:0.19	R:0.26	R:0.06
G:0.16	G:0.22	G:0.06	G:0.22	G:0.16	G:0.06
B:0.06	B:0.22	B:0.03	B:0.22	B:0.06	B:0.03
<8>	<1>	<14>	<1>	<10>	<1>
R:0.09	R:0.06	R:0.22	R:0.06	R:0.19	R:0.06
G:0.26	G:0.06	G:0.26	G:0.06	G:0.22	G:0.06
B:0.03	B:0.03	B:0.26	B:0.03	B:0.22	B:0.03
<10>	<10>	<10>	<1>	<10>	<1>
R:0.19	R:0.19	R:0.19	R:0.06	R:0.19	R:0.06
G:0.22	G:0.22	G:0.22	G:0.06	G:0.22	G:0.06
B:0.22	B:0.22	B:0.22	B:0.03	B:0.22	B:0.03
<1>	<1>	<10>	<1>	<1>	<10>
R:0.06	R:0.06	R:0.19	R:0.06	R:0.06	R:0.19
G:0.06	G:0.06	G:0.22	G:0.06	G:0.06	G:0.22
B:0.03	B:0.03	B:0.22	B:0.03	B:0.03	B:0.22

索引值為 1
顏色對應表裡的第 2 列

索引陣列
(型態為 uint8)

10	10	10	10	1	10
7	10	1	10	7	1
8	1	14	1	10	1
10	10	10	1	10	1
1	1	10	1	1	10

顏色對應表

r	g	b
0.00	0.00	0.00
0.06	0.06	0.03
0.29	0.03	0.00
0.29	0.04	0.02
0.29	0.06	0.06
0.39	0.03	0.09
0.45	0.06	0.00
0.26	0.16	0.08
0.09	0.26	0.03
0.22	0.19	0.22
0.19	0.22	0.22
⋮	⋮	⋮

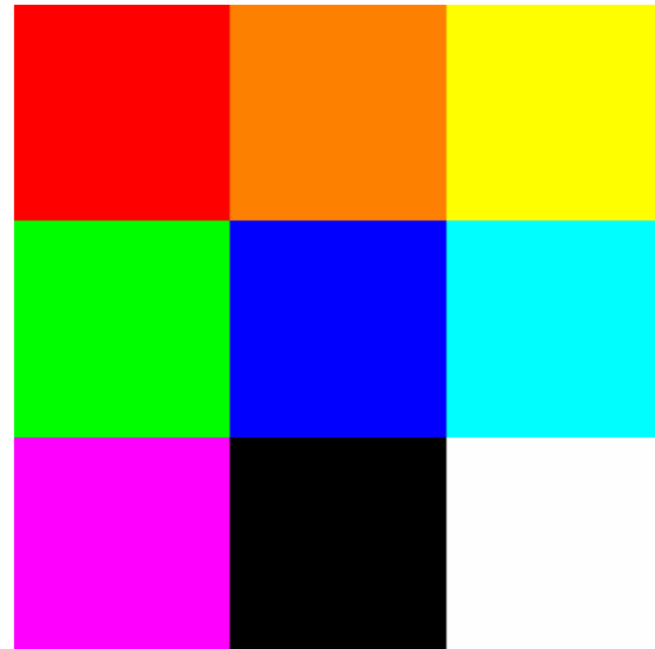
因為索引陣列的型態為 uint8，所以索引值為 1 的元素，所使用的顏色是顏色對應表裡第 2 列所指定的顏色



```
>> map=[
    1  0  0;
    1 0.5 0;
    1  1  0;
    0  1  0;
    0  0  1;
    0  1  1;
    1  0  1;
    0  0  0;
    1  1  1];
```

```
>> A=[1 2 3;
     4 5 6;
     7 8 9];
```

```
>> imshow(A,map,'InitialMagnification','fit')
```



-
- 在讀取索引影像時，必須利用兩個引數來接收 `imread` 的傳回值：

```
>> [x,map2]=imread('kids.tif');  
>> imshow(x,map2)
```



各種影像類型的整理

- 下表列出了影像類別與儲存格式的比較：

表 20.2.1 影像類別與儲存格式的比較

影像類別	儲存格式	說明
二元影像	logical	由 0 與 1 所組成的陣列，其中 0 與 1 為 logical 型態
灰階影像	double	由浮點數所組成的陣列，範圍從 0~1
	uint8	由 8-bit 整數所組成的陣列，範圍從 0~255
	uint16	由 16-bit 整數所組成的陣列，範圍從 0~65535
RGB 影像	double	由浮點數所組成的 $M \times N \times 3$ 陣列，範圍從 0~1
	uint8	由 8-bit 整數所組成的 $M \times N \times 3$ 陣列，範圍從 0~255
	uint16	由 16-bit 整數所組成的 $M \times N \times 3$ 陣列，範圍從 0~65535
索引影像 (設顏色對應表為 $k \times 3$ 的陣列)	double	由浮點數所組成的陣列，元素值的範圍從 1~ k
	uint8	由 8-bit 整數所組成的陣列，元素值的範圍從 0~ $k-1$
	uint16	由 16-bit 整數所組成的陣列，元素值的範圍從 0~ $k-1$


20.3 儲存與顯示影像

20.3.1 儲存影像資料

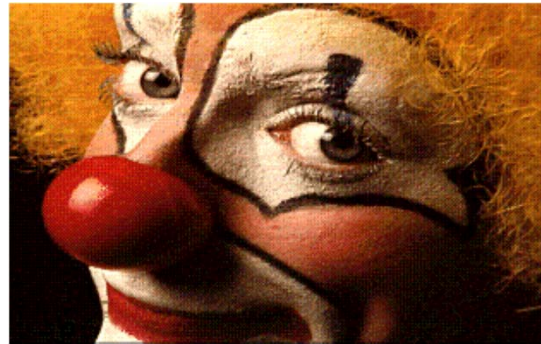
- 如果要將影像資訊寫入檔案中，可用`imwrite`函數：

表 20.3.1 將影像寫入檔案的函數

函 數	說 明
<code>imwrite(x, filename)</code>	將影像陣列 x 寫入檔案 $filename$ 中
<code>imwrite(x, map, filename)</code>	將索引影像的資料寫入檔案 $filename$
<code>imwrite(x, filename, p1, v1, p2, v2, ...)</code>	將影像資料寫入檔案 $filename$ 中



```
>> load clown
>> imshow(X,map)
```



```
>> imwrite(X,map,'cln2.jpg','Quality',25)
>> imshow('cln2.jpg')
```





20.3.2 利用imtool函數來顯示與分析影像

- imtool函數的語法如下：

表 20.3.3 imtool 函數的用法

函 數	說 明
<code>imtool(filename)</code>	利用影像分析工具開啟圖檔 <i>filename</i>
<code>imtool(x)</code>	利用影像分析工具開啟影像陣列 <i>x</i>
<code>imtool(x, map)</code>	利用影像分析工具開啟索引陣列 <i>x</i> 與顏色對應表 <i>map</i>

>> `imtool('pears.png')`

按此鈕可放大
要顯示的範圍

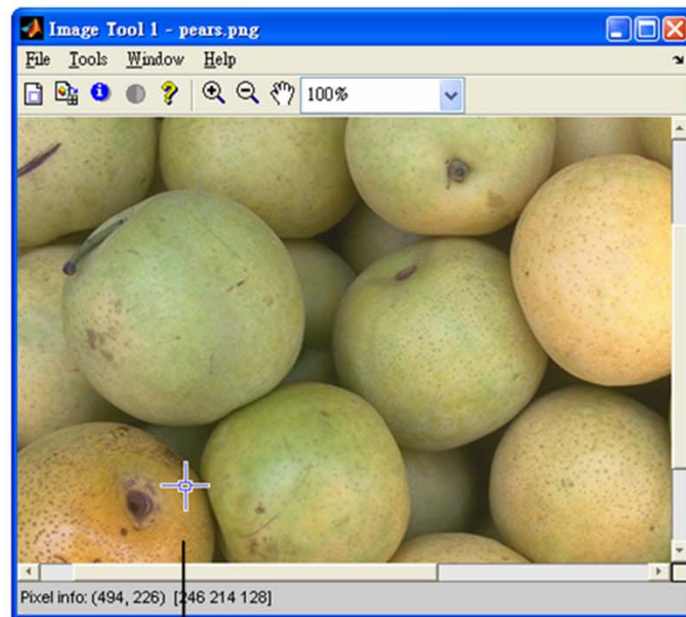
Overview 視窗

Image Tool 視窗



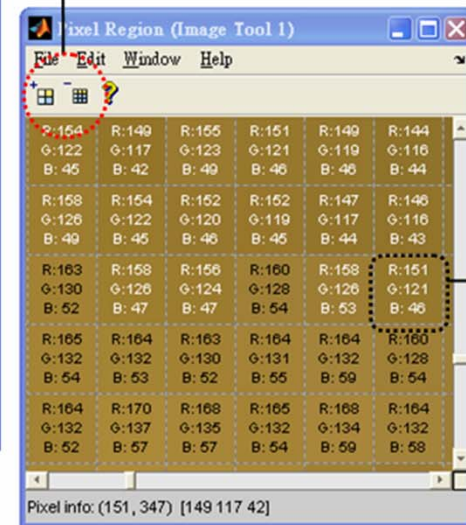
移動此框，右邊的視
窗會顯示出此框內的
影像

- 按下Pixel region按鈕可顯示像素的放大圖：



移動此十字框線，右邊的視窗會顯示此十字框線內的影像，以及每一個像素值

點選此處可縮放每一個像素的大小



每個方格代表一個像素




基本的影像處理

影像的資料型態與轉換

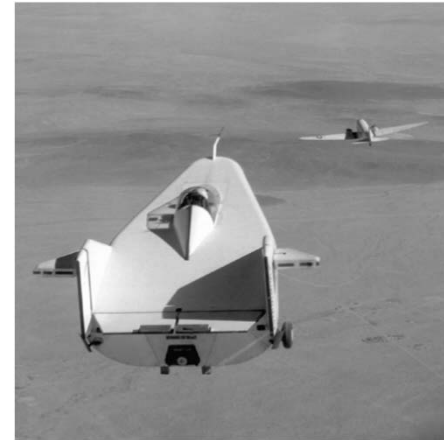
○ Matlab的影像轉換函數：

表 20.4.1 影像轉換函數

函 數	說 明
<code>y=ind2gray(x, map)</code>	將索引影像轉換成灰階影像
<code>[y, map]=gray2ind(x)</code>	將灰階影像轉換成索引影像
<code>y=rgb2gray(x)</code>	將 RGB 影像轉換成灰階影像
<code>[y, map]=rgb2ind(x, n)</code>	將 RGB 影像轉換成具有 n 個顏色的索引影像
<code>y=ind2rgb(x, map)</code>	將索引影像轉換成 RGB 影像



```
>> x=imread('liftingbody.png');  
>> imshow(x)  
>> [y,map]=gray2ind(x);
```



```
>> x=imread('pears.png');  
>> imshow(x)  
>> [y,map]=rgb2ind(x,16);
```





影像的數學運算

- 下表列出了Matlab提供的影像數學運算函數：

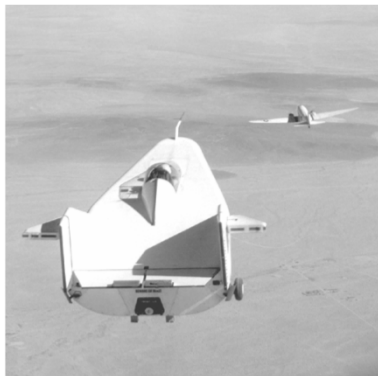
表 20.4.2 影像數學運算函數

函 數	說 明
<code>imabsdiff(x,y)</code>	計算兩張影像 x 與 y 之差的絕對值
<code>imcomplement(x)</code>	計算影像 x 的補數
<code>imadd(x,y)</code>	計算兩張影像 x 與 y 之和
<code>imsubtract(x,y)</code>	計算兩張影像 x 與 y 之差
<code>immultiply(x,y)</code>	計算兩張影像 x 與 y 之積
<code>imdivide(x,y)</code>	計算兩張影像 x 與 y 之商

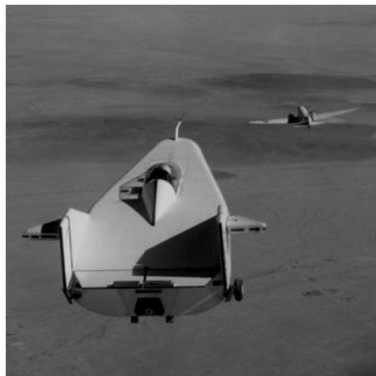

```
>> x=imread('liftingbody.png');  
>> imshow(x)
```



`imadd(x,50)`



`imsubtract(x,50)`



`imcomplement(x)`





影像的縮放、旋轉與剪裁

- 影像處理工具箱也可以進行影像的縮放、旋轉與剪裁：

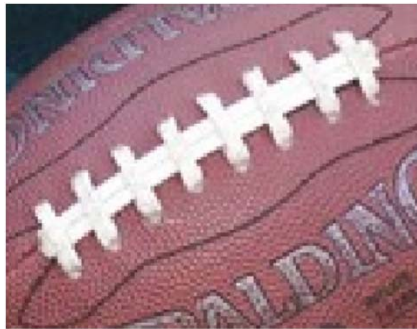
表 20.4.3 影像的縮放、旋轉與剪裁

函 數	說 明
<code>y=imresize(x,m)</code>	將影像 x 放大 m 倍
<code>y=imresize(x,[row col])</code>	將影像 x 縮放到 row 列、 col 行的大小
<code>y=imrotate(x,angle)</code>	將影像 x 逆時針旋轉 $angle$ 個角度
<code>y=imcrop(x)</code>	將影像 x 進行剪裁

```
>> x=imread('football.jpg');  
>> imshow(x)
```



```
>> imresize(x,2.5)
```



```
>> imresize(x,2.5,'bicubic')
```





空間域的影像處理

影像直方圖等化法

- 繪製影像直方圖與直方圖等化之相關函數的用法：

表 20.5.1 影像直方圖與等化函數

函 數	說 明
<code>imhist(x)</code>	繪出影像陣列 x 裡，像素值分佈的直方圖
<code>histeq(x)</code>	將影像陣列 x 等化

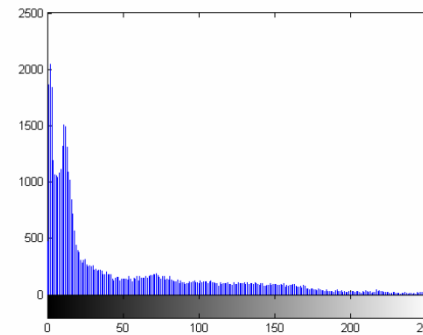


```
>> x=imread('tire.tif');
```

```
>> imshow(x)
```



```
>> imhist(x),axis([0,255,0,2500])
```

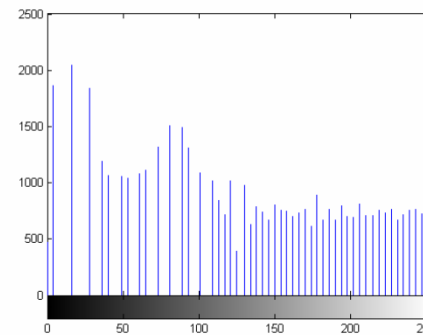


```
>> x2=histeq(x);
```

```
>> imshow(x2)
```



```
>> imhist(x2),axis([0,255,0,2500])
```



線性濾波

- 利用`imfilter`與`fspecial`函數可指定濾波器：

表 20.5.2 線性濾波函數

函 數	說 明
<code>imfilter(x,h)</code>	對影像陣列 x 以濾波器 h 進行線性濾波
<code>h=fspecial('type',p)</code>	以參數 p 建立濾波器 $type$ ，若 p 省略，則以預設值代替

```
>> x=imread('peppers.png');  
>> imshow(x)
```



```
>> h=ones(5,5)/25
```

```
h =
```

```
0.0400    0.0400    0.0400    0.0400    0.0400  
0.0400    0.0400    0.0400    0.0400    0.0400  
0.0400    0.0400    0.0400    0.0400    0.0400  
0.0400    0.0400    0.0400    0.0400    0.0400  
0.0400    0.0400    0.0400    0.0400    0.0400
```

```
>> imshow(imfilter(x,h))
```



```
>> h2=fspecial('unsharp')
h2 =
  -0.1667   -0.6667   -0.1667
  -0.6667    4.3333   -0.6667
  -0.1667   -0.6667   -0.1667
```

```
>> imshow(imfilter(x,h2))
```





頻率域的影像處理

二維離散傅利葉轉換

- 下表列出了離散傅利葉轉換的相關函數：

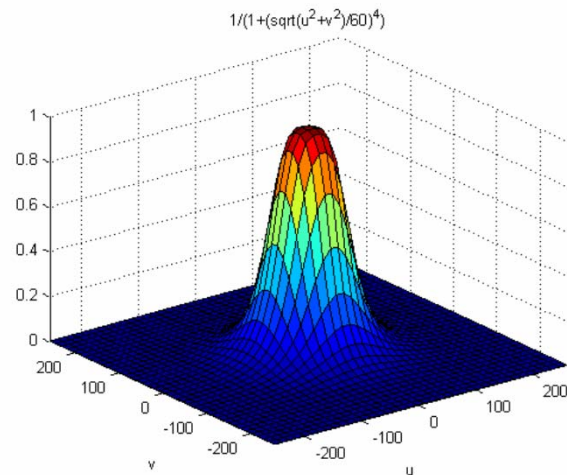
表 20.6.1 傅利葉轉換函數

函 數	說 明
<code>fft2(x)</code>	計算二維陣列 x 的離散傅利葉轉換
<code>ifft2(x)</code>	計算二維陣列 x 的離散反傅利葉轉換
<code>fftshift(x)</code>	將傅利葉轉換後，頻率域的原點移到頻率域方塊的正中間

傅利葉轉換的應用

- 下面是利用Butterworth濾波器來處理影像的例子：

```
>> x=imread('liftingbody.png');  
>> [u,v]=meshgrid(-255:256,-255:256);  
>> H=1./(1+(sqrt(u.^2+v.^2)/60).^4);  
>> ezsurf('1/(1+(sqrt(u^2+v^2)/60)^4)',[-255 256],[-255 256],40)
```



$$\frac{1}{1+(\sqrt{u^2+v^2}/60)^4}$$

```
>> F=fftshift(fft2(double(x)));  
>> y=ifft2(fftshift(H.*F));  
>> imshow(uint8(real(y)))
```

