

# 影像處理

## (Image Processing)

真理大學 資訊工程系 吳汶涓老師

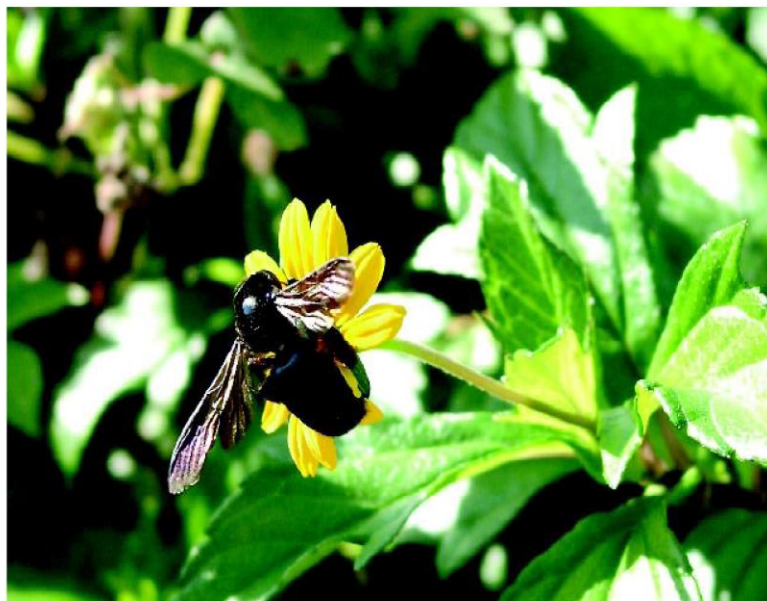
### Course 1

#### 導論



# Outline

- 1.1 數位與圖片
- 1.2 何謂影像處理
- 1.3 影像取樣與擷取
- 1.4 影像與數位影像
- 1.5 應用
- 1.6 影像處理的面向
- 1.7 影像處理工作
- 1.8 數位影像的類型
- 1.9 影像檔案大小
- 1.10 影像感知



# 1.1 影像與圖片

- 人類是一種**視覺**動物，強烈依賴視覺來感知周遭的世界。
  - 功能?
- 視覺景象的單一影像(image)，可稱為快照。

# 1.2 何謂影像處理

- Image Processing，是**改變影像的本質**，以便
  - 強化圖片資訊，便於人眼辨識
  - 轉化成更適合機器自動辨識的格式



圖 1.1 影像銳利化 (a) 原始影像 (b) 銳利化結果



(a)



(b)

圖 1.2 去除影像雜訊 (a) 原始影像 (b) 去除雜訊結果



(a)



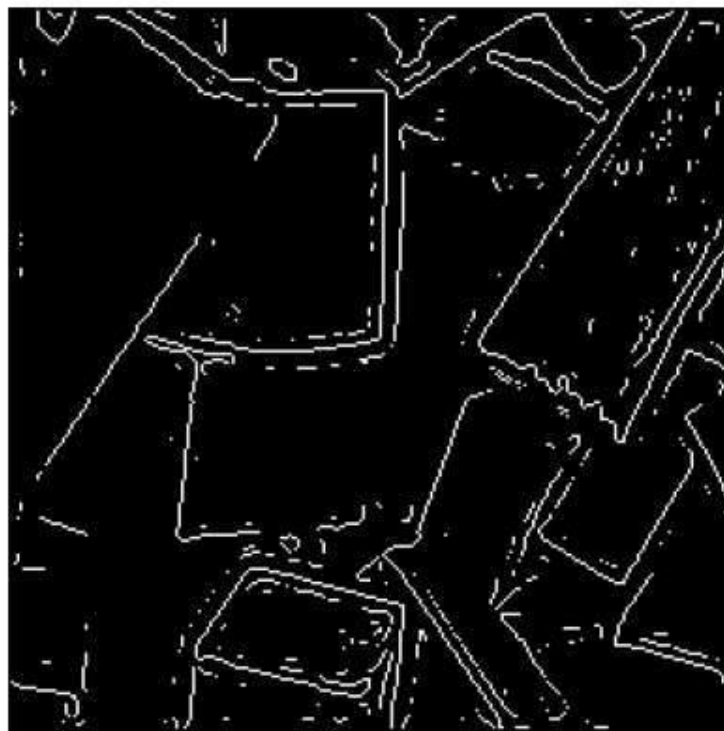
(b)

**圖 1.3** 去除影像模糊現象 (a) 原始影像 (b) 去除模糊現象結果



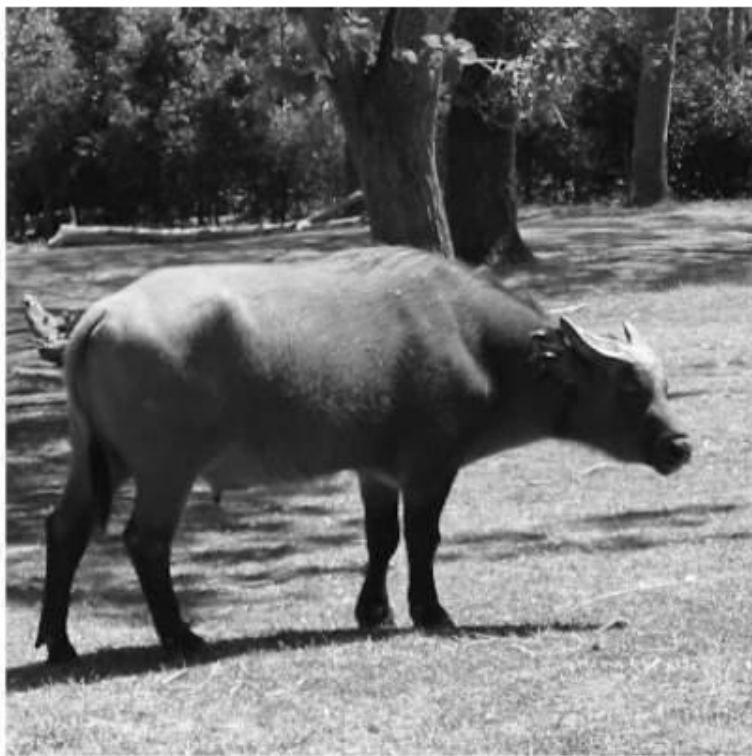


(a)



(b)

圖 1.4 取得影像邊緣線條 (a) 原始影像 (b) 物體邊緣線條



(a)



(b)

**圖 1.5** 影像模糊化 (a) 原本影像 (b) 模糊化後去除細節結果



## 1.3 影像取樣與擷取

- **取樣**（sampling）是將連續函數數位化的過程，舉例，假設將下列函數：

$$y = \sin(x) + \frac{1}{3} \sin(3x)$$

- 在 $x$ 軸上以相等的間隔取樣10個點
- 在 $x$ 軸上以相等的間隔取樣100個點



圖 1.6 函數取樣：低度取樣

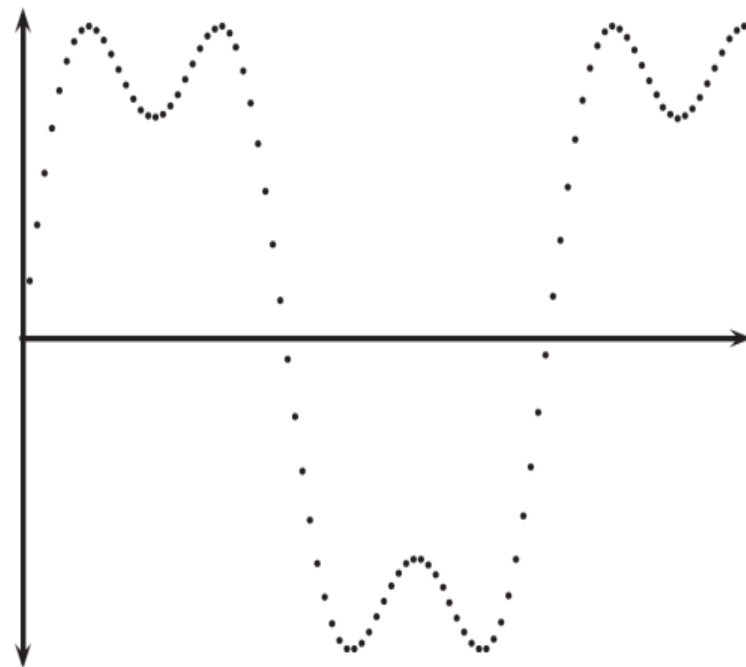
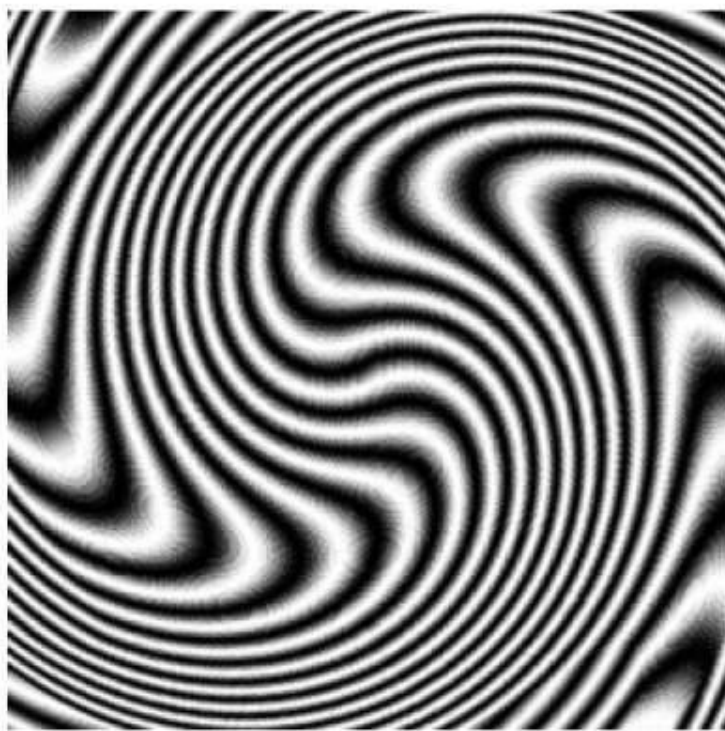
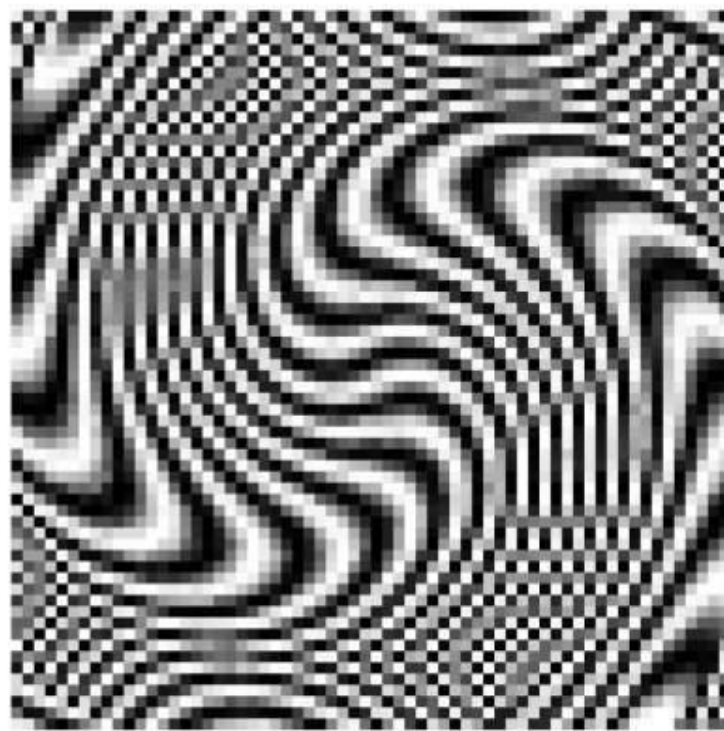


圖 1.7 函數取樣：使用更多取樣點



(a)



(b)

**圖 1.8** 取樣的效果 (a) 正確取樣，無混疊 (b) 低度取樣版本，有混疊現象

- 影像擷取
  - CCD相機

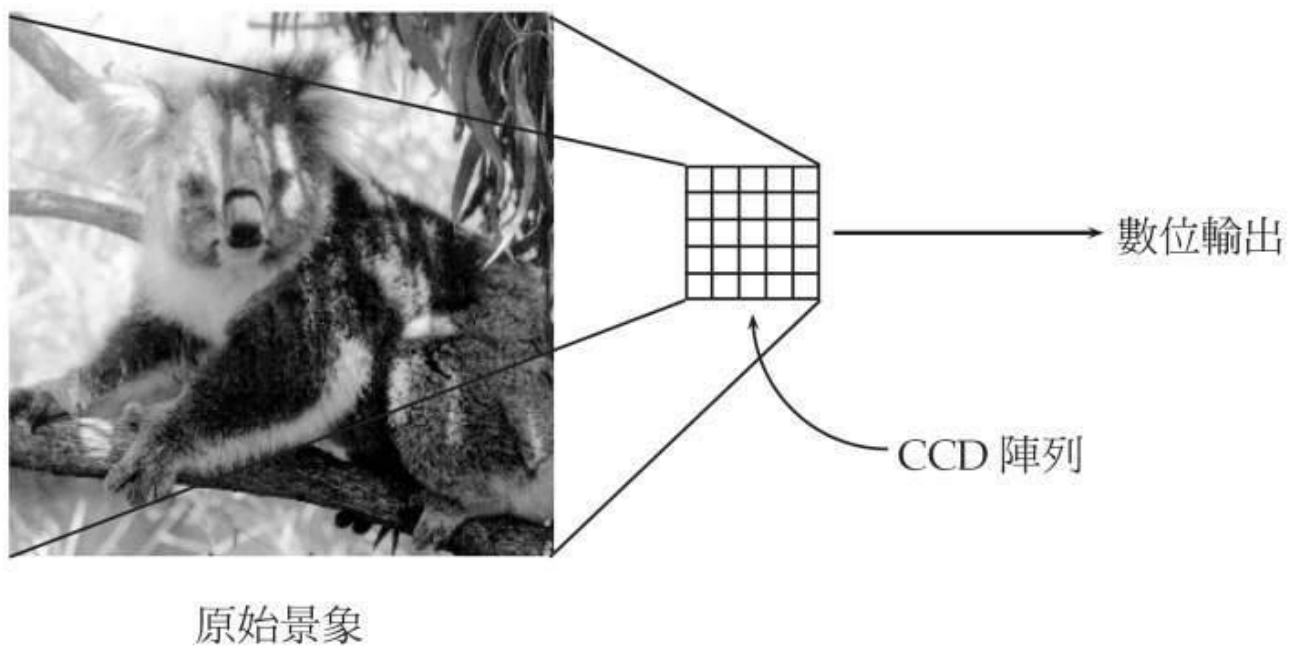


圖 1.9 原始景象使用 CCD 陣列擷取影像

## □ 平台式掃描器

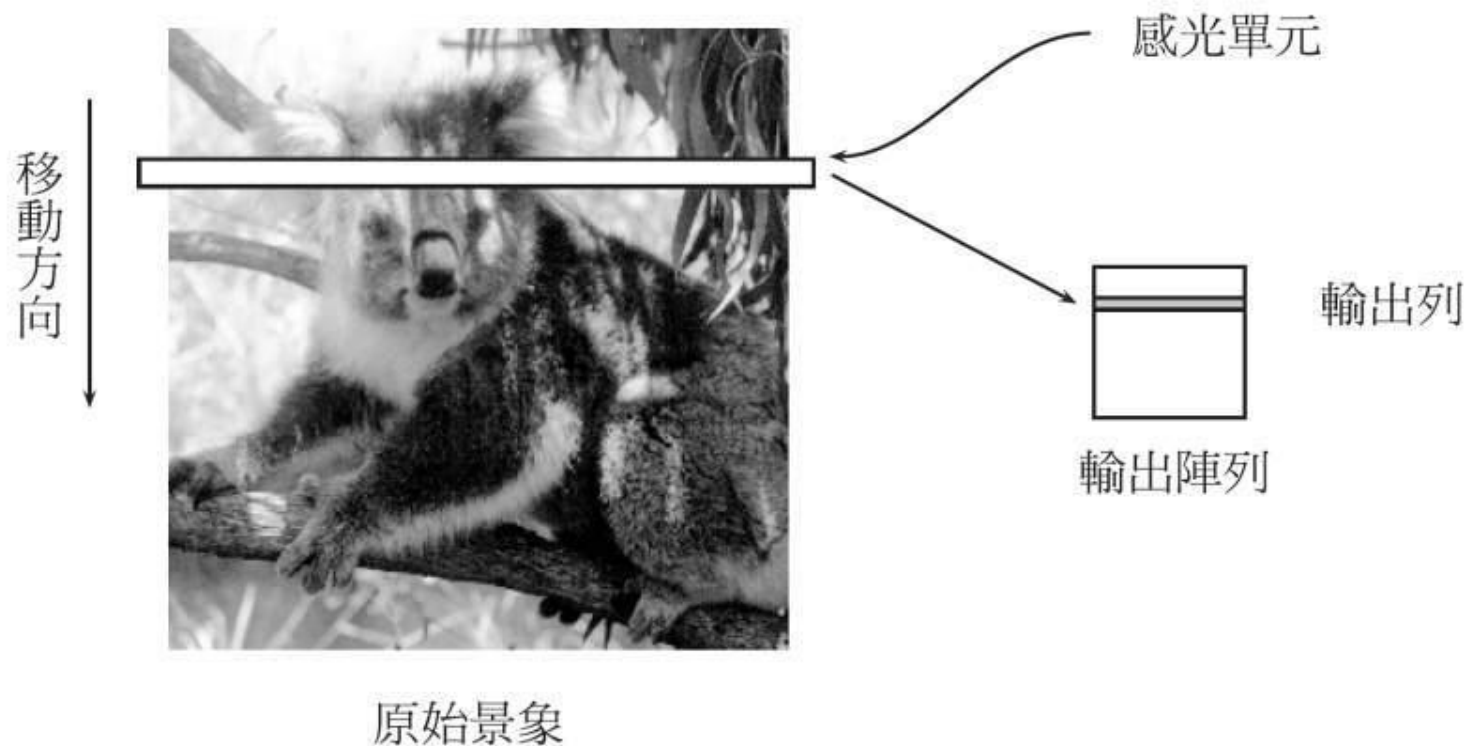


圖 1.10 原始景象使用 CCD 掃描器擷取影像

## □ 其他能量來源

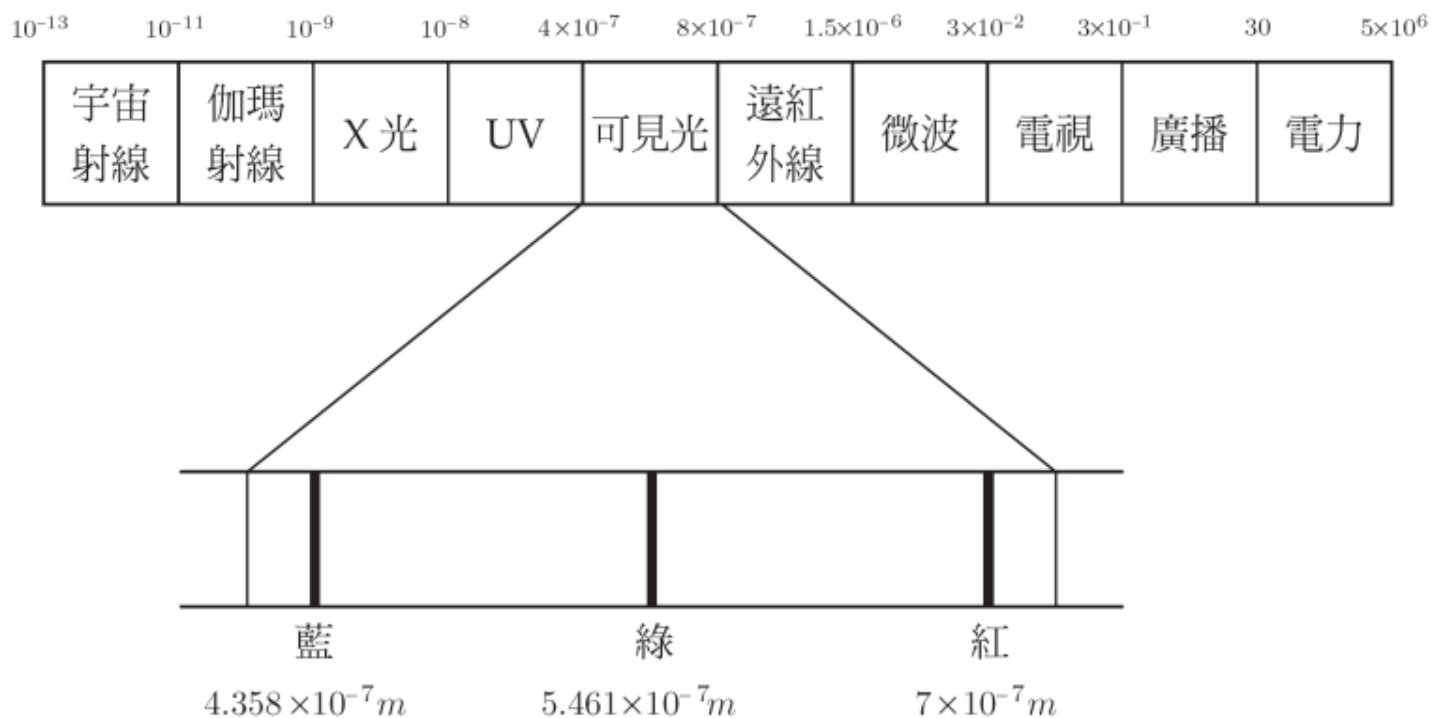


圖 1.11 電磁波頻譜



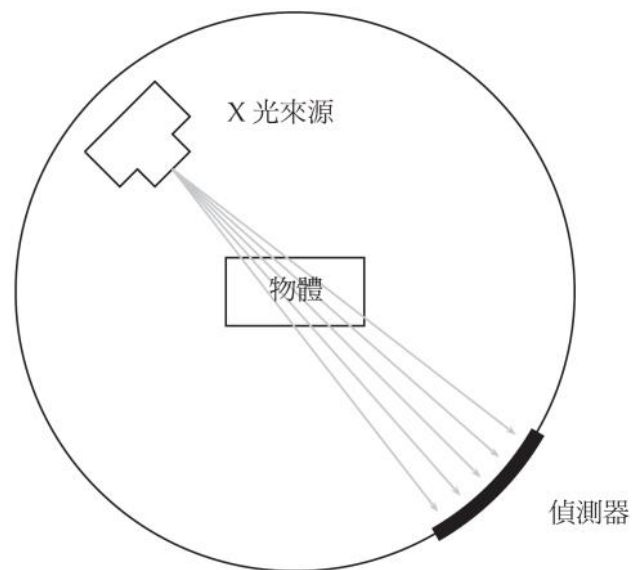
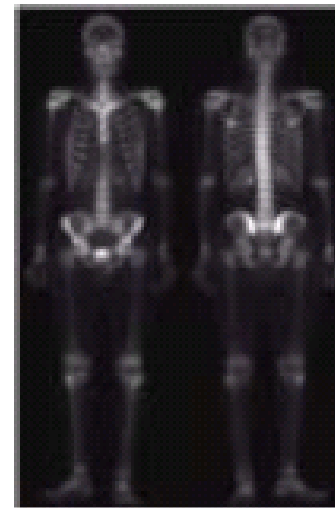


圖 1.12 X 光斷層攝影法



骨骼掃描  
(Bone scan)

# 1.4 影像與數位影像

- 假設一張照片是灰階的(只有不同深淺的灰影)，沒有色彩。這張影像視為一個二維函數：
  - $x$ 、 $y$ 、 $f(x,y)$  的值是不連續(離散)的
  - $f(x,y)$  值假設從0到 1

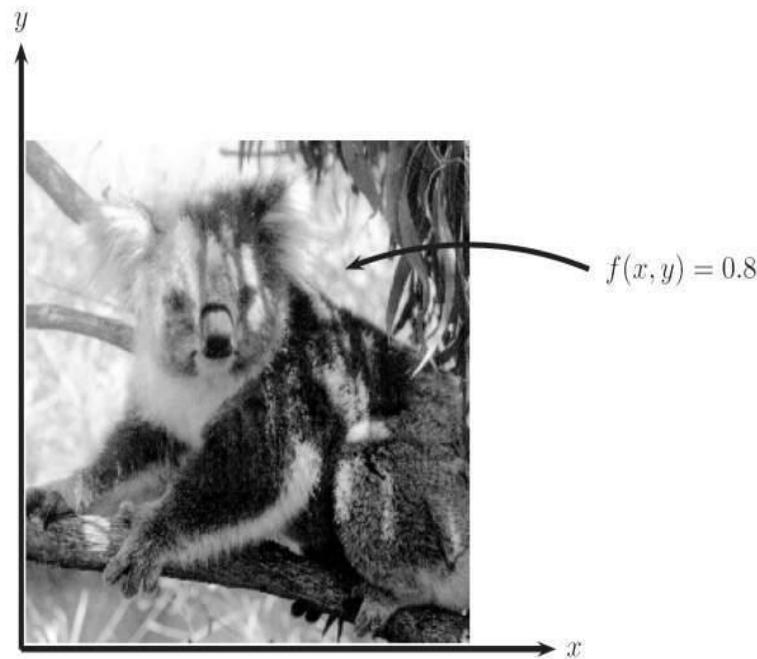


圖 1.13 影像視為函數

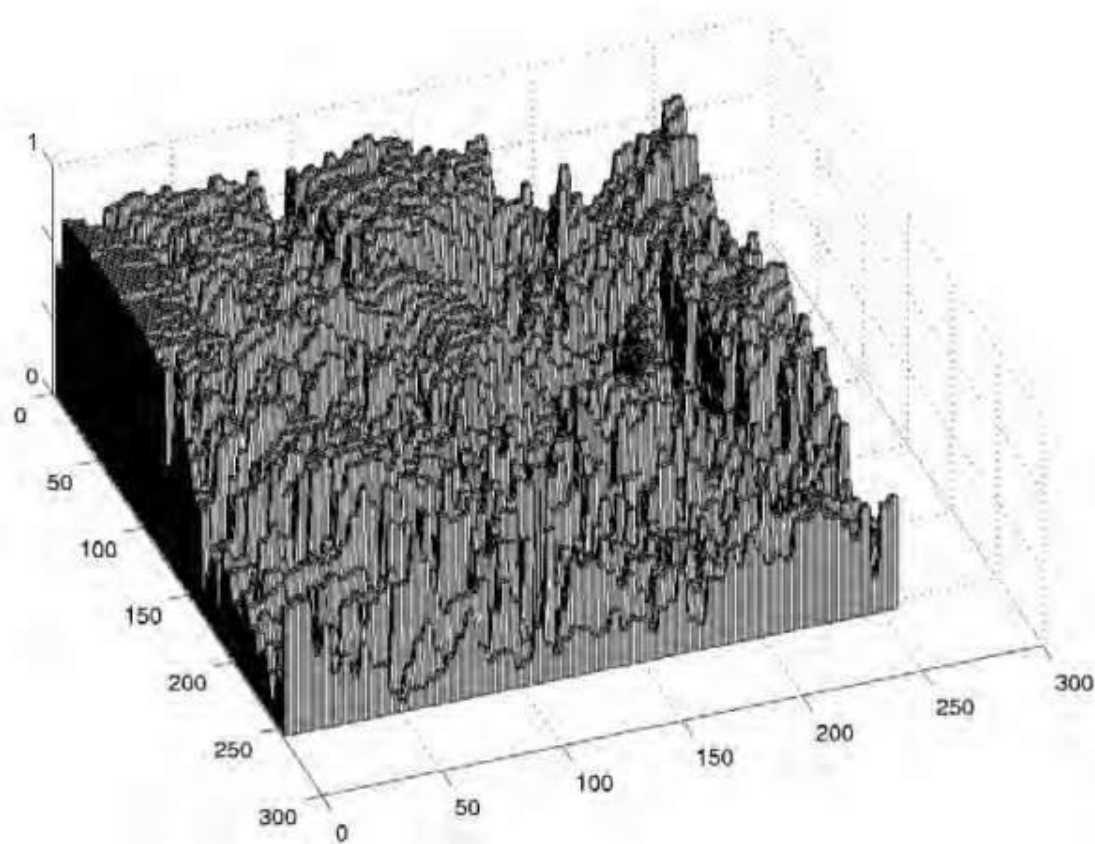


圖 1.14 圖 1.13 之二維變數函數製圖

- 數位影像可看成一個連續影像經過取樣而成的大型陣列，這些取樣點就是所謂的**像素**(pixels)

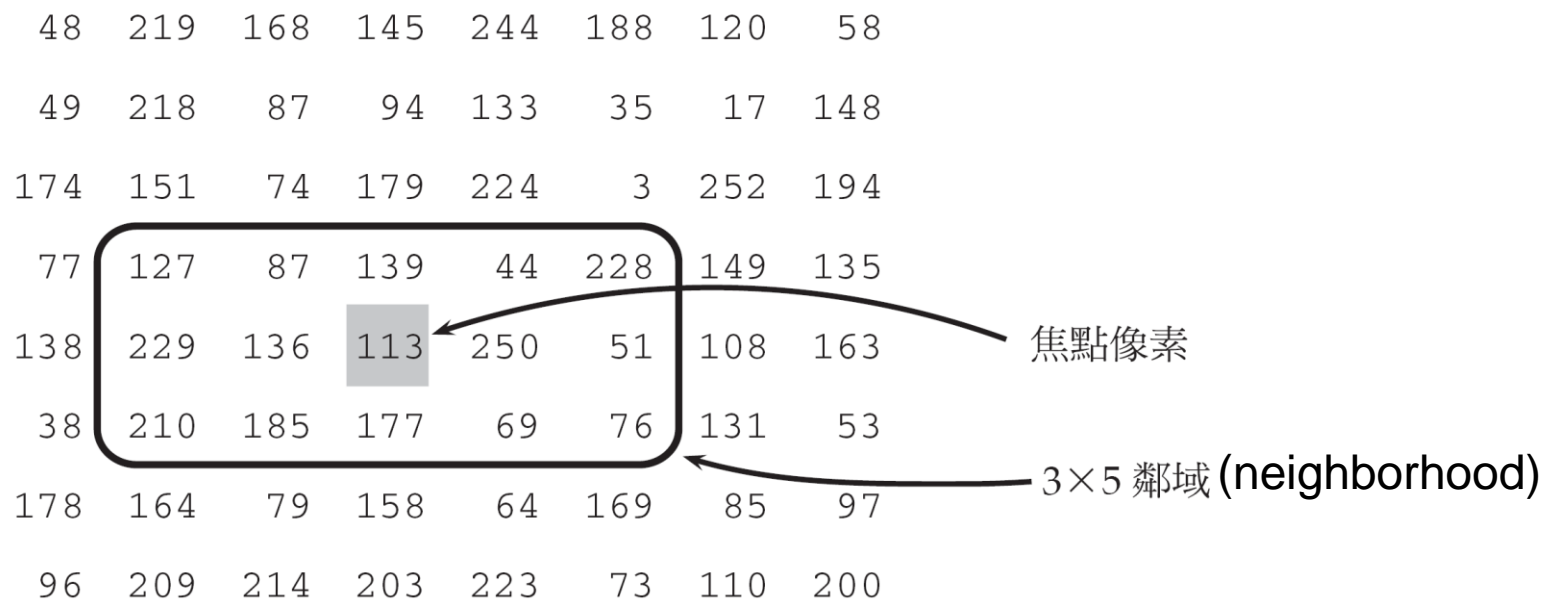


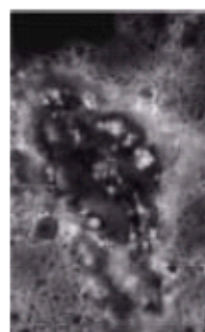
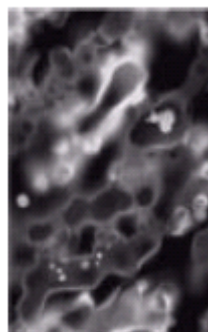
圖 1.15 像素及其鄰域

# 1.5 應用

- 影像處理的用途很廣，只要可透過視覺方式呈現，皆可以應用

- 醫療
- 農業
- 工業
- 法律

正常



患黑穗病玉米微影



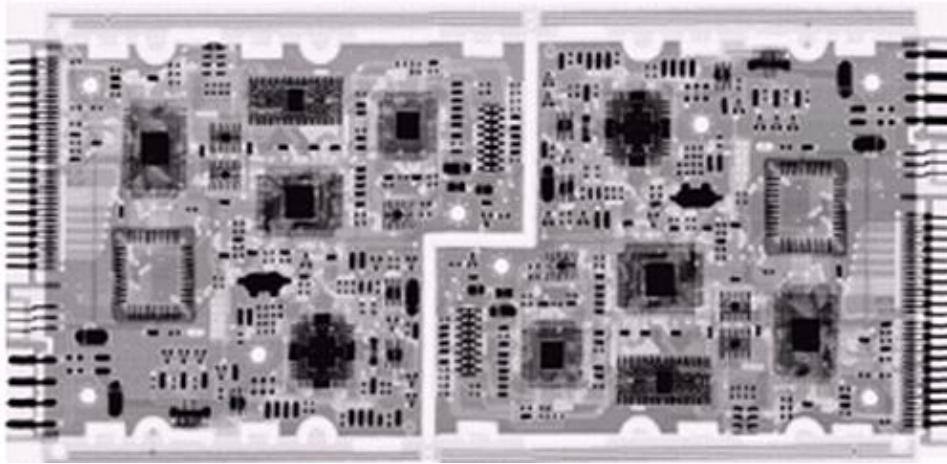
骨骼掃描  
(Bone scan)



美國 D.C.  
遙測影像  
(Remote  
sensing)



指紋影像(Fingerprint)

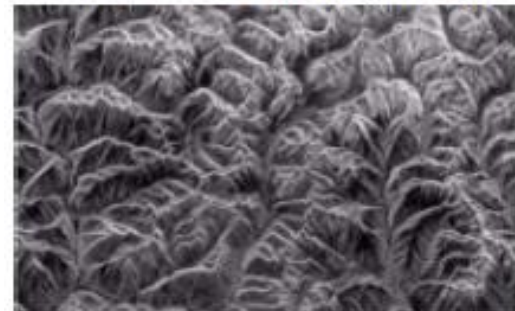


嬰兒超音波影像  
(Ultrasound)

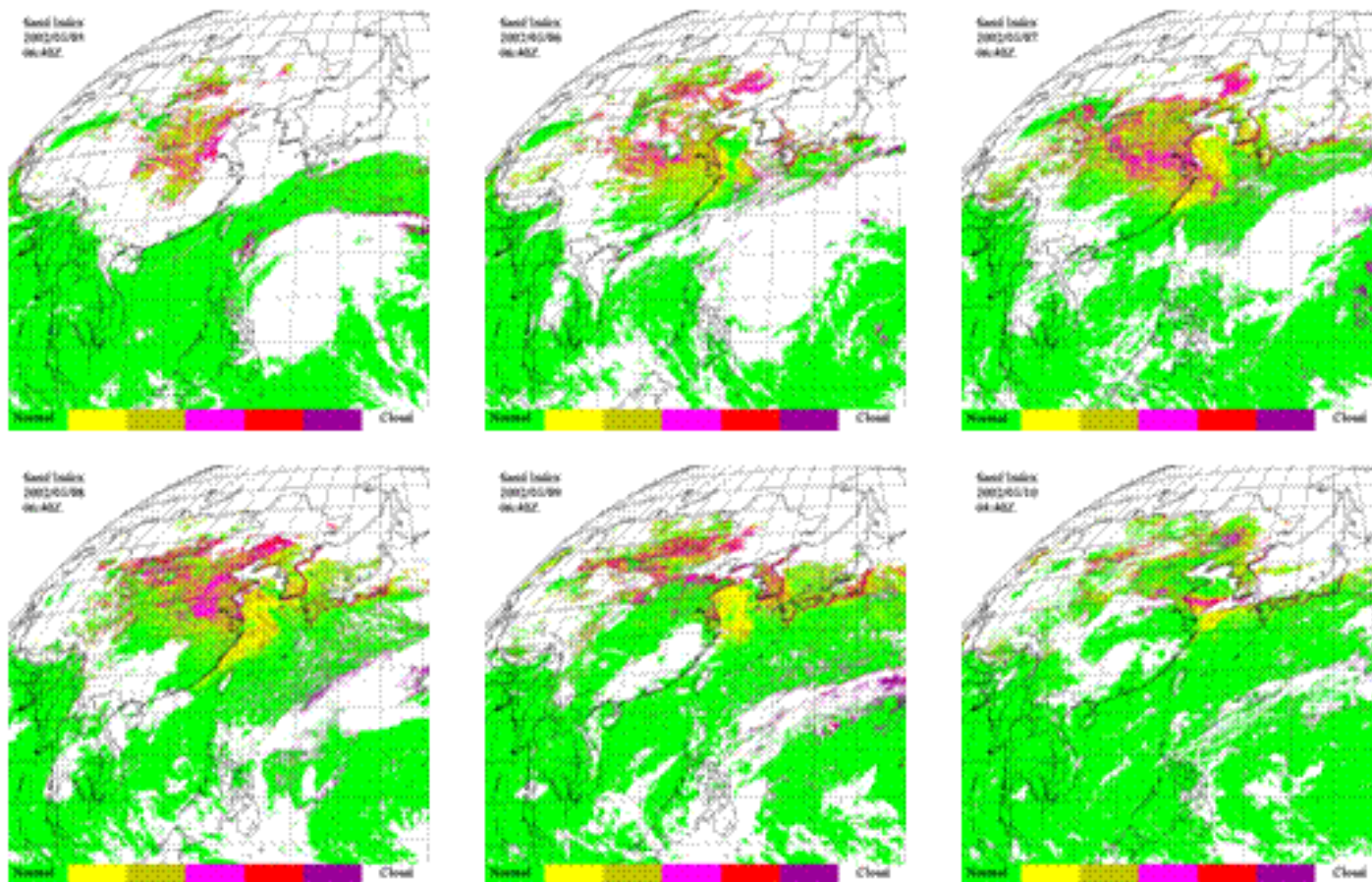


紅外線影像  
(美國)

微波空照影像(西藏)







2002 年 3 月 5 日到 3 月 10 日沙塵暴連續結果(05:40Z)，圖中之綠色系列表未受沙塵暴影響之正常地區，紅黃色系為受沙塵暴影響之區域，而白色則表示為雲所覆蓋之區域。

## 遙測資料分析



## Image Classification



# 1.6 影像處理的面向

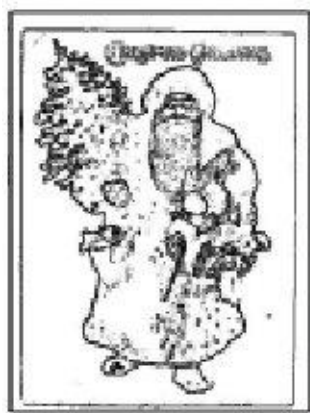
- 影像增強
- 影像回復
- 影像切割



低對比度影像



經影像加強之影像



Edge image

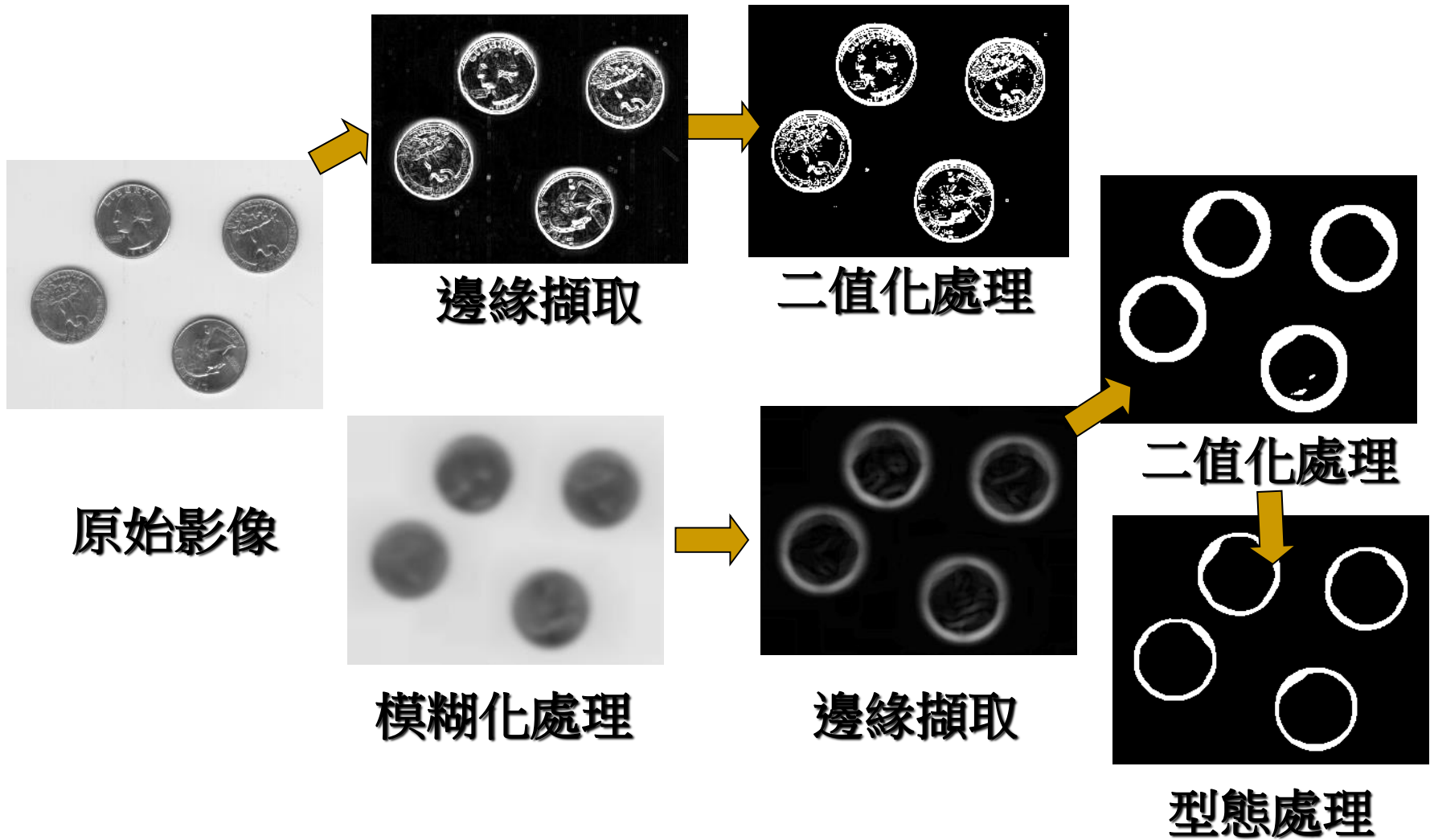


Segmented image



## 1.7 影像處理工作

- 擷取影像（acquiring the image）
- 前置處理（preprocessing）
- 切割（segmentation）
- 表示與描述（representation and description）
- 辨識與解讀（recognition and interpretation）



# 1.8 數位影像的類型

## ■ 二元數位影像 (binary)

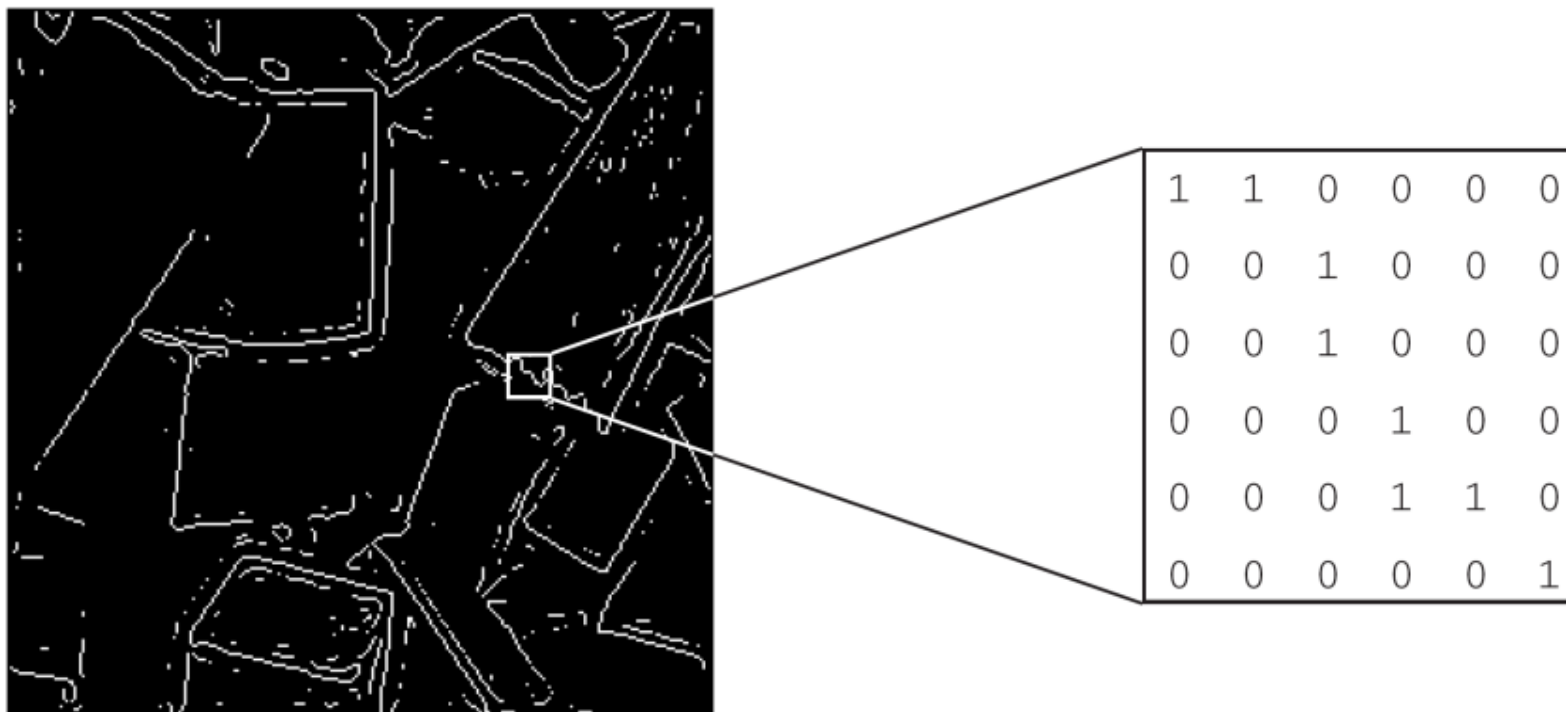


圖 1.16 二元數位影像



## ■ 灰階影像 (grayscale)



圖 1.17 灰階影像

## ■ 索引影像 (Indexed)



4	5	5	5	5	5
5	4	5	5	6	6
5	5	5	0	8	9
5	5	5	5	11	11
5	5	5	8	16	20
8	11	11	26	33	20
11	20	33	33	58	37

索引值

0.1211	0.1211	0.1416
0.1807	0.2549	0.1729
0.2197	0.3447	0.1807
0.1611	0.1768	0.1924
0.2432	0.2471	0.1924
0.2119	0.1963	0.2002
0.2627	0.2588	0.2549
0.2197	0.2432	0.2588
⋮	⋮	⋮

色譜

圖 1.19 索引彩色影像

## ■ 全彩或RGB（紅綠藍）影像

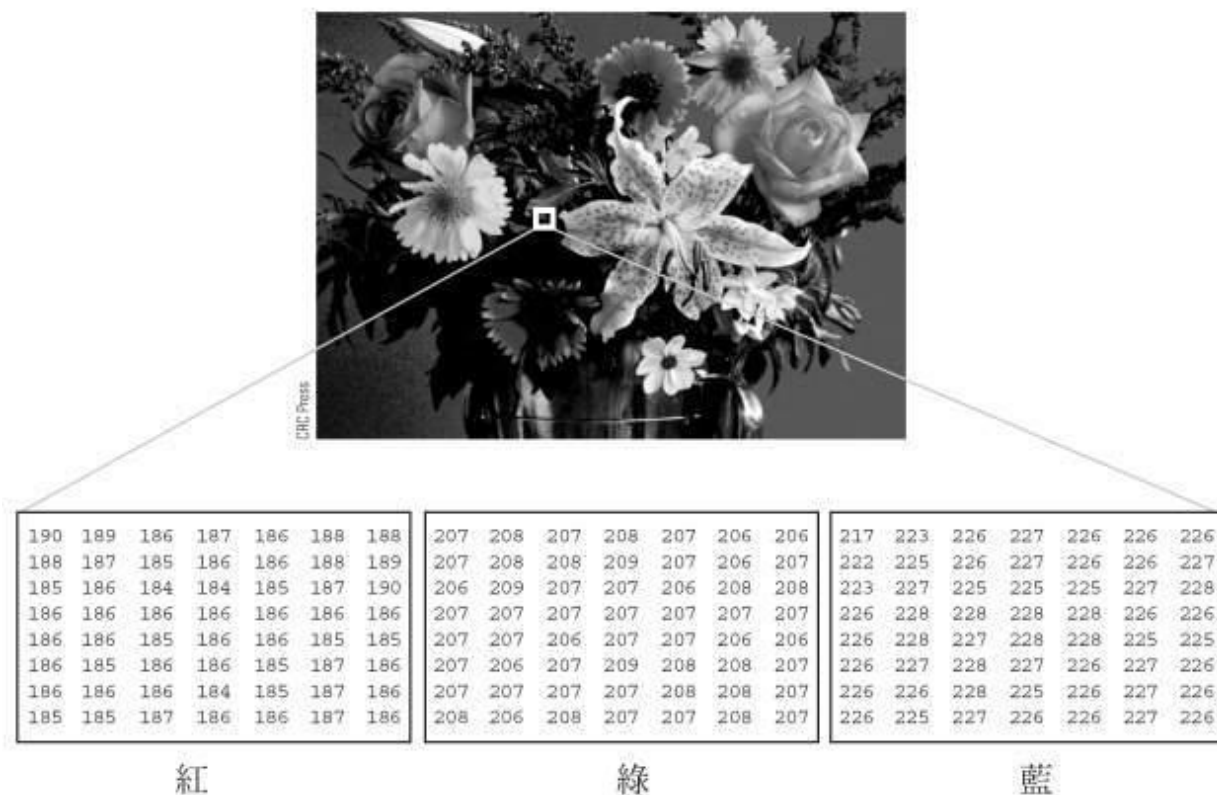


圖 1.18 全彩影像

## 1.9 影像檔案大小

- 512×512 的二元數位影像檔

$$\begin{aligned} 512 \times 512 \times 1 &= 262,144 \text{ bits} \\ &= 32768 \text{ bytes} \\ &= 32.768 \text{ Kb} \\ &\approx 0.033 \text{ Mb} \end{aligned}$$

- 同樣大小的灰階影像則需要：

$$\begin{aligned} 512 \times 512 \times 1 &= 262,144 \text{ bytes} \\ &= 262.14 \text{ Kb} \\ &\approx 0.262 \text{ Mb} \end{aligned}$$

## ■ 彩色影像

$$\begin{aligned} 512 \times 512 \times 3 &= 786,432 \text{ bytes} \\ &= 786.43 \text{ Kb} \\ &\approx 0.786 \text{ Mb} \end{aligned}$$

## 1.10 影像感知

- 影像處理是為了讓影像呈現較符合**人眼所需**，因此必須考慮人類視覺系統的限制

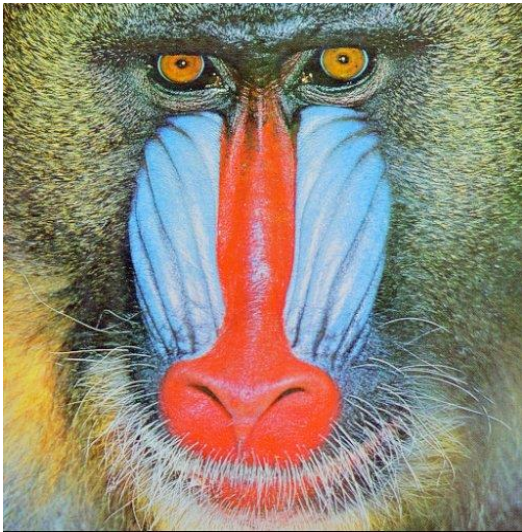


圖 1.20 不同背景的灰色方塊



# 補充

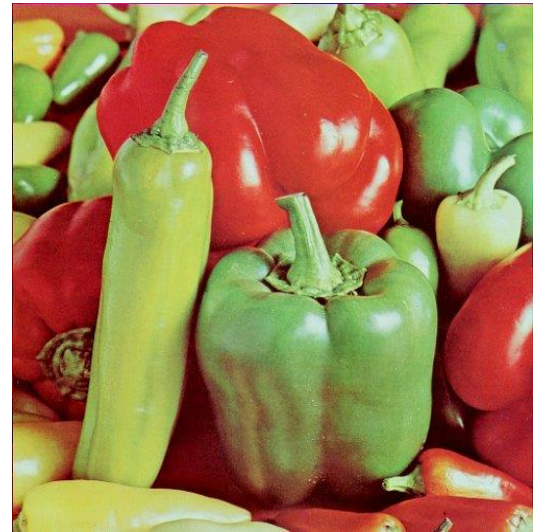
## ■ Test images



Baboon



Lena



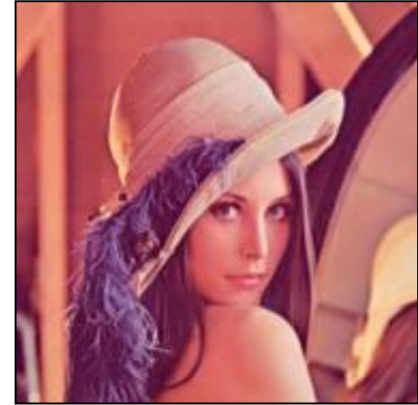
Pepper

## ■ 檔案格式

□ 原始檔 (raw file), BMP, GIF, JPG, PNG, ...

圖檔格式	黑白	16 色	256 色	灰階	全彩	CMYK	壓縮方式	其 它
BMP	✓	✓	✓	✓	✓		RLE 非破壞性	只有索引色和灰階 能壓縮
GIF	✓	✓	✓	✓			LZW 非破壞性	可存成透明圖、交 錯圖與動畫
JPG				✓	✓	✓	破壞性	可設定壓縮比率
PNG	✓	✓	✓	✓	✓		非破壞性	新的網頁圖檔格式
PCD			✓	✓	✓			PhotoImpact 無法存 成PCD,但可開啓
TIF	✓	✓	✓	✓	✓	✓	LZW 非破壞性	另支援 16 bit 灰階 與48 bit 全彩
UFO	✓	✓	✓	✓	✓		非破壞性	可儲存未合併的物 件、路徑及選取區

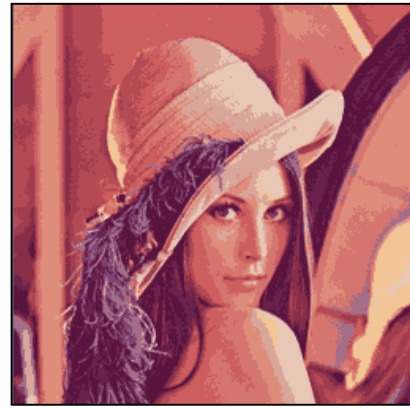
全彩



二元



灰階



64色索引色

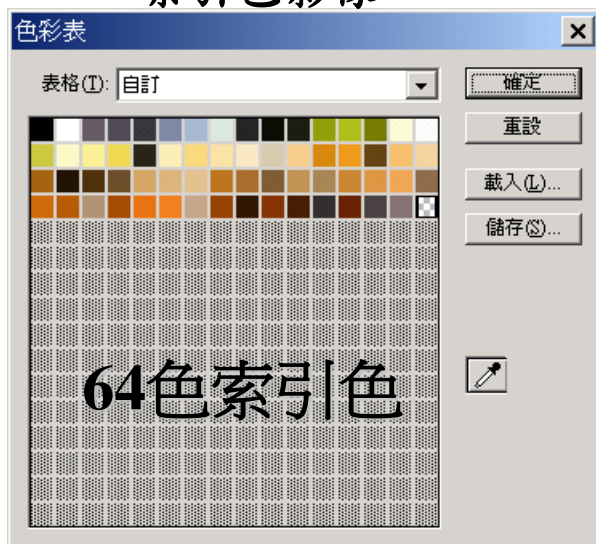


256色索引色





索引色影像



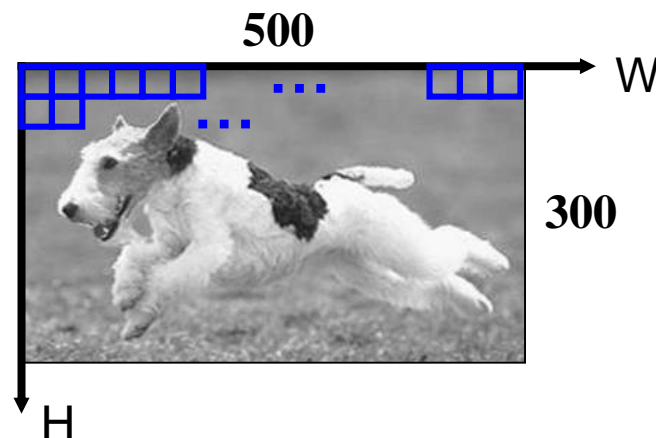
## ■ 宣告-灰階影像

- ❑ 大小為  $H \times W$  pixels
- ❑  $W=500$ ,  $H=300$
- ❑ C語言語法：

```
#define W 500
```

```
#define H 300
```

```
unsigned char Image[H][W];
```



## ■ 讀取(寫入)影像

```
FILE *rfile;
```

```
rfile=fopen("DOG.raw", "rb");
```

"wb"

```
for (i=0; i<H; i++)
```

```
    for (j=0; j<W; j++)
```

```
        Image[i][j]=(int) fgetc(rfile);
```

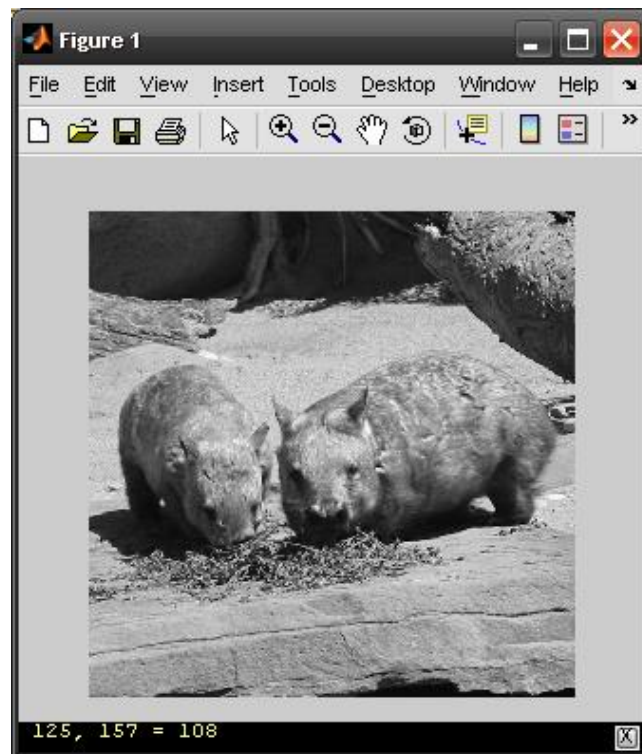
```
fclose(rfile);
```

```
fwrite("output.raw", sizeof(unsigned char), H*W, rfile);
```



## ■ Matlab 語法：

- ❑ Matlab 指令視窗
- ❑ Matlab 提示符號：>>
- ❑ >> **w**=imread( ' wombats.tif ' );
- ❑ >> size(**w**)  
ans = 256 256
- ❑ figure, imshow(**w**), pixval on
- ❑ Imshow ( ' wombats.tif ' );



# Exercise

## ■ 學習如何撰寫Matlab函數

- 開啟Matlab，用滑鼠點選 File->New->M-File，開啟一個新的M-File
- 用編輯器輸入

```
function [output]=Add(Input1, Input2)
    a=Input1;
    b=Input2;
    output=a+b;
```
- 儲存Add.m檔案於目前路徑下
- 回到command window 打入 Add(10, 10)，輸出結果？
- 自己改寫成減法函數