

影像處理

(Image Processing)

真理大學 資訊工程系 吳汶涓老師

Course 4
點處理



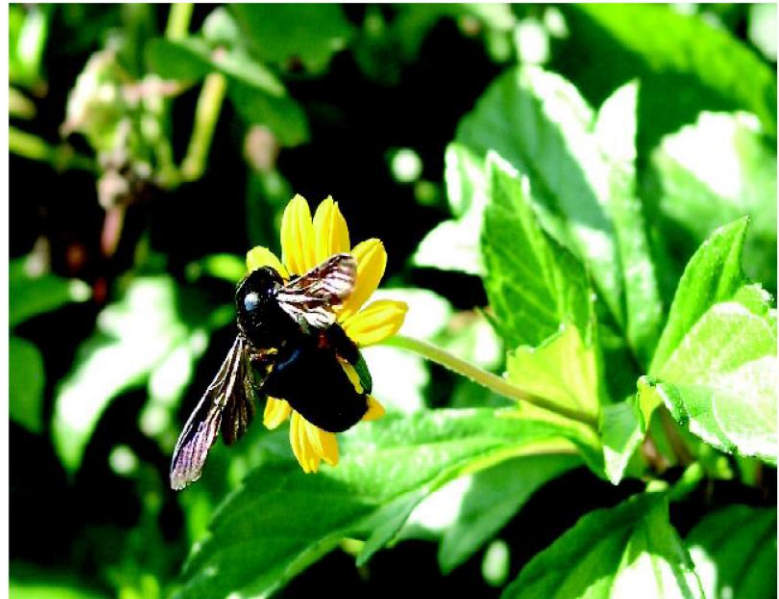
Outline

4.1 前言

4.2 數學運算

4.3 直方圖(灰階值分布圖)

4.4 查詢表格



4.1 前言

- 影像處理運算可分成三個等級：
 - 轉換 (transforms)
 - 鄰域處理 (neighborhood processing)
 - 點運算 (point operations)

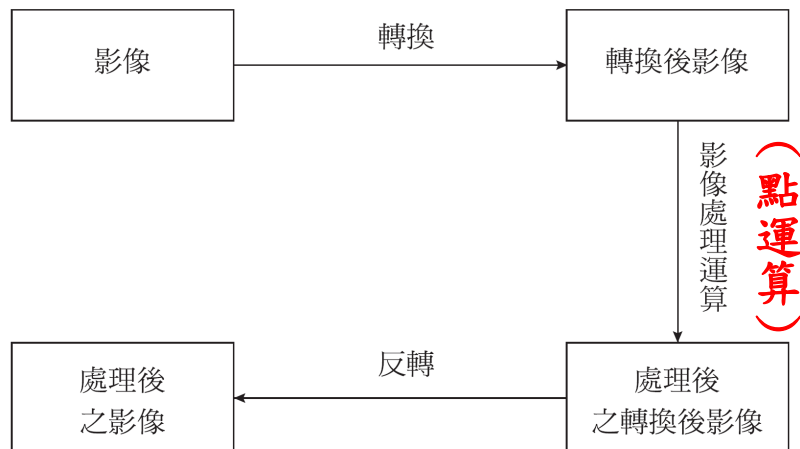


圖 4.1 轉換處理的結構

4.2 數學運算

- 數學運算是對每個像素進行某些簡單的函數運算。

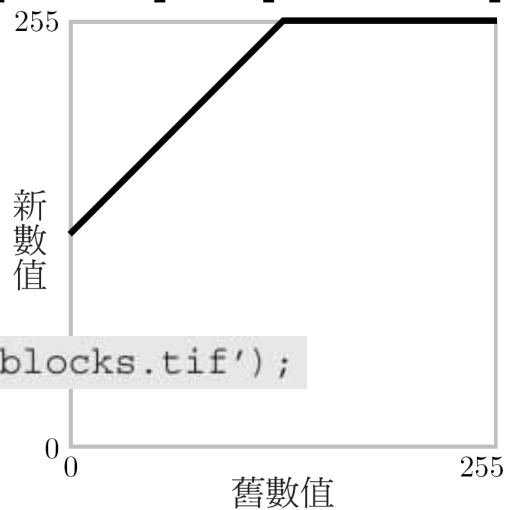
$$y = f(x) \leftarrow \begin{matrix} y = x + C \\ y = x * C \\ \dots \end{matrix}$$

- 不管是哪一種函數都必須調整輸出結果，讓結果落在0...255的範圍之內。

$$y \leftarrow \begin{cases} 255 & \text{若 } y > 255 \\ 0 & \text{若 } y < 0 \end{cases}$$

$$y = x + 128$$

$$[0, 255] \rightarrow [128, 255]$$



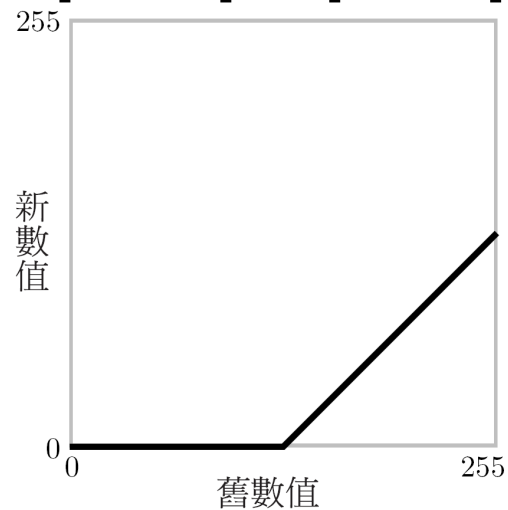
```
>> b=imread('blocks.tif');
```

```
>> b1=imadd(b,128);
```

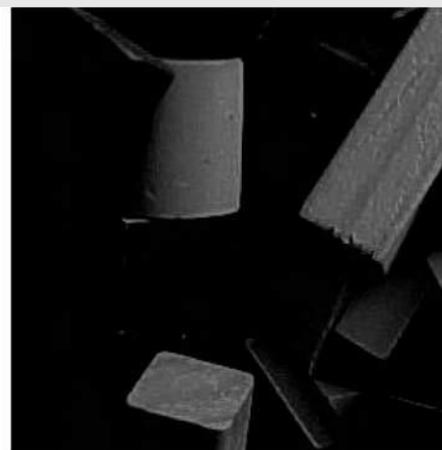


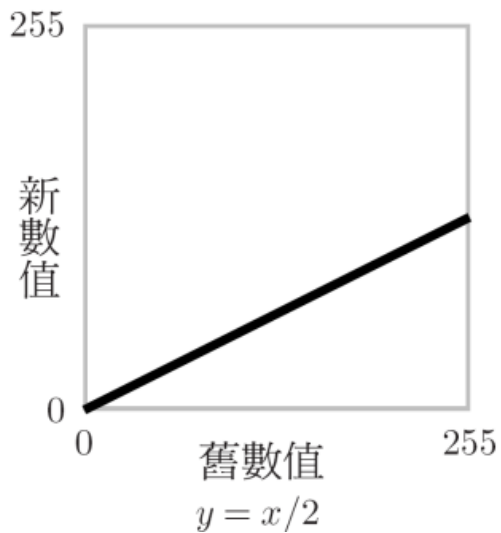
$$y = x - 128$$

$$[0, 255] \rightarrow [0, 127]$$

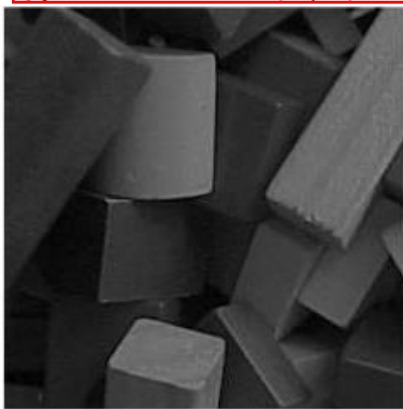


```
>> b2=imsubtract(b,128);
```

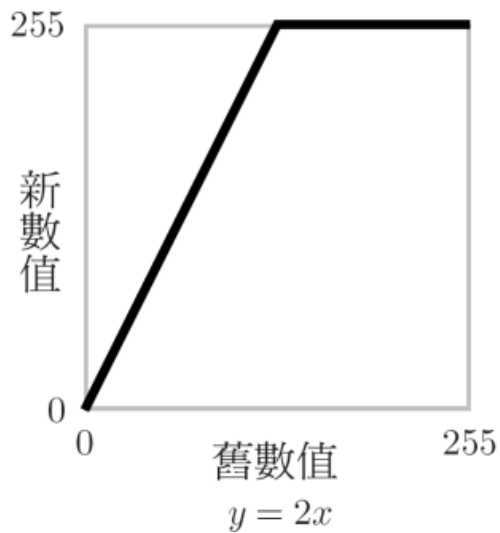




```
b3=immultiply(b,0.5);  
或 b3=imdivide(b,2)
```



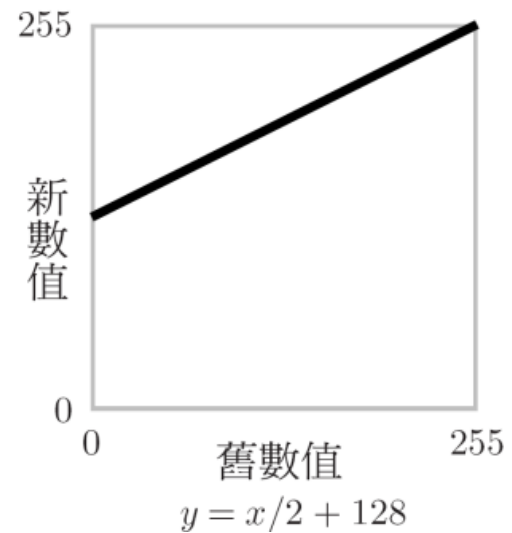
b3 : $y = x/2$



```
b4=immultiply(b,2);
```



b4 : $y = 2x$



```
b5=imadd(immultiply(b,0.5),128);  
b5=imadd(imdivide(b,2),128);
```

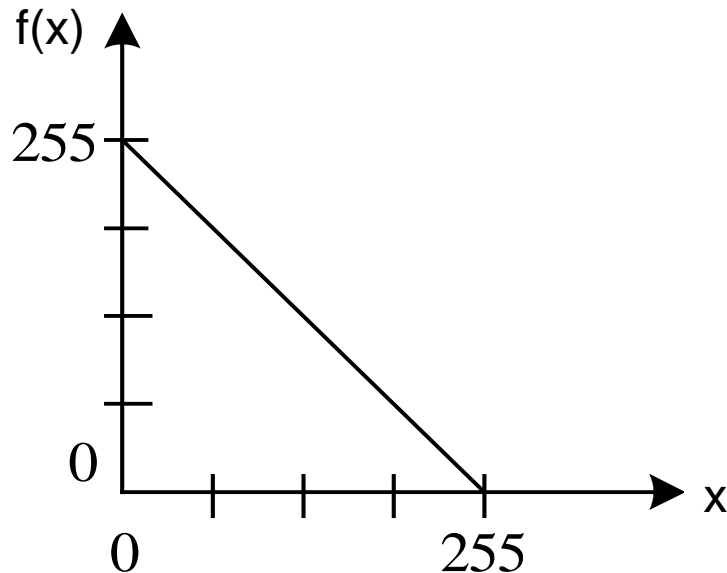


b5 : $y = x/2 + 128$

- **補色**(complement)：就是對應於相片的負片

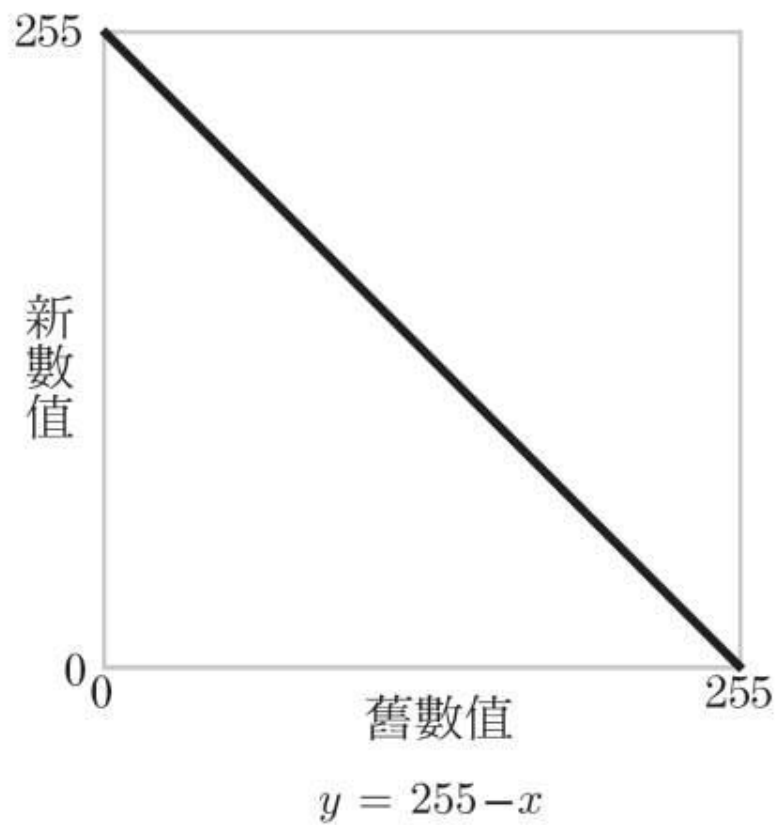
$$f(x) = 255 - x$$

→ 白變黑、黑變白



練習：

```
>> x= imread("blocks.tif");  
>> xc=imcomplement(x);  
>> imshow(xc)
```



```
>> bc=imcomplement(b);  
>> imshow(bc)
```



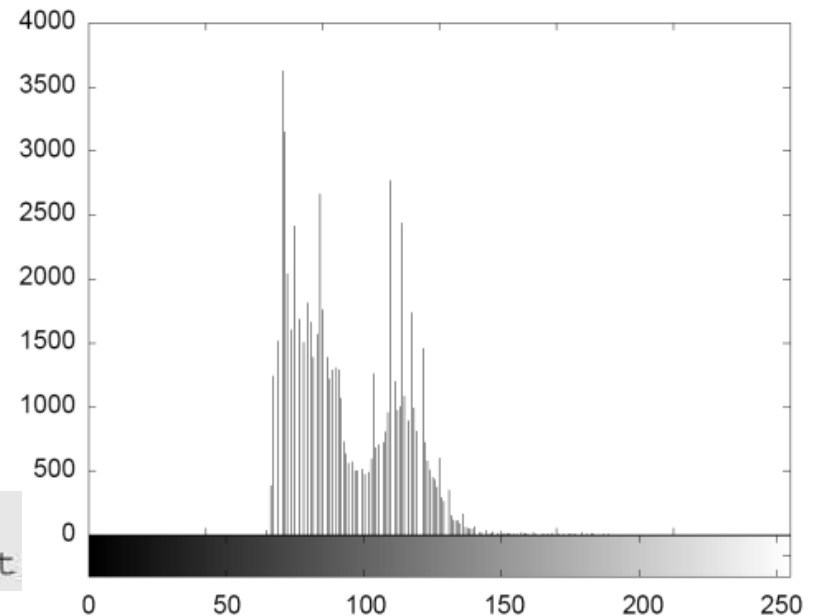
圖 4.6 影像補色

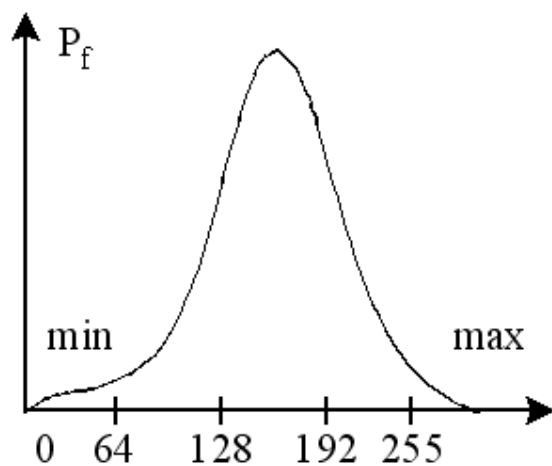
4.3 直方圖(灰階值分布圖)

- **直方圖**(histogram)是一張標示每個灰階值在影像中出現次數的圖表
 - 使用**imhist**函數來檢視影像的直方圖

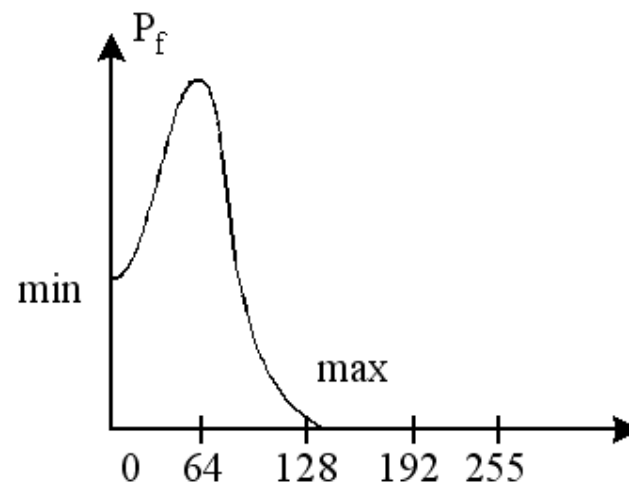


```
>> p=imread('pout.tif');  
>> imshow(p),figure,imhist(p),axis tight
```

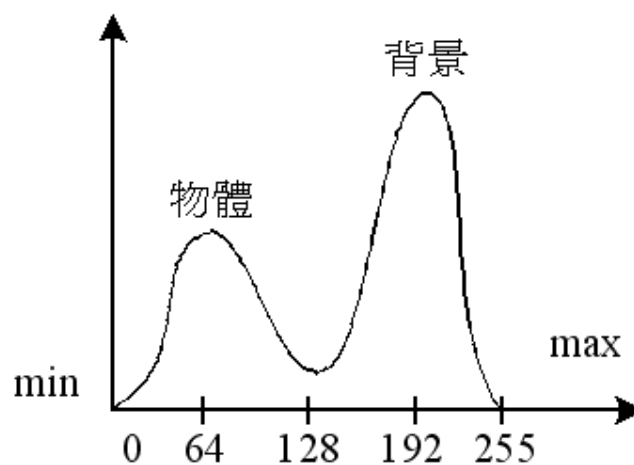




(a) 全動態範圍，對比低



(b) 偏暗



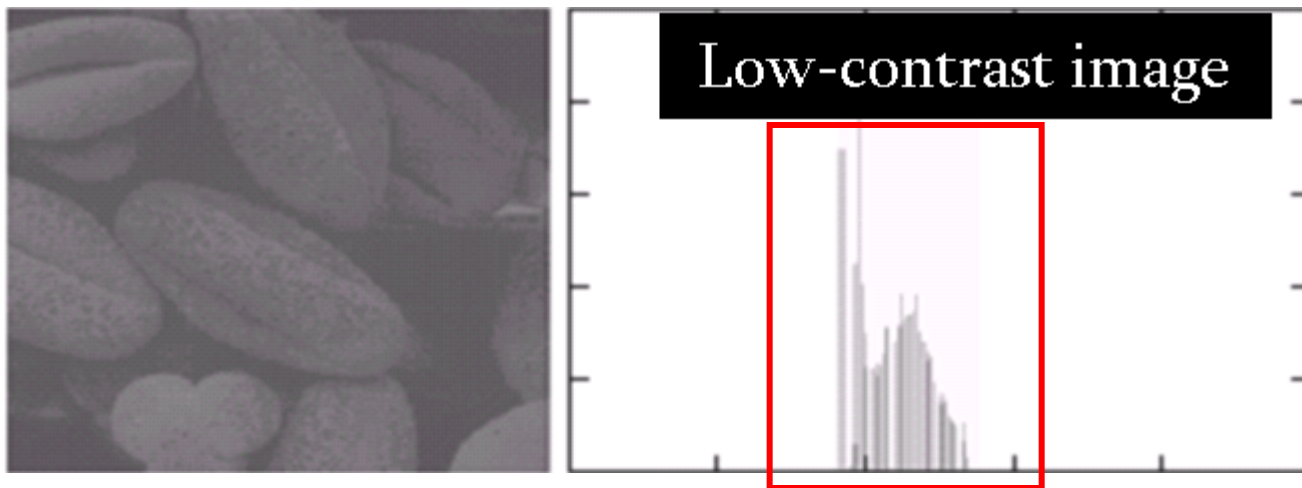
(c) 前景和背景分明

■ 直方圖擴展法(對比擴展法)

→ 動態範圍太窄 (min, max)

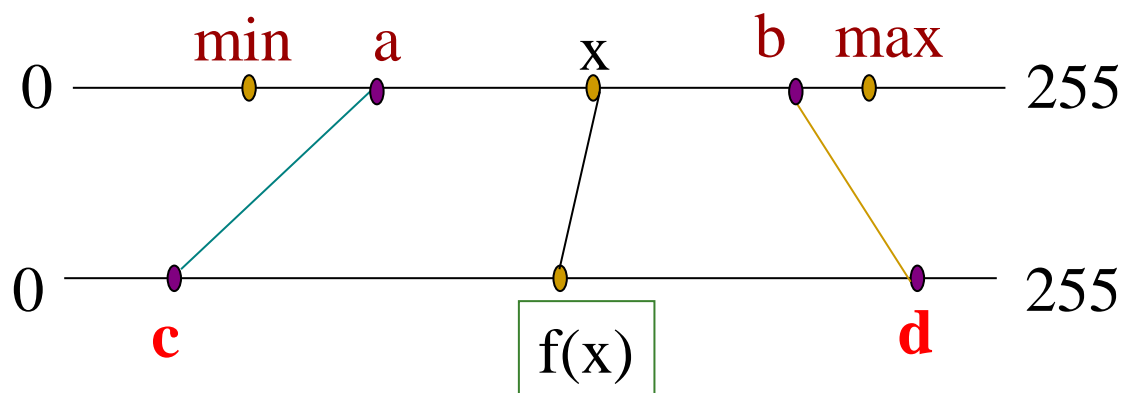


映設至最大範圍(0, 255)



讓使用者設定伸展後的像素最大最小值

灰階層次	i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	n_i	15	0	0	0	0	70	110	45	70	35	0	0	0	0	0	15



$$f(x) = \frac{(x-5)}{(9-5)} * (14-2) + 2$$

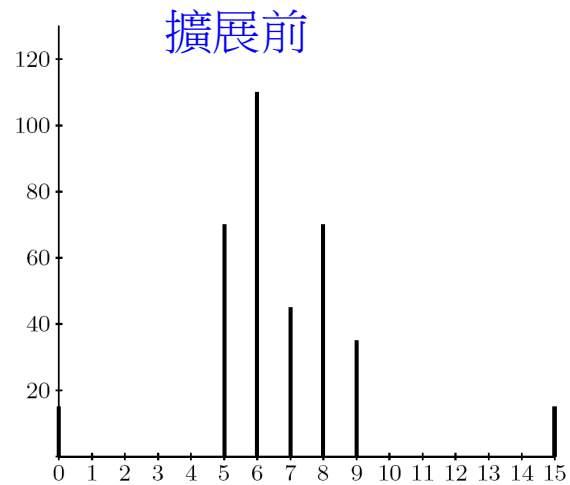
```
imadjust(im, [a,b], [c,d])
```

```
p=imread('pout.tif');  
imadjust(p, [5,9], [2,14]);
```

$$\frac{b-a}{d-c} = \frac{x-a}{f(x)-c}$$

$$(b-a) * (f(x)-c) = (x-a) * (d-c)$$

$$f(x) = c + \frac{(x-a)}{(b-a)} * (d-c)$$



x	5	6	7	8	9
$f(x)$	2	5	8	11	14

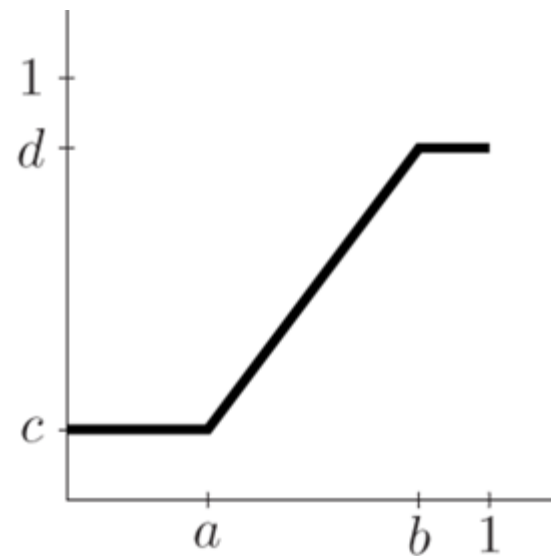
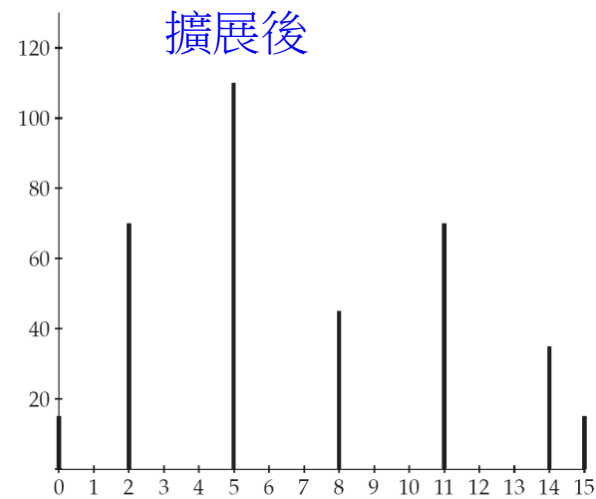


圖 4.10 imadjust 的擴展函數

- imadjust函數還有另一個可選用的參數：
gamma值

$$y = \left(\frac{x - a}{b - a} \right)^\gamma (d - c) + c$$

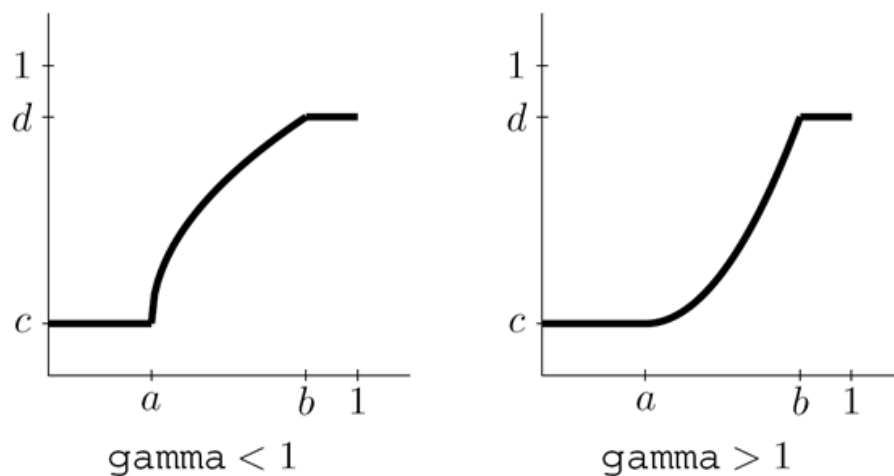


圖 4.11 gamma 值不等於 1 時的 imadjust 函數

```
>> t=imread('tire.tif');  
>> th=imadjust(t,[],[],0.5);  
>> imshow(t),figure,imshow(th)
```



圖 4.12 輪胎影像及 gamma 值修正後的輪胎影像

```
>> plot(t,th,'.'),axis tight
```

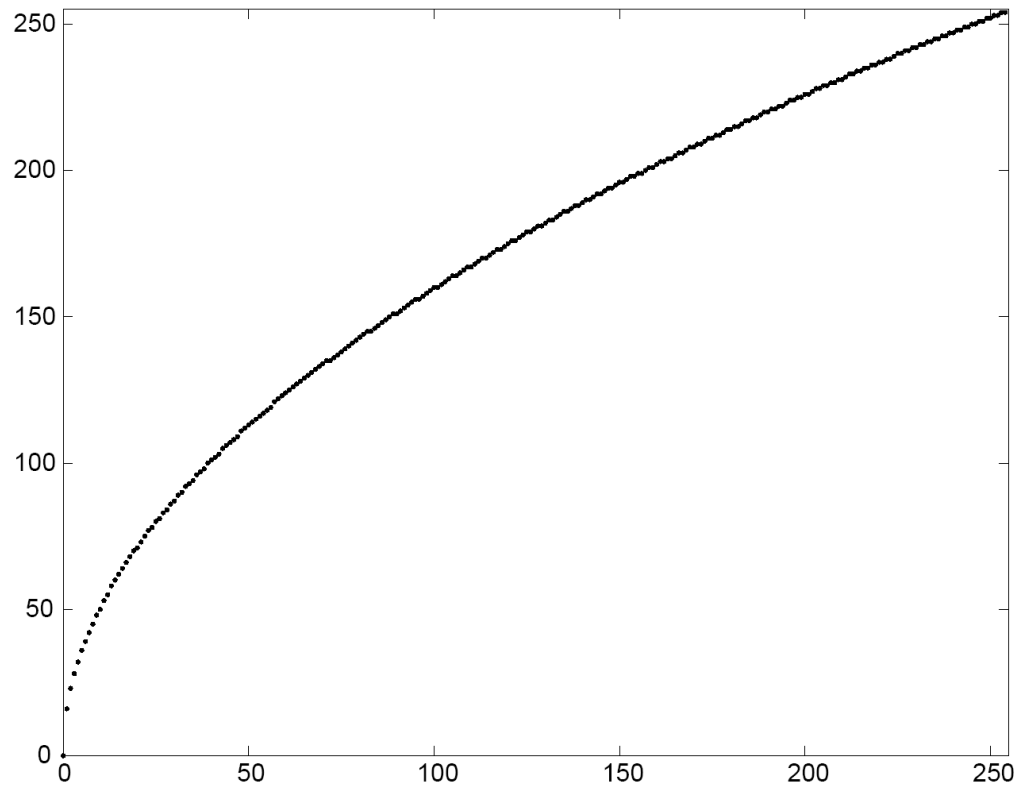


圖 4.13 圖 4.12 中使用的函數

■ 直方圖等化法 (histogram equalization)

→ 達到影像全動態範圍且灰階值分佈均勻

□ 全自動化(適性、不需人工操作選a、b值)

□ 最佳的對比

□ 使用 histeq 函數

```
>> p=imread('pout.tif');  
>> ph=histeq(p);  
>> imshow(ph),figure,imhist(ph)
```

□ 作法:將影像看作連續函數 $f(x, y)$ ，而直方圖就當作機率密度函數來處理。

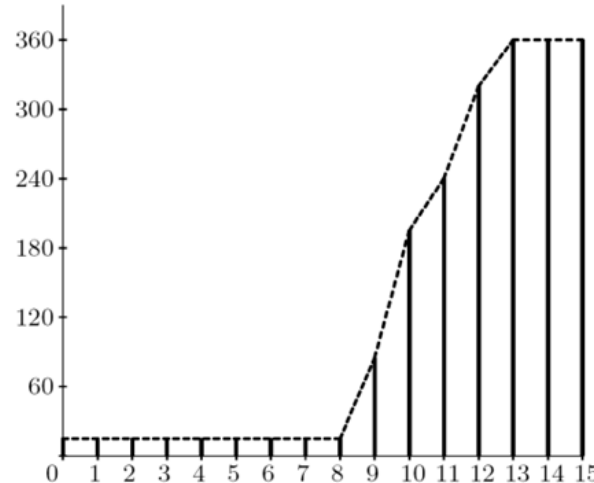


圖 4.22 累加的直方圖

```
>> en=imread('engineer.tif');  
>> e=imdivide(en,4);  
>> imshow(e),figure,imhist(e),axis tight
```

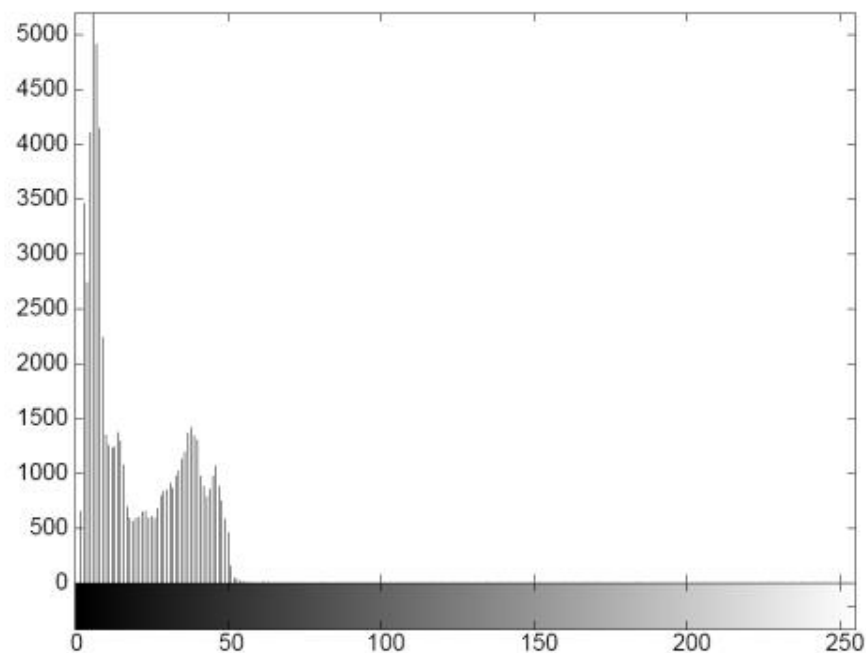
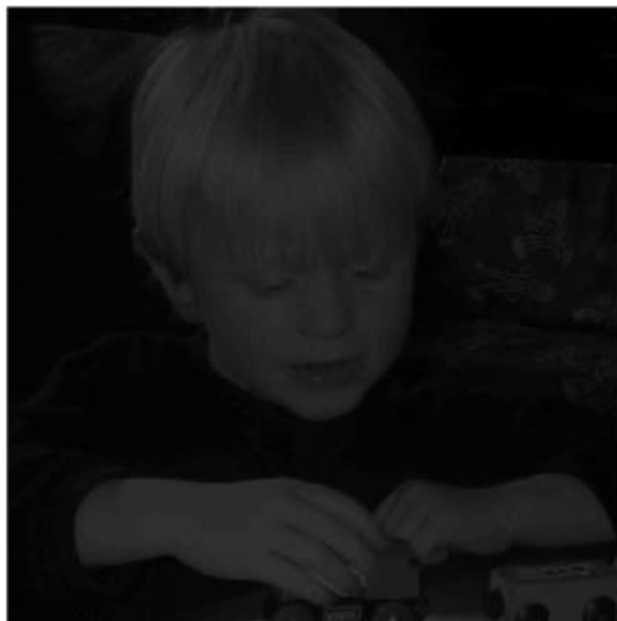


圖 4.20 engineer.tif 的暗色版本及其直方圖

```
>> eh=histeq(e);  
>> imshow(eh),figure,imhist(eh),axis tight
```

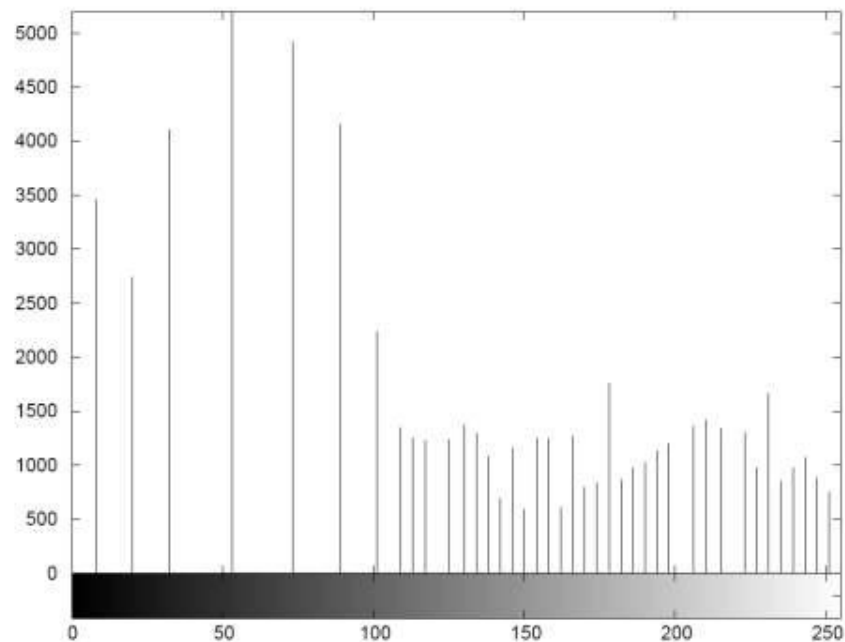
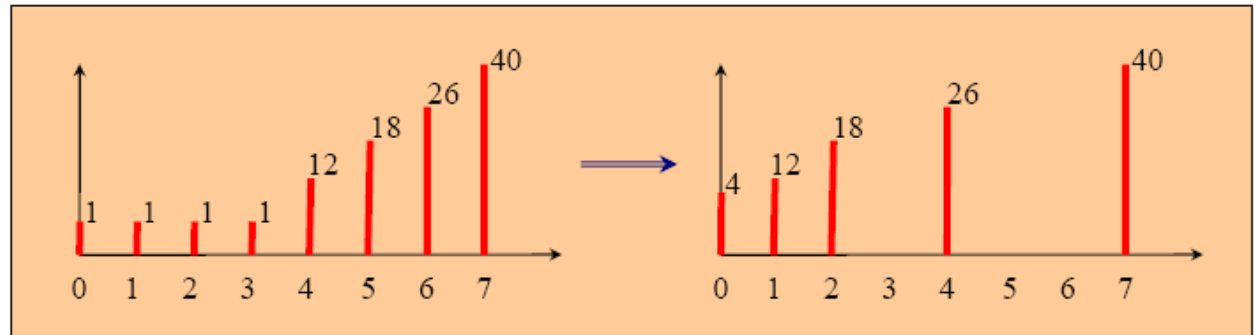


圖 4.21 圖 4.20 等化後的影像及其直方圖



練習題：

Example 1: 像素總數100, 灰階值0~7

輸入灰階 r	像素 個數	$p(r)$	cdf	輸出灰階 s
0	1	0.01	0.01	$7 \times 0.01 = 0$
1	1	0.01	0.02	$7 \times 0.02 = 0$
2	1	0.01	0.03	$7 \times 0.03 = 0$
3	1	0.01	0.04	$7 \times 0.04 = 0$
4	12	0.12	0.16	$7 \times 0.16 = 1$
5	18	0.18	0.34	$7 \times 0.34 = 2$
6	26	0.26	0.60	$7 \times 0.60 = 4$
7	40	0.40	1.00	$7 \times 1.00 = 7$

灰階 s	像素個數
0	4
1	12
2	18
3	0
4	26
5	0
6	0
7	40

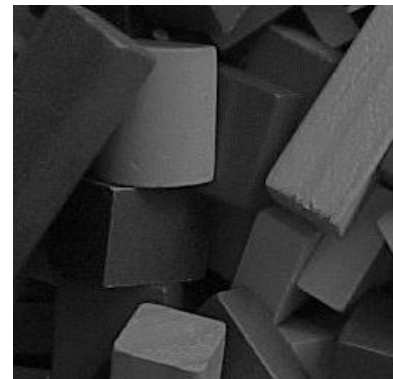
4.4 查詢表格

- 使用 **查詢表格** (lookup table, **LUT**) 執行點運算會十分有效率

□ Example:

索引值：	0	1	2	3	4	5	...	250	251	252	253	254	255
LUT：	0	0	1	1	2	2	...	125	125	126	126	127	127

```
>> T=uint8(floor([0:255]/2))  
>> b=imread('blocks.tif');  
>> b2=T(b+1);  
>> imshow(b2)
```



■ Example: 執行對比擴展函數

$$y = \frac{64}{96}x$$

$$y = \frac{192 - 64}{160 - 96}(x - 96) + 64$$

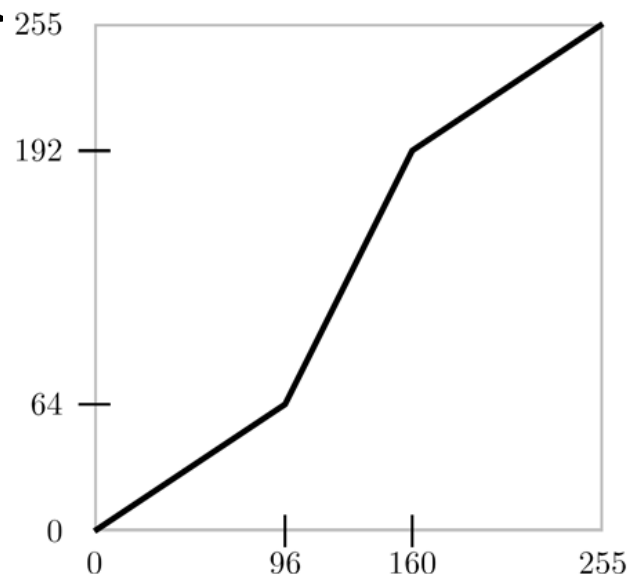
$$y = \frac{255 - 192}{255 - 160}(x - 160) + 192$$



$$y = 0.6667x$$

$$y = 2x - 128$$

$$y = 0.6632x + 85.8947$$



```
>> t1=0.667*[0:96];  
>> t2=2*[97:160]-128;  
>> t3=0.6632*[161:255]+85.8947;  
>> T=uint8(floor([t1 t2 t3]));
```

```
>>b2=T(b+1);
```

```
>> imshow(b2)
```

練習

- 在M-file中，撰寫以下運算對應的LUT並將影像的原圖、結果圖以及對應的直方圖顯示出來。(灰階影像任意選擇)
 - 灰階值乘以2
 - 影像補色
 - 灰階值加150