影像處理
Chapter 02 數位影像基礎

Chien-Chang Chen
Tamkang University
Department of Computer Science and Information Engineering

人類視覺系統

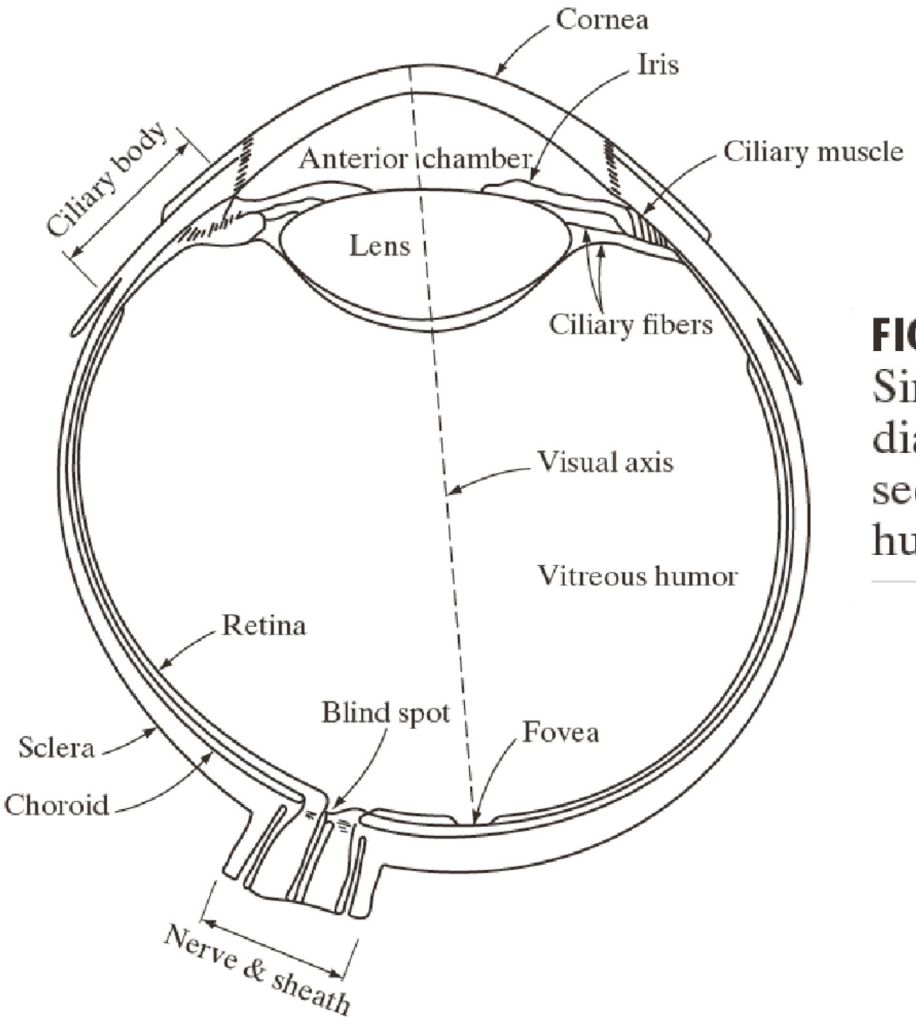


FIGURE 2.1 Simplified diagram of a cross section of the human eye.

- The eye works like a camera, with the lens focusing an image onto the retina (upside-down and left-right reversed).
- The retina consists of an array of rods and three kinds of cones.
- The rods（柱狀體）come into play when light levels are low and produce a image in shades of gray ("all cats are gray at night!").
- For higher light levels, the cones（錐狀體）each produce a signal. Be- cause of their differing pigments, the three kinds of cones are most sensitive to red (R), green (G), and blue (B) light.
- It seems likely that the brain makes use of differences R-G, G-B, and B-R, as well as combining all of R, G, and B into a high-light-level achromatic channel.

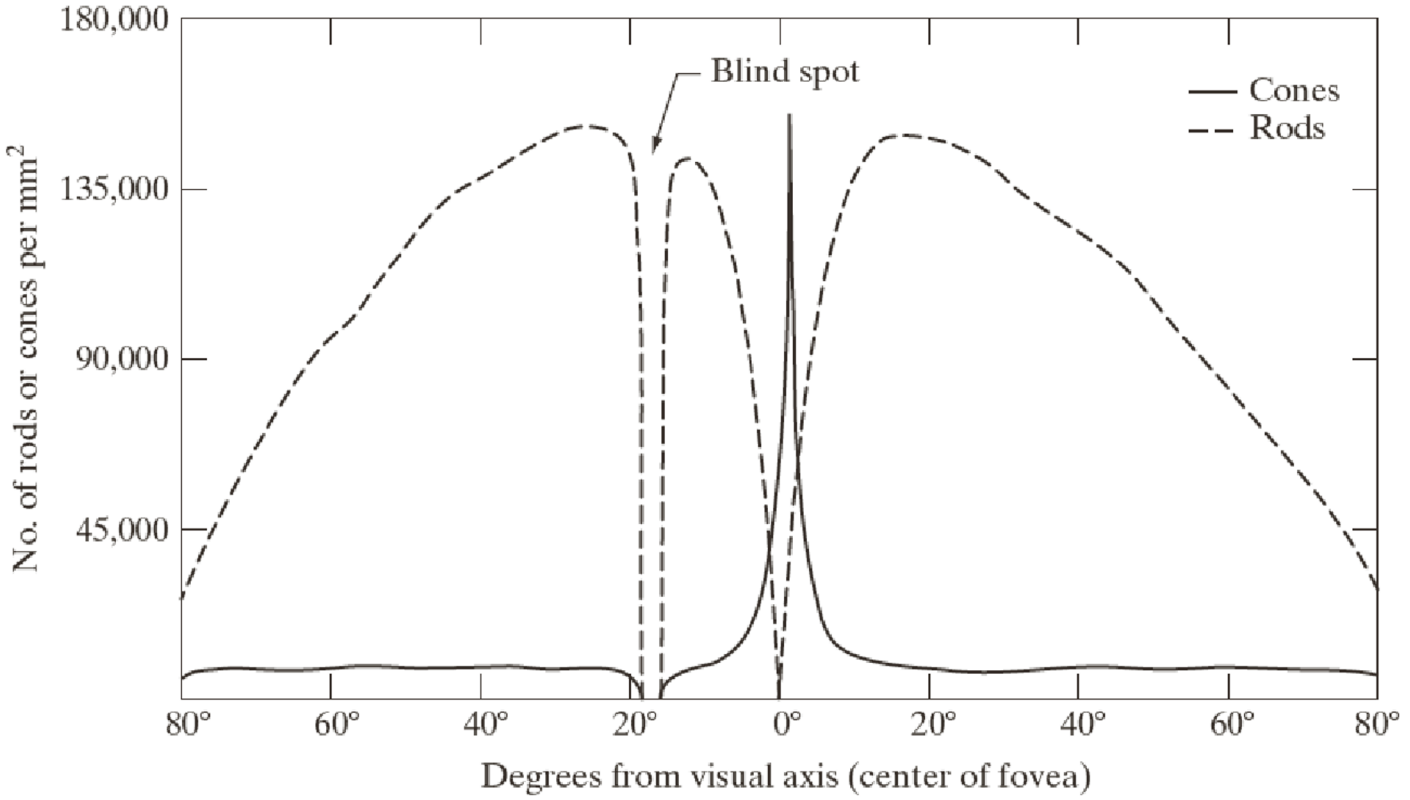


FIGURE 2.2
Distribution of rods and cones in the retina.

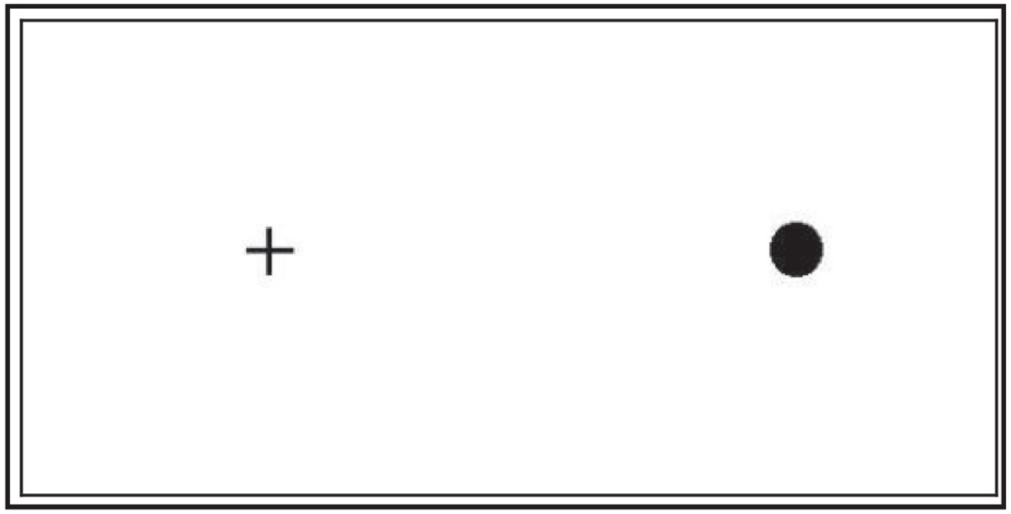
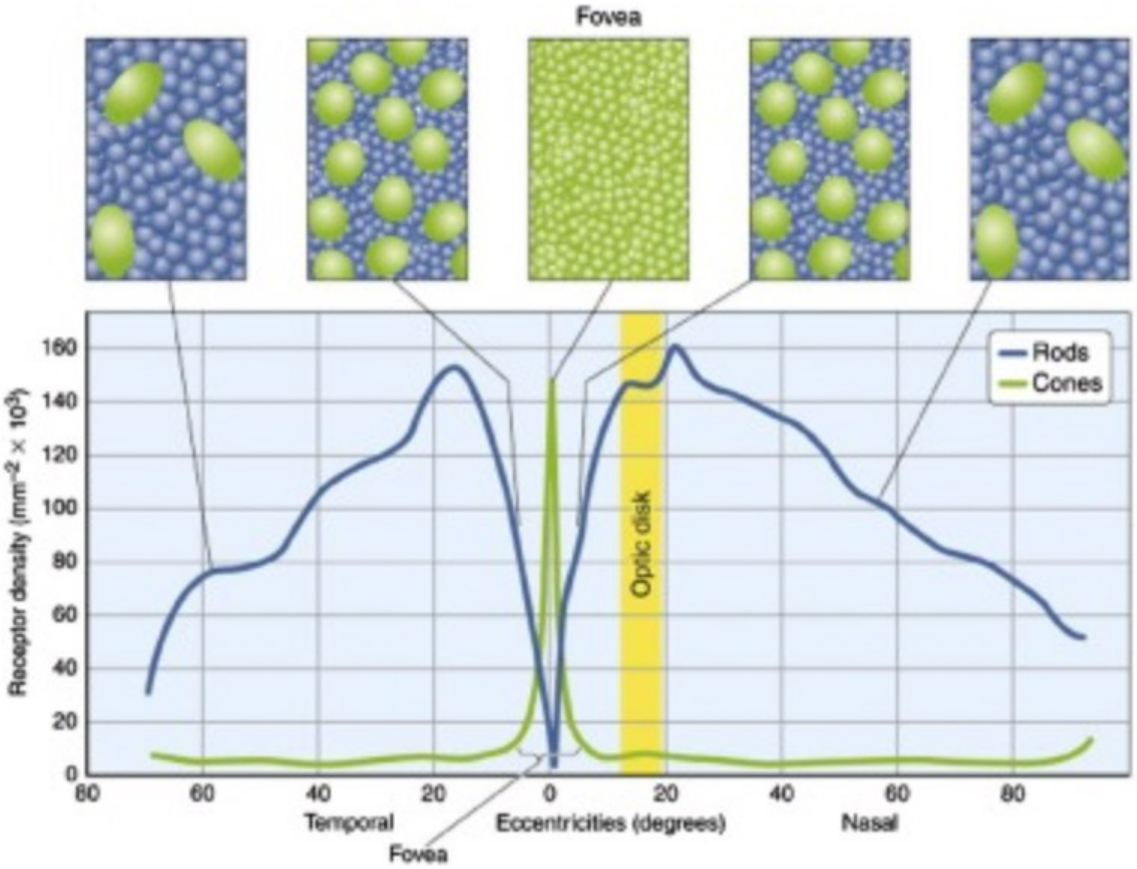


圖 3-3　人類眼睛的盲點測試圖

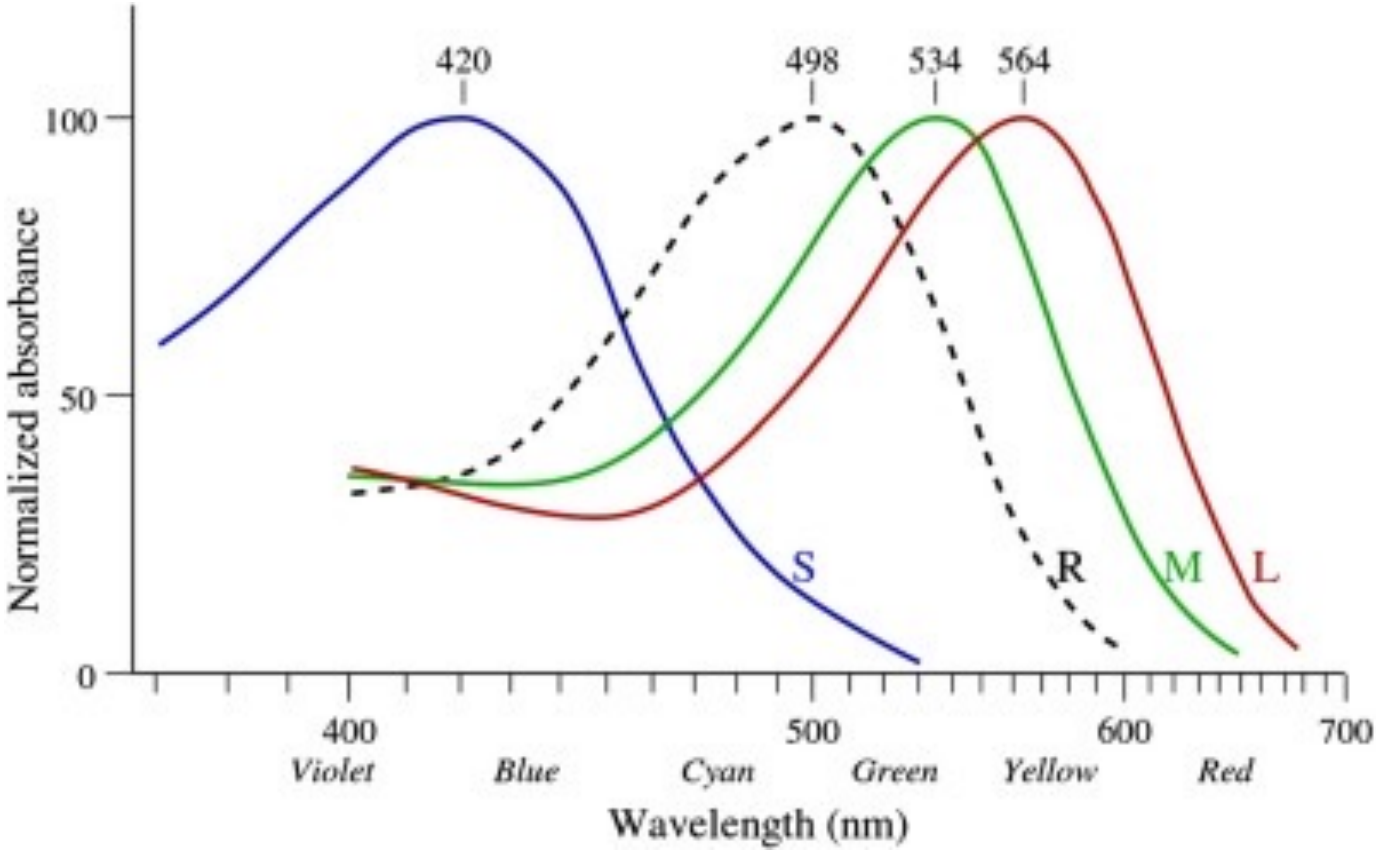


Download : Download full-size image

Figure 2.5. Photoreceptor distribution in the retina.

Reproduced with permission from: Mustafia et al. [12].

https://www.sciencedirect.com/science/article/pii/B9780124059061000027



https://www.sciencedirect.com/science/article/pii/B9780124059061000027

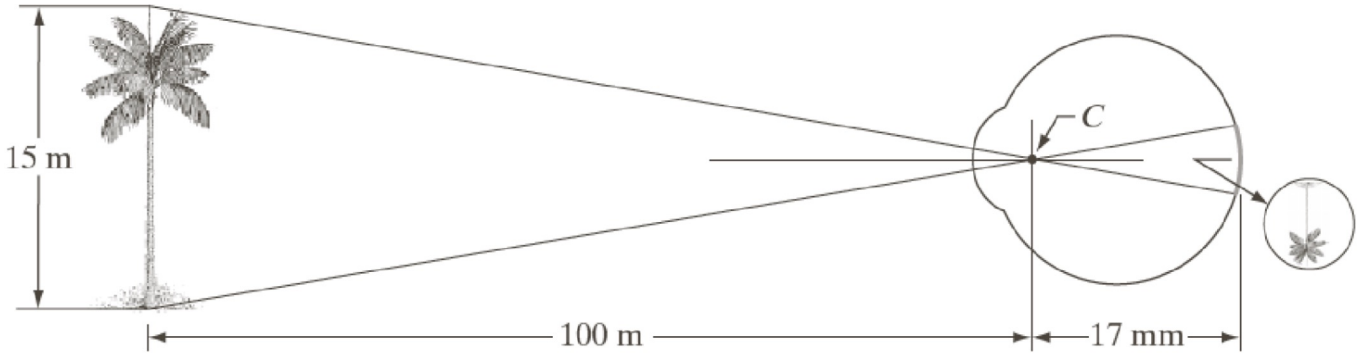
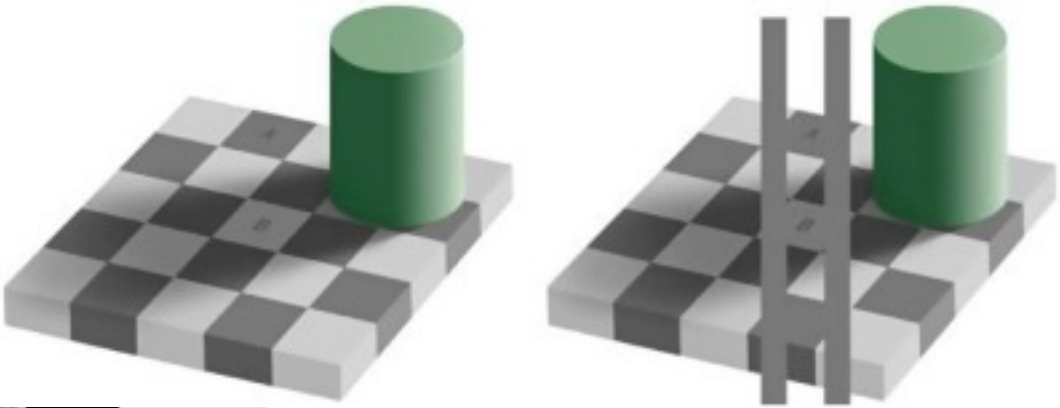


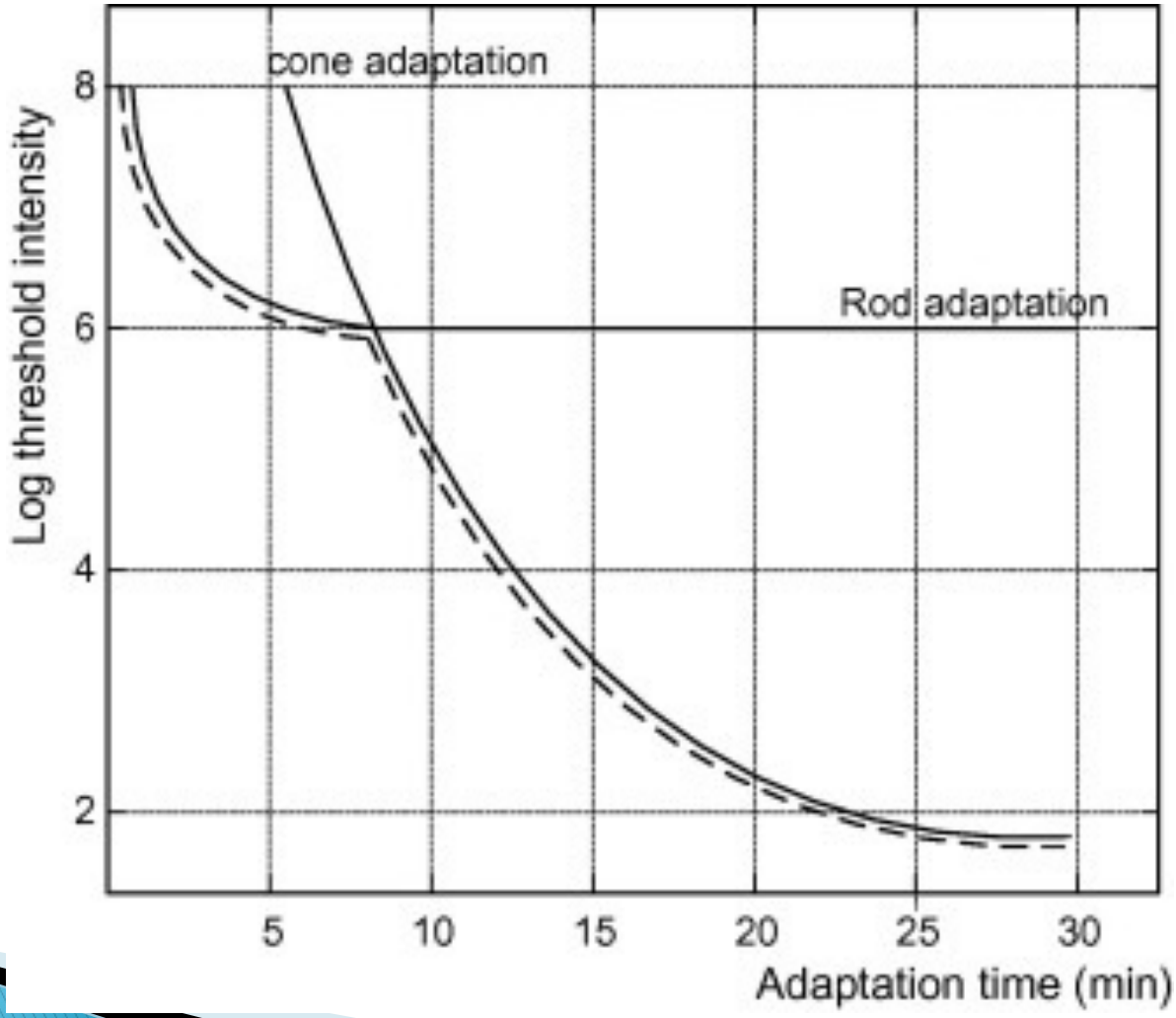
FIGURE 2.3 Graphical representation of the eye looking at a palm tree. Point *C* is the optical center of the lens.

- Mach band effect

- Adelson's grid



https://www.sciencedirect.com/sci…ticle/pii/B9780124059061000027#s0105



https://ars.els-cdn.com/content/ima[illegible]0-B9780124059061000027-f02-13-9780124059061.jpg

Increased immersion from color images

- on the right, shows a color image of Yellow Water in Kakadu, Australia and, on the left, a monochrome version of the same image. Most would agree that the color image is easier to interpret, providing improved differentiation between natural objects and features in the scene and a better sense of depth and engagement.

https://www.sciencedirect.com/[illegible]/article/pii/B9780124059061000027#s0105

影像擷取

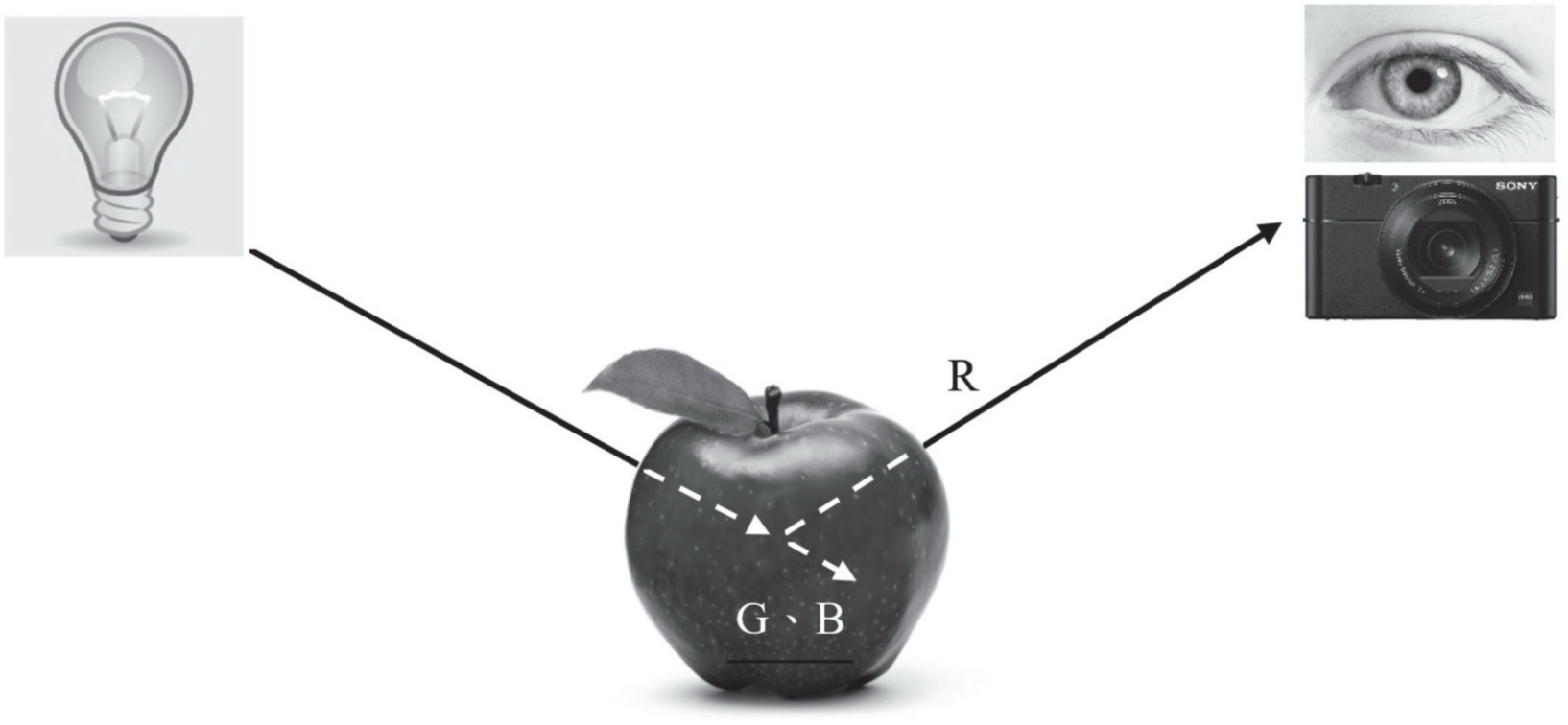
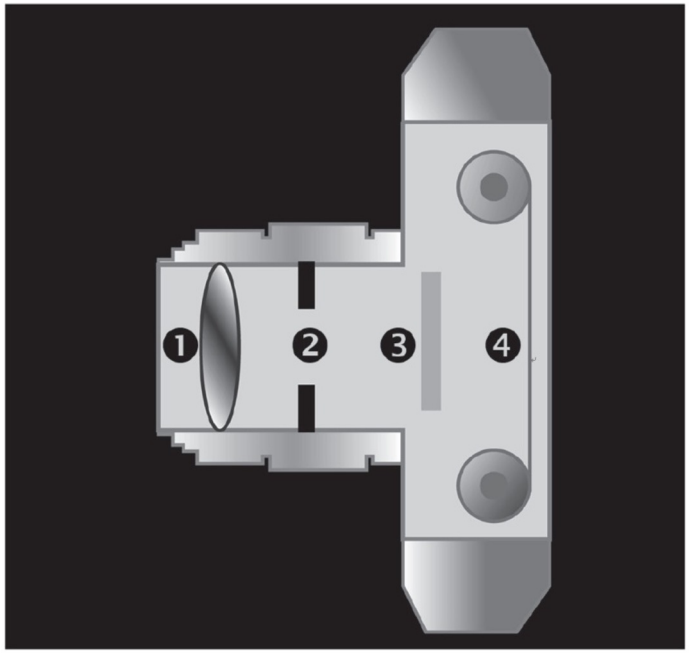


圖 3-4　影像擷取過程



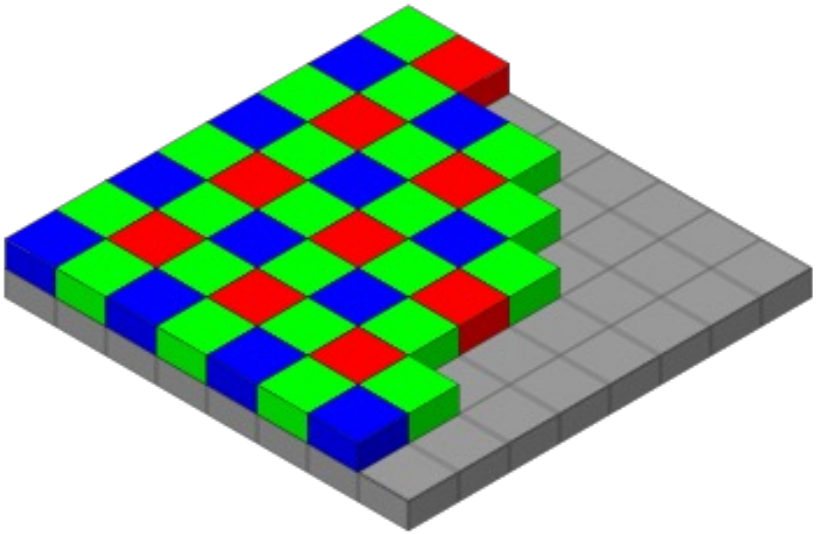
1. 鏡片 (Lens)　　2. 光圈 (Aperture)
3. 快門 (Shutter)　　4. CCD / CMOS

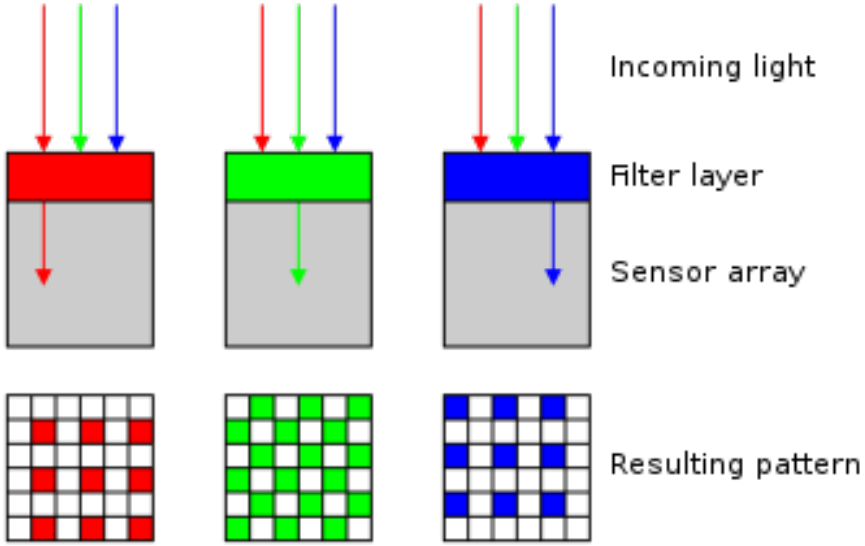
圖 3-5　相機的剖面結構圖

R	G	R	G	R	G	
G	B	G	B	G	B	
R	G	R	G			
G	B	G	B			
R	G					
G	B					

圖 3-6　貝爾濾波器

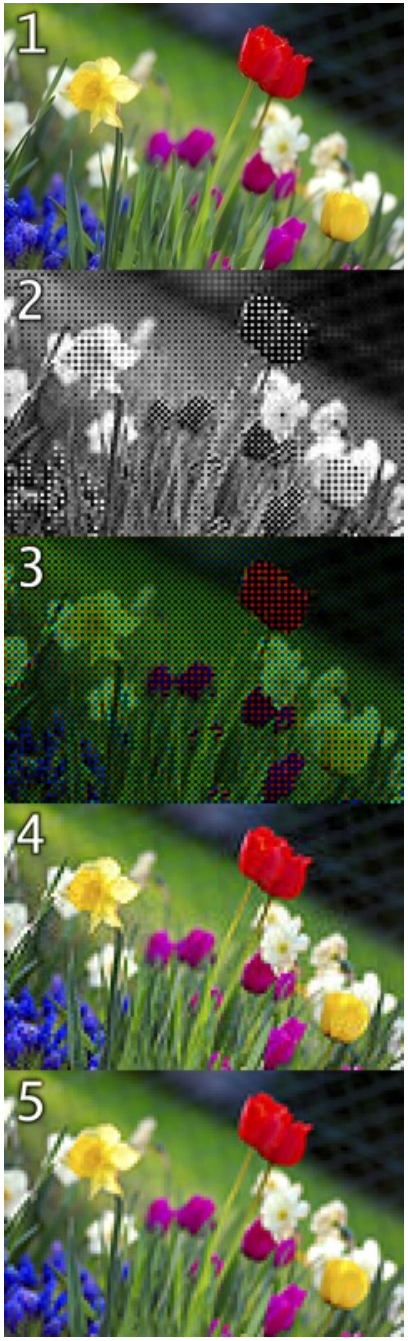
- 貝爾濾波器採用4個像素的強度值，構成一個RGB色彩像素





https://en.wikipedia.org/wiki/Bayer_filter

- 1. Original scene
- 2. Output of a 120×80-pixel sensor with a Bayer filter
- 3. Output color-coded with Bayer filter colors
- 4. Reconstructed image after interpolating missing color information
- F5. ull RGB version at 120×80-pixels for comparison (e.g. as a film scan, Foveon or pixel shift image might appear)



https://en.wikipedia.org/wiki/Bayer_filter

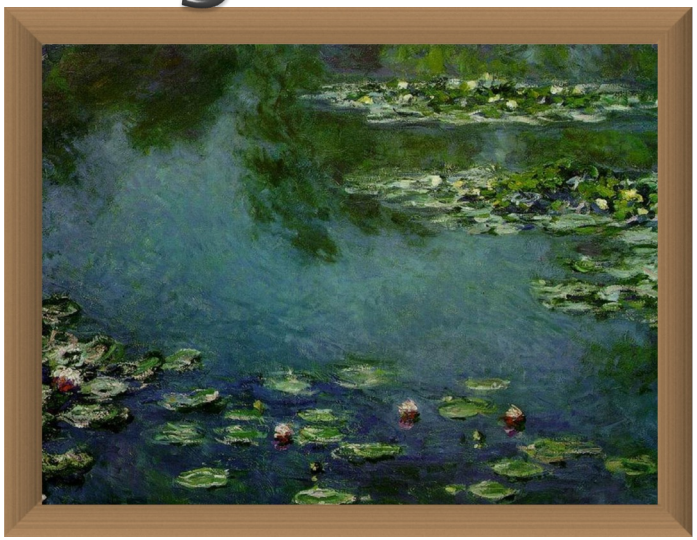
Image Formation Model

- Function $f(x, y)$ is characterized by two components
  - *illumination* $i(x, y)$打光函數: the amount of source illumination incident on the scene being viewed
  - *reflectance* $r(x, y)$反射函數: the amount of illumination reflected by the objects in the scene
- The two functions combine as a product to form

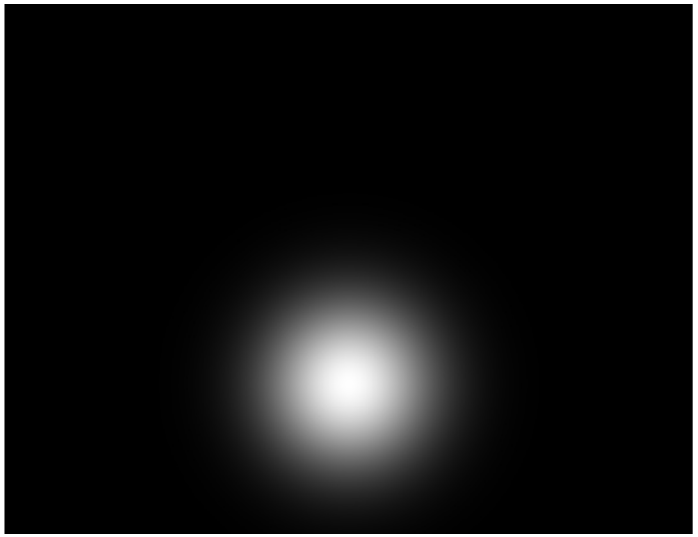
$$f(x, y) = i(x, y) \cdot r(x, y)$$

$$0 \leq i(x, y) < \infty \quad \text{and} \quad 0 \leq r(x, y) \leq 1$$

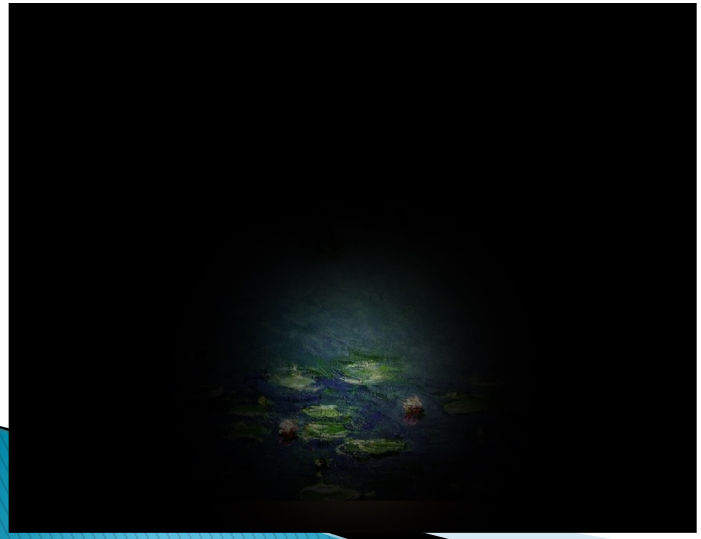
Image Formation Model



illumination $i(x, y)$



reflectance $r(x, y)$



打光函數(二維高斯函數)

$$r(x,y) = e^{-\frac{(x-x_0)^2+(y-y_0)^2}{2\sigma^2}}$$

$$f(x, y) = i(x, y) \cdot r(x, y)$$

$$f(x, y) = i(x, y) \cdot r(x, y)$$

$$0 \le i(x, y) < \infty \quad \text{and} \quad 0 \le r(x, y) \le 1$$

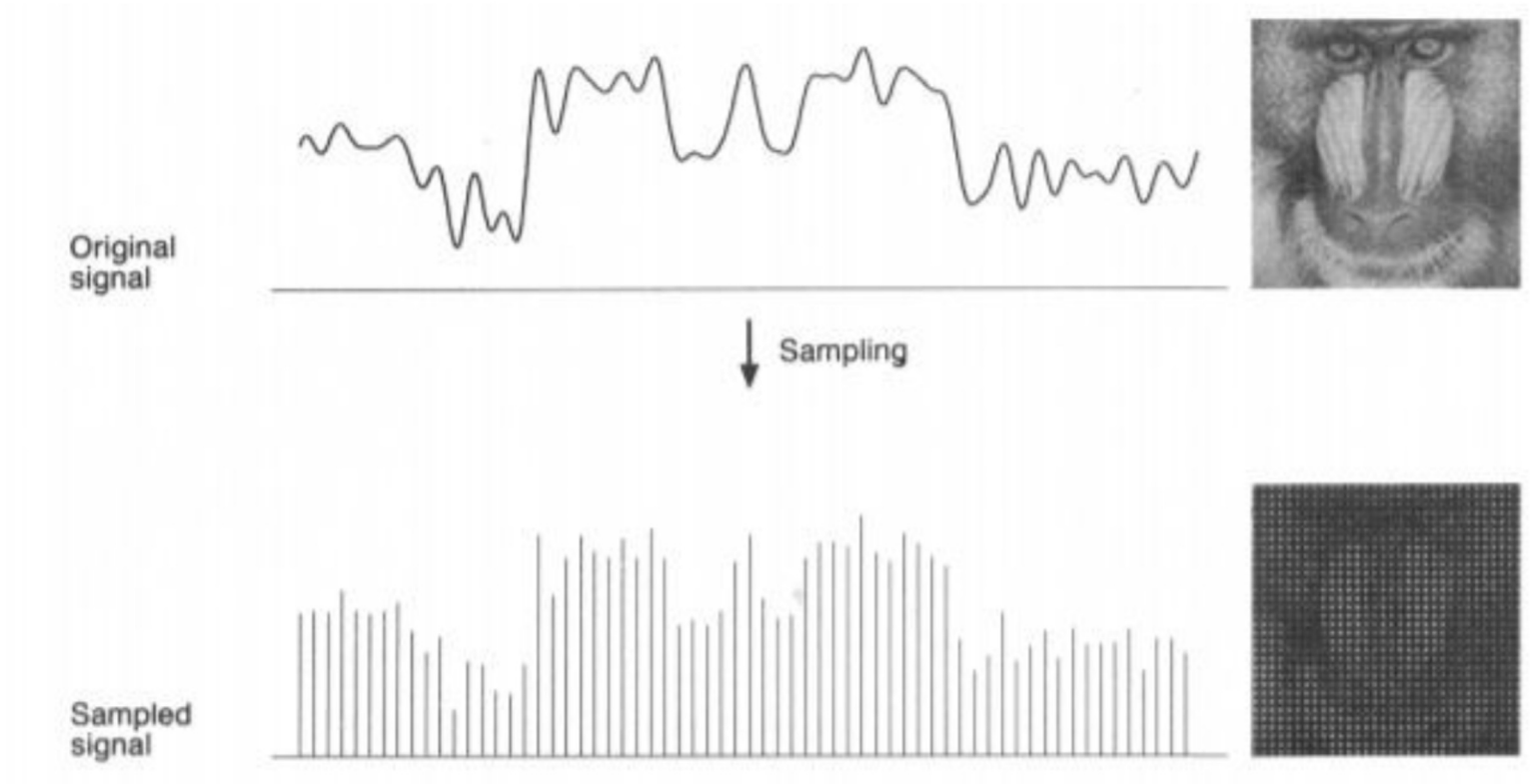
Check IP02.ipynb

- 2.1 Image Formation Model
- 2.2 Practice: Modify python program to acquire the image formation result

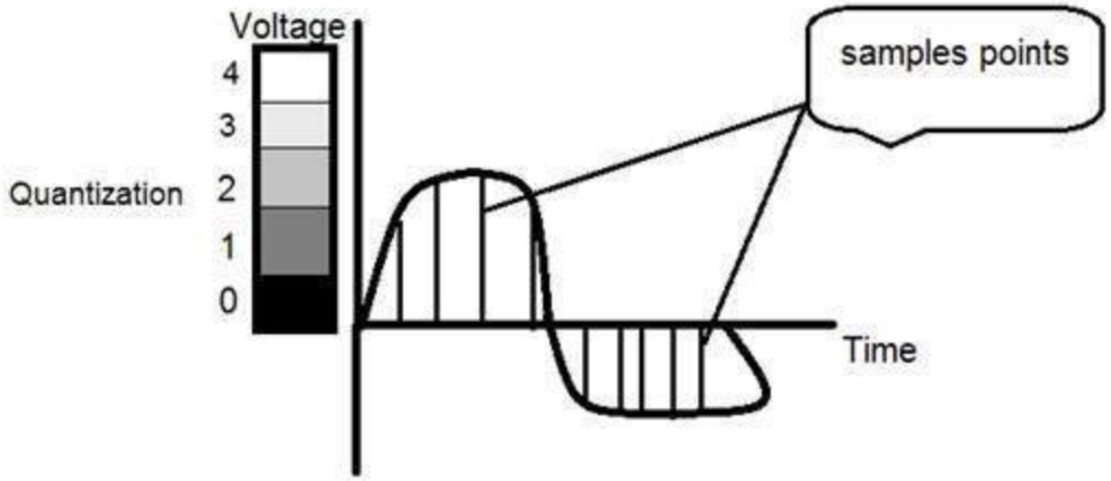
數位影像的取樣與量化

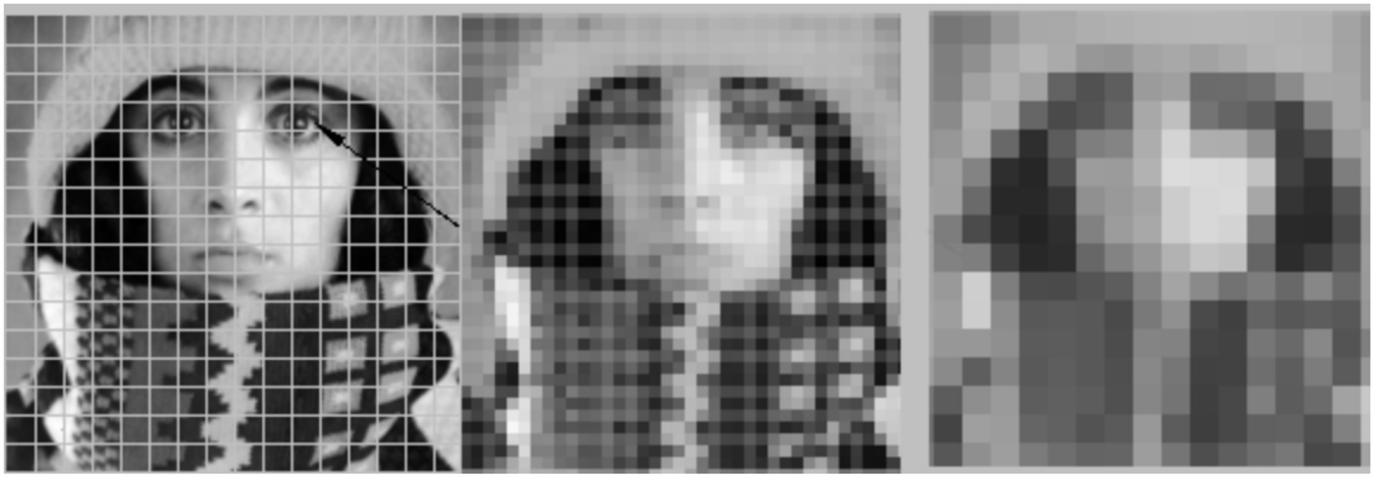
- 取樣(Sampling)：影像空間座標的數位化
- 量化(Quantization)：影像像素強度的數位化

Image Sampling and Quantization

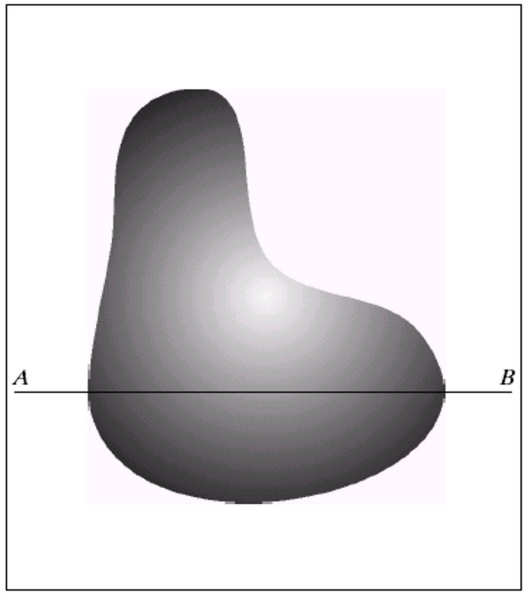


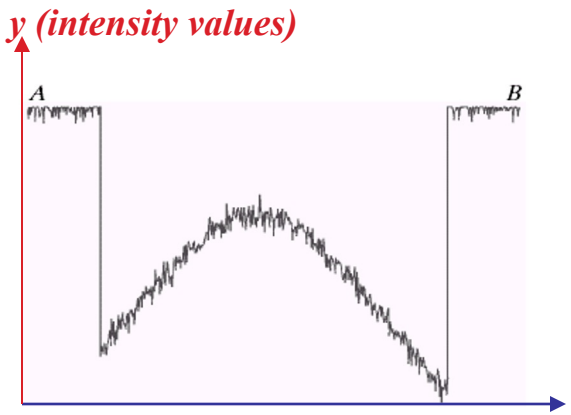
https://wimarshikathamali1993.medium.com/sampling-quantization-in-digital-image-processing-8c4490357039

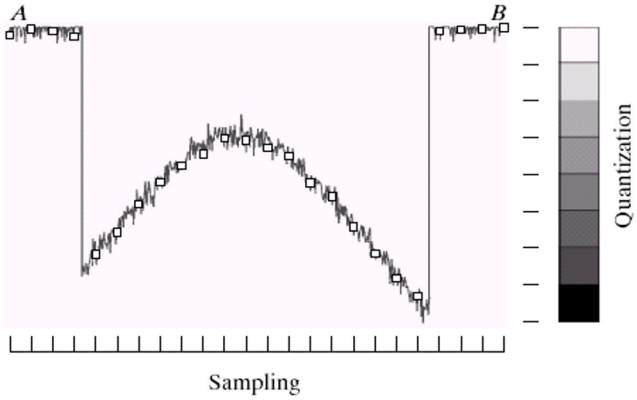


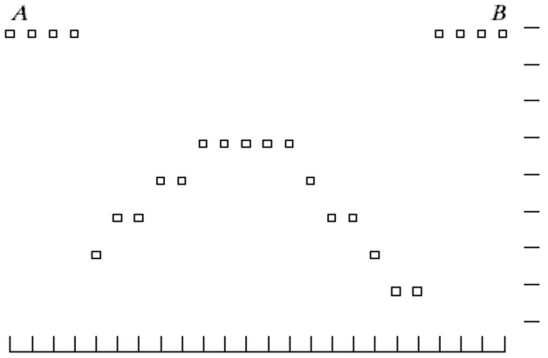


https://wimarshikathamali1993.medium.com/sampling-quantization-in-digital-image-processing-8c4490357039



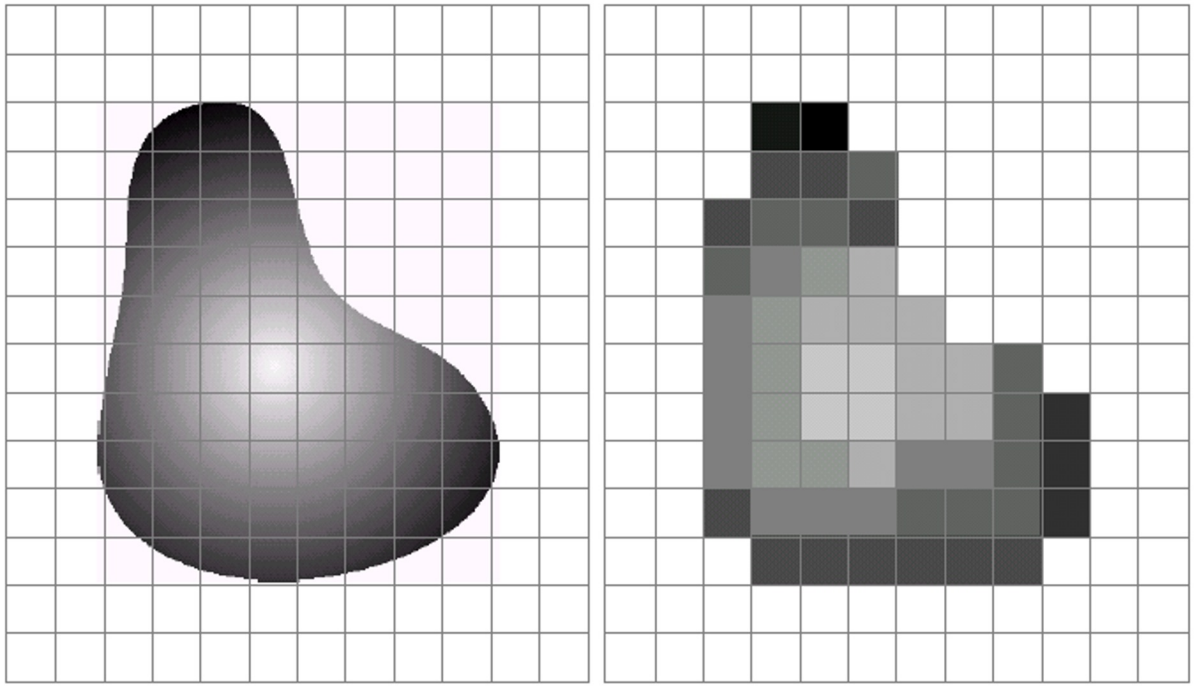






a b
c d

Generating a digital image. (a) Continuous image. (b) A scaling line from A to B in the continuous image, used to illustrate the concepts of sampling and quantization. (c) sampling and quantization. (d) Digital scan line.



a b

FIGURE 2.17 (a) Continuos image projected onto a sensor array. (b) Result of image sampling and quantization.

a b

(a) Continuous image projected onto a sensor array. (b) Result of image sampling and quantization.

Sampling

1024

512

256

128

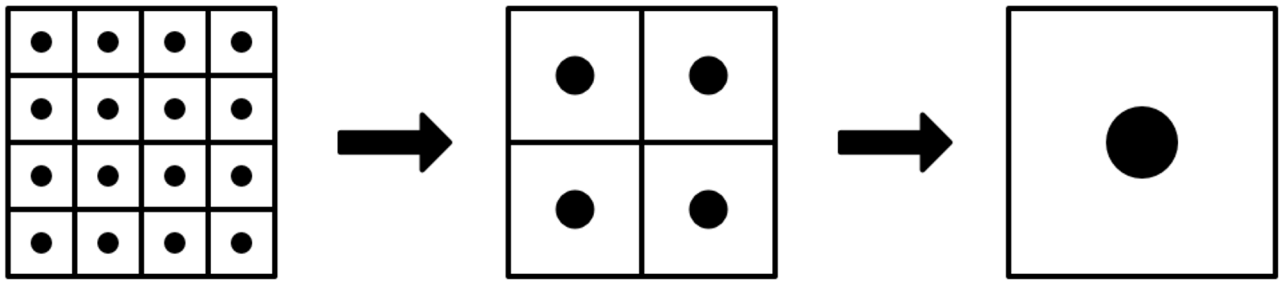
64

32

https://sisu.ut.ee/sites/default/files/imageprocessing/files/vid_lecture02fin.ppt

Image Downsampling

- 整數的downsampling

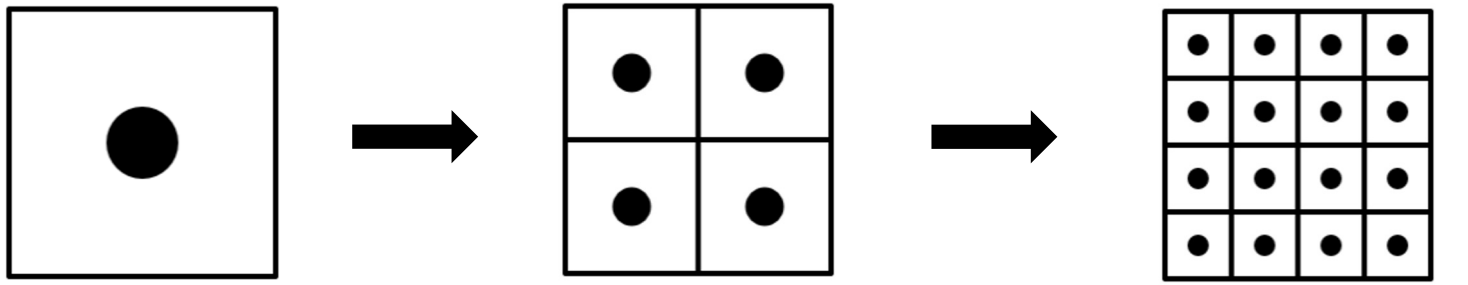


Check IP02.ipynb

- 2.3 Image Downsampling
- 2.4 Image downsampleing by the average of each nxn image block
  - n : sampling_rate
  - (1) use for loop to calculate block average
  - (2) use np.mean to calculate block average

Image Upsampling

- 整數的 upsampling



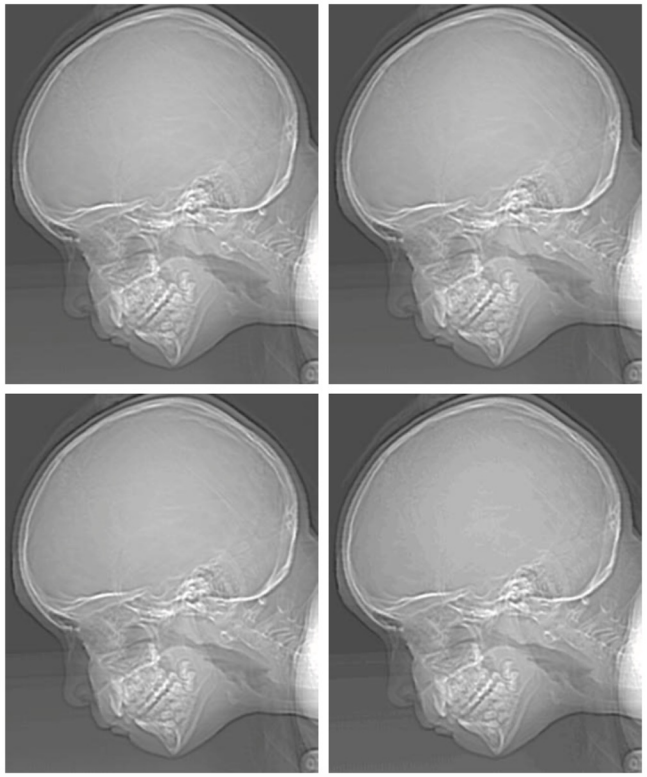
Check IP02.ipynb

- 2.5 Image Upsampling
  - integer extension like slide p.27

Image Quantization

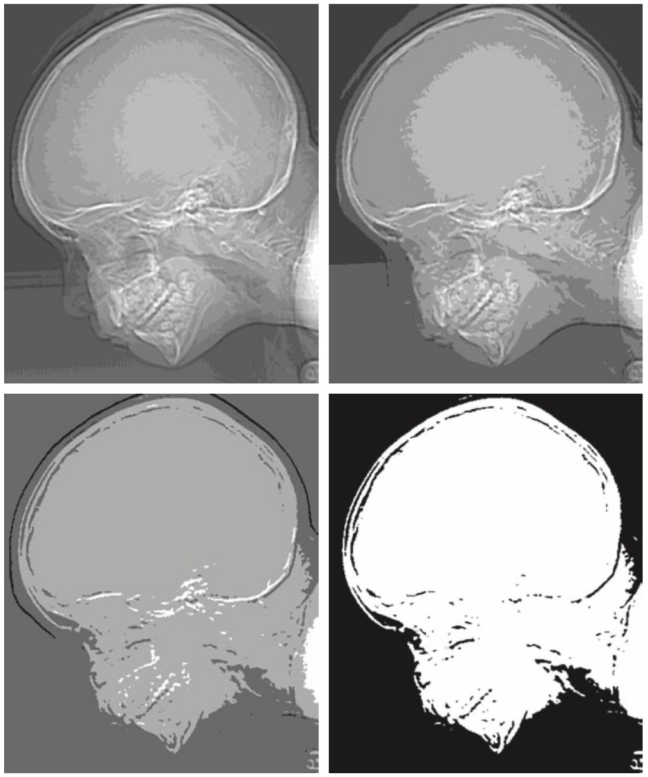
a b
c d

FIGURE 2.24
(a) 774 × 640, 256-level image. (b)-(d) Image displayed in 128, 64, and 32 intensity levels, while keeping the spatial resolution constant. (Original image courtesy of the Dr. David R. Pickens, Department of Radiology & Radiological Sciences, Vanderbilt University Medical Center.)



e f
g h

FIGURE 2.24
(*Continued*)
(e)-(h) Image displayed in 16, 8, 4, and 2 intensity levels.



Check IP02.ipynb

- 2.6 Image Quantization
- 2.7 Image Quantization version 2

Image Addition & Subtraction

- Image Addition

$$f(u) = \begin{cases} u + c, & if\ u + c < 255 \\ 255, & else. \end{cases}$$

- Image Subtraction

$$f(u) = \begin{cases} u - c, & if\ u - c > 0 \\ 0, & else. \end{cases}$$

Check IP02.ipynb

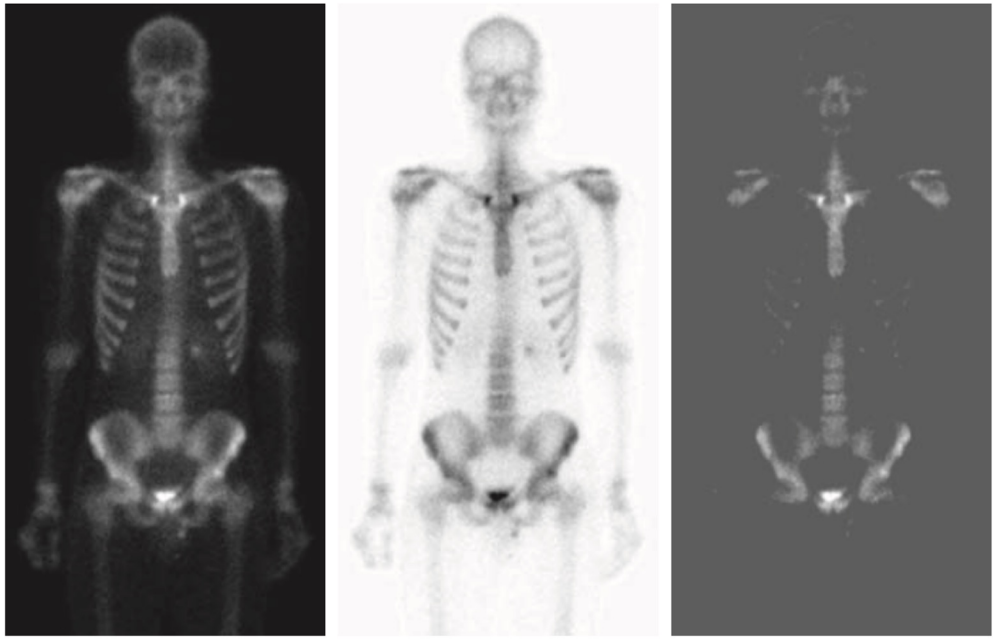
- 2.8 Image Addition & Subtraction(gray)

Complement 負片

- Complement of an image $f(x,y)$
- $c(x,y) = 255 - f(x,y)$
- Union of two images $f(x,y)$ and $g(x,y)$
- $A \cup B = \{\max(a,b) \,|\, a \in f, b \in g\}$

a b c

FIGURE 2.36
Set operations involving grayscale images. (a) Original image. (b) Image negative obtained using grayscale set complementation. (c) The union of image (a) and a constant image. (Original image courtesy of G.E. Medical Systems.)

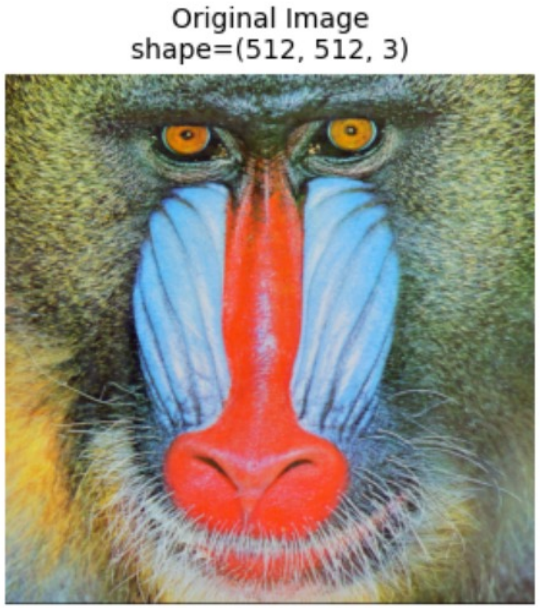


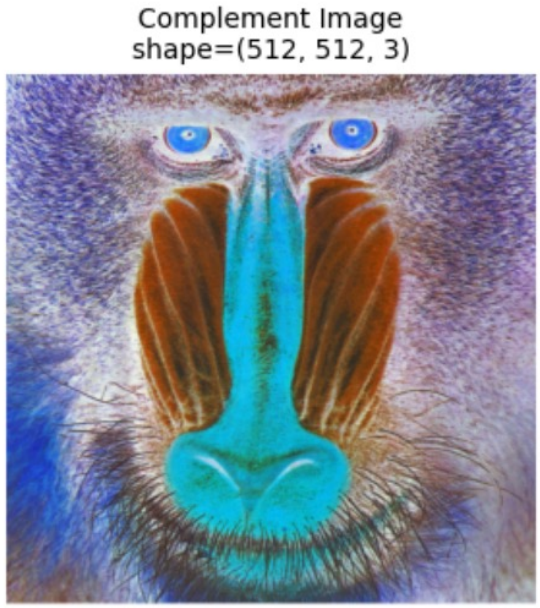
Check IP02.ipynb

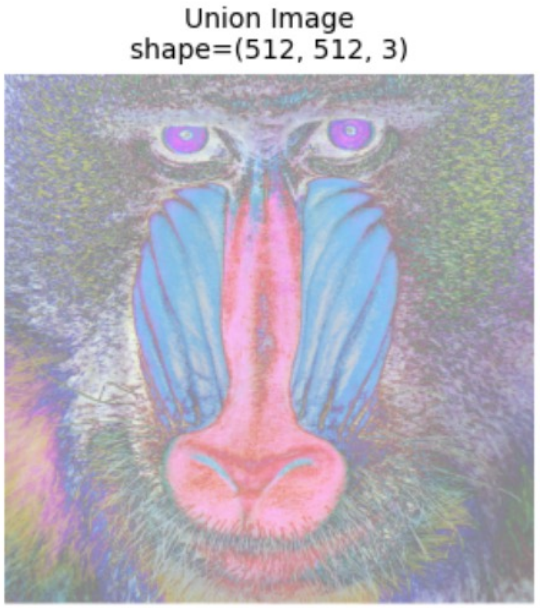
- 2.9 Practice Complement(gray)
- Finish by implementing two functions
  - def image_complement(f)
  - def image_union(f, g)

Check IP02.ipynb

- 2.10 Practice Complement(color)
- Finish by implementing two functions
  - def image_complement(f)
  - def image_union(f, g)





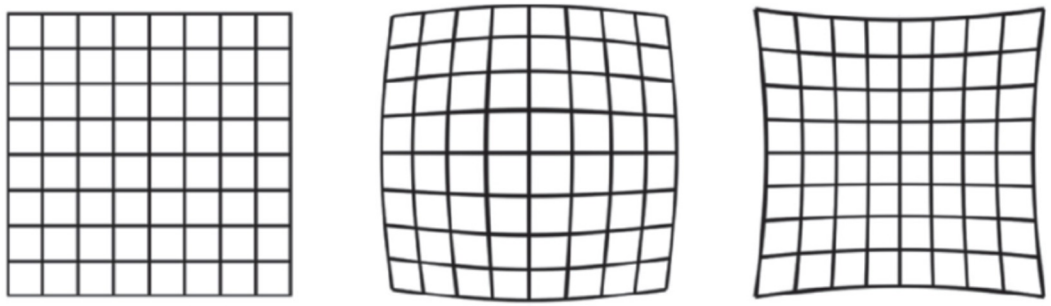


Geometric Transformations

- modify the spatial arrangement of pixels in an image
- Geometric transformations of digital images consist of two basic operations:
  - Spatial transformation of coordinates
  - Intensity interpolation that assigns intensity values to the spatially transformed pixels

(a)

(b)



(c)

- transformation of coordinates may be expressed as

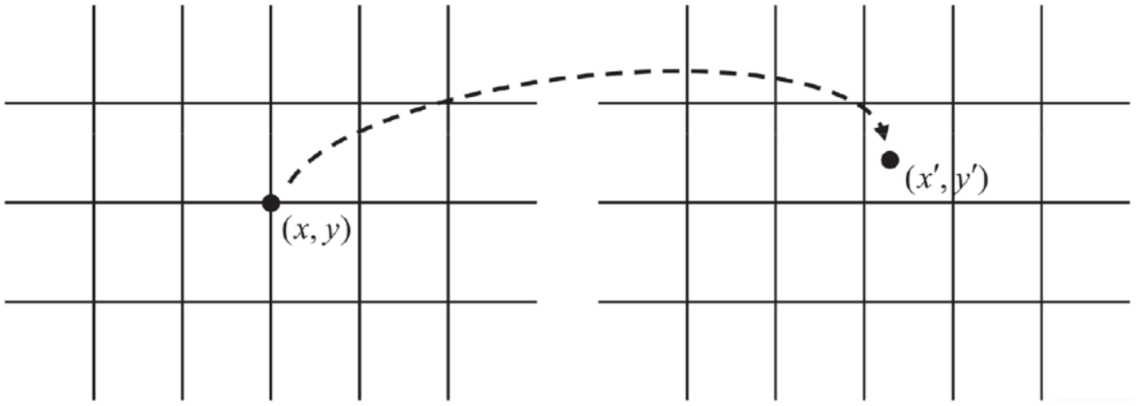
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = T \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

(x, y): pixel coordinates in the original image

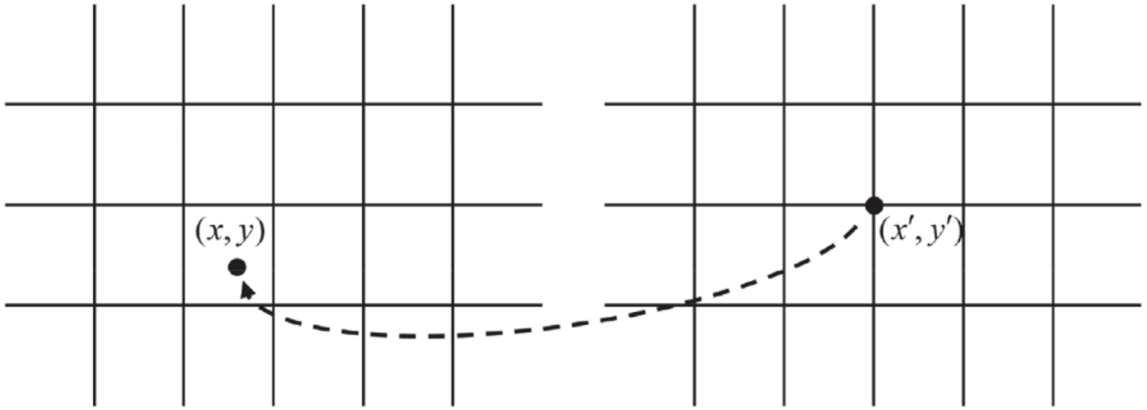
(x', y') : corresponding pixel coordinates of the transformed image

正向映射(Forward Mapping) : $(x', y') = T\{(x, y)\}$

反向映射(Inverse Mapping) : $(x, y) = T^{-1}\{(x', y')\}$

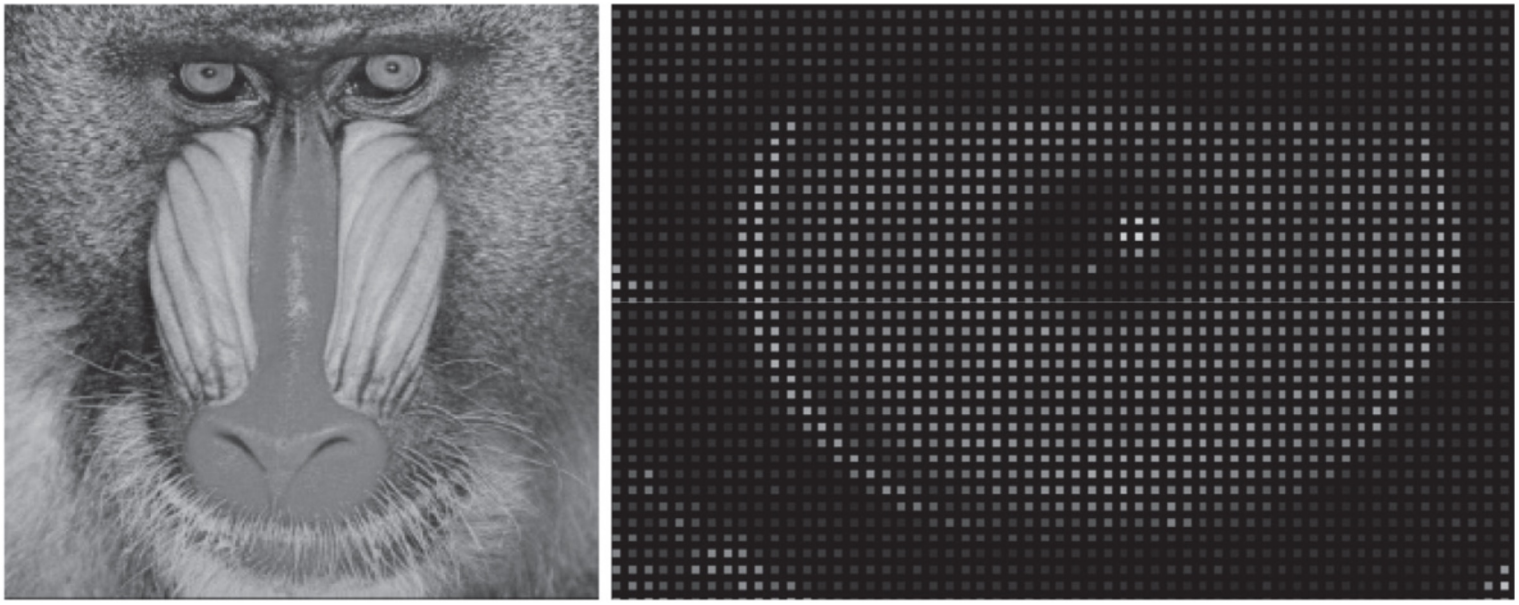


正向映射 (Forward Mapping)



反向映射 (Inverse Mapping)

- 正向映射(Forward Mapping)可能會產生的問題



原始影像　　放大兩倍 (局部)

圖 4-3　使用正向映射產生的破洞問題

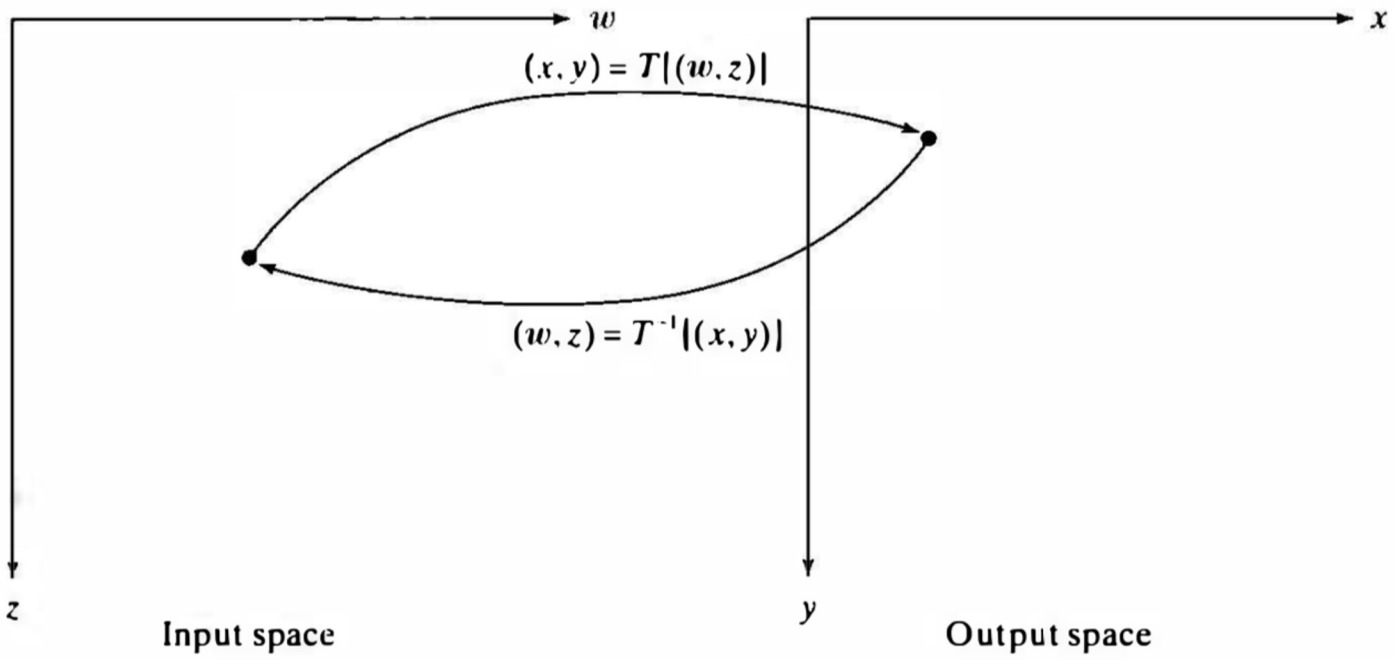


FIGURE 6.1 Forward and inverse transformation of a point for $T[(w, z)] = (w/2, z/2)$.

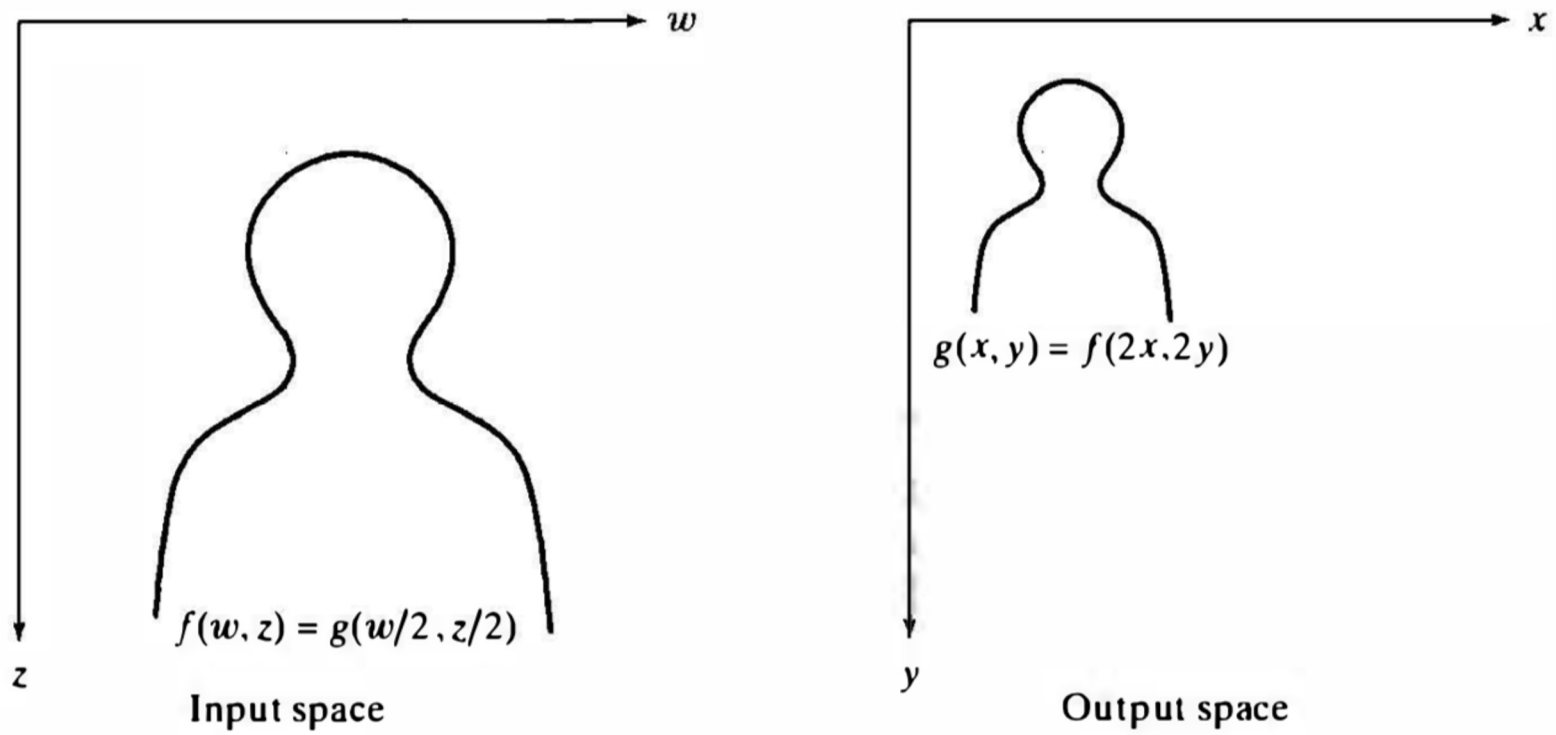


FIGURE 6.2 Forward and inverse transformation of a simple image for the transformation $T[(w, z)] = (w/2, z/2)$.

影像內插

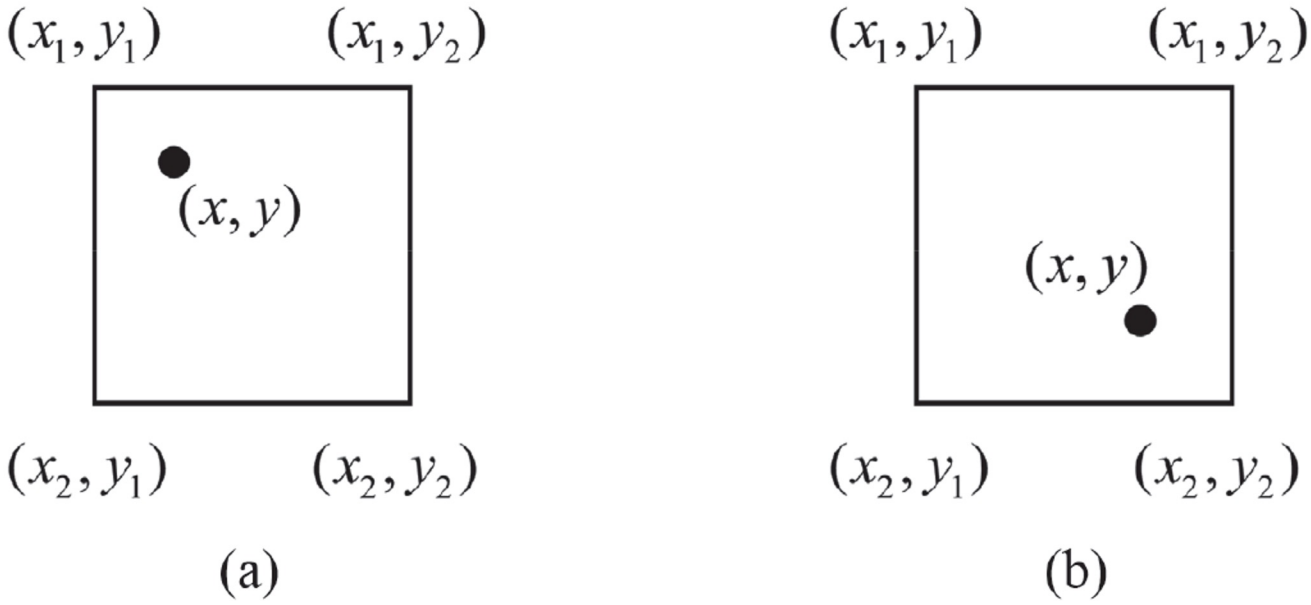
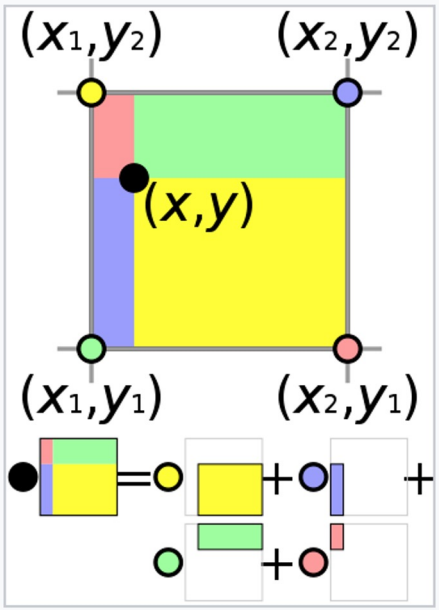


圖 4-4　最近鄰內插法

雙線性內插法(Bilinear Interpolation)



A geometric visualisation of bilinear interpolation. The product of the value at the desired point (black) and the entire area is equal to the sum of the products of the value at each corner and the partial area diagonally opposite the corner (corresponding colours).

from Wikipedia

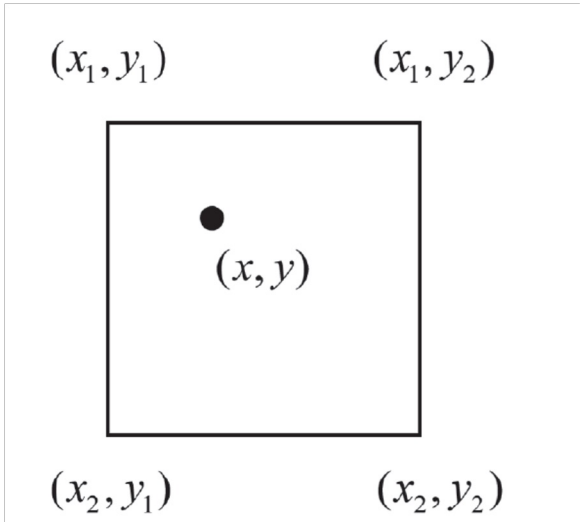


圖 4-5　雙線性內插法

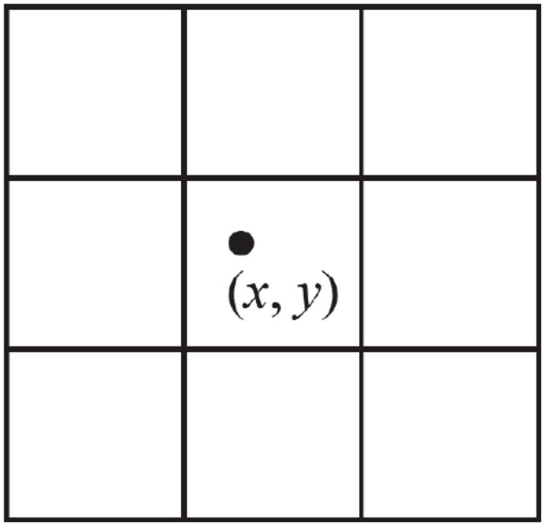


圖 4-6　雙立方內插法

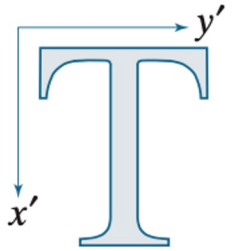
計算量大
保留更精確的影像細節
即時影像處理應用中不實用

Image Scaling 影像縮放

Scaling/Reflection
(For reflection, set one scaling factor to –1 and the other to 0)

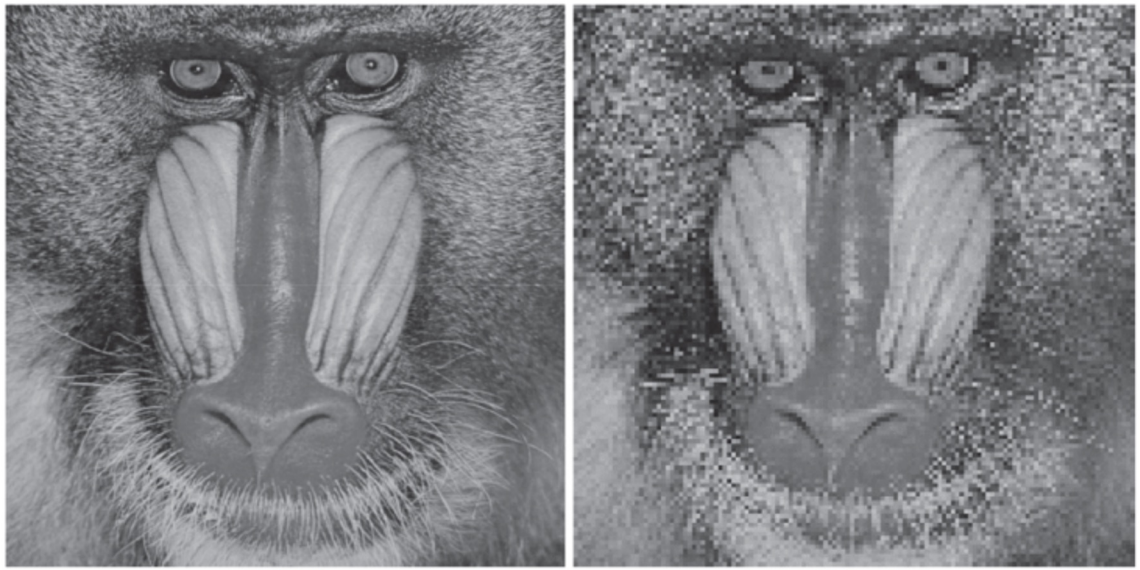
$$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$x' = c_x x$$
$$y' = c_y y$$



Check IP02.ipynb

- 2.11 Image scaling



原始影像　　最近鄰內插法

雙線性內插法　　雙立方內插法

圖 4-8　使用不同內插法的影像縮放比較

Which one is the best? worst?

Check IP02.ipynb

- 2.12 Image Rescaling

Images difference measure

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I(i,j) - I'(i,j)]^2$$

$$PSNR = 20 \cdot log_{10}(\frac{255}{\sqrt{MSE}})$$

$$PSNR = 10 \cdot log_{10}(\frac{255^2}{MSE})$$

Check IP02.ipynb

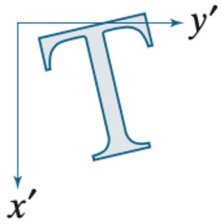
- 2.13 Homework PSNR
  - Design a PSNR function to measure the difference between (img,img1), (img,img2), and (img,img3)

Image Rotation 影像旋轉

Rotation (about the origin)

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$x' = x\cos\theta - y\sin\theta$$
$$y' = x\sin\theta + y\cos\theta$$



- 以上公式的旋轉中心為(0,0)，是影像的正中心
- 然而影像的(0,0)是左上角，因此，使用旋轉函數時，需要移至影像的正中心點

原始影像　　旋轉 $\theta = 30°$

圖 4-9　影像旋轉

Check IP02.ipynb

- 2.14 Image Rotation

練習

- 已知影像旋轉的空間轉換函數可以定義為:
  - $x' = xcos\theta - ysin\theta$
  - $y' = xsin\theta + ycos\theta$
- 請計算反轉換函數

Ans

原式可以表示成：

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

因此求反矩陣：

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}^{-1} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}$$

可得：

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix}$$

即：

$$x = x'\cos\theta + y'\sin\theta$$

$$y = -x'\sin\theta + y'\cos\theta$$

練習

- 已知影像旋轉的空間轉換函數可以定義為:
  - $x' = xcos\theta - ysin\theta$
  - $y' = xsin\theta + ycos\theta$
- 假設將旋轉中心移至影像中心(x_0, y_0)，請計算空間轉換函數及反轉換函數

Ans

$$x' = (x - x_0)\cos\theta - (y - y_0)\sin\theta$$
$$y' = (x - x_0)\sin\theta + (y - y_0)\cos\theta$$

或

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix}$$

因此可得：

$$\begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix}$$

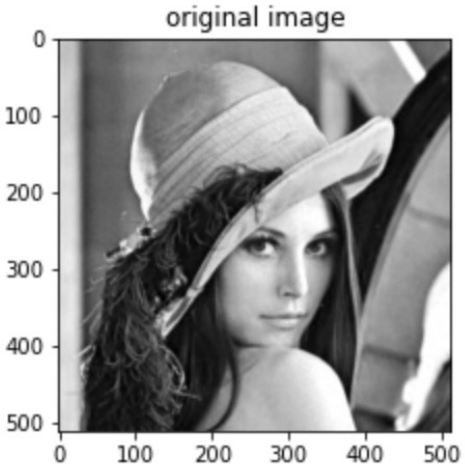
即：

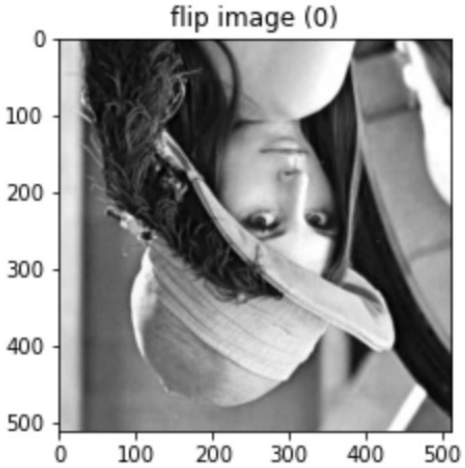
$$x = x'\cos\theta + y'\sin\theta + x_0$$
$$y = -x'\sin\theta + y'\cos\theta + y_0$$

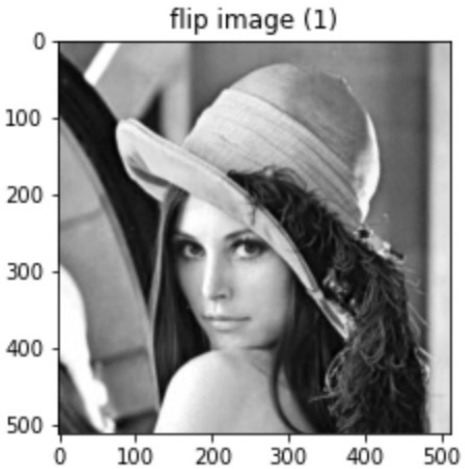
Check IP02.ipynb

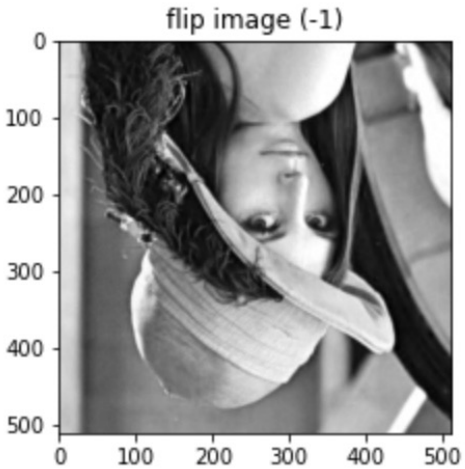
- 2.15 Image Flip

0:vertical flip
+:horizontal flip
-:dual direction









Check IP02.ipynb

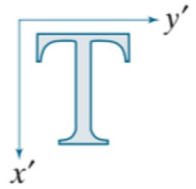
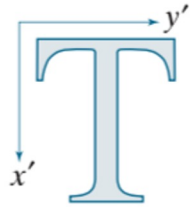
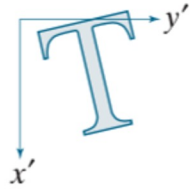
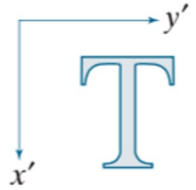
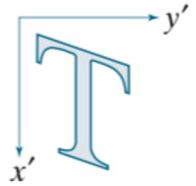
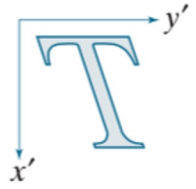
- **2.16 Homework Flip**
  - Write Flip function yourself, don't use cv2.flip
  - img2 = my_flip(img1,0) #same as cv2.flip(img1,0)
  - img3 = my_flip(img1,1) #same as cv2.flip(img1,1)
  - img4 = my_flip(img1,-1) #same as cv2.flip(img1,-1)

Affine Transformation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = A \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- This transformation can scale(縮放), rotate(旋轉), translate(平移) an image, depending on the values chosen for the elements of matrix A

TABLE 2.3
Affine transformations based on Eq. (2-45).

Transformation Name	Affine Matrix, A	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x$ $y' = y$	
Scaling/Reflection (For reflection, set one scaling factor to −1 and the other to 0)	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = c_x x$ $y' = c_y y$	
Rotation (about the origin)	$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x\cos\theta - y\sin\theta$ $y' = x\sin\theta + y\cos\theta$	
Translation	$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x + t_x$ $y' = y + t_y$	
Shear (vertical)	$\begin{bmatrix} 1 & s_v & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x + s_v y$ $y' = y$	
Shear (horizontal)	$\begin{bmatrix} 1 & 0 & 0 \\ s_h & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x$ $y' = s_h x + y$	

Check IP02.ipynb

- 2.17 Affine Transform
- 2.18 Homework Affine Transformation
  - The followings are code from ChatGPT
  - Please modify these codes for image affine transform to fit the following requirement
  - 1. only use OpenCV on image read and image display
  - 2. don't use PIL on image processing

Perspective Transformation

- Understanding Homography

透視轉換

定義 透視轉換

透視轉換 (Perspective Transformation) 可以定義爲：

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \boldsymbol{T} \cdot \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

其中，$\boldsymbol{T}$ 稱爲**透視轉換矩陣** (Perspective Transformation Matrix) 。

Check IP02.ipynb

- 2.19 Perspective Transform
- 2.20 Practice: Perspective transform
  - use perspective transform to find the following result

Check IP02.ipynb

- 2.21 Practice: Rotation and Scaling
  - Please use the combination of cv2.getRotationMatrix2D and cv2.warpAffine to rotate an image with given degree and fit in the image under size 512x512