# *Digital Image Processing*

李东晓

lidx@zju.edu.cn



浙江大学信息与电子工程学院
College of Information Science & Electronic Engineering, Zhejiang University

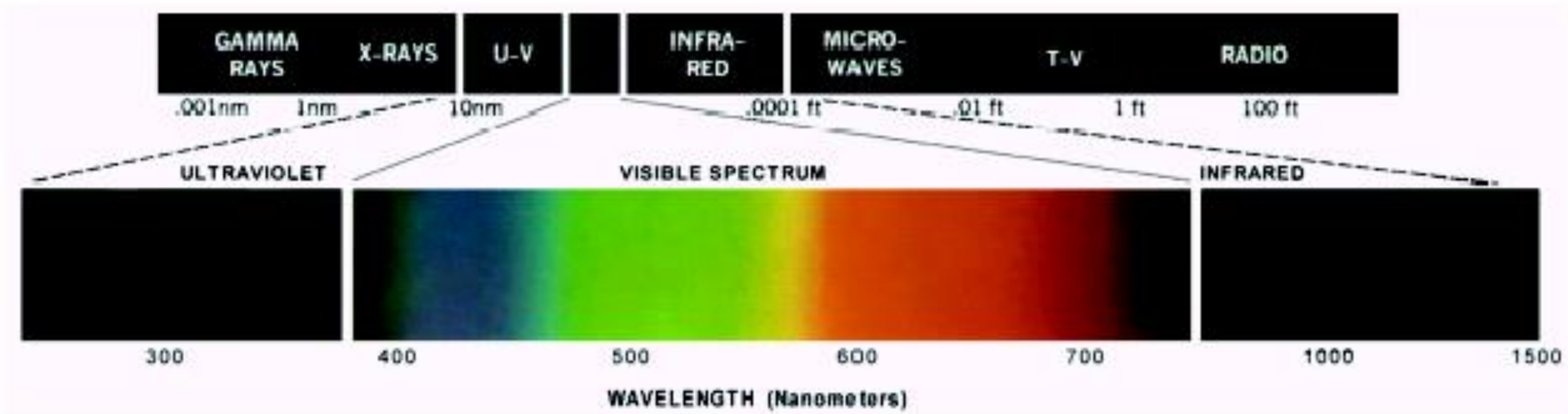
# Contents

# Light And The Electromagnetic Spectrum

Light is just a particular part of the electromagnetic spectrum that can be sensed by the human eye

The electromagnetic spectrum is split up according to the wavelengths of different forms of energy

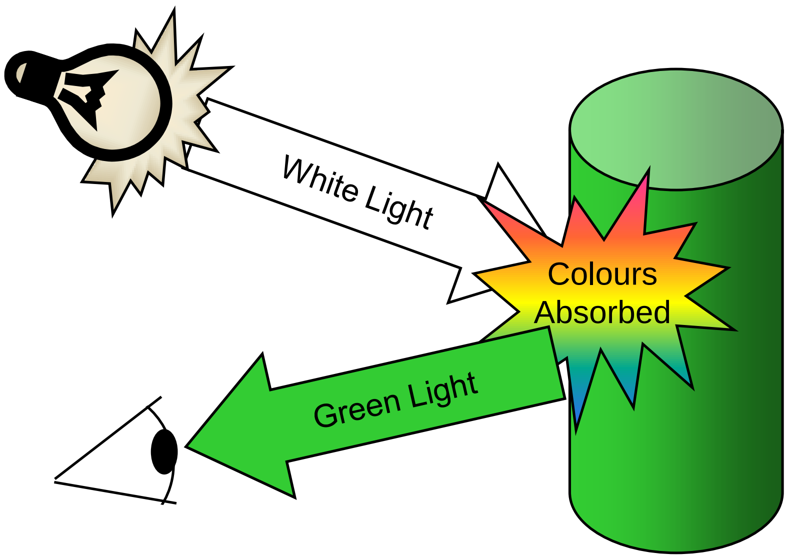
$$c = \lambda f$$



# Reflected Light

The colours that we perceive are determined by the nature of the light reflected from an object

For example, if white light is shone onto a green object most wavelengths are absorbed, while green light is reflected from the object

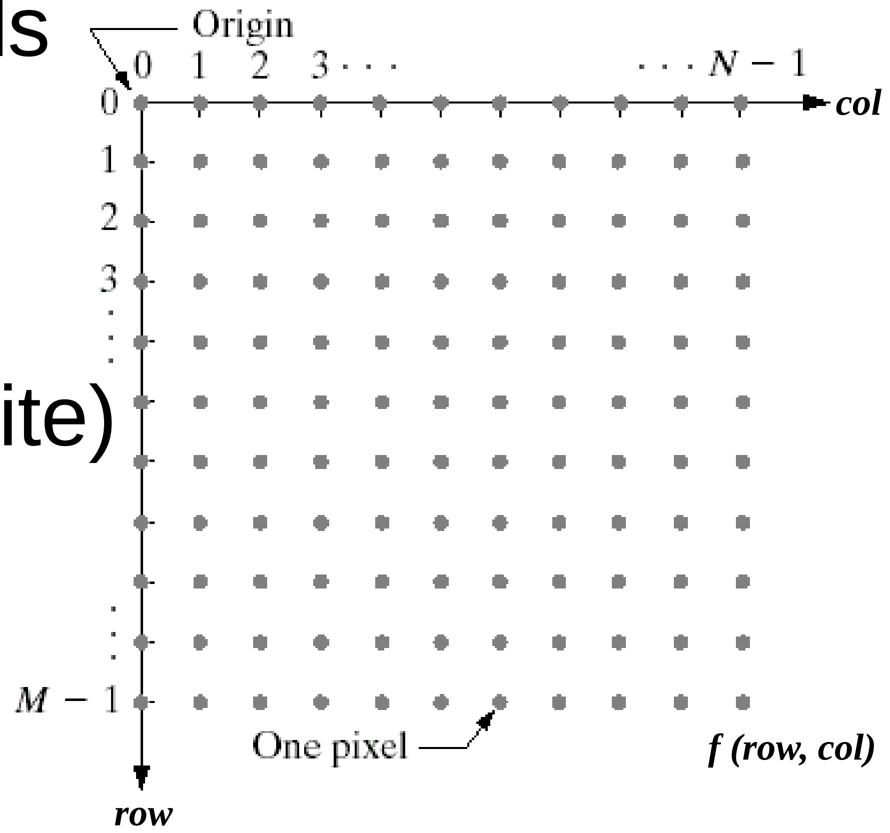


# 2D Digital Image Representation

Before we discuss image acquisition recall that a digital image is composed of *M* rows and *N* columns of pixels each storing a value

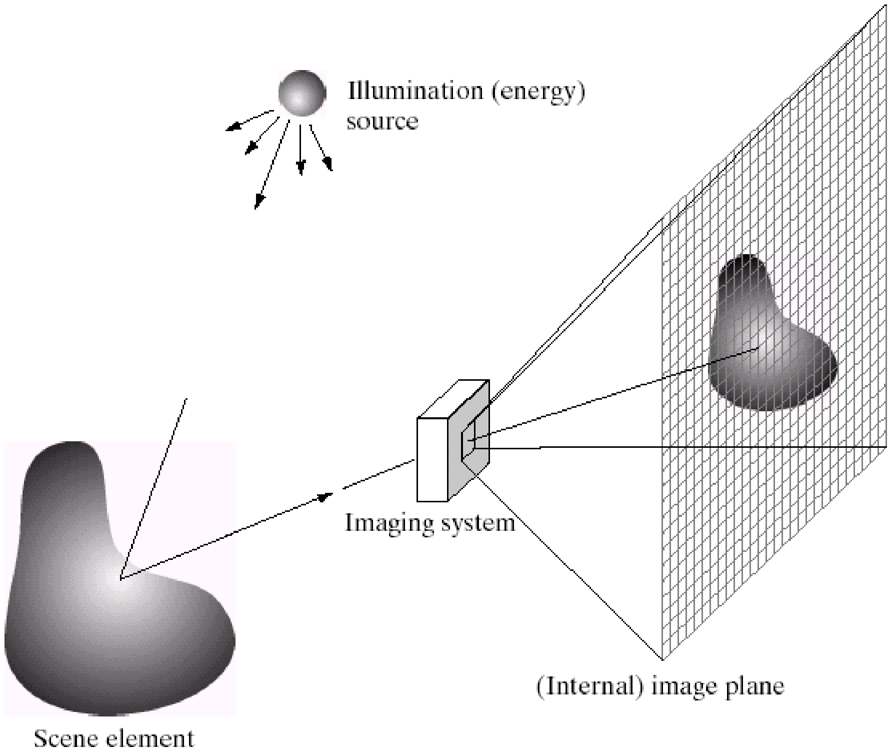
Pixel values are most often grey levels in the range 0-255 (black-white)

We will see later on that images can easily be represented as matrices



# Image Acquisition

Images are typically generated by *illuminating a scene* and absorbing the energy reflected by the objects in that scene



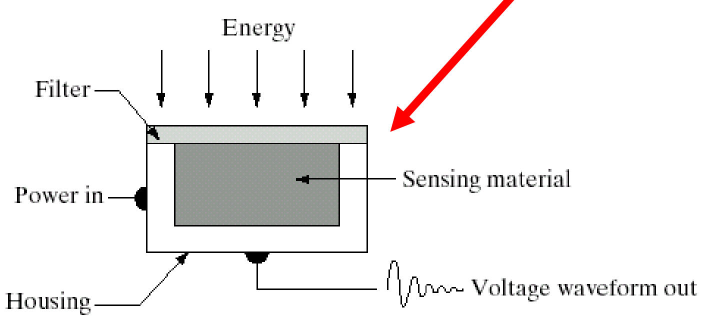

– Typical notions of illumination and scene can be way off:
  - X-rays of a skeleton
  - Ultrasound of an unborn baby
  - Electro-microscopic images of molecules

# Image Sensing

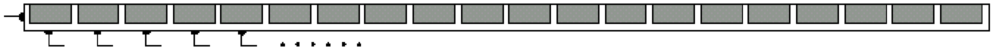
Incoming energy lands on a sensor material responsive to that type of energy and this generates a voltage

Collections of sensors are arranged to capture images

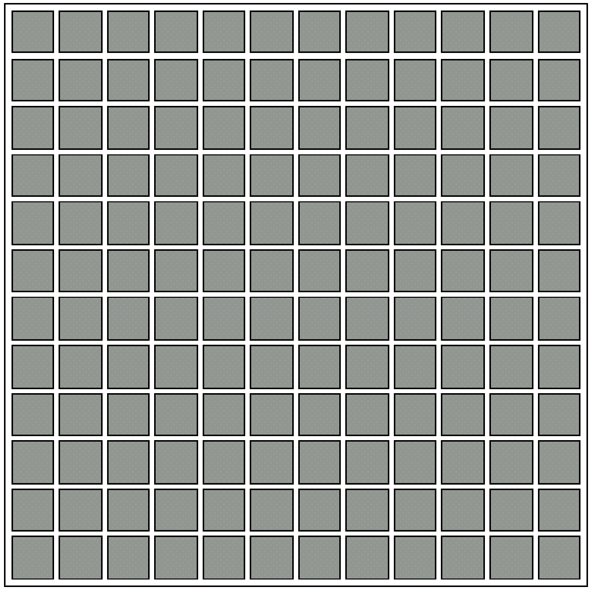
pixel




Imaging Sensor

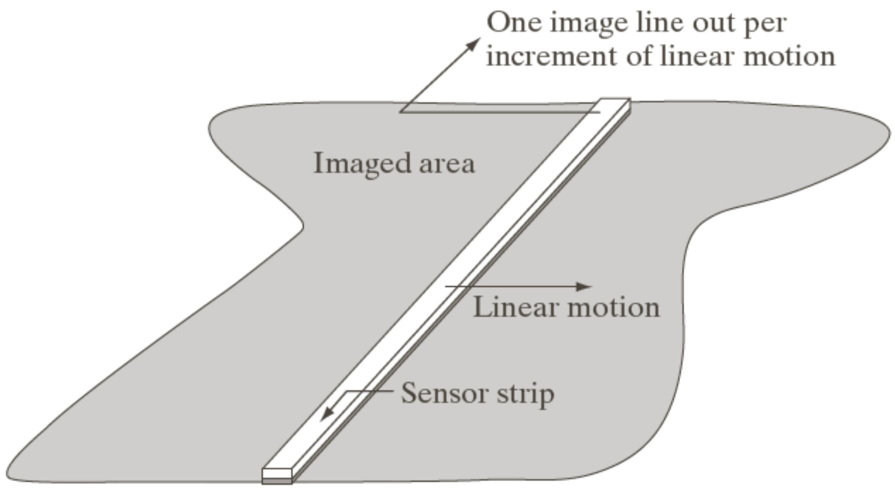


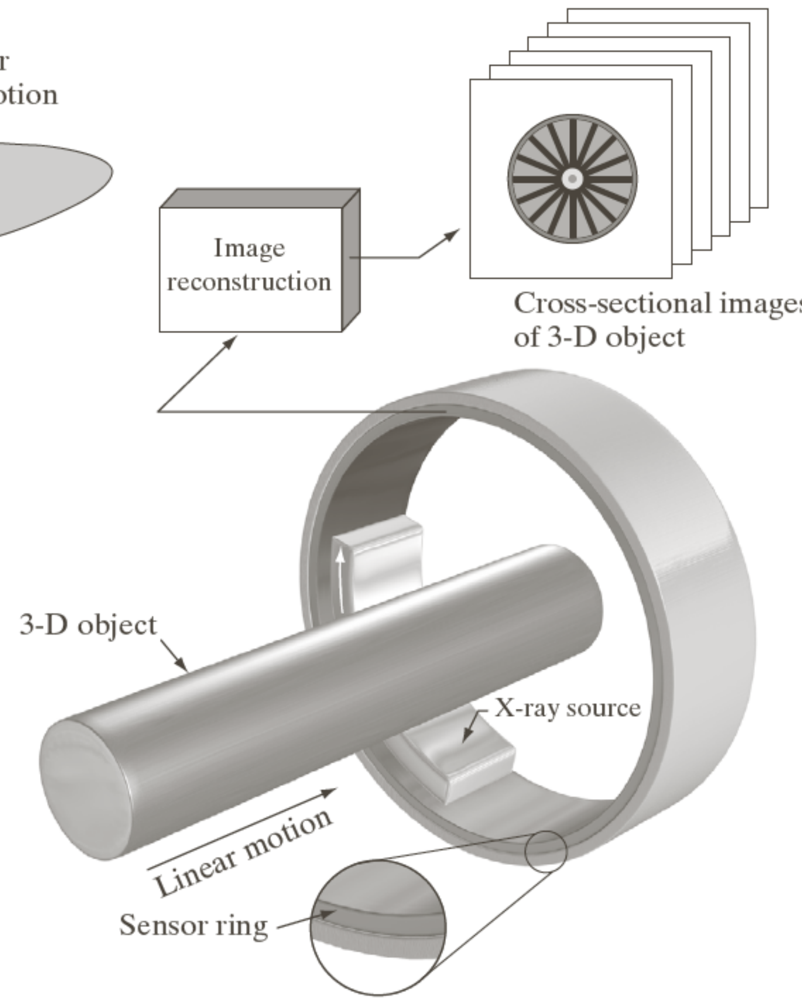
Line of Image Sensors



Array of Image Sensors

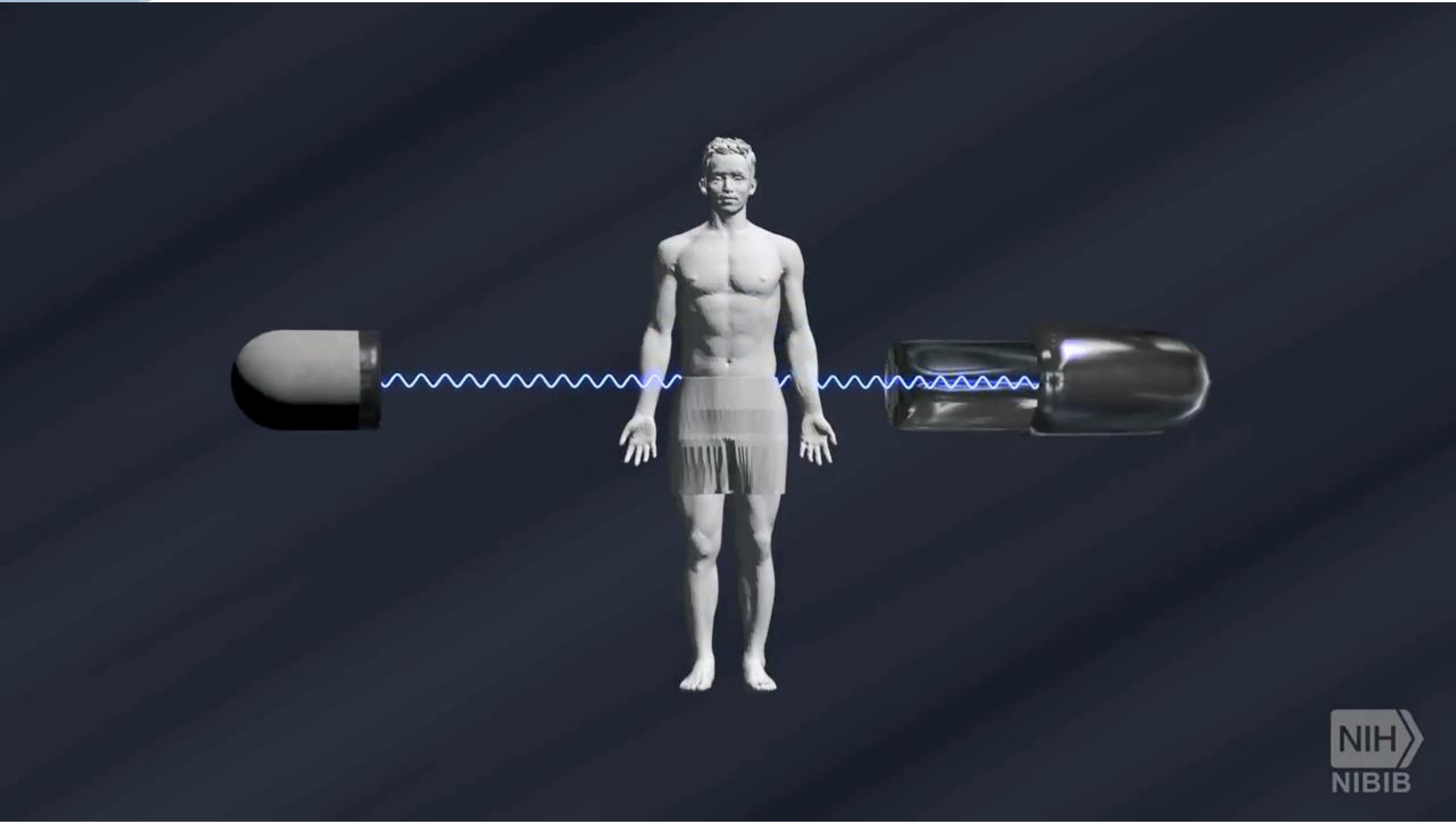
# Image Sensing



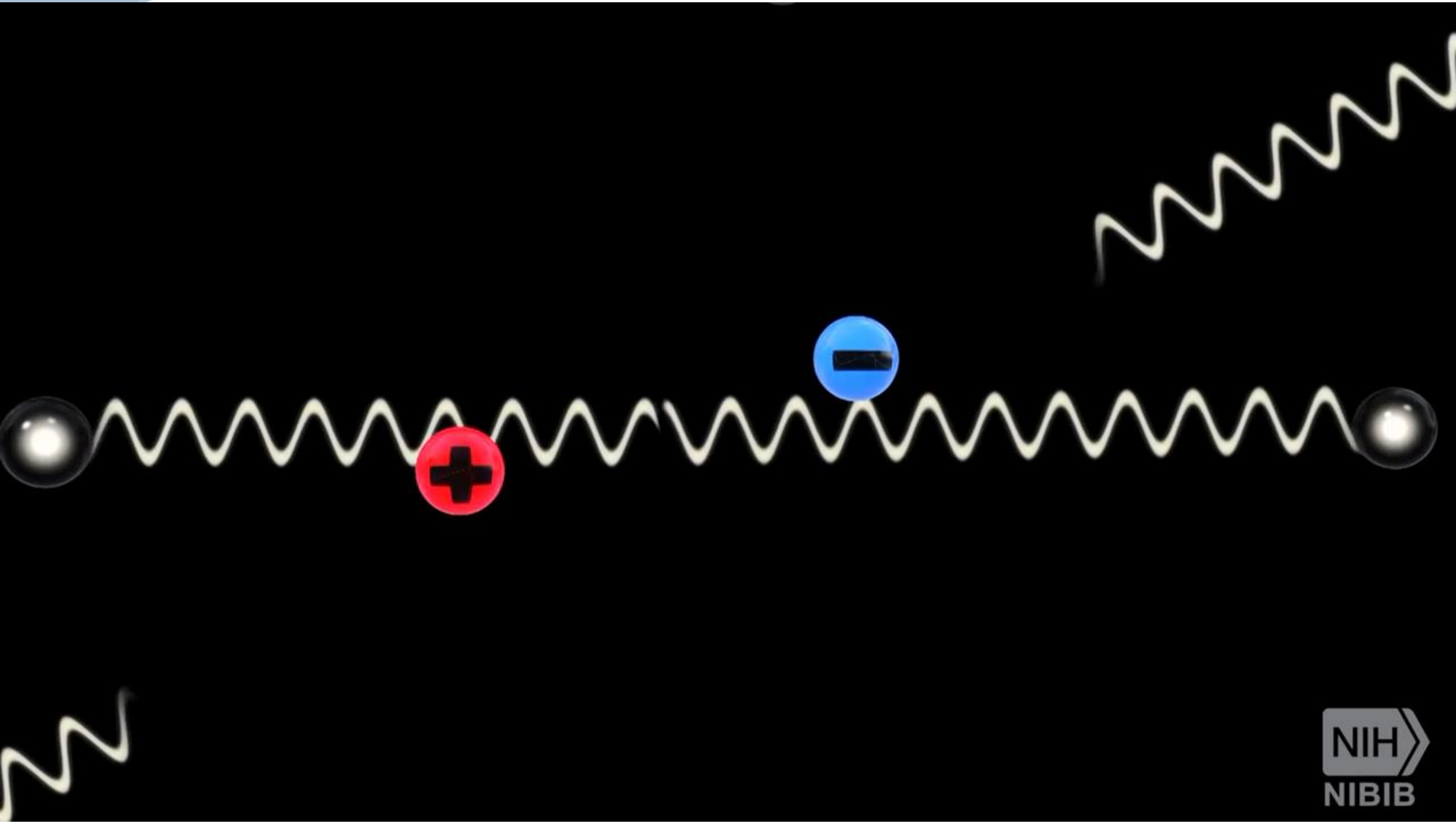




Using Sensor Strips and Rings

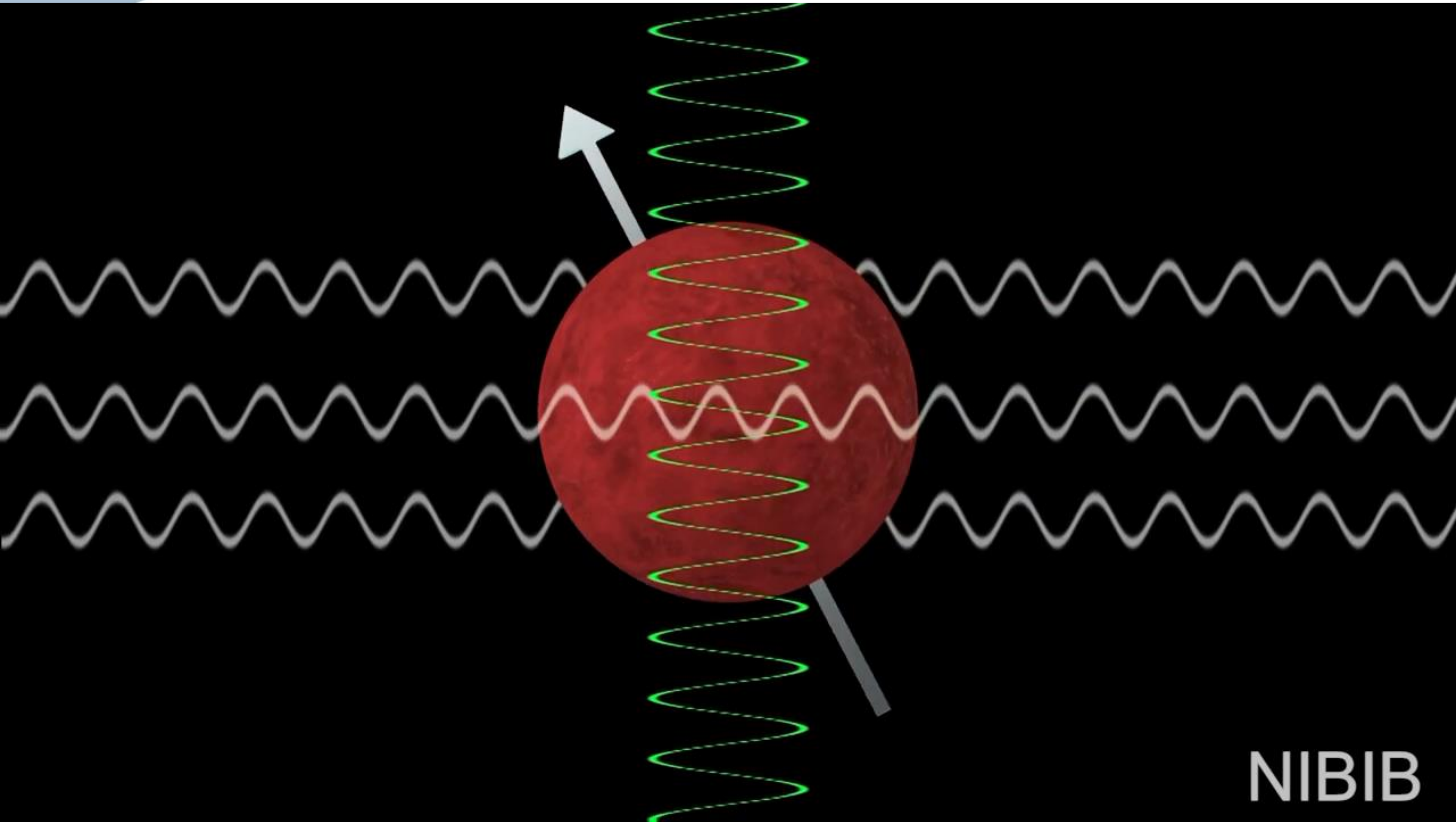
# How Does a CT Scan Work?



# How Does a PET Scan Work?



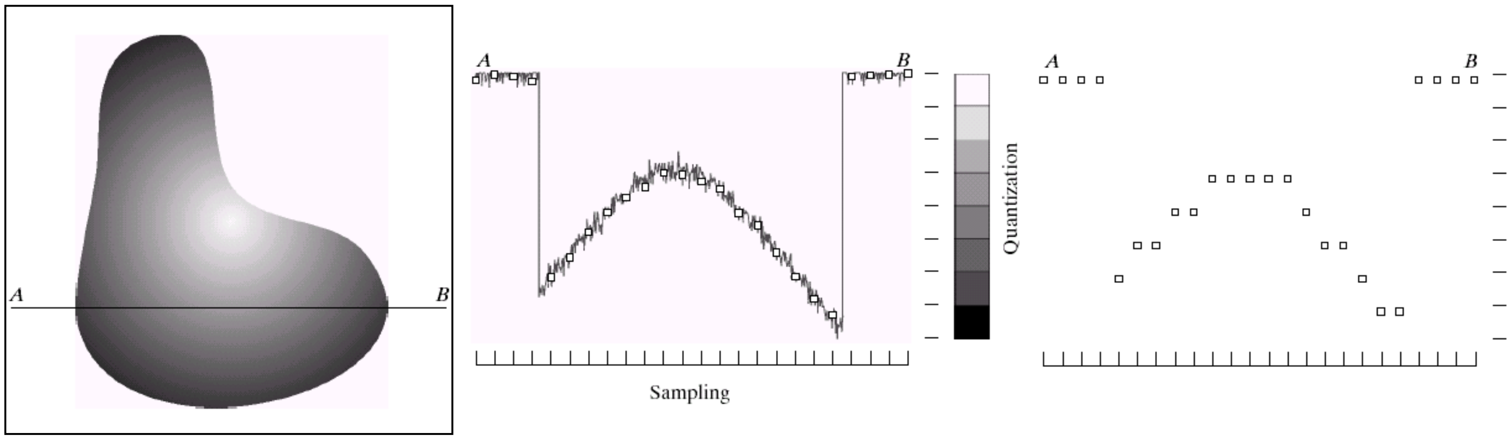
# How Does an MRI Scan Work?



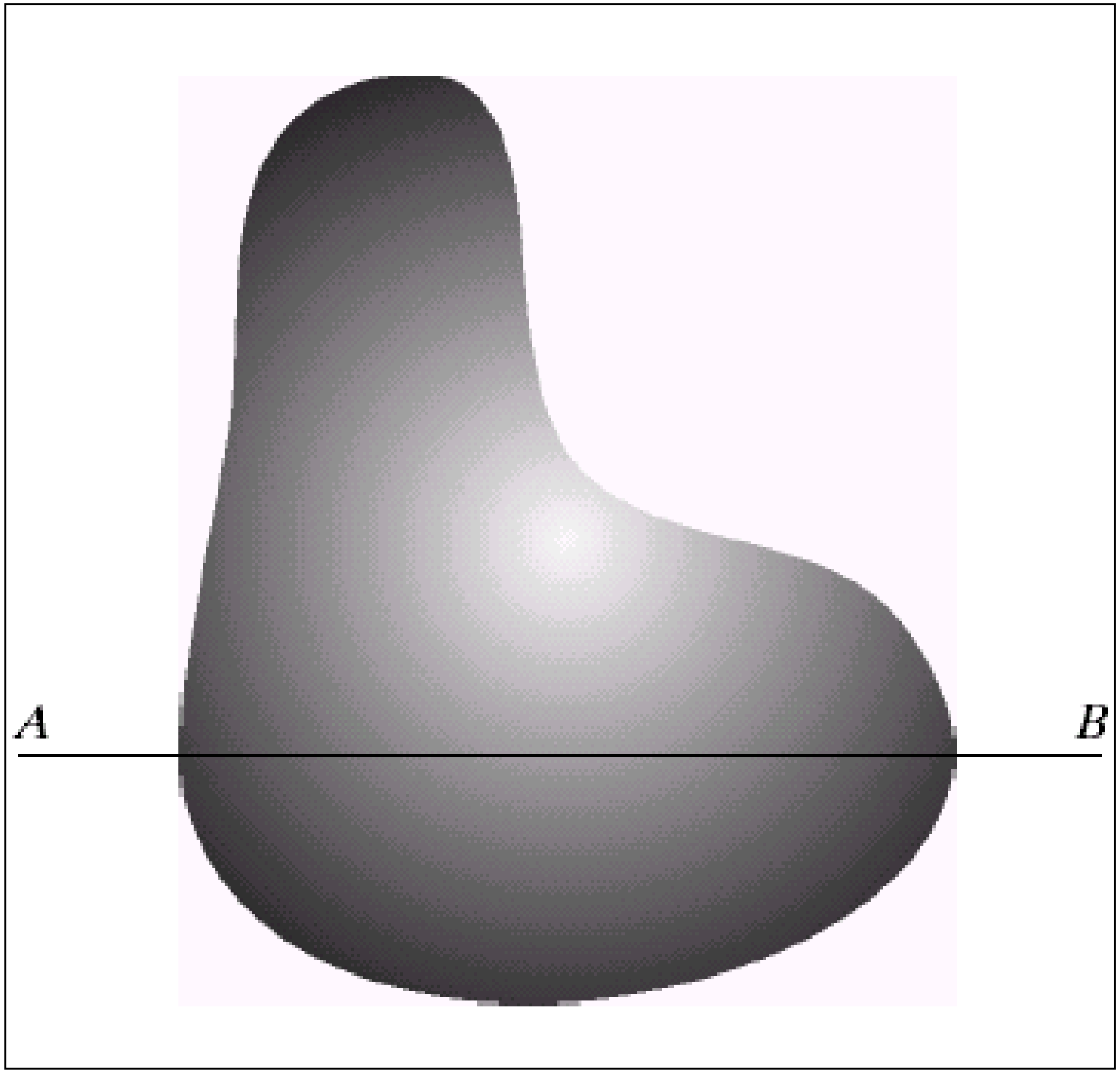
# Image Sampling And Quantisation

A digital sensor can only measure a limited number of **samples** at a **discrete** set of energy levels

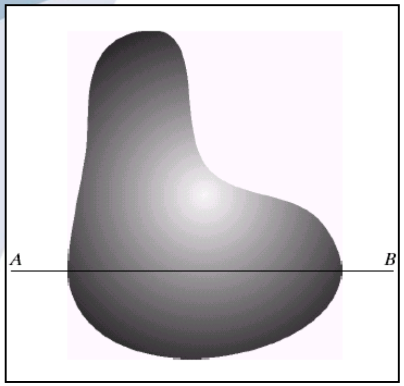
*Quantisation* is the process of converting a continuous **analogue** signal into a digital representation of this signal

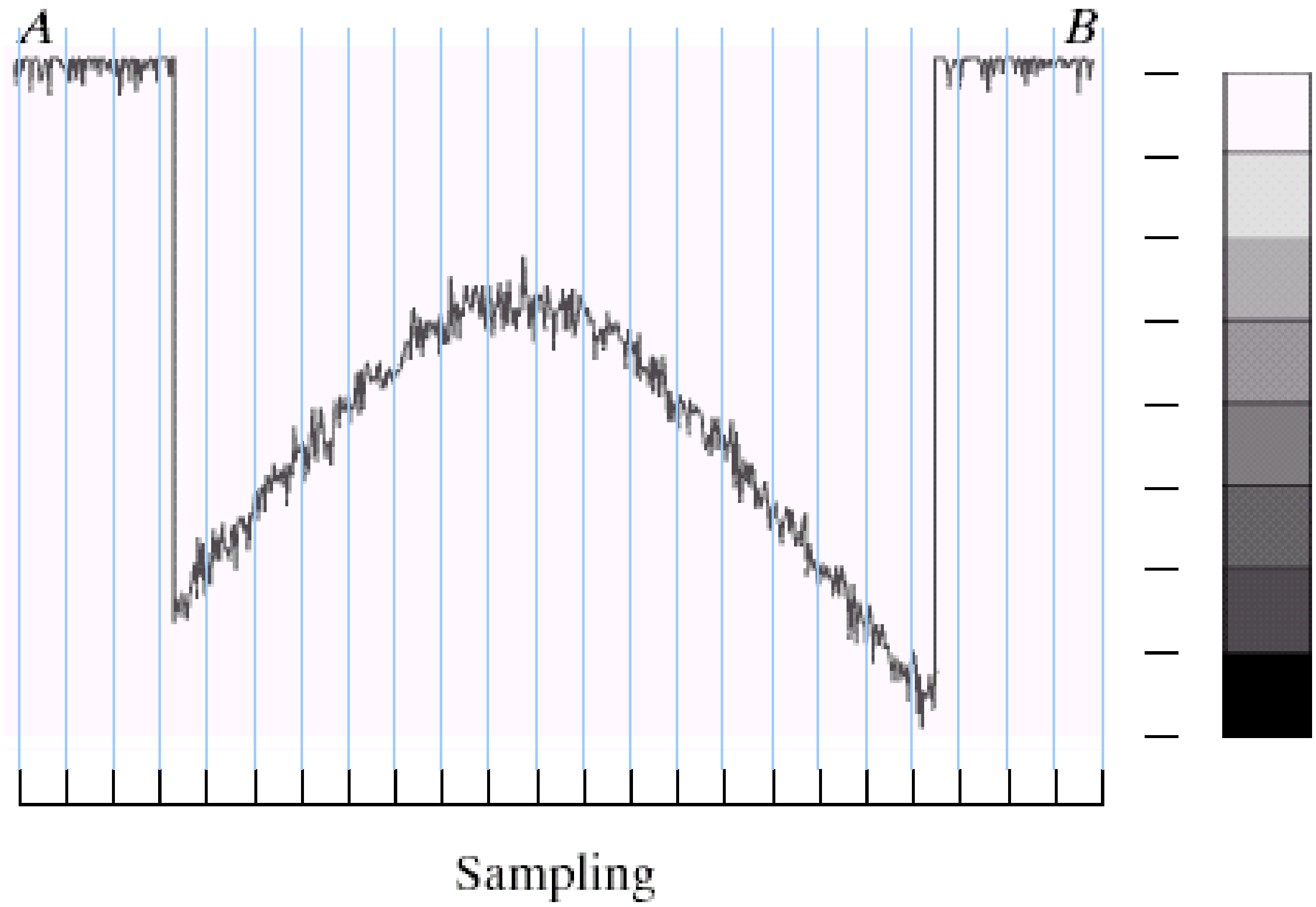


# Image Sampling And Quantisation



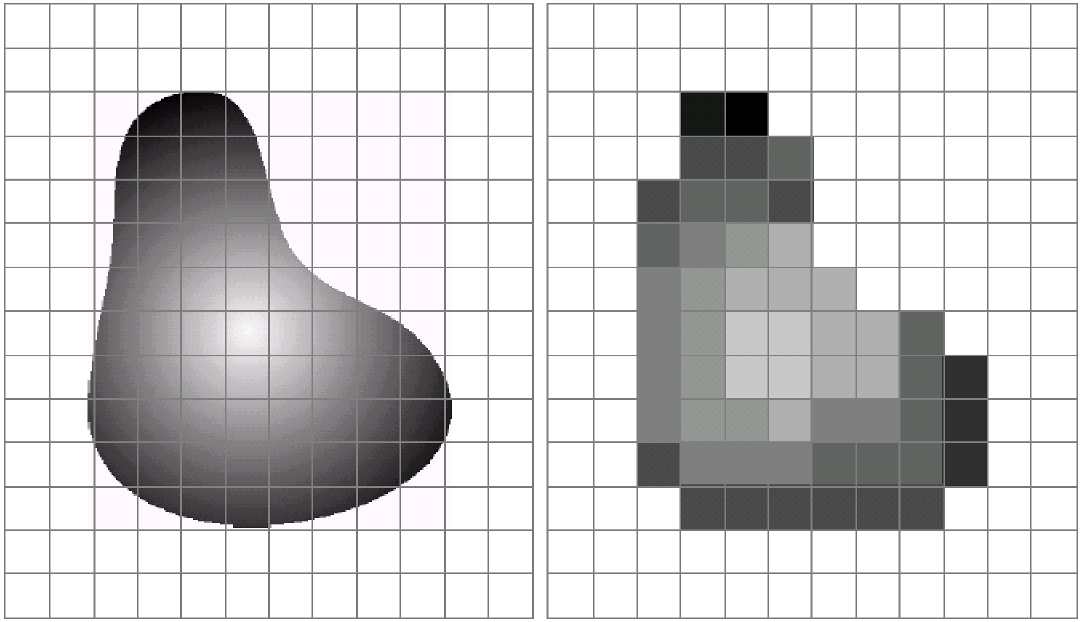
# Image Sampling And Quantisation



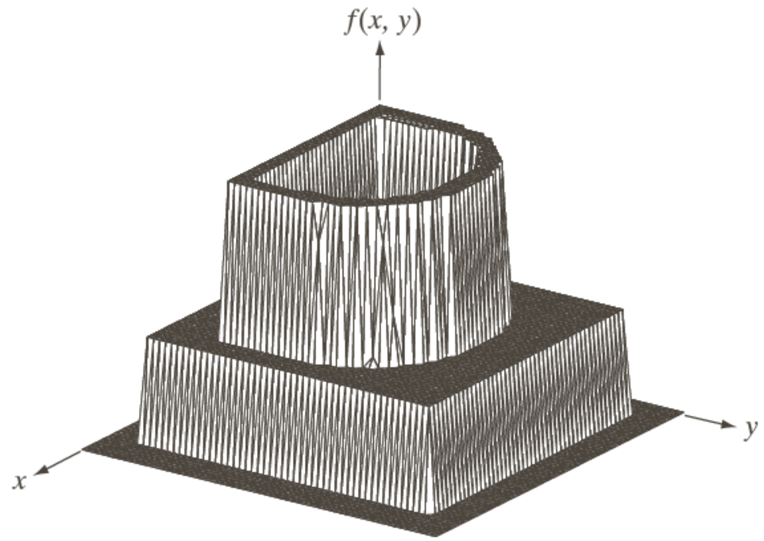



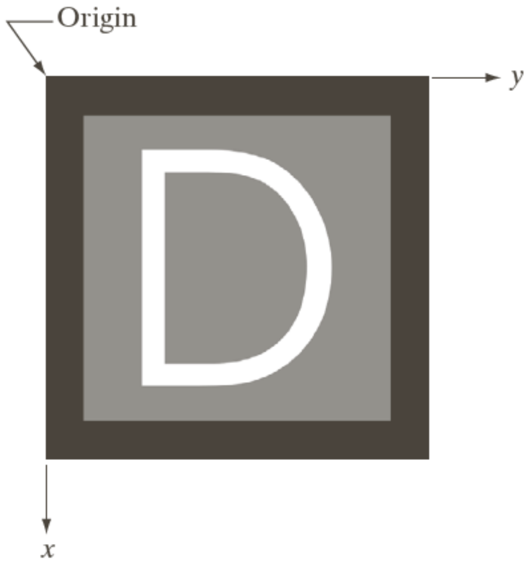
# Image Sampling And Quantisation

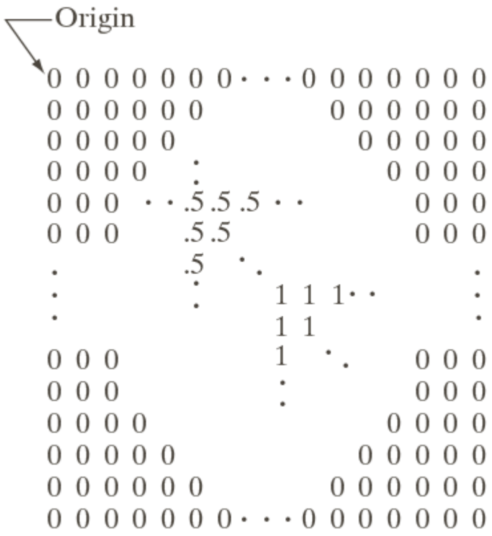
Remember that a digital image is always only an **approximation** of a real world scene



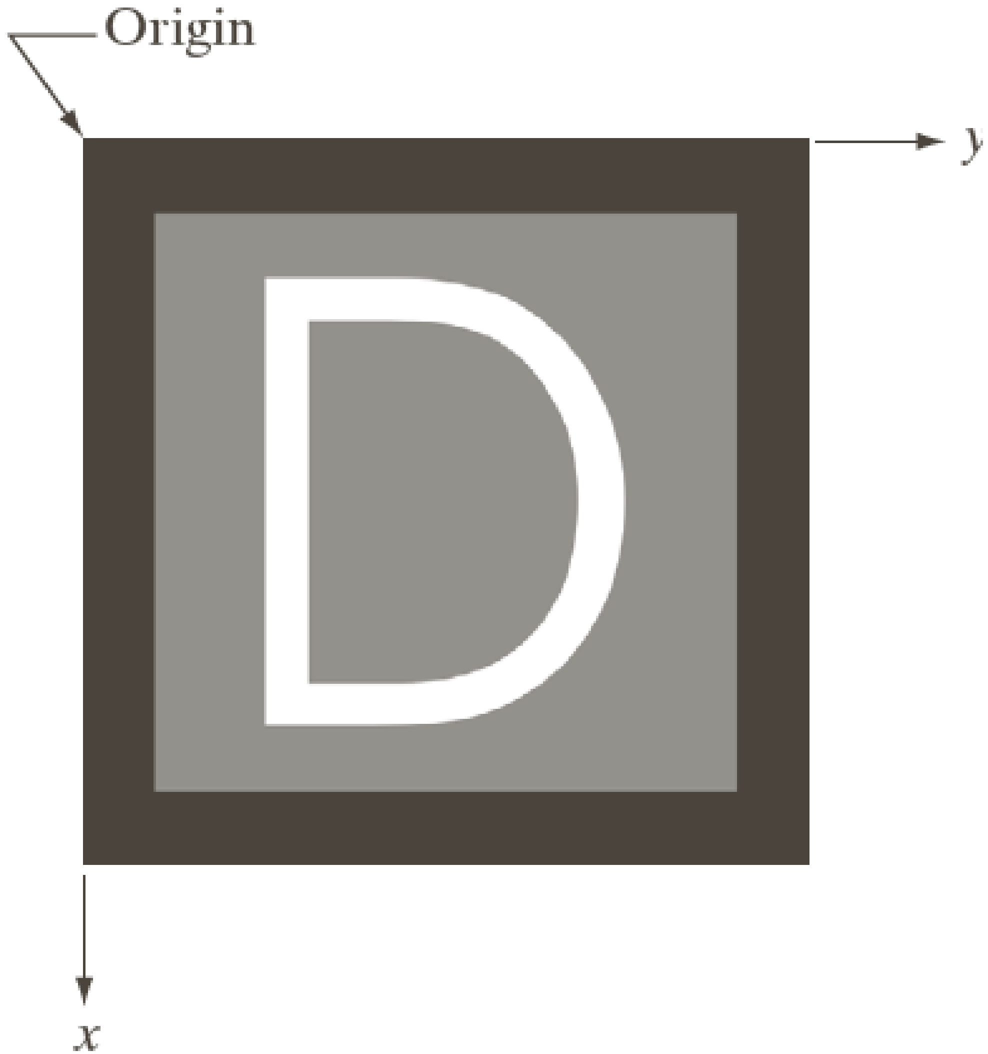
# Image Representation



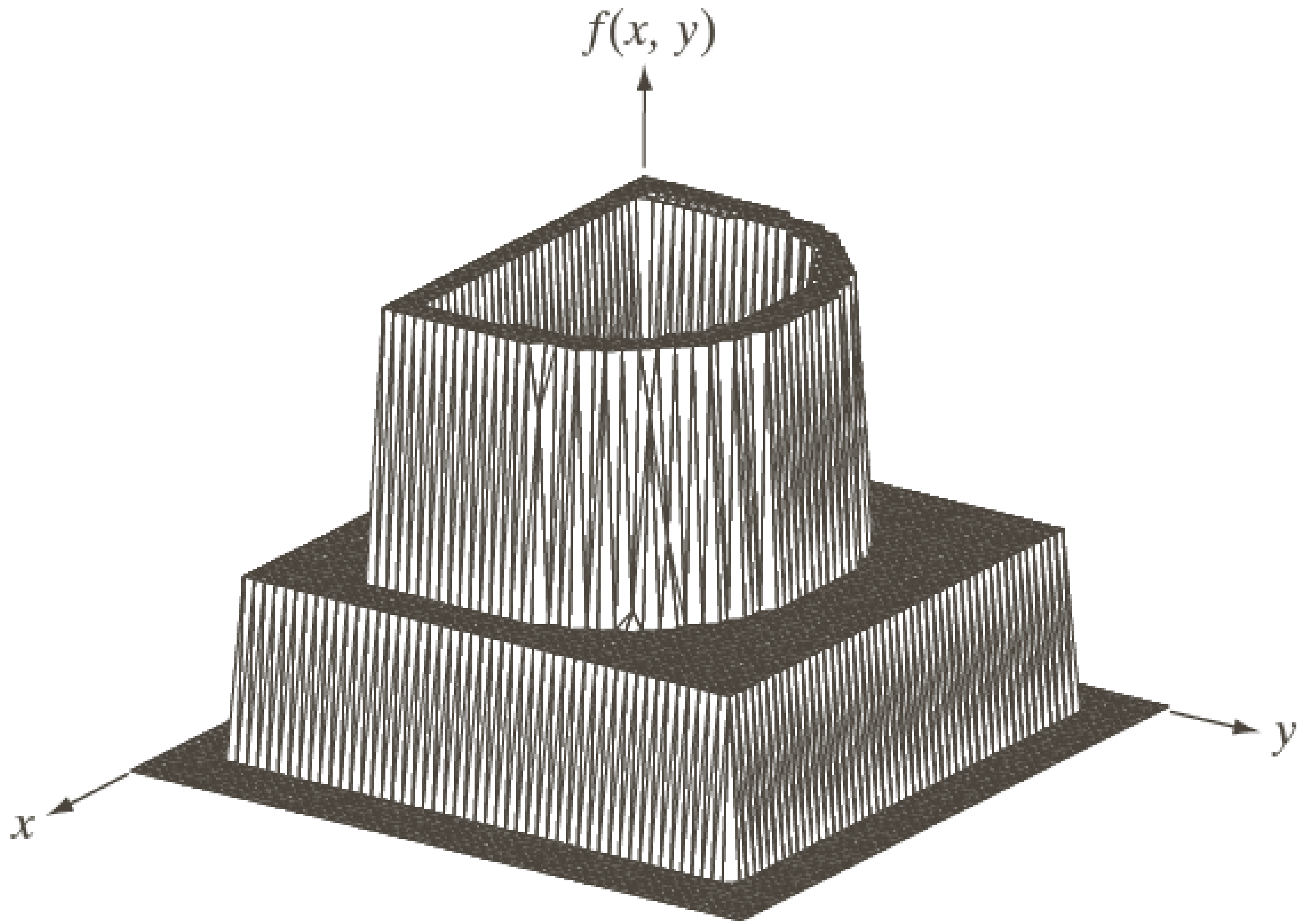



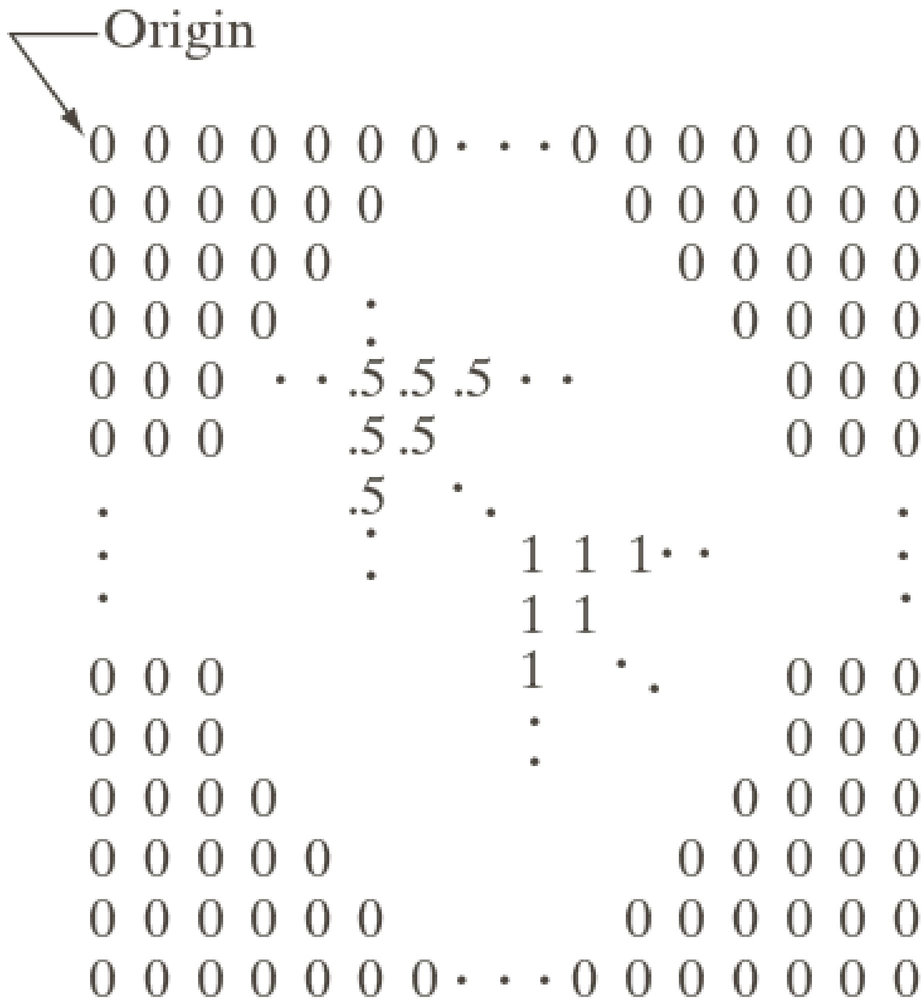
# Image Representation



# Image Representation



# Image Representation



# Spatial Resolution

*The spatial resolution* of an image is determined by how sampling was carried out

Spatial resolution simply refers to the smallest discernable detail in an image

- Vision specialists will often talk about pixel size
- Graphic designers will talk about *dots per inch* (DPI)

# Spatial Resolution (cont…)


Same Field-of-View (FOV)

# Spatial Resolution (cont…)

1024 X 1024

# Spatial Resolution (cont…)

512 X 512

# Spatial Resolution (cont…)

256 X 256

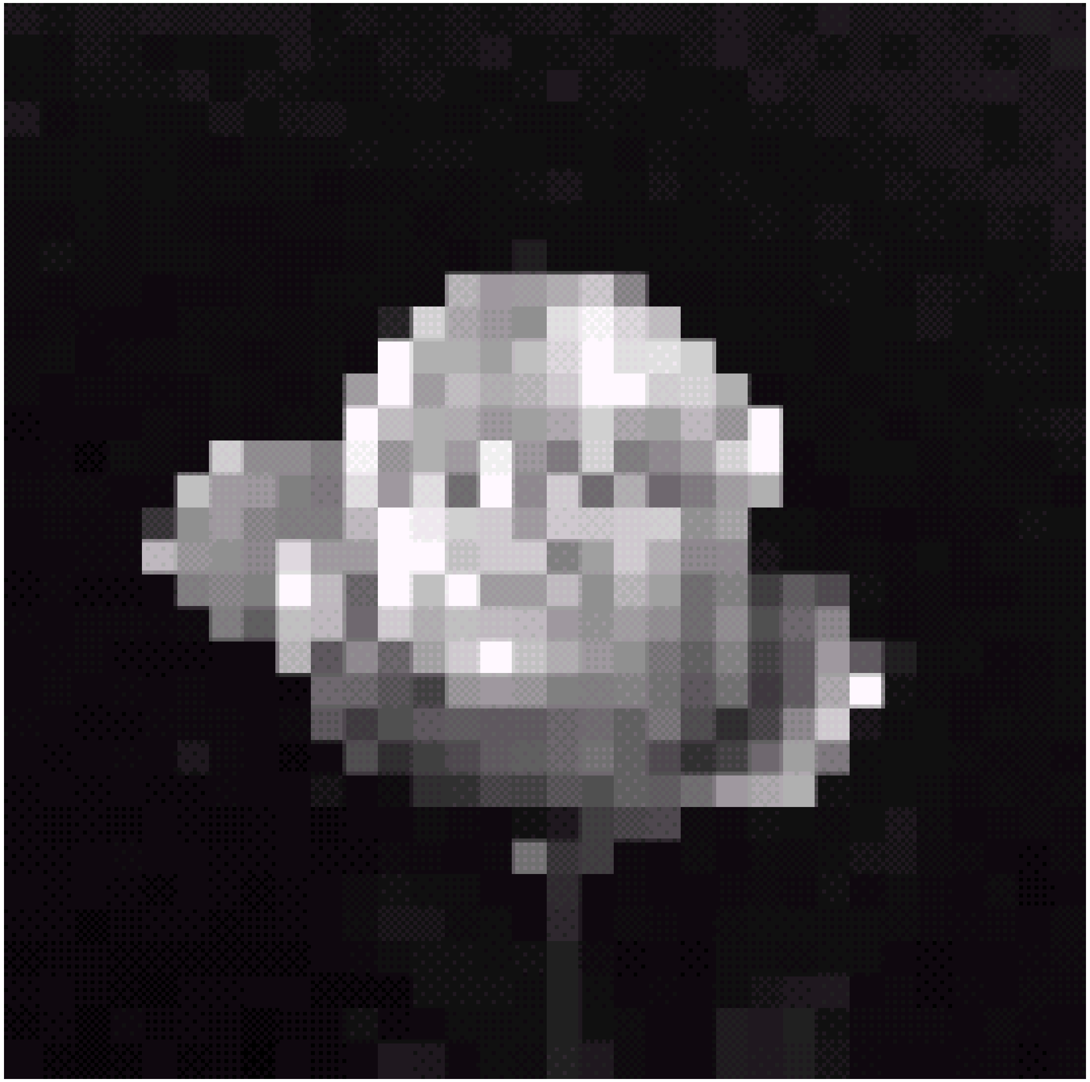
# Spatial Resolution (cont…)

128 X 128

# Spatial Resolution (cont…)

64 X 64

# Spatial Resolution (cont…)



32 X 32

# Intensity Level Resolution

*Intensity level resolution* refers to the number of intensity levels used to represent the image

- The more intensity levels used, the finer the level of detail discernable in an image
- Intensity level resolution is usually given in terms of the number of bits used to store each intensity level

| Number of Bits | Number of Intensity Levels | Examples |
|---|---|---|
| 1 | 2 | 0, 1 |
| 2 | 4 | 00, 01, 10, 11 |
| 4 | 16 | 0000, 0101, 1111 |
| 8 | 256 | 00110011, 01010101 |
| 16 | 65,536 | 1010101010101010 |

# Resolution and Raw Data Size

**TABLE 2.1**
Number of storage bits for various values of $N$ and $k$.

| $N/k$ | 1 ($L$ = 2) | 2 ($L$ = 4) | 3 ($L$ = 8) | 4 ($L$ = 16) | 5 ($L$ = 32) | 6 ($L$ = 64) | 7 ($L$ = 128) | 8 ($L$ = 256) |
|---|---|---|---|---|---|---|---|---|
| 32 | 1,024 | 2,048 | 3,072 | 4,096 | 5,120 | 6,144 | 7,168 | 8,192 |
| 64 | 4,096 | 8,192 | 12,288 | 16,384 | 20,480 | 24,576 | 28,672 | 32,768 |
| 128 | 16,384 | 32,768 | 49,152 | 65,536 | 81,920 | 98,304 | 114,688 | 131,072 |
| 256 | 65,536 | 131,072 | 196,608 | 262,144 | 327,680 | 393,216 | 458,752 | 524,288 |
| 512 | 262,144 | 524,288 | 786,432 | 1,048,576 | 1,310,720 | 1,572,864 | 1,835,008 | 2,097,152 |
| 1024 | 1,048,576 | 2,097,152 | 3,145,728 | 4,194,304 | 5,242,880 | 6,291,456 | 7,340,032 | 8,388,608 |
| 2048 | 4,194,304 | 8,388,608 | 12,582,912 | 16,777,216 | 20,971,520 | 25,165,824 | 29,369,128 | 33,554,432 |
| 4096 | 16,777,216 | 33,554,432 | 50,331,648 | 67,108,864 | 83,886,080 | 100,663,296 | 117,440,512 | 134,217,728 |
| 8192 | 67,108,864 | 134,217,728 | 201,326,592 | 268,435,456 | 335,544,320 | 402,653,184 | 469,762,048 | 536,870,912 |

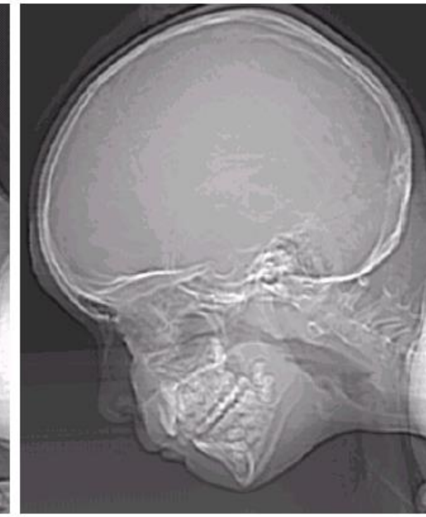
# Intensity Level Resolution (cont…)

256 grey levels (8 bits per pixel)

128 grey levels (7 bpp)

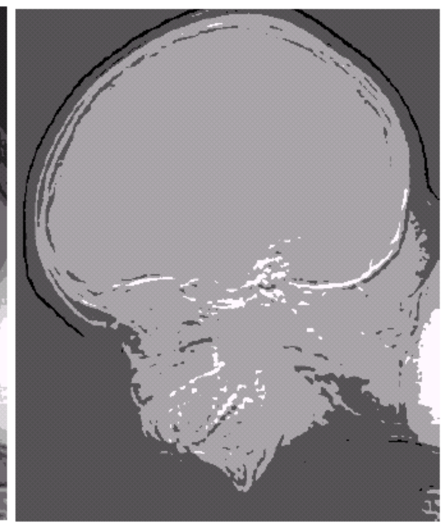
64 grey levels (6 bpp)

32 grey levels (5 bpp)



16 grey levels (4 bpp)

8 grey levels (3 bpp)



4 grey levels (2 bpp)

2 grey levels (1 bpp)

# Intensity Level Resolution (cont…)

bits per pixel

8 bpp

256 grey levels

# Intensity Level Resolution (cont…)

7 bpp

128 grey levels

# Intensity Level Resolution (cont…)

6 bpp

64 grey levels

# Intensity Level Resolution (cont…)

5 bpp

32 grey levels

# Intensity Level Resolution (cont…)

4 bpp

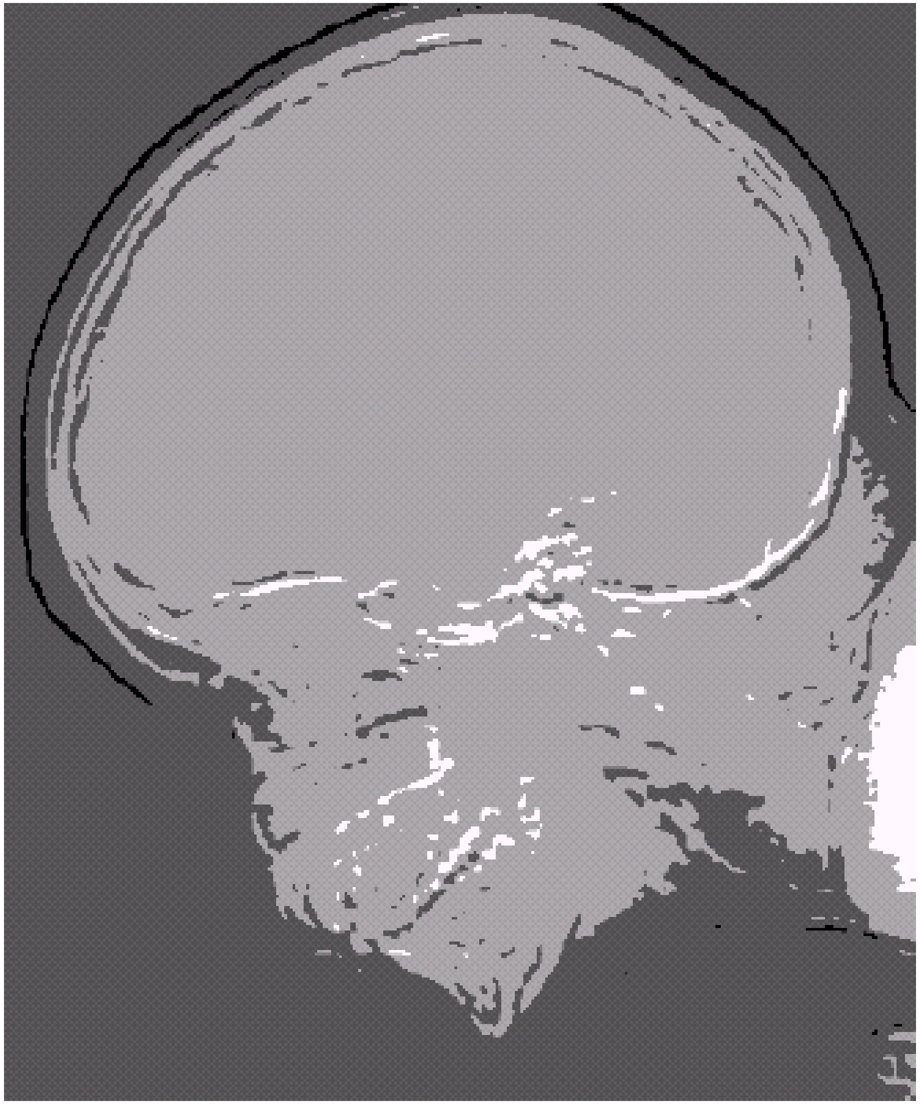
16 grey levels

# Intensity Level Resolution (cont…)

3 bpp

8 grey levels

# Intensity Level Resolution (cont…)



2 bpp

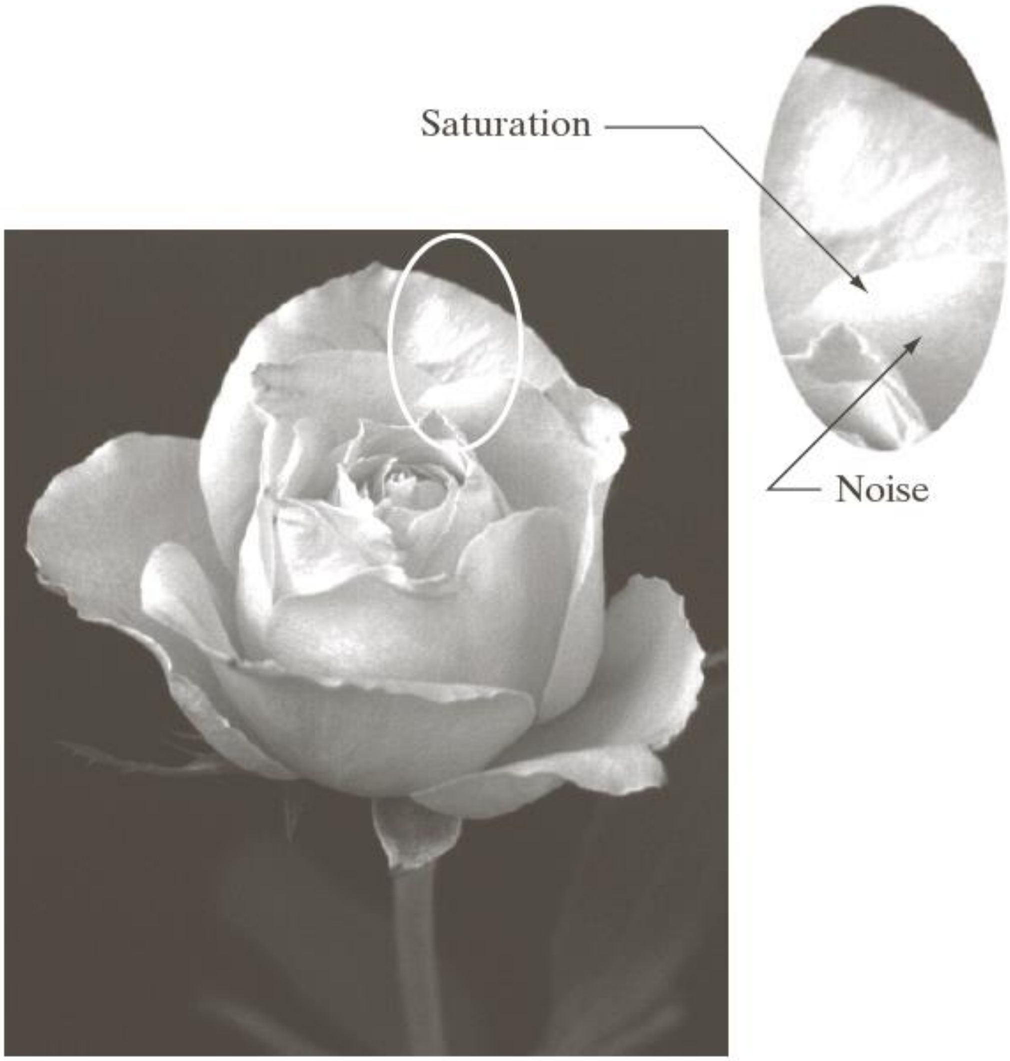
4 grey levels

# Intensity Level Resolution (cont…)

1 bpp

2 grey levels

# Saturation & Noise



# Resolution: How Much Is Enough?

The big question with resolution is always *how much is enough*?

- This all depends on what is in the image and what you would like to do with it
- Key questions include
  - Does the image look aesthetically pleasing?
  - Can you see what you need to see within the image?

# Resolution: How Much Is Enough? (cont…)


The picture on the right is fine for counting the number of cars, but NOT for reading the number plate

# Question?

对于人脸识别，脸部需要多大分辨率？

- 至少 ≥ 64x64
- 最好 ≥ 128x128


# Intensity Level Resolution (cont…)

Low Detail

Medium Detail

High Detail

# Intensity Level Resolution (cont…)

Lena

# Intensity Level Resolution (cont…)

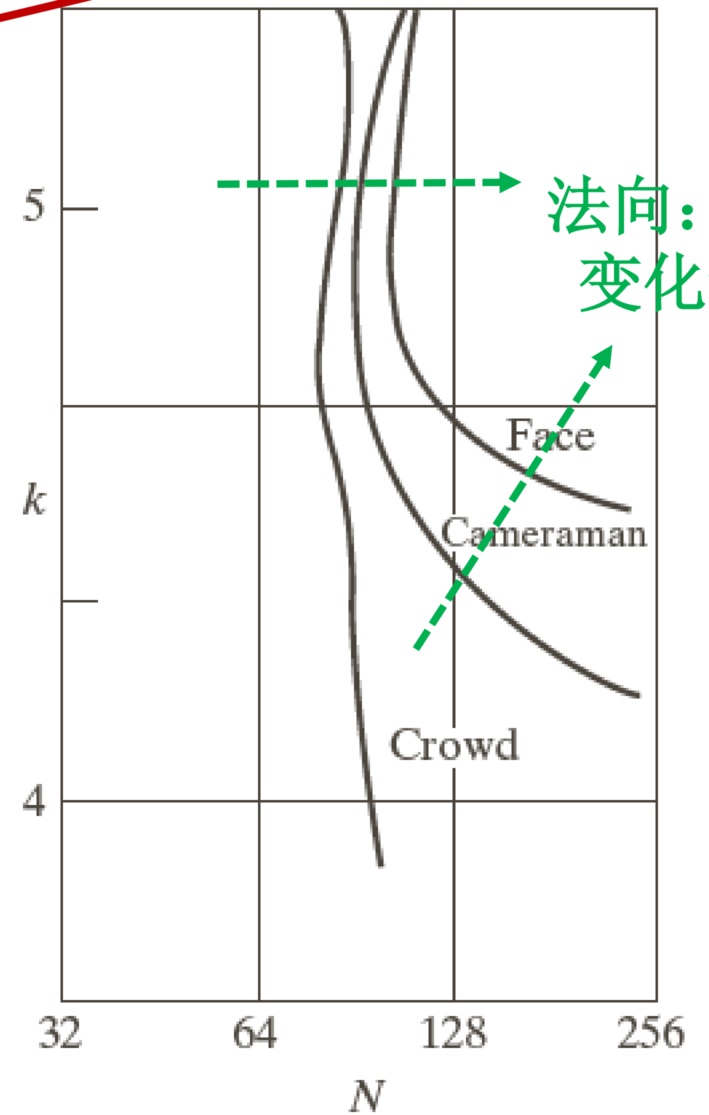
Cameraman

# Intensity Level Resolution (cont…)

Crowd

# Typical isopreference curve

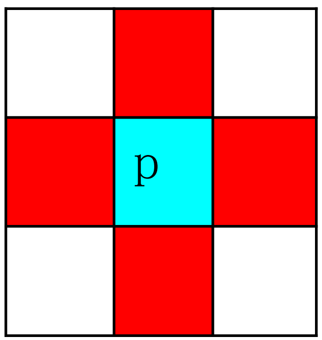
等偏爱
（相同主观质量）



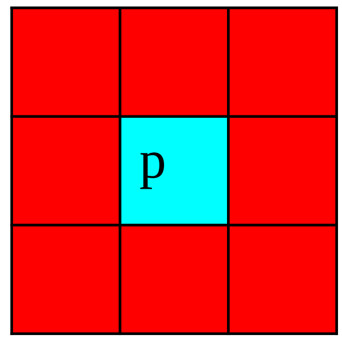

法向：主观质量变化最大方向

# 2.5 Some Basic relationships between pixels

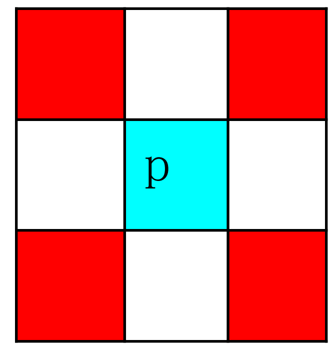
## Neighborhood of a Pixel




$N_4(p)$




$N_8(p)$



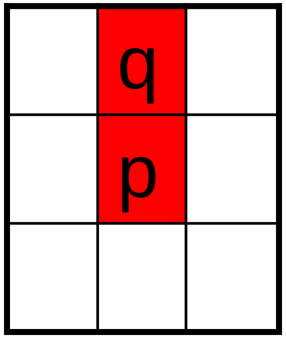

$N_D(p)$

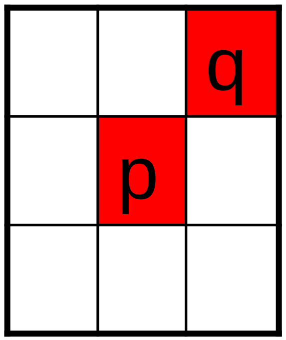
- Three types of adjacency

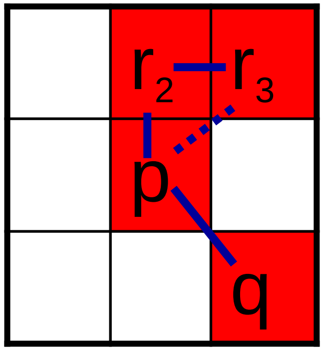
1. 4 – adjacency: $q \in N_4(p)$
2. 8 – adjacency: $q \in N_8(p)$
3. *m* – adjacency (mixed - adjacency)

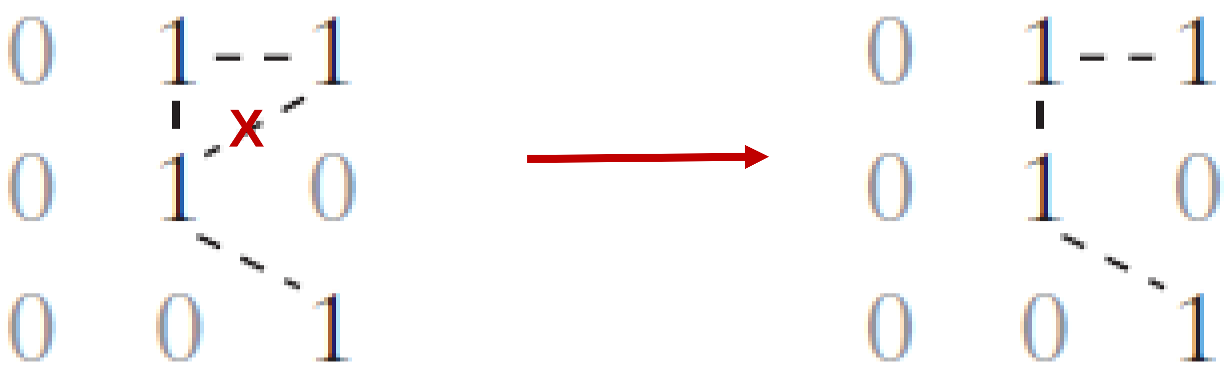
- $q \in N_4(p)$ OR
- $q \in N_D(p)$ AND $(N_4(p) \cap N_4(q) = \emptyset)$

Eliminate ambiguities in 8 – adjacency



- Path $(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$

$(x_i, y_i)$ and $(x_{i-1}, y_{i-1})$ are adjacent for $1 \leq i \leq n$

1. 4 – paths
2. 8 – paths
3. *m* - paths

- p and q are Connected if there exists a path between them
- Connected component: a set of all connected pixels
- Region: a subset of connected pixels

- Adjacent / Disjoint Regions

depend on the type of adjacency

- Inner Boundary (border, contour)

set of pixels that are adjacent to pixels in the complement of region R

- Outer Boundary

corresponding border in the background

- Image Boundary
- Edge vs Border: local vs global

$$
\left.\begin{matrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{matrix}\right\} R_i
$$
$$
\left.\begin{matrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}\right\} R_j
$$

$$
\begin{matrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{matrix}
$$

# Distance Measures

- Distance metric

(a) $D(p, q) \geq 0 \quad (D(p, q) = 0 \quad \text{iff} \quad p = q),$

(b) $D(p, q) = D(q, p)$, and

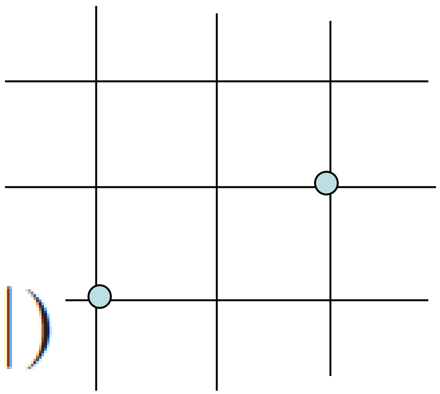
(c) $D(p, z) \leq D(p, q) + D(q, z).$

- Euclidean distance

$$D_e(p, q) = \left[(x - s)^2 + (y - t)^2\right]^{\frac{1}{2}}$$

- City-block distance

$$D_4(p, q) = |x - s| + |y - t|$$



- Chessboard distance

$$D_8(p, q) = \max(|x - s|, |y - t|)$$

- $D_m$ distance: length of shortest *m* - path

# 2.6 Mathematical Tools Used in DIP

- Array vs Matrix Operations
  - Array Product

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$$

  - Matrix Product

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

# Mathematical Tools (cont…)

- Linear vs Nonlinear Operations

$$H\left[f(x,y)\right] = g(x,y)$$

  – Linear operator (additivity & homogeneity)

$$H\left[a_i f_i(x,y) + a_j f_j(x,y)\right] = a_i H\left[f_i(x,y)\right] + a_j H\left[f_j(x,y)\right]$$

$$= a_i g_i(x,y) + a_j g_j(x,y)$$

  – Nonlinear operator
    - max, min, median, etc.

# Arithmetic Operations

- Four arithmetic array operations

$$
\begin{aligned}
s(x, y) &= f(x, y) \boxed{+} g(x, y) \\
d(x, y) &= f(x, y) \boxed{-} g(x, y) \\
p(x, y) &= f(x, y) \boxed{\times} g(x, y) \\
v(x, y) &= f(x, y) \boxed{\div} g(x, y)
\end{aligned}
$$

# Arithmetic Operations

- Addition (e.g. image denoising)

$$g(x, y) = f(x, y) + \eta(x, y)$$

$$\bar{g}(x, y) = \frac{1}{K}\sum_{i=1}^{K} g_i(x, y)$$

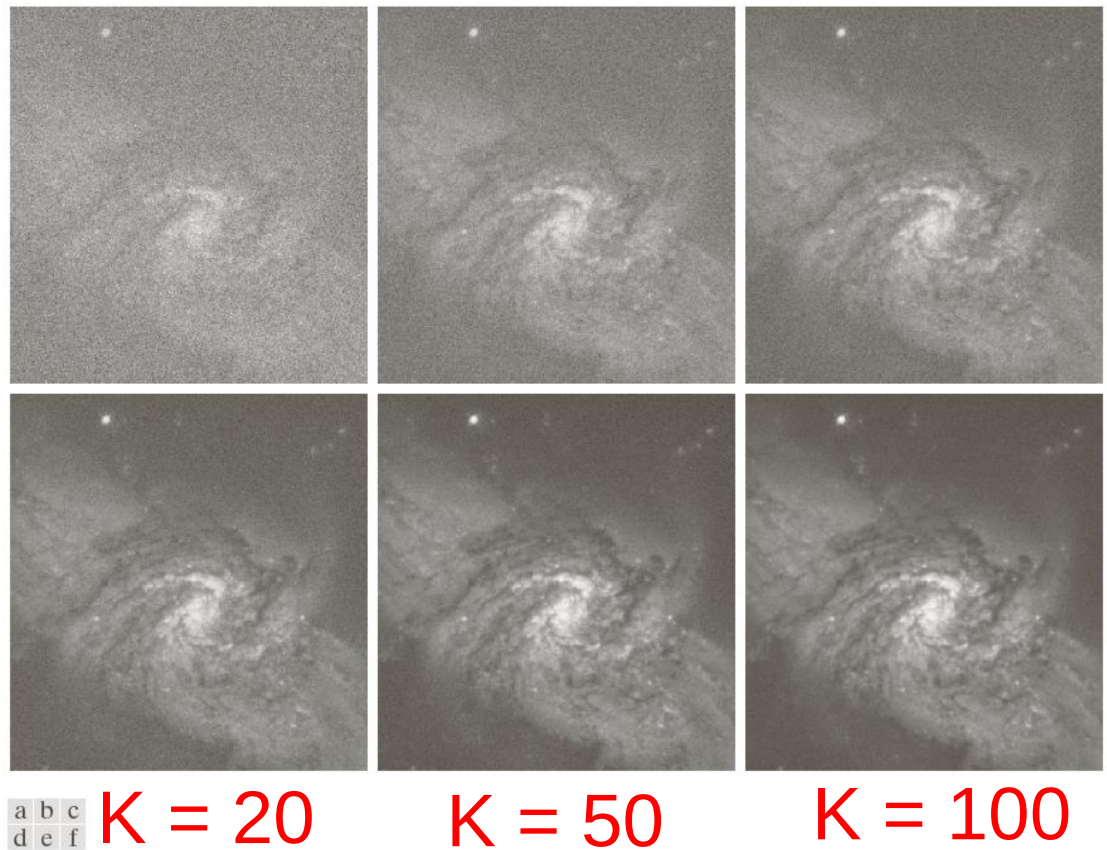
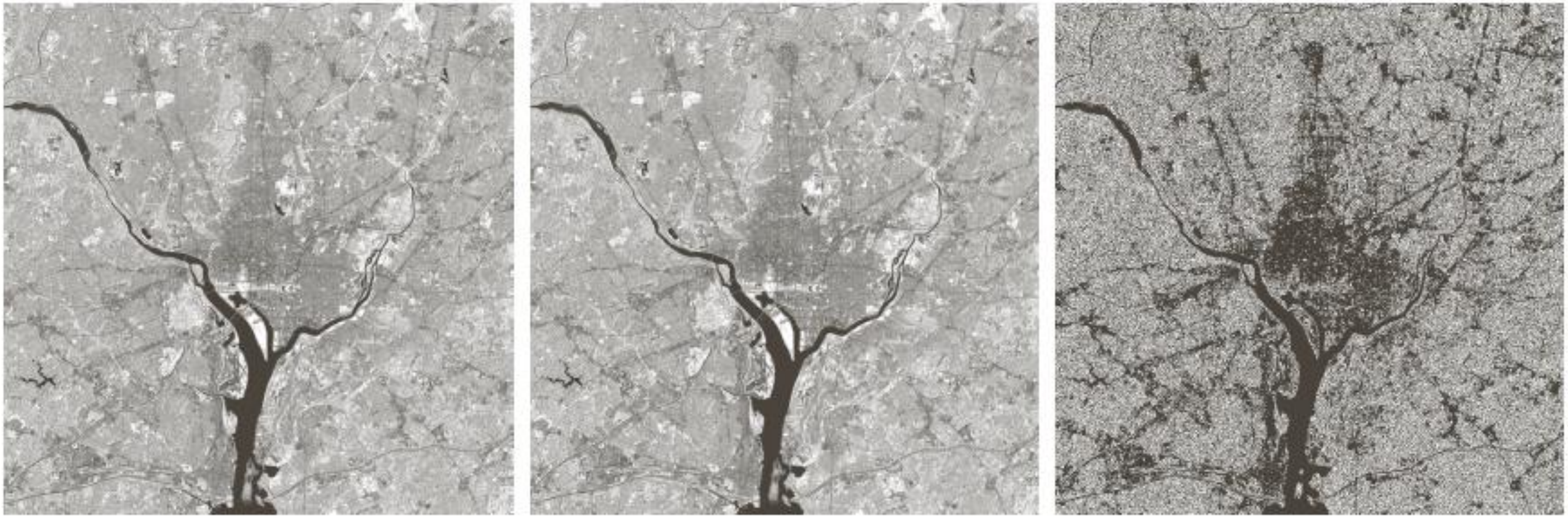



**FIGURE 2.26** (a) Image of Galaxy Pair NGC 3314 corrupted by additive Gaussian noise. (b)–(f) Results of averaging 5, 10, 20, 50, and 100 noisy images, respectively. (Original image courtesy of NASA.)

# Arithmetic Operations

- Subtraction (e.g. difference enhancement)



a b c

**FIGURE 2.27** (a) Infrared image of the Washington, D.C. area. (b) Image obtained by setting to zero the least significant bit of every pixel in (a). (c) Difference of the two images, scaled to the range [0, 255] for clarity.

# Arithmetic Operations

- Subtraction (e.g. mask mode radiography)

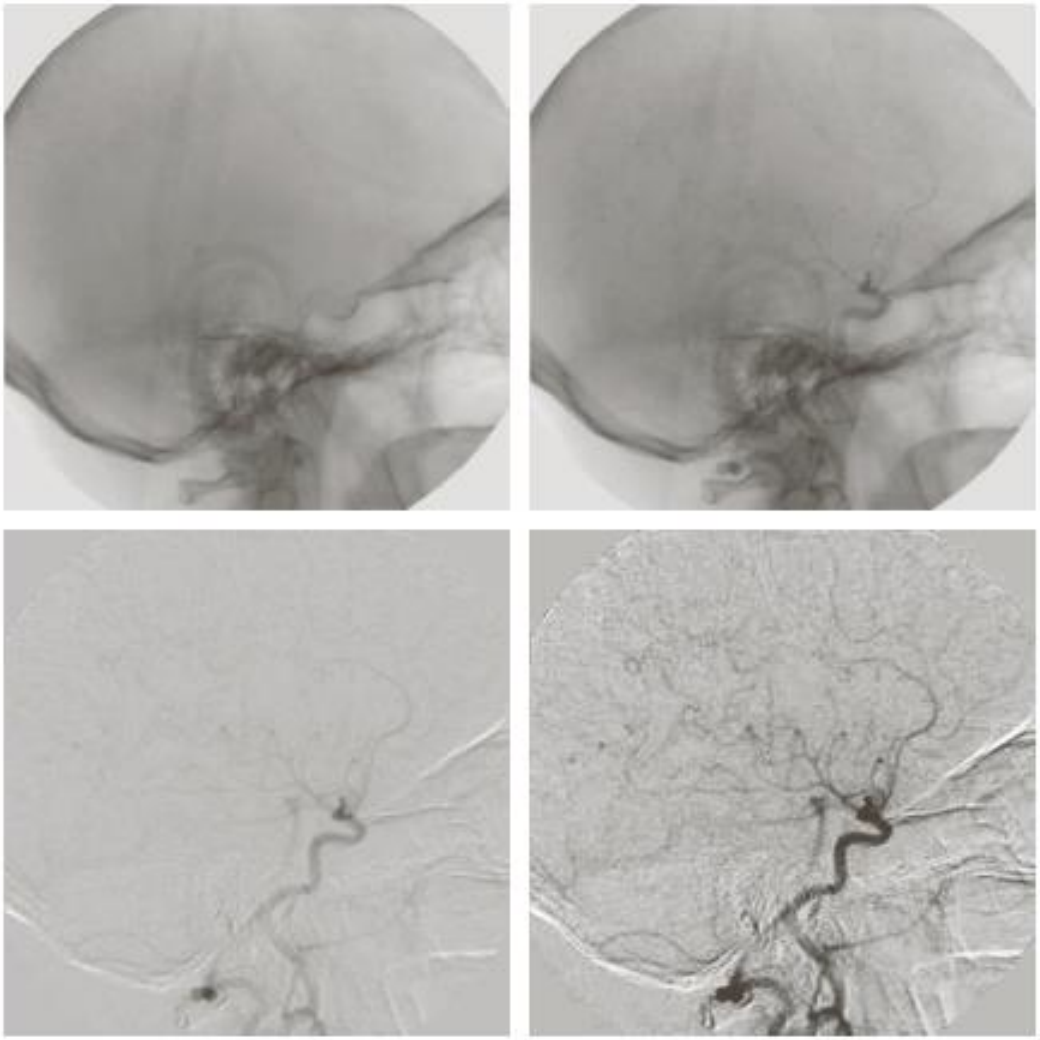



**FIGURE 2.28** Digital subtraction angiography. (a) Mask image. (b) A live image. (c) Difference between (a) and (b). (d) Enhanced difference image. (Figures (a) and (b) courtesy of The Image Sciences Institute, University Medical Center, Utrecht, The Netherlands.)

# Arithmetic Operations

- Multiplication & Division (e.g. shading correction)

a b c

**FIGURE 2.29** Shading correction. (a) Shaded SEM image of a tungsten filament and support, magnified approximately 130 times. (b) The shading pattern. (c) Product of (a) by the reciprocal of (b). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

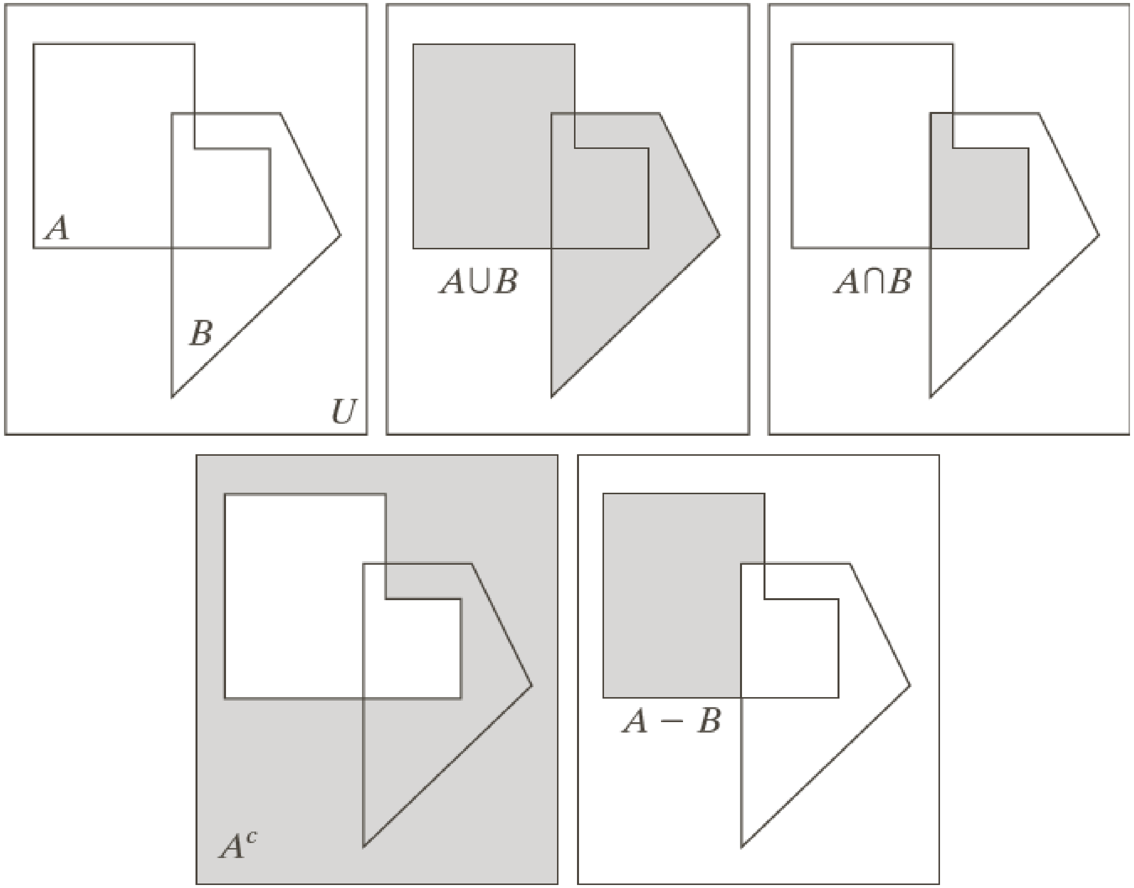
- 利用模版相乘做ROI提取操作

a b c

**FIGURE 2.30** (a) Digital dental X-ray image. (b) ROI mask for isolating teeth with fillings (white corresponds to 1 and black corresponds to 0). (c) Product of (a) and (b).

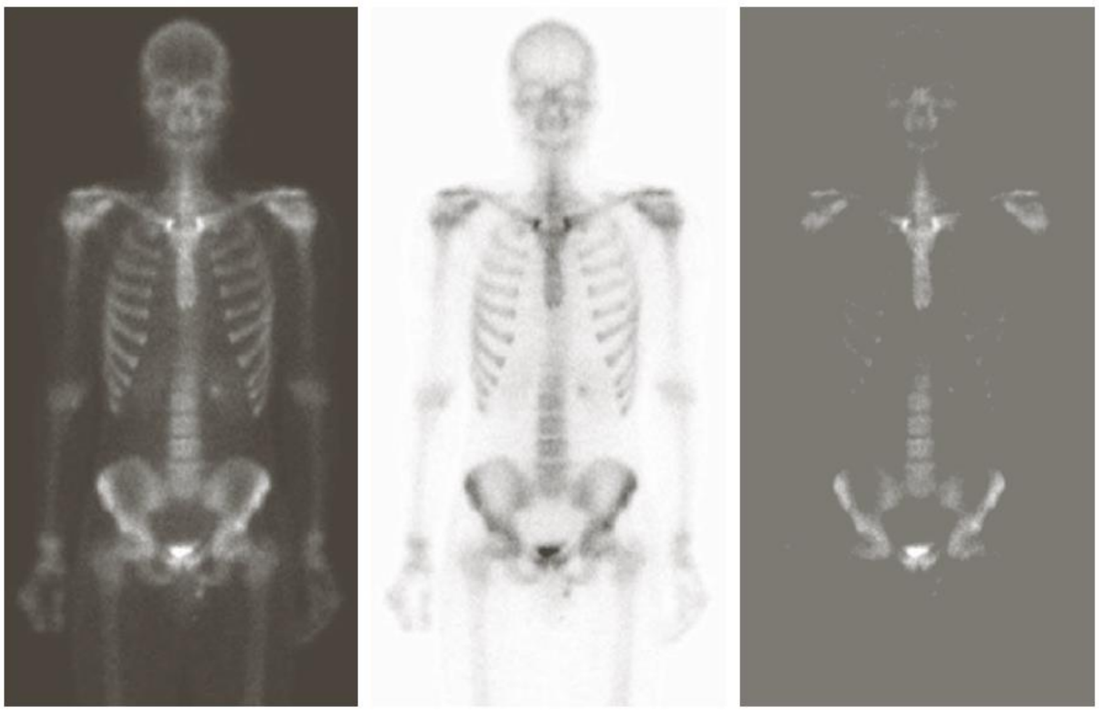
# 集合和逻辑操作

- 基本集合操作（以坐标为基础，图像二值）



- 灰度图像的集合操作
  - 由空间相应元素对间的最大灰度形成的阵列

$$A \cup B = \left\{ \max_{z}(a, b) \mid a \in A, b \in B \right\}$$

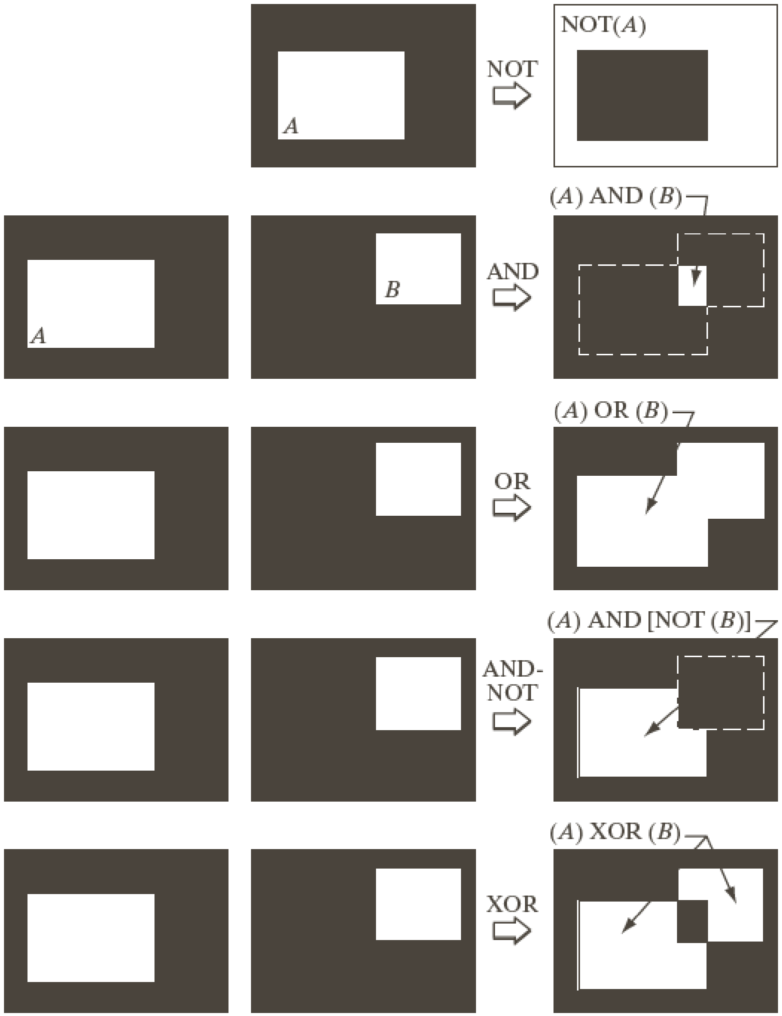


A　　$A^c$　　A U constant image

a b c

**FIGURE 2.32** Set operations involving gray-scale images. (a) Original image. (b) Image negative obtained using set complementation. (c) The union of (a) and a constant image. (Original image courtesy of G.E. Medical Systems.)

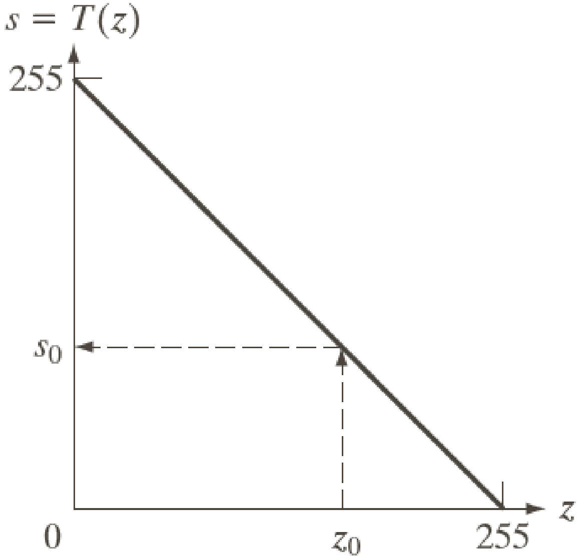
- 逻辑操作



# 2.6.5 空间操作

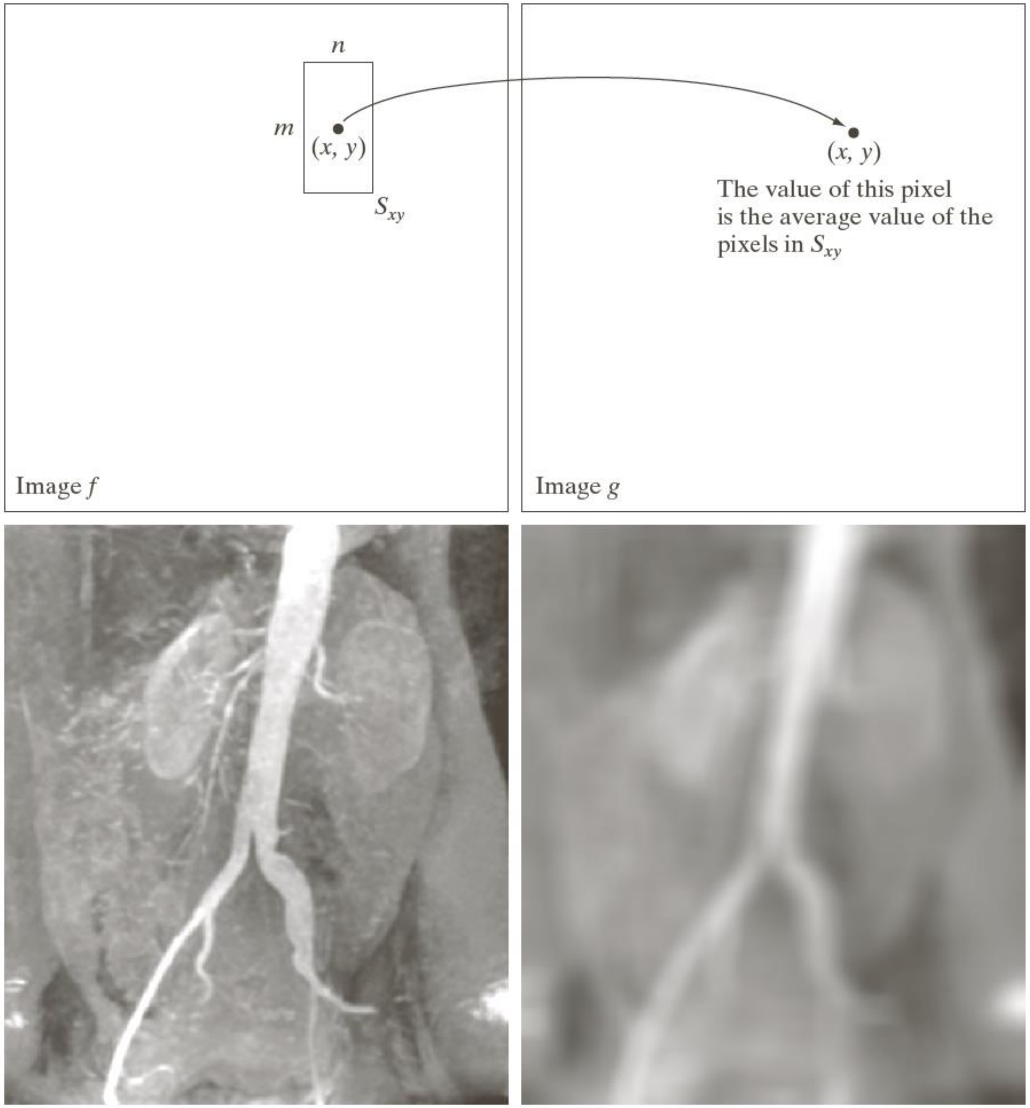
- 单像素操作

$$s = T(z)$$



- 邻域操作

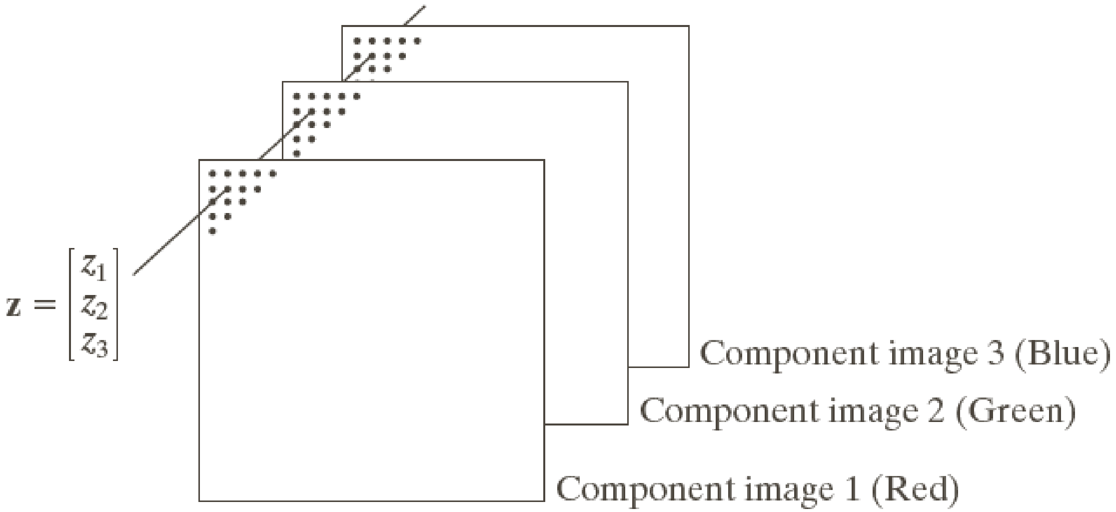
$$g(x,y) = \frac{1}{mn}\sum_{(r,c)\in S_{xy}} f(r,c)$$



# 2.6.6 向量与矩阵操作

- 1 彩色图像可以看做是三维向量
- 2. 整幅图像可以看做矩阵或者向量来处理。把尺寸为M×N的图像描述为一个MN×1维向量来处理，如线性处理：

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \mathbf{n}$$



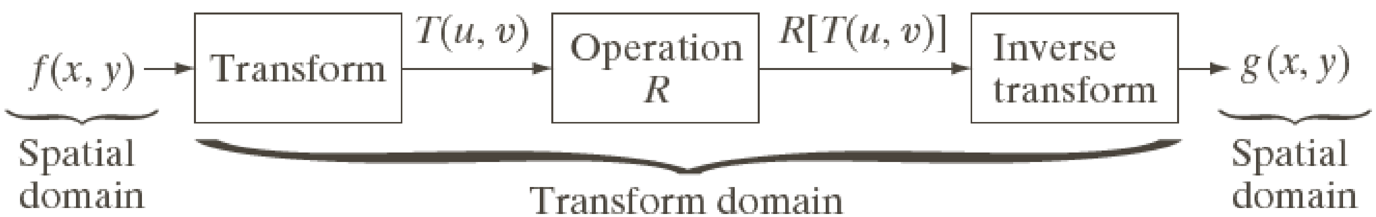
# 2.6.7 图像变换

- 很多时候，除了空间域，在图像的变换域做处理可能会更好。
- 正变换：

$$T(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)r(x,y,u,v)$$

- 反变换：

$$f(x,y)=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}T(u,v)s(x,y,u,v)$$

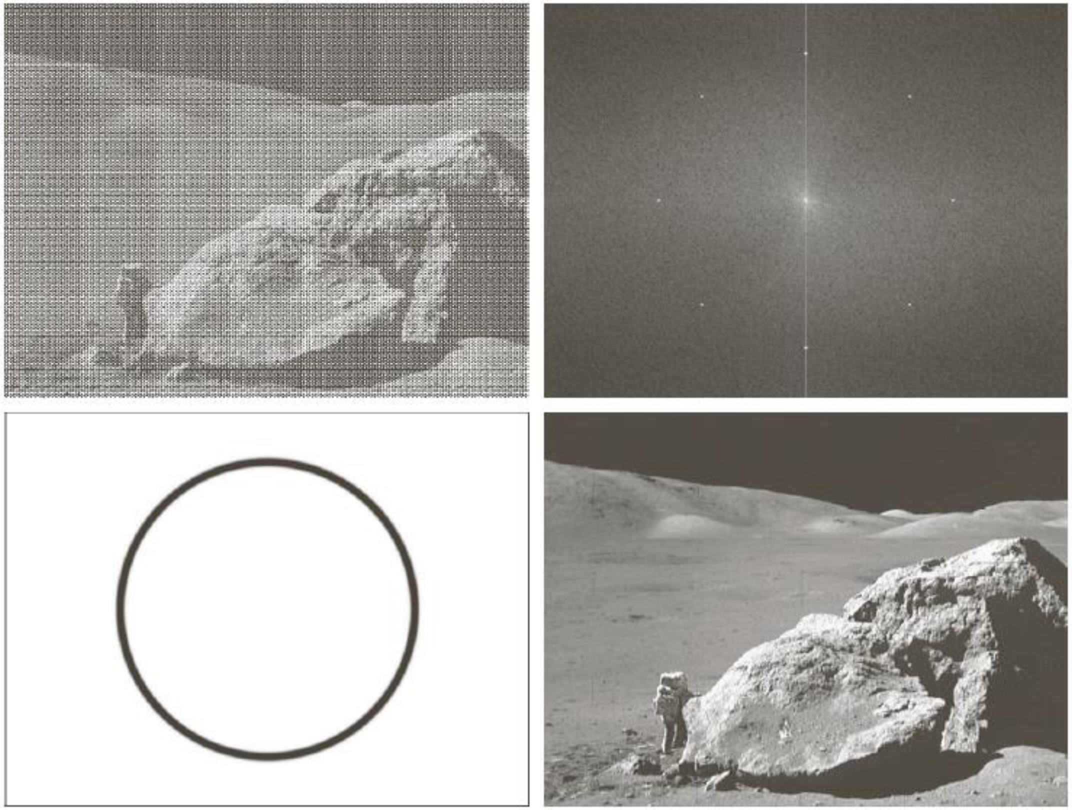


- 傅里叶变换

$$T(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)e^{-j2\pi(ux/M+vy/N)}$$

$$f(x,y)=\frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}T(u,v)e^{j2\pi(ux/M+vy/N)}$$

# 变换域图像处理



# 2.6.8 概率方法

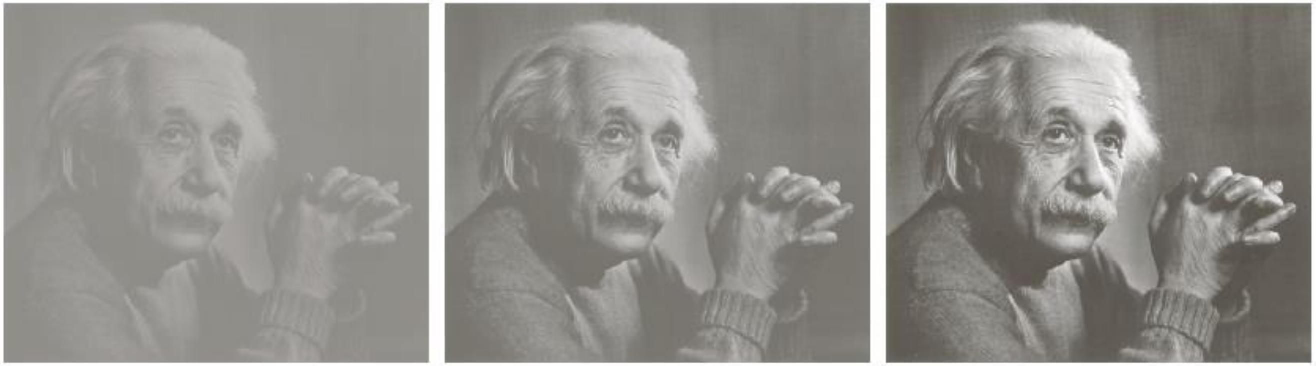
- 将图像看成为是一个随机场，可以获得很多相关的统计量。如：

$$p(z_k) = \frac{n_k}{MN}$$

$$m = \sum_{k=0}^{L-1} z_k p(z_k)$$

$$\sigma^2 = \sum_{k=0}^{L-1} (z_k - m)^2 p(z_k)$$

- 图像灰度标准差 ↔ 对比度



标准差分别为**14.3, 31.6, 49.2**

# Geometric Transformation

- 几何变换的目的
  - 几何失真 → 校正
  - 为了满足观测需要
    - 实现某种特殊效果
    - 信息隐藏
- 几何变换例子
  - 图像的缩放、旋转、平移和镜像；拼接、融合也常涉及图像的几何变换；立体电视的视点重构
- 几何变换一般不改变像素值，只是改变像素所在的位置

$$g(x,y)=f(x',y')=f[p(x,y),q(x,y)]$$

$(x',y')$ 可能不在图像的取样点阵上，这需要重采样

- 图像变形：同时改变像素位置和像素值

# 图像的仿射变换

- 仿射变换是对图像的最基本的几何变换

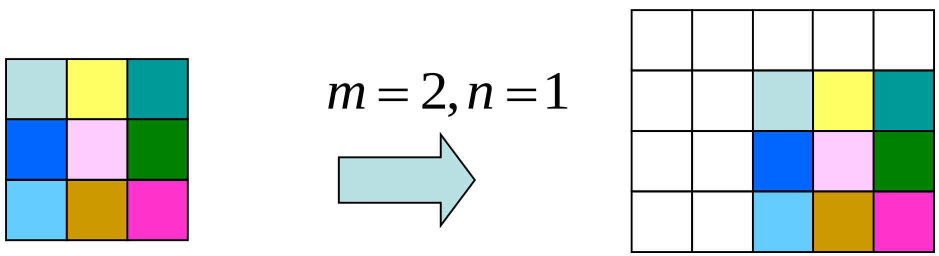
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} m \\ n \end{bmatrix} \pmod N$$

其中(x,y)是原图像中像素坐标，(x’,y’)是变换后的像素坐标。

- 坐标映射关系：线性（一次多项式）
- 平移、旋转、缩放等都是仿射变换

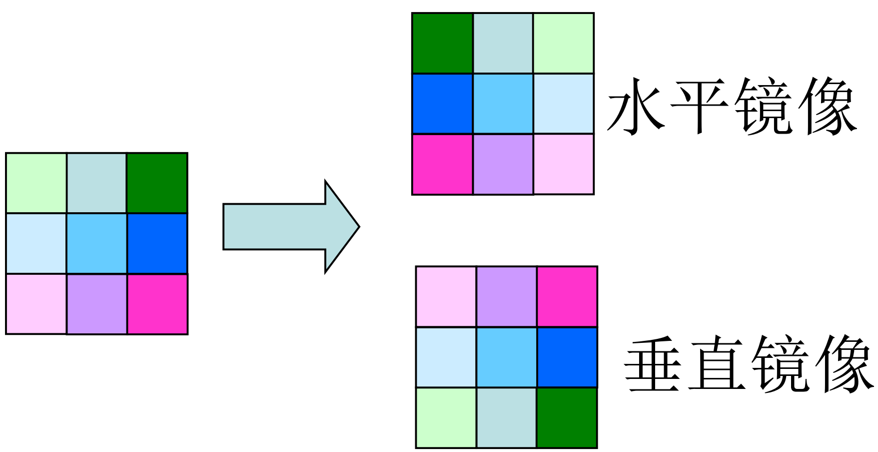
- **1. 图像的平移**

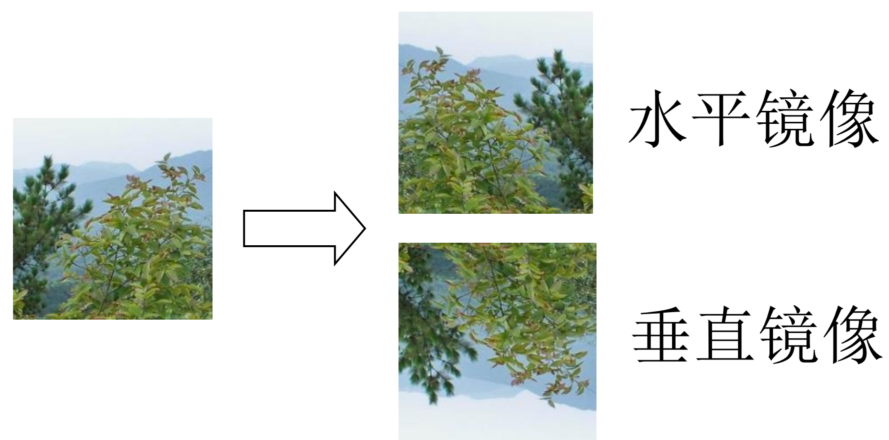
$$\begin{cases} x' = x + m \\ y' = y + n \end{cases}$$




平移后的景物与原图像相同

- **2. 图像的镜像**
  - 水平镜像 $\begin{cases} x' = -x \quad or \quad x' = W - 1 - x \\ y' = y \end{cases}$
  - 垂直镜像 $\begin{cases} x' = x \\ y' = -y \quad or \quad y' = H - 1 - y \end{cases}$



同时进行水平和垂直镜像则成为原点镜像

$$\begin{cases} x' = -x \\ y' = -y \end{cases}$$

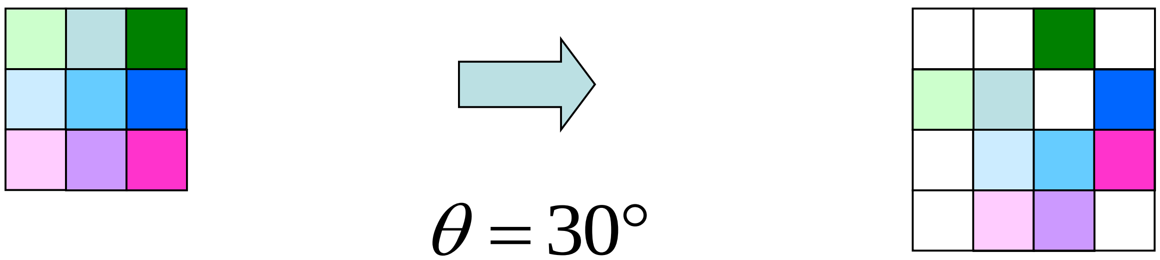
或

$$\begin{cases} x' = W - 1 - x \\ y' = H - 1 - y \end{cases}$$

也可对某个给定直线（垂直、水平）做镜像变换

- **3. 图像的旋转（以某点为原点）**

$$\begin{cases} x' = x\cos\theta + y\sin\theta \\ y' = -x\sin\theta + y\cos\theta \end{cases}$$

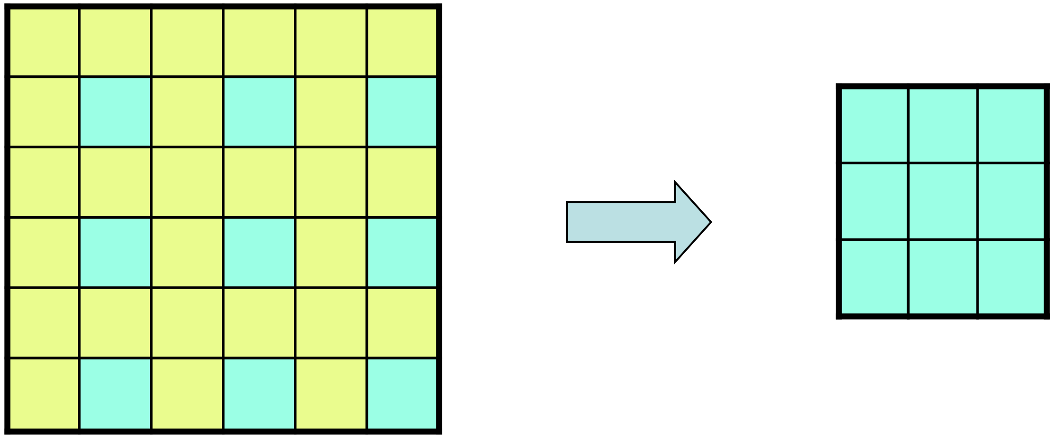



$$\begin{cases} x' = 0.866x + 0.5y \\ y' = -0.5x + 0.866y \end{cases}$$

# 4. 图像的缩小

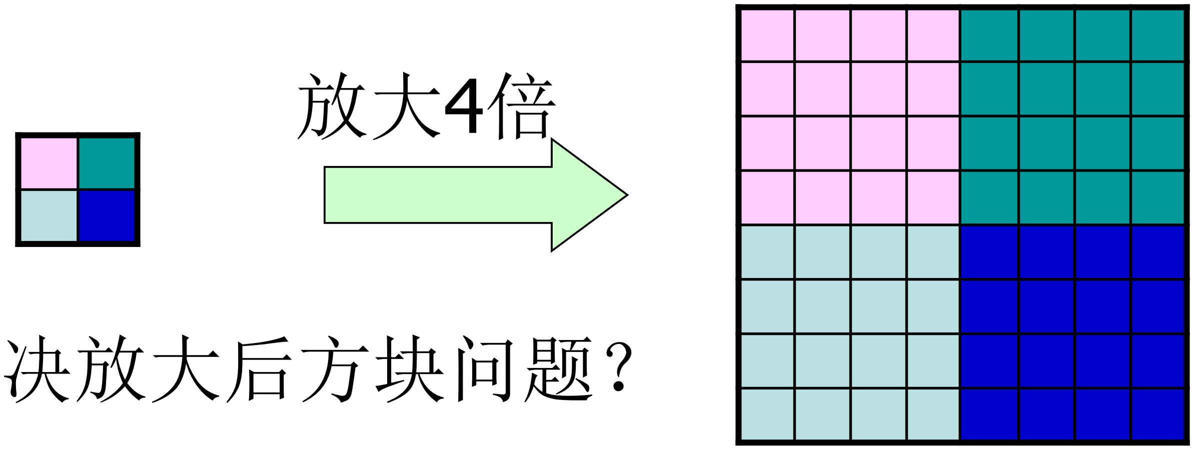
- 横向和纵向相同比例缩小
  - 整数倍缩小（如缩小一半）

    只需取原图的偶（奇）数行和偶（奇）数列构成新的图像



- 非整数倍缩小
  - 缩小后按原图上最近点取值
  - 插值后取值
- 横向和纵向不按比例缩小
  - 因x方向和y方向的缩小比例不同，会产生图像的几何畸变。
- 频谱混迭问题
  - 低通滤波后再缩小

- 5. **图像的放大**
  - **简单整数倍放大**
    - 如果需要将原图像放大k倍，则将一个像素值添在新图像的k*k的子块中




怎么解决放大后方块问题？

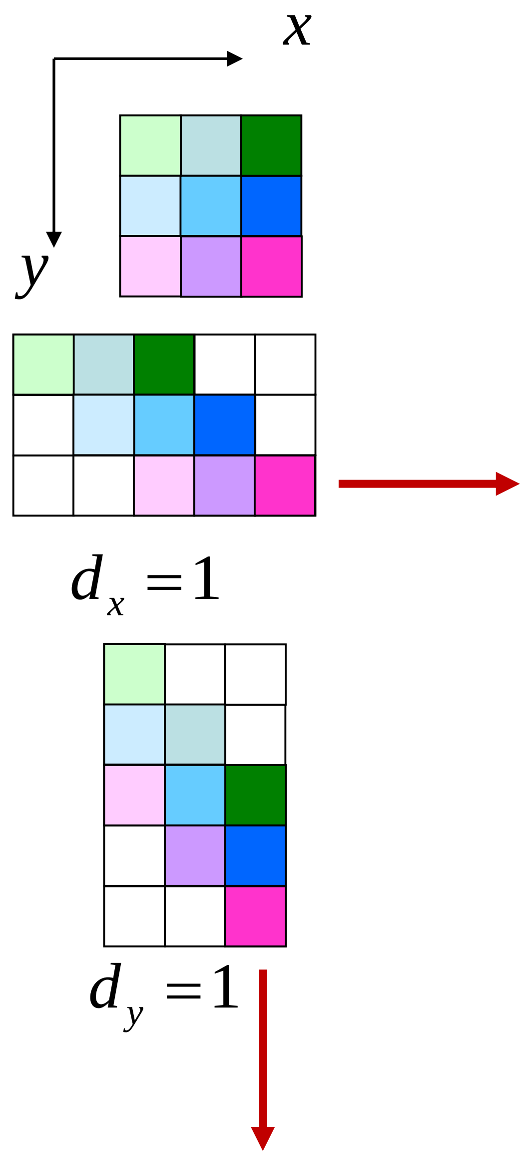
1)插值；2)放大后低通滤波

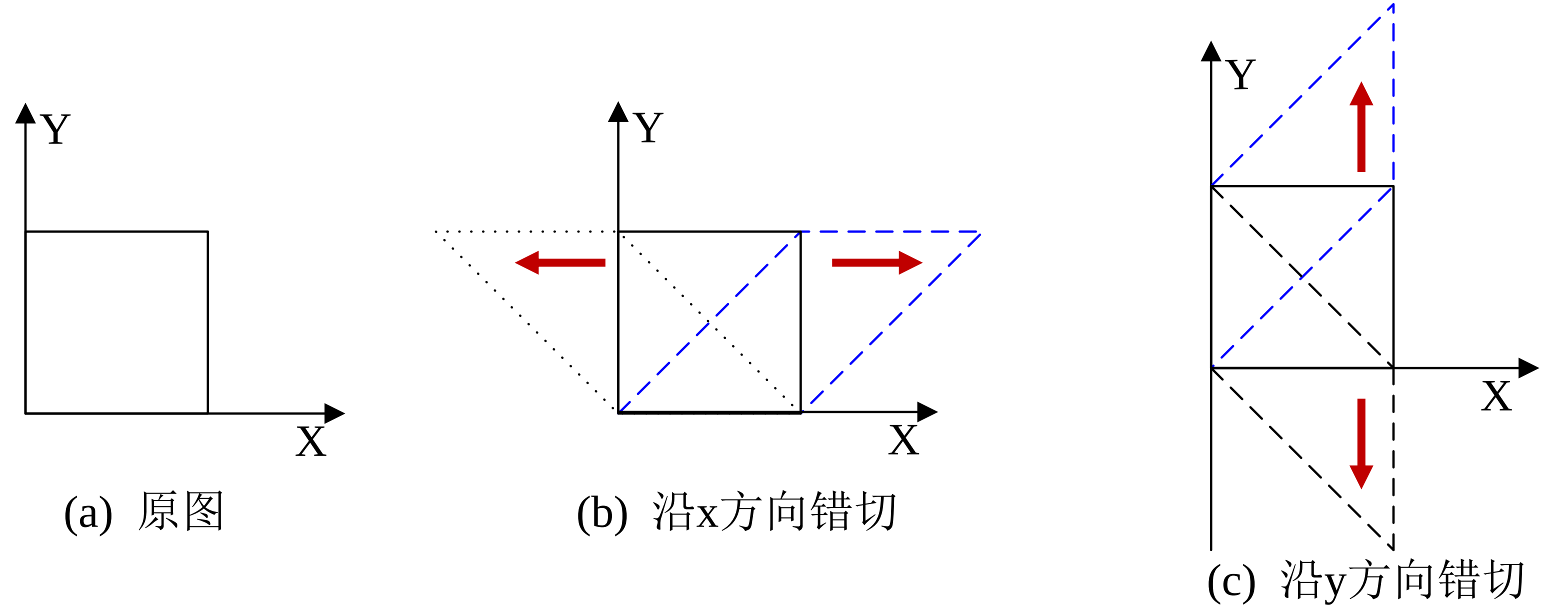
# • 6. 图像的错切变换

- 错切变换是指保持二维图像上各点的某一坐标值不变，而另一坐标值作关于该坐标值的线性变换。
  - 不变的坐标轴称为依赖轴，另一个改变的坐标轴称为方向轴
  - 错切变换也称为剪切、错位变换，动画设计中常用于产生弹性物体的变形
- 图像的错切变换可看作是景物在平面上的非垂直投影效果

$$\begin{cases} x' = x + d_x y \\ y' = y \end{cases} (x\text{方向的错切})$$

$$\begin{cases} x' = x \\ y' = y + d_y x \end{cases} (y\text{方向的错切})$$



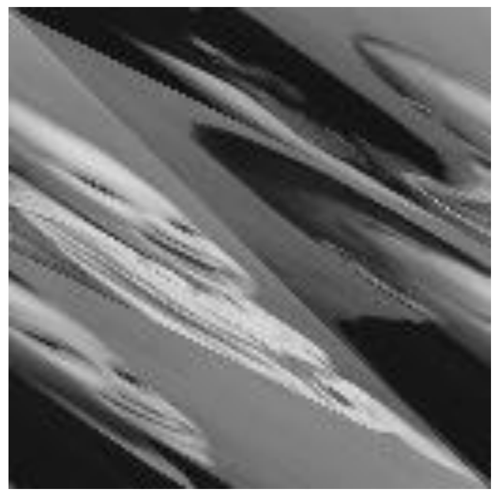
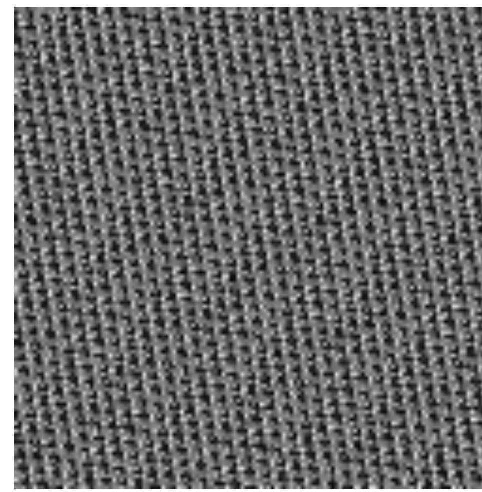



# 图像的错切变换

基于几何变换的图像像素置乱 $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \pmod N$

图像置乱（幻方变换、Hilbert变换等）：多次变换有图像加密效果

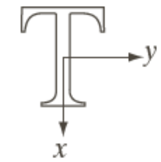
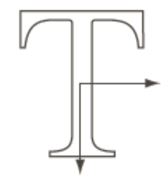
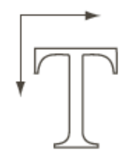
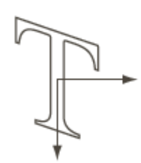
2维Arnold变换: $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \pmod N$ N×N是数字图像尺寸

Arnold变换具有周期性，即当迭代到某一步时，将重新得到原始图像

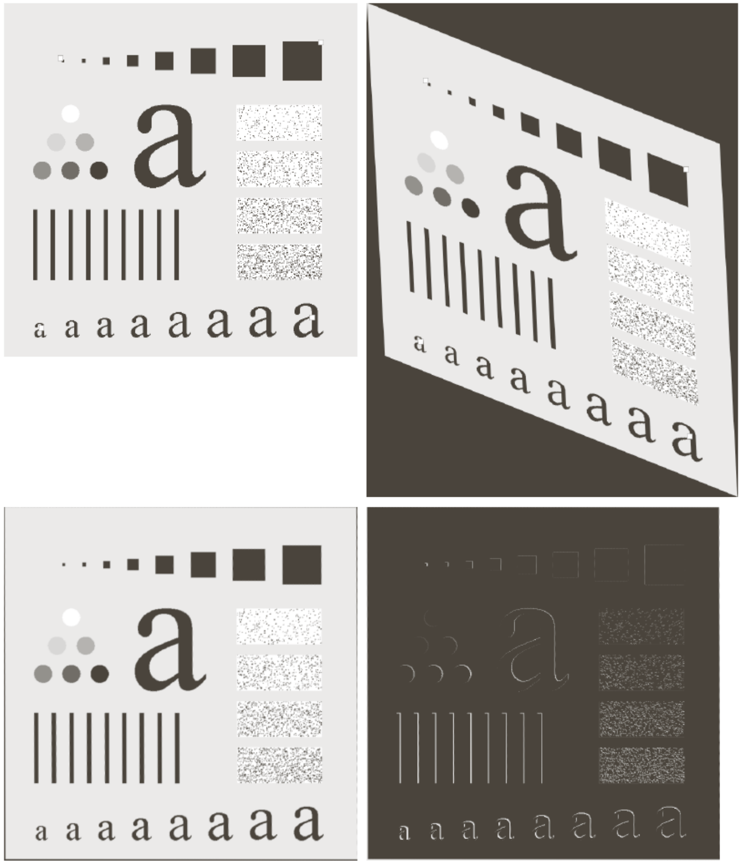
# 仿射变换

**TABLE 2.2**
Affine transformations based on Eq. (2.6.–23).

| Transformation Name | Affine Matrix, T | Coordinate Equations | Example |
|---|---|---|---|
| Identity | $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $x = v$<br>$y = w$ |  |
| Scaling | $\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $x = c_x v$<br>$y = c_y w$ |  |
| Rotation | $\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $x = v\cos\theta - w\sin\theta$<br>$y = v\cos\theta + w\sin\theta$ |  |
| Translation | $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$ | $x = v + t_x$<br>$y = w + t_y$ |  |
| Shear (vertical) | $\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $x = v + s_v w$<br>$y = w$ |  |
| Shear (horizontal) | $\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $x = v$<br>$y = s_h v + w$ |  |

# 基于仿射变换的图像配准

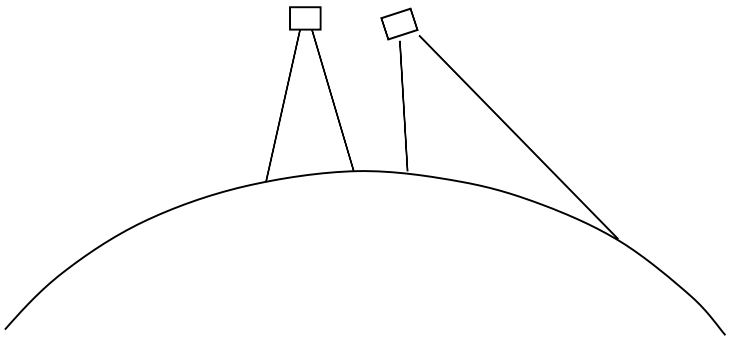
$$[x \ \ y \ \ 1] = [v \ \ w \ \ 1]\,\mathbf{T} = [v \ \ w \ \ 1]\begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix}$$



# 几何失真的校正

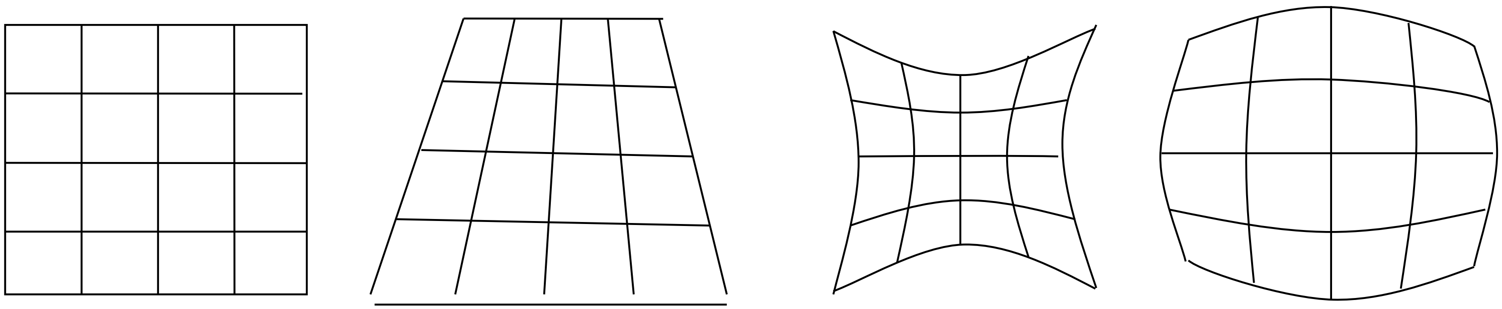
- 卫星遥感图像的几何失真校正
  - 由于地球是球形的、拍摄姿态等引起，不确定，属于非系统失真

    一般用一幅测绘好的基准图像去校正另一幅图像
  - 由于多光谱扫描镜线速不匀、检测器采样延迟造成各波段间不配准等，有规律的失真，属于系统失真



卫星遥感图像失真示意图

- 常见的几何失真



原图　　透视失真　　枕形失真　　桶形失真

- 几何校正的几个步骤

1. 选取控制点

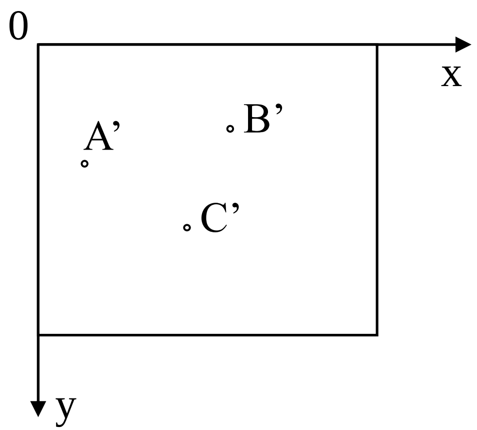
2. 确定图像空间坐标变换

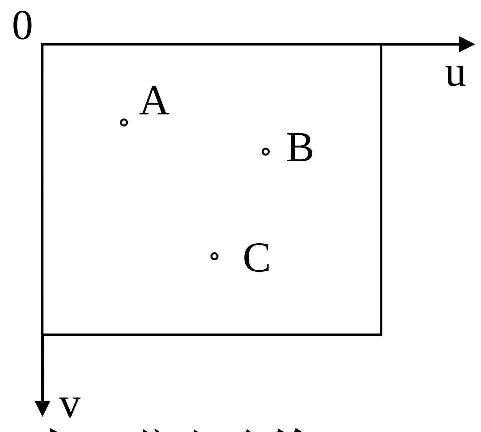
（包括求变换方程）

3. 确定在校正空间中的各像点的值

# 1. 选取控制点

一幅标准图像g(u,v) 和一幅几何失真图像f(x,y)，选取两幅图像上的对应点（控制点），使变换后被校正图像上的点几何位置与标准图像相同



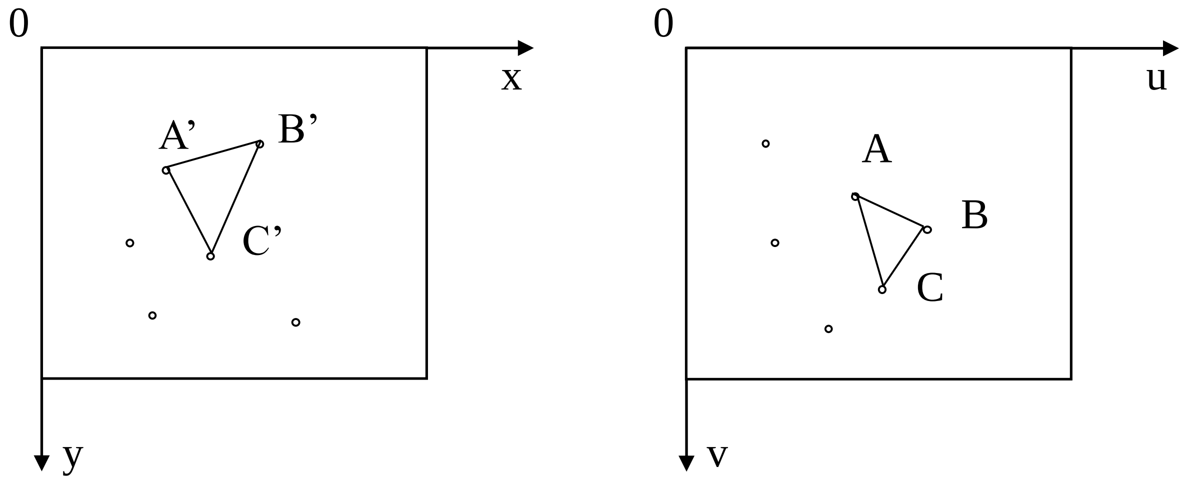

对自校正系统，需要自动检测匹配的控制点

选取地标为控制点

# 2. 空间几何坐标变换

利用两幅图像间控制点的对应关系，建立坐标间的函数关系

## (1) 三角形线性法




失真图像f(x,y)　　标准图像g(u,v)

- 在一个局部小区域内可认为是线性的

$$\begin{cases} x = au + bv + c \\ y = du + ev + f \end{cases}$$

- 将整幅图像分成一个个小三角形，对应三个顶点，得到三对控制点:

$$(x_1, y_1) \leftrightarrow (u_1, v_1) \quad (x_2, y_2) \leftrightarrow (u_2, v_2) \quad (x_3, y_3) \leftrightarrow (u_3, v_3)$$

即

$$\begin{cases} x_1 = au_1 + bv_1 + c \\ x_2 = au_2 + bv_2 + c \\ x_3 = au_3 + bv_3 + c \end{cases} \qquad \begin{cases} y_1 = du_1 + ev_1 + f \\ y_2 = du_2 + ev_2 + f \\ y_3 = du_3 + ev_3 + f \end{cases}$$

坐标变换一般采用**反向投影**方式。为什么？？？

$$(x, y) \Leftarrow (u, v)$$

# 算法的特点

- 计算简单
- 精度受三角形的选取影响

  三角形区域变小，则可提高精度，但计算量增加
- 一般要求控制点尽量均匀覆盖整个待校正区域，且控制点位置要找得准确
- 如果整幅图像的几何失真是线性的，则在整幅图像上只要取三个控制点

## (2) 二元多项式法

- 用一个二元n次多项式

$$
\left.\begin{aligned}
x &= \sum_{i=0}^{n}\sum_{j=0}^{n-i} a_{ij} u^i v^j \\
y &= \sum_{i=0}^{n}\sum_{j=0}^{n-i} b_{ij} u^i v^j
\end{aligned}\right\}
$$

- 通常可取 n=2

$$
\left.\begin{aligned}
x &= a_{00} + a_{01}v + a_{02}v^2 + a_{10}u + a_{11}uv + a_{20}u^2 \\
y &= b_{00} + b_{01}v + b_{02}v^2 + b_{10}u + b_{11}uv + b_{20}u^2
\end{aligned}\right\}
$$

# 对于共L对控制点，采用最小二乘方准则

$$\varepsilon=\sum_{l=1}^{L}\left(x_l-\sum_{i=0}^{n}\sum_{j=0}^{n-i}a_{ij}u_l{}^i v_l{}^j\right)^2$$

- 为使ε最小，对每个系数 $\dfrac{\partial\varepsilon}{\partial a_{st}}=0$

$a_{st}, s=0,1,\ldots,n; t=0,1,\ldots,n-s$

– 得 $$\sum_{l=1}^{L}\left(\sum_{i=0}^{n}\sum_{j=0}^{n-i}a_{ij}u_l{}^i v_l{}^j\right)u_l{}^s v_l{}^t=\sum_{l=1}^{L}x_l u_l{}^s v_l{}^t$$

$$s=0,1,\ldots,n; t=0,1,\ldots,n-s$$

– 类似地 $$\sum_{l=1}^{L}\left(\sum_{i=0}^{n}\sum_{j=0}^{n-i}b_{ij}u_l{}^i v_l{}^j\right)u_l{}^s v_l{}^t=\sum_{l=1}^{L}y_l u_l{}^s v_l{}^t$$

- 对于不同的s,t，对应有M=(n+1)(n+2)/2 个系数 $a_{st}$，组成有M个方程的方程组
- 同样，对应 $b_{st}$ 也有M个方程的方程组
- 解上面两方程组，则可求得系数 $a_{ij}$ 和 $b_{ij}$

• 这两个方程组也可写成矩阵的形式 $\mathbf{T}\vec{a} = \vec{X}$

T 是由 $u_l, v_l$ 构成的矩阵，$\vec{a}$ 是系数 $a_{ij}$ 组成的列矢量，

$\vec{X}$ 是由 $x_l$ 和 $u_l, v_l$ 构成的列矢量

$b_{ij}$ 也有类似的表达式

– 精度与多项式次数n有关，次数越高，则拟合后误差越小，但方程数目M也随之增加，也即必须大大增加控制点对数，才能得解。

多项式次数n越高越好吗？

– 通常取 n=2，L≥9

– 一种简化：

$$\left.\begin{aligned} x &= a_{00} + a_{01}v + a_{10}u + a_{11}uv \\ y &= b_{00} + b_{01}v + b_{10}u + b_{11}uv \end{aligned}\right\}$$

- 两个方程组都有4个未知系数，用4对一般的（不要三点成一直线）控制点就可得解
- 可把图像用控制点对分成一个个四边形区域

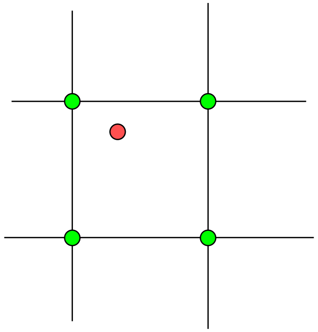
# 3. 像素值的确定

## (1) 最近邻点法

标准空间中点(u,v)，其灰度应是点(x,y)的灰度

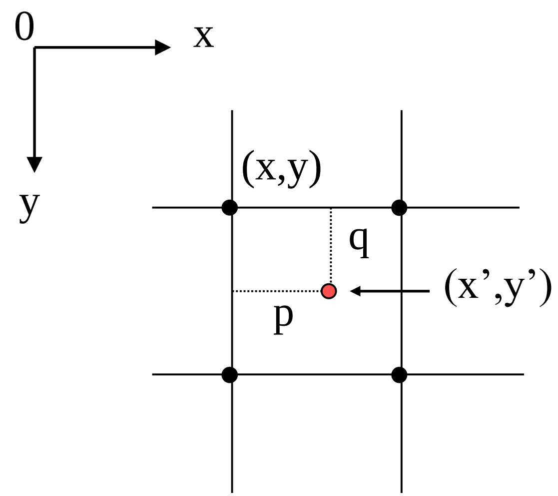
但(x,y)一般不在栅格上，取它周围最近的邻点的灰度作为点(u,v)的灰度

- 特点：
  - –计算简单
  - –校正后的图像会有亮度的明显不连续性



## (2) 双线性内插法

$$f(x+p,y+q)=(1-p)(1-q)f(x,y)+(1-p)qf(x,y+1)+$$
$$p(1-q)f(x+1,y)+pqf(x+1,y+1)$$

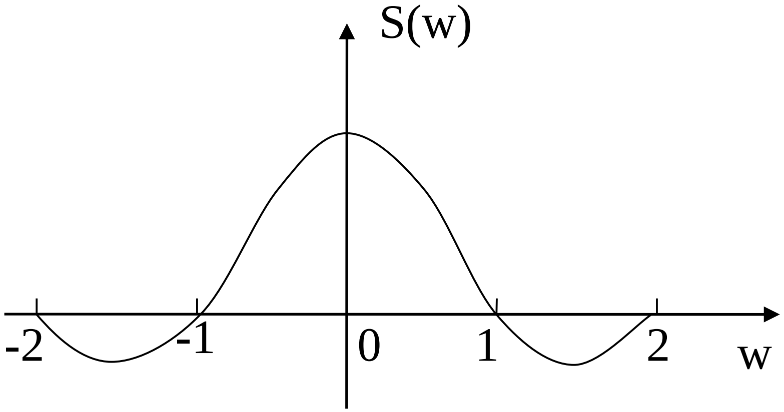



双线性内插有低通效应，会引起边缘模糊

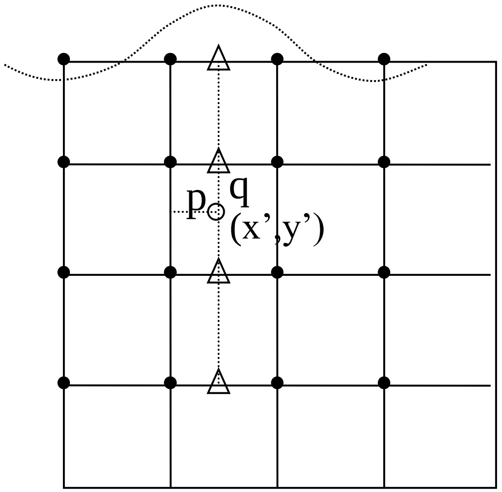
## (3) 三次卷积法

也叫**立方样条插值**，用三次多项式S(w)来内插

$$S(w)=\begin{cases} 1-2|w|^2+|w|^3 & |w|<1 \\ 4-8|w|+5|w|^2-|w|^3 & 1\le |w|\le 2 \\ 0 & |w|\ge 2 \end{cases}$$




立方样条函数




立方样条内插

用矩阵表示

$$f(x',y') = \mathbf{A}^T\mathbf{B}\mathbf{C}$$

其中（权重归一化问题？可以证明）

$$\mathbf{A} = \begin{bmatrix} S(1+q) & S(q) & S(1-q) & S(2-q) \end{bmatrix}^T$$

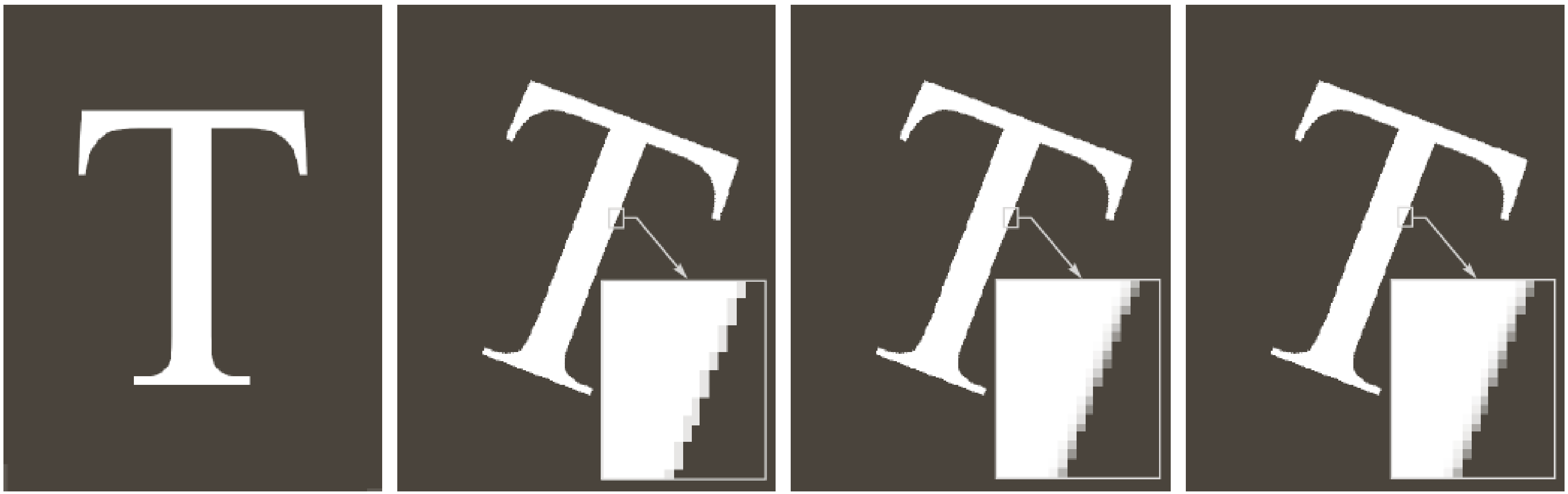
$$\mathbf{C} = \begin{bmatrix} S(1+p) & S(p) & S(1-p) & S(2-p) \end{bmatrix}^T$$

$$\mathbf{B} = \begin{bmatrix} f(y-1,x-1) & f(y-1,x) & f(y-1,x+1) & f(y-1,x+2) \\ f(y,x-1) & f(y,x) & f(y,x+1) & f(y,x+2) \\ f(y+1,x-1) & f(y+1,x) & f(y+1,x+1) & f(y+1,x+2) \\ f(y+2,x-1) & f(y+2,x) & f(y+2,x+1) & f(y+2,x+2) \end{bmatrix}$$

Nearest neighbor Bilinear Bicubic

**FIGURE 2.24** (a) Image reduced to 72 dpi and zoomed back to its original size (3692 × 2812 pixels) using nearest neighbor interpolation. This figure is the same as Fig. 2.20(d). (b) Image shrunk and zoomed using bilinear interpolation. (c) Same as (b) but using bicubic interpolation. (d)–(f) Same sequence, but shrinking down to 150 dpi instead of 72 dpi [Fig. 2.24(d) is the same as Fig. 2.20(c)]. Compare Figs. 2.24(e) and (f), especially the latter, with the original image in Fig. 2.20(a).

• 图像旋转与灰度内插



a b c d

Nearest neighbor Bilinear Bicubic

**FIGURE 2.36** (a) A 300 dpi image of the letter T. (b) Image rotated 21° clockwise using nearest neighbor interpolation to assign intensity values to the spatially transformed pixels. (c) Image rotated 21° using bilinear interpolation. (d) Image rotated 21° using bicubic interpolation. The enlarged sections show edge detail for the three interpolation approaches.

# 图像变形

- 图像变形指景物的形体变化，它是使一幅图像逐步变化到另一幅图像的处理方法
  - 同时发生坐标和像素值的变化
  - 图像变形用二幅图像生成一个视频序列
- 变形的起始图像和结束图像分别为两幅关键帧，从起始形状变化到结束形状的关键在于自动地生成中间形状，也即自动生成中间帧

# 渐隐（dissolve）

渐隐：静态变换，淡入，淡出

– 大小相同的两幅图，从图a逐渐变化成图b

– 原理：让图a中每个像素的颜色，逐渐变成图b相同位置像素的颜色

– 方法：根据变换的快慢，设置相应的步长，将图a每一像素的值经过N帧图像后，逐渐变成图b相同位置像素的值。可以选择等比或等差的方式，或其它方式

- 对于彩色图像

RGB可同时从原图变到目标图像，可分别变化，也可考虑RGB的相对比例关系同时变化

$$r_{i,j} = r_{i,j}^{\mathrm{a}} + \frac{r_{i,j}^{\mathrm{b}} - r_{i,j}^{\mathrm{a}}}{N} * n \ldots\ldots\ldots n = 0 \ldots N-1$$

$$g_{i,j} = g_{i,j}^{\mathrm{a}} + \frac{g_{i,j}^{\mathrm{b}} - g_{i,j}^{\mathrm{a}}}{N} * n \ldots\ldots\ldots n = 0 \ldots N-1$$

$$b_{i,j} = b_{i,j}^{\mathrm{a}} + \frac{b_{i,j}^{\mathrm{b}} - b_{i,j}^{\mathrm{a}}}{N} * n \ldots\ldots\ldots n = 0 \ldots N-1$$

# Dissolve Examples

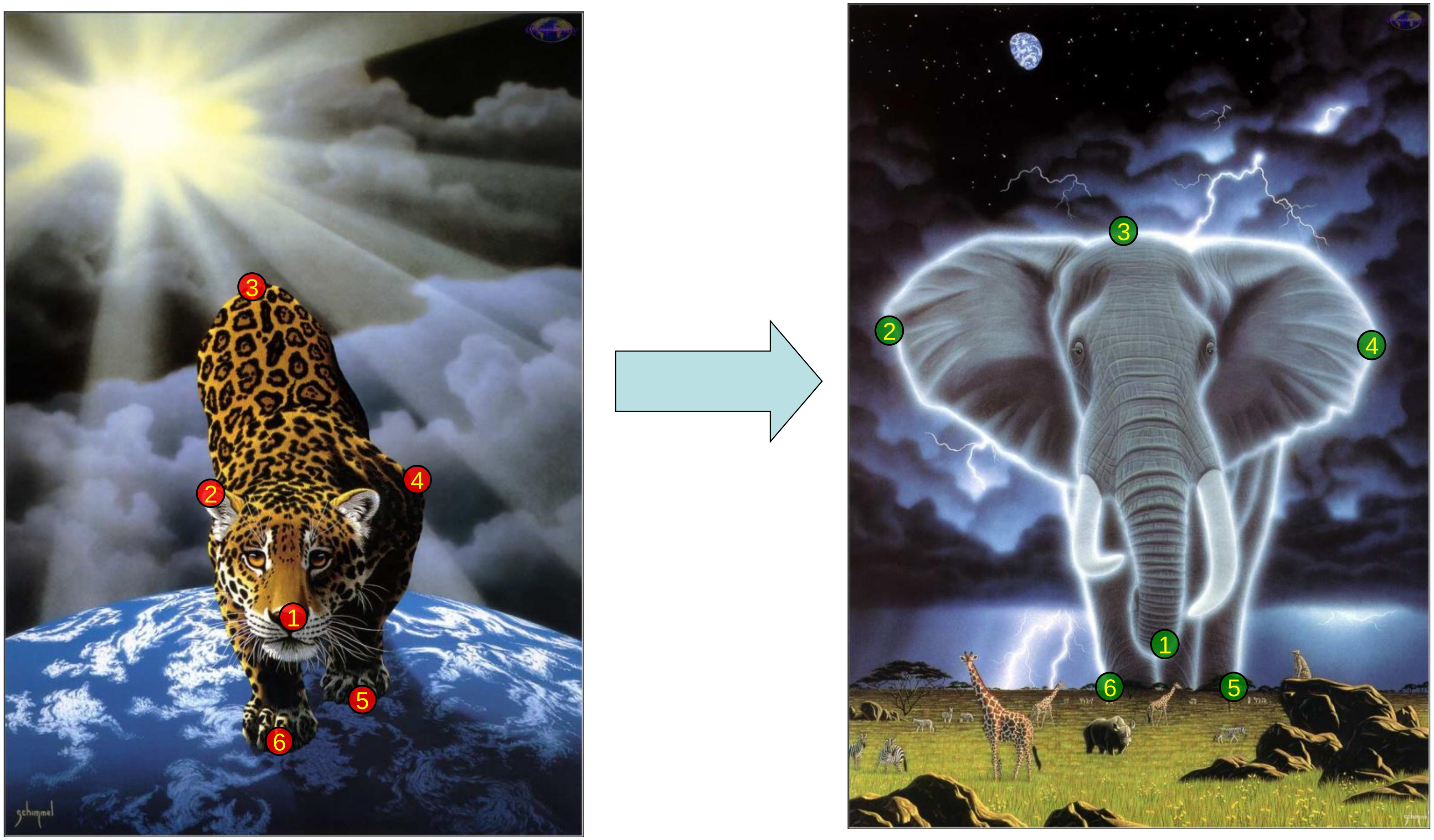
# 图像变形（Image Morphing）

- 先在起始帧和结束帧上确定结构对应关系，也即从起始画面上的一个点变到结束画面上的另一个对应点的位置（即控制点，这是变形运算所需要的参数）
- 确定中间帧帧数
- 确定控制点在中间帧上的移动方式
  - 当前帧上的每个点作向着结束点方向的步进运动，步进长度为移动距离除以中间帧数，以求出下一帧对应点的位置及颜色，并对其它相邻点作插值处理
  - 可定义不同的动画效果

# 图像变形（Image Morphing）

– 确定像素值在中间帧上的变化方式

采用几何校正技术，确定中间帧上的像素对应到起始帧和结束帧上的坐标位置，从而确定其像素值（利用内插后的像素值，像素值在中间帧的变化规律）


3
2
4
1
5
6
3
2
4
1
6
5

# Image Morphing

# Assignments

- 2.8, 2.14, 2.16, 2.18, 2.22, 2.26, 2.28, 2.37

  一定要对照英文版题目，帮助理解