

第2章 数字图像基础

§ 2.1 视觉感知要素

§ 2.2 光和电磁波谱

§ 2.3 图像感知和获取

§ 2.4 图像取样和量化

§ 2.5 像素间的基本关系

§ 2.6 线性和非线性操作(略)



§ 2.3 图像感知和获取

§ 2.3.1 用单个传感器获取图像

§ 2.3.2 用带状传感器获取图像

§ 2.3.3 用传感器阵列获取图像

§ 2.3.4 图像灰度级

§ 2.3 图像感知和获取

§ 2.3.1 用单个传感器获取图像

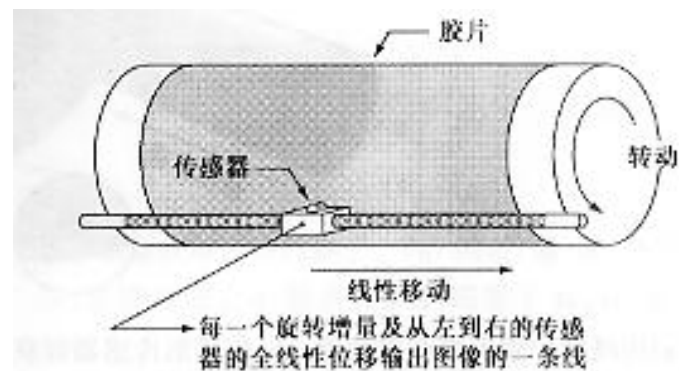
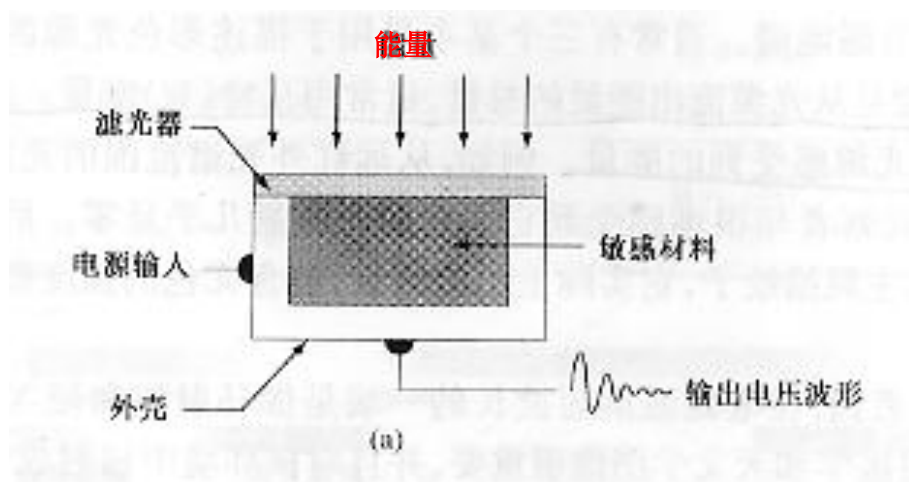
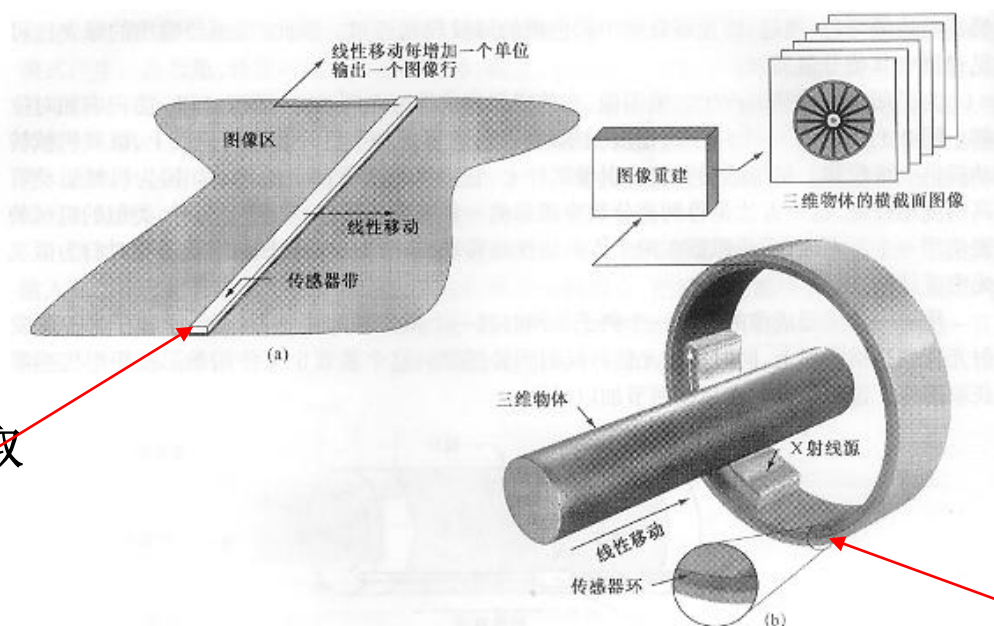
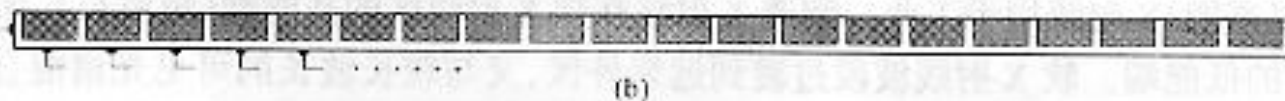


图 2.13 单个传感器组通过运动产生二维图像

单个传感器如光二极管；前面的滤光器改善选择性能；需要通过x和y两个方向的位移才能产生二维图像。

§ 2.3.2 用带状传感器获取图像

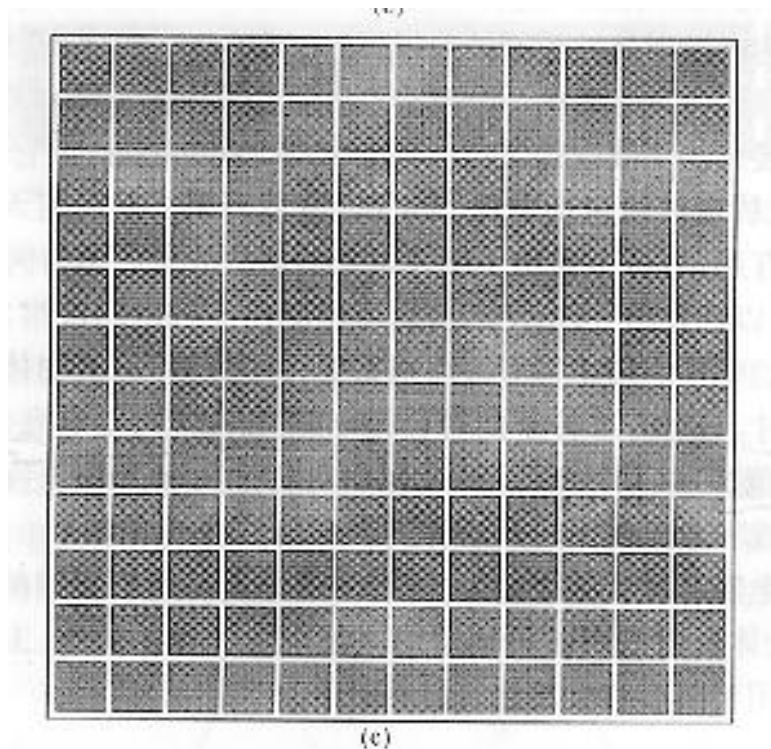


传感器带一次获取
图像的一行；

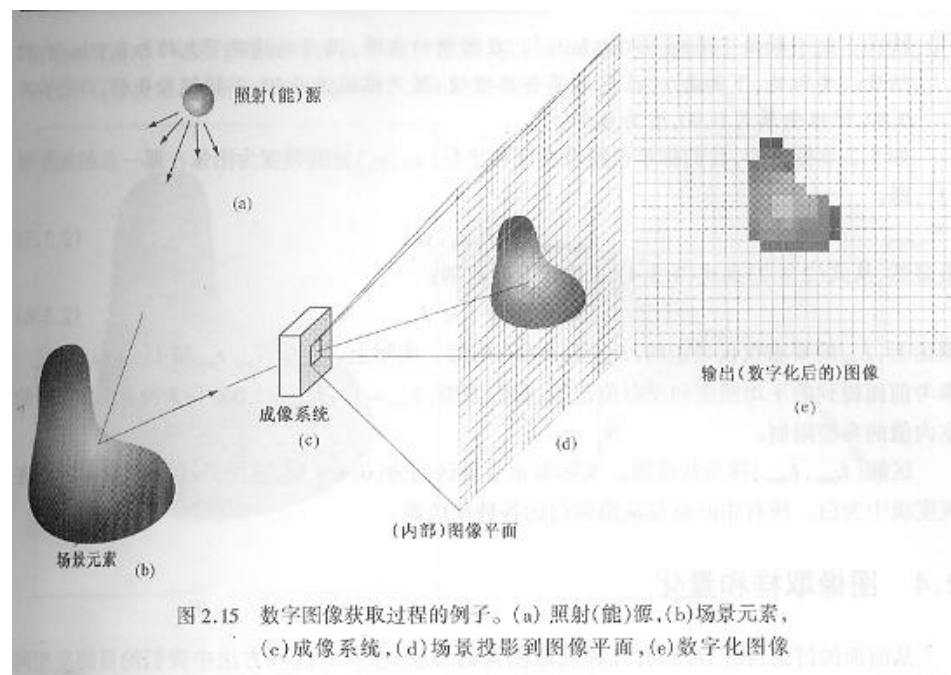
圆环状传感器带
一次给出一个横
切面图像；

图 2.14 (a)用线性传感器带获取图像, (b)用环形传感器带获取图像

§ 2.3.3 用传感器阵列获取图像



传感器阵列



CCD传感器阵列：数码相机

§ 2.3.4 图像灰度级

单色图像视觉模型已在 § 2.2.3 小节中介绍。

图像值称为**灰度级**，正比于物理源(电磁波)的**辐射能量**。

由图像的物理产生过程可知，图像值表示为一个两分量的函数，入射分量和反射分量。**图像值是非负和有限的。**

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

单色图像在任意坐标处的强度，就是图像在该点的灰度级 l ，满足 $L_{\min} < l < L_{\max}$ 。

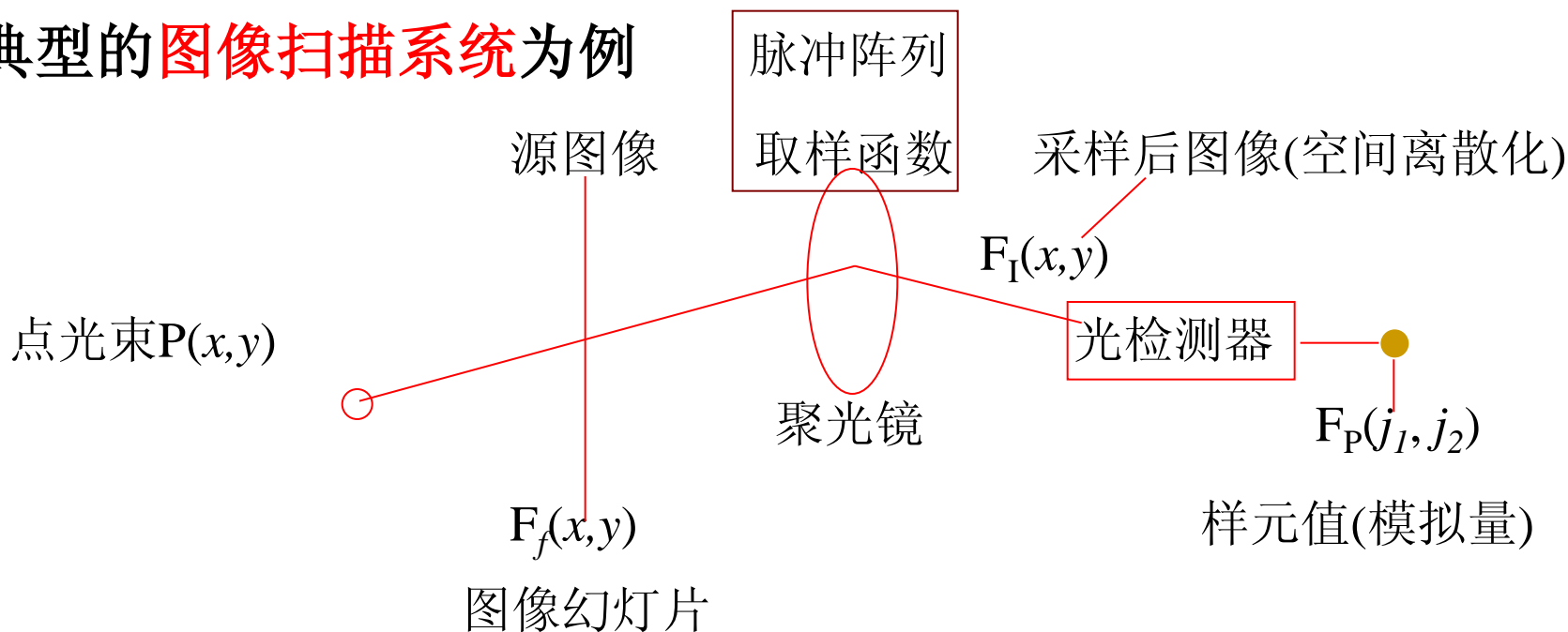
§ 2.4 图像取样和量化

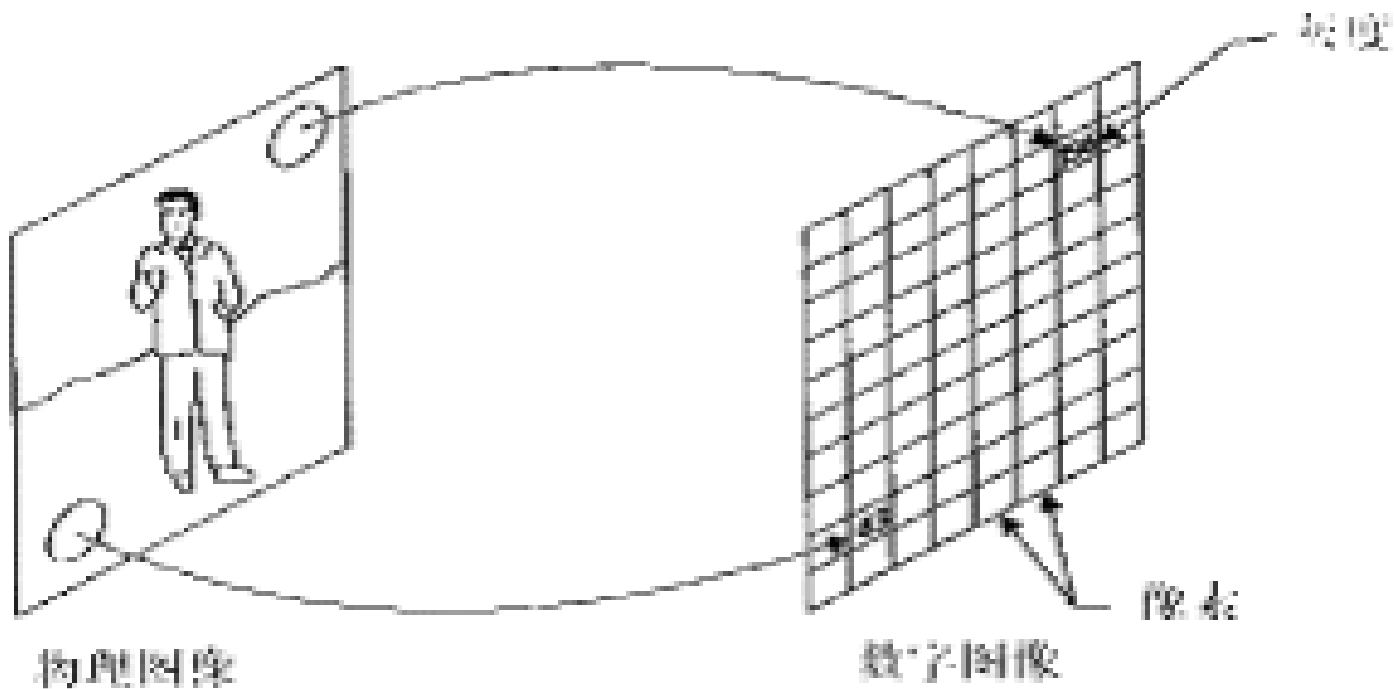
空间坐标的离散化叫做空间采样；

性质空间(灰度)的离散化叫做量化；

采样和量化主要在图像输入设备中处理。

以典型的**图像扫描系统**为例





图像信号的数字化过程和结果:

$$f(x, y) \rightarrow \text{空间离散} \rightarrow \text{幅值量化} \rightarrow \begin{bmatrix} f(0,0) & f(0,1) & L & f(0,N-1) \\ f(1,0) & f(1,1) & L & f(1,N-1) \\ L & L & L & L \\ f(M-1,0) & f(M-1,1) & L & f(M-1,N-1) \end{bmatrix}$$

§ 2.4.1 采样和量化的基本概念

§ 2.4.1.1 采样过程

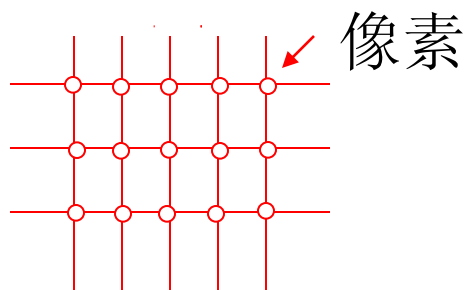
用点光束扫描透明图像片，然后由光电检测器(有一定空间分辨率)检测出样元值 $F_P(j_1, j_2)$ ；取样的样元值是模拟量表示的幅值，采样过程仅把连续图像在空间上离散化。

设 $F_f(x, y)$ 为一连续图像， $S(x, y)$ 为空间取样函数(δ 脉冲阵列)，则采样后的图像为：(应满足奈奎斯特Nyquist准则)

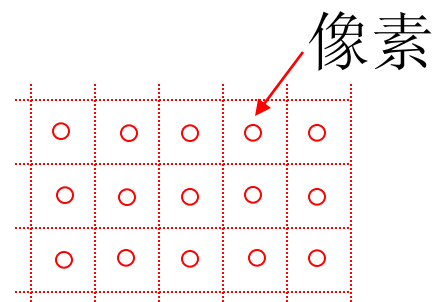
$$F_I(x, y) = F_f(x, y) S(x, y);$$

图像尺寸为 $M \times N$ 。

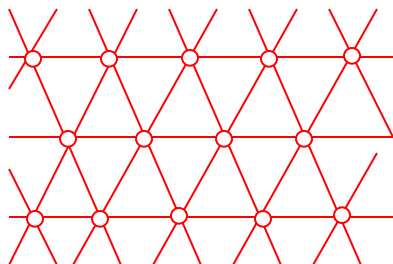
采样阵列： δ 脉冲阵列



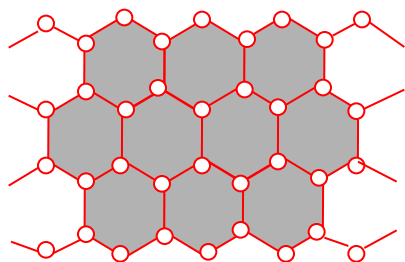
正方形点阵



正方形阵列

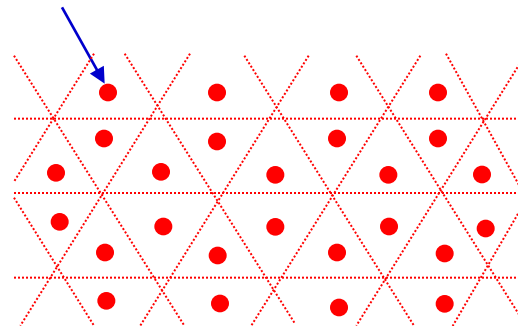


正三角形点阵

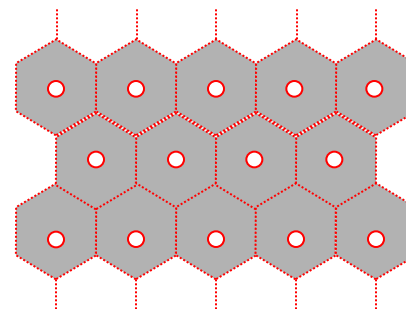


正六角形点阵

像素



正三角形阵列



正六角形阵列

§ 2.4.1 采样和量化的基本概念

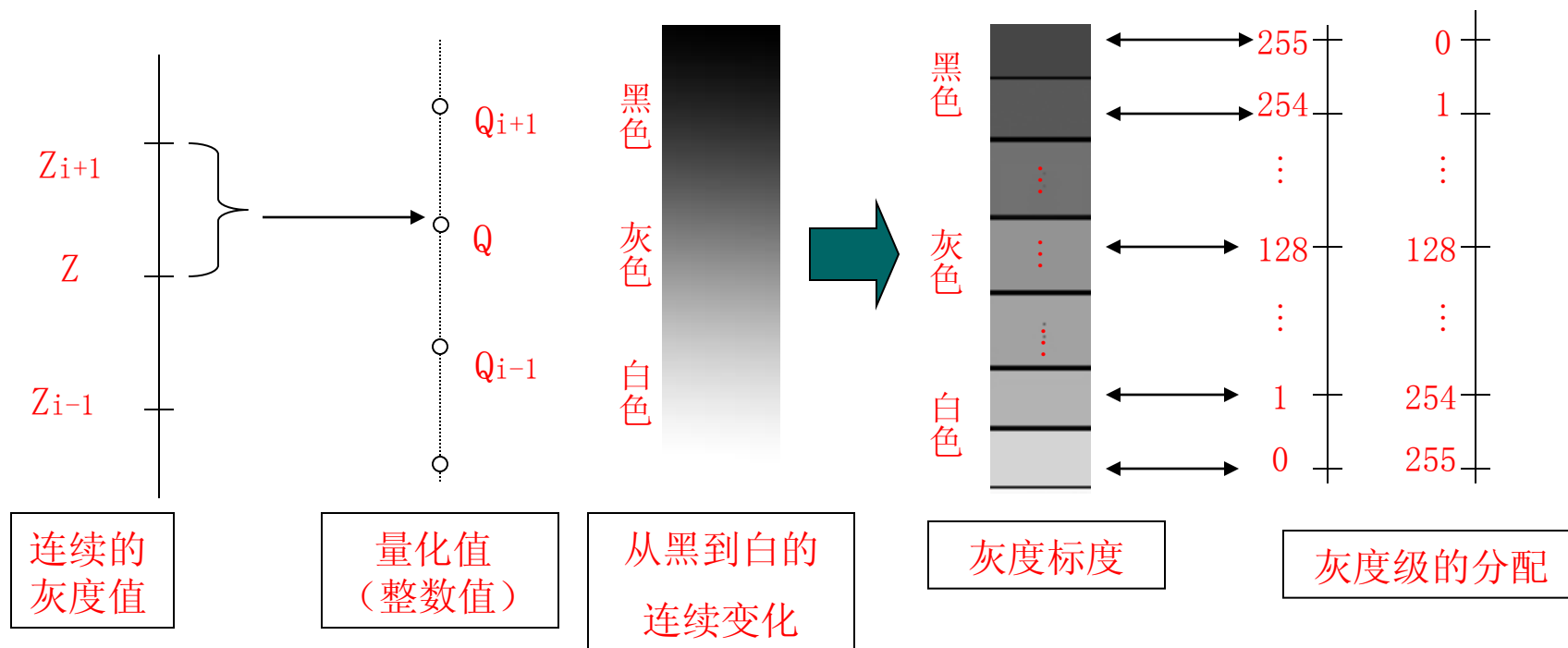
§ 2.4.1.2 量化过程

量化(QUANTIZATION)：样元值离散化的过程；

将模拟量样元值按一定方式离散化为数字量，完成模 / 数转换，得到灰度值。

量化过程把图像在性质上离散化。

量化过程示意图



量化

把从黑到白的灰度级(gray level)
量化成8比特示意图

§ 2.4.1.3 采样和量化的误差

采样：取样点(样元、像素)与图像分辨率是相关的，取样点越多，与原始图像越接近，但数据量迅速增加；若取样点太少(欠采样)，图像质量变差，造成**取样误差**。

量化：将样点量化为Q级，Q如何取值？

1) Q总是取2的整数次幂，如 $Q=2^b$ ；

2) b取值越大，重建图像失真越小，若要完全不失真重建图像，b必须取无穷大，否则一定存在失真。这就是所谓**量化误差**。

采样误差

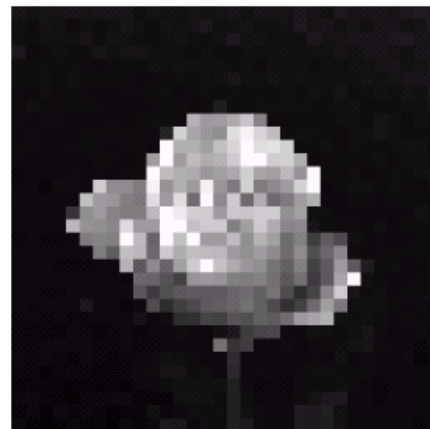
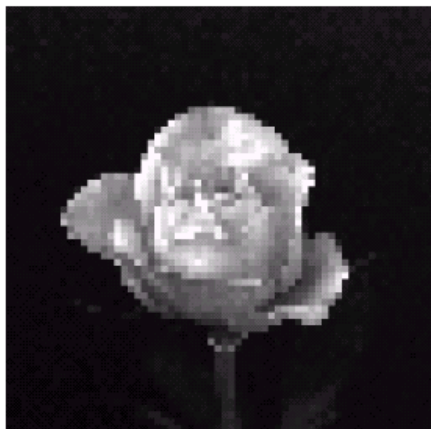
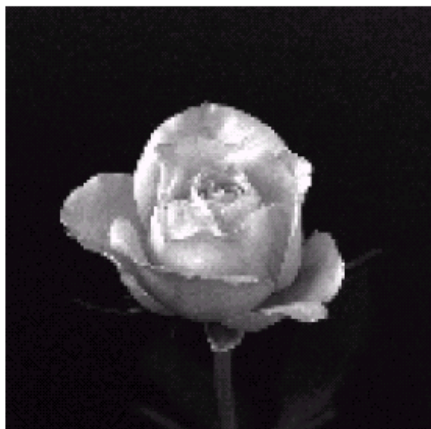
比较不同采样分辨率获取 1024×1024 图像的结果。

1024×1024



256×256

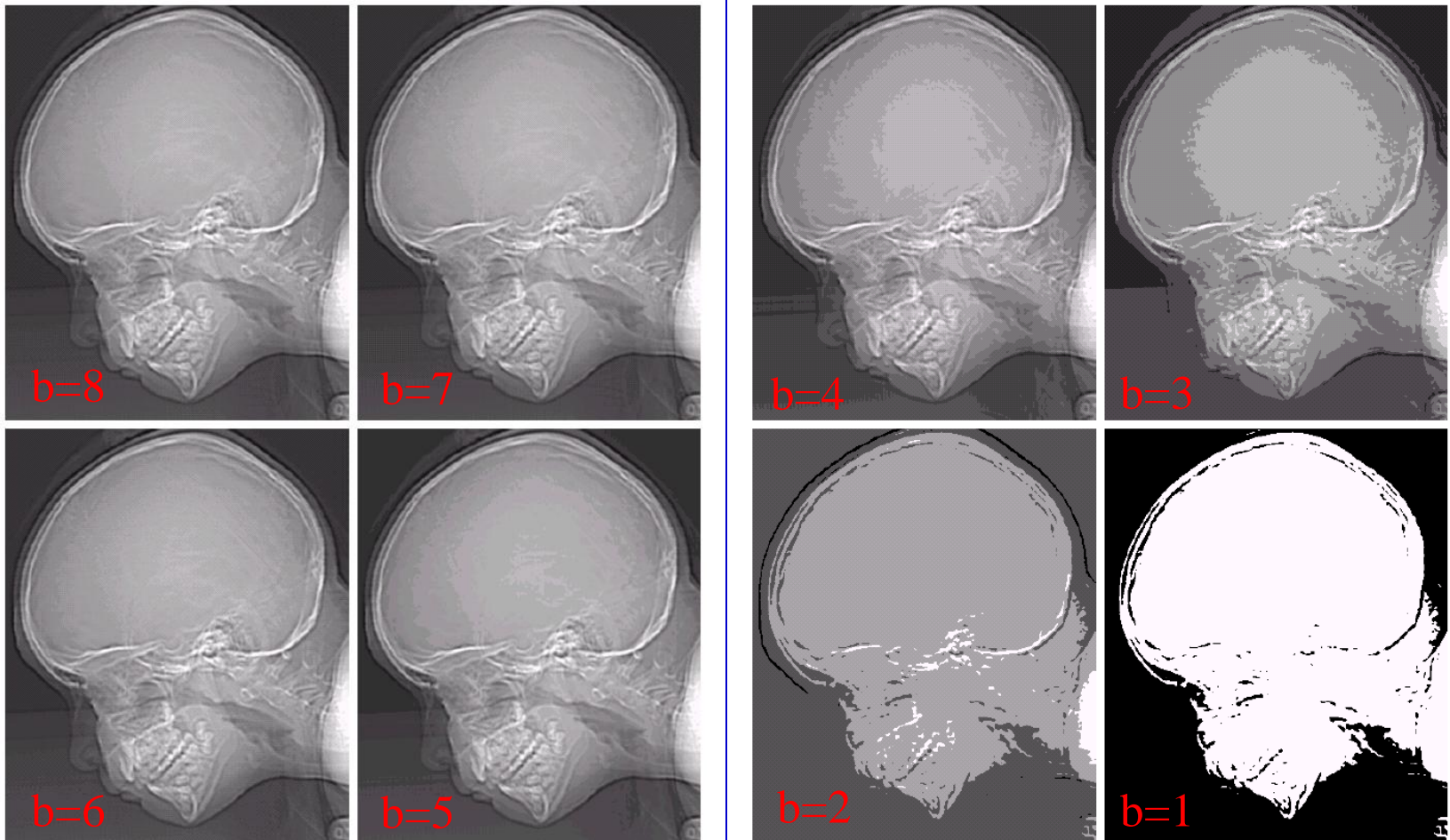
128×128



32×32

量化误差

采用不同分辨率量化图像的结果 2^b , $b=8, \dots, 1$ 。





采样和量化总误差

总的采样误差 = 取样误差 + 量化误差;
取样脉冲宽度有限 量化级有限

取样误差是取样脉冲宽度有限造成的;

量化误差是量化级有限造成的;

§ 2.4.1.4 均匀采样和量化

假定一幅图像取 $M \times N$ 个样点，空间分辨率一般取 $M=N$ ，且都是2的整数幂，便于计算机处理。

如图像尺寸为 128×128 、 256×256 、 512×512 或 1024×1024 等。

灰度量级数一般取2的8次幂(256级灰度)。

需要的存储比特数为： $M \times N \times 8$ ；参见表2.1；1次幂为黑白图像。

§ 2.4.1.5 非均匀采样和量化

- **非均匀采样**是自适应的采样过程，改善总的采样效果。

具体方法是根据图像局部特征进行采样：

对图像中变化剧烈的区域采用较密的采样；

对图像中变化平滑的区域采用较稀的采样；

缺点：需要先确定边缘，采样过程较复杂；

- **非均匀量化**是一种最佳量化。

根据图像像素的灰度分布概率密度函数，确定分层灰度和等级灰度，使量化后图像与原图像的均方误差最小。

§ 2.4.2 空间和灰度级分辨率

空间分辨率：单位距离可分辨的最小线对数目；

如每毫米100线对；每英寸2540线对。

灰度级分辨率：灰度级别中可分辨的最小变化。

即灰度分辨率是指图像中可以辨别的最小明暗变化，
灰度分辨率受量化等级的影响。

一般灰度级取2的8次幂；256级灰度等级。

通常将两者结合描述，如**128*128*8、256*256*8等**。

空间分辨率



1024



512



256



128

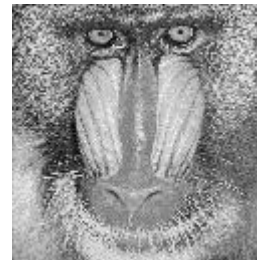
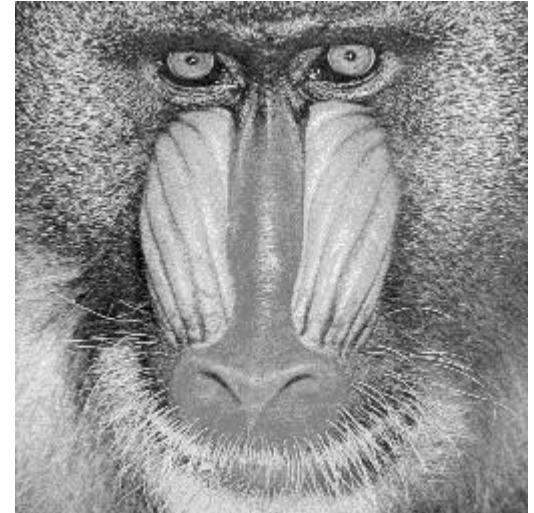
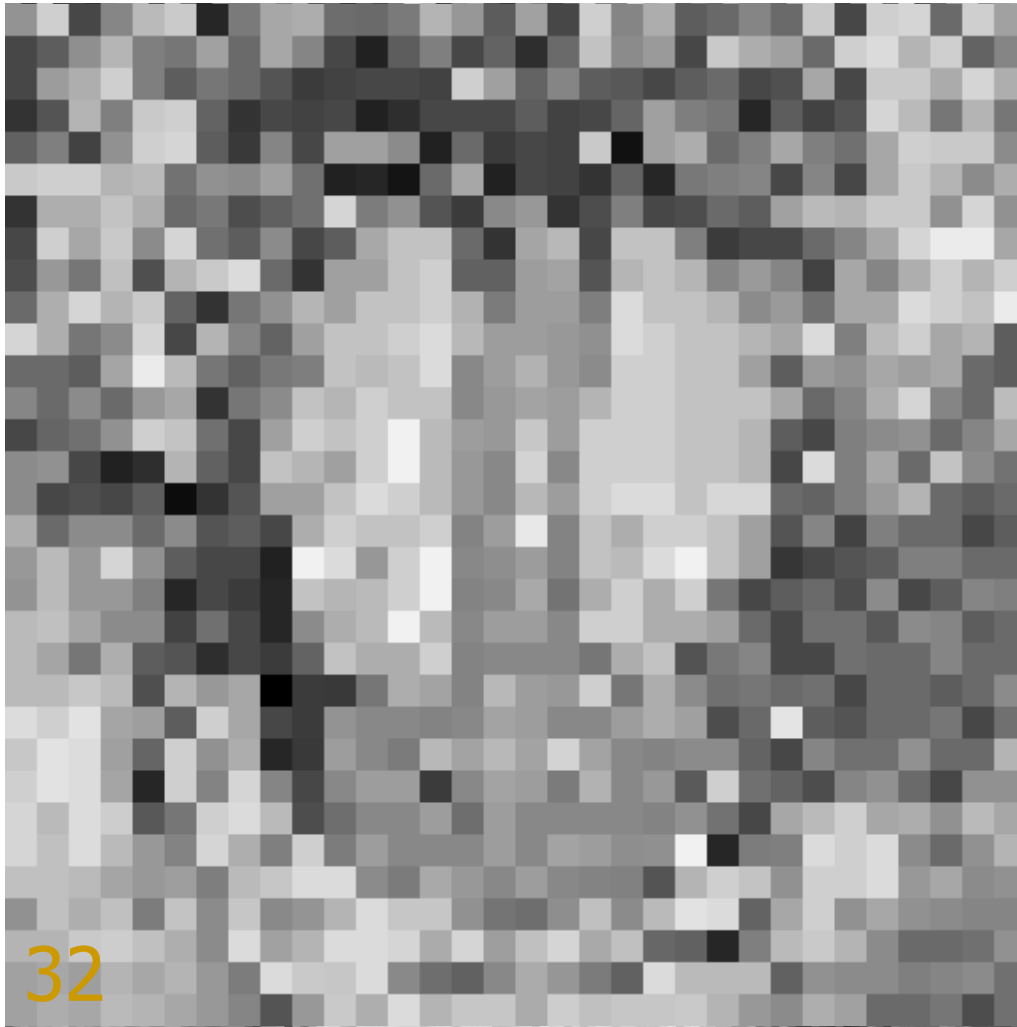


64



32

空间分辨率



- $G_{M \times N}$
- 512×512
- 256×256
- 128×128
- 64×64
- 32×32



图像灰度级

$$2^8=256(8\text{bits})$$

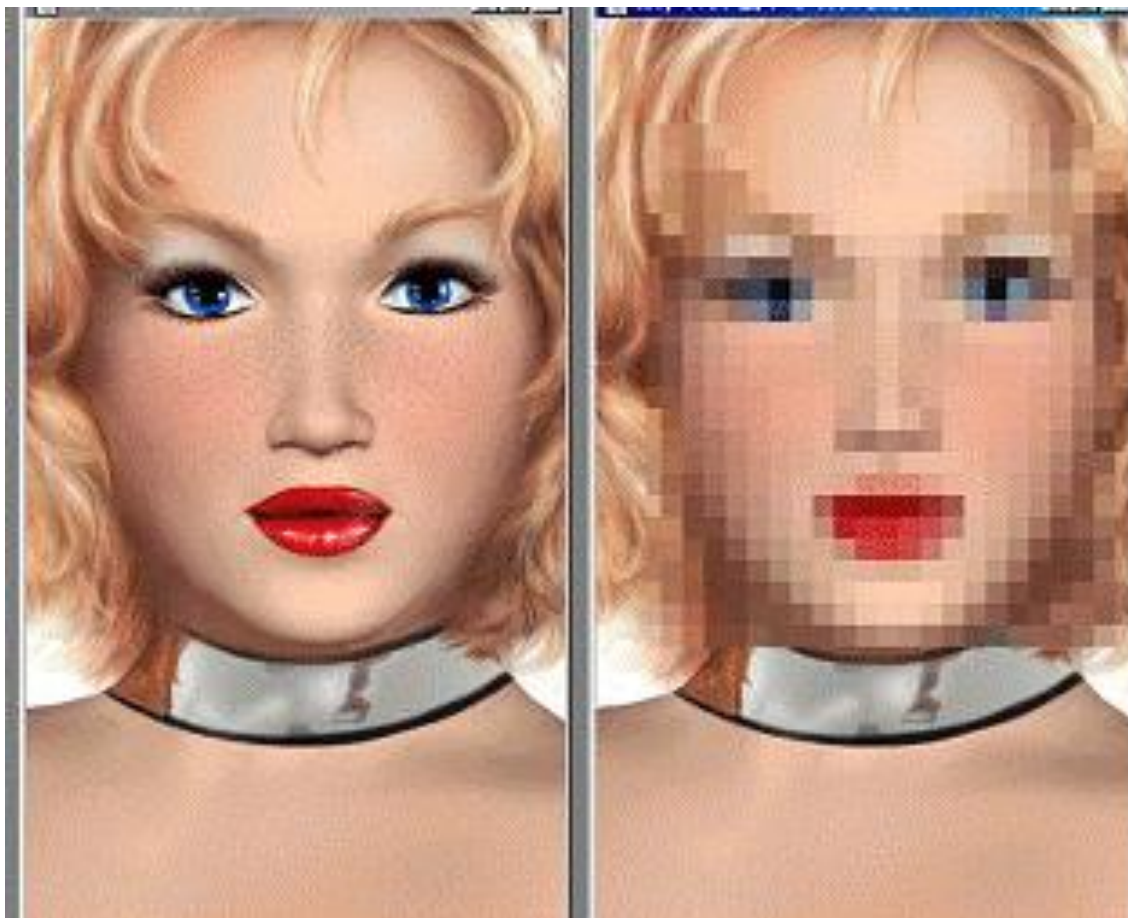
$$2^5=32 \quad (5\text{bits})$$

$$2^4=16 \quad (4\text{bits})$$

$$2^3=8 \quad (3\text{bits})$$

$$2^1=2 \quad (1\text{bits})$$

马赛克效果



§ 2.4.3 数字图像存储表示

一、矩阵表示（普遍方式）

矩阵的元素对应于图像的像素pixel、取样的量化值；

$$f = \begin{bmatrix} f_{11} & f_{12} & \dots & f_{1N} \\ f_{21} & f_{22} & \dots & f_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ f_{M1} & f_{M2} & \dots & f_{MN} \end{bmatrix} = \begin{bmatrix} f_{jk} \end{bmatrix} \quad (j=1,2,\dots,M; \quad k=1,2,\dots,N)$$

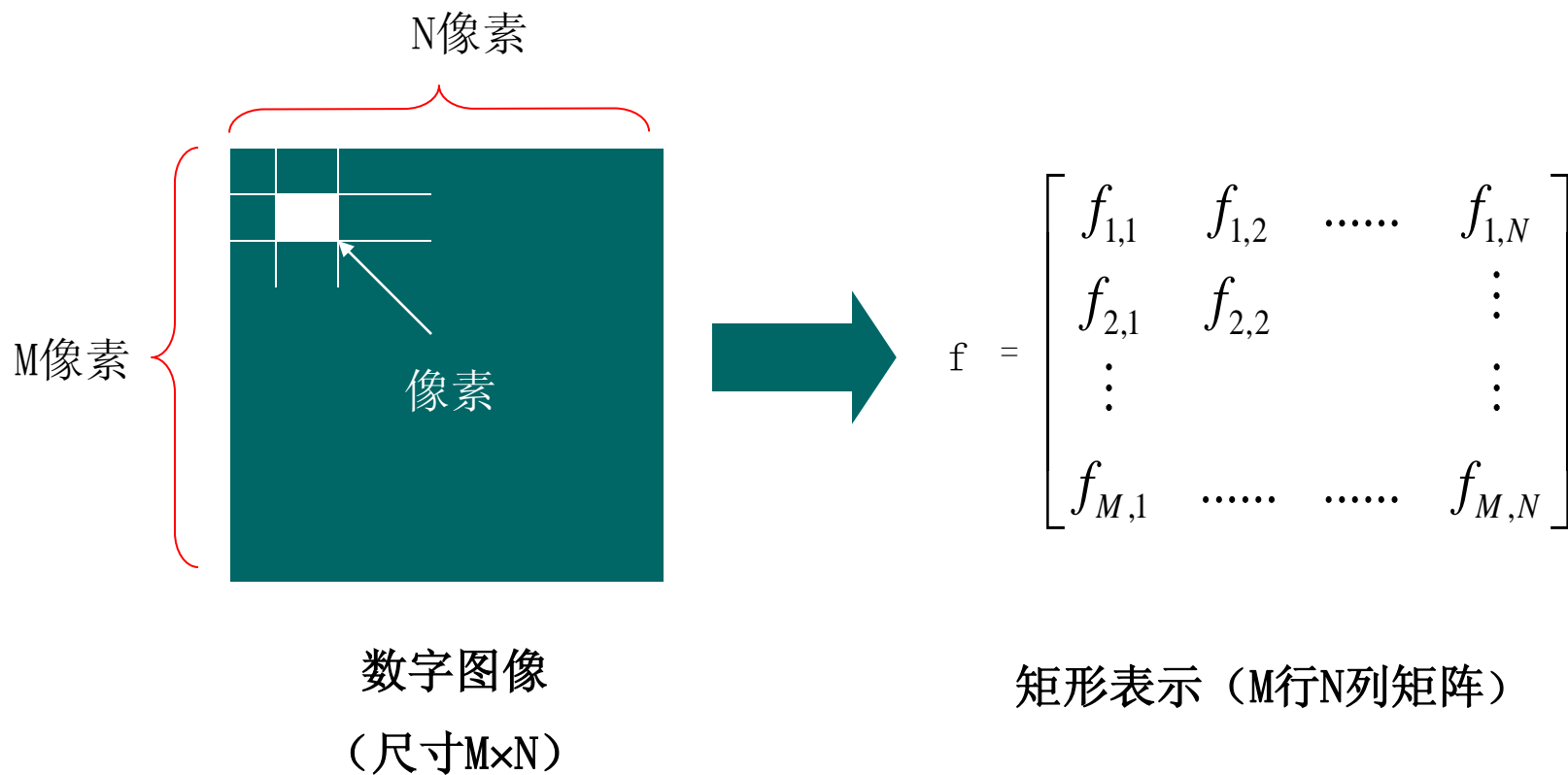
像素具有非负性(0-255)和有界性；**Matlab数据类型uint8**；

VC中数据结构 **unsigned char** $f[128][128]$, $*fima[256]$ ；

若 f_{jk} 表示像素，图像的能量 $E = \sum \sum f_{jk}^2 < \text{上限值}$ ；有界；
(C语言中下标是从(0,0)开始，到(M-1,N-1))。

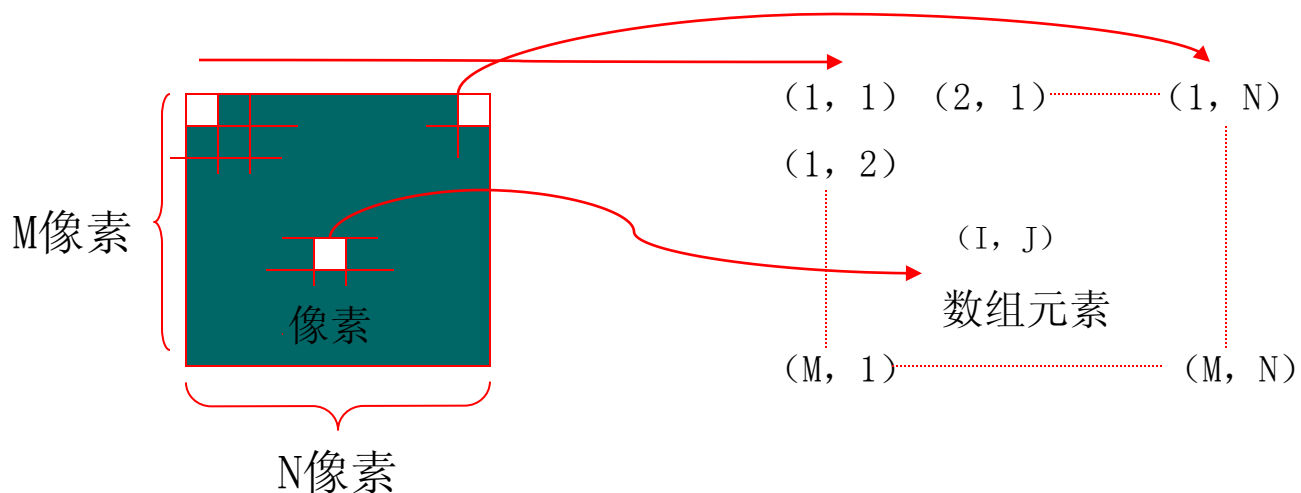
矩阵表示的优点：与图像直观对应；

矩阵运算易处理；用**Matlab**更方便。



数字图像的矩阵表示

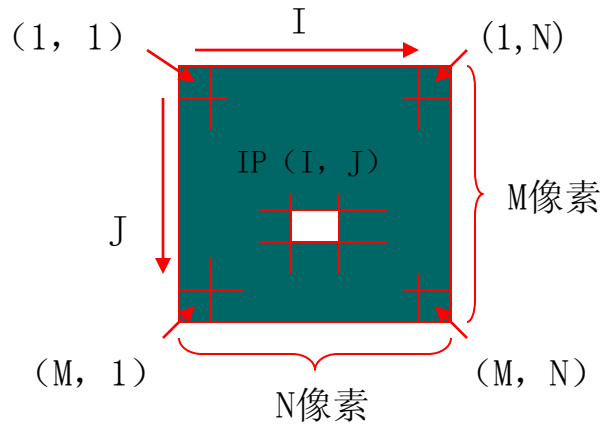
二维数组和图像的关系



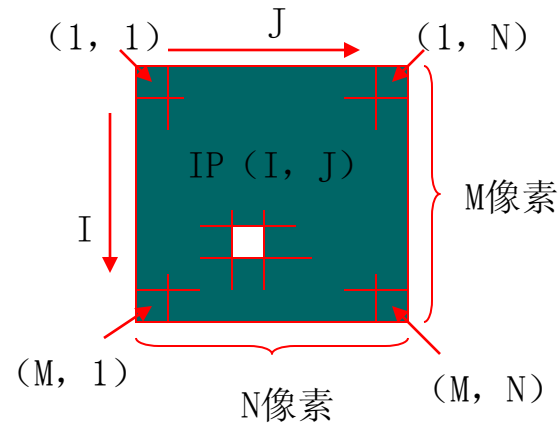
数字图像

大小为 $M \times N$ 的二维数组IP

把数字图像存储到二维数组中



F型



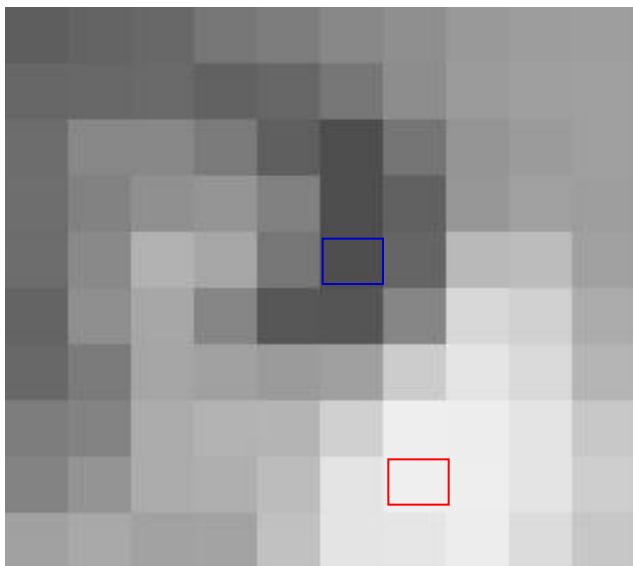
M型

二维数组的标注和数字图像的对应关系有两种形式：

1) 数组的第1,2标注 I, J 分别对应于画面的水平 I 和垂直 J 两方向——**F型**(循环语句取数组元素时的顺序-逐行读取)

2) 数组的第1,2标注 I, J 分别对应于画面的垂直 I 和水平 J 两方向——**M型**(循环语句取数组元素时的顺序-逐列读取)

10*10图像及数组表示



数字图像I
(尺寸10×10)



94	100	104	119	125	136	143	153	157	158
103	104	106	98	103	119	141	155	159	160
109	136	136	123	95	78	117	149	155	160
110	130	144	149	129	78	97	151	161	158
109	137	178	167	119	78	101	185	188	161
100	143	167	134	87	85	134	216	209	172
104	123	166	161	155	160	205	229	218	181
125	131	172	179	180	208	238	237	228	200
131	148	172	175	188	228	239	238	228	206
161	169	162	163	193	228	230	237	220	199

size(I)为10, 10

min(I(:))为78;

max(I(:))为239;

§ 2.4.3 数字图像存储表示

二、矢量表示

用按行或按列的顺序排列所有像素为矢量，即矢量形式；

如按列排M*N列的图像，可表示为：

$$f = [f^1, f^2, \dots, f^N];$$

$$\begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,N} \\ f_{2,1} & f_{2,2} & & \\ \vdots & & & \\ f_{M,1} & L & L & f_{M,N} \end{bmatrix}$$

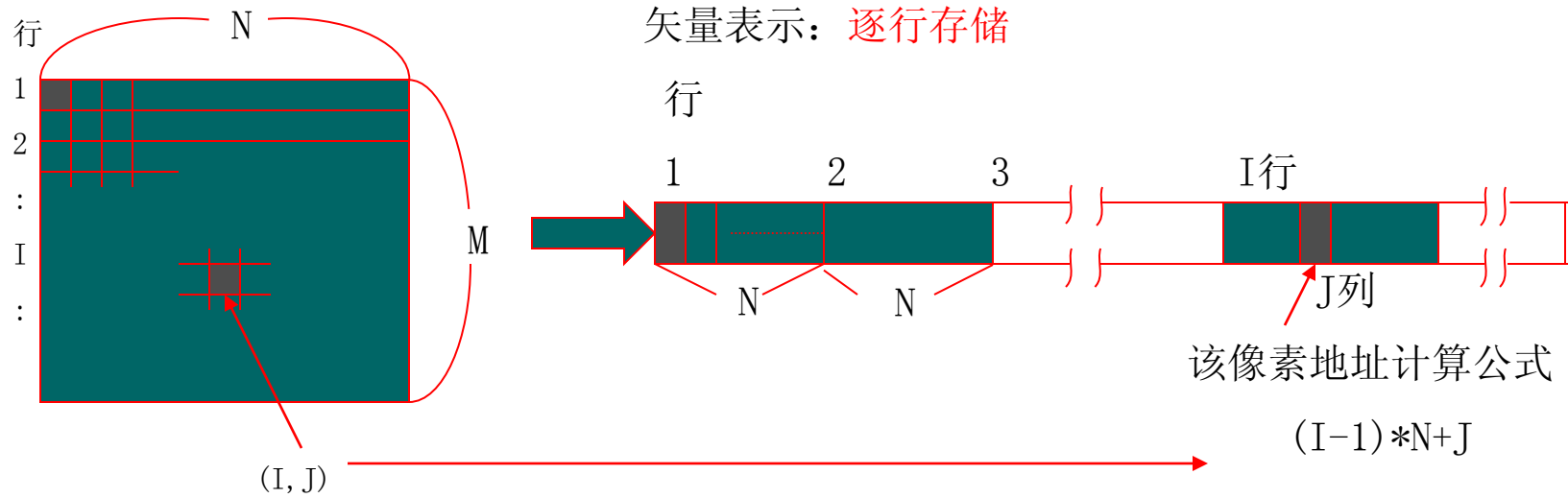
式中 $f^i = [f_{1i}, f_{2i}, \dots, f_{Mi}]^T$; ($i=1, 2, \dots, N$) ; 每列有M行(M个像素)

图像MN*1的列矢量 $f = [f_{11}, f_{21}, \dots, f_{M1}, f_{12}, f_{22}, \dots, f_{M2}, \dots \dots \dots$
 $f_{1N}, f_{2N}, \dots, f_{MN}]$;

第一列

第二列

第N列



把图像数据存储到一维数组或矢量中
第(I,J)个像素的位置计算呈线性公式
 $(I-1)*N+J$, C语言适合用指针变量。

矢量表示的优点：

图像的能量计算公式很简洁 $E = f^T f = \sum f_i^2$ ($i = M * N$);

相邻点的表示有规律可循，便于公式计算；

空间分配上比矩阵方式灵活；

三、图像矩阵表示与矢量表示之间的互换

矩阵到矢量(略)

矢量到矩阵(略)

§ 2.4.3 数字图像存储表示

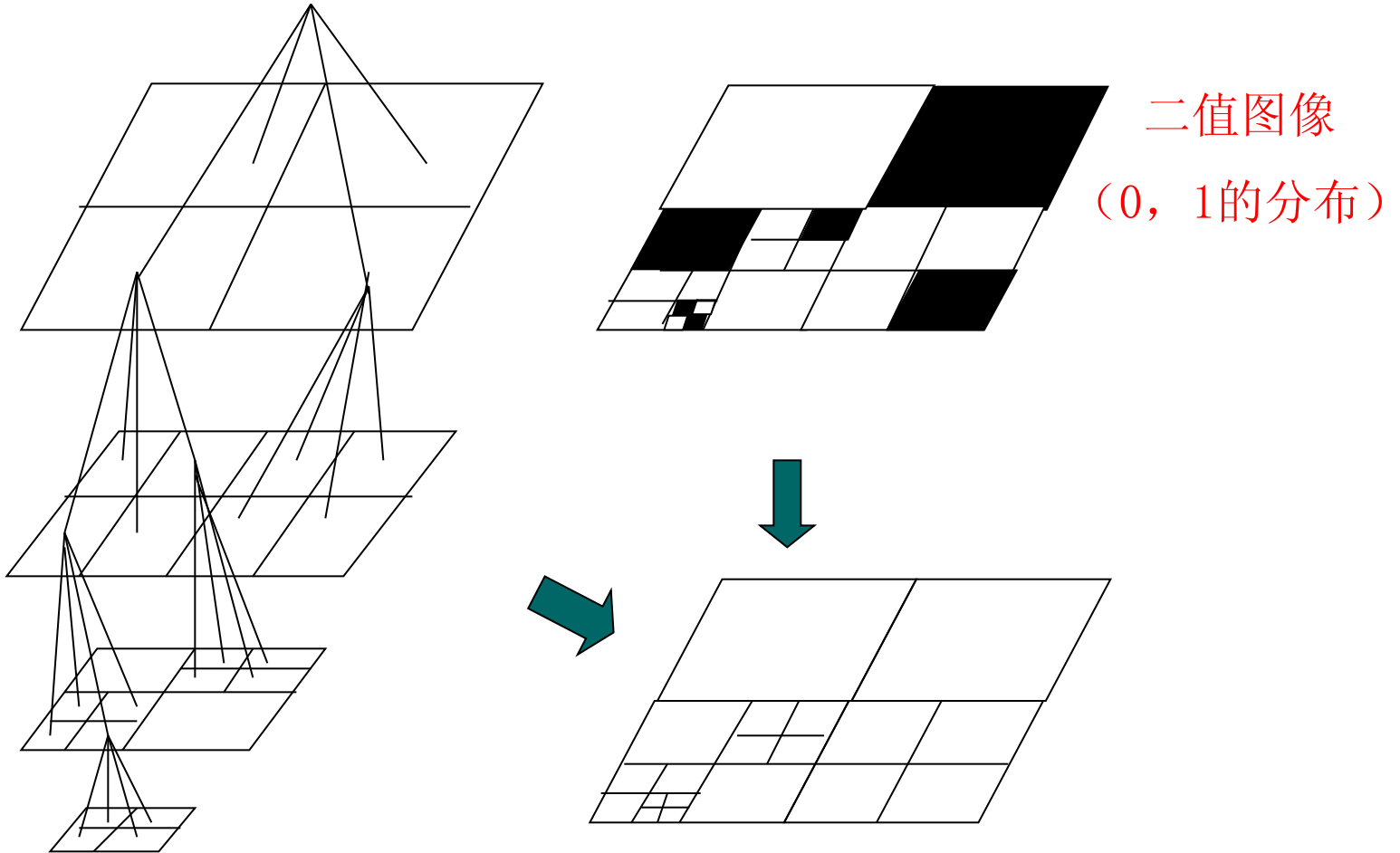
四、二值图像的树结构表示

对于二值图像，可以采用稀疏方法存储。

将二值图像横向纵向都接连不断地二等分，如果被划分部分的图像中全体都是白色或黑色时，这一部分则不再划分。用这种方法，可以将图像用树结构（四叉树）来表示。

树结构表示可以应用在二值图像特征提取和信息压缩等方面，节省存储空间。

二值图像的树结构表示说明



图像的四叉树表示

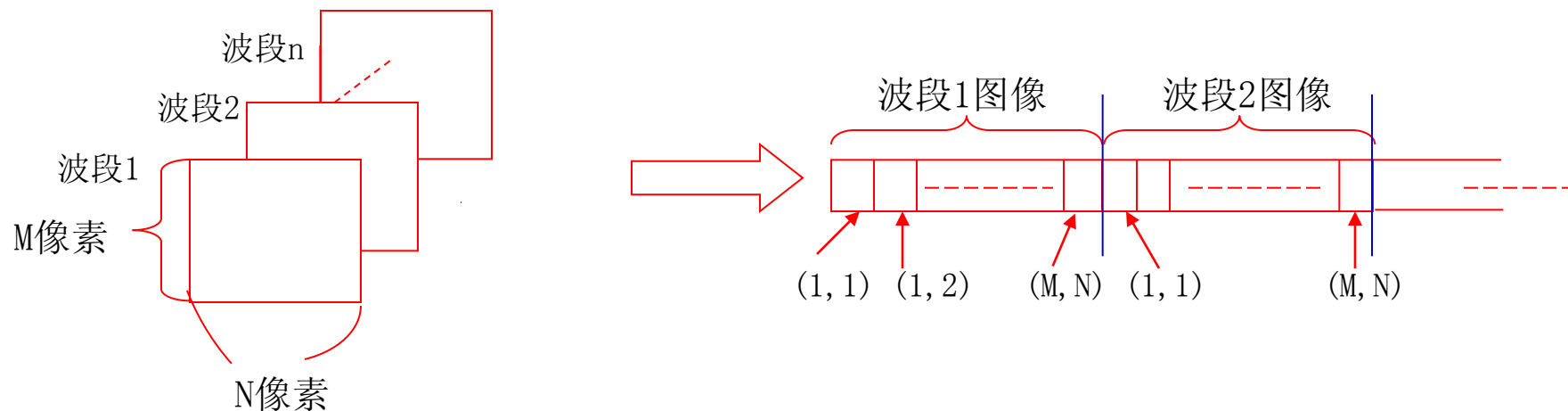
§ 2.4.3 数字图像存储表示

五、多波段图像的表达—数据结构

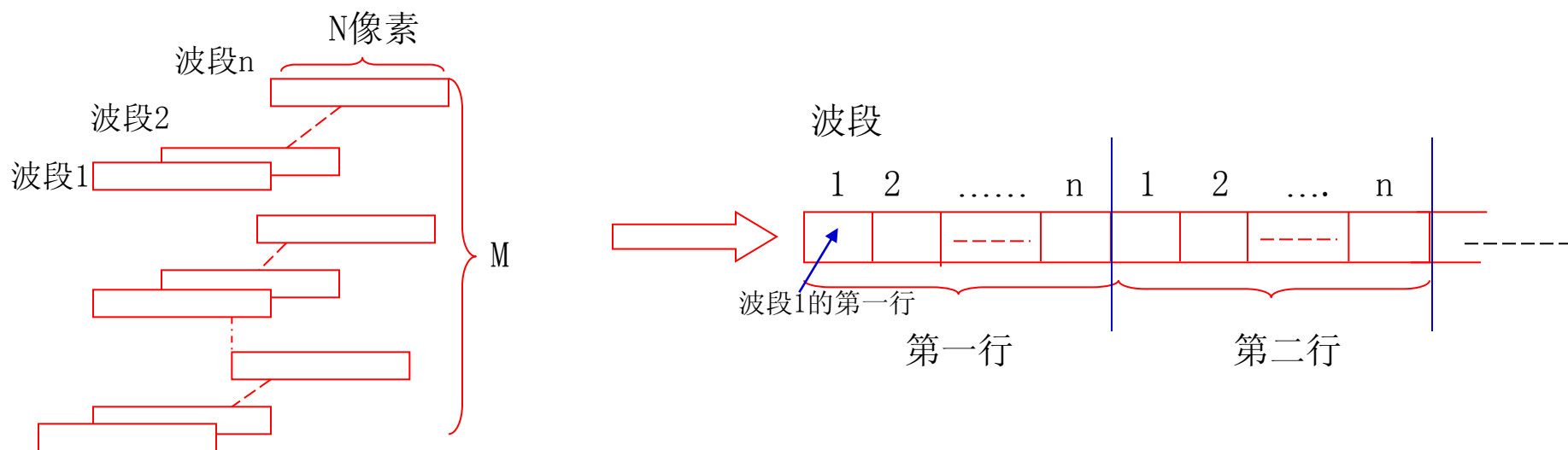
在彩色图像（红、绿、蓝），或多谱图像（把同一对象用多个不同的波长拍摄得到的多波段图像）中，各个像素包含着多个与图像有关的信息。这类图像数据的存储，以多谱图像为例分析，有下列方法：

- ① 按每个波段存储图像；
- ② 按每个扫描线存储各个波段的数据；
- ③ 按每个像素存储各个波段的数据。

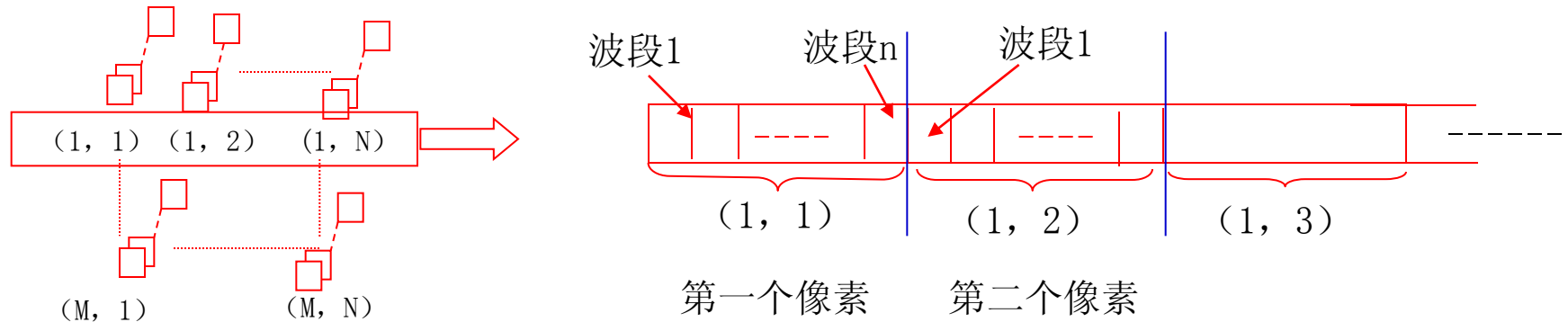
多谱图像的数据存储方法



(a) 按每个波段存储图像



(b) 按各条扫描线(行)存储各波段数据



(c) 按每个像素存储各波段数据

比较:

在进行每个波段的处理时, 采用存储方法①(按每个波段存储图像), 是最常采用的一种方法;

类似的, 以视频图像或者电影胶片等为代表的运动图像数据, 也可以看成是一种多重图像。按一定时间间隔顺序得到的一个一个场面的图像(叫做帧)的集合, 可以用对应于①(按每个波段存储图像)的形式进行处理。

若把各波段的数据装配起来进行图案分类的场合, 采用方法③(按每个像素存储各个波段的数据)存储比较有利, 容易对每个像素的 n 个波段值进行分析后分类。

§ 2.4.4 图像文件格式

常用的图像文件格式有很多，可用后缀来区分具体格式。

原始图像：文件扩展名为.raw(没有文件头、只有像素数据)

BMP格式：文件扩展名为.bmp(位图，Bitmap))

JPEG格式：文件扩展名为.jpg

TIFF格式：文件扩展名为.tif

GIF格式：文件扩展名为.gif(动画图像)

以BMP格式为例进行介绍。

注：不同文件格式的图像通常可以相互转换(另存为即可)；

Bitmap 的文件头说明部分(54字节)

偏移	长度	名称	描述
0	2	bfType	ASCII码 “BM” (0x424D)
2	4	bfSize	文件长度 (文件总字节数)
6	2	bfReserved	0
8	2	bfReserved	0
10	4	bfOffbits	图像开始偏移量

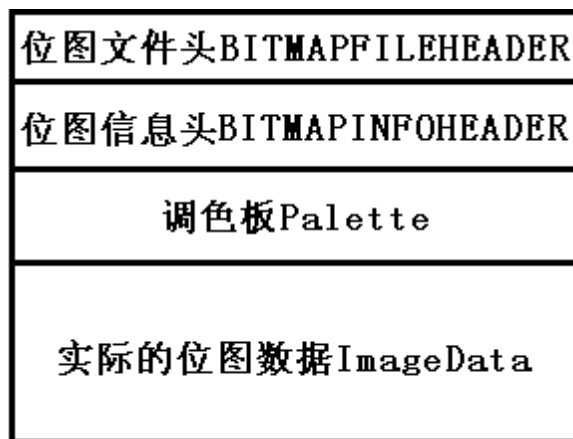
注： VC中**CDIB**类将上述内容14字节用一个结构表示 (文件头结构)
以下40字节用一个结构表示(信息头结构)

偏移	长度	名称	描述
14	4	biSize	头的大小，40字节（信息头结构大小）
18	4	biWidth	图像宽度
22	4	biHeight	图像高度
26	2	biPlanes	图像位平面数目（恒定为1）
28	2	biBitCount	像素位数目（色彩：8为256色，24为RGB）
30	4	biCompressin	压缩类型
34	4	biSizeImage	压缩图像大小
38	4	biXPelsPerMeter	水平分辨率
42	4	biYPelsPerMeter	垂直分辨率
46	4	biClrUsed	彩色数目（为0时，是2的biBitCount次方）
50	4	biClrImportant	“重要颜色”数目（为0时，都重要）
54	4*N	bmiColors	彩色“影射”（索引表，RGB和一个保留）

Windows的位图文件(.bmp文件)的格式

Windows位图文件结构示意图

bmp文件大体上分成四个部分



第一部分为位图文件头结构BITMAPFILEHEADER;

第二部分为位图信息头结构BITMAPINFOHEADER;

这个结构的长度是固定的，为40个字节;



第三部分为调色板(Palette)，注意，调色板部分是对那些需要调色板的位图文件而言的。有些位图，如真彩色图，是不需要调色板的，在BITMAPINFOHEADER后直接就是位图数据。

调色板(Palette)，另一种叫法是颜色查找表 (LookUpTable) LUT，似乎更确切一些。Windows位图中用到了调色板技术，将需要的颜色作成一张表，如果表的第0行设为255, 0, 0, 0（红色），那么当某个像素为红色时，只需要标明0即可，可以节省图像尺寸。

第四部分就是实际的图像数据了。

.BMP文件的数据是从下到上，从左到右安排的。也就是说，从文件中最先读到的是图像最下面一行的左边第一个像素，然后是左边第二个像素...接下来是倒数第二行左边第一个像素，左边第二个像素...依次类推，最后得到的是最上面一行的最右一个像素。

对于用到调色板的位图（索引位图**index**），图像数据就是该像素颜色在调色板中的索引值，对于真彩色图，图像数据就是实际的**R、G、B**值。

§ 2.4.5 放大和收缩图像

实际应用中，图像的缩放(zoom) 采用下式表示尺寸的变化。

$$\begin{bmatrix} x0 \\ y0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1/ratio & 0 & 0 \\ 0 & 1/ratio & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x1 \\ y1 \\ 1 \end{bmatrix}$$

ratio 为缩放因子；缩小为欠采样，放大为过采样。

图像放大一般需要执行二步操作：

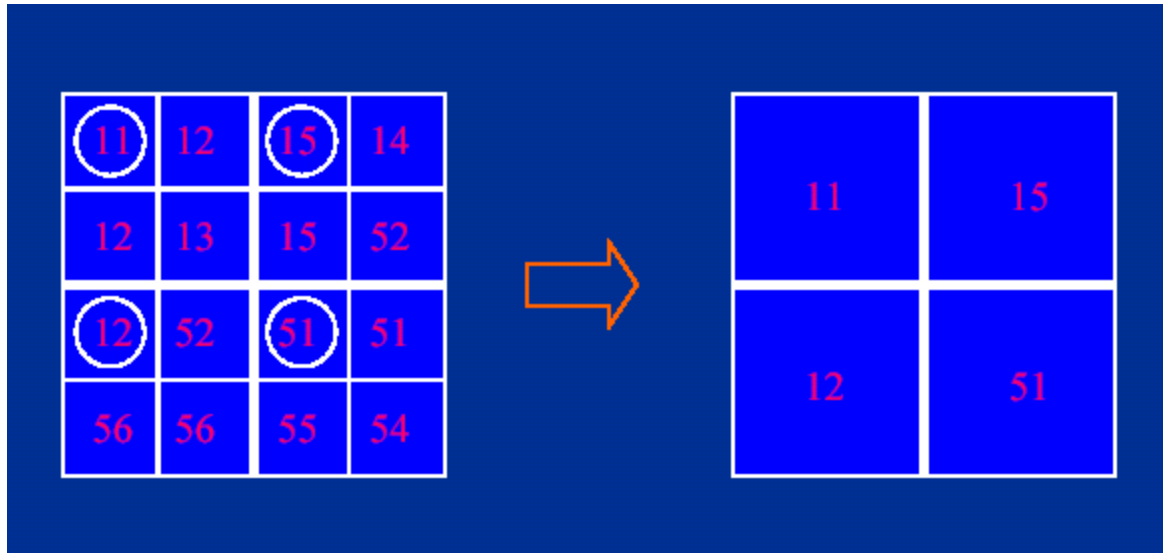
1)增加新的像素位置；按放大因子扩大空间尺寸；

2)对新位置的像素赋灰度值；称为像素插值；如整数倍放大时，简单复制同样的值，像素插值方法在第5章中再介绍。

§ 2.4.5 放大和收缩图像

图像收缩与图像放大相似，但图像空间是缩小的；

例如亚采样为二中取一：隔一行(列)，删除一行(列)；



多倍数收缩是多行(列) 中取一行(列) ， 删除其余行(列)。

图像放缩举例



放大2倍



原图像



缩小2倍

§ 2.5 像素间的一些基本关系

§ 2.5.1 相邻像素

一、4-邻域(近邻像素)

设像素 p ，坐标为 (x,y) ，有4个近邻像素 r_i ，坐标分别为 $(x+1,y)$ ， $(x-1,y)$ ， $(x,y+1)$ ， $(x,y-1)$ ，组成 p 的4-邻域，记为 $N_4(p)$ ；

4-邻域近邻像素间相距一个单位距离；如果 p 在图像的边缘，它的近邻像素会在图像外，需特殊处理；

4 - 邻域 (4 - neighbor) 像素示意图: (红色像素)

$$N_4(P) = \{(x+1, y), (x, y+1), (x-1, y), (x, y-1)\}$$

s_1 $(x-1, y-1)$	r_1 $(x-1, y)$	s_2 $(x-1, y+1)$
r_4 $(x, y-1)$	P (x, y)	r_2 $(x, y+1)$
s_4 $(x+1, y-1)$	r_3 $(x+1, y)$	s_3 $(x+1, y+1)$

对角近邻像素示意图: (灰色像素)

$$N_D(P) = \{(x-1, y-1), (x-1, y+1), (x+1, y+1), (x+1, y-1)\}$$

二、对角近邻像素

像素 s_i : $(x-1, y-1)$, $(x-1, y+1)$, $(x+1, y+1)$, $(x+1, y-1)$, 组成 p 的对角近邻像素, 记为 $N_D(p)$;

三、8-邻域

p 的周围8个近邻像素全体称为 p 的8-邻域, 记为 $N_8(p)$;

8-邻域是 $N_4(p)$ 和 $N_D(p)$ 之和。

§ 2.5 像素间的一些基本关系

§ 2.5.2 邻接性、连通性、区域和边界

判断方法：像素是否接触、灰度值是否满足某个特定的相似准则；

一、连接(连通的特例)

设 V 表示连接的灰度值集合，如 $V=\{8,9,\dots,16\}$

4-连接：2个像素 p 和 r ，在 V 中取值，且 r 在 $N_4(p)$ 中，则它们为4-连接；

8-连接：2个像素 p 和 r ，在 V 中取值，且 r 在 $N_8(p)$ 中，则它们为8-连接；(存在歧义性)

m-连接(混合连接): 是8-连接的一种变形, 消除8-连接中可能出现的多路连接问题。

定义: 2个像素 p 和 r , 在 V 中取值, 且满足下列条件之一, 则它们为**m-连接**;

条件一: r 在 $N_4(p)$ 中;

条件二: r 在 $N_D(p)$ 中, 且 $N_4(p) \cap N_4(r)$ 是空集;

即不能有元素同时出现在 $N_4(p)$ 和 $N_4(r)$ 中。

例： $V=\{1\}$ ，下图8-连接存在多路连接，不是m-连接。

	0	1	1	$N_D(p)$
p	0	1	0	
	0	0	1	

二、毗邻

如果一个像素 p 和另一个像素 q 相连接，则它们相毗邻；

所以对应4-连接有4-毗邻、8-连接有8-毗邻、 m -连接有 m -毗邻。

对于图像中子集 S 和 T ，如果 S 中的一些像素与 T 中的一些像素毗邻，则 S 与 T 是毗邻的。

三、通路

由相互毗邻定义通路，对应**有4-通路、8-通路、m-通路**。

四、连通

图像子集 S 中的像素 p 和 q ，如果存在一条从 p 到 q 的通路，称 p 在 S 中与 q 相连通。

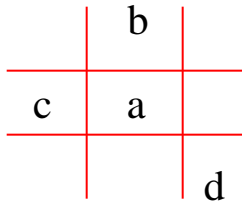
连通组元： S 中与 p 连通的像素的集合(包括 p)称为一个连通组元。

五、等价关系和传递闭包

1. 关系的定义：设集合A，定义如下二元关系R，

$R = \{ \langle a, b \rangle \mid a, b \in A \text{ 且 } a \text{ 与 } b \text{ 相连} \}$ ，写着 $a \sim_R b$ ；

例：设点集 $A = \{a, b, c, d\}$ ，分布如图，R表示4-连接关系



则有 $R = \{(a, b), (b, a), (a, c), (c, a)\}$ ；

d无任何4-连接关系。

2. 关系的描述方法

关系常用二值矩阵**B**来描述；

矩阵由元素构成行和列，存在关系的分量为1，否则为0，

上述关系的矩阵**B**如下。

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

3. 关系的性质

★ 反射性：如果对A中的每个a， $a \sim R \sim a$ 成立，称R具有反射性

特点：矩阵中主对角线上元素均应为1；

★ 对称性：如果对A中的a和b， $a \sim R \sim b$ 成立， $b \sim R \sim a$ 亦成立，称R具有对称性。例中存在 (a,b) 和 (b,a)，R为对称矩阵；

特点：矩阵中对称分量均应为1；

★ 传递性：如果对A中的a,b和c， $a \sim R \sim b$ ， $b \sim R \sim c$ ，则 $a \sim R \sim c$ 成立，称R具有传递性。这种隐含关系可通过算法导出。见B⁺算法。



4. 等价关系

满足反射性、对称性、传递性的关系，称为等价关系。等价关系中， A 可分为多个等价组(子集)，子集内的元素才有关系 R 。

5. 传递闭包

R的传递闭包，记为 **R^+** ，是包含所有隐含关系的集合。

若**B**是**R**的关系矩阵，则 **B^+** 是 **R^+** 的矩阵。

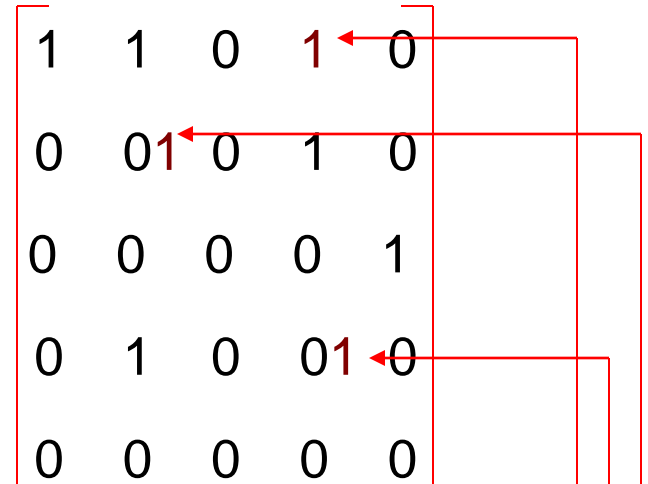
定义： **$B^+ = B + BB + BBB + \dots + B^n$** 。

求 **B^+** 的快速算法思路：

1) 逐列分析计算；

2) 对**j**列，若**i**行元素为1，则将**j**行布尔加到**i**行上。

$$B = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

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


例：第一列 $j=1$, $i=1$ 时, $B(1,1)=1$, 第一行加到第一行上
 第二列 $j=2$, $i=1$ 时, $B(1,2)=1$, 第二行加到第一行上
 第二列 $j=2$, $i=4$ 时, $B(4,2)=1$, 第二行加到第四行上
 第四列 $j=4$, $i=2$ 时, $B(2,4)=1$, 第四行加到第二行上
 第五列 $j=5$, $i=3$ 时, $B(3,5)=1$, 第五行加到第三行上

§ 2.5 像素间的一些基本关系

§ 2.5.3 距离度量

距离的概念对定义图像中的对象图形的形状或位置关系等计量性质方面是重要的。例如用距离作为分类的依据。

总的来说,对某一集合 s 的元素 p, q, r , 把满足下面性质(称为距离的三定理) 的函数 d 叫做距离(distance)。

(1) 只有当 $p = q$ 时, 才有 $d(p, q) = 0$

(2) $d(p, q) = d(q, p)$

(3) $d(p, r) \leq d(p, q) + d(q, r)$

虽然能够定义满足上式的有种种距离函数，但只有少数在实际中经常被采用。以下仅介绍其中的三种距离。

★欧几里德距离 (Euclidean distance)

$$d_e((i, j), (h, k)) = \sqrt{(i-h)^2 + (j-k)^2}$$

$$\begin{array}{ccccc} & & \sqrt{5} & 2 & \sqrt{5} \\ \sqrt{5} & \sqrt{2} & 1 & \sqrt{2} & \sqrt{5} \\ 2 & 1 & 0 & 1 & 2 \\ \sqrt{5} & \sqrt{2} & 1 & \sqrt{2} & \sqrt{5} \\ & & \sqrt{5} & 2 & \sqrt{5} \end{array}$$

离开单个像素的等距离线大致呈圆形

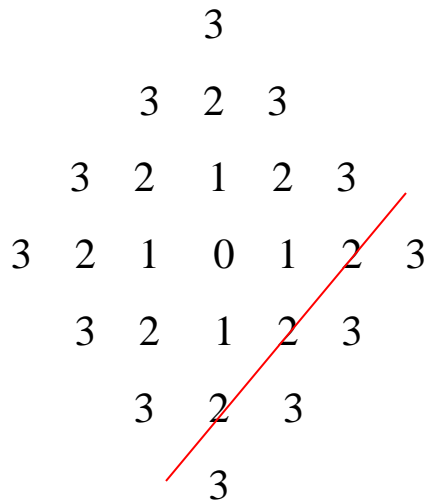
对于一幅 $n \times n$ 的数字图像，其对角距离 $d_e = \sqrt{2}n$

欧几里德距离比较直观，但运算量大，需要开方运算。

★ 4-邻域距离 (4 -neighbor distance)

$$d_4((i, j), (h, k)) = |i - h| + |j - k|$$

d_4 的别名为街区距离 (city - block distance)



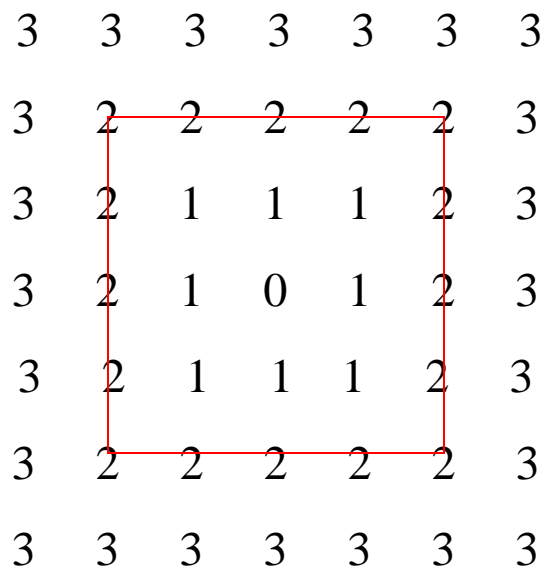
等距离线 呈倾斜45° 的正方形

对于一幅 $n \times n$ 的数字图像，其对角距离 $d_4 = 2n$

★ 8-邻域距离 (8 -neighbor distance)

$$d_8((i, j), (h, k)) = \max \{ |i - h|, |j - k| \}$$

d_8 的别名称为国际象棋盘距离 (chess - board distance)



等距离线 呈正方形

对于一幅 $n \times n$ 的数字图像，其对角距离 $d_8 = n$

§ 2.5.4 基于像素间的图像操作

一、逐像素处理

1. 算术运算：原地运算 $p=p+c$ (或 $p=p-c$)；灰度增减运算。

逻辑运算： p (与、或) q ，像素间运算；

逐像素处理具有串行处理的算法特征；

2. 串行处理：输出图像 $JP(I,J)$ 的值，用输入图像的第 (I,J) 像素 $IP(I,J)$ 的邻域中包含 $IP(I,J)$ 在内的未扫描部分的像素值，以及不包含 $JP(I,J)$ 在内的已处理部分的像素值(两者)进行计算。

用相应的处理算法，按一定顺序依次进行计算。

(注：编程时输入图像和输出图像可以采用同一数组。)

二、模板运算（适合于邻域处理）

采用模板、窗、滤波器进行运算，求均值、方差等，算法灵活。编程时**输入输出图像不能使用同一个数组。**

新灰度值 = 它本身灰度值与其相邻像素灰度值的函数；

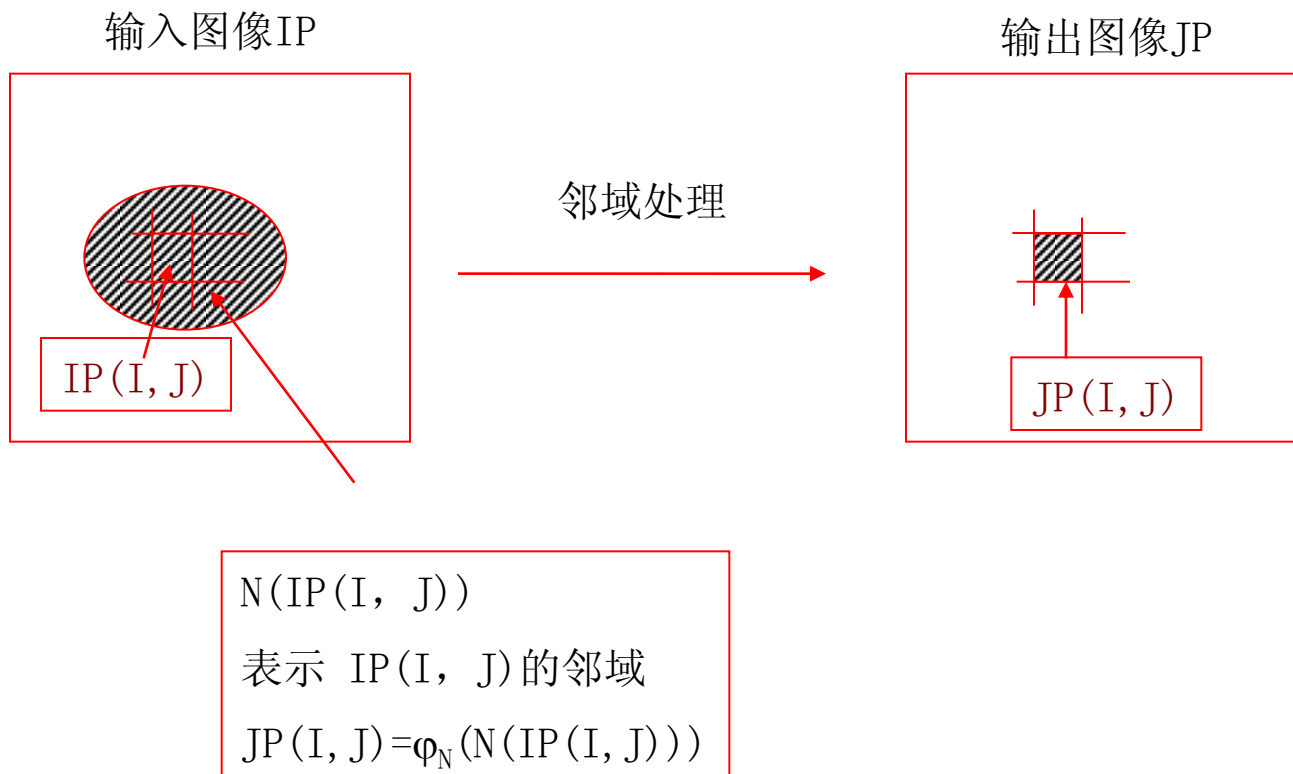
例：求3行3列窗口的平均值，

$$\text{中心像素值 } p_5 = (p_1 + p_2 + \dots + p_9) / 9;$$

优点：可适当选择模板系数，移动模板灵活进行一系列有用的运算；

缺点：运算量较大，可采用专用芯片来加速计算。

邻域处理(Neighborhood Operation)示意图



三、坐标变换

通过坐标变换完成图像整体的平移、旋转和尺度变换(变比、放大、缩小)。

采用矩阵运算实现。通常采用齐次坐标系，将三维坐标放到四维空间进行处理，更方便灵活。

1. 尺度变换(放缩)

$$S = \begin{bmatrix} S_x & & & \\ & S_y & & \\ & & S_z & \\ & & & 1 \end{bmatrix}$$

S_x 表示x方向的尺度变换系数;

S_y 表示y方向的尺度变换系数;

S_z 表示z方向的尺度变换系数;

注: $S_x > 1$ 时, 放大, $S_x < 1$ 时, 缩小;

逆变换为 $1/S_x$

2. 平移变换

$$T = \begin{bmatrix} 1 & & & T_x \\ & 1 & & T_y \\ & & 1 & T_z \\ & & & 1 \end{bmatrix}$$

T_x 表示x方向的平移分量;
 T_y 表示y方向的平移分量;
 T_z 表示z方向的平移分量;

注：原坐标变换至新坐标处，偏移量为 T_x 、 T_y 、 T_z ；
逆变换为 $-T_x$ 、 $-T_y$ 、 $-T_z$ ；

平移后的图像是否要放大？一种做法是不放大，移出的部分被截断，这种处理，文件大小不会改变。还有一种做法是：将图像放大，使得能够显示出所有部分。



3. 旋转变换

旋转与选择的旋转轴有关。

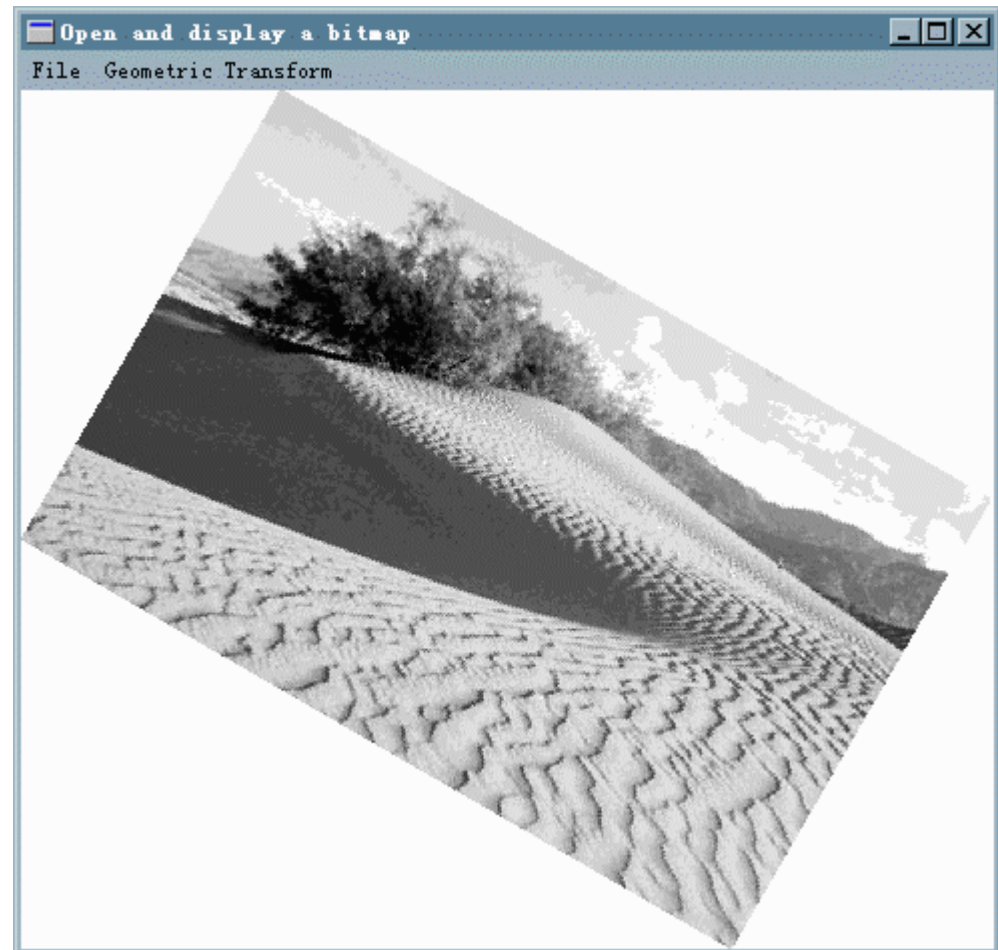
设绕 z 轴旋转 γ 角，旋转变换矩阵为

$$T = \begin{bmatrix} \cos\gamma & \sin\gamma & 0 & 0 \\ -\sin\gamma & \cos\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

注：必须满足的条件是： $\cos\gamma \cos\gamma - \sin\gamma (-\sin\gamma) = 1$

图像的旋转是以图像的中心为圆心旋转，一种做法是旋转后，将图像变大。另一种做法是不让图像变大，转出的图像空间的部分被裁剪掉。

$$\begin{bmatrix} x0 \\ y0 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(a) & -\sin(a) & 0 \\ \sin(a) & \cos(a) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x1 \\ y1 \\ 1 \end{bmatrix}$$

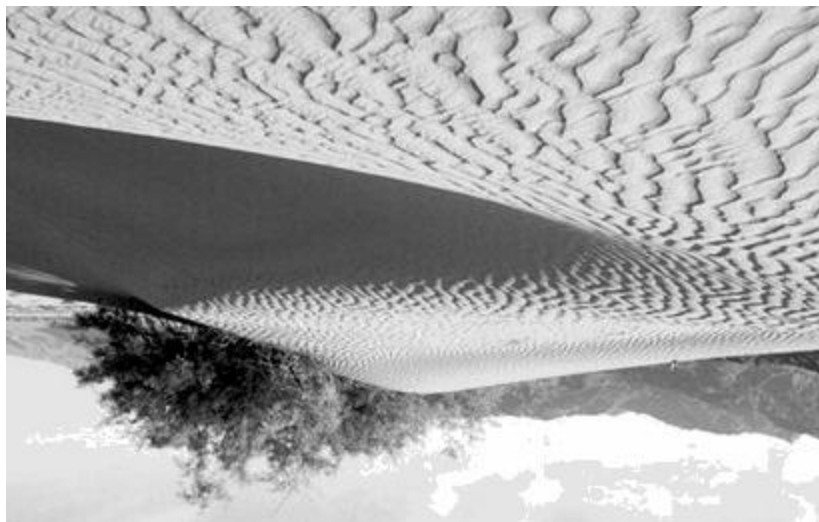


图像的镜像

垂直镜像

$$\begin{bmatrix} x0 \\ y0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & h & 1 \end{bmatrix} \begin{bmatrix} x1 \\ y1 \\ 1 \end{bmatrix}$$

上下像素置换



水平镜像

$$\begin{bmatrix} x0 \\ y0 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ w & 0 & 1 \end{bmatrix} \begin{bmatrix} x1 \\ y1 \\ 1 \end{bmatrix}$$

左右像素置换



图像的转置（长宽互换）



§ 2.5.5 数字图像的统计特性

像素灰度是图像各离散点量测幅度的样本值，在图像处理、识别工作所采用的所有图像特征中，它是最基本、最原始的特征数据。像素灰度分布包含着图像总体的特征，它可以在图像的特定像素点上或其邻域内测定。

当数字图像作为随机过程来处理时，采用统计特性来研究。

设图像 $f = [f_{11}, f_{12}, \dots, f_{MN}]^T$ ，或矩阵形式 $F = [f_{ik}]$

其尺寸为 $N \times N$ ，常用的灰度分布特征有：均值、方差、相关函数等。

一、均值

均值是图像集的一阶矩，数学表示与计算公式为：

$$E\{F\} = [E\{f_{ik}\}], \text{ 或}$$

$$\mu = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y)$$

图像灰度均值是图像包含的**平均能量**。

二、方差函数

方差函数是图像集的二阶中心矩，可直接从均值得到。

$$\sigma = \frac{1}{N} \left[\sum_{x=0}^{N-1} \sum_{y=0}^{N-1} (f(x, y) - \mu)^2 \right]^{\frac{1}{2}}$$

图像灰度方差(标准差)是像素灰度分布的分散程度。

三、图像的对比度

图像的对比度计算公式如下：

$$e = \frac{f_{\max}(x, y) - f_{\min}(x, y)}{\mu}$$

§ 2.5.6 数字图像的主要特征

图像是一个复杂的信息载体，精确地描述图像是困难的。只用简单的几个统计量来描述复杂的客体，会带来一些不完整、不充分的地方，比如灰度特征并没有考虑像素灰度的空间分布，具有同样均值、方差的图像，两者可能完全不同。

因此，在分析处理图像时，应根据实际情况有针对性地研究图像的特征描述。

常用的图像特征及描述方法如下：

一、像素灰度分布

图像每一个像素的灰度值是最基本、最原始的测量值和特征，由它可以组成更大基元的特征。

表征灰度分布的特征描述有总体或局部的均值、方差等。

二、纹理特征

图像纹理是像素灰度的分布，宏观上呈现周期性的结构特征，它是图像中某些结构单元按某种规则排列而成的规则图案，反映图像纹理基元灰度周期性重复变化的规律。

三、图像的频率特性

与一维时间函数波形类似，图像也有空间频率的概念。

如果图像灰度按一定周期变化，它的频率就是在某一坐标轴方向上一个单位长度的距离内，周期函数重复出现的次数，周期表示在同一方向上图像波形重复出现的最小距离。

四、图像灰度变化的梯度特征

图像灰度的梯度反映图像内物体边缘处灰度变化的情况，它描述了图像灰度分布的总体特征。



第2章 习题

P55 (P58第三版) : 2.2、2.11、2.15。

补充题:

1. 能否说物体的亮度越大，人眼观测到的明亮度一定越亮？为什么？