

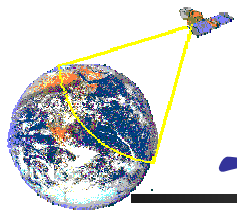
# 数字图像处理与分析

## 第三章 数字图像处理基本运算

刘定生

中科院中国遥感卫星地面站

2004年春季学期

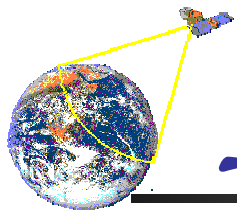


# 上次课复习

## ■ 基本概念

- 图像的表达—— $I=f(x, y)$
- 图像的质量——灰度、对比度、客观评价指标
- 人类的视觉模型、图像的颜色——三基色假说、**RGB**、**HSI**模型、**CIE**色度图
- 图像的描述——二值图像、灰度图像、彩色图像、图像文件格式
- 图像的直方图——定义、基本性质

## ■ 图像的数字化的——**Nyquist**采样定律、量化



# 上次课复习

## ■ 思考题

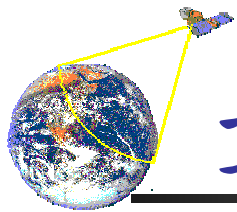
- 我们知道，要构成一幅数字图像需要采样和量化，如果采样和量化都充分细的话，就可以得到好的画质。但是数据量也就变得很大。问当数据量设为一个定值时，在什么时候将采样优先考虑？什么情况下将量化优先考虑？为什么？
- 在图像量化中，有非均匀量化技术。当灰度级低的时候用它比较有效。但是为什么在灰度级级数高时几乎不用？

## ■ 作业

- 自习“视觉的空间性质”与“视觉的时间特性”，初步了解图像时空特性在人眼中的作用。

## ■ 实验

- 利用已有程序文件，通过编程练习打开和显示BMP图像
- 利用Photoshop软件或通过编程，对不同的图像，进行直方图显示、彩色变换、伪彩色处理等，巩固本单元所学内容



## 第三章 数字图像处理基本运算

### ■ 图像的像素级运算

#### ➤ 点运算

- ✓ 线性点运算、非线性点运算

#### ➤ 代数运算

- ✓ 加法、减法、乘法、除法

#### ➤ 逻辑运算

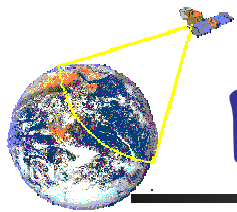
- ✓ 求反、异或、或、与

### ■ 图像的空域变换

- ✓ 几何变换

- ✓ 非几何变换

  - ◆ 直方图变换

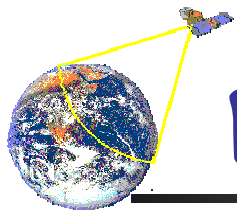


# 图像运算——点运算

## ■ 线性点运算

$$I_{out}(x, y) = a * I_{in}(x, y) + b$$

- $a=1, b=0$ : 恒等
- $a<0$ : 黑白反转
- $|a|>1$ : 增加对比度
- $|a|<1$ : 减小对比度
- $b>0$ : 增加亮度
- $b<0$ : 减小亮度



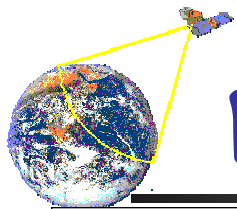
# 图像运算——点运算

## ■ 非线性点运算

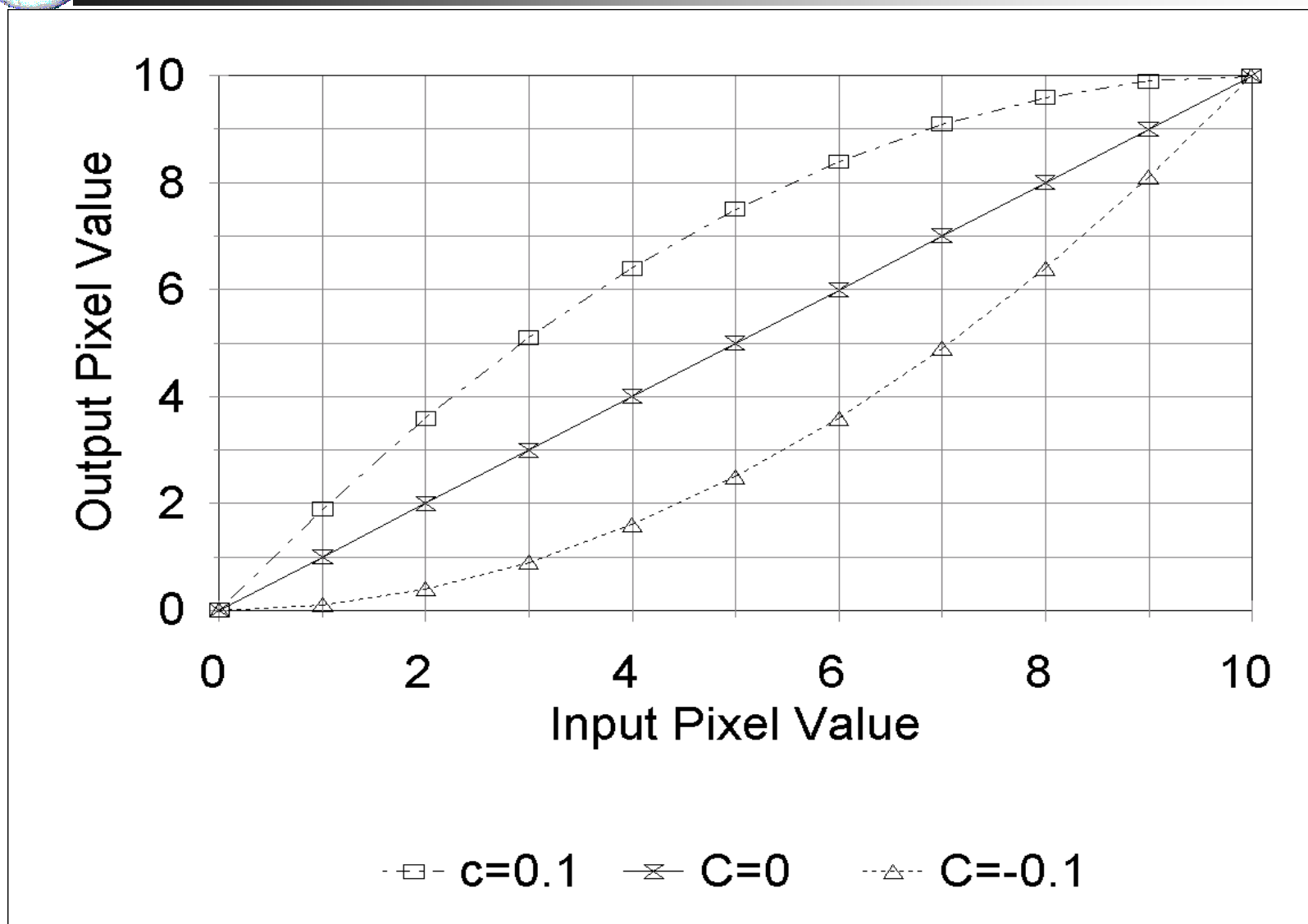
$$f(I(x, y)) = I(x, y) + C * I(x, y) * (I(x, y)_m - I(x, y))$$

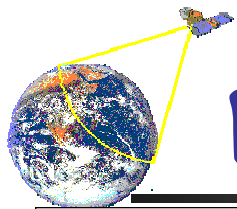
➤  $C < 0$ , 增强中间部分亮度

➤  $C > 0$ , 减小中间部分亮度

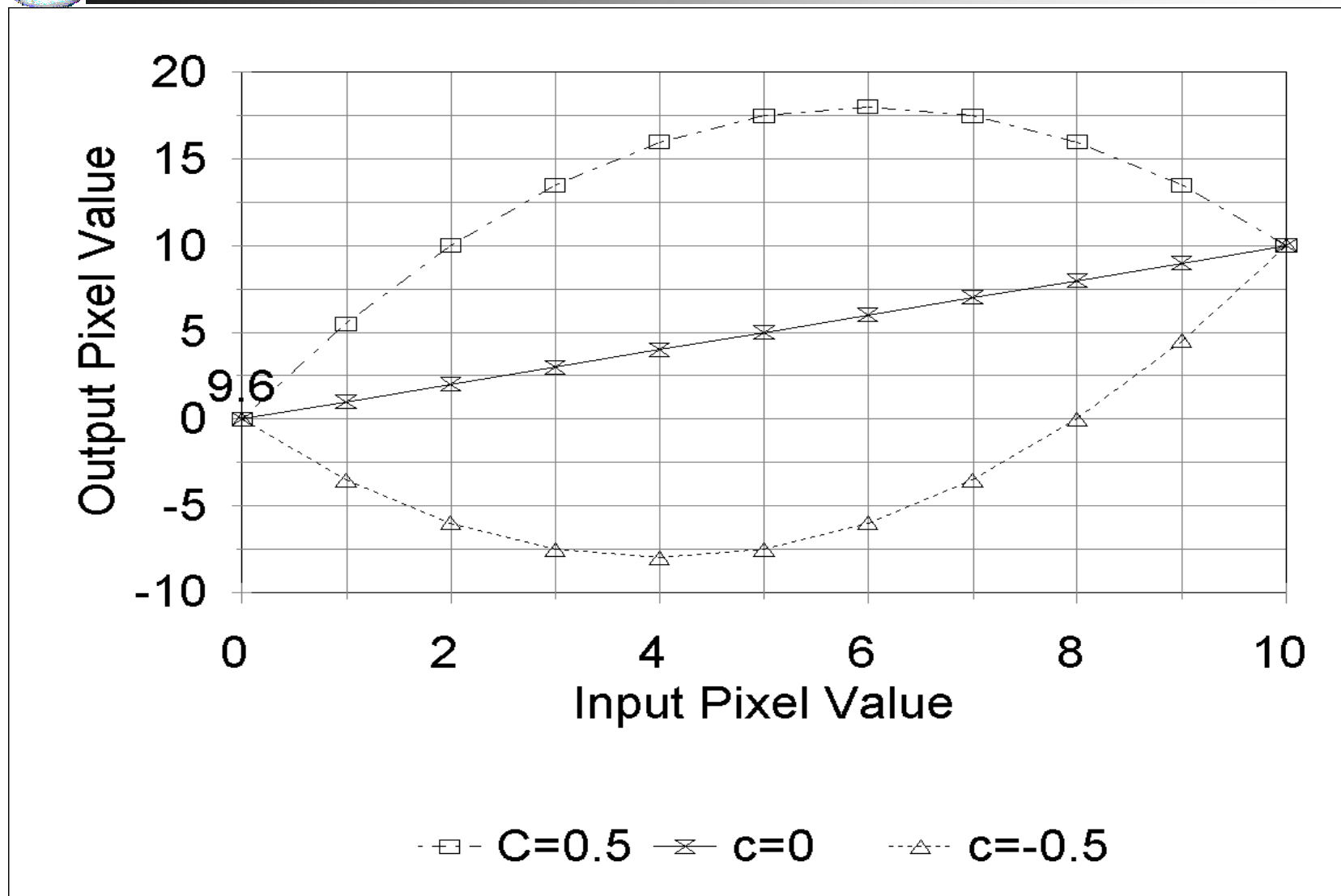


# 图像运算—点运算

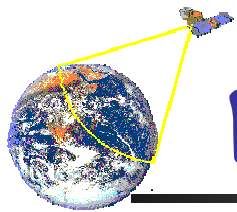




## 图像运算—点运算







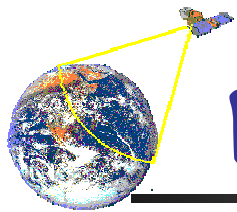
# 图像运算—代数运算

## ■ 加法运算的定义

$$C(x,y) = A(x,y) + B(x,y)$$

## ■ 主要应用举例

- 去除“叠加性”噪音
- 生成图像叠加效果



## 图像运算—代数运算

### ■ 去除“叠加性”噪音

对于原图像 $f(x, y)$ ，有一个噪音图像集

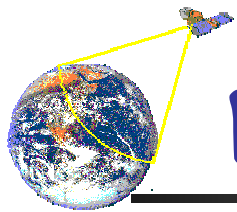
$$\{ g_i(x, y) \} \quad i=1, 2, \dots, M$$

其中： $g_i(x, y) = f(x, y) + h(x, y)_i$

$M$ 个图像的均值定义为：

$$g(x, y) = (g_0(x, y) + g_1(x, y) + \dots + g_M(x, y)) / M$$

当：噪音 $h(x, y)_i$ 为互不相关，且均值为0时，  
上述图像均值将降低噪音的影响。



# 图像运算—代数运算

## ■ 生成图像叠加效果

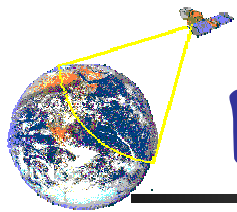
对于两个图像 $f(x,y)$ 和 $h(x,y)$ 的均值有:

$$g(x, y) = \frac{1}{2} f(x, y) + \frac{1}{2} h(x, y)$$

会得到二次曝光的效果。推广这个公式为:

$$g(x,y) = \alpha f(x,y) + \beta h(x,y) \quad \text{其中 } \alpha + \beta = 1$$

我们可以得到各种图像合成的效果，也可以用于两张图片的衔接



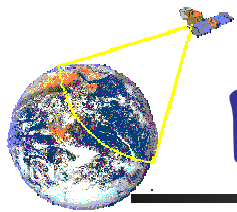
# 图像运算—代数运算

## ■ 减法的定义

$$C(x,y) = A(x,y) - B(x,y)$$

## ■ 主要应用举例

- 去除不需要的叠加性图案
- 检测同一场景两幅图像之间的变化



# 图像运算—代数运算

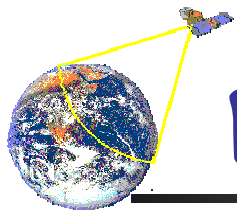
## ■ 去除不需要的叠加性图案

设：背景图像 $b(x,y)$ ，前景背景混合图像 $f(x,y)$

$$g(x,y) = f(x,y) - b(x,y)$$

$g(x,y)$  为去除了背景的图像。

电视制作的蓝屏技术就基于此



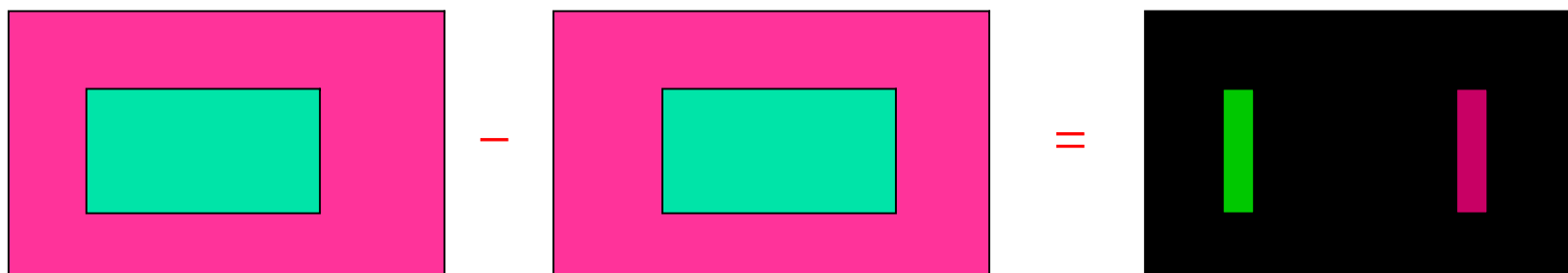
## 图像运算—代数运算

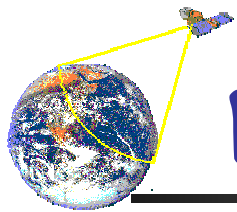
- 检测同一场景两幅图像之间的变化

设： 时间1的图像为 $T_1(x,y)$ ,

时间2的图像为 $T_2(x,y)$

$$g(x,y) = T_2(x,y) - T_1(x,y)$$





# 图像运算—代数运算

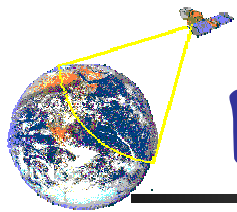
## ■ 乘法的定义

$$C(x,y) = A(x,y) \times B(x,y)$$

## ■ 主要应用举例

### ➤ 图像的局部显示

✓ 用二值蒙板图像与原图像做乘法



# 图像运算——逻辑运算

## ■ 逻辑运算——求反、异或、或、与

### ➤ 求反的定义

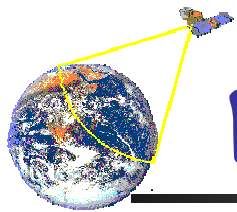
$$g(x,y) = R - f(x,y)$$

$R$ 为 $f(x, y)$ 的灰度级。

### ➤ 主要应用举例

- ✓ 获得一个图像的负像
- ✓ 获得一个子图像的补图像





# 图像运算——逻辑运算

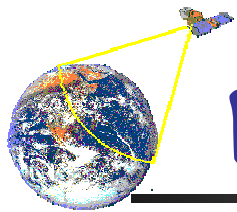
## ■ 逻辑运算——求反、异或、或、与

➤ 异或运算的定义

$$g(x,y) = f(x,y) \oplus h(x,y)$$

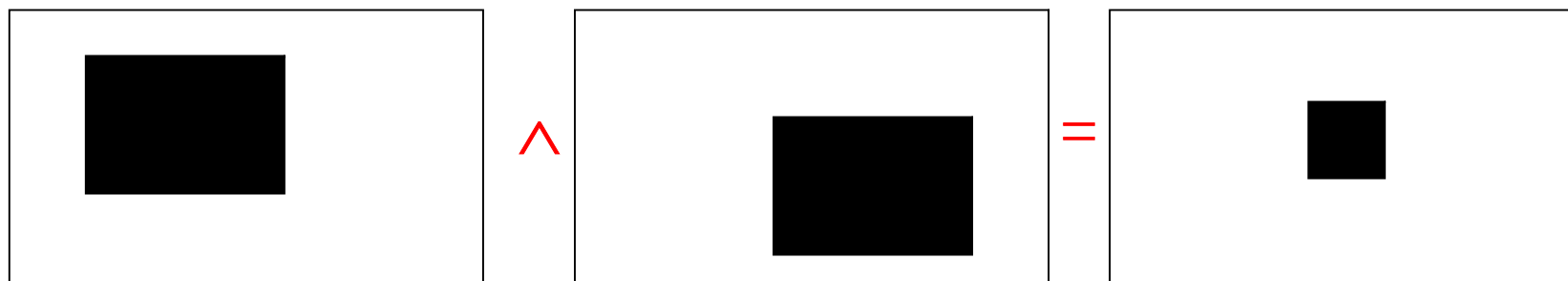
➤ 主要应用举例

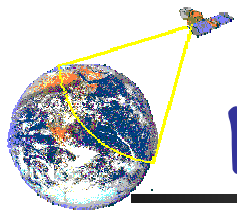
✓ 获得相交子图像



## 图像运算——逻辑运算

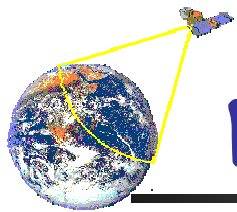
- 逻辑运算——求反、异或、或、与
- 与运算的定义
  - $g(x,y) = f(x,y) \wedge h(x,y)$
- 主要应用举例
  - 求两个子图像的相交子图





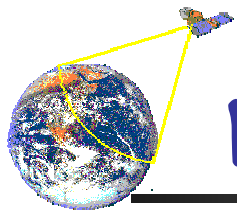
## 图像的空域变换

- 几何变换
  - 基本变换
  - 灰度插值
- 非几何变换
  - 模板运算
  - 灰度变换
  - 直方图变换



## 图像的空域变换——几何变换

- 基本几何变换的定义
- 常用的基本几何变换
  - 平移变换
  - 旋转变换
  - 镜像变换：水平镜像、垂直镜像
  - 放缩变换
  - 拉伸变换
- 数字图像几何变换的计算



## 图像的空域变换——几何变换

### ■ 基本几何变换的定义

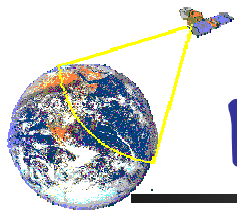
对于原图像 $f(x,y)$ ，坐标变换函数

$$x' = a(x,y); \quad y' = b(x,y)$$

唯一确定了几何变换：

$$g(x',y') = f(a(x,y), b(x,y))$$

$g(x,y)$ 是目标图像



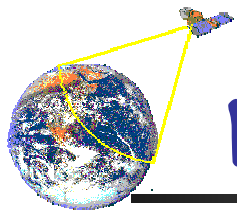
## 图像的空域变换—几何变换

### ■ 平移变换

设:  $a(x,y) = x + x_0$ ;       $b(x,y) = y + y_0$ ;

可有:  $g(x', y') = f(x + x_0, y + y_0)$

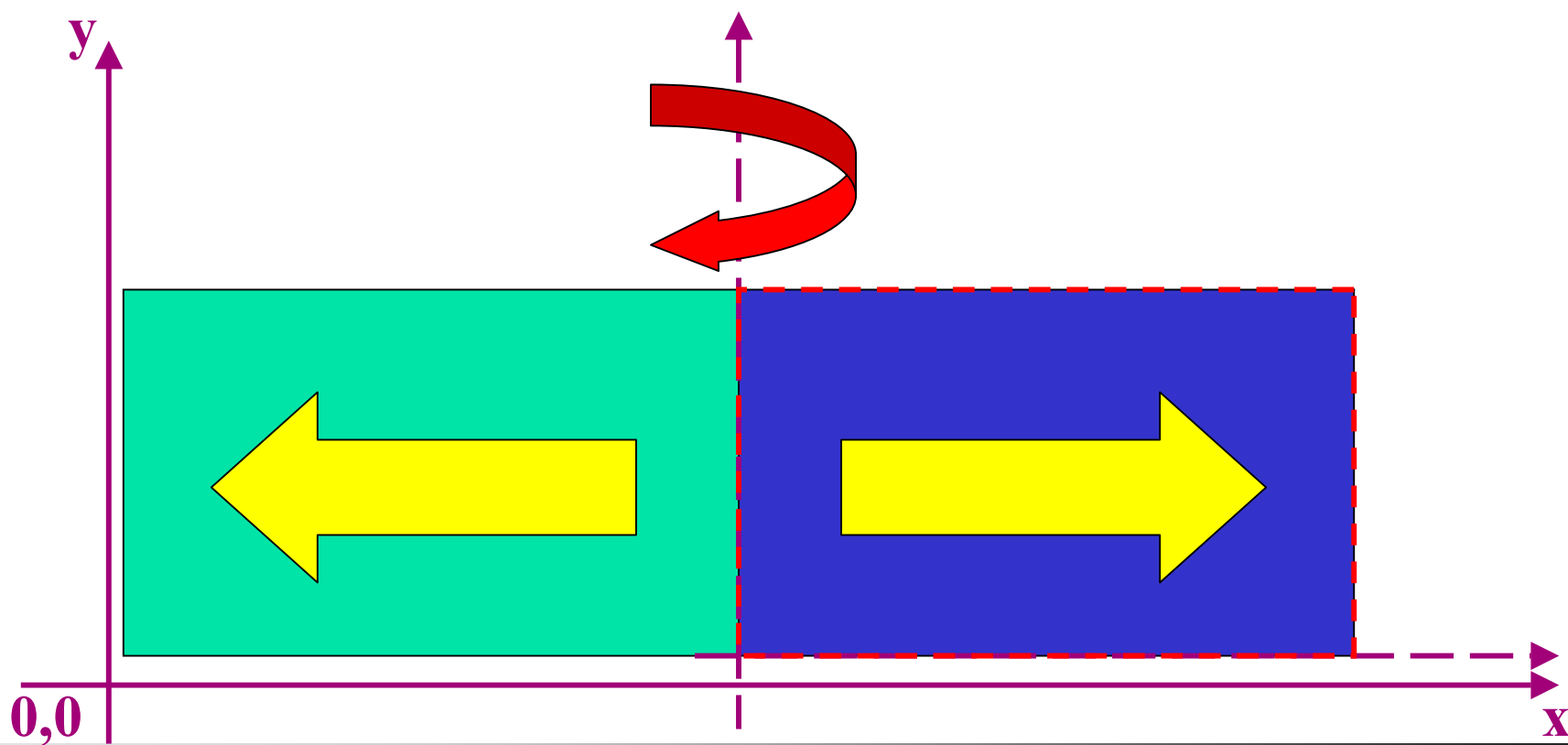
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}_{new} = \begin{bmatrix} a(x,y) \\ b(x,y) \\ 1 \end{bmatrix}_{new} = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{old}$$

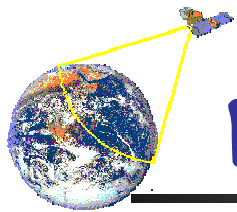


## 图像的空域变换—几何变换

### ■ 水平镜像

$$a(x,y) = -x; \quad b(x,y) = y;$$



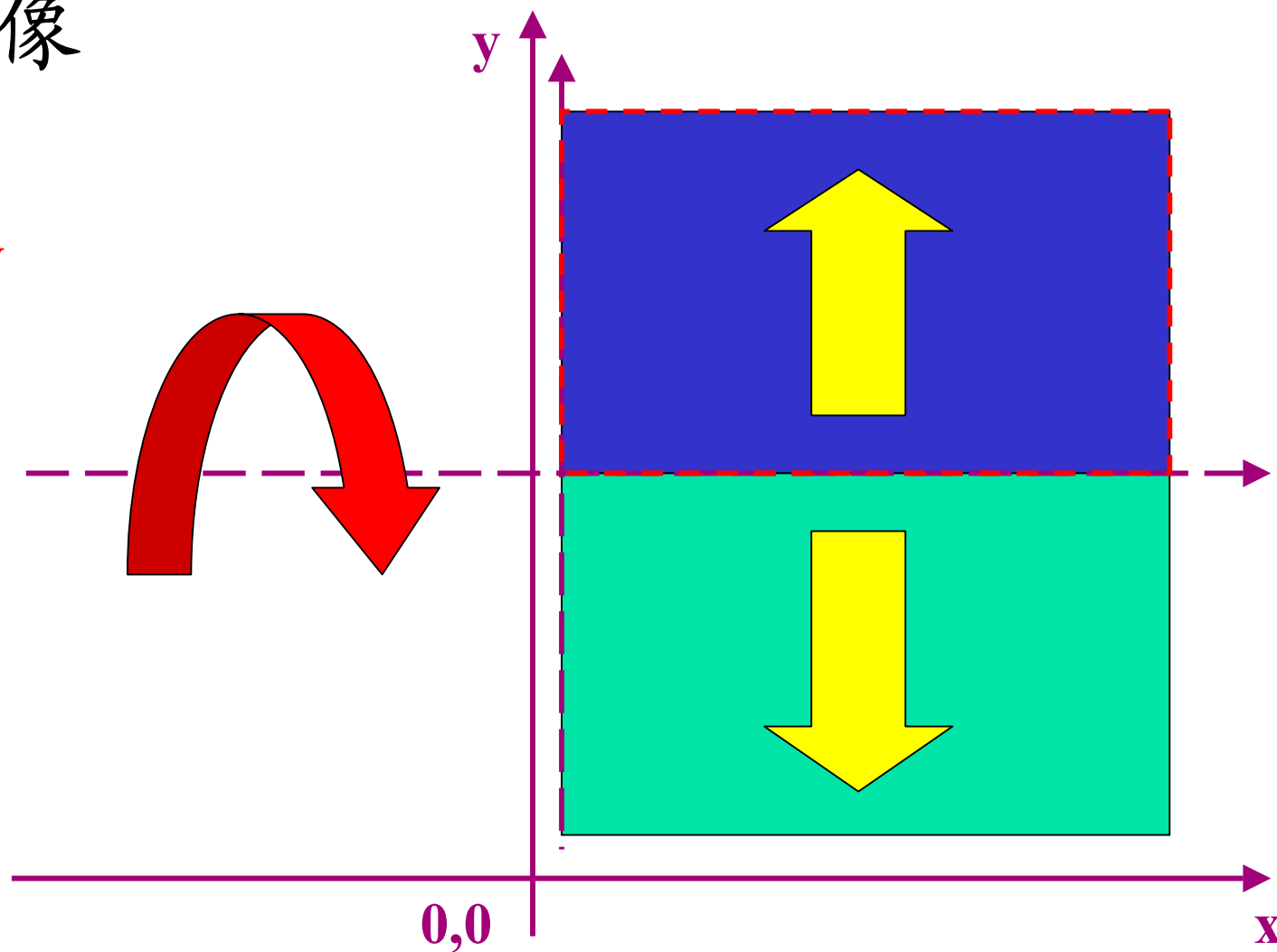


## 图像的空域变换—几何变换

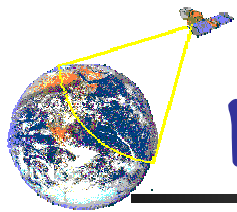
### ■ 垂直镜像

$$a(x,y) = x;$$

$$b(x,y) = -y$$







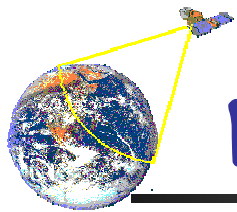
## 图像的空域变换—几何变换

### ■ 旋转变换:绕原点旋转 $\alpha$ 度

设:  $a(x,y) = x * \cos(\alpha) - y * \sin(\alpha)$

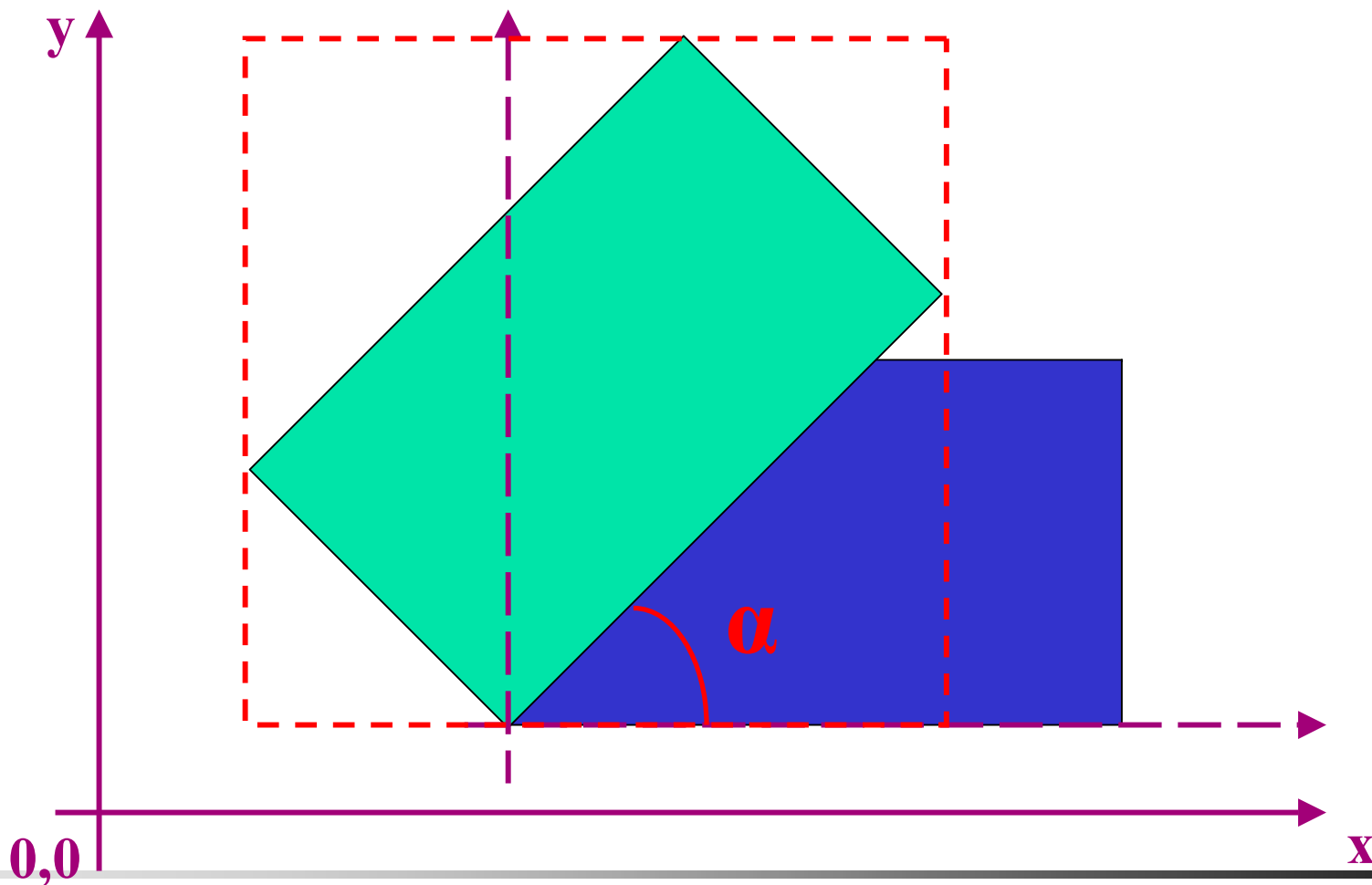
$b(x,y) = x * \sin(\alpha) + y * \cos(\alpha)$

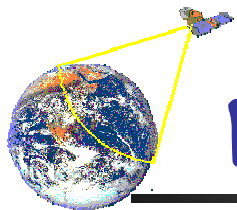
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}_{new} = \begin{bmatrix} a(x,y) \\ b(x,y) \\ 1 \end{bmatrix}_{new} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{old}$$



## 图像的空域变换—几何变换

### ■ 旋转变换:绕原点旋转 $\alpha$ 度

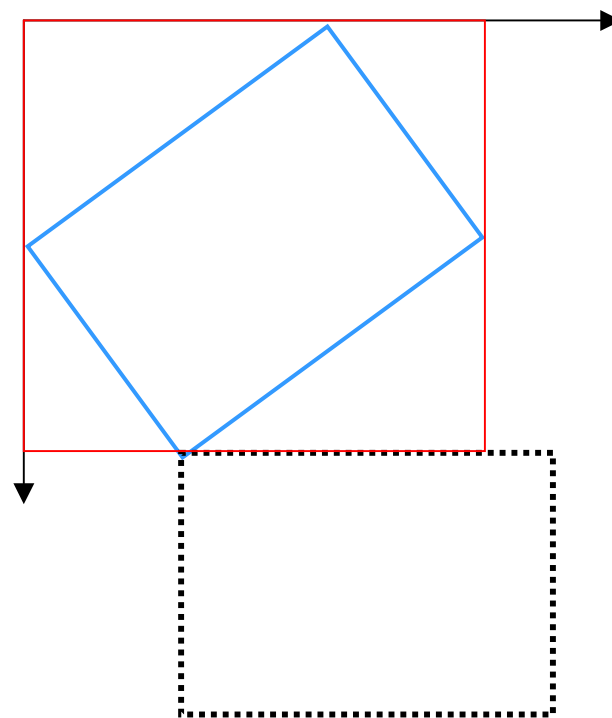
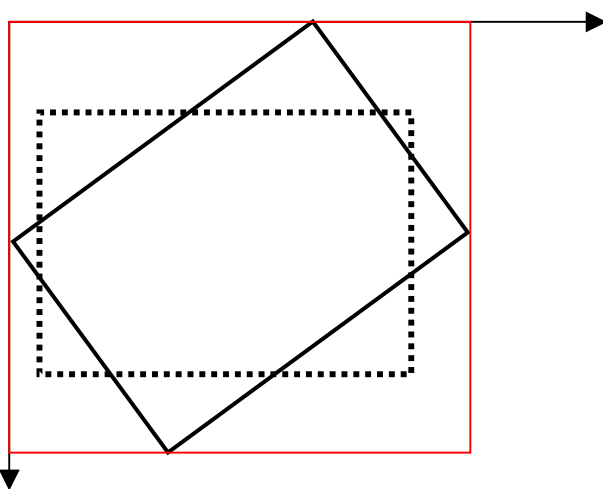


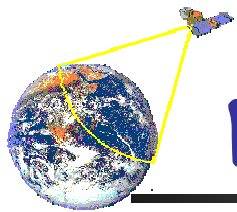


# 图像的空域变换—几何变换

## ■ 旋转变换的注意点

- 1) 图像旋转之前，为了避免信息的丢失，一定有平移坐标，具体的做法有如图所示的两种方法。



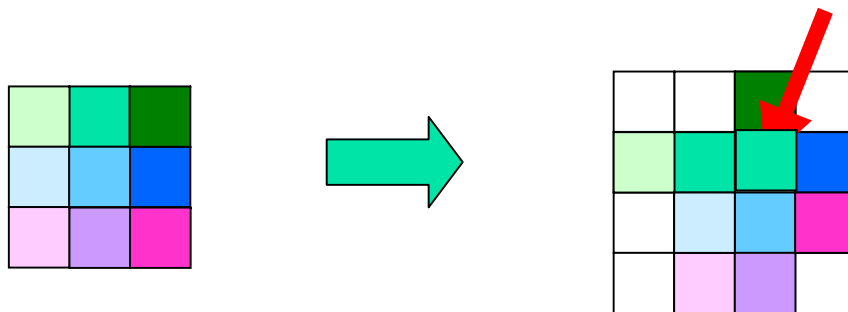


# 图像的空域变换——几何变换

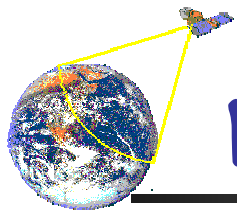
## ■ 旋转变换的注意点

图像的旋转注意点:

- 2) 图像旋转之后, 会出现许多的空洞点, 对这些空洞点必须进行填充处理, 否则画面效果不好。称这种操作为插值处理。



经过插值处理之后, 图像效果就变得自然。



## 图像的空域变换——几何变换

### ■ 缩放变换:

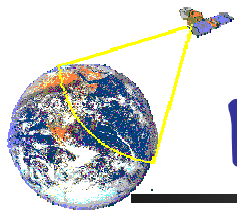
x方向缩放c倍, y方向缩放d倍

$$a(x,y) = x \times c; \quad b(x,y) = y \times d;$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}_{new} = \begin{bmatrix} a(x,y) \\ b(x,y) \\ 1 \end{bmatrix}_{new} = \begin{bmatrix} 1/c & 0 & 0 \\ 0 & 1/d & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{old}$$

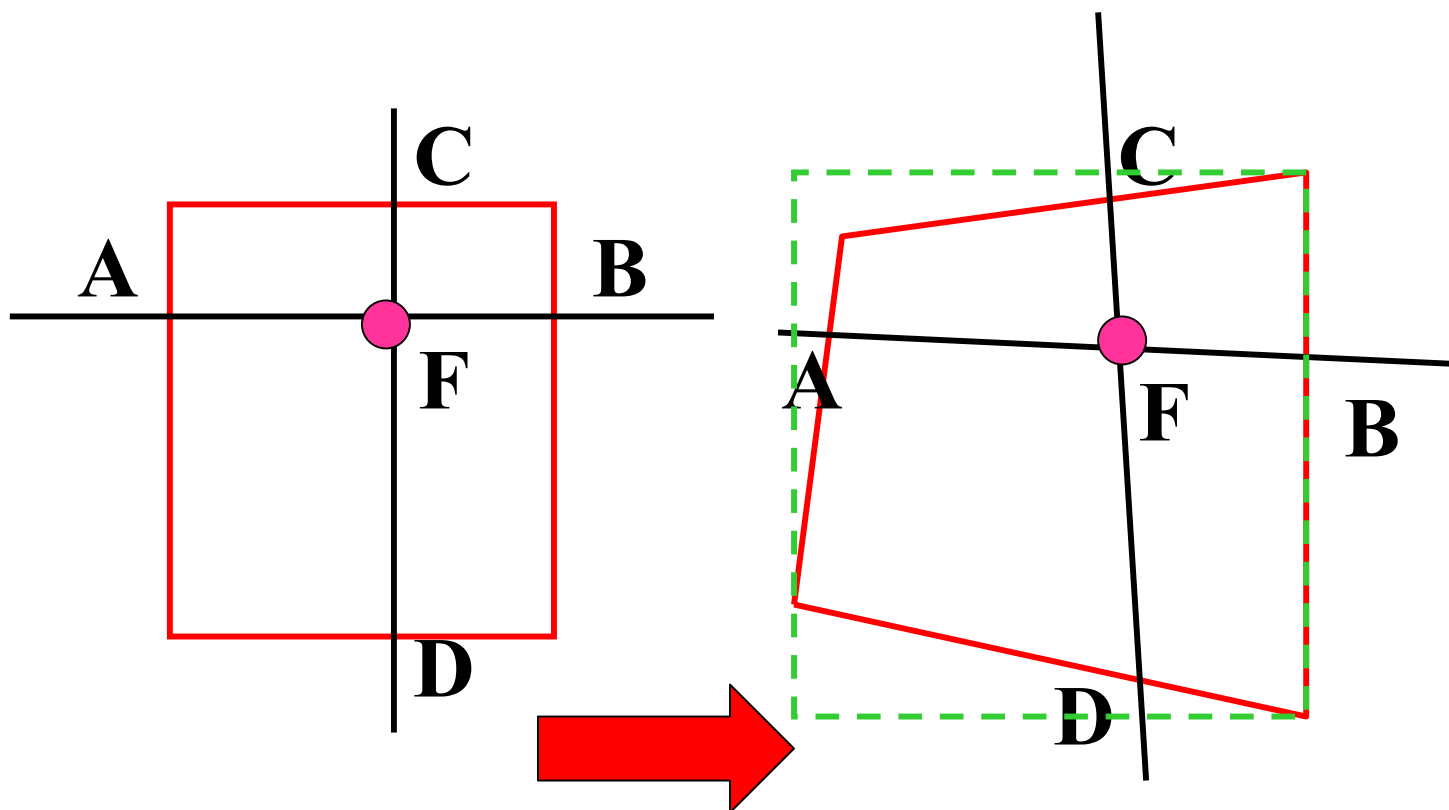
c, d 相等, 按比例缩放

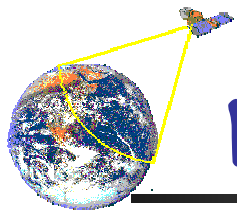
c, d 不相等, 不按比例缩放——几何畸变



# 图像的空域变换—几何变换

## ■ 变形变换





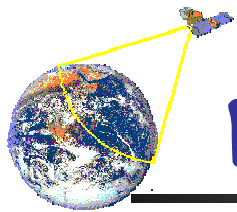
## 图像的空域变换——几何变换

### ➤ 例：图像的错切变换

图像的错切变换实际上是景物在平面上的非垂直投影效果。

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

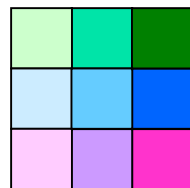
$$\begin{cases} x' = x \\ y' = y + d_y x \end{cases} \quad (y \text{ 方向的错切})$$



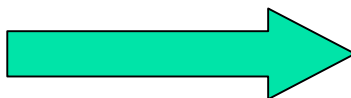
## 图像的空域变换—几何变换

### ➤ 例：图像的错切变换

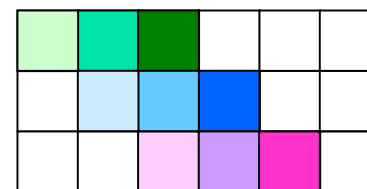
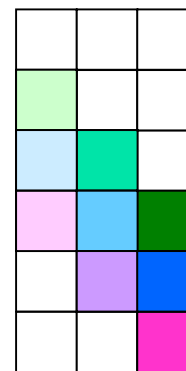
- ✓ 可以看到，错切之后原图像的像素排列方向改变。与旋转不同的是，x方向与y方向独立变化。



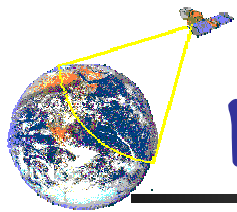
$$d_y = -1$$



$$d_x = 1$$







# 图像的空域变换——几何变换

## ■ 基本几何变换的特征

### ➤ 坐标空间的变化

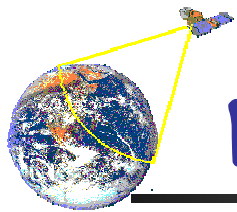
- ✓ 范围发生变化

- ✓ 大小发生变化

### ➤ 像素值的变化

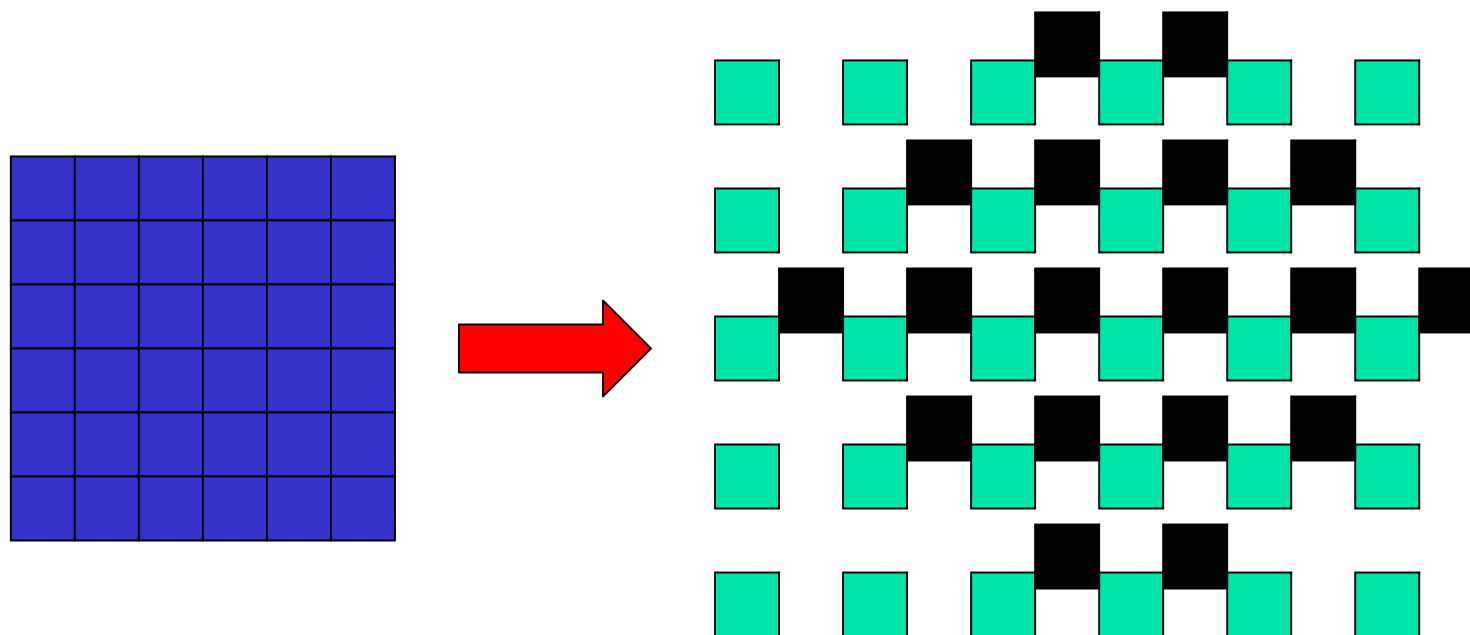
- ✓ 像素值不发生变化——位置改变

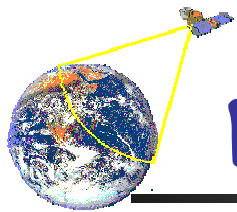
- ✓ 像素值发生变化——旋转、缩放、变形变换



## 图像的空域变换—几何变换

➤ 旋转、缩放、变形变换中的漏点、不规则点问题





# 图像的空域变换——几何变换

## ■ 离散几何变换的计算问题

### ➤ 空间坐标

- ✓ 向前映射法

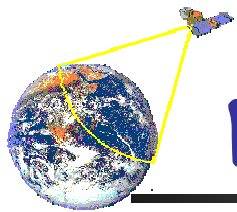
- ✓ 向后映射法

### ➤ 像素值计算——灰度插值（重采样）

- ✓ 最近邻插值法

- ✓ 双线性插值（一阶插值）

- ✓ 高阶插值



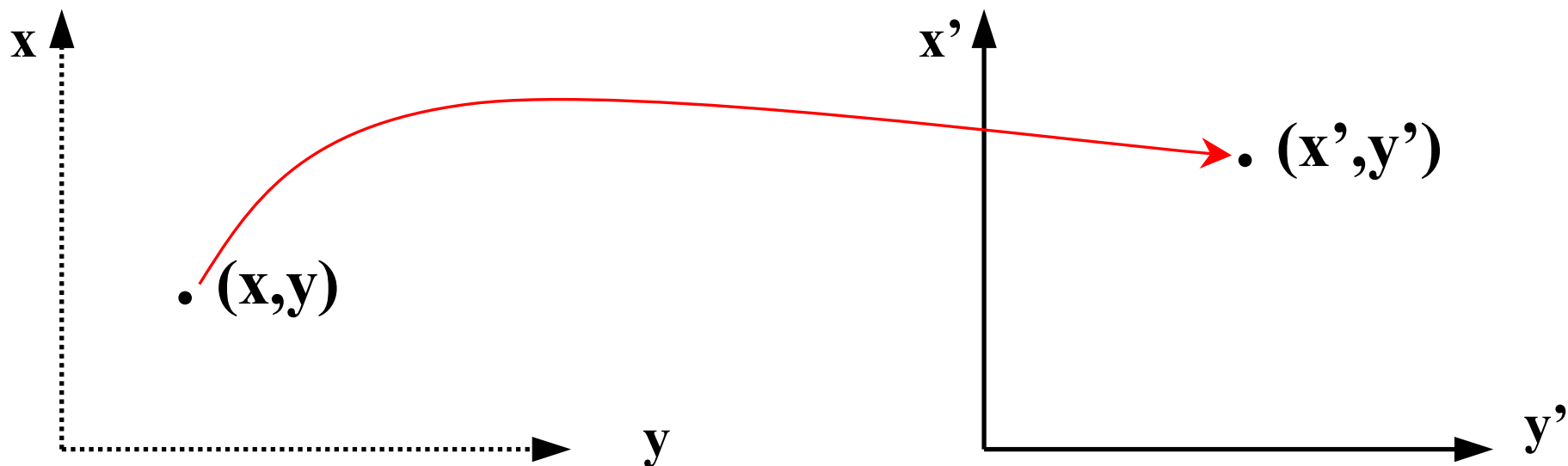
## 图像的空域变换—几何变换

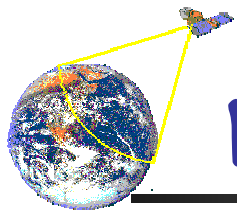
### ■ 向前映射算法

$$g(x',y') = f(a(x,y), b(x,y));$$

➤ 从原图像坐标计算出目标图像坐标

✓ 镜像、平移变换使用这种计算方法





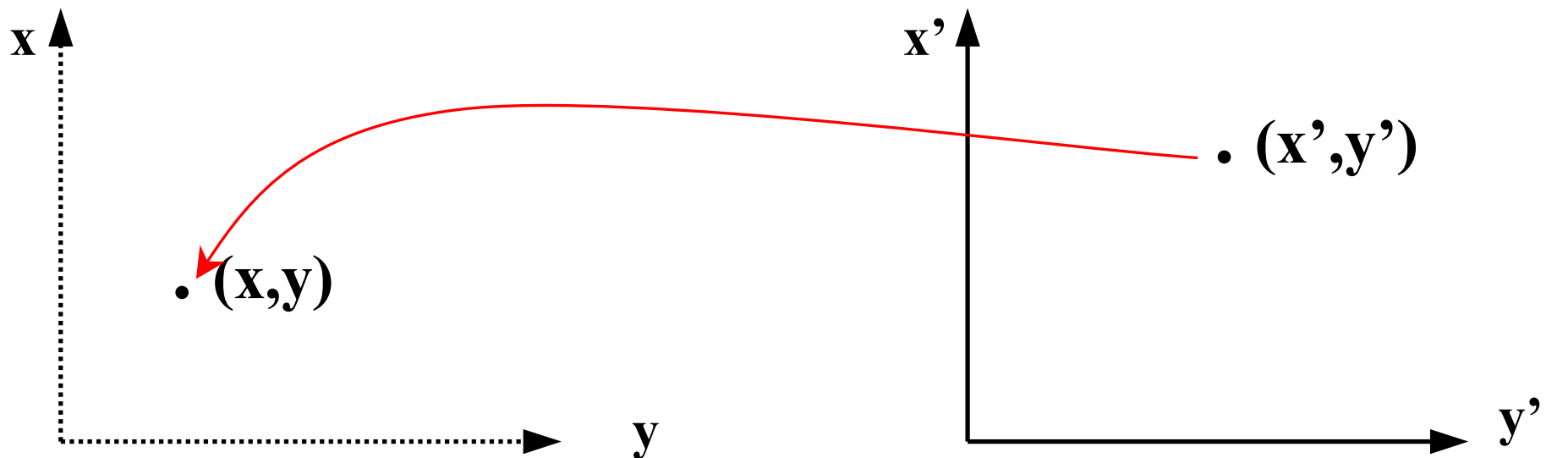
## 图像的空域变换—几何变换

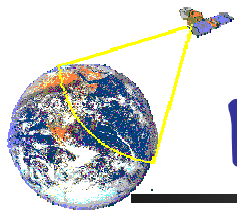
### ■ 向后映射算法

$$g(a'(x,y), b'(x,y)) = f(x,y);$$

➤ 从结果图像的坐标计算原图像的坐标

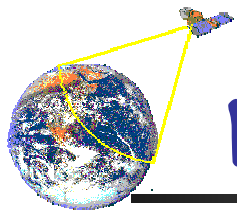
✓ 旋转、缩放、变形可以使用





## 图像的空域变换——几何变换

- 复合变换——多种变换的计算
  - 注意计算的顺序
  - 将多级变换合并为一级变换
- 例：围绕任意坐标点的旋转（ $x_0, y_0$ ）
  - (1) 将（ $x_0, y_0$ ）点平移至坐标原点（ $0, 0$ ）
  - (2) 旋转
  - (3) 平移回（ $x_0, y_0$ ）点

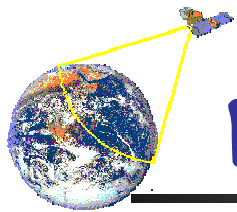


## 图像的空域变换——几何变换

### ■ 复合变换——多种变换的计算

$$(1) \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}_{new} = \begin{bmatrix} a(x, y) \\ b(x, y) \\ 1 \end{bmatrix}_{new} = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{transl}$$

$$(2) \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{transl} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{trans \& rot}$$



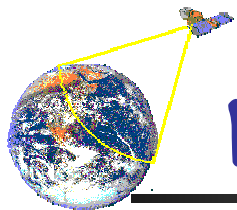
## 图像的空域变换—几何变换

$$(3) \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{trans \& rot} = \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{old}$$

通过矩阵的线性运算形成一次变换式

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}_{new} = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{tl \& r \& rtl}$$





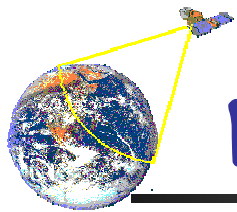
## 图像的空域变换——灰度插值

### ■ 灰度插值——最近邻插值法

➤ 选择最临近点像素灰度值

$(x',y')$  点像素的灰度值为原图中  $(x,y)$  点的像素值

$(x,y)$		$(x+1,y)$	
	$\bullet (x',y')$		
$(x,y+1)$		$(x+1,y+1)$	



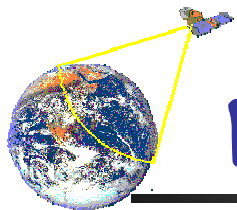
## 图像的空域变换——灰度插值

### ■ 灰度插值——最近邻插值法

#### ➤ 特点

- ✓ 简单快速
  - ✓ 灰度保真性好
  - ✓ 误差较大
  - ✓ 视觉特性较差
- 马赛克效应





## 图像的空域变换——灰度插值

### ■ 灰度插值——双线性插值（一阶插值）

$$f'(x',y')=$$

$$a \cdot f(x,y) + b \cdot f(x,y+1)$$

$$f''(x',y')=$$

$$c \cdot f(x,y) + d \cdot f(x+1,y)$$

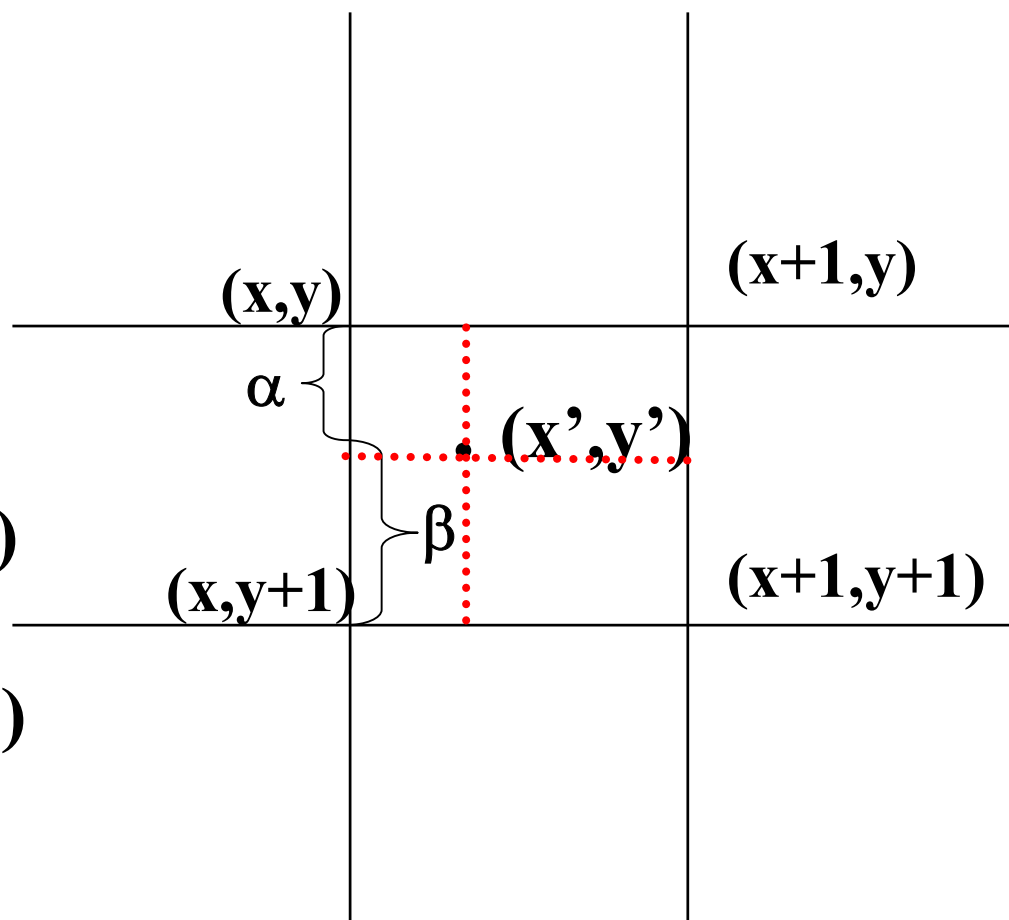
$$f'''(x',y')=$$

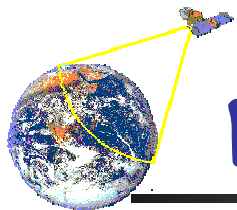
$$u \cdot f(x,y+1) + v \cdot f(x+1,y+1)$$

$$f''''(x',y')=$$

$$w \cdot f(x+1,y) + z \cdot f(x+1,y+1)$$

$$f(x',y')=.....$$





## 图像的空域变换——灰度插值

### ■ 双线性插值——简化计算方法

➤ 应用双曲抛物面方程

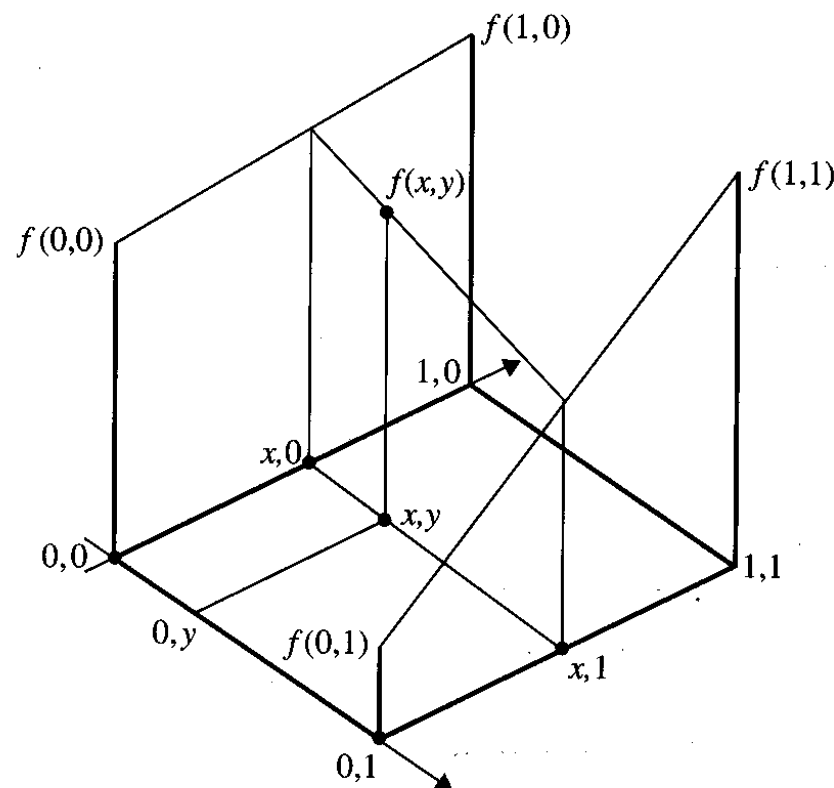
$$f(x, y) = ax + by + cxy + d$$

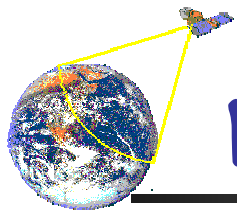
➤ 归一化坐标值

$$0 > x > 1, 0 > y > 1$$

➤ 可有:

$$\begin{aligned} f(x, y) = & [f(1, 0) - f(0, 0)]x + [f(0, 1) - f(0, 0)]y \\ & + [f(1, 1) + f(0, 0) - f(0, 1) - f(1, 0)]xy + f(0, 0) \end{aligned}$$





## 图像的空域变换——灰度插值

### ■ 线性运算理论

➤ 如果 $f$ 是一个线性运算，则

$$f(x_1 + x_2, y) = f(x_1, y) + f(x_2, y)$$

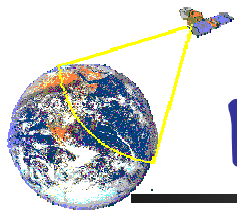
➤ 但对

$$f(x, y) = ax + by + cxy + d$$

➤ 则有

$$f(x_1 + x_2, y) = a(x_1 + x_2) + by + c(x_1 + x_2)y + d$$

$$f(x_1 + x_2, y) = ax_1 + ax_2 + by + cx_1y + cx_2y + d$$



## 图像的空域变换——灰度插值

### ■ 灰度插值——双线性插值（一阶插值）

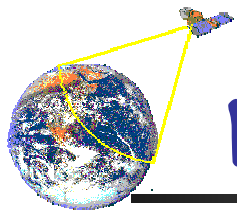
- 双线性插值一般理论——双曲抛物面方程插值

$$f(x, y) = ax + by + cxy + d$$

- 需得到四个未知参数——利用四个已知点

- 特点

- ✓ 计算中较为充分地考虑相邻各点的特征，具有灰度平滑过渡特点
- ✓ 一般情况下可得到满意结果
- ✓ 具有低通滤波特性，使图像轮廓模糊
- ✓ 平滑作用使图像细节退化，尤其在放大时
- ✓ 不连续性会产生不希望的结果

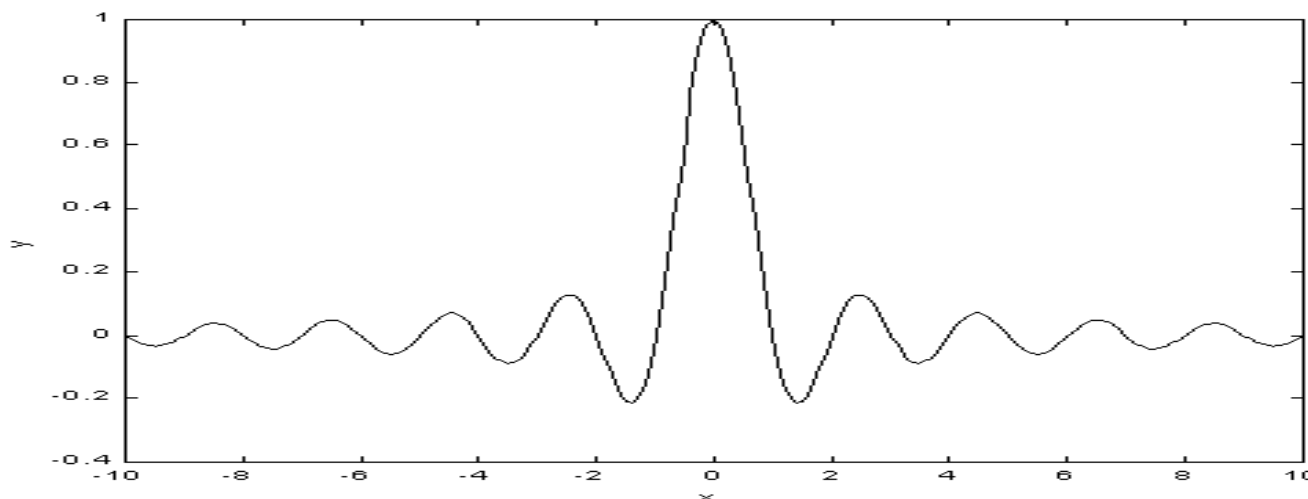


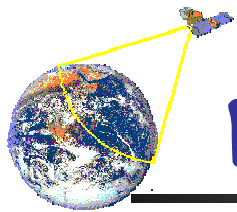
## 图像的空域变换——灰度插值

### ■ 灰度插值——最佳插值函数

➤ 在满足Nyquist条件下，从离散信号 $x(nT_s)$ 可恢复连续信号 $x(t)$ ：

$$\therefore x(t) = \sum_{i=-\infty}^{+\infty} x(nT_s) \operatorname{sinc}\left(\frac{\pi}{T_s}(t - nT_s)\right)$$

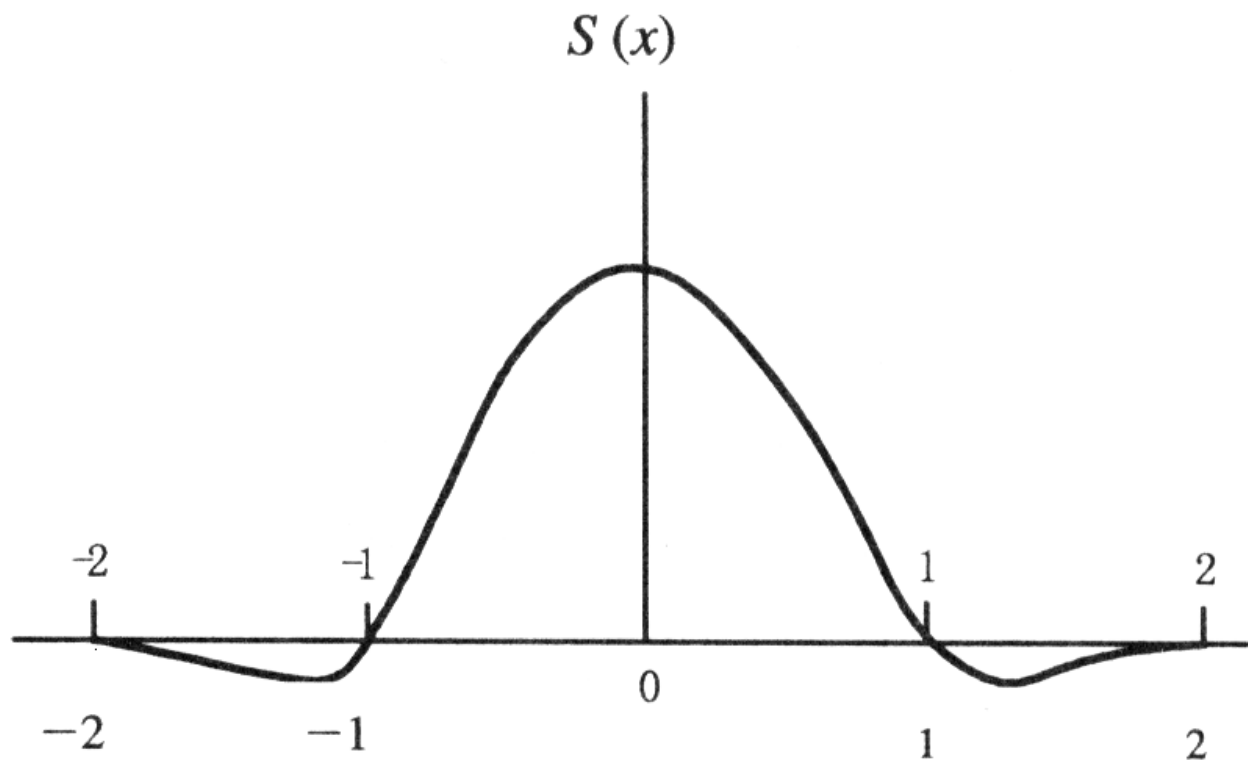




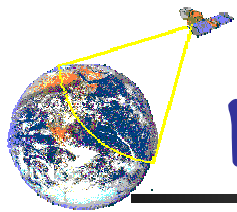
## 图像的空域变换——灰度插值

### ■ 灰度插值——高阶插值

➤ 如果简化计算，仅取原点周围有限范围函数：







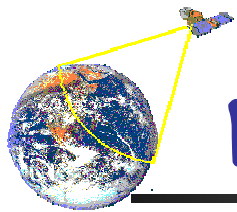
## 图像的空域变换——灰度插值

### ■ 灰度插值——高阶插值

- 并利用三次多项式来近似理论上的最佳插值函数  $\text{sinc}(x)$ :

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

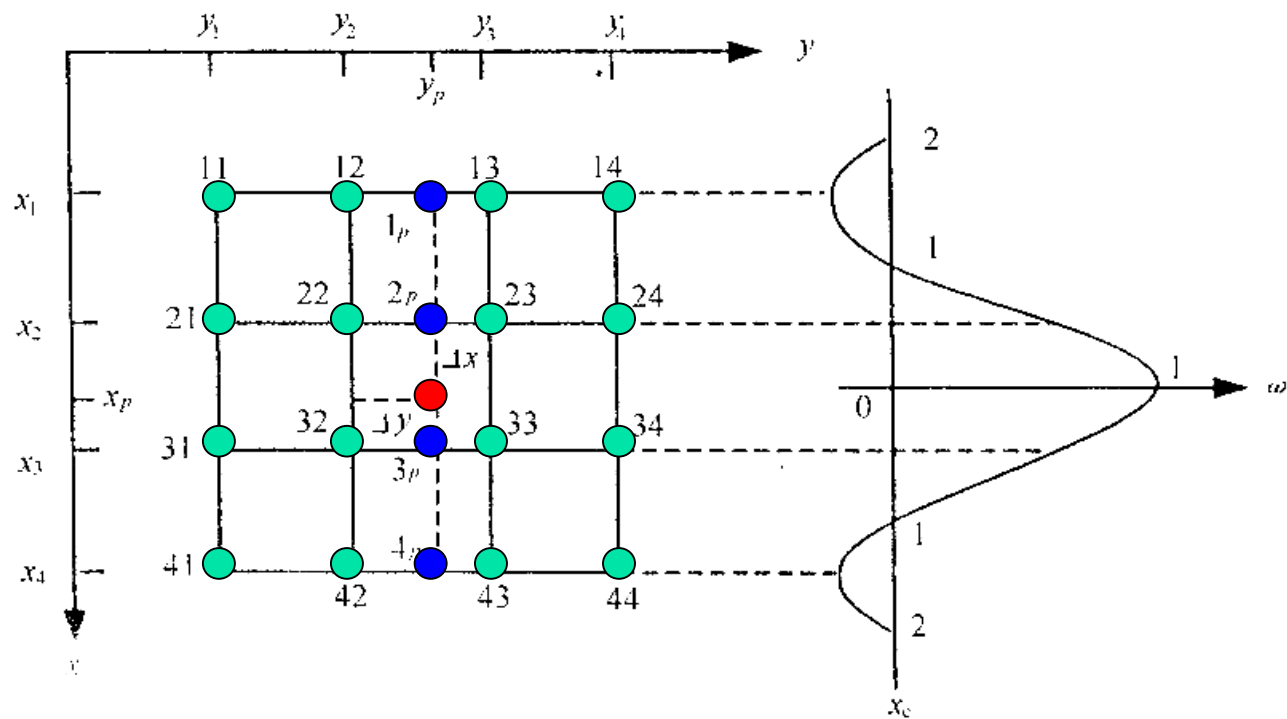
- 由此形成常用的三次卷积插值算法，又称三次内插法，两次立方方法 (Cubic)。CC插值法等



## 图像的空域变换——灰度插值

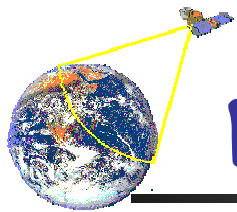
### ■ 灰度插值——三次卷积插值算法实现

➤ 利用待插值点周围的16个邻点像素值:



➤ 首先确定辅助点位 $1p$ ,  $2p$ ,  $3p$ ,  $4p$ 各点亮度值

➤ 再由确定 $p$ 点亮度值



## 图像的空域变换——灰度插值

### ■ 灰度插值——三次卷积插值算法实现

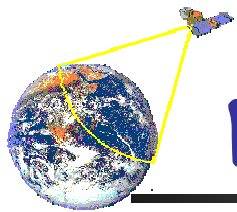
$$I_p = S_x \bullet I \bullet S_y$$

➤ 其中

$$S_x = [S(x_c(1p)) \quad S(x_c(2p)) \quad S(x_c(3p)) \quad S(x_c(4p))]$$

$$S_y = [S(y_c(i1)) \quad S(y_c(i2)) \quad S(y_c(i3)) \quad S(y_c(i4))]^T$$

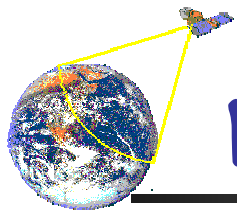
$$I = \begin{bmatrix} I_{11} & I_{12} & I_{13} & I_{14} \\ I_{21} & I_{22} & I_{23} & I_{24} \\ I_{31} & I_{32} & I_{33} & I_{34} \\ I_{41} & I_{42} & I_{43} & I_{44} \end{bmatrix}$$



## 图像的空域变换——灰度插值

### ■ 灰度插值——三次卷积插值算法特点

- 为满足二维Nyquist条件下，最佳重构公式的近似
- 只有图像满足特定的条件，三次卷积插值算法才可获得最佳结果
- 可使待求点的灰度值更好地模拟实际可能值
- 可取得更好的视觉效果
- 三次卷积内插突出的优点是高频信息损失少，可将噪声平滑
- $4 \times 4$ 时，像元均值和标准差信息损失小
- 计算量大为增加

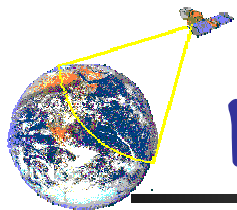


## 图像的空域变换——灰度插值

### ■ 灰度插值——图像处理中内插方法的选择

➤ 内插方法的选择除了考虑图像的显示要求及计算量，还要考虑内插结果对分析的影响

- ✓ 当纹理信息为主要信息时，最近邻采样将严重改变原图像的纹理信息
- ✓ 但灰度信息为主要信息时，双线性内插及三次卷积内插将减少图像异质性，增加图像同质性，其中，双线性内插方法使这种变化更为明显



## 图像的空域变换——非几何变换

- 非几何变换的定义

对于原图像 $f(x,y)$ ，灰度值变换函数

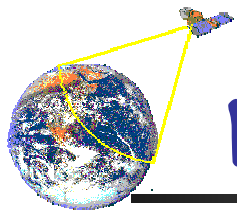
$$T(f(x,y))$$

唯一确定了非几何变换:

$$g(x,y) = T(f(x,y))$$

$g(x,y)$ 是目标图像

- 非几何变换属于像素值的变换，没有几何位置的改变——灰度变换
- 灰度变换的目的是为了改善画质，使图像的显示效果更加清晰



## 图像的空域变换——非几何变换

### ■ 非几何变换的定义

对于彩色原图像 $f(x,y)$ ，颜色值变换函数

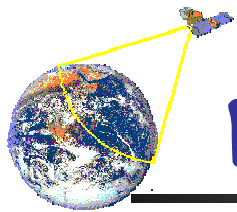
$$T_r(f(x,y)); T_g(f(x,y)); T_b(f(x,y));$$

唯一确定了非几何变换：

$$g_r(x,y) = T_r(f(x,y))$$

$$g_g(x,y) = T_g(f(x,y))$$

$$g_b(x,y) = T_b(f(x,y))$$



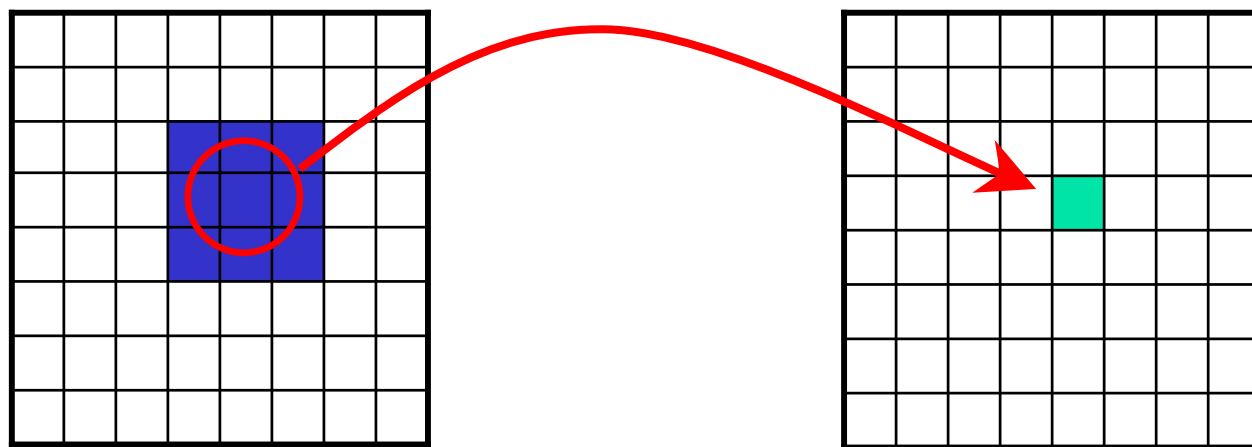
## 图像的空域变换—非几何变换

### ■ 离散非几何变换的计算

➤ 简单变换—像素值一一对应的映射

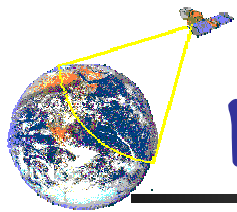
✓ 如伪彩色变换

➤ 复杂变换—同时考虑相邻各点的像素值



✓ 通常通过模板运算进行





# 图像的空域变换——非几何变换

## ■ 模板的定义

➤ 所谓模板就是一个系数矩阵

➤ 模板大小：经常是奇数，如：

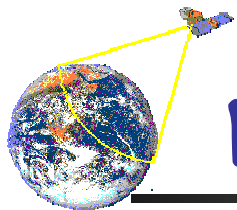
**3x3   5x5   7x7   .....**

➤ 模板系数：矩阵的元素

**$w_1$   $w_2$   $w_3$**

**$w_4$   $w_5$   $w_6$**

**$w_7$   $w_8$   $w_9$**



## 图像的空域变换——非几何变换

### ■ 模板运算的定义

对于某图像的子图像:

$z_1 z_2 z_3$

$z_4 z_5 z_6$

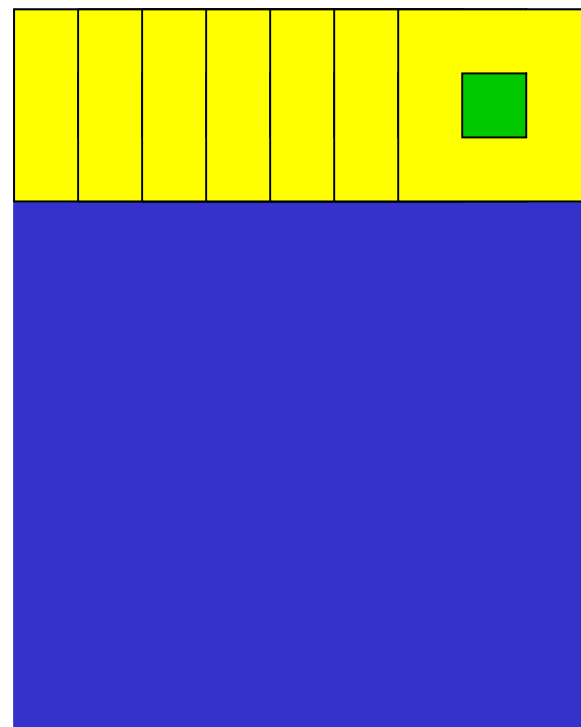
$z_7 z_8 z_9$

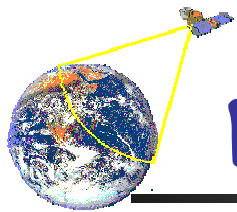
$z_5$ 的模板运算公式为:

$$R = w_1 z_1 + w_2 z_2 + \dots + w_9 z_9$$

$w_1, w_2, \dots, w_9$

为加权系数。

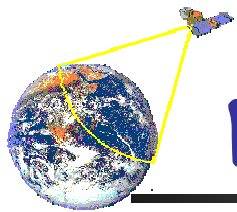




## 图像的空域变换——非几何变换

- 灰度变换（点运算）的定义(1)
  - 对于输入图像 $f(x, y)$ ，灰度变换 $T$ 将产生一个输出图像 $g(x, y)$ ， $g(x, y)$ 的每一个像素值，均取决于 $f(x, y)$ 中对应点的像素值

$$g(x, y) = T(f(x, y))$$



## 图像的空域变换—非几何变换

- 灰度变换（点运算）的定义(2)

对于原图像 $f(x,y)$ ，灰度值变换函数

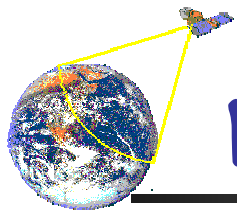
$$T(f(x,y))$$

由于灰度值总是有限个，如：0-255

非几何变换可定义为：

$$R = T(r)$$

其中 $R, r$ 在0-255之间取值



## 图像的空域变换——非几何变换

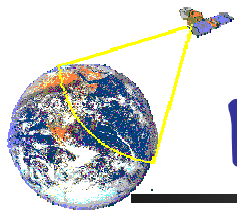
- 灰度变换（点运算）的实现

$R = T(r)$  定义了输入像素值与输出像素之间的映射关系，通常通过查表来实现。

因此灰度级变换也被称为LUT（Look Up Table）变换。

0 1 2 3 4 5 6 7 8 9 ... 250 251 252 253 254 255

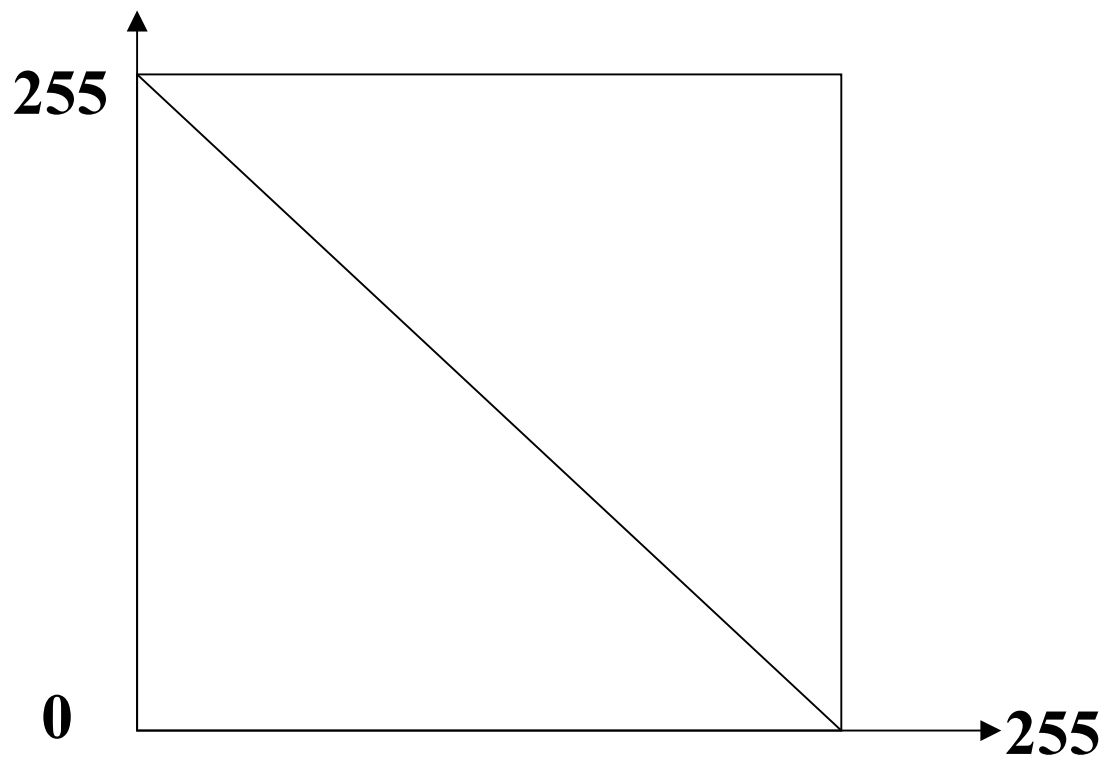
0 3 5 7 9 11 13 15 17 19 ... 252 253 254 254 254 255

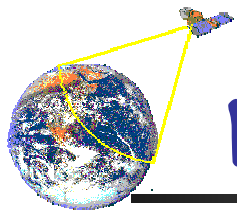


# 图像的空域变换——非几何变换

## ■ 灰度变换例1

➤ 图像求反

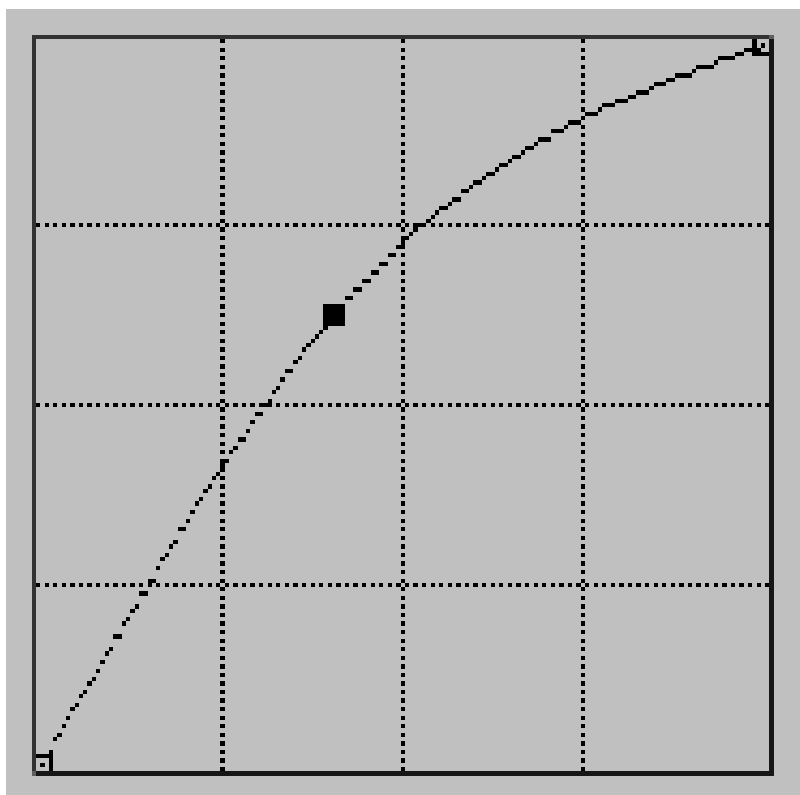


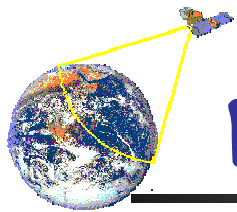


# 图像的空域变换——非几何变换

## ■ 灰度变换例2

➤ 对比度拉伸





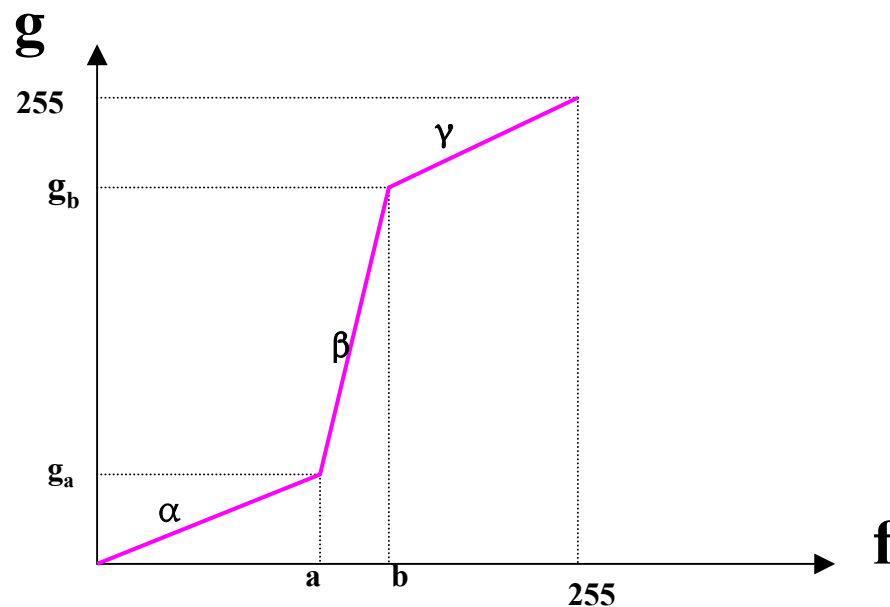
## 图像的空域变换——非几何变换

### ■ 灰度变换例3

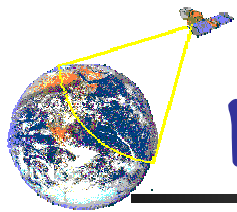
➤ 对比度展宽——突出图像中关心的部分

➤ 方法:

$$g = \begin{cases} \alpha f & 0 \leq f < a \\ \beta(f - a) + g_a & a \leq f < b \\ \gamma(f - b) + g_b & b \leq f < L \end{cases}$$



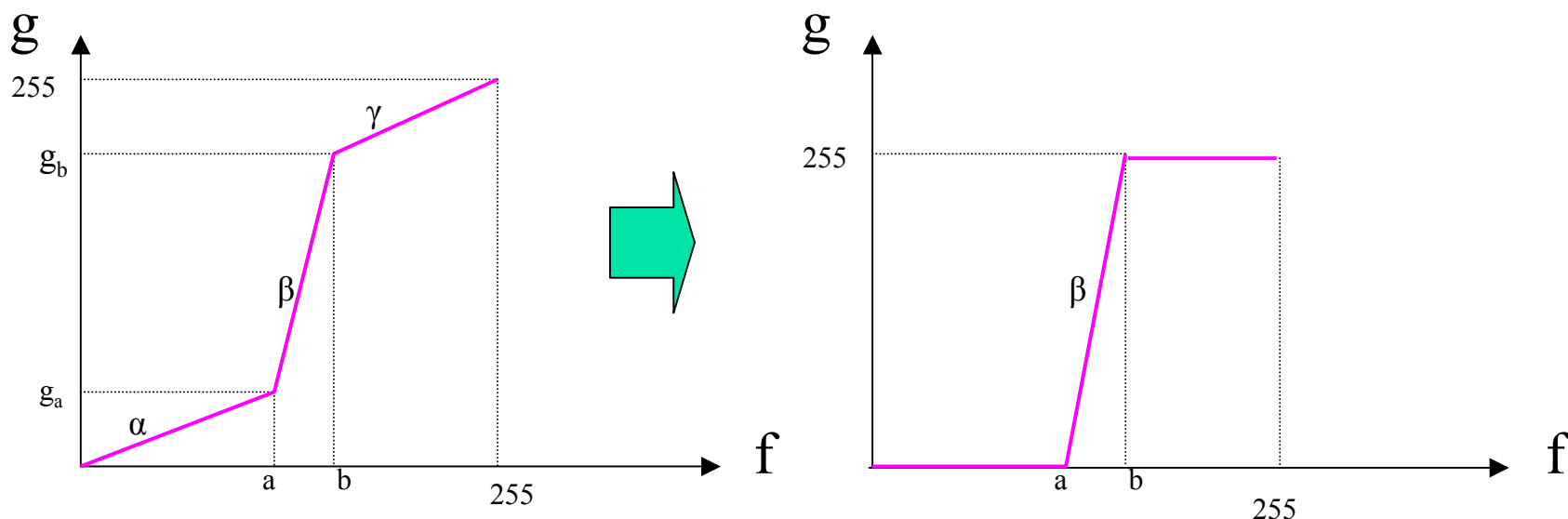


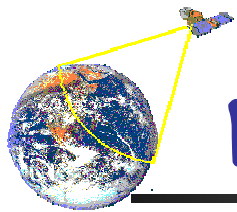


## 图像的空域变换—非几何变换

### ■ 灰度变换例4

➤ 灰度窗——只显示指定灰度级范围内的信息。如：  
 $\alpha = \gamma = 0$

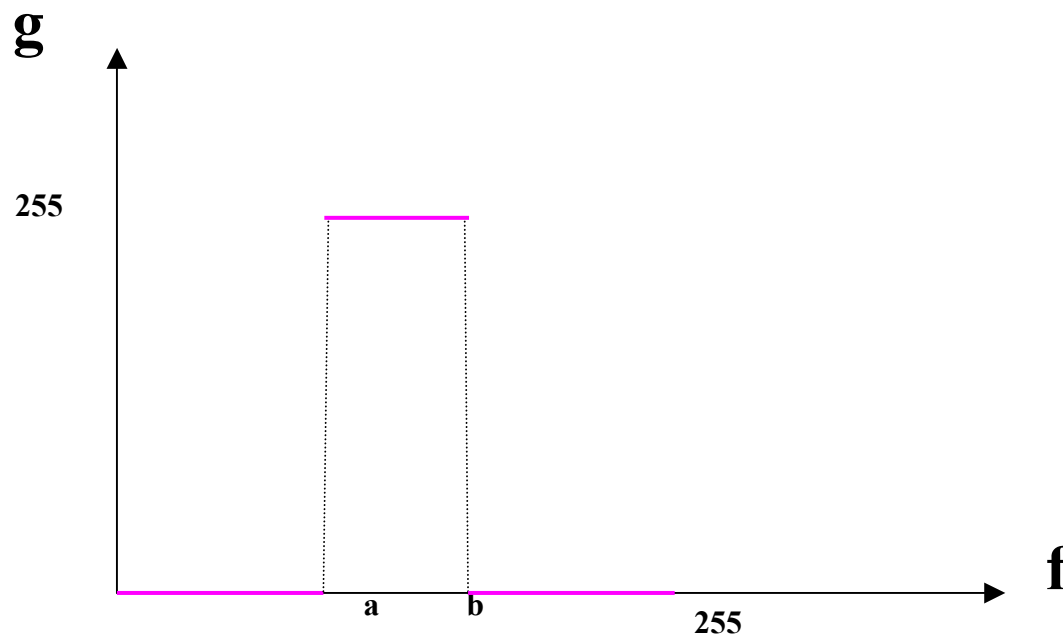


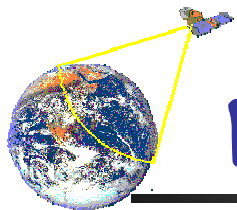


## 图像的空域变换——非几何变换

### ■ 灰度变换例5

➤ 灰度级切片——只保留感兴趣的部分，其余部分置为0





# 图像的空域变换——非几何变换

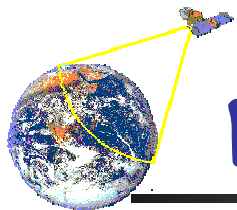
## ■ 灰度变换例6

### ➤ 灰度级修正

- ✓ 通过记录装置把一景物变成一幅图像时，景物上每一点所反射的光，并不是按同一比例转化成图像上相应点的灰度，靠近光轴的光要比远离光轴的光衰减得要少一些。
- ✓ 灰度级修正的目的是：使画面中的每个关心的细节信息通过灰度级修正之后，可以变得清楚可见。

$$I(x, y) = e(x, y) * g(x, y)$$

$$g(x, y) = e^{-1}(x, y) * I(x, y)$$



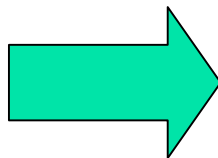
# 图像的空域变换—非几何变换

## ■ 灰度变换例7

➤ 线性动态范围调整:

$$h^*(x, y) = \begin{cases} 0 & h(x, y) \leq a \\ \frac{255}{b-a} h(x, y) - \frac{255a}{(b-a)} & h(x, y) \in (a, b) \\ 255 & h(x, y) \geq b \end{cases}$$

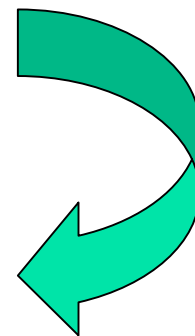
1	3	9	9	8
2	1	3	7	3
3	6	0	6	4
6	8	2	0	5
2	9	2	6	0



2	3	7	7	7
2	2	3	7	3
3	6	2	6	4
6	7	2	2	5
0	7	2	6	2

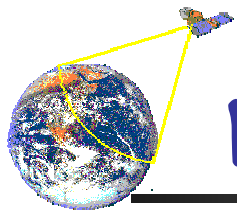
黑: 0 → 2  
白: 9 → 7

$$y = 1.8 * x - 3.6$$



0	2	9	9	9
0	0	2	9	2
2	7	0	7	4
7	9	0	0	5
0	9	0	7	0

作用: 进行亮暗限幅



## 图像的空域变换—直方图变换

### ■ 图像直方图的定义（1）

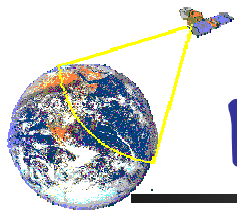
一个灰度级在范围 $[0, L-1]$ 的数字图像，其的直方图是一个离散函数

$$p(r_k) = n_k/n$$

$n$  是图像的像素总数

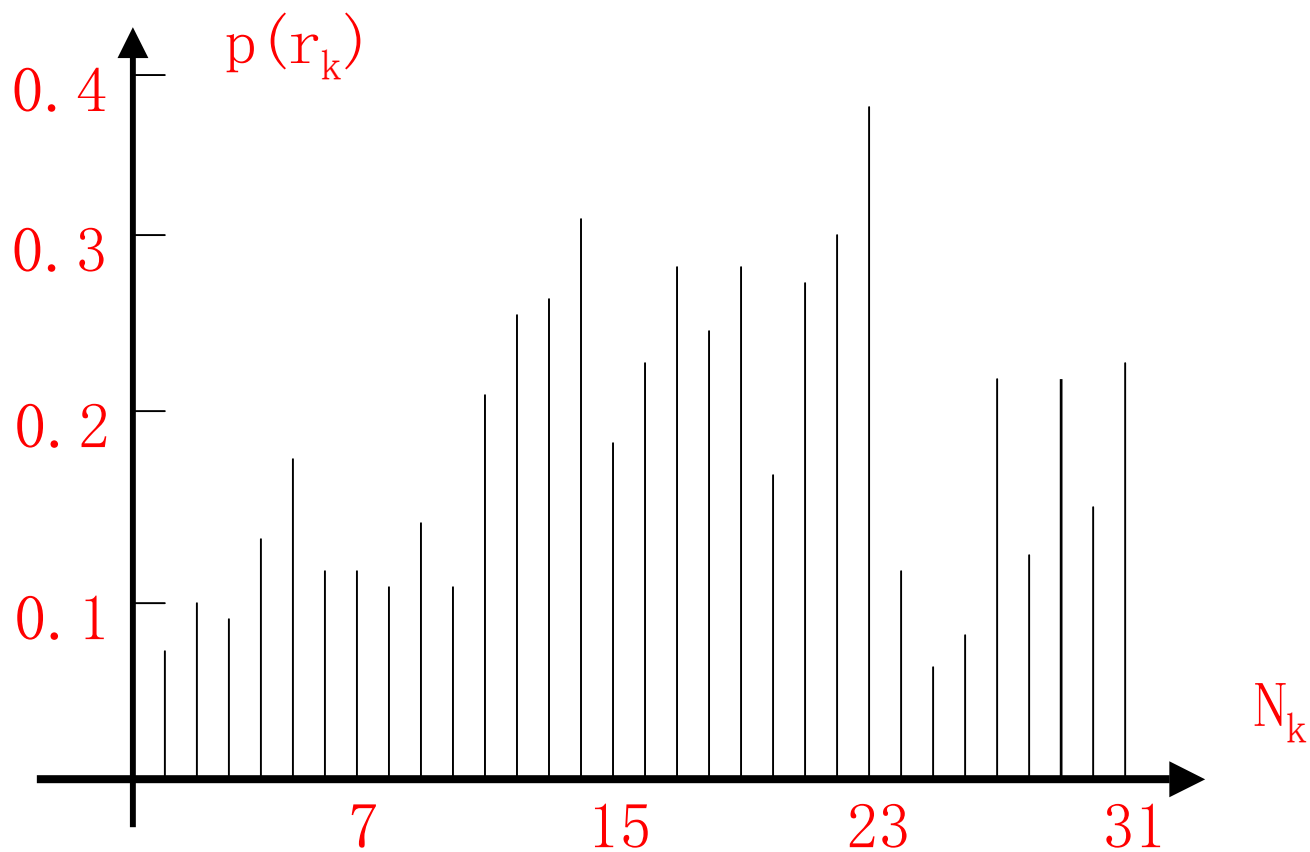
$n_k$  是图像中第 $k$ 个灰度级的像素总数

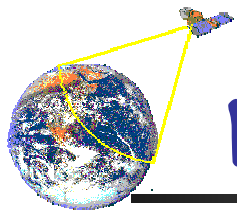
$r_k$  是第 $k$ 个灰度级，  $k = 0, 1, 2, \dots, L-1$



# 图像的空域变换—直方图变换

## ➤ 图像直方图例





## 图像的空域变换—直方图变换

### ■ 图像直方图的定义（2）

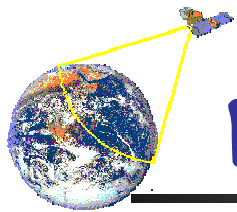
一个灰度级别在范围 $[0, L-1]$ 的数字图像的直方图是一个离散函数

$$p(r_k) = n_k \quad k = 0, 1, 2, \dots, L-1$$

由于 $r_k$ 的增量是1，直方图可表示为：

$$p(k) = n_k$$

即，图像中不同灰度级像素出现的次数



## 图像的空域变换—直方图变换

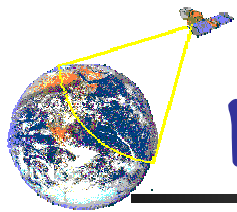
### ■ 两种图像直方图定义的比较

$$p(r_k) = n_k$$

$$p(r_k) = n_k/n$$

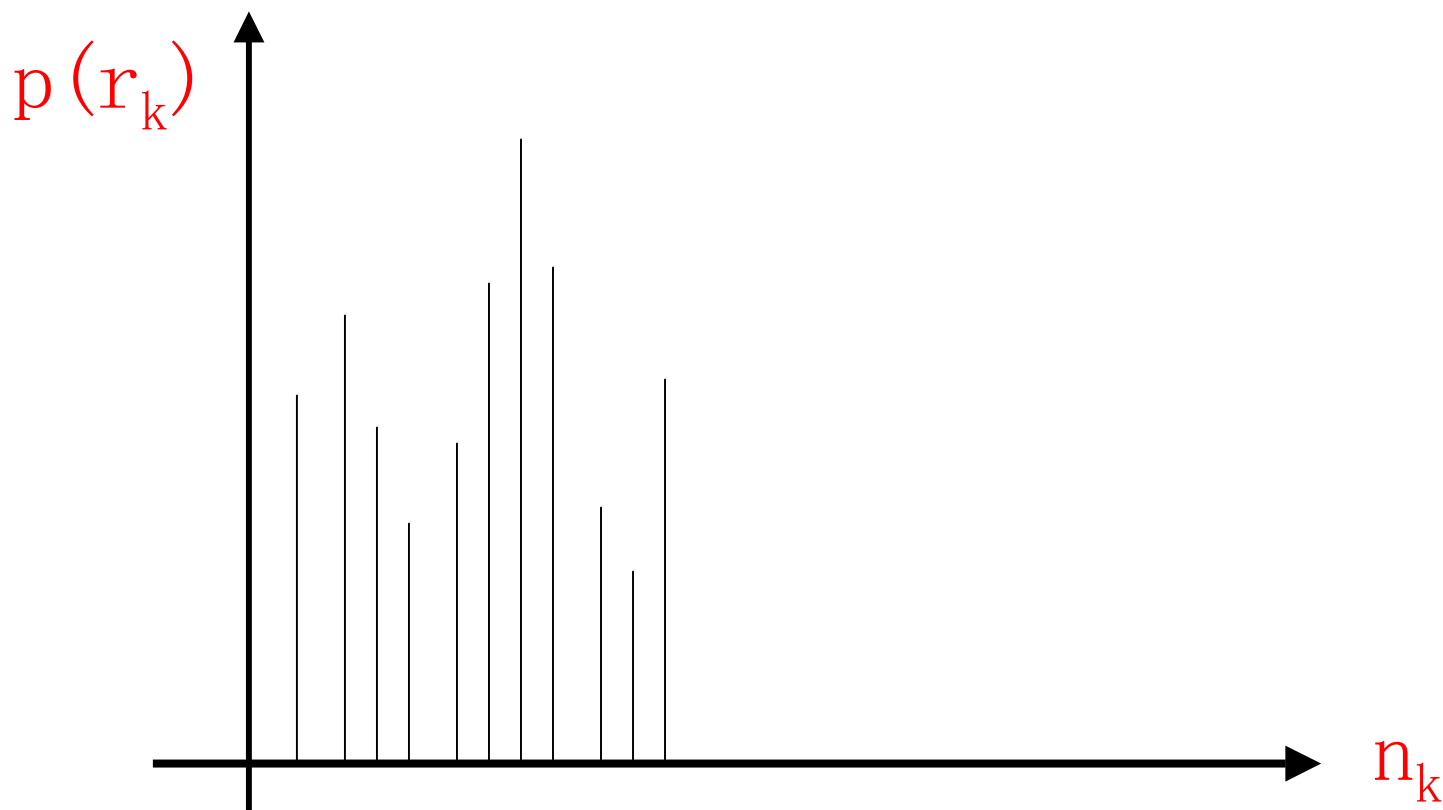
- 使函数值正则化到[0,1]区间，成为实数函数
- 函数值的范围与象素的总数无关
- 给出灰度级 $r_k$ 在图像中出现的概率密度统计

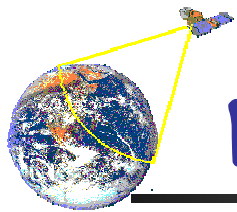




## 图像的空域变换—直方图变换

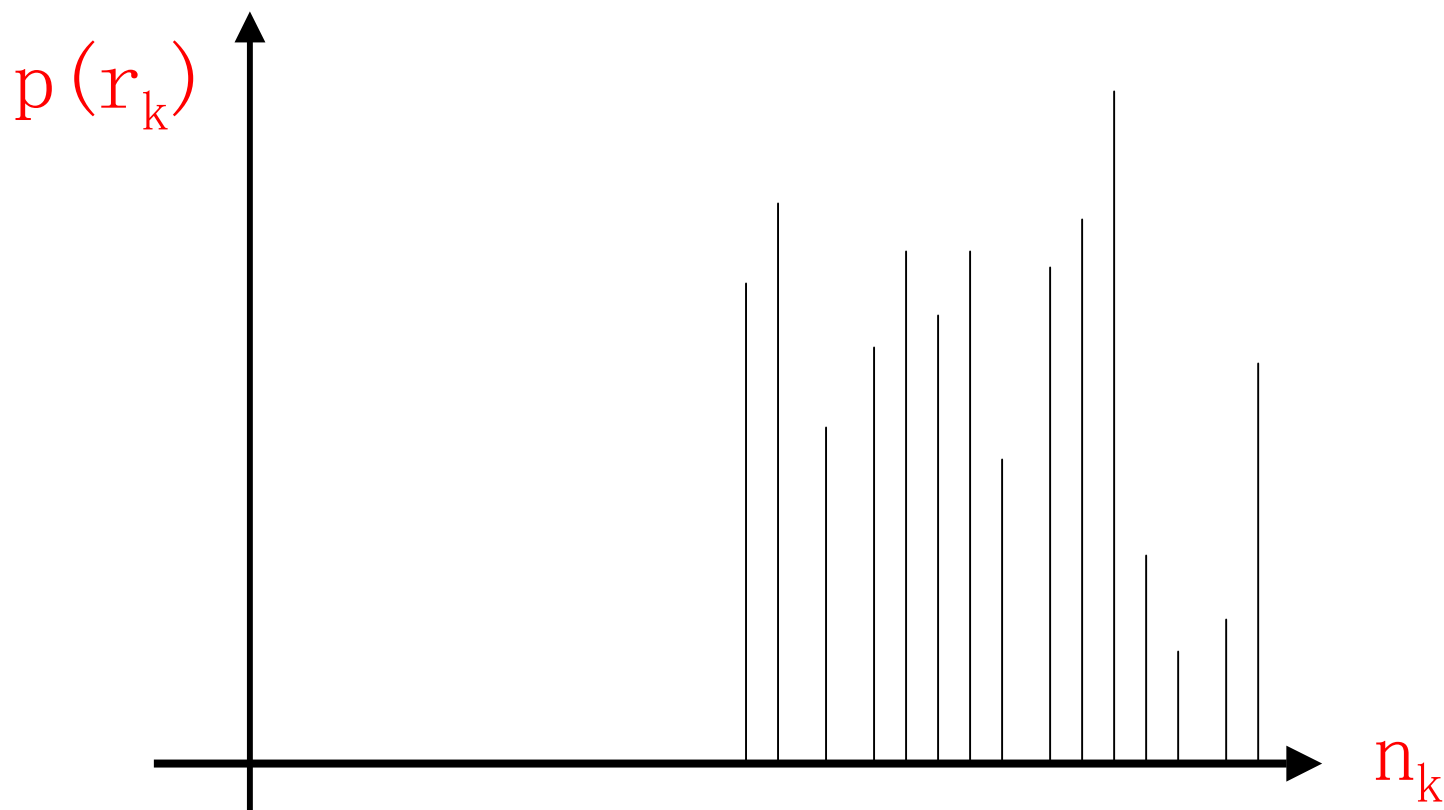
### ■ 较暗图像的直方图

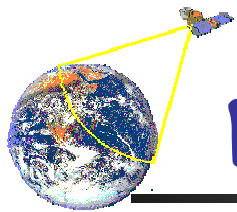




## 图像的空域变换—直方图变换

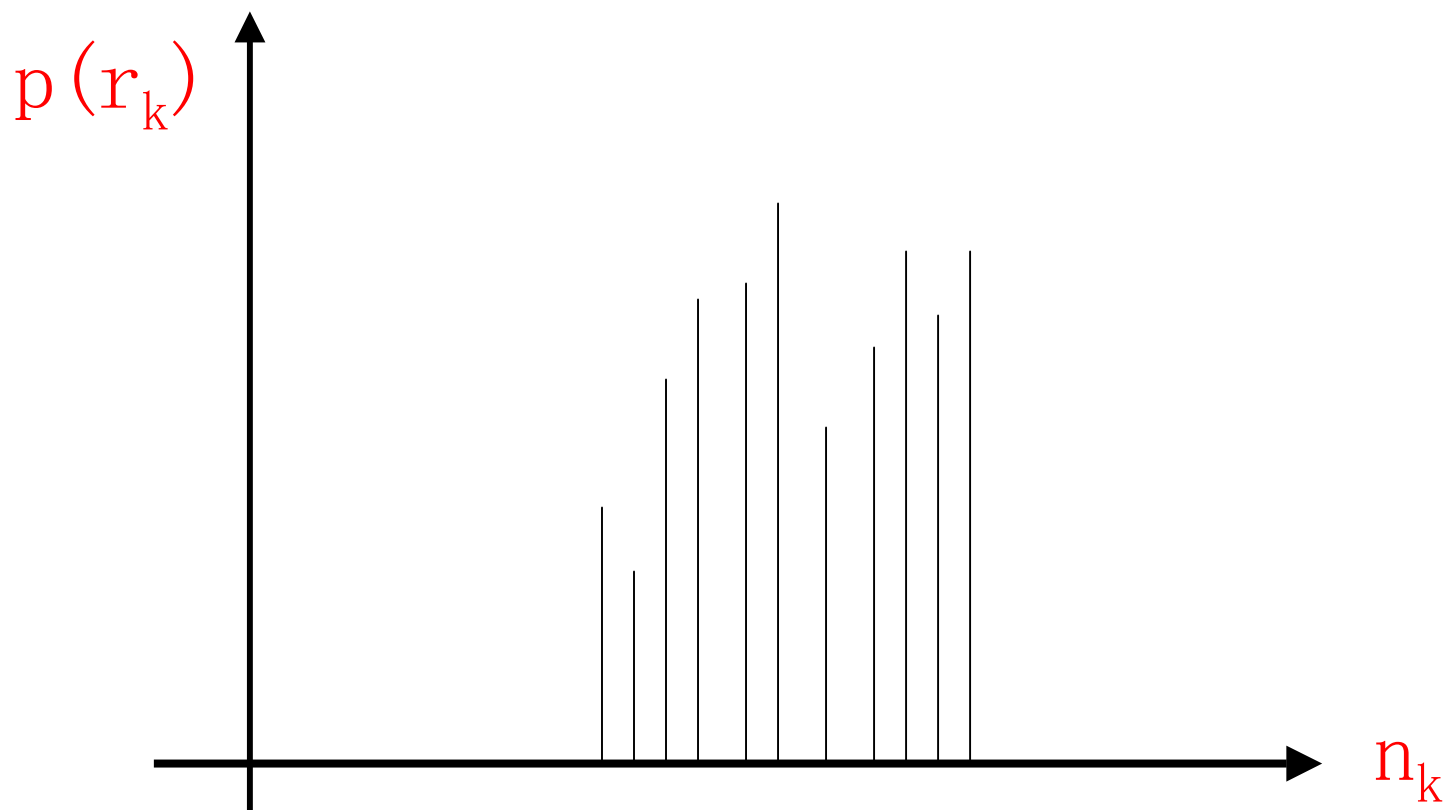
### ■ 较亮图像的直方图

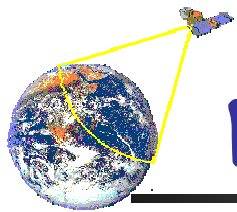




## 图像的空域变换—直方图变换

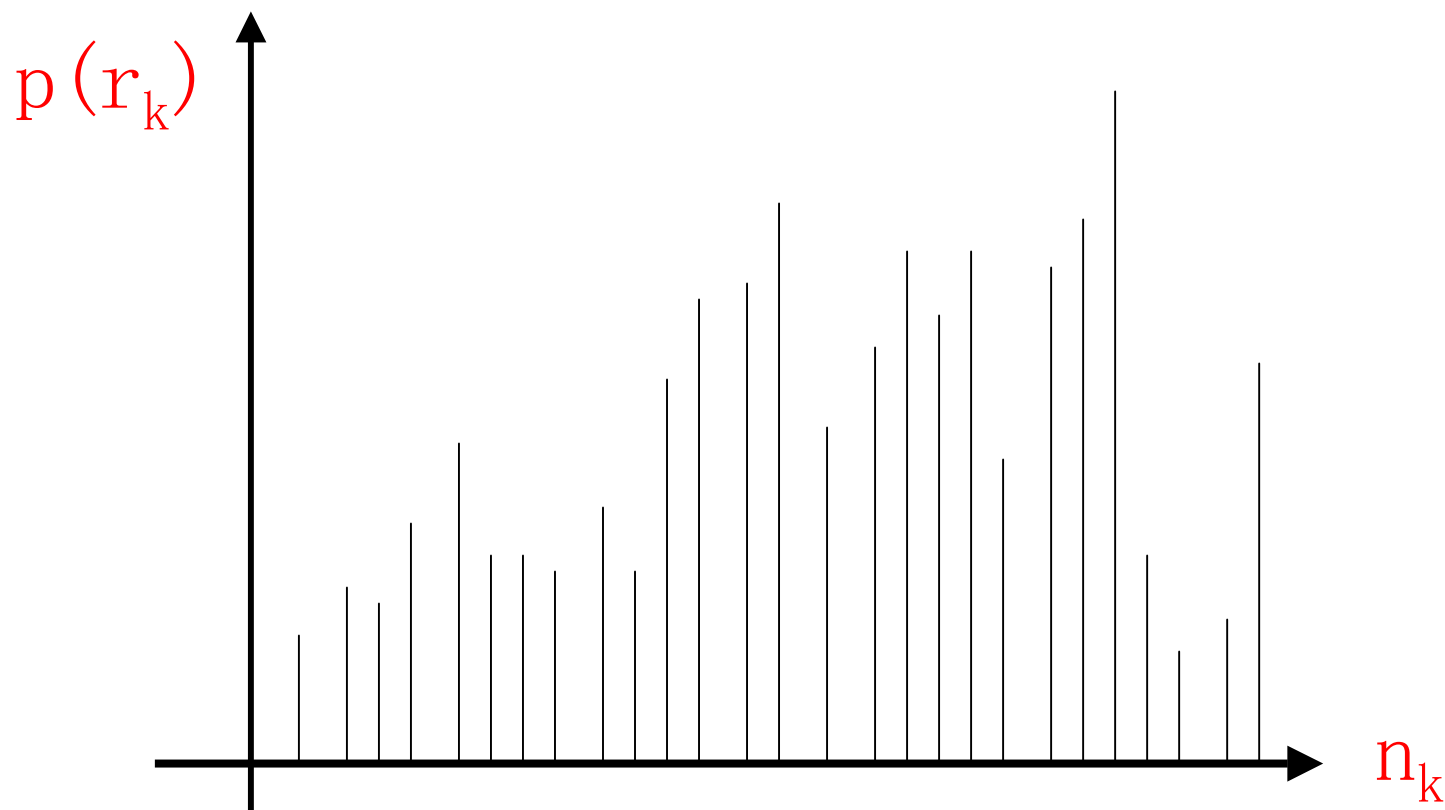
### ■ 对比度较低图像的直方图

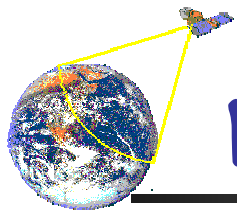




## 图像的空域变换—直方图变换

### ■ 对比度较高图像的直方图





# 图像的空域变换—直方图变换

## ■ 直方图变换的理论基础

➤ 设连续图像的概率分布为：

$$P(r) = \lim_{\Delta r \rightarrow 0} \frac{A(r + \Delta r) - A(r)}{\Delta r A} \quad \begin{array}{l} r \text{ — 灰度} \\ 0 \leq r \leq 1 \end{array}$$

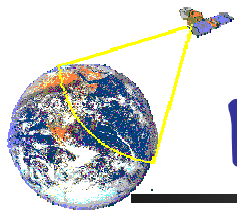
$$\int_{r_{\min}}^{r_{\max}} P(r) dr = 1$$

其中  $A$  为图像的面积。

对于离散图像

$$P(r_i) = \frac{n_i}{n}$$

$$\sum_{i=0}^{k-1} P(r_i) = 1$$



## 图像的空域变换—直方图变换

### ■ 直方图变换的理论基础

➤ 对 $[0, 1]$ 区间内任意 $r$ 值，按下式变换：

$$s = T(r)$$

➤ 上述变换式满足条件：

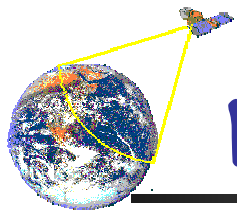
(1) 对于 $0 \leq r \leq 1$ ，有 $0 \leq s \leq 1$

(2) 在 $0 \leq r \leq 1$ 区间内， $T(r)$  为单值单调增加

➤ 从 $s$ 到 $r$ 的反变换为：

$$r = T^{-1}(s) \quad 0 \leq s \leq 1$$

➤ 反变换 $T^{-1}(s)$ 同样满足条件(1) (2)。



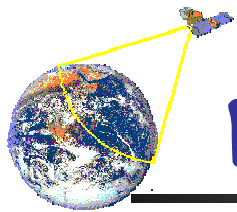
# 图像的空域变换—直方图变换

## ■ 直方图变换的理论基础

- 由概率论知，若 $P_r(r)$ 和变换函数 $s=T(r)$ 已知， $r=T^{-1}(s)$ 是单调增长函数，则变换后的图像灰度级的概率密度函数 $P_s(s)$ 如下所示：

$$P_s(s) = \left( P_r(r) \frac{dr}{ds} \right) \Big|_{r=T^{-1}(s)}$$

直方图变换技术正是通过选择变换函数 $T(r)$ ，使目标图像的直方图具有期望的形状。

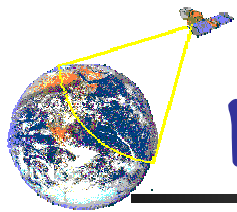


# 图像的空域变换——直方图变换

## ■ 直方图均衡

- 直方图均衡方法的基本思想是使目标图像的直方图具有平直的直方图
- 直观概念是对在图像中像素个数多的灰度级进行展宽，而对像素个数少的灰度级进行缩减。从而达到清晰图像的目的
- 基本方法是通过灰度  $r$  的概率密度函数  $p(r_k)$ ，求出灰度变换  $T(r)$ ，建立等值像素出现的次数与结果图像像素值之间的关系。
- 形成一种自动调节图像对比度质量的算法





## 图像的空域变换——直方图变换

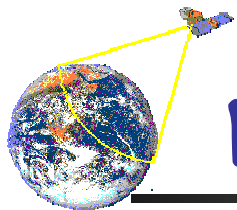
### ■ 直方图均衡

➤ 算法:

设 $f$ 、 $g$ 分别为原图像和处理后的图像。

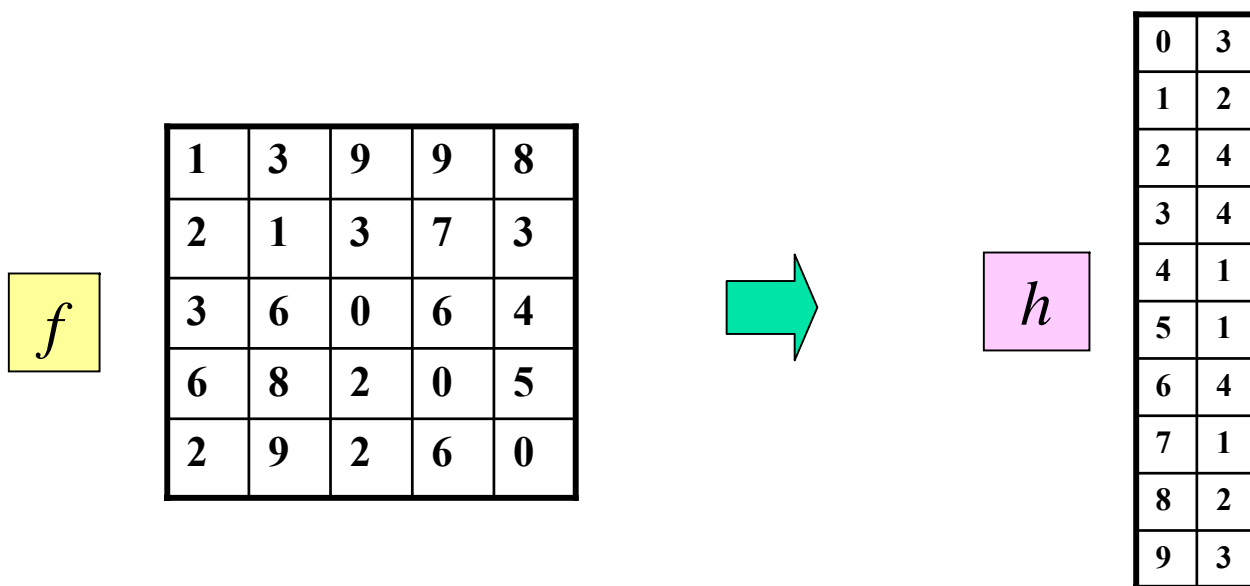
1) 求出原图 $f$ 的灰度直方图，设为 $h$ 。

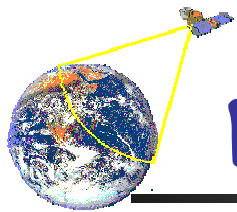
$h$ 为一个256维的向量。



# 图像的空域变换—直方图变换

## ■ 直方图均衡





## 图像的空域变换—直方图变换

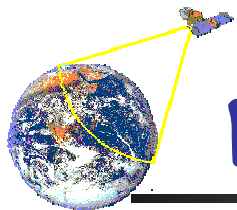
### ■ 直方图均衡

2) 求出图像  $f$  的总体像素个数

$N_f = m \times n$  ( $m, n$  分别为图像的长和宽)

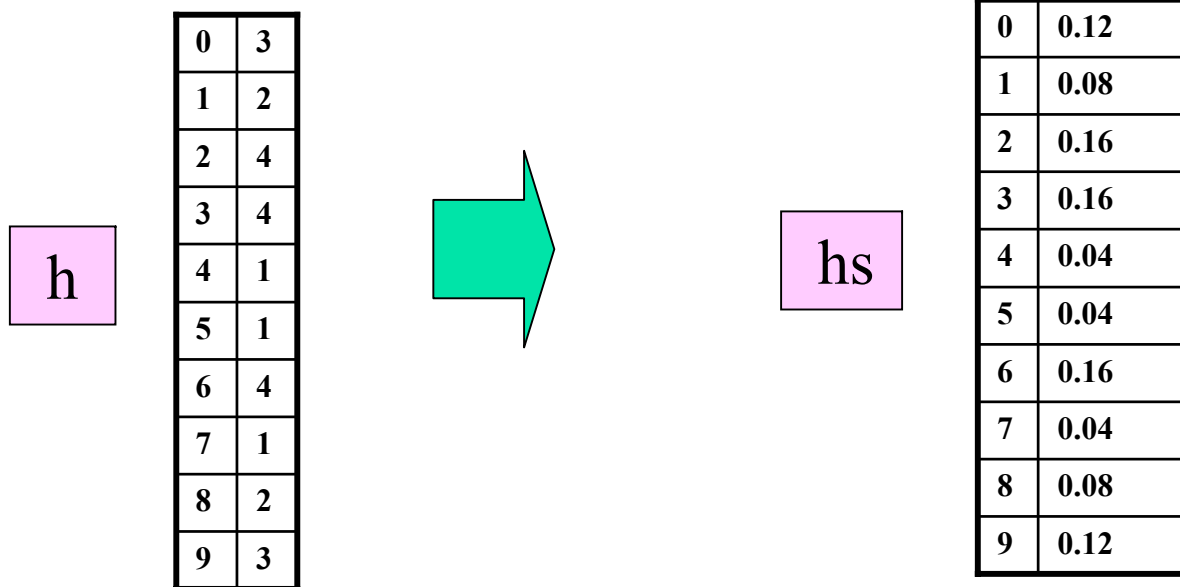
计算每个灰度级的像素个数在整个图像中所占的百分比。

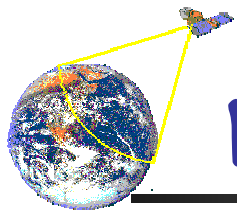
$$hs(i) = h(i) / N_f \quad (i=0,1,\dots,255)$$



# 图像的空域变换—直方图变换

## ■ 直方图均衡





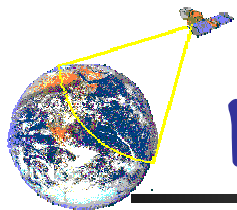
## 图像的空域变换—直方图变换

### ■ 直方图均衡

3) 计算图像各灰度级的累计分布**hp**。

$$hp(i) = \sum_{k=0}^i h(k)$$

$$i = 1, 2, \dots, 255$$

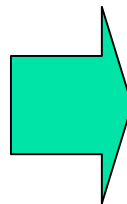


# 图像的空域变换—直方图变换

## ■ 直方图均衡

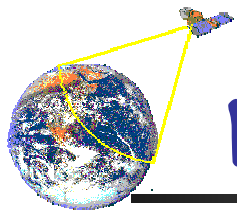
hs

0	0.12
1	0.08
2	0.16
3	0.16
4	0.04
5	0.04
6	0.16
7	0.04
8	0.08
9	0.12



hp

0	0.12
1	0.20
2	0.36
3	0.52
4	0.56
5	0.60
6	0.76
7	0.80
8	0.88
9	1.00



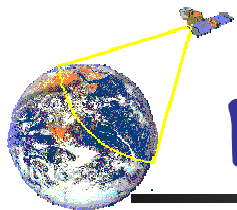
## 图像的空域变换—直方图变换

### ■ 直方图均衡

求出新图像 $g$ 的灰度值。

$$g = 255 \cdot hp(i) \quad i = 1, 2, \dots, 255$$

$$g = 0 \quad i = 0$$

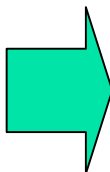


# 图像的空域变换—直方图变换

## ■ 直方图均衡

hp

0	0.12
1	0.20
2	0.36
3	0.52
4	0.56
5	0.60
6	0.76
7	0.80
8	0.88
9	1.00



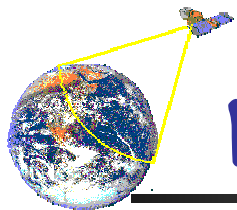
g

51	133	255	255	224
92	51	133	204	133
133	194	0	194	143
194	224	92	0	153
92	255	92	194	0

f

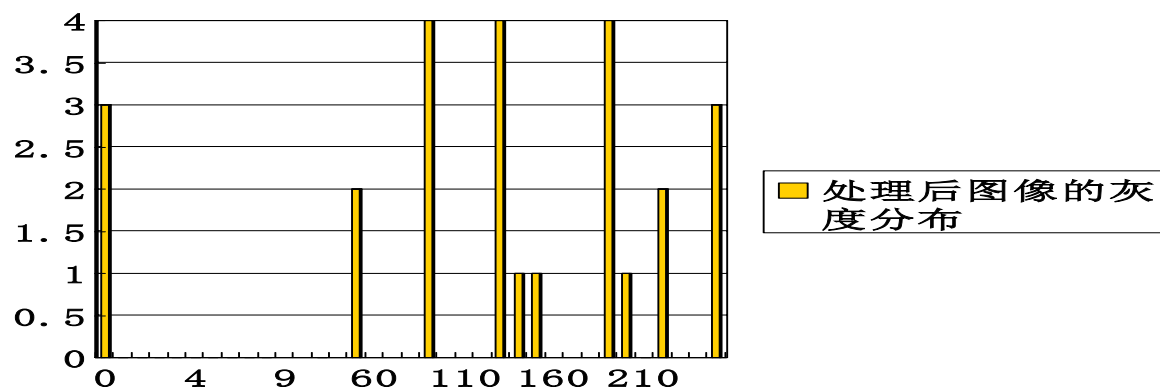
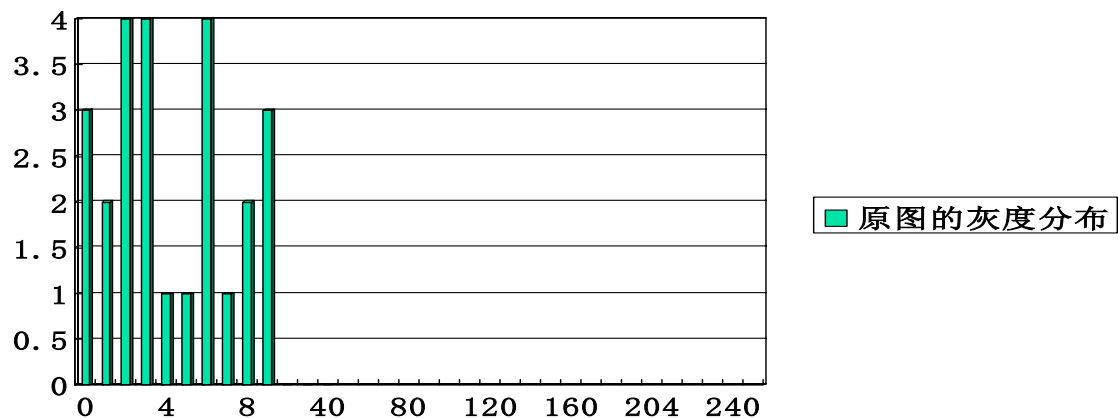
1	3	9	9	8
2	1	3	7	3
3	6	0	6	4
6	8	2	0	5
2	9	2	6	0

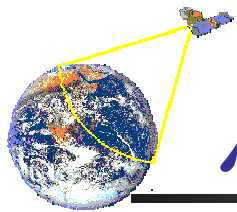




# 图像的空域变换—直方图变换

## ■ 直方图均衡





# 总结

## ■ 图像的像素级运算

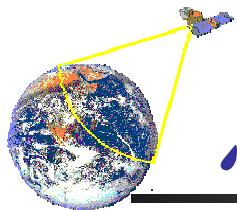
- 点运算—线性点运算、非线性点运算
- 代数运算—加法、减法、乘法、除法
- 逻辑运算—求反、异或、或、与

## ■ 图像的空域变换

- 几何变换，非几何变换

## ■ 图像的空域运算

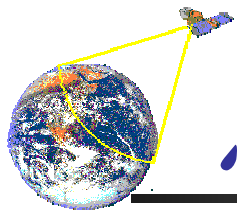
- 仅灰度值变化，空间关系不变
- 空间关系改变，灰度值不变
- 空间关系改变，灰度值改变
- 输入图像中各像素独立运算
- 输入图像中一定区域的像素参与运算—领域运算



## 思考题与习题

### ■ 思考题

- 假设你有两张同一建筑物的照片，这两张照片是不同的人在同一天站在同一地点上拍下的，时间相差4个小时，在这4个小时中，有人从该建筑物的某个窗口射击了三枪。负责调查的侦探不知道哪间办公室是事发房间，而对胶片的肉眼视觉检查也无法表明在这段时间内有哪扇窗户被打开或关上，你能帮助他吗？



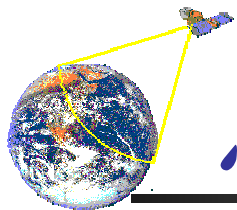
## 思考题与习题

### ■ 习题-1

设图像为：

$$F = \begin{bmatrix} 59 & 60 & 58 & 57 \\ 61 & 59 & 59 & 57 \\ 62 & 59 & 60 & 58 \\ 59 & 61 & 60 & 56 \end{bmatrix}$$

- 1) 请将它旋转45度;
- 2) 请将它x方向进行45度错切。



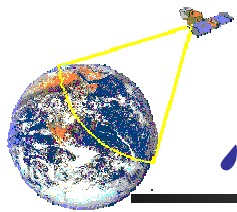
## 思考题与习题

### ■ 习题-2

已知一幅图像为：

$$f = \begin{bmatrix} 1 & 5 & 255 & 225 & 100 & 200 & 255 & 200 \\ 1 & 7 & 254 & 255 & 100 & 10 & 10 & 9 \\ 3 & 7 & 10 & 100 & 100 & 2 & 9 & 6 \\ 3 & 6 & 10 & 10 & 9 & 2 & 8 & 2 \\ 2 & 1 & 8 & 8 & 9 & 3 & 4 & 2 \\ 1 & 0 & 7 & 8 & 8 & 3 & 2 & 1 \\ 1 & 1 & 8 & 8 & 7 & 2 & 2 & 1 \\ 2 & 3 & 9 & 8 & 7 & 2 & 2 & 0 \end{bmatrix}$$

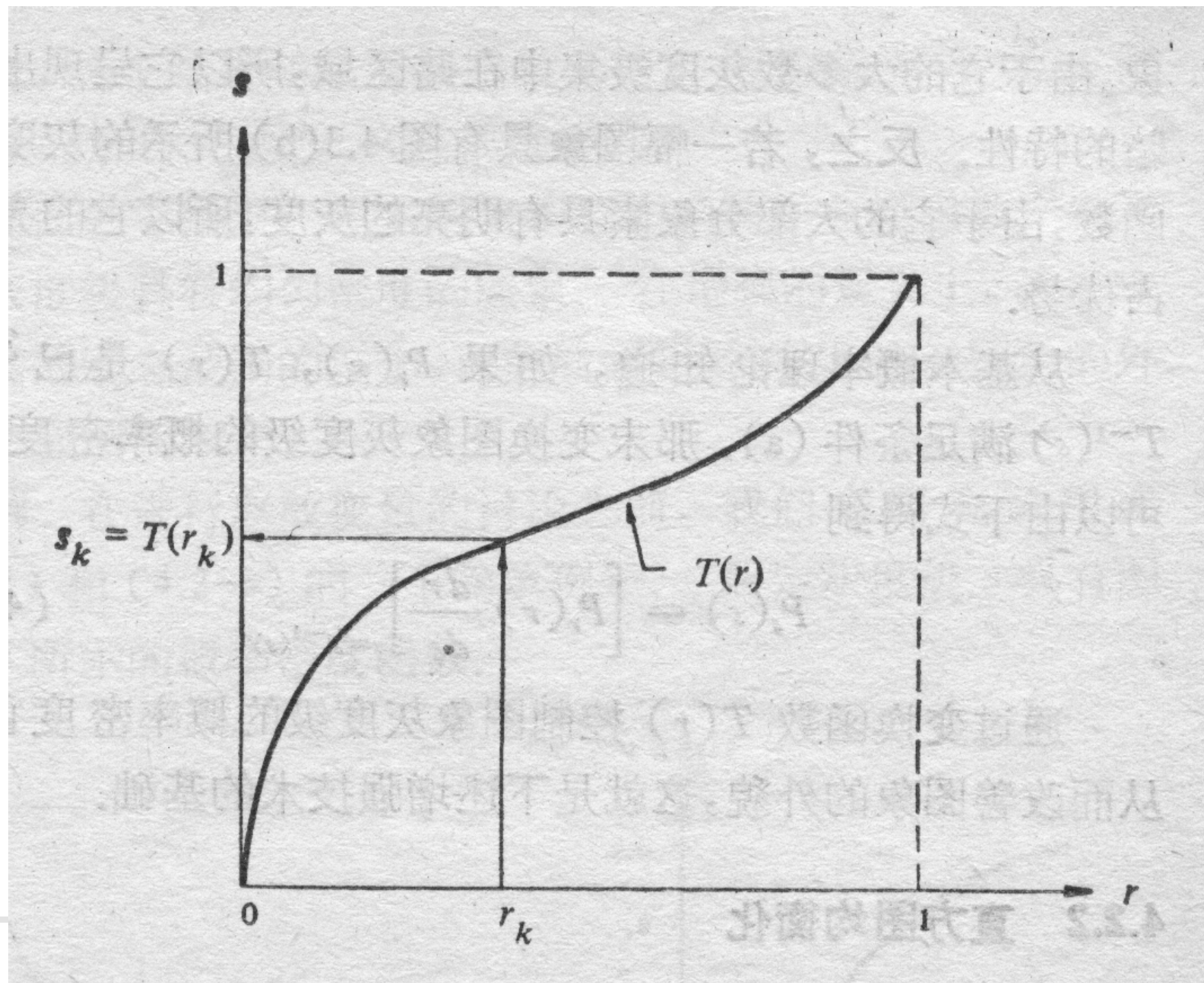
请对其进行灰度直方图的均衡化处理。

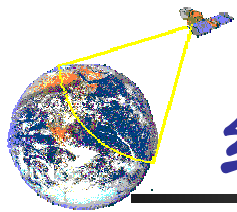


## 思考题与习题

### ■ 上机实验

- 熟悉实验软件、Photoshop中的各种灰度变换操作及观察效果，通过实验巩固本节课所学内容
- 对所附扫描图像，通过灰度变换方法使其清晰可用
- 编程试验
  - ✓ 编制一个程序，对图像进行灰度变换，以增强其对比度
  - ✓ 编制一个程序对图像进行旋转，要求适应任意角度





结束

# 第三章

# 结束