



# 数字图象处理

---

中国科学技术大学  
电子工程与信息科学系

主讲教师：李厚强 (lihq@ustc.edu.cn)  
周文罡 (zhwg@ustc.edu.cn)

助教：冯 浩 ([fh1995@mail.ustc.edu.cn](mailto:fh1995@mail.ustc.edu.cn))

课程主页：<https://ustc-dip.github.io/>



# 第1章 绪论

---

- 1.1 什么是数字图像处理
- 1.2 数字图像处理的起源
- 1.3 数字图像处理应用领域
- 1.4 数字图像处理的基本步骤
- 1.5 图像处理系统的组成
- 1.6 数字图像处理应用举例





# 第1章 绪论

---

- 1.1 **什么是数字图像处理**
- 1.2 数字图像处理的起源
- 1.3 数字图像处理应用领域
- 1.4 数字图像处理的基本步骤
- 1.5 图像处理系统的组成
- 1.6 数字图像处理应用举例



# 1.1 什么是数字图像处理

---

## ◆ 图像

用各种观测系统以不同形式和手段观测客观世界而获得的，可以直接或间接作用于人眼并进而产生视知觉的实体。

## ◆ 图像和信息

人类从外界（客观世界）获得的信息约有75%来自视觉系统。



# 1.1 什么是数字图像处理

## ◆ 图像表示

2-D数组  $f(x, y)$

$x, y$ : 2-D空间 $XY$ 中坐标点的位置

$f$ : 代表图像在 $(x, y)$ 的性质 $F$ 的数值

$f, x, y$  的值可以是任意实数

性质 $F$ : 可对应不同物理量

灰度图象里用灰度表示



# 1.1 什么是数字图像处理

## ◆ 图像单元

一幅图像是许多图像单元的集合体

2-D图象： 像素（picture element）常用pixel表示

3-D图象： 体素（volume element）常用voxel表示

$$f(x, y) \rightarrow f(x, y, z), f(x, y, t)$$

# 1.1 什么是数字图像处理

## ◆ 图像显示示例



(a)



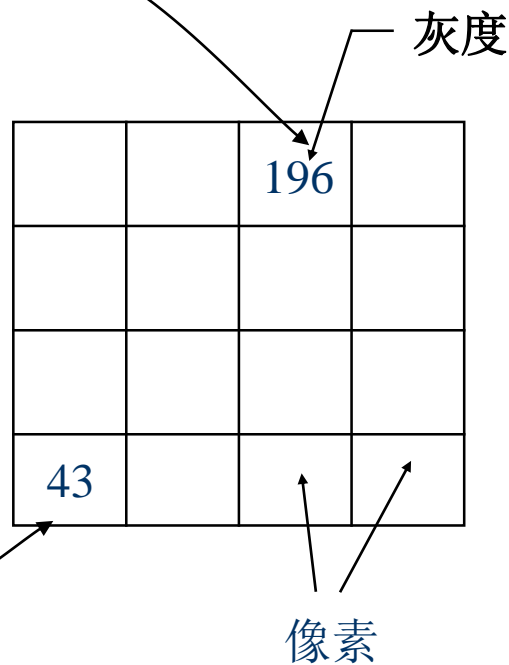
(b)

# 1.1 什么是数字图像处理

## ◆ 数字图像表示



(a) 物理图像



(b) 数字图像



# 1.1 什么是数字图像处理

## ◆ 数字图像处理的定义

从图像处理到计算机视觉是一个连续的统一体

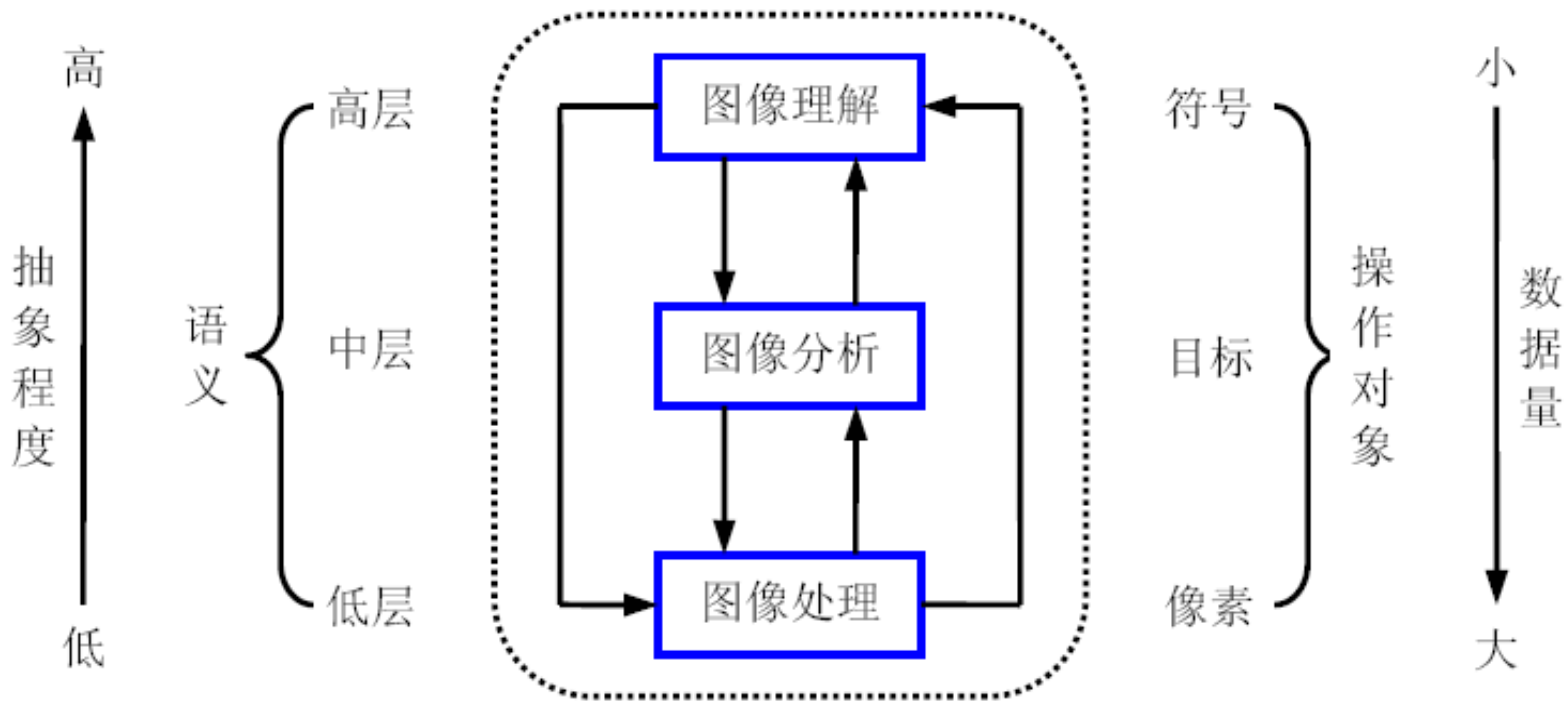
图像处理（图像  图像）

图像分析（图像  数据）

图像理解（图像  解释）

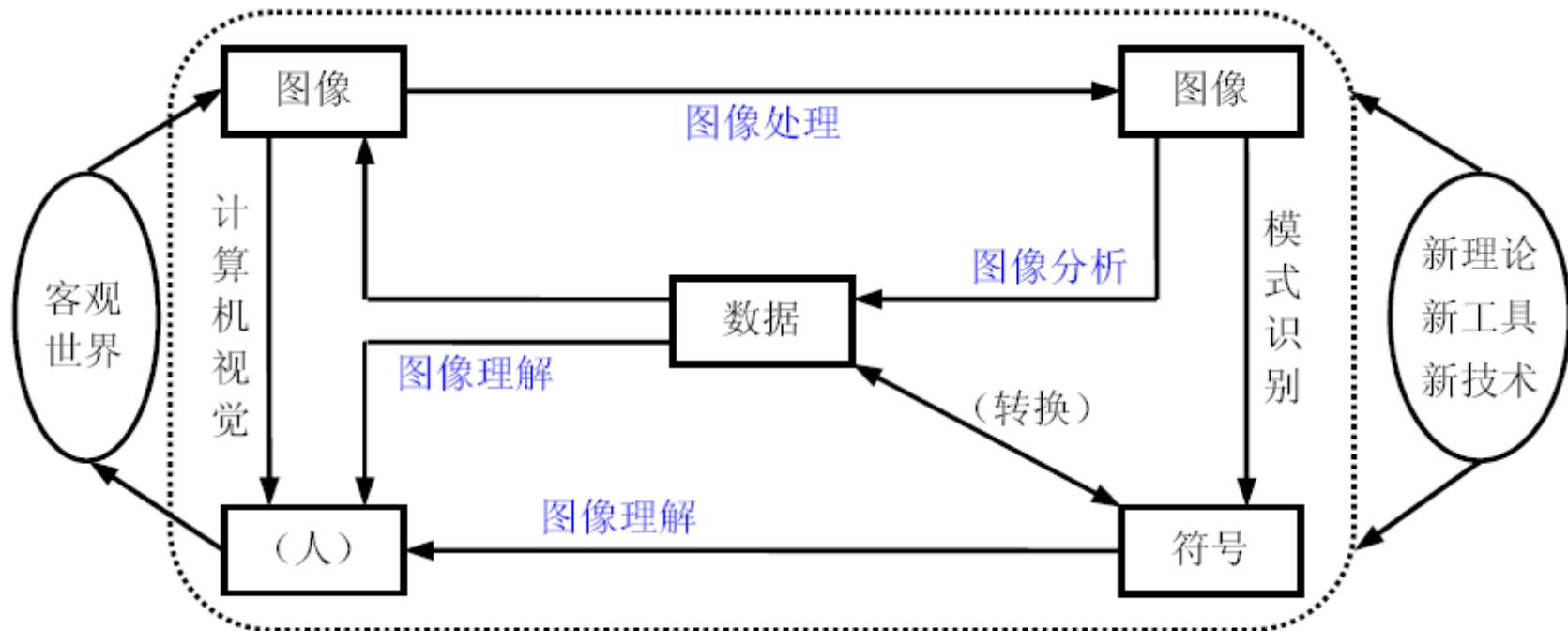
我们将**数字图像处理**界定为其**输入**和**输出**都是图像的处理

# 1.1 什么是数字图像处理





# 1.1 什么是数字图像处理





# 第1章 绪论

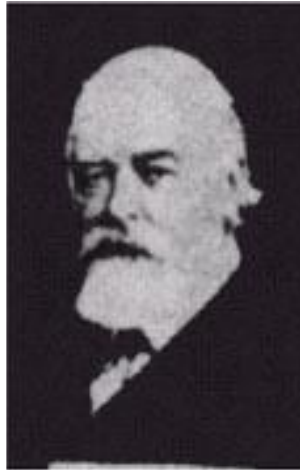
---

- 1.1 什么是数字图像处理
- 1.2 **数字图像处理的起源**
- 1.3 数字图像处理应用领域
- 1.4 数字图像处理的基本步骤
- 1.5 图像处理系统的组成
- 1.6 数字图像处理应用举例

# 1.2 数字图像处理的起源

## 巴特兰 (Bartlane) 电缆图片传输系统

**1921年** 电报打印机采用特殊字符在编码纸带打印。输出设备从通用到专用



**1922年** 两次穿越大西洋，穿孔纸得到图像检测误差。图像通信系统  
信源编码和信道编码

**1929年** 从伦敦到纽约从早期5个灰度到15个通过电缆传输



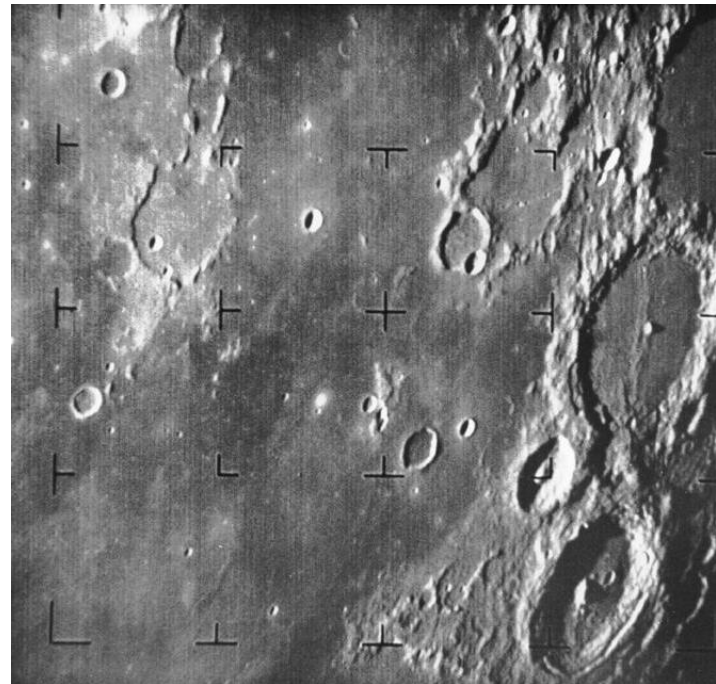


# 1.2 数字图像处理的起源

第一台功能强大到足以执行有意义的图像处理任务的大型计算机，出现于20世纪60年代初

处理卫星图像：校正航天器上电视摄像机中各种类型的图像畸变

“徘徊者7号”撞击月球表面前拍摄的第一张月球图像





# 1.2 数字图像处理的起源

- 20世纪60年代末和70年代，数字图像处理技术开始用于医学成像、地球资源遥感监测和天文学领域。
- 1895年伦琴发现X射线，获1901年诺贝尔物理学奖。
- 1975年Godfrey N. Hounsfield和Allan M. Cormack发明了计算机断层技术（CT），获1979年诺贝尔医学奖。
- 从20世纪60年代至今，数字图像处理技术除了用于上述领域，在地理学、考古学、生物学、国防和工业领域中也有广泛的应用。我们将在下一节中说明某些这样的应用领域



# 第1章 绪论

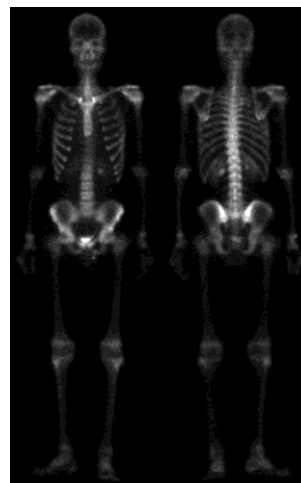
---

- 1.1 什么是数字图像处理
- 1.2 数字图像处理的起源
- 1.3 **数字图像处理应用领域**
- 1.4 数字图像处理的基本步骤
- 1.5 图像处理系统的组成
- 1.6 数字图像处理应用举例

# 1.3 数字图像处理应用领域

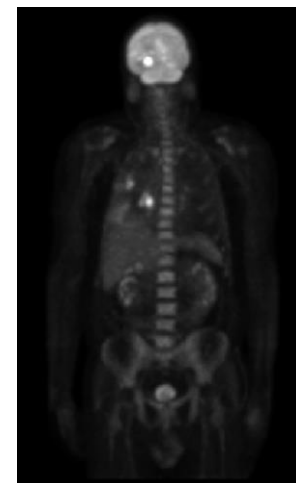
## ◆ 伽马射线成像

(a) 骨骼扫描图像



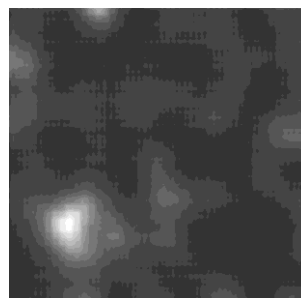
(a)

(b) PET图像



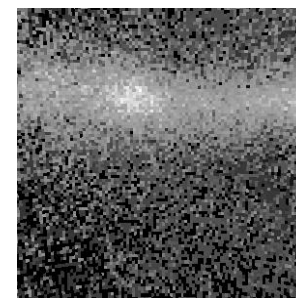
(b)

(c) 天鹅星座环图像



(c)

(d) 来自反应堆真空管的伽马辐射（亮点）



(d)



# 1.3 数字图像处理应用领域

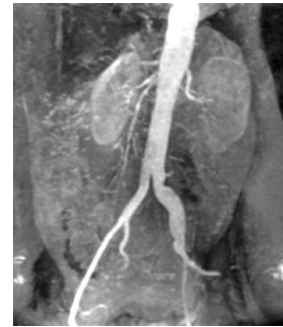
## ◆ X射线成像

- (a) 胸部X射线图像
- (b) 主动脉造影图像
- (c) 头部CT图像
- (d) 电路板图像
- (e) 天鹅星座环图像

**X射线是最早用于成像的电磁辐射源之一**



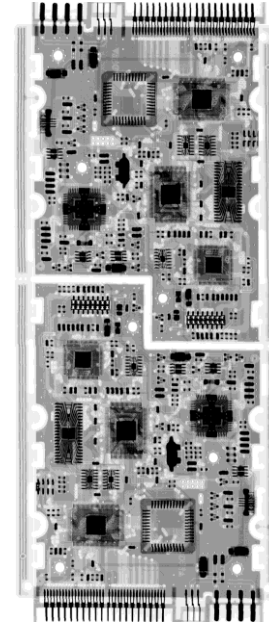
(a)



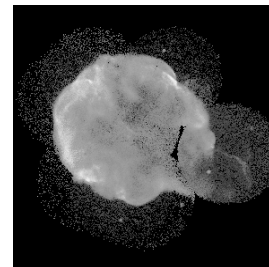
(b)



(c)



(d)

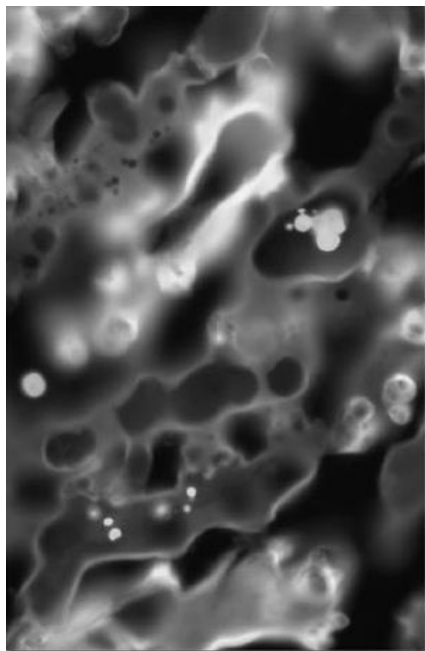


(e)

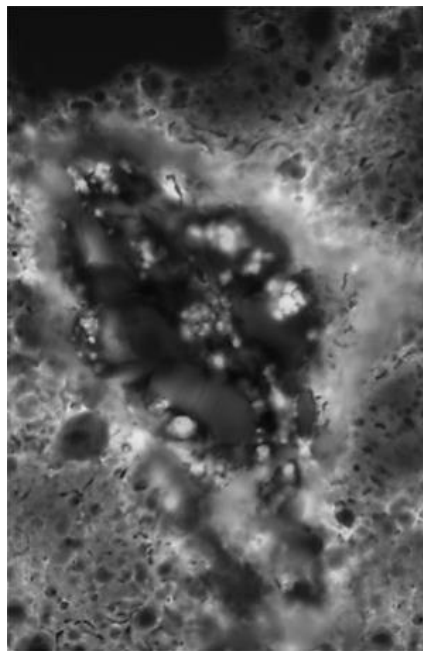


# 1.3 数字图像处理应用领域

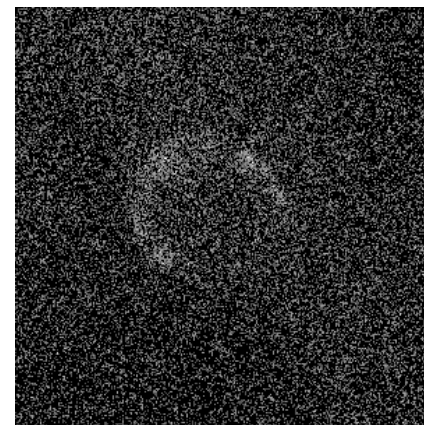
## ◆ 紫外波段成像



(a)



(b)

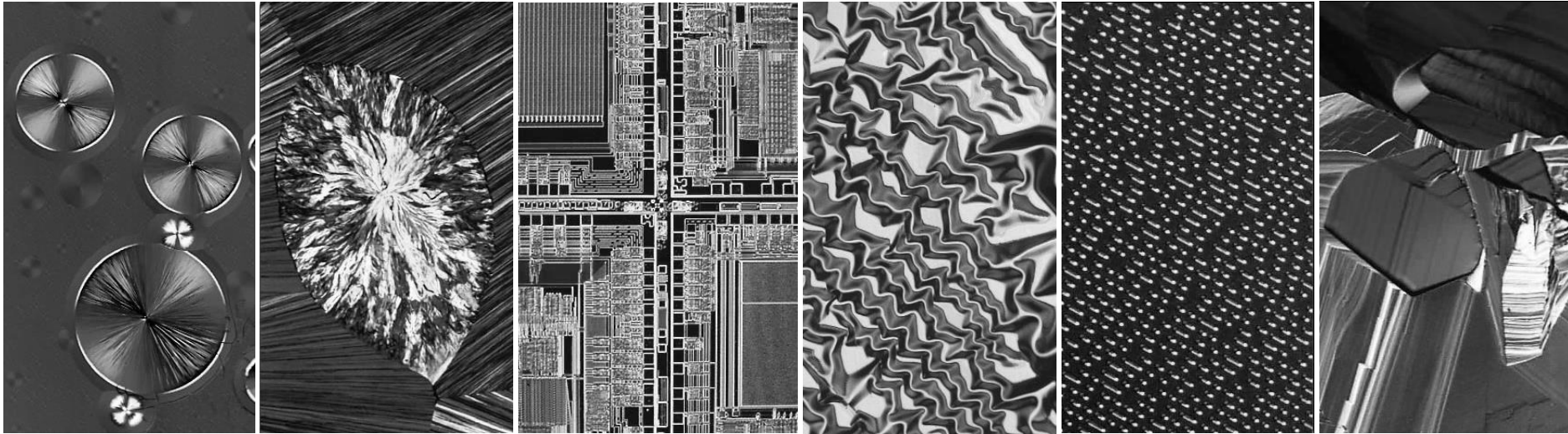


(c)

(a) 普通玉米图像； (b) 患黑穗病的玉米图像； (c) 天鹅星座环图像

# 1.3 数字图像处理应用领域

## ◆ 可见光与红外波段成像-光显微镜图像



(a)

(b)

(c)

(d)

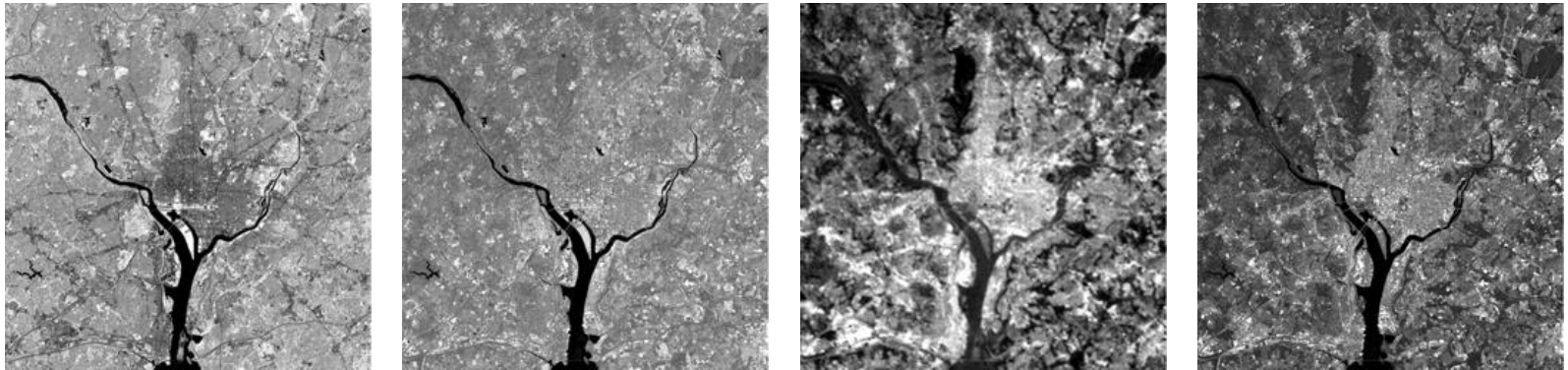
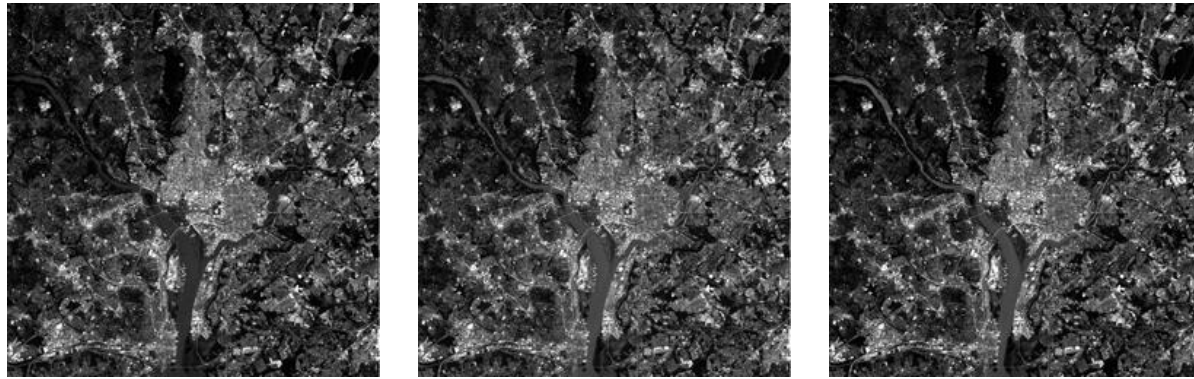
(e)

(f)

(a) 放大250倍的紫衫酚（抗癌剂）；(b) 放大40倍的胆固醇；(c) 已放大60倍微处理器；(d) 已放大600倍的镍氧化物胶片；(e) 已放大1750倍的音频CD表面；(f) 已放大450倍的有机超导体

# 1.3 数字图像处理应用领域

## ◆ 可见光与红外波段成像-多光谱遥感图像分析

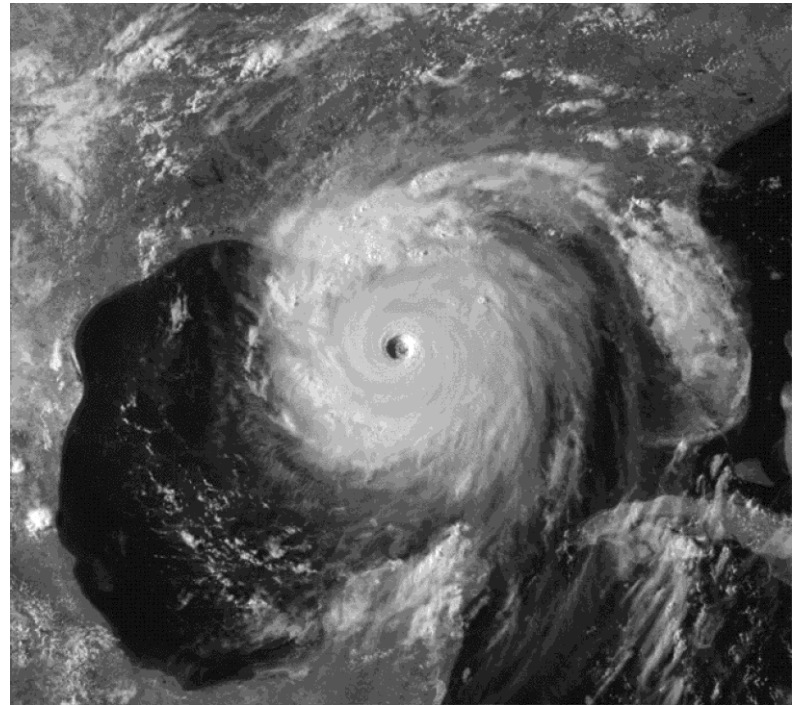




# 1.3 数字图像处理应用领域

## ◆ 可见光与红外波段成像-天气观测与预报

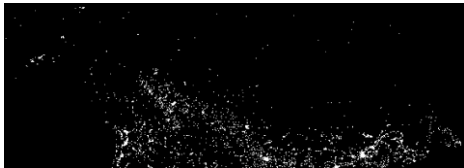
(a) 2005年8月29日拍摄的“卡特琳娜”飓风的卫星图像



(a)

# 1.3 数字图像处理应用领域

## ◆ 可见光与红外波段成像-红外成像系统

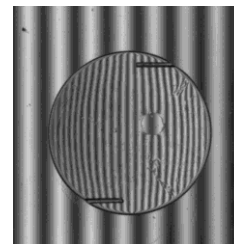
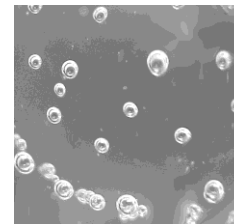
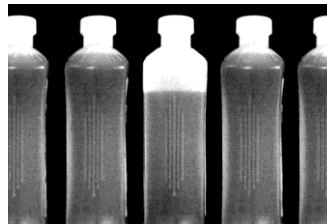
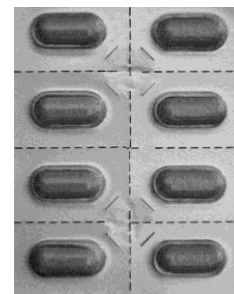
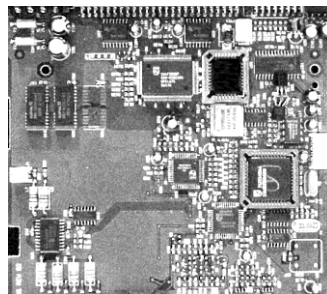


左：美洲红外卫星图像  
右：实际地形参考图像

# 1.3 数字图像处理应用领域

## ◆ 可见光与红外波段成像-工业检测

可见光谱中一个主要成像领域是生产产品的自动视觉检测



# 1.3 数字图像处理应用领域

## ◆ 可见光与红外波段成像-其他例子

(a)



(b)



(a) 拇指指纹图像

(b) 纸币图像

(c) ~ (d) 车牌图像识别



(c)

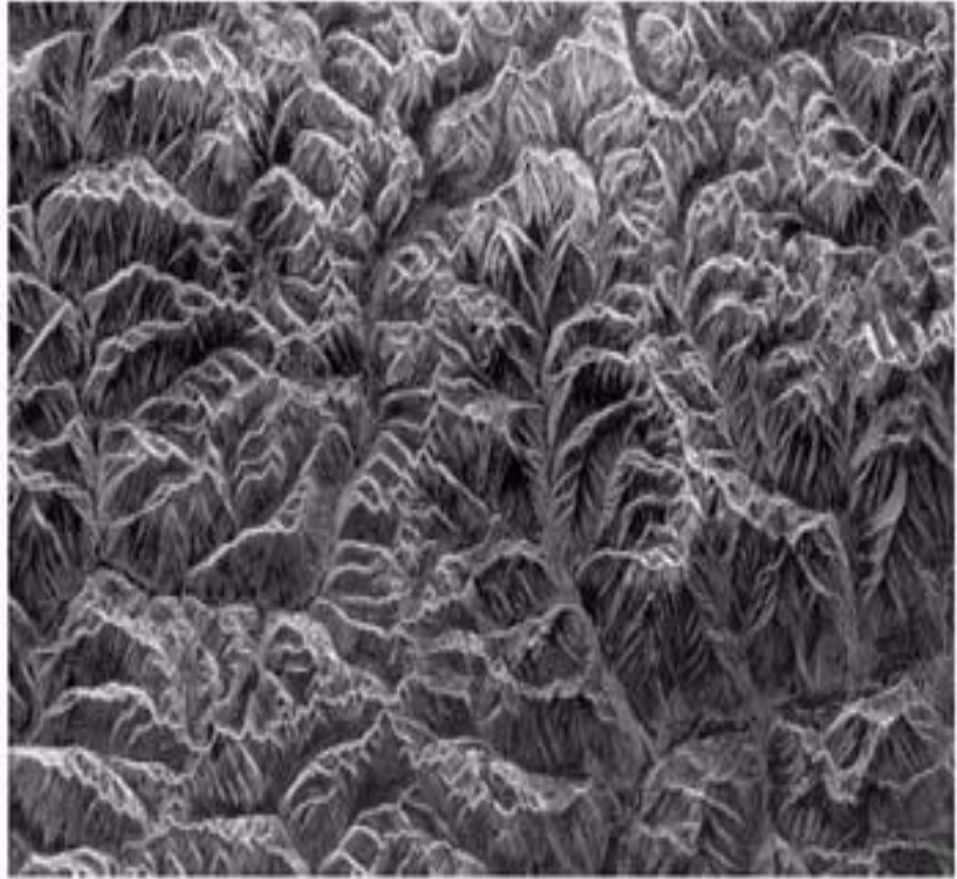


(d)

# 1.3 数字图像处理应用领域

## ◆ 微波波段成像

航天器拍摄的  
西藏东南山区  
雷达图像





# 1.3 数字图像处理应用领域

## ◆ 无线电波段成像



(a)

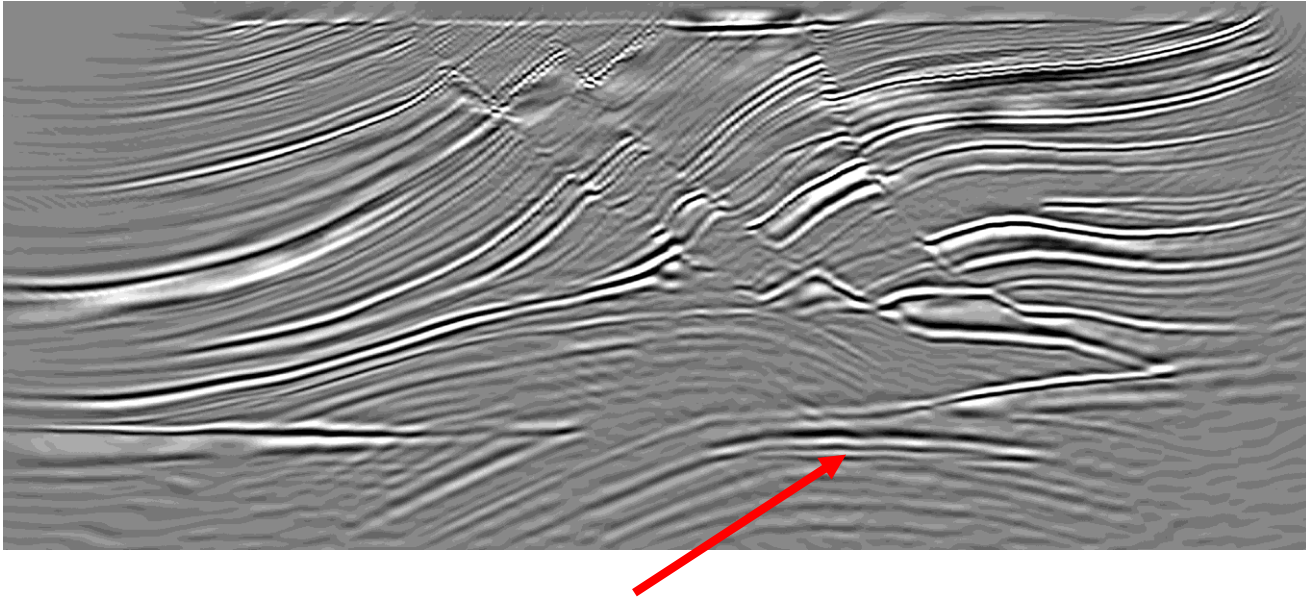


(b)

人的磁共振 (MRI) 图像：(a) 膝盖图像；(b) 脊椎图像

# 1.3 数字图像处理应用领域

## ◆ 使用其他成像方式的例子-低端声波成像



地震模型的剖面图像。箭头指向碳氢化合物（油或气）的油气阱

# 1.3 数字图像处理应用领域

## ◆ 使用其他成像方式的例子-超声成像

(a) 胎儿图像



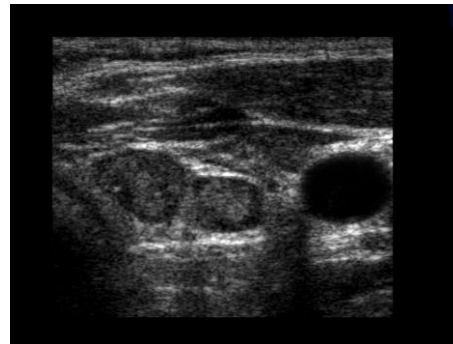
(a)

(b) 胎儿的另一幅图像



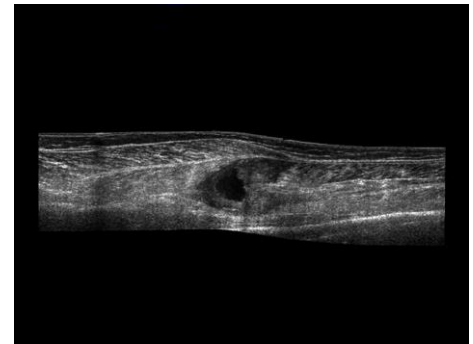
(b)

(c) 甲状腺图像



(c)

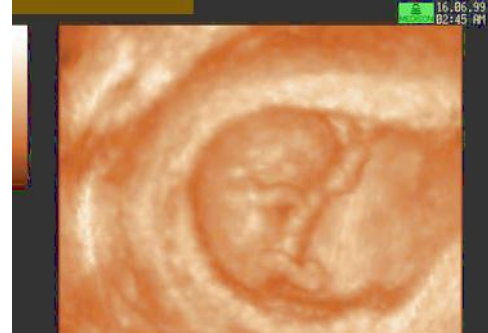
(d) 有损伤的肌肉层图像



(d)

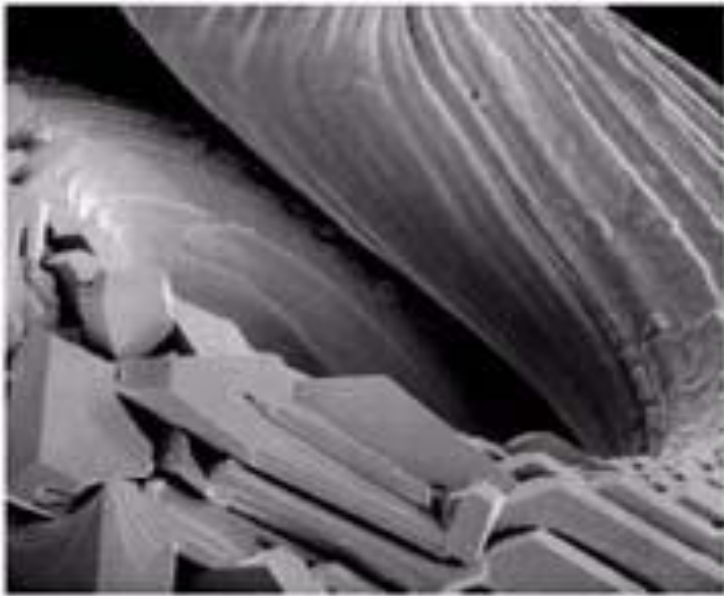
# 1.3 数字图像处理应用领域

## ◆ 使用其他成像方式的例子-超声成像

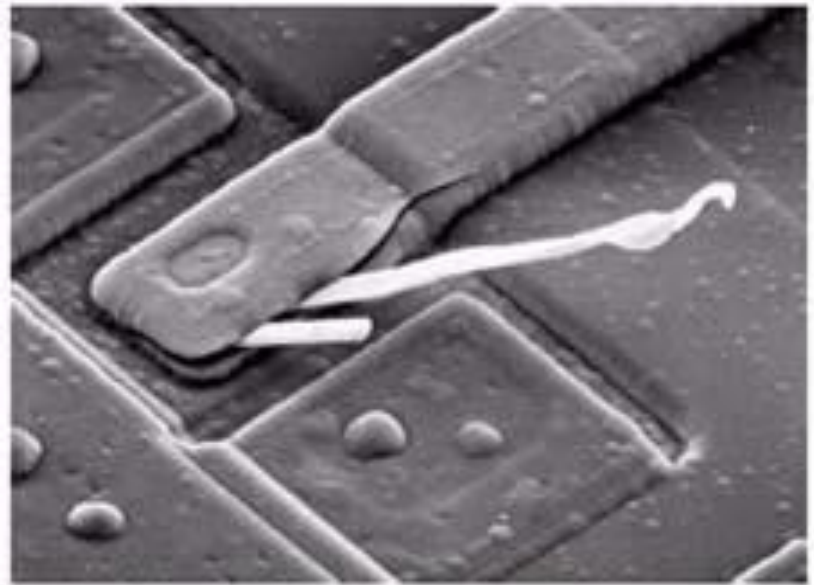


# 1.3 数字图像处理应用领域

## ◆ 使用其他成像方式的例子-电子显微镜成像



(a) 过热损坏的钨丝 (250倍)

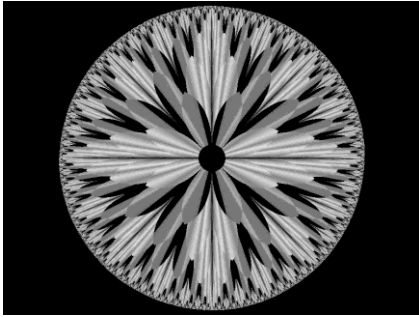


(b) 损坏的IC电路 (2500倍)

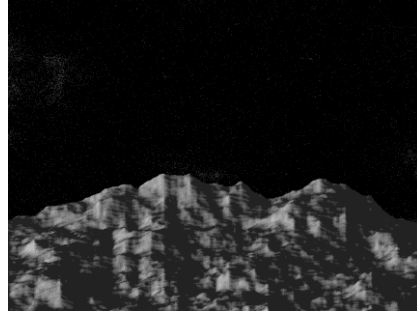
扫描式电子显微镜图像 (SEM)

# 1.3 数字图像处理应用领域

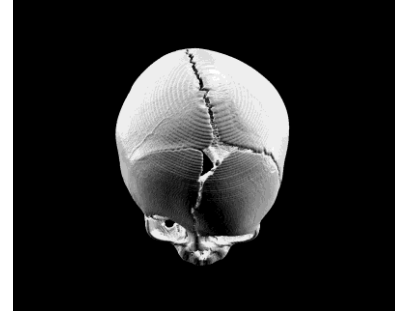
## ◆ 计算机生成的图像



(a)



(b)



(c)



(d)

(a) ~ (b) 分形图像；(c) ~ (d) 所示物体的三维计算机模型生成的图像





# 第1章 绪论

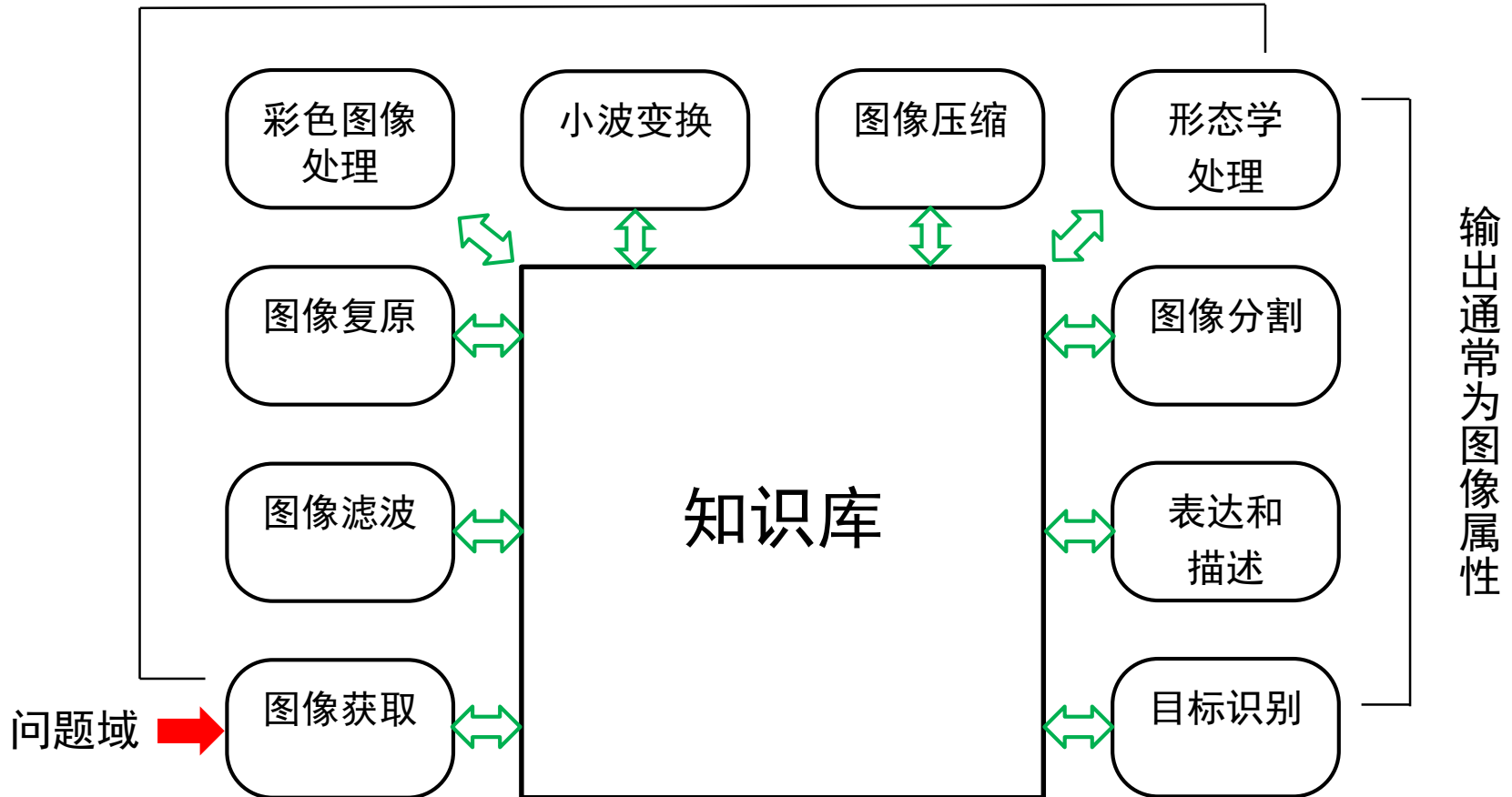
---

- 1.1 什么是数字图像处理
- 1.2 数字图像处理的起源
- 1.3 数字图像处理应用领域
- 1.4 **数字图像处理的基本步骤**
- 1.5 图像处理系统的组成
- 1.6 数字图像处理应用举例



# 1.4 数字图像处理的基本步骤

输出通常为图像







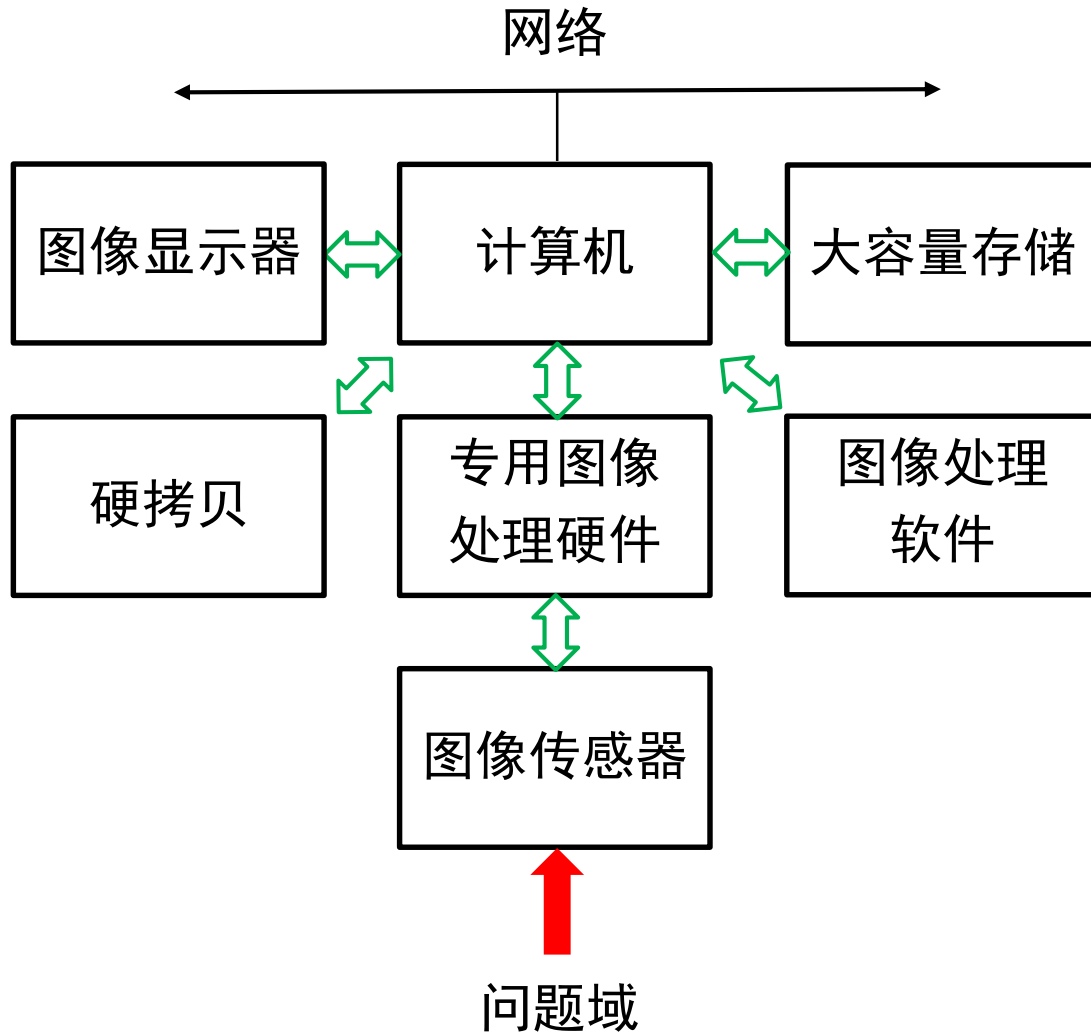
# 第1章 绪论

---

- 1.1 什么是数字图像处理
- 1.2 数字图像处理的起源
- 1.3 数字图像处理应用领域
- 1.4 数字图像处理的基本步骤
- 1.5 **图像处理系统的组成**
- 1.6 数字图像处理应用举例



# 1.5 图像处理系统的组成





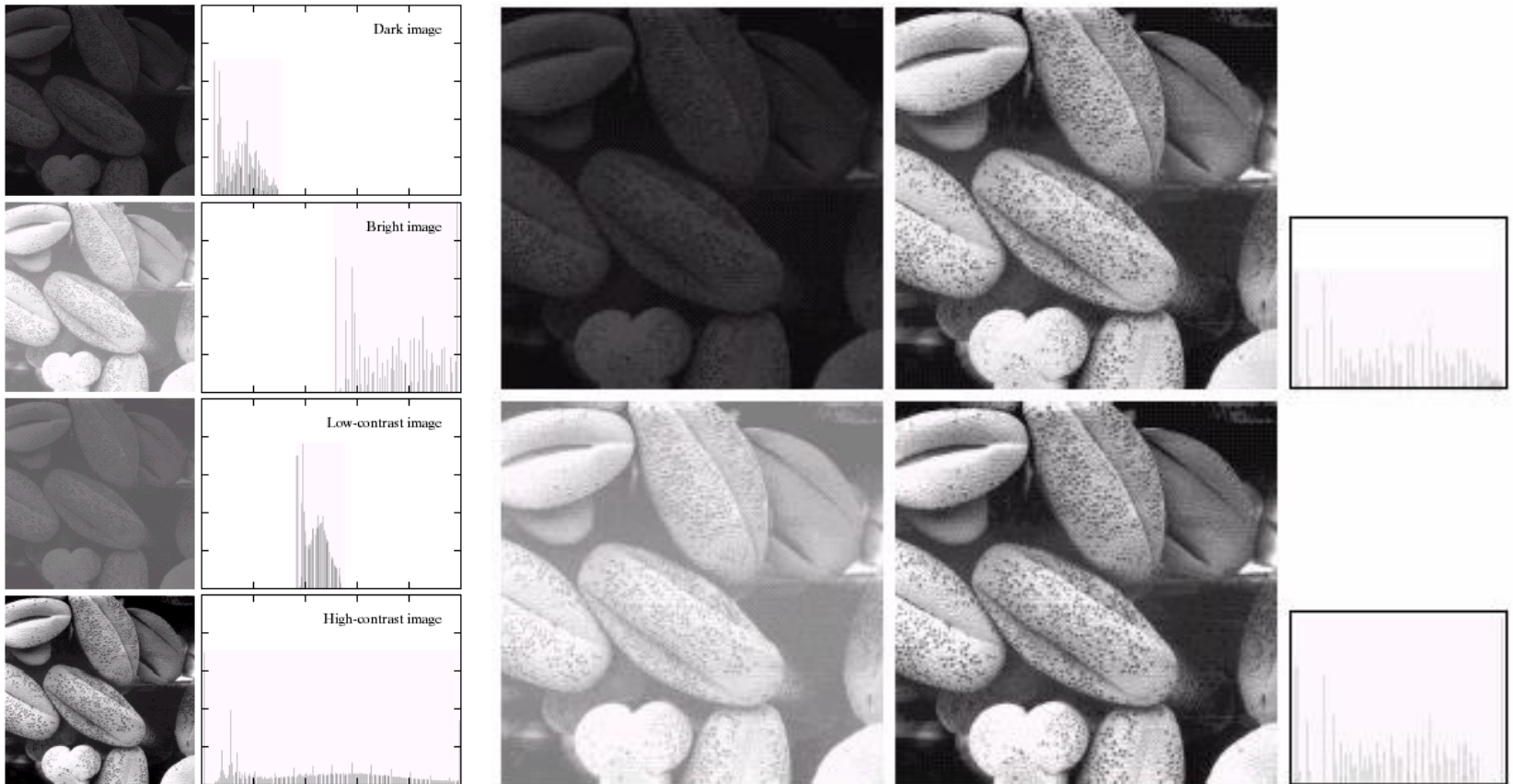
# 第1章 绪论

---

- 1.1 什么是数字图像处理
- 1.2 数字图像处理的起源
- 1.3 数字图像处理应用领域
- 1.4 数字图像处理的基本步骤
- 1.5 图像处理系统的组成
- 1.6 **数字图像处理应用举例**

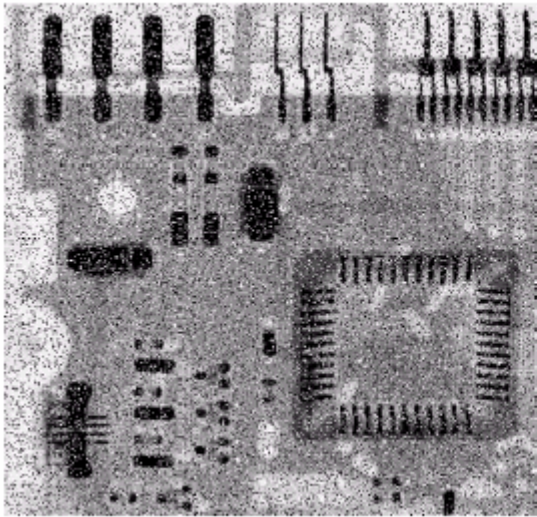
# 1.6 数字图像处理应用举例

## ➤ 直方图均衡

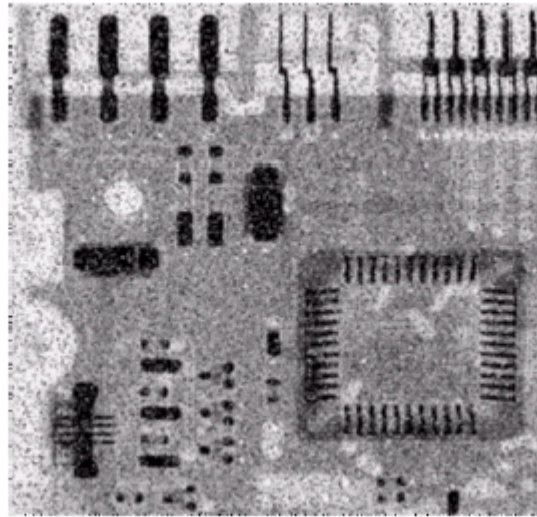


# 1.6 数字图像处理应用举例

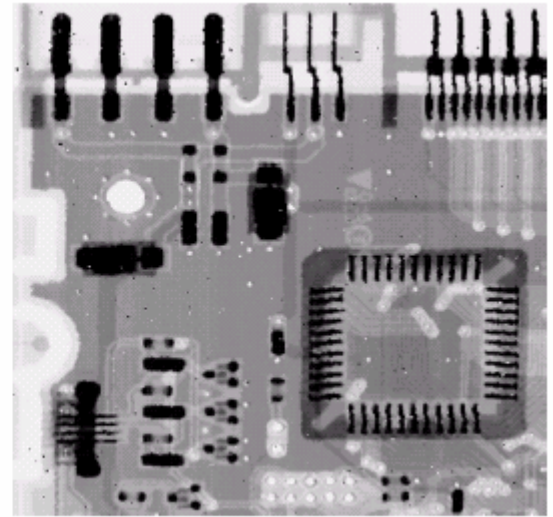
## ➤ 空域滤波



(a) 椒盐噪声图像



(b) 均值滤波

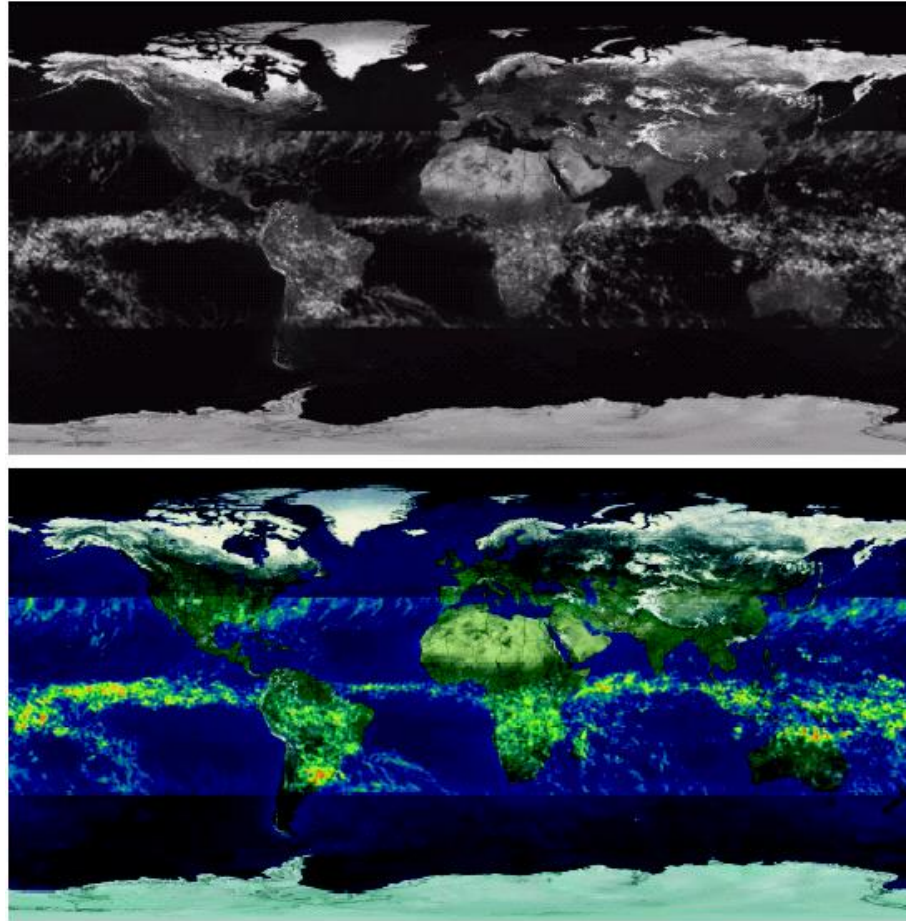


(c) 中值滤波



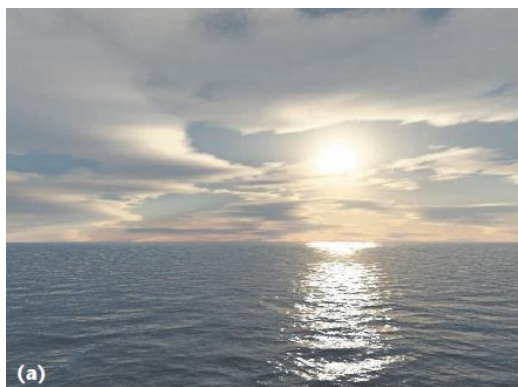
# 1.6 数字图像处理应用举例

## ➤ 伪彩色变换

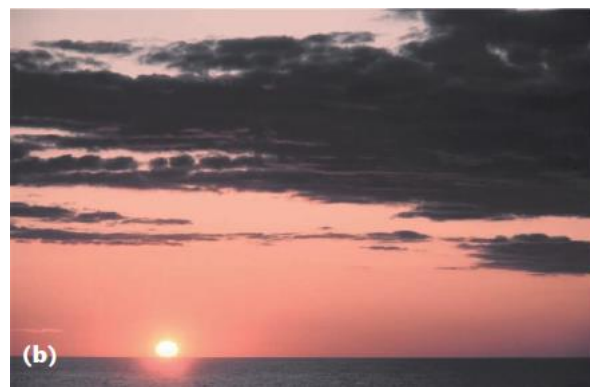


# 1.6 数字图像处理应用举例

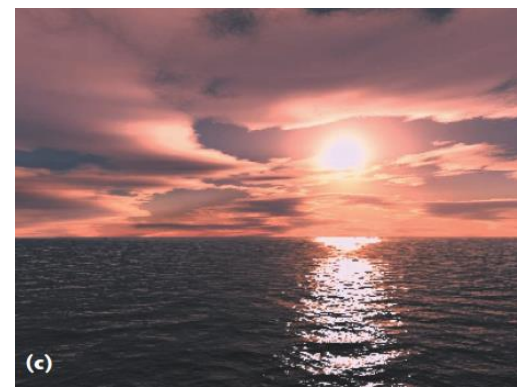
## ▶ 颜色迁移 (Color Transfer)



(a) 目标图像



(b) 参考图像

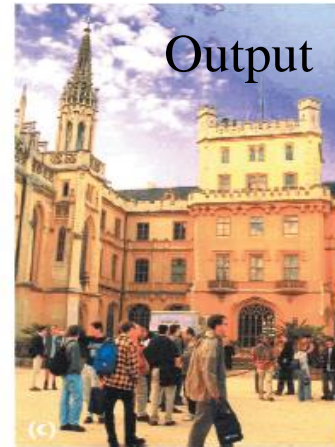
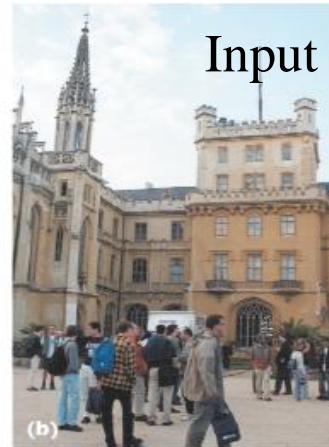
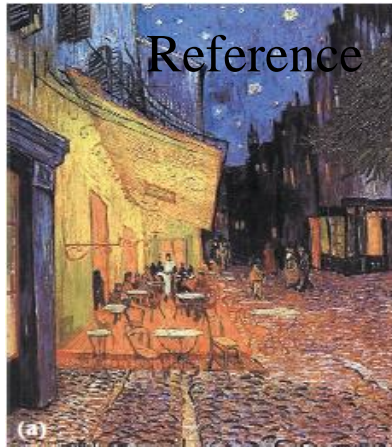


(c) 迁移结果



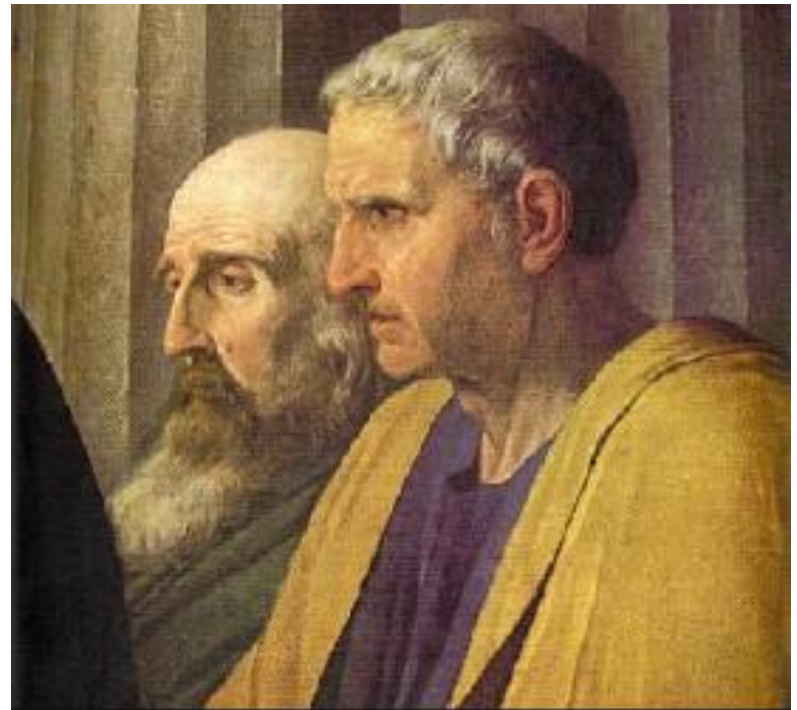
# 1.6 数字图像处理应用举例

## ➤ 颜色迁移 (Color Transfer)



# 1.6 数字图像处理应用举例

## ▶ 图像修复



# 1.6 数字图像处理应用举例

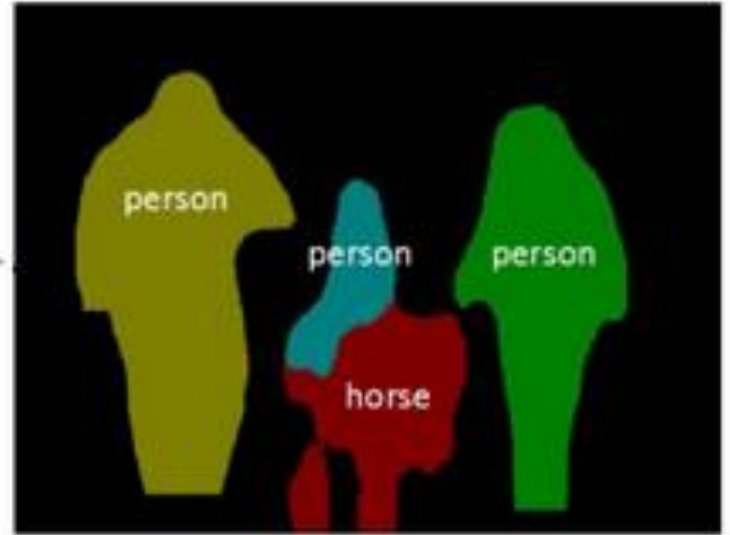
## ▶ 图像修复





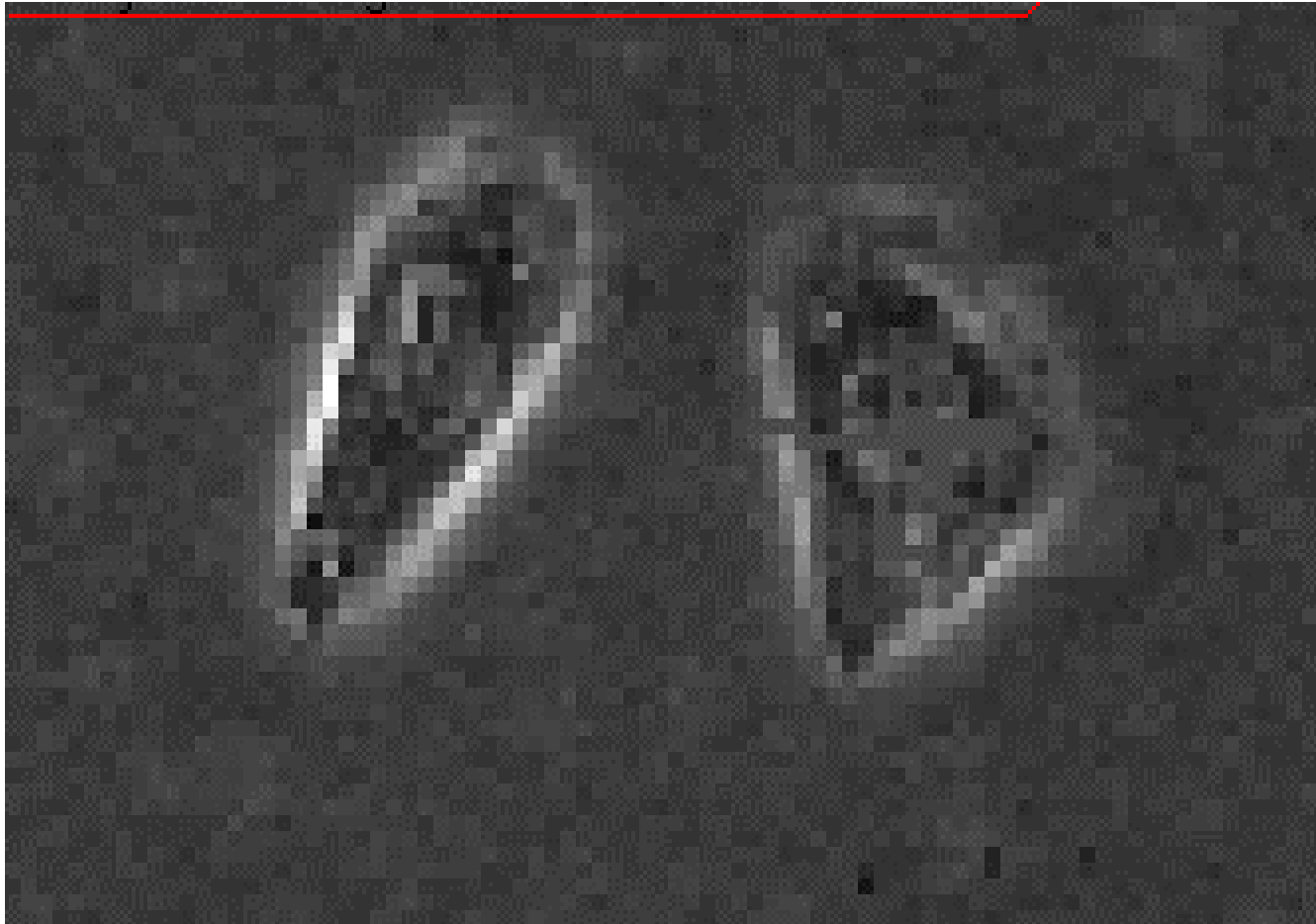
# 1.6 数字图像处理应用举例

## ➤ 图像分割



# 1.6 数字图像处理应用举例

## ➤ 图像分割



# 1.6 数字图像处理应用举例

## ► 航空遥感图像分析



# 1.6 数字图像处理应用举例

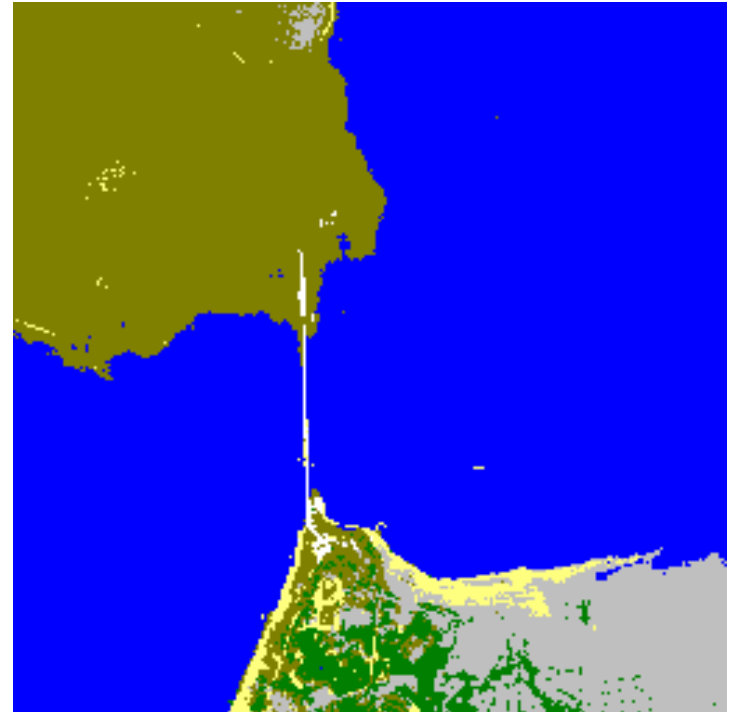
## ▶ 卫星遥感图像分析





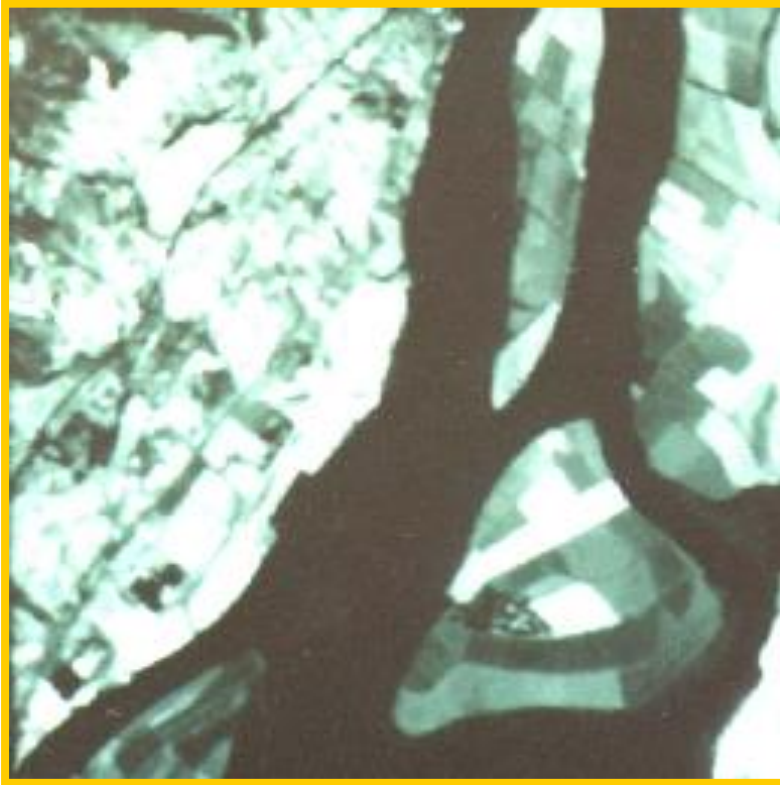
# 1.6 数字图像处理应用举例

## ➤ 遥感分类

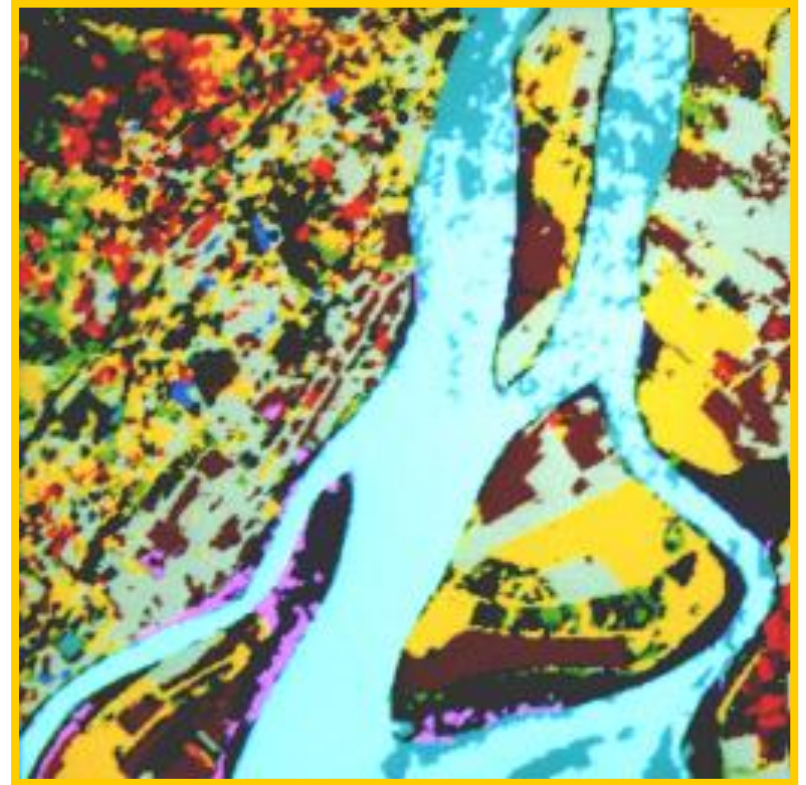


# 1.6 数字图像处理应用举例

## ▶ 遥感分类



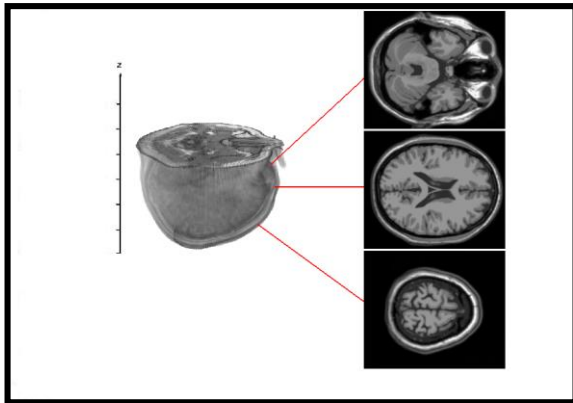
(a) 里斯本地区TM遥感图像（第六波段）



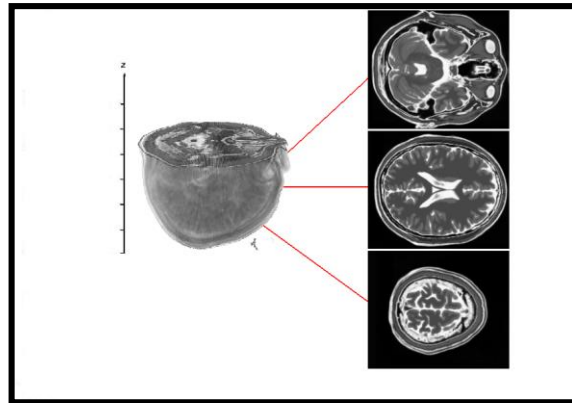
(b) 采用有监督分类的十五大类分类结果

# 1.6 数字图像处理应用举例

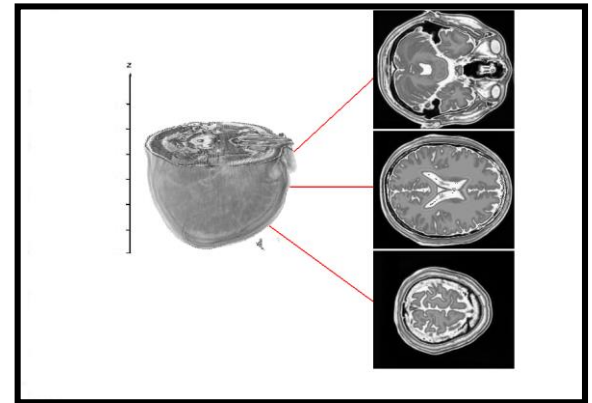
## ▶ 图像融合



(a) MR Image T1 Band



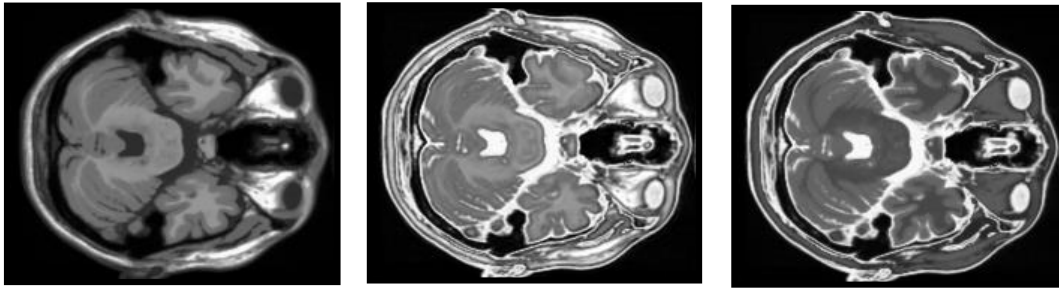
(b) MR Image T2 Band



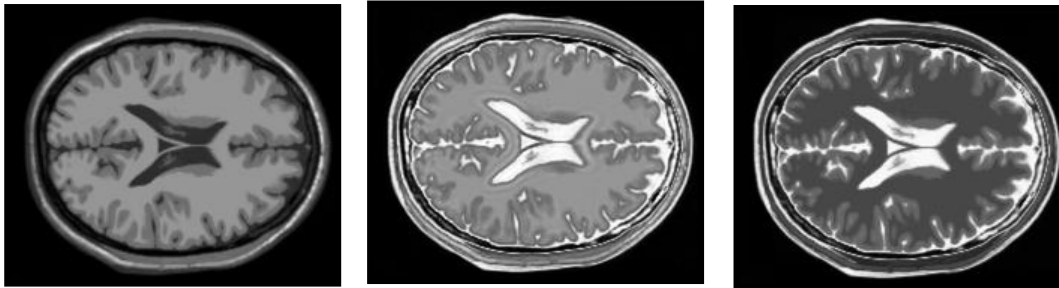
(c) Fused Image

# 1.6 数字图像处理应用举例

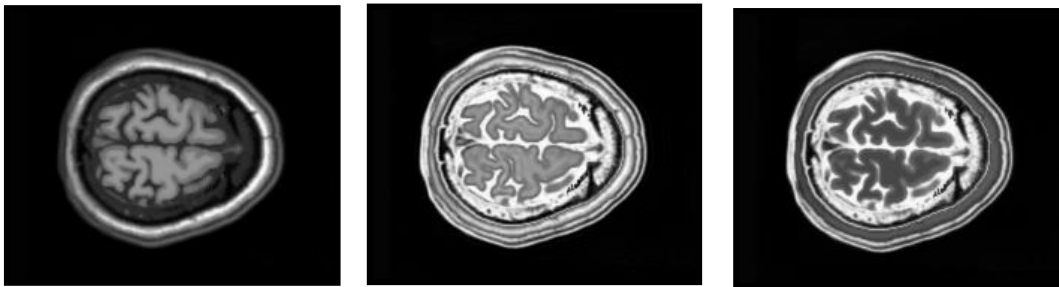
## ➤ 图像融合



(a) Result 1



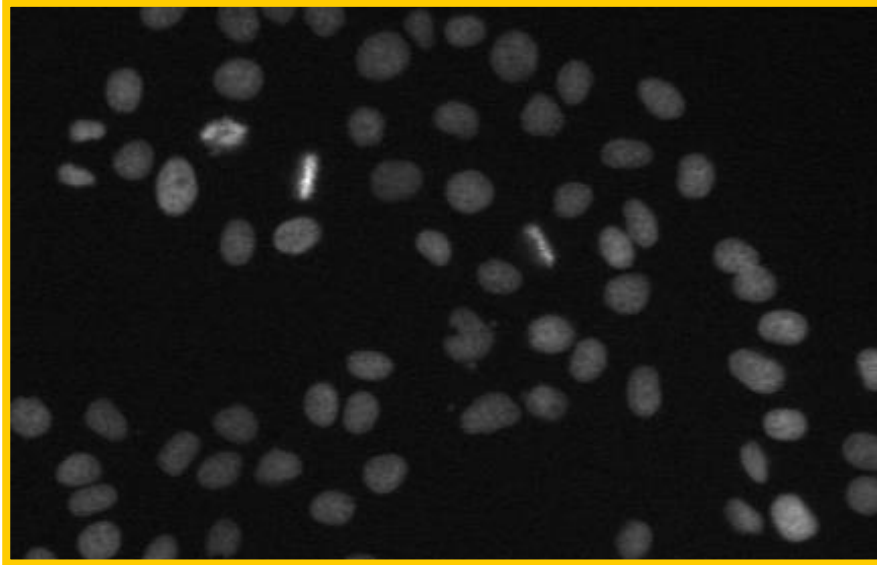
(b) Result 2



(c) Result 3

# 1.6 数字图像处理应用举例

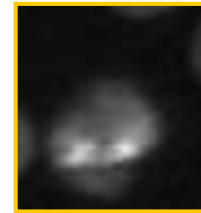
## ▶ 辅助制药



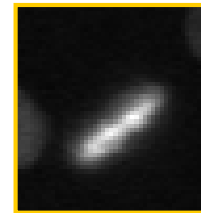
(a) 细胞显微荧光图像



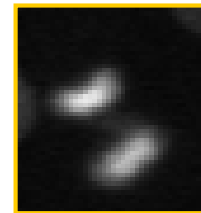
(a) 分裂间期



(b) 分裂前期



(c) 分裂中期

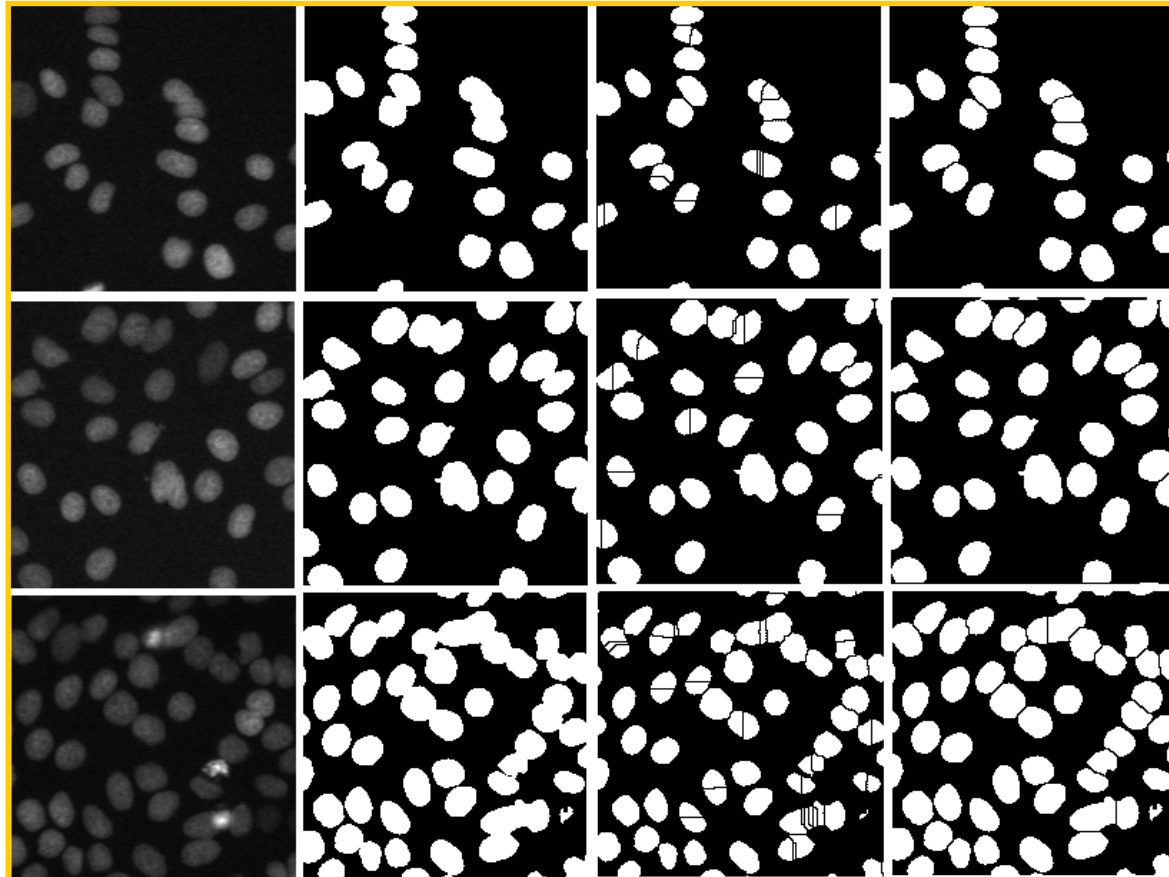


(d) 分裂后期



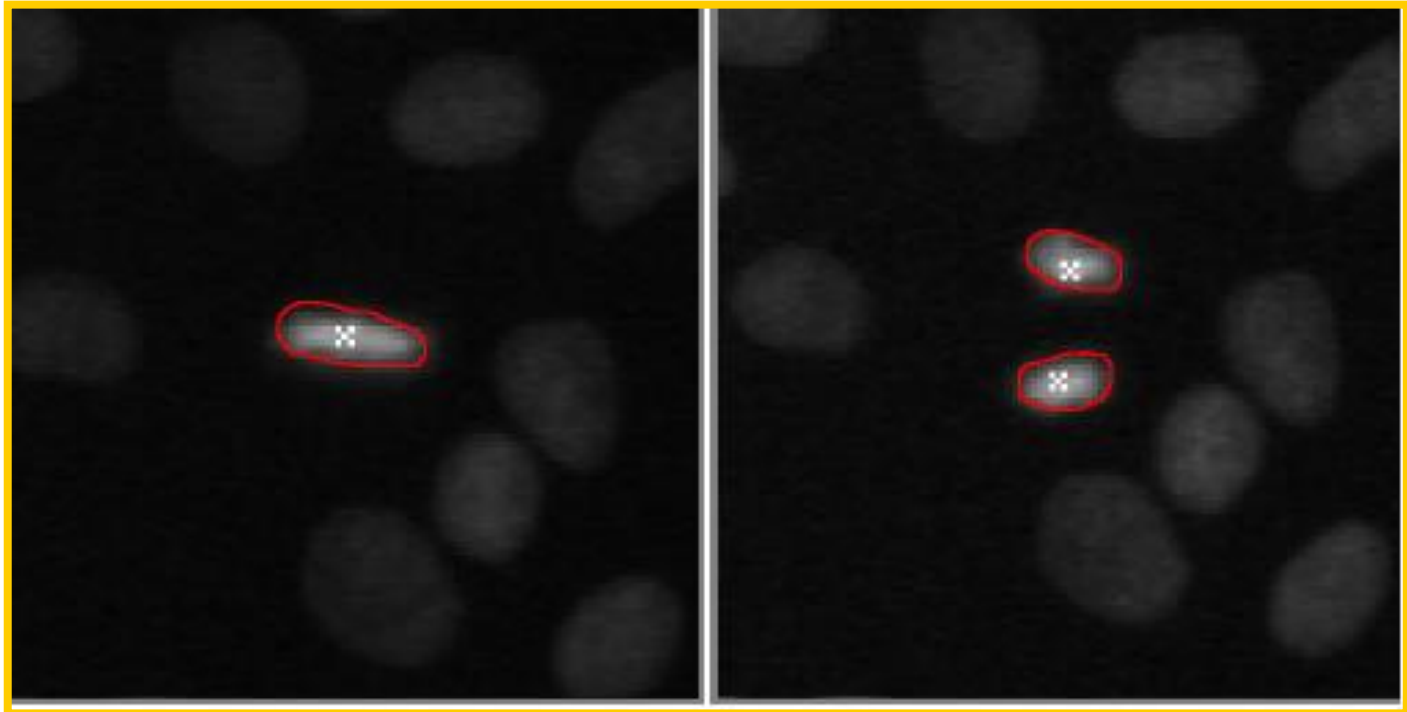
# 1.6 数字图像处理应用举例

## ➤ 辅助制药



# 1.6 数字图像处理应用举例

## ➤ 辅助制药

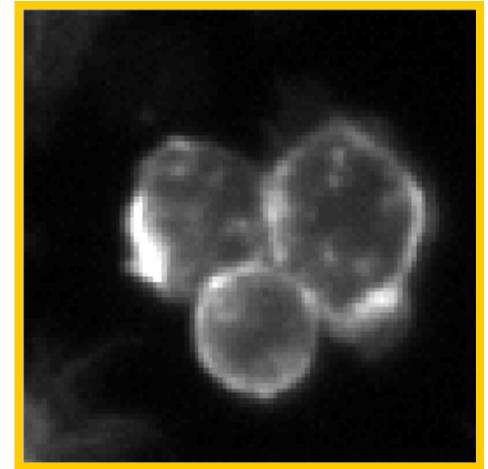
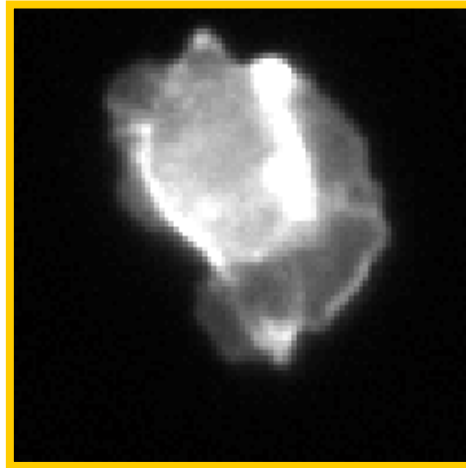


跟踪



# 1.6 数字图像处理应用举例

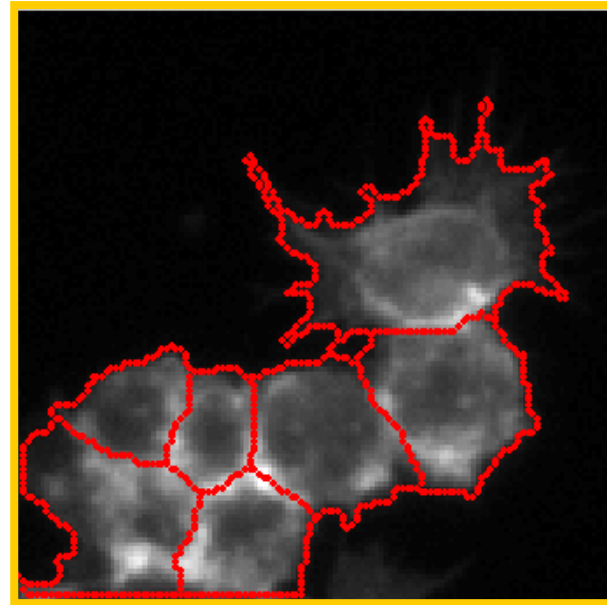
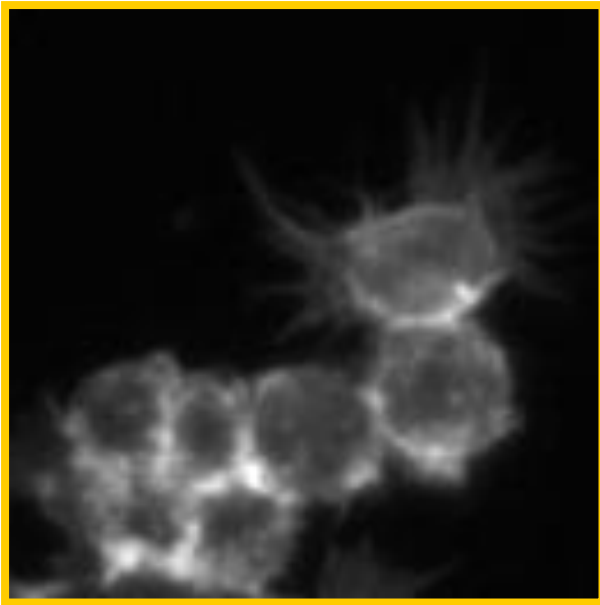
## ▶ 基于图像分析的生物学



RNAi Cell Image分割

# 1.6 数字图像处理应用举例

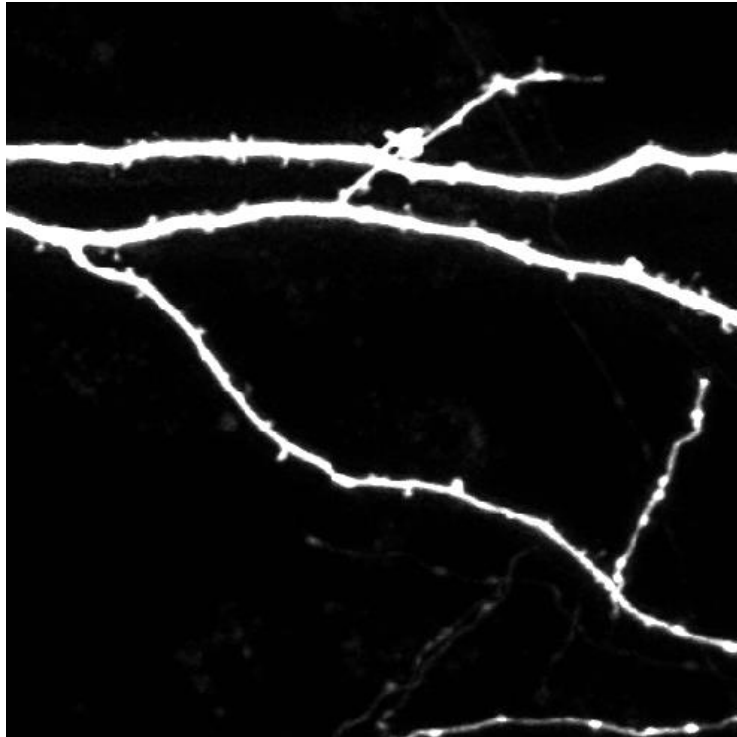
## ➤ 基于图像分析的生物学



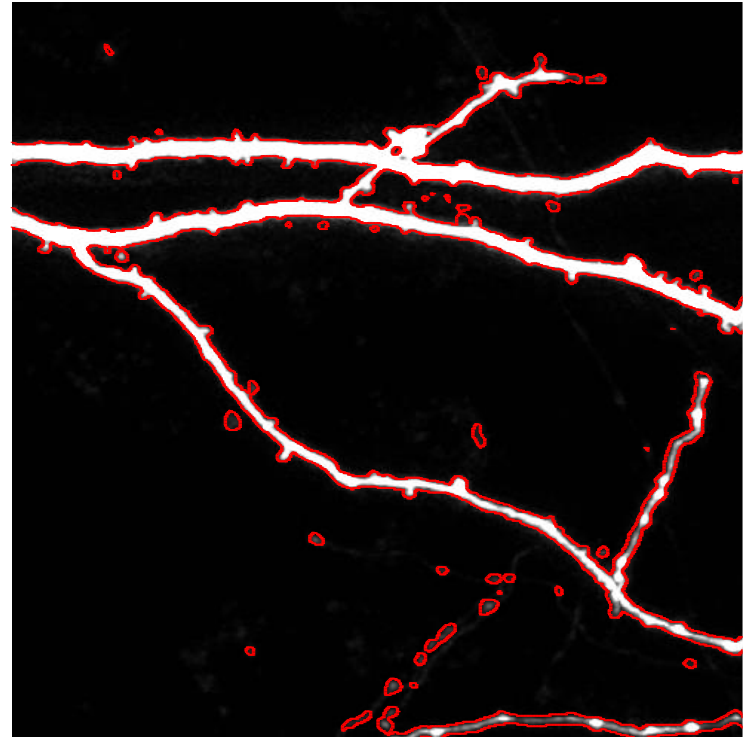
RNAi Cell Image分割

# 1.6 数字图像处理应用举例

## ▶ 神经细胞图像处理与分析



(a). Original image

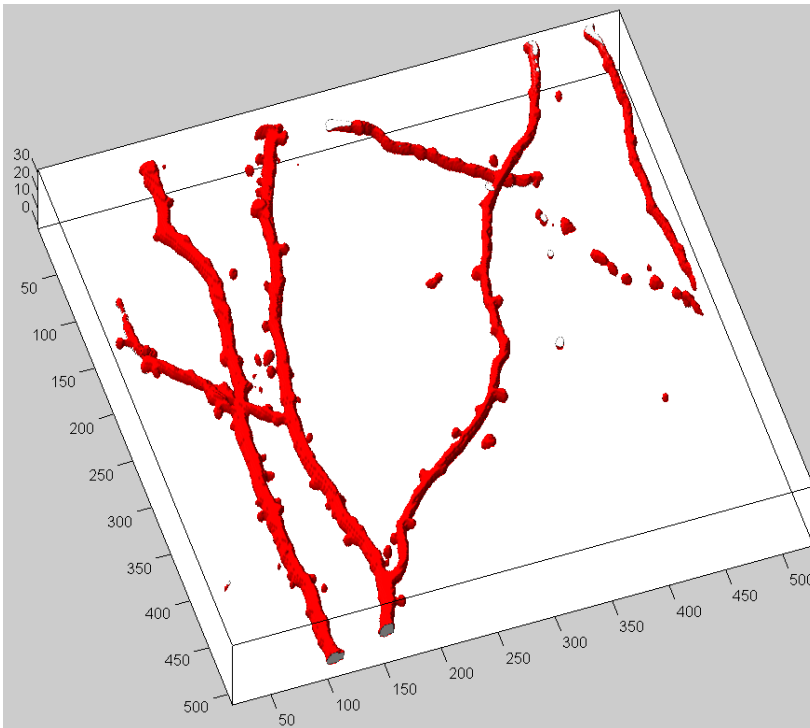


(b). Segmentation result

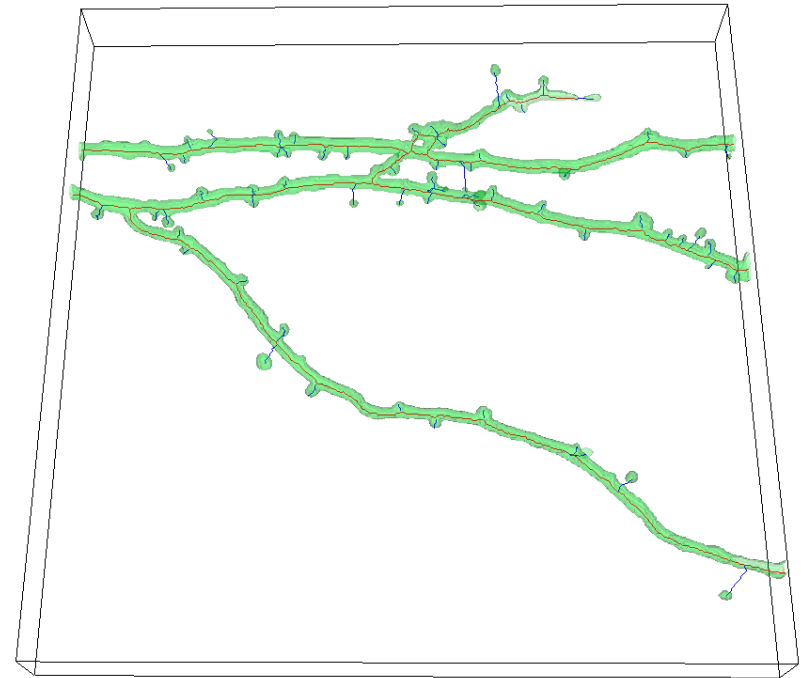
**Denrite Image分割**

# 1.6 数字图像处理应用举例

## ➤ 神经细胞图像处理与分析



(c). 3D view of segmentation result



(d). Surface rendering and skeleton, each branch in the skeleton corresponds to a spine

# 1.6 数字图像处理应用举例

## ➤ 人脸整容



# 1.6 数字图像处理应用举例

## ➤ 人脸整容



# 1.6 数字图像处理应用举例

## ➤ 图像检索

Database: thousands -> millions -> billions

- Retrieve identical images (√)
- Retrieve nearly duplicated images: cropped, edited, resized (√)



- Retrieve images containing similar objects (√)



- Retrieve similar semantics (X)





# 1.6 数字图像处理应用举例

## □ Problem definition

- Goal: Search images with partial-duplicated patches in a large corpus of web image dataset.



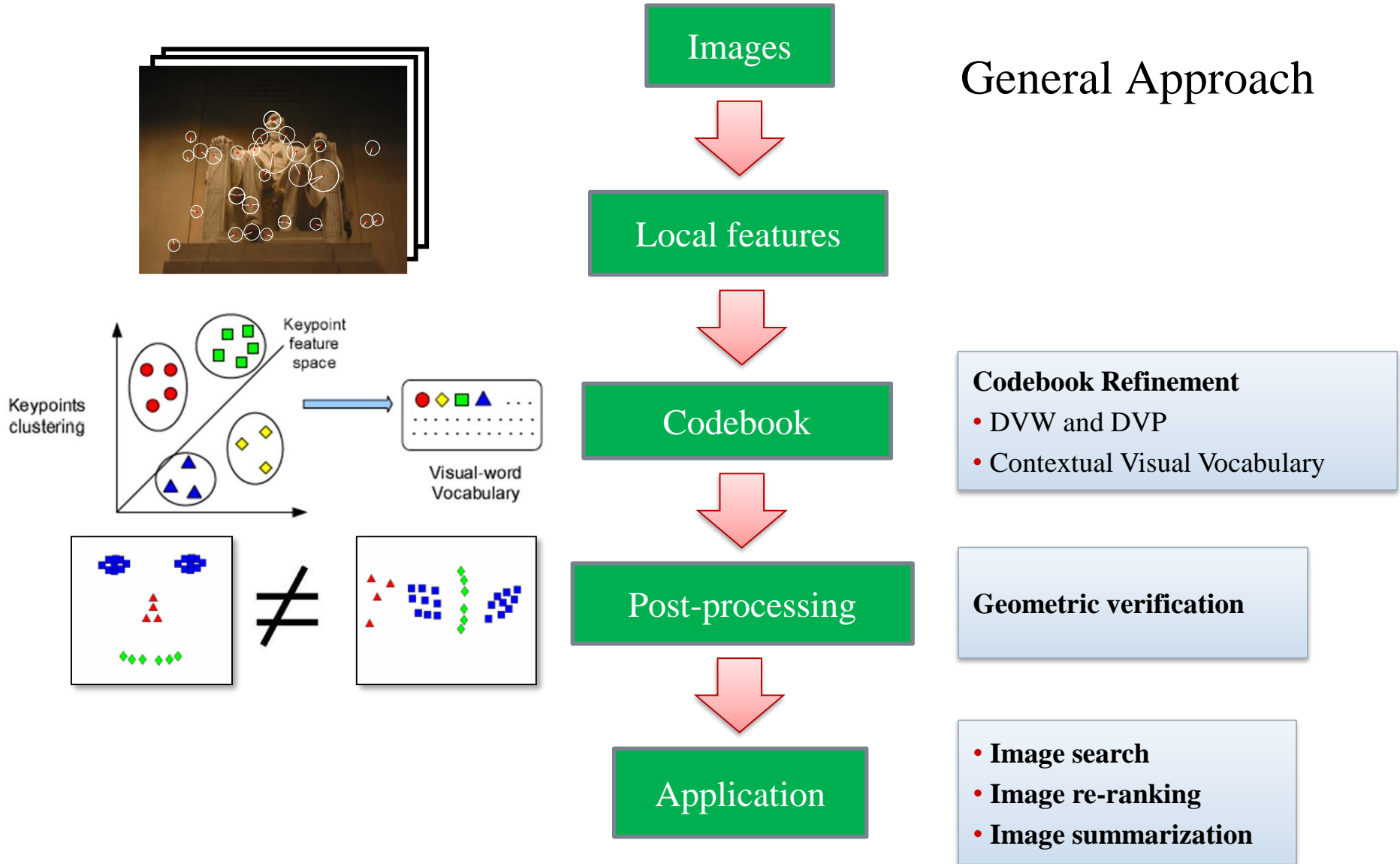
## □ Partially duplicated target image

- Edit the original image with changes in scale, cropping, and partial occlusion, *etc.*

## □ Different from image-based object retrieval

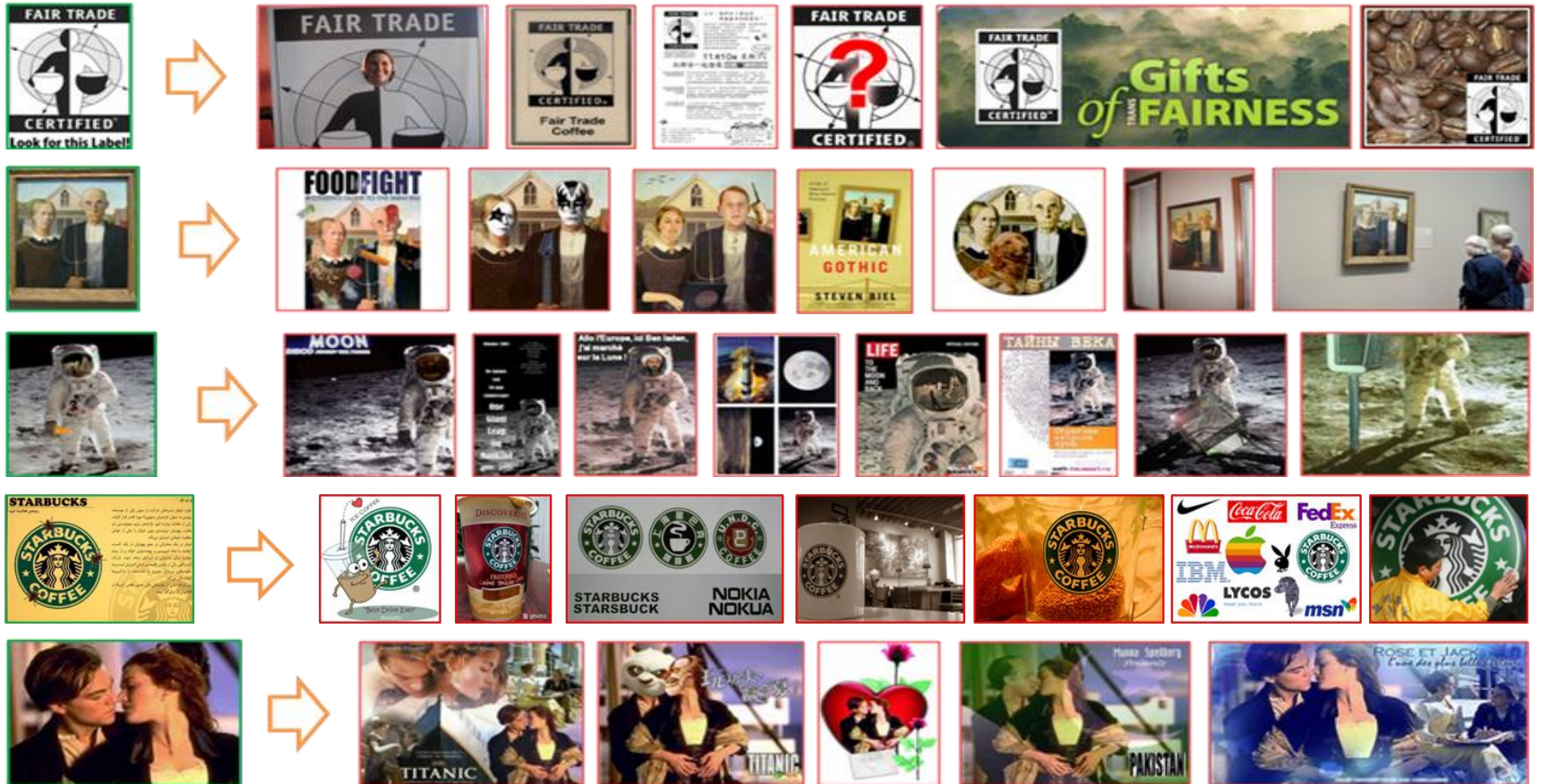
- Variations due to 3D view-point change, object-class variability, *etc.*

# 1.6 数字图像处理应用举例



# 1.6 数字图像处理应用举例

## Sample Results

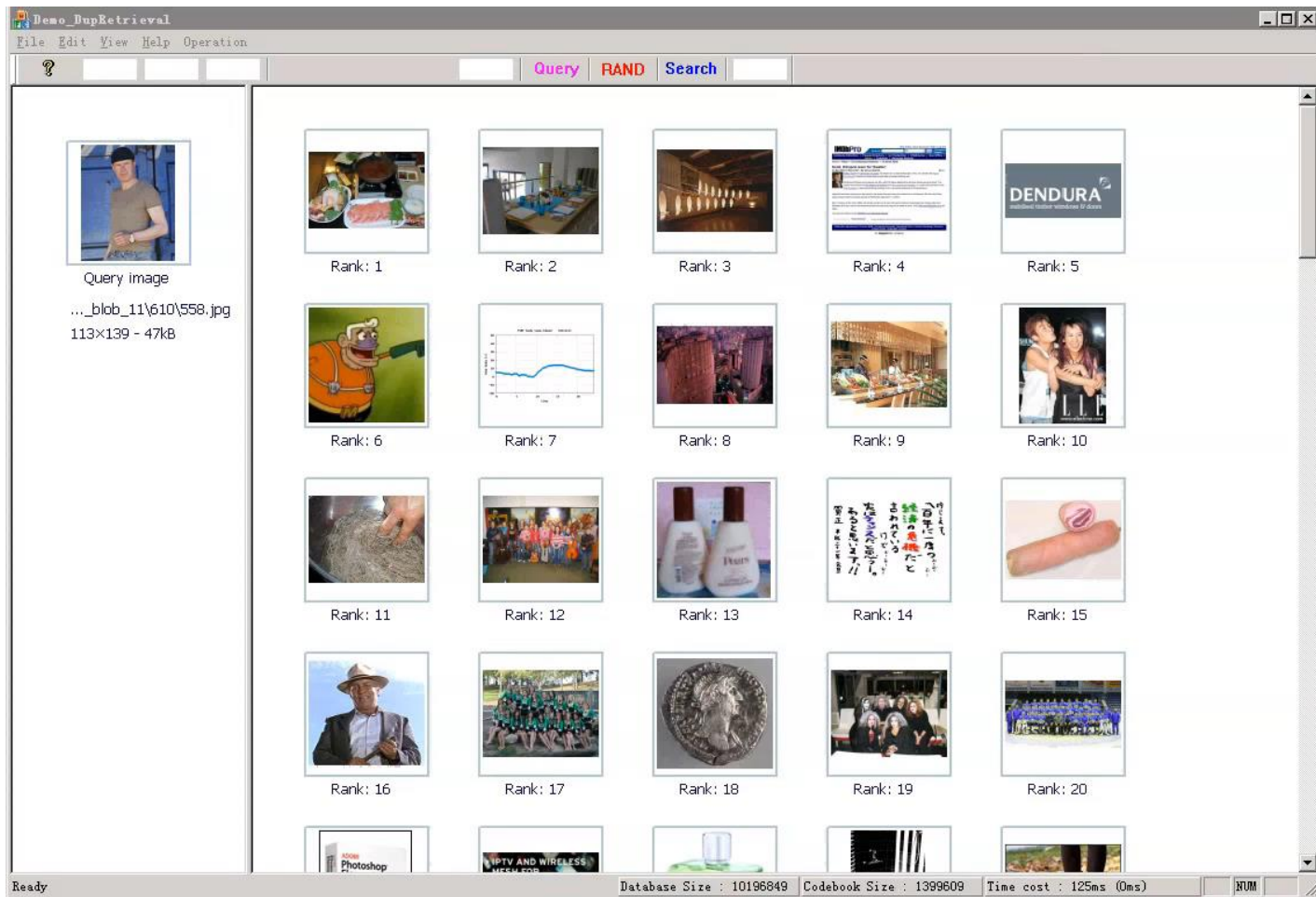


Queries

(selected from those before the first false positive)

# 1.6 数字图像处理应用举例

## Demo





# 1.6 数字图像处理应用举例

## ➤ 视频分析：运动目标跟踪



# 1.6 数字图像处理应用举例

## ▶ 视频分析：Video Inpainting





# 1.6 数字图像处理应用举例

## ▶ 视频分析：Video Summary



(a) Original Video



(b) Video Summary

# 1.6 数字图像处理应用举例

## ▶ 视频分析: Video Analysis Coding



(a) Sample



(b) Sample



(c) Ours



# 参考书目

教材：

[美] Rafael C. Gonzalez著；阮秋琦 等 译. 《数字图像处理》  
(第三版)，2017.

参考书：

1. 章毓晋. 《图像工程》（上册）：图像处理（第3版），2012.
2. [美] Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins著；  
阮秋琦 译. 国外电子与通信教材系列：《数字图像处理》  
**(MATLAB版) (本科教学版)**，电子工业出版社，**第2版，2014.**
3. Russ J C, The Image Processing Handbook, 4th Ed. CRC Press, 2002.
4. Kenneth. R. Casteman, Digital Image Processing, Printice Hall Inc. 1996
5. W. K. PRATT, Digital Image Processing, 3rd ed., John Wiley and Sons,  
2001



# 数字图象处理

---

中国科学技术大学  
电子工程与信息科学系

主讲教师：李厚强 (lihq@ustc.edu.cn)  
周文罡 (zhwg@ustc.edu.cn)

助教：冯 浩 ([fh1995@mail.ustc.edu.cn](mailto:fh1995@mail.ustc.edu.cn))

课程主页：<https://ustc-dip.github.io/>



# 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



# 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



# 2.1 视觉感知要素

## ◆ 人眼的剖面简图

视网膜有两类感光器：

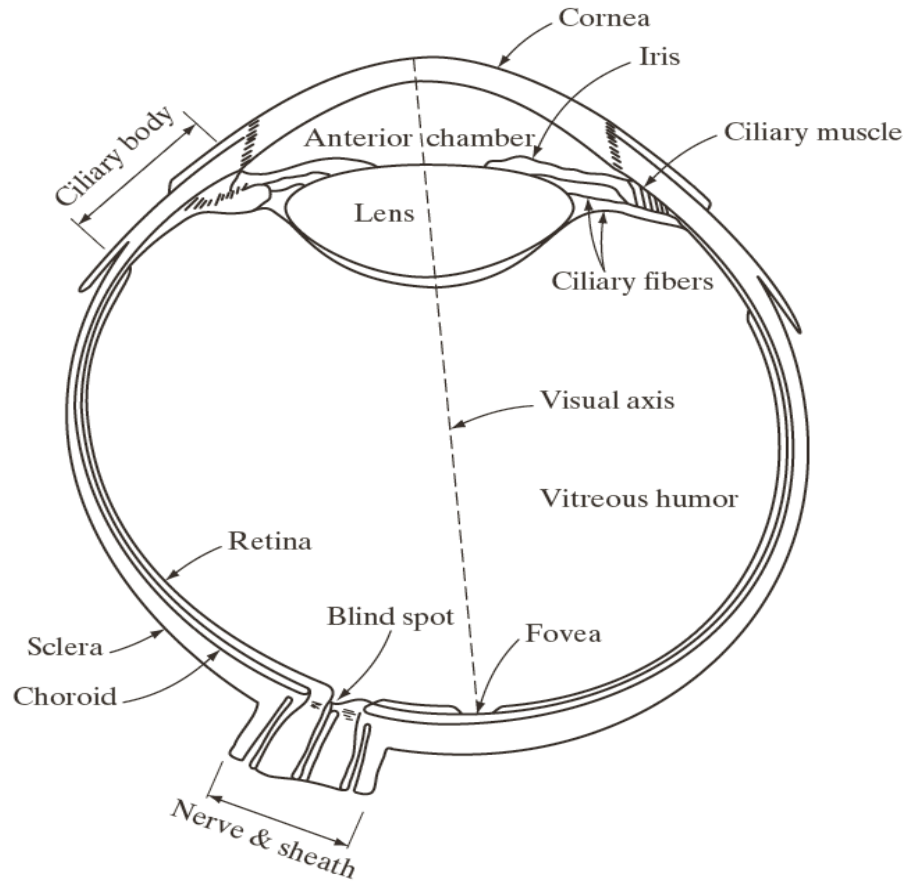
### 锥状体和杆状体

**锥状体：**白昼视觉或亮视觉 (600-700万)

对颜色高度敏感，而且每个锥状体都连接到神经末梢，人可以充分地分辨图像细节

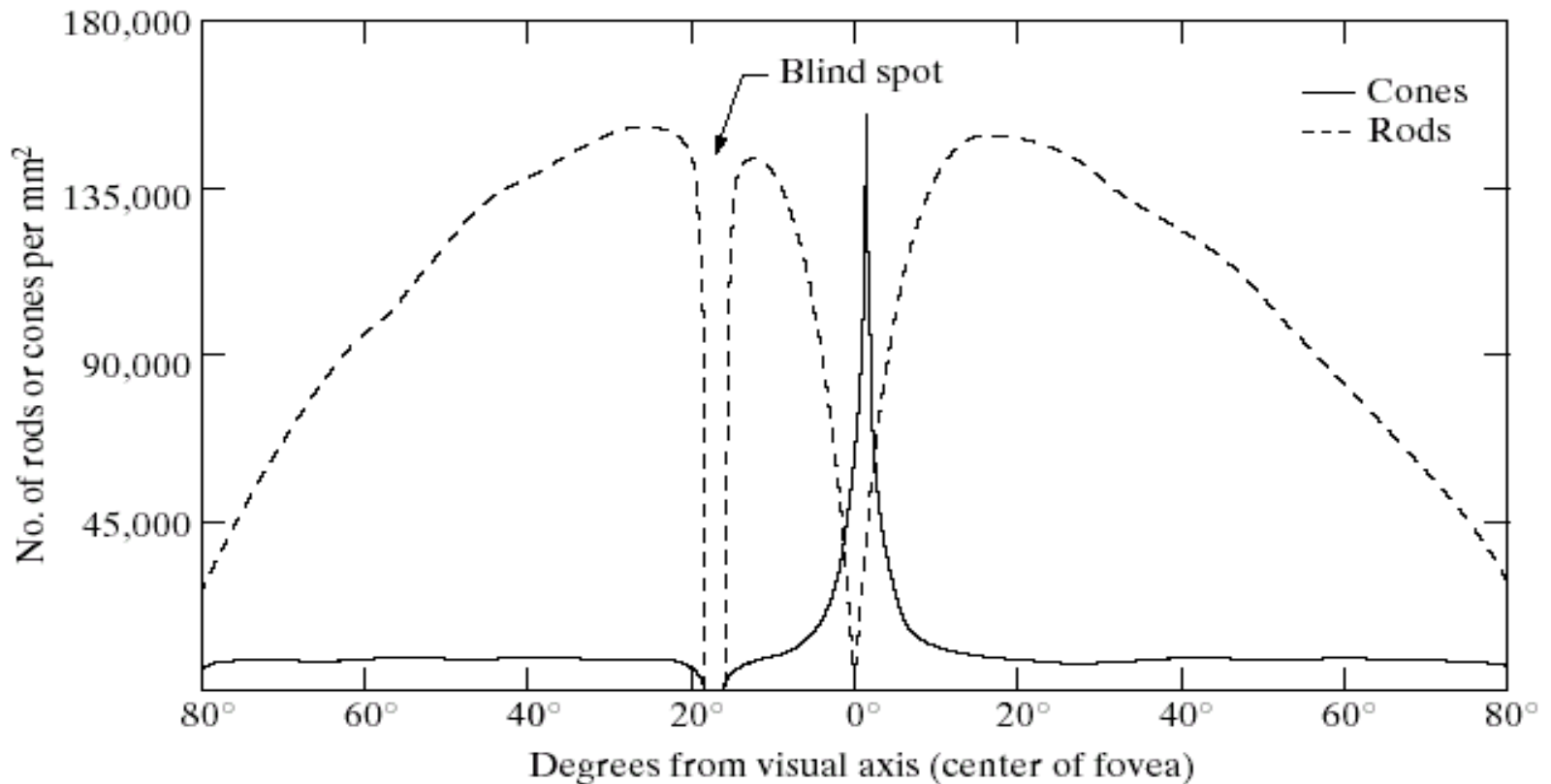
**杆状体：**暗视觉或微光视觉 (7500-15000万)

它们没有色彩感觉，而对低照明度敏感。几个杆状体才连到一个神经末梢，所以不感知事物的细节，只感知一般的总体图像。



# 2.1 视觉感知要素

视网膜上杆状体 (Rods) 与锥状体 (Cones) 的分布



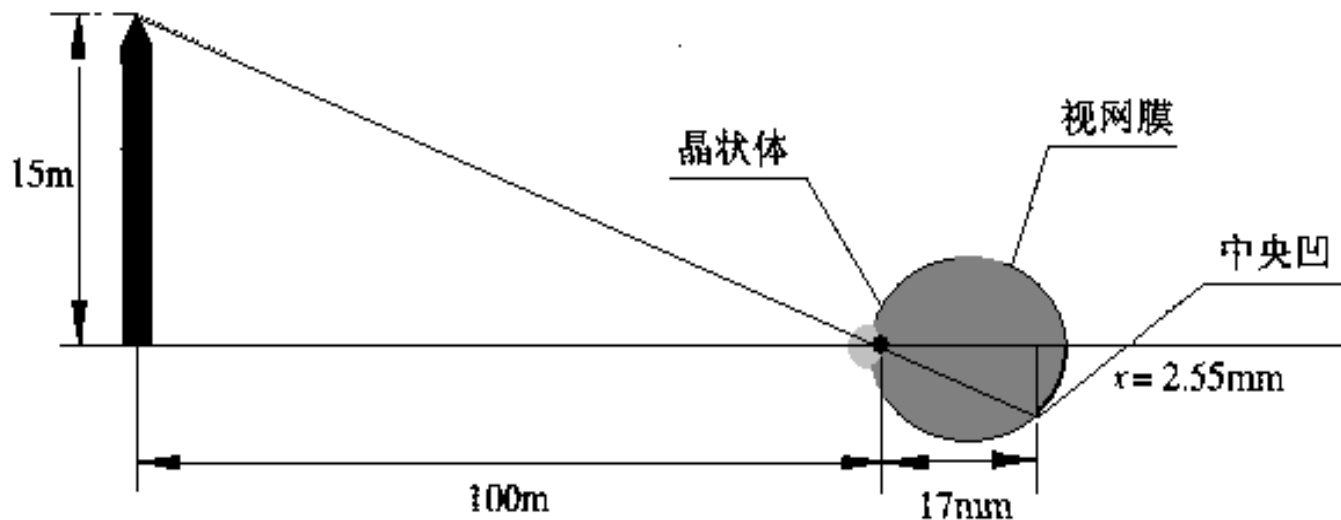


## 2.1 视觉感知要素

- 感受器的分布式关于中央凹对称的。锥状体在视网膜的中心也就是中央凹的中心区域最密。
- 从该中心向外到偏离视轴大约**20度**，杆状体的密度逐渐增大，然后向外到视网膜的极限边缘处，密度逐渐下降。
- 中央凹本身是视网膜中直径约为**1.5mm**的圆形凹坑。我们可以把中央凹看成是大小为**1.5mm\*1.5mm**的方形传感器阵列。

## 2.1 视觉感知要素

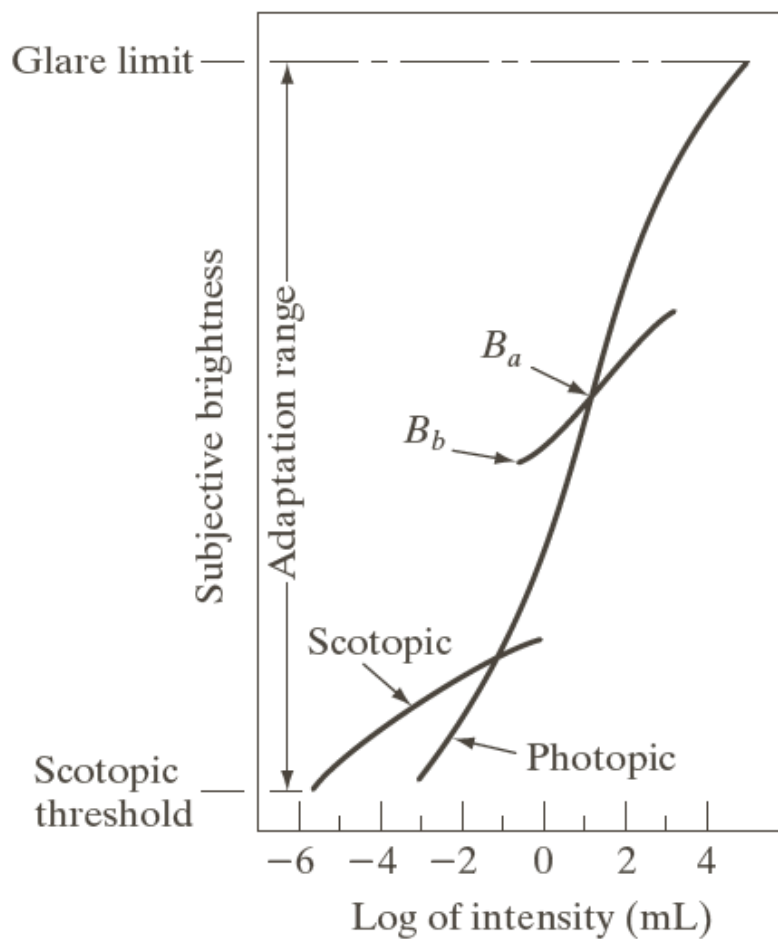
### ◆ 眼睛中图像的形成



球体前端为晶状体，相当于镜头，内壁为视网膜，相当于胶片。眼睛实现正确聚焦的焦距是通过改变晶状体的形状来得到的。这点跟相机成像不同。

# 2.1 视觉感知要素

## ◆ 亮度适应和辨别



**FIGURE 2.4**  
Range of subjective brightness sensations showing a particular adaptation level.

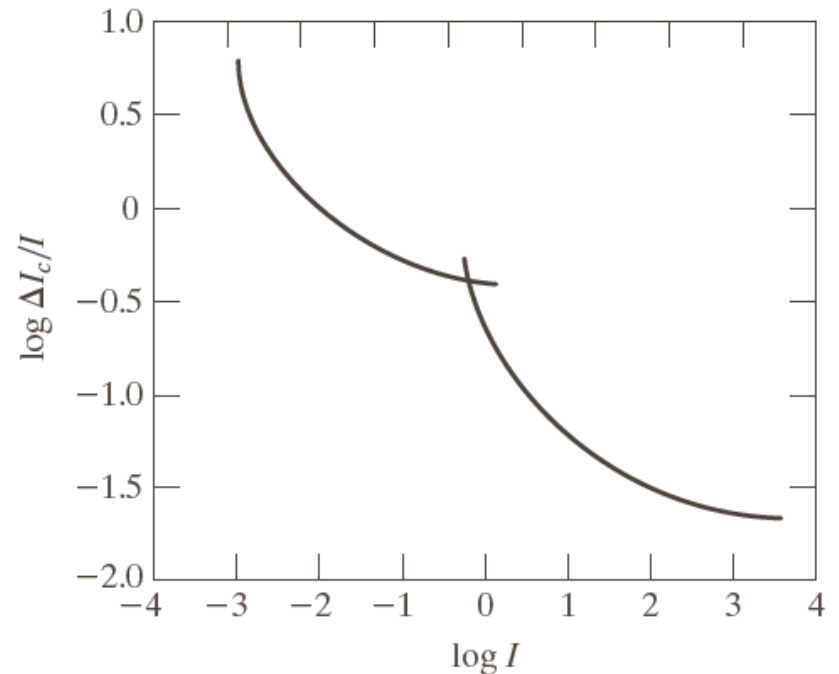
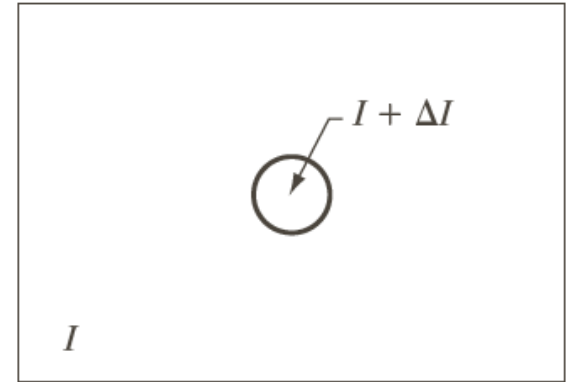
# 2.1 视觉感知要素

## ◆ 亮度适应和辨别

$\Delta I_c$ : 可辨别照明增量的50%

曲线表明，在低照明级别，亮度辨别较差，且他会随着背景照明的增加而明显改善。

2条曲线反映了这样一个事实，在低照明水平下，视觉有杆状体执行，在高照明水平下，视觉由锥状体执行。



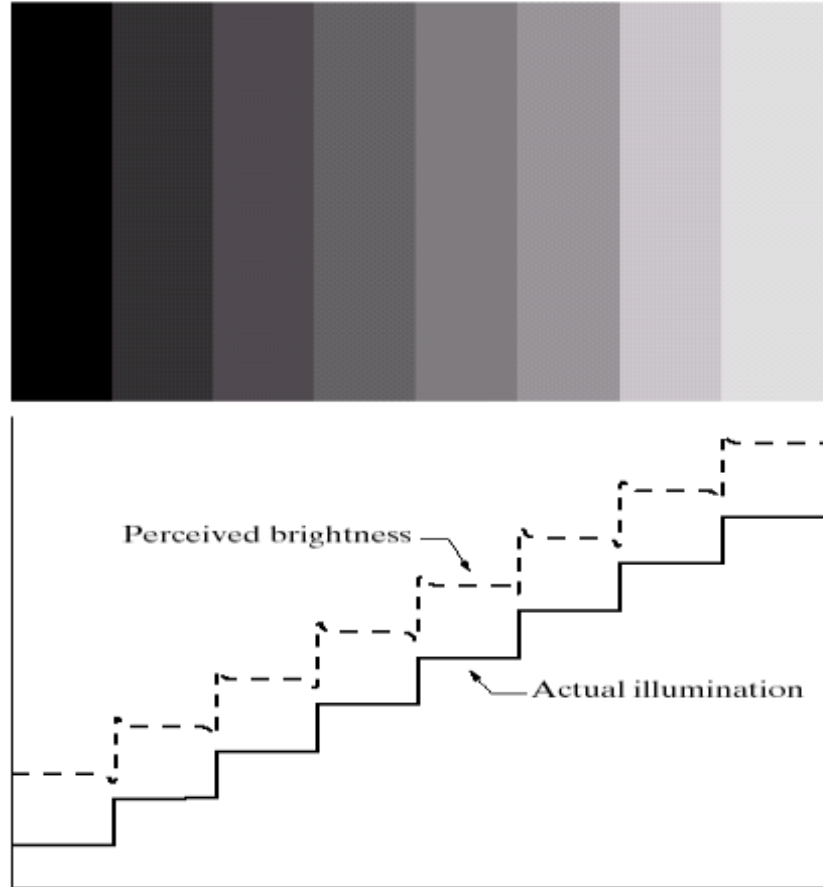


# 2.1 视觉感知要素

## 马赫带效应

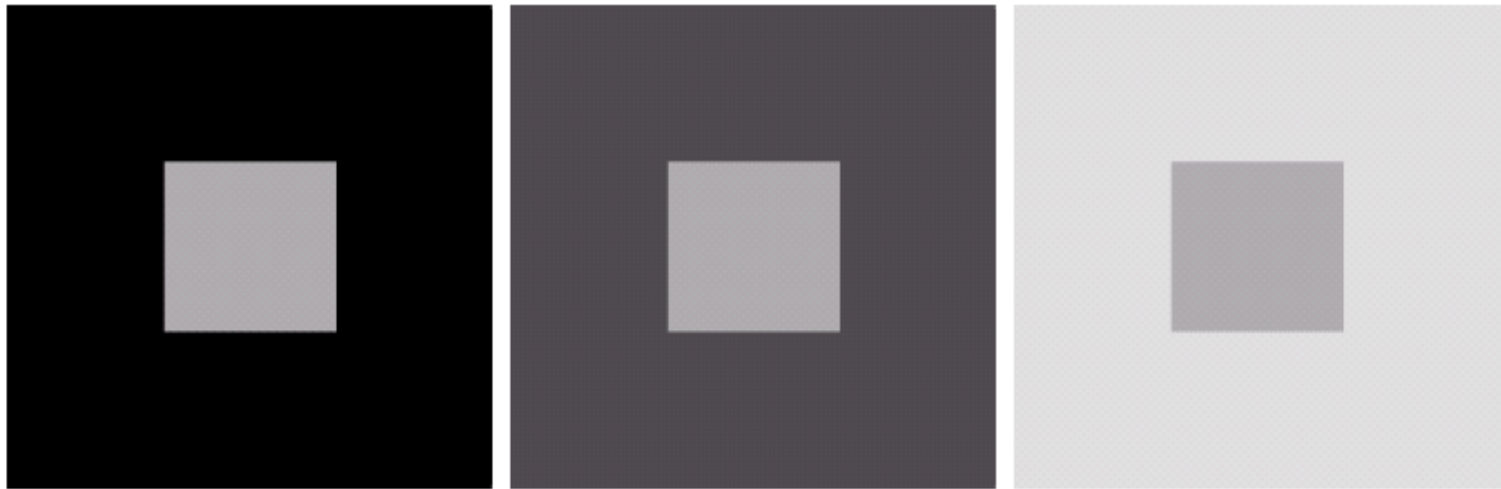
马赫带：1865年首先描述视觉系统倾向不同强度区域边界周围的”欠调”或”过调”。

证明了感觉亮度不是简单的强度函数。



## 2.1 视觉感知要素

### 同时对比现象



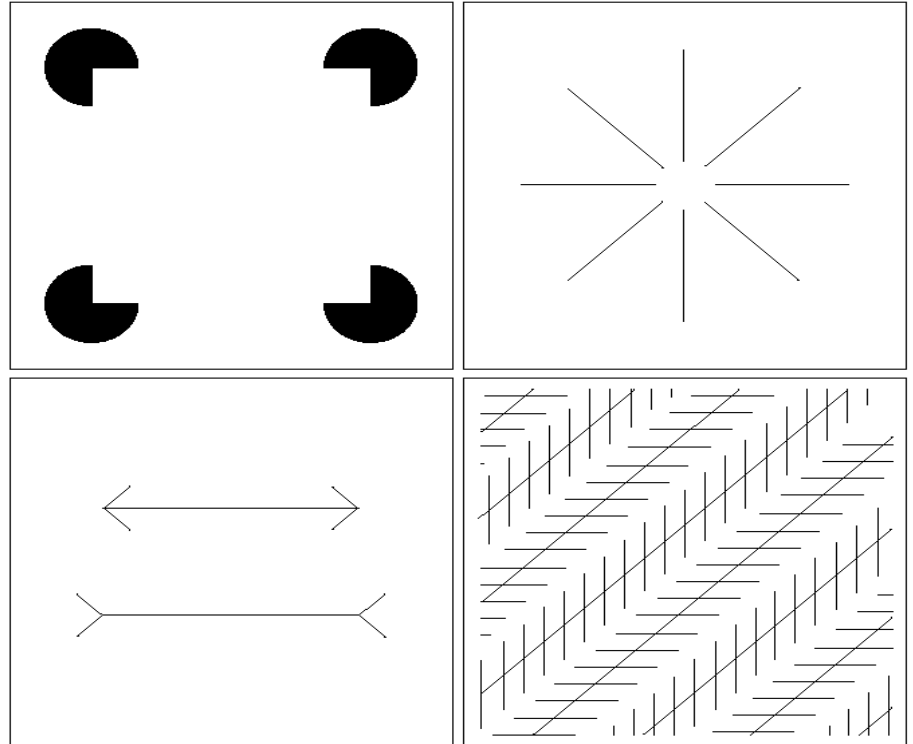
所有的中心方块有同样的亮度，但当背景变亮时，感觉它们就逐渐变暗。即感知区域的亮度并不简单地取决于其强度。同样证明了感觉亮度不是简单的强度函数。

# 2.1 视觉感知要素

## 视觉错觉

眼睛填充了不存在的信息或者错误地感知了物体的几何特点。

错觉是人类视觉系统的一种特性，这一特性尚未被人类完全了解。



左上：正方形轮廓；右上：圆轮廓；  
左下：长度错觉；右下：平行线错觉



# 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

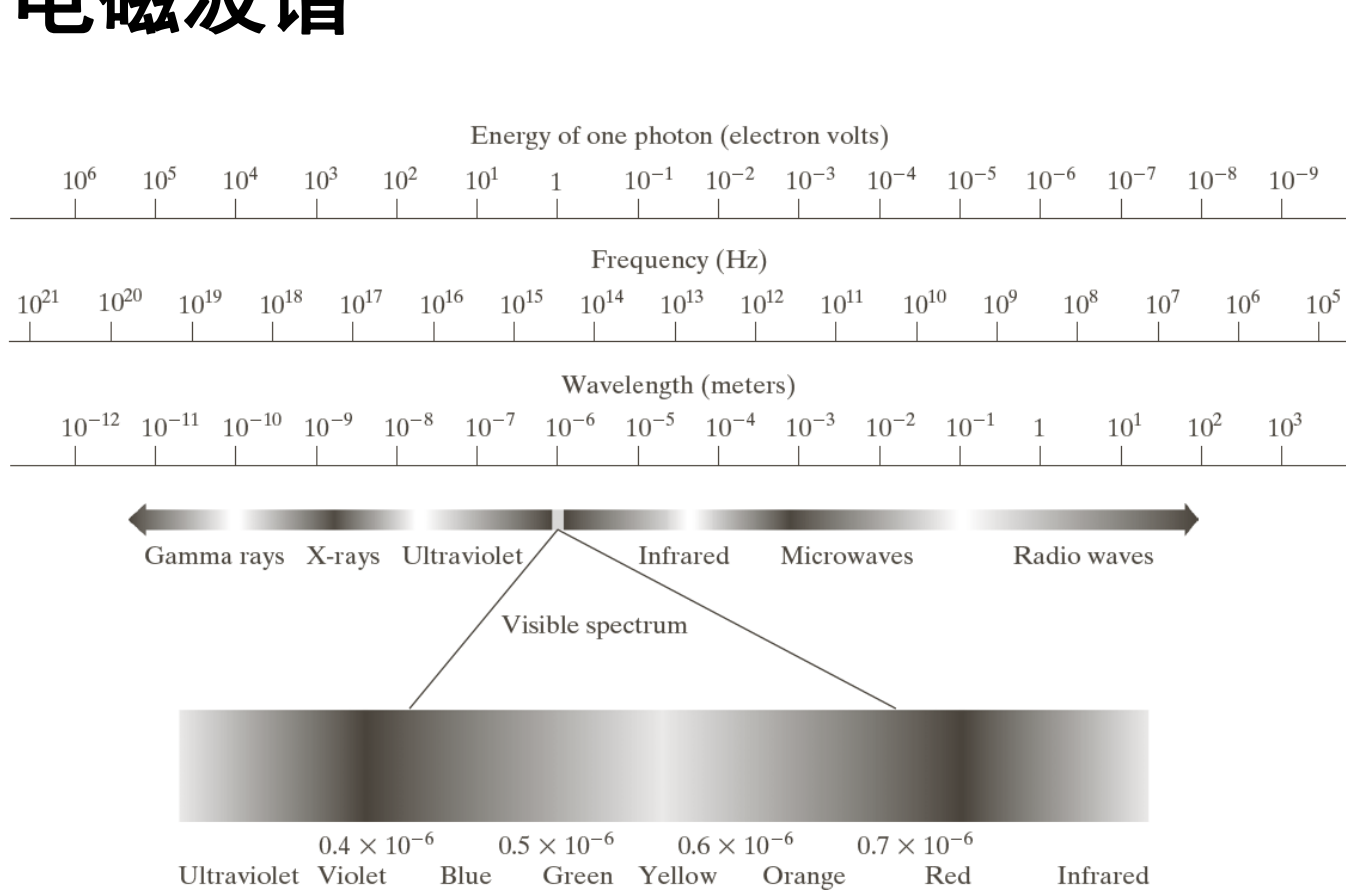
2.4 图像取样和量化

2.5 像素间的基本关系



# 2.2 光和电磁波谱

## ◆ 电磁波谱



普朗克常数

$$E = h\nu$$

$\nu$

$$\lambda = \frac{c}{\nu}$$



## 2.2 光和电磁波谱

### ◆ 可见光

- 我们感受到的可见光的色彩范围只占电磁波的一小部分。**光是一种特殊的电磁辐射**，它可以被人眼感知。
- 彩色谱分**6个区域**：红、橙、黄、绿、蓝、紫。颜色之间是**混合平滑过渡**的。
- 人感知物体的颜色由物体**反射光**的特性决定。
  - ✓ 如果一个物体以所有可见波长相对平衡地反射光，那么这个物体对观察者而言是白色。
  - ✓ 如果一个物体在可见光谱的有限范围内反射时，会呈现各种颜色色调。例如，绿色物体的反射波是范围为**500~570nm**的光，而吸收其他波长的大部分能量。
- 没有颜色的光称为**单色光或无色光**，其唯一属性是**强度**。常用灰度级表示表的强度。





## 2.2 光和电磁波谱

### ◆ 电磁波

- 电磁波可以看成是以波长 $\lambda$ 传播的正弦波，每个粒子以波的模式以光速传播和移动。
- 根据能量公式： $E = h\nu$ ，( $h$ 为普朗克常数)，能量与频率成正比。
  - ✓ 无线电波有低能量光子，而伽马射线的能量最高，这就是伽马射线对活体组织危害大的原因。
- 伽马辐射对医学和天文学成像很重要；红外线波段会辐射热，这使得它在依靠“热特性”成像的应用非常有用。
- 虽然成像主要以电磁波发射的能量为基础，但并不是唯一成像方法，例如用物体反射的声波也可用于形成超声波图像。



# 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



## 2.3 图像感知和获取

我们感兴趣的多数图像都是由“照射”源和形成图像的“场景”元素对光能的反射或吸收而产生的。

### “照射”源：

- 电磁源，如雷达、红外线或X射线系统。
- 非传统光源，如超声波，甚至计算机产生的照射模式。

### “场景”元素：

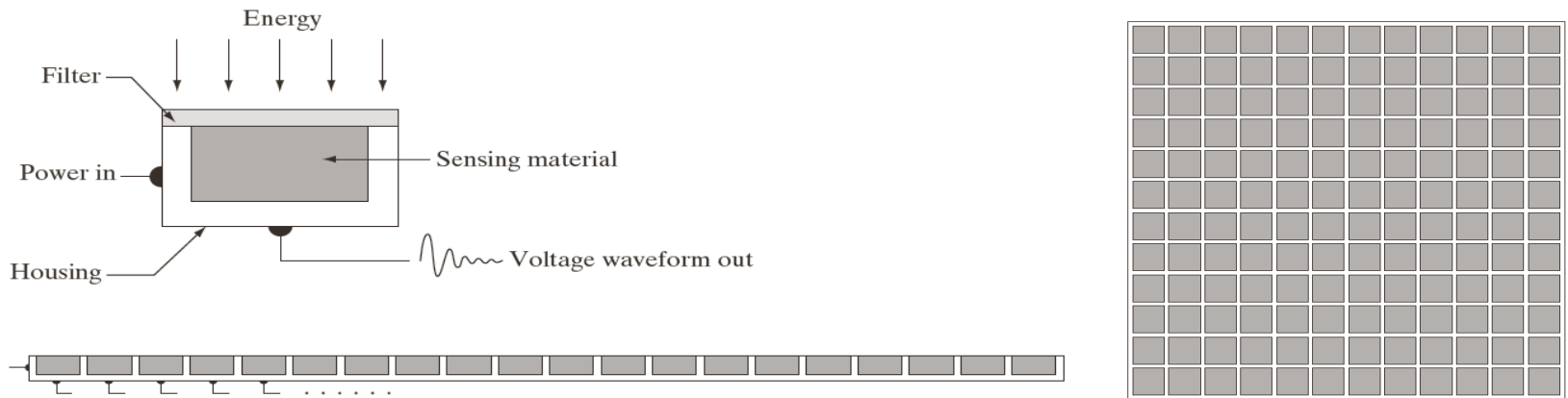
- 物体、分子、沉积岩或人类大脑。

反射能或透射能可聚焦到一个**光转换器**上（如荧光屏），光转换器再把能量转换为可见光。

## 2.3 图像感知和获取

### 成像原理

通过将输入电能和对特殊类型检测能源敏感的传感材料相组合，把输入能源转变为电压。输出连续的电压波形，这些波形的幅度和空间特性都与感知的物理现象有关。

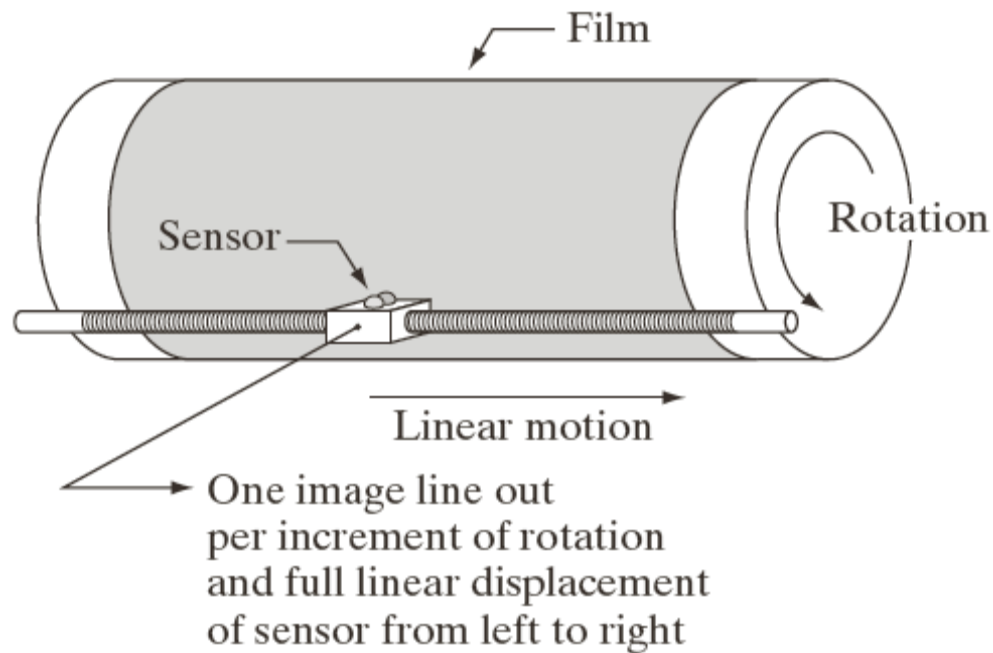


左上：单个成像传感器；左下：条带传感器；右：阵列传感器

## 2.3 图像感知和获取

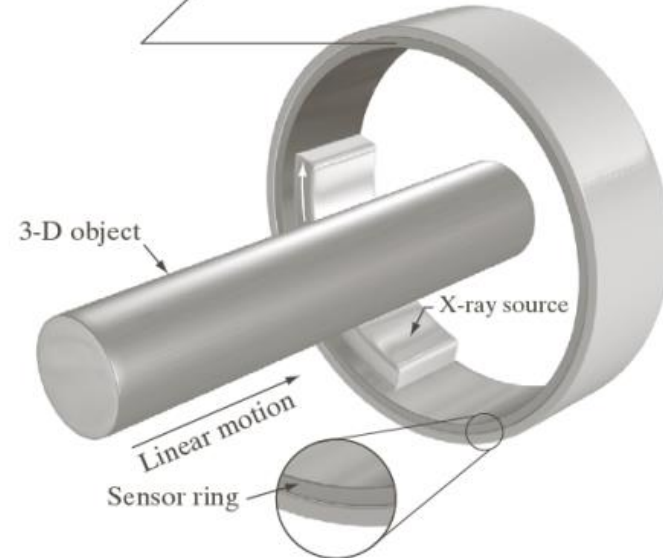
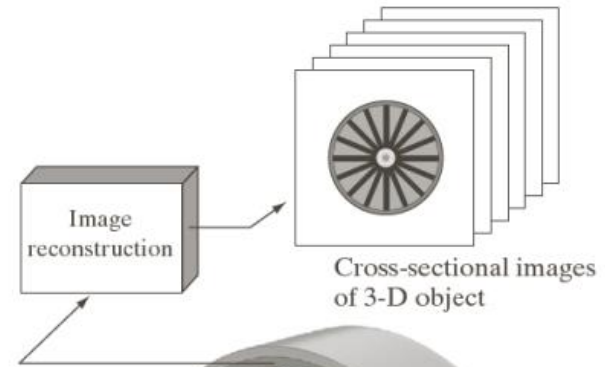
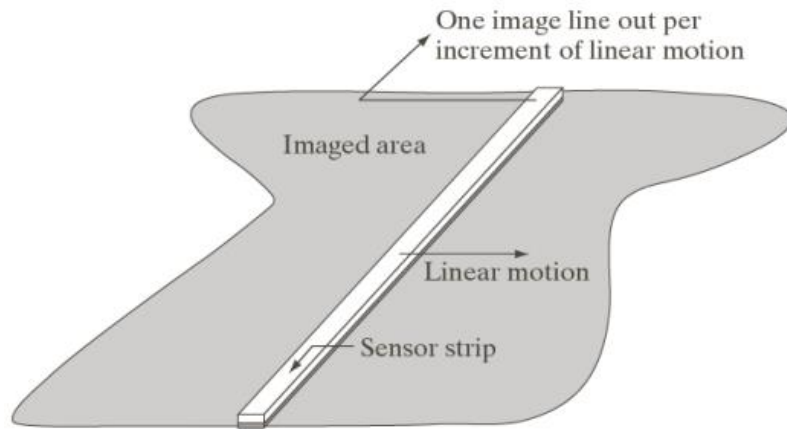
### ◆ 单个传感器通过运动来生成二维图像

胶片每旋转一个增量且传感器完整地  
从左到右线性移动  
一次，输出图像的一行。



## 2.3 图像感知和获取

### ◆ 使用条带传感器获取图像



1. (上图) 使用一个线性传感器带获取图像, 如平板扫描仪
2. (右图) 使用一个环形传感器带获取图像, 如CT和MRI成像



## 2.3 图像感知和获取

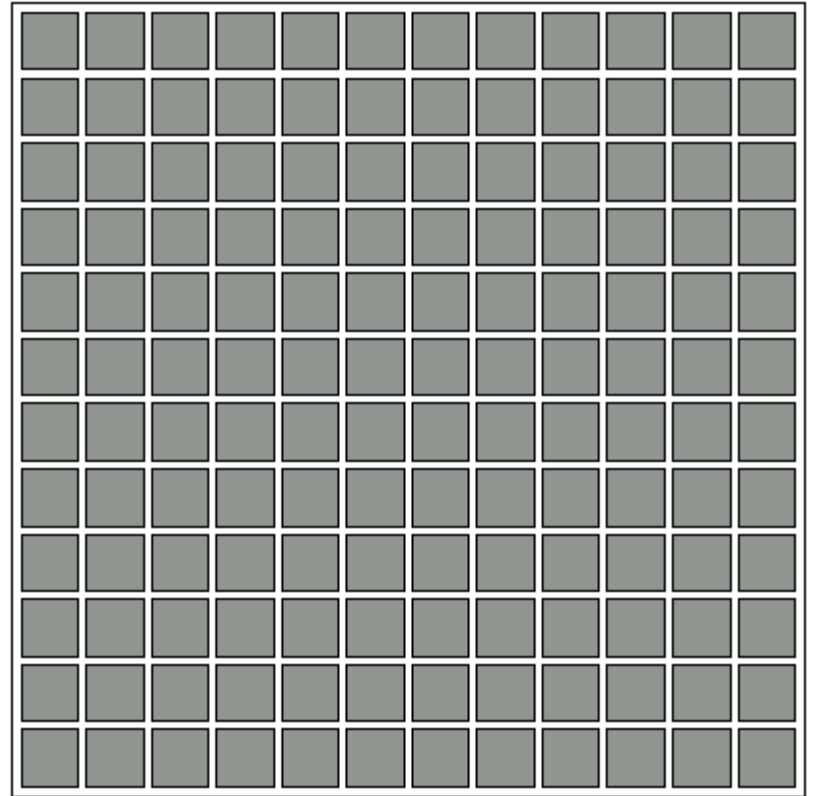
### ◆ 阵列成像传感器

#### CCD

英文全称：Charge-coupled Device

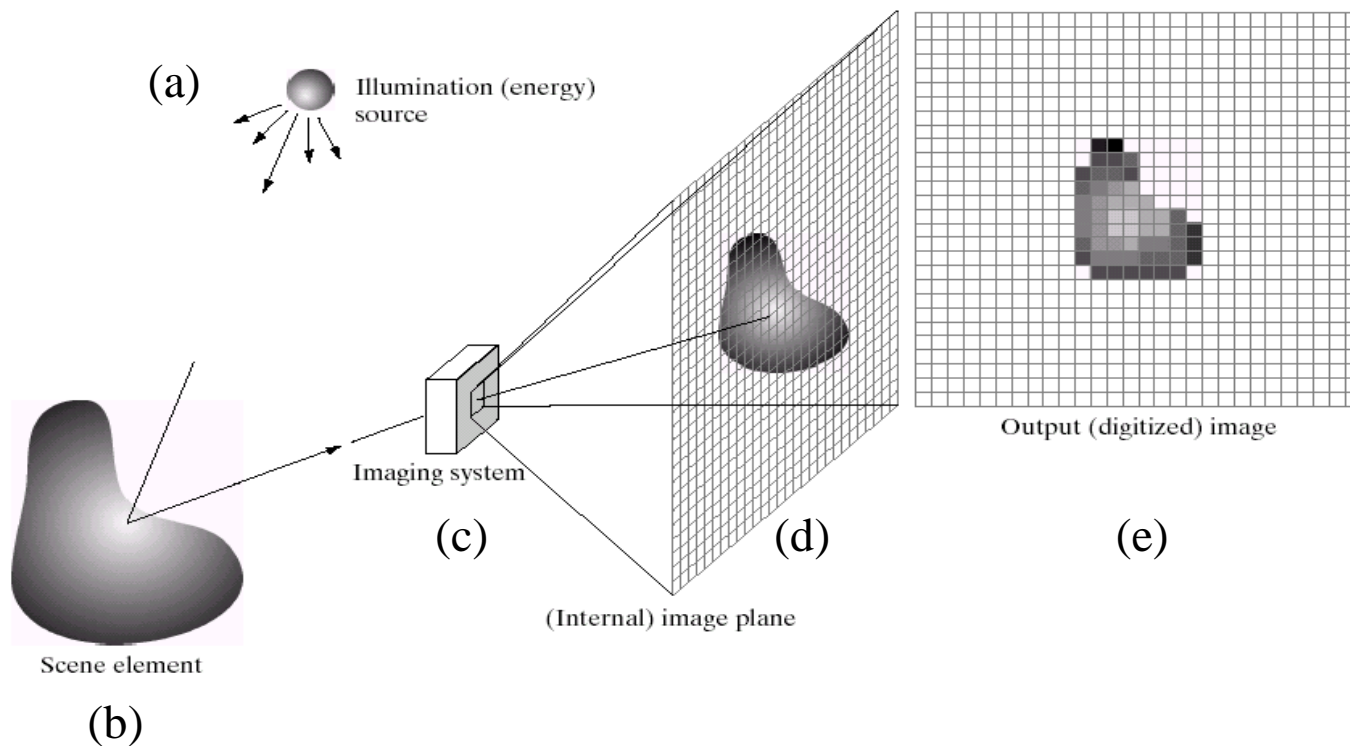
中文全称：电荷耦合元件。

不需要扫描，就能形成二维图像。



## 2.3 图像感知和获取

### 数字图像获取过程的一个例子



(a) 能量源；(b) 场景元素；(c) 成像系统；(d) 场景到图像平面的投影；(e) 数字化后的图像



## 2.3 图像感知和获取

### 简单的图像形成模型

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

其中:

$f(x, y)$ 是图像幅度分布

$i(x, y)$ 是入射场分布, 取决于照射源特性

$r(x, y)$ 是反射系数分布, 取决于成像物体的特性

$$0 < i(x, y) < \infty$$

$$0 < r(x, y) < 1$$



# 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

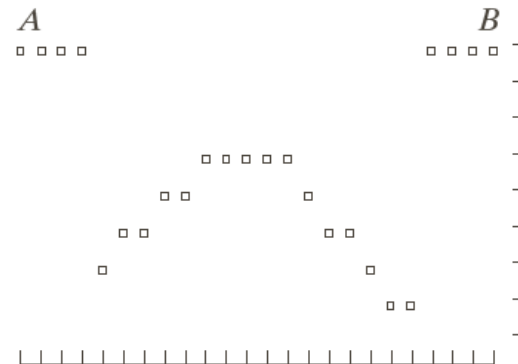
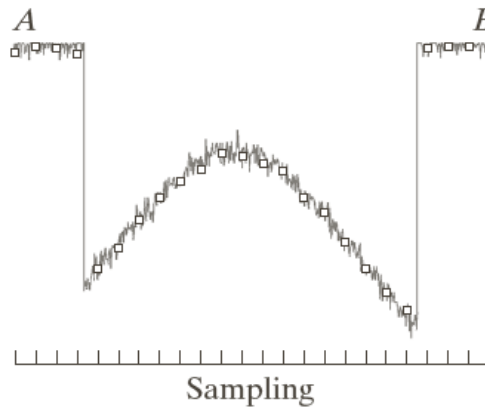
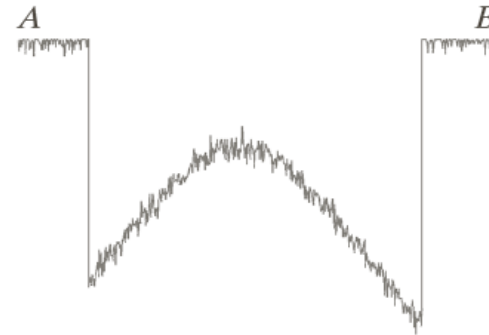
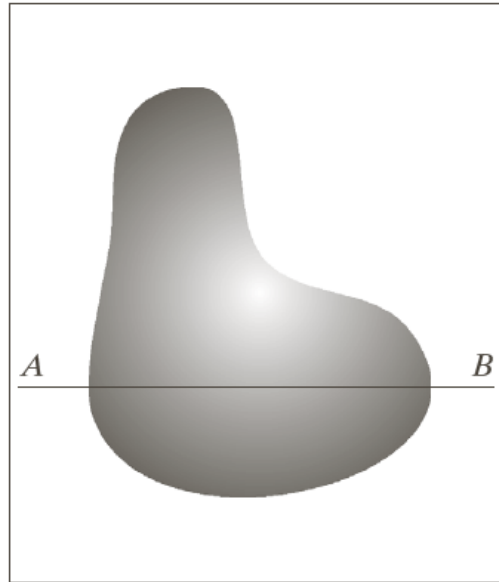


## 2.4 图像取样和量化

### ◆ 基本概念

1. **图像数字化**：将模拟图像经过离散化之后，得到用数字表示的图像。图像的数字化的包括取样和量化两个过程。
2. **取样**：是将在空间上连续的图像转换成离散的采样点（即像素）集的操作。即：**空间坐标**的离散化。
3. **量化**：把取样后所得的各像素的灰度值从模拟量到离散量的转换称为图像灰度的量化。即：**灰度**的离散化。

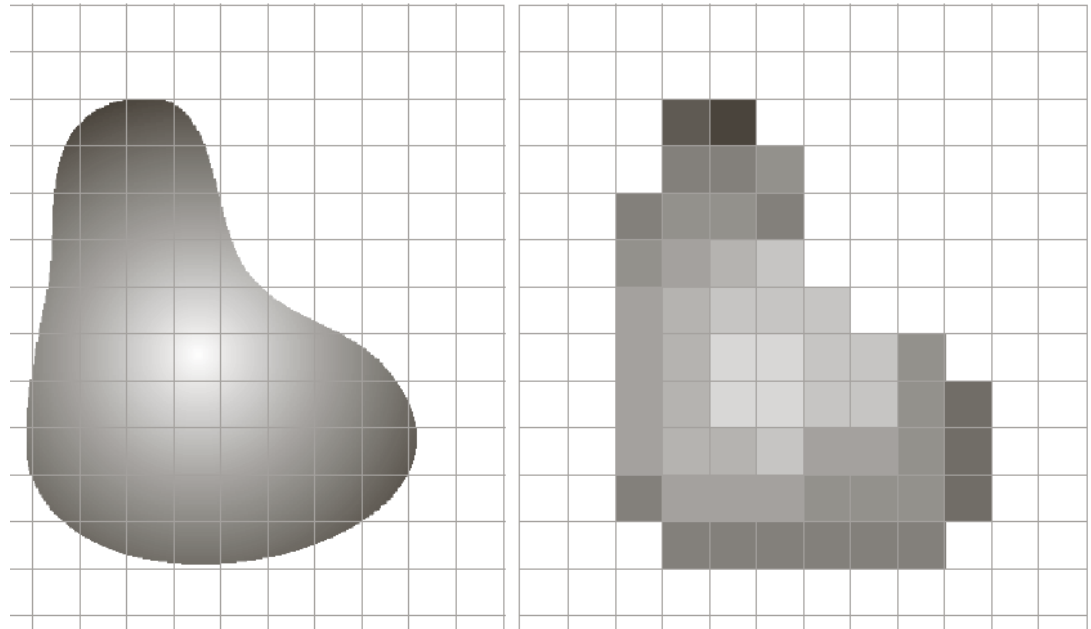
# 2.4 图像取样和量化





## 2.4 图像取样和量化

数字图像的性质在很大程度上取决于取样和量化中所用的**样本数**和**灰度级**。



(a)

(b)

(a) 连续图像； (b) 取样和量化后的结果



## 2.4 图像取样和量化

### ◆ 数字图像表示

#### 1. 矩阵表示

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0, N-1) \\ f(1,0) & f(1,1) & \cdots & f(1, N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1, N-1) \end{bmatrix}$$

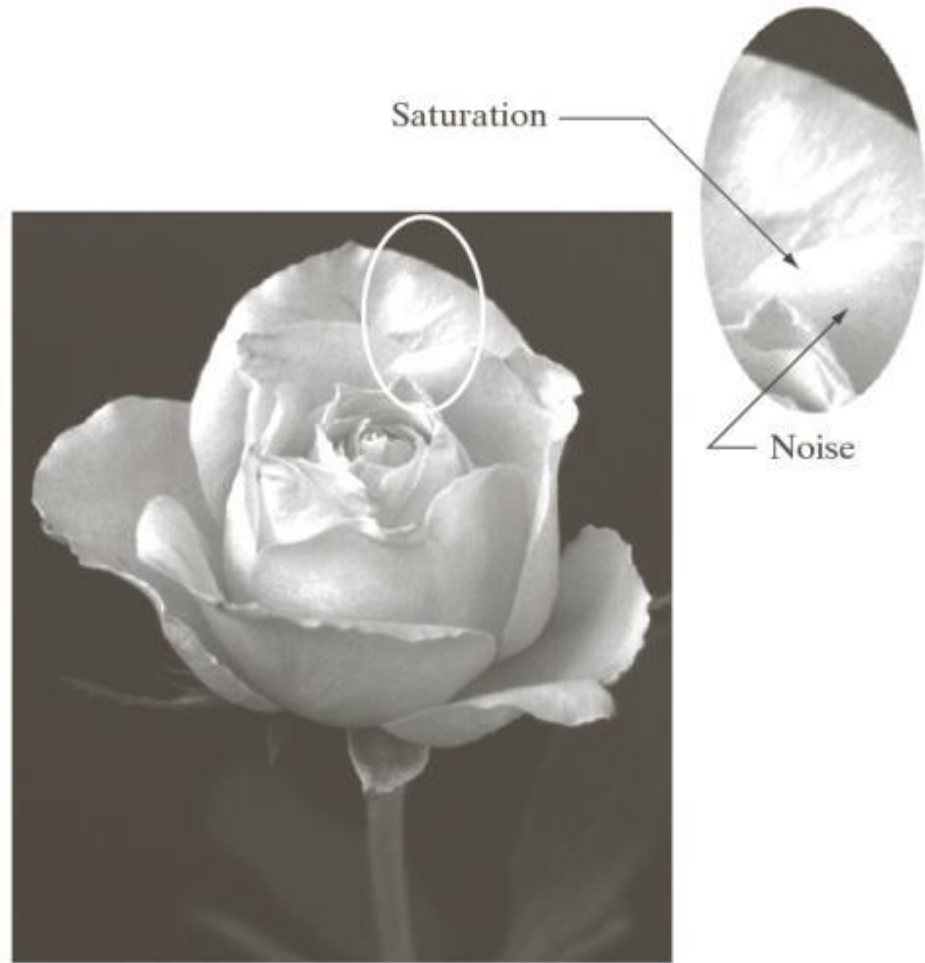
#### 2. 传统表示形式

$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix}$$



## 2.4 图像取样和量化

饱和度指的是一个最大值。超过这个值的灰度级将被剪切掉。





## 2.4 图像取样和量化

储存数字图像所需的比特数**b**为：

$$b = M \times N \times k$$

当  $M = N$  时，该式变为：

$$b = N^2 k$$

$N/k$	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

**N**和**k**取不同值时存储所需的比特数



## 2.4 图像取样和量化

### ◆ 空间和灰度分辨率

- **空间分辨率：** 图像空间中可分辨的最小细节。一般用单位长度上采样的像素数目或单位长度上的线对数目表示。例如，举一个质量概念，报纸用75dpi的分辨率来印刷，杂志是133dpi，光鲜的小册子是175dpi，您正在看的书是以2044dpi印刷的。**空间分辨率的度量必须针对空间单位来规定才有意义。**





## 2.4 图像取样和量化

---

### □ 空间分辨率与图像质量的关系:

空间分辨率越高，图像质量越好；空间分辨率越低，图像质量越差，会出现棋盘模式。

## 2.4 图像取样和量化

图像分别以  
250dpi, 300dpi,  
150dpi, 72dpi  
来显示。





## 2.4 图像取样和量化

- **灰度分辨率**：图像灰度级中可分辨的最小变化。一般用灰度级或比特数表示。灰度级数通常是2的整数次幂。最通用的是8比特。通常说一幅被量化为256级的图像有8比特的灰度分辨率。



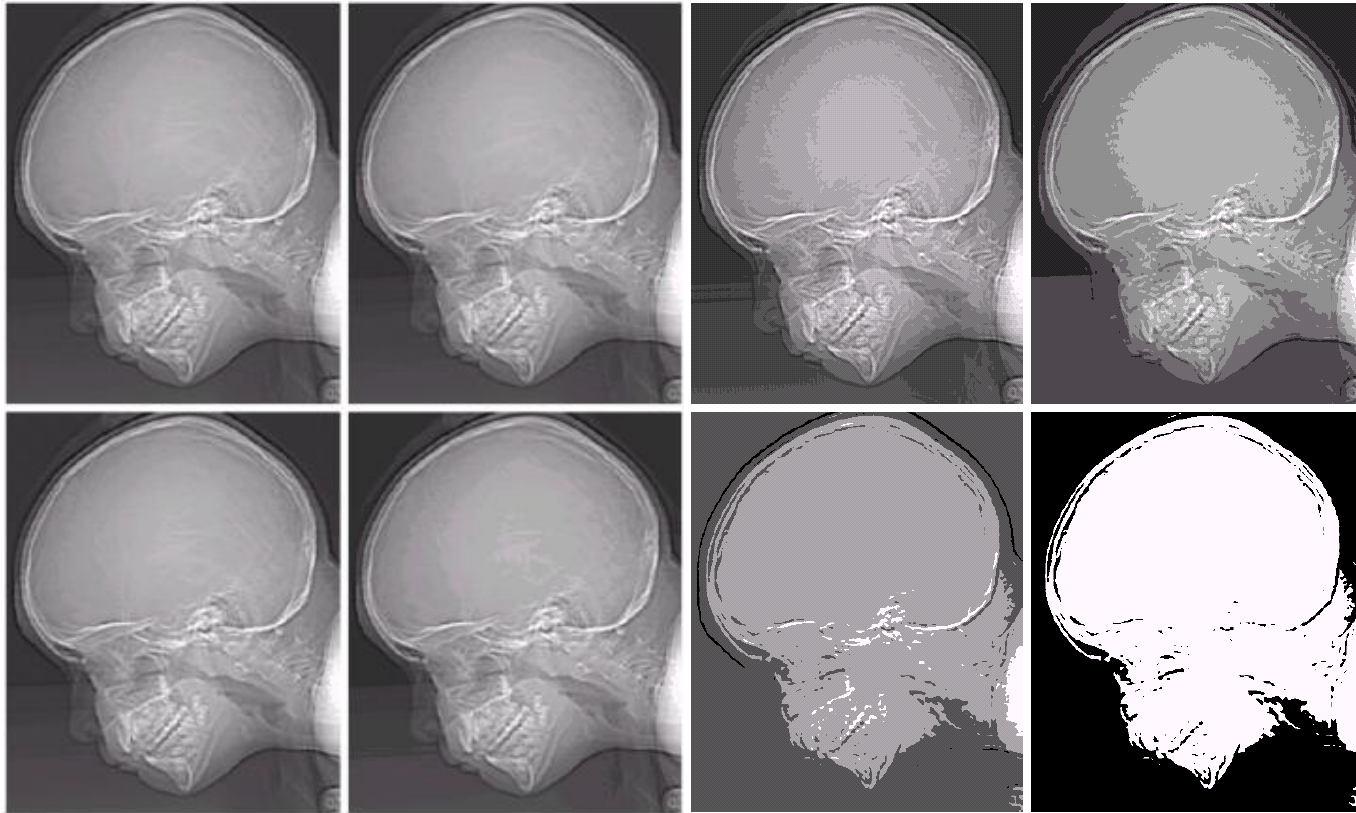
## 2.4 图像取样和量化

---

### □ 灰度分辨率与图像质量的关系:

灰度分辨率越高，图像质量越好；灰度分辨率越低，图像质量越差，会出现虚假轮廓。

## 2.4 图像取样和量化



图像大小为 $452 \times 374$ ，分别以256, 128, 64, 32, 16, 8, 4, 2的灰度级递减

## 2.4 图像取样和量化



256灰度级



16灰度级



8灰度级



4灰度级



## 2.4 图像取样和量化

- 空间分辨率和灰度分辨率同时变化，对图像质量的影响：  
图像细节含量由少到多，同时改变空间分辨率和量化级，  
要求观察者按照主观质量进行排序。



(a) 少量细节图像



(b) 中等程度细节图像



(c) 大量细节图像



## 2.4 图像取样和量化

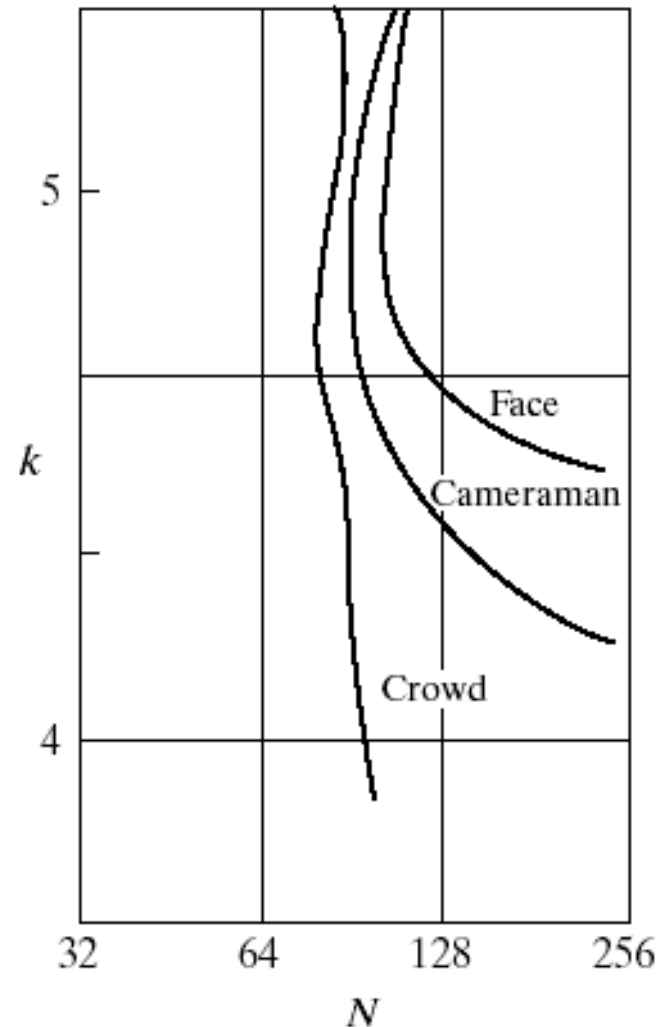
### 等偏爱曲线

(Isopreference curve):

在 $N$ - $k$ 平面内对应于主观感觉质量相等的曲线;

**实验结果:**

当图像中细节增加时, 等偏爱曲线趋于更接近 $N$ - $k$ 平面内的垂直线。





## 2.4 图像取样和量化

### □ 总结:

- A. 图像质量一般随 $N$ 和 $k$ 的增加而增加。在极少数情况下对固定的 $N$ ，减小 $k$ 能改进质量。最有可能的原因是减小 $k$ 常能可增加图像的视觉反差。
- B. 对具有大量细节的图像只需很少的灰度级数就可较好地表示。
- C.  $N \times k$ 为常数的图像主观看起来可以有较大的差异。



## 2.4 图像取样和量化

### □ 图像内插

- 内插是用已知数据来估计未知位置的数值的处理。

➤ 最邻近内插

➤ 双线性内插

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

➤ 双三次内插

$$v(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{xy} x^i y^j$$



# 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**

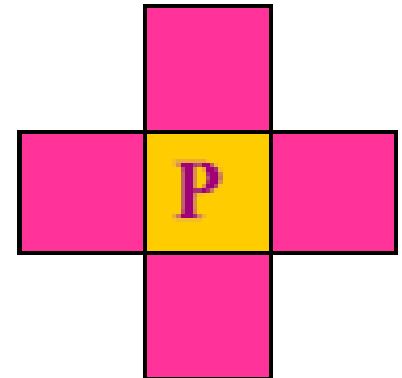


## 2.5 像素间的基本关系

### □ 相邻像素

#### ◆ 4邻域

坐标为  $(x,y)$  的像素 $p$ 有4个水平和垂直的相邻像素，坐标分别为:  $(x+1, y)$ ,  $(x-1, y)$ ,  $(x, y+1)$ ,  $(x, y-1)$ ，这个像素集称为 $p$ 的4邻域，用 $N_4(p)$ 表示。



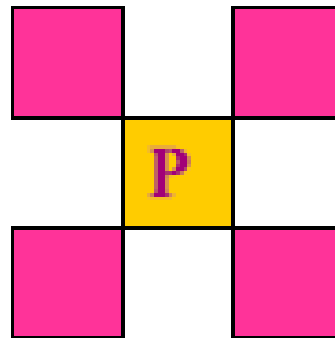
## 2.5 像素间的基本关系

### ◆ 对角邻域

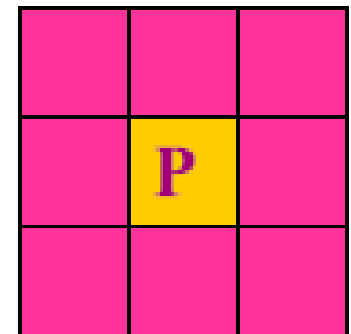
坐标分别为:  $(x+1, y+1)$ ,  $(x-1, y-1)$ ,  $(x-1, y+1)$ ,  $(x+1, y-1)$ ,  
用  $N_D(p)$  表示。

### ◆ 8邻域

$N_4(p) \cup N_D(p)$



(a) 对角邻域



(b) 8邻域

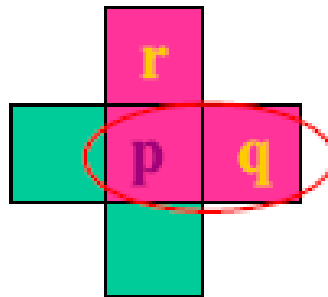
## 2.5 像素间的基本关系

### □ 邻接性、连通性、区域和边界

- 邻接性

令 $V$ 是具有特定**相似性准则**的灰度值集合。

1) 4邻接: 如果 $q$ 在 $N_4(p)$ 中, 具有 $V$ 中数值的两个像素 $p$ 和 $q$ 是4邻接的。

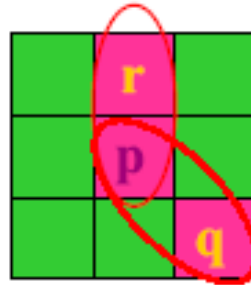






## 2.5 像素间的基本关系

2) 8邻接: 如果 $q$ 在 $N_8(p)$ 中, 具有 $V$ 中数值的两个像素 $p$ 和 $q$ 是8邻接的。





## 2.5 像素间的基本关系

### 3) m邻接（混合邻接）：

如果 $q$ 在 $N_4(p)$ 中，或者 $q$ 在 $N_D(p)$ 中且集合 $N_4(p) \cap N_4(q)$ 没有 $V$ 值的像素，则具有 $V$ 值的像素 $p$ 和 $q$ 是 $m$ 邻接的。

$m$ 邻接的引入是为了消除采用8邻接常常发生的二义性。

满足下面两个条件之一即可：

1.  $q$ 在 $p$ 的4邻域中；
2.  $q$ 在 $p$ 的对角领域中，并且 $q$ 的4邻域与 $p$ 的4邻域相交为空集（交集无点属于 $V$ ）。

0	1	1
0	1	0
0	0	1

0	1	1
0	1	0
0	0	1



## 2.5 像素间的基本关系

### □ 连通性

- ◆ 从具有坐标 $(x, y)$ 的像素 $p$ 到具有坐标 $(s, t)$ 的像素 $q$ 的一条通路由一系列具有坐标 $(x_0, y_0)$ ,  $(x_1, y_1)$ , ...,  $(x_n, y_n)$ 的独立像素组成。这里 $(x_0, y_0) = (x, y)$ ,  $(x_n, y_n) = (s, t)$ , 且 $(x_i, y_i)$ 与 $(x_{i-1}, y_{i-1})$ 邻接, 其中 $1 \leq i \leq n$ ,  $n$ 为通路长度。
- ◆ 邻接是连通的一种特例, 连通是由一系列依次邻接的像素组成。



## 2.5 像素间的基本关系

---

### □ 区域

- ◆ 连通分量:  $S$ 是图像中的一个子集, 对 $S$ 中任何一个像素 $p$ , 连通到该像素的像素集称为 $S$ 的连通分量。
- ◆ 如果 $S$ 仅有一个连通分量, 则称为连通集, 也称作一个区域。



## 2.5 像素间的基本关系

---

### □ 边界

- ◆ 邻接区域：两个区域联合起来形成一个连通集。
- ◆ 不连接区域：非邻接区域。
- ◆ 背景：一幅图像的 $K$ 个不连接区域的并集的补集。
- ◆ 边界：一个区域的边界，是该区域中至少有一个背景邻点的像素集。



## 2.5 像素间的基本关系

### □ 距离度量

#### ◆ 距离度量函数的定义

- 给定3个像素 $\mathbf{x}$ ,  $\mathbf{y}$ ,  $\mathbf{z}$ , 坐标分别为 $(x_1, x_2)$ ,  $(y_1, y_2)$ ,  $(z_1, z_2)$ , 如果下列条件满足, 则  $D$  是距离度量函数:

①  $D(\mathbf{x}, \mathbf{y}) \geq 0$  ( $D(\mathbf{x}, \mathbf{y}) = 0$ , 当且仅当  $\mathbf{x} = \mathbf{y}$ )

②  $D(\mathbf{x}, \mathbf{y}) = D(\mathbf{y}, \mathbf{x})$

③  $D(\mathbf{x}, \mathbf{z}) \leq D(\mathbf{x}, \mathbf{y}) + D(\mathbf{y}, \mathbf{z})$

#### ◆ $L_p$ 范数 (norm) 及其诱导出的距离度量: 对于 $k$ 维向量 $\mathbf{u} = [u_1, u_2, \dots, u_k]^T$

- $L_p$  范数:  $\|\mathbf{u}\|_p = (\sum_{i=1}^k |u_i|^p)^{\frac{1}{p}}$

- $L_p$  范数诱导的距离度量:  $D(\mathbf{u}, \mathbf{v}) = \|\mathbf{u} - \mathbf{v}\|_p = (\sum_{i=1}^k |u_i - v_i|^p)^{\frac{1}{p}}$



## 2.5 像素间的基本关系

□ 欧氏(Euclidean)距离：模为2的距离

- ◆ 根据这个距离量度，与 $(x_1, x_2)$ 的距离小于或等于某个值 $d$ 的像素都包括在以 $(x_1, x_2)$ 为中心以 $d$ 为半径的圆。

$$D_e(\mathbf{x}, \mathbf{y}) = [(x_1 - y_1)^2 + (x_2 - y_2)^2]^{\frac{1}{2}}$$

- ◆ 欧式距离等价于 $L_2$ 距离（以 $L_p$ 范数（ $p = 2$ ）诱导的距离度量）



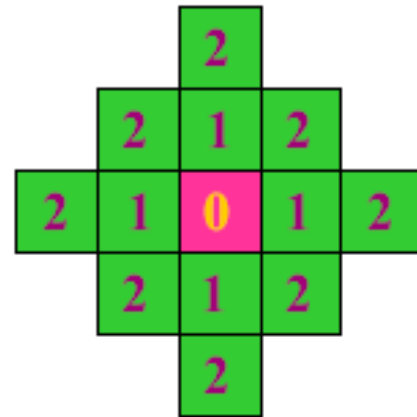
## 2.5 像素间的基本关系

□ 城区(city-block)距离：模为1的距离或 $D_4$ 距离

◆ 根据这个距离量度，与 $(x_1, x_2)$ 的距离小于或等于某个值 $d$ 的像素组成以 $(x_1, x_2)$ 为中心的菱形。

$$D_4(\mathbf{x}, \mathbf{y}) = |x_1 - y_1| + |x_2 - y_2|$$

◆ 城区(city-block)距离等价于 $L_1$ 距离（以 $L_p$ 范数（ $p = 1$ ）诱导的距离度量）





## 2.5 像素间的基本关系

□ 棋盘(chessboard)距离：模为 $\infty$ 的距离或 $D_8$ 距离

◆ 根据这个距离量度，与 $(x_1, x_2)$ 的距离小于或等于某个值 $d$ 的象素组成以 $(x_1, x_2)$ 为中心的正方形。

$$D_8(\mathbf{x}, \mathbf{y}) = \max(|x_1 - y_1|, |x_2 - y_2|)$$

◆ 棋盘(chessboard)距离等价于 $L_\infty$ 距离（以 $L_p$ 范数  
( $p = \infty$ )诱导的距离度量)

✓ (请证明该结论)

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2



# 数字图象处理

---

中国科学技术大学  
电子工程与信息科学系

主讲教师：李厚强 ([lihq@ustc.edu.cn](mailto:lihq@ustc.edu.cn))  
周文罡 ([zhwg@ustc.edu.cn](mailto:zhwg@ustc.edu.cn))

助教：谢乔康 ([xieqiaok@mail.ustc.edu.cn](mailto:xieqiaok@mail.ustc.edu.cn))  
周 浩 ([zhouh156@mail.ustc.edu.cn](mailto:zhouh156@mail.ustc.edu.cn))



# 第3章 灰度变换与空域滤波

## 3.1 背景知识

## 3.2 基本灰度变换

## 3.3 直方图处理

## 3.4 空间滤波基础

## 3.5 空域平滑

## 3.6 空域锐化

# 灰度变换

- 空间域：包含图像像素的简单平面
- 空间域灰度变换

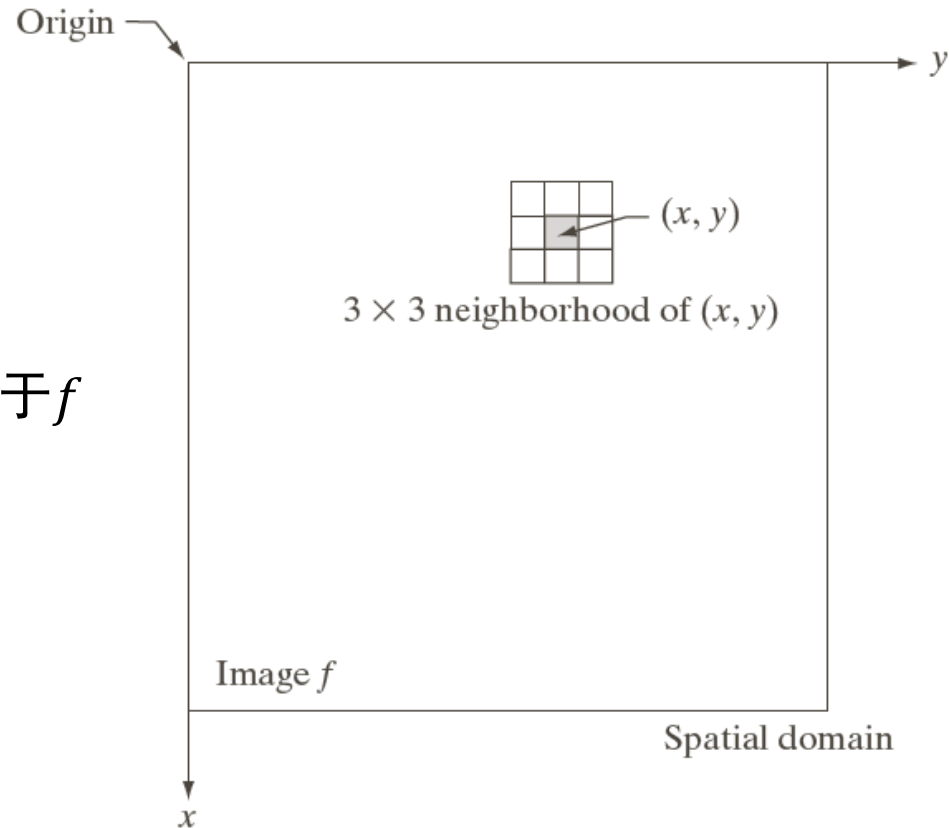
$$g(x, y) = T(f(x, y))$$

$f(x, y)$ : 输入图像

$g(x, y)$ : 输出图像

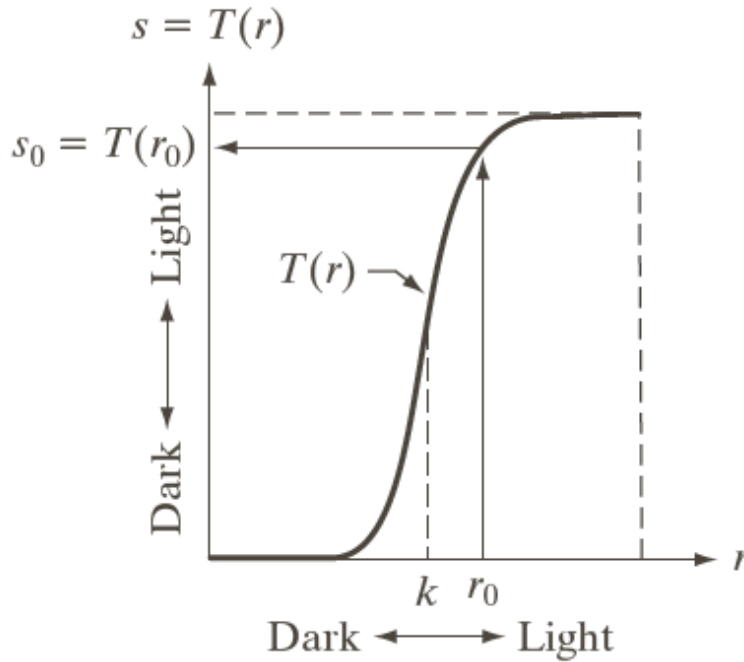
$T$ : 在点 $(x, y)$ 的邻域上定义的关于 $f$ 的一种算子

- 空间滤波邻域
  - 决定操作的图像局部范围
  - 最小邻域： $1 \times 1$

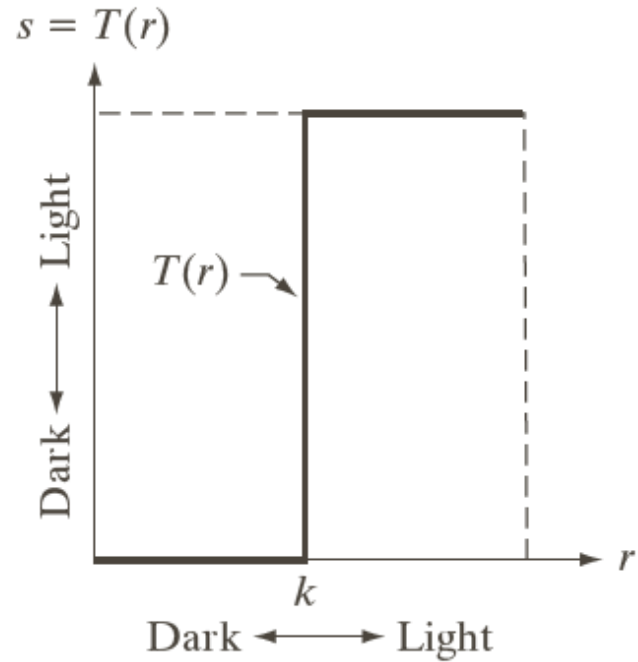


# 灰度变换

## □ 邻域为 $1 \times 1$ 时的灰度变换



对比度拉伸函数



阈值处理函数



# 第3章 灰度变换与空域滤波

---

3.1 背景知识

3.2 基本灰度变换

3.3 直方图处理

3.4 空间滤波基础

3.5 空域平滑

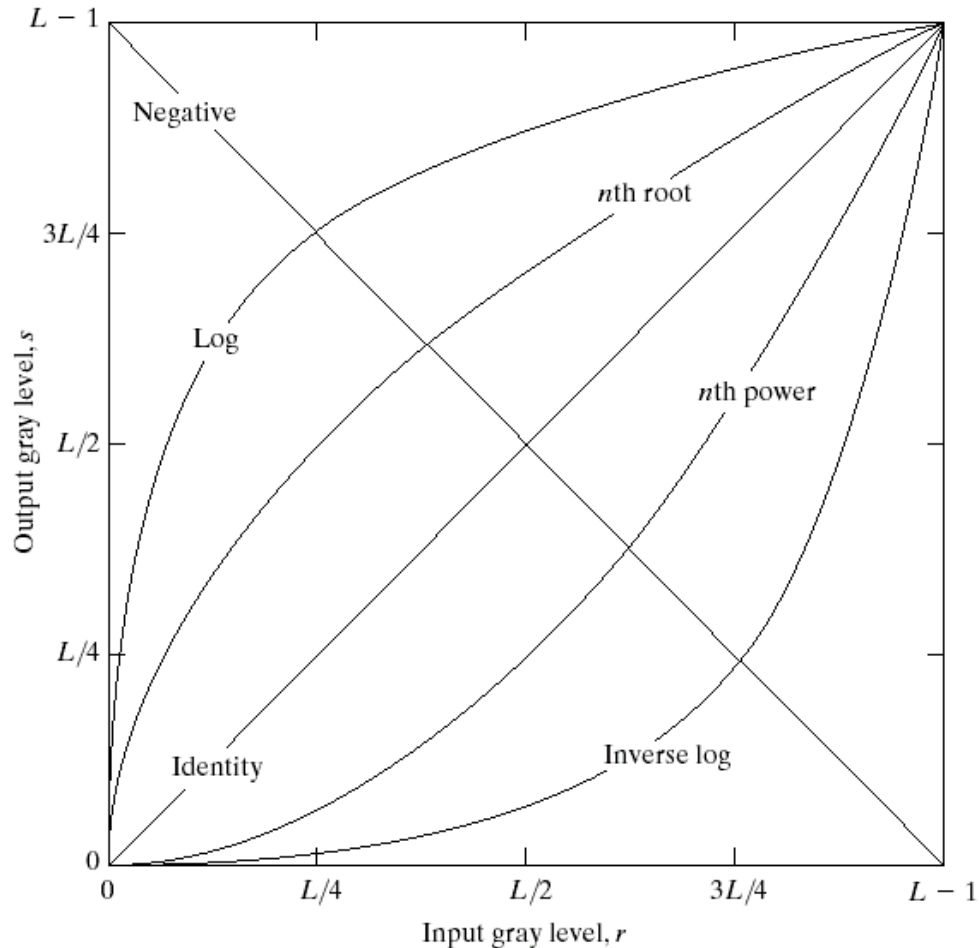
3.6 空域锐化



# 常见灰度变换

□ 灰度变换  $s = T(r)$

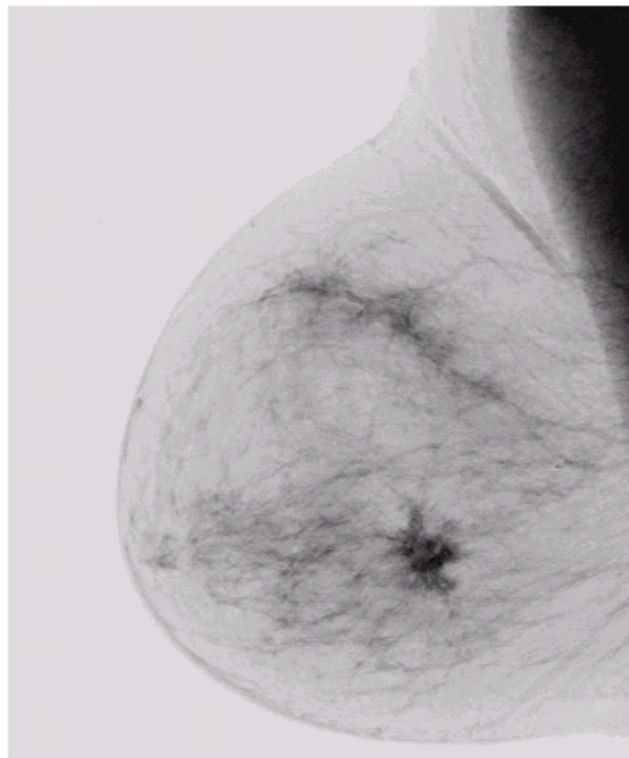
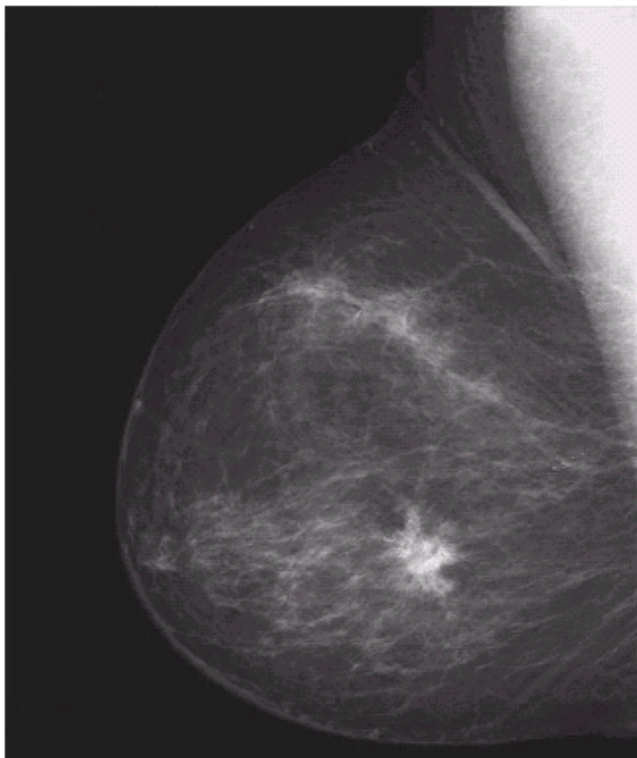
■  $r$ 为数字量且取值有限，该映射可通过查找表实现



# 图象求反

□ 灰度变换函数： $s = L - 1 - r$

- 适用于增强嵌入图像暗色区域中的白色或灰色细节



a b

## FIGURE 3.4

(a) Original digital mammogram.  
(b) Negative image obtained using the negative transformation in Eq. (3.2-1).  
(Courtesy of G.E. Medical Systems.)

# 对数变换

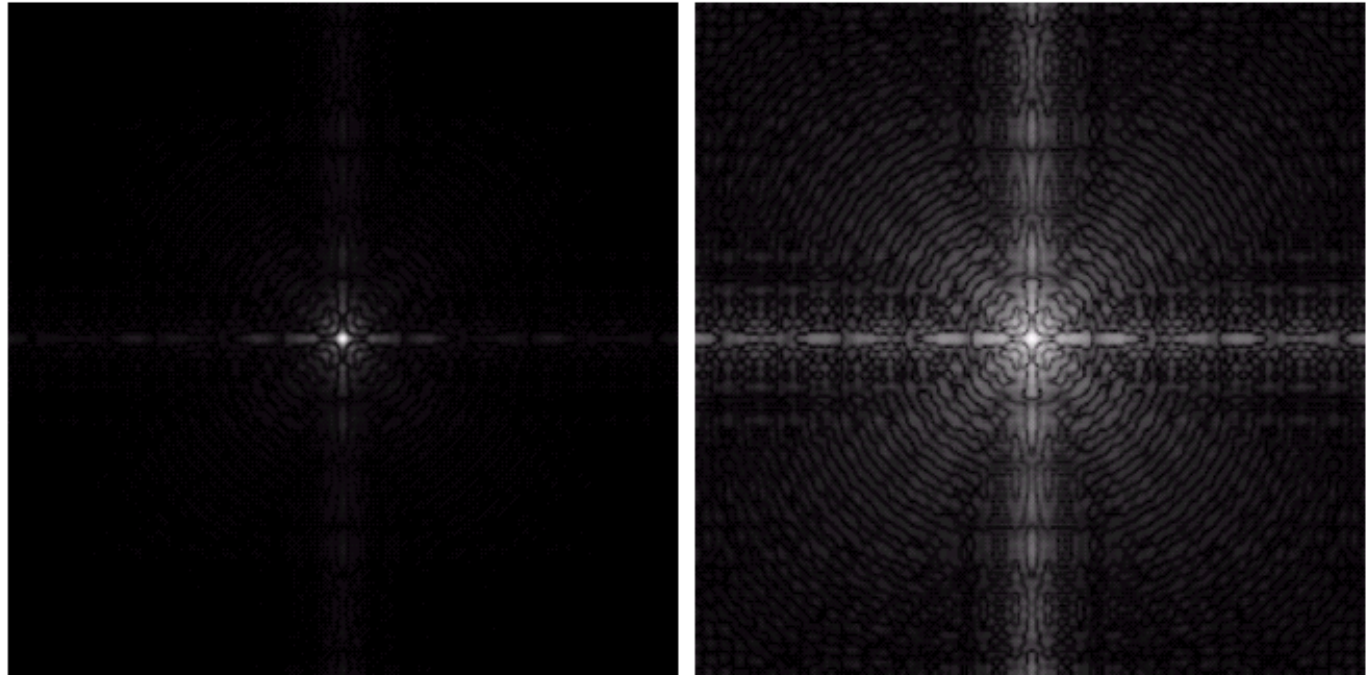
□ 灰度变换函数： $s = c \cdot \log(1 + r)$

a b

## FIGURE 3.5

(a) Fourier spectrum.

(b) Result of applying the log transformation given in Eq. (3.2-2) with  $c = 1$ .

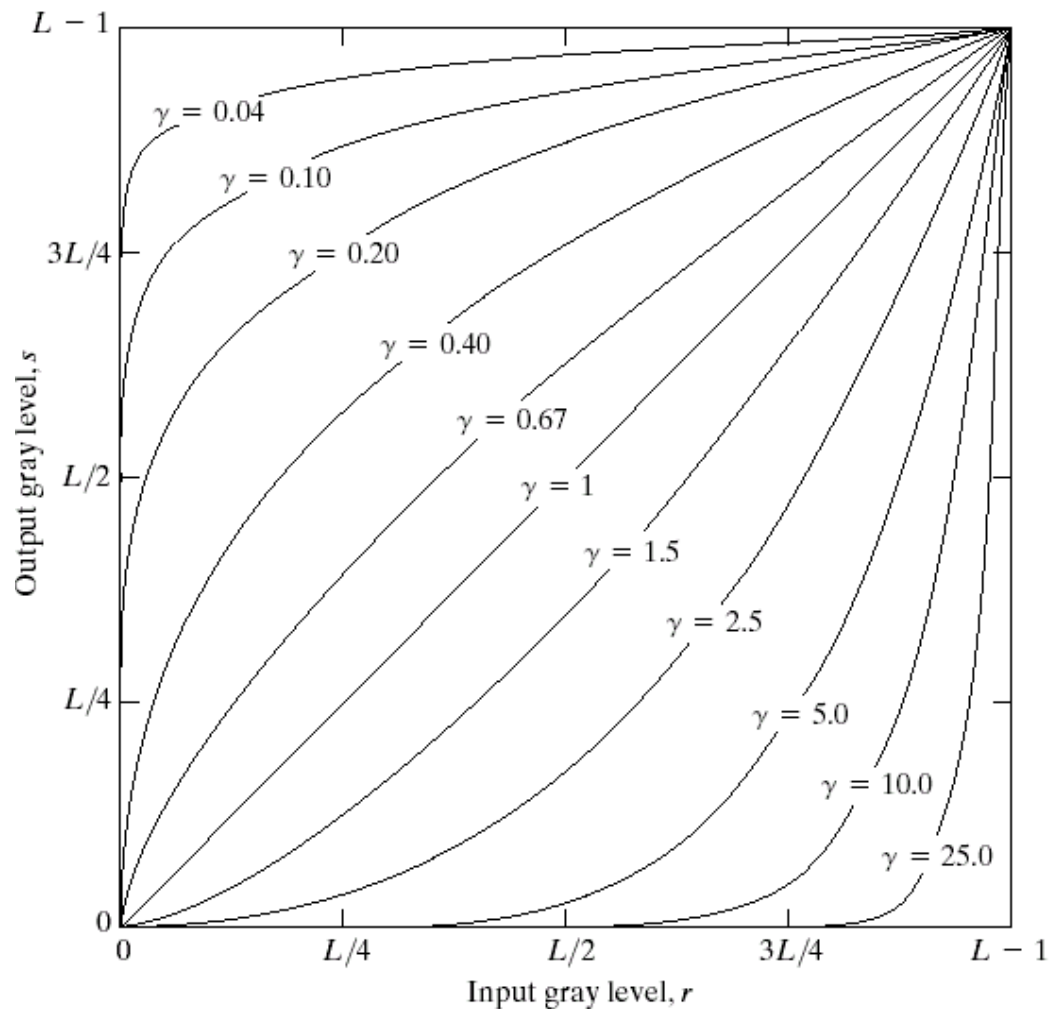


傅里叶频谱：

$0 \sim 1.5 \times 10^6$

# 常见灰度变换—指数变换

灰度变换函数： $s = c \cdot r^\gamma$



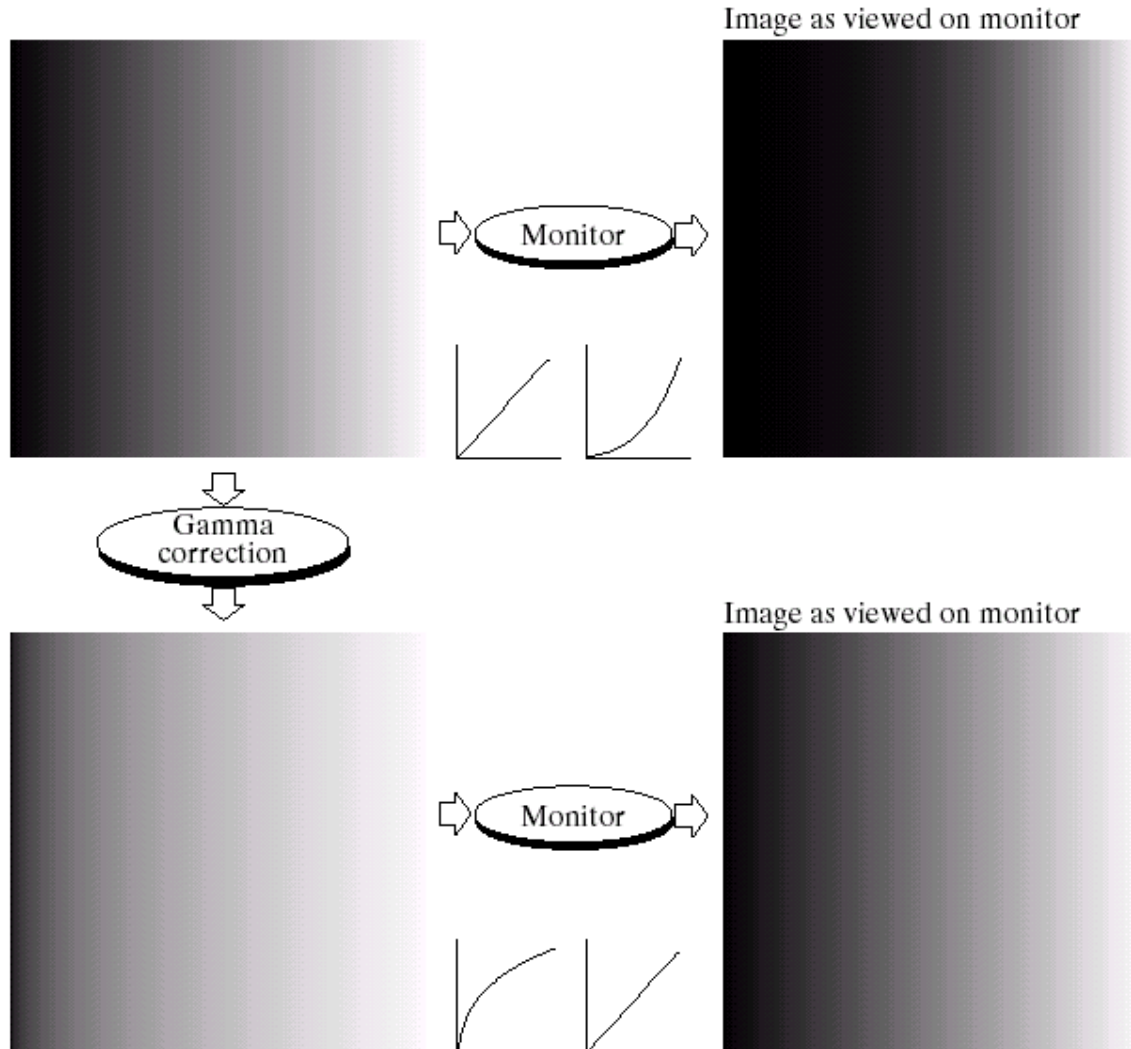
**FIGURE 3.6** Plots of the equation  $s = cr^\gamma$  for various values of  $\gamma$  ( $c = 1$  in all cases).

# 伽马校正

a	b
c	d

**FIGURE 3.7**

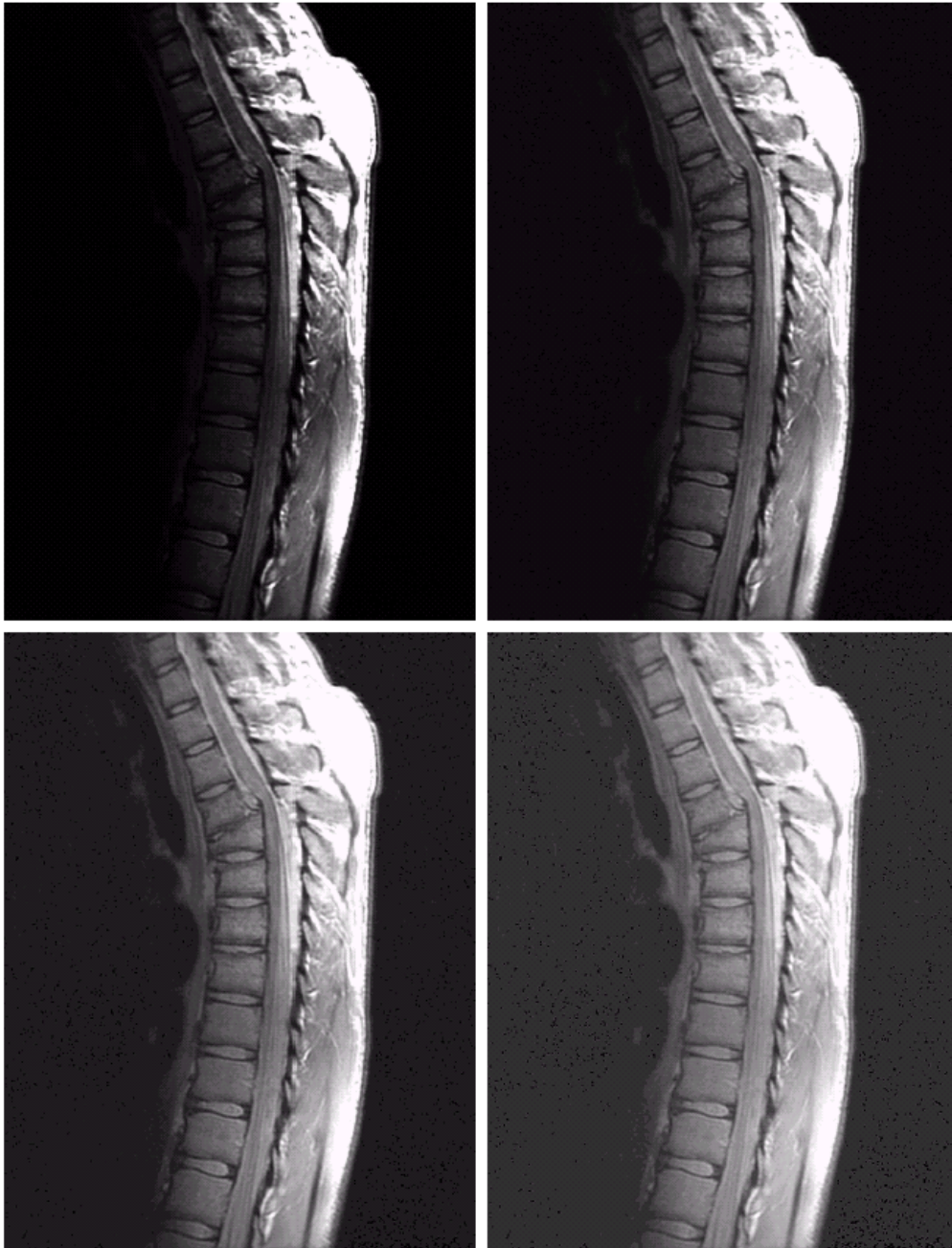
(a) Linear-wedge gray-scale image.  
 (b) Response of monitor to linear wedge.  
 (c) Gamma-corrected wedge.  
 (d) Output of monitor.



(b):  $\gamma=2.5$

(c):  $\gamma=1/2.5=0.4$





a	b
c	d

**FIGURE 3.8**

(a) Magnetic resonance (MR) image of a fractured human spine.  
(b)–(d) Results of applying the transformation in Eq. (3.2-3) with  $c = 1$  and  $\gamma = 0.6, 0.4,$  and  $0.3,$  respectively. (Original image for this example courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

a b  
c d

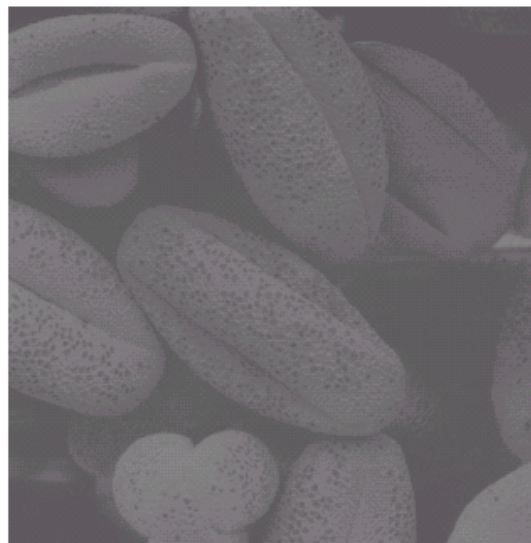
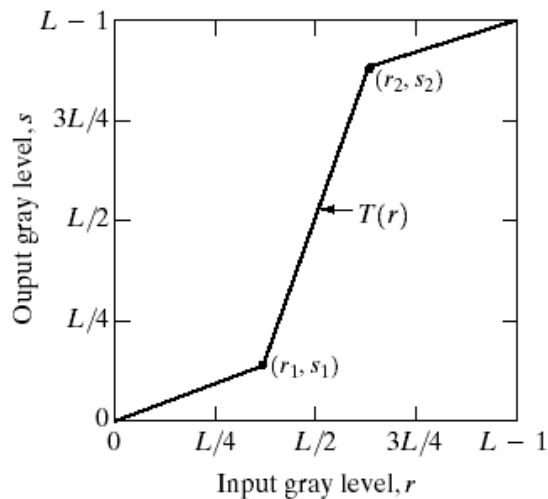
**FIGURE 3.9**

(a) Aerial image.  
(b)–(d) Results of applying the transformation in Eq. (3.2-3) with  $c = 1$  and  $\gamma = 3.0, 4.0,$  and  $5.0,$  respectively. (Original image for this example courtesy of NASA.)





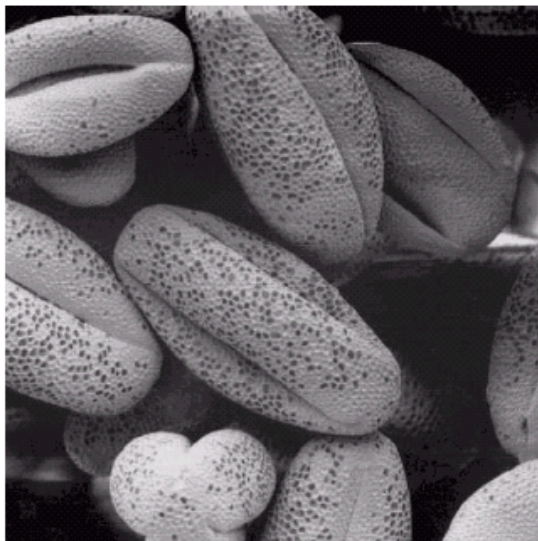
# 分段线性变换函数—对比度拉伸



a b  
c d

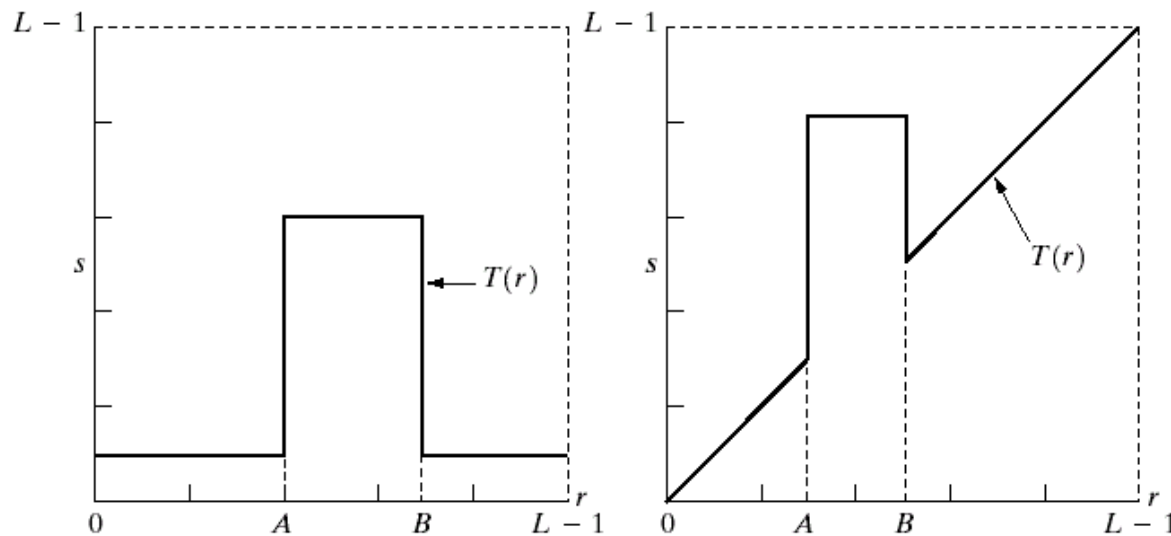
**FIGURE 3.10**

Contrast stretching.  
(a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)



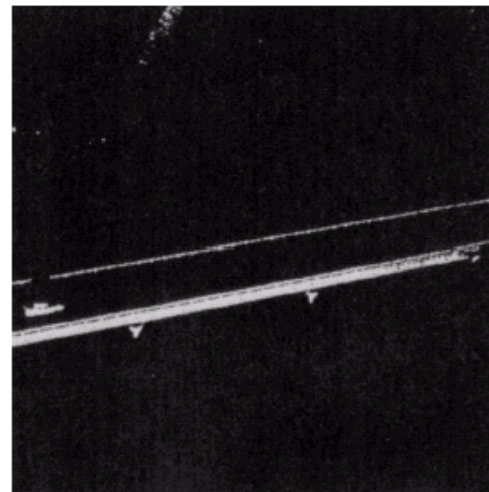
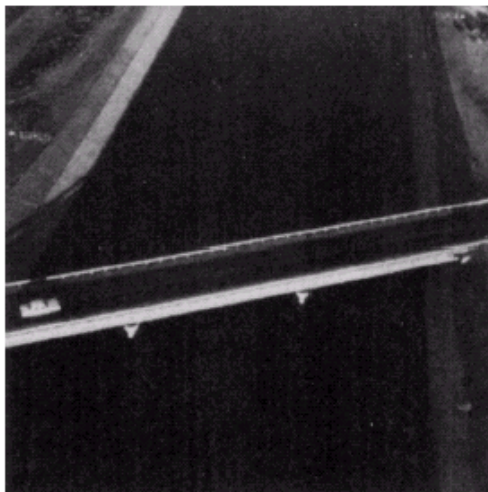


# 分段线性变换函数—灰度级切分

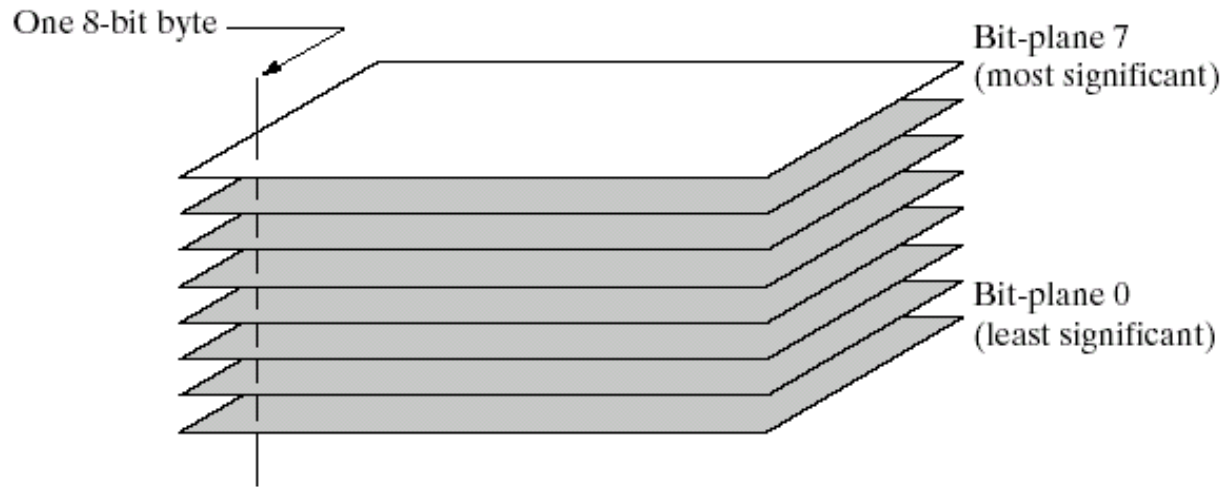


a	b
c	d

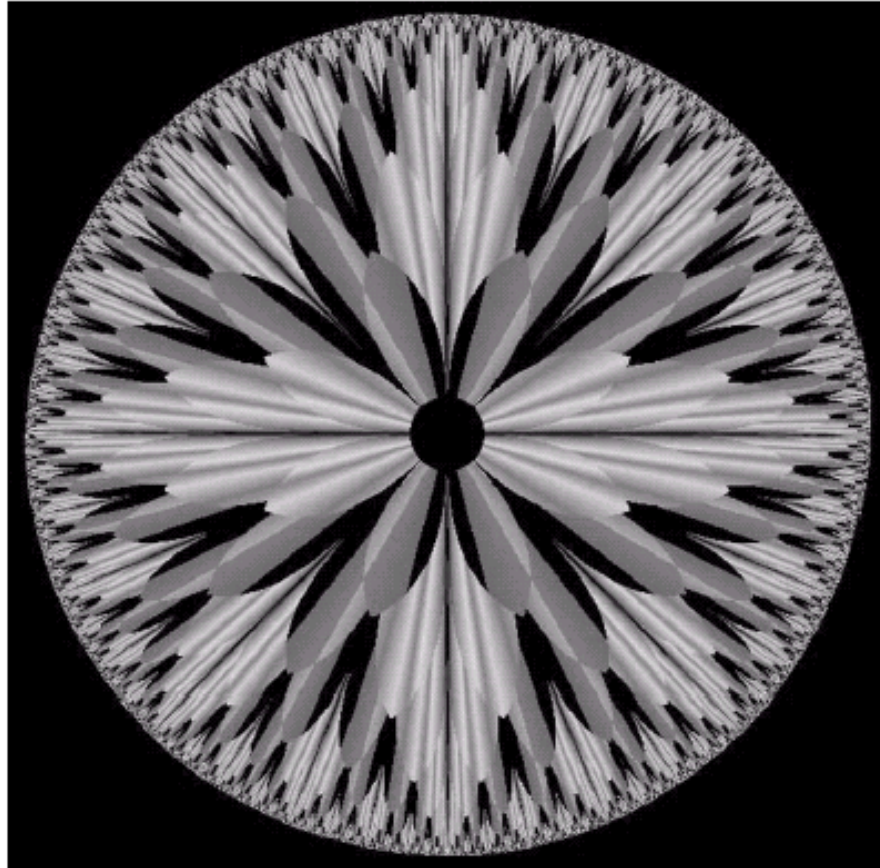
**FIGURE 3.11**  
 (a) This transformation highlights range  $[A, B]$  of gray levels and reduces all others to a constant level.  
 (b) This transformation highlights range  $[A, B]$  but preserves all other levels.  
 (c) An image.  
 (d) Result of using the transformation in (a).



# 比特面分割

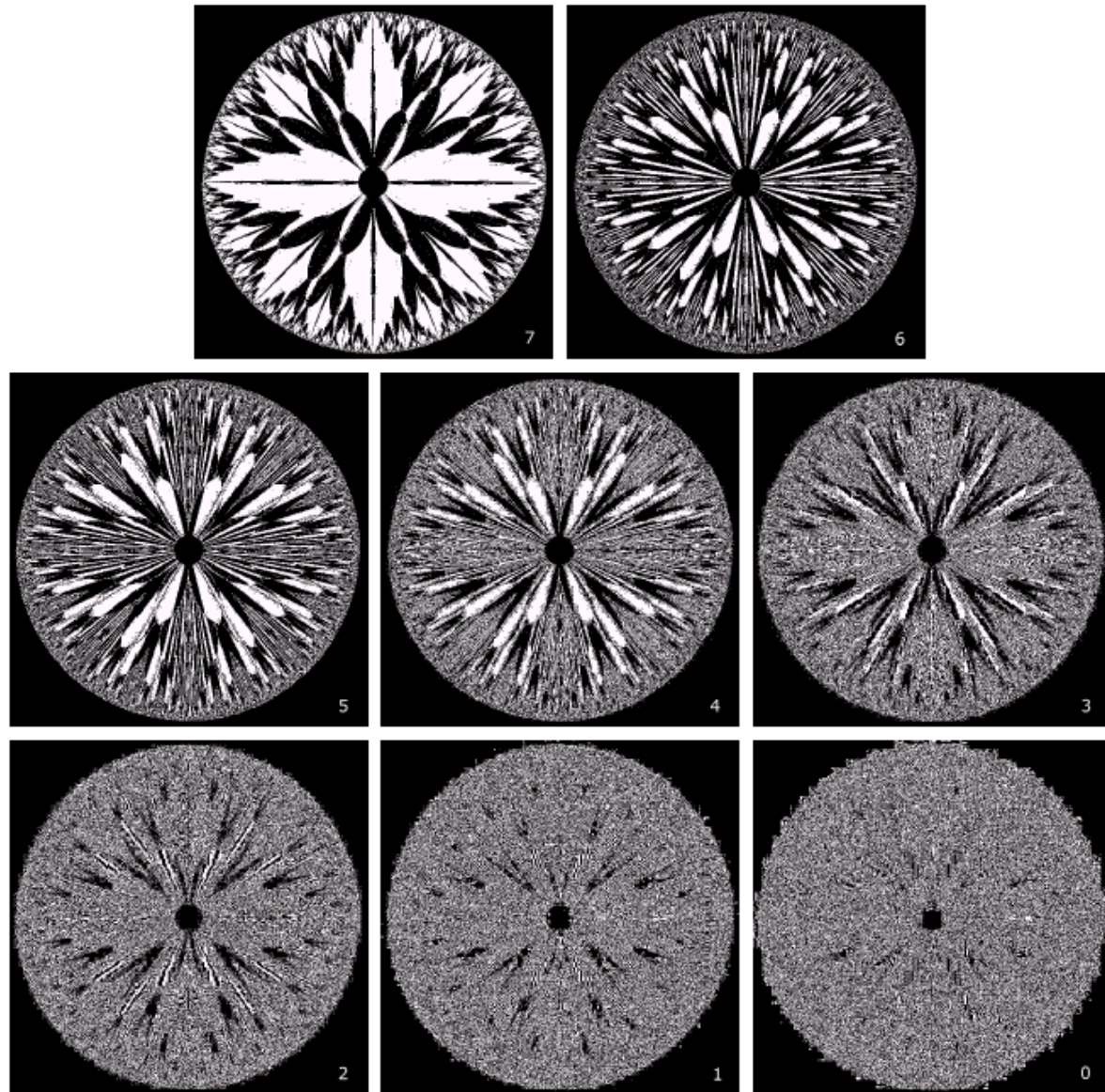


**FIGURE 3.12**  
Bit-plane  
representation of  
an 8-bit image.



**FIGURE 3.13** An 8-bit fractal image. (A fractal is an image generated from mathematical expressions). (Courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA.)





**FIGURE 3.14** The eight bit planes of the image in Fig. 3.13. The number at the bottom, right of each image identifies the bit plane.



# 第3章 灰度变换与空域滤波

---

- 3.1 背景知识
- 3.2 基本灰度变换
- 3.3 直方图处理
- 3.4 空间滤波基础
- 3.5 空域平滑
- 3.6 空域锐化



# 灰度直方图

## □ 灰度直方图

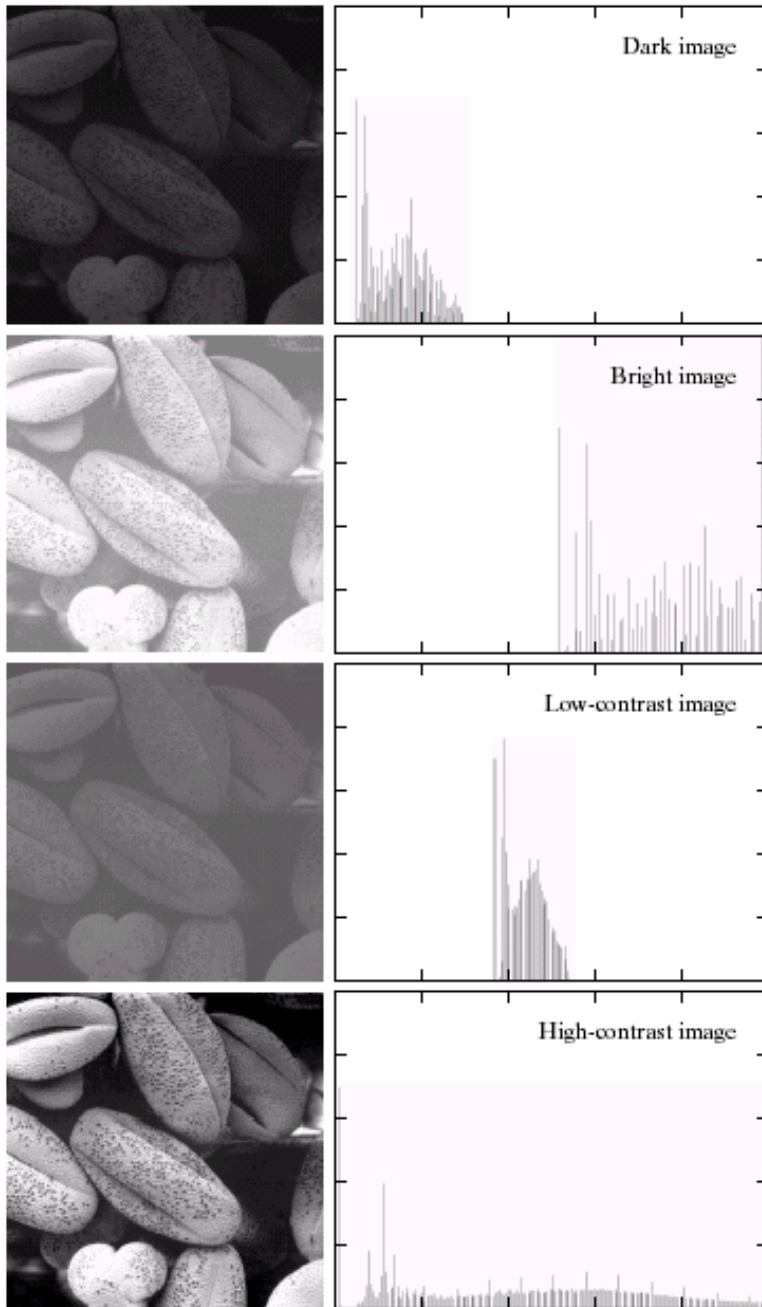
- 是一种灰度级的函数，它表示数字图象中每一灰度级与该灰度出现的频数之间的对应关系。
- 一幅特定的图有唯一的直方图，但反之不成立。
- 直方图在一定程度上可以反映图像的状况，例如：有时可以根据直方图确定分割物体和背景的背景。

## □ 基于直方图的灰度变换

- 不改变像素的位置，只改变像素的灰度值。

## □ 通过修改直方图可以达到增强图象的对比度、使感兴趣的部分看得更清楚。

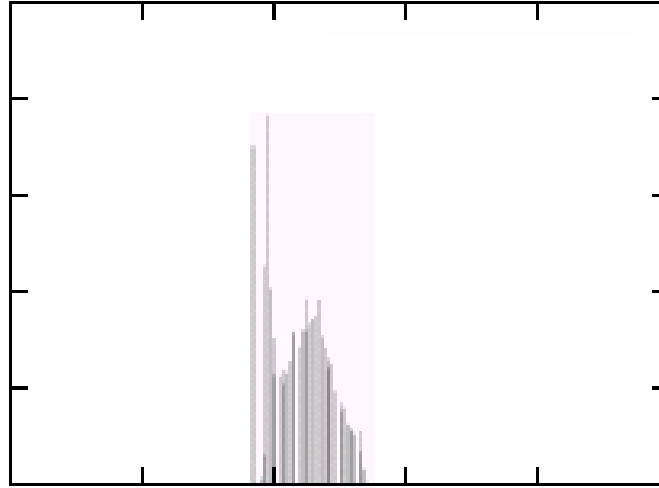
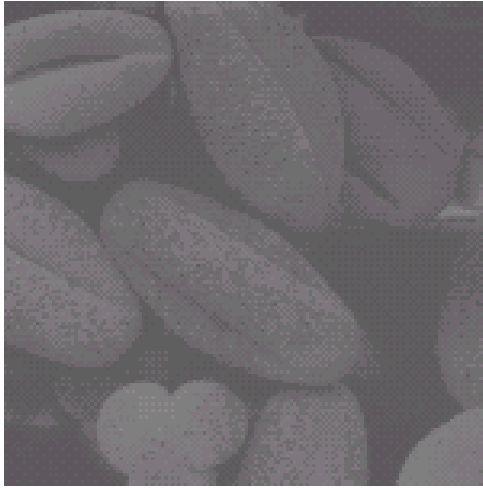
## □ 常用的有一维灰度特征直方图，也有其他特征直方图，还有多维直方图。



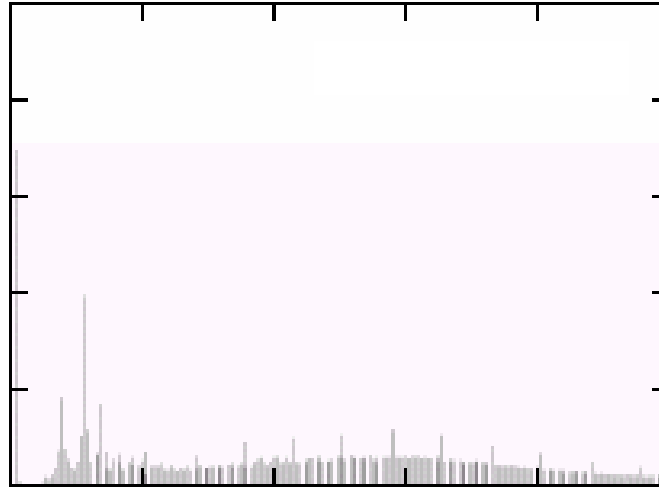
- 低对比度图像的灰度级较窄
- 高对比度图像中直方图分量覆盖了很宽的灰度级范围



# 直方图修正

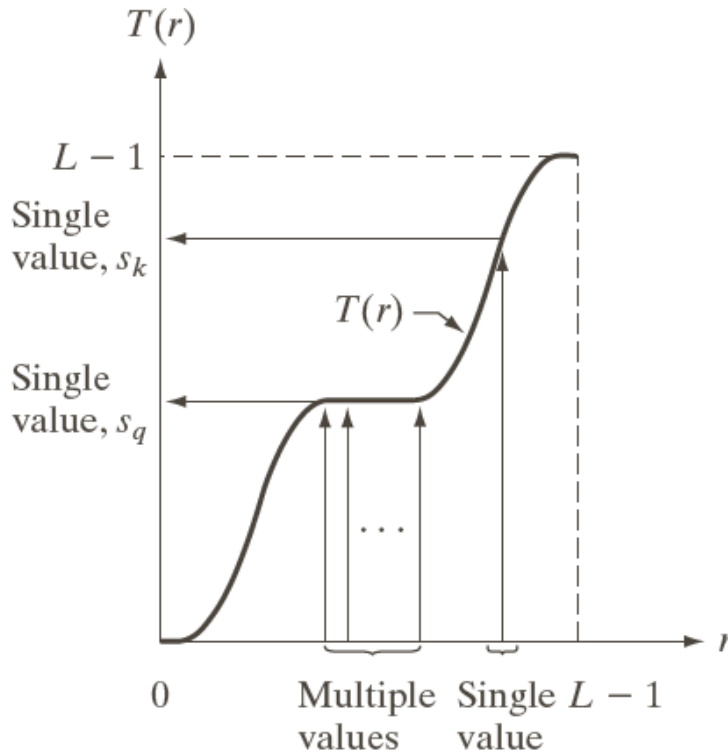


低对比度

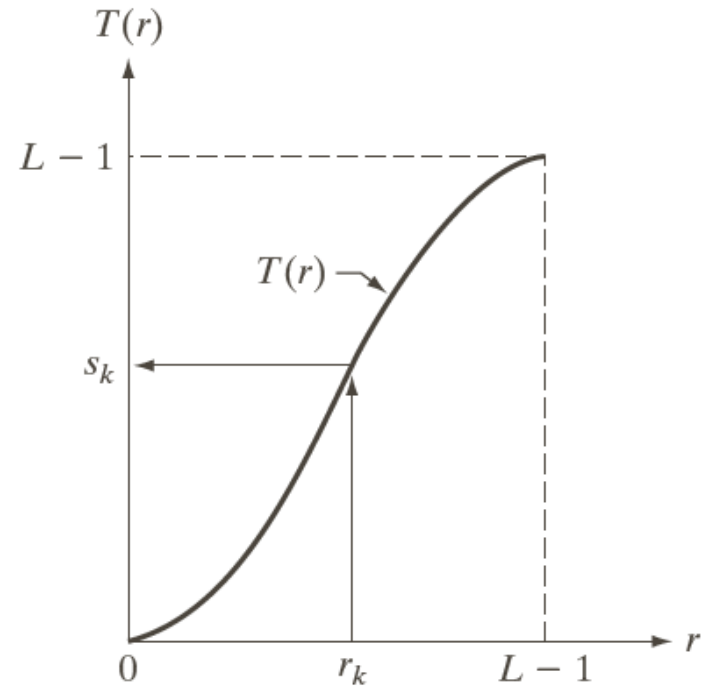


高对比度

# 灰度变换函数



非单调递增函数

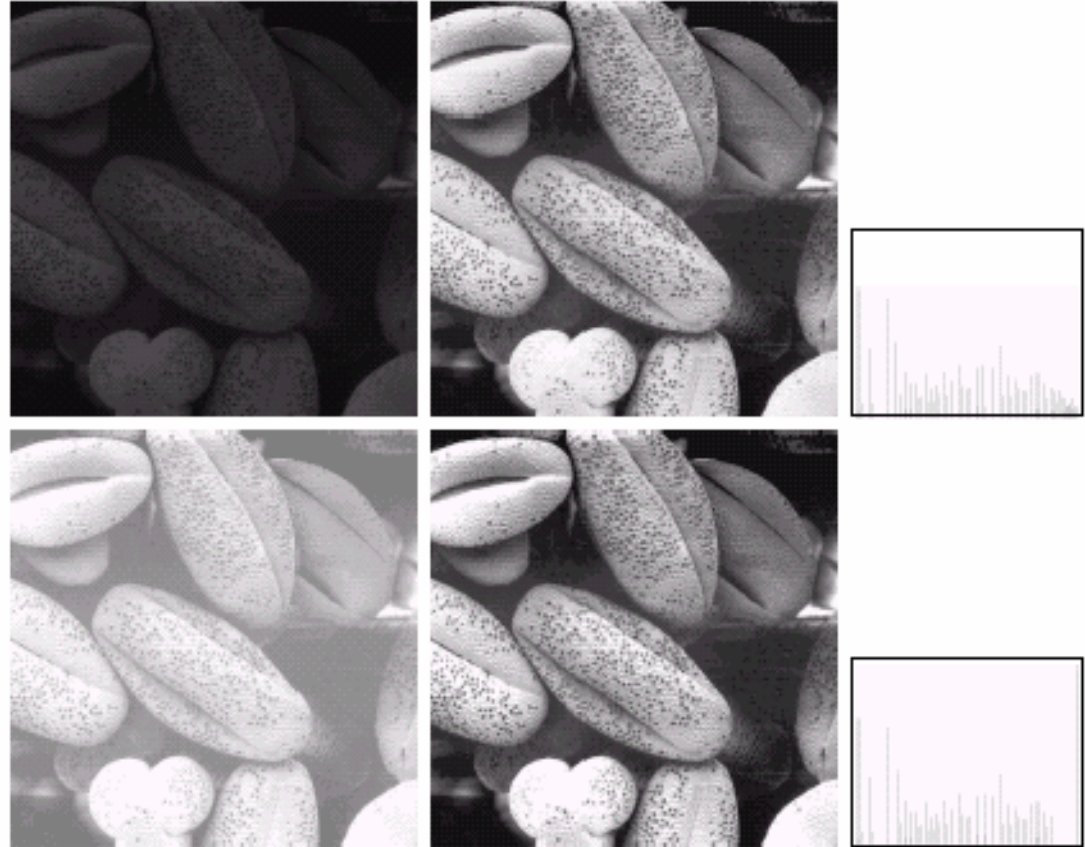


严格单调递增函数

# 直方图均衡

- 是基于直方图的灰度变换中常用的一种处理方法。
- 通常，用直方图拉伸来执行直方图均衡处理。

$$s = T(r) = (L - 1) \int_0^r p_r(w) dw$$

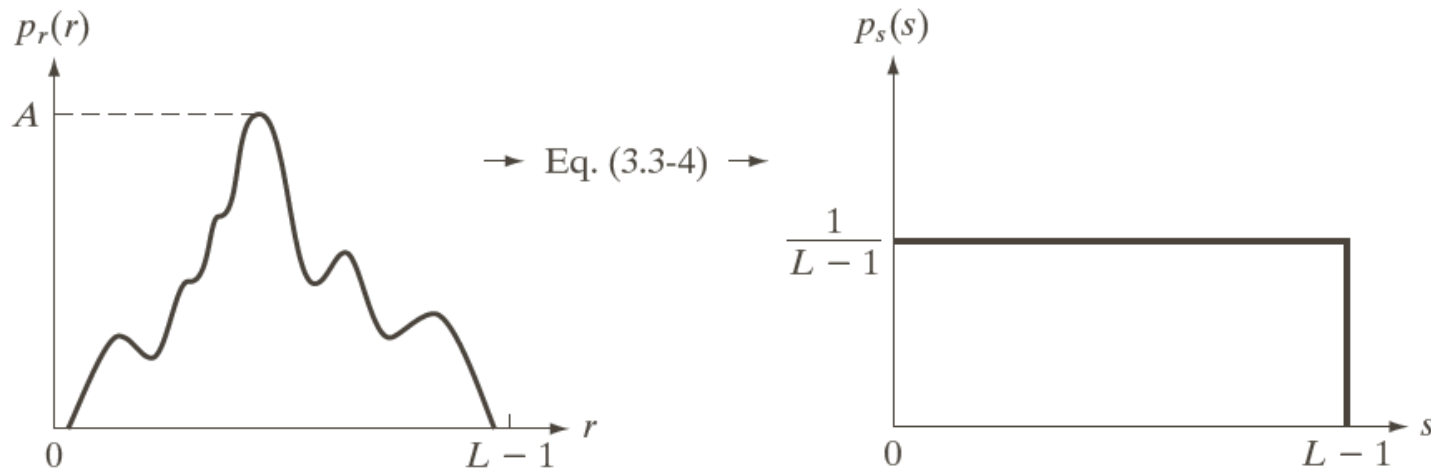


# 直方图均衡

$$s = T(r) = (L - 1) \int_0^r p_r(w) dw \quad \Rightarrow \quad \frac{ds}{dr} = (L - 1) p_r(r)$$

由莱布尼茨准则,

$$p_s(s) = p_r(r) \left| \frac{dr}{ds} \right| = p_r(r) \left| \frac{1}{(L - 1) p_r(r)} \right| = \frac{1}{L - 1}, \quad 0 \leq s \leq L - 1$$



a b

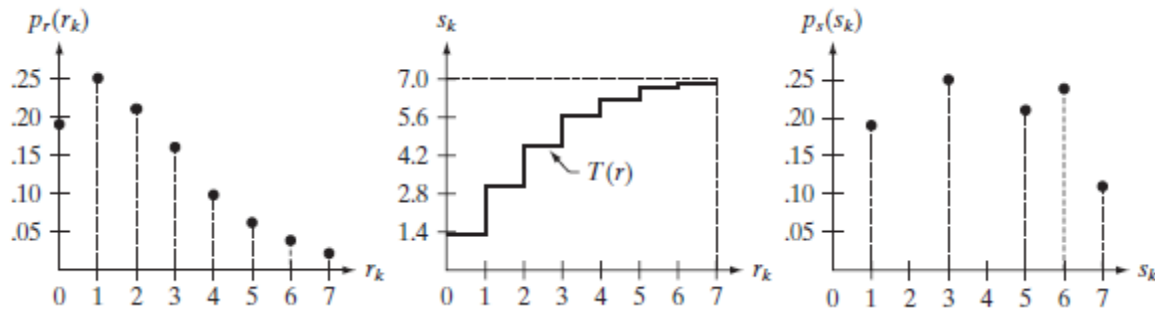
**FIGURE 3.18** (a) An arbitrary PDF. (b) Result of applying the transformation in Eq. (3.3-4) to all intensity levels,  $r$ . The resulting intensities,  $s$ , have a uniform PDF, independently of the form of the PDF of the  $r$ 's.



# 直方图均衡

$$p_r(r_k) = \frac{n_k}{MN} \quad k = 0, 1, 2, \dots, L - 1$$

$$\begin{aligned} s_k = T(r_k) &= (L - 1) \sum_{j=0}^k p_r(r_j) \\ &= \frac{(L - 1)}{MN} \sum_{j=0}^k n_j \quad k = 0, 1, 2, \dots, L - 1 \end{aligned}$$





# 直方图匹配（直方图规定化）

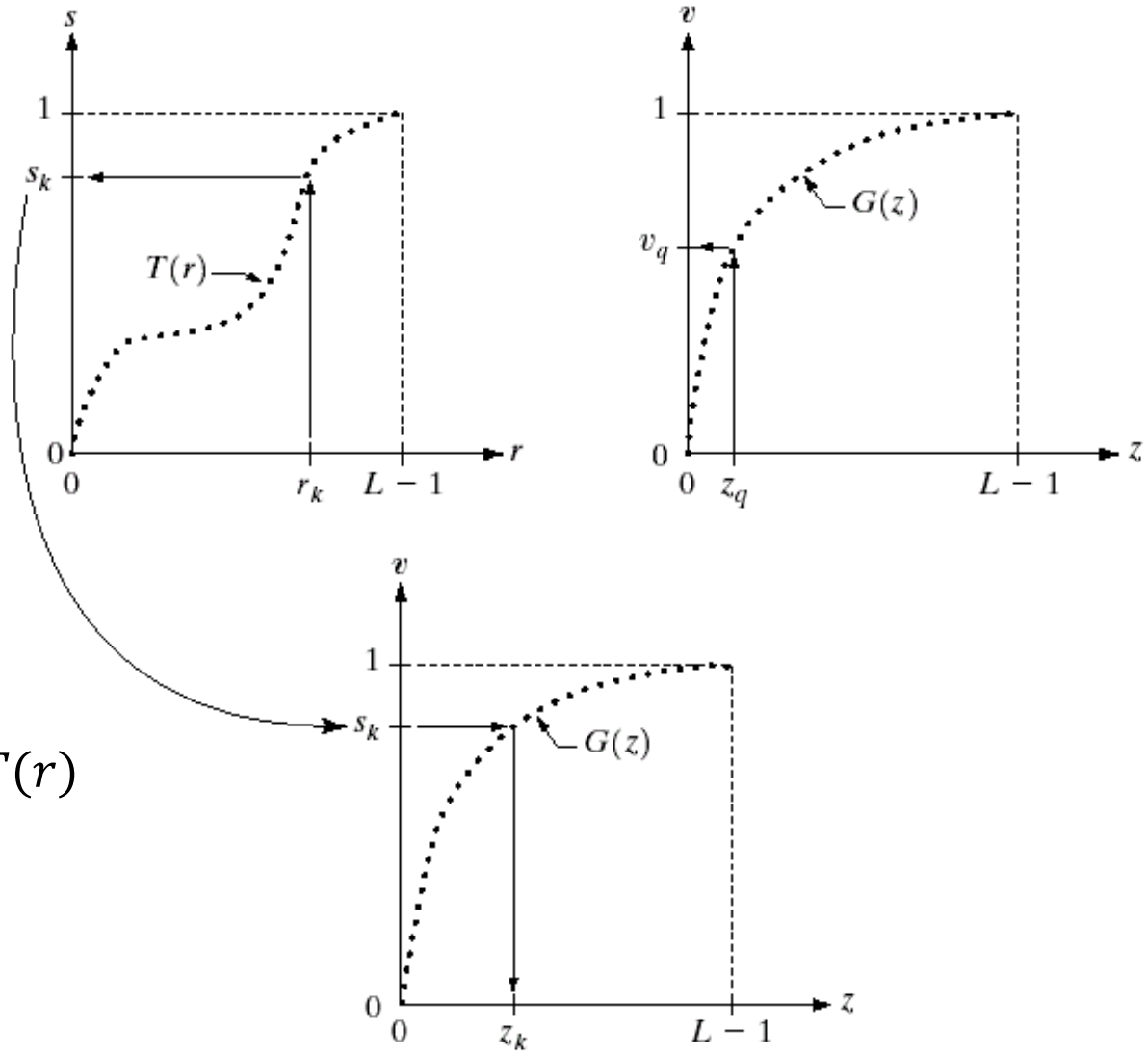
- 借助直方图变换实现规定/特定的灰度映射
  
- 三个步骤
  1. 对原始直方图进行灰度均衡化
  2. 规定需要的直方图，计算能使规定直方图均衡化的变换
  3. 将原始直方图对应映射到规定直方图

# 直方图匹配 (直方图规定化)

a b  
c

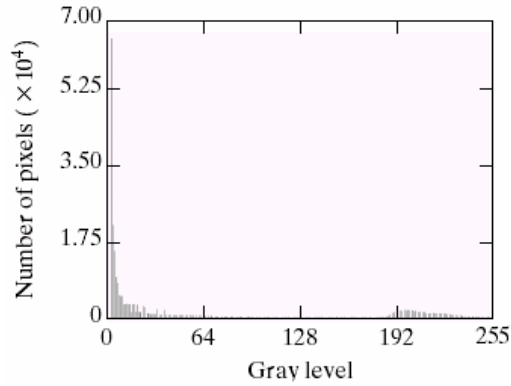
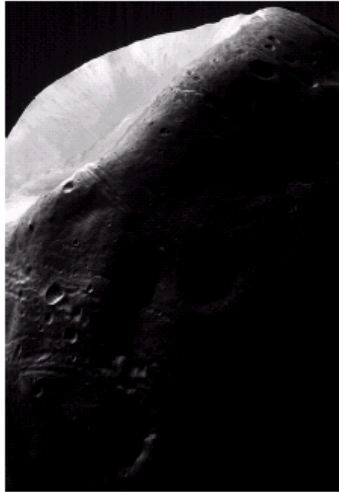
**FIGURE 3.19**

(a) Graphical interpretation of mapping from  $r_k$  to  $s_k$  via  $T(r)$ .  
 (b) Mapping of  $z_q$  to its corresponding value  $v_q$  via  $G(z)$ .  
 (c) Inverse mapping from  $s_k$  to its corresponding value of  $z_k$ .



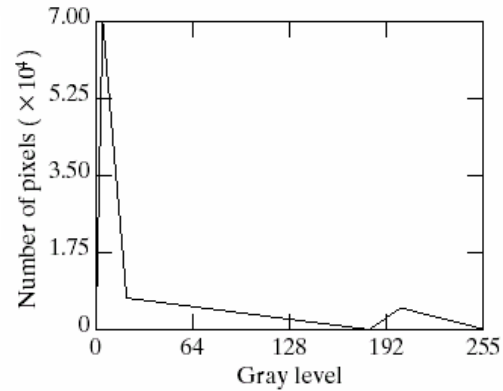


# 直方图匹配



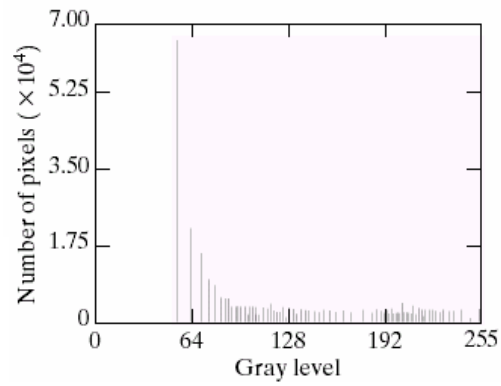
原图

原图直  
方图



处理后  
图像

匹配目  
标直方  
图



处理后  
直方图

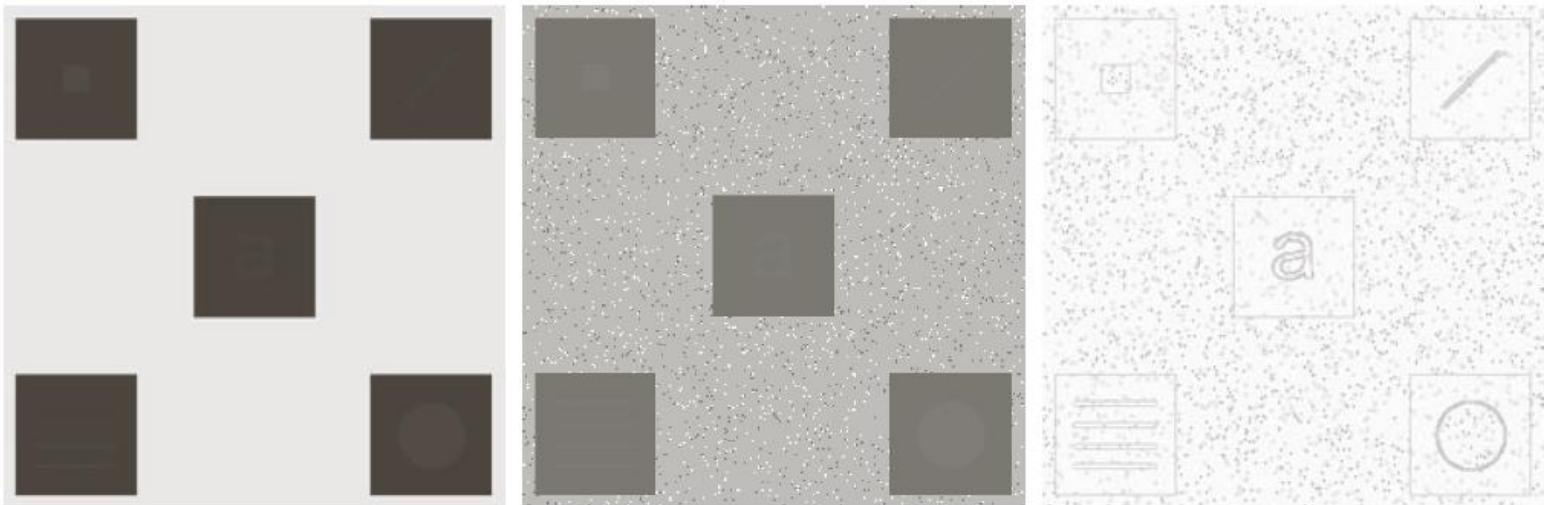


# 直方图均衡 vs. 直方图匹配

- 直方图均衡：
  - 自动增强
  - 效果不易控制
  - 总得到全图增强的结果
- 直方图匹配：
  - 有选择地增强
  - 须给定需要的直方图
  - 可特定增强的结果
- 直方图均衡可以看作是匹配直方图为均匀分布的直方图匹配

# 局部直方图处理

## □ 图像局部邻域的直方图均衡化

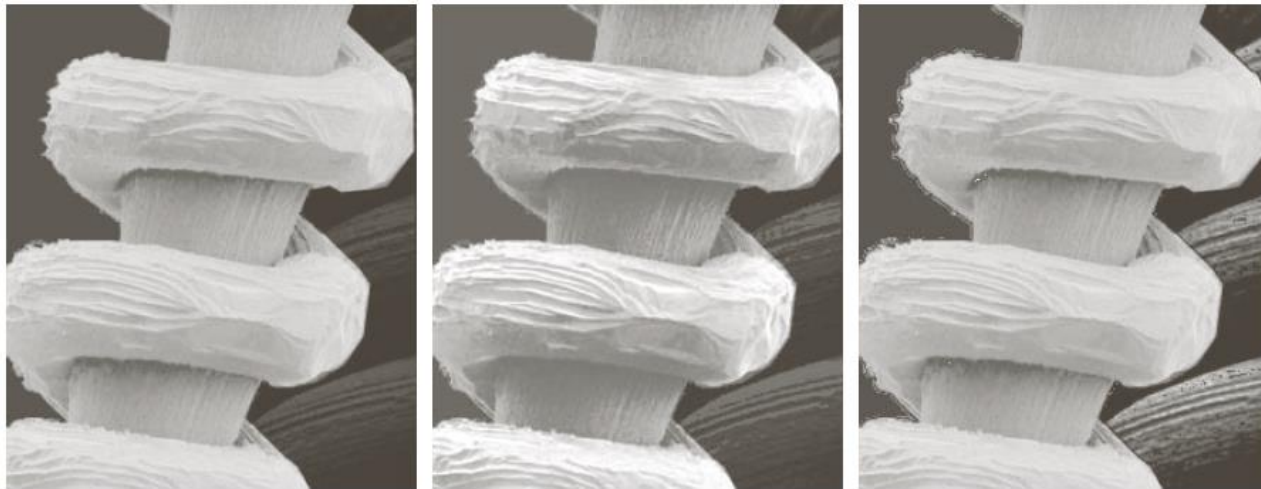


a b c

**FIGURE 3.26** (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization applied to (a), using a neighborhood of size  $3 \times 3$ .

# 基于直方图统计的图像增强

- 基于图像的直方图，很容易计算出图像像素灰度的均值和标准差
- 基于像素邻域的均值和标准差，可以选择性地对某些像素灰度进行增强



a b c

**FIGURE 3.27** (a) SEM image of a tungsten filament magnified approximately 130 $\times$ . (b) Result of global histogram equalization. (c) Image enhanced using local histogram statistics. (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)



# 第3章 灰度变换与空域滤波

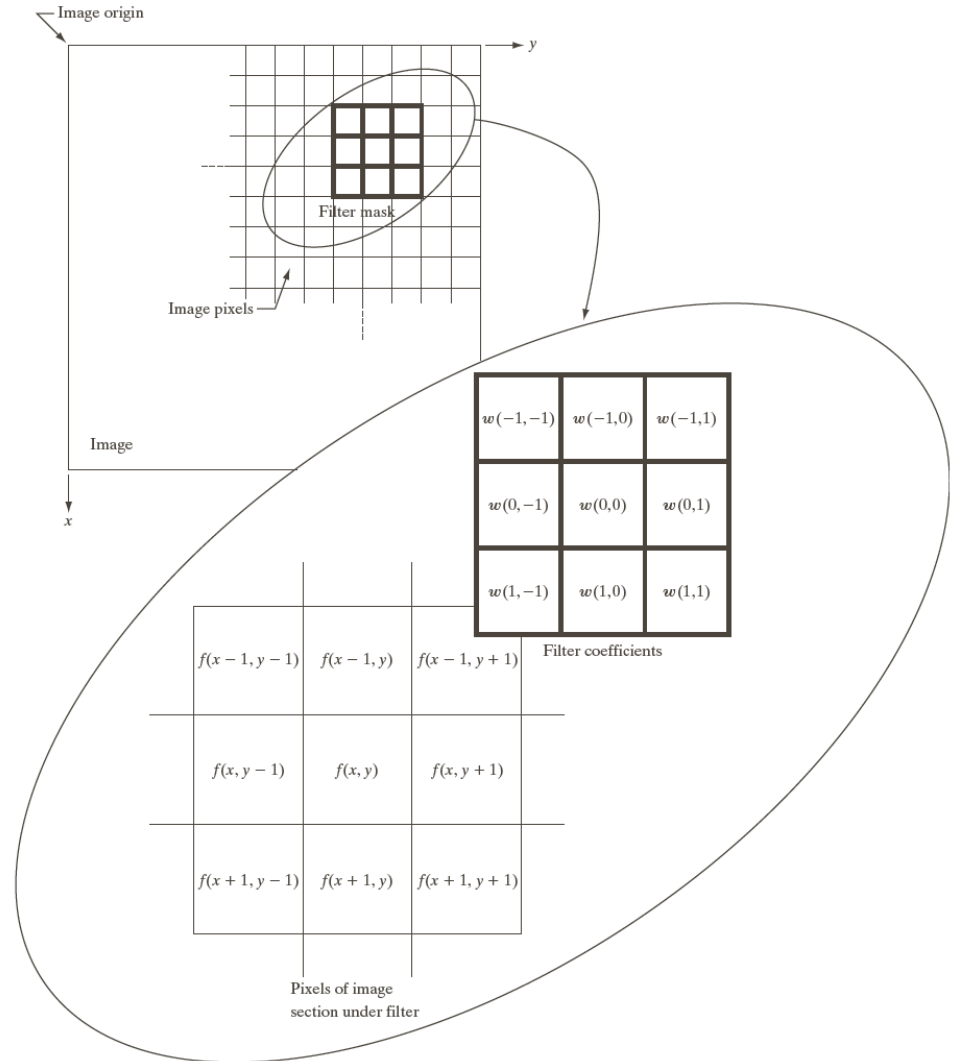
---

- 3.1 背景知识
- 3.2 基本灰度变换
- 3.3 直方图处理
- 3.4 空间滤波基础**
- 3.5 空域平滑
- 3.6 空域锐化

# 空域滤波机制

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)$$

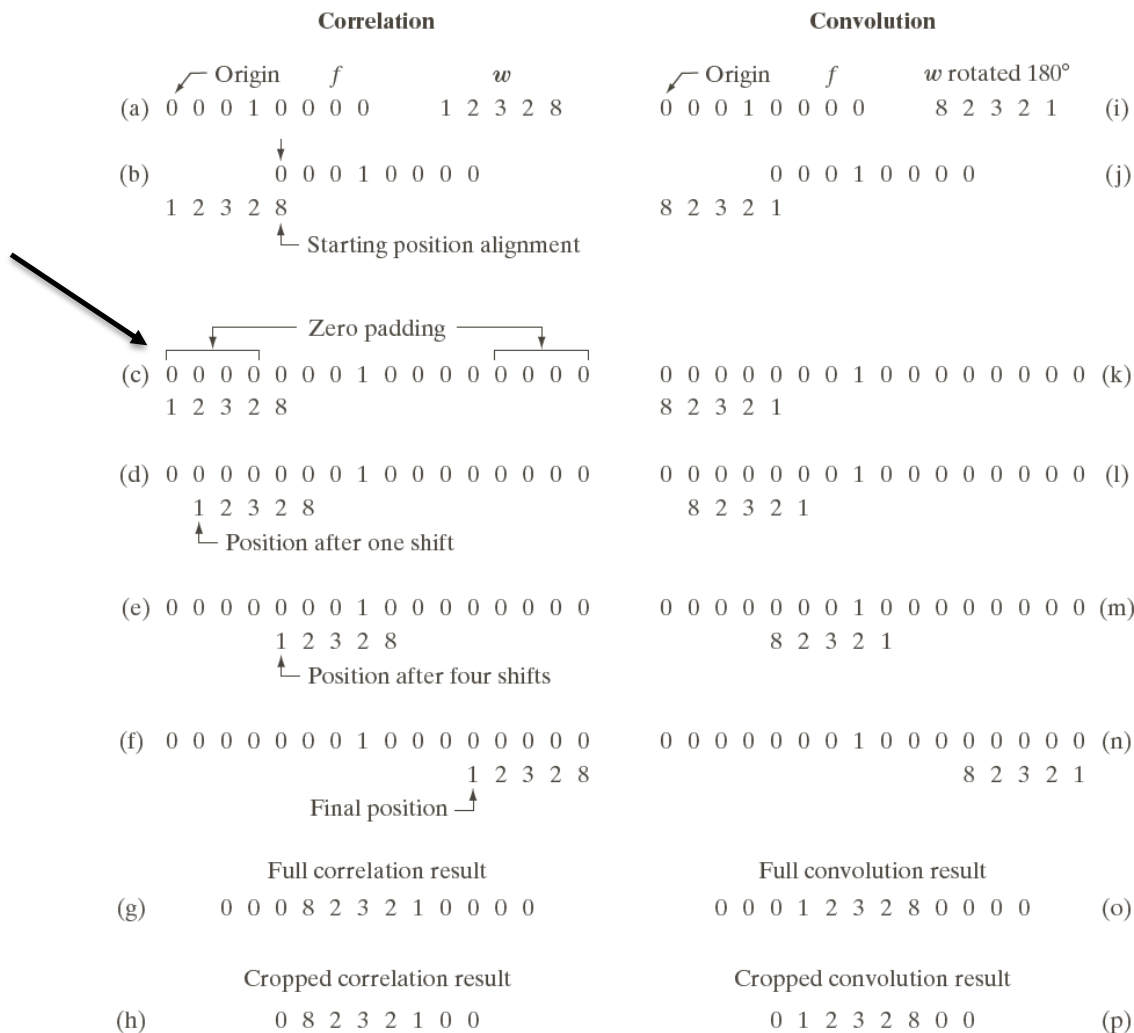
**相关 vs. 卷积：**卷积运算需要对卷积核进行反转操作，而相关不需要。





# 空间相关与卷积：一维情况

- 边界延拓：
- 1) 补零延拓
  - 2) 对称延拓
  - 3) 复制延拓
  - 4) 循环延拓

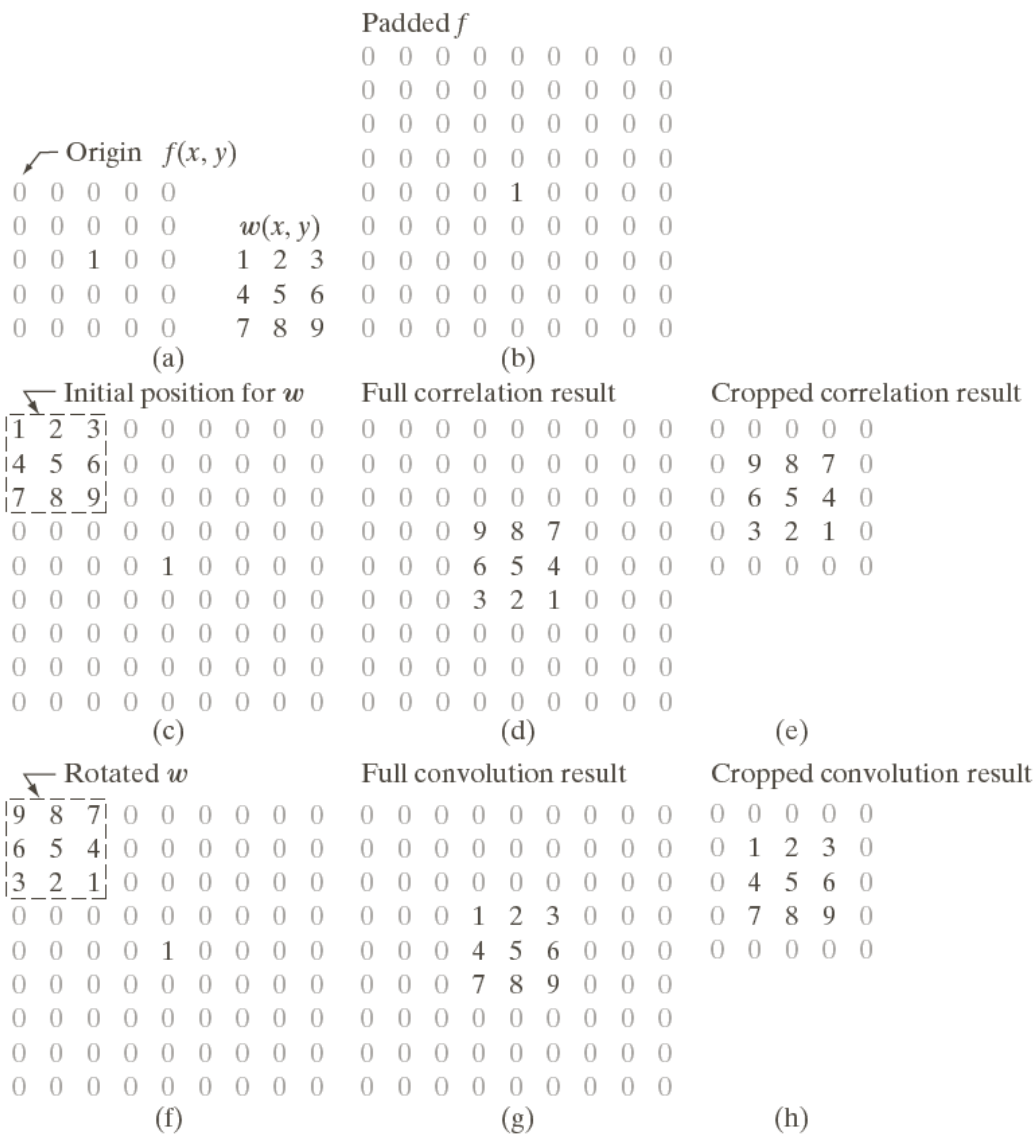


**FIGURE 3.29** Illustration of 1-D correlation and convolution of a filter with a discrete unit impulse. Note that correlation and convolution are functions of *displacement*.





# 空间相关与卷积：二维情况



**FIGURE 3.30**  
Correlation (middle row) and convolution (last row) of a 2-D discrete, unit impulse. The 0s are shown in gray to simplify visual analysis.

相关运算：

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)$$

卷积运算：

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x - s, y - t)$$



# 滤波器的向量表示与模板生成

- 当关注相关或卷积的响应特性时，可以将响应写为乘积的求和形式

$$R = w_1 z_1 + w_2 z_2 + \cdots + w_{MN} z_{MN} = \mathbf{w}^T \mathbf{z}$$

- 等价于将二维滤波器按一维进行编号

- 空间滤波器的模板生成

- 滤波器的系数根据其支持怎么样的操作来选择
- **线性滤波**的作用是实现乘积求和操作：加权求和

$$w_{mn} = \frac{1}{MN} \quad w_{mn} = \exp\left(-\frac{\left(m - \left\lfloor \frac{M}{2} \right\rfloor\right)^2 + \left(n - \left\lfloor \frac{N}{2} \right\rfloor\right)^2}{2\sigma}\right)$$

- **非线性滤波**要求确定领域大小，以及对包含在领域内的图像像素执行的操作



# 第3章 灰度变换与空域滤波

---

- 3.1 背景知识
- 3.2 基本灰度变换
- 3.3 直方图处理
- 3.4 空间滤波基础
- 3.5 空域平滑
- 3.6 空域锐化

# 平滑滤波器

## 局部平均法

局部平均法是用某象素邻域内的各点灰度的平均值来代替该象素原来的灰度值。

$$g(x, y) = \frac{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t)}{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t)}$$

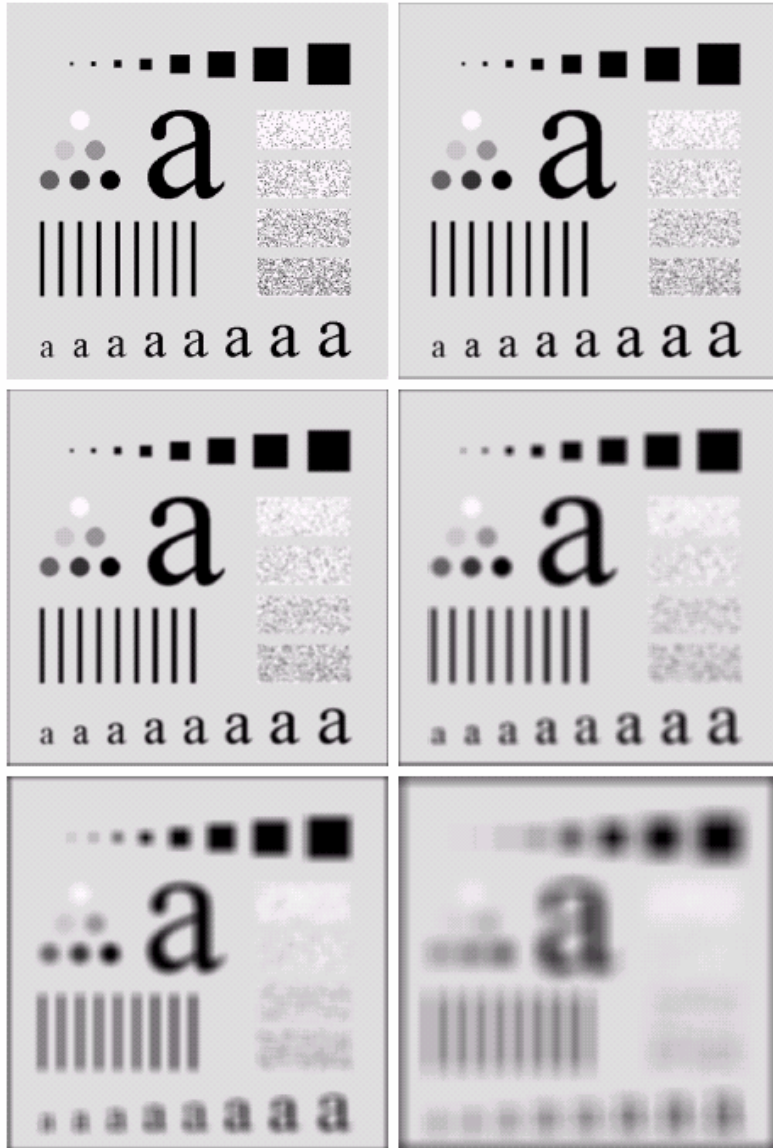
$$\frac{1}{9} \times$$

1	1	1
1	1	1
1	1	1

$$\frac{1}{16} \times$$

1	2	1
2	4	2
1	2	1

# 图像平滑



原图

n=3方形均值滤波

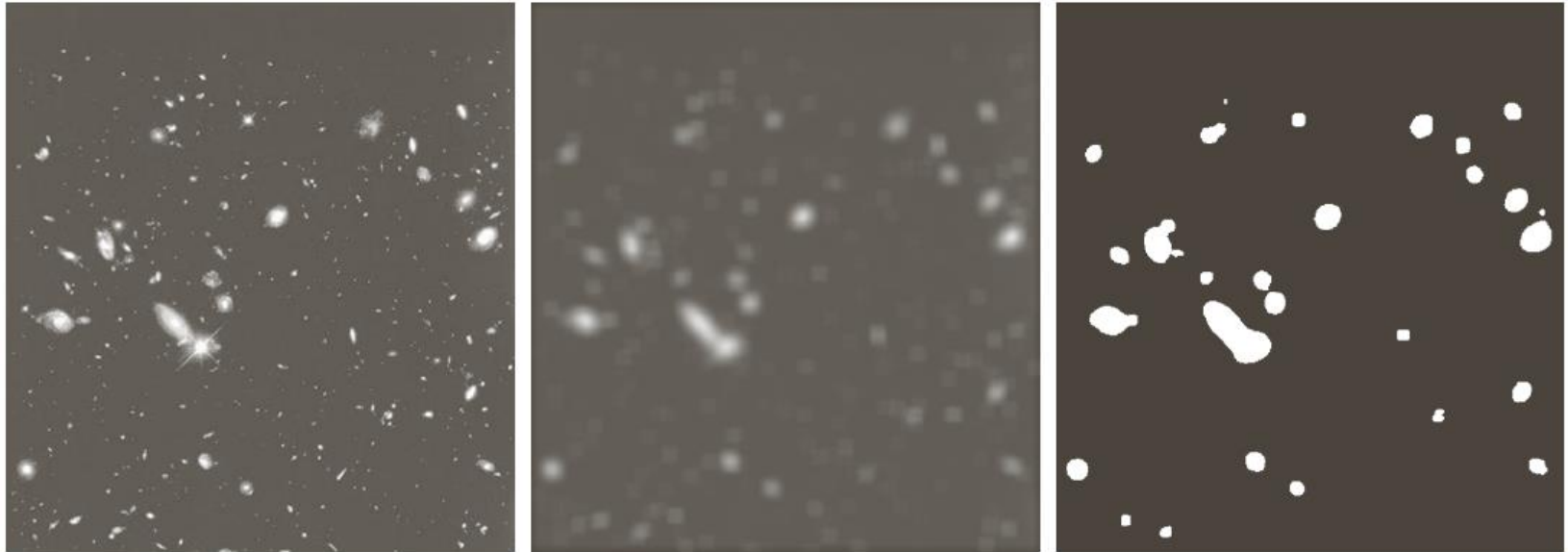
n=5方形均值滤波

n=9方形均值滤波

n=15方形均值滤波

n=35方形均值滤波

# 图像平滑



a b c

**FIGURE 3.34** (a) Image of size  $528 \times 485$  pixels from the Hubble Space Telescope. (b) Image filtered with a  $15 \times 15$  averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)



# 统计排序滤波器

## □ 中值滤波

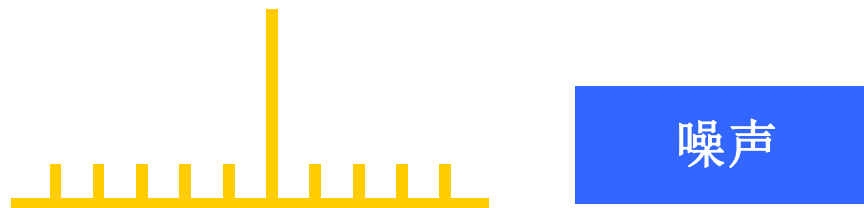
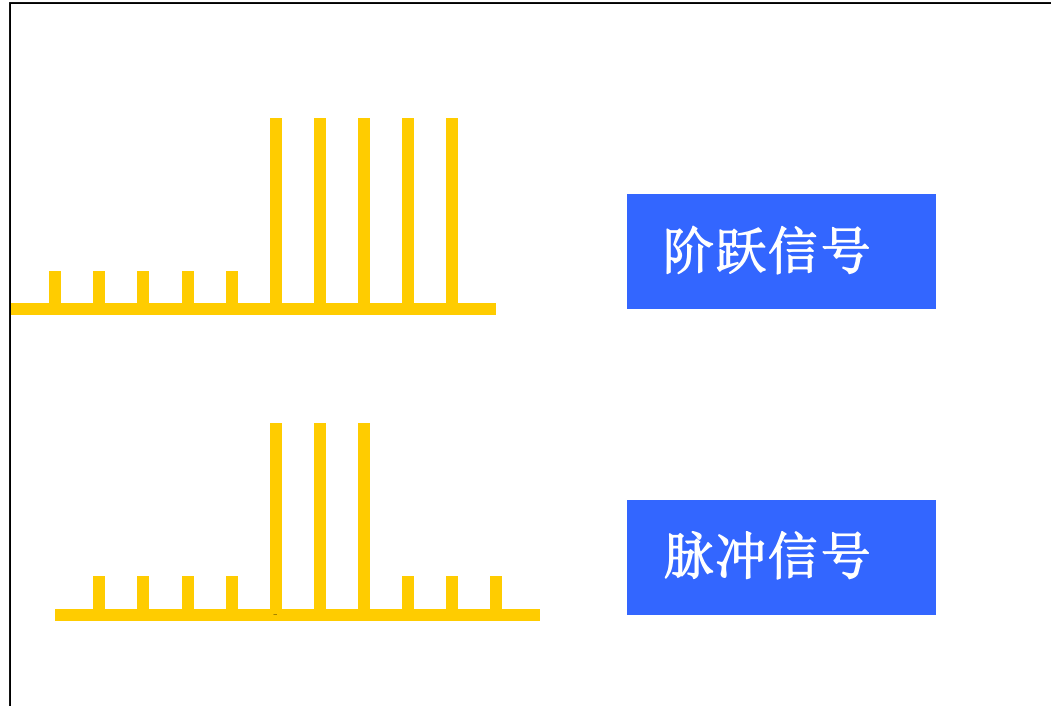
- 用局部邻域（窗口）里的中值来代替上述局部平均法中的局部平均值。即将以该点为中心的某个窗口框住的各象素的中间值作为处理后图象中该点象素的值。
- 二维中值滤波的概念是一维的推广。
- 以一维为例：
  - 奇数窗口中五点的值为 88、95、230、110、120，按大小重排这五个值得 88、95、110、120、230，其位于中间位置上的“中值”是110。
  - 实际应用：美国家庭收入统计（中位数）

## □ 其他统计排序滤波器

- 最大值滤波器，最小值滤波器



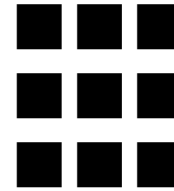
# 中值滤波保边缘的性能



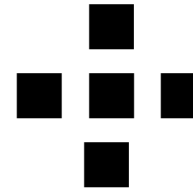
# 二维中值滤波窗口

大小、形状  
均可调节

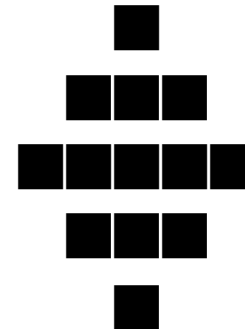
方形



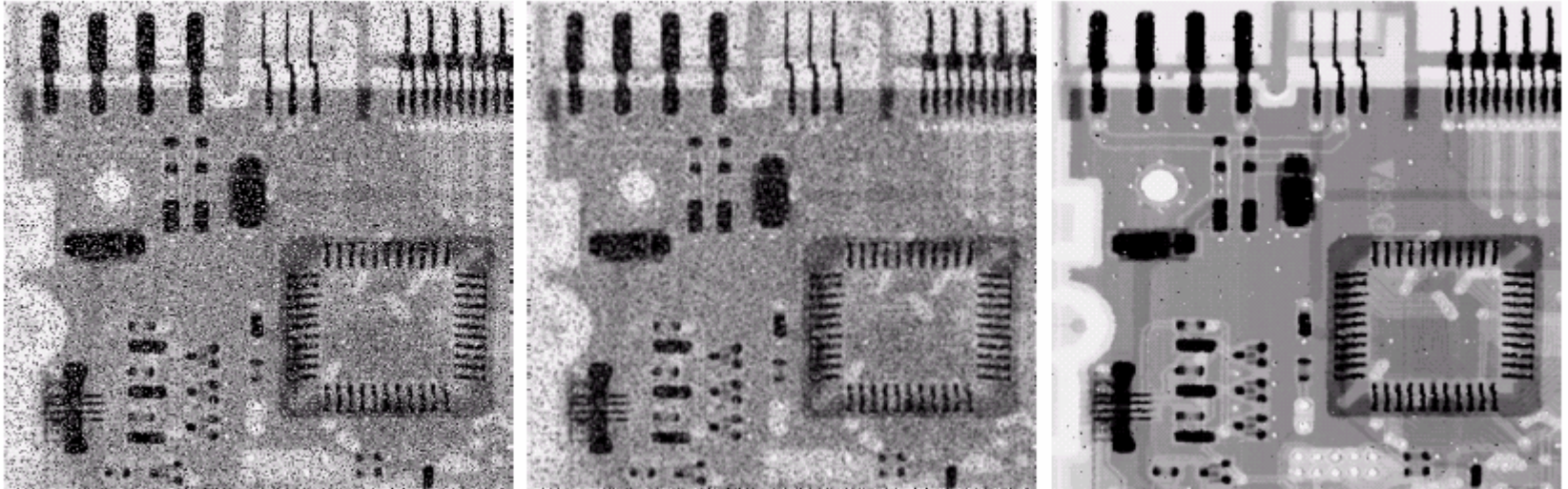
十字形



菱形



# 例



从左到右依次分别为

椒盐噪声污染的电路板X光图像

用 $3 \times 3$ 均值掩模去除噪声

用 $3 \times 3$ 中值滤波器去除噪声

去除椒盐噪声使用自适应中值滤波器的效果会更好一些，详情参见《数字图像处理》（冈萨雷斯）



# 第3章 灰度变换与空域滤波

---

- 3.1 背景知识
- 3.2 基本灰度变换
- 3.3 直方图处理
- 3.4 空间滤波基础
- 3.5 空域平滑
- 3.6 空域锐化



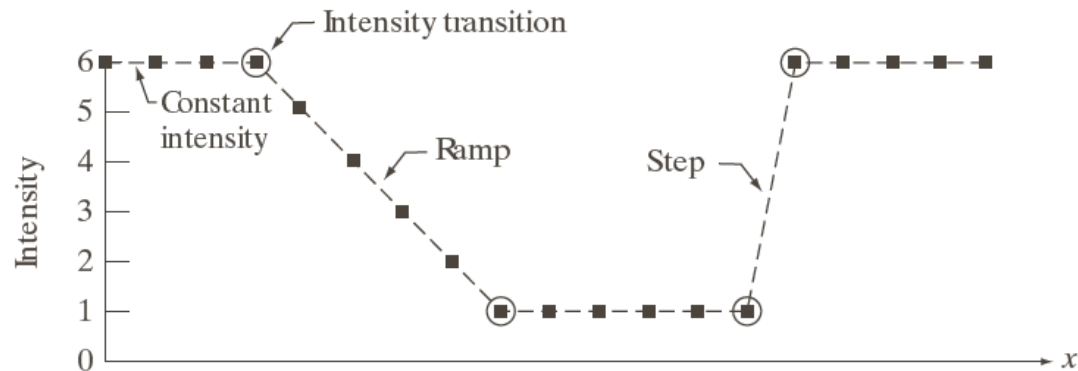
# 空域锐化

- 图象锐化针对常见的图像模糊、边缘不清晰所采取的处理方法，它能加强图象的轮廓，使图象看起来比较清楚。
- 锐化处理的主要目的：突出灰度的过渡部分
  - 常用一阶微分和二阶微分的锐化滤波器

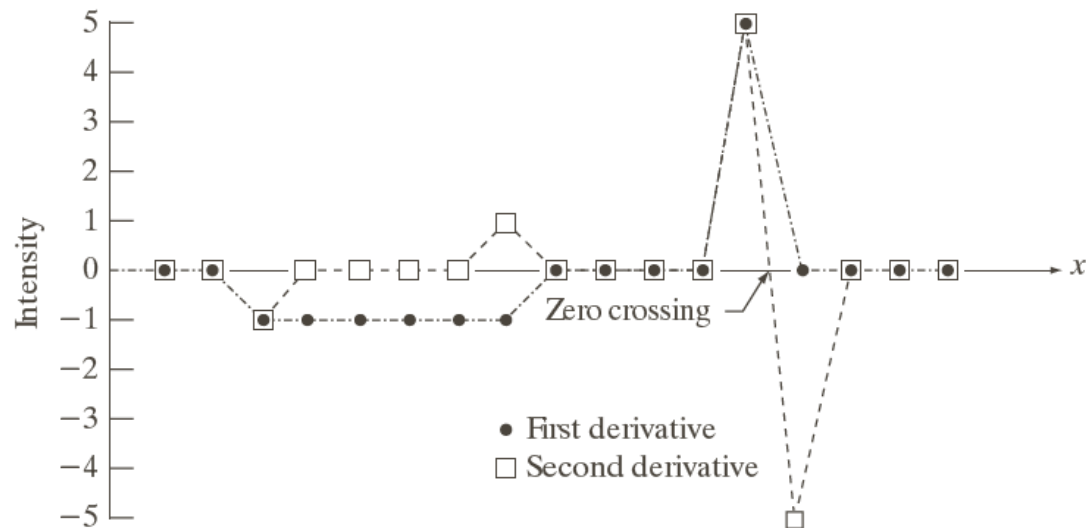
$$\frac{\partial f}{\partial x} = f(x + 1) - f(x)$$

$$\frac{\partial^2 f}{\partial x^2} = f(x + 1) + f(x - 1) - 2f(x)$$

# 空域锐化



Scan line	6	6	6	6	5	4	3	2	1	1	1	1	1	1	6	6	6	6	6
1st derivative	0	0	-1	-1	-1	-1	-1	0	0	0	0	0	0	5	0	0	0	0	0
2nd derivative	0	0	-1	0	0	0	0	1	0	0	0	0	0	5	-5	0	0	0	0



a  
b  
c

**FIGURE 3.36** Illustration of the first and second derivatives of a 1-D digital function representing a section of a horizontal intensity profile from an image. In (a) and (c) data points are joined by dashed lines as a visualization aid.



# 一阶微分锐化图像

- 分析表明微分可以突出高频成分，从而使图象轮廓清晰。因此在空间域可用微分法锐化图象。
- 图象处理中常用的微分方法是“梯度法”，即用梯度作为锐化图象的度量。对于数字图象，则用差分来表示梯度。
- 实际应用中，为了获得各向同性的度量，用梯度的模代替梯度。



# 一阶微分锐化图像

□ 典型的差分运算，可表示为：

$$G[f(x, y)] \approx |f(x, y) - f(x - 1, y)| + |f(x, y) - f(x, y - 1)|$$

<b>0</b>	<b>0</b>
<b>-1</b>	<b>1</b>

<b>0</b>	<b>-1</b>
<b>0</b>	<b>1</b>

直接差分算子



# 梯度算子

- Roberts算子
- Sobel算子
- Prewitt算子
- Kirsch算子

空间频率的滤波  
也可以在空间域  
来实现

-1	0
0	1

0	-1
1	0

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

5	5	5
-3	0	-3
-3	-3	-3

-3	5	5
-3	0	5
-3	-3	-3

# 梯度算子

原图



$|G_y|$

$|G_x|$

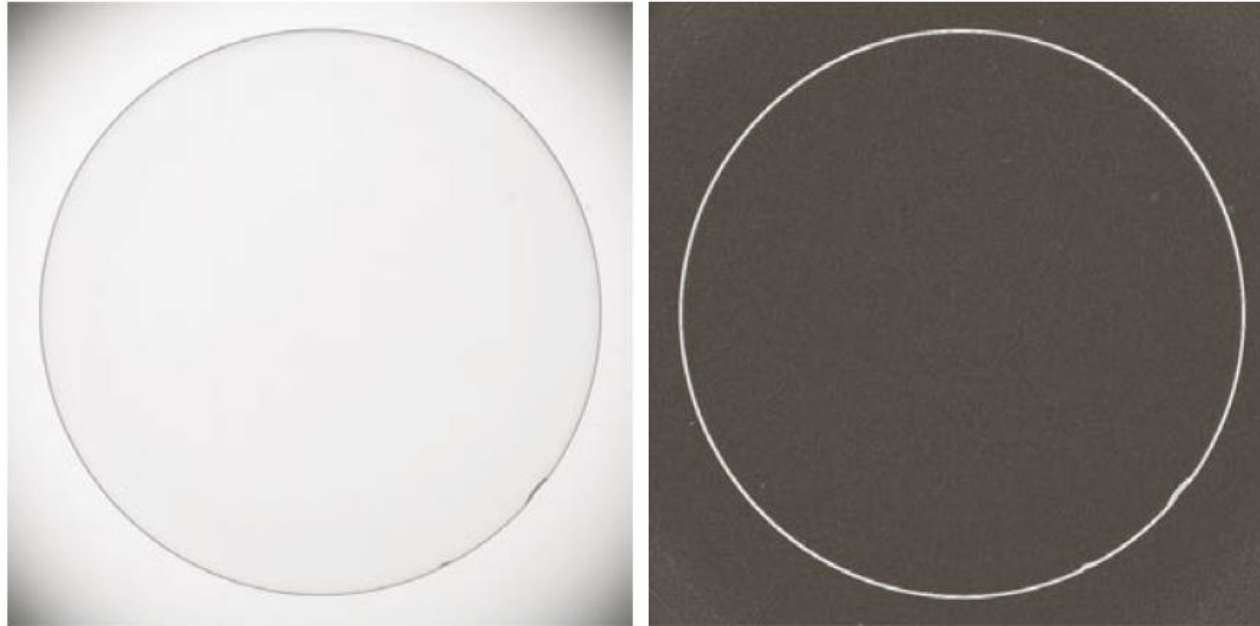


$|G_x|$

+

$|G_y|$

# 一阶微分锐化图像



a b

**FIGURE 3.42**  
(a) Optical image of contact lens (note defects on the boundary at 4 and 5 o'clock).  
(b) Sobel gradient.  
(Original image courtesy of Pete Sites, Perceptics Corporation.)



# 拉普拉斯算子

一个二元图像函数 $f(x, y)$ 的拉普拉斯变换定义为

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

几种常用的拉普拉斯算子:

0	1	0
1	-4	1
0	1	0

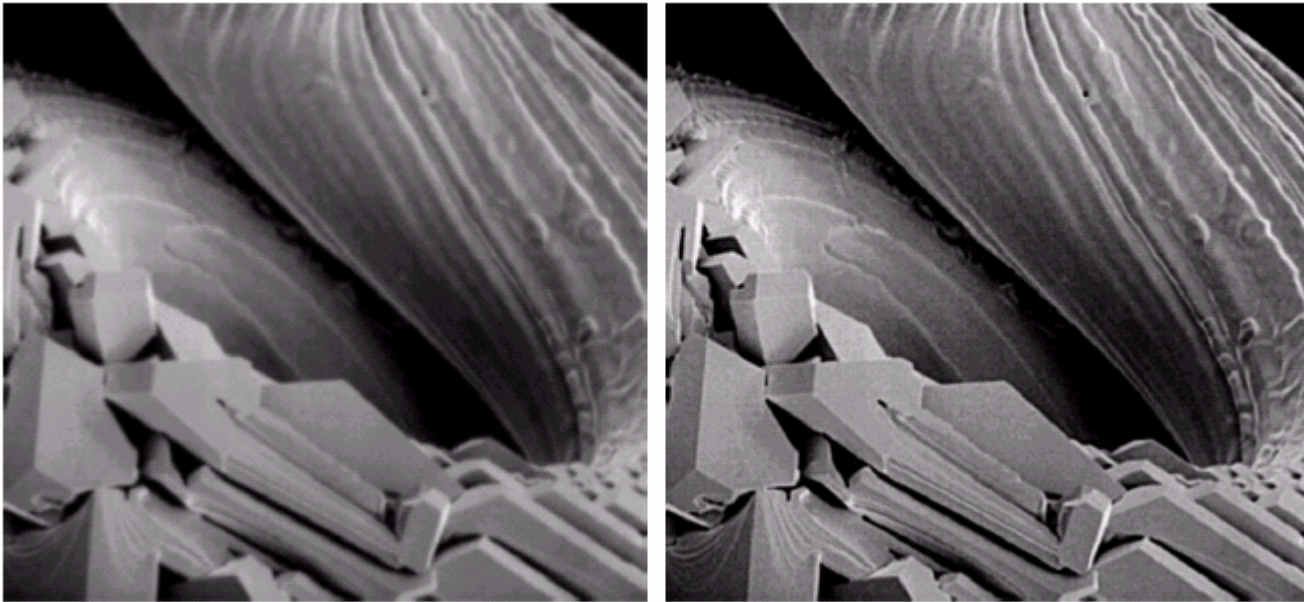
1	1	1
1	-8	1
1	1	1

0	-1	0
-1	4	-1
0	-1	0

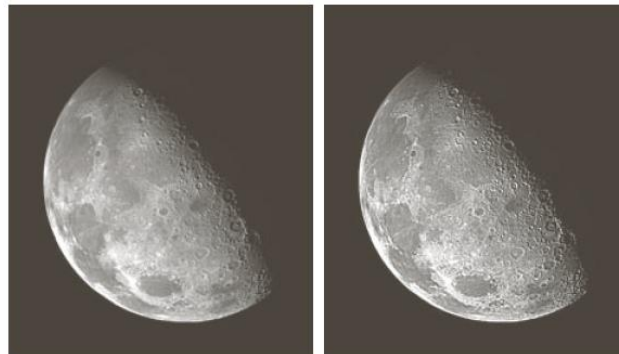
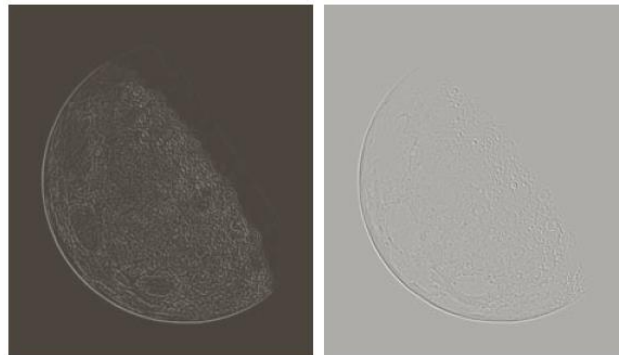
-1	-1	-1
-1	8	-1
-1	-1	-1

# 使用拉普拉斯算子进行图像增强

$$g(x, y) = \begin{cases} f(x, y) - \nabla^2 f(x, y) & \text{如果拉普拉斯掩模中心系数为负} \\ f(x, y) + \nabla^2 f(x, y) & \text{如果拉普拉斯掩模中心系数为正} \end{cases}$$



# 使用拉普拉斯算子进行图像增强



0	-1	0
-1	4	-1
0	-1	0

a  
b c  
d e

**FIGURE 3.38**

(a) Blurred image of the North Pole of the moon.  
 (b) Laplacian without scaling.  
 (c) Laplacian with scaling.  
 (d) Image sharpened using the mask in Fig. 3.37(a).  
 (e) Result of using the mask in Fig. 3.37(b).  
 (Original image courtesy of NASA.)

-1	-1	-1
-1	8	-1
-1	-1	-1





# 数字图象处理

---

中国科学技术大学  
电子工程与信息科学系

主讲教师：李厚强 ([lihq@ustc.edu.cn](mailto:lihq@ustc.edu.cn))  
周文罡 ([zhwg@ustc.edu.cn](mailto:zhwg@ustc.edu.cn))

助教：谢乔康 ([xieqiaok@mail.ustc.edu.cn](mailto:xieqiaok@mail.ustc.edu.cn))  
周 浩([zhouh156@mail.ustc.edu.cn](mailto:zhouh156@mail.ustc.edu.cn))



# 第4章 频率域滤波

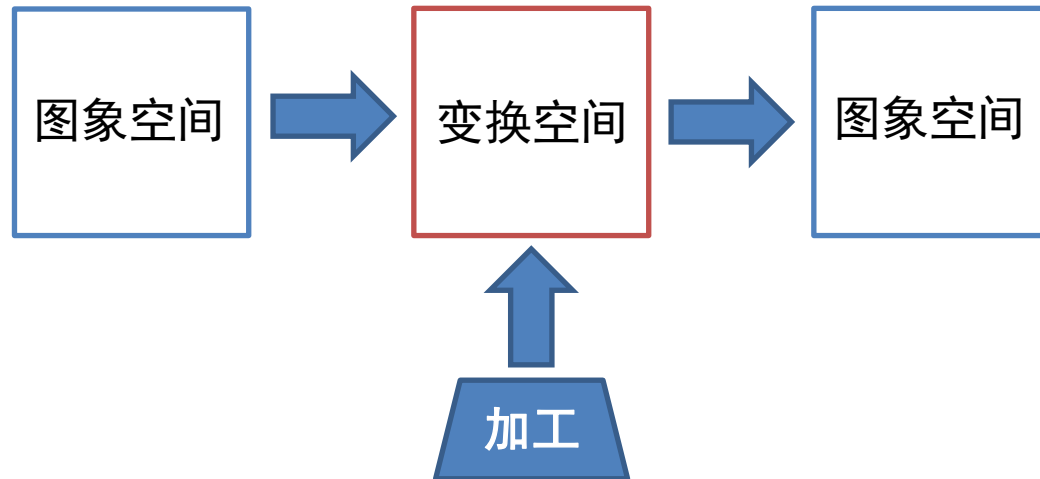
---

- 4.1 离散傅立叶变换(DFT)
- 4.2 频率域滤波基础
- 4.3 频率域滤波器平滑图像
- 4.4 频率域滤波器锐化图像



# 第4章 频率域滤波：图像变换

**动机：**为了有效和快速地对图象进行处理，常常需要将原定义在图象空间的图象以某种形式转换到另外一些空间（频率域空间）并加工，最后再转换回图象空间以得到所需的效果。





# 第4章 频率域滤波

---

- 4.1 离散傅立叶变换(DFT)
- 4.2 频率域滤波基础
- 4.3 频率域滤波器平滑图像
- 4.4 频率域滤波器锐化图像

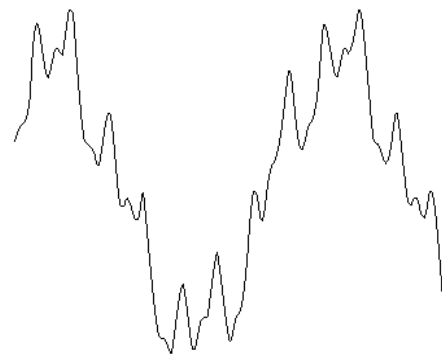
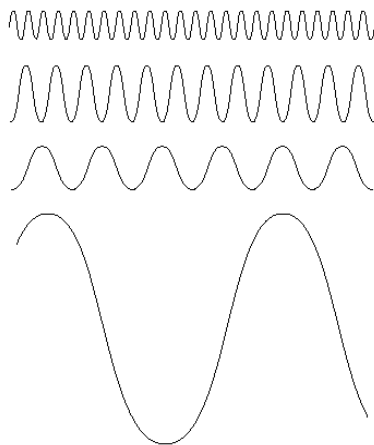
# 4.1 傅立叶变换基础

## □ 傅里叶级数 (FS)

- 任何周期函数都可以表示为不同频率的正弦或余弦之和的形式，每个正弦项和余弦项都乘以不同的系数

## □ 傅里叶变换 (FT)

- 任何非周期函数（其曲线下的面积是有限的）可以表示为正弦和/或余弦乘以加权函数的积分来表示



**FIGURE 4.1** The function at the bottom is the sum of the four functions above it. Fourier's idea in 1807 that periodic functions could be represented as a weighted sum of sines and cosines was met with skepticism.



# 4.1 傅立叶变换基础

## 一维傅立叶变换及其反变换

$$F(u) = \int_{-\infty}^{\infty} f(x) e^{-j2\pi ux} dx$$

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux+vy)} dx dy$$

$$F(x) = \int_{-\infty}^{\infty} f(u) e^{j2\pi ux} du$$

$$F(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(u, v) e^{j2\pi(ux+vy)} du dv$$

离散形式:

$$F(u) = \sum_{x=0}^{M-1} f(x) e^{-j2\pi ux/M} \quad x = 0, 1, 2, \dots, M-1$$

$$f(x) = \frac{1}{M} \sum_{u=0}^{M-1} F(u) e^{j2\pi ux/M} \quad u = 0, 1, 2, \dots, M-1$$

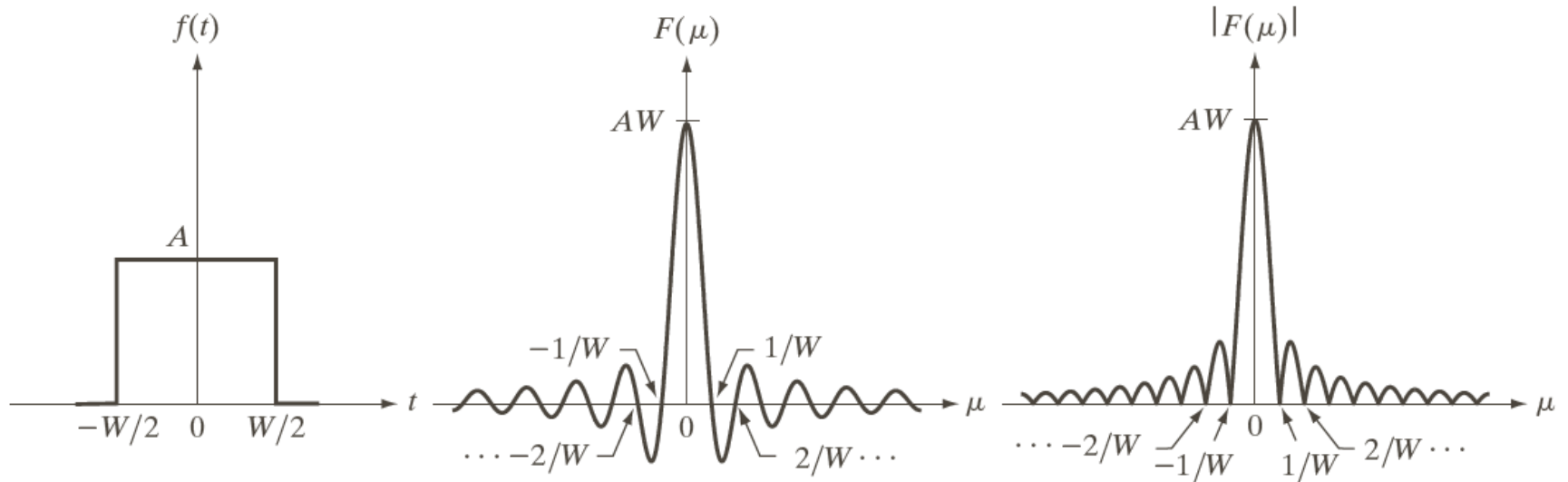
频域→不同的频域成份, 可以表示成极坐标形式:

$$F(u) = |F(u)| e^{-j\phi(u)}$$

$$|F(u)| = \left[ R^2(u) + I^2(u) \right]^{\frac{1}{2}}$$

$$\phi(u) = \arctan \left[ \frac{I(u)}{R(u)} \right]$$

# 4.1 傅立叶变换基础

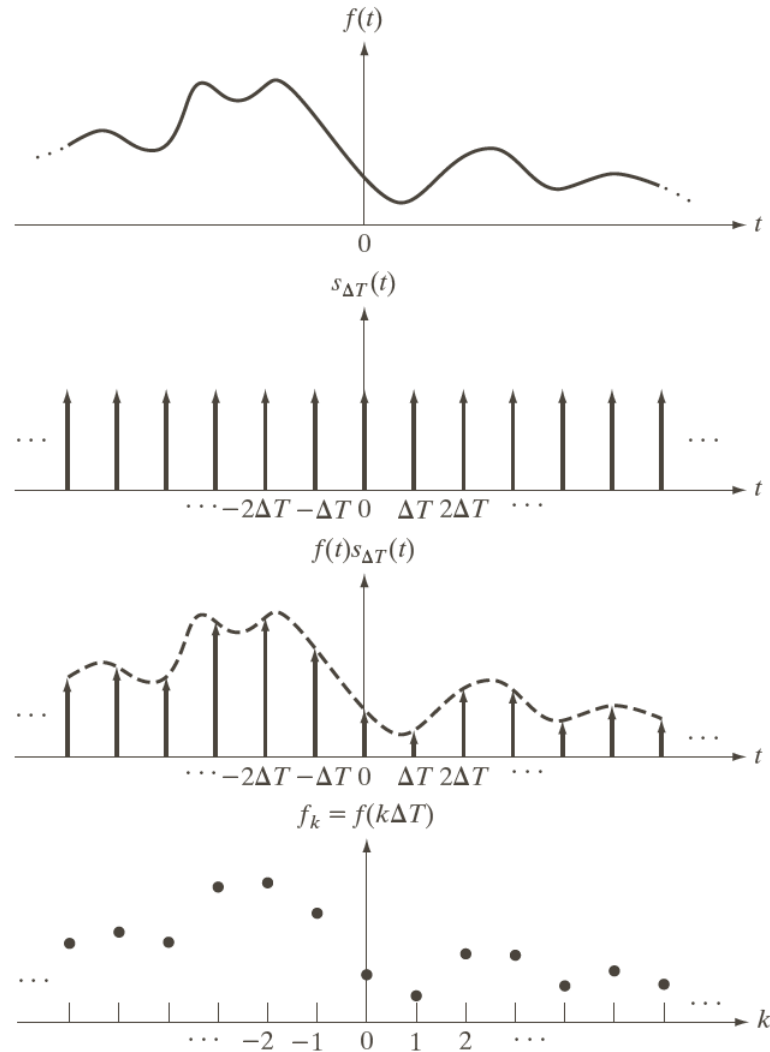


a b c

**FIGURE 4.4** (a) A simple function; (b) its Fourier transform; and (c) the spectrum. All functions extend to infinity in both directions.



# 取样：连续函数转换为离散序列

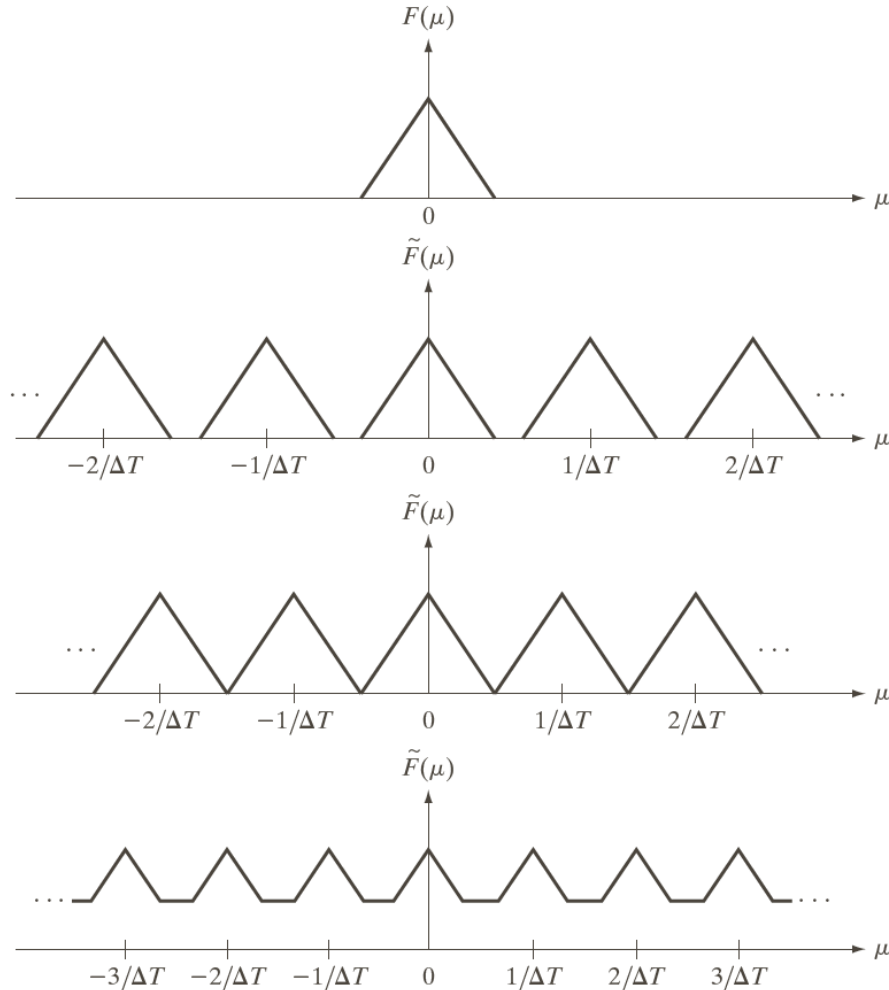


a  
b  
c  
d

**FIGURE 4.5**  
 (a) A continuous function. (b) Train of impulses used to model the sampling process. (c) Sampled function formed as the product of (a) and (b). (d) Sample values obtained by integration and using the sifting property of the impulse. (The dashed line in (c) is shown for reference. It is not part of the data.)

# 取样函数的傅里叶变换

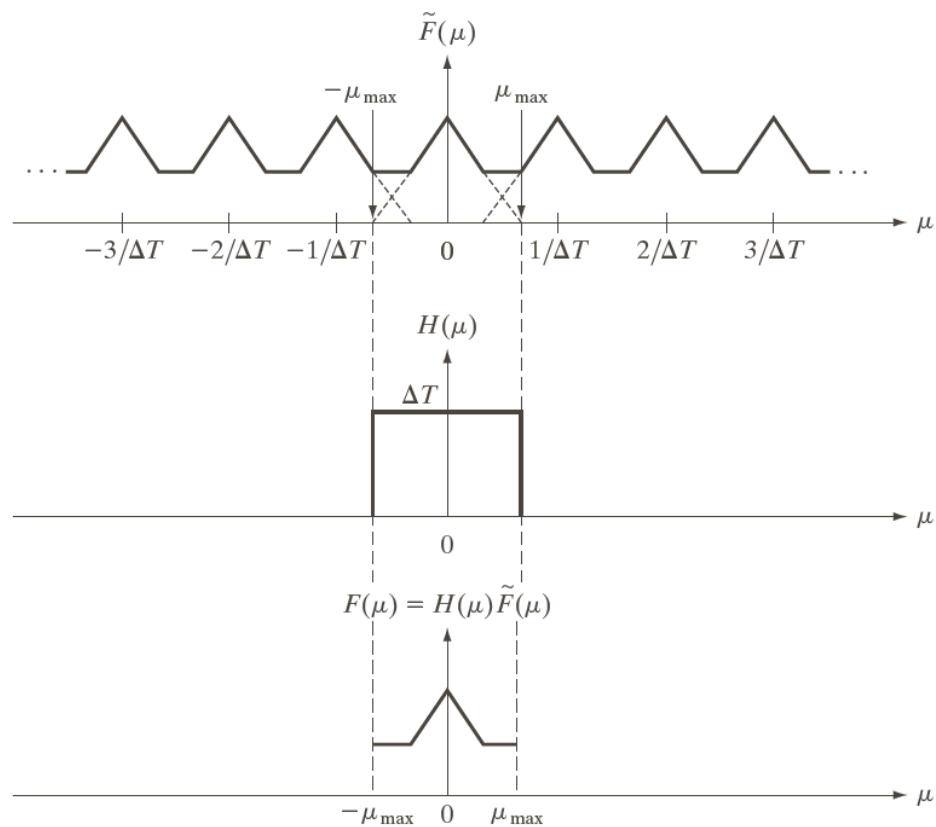
**奈奎斯特采样率：**  
完全等于信号最高频率的两倍的  
取样率。



a  
b  
c  
d

**FIGURE 4.6**  
(a) Fourier transform of a band-limited function.  
(b)–(d) Transforms of the corresponding sampled function under the conditions of over-sampling, critically-sampling, and under-sampling, respectively.

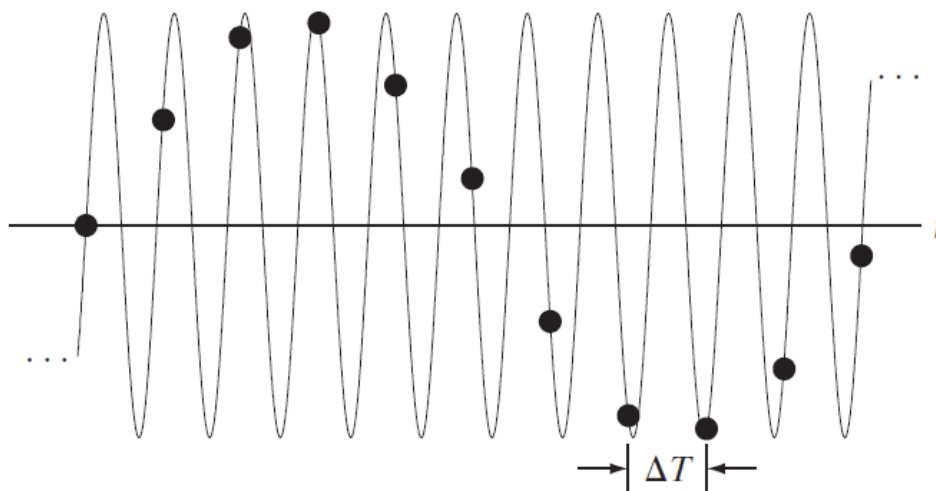
# 混淆



a  
b  
c

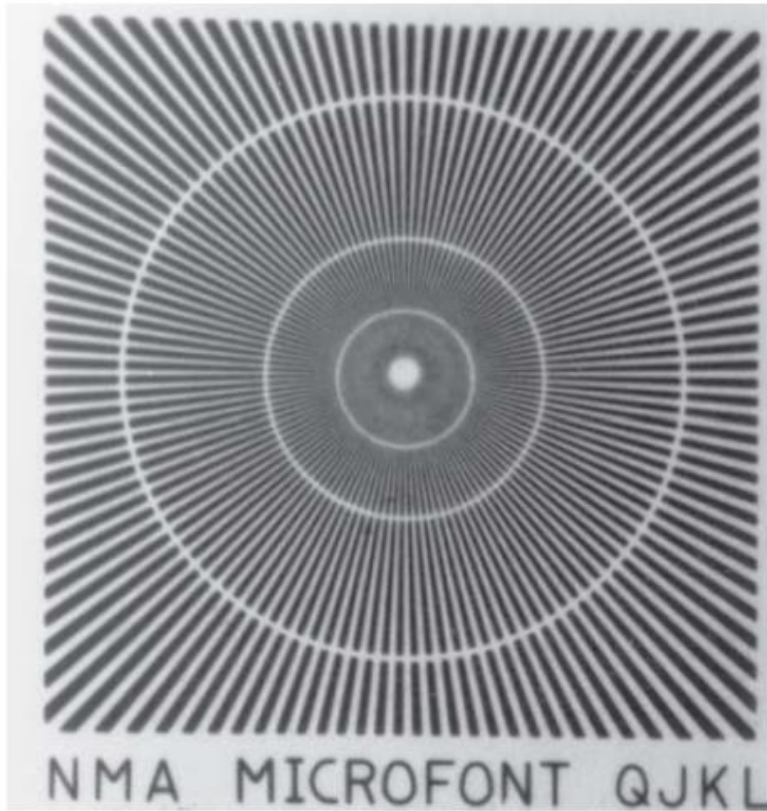
**FIGURE 4.9** (a) Fourier transform of an under-sampled, band-limited function. (Interference from adjacent periods is shown dashed in this figure). (b) The same ideal lowpass filter used in Fig. 4.8(b). (c) The product of (a) and (b). The interference from adjacent periods results in aliasing that prevents perfect recovery of  $F(\mu)$  and, therefore, of the original, band-limited continuous function. Compare with Fig. 4.8.

# 混淆

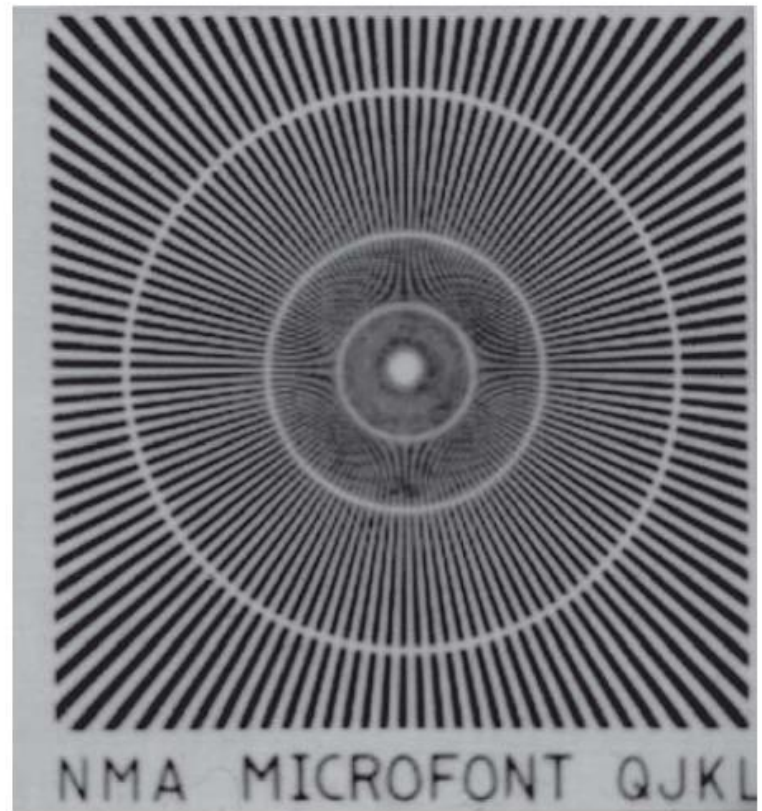


**FIGURE 4.10** Illustration of aliasing. The under-sampled function (black dots) looks like a sine wave having a frequency much lower than the frequency of the continuous signal. The period of the sine wave is 2 s, so the zero crossings of the horizontal axis occur every second.  $\Delta T$  is the separation between samples.

# 数字图像中的混淆

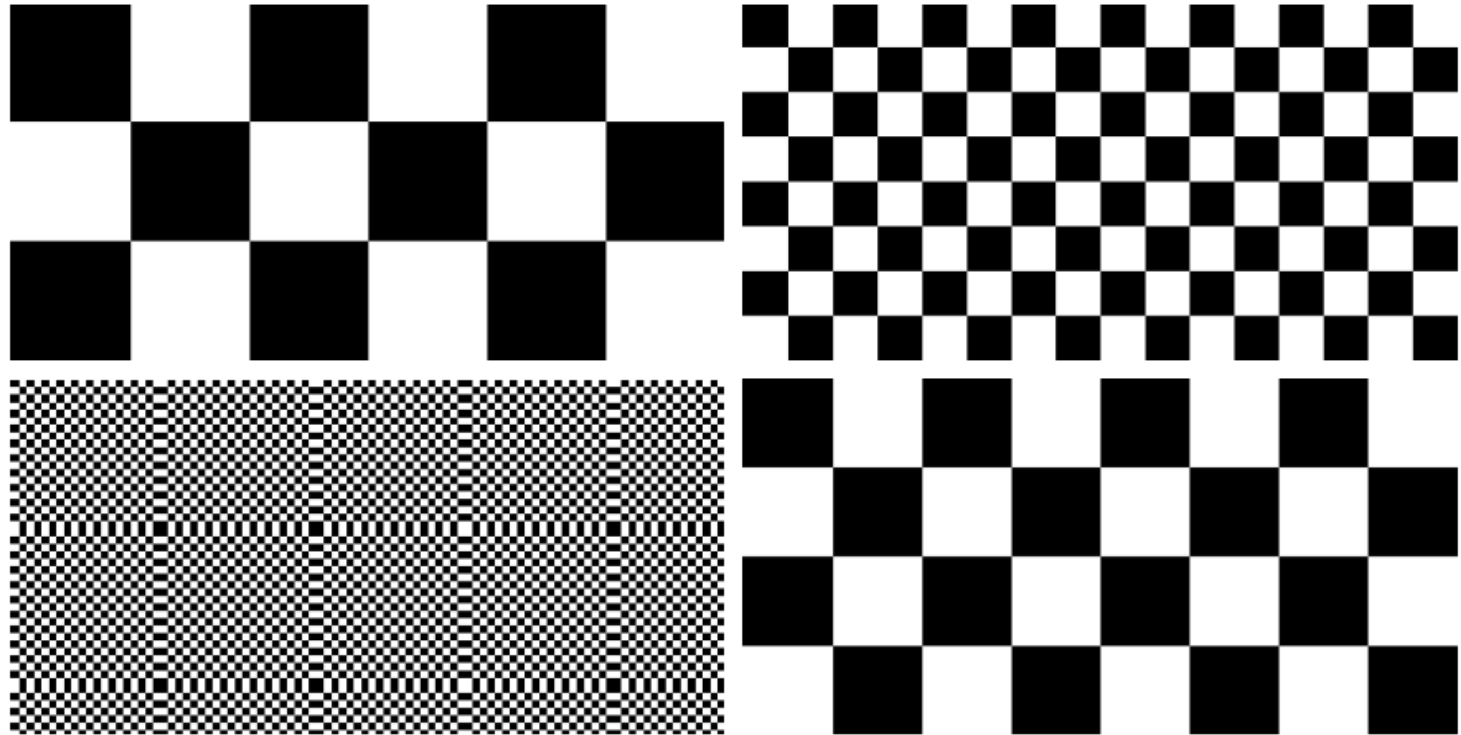


(a) Original image



(b) Sampled image

# 数字图像中的混淆



a	b
c	d

**FIGURE 4.16** Aliasing in images. In (a) and (b), the lengths of the sides of the squares are 16 and 6 pixels, respectively, and aliasing is visually negligible. In (c) and (d), the sides of the squares are 0.9174 and 0.4798 pixels, respectively, and the results show significant aliasing. Note that (d) masquerades as a “normal” image.



# 数字图像中的混淆

## □ 图像内插和重采样

- 放大可视为过采样，缩写可视为欠采样



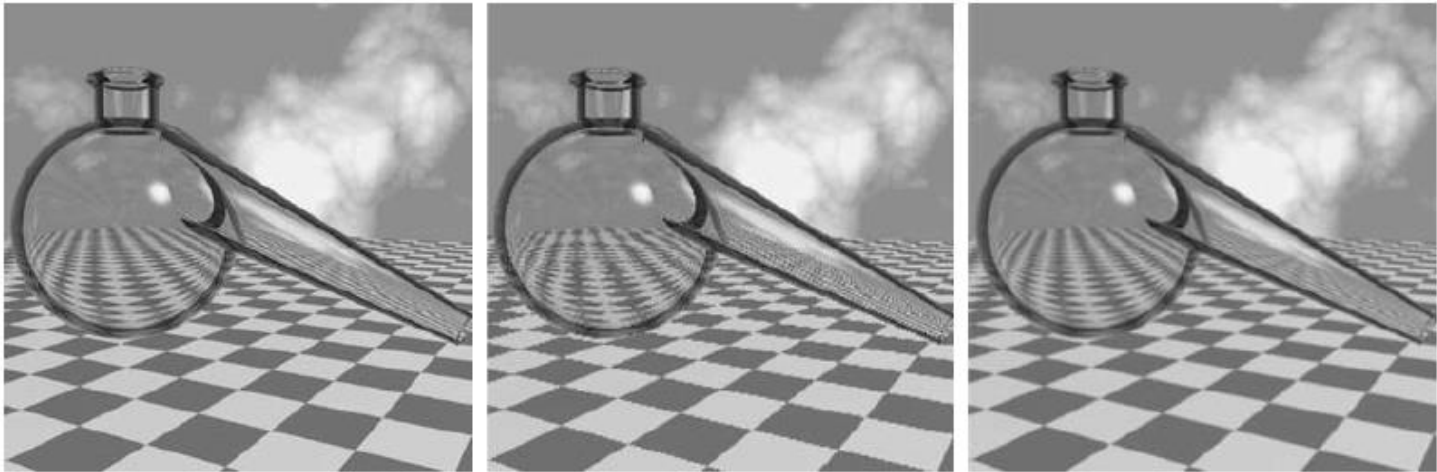
a b c

**FIGURE 4.17** Illustration of aliasing on resampled images. (a) A digital image with negligible visual aliasing. (b) Result of resizing the image to 50% of its original size by pixel deletion. Aliasing is clearly visible. (c) Result of blurring the image in (a) with a  $3 \times 3$  averaging filter prior to resizing. The image is slightly more blurred than (b), but aliasing is no longer objectionable. (Original image courtesy of the Signal Compression Laboratory, University of California, Santa Barbara.)



# 数字图像中的混淆

## □ 图像内插和重采样

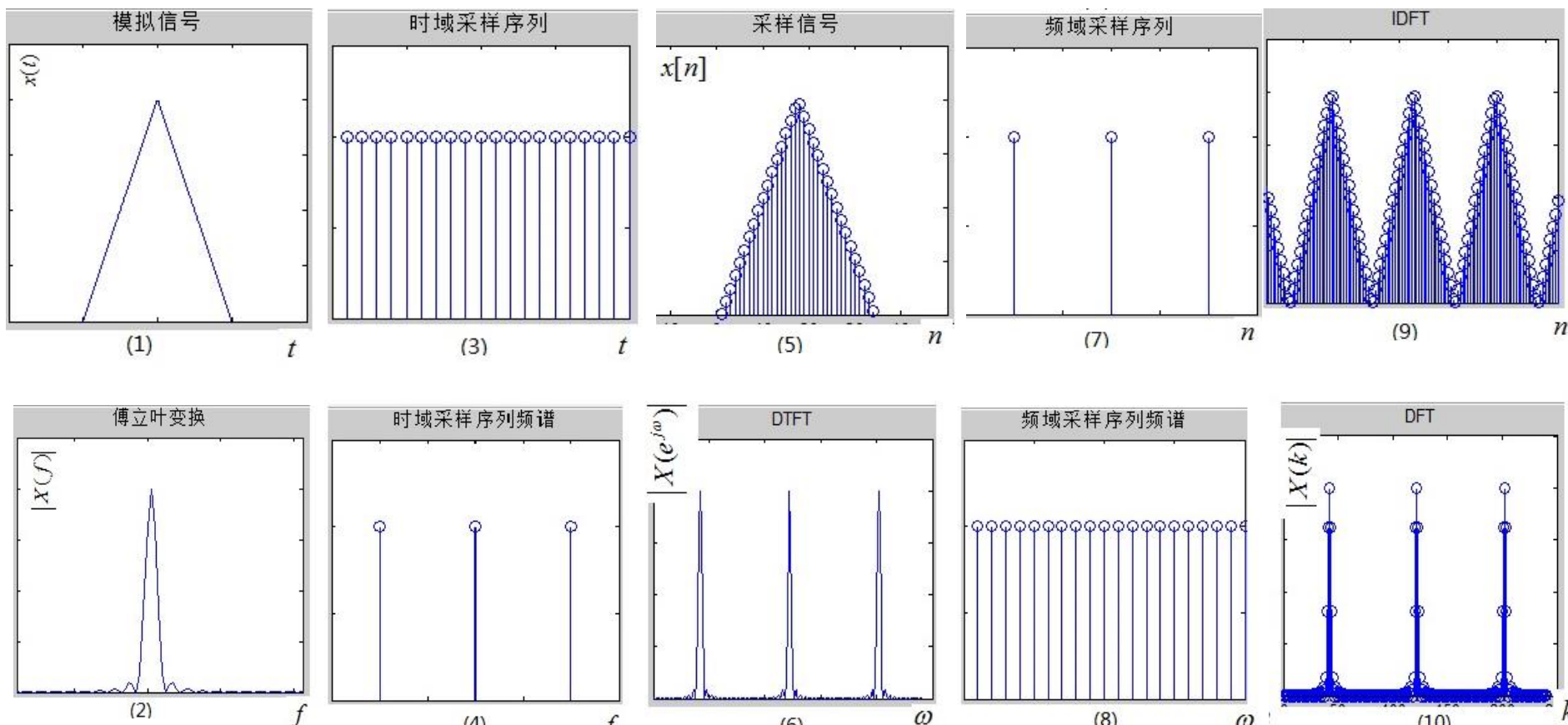


a b c

**FIGURE 4.18** Illustration of jaggies. (a) A  $1024 \times 1024$  digital image of a computer-generated scene with negligible visible aliasing. (b) Result of reducing (a) to 25% of its original size using bilinear interpolation. (c) Result of blurring the image in (a) with a  $5 \times 5$  averaging filter prior to resizing it to 25% using bilinear interpolation. (Original image courtesy of D. P. Mitchell, Mental Landscape, LLC.)

# 4.1 傅立叶变换基础

- 傅里叶变换 (FT)、离散时间傅里叶变换 (DTFT) 和离散傅里叶变换 (DFT) 之间的关系
  - DFT: 对一个周期取样





# 4.1 离散傅立叶变换(DFT)

## 二维DFT及其反变换

一个图像关于的函数的离散

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

$$u = 0.1.2.\dots M - 1$$

$$v = 0.1.2.\dots N - 1$$

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

$$x = 0.1.2.\dots M - 1$$

$$y = 0.1.2.\dots N - 1$$

同理:

$$\text{谱: } |F(u, v)| = \left[ K^2(x, y) + I^2(x, y) \right]^{\frac{1}{2}}$$

$$\text{相角: } F(u, v) = \arctg \left[ \frac{I(u, v)}{R(u, v)} \right]$$

功率谱:

$$P(u, v) = |F(u, v)|^2 = R^2(u, v) + I^2(u, v)$$

可以证明:

$$\wp \left[ f(x, y) (-1)^{x+y} \right] = F \left( u - \frac{M}{2}, v - \frac{N}{2} \right)$$

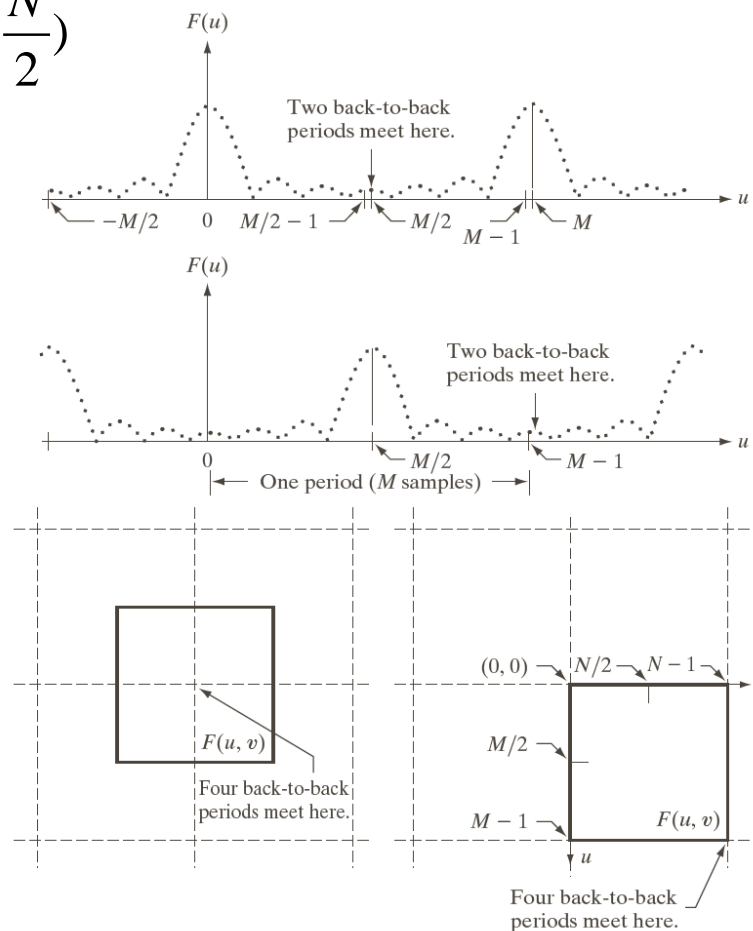
原点被放置在  $u = \frac{M}{2}, v = \frac{N}{2}$  上

$$F(0, 0) = MN \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) = MN \bar{f}$$

$\bar{f}$  为图象平均灰度。

# 2D DFT的性质：周期性

$$\mathcal{F}\left[f(x, y)(-1)^{x+y}\right] = F\left(u - \frac{M}{2}, v - \frac{N}{2}\right)$$



a  
b  
c d

**FIGURE 4.23** Centering the Fourier transform. (a) A 1-D DFT showing an infinite number of periods. (b) Shifted DFT obtained by multiplying  $f(x)$  by  $(-1)^x$  before computing  $F(u)$ . (c) A 2-D DFT showing an infinite number of periods. The solid area is the  $M \times N$  data array,  $F(u, v)$ , obtained with Eq. (4.5-15). This array consists of four quarter periods. (d) A Shifted DFT obtained by multiplying  $f(x, y)$  by  $(-1)^{x+y}$  before computing  $F(u, v)$ . The data now contains one complete, centered period, as in (b).

$\square$  = Periods of the DFT.  
 $\blacksquare$  =  $M \times N$  data array,  $F(u, v)$ .



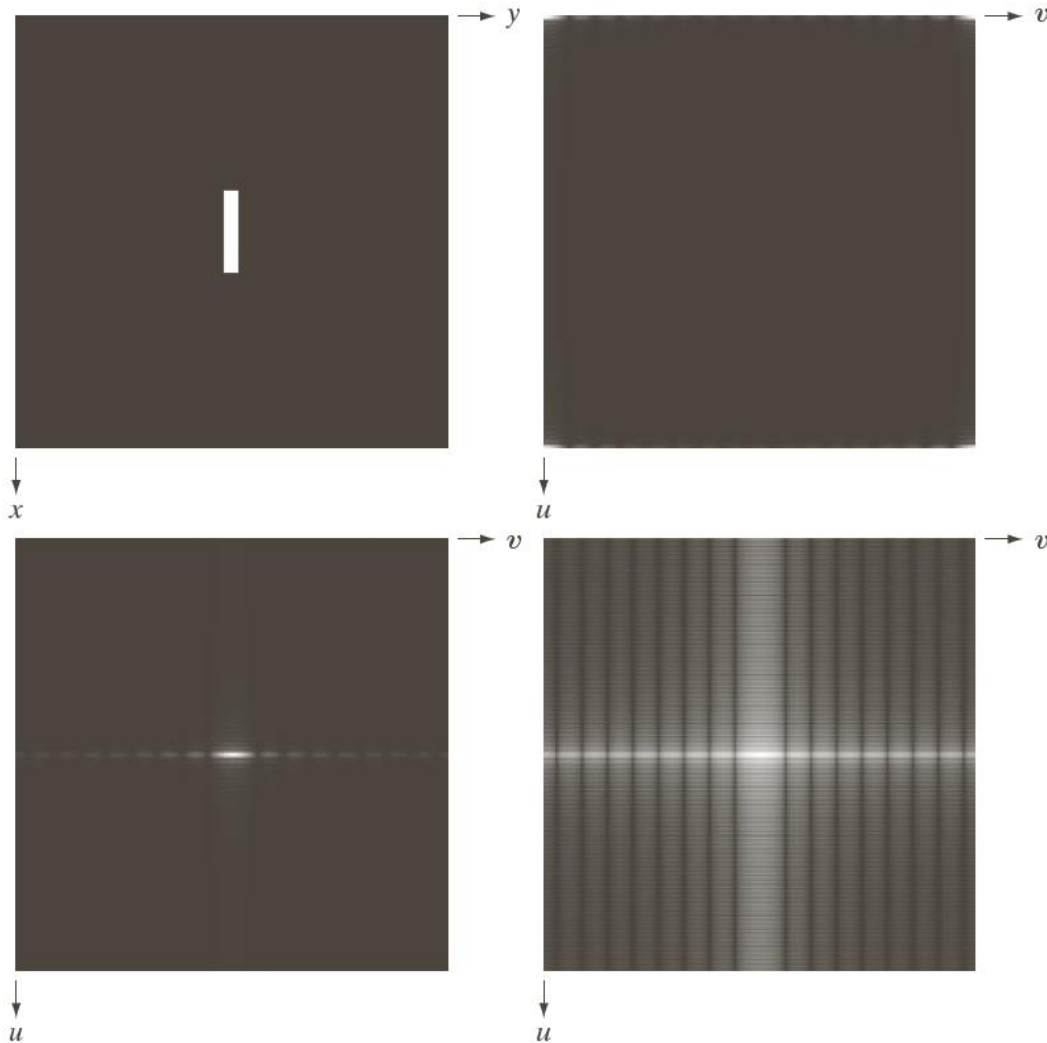
# 2D DFT及其反变换的一些性质

Spatial Domain <sup>†</sup>			Frequency Domain <sup>†</sup>	
1)	$f(x, y)$ real	$\Leftrightarrow$	$F^*(u, v) = F(-u, -v)$	
2)	$f(x, y)$ imaginary	$\Leftrightarrow$	$F^*(-u, -v) = -F(u, v)$	
3)	$f(x, y)$ real	$\Leftrightarrow$	$R(u, v)$ even; $I(u, v)$ odd	
4)	$f(x, y)$ imaginary	$\Leftrightarrow$	$R(u, v)$ odd; $I(u, v)$ even	
5)	$f(-x, -y)$ real	$\Leftrightarrow$	$F^*(u, v)$ complex	
6)	$f(-x, -y)$ complex	$\Leftrightarrow$	$F(-u, -v)$ complex	
7)	$f^*(x, y)$ complex	$\Leftrightarrow$	$F^*(-u - v)$ complex	
8)	$f(x, y)$ real and even	$\Leftrightarrow$	$F(u, v)$ real and even	
9)	$f(x, y)$ real and odd	$\Leftrightarrow$	$F(u, v)$ imaginary and odd	
10)	$f(x, y)$ imaginary and even	$\Leftrightarrow$	$F(u, v)$ imaginary and even	
11)	$f(x, y)$ imaginary and odd	$\Leftrightarrow$	$F(u, v)$ real and odd	
12)	$f(x, y)$ complex and even	$\Leftrightarrow$	$F(u, v)$ complex and even	
13)	$f(x, y)$ complex and odd	$\Leftrightarrow$	$F(u, v)$ complex and odd	

**TABLE 4.1** Some symmetry properties of the 2-D DFT and its inverse.  $R(u, v)$  and  $I(u, v)$  are the real and imaginary parts of  $F(u, v)$ , respectively. The term *complex* indicates that a function has nonzero real and imaginary parts.

<sup>†</sup>Recall that  $x, y, u,$  and  $v$  are *discrete* (integer) variables, with  $x$  and  $u$  in the range  $[0, M - 1]$ , and  $y,$  and  $v$  in the range  $[0, N - 1]$ . To say that a complex function is *even* means that its real *and* imaginary parts are even, and similarly for an odd complex function.

# 数字图像的DFT变换



a	b
c	d

**FIGURE 4.24**

(a) Image. (b) Spectrum showing bright spots in the four corners. (c) Centered spectrum. (d) Result showing increased detail after a log transformation. The zero crossings of the spectrum are closer in the vertical direction because the rectangle in (a) is longer in that direction. The coordinate convention used throughout the book places the origin of the spatial and frequency domains at the top left.

# 数字图像的DFT变换

## 二维DFT及其反变换

下图实例：

中心化，矩形宽高化为 2 : 1

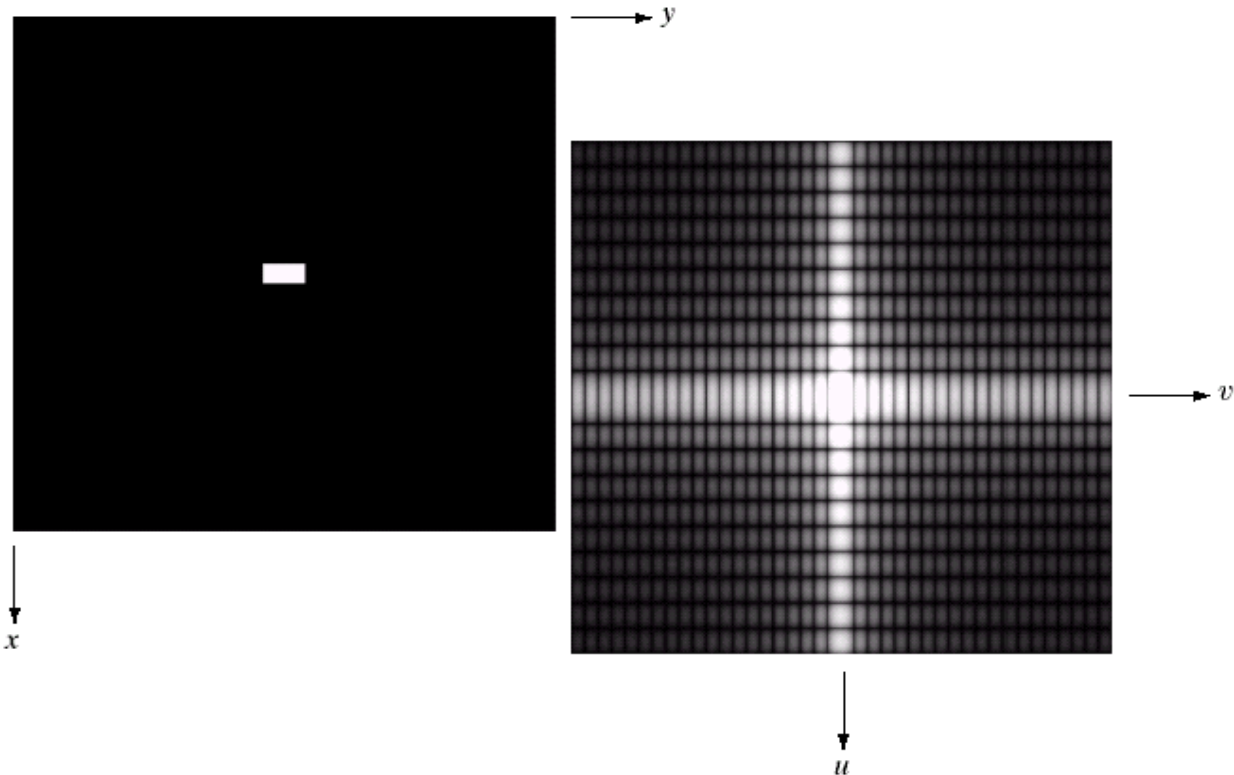
反映到频域轴亮点间距恰好相反

a b

**FIGURE 4.3**

(a) Image of a  $20 \times 40$  white rectangle on a black background of size  $512 \times 512$  pixels.

(b) Centered Fourier spectrum shown after application of the log transformation given in Eq. (3.2-2). Compare with Fig. 4.2.



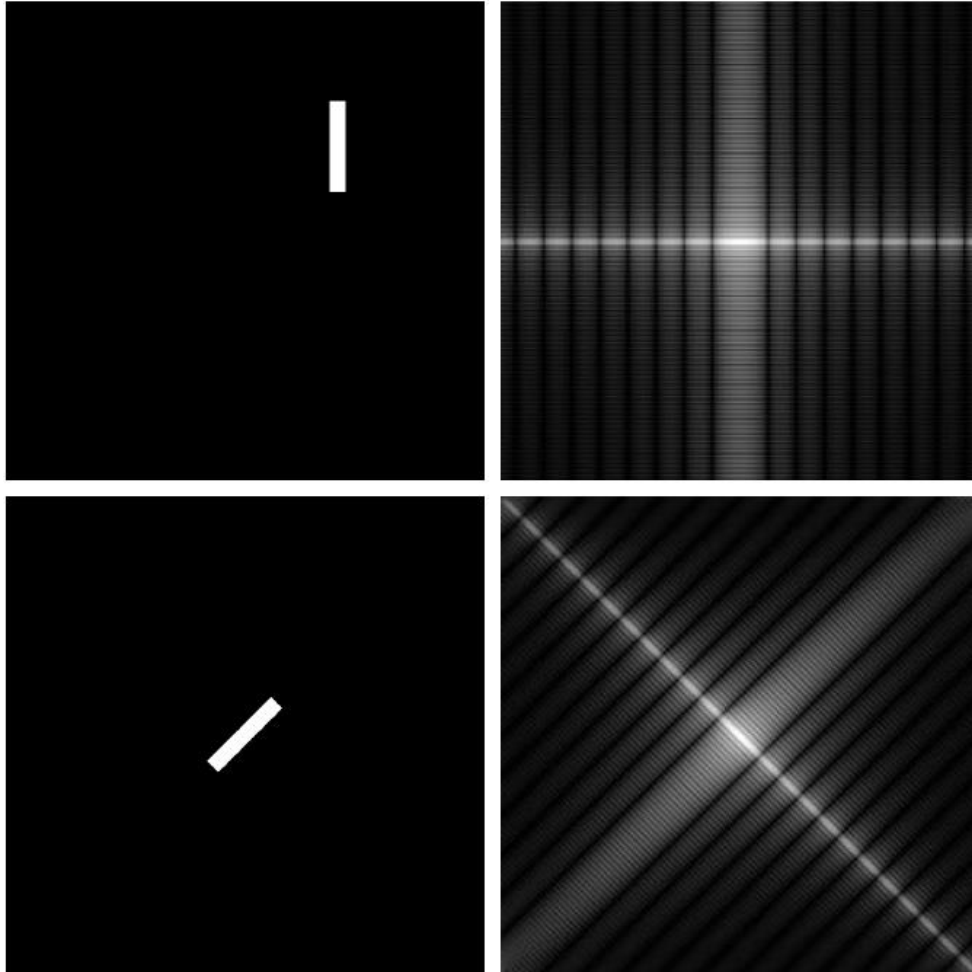


# 数字图像的DFT变换

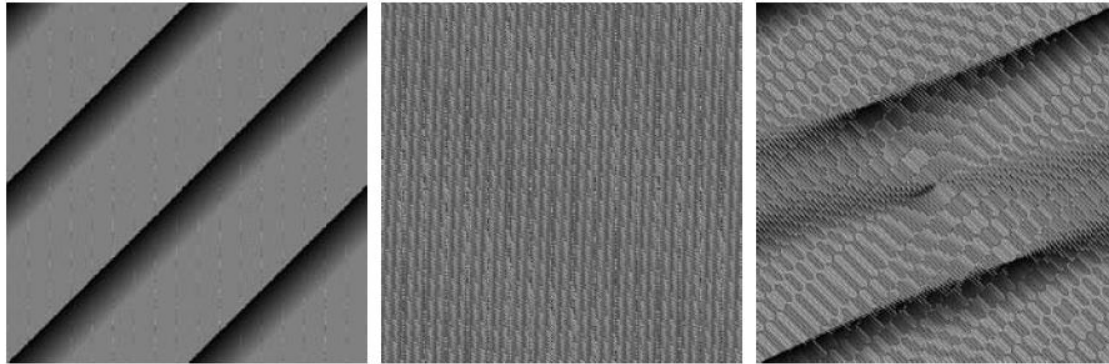
a	b
c	d

**FIGURE 4.25**

(a) The rectangle in Fig. 4.24(a) translated, and (b) the corresponding spectrum. (c) Rotated rectangle, and (d) the corresponding spectrum. The spectrum corresponding to the translated rectangle is identical to the spectrum corresponding to the original image in Fig. 4.24(a).



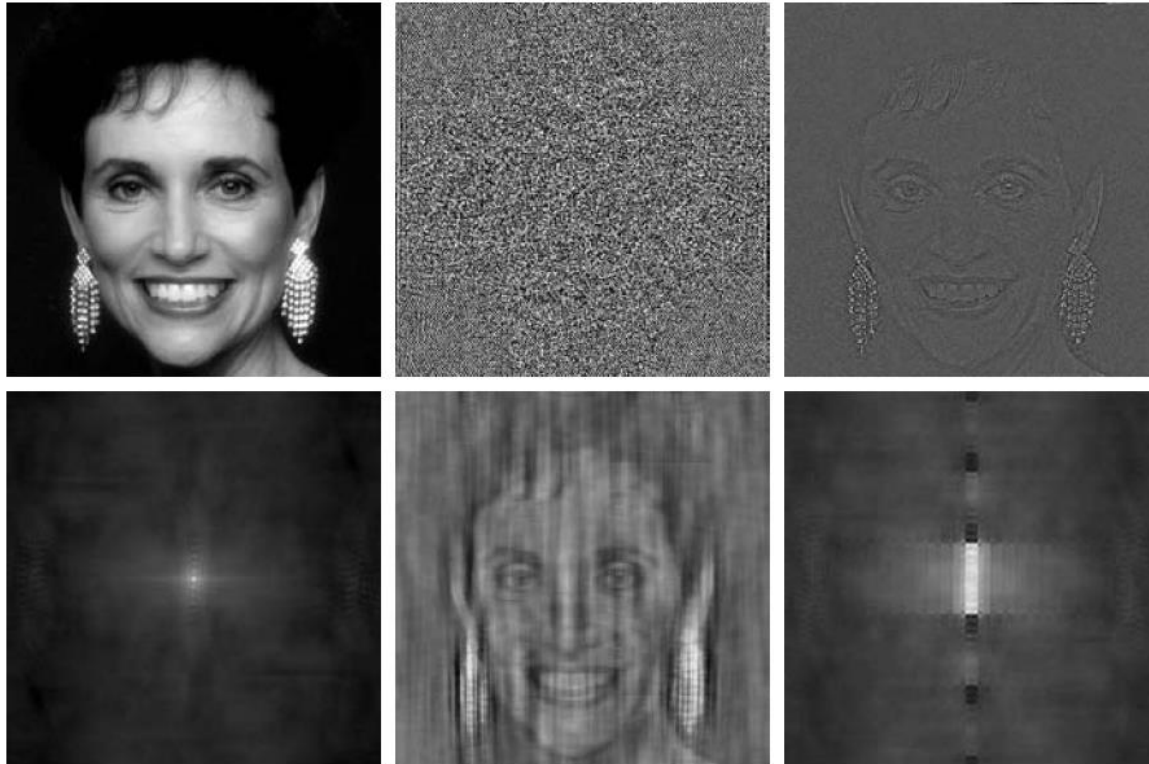
# 数字图像的DFT变换



a b c

**FIGURE 4.26** Phase angle array corresponding (a) to the image of the centered rectangle in Fig. 4.24(a), (b) to the translated image in Fig. 4.25(a), and (c) to the rotated image in Fig. 4.25(c).

# 基于二维DFT频谱与相位的反变换



a	b	c
d	e	f

**FIGURE 4.27** (a) Woman. (b) Phase angle. (c) Woman reconstructed using only the phase angle. (d) Woman reconstructed using only the spectrum. (e) Reconstruction using the phase angle corresponding to the woman and the spectrum corresponding to the rectangle in Fig. 4.24(a). (f) Reconstruction using the phase of the rectangle and the spectrum of the woman.

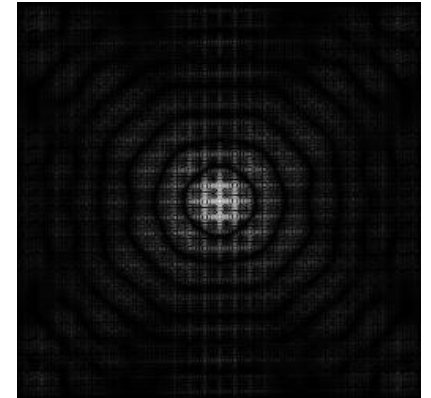
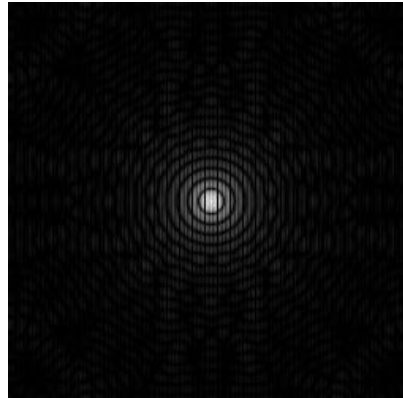
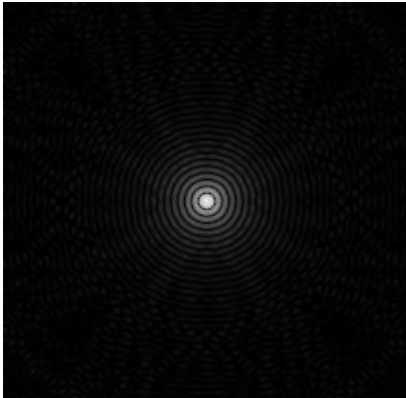
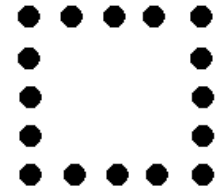


# 二维DFT的性质

- **线性**      $f_1(x, y) + f_2(x, y)$       $F_1(u, v) + F_2(u, v)$
- **比例**      $f(ax, by)$       $\frac{1}{ab} F\left(\frac{u}{a}, \frac{v}{b}\right)$
- **平移**      $f(x - a, y - b)$       $e^{-j2\pi(au+bv)} F(u, v)$   
                   $e^{j2\pi(cx+dy)} f(x, y)$       $F(u - c, v - d)$
- **卷积**      $f_1(x, y) * f_2(x, y)$       $F_1(u, v)F_2(u, v)$   
                   $f_1(x, y)f_2(x, y)$       $F_1(u, v) * F_2(u, v)$
- **旋转**      $f(x \cos \theta + y \sin \theta, -x \sin \theta + y \cos \theta)$   
                   $F(u \cos \theta + v \sin \theta, -u \sin \theta + v \cos \theta)$

# 4.1 离散傅立叶变换(DFT)

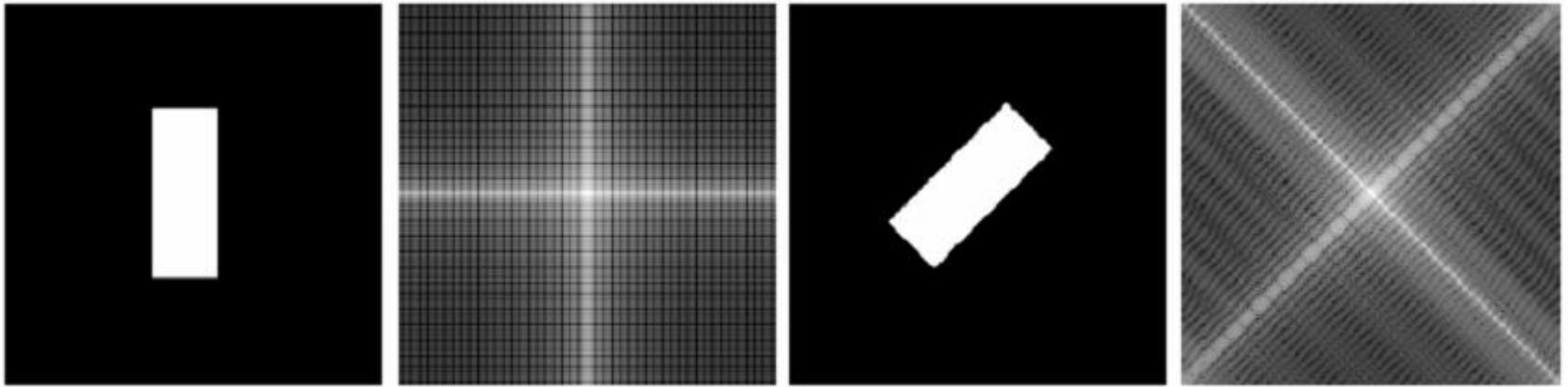
## 线性叠加及尺度变化





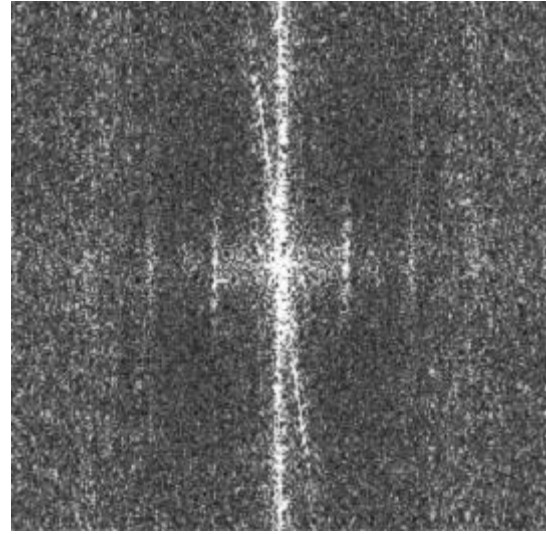
# 4.1 离散傅立叶变换(DFT)

## 旋转性





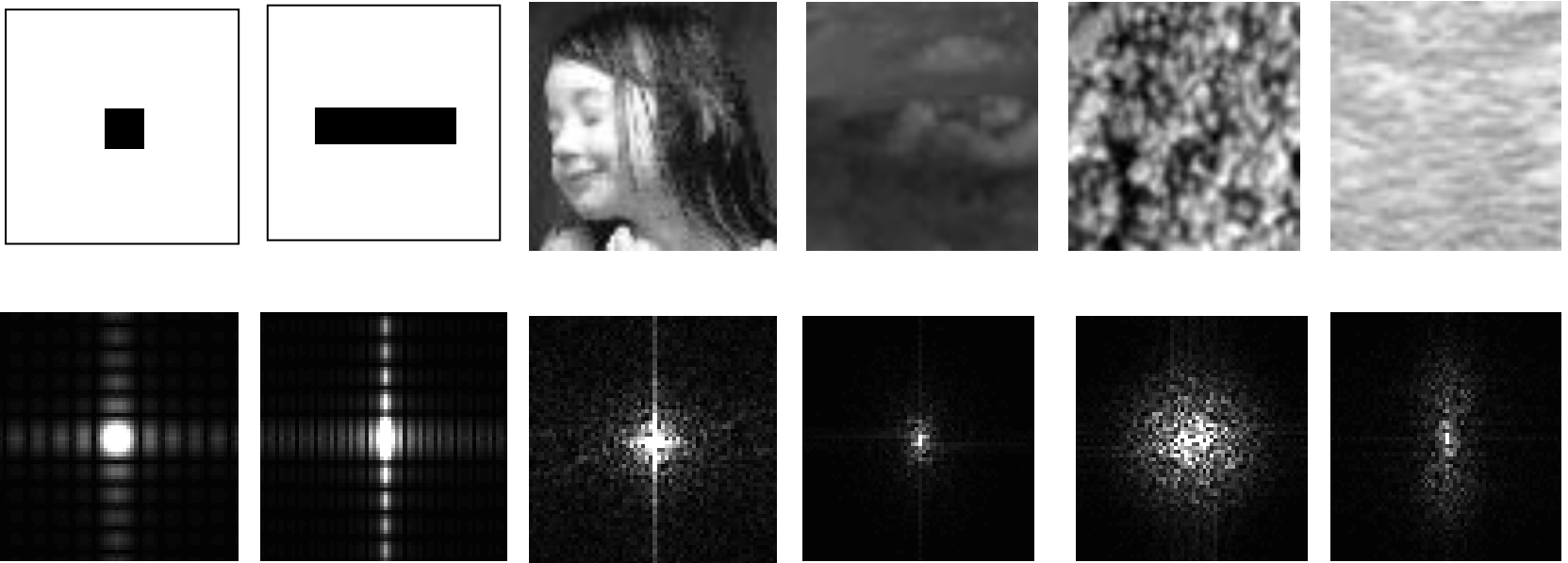
# 4.1 离散傅立叶变换(DFT)



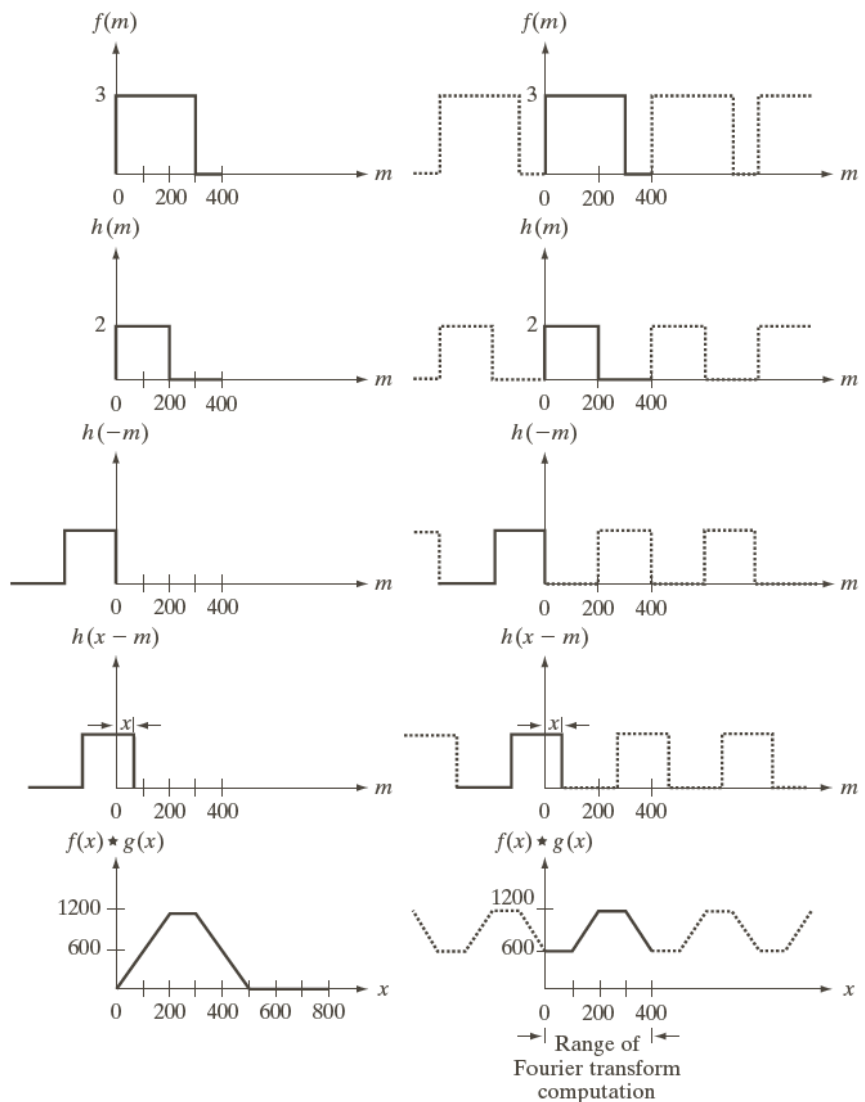


# 4.1 离散傅立叶变换(DFT)

## 典型图象的频谱



# 空域卷积 vs. DFT



a f  
b g  
c h  
d i  
e j

**FIGURE 4.28** Left column: convolution of two discrete functions obtained using the approach discussed in Section 3.4.2. The result in (e) is correct. Right column: Convolution of the same functions, but taking into account the periodicity implied by the DFT. Note in (j) how data from adjacent periods produce wraparound error, yielding an incorrect convolution result. To obtain the correct result, function padding must be used.



# DFT定义及相关表达式小结

Name	Expression(s)
1) Discrete Fourier transform (DFT) of $f(x, y)$	$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M+vy/N)}$
2) Inverse discrete Fourier transform (IDFT) of $F(u, v)$	$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M+vy/N)}$
3) Polar representation	$F(u, v) =  F(u, v)  e^{j\phi(u, v)}$
4) Spectrum	$ F(u, v)  = [R^2(u, v) + I^2(u, v)]^{1/2}$ $R = \text{Real}(F); \quad I = \text{Imag}(F)$
5) Phase angle	$\phi(u, v) = \tan^{-1} \left[ \frac{I(u, v)}{R(u, v)} \right]$
6) Power spectrum	$P(u, v) =  F(u, v) ^2$
7) Average value	$\bar{f}(x, y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) = \frac{1}{MN} F(0, 0)$
8) Periodicity ( $k_1$ and $k_2$ are integers)	$F(u, v) = F(u + k_1M, v) = F(u, v + k_2N)$ $= F(u + k_1M, v + k_2N)$ $f(x, y) = f(x + k_1M, y) = f(x, y + k_2N)$ $= f(x + k_1M, y + k_2N)$
9) Convolution	$f(x, y) \star h(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) h(x - m, y - n)$
10) Correlation	$f(x, y) \star\star h(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f^*(m, n) h(x + m, y + n)$
11) Separability	The 2-D DFT can be computed by computing 1-D DFT transforms along the rows (columns) of the image, followed by 1-D transforms along the columns (rows) of the result. See Section 4.11.1.
12) Obtaining the inverse Fourier transform using a forward transform algorithm.	$MNf^*(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F^*(u, v) e^{-j2\pi(ux/M+vy/N)}$ This equation indicates that inputting $F^*(u, v)$ into an algorithm that computes the forward transform (right side of above equation) yields $MNf^*(x, y)$ . Taking the complex conjugate and dividing by $MN$ gives the desired inverse. See Section 4.11.2.



# 第4章 频率域滤波

---

- 4.1 离散傅立叶变换(DFT)
- 4.2 频率域滤波基础
- 4.3 频率域滤波器平滑图像
- 4.4 频率域滤波器锐化图像

## 4.2 频率域滤波基础

频率域滤波基本步骤:

- 1、  $(-1)^{x+y} \times$  原图像
  - 2、  $F(u, v)$
  - 3、  $H(u, v) \times F(u, v)$
  - 4、 反DEF
  - 5、 实部
  - 6、 用  $(-1)^{x+y} \times$  (5) 结果。
- 被滤波图像 =  $\mathcal{F}^{-1}[G(u, v)]$

$$G(u, v) = H(u, v)F(u, v)$$

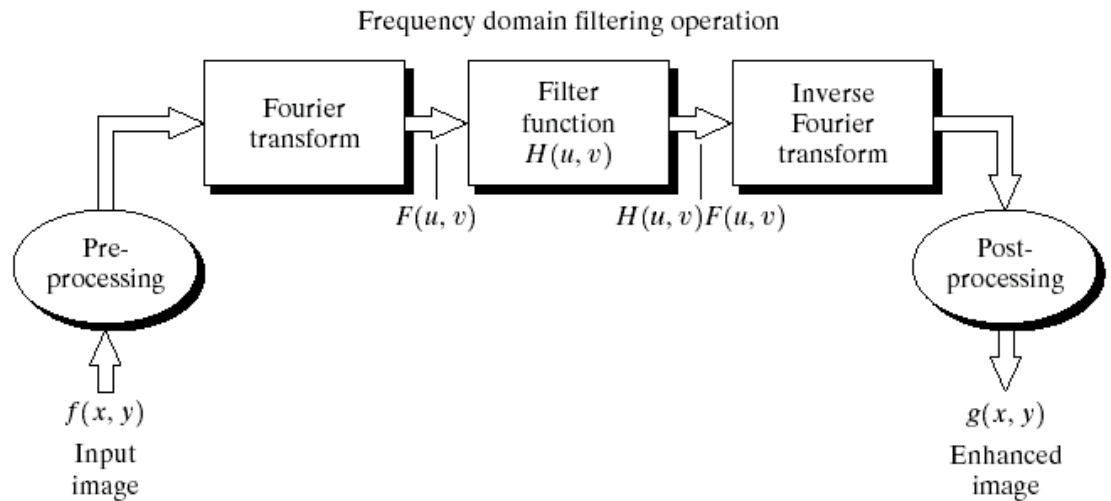
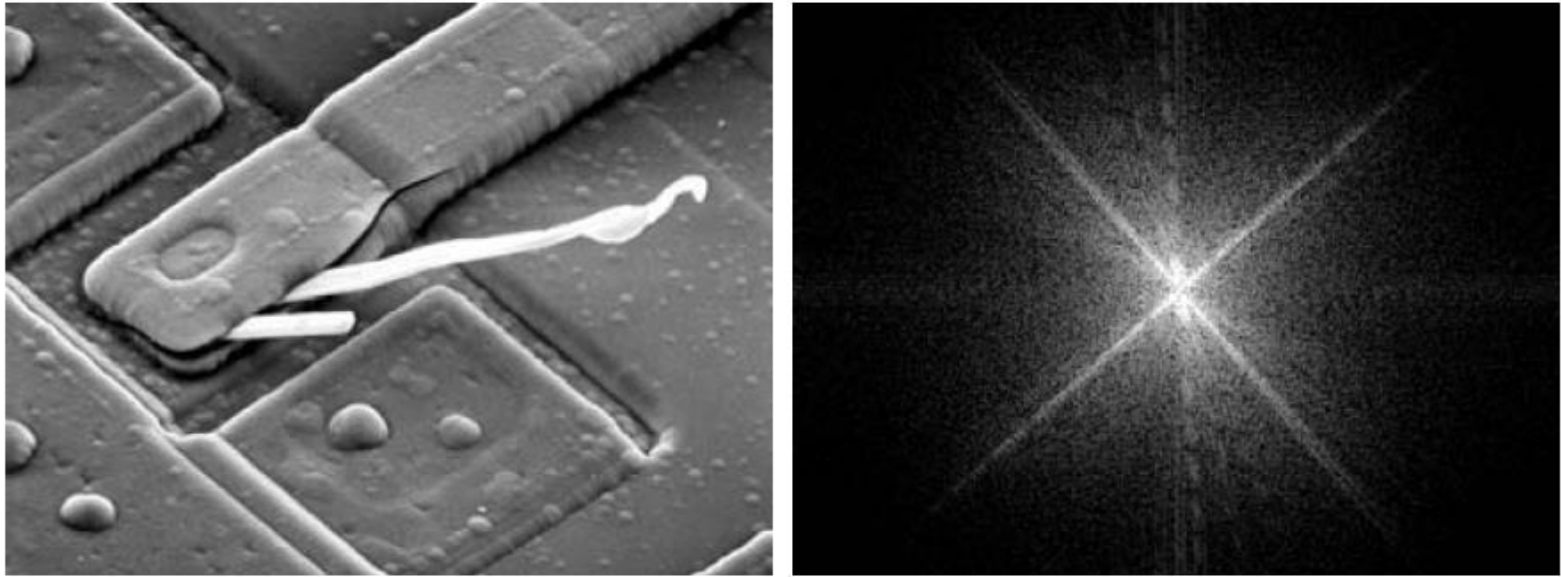


FIGURE 4.5 Basic steps for filtering in the frequency domain.

# 图像空域与频域的定性分析



a b

**FIGURE 4.29** (a) SEM image of a damaged integrated circuit. (b) Fourier spectrum of (a). (Original image courtesy of Dr. J. M. Hudak, Brockhouse Institute for Materials Research, McMaster University, Hamilton, Ontario, Canada.)

## 4.2 频率域滤波基础

一些基本的滤波器及其性质：

陷波滤波器： $F(0,0) \rightarrow$  平均值

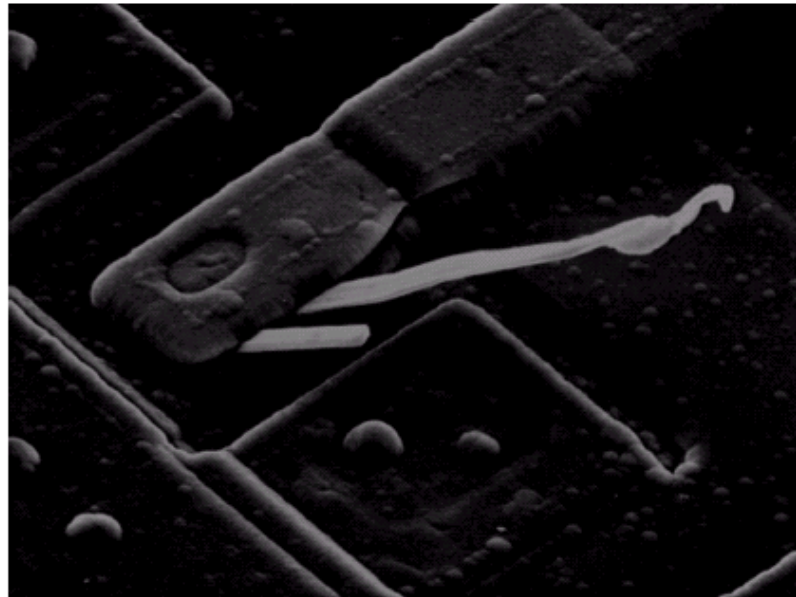
如果想使其平均灰度为0，则可：

$$H(u, v) = \begin{cases} 0 & (u, v) = (\frac{M}{2}, \frac{N}{2}) \\ 1 & \text{其它} \end{cases}$$

具体见如下图示：

**FIGURE 4.6**

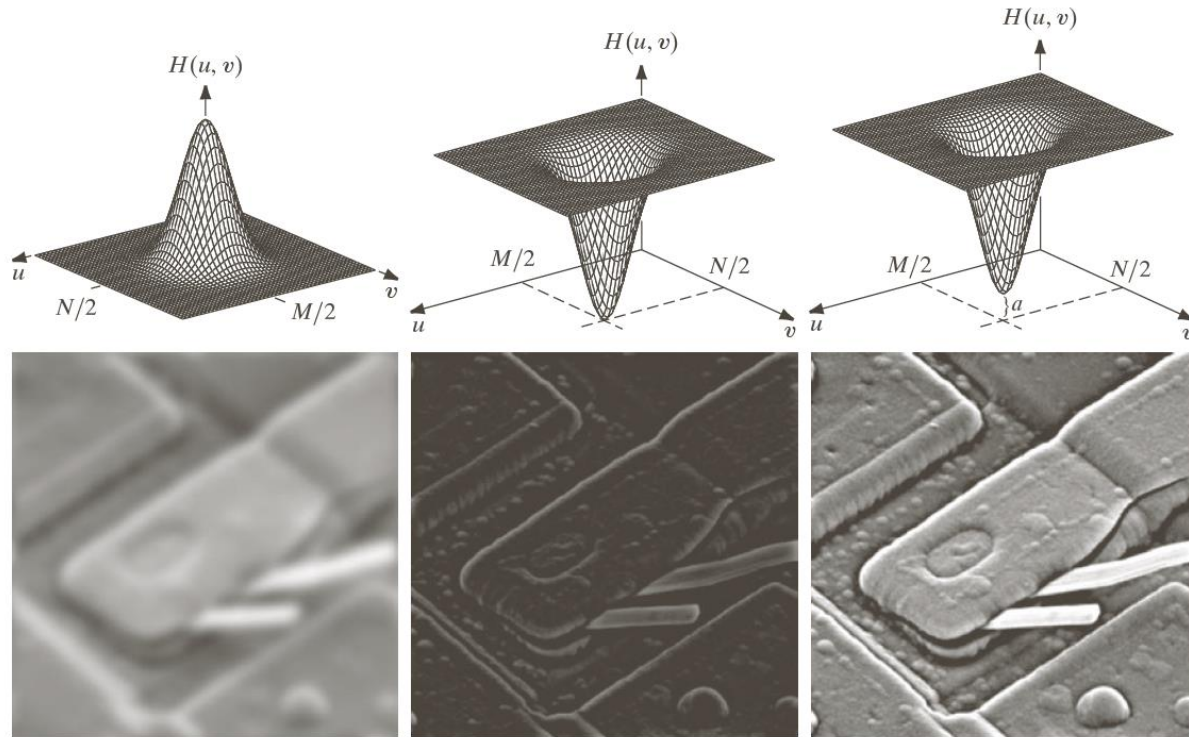
Result of filtering the image in Fig. 4.4(a) with a notch filter that set to 0 the  $F(0, 0)$  term in the Fourier transform.





# 4.2 频率域滤波基础

## □ 高通与低通滤波



a b c  
d e f

**FIGURE 4.31** Top row: frequency domain filters. Bottom row: corresponding filtered images obtained using Eq. (4.7-1). We used  $a = 0.85$  in (c) to obtain (f) (the height of the filter itself is 1). Compare (f) with Fig. 4.29(a).

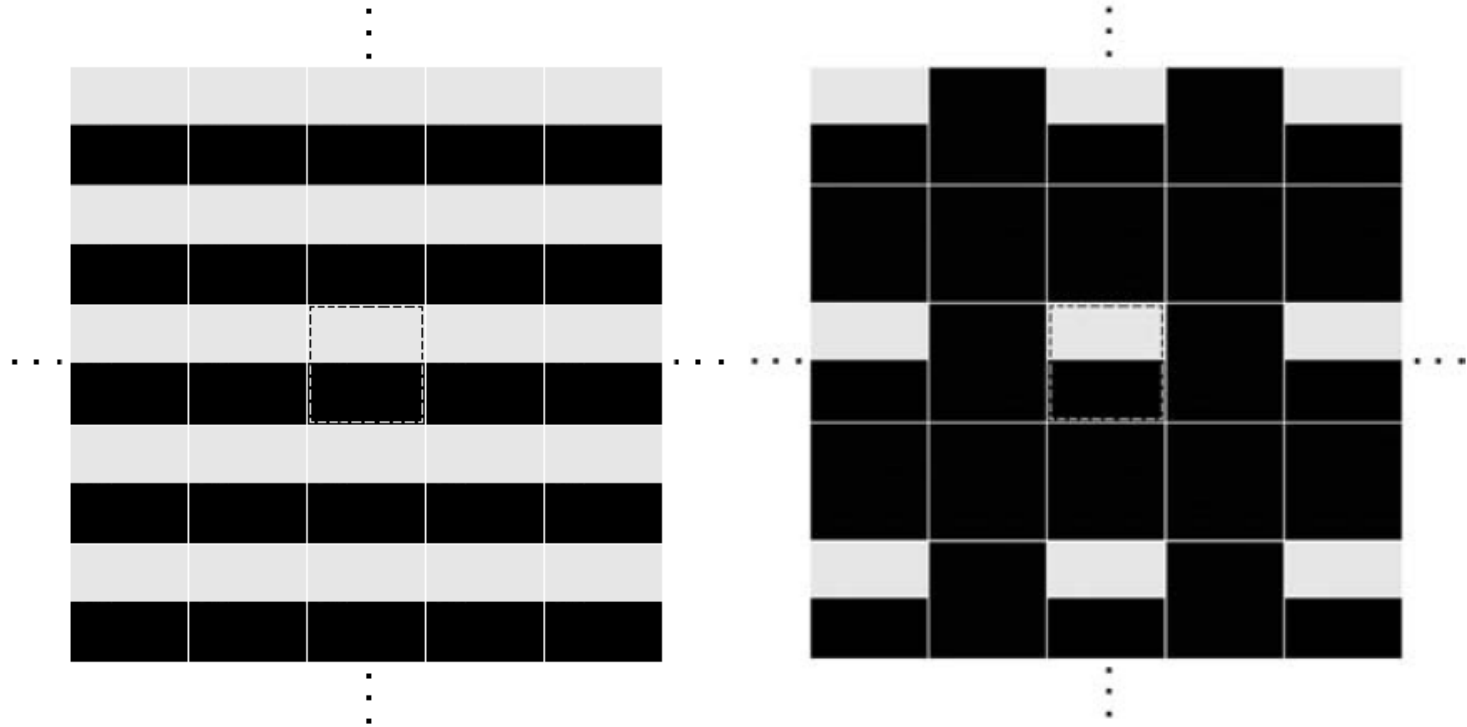
# 使用DFT时二维图像固有的周期性



a b c

**FIGURE 4.32** (a) A simple image. (b) Result of blurring with a Gaussian lowpass filter without padding. (c) Result of lowpass filtering with padding. Compare the light area of the vertical edges in (b) and (c).

# 使用DFT时二维图像固有的周期性



a b

**FIGURE 4.33** 2-D image periodicity inherent in using the DFT. (a) Periodicity without image padding. (b) Periodicity after padding with 0s (black). The dashed areas in the center correspond to the image in Fig. 4.32(a). (The thin white lines in both images are superimposed for clarity; they are not part of the data.)



# 空间和频率域滤波间的对应

## 空间域滤波和频率域滤波之间的对应关系

基本联系就是卷积定理

离散卷积:

$$f(x, y) * h(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) h(x-m, y-n)$$

$$f(x, y) * h(x, y) \Leftrightarrow F(u, v) H(u, v)$$

$$f(x, y) h(x, y) \Leftrightarrow F(u, v) * H(u, v)$$

在  $(x_0, y_0)$  处, 强度为A的冲激函数  $A\delta(x-x_0, y-y_0)$

并且:

$$\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} S(x, y) A\delta(x-x_0, y-y_0) = AS(x_0, y_0)$$

把  $A\delta(x-x_0, y-y_0)$  看作  $-M \times N$  大小的图像, 在  $(x_0, y_0)$  有值, 其它全为0。

$$\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} S(x, y) \delta(x, y) = S(0, 0)$$



# 空间和频率域滤波间的对应

单位冲激的傅氏变换:

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \delta(x, y) \zeta^{-2\pi(ux/M + vy/N)}$$
$$= \frac{1}{MN}$$

可得到:  $f(x, y) = \delta(x, y)$ ,

令

$$f(x, y) * h(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \delta(m, n) h(x - m, y - n)$$
$$= \frac{1}{MN} h(x, y)$$

可以输入  $\delta(x, y)$  出  $h(x, y)$

**结论: 空间域和频率域中的滤波器组成了一个傅立叶变换对。一般空间域更适合使用小的滤波器。**

# 空间和频率域滤波间的对应

## 空间域滤波和频率域滤波之间的对应关系

例如：高斯滤波函数

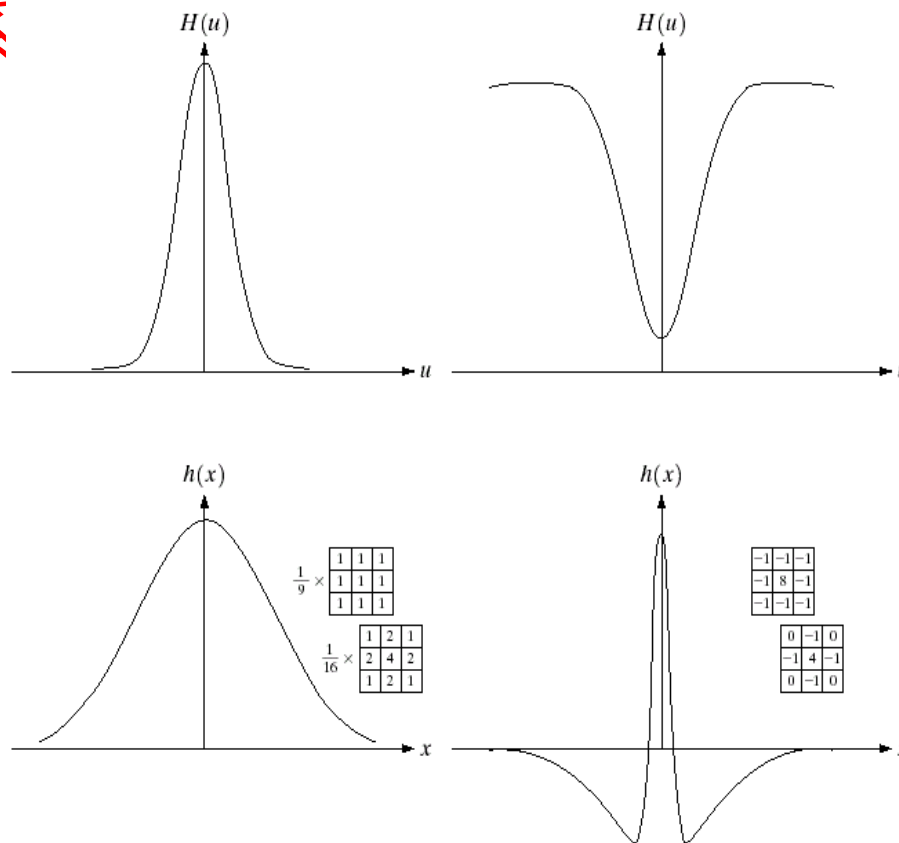
$$H(u) = Ae^{-u^2/2\sigma^2}$$

$\sigma$  为标准差：

$$h(x) = \sqrt{2\pi}\sigma Ae^{-2\pi^2\sigma^2x^2}$$

傅立叶变换及其反变换  
因为实数。

曲线形状如右图所示：



a b  
c d

**FIGURE 4.9**  
(a) Gaussian frequency domain lowpass filter.  
(b) Gaussian frequency domain highpass filter.  
(c) Corresponding lowpass spatial filter.  
(d) Corresponding highpass spatial filter. The masks shown are used in Chapter 3 for lowpass and highpass filtering.



# 第4章 频率域滤波

---

- 4.1 离散傅立叶变换(DFT)
- 4.2 频率域滤波基础
- 4.3 频率域滤波器平滑图像
- 4.4 频率域滤波器锐化图像

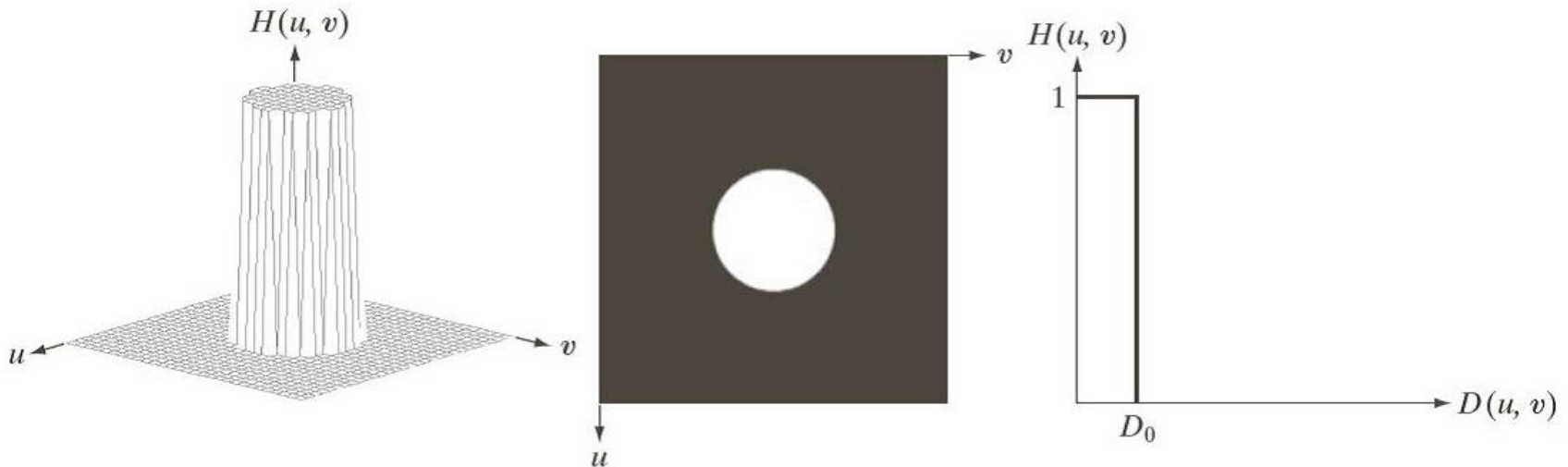


# 4.3 频率域滤波平滑

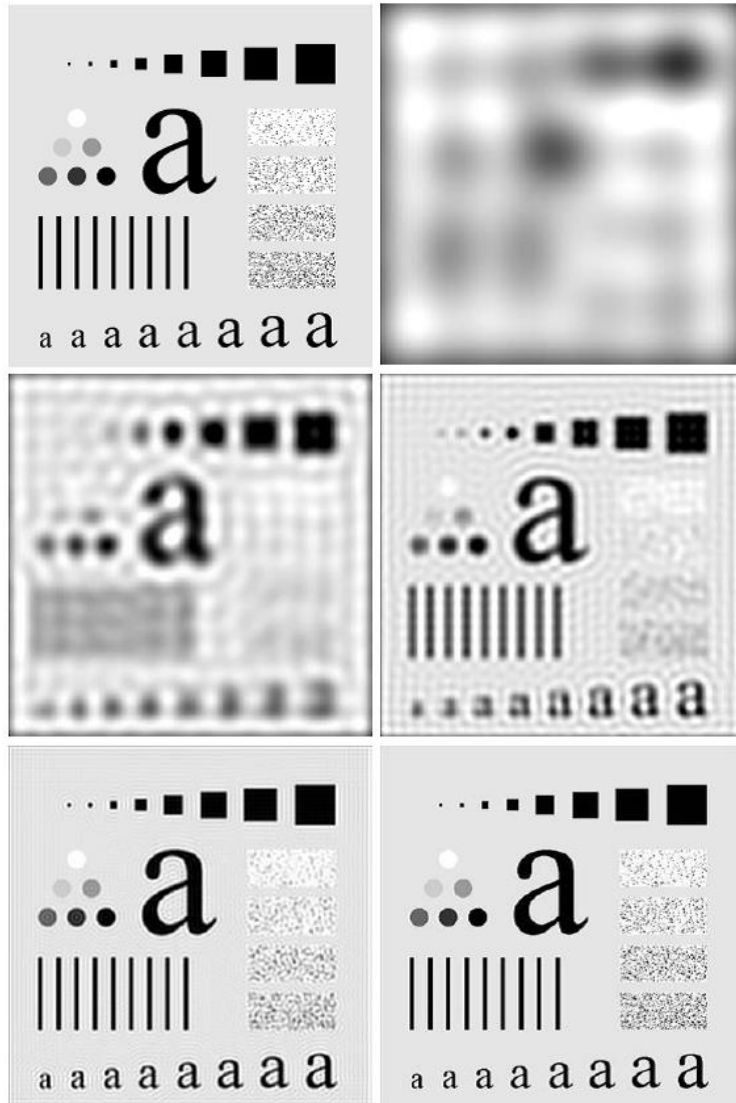
## 理想低通滤波器

$$H(u, v) = \begin{cases} 1 & D(u, v) \leq D_0 \\ 0 & D(u, v) > D_0 \end{cases}$$

其中， $D(u, v)$ 是频率域中点 $(u, v)$ 与频率矩形中心的距离



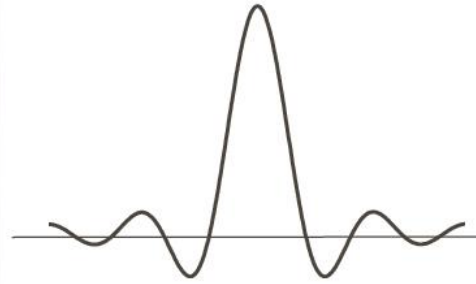
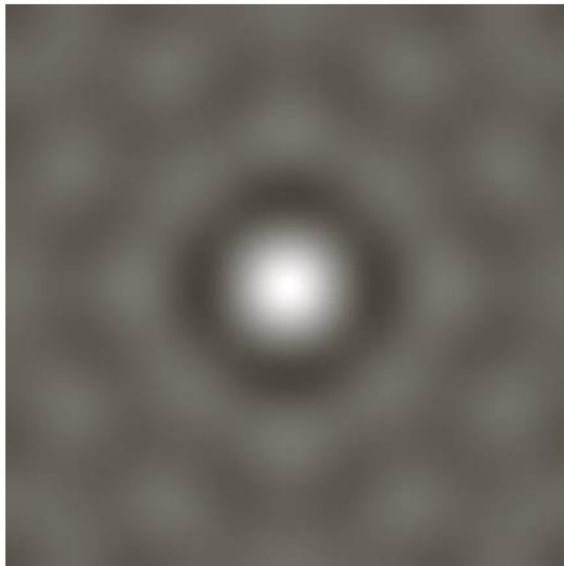
# 理想低通滤波器示例



a b  
c d  
e f

(a) 原图  
(b)-(f) 使用理想低通滤波器，截止频率设置10,30,60,160和460。这些滤波器移除的功率分别为总功率的13%，6.9%，4.3%，2.2%和0.8%。

# 振铃效应解释



a b

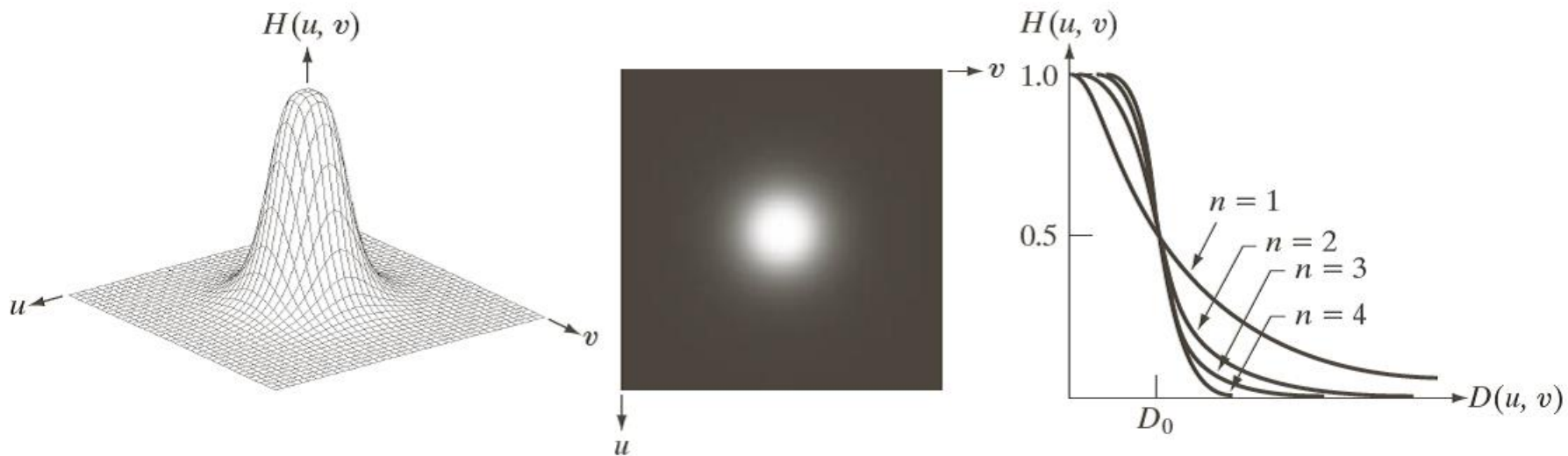
**FIGURE 4.43**

(a) Representation in the spatial domain of an ILPF of radius 5 and size  $1000 \times 1000$ .  
(b) Intensity profile of a horizontal line passing through the center of the image.

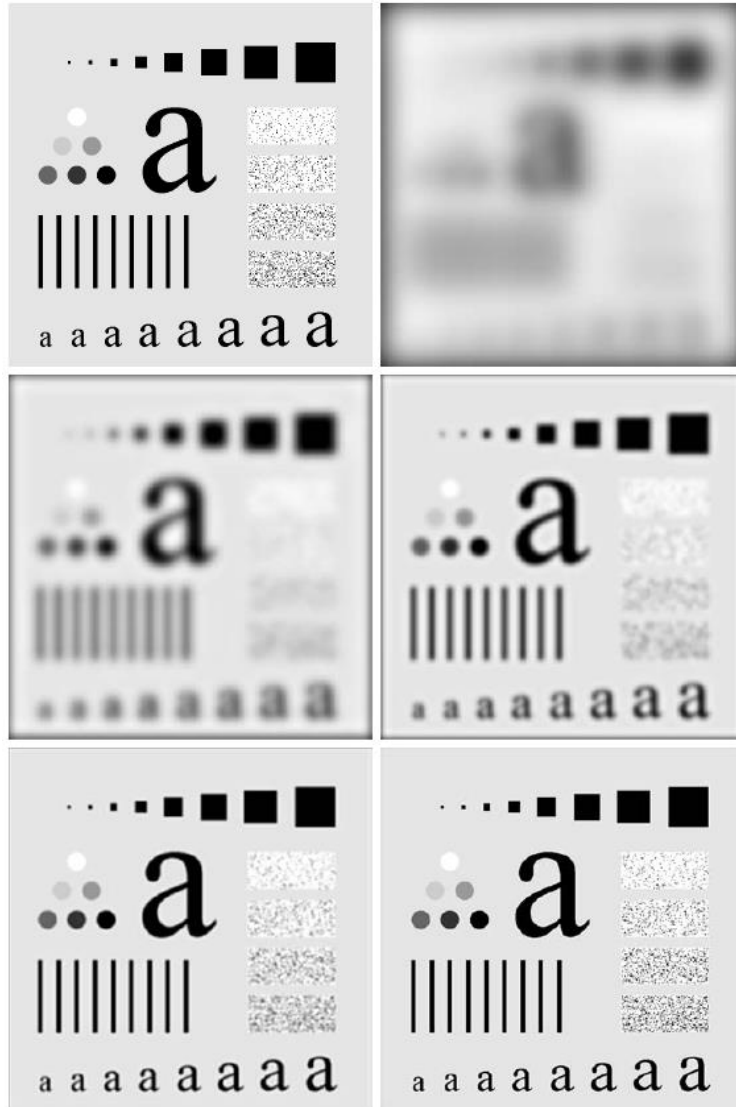
# 4.3 频率域滤波平滑

## n 阶布特沃斯低通滤波器

$$H(u, v) = \frac{1}{1 + [D(u, v) / D_0]^{2n}}$$



# 布特沃斯滤波器示例



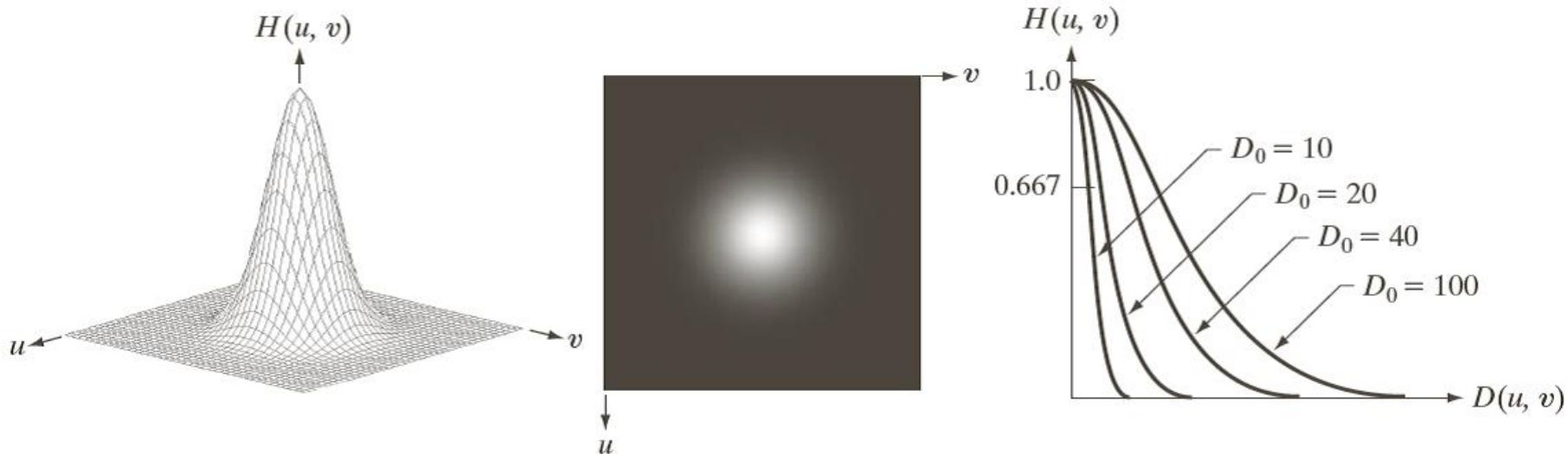
a b  
c d  
e f

(a) 原图  
(b)-(f) 使用二阶布特沃斯低通滤波器的结果，截止频率仍为10，30，60，160和460。

## 4.3 频率域滤波平滑

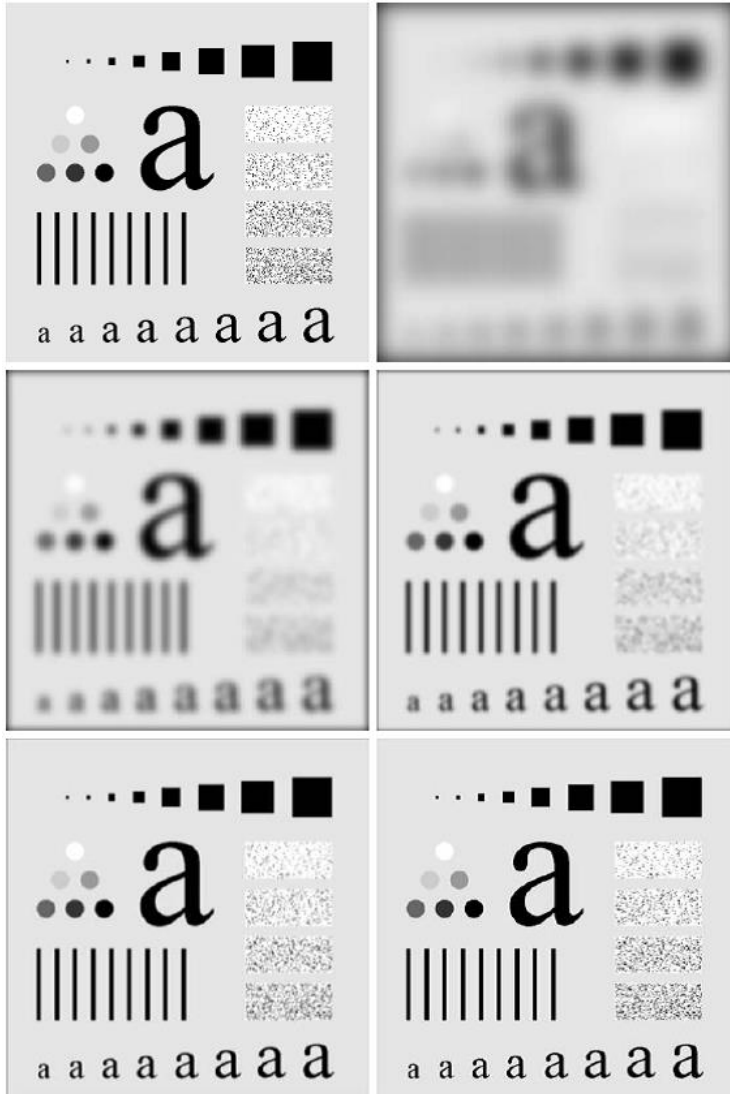
### 高斯低通滤波器

$$H(u, v) = e^{-D^2(u, v)/2D_0^2}$$



高斯低通滤波器（GLPF）的傅里叶反变换也是高斯的，因此通过上式的IDFT得到的空间高斯滤波器没有振铃。

# 高斯低通滤波器示例



a	b
c	d
e	f

(a) 原图  
(b)-(f) 使用高斯低通滤波器的结果，截止频率仍为10，30，60，160和460。



# 低通滤波的其他例子

## 1. 字符识别:

下图：断裂现象

a b

**FIGURE 4.19**

(a) Sample text of poor resolution (note broken characters in magnified view).  
(b) Result of filtering with a GLPF (broken character segments were joined).

Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.



# 低通滤波的其他例子

## 2. 印刷和出版业：预处理

下图：减少面部细纹



a b c

**FIGURE 4.20** (a) Original image ( $1028 \times 732$  pixels). (b) Result of filtering with a GLPF with  $D_0 = 100$ . (c) Result of filtering with a GLPF with  $D_0 = 80$ . Note reduction in skin fine lines in the magnified sections of (b) and (c).

# 低通滤波的其他例子

## 3. 卫星和航空图像:

下图: 墨西哥湾和佛罗里达图像存在“扫描线”  
(用高斯低通来处理)



a b c

**FIGURE 4.21** (a) Image showing prominent scan lines. (b) Result of using a GLPF with  $D_0 = 30$ . (c) Result of using a GLPF with  $D_0 = 10$ . (Original image courtesy of NOAA.)



# 第4章 频率域滤波

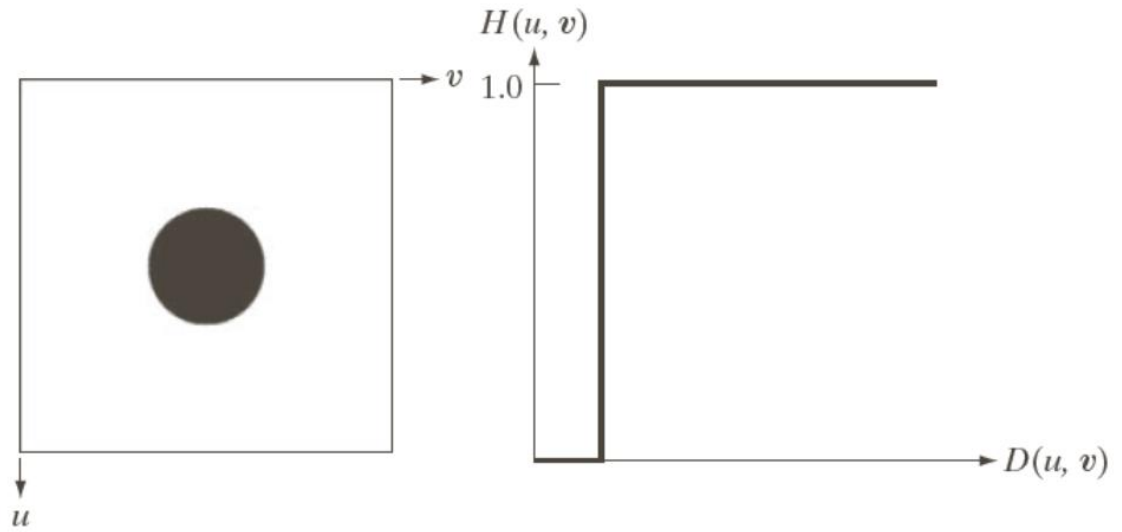
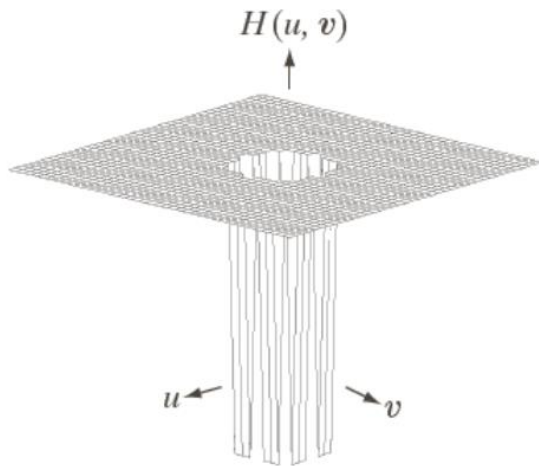
---

- 4.1 离散傅立叶变换(DFT)
- 4.2 频率域滤波基础
- 4.3 频率域滤波器平滑图像
- 4.4 频率域滤波器锐化图像

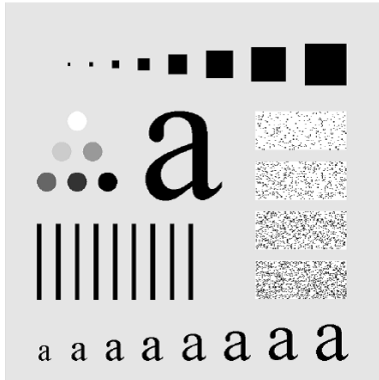
# 4.4 频率域滤波锐化

## 理想高通滤波器

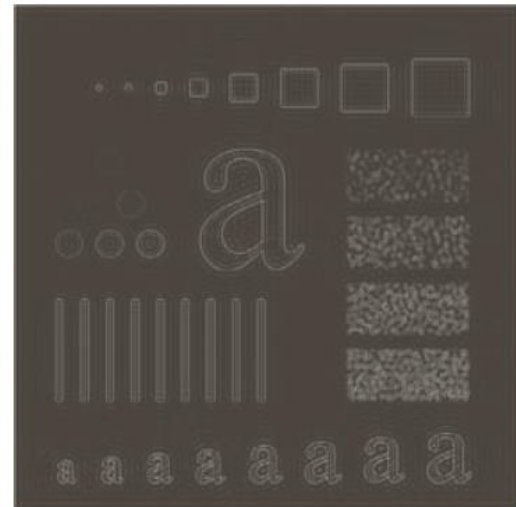
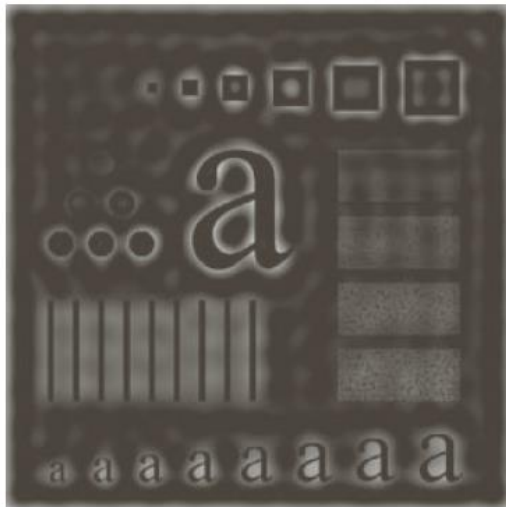
$$H(u, v) = \begin{cases} 0 & D(u, v) \leq D_0 \\ 1 & D(u, v) > D_0 \end{cases}$$



# 理想高通滤波器示例



原图



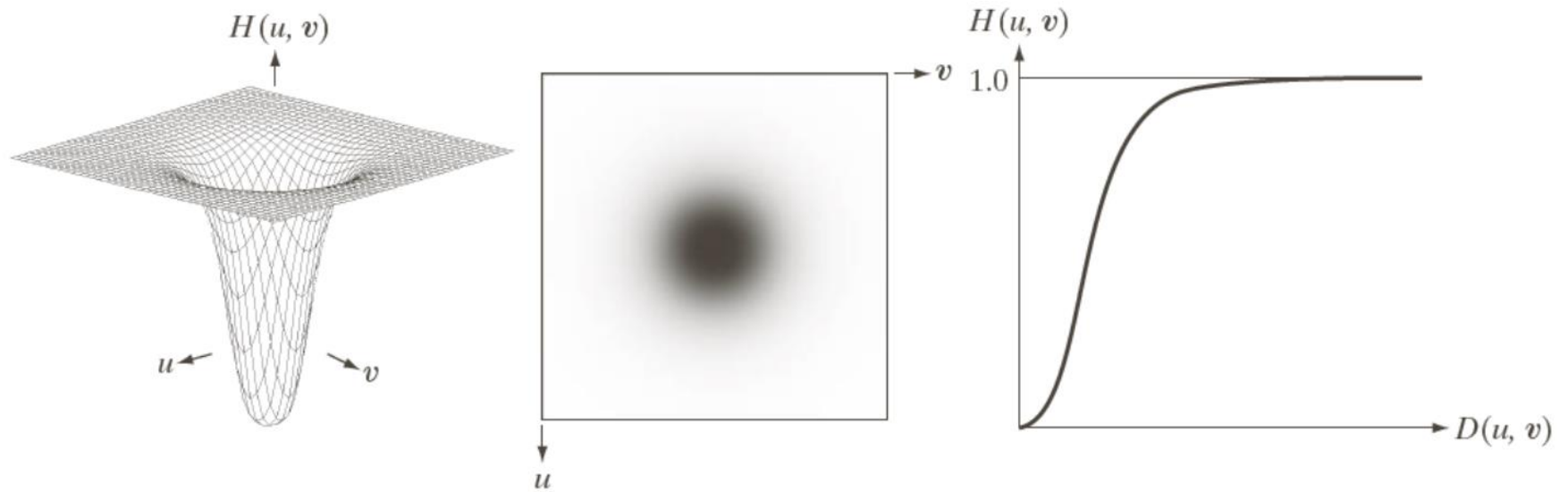
a b c

$D_0 = 30, 60, 100$  的理想高通滤波器结果

# 4.4 频率域滤波锐化

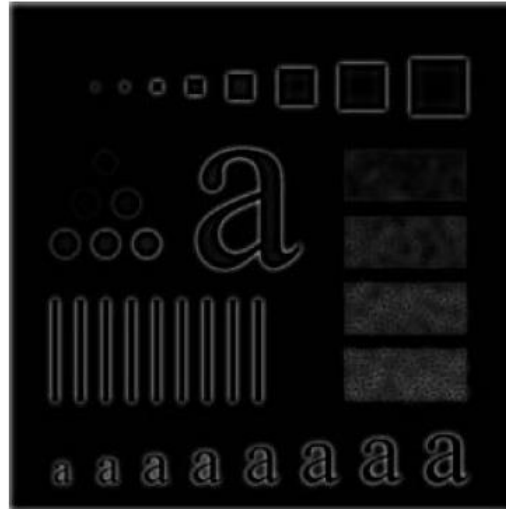
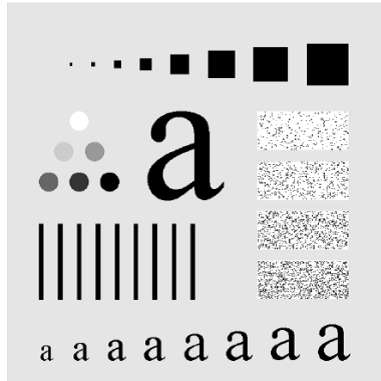
## 布特沃斯高通滤波器

$$H(u, v) = \frac{1}{1 + [D_0 / D(u, v)]^{2n}}$$





# 布特沃斯高通滤波器示例



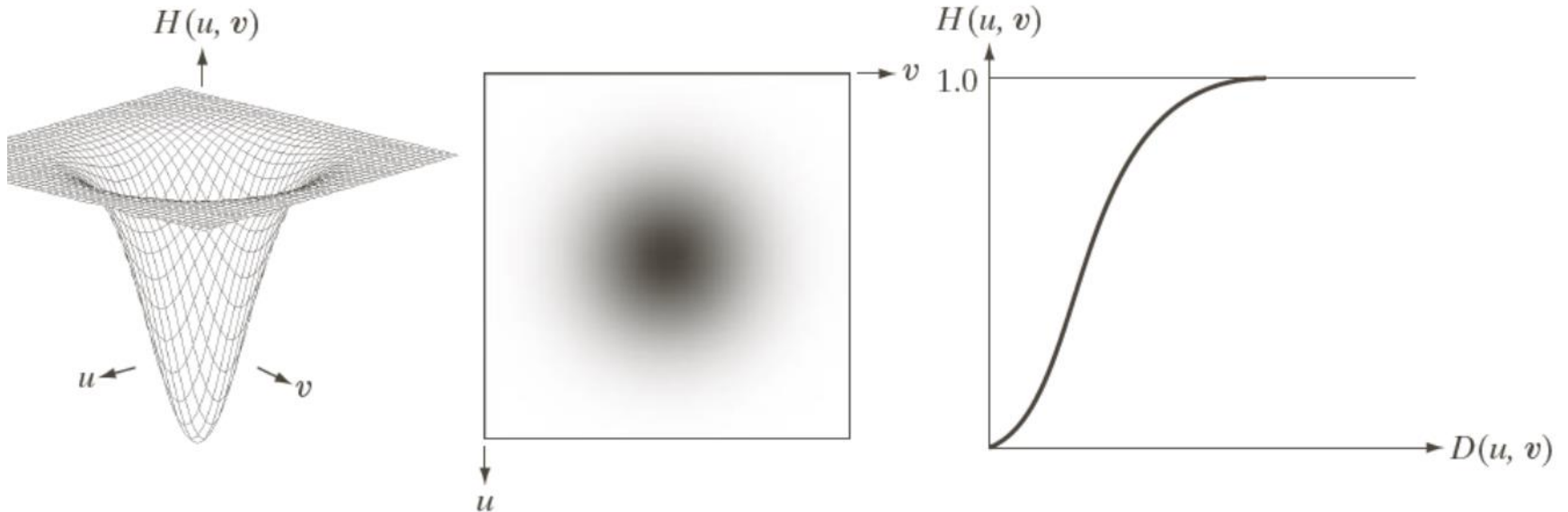
a b c

$D_0 = 30, 60, 100$  的2阶布特沃斯高通滤波器结果

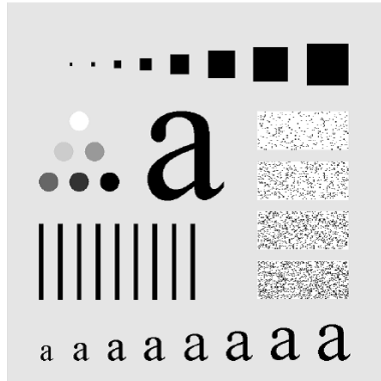
# 4.4 频率域滤波锐化

## 高斯高通滤波器

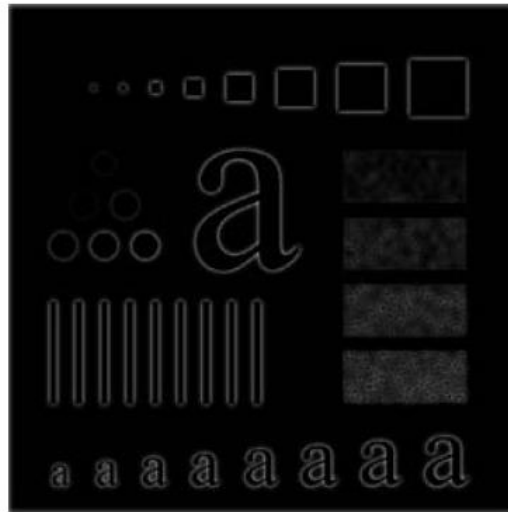
$$H(u, v) = 1 - e^{-D^2(u, v) / 2D_0^2}$$



# 高斯高通滤波器示例



原图



a b c

$D_0 = 30, 60, 100$  的高斯高通滤波器结果



## 4.4 频率域滤波锐化：同态滤波

- 一幅图像 $f(x,y)$ 可以表示为照射分量和反射分量的乘积。

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

- 然而上式不能用来直接对两部分分量分别进行操作，因为两个函数乘积的傅立叶变换是不可分的。

$$F\{f(x,y)\} \neq F\{i(x,y)\}F\{r(x,y)\}$$

- 我们对图像函数两边取对数，则可以将两个分量分开。

$$z(x,y) = \ln f(x,y) = \ln i(x,y) + \ln r(x,y)$$

$$F\{\ln f(x,y)\} = F\{\ln i(x,y)\} + F\{\ln r(x,y)\}$$



# 同态滤波步骤

1. 两边取对数:  $f(x,y) = \ln i(x,y) + \ln r(x,y)$

2. 两边取付氏变换:  $F(u,v) = I(u,v) + R(u,v)$

3. 用一频域函数  $H(u,v)$  处理  $F(u,v)$ :

$$H(u,v)F(u,v) = H(u,v) I(u,v) + H(u,v) R(u,v)$$

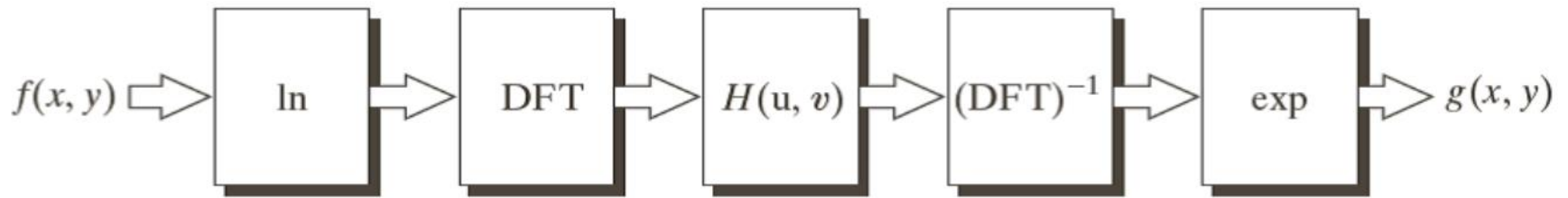
4. 反变换到空域:

$$s(x,y) = i'(x,y) + r'(x,y)$$

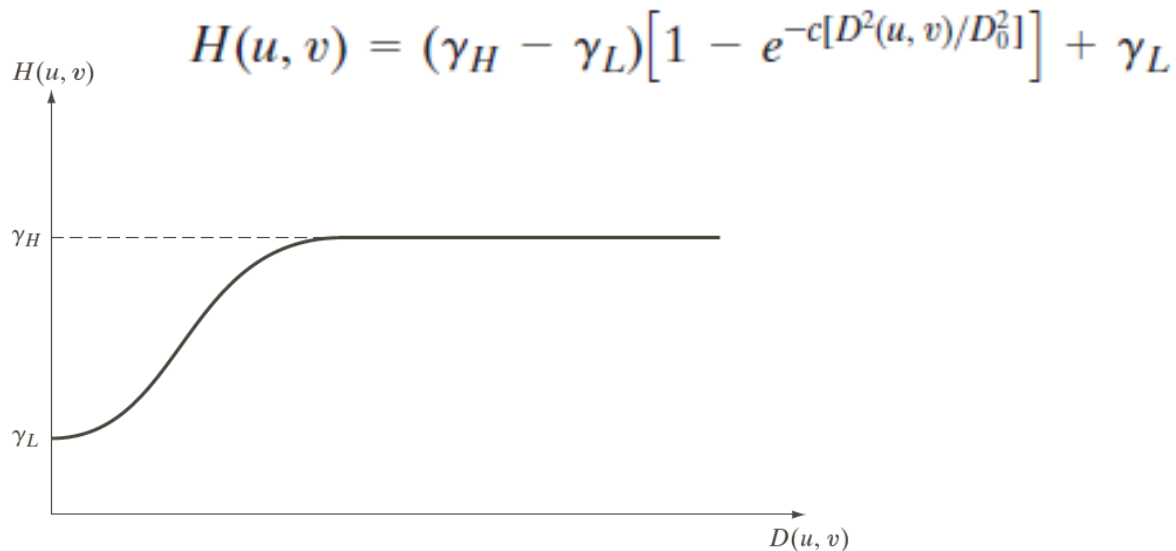
5. 两边取指数:

$$g(x,y) = \exp\{i'(x,y)\} \exp\{r'(x,y)\} = i_0(x,y) r_0(x,y)$$

# 同态滤波步骤及滤波器径向剖面图

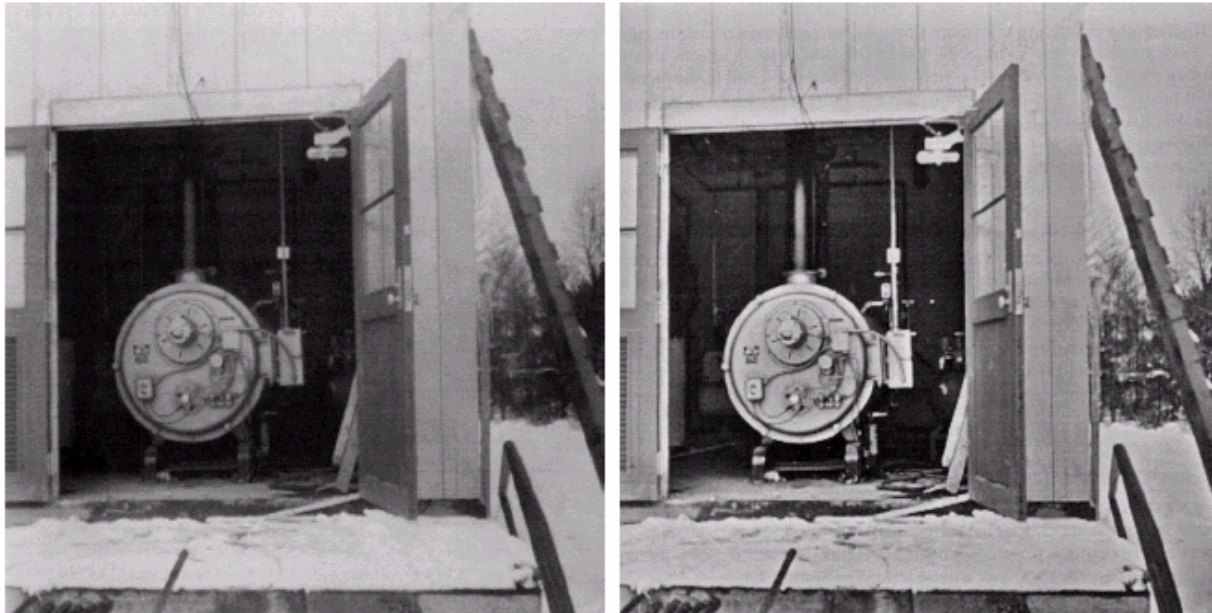


同态滤波器函数径向剖面图：滤波器函数趋向于衰减低频的贡献，而增强高频的贡献。



# 同态滤波示例

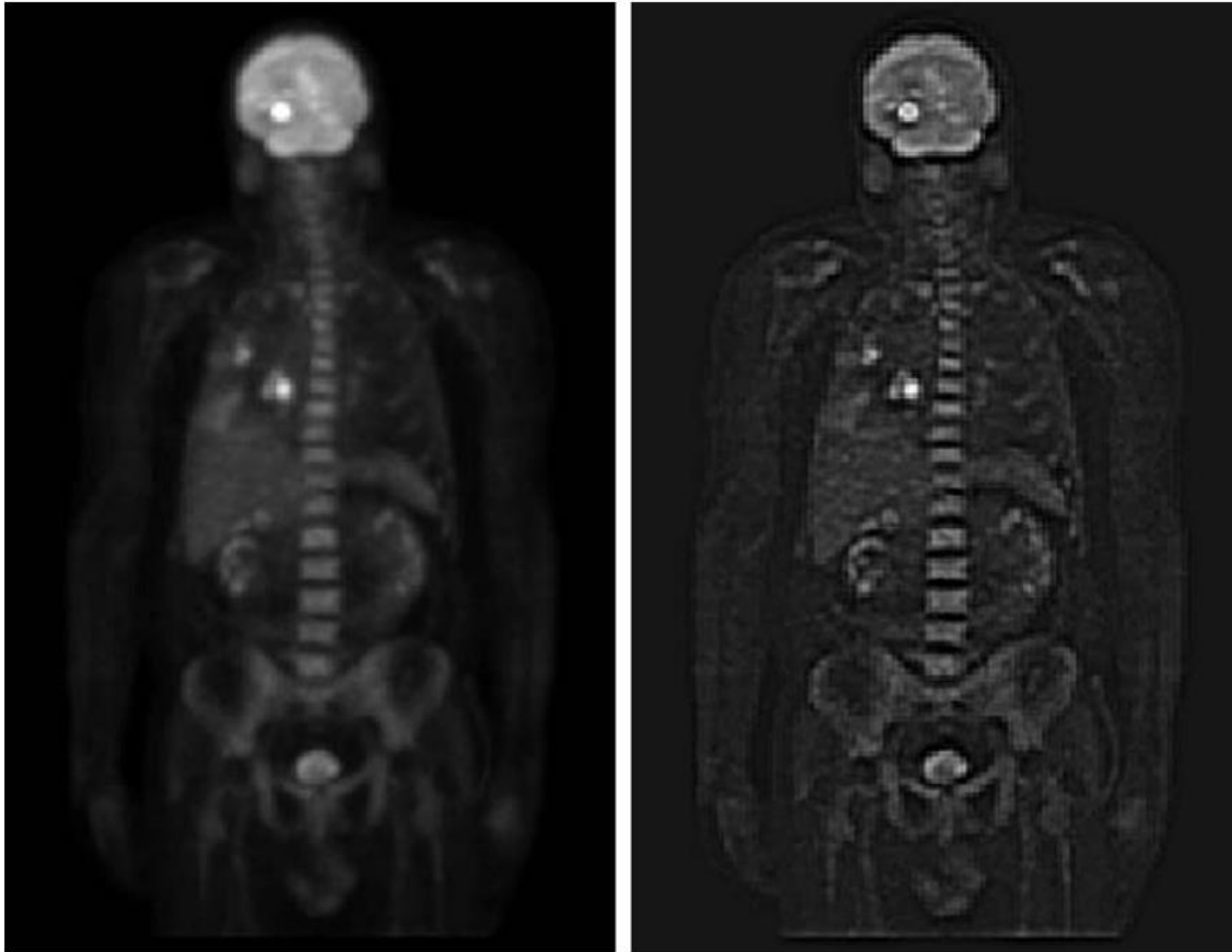
- **特点：**能消除乘性噪声，能同时压缩图象的整体动态范围和增加图象中相邻区域间的对比度
- 示例一：





# 同态滤波示例

同态滤波示例二：





# 第5章 图像复原与重建

- 数字图象恢复与增强的目的类似，也是旨在改善图象的质量。
- 但恢复是力求保持图象的本来面目，即以**保真**原则为其前提，这是区别于增强的基本不同点。
- 因而恢复时要了解图象质量下降的物理过程，找出或估计其物理模型。恢复的过程就是沿着质量下降的逆过程来重现原始图象。



# 第5章 图像复原与重建

- 5.1 图像退化/复原过程的模型
- 5.2 噪声模型
- 5.3 只存在噪声的复原——空间滤波
- 5.4 用频率域消除周期噪声
- 5.5 线性、位置不变的退化
- 5.6 估计退化函数
- 5.7 逆滤波
- 5.8 最小均方误差（维纳）滤波
- 5.9 约束最小二乘方滤波
- 5.10 由投影重建图像

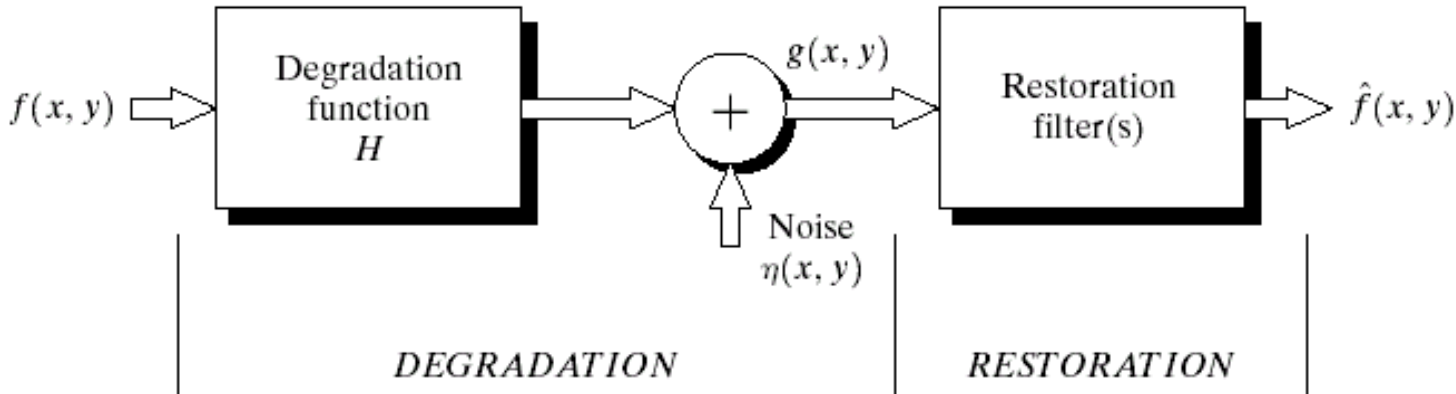


# 第5章 图像复原与重建

- 5.1 图像退化/复原过程的模型
- 5.2 噪声模型
- 5.3 只存在噪声的复原——空间滤波
- 5.4 用频率域消除周期噪声
- 5.5 线性、位置不变的退化
- 5.6 估计退化函数
- 5.7 逆滤波
- 5.8 最小均方误差（维纳）滤波
- 5.9 约束最小二乘方滤波
- 5.10 由投影重建图像

# 5.1 图像退化/复原过程的模型

□ 线性位移不变系统，加性噪声



**FIGURE 5.1** A model of the image degradation/restoration process.

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

$$G(u, v) = H(u, v)F(u, v) + N(u, v)$$



# 第5章 图像复原与重建

- 5.1 图像退化/复原过程的模型
- 5.2 **噪声模型**
- 5.3 只存在噪声的复原——空间滤波
- 5.4 用频率域消除周期噪声
- 5.5 线性、位置不变的退化
- 5.6 估计退化函数
- 5.7 逆滤波
- 5.8 最小均方误差（维纳）滤波
- 5.9 约束最小二乘方滤波
- 5.10 由投影重建图像



## 5.2 噪声模型

### □ 噪声的产生

- 图像获取：CCD相机
- 图像传输：无线传输

### □ 噪声的空域特性和频域特性

- 空域特性：噪声的直方图分布
- 频域特性：噪声在Fourier频谱上的分布

### □ 几种典型的噪声模型

- 高斯噪声 (Gaussian noise), 瑞利噪声(Rayleigh noise), 厄兰噪声(Erlang noise), 指数噪声(Exponential noise), 均匀噪声(Uniform noise), 椒盐噪声(salt-and-pepper),





# 几种典型的噪声模型

$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(z-\mu)^2/2\sigma^2} \quad (\text{高斯噪声})$$

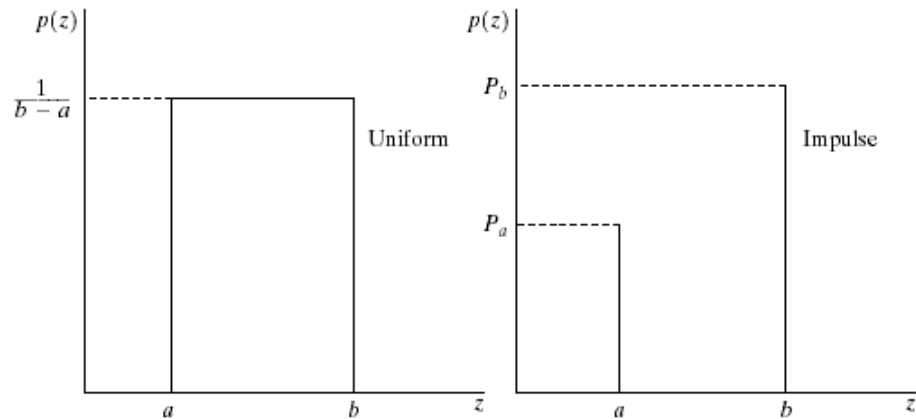
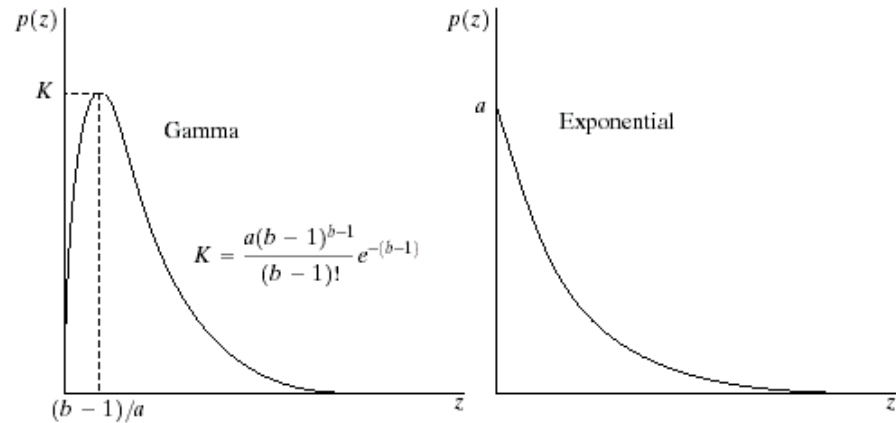
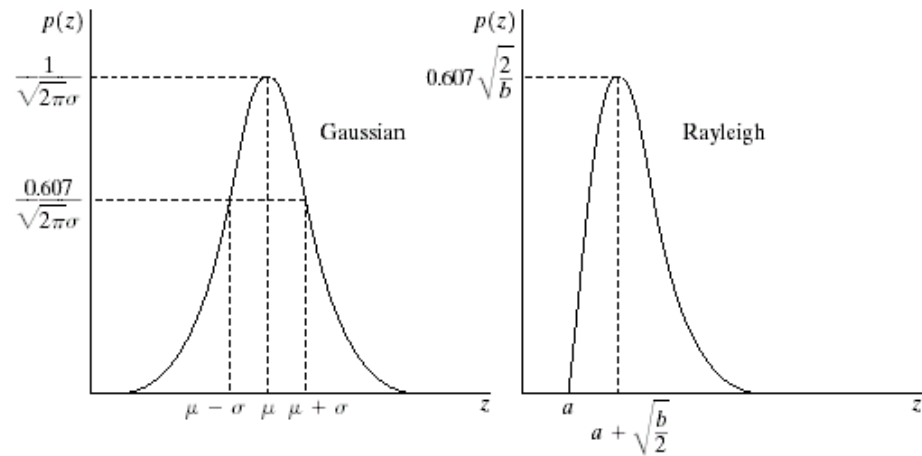
$$p(z) = \begin{cases} \frac{2}{b} (z-a) e^{-(z-a)^2/b} & \text{for } z \geq a \\ 0 & \text{for } z < a \end{cases} \quad (\text{瑞利噪声})$$

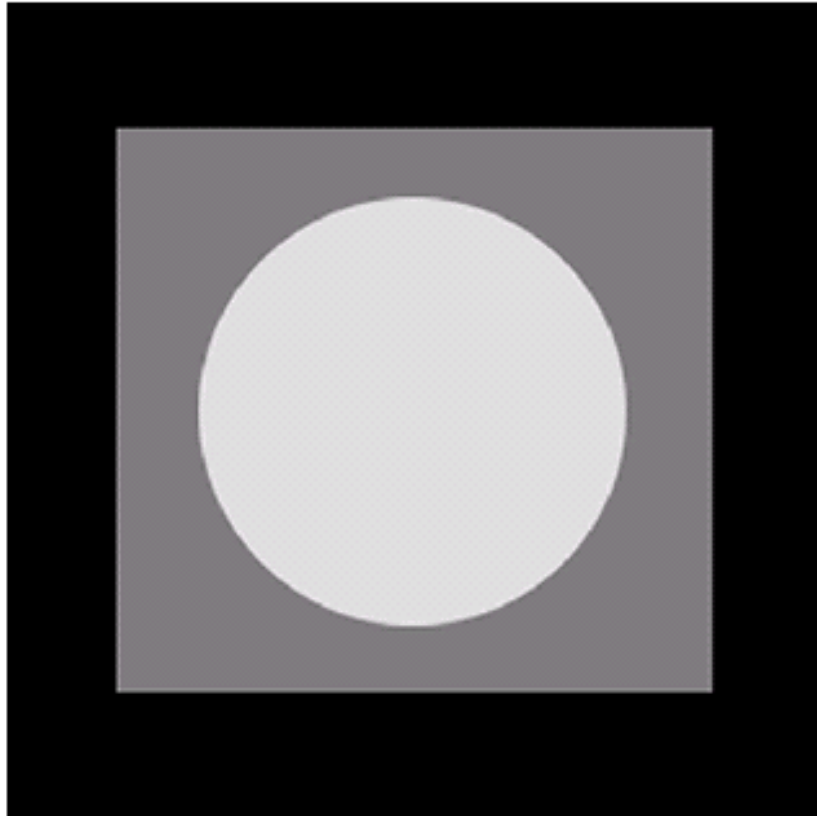
$$p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases} \quad (\text{厄兰噪声})$$

$$p(z) = \begin{cases} a e^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases} \quad (\text{指数噪声})$$

$$p(z) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq z \leq b \\ 0 & \text{otherwise} \end{cases} \quad (\text{均匀噪声})$$

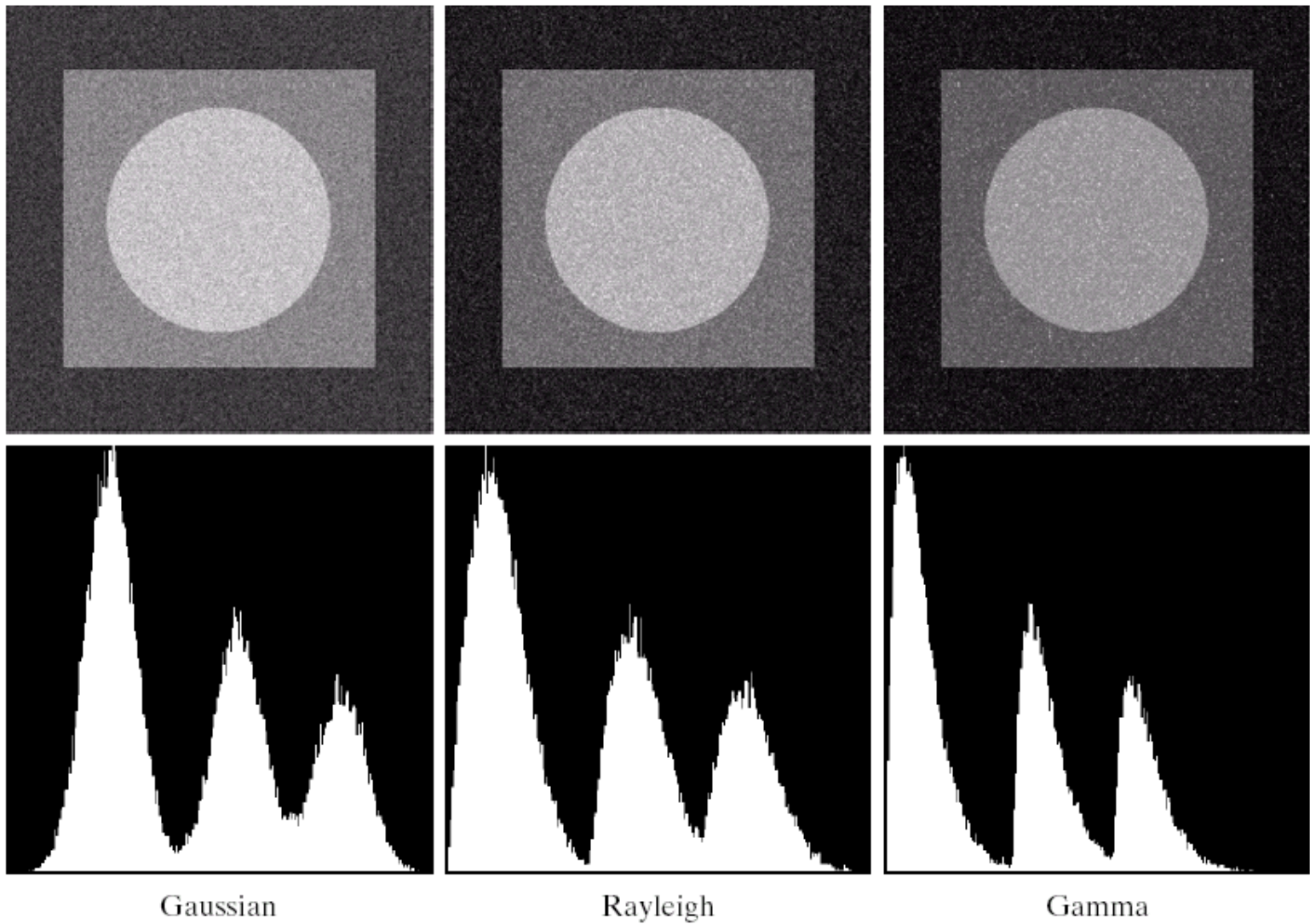
$$p(z) = \begin{cases} p_a & \text{for } z = a \\ p_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases} \quad (\text{椒盐噪声})$$





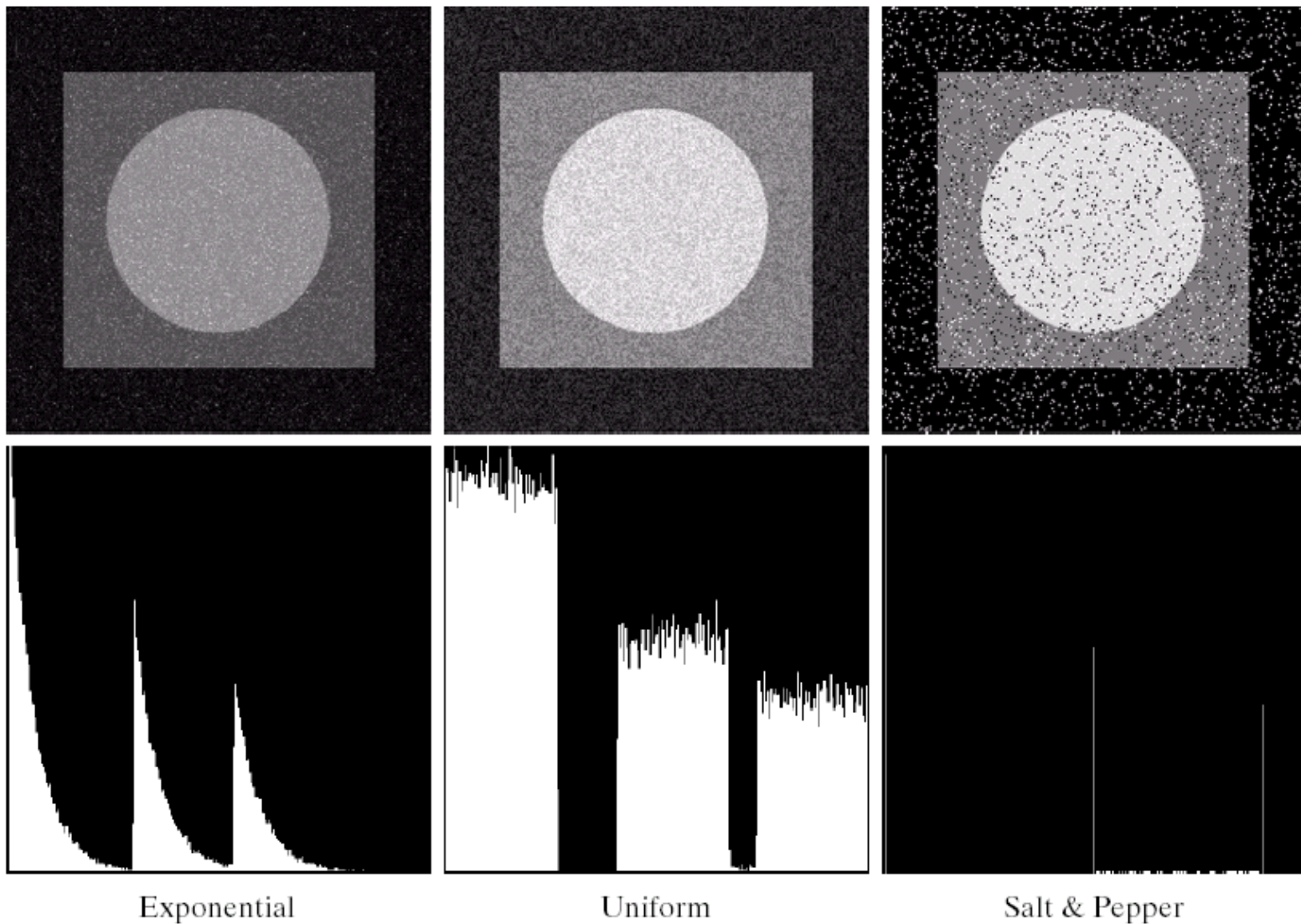
**FIGURE 5.3** Test pattern used to illustrate the characteristics of the noise PDFs shown in Fig. 5.2.

---



a	b	c
d	e	f

**FIGURE 5.4** Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the image in Fig. 5.3.



g h i  
j k l

**FIGURE 5.4 (Continued)** Images and histograms resulting from adding exponential, uniform, and impulse noise to the image in Fig. 5.3.

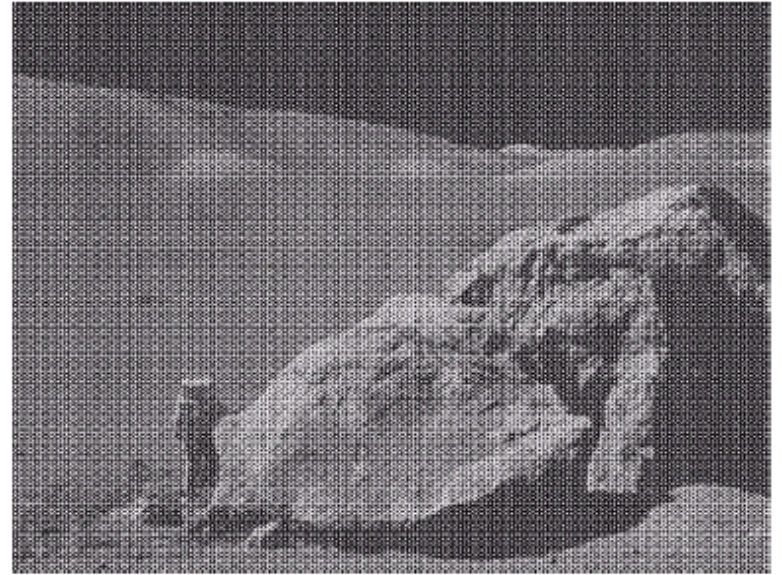


# 周期噪声(Periodic Noise)

a  
b

**FIGURE 5.5**

(a) Image corrupted by sinusoidal noise.  
 (b) Spectrum (each pair of conjugate impulses corresponds to one sine wave).  
 (Original image courtesy of NASA.)

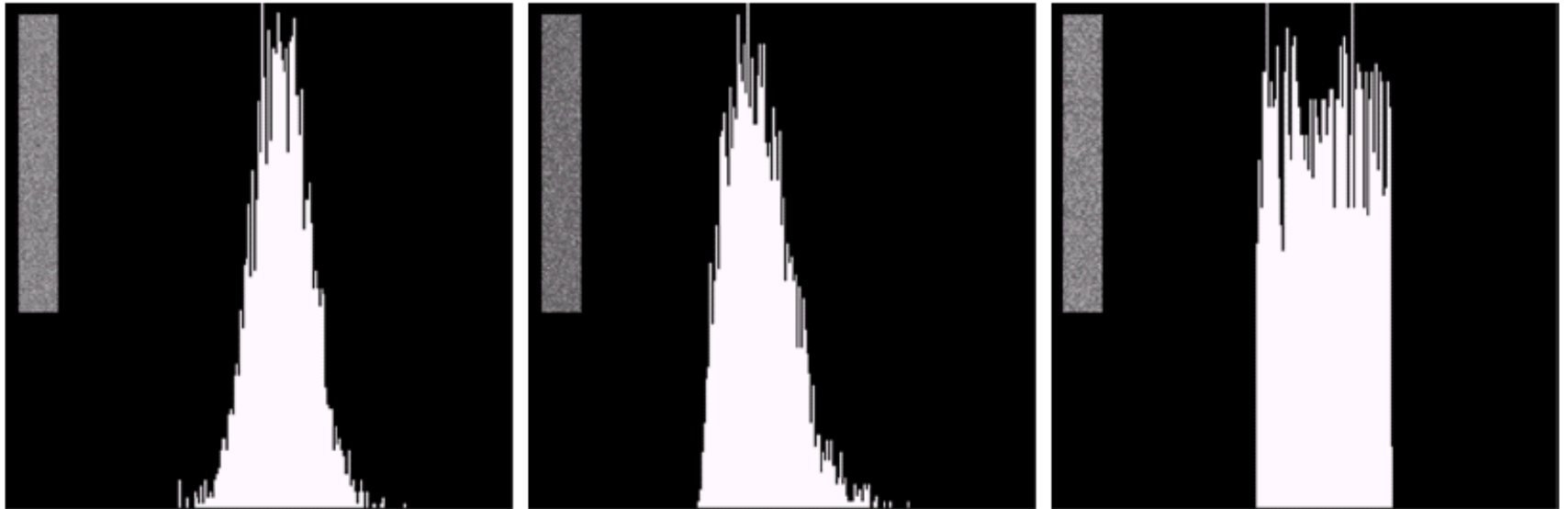


$$\cos(2\pi u_0 x + 2\pi v_0 y)$$



$$\frac{1}{2} [\delta(u + Mu_0, v + Nv_0) + \delta(u - Mu_0, v - Nv_0)]$$

# 估计噪声参数



a b c

**FIGURE 5.6** Histograms computed using small strips (shown as inserts) from (a) the Gaussian, (b) the Rayleigh, and (c) the uniform noisy images in Fig. 5.4.



# 第5章 图像复原与重建

- 5.1 图像退化/复原过程的模型
- 5.2 噪声模型
- 5.3 只存在噪声的复原——空间滤波
- 5.4 用频率域消除周期噪声
- 5.5 线性、位置不变的退化
- 5.6 估计退化函数
- 5.7 逆滤波
- 5.8 最小均方误差（维纳）滤波
- 5.9 约束最小二乘方滤波
- 5.10 由投影重建图像





## 5.3 只存在噪声的复原——空间滤波

- 基于空域滤波的方法仅针对加性噪声
- 与空域增强原理相同
- 几种常用滤波器
  - 均值滤波器 (Mean Filters)
  - 次序统计滤波器 (Order Statistic Filters)
  - 自适应滤波器 (Adaptive Filters)



# 几种常用滤波器

- 算术平均滤波器(Arithmetic mean filters)

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s, t)$$

- 几何平均滤波器(Geometric mean filter)

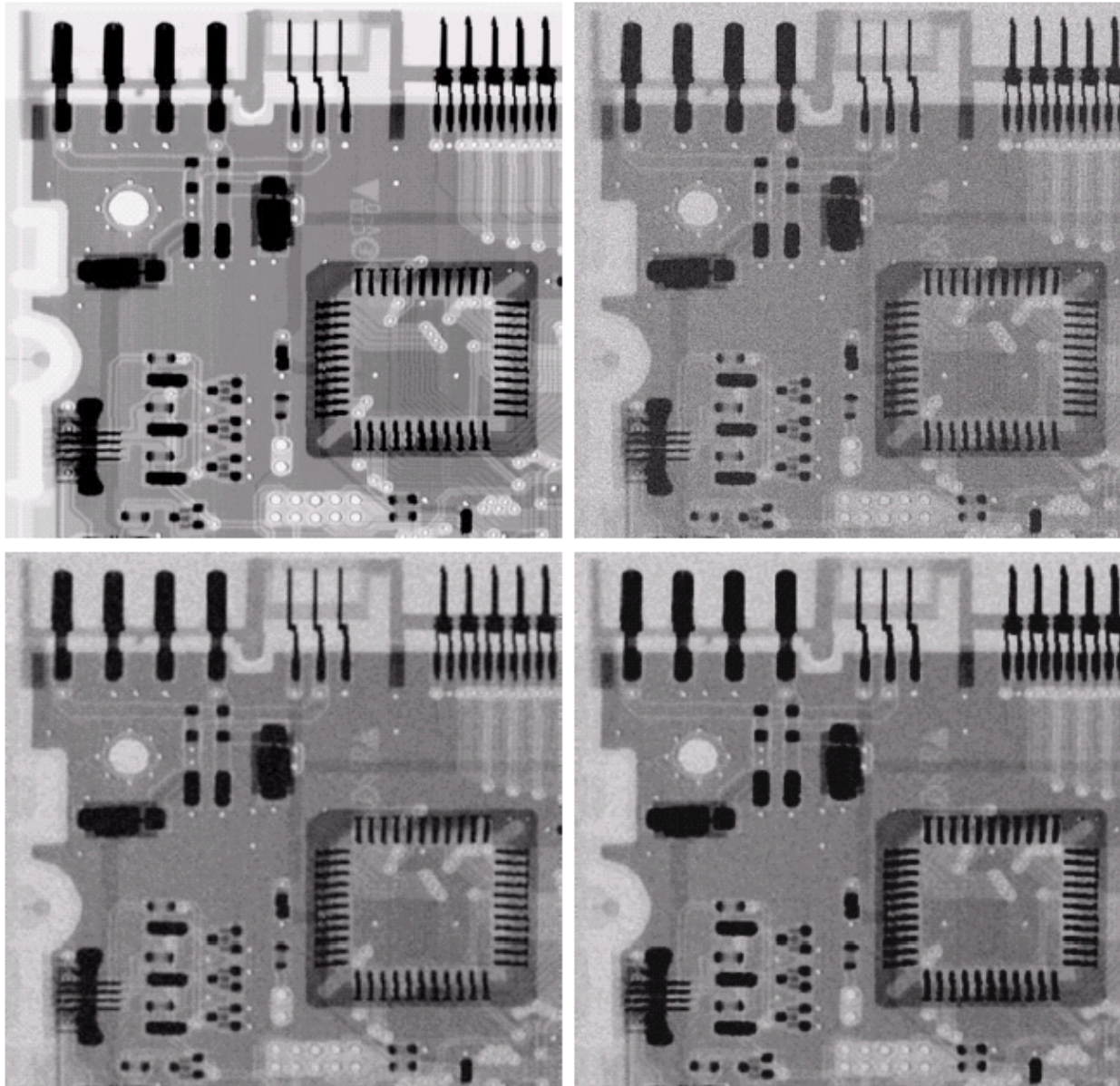
$$\hat{f}(x, y) = \left[ \prod_{(s,t) \in S_{xy}} g(s, t) \right]^{\frac{1}{mn}}$$

- 调和平均滤波器(Harmonic mean filter)

$$\hat{f}(x, y) = \frac{mn}{\sum_{(s,t) \in S_{xy}} \frac{1}{g(s, t)}}$$

- 反调和平均滤波器(Contraharmonic filter)

$$\hat{f}(x, y) = \frac{\sum_{(s,t) \in S_{xy}} g(s, t)^{Q+1}}{\sum_{(s,t) \in S_{xy}} g(s, t)^Q}$$



a	b
c	d

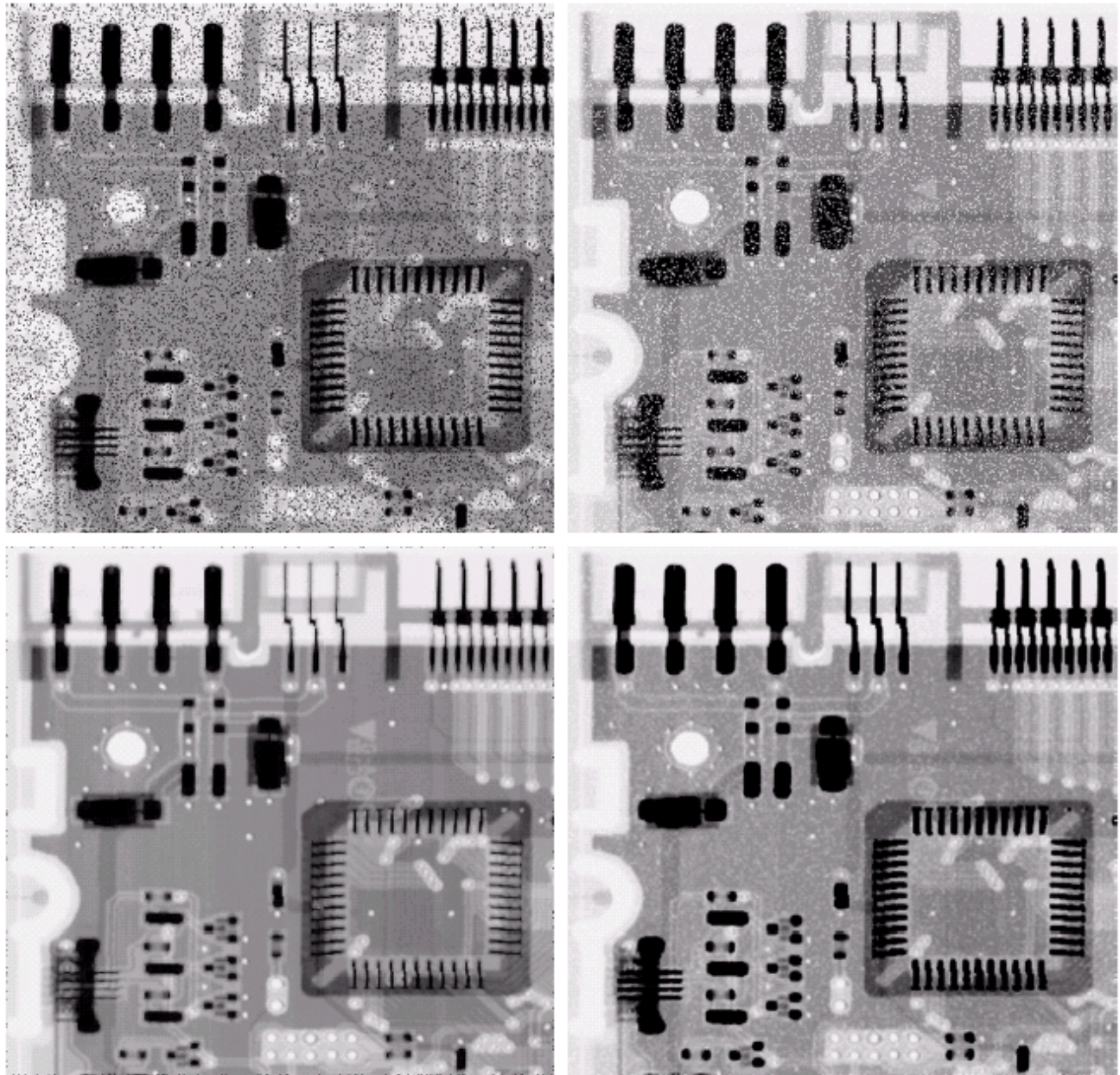
**FIGURE 5.7** (a) X-ray image. (b) Image corrupted by additive Gaussian noise. (c) Result of filtering with an arithmetic mean filter of size  $3 \times 3$ . (d) Result of filtering with a geometric mean filter of the same size. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

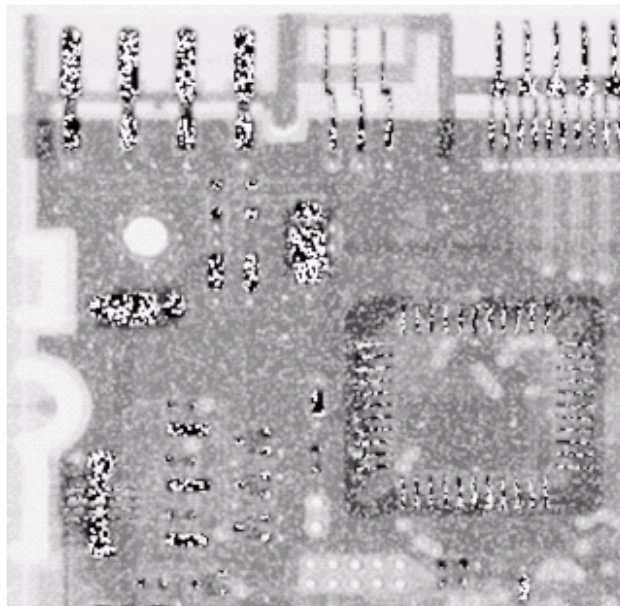
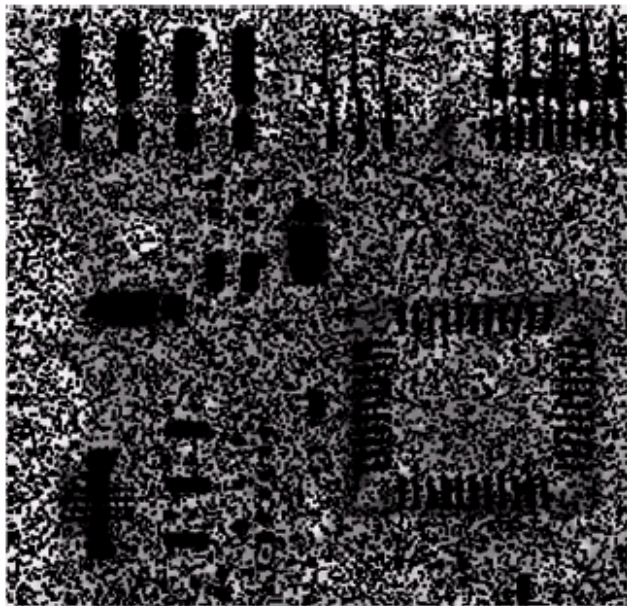


a b  
c d

**FIGURE 5.8**

(a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability. (c) Result of filtering (a) with a  $3 \times 3$  contra-harmonic filter of order 1.5. (d) Result of filtering (b) with  $Q = -1.5$ .





a b

**FIGURE 5.9** Results of selecting the wrong sign in contra-harmonic filtering. (a) Result of filtering Fig. 5.8(a) with a contra-harmonic filter of size  $3 \times 3$  and  $Q = -1.5$ . (b) Result of filtering 5.8(b) with  $Q = 1.5$ .





# 次序统计滤波器

中值滤波器(Median filter)

$$\hat{f}(x, y) = \underset{(s,t) \in \mathcal{S}_{xy}}{\text{median}} \{g(s, t)\}$$

最大最小滤波器(Max and min filters)

$$\left\{ \begin{array}{l} \hat{f}(x, y) = \max_{(s,t) \in \mathcal{S}_{xy}} \{g(s, t)\} \\ \hat{f}(x, y) = \min_{(s,t) \in \mathcal{S}_{xy}} \{g(s, t)\} \end{array} \right.$$

中点滤波器(Midpoint filter)

$$\hat{f}(x, y) = \frac{1}{2} \left[ \max_{(s,t) \in \mathcal{S}_{xy}} \{g(s, t)\} + \min_{(s,t) \in \mathcal{S}_{xy}} \{g(s, t)\} \right]$$

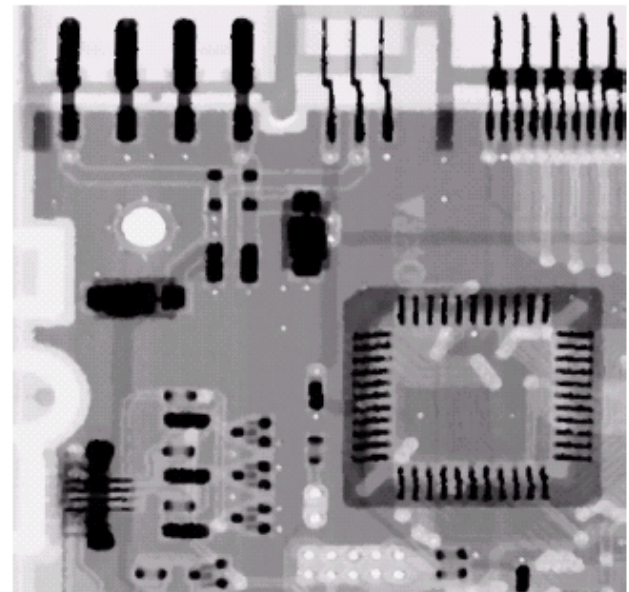
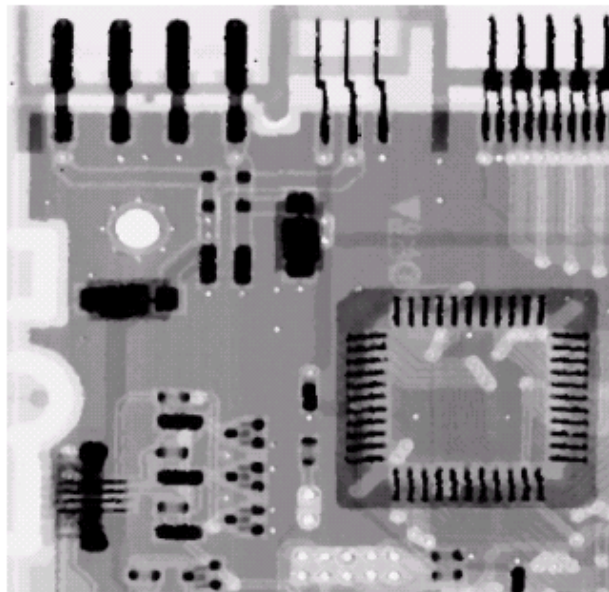
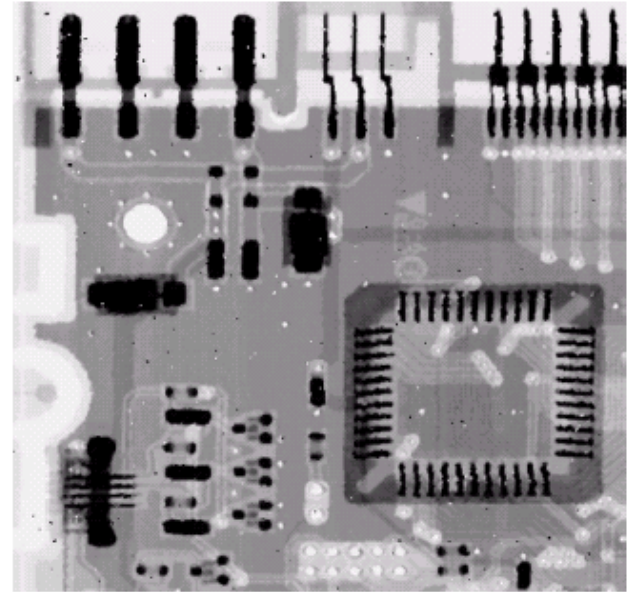
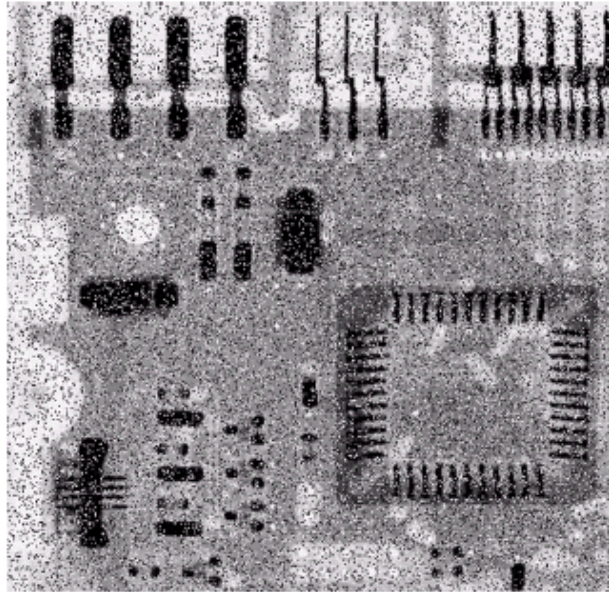
Alpha截取中值滤波器(Alpha-trimmed mean filter)

$$\hat{f}(x, y) = \frac{1}{mn - d} \sum_{(s,t) \in \mathcal{S}_{xy}} g_r(s, t)$$

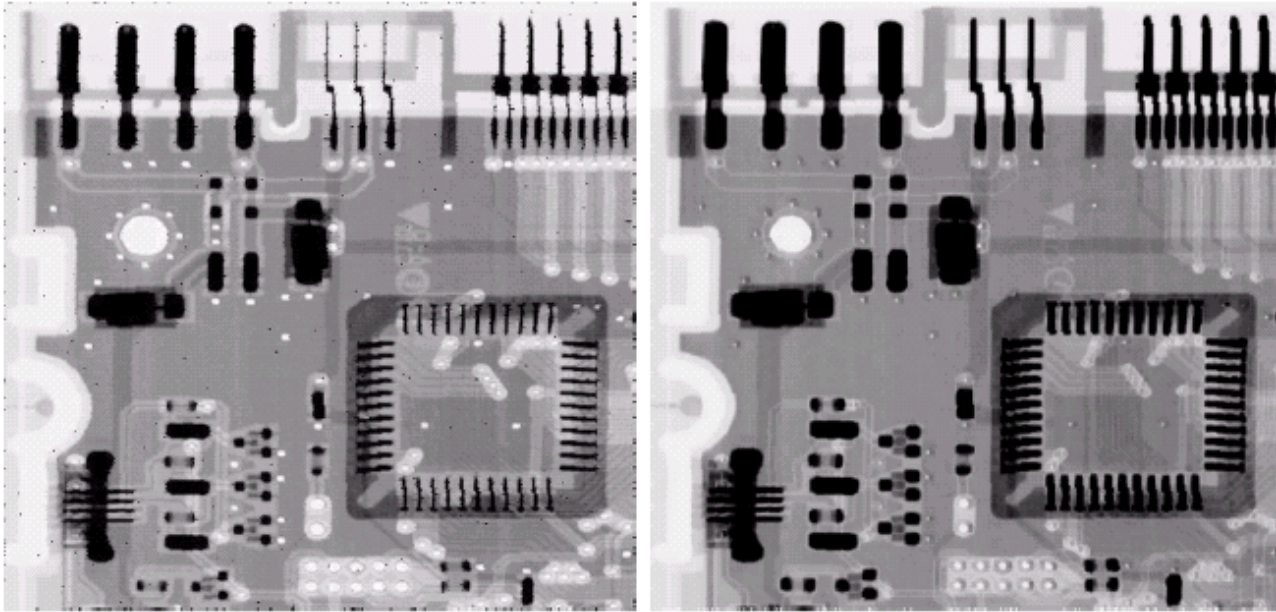
a b  
c d

**FIGURE 5.10**

- (a) Image corrupted by salt-and-pepper noise with probabilities  $P_a = P_b = 0.1$ .  
(b) Result of one pass with a median filter of size  $3 \times 3$ .  
(c) Result of processing (b) with this filter.  
(d) Result of processing (c) with the same filter.



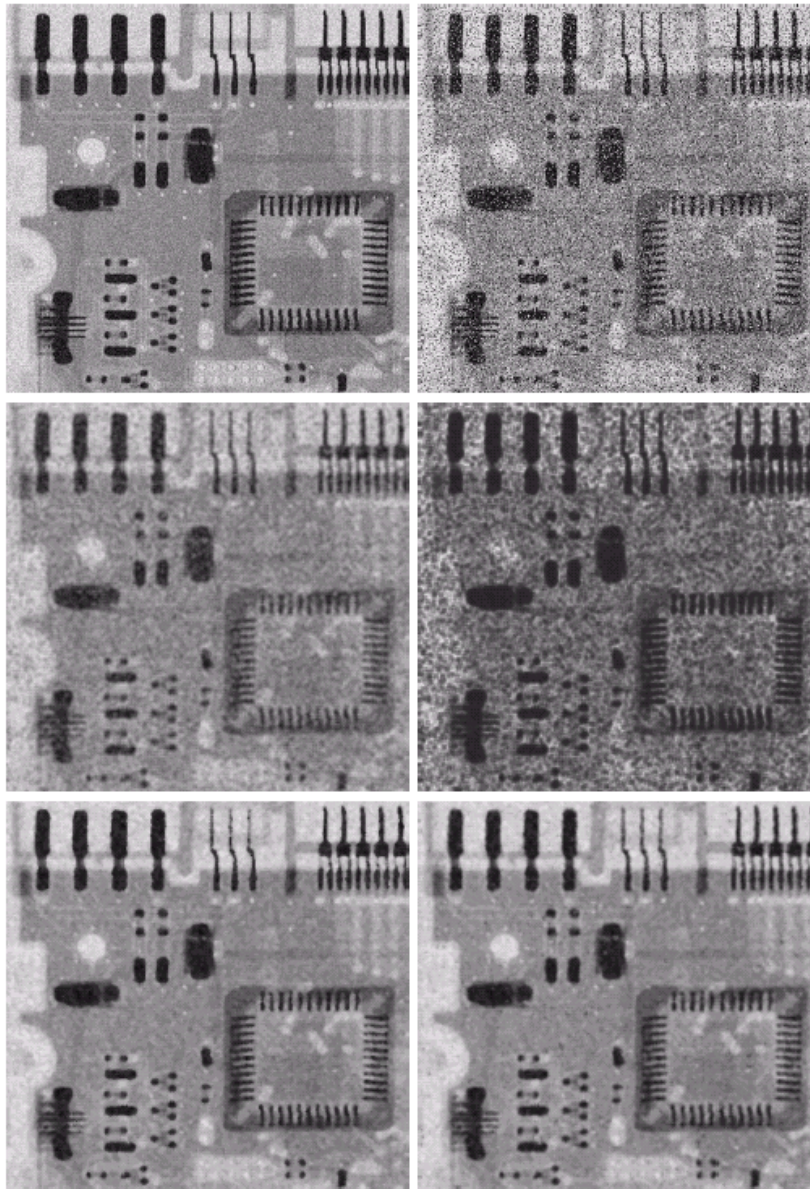




a b

**FIGURE 5.11**  
(a) Result of  
filtering

Fig. 5.8(a) with a  
max filter of size  
 $3 \times 3$ . (b) Result  
of filtering 5.8(b)  
with a min filter  
of the same size.



**FIGURE 5.12** (a) Image corrupted by additive uniform noise. (b) Image additionally corrupted by additive salt-and-pepper noise. Image in (b) filtered with a  $5 \times 5$ : (c) arithmetic mean filter; (d) geometric mean filter; (e) median filter; and (f) alpha-trimmed mean filter with  $d = 5$ .



# 第5章 图像复原与重建

- 5.1 图像退化/复原过程的模型
- 5.2 噪声模型
- 5.3 只存在噪声的复原——空间滤波
- 5.4 用频率域消除周期噪声
- 5.5 线性、位置不变的退化
- 5.6 估计退化函数
- 5.7 逆滤波
- 5.8 最小均方误差（维纳）滤波
- 5.9 约束最小二乘方滤波
- 5.10 由投影重建图像



## 5.4 用频率域消除周期噪声

### Periodic Noise Reduction by Frequency Domain Filtering

- 带阻滤波器(Bandreject Filters)
- 带通滤波器(Bandpass Filters)
- 槽口滤波器(Notch Filters)



# 带阻滤波器 (Bandreject Filters)

理想带阻滤波器(Ideal Bandreject Filters)

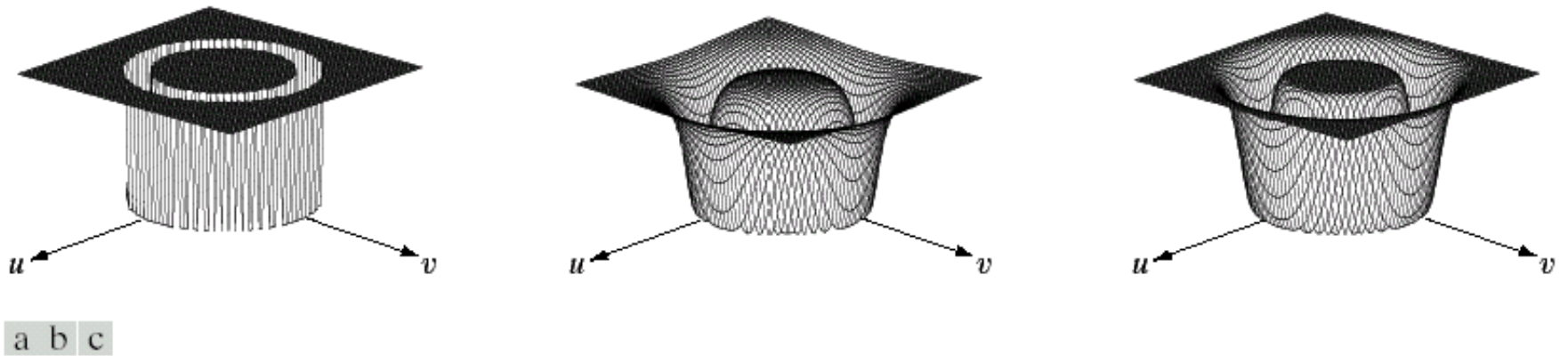
$$H(u, v) = \begin{cases} 1 & \text{if } D(u, v) < D_0 - \frac{W}{2} \\ 0 & \text{if } D_0 - \frac{W}{2} \leq D(u, v) \leq D_0 + \frac{W}{2} \\ 1 & \text{if } D(u, v) > D_0 + \frac{W}{2} \end{cases}$$

Butterworth带阻滤波器(Butterworth Bandreject Filters)

$$H(u, v) = \frac{1}{1 + \left[ \frac{D(u, v)W}{D^2(u, v) + D_0^2} \right]^{2n}}$$

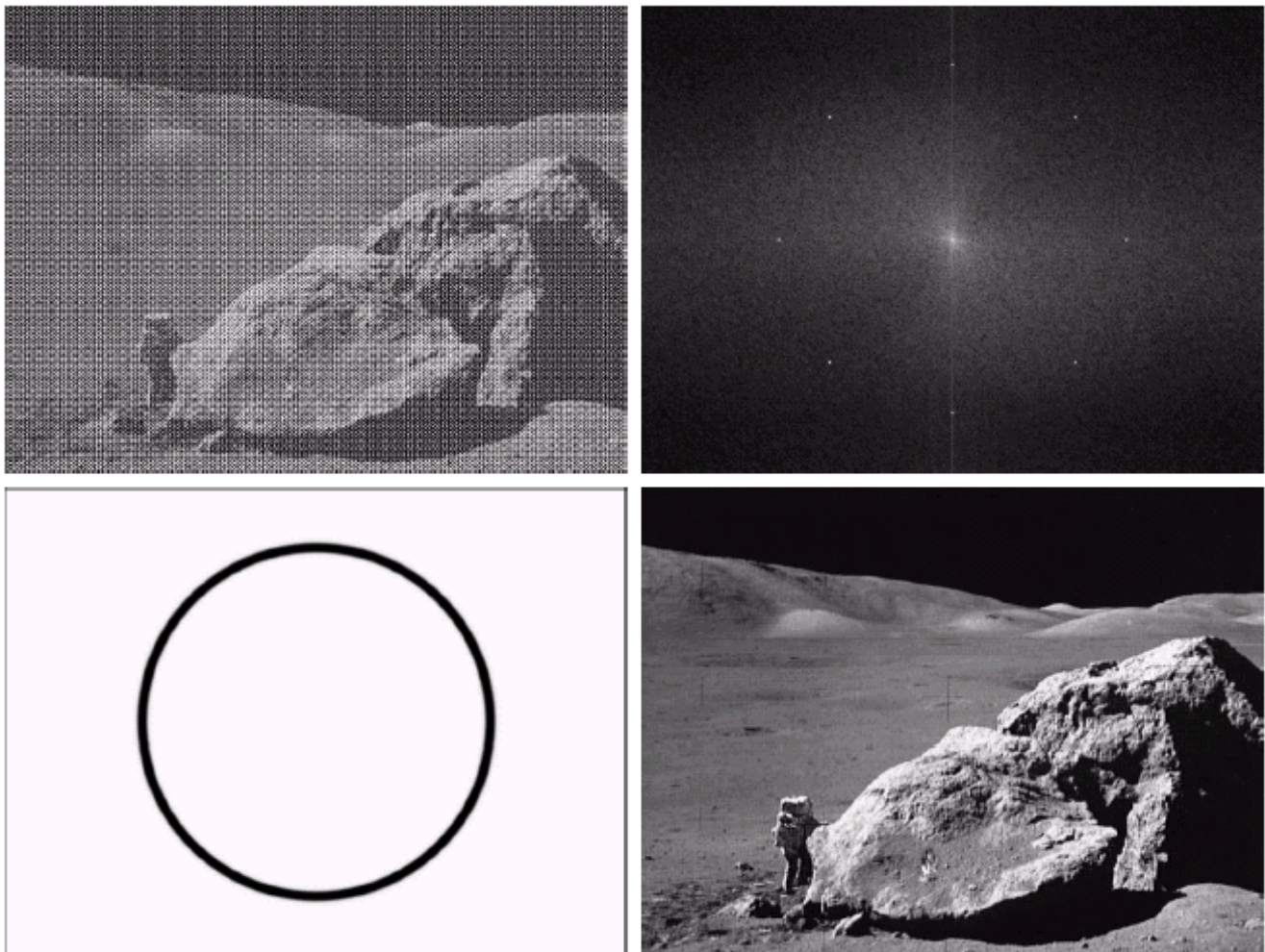
高斯带阻滤波器(Gaussian Bandreject Filters)

$$H(u, v) = 1 - e^{-\frac{1}{2} \left[ \frac{D^2(u, v) + D_0^2}{D(u, v)W} \right]^2}$$



**FIGURE 5.15** From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters.





a	b
c	d

**FIGURE 5.16**

(a) Image corrupted by sinusoidal noise.  
(b) Spectrum of (a).  
(c) Butterworth bandreject filter (white represents 1).  
(d) Result of filtering. (Original image courtesy of NASA.)

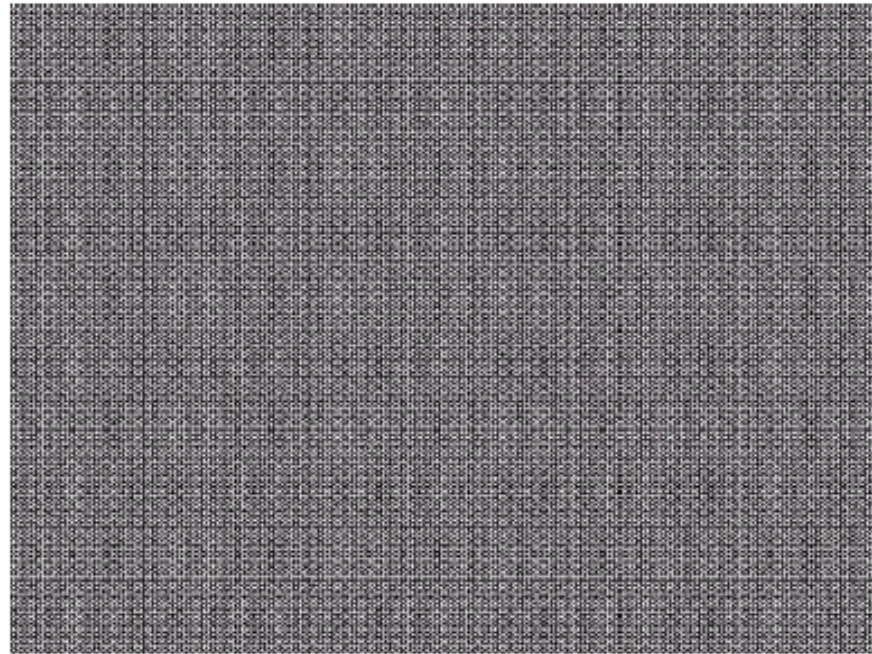


# 带通滤波器(Bandpass Filters)

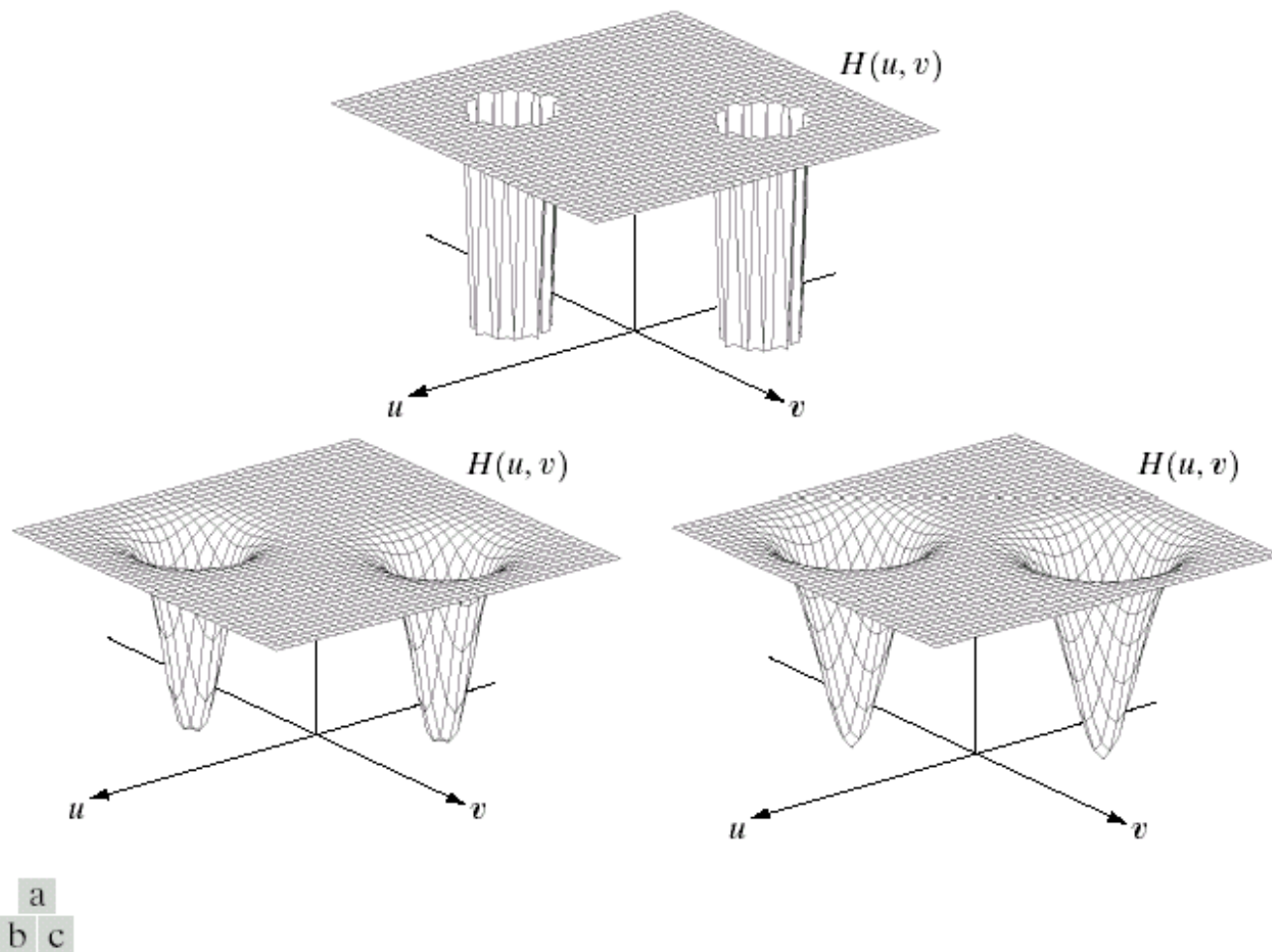
带通滤波器与带阻滤波器功能相反

$$H_{bp}(u, v) = 1 - H_{br}(u, v)$$

**FIGURE 5.17**  
Noise pattern of  
the image in  
Fig. 5.16(a)  
obtained by  
bandpass filtering.



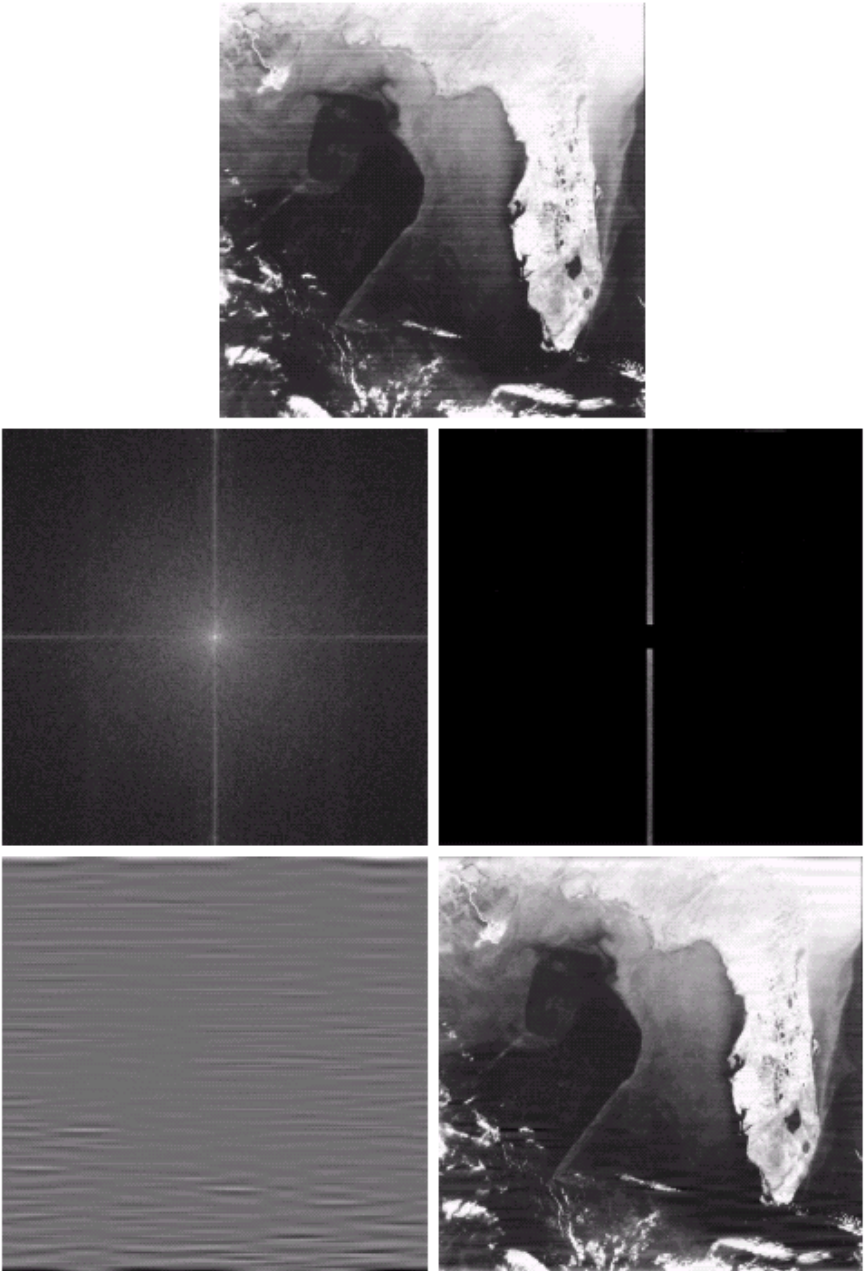
# 槽口滤波器



**FIGURE 5.18** Perspective plots of (a) ideal, (b) Butterworth (of order 2), and (c) Gaussian notch (reject) filters.

a  
b c  
d e

**FIGURE 5.19** (a) Satellite image of Florida and the Gulf of Mexico (note horizontal sensor scan lines). (b) Spectrum of (a). (c) Notch pass filter shown superimposed on (b). (d) Inverse Fourier transform of filtered image, showing noise pattern in the spatial domain. (e) Result of notch reject filtering. (Original image courtesy of NOAA.)





# 第5章 图像复原与重建

- 5.1 图像退化/复原过程的模型
- 5.2 噪声模型
- 5.3 只存在噪声的复原——空间滤波
- 5.4 用频率域消除周期噪声
- 5.5 线性、位置不变的退化
- 5.6 估计退化函数
- 5.7 逆滤波
- 5.8 最小均方误差（维纳）滤波
- 5.9 约束最小二乘方滤波
- 5.10 由投影重建图像



## 5.5 线性、位置不变的退化

降级过程:

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

线性系统:

$$H[af_1(x, y) + bf_2(x, y)] = aH[f_1(x, y)] + bH[f_2(x, y)]$$

位置不变系统:

$$H[f(x - \alpha, y - \beta)] = g(x - \alpha, y - \beta)$$

线性位置不变降级模型

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

$$G(u, v) = H(u, v)F(u, v) + N(u, v)$$





# 第5章 图像复原与重建

- 5.1 图像退化/复原过程的模型
- 5.2 噪声模型
- 5.3 只存在噪声的复原——空间滤波
- 5.4 用频率域消除周期噪声
- 5.5 线性、位置不变的退化
- 5.6 估计退化函数
- 5.7 逆滤波
- 5.8 最小均方误差（维纳）滤波
- 5.9 约束最小二乘方滤波
- 5.10 由投影重建图像



## 5.6 估计退化函数

### Estimating the Degradation Function

- 通过观测图像估计(Estimation by Image Observation)
- 通过实验估计(Estimation by Experimentation)
- 通过数学建模估计(Estimation by Modeling)



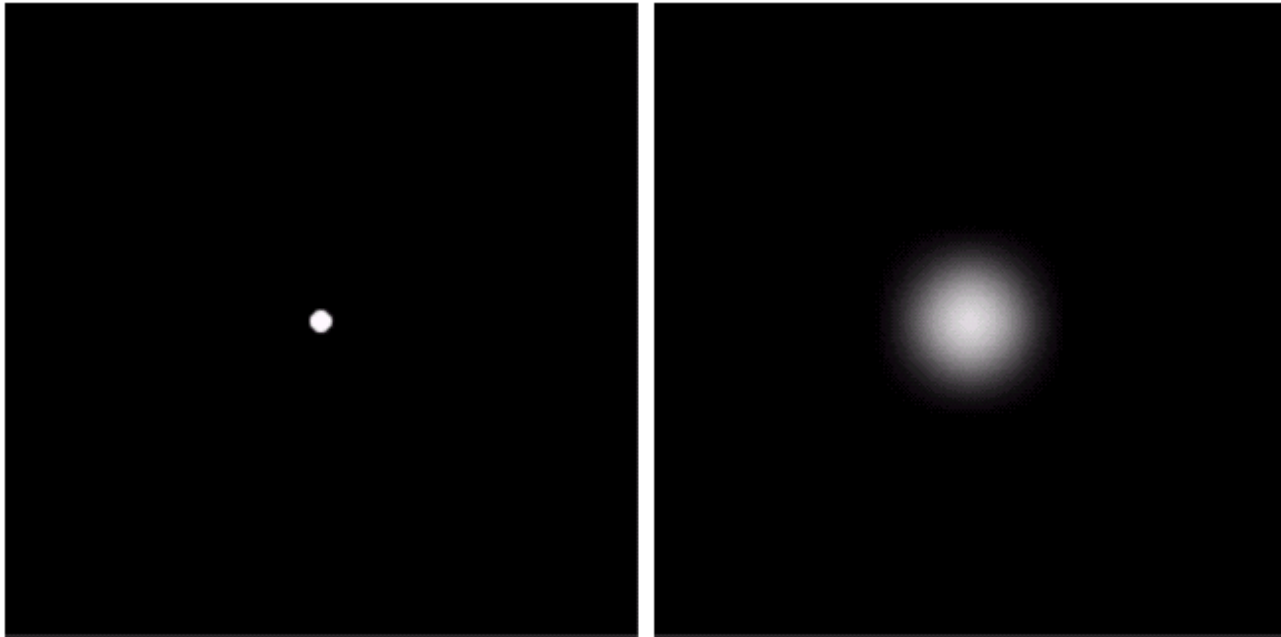


# 图像观测估计

- 假设：退化过程为线性、位置不变的
- 通过图像本身收集信息来估计退化函数 $H$ 
  - 从图像中选取有很强信号内容的区域（高对比度）
  - 对该图像进行处理，得到尽可能不模糊的结果
    - ✓ 如锐化处理、甚至手工方法处理
  - 基于处理之后后的图像频谱，估计退化函数频谱

$$H(u, v) = \frac{G(u, v)}{F(u, v)}$$

# 试验估计



a b

**FIGURE 5.24**

Degradation estimation by impulse characterization.  
(a) An impulse of light (shown magnified).  
(b) Imaged (degraded) impulse.

# 建模估计

a b  
c d

**FIGURE 5.25**

Illustration of the atmospheric turbulence model.

(a) Negligible turbulence.

(b) Severe turbulence,  $k = 0.0025$ .

(c) Mild turbulence,  $k = 0.001$ .

(d) Low turbulence,  $k = 0.00025$ .

(Original image courtesy of NASA.)



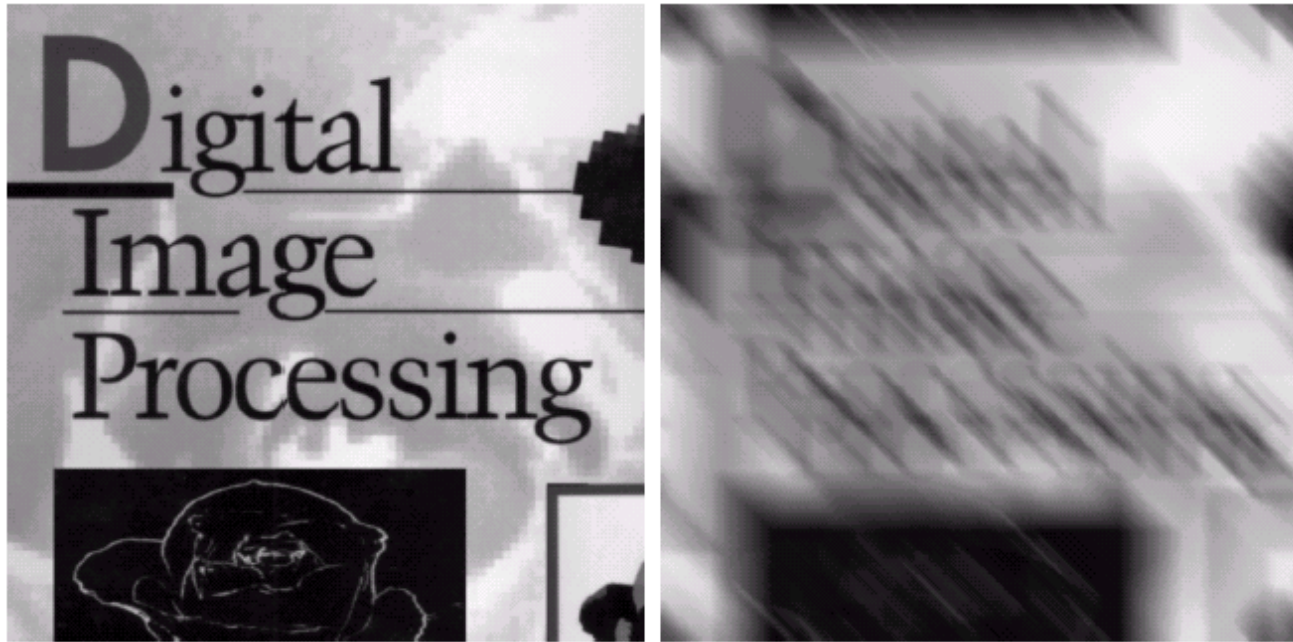
$$H(u, v) = e^{-k(u^2 + v^2)^{5/6}}$$

# 建模估计

## □ 从基本原理推导数学模型

$$g(x, y) = \int_0^T f[x - x_0(t), y - y_0(t)] dt$$

$$H(u, v) = \frac{T}{\pi(ua + vb)} \sin[\pi(ua + vb)] e^{-j\pi(ua + vb)} \quad x(t) = at / T; \quad y(t) = bt / T$$



a b

**FIGURE 5.26** (a) Original image. (b) Result of blurring using the function in Eq. (5.6-11) with  $a = b = 0.1$  and  $T = 1$ .



# 第5章 图像复原与重建

- 5.1 图像退化/复原过程的模型
- 5.2 噪声模型
- 5.3 只存在噪声的复原——空间滤波
- 5.4 用频率域消除周期噪声
- 5.5 线性、位置不变的退化
- 5.6 估计退化函数
- 5.7 **逆滤波**
- 5.8 最小均方误差（维纳）滤波
- 5.9 约束最小二乘方滤波
- 5.10 由投影重建图像



## 5.7 逆滤波

- 最简单且粗糙的恢复方法
- 降质图像频谱直接除以降质函数：

$$\hat{F}(u, v) = \frac{G(u, v)}{H(u, v)} \quad \hat{F}(u, v) = F(u, v) + \frac{N(u, v)}{H(u, v)}$$

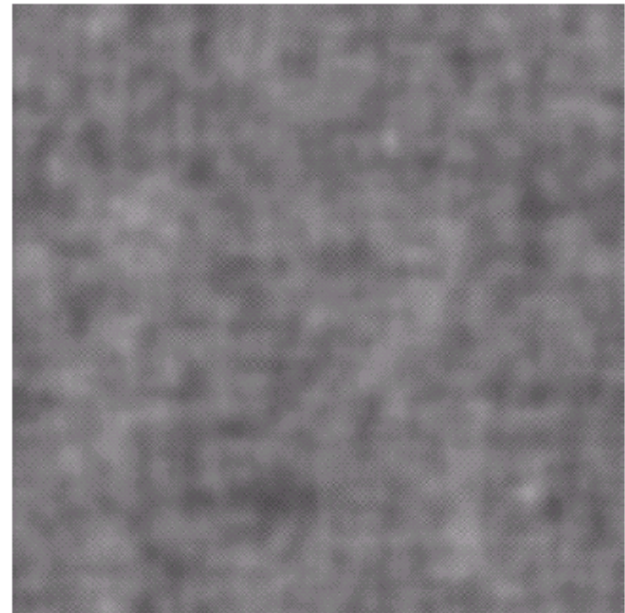
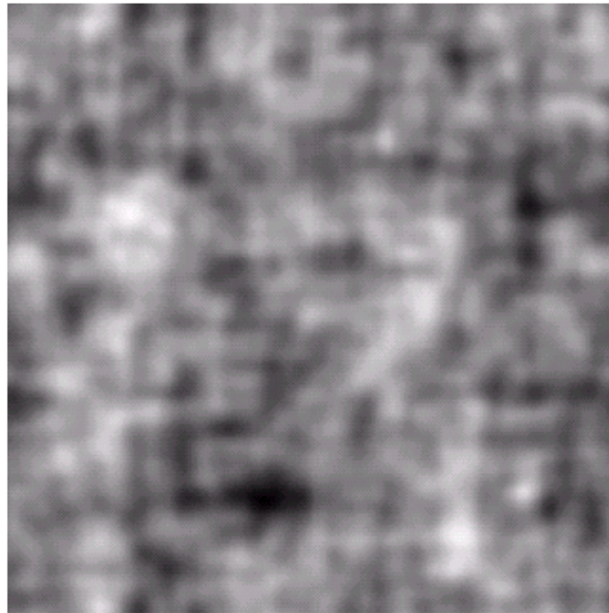
- 由于 $N(u, v)$ 未知，即使得到降质函数 $H(u, v)$ ，也难以精确重建
- $H(u, v)$ 在某些位置为0或者是非常小的值， $F(u, v)$ 被淹没
  - 解决方法：限制频谱范围



a b  
c d

**FIGURE 5.27**

Restoring Fig. 5.25(b) with Eq. (5.7-1). (a) Result of using the full filter. (b) Result with  $H$  cut off outside a radius of 40; (c) outside a radius of 70; and (d) outside a radius of 85.







# 第5章 图像复原与重建

- 5.1 图像退化/复原过程的模型
- 5.2 噪声模型
- 5.3 只存在噪声的复原——空间滤波
- 5.4 用频率域消除周期噪声
- 5.5 线性、位置不变的退化
- 5.6 估计退化函数
- 5.7 逆滤波
- 5.8 最小均方误差（维纳）滤波
- 5.9 约束最小二乘方滤波
- 5.10 由投影重建图像



## 5.8 最小均方误差（维纳）滤波

- 确定未污染图像  $f$  的一个估计  $\hat{f}$ ，使得他们之间的均方误差最小

$$e^2 = E \left\{ (f - \hat{f})^2 \right\}$$

$$F(u, v) = \left[ \frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + S_\eta(u, v) / S_f(u, v)} \right] G(u, v)$$

$$F(u, v) = \left[ \frac{1}{H(u, v)} \frac{|H(u, v)|^2}{|H(u, v)|^2 + K} \right] G(u, v)$$

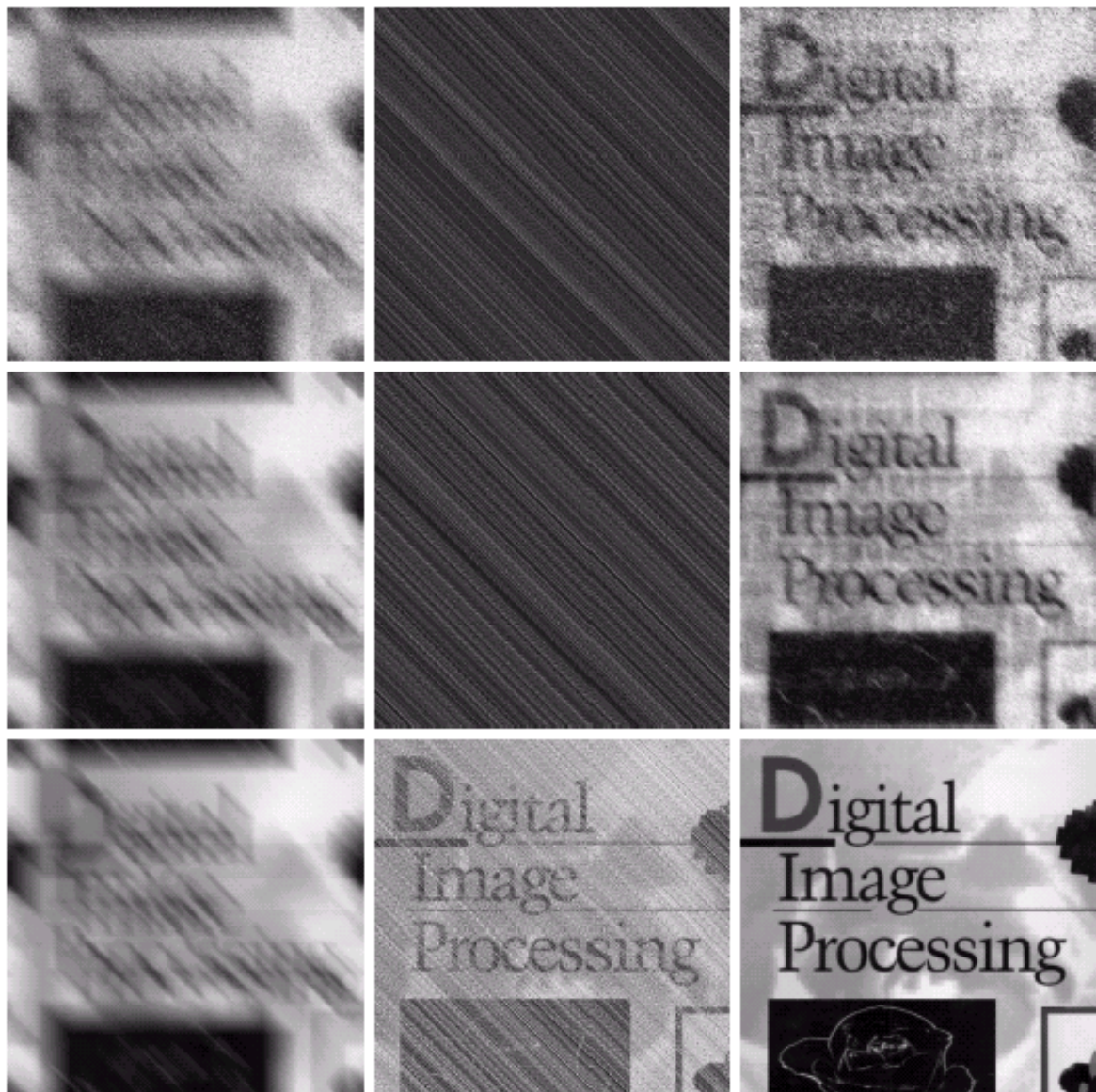
# 逆滤波与维纳滤波比较



a b c

**FIGURE 5.28** Comparison of inverse- and Wiener filtering. (a) Result of full inverse filtering of Fig. 5.25(b). (b) Radially limited inverse filter result. (c) Wiener filter result.

# 进一步比较



a b c  
d e f  
g h i

**FIGURE 5.29** (a) Image corrupted by motion blur and additive noise. (b) Result of inverse filtering. (c) Result of Wiener filtering. (d)–(f) Same sequence, but with noise variance one order of magnitude less. (g)–(i) Same sequence, but noise variance reduced by five orders of magnitude from (a). Note in (h) how the deblurred image is quite visible through a “curtain” of noise.



# 第5章 图像复原与重建

- 5.1 图像退化/复原过程的模型
- 5.2 噪声模型
- 5.3 只存在噪声的复原——空间滤波
- 5.4 用频率域消除周期噪声
- 5.5 线性、位置不变的退化
- 5.6 估计退化函数
- 5.7 逆滤波
- 5.8 最小均方误差（维纳）滤波
- 5.9 约束最小二乘方滤波
- 5.10 由投影重建图像





## 5.9 约束最小二乘方滤波

### □ 与维纳滤波相比

- 不需要知道信号和噪声的频谱，仅需知道噪声的均值和方差
- 维纳滤波的最优是平均意义上的，代数法是针对每一幅具体图像

$$g(x, y) = h(x, y) * f(x, y) + \eta(x, y) \quad \longleftrightarrow \quad g = Hf + \eta$$

$$\begin{bmatrix} 50 & 24 & 33 & 39 \\ 79 & 12 & 14 & 20 \\ 103 & 102 & 98 & 32 \\ 210 & 101 & 19 & 17 \end{bmatrix}$$

$$g, f, \eta : M \times N \Rightarrow MN \times 1$$

$$H : MN \times MN$$



## 5.9 约束最小二乘方滤波

$$\min C = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[ \nabla^2 f(x, y) \right]^2$$

$$\text{subject to } \|g - H\hat{f}\|^2 = \|\eta\|^2 \quad \|\eta\|^2 = MN(\sigma_\eta^2 + m_\eta^2)$$

$$F(u, v) = \left[ \frac{H^*(u, v)}{|H(u, v)|^2 + \gamma |P(u, v)|^2} \right] G(u, v)$$

$$p(x, y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \begin{array}{c} \text{FFT} \\ \longleftrightarrow \end{array} P(u, v)$$



# 5.9 约束最小二乘方滤波



a b c

**FIGURE 5.30** Results of constrained least squares filtering. Compare (a), (b), and (c) with the Wiener filtering results in Figs. 5.29(c), (f), and (i), respectively.



# 第5章 图像复原与重建

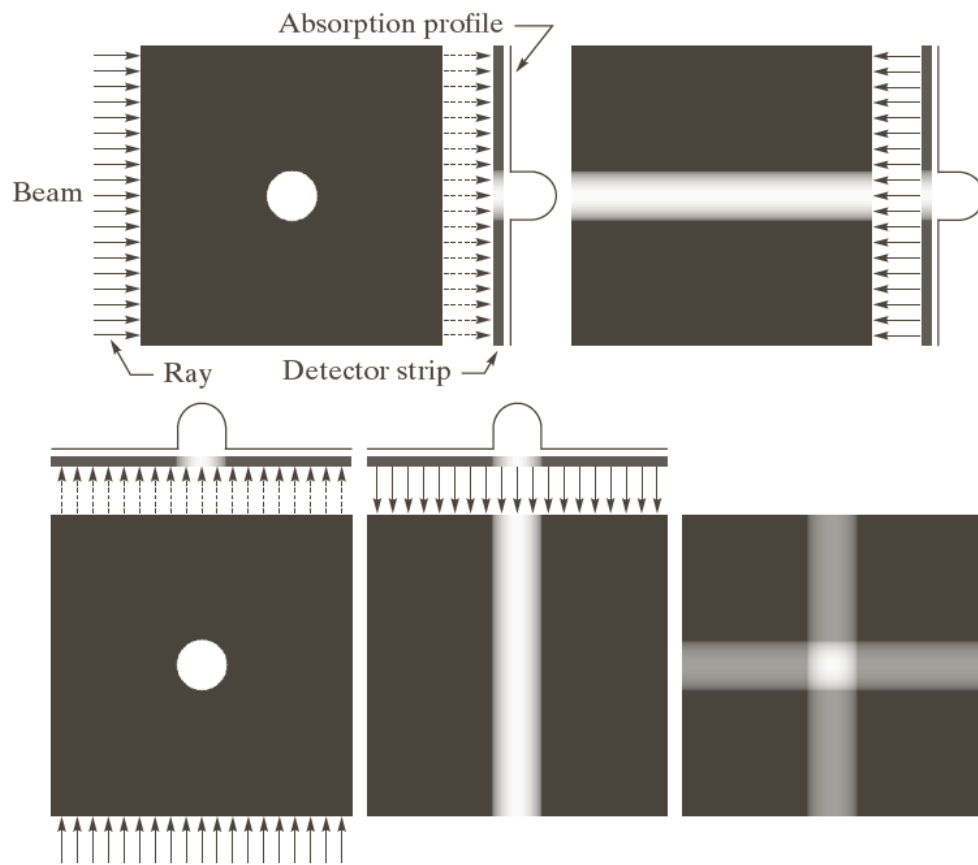
- 5.1 图像退化/复原过程的模型
- 5.2 噪声模型
- 5.3 只存在噪声的复原——空间滤波
- 5.4 用频率域消除周期噪声
- 5.5 线性、位置不变的退化
- 5.6 估计退化函数
- 5.7 逆滤波
- 5.8 最小均方误差（维纳）滤波
- 5.9 约束最小二乘方滤波
- 5.10 由投影重建图像

# 5.10 由投影重建图像

## 计算机断层 (CT) 原理

a b  
c d e

**FIGURE 5.32**  
 (a) Flat region showing a simple object, an input parallel beam, and a detector strip.  
 (b) Result of back-projecting the sensed strip data (i.e., the 1-D absorption profile).  
 (c) The beam and detectors rotated by 90°.  
 (d) Back-projection.  
 (e) The sum of (b) and (d). The intensity where the back-projections intersect is twice the intensity of the individual back-projections.



# 5.10 由投影重建图像

## 计算机断层 (CT) 原理

a	b	c
d	e	f

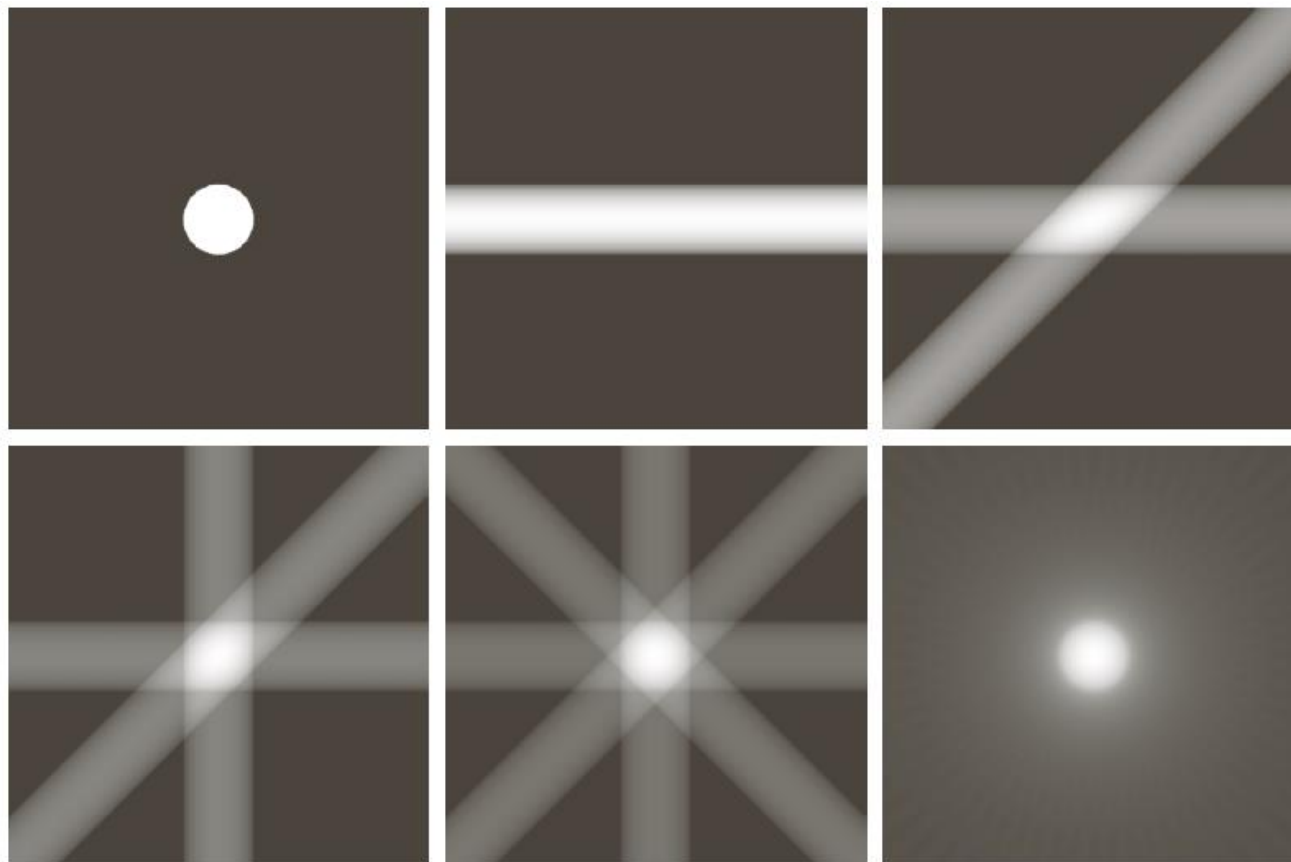
**FIGURE 5.33**

(a) Same as Fig. 5.32(a).

(b)–(e)

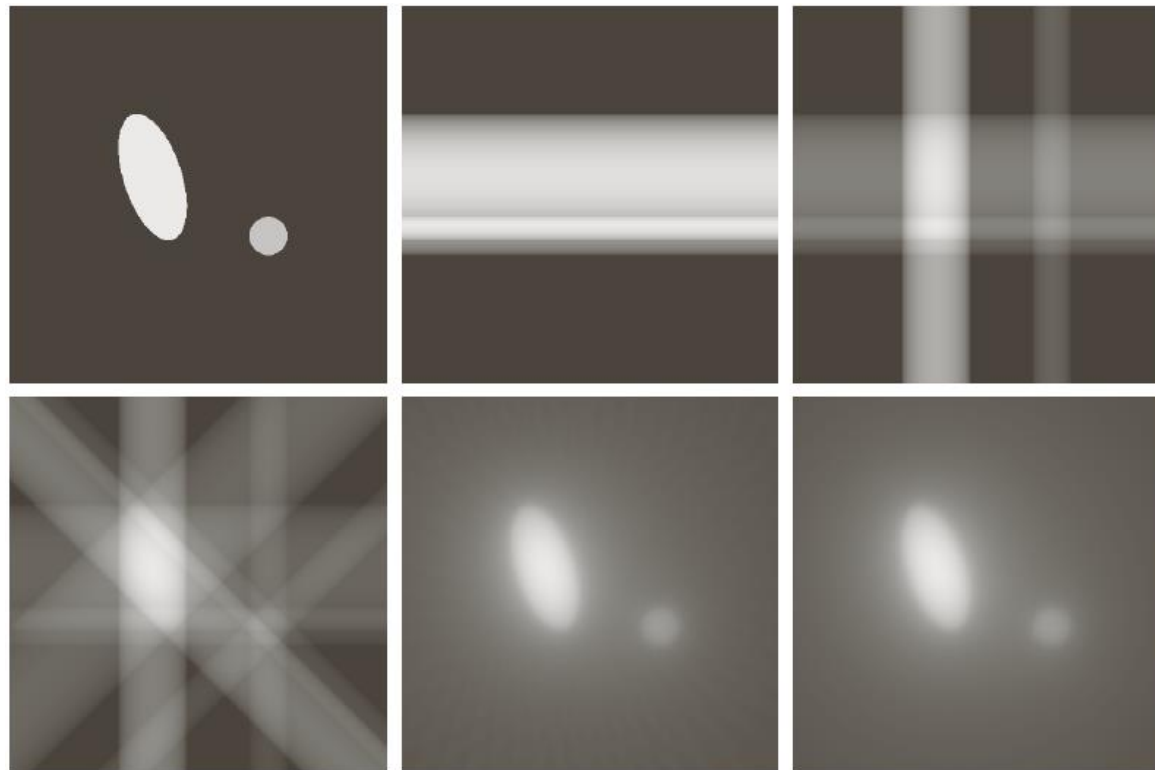
Reconstruction using 1, 2, 3, and 4 backprojections  $45^\circ$  apart.

(f) Reconstruction with 32 backprojections  $5.625^\circ$  apart (note the blurring).



# 5.10 由投影重建图像

## 计算机断层 (CT) 原理



a b c  
d e f

**FIGURE 5.34** (a) A region with two objects. (b)–(d) Reconstruction using 1, 2, and 4 backprojections  $45^\circ$  apart. (e) Reconstruction with 32 backprojections  $5.625^\circ$  apart. (f) Reconstruction with 64 backprojections  $2.8125^\circ$  apart.

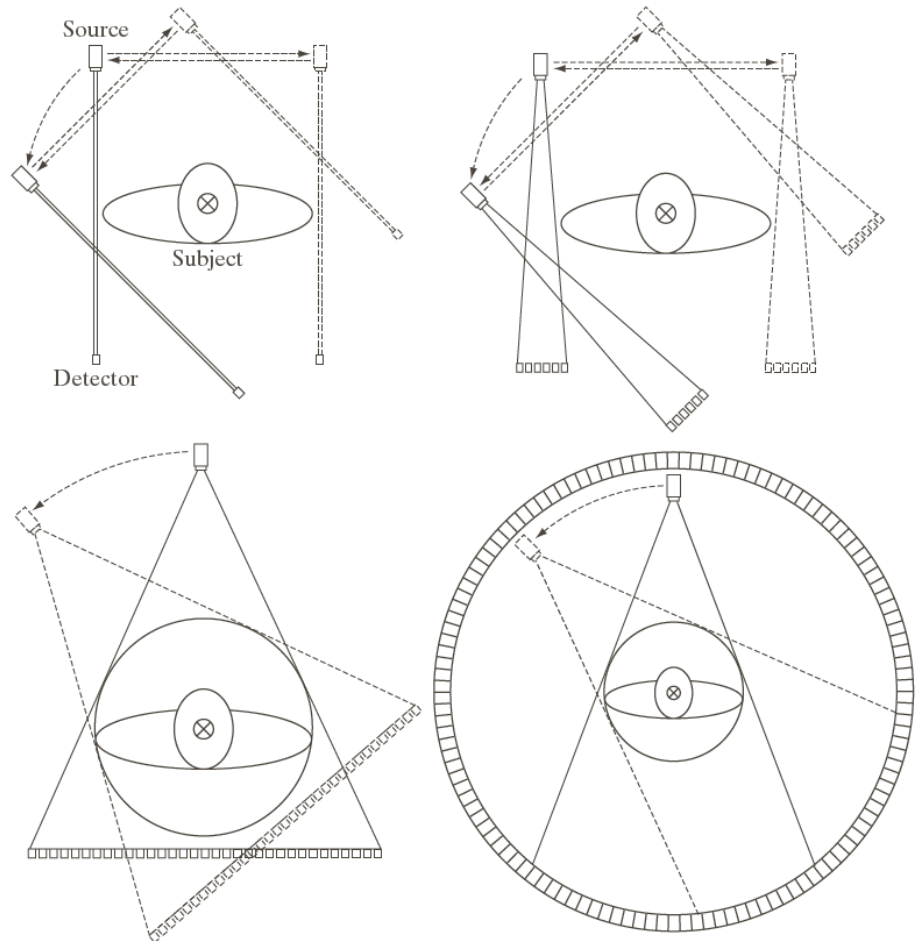
# 5.10 由投影重建图像

## 计算机断层 (CT) 原理

- 1919 Johann Radon 提出Radon变换
- 1962年Allan M. Cormack, Tuffs Univ. 设计CT原型
- G. N. Hounsfield, 英国伦敦EMI公司工程师, 同时设计第一台医用CT
- A. M. Cormack和G. N. Hounsfield获1979年诺贝尔医学奖

a b  
c d

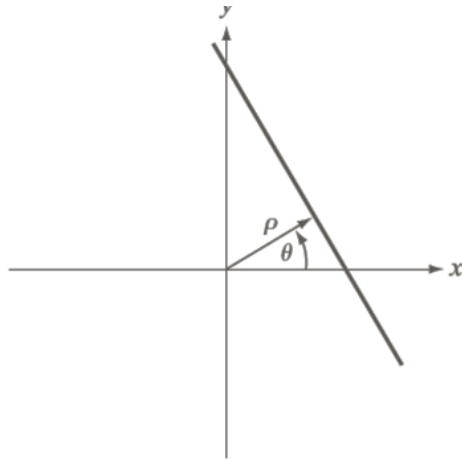
**FIGURE 5.35** Four generations of CT scanners. The dotted arrow lines indicate incremental linear motion. The dotted arrow arcs indicate incremental rotation. The cross-mark on the subject's head indicates linear motion perpendicular to the plane of the paper. The double arrows in (a) and (b) indicate that the source/detector unit is translated and then brought back into its original position.



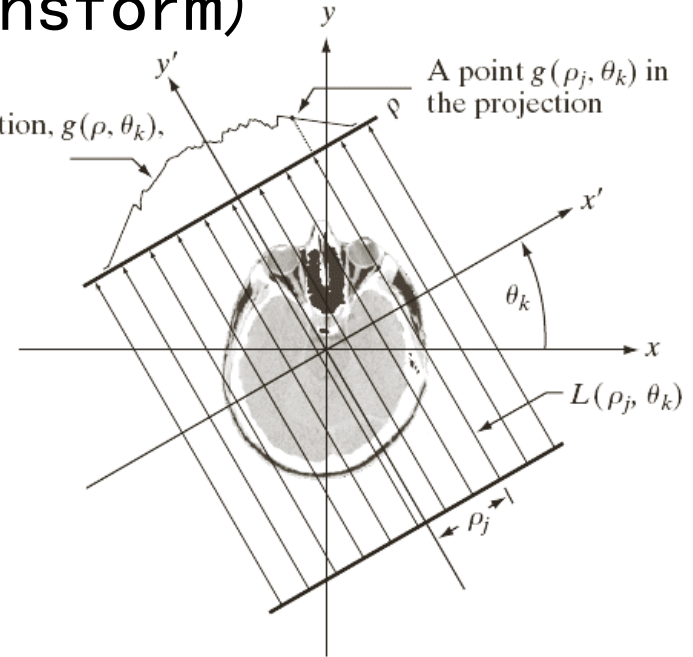
# 5.10 由投影重建图像

## 投影和雷登变换 (Radon Transform)

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



Complete projection,  $g(\rho, \theta_k)$ ,  
for a fixed angle



$$g(\rho_j, \theta_k) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) \delta(x \cos \theta_k + y \sin \theta_k - \rho_j) dx dy$$

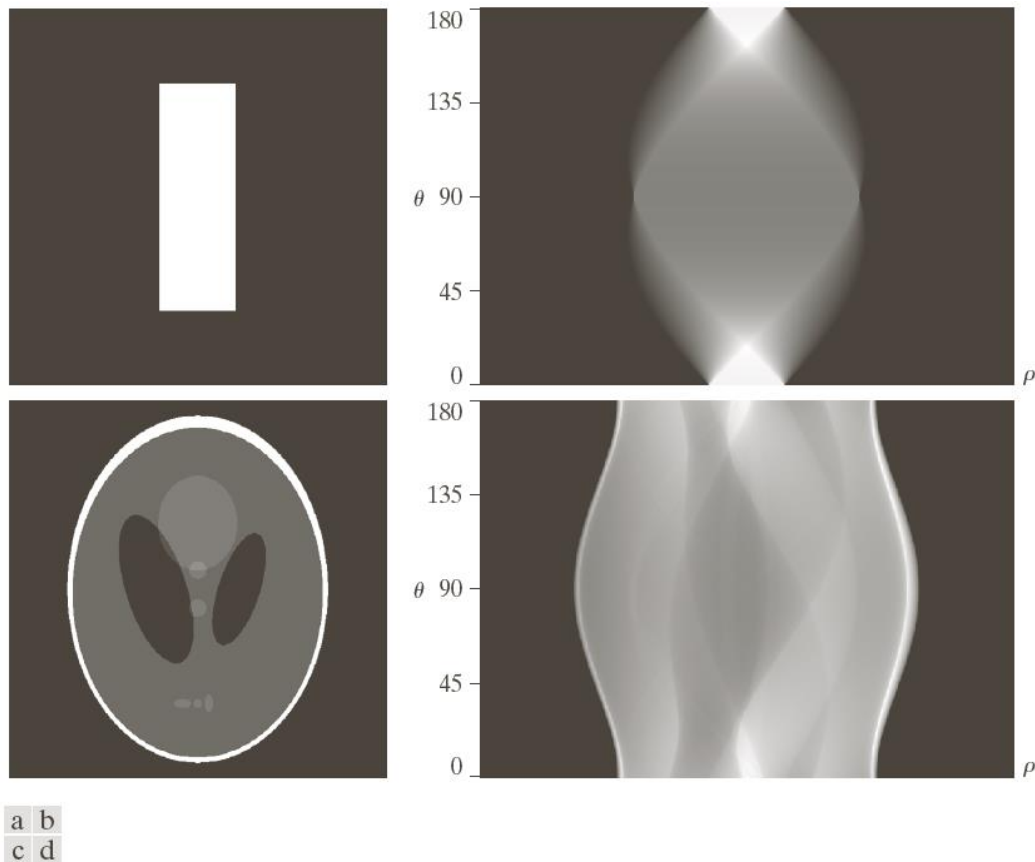
$$g(\rho, \theta) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - \rho) dx dy$$

$$g(\rho, \theta) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \delta(x \cos \theta + y \sin \theta - \rho) dx dy$$



# 5.10 由投影重建图像

正弦图(sinogram):  $g(\rho, \theta)$



**FIGURE 5.39** Two images and their sinograms (Radon transforms). Each row of a sinogram is a projection along the corresponding angle on the vertical axis. Image (c) is called the *Shepp-Logan phantom*. In its original form, the contrast of the phantom is quite low. It is shown enhanced here to facilitate viewing.

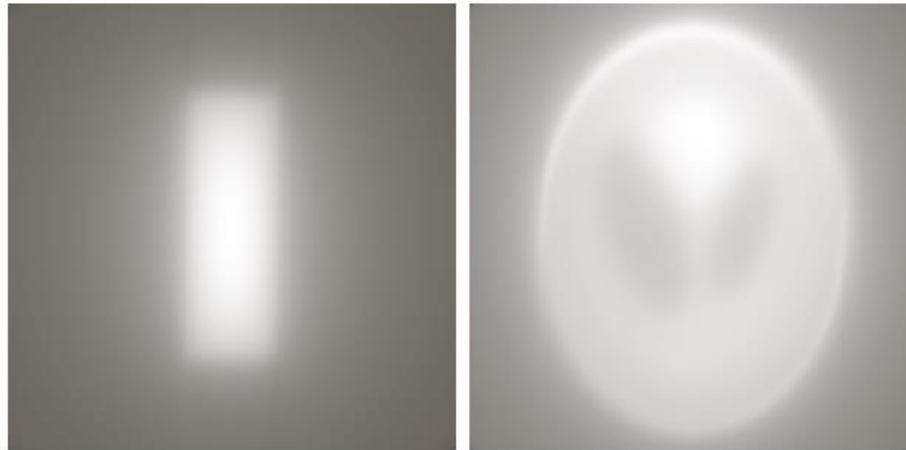
## 5.10 由投影重建图像

从正弦图  $g(\rho, \theta)$  得到反投影图像

$$f_{\theta_k}(x, y) = g(\rho, \theta_k) = g(x \cos \theta_k + y \sin \theta_k, \theta_k) dx dy$$

$$f_{\theta}(x, y) = g(x \cos \theta + y \sin \theta, \theta) dx dy$$

$$f(x, y) = \int_0^{\pi} f_{\theta}(x, y) \quad f(x, y) = \sum_{\theta=0}^{\pi} f_{\theta}(x, y)$$



a b

**FIGURE 5.40**  
Backprojections  
of the sinograms  
in Fig. 5.39.



## 5.10 由投影重建图像

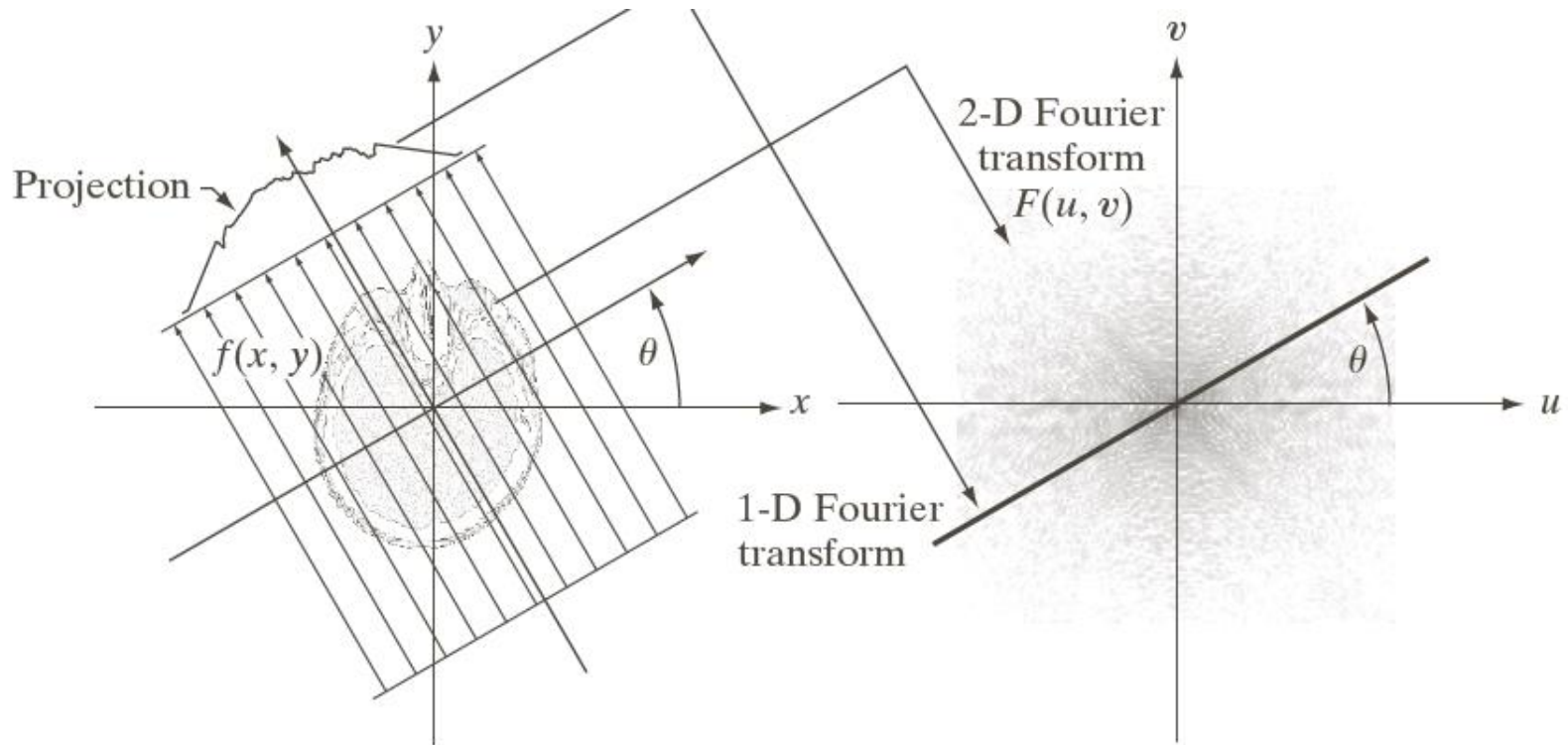
### 傅里叶切片定理

$\rho$  投影的一维傅里叶变换

$$\begin{aligned} G(\omega, \theta) &= \int_{-\infty}^{+\infty} g(\rho, \theta) e^{-j2\pi\omega\rho} d\rho \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - \rho) e^{-j2\pi\omega\rho} dx dy d\rho \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) \left[ \int_{-\infty}^{+\infty} \delta(x \cos \theta + y \sin \theta - \rho) e^{-j2\pi\omega\rho} d\rho \right] dx dy \\ &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) e^{-j2\pi\omega(x \cos \theta + y \sin \theta)} dx dy \\ &= \left[ \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) e^{-j2\pi(ux+vy)} dx dy \right]_{u=\omega \cos \theta; v=\omega \sin \theta} \\ &= [F(u, v)]_{u=\omega \cos \theta; v=\omega \sin \theta} \quad (F(u, v) \text{ 是 } f(x, y) \text{ 的傅里叶变换}) \\ &= F(\omega \cos \theta, \omega \sin \theta) \end{aligned}$$

# 5.10 由投影重建图像

## 傅里叶切片定理





## 5.10 由投影重建图像

### 使用平行射线束滤波反投影的重建

$$\begin{aligned} f(x, y) &= \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} F(u, v) e^{j2\pi(ux+vy)} dudv \\ &= \int_0^{2\pi} \int_0^{+\infty} F(\omega \cos \theta, \omega \sin \theta) e^{j2\pi\omega(x \cos \theta + y \sin \theta)} \omega d\omega d\theta \quad (u = \omega \cos \theta, v = \omega \sin \theta) \\ &= \int_0^{2\pi} \int_0^{+\infty} G(\omega, \theta) e^{j2\pi\omega(x \cos \theta + y \sin \theta)} \omega d\omega d\theta \quad (F(\omega \cos \theta, \omega \sin \theta) = G(\omega, \theta)) \\ &= \int_0^{\pi} \int_{-\infty}^{+\infty} |\omega| G(\omega, \theta) e^{j2\pi\omega(x \cos \theta + y \sin \theta)} d\omega d\theta \\ &= \int_0^{\pi} \left[ \int_{-\infty}^{+\infty} |\omega| G(\omega, \theta) e^{j2\pi\omega\rho} d\omega \right]_{\rho=x \cos \theta + y \sin \theta} d\theta \end{aligned}$$

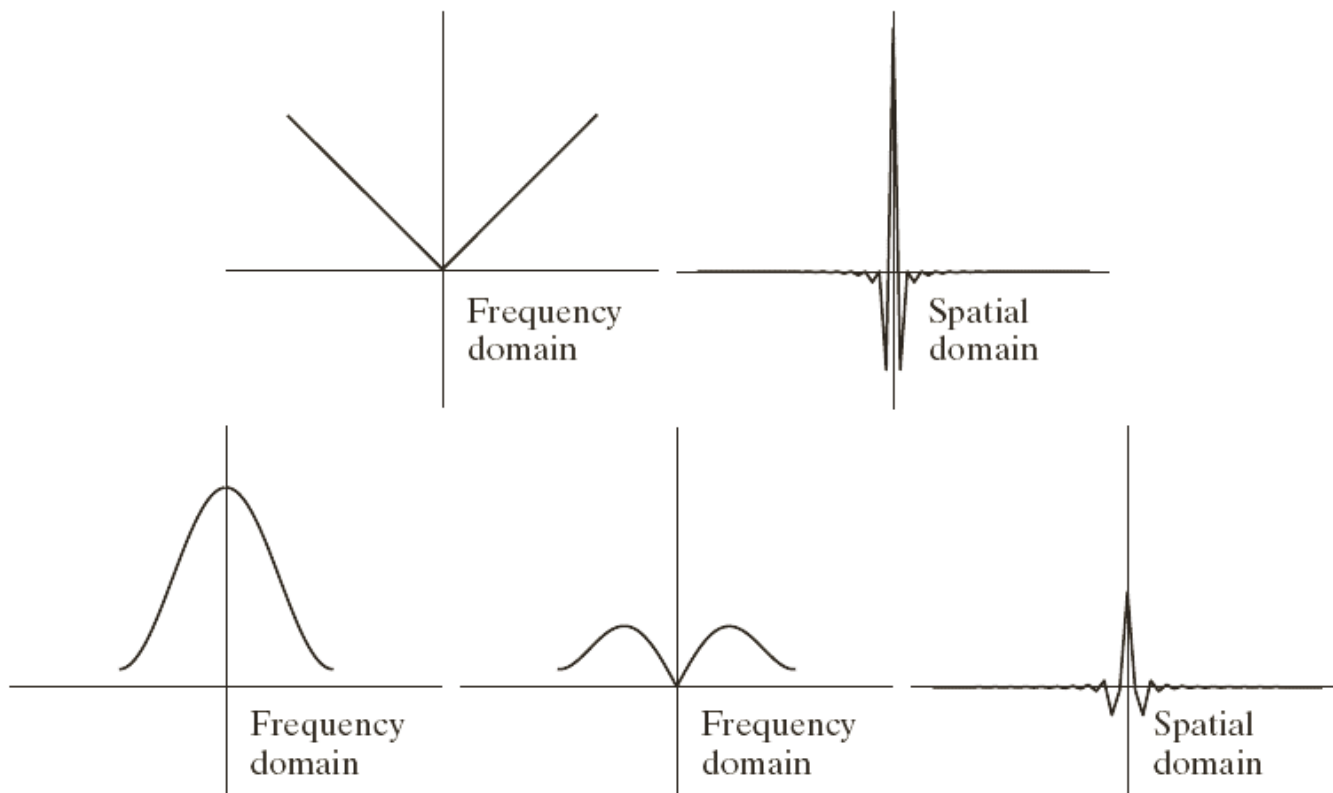
# 5.10 由投影重建图像

## 使用平行射线束滤波反投影的重建

a b  
c d e

**FIGURE 5.42**

(a) Frequency domain plot of the filter  $|\omega|$  after band-limiting it with a box filter. (b) Spatial domain representation. (c) Hamming windowing function. (d) Windowed ramp filter, formed as the product of (a) and (c). (e) Spatial representation of the product (note the decrease in ringing).



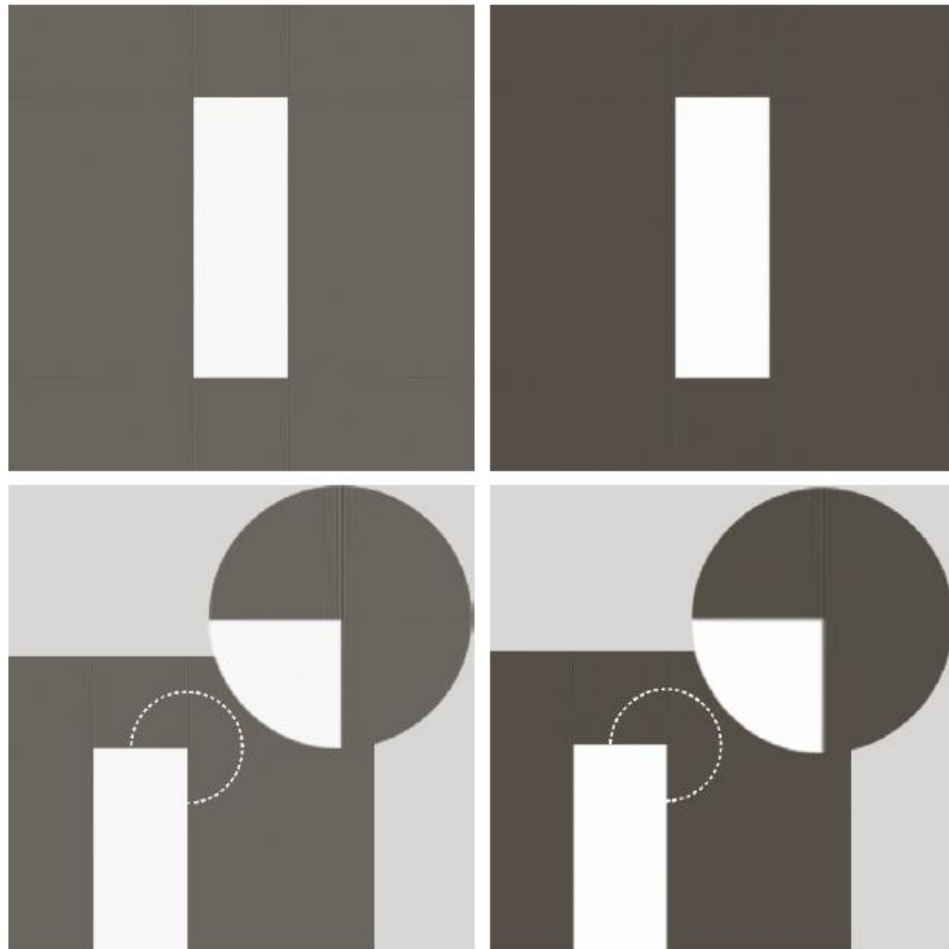
# 5.10 由投影重建图像

## 使用平行射线约束滤波反投影的重建

a	b
c	d

**FIGURE 5.43**

Filtered back-projections of the rectangle using (a) a ramp filter, and (b) a Hamming-windowed ramp filter. The second row shows zoomed details of the images in the first row. Compare with Fig. 5.40(a).





# 5.10 由投影重建图像

## 使用平行射线束滤波反投影的重建

a b

**FIGURE 5.44**  
Filtered backprojections of the head phantom using (a) a ramp filter, and (b) a Hamming-windowed ramp filter. Compare with Fig. 5.40(b).





## 5.10 由投影重建图像

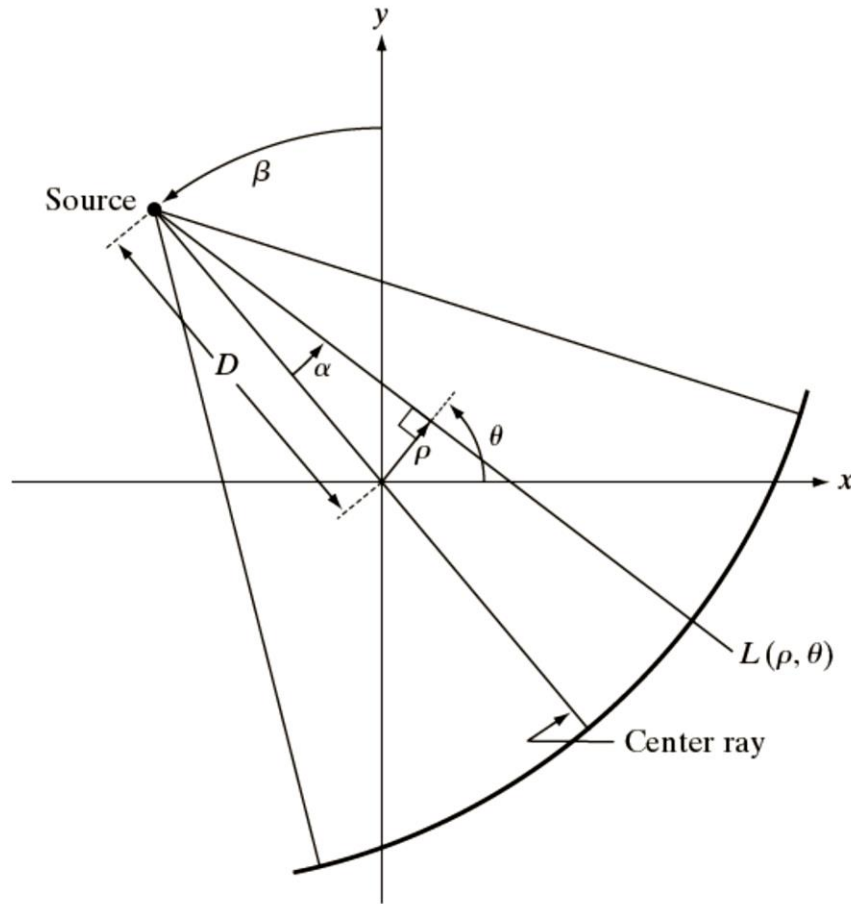
### 使用平行射线束滤波反投影的重建

$$\begin{aligned} f(x, y) &= \int_0^\pi \left[ \int_{-\infty}^{+\infty} |\omega| G(\omega, \theta) e^{j2\pi\omega\rho} d\omega \right]_{\rho=x\cos\theta+y\sin\theta} d\theta \\ &= \int_0^\pi \left[ s(\rho) * g(\rho, \theta) \right]_{\rho=x\cos\theta+y\sin\theta} d\theta \\ &= \int_0^\pi \left[ \int_{-\infty}^{+\infty} g(\rho, \theta) s(x\cos\theta + y\sin\theta - \rho) d\rho \right] d\theta \end{aligned}$$

# 5.10 由投影重建图像

## 使用平行射线约束滤波反投影的重建

**FIGURE 5.45**  
Basic fan-beam geometry. The line passing through the center of the source and the origin (assumed here to be the center of rotation of the source) is called the *center ray*.





## 5.10 由投影重建图像

### 使用扇形射线束滤波反投影的重建

$$\begin{aligned} f(x, y) &= \frac{1}{2} \int_0^{2\pi} \left[ \int_{-T}^{+T} g(\rho, \theta) s(x \cos \theta + y \sin \theta - \rho) d\rho \right] d\theta \\ &= \frac{1}{2} \int_0^{2\pi} \int_{-T}^{+T} g(\rho, \theta) s[r \cos(\theta - \varphi) - \rho] d\rho d\theta \end{aligned}$$

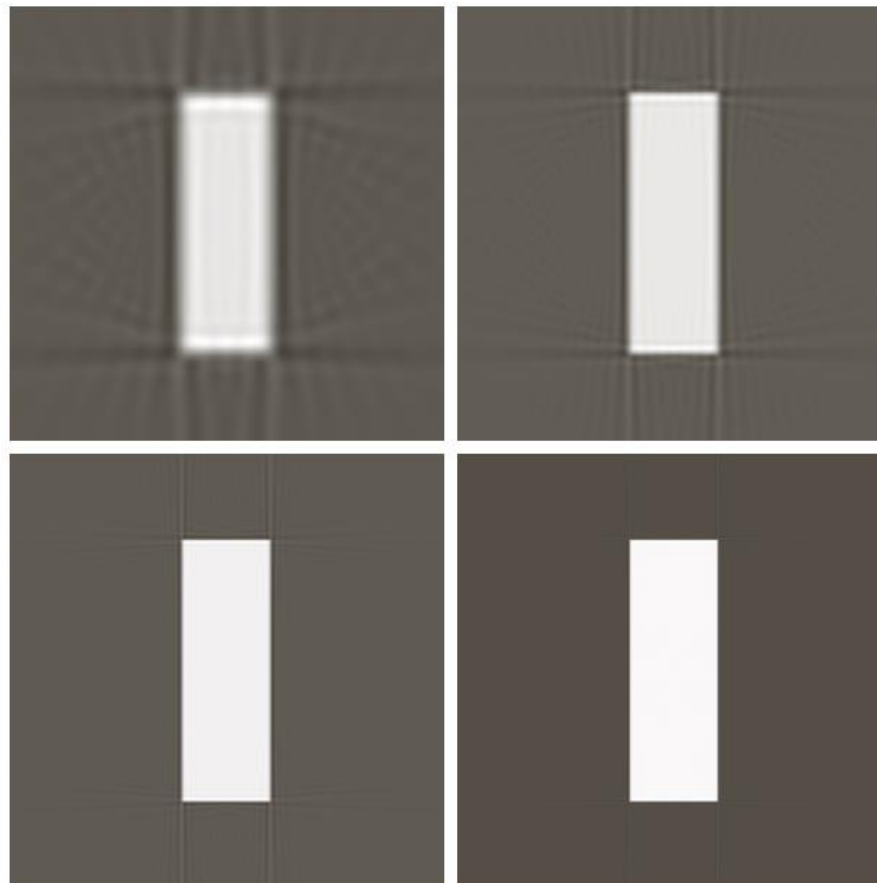
$$\begin{aligned} f(r, \varphi) &= \frac{1}{2} \int_{\alpha}^{2\pi - \alpha} \int_{\arcsin(-T/D)}^{\arcsin(T/D)} g(D \sin \alpha, \alpha + \beta) s[r \cos(\beta + \alpha - \varphi) - D \sin \alpha] D d\alpha d\beta \\ &= \frac{1}{2} \int_0^{2\pi} \int_{-\alpha_m}^{+\alpha_m} p(\alpha, \beta) s[r \cos(\beta + \alpha - \varphi) - D \sin \alpha] D d\alpha d\beta \end{aligned}$$

## 5.10 由投影重建图像

### 使用扇形射线束滤波反投影的重建

a b  
c d

**FIGURE 5.48**  
Reconstruction of  
the rectangle  
image from  
filtered fan  
backprojections.  
(a)  $1^\circ$  increments  
of  $\alpha$  and  $\beta$ .  
(b)  $0.5^\circ$   
increments.  
(c)  $0.25^\circ$   
increments.  
(d)  $0.125^\circ$   
increments.  
Compare (d) with  
Fig. 5.43(b).



## 5.10 由投影重建图像

### 使用扇形射线束滤波反投影的重建

a b  
c d

**FIGURE 5.49**

Reconstruction of the head phantom image from filtered fan backprojections.

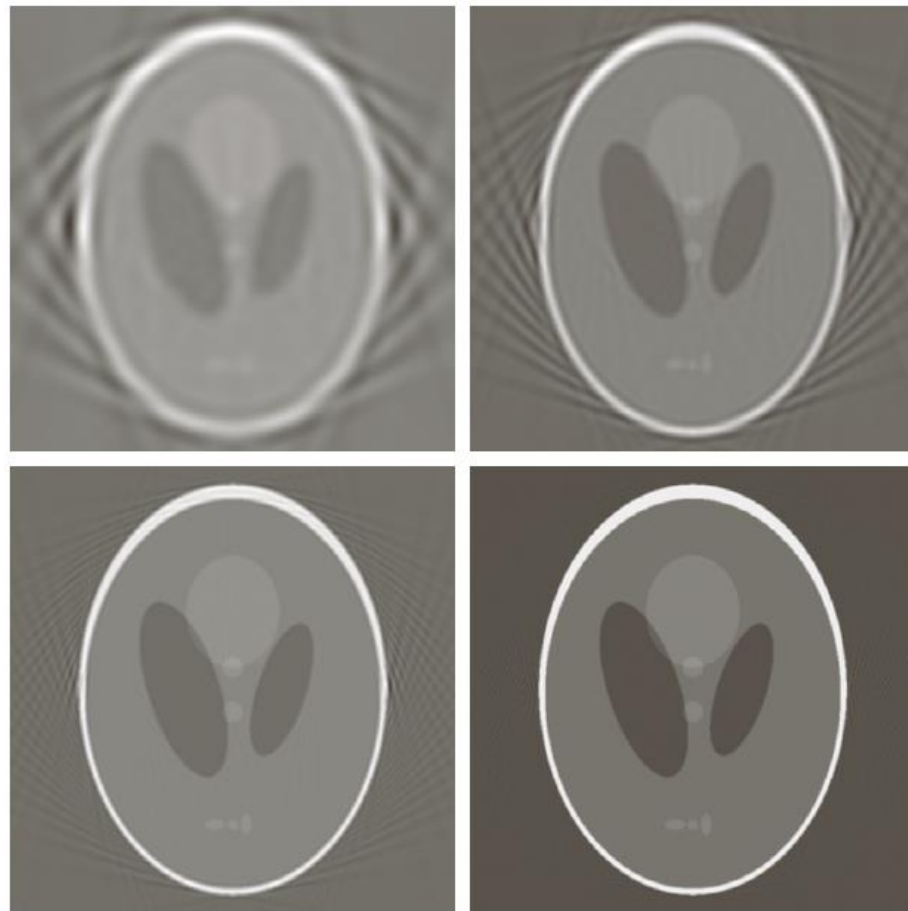
(a)  $1^\circ$  increments of  $\alpha$  and  $\beta$ .

(b)  $0.5^\circ$  increments.

(c)  $0.25^\circ$  increments.

(d)  $0.125^\circ$  increments.

Compare (d) with Fig. 5.44(b).





# 第6章 彩色图像处理

- 彩色是一个强有力的描绘子，常常可简化从场景中提取和识别目标
- 人可以辨别几千种彩色色调和亮度，但只能辨别几十种灰度色调
- 彩色图像处理可分为两个主要领域：全彩色处理和伪彩色处理
  - 全彩色处理：通常要求图像用全彩色传感器获取
  - 伪彩色处理：对一种特定的单色灰度或灰度范围赋予一种颜色
- 前面几章介绍的一些灰度方法可直接用于彩色图像。





# 第6章 彩色图像处理

- 6.1 彩色基础
- 6.2 彩色模型
- 6.3 伪彩色图像处理
- 6.4 真彩色图像处理
- 6.5 彩色变换
- 6.6 平滑和锐化
- 6.7 基于彩色的图像分割
- 6.8 彩色图像中的噪声
- 6.9 彩色图像压缩

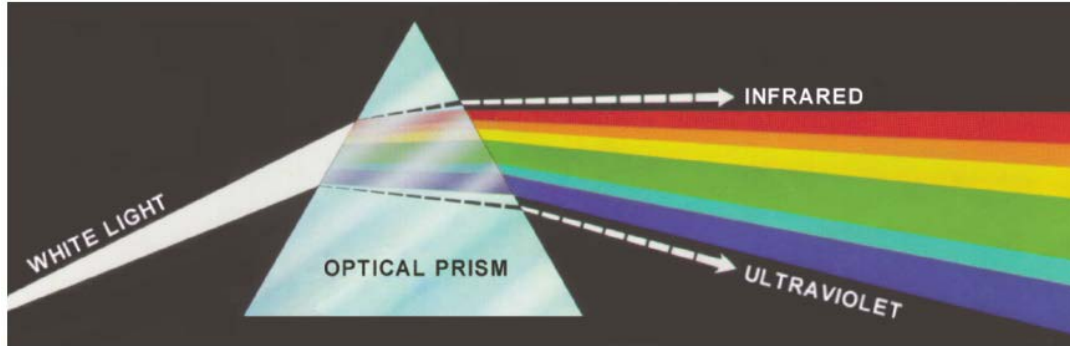


# 第6章 彩色图像处理

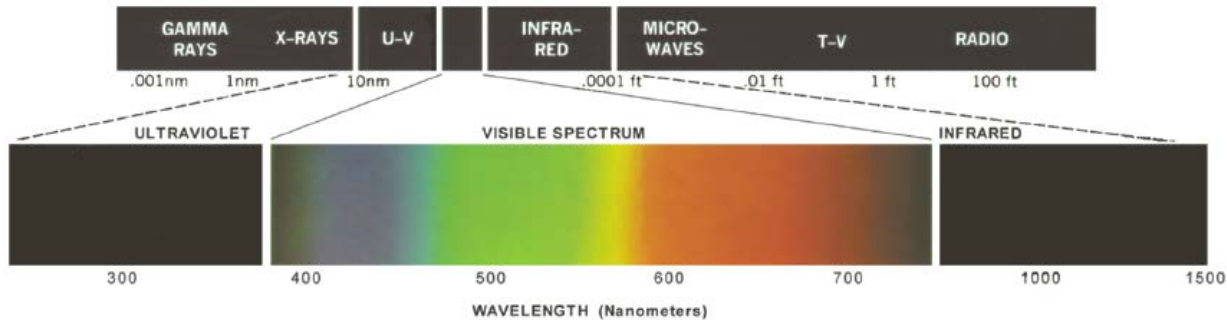
- 6.1 彩色基础
- 6.2 彩色模型
- 6.3 伪彩色图像处理
- 6.4 真彩色图像处理
- 6.5 彩色变换
- 6.6 平滑和锐化
- 6.7 基于彩色的图像分割
- 6.8 彩色图像中的噪声
- 6.9 彩色图像压缩

# 6.1 彩色基础

- 1661年，艾萨克·牛顿发现了太阳光的分解



- 彩色光大约覆盖电磁波谱中的400~700nm的范围

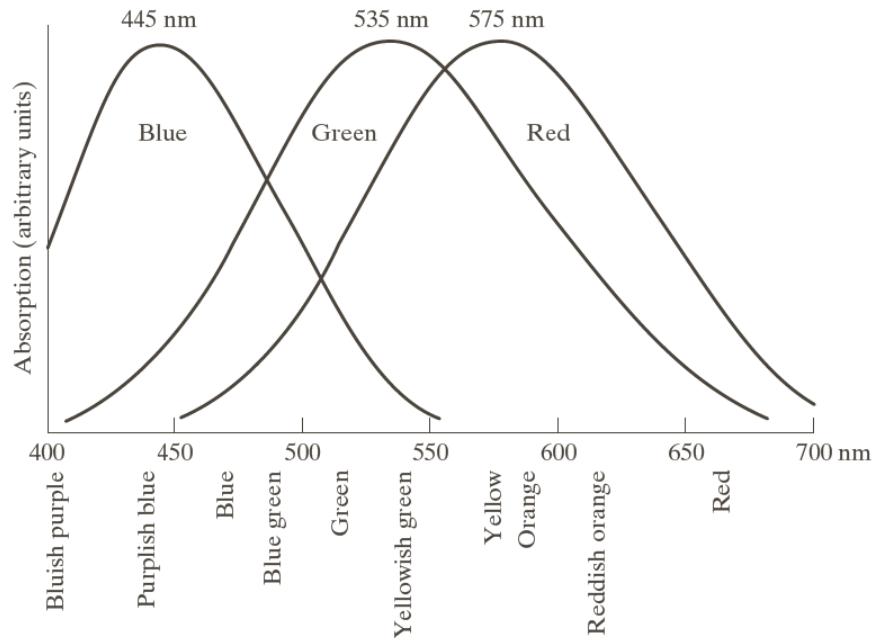




# 6.1 彩色基础

## □ 人的视觉

- 人眼中的锥状细胞是负责彩色视觉的传感器
- 人眼中600~700万个锥状细胞分为3个主要的感知类别，分别对应于红色、绿色和蓝色。65%细胞对红光敏感，33%对绿光敏感，只有2%对蓝光敏感



以波长为函数，人眼中的红色、绿色和蓝色锥状细胞对光的吸收曲线

# 6.1 彩色基础

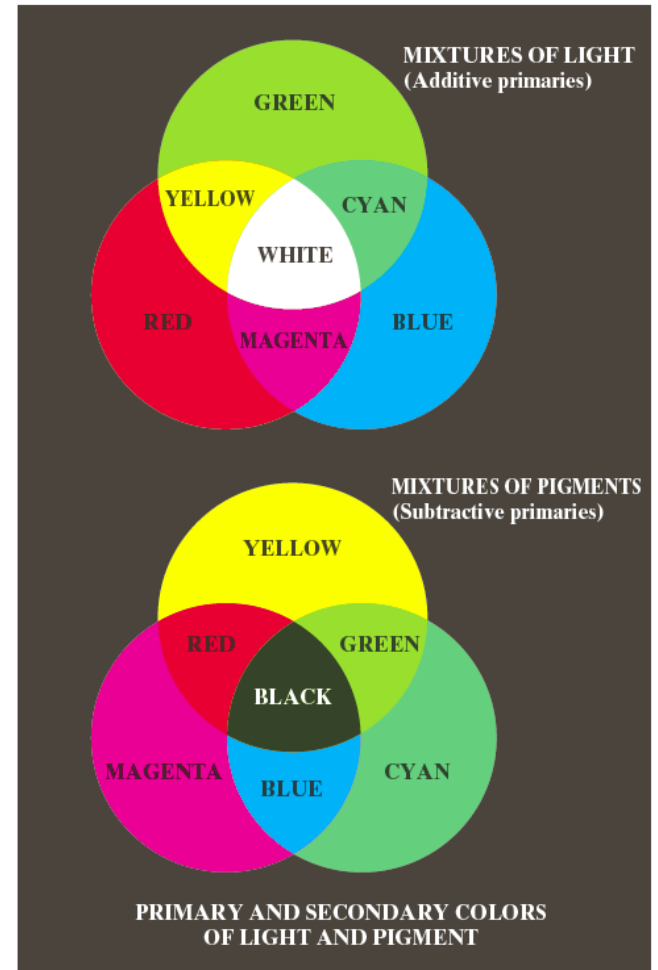
## □ 光的原色与颜料的原色

### □ 光的原色

- 红(R)、绿(G)、蓝(B)
- 满足加色原理
- 应用：CRT显示器

### □ 颜料、着色剂的原色

- 深红色(M)、青色(C)、黄色(Y)
- 满足减色原理
- 应用：彩色打印机



# 色度图

- 颜色的物理特征：亮度(Intensity)、色调(Hue)、饱和度(Saturation)
  - 亮度：体现了无色的强度概念
  - 色调：光波混合中与主波长有关的属性，观察者感知的主要颜色
  - 饱和度：一种颜色混合白光的数量，与所加白光数量成反比
- 形成任何特殊颜色的红、绿、蓝的数量称为三色值，并分别表示为X、Y、Z，这样，一种颜色可由其三色值系数定义为：

$$x = \frac{X}{X + Y + Z}$$

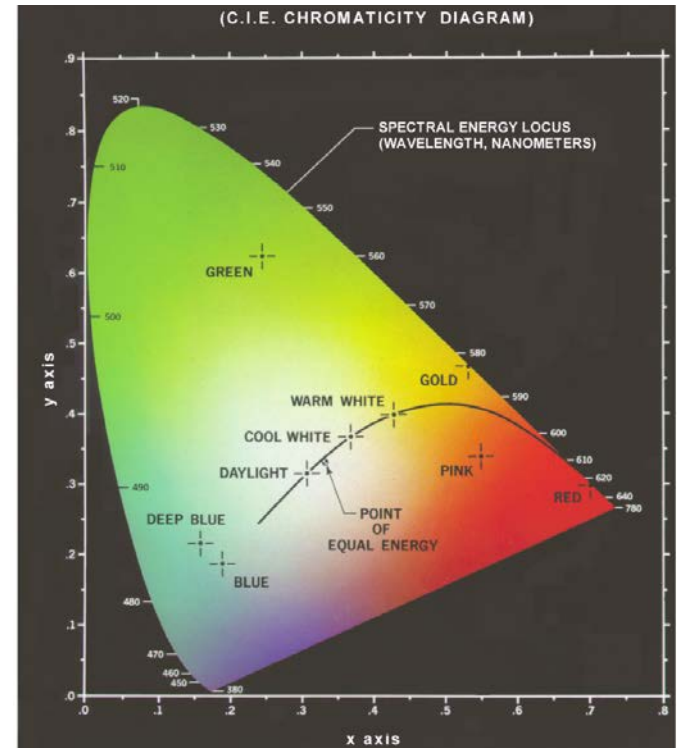
$$y = \frac{Y}{X + Y + Z}$$

$$z = \frac{Z}{X + Y + Z}$$

- CIE色度图

- 以x和y的函数表示颜色的组成
- 相应的z值可由下式得到

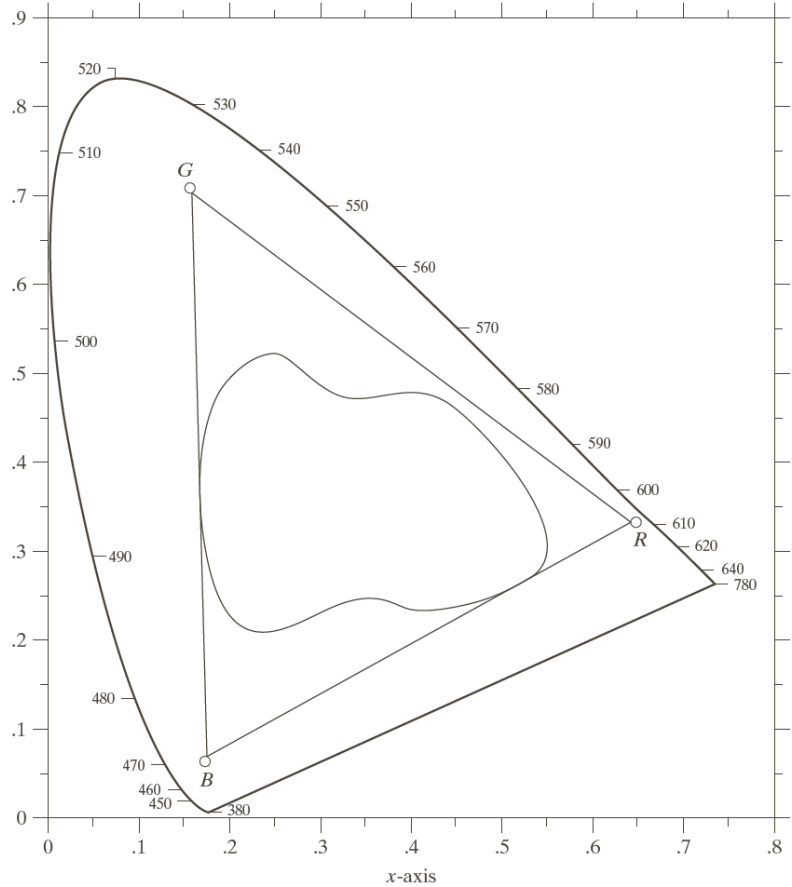
$$x + y + z = 1$$



# 6.1 彩色基础

## □ 彩色监视器的彩色域

- 右图为彩色监视器的典型彩色域和彩色打印设备的典型彩色域
- 三角形内的不规则区域是今天高质量彩色打印设备的代表性彩色域。
- 彩色打印设备的彩色域的边界是不规则的，因为彩色打印是相加色彩和相减色彩混合的组合，不易控制。







# 第6章 彩色图像处理

- 6.1 彩色基础
- 6.2 彩色模型**
- 6.3 伪彩色图像处理
- 6.4 真彩色图像处理
- 6.5 彩色变换
- 6.6 平滑和锐化
- 6.7 基于彩色的图像分割
- 6.8 彩色图像中的噪声
- 6.9 彩色图像压缩



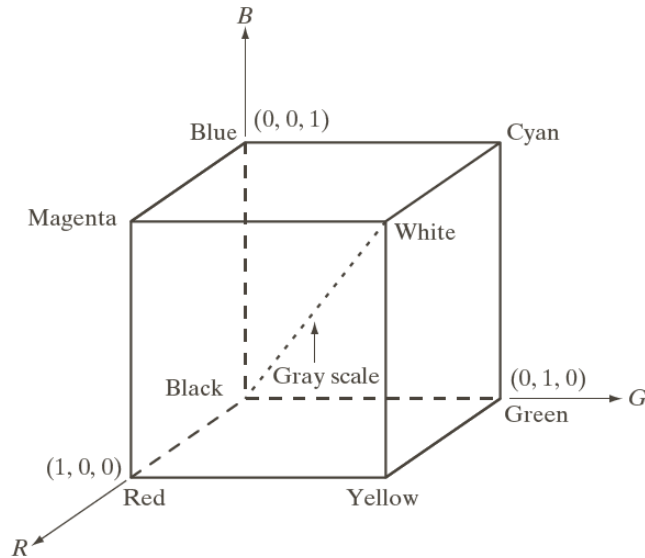
## 6.2 彩色模型

- 彩色模型的目的：在某些标准下用通常可以接受的方式方便地对彩色加以说明
- 在数字图像处理中，最通用的模型如下：
  - 面向硬件的RGB（红、绿、蓝）模型，用于彩色摄像机
  - CMY（青、粉红、黄）模型和CMYK（青、粉红、黄、黑）模型，用于彩色打印机
  - HIS（色调、饱和度、亮度）模型，更符合人描述和解释颜色的方式

# 6.2.1 RGB彩色模型

## □ RGB彩色模型

- 基于笛卡尔坐标系，三原色位于3个角上
- 在RGB空间中，表示每个像素的比特数称为像素深度。考虑一幅RGB图像，其中每幅红绿蓝图像都是8比特图像，所以每个RGB像素有24比特的深度



RGB彩色立方体

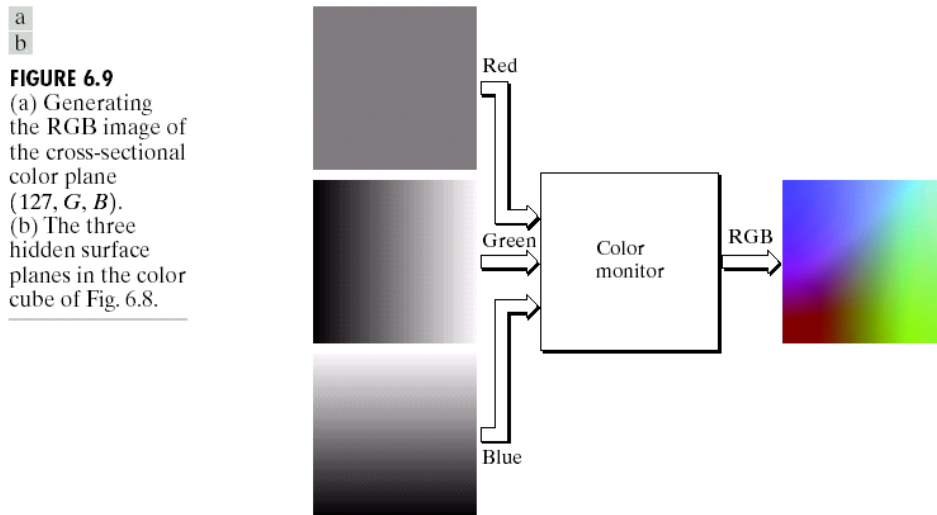


RGB 24比特彩色立方体

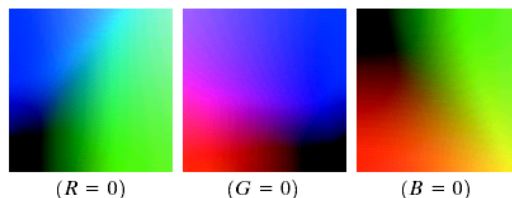
# 6.2.1 RGB彩色模型

## □ 生成RGB彩色立方体的隐藏面和剖面

- 彩色平面通过固定三种颜色中的一种并让其他两种颜色变化
- a图说明，简单把三幅独立的分量图像送入彩色监视器，可观察一幅横截面图像，b图以同样方式生成立方体的3个隐藏面



a  
b  
**FIGURE 6.9**  
(a) Generating the RGB image of the cross-sectional color plane ( $127, G, B$ ).  
(b) The three hidden surface planes in the color cube of Fig. 6.8.





# 6.2.1 RGB彩色模型

## □ 稳定色

- 假定256种颜色是最小颜色数，已知256颜色中的40种可被各种操作系统进行不同处理，剩下的216中颜色已成为稳定色
- 216种稳定色中的每种可由3个RGB值形成，但每个值只能是0,51,102,153,204或255，这些RGB三元组提供 $(6)^3 = 216$ 种

Number System	Color Equivalent					
Hex	00	33	66	99	CC	FF
Decimal	0	51	102	153	204	255

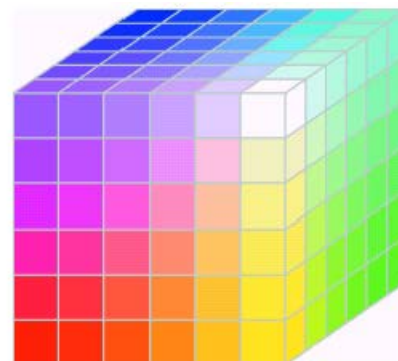
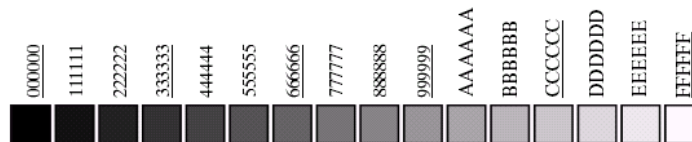
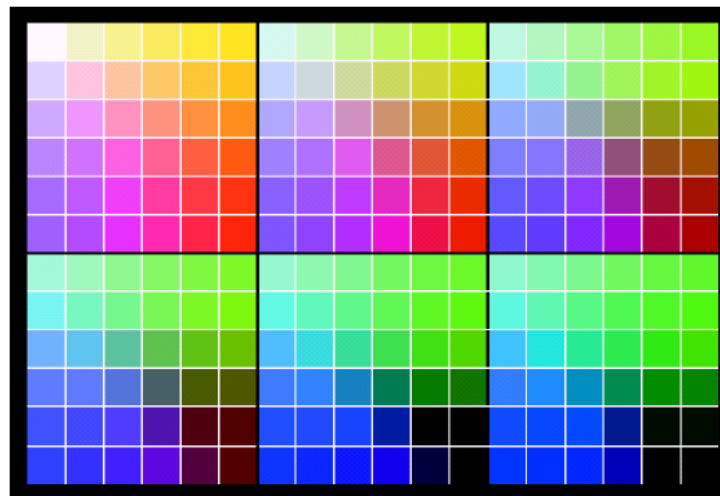
**TABLE 6.1**  
Valid values of each RGB component in a safe color.

- 因为要取3个数来形成RGB彩色，每种稳定色由上表中的3个两位十六进制数形成。例如，最纯净的红色是FF0000。使用更为熟悉的十进制表示可得到相同的结果，例如，以十进制表示的最亮红色为R=255(FF)和G=B=0。

# 6.2.1 RGB彩色模型

## □ 稳定色

- 右图显示了按RGB的值降序排列的216种稳定色
- 不是所有可能的8比特灰色都包含在216中稳定色中
- RGB稳定色立方体仅在表面上存在有效的颜色，整个表面被216种颜色覆盖



a  
b

**FIGURE 6.10**  
 (a) The 216 safe RGB colors.  
 (b) All the grays in the 256-color RGB system (grays that are part of the safe color group are shown underlined).

**FIGURE 6.11** The RGB safe-color cube.



## 6.2.2 CMY和CMYK彩色模型

### □ CMY彩色模型

- 大多数在纸上沉积彩色颜料的设备，如彩色打印机和复印机，要求输入CMY数据或在内部进行RGB到CMY的转换：

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

### □ CMYK模型

- 等量的颜料原色可以生成黑色，但为打印目的组合这些颜色产生的黑色是不纯的。
- 为了生成真正的黑色（在打印中起主要作用的颜色），加入了第4中颜色——黑色，提出了CMYK彩色模型



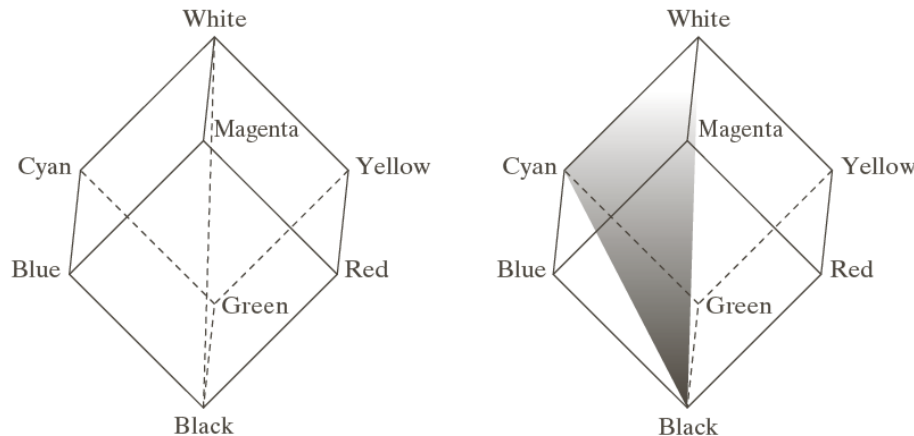
# 6.2.3 HSI彩色模型

## □ HSI彩色模型

- 即色调 (hue)、饱和度 (saturation)、密度 (intensity, 对应图像亮度)
- I分量与图像的彩色信息无关
- H和S分量与人感受颜色的方式紧密相连 (合成色度)

## □ HSI模型与RGB模型间的概念关系

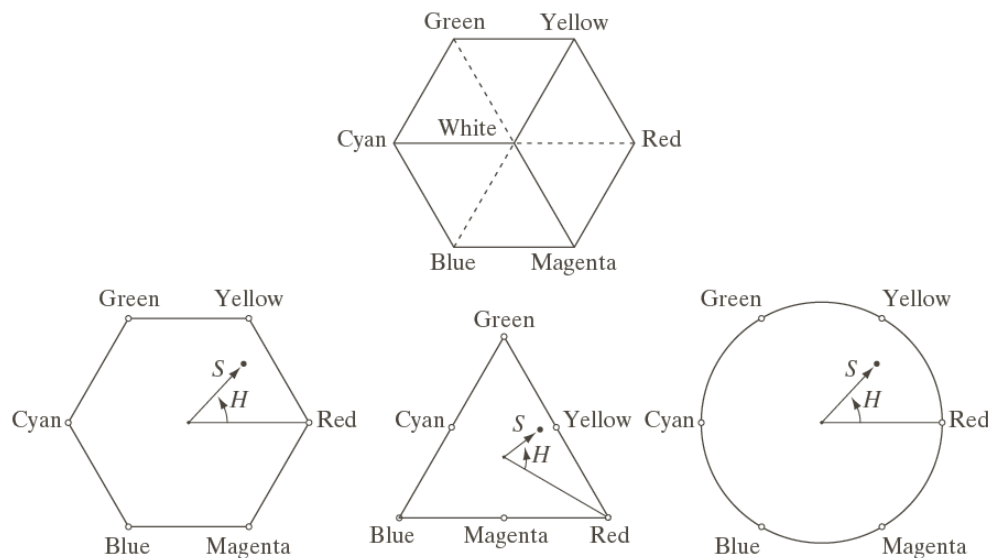
- 平面上点都有相同的色调, 强度和饱和度是不同的
- 形成HSI空间所要求的色调、饱和度和强度值可由RGB彩色立方体得到, 可将任何RGB点转换为相应的HSI彩色模型中的点



## 6.2.3 HSI彩色模型

### □ HSI彩色模型的色调和饱和度

- HSI空间由一个垂直强度轴和位于该轴垂直的平面内的彩色点的轨迹表示。
- HSI平面什么形状（六边形、三角形或圆形）无关紧要，因为这些形状中的任何一个都可以通过几何变换为其他两种形状

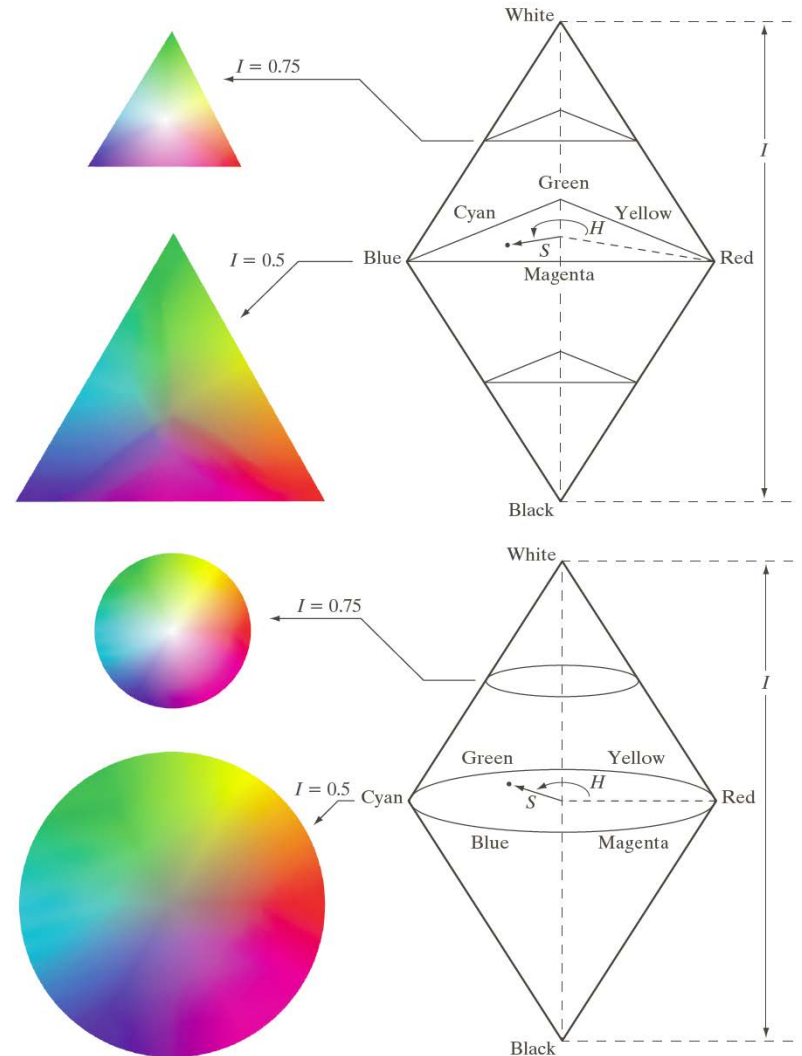


HSI彩色模型中的色调和饱和度：黑点是一个任意彩色点。与红轴的夹角给出了色调，向量的长度是饱和度。这些平面中的所有彩色的强度由垂直强度轴上的平面的位置给出

# 6.2.3 HSI彩色模型

## □ HSI彩色模型

- 右图显示了基于彩色三角形和圆形的HSI模型
- 三角形和圆形平面垂直于强度轴





## 6.2.3 HSI彩色模型

□ 从RGB转换到HIS

$$H = \begin{cases} \theta, & B \leq G \\ 360 - \theta, & B > G \end{cases}$$

$$H = \arccos \left\{ \frac{[(R - G) + (R - B)] / 2}{[(R - G)^2 + (R - B)(G - B)]^{1/2}} \right\}$$

$$I = \frac{1}{3}(R + G + B) \quad S = 1 - \frac{3}{(R + G + B)}[\min(R, G, B)]$$



## 6.2.3 HSI彩色模型

### □ 从HSI转换到RGB

- 当H在[0, 120]之间

$$B = I(1 - S) \quad R = I \left[ 1 + \frac{S \cos H}{\cos(60^\circ - H)} \right] \quad G = 3I - (B + R)$$

- 当H在[120, 240]之间

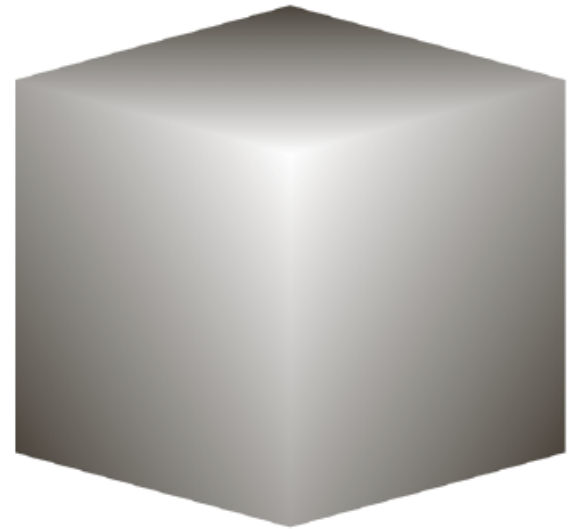
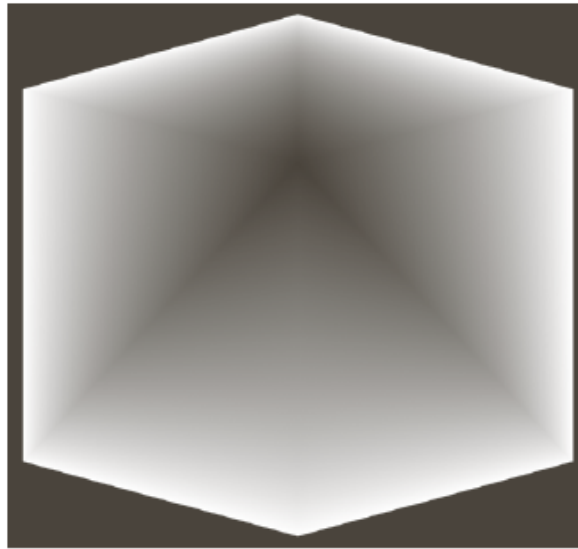
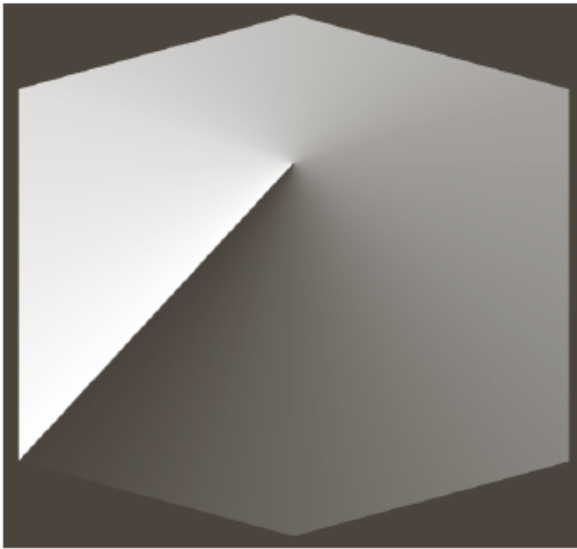
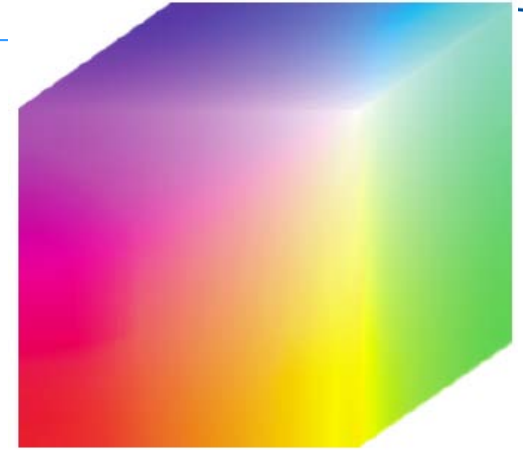
$$R = I(1 - S) \quad G = I \left[ 1 + \frac{S \cos(H - 120^\circ)}{\cos(180^\circ - H)} \right] \quad B = 3I - (R + G)$$

- 当H在[240, 360]之间

$$G = I(1 - S) \quad B = I \left[ 1 + \frac{S \cos(H - 240^\circ)}{\cos(300^\circ - H)} \right] \quad R = 3I - (G + B)$$

## 6.2.3 HSI彩色模型

- 对应于RGB彩色立方体图像的HSI值
  - RGB立方体的色调、饱和度和强度图像。



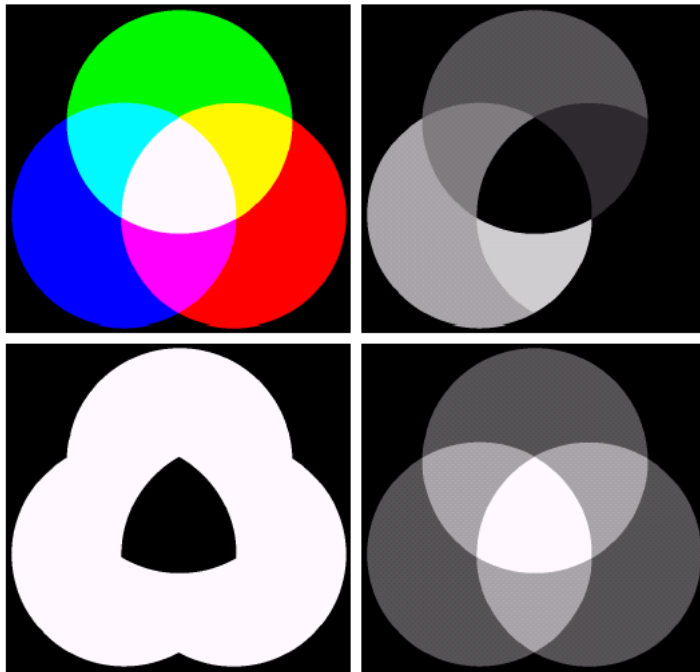
a b c

**FIGURE 6.15** HSI components of the image in Fig. 6.8. (a) Hue, (b) saturation, and (c) intensity images.

# 6.2.3 HSI彩色模型

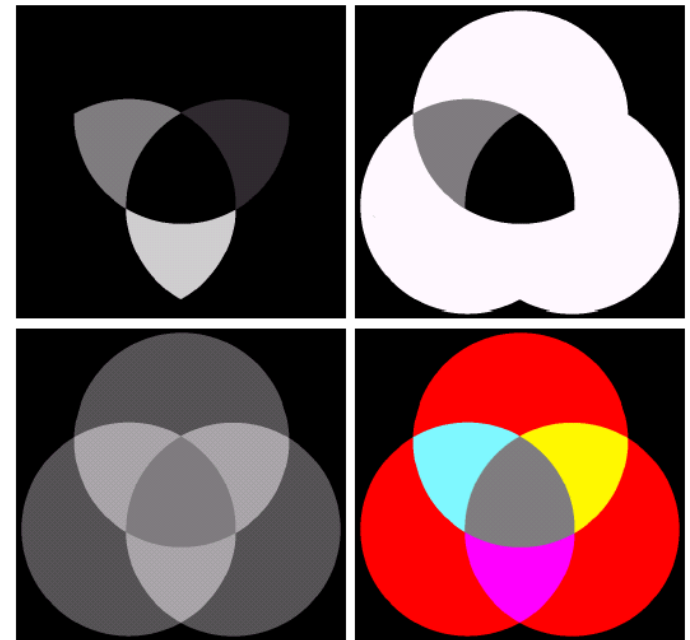
## □ 处理HSI分量图像

- 要在RGB图像中更改任何区域的个别颜色，可以改变色调图像相应区域的值，将新的H图像和没有变化的S图像与I图像一起变换回RGB图像。改变饱和度和强度类似，可同时进行



a b  
c d

**FIGURE 6.16** (a) RGB image and the components of its corresponding HSI image: (b) hue, (c) saturation, and (d) intensity.



a b  
c d

**FIGURE 6.17** (a)–(c) Modified HSI component images. (d) Resulting RGB image. (See Fig. 6.16 for the original HSI images.)



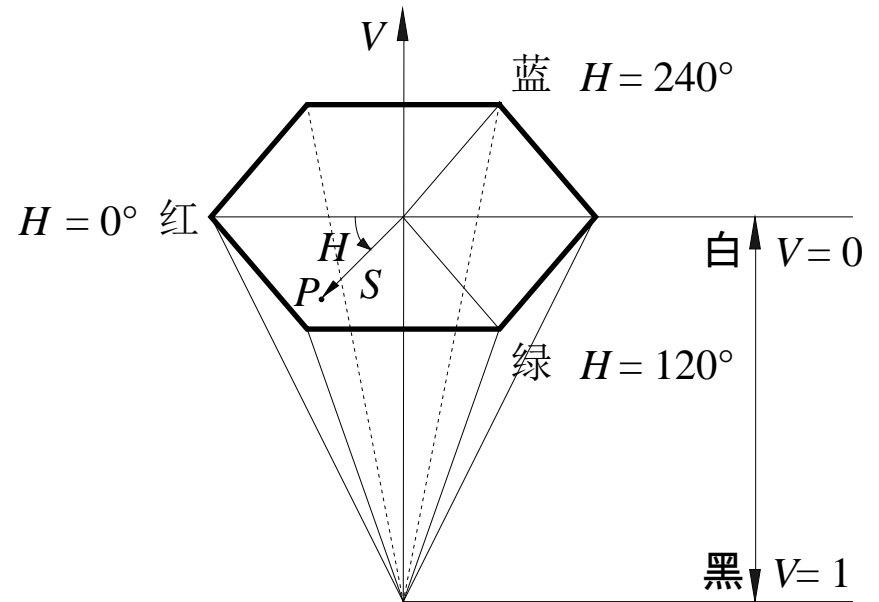
# 6.2 彩色模型

## □ HSV模型

- 一般用六棱锥来表示
- H同HSI模型

$$S = \frac{\max(R, G, B) - \min(R, G, B)}{\max(R + G + B)}$$

$$V = \frac{\max(R, G, B)}{255}$$





# 第6章 彩色图像处理

- 6.1 彩色基础
- 6.2 彩色模型
- 6.3 伪彩色图像处理
- 6.4 真彩色图像处理
- 6.5 彩色变换
- 6.6 平滑和锐化
- 6.7 基于彩色的图像分割
- 6.8 彩色图像中的噪声
- 6.9 彩色图像压缩



# 6.3 伪彩色图像处理

## □ 特点

- 伪彩色（也称假彩色）图像处理是基于一种指定的规则对灰度值赋以颜色的处理
- 不同灰度区域 → 赋予不同颜色

## □ 原理

- 人眼对颜色比对灰度有较大的分辨能力
- 对灰度：几十
- 对彩色：几千

## □ 分类

- 灰度分层
- 灰度到彩色的变换

# 6.3.1 灰度分层

## □ 灰度分层（也称密度分层）

- 一幅图像被描述为三维函数，如图
- 用一些平行于该图像的坐标平面的平面去切割图像函数，从而把灰度函数分层一些灰度区间
- 每个灰度区间赋予不同的颜色

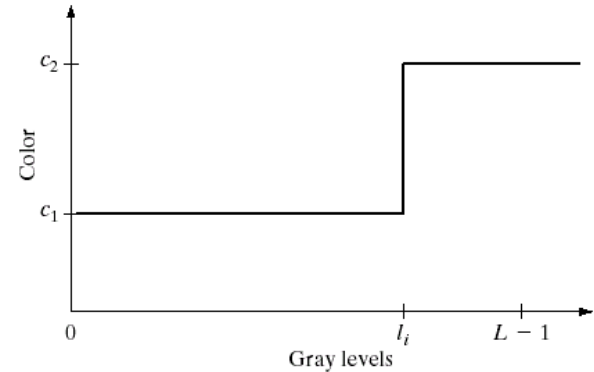
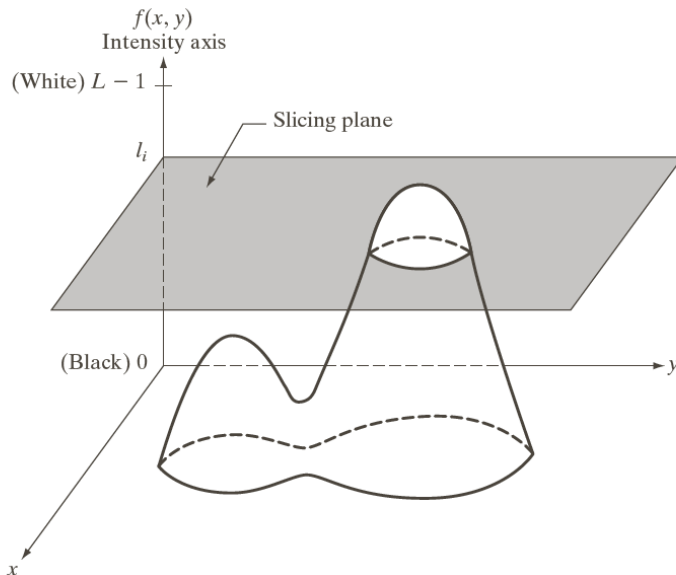


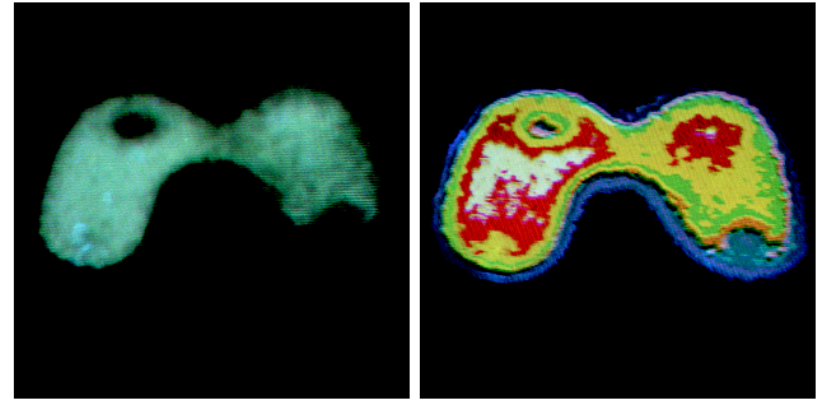
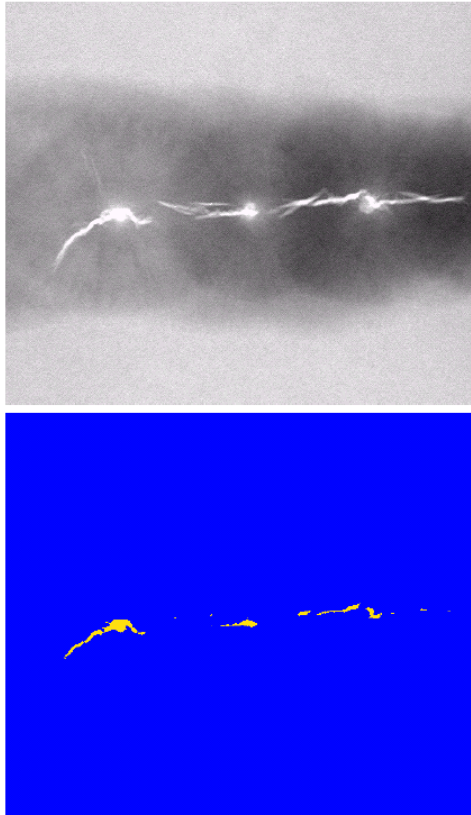
FIGURE 6.19 An alternative representation of the intensity-slicing technique.

# 6.3.1 灰度分层

## □ 例

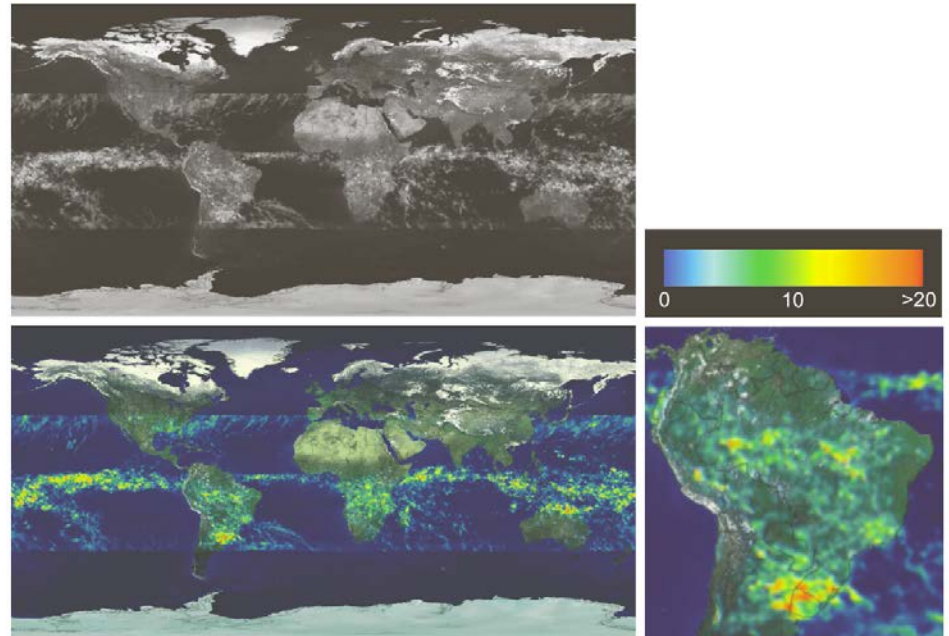
a  
b

**FIGURE 6.21**  
(a) Monochrome X-ray image of a weld. (b) Result of color coding.  
(Original image courtesy of X-TEK Systems, Ltd.)



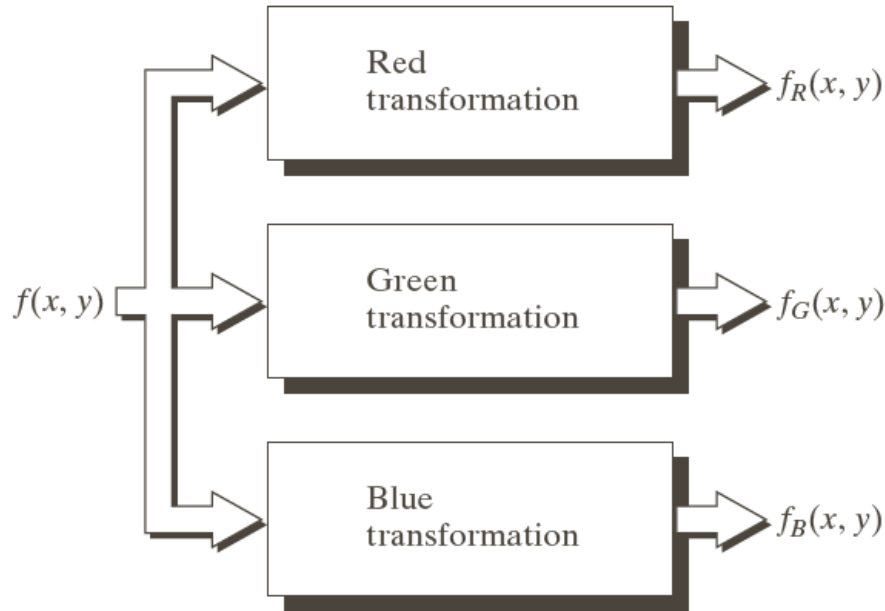
a b

**FIGURE 6.20** (a) Monochrome image of the Picker Thyroid Phantom. (b) Result of density slicing into eight colors. (Courtesy of Dr. J. L. Blankenship, Instrumentation and Controls Division, Oak Ridge National Laboratory.)



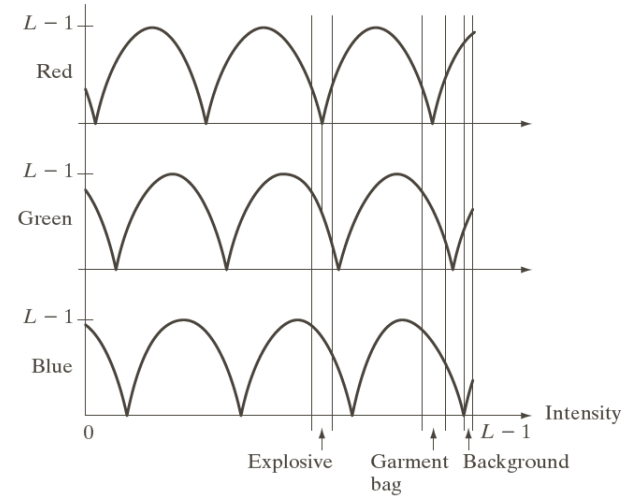
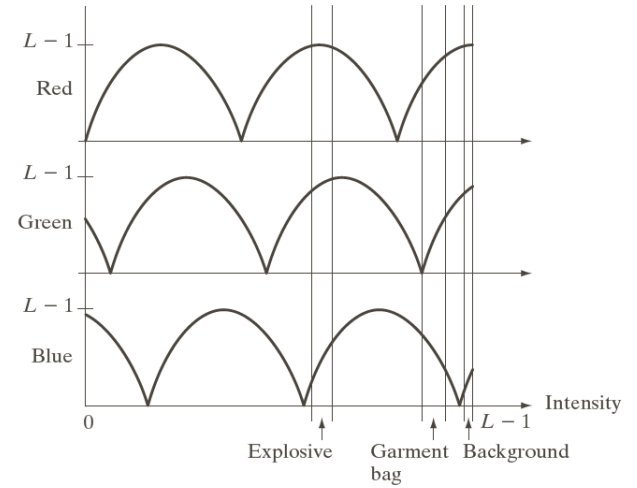
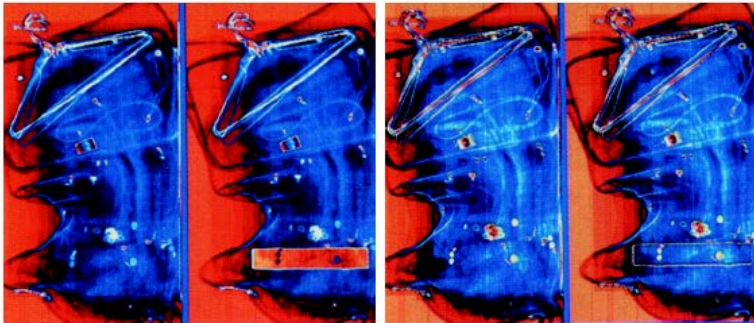
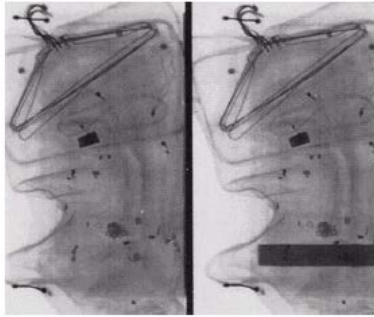
## 6.3.2 灰度到彩色的变换

- 从灰度到彩色的变换（映射）
  - 对任何输入像素的灰度执行3个独立的变换
  - 将3个变换结果分别送入彩色监视器的红、绿、蓝通道
  - 产生一幅合成图像，图像的彩色内容有变换函数的特性调制
  - 该方法是灰度分层方法的推广



# 6.3.2 灰度到彩色的变换

## □ 例



a  
b c

**FIGURE 6.24** Pseudocolor enhancement by using the gray-level to color transformations in Fig. 6.25. (Original image courtesy of Dr. Mike Hurwitz, Westinghouse.)

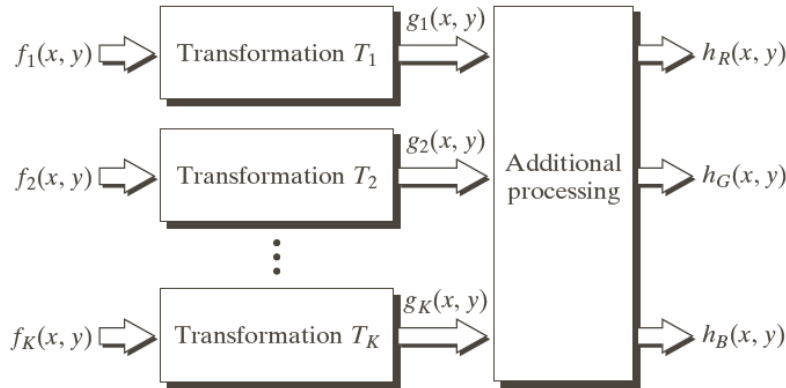




# 6.3.2 灰度到彩色的变换

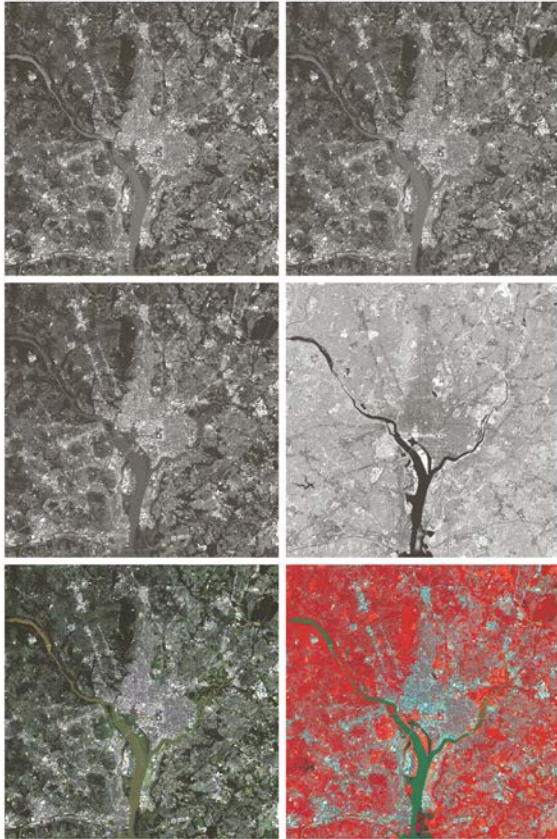
## □ 多光谱图像处理

- 将多幅单色图像组合为一幅彩色图像合成图像是很重要的
- 不同的传感器在不同的谱段产生独立的单色图像
- 附加处理可以是彩色平衡、混合图像，以及以产生图像的传感器响应特性的知识为基础来选择三幅用于显示的图像等技术



# 6.3.2 灰度到彩色的变换

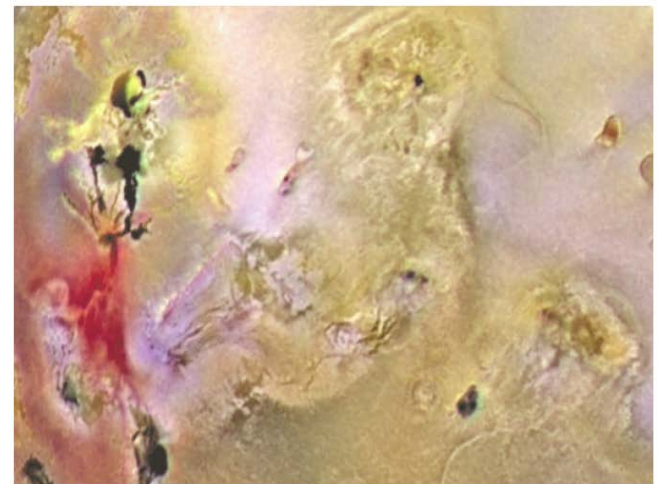
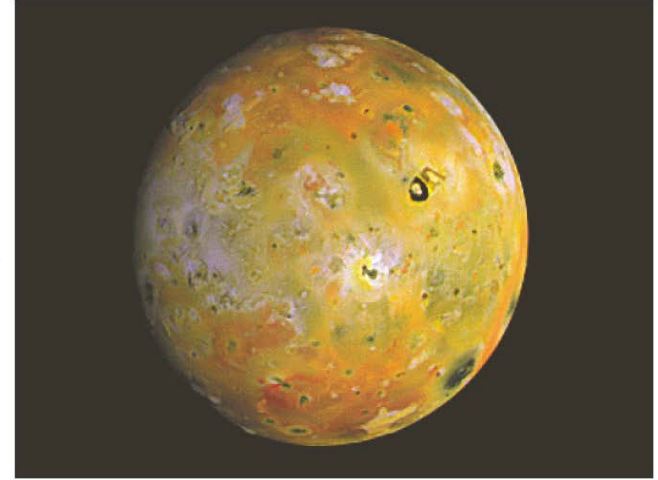
## □ 多光谱图像的彩色编码



**FIGURE 6.27** (a)–(d) Images in bands 1–4 in Fig. 1.10 (see Table 1.1). (e) Color composite image obtained by treating (a), (b), and (c) as the red, green, blue components of an RGB image. (f) Image obtained in the same manner, but using in the red channel the near-infrared image in (d). (Original multispectral images courtesy of NASA.)

a  
b

**FIGURE 6.28**  
(a) Pseudocolor rendition of Jupiter Moon Io.  
(b) A close-up.  
(Courtesy of NASA.)



a b  
c d  
e f



# 第6章 彩色图像处理

- 6.1 彩色基础
- 6.2 彩色模型
- 6.3 伪彩色图像处理
- 6.4 真彩色图像处理**
- 6.5 彩色变换
- 6.6 平滑和锐化
- 6.7 基于彩色的图像分割
- 6.8 彩色图像中的噪声
- 6.9 彩色图像压缩



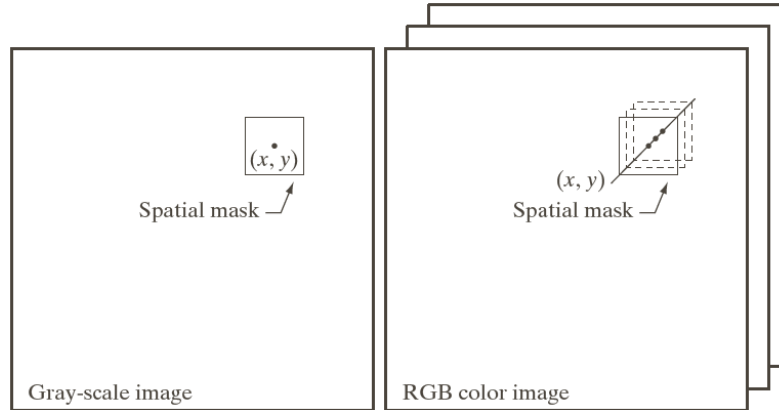
## 6.4 真彩色图像处理

- 真（全）彩色图像处理方法分为两大类
  - 将一幅彩色图象看作三幅分量图象的组合体，先分别单独处理，再将结果合成
  - 将一幅彩色图象中的每个象素看作具有三个属性值，即属性现在为一个矢量，利用对矢量的表达方法进行处理

$$\begin{aligned} \sum_{(x,y) \in N} \mathbf{C}(x,y) &= \sum_{(x,y) \in N} [R(x,y) + G(x,y) + B(x,y)] \\ &= \left\{ \sum_{(x,y) \in N} R(x,y) + \sum_{(x,y) \in N} G(x,y) + \sum_{(x,y) \in N} B(x,y) \right\} \end{aligned}$$

# 6.4 真彩色图像处理

- 每种彩色分量的处理等同于基于向量的处理，两个条件
  - 第一，处理必须对向量和标量都可用；
  - 第二，对向量的每个分量的操作对于其他分量必须是独立的



a b

**FIGURE 6.29**  
Spatial masks for  
gray-scale and  
RGB color  
images.

- 图显示了灰度图像和全彩色图像的邻域空间处理，邻域平均
- (a)中平均是把邻域内的所有像素的灰度相加除以像素总数
- (b)中是把邻域内所有向量相加，并用邻域向量总数去除每个分量
- 这两个条件满足，两种处理结果是相同的



# 第6章 彩色图像处理

- 6.1 彩色基础
- 6.2 彩色模型
- 6.3 伪彩色图像处理
- 6.4 真彩色图像处理
- 6.5 彩色变换**
- 6.6 平滑和锐化
- 6.7 基于彩色的图像分割
- 6.8 彩色图像中的噪声
- 6.9 彩色图像压缩



# 6.5 彩色变换

## □ 变换公式

- 如灰度变换技术那样，用下式表达的彩色变换作为模型

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

- 其中  $f(x, y)$  是输入图像， $g(x, y)$  是变换后处理过的彩色输出图像， $T$  是在  $(x, y)$  的空间领域上对  $f$  操作的一个算子
- 类似于灰度变换的方法，本节只关注如下的彩色变换

$$s_i = T_i(r_1, r_2, \dots, r_n), \quad i = 1, 2, \dots, n$$

- 为标记简单起见， $r_i$  和  $s_i$  是  $f(x, y)$  和  $g(x, y)$  在任何点处彩色分量的变量， $n$  是彩色分量数， $\{T_1, T_2, \dots, T_n\}$  是对  $r_i$  操作产生  $s_i$  的一组变换或彩色映射函数。注意， $n$  个变换合并可执行单一变换函数  $T$ 。用于描述  $f$  和  $g$  的像素的所选彩色空间决定  $n$  的值。例如，如果选择 RGB 彩色空间，则  $n=3$ ，且  $r_1, r_2, r_3$  分别表示输入图像的红、绿、蓝分量。



# 6.5 彩色变换

## □ 全彩色图像及其各种彩色空间分量



Full color



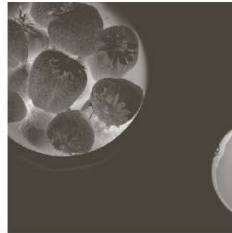
Cyan



Magenta



Yellow



Black



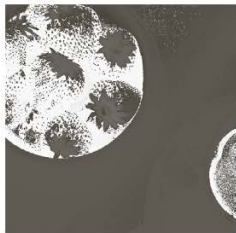
Red



Green



Blue



Hue



Saturation



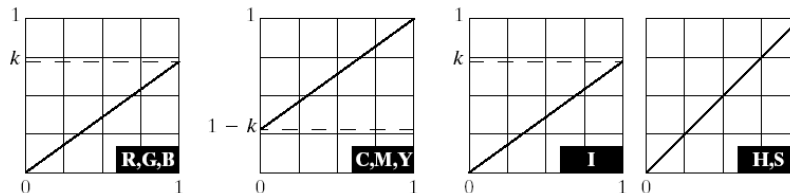
Intensity

# 6.5 彩色变换

- 理论上，任何变换都可在任何彩色模型中执行。但实际上，某些操作对特定的模型比较适用
- 假设，要降低彩色图像的亮度。在RGB、CMY和HSI模型中的变换函数是不同的

a b  
c d e

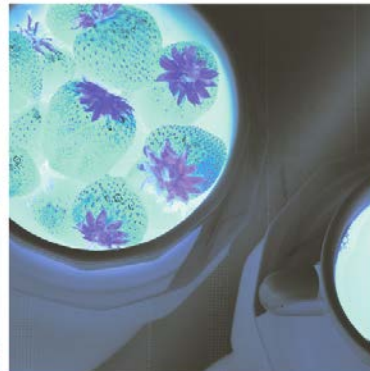
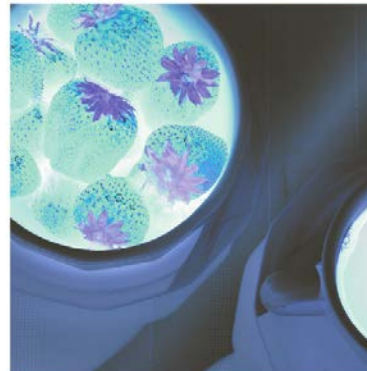
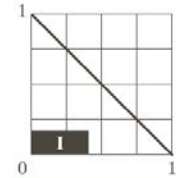
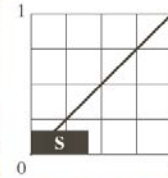
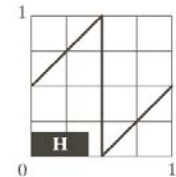
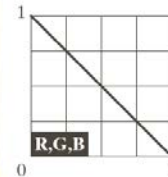
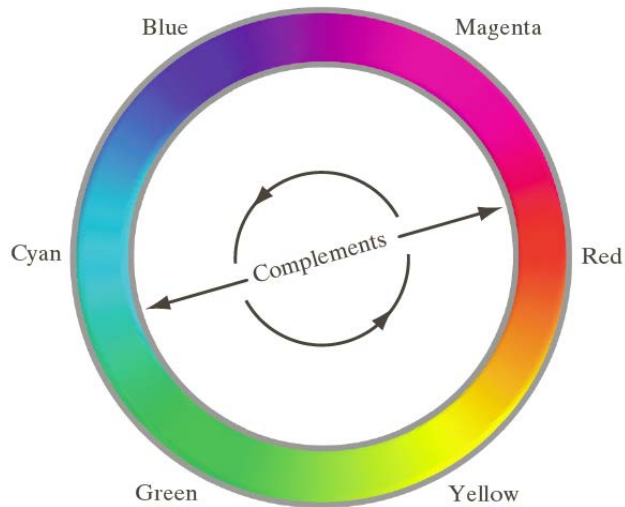
**FIGURE 6.31**  
Adjusting the intensity of an image using color transformations. (a) Original image. (b) Result of decreasing its intensity by 30% (i.e., letting  $k = 0.7$ ). (c)–(e) The required RGB, CMY, and HSI transformation functions. (Original image courtesy of MedData Interactive.)



# 6.5 彩色变换

## □ 补色

- 彩色环上与色调直接相对的另一端被称为补色
- 补色对于增强嵌在彩色图像暗区的细节很有用





# 6.5 彩色变换

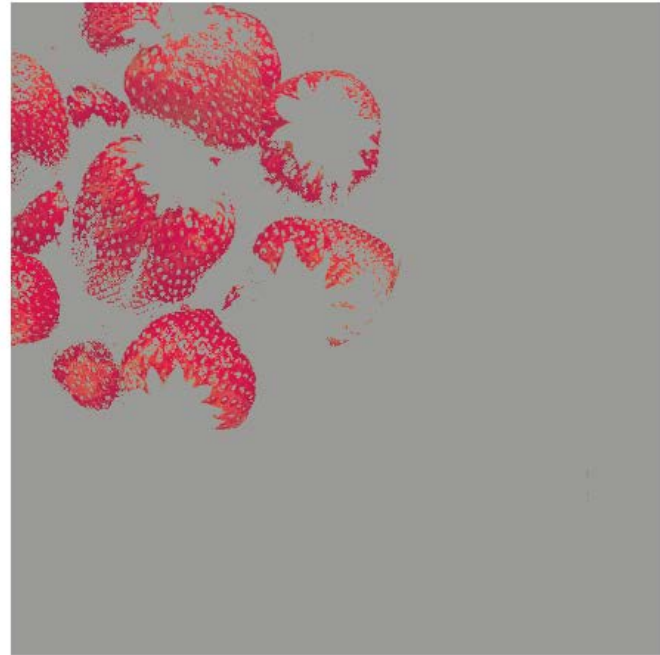
## □ 彩色分层

- 突出图像中某个特定彩色区域对从周围分离出目标物体很有用
- 变换函数很复杂，其中较为简单的方法之一是，把某些感兴趣区域之外的彩色映射为不突出的无确定性质的颜色

$$s_i = \begin{cases} 0.5 & \text{if } \left[ |r_j - a_j| > \frac{W}{2} \right]_{\text{any } 1 \leq j \leq n} \\ r_i & \text{otherwise} \end{cases} \quad i = 1, 2, \dots, n$$

$$s_i = \begin{cases} 0.5 & \text{if } \sum_{j=1}^n (r_j - a_j)^2 > R_0^2 \\ r_i & \text{otherwise} \end{cases} \quad i = 1, 2, \dots, n$$

# 6.5 彩色变换





# 6.5 彩色变换

## □ CIE L\*a\*b\* 模型 (CIELAB)

L\*表示亮度， a\*表示红减绿， b\*表示绿减蓝

$$L^* = 116 \cdot h\left(\frac{Y}{Y_w}\right) - 16 \quad (6.5-9)$$

$$a^* = 500 \left[ h\left(\frac{X}{X_w}\right) - h\left(\frac{Y}{Y_w}\right) \right] \quad (6.5-10)$$

$$b^* = 200 \left[ h\left(\frac{Y}{Y_w}\right) - h\left(\frac{Z}{Z_w}\right) \right] \quad (6.5-11)$$

$$h(q) = \begin{cases} \sqrt[3]{q} & q > 0.008856 \\ 7.787q + 16/116 & q \leq 0.008856 \end{cases} \quad (6.5-12)$$

$X_w, Y_w, Z_w$ 是参考白色三激励值，即CIE色度图中由 $x=0.3127$ 和 $y=0.3290$ 定义



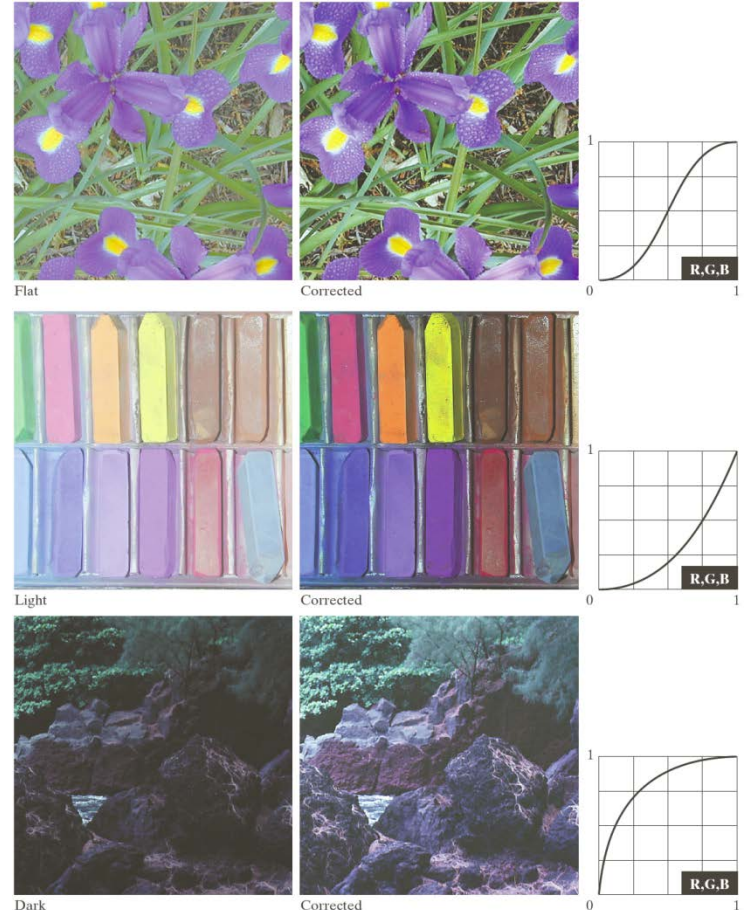
# 6.5 彩色变换

## □ 色调和彩色校正

- 照片增强和颜色再现

## □ 色调变换

- 色调变换通常需要交互选择，试验性调整图像亮度和对比度，在合适的范围提供更多的细节，彩色本身不改变。在RGB和CMY空间中，这意味着使用相同的变换函数映射所有的3个彩色分量；在HSI彩色空间中，则改进了亮度分量
- 图为校正3种常见色调不平衡的变换即平淡、较亮、较暗





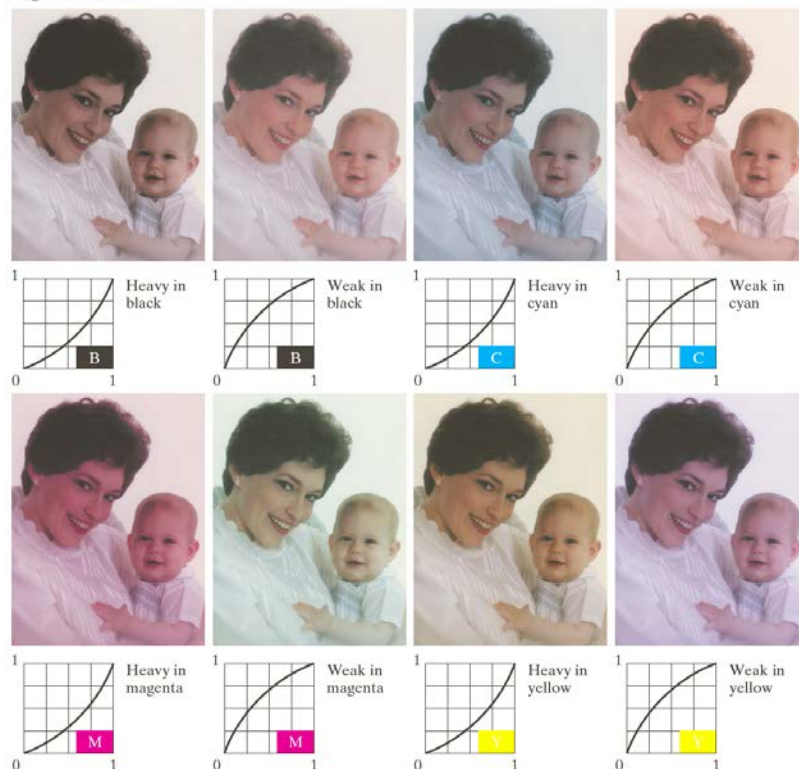
# 6.5 彩色变换

## □ 彩色平衡

- 在调整一幅图像的彩色分量时，要意识到每个操作都会影响到图像的全部彩色平衡
- 彩色环可用于预测一个彩色分量如何影响其他彩色分量
- 基于彩色环，任何颜色的比例都可通过减小图像中相对色（或补色）数量来增大
- 右图显示了用于校正简单CMYK输出不平衡的变换



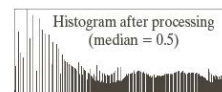
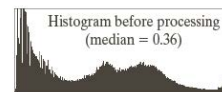
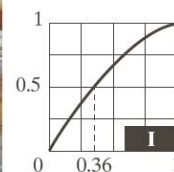
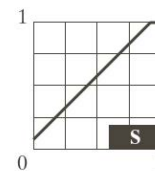
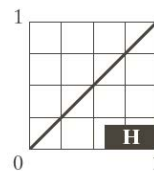
Original/Corrected



# 6.5 彩色变换

## □ 直方图处理

- 直方图均衡会自动确定一种变换，产生具有均匀灰度值的直方图的图像
- 单独对彩色图像的分量进行直方图均衡是不明智的
- 应该均匀展开这种彩色灰度，而保持彩色本身（色调）不变
- 图为HSI彩色空间的直方图均衡





# 第6章 彩色图像处理

- 6.1 彩色基础
- 6.2 彩色模型
- 6.3 伪彩色图像处理
- 6.4 真彩色图像处理
- 6.5 彩色变换
- 6.6 平滑和锐化**
- 6.7 基于彩色的图像分割
- 6.8 彩色图像中的噪声
- 6.9 彩色图像压缩

# 6.6 平滑和锐化

## □ 彩色图像平滑

- 灰度图像平滑是一种空间滤波操作，每个像素被由模板定义的领域中的像素的平均值代替，推广到彩色图像处理上
- 邻域平均平滑可以在每个彩色平面的基础上执行，其结果与使用RGB彩色向量执行平均是相同的
- 下图为一副图像的RGB分量图像





# 6.6 平滑和锐化

- 上图为该图的HSI分量图像，下图为平滑这2种彩色模型的区别



用 $5 \times 5$ 模板平滑图像：处理每个RGB分量结果（左）；处理HSI（中）；差别（右）

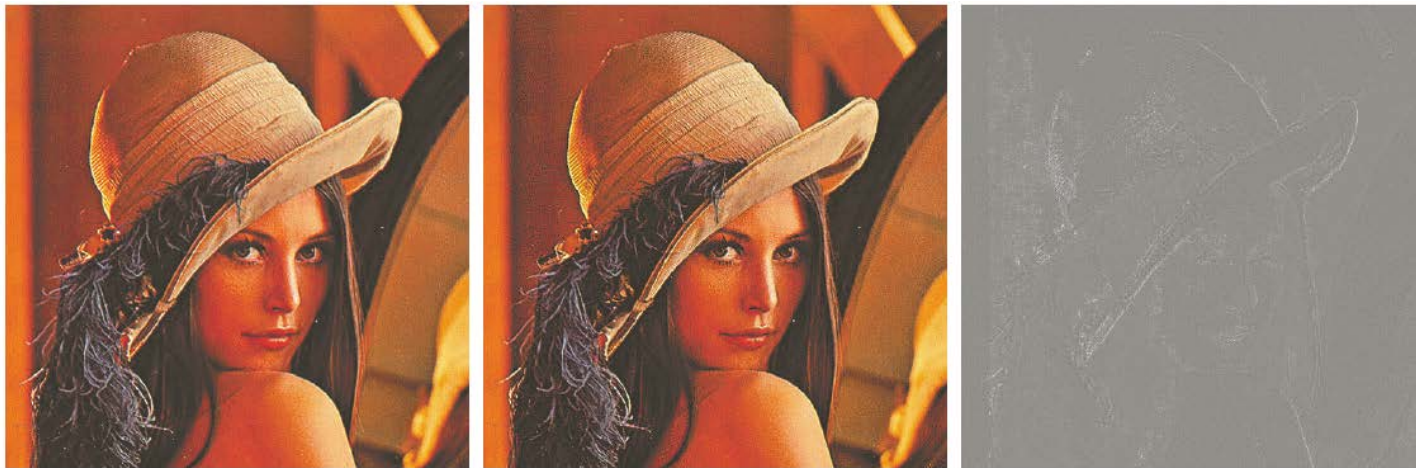
# 6.6 平滑和锐化

## □ 彩色图像锐化

- 采用拉普拉斯变换方法的图像锐化处理
- 可通过分别计算每幅分量图像的拉普拉斯变换来计算彩色图像的拉普拉斯变换

$$g(x, y) = f(x, y) + c[\nabla^2 f(x, y)]$$

0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1



拉普拉斯锐化图像：处理每个RGB分量结果（左）；处理HSI（中）；差别（右）



# 第6章 彩色图像处理

- 6.1 彩色基础
- 6.2 彩色模型
- 6.3 伪彩色图像处理
- 6.4 真彩色图像处理
- 6.5 彩色变换
- 6.6 平滑和锐化
- 6.7 基于彩色的图像分割**
- 6.8 彩色图像中的噪声
- 6.9 彩色图像压缩





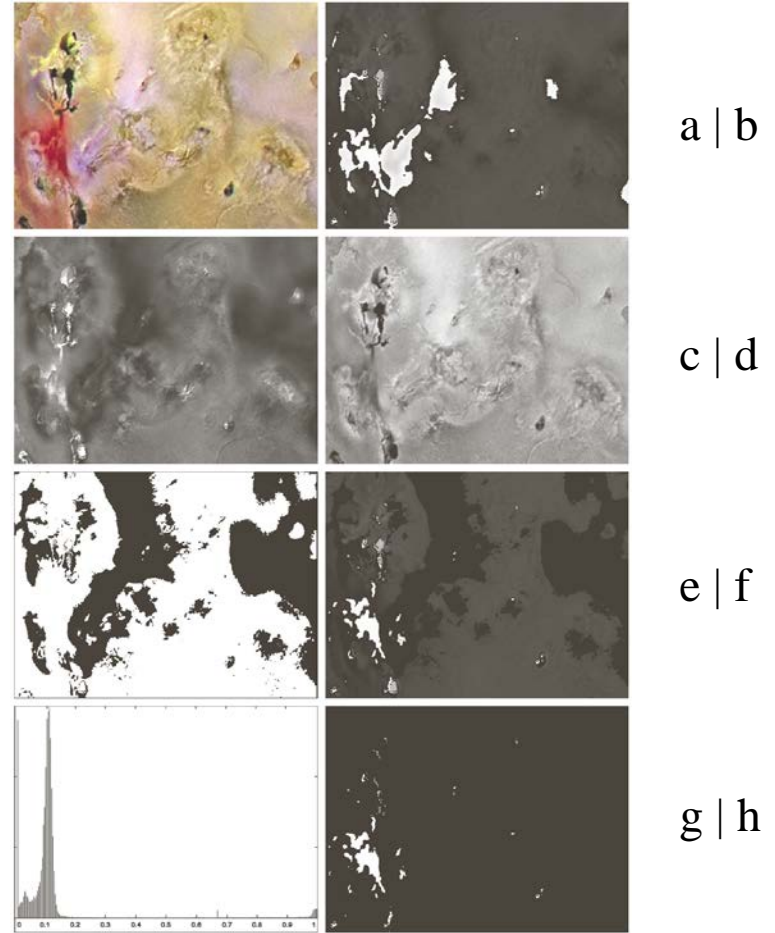
# 6.7 基于彩色的图像分割

- 分割是把一幅图像分成多个区域的处理，虽然分割是后面几章讨论的问题，但出于连续性原因，这里简单地介绍一下彩色分割
  - HSI彩色空间的分割
  - RGB向量空间中的分割
  - 彩色边缘检测

# 6.7 基于彩色的图像分割

## □ HSI彩色空间的分割

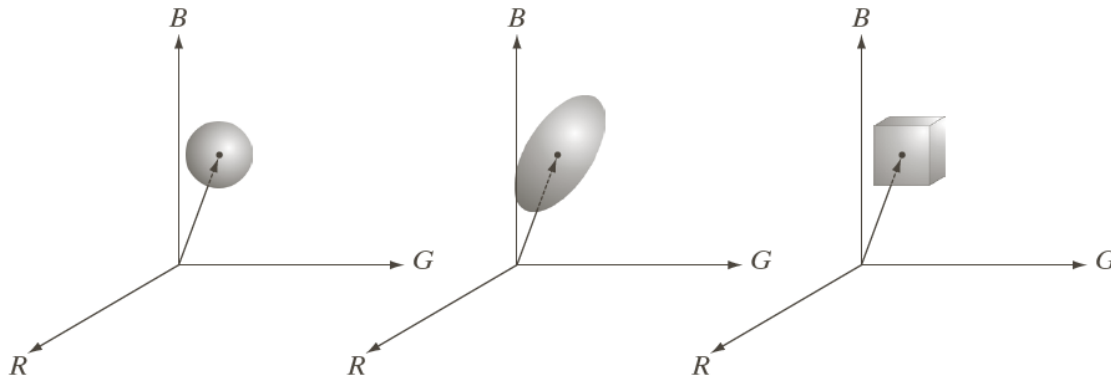
- (a) 原图像，为分割微红色区
- (b) 色调图像
- (c) 饱和度图像
- (d) 亮度图像
- (e) 二值饱和度模板（黑=0）
- (f) 图b和图e相乘后的结果
- (g) 图f的直方图
- (h) 图a中红色分量的分割



# 6.7 基于彩色的图像分割

## □ RGB向量空间中的分割

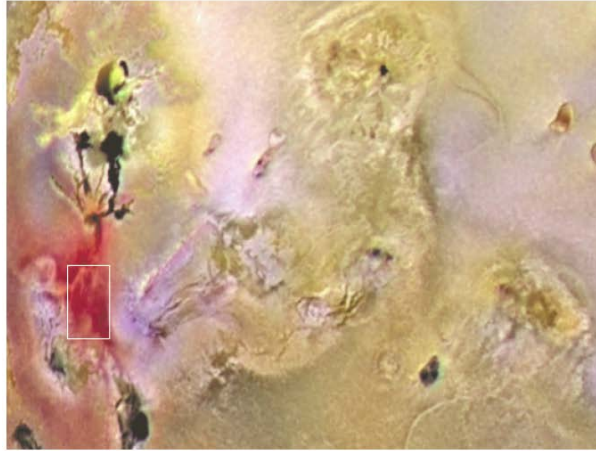
- 给定感兴趣的有代表性彩色的样点集，得到平均彩色用 $a$ 表示
- 相似性度量，比较空间任一点 $z$ 与 $a$ 的距离
  - ✓ 欧式距离（轨迹是球体）  $D(z, a) = \|z - a\|$
  - ✓ 马氏距离（轨迹是椭球体）  $D(z, a) = [(z - a)^T C^{-1} (z - a)]^{\frac{1}{2}}$
  - ✓ 边界盒（避免开方运算）
- 通过确定它是否在球体或盒子内部或外部来进行分割



# 6.7 基于彩色的图像分割

## □ RGB向量空间中的分割

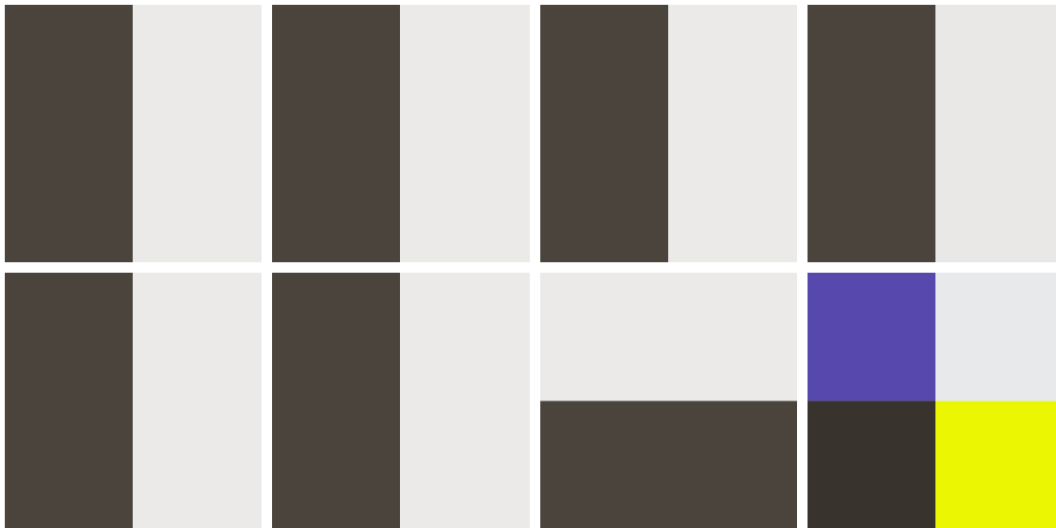
### ■ 例



# 6.7 基于彩色的图像分割

## □ 彩色边缘检测

- 以单一图像为基础计算边缘还是直接在彩色空间计算边缘
- 例下图，如果计算每个分量图像的梯度，并将结果相加形成两幅相应的RGB梯度图像，则中心点处的梯度值在两种情况下都将相同。直观看，我们希望图d中图像那一点梯度更强，因为R, G, B图像的边缘在该图像中的方向相同，而对于图h的图像只有两个边缘具有相同的方向。
- 所以，处理3个单独的平面所形成的合成梯度图像可导致错误



(a)~(c)R,G,B分量图像；(d)产生的RGB彩色图像；

(e)~(g)R,G,B分量图像；(h)产生的RGB彩色图像；



# 6.7 基于彩色的图像分割

## □ 彩色边缘检测 (Di Zenzo[1986])

- 对于标量函数，梯度是坐标点指向f的最大变化率方向的向量
- 令 $r$ ,  $g$ 和 $b$ 是沿RGB彩色空间的R,G,B轴的单位向量，并定义

$$\vec{u} = \frac{\partial R}{\partial x} \vec{r} + \frac{\partial G}{\partial x} \vec{g} + \frac{\partial B}{\partial x} \vec{b} \quad \vec{v} = \frac{\partial R}{\partial y} \vec{r} + \frac{\partial G}{\partial y} \vec{g} + \frac{\partial B}{\partial y} \vec{b}$$

- 令  $g_{xx}, g_{yy}, g_{xy}$  表示这些向量的点积，如下所示

$$g_{xx} = \vec{u} \cdot \vec{u} = \left| \frac{\partial R}{\partial x} \right|^2 + \left| \frac{\partial G}{\partial x} \right|^2 + \left| \frac{\partial B}{\partial x} \right|^2 \quad g_{yy} = \vec{v} \cdot \vec{v} = \left| \frac{\partial R}{\partial y} \right|^2 + \left| \frac{\partial G}{\partial y} \right|^2 + \left| \frac{\partial B}{\partial y} \right|^2 \quad g_{xy} = \vec{u} \cdot \vec{v} = \frac{\partial R}{\partial x} \frac{\partial R}{\partial y} + \frac{\partial G}{\partial x} \frac{\partial G}{\partial y} + \frac{\partial B}{\partial x} \frac{\partial B}{\partial y}$$

- 可以证明 $c(x,y)$ 的最大变化率方向可以由以下角度给出

$$\theta(x, y) = \frac{1}{2} \arctan \left[ \frac{2g_{xy}}{g_{xx} - g_{yy}} \right]$$

- 且角度方向上点 $(x,y)$ 处的变化率的值由下式给出

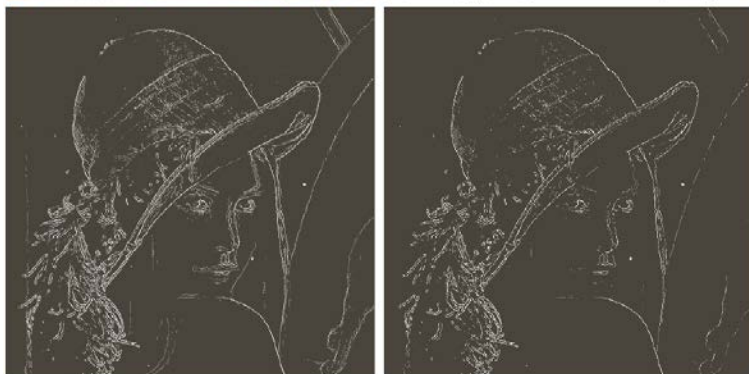
$$F_{\theta}(x, y) = \left\{ \frac{1}{2} [(g_{xx} + g_{yy}) + (g_{xx} - g_{yy}) \cos 2\theta(x, y) + 2g_{xy} \sin 2\theta(x, y)] \right\}^{\frac{1}{2}}$$

# 6.7 基于彩色的图像分割

## □ 例



(a)RGB图像；(b)在RGB彩色空间中计算的梯度；(c)在每幅分量图像基础上计算梯度，然后相加；(d)图b和图c的差



彩色图像的分量梯度图像：  
(a)红分量；(b)绿分量；(c)蓝分量。这3幅图像相加并标定后产生了上图c中的图像







# 第6章 彩色图像处理

- 6.1 彩色基础
- 6.2 彩色模型
- 6.3 伪彩色图像处理
- 6.4 真彩色图像处理
- 6.5 彩色变换
- 6.6 平滑和锐化
- 6.7 基于彩色的图像分割
- 6.8 彩色图像中的噪声**
- 6.9 彩色图像压缩



## 6.8 彩色图像中的噪声

- 5.2节讨论的噪声模型可用于彩色图像
- 通常，彩色图像的噪声内容在每个彩色通道中具有相同的特性，但噪声对不同彩色通道所造成的影响不同
- 然而，不同的噪声水平像是由每个彩色通道的相对照射强度的差异造成的
  - 例如，在CCD摄像机中，红色滤镜的使用将减少用于红色传感器的照射强度。CCD传感器在低照明水平下就是噪声源，因此，这种情况下得到的RGB图像红色分量图像与其他两幅分量图像相比往往是噪声源

# 6.8 彩色图像中的噪声

- 与在单色图像中相比，细粒噪声在彩色图像中不太引人注意



(a)~(c)由均值为0、方差为800的高斯噪声污染的红、绿和蓝分量图像；  
(d)最终的RGB图像

## 6.8 彩色图像中的噪声

- 噪声从一个彩色模型转换为另一个模型
  - 噪声图像的色调与饱和度分量明显降质了，这分别是由转换函数中求余弦与取最小值操作的非线性造成的
  - 图c中的强度分量比3个带噪声的RGB分量图像中的任何一个都要平滑一些，这是由亮度图像是RGB图像的平均这一事实造成的

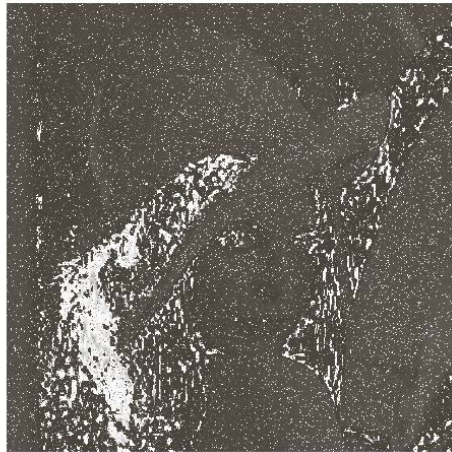


上图(d)中带噪声的彩色图像的HSI分量：(a)色调图像；(b)饱和度图像；(c)亮度图像



## 6.8 彩色图像中的噪声

- 在仅有一个RGB通道受噪声影响的情况下，到HSI的转换才将噪声扩散到所有HSI分量图像



- (a) 绿平面被椒盐噪声污染的RGB图像；
- (b) HSI图像的色调分量；
- (c) 饱和度分量；
- (d) 亮度分量



# 第6章 彩色图像处理

- 6.1 彩色基础
- 6.2 彩色模型
- 6.3 伪彩色图像处理
- 6.4 真彩色图像处理
- 6.5 彩色变换
- 6.6 平滑和锐化
- 6.7 基于彩色的图像分割
- 6.8 彩色图像中的噪声
- 6.9 彩色图像压缩**



## 6.9 彩色图像压缩

- 因为描述彩色所要求的比特数比描述灰度级所要求的比特数大3~4倍，所以数据压缩再存储和传输彩色图像中起着核心的作用。
- 对于前几节描述的RGB，CMY(K)和HSI图像，任何压缩对象的数据是每个彩色像素的分量，它们是传达彩色信息的方式。
- 压缩是减少或消除冗余和不相干数据的处理。



# 6.9 彩色图像压缩

## □ 实例

- 一幅24比特RGB全彩色图像
- 采用JPEG 2000压缩算法
- 原图每230比特数据，压缩后仅包含1个数据比特
- 假定压缩后图像在Internet上能于1分钟内传输完成，那么原图像的传输至少需要4小时





# 第7章 小波和多分辨率处理

- 小波变换的基函数是小型波，它具有变化的频率和有限的持续时间
- 小波是多分辨率理论这种信号处理和分析方法的基础，多分辨率理论涉及多个分辨率下的信号（或图像）表示与分析。
  - 这种方法的优势是：某种分辨率下无法检测的特性，在另一种分辨率下容易检测
- 本章从多分辨率的角度来审视基于小波的变换。
  - 主要内容集中于离散小波变换的开发和利用



# 第7章 小波和多分辨率处理

---

7.1 背景

7.2 多分辨率展开

7.3 一维小波变换

7.4 快速小波变换

7.5 二维小波变换

7.6 小波包



# 第7章 小波和多分辨率处理

---

## 7.1 背景

## 7.2 多分辨率展开

## 7.3 一维小波变换

## 7.4 快速小波变换

## 7.5 二维小波变换

## 7.6 小波包

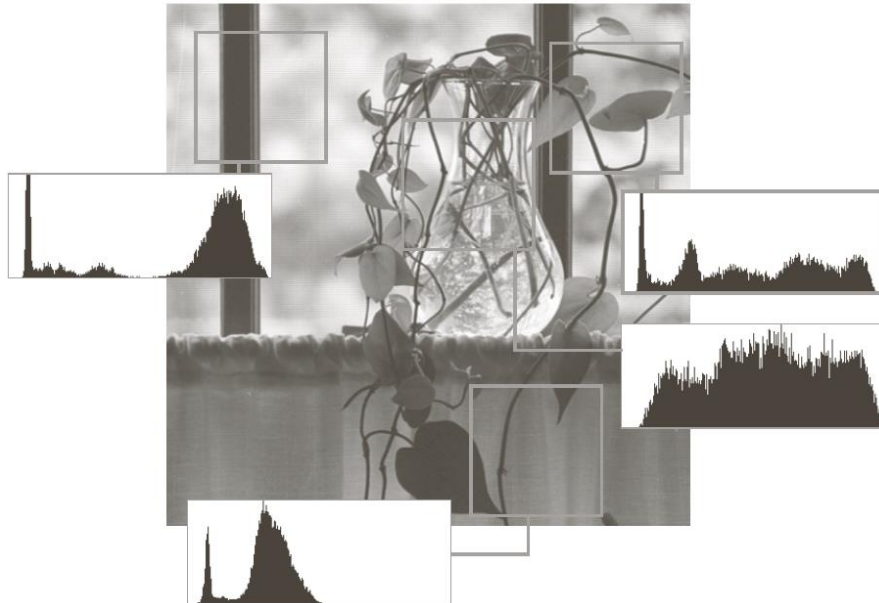
# 7.1 背景：多分辨率分析（MRA）

## □ 动机

- 图像中，区域由相似纹理和灰度级组成，不同区域结合形成物体
- 较小物体适合用较高分辨率分析
- 对于较大物体，用低分辨率分析即可

## □ 三个与MRA相关的图像操作：

- 图像金字塔，子带编码，哈尔变换



图像不同区域的局部直方图差异很大

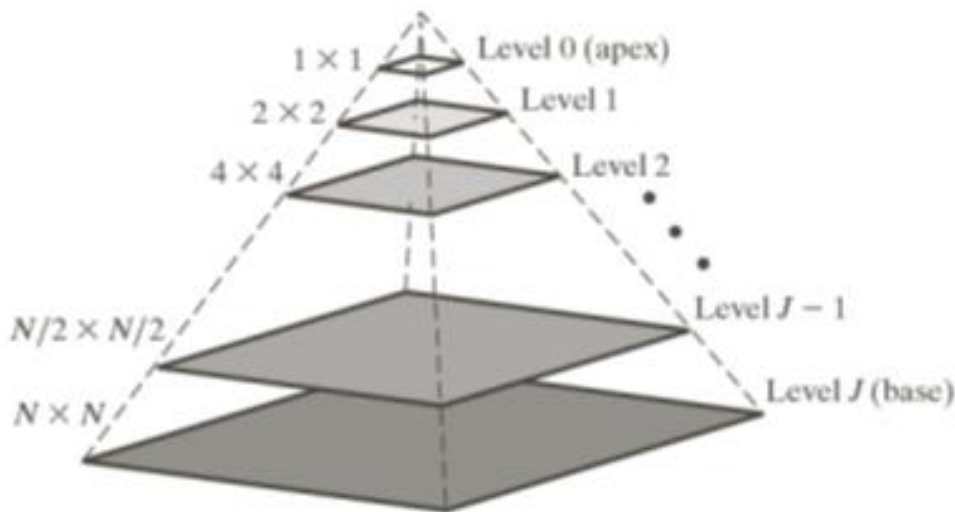
# 图像金字塔

## □ 图像金字塔：

- 一系列以金字塔形状排列的、分辨率逐渐降低的图像集合
- 底部：高分辨率表示；顶部：低分辨率近似
- 最初用机器视觉和图像压缩

## □ $P + 1$ 级图像金字塔像素总数是

$$N^2 \left( 1 + \frac{1}{4^1} + \frac{1}{4^2} + \cdots + \frac{1}{4^P} \right) \leq \frac{4}{3} N^2$$



两个问题：

第一，如何构建该金字塔？

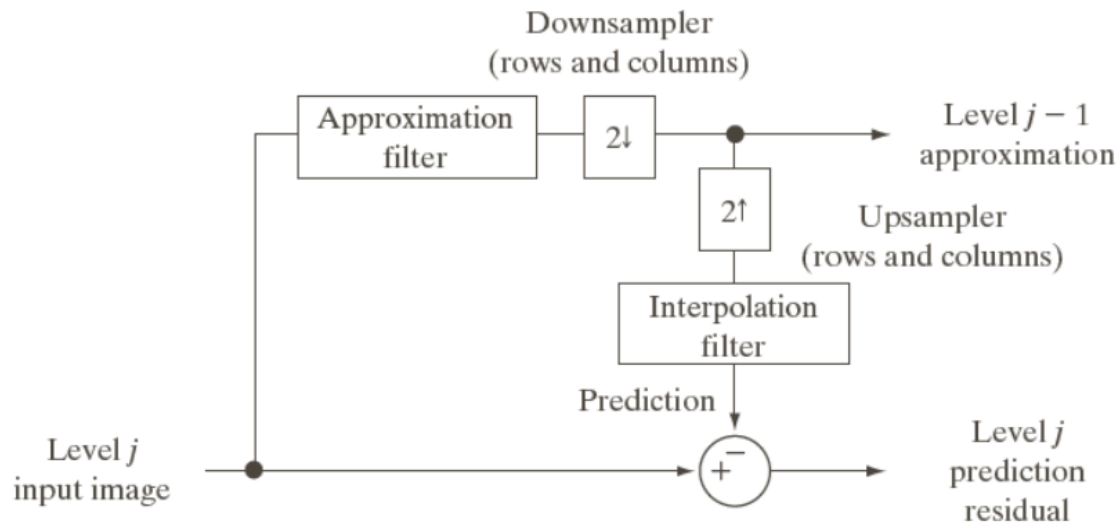
第二，金字塔相邻两级差别是什么？



# 图像金字塔

## □ 创建近似和残差金字塔

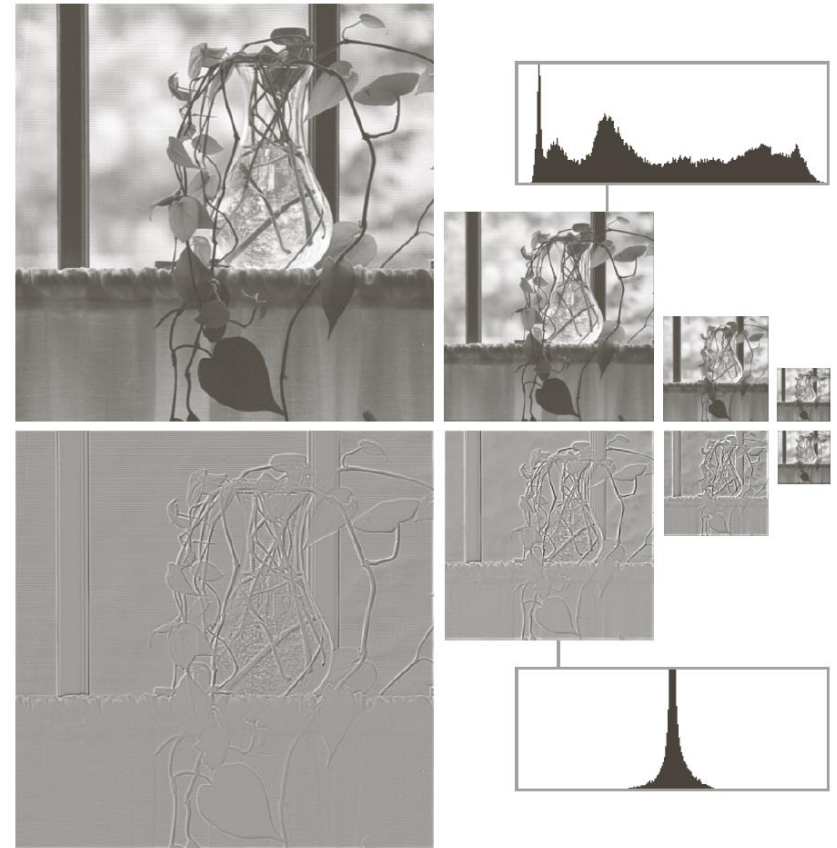
- 步骤一：通过近似滤波器后，2倍下采样，得到第 $j-1$ 级近似
  - ✓ 近似滤波器：邻域平均、高斯低通滤波、无滤波
- 步骤二：由步骤一产生的分辨率降低的近似，创建第 $j$ 级输入图像的一个估计
  - ✓ 插值方法：最近邻、双线性、双三次内插
- 步骤三：计算步骤二的预测图像和步骤一的输入之间的差，即预测残差





# 图像金字塔

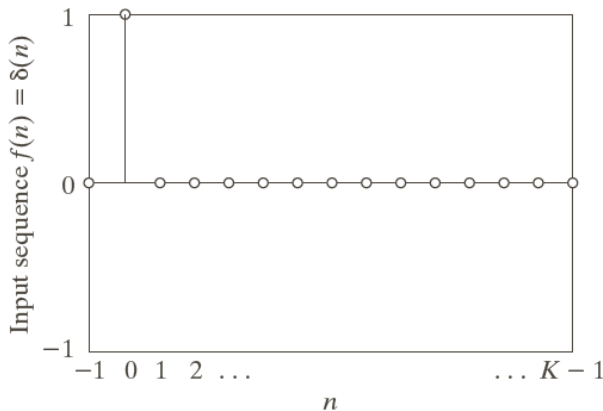
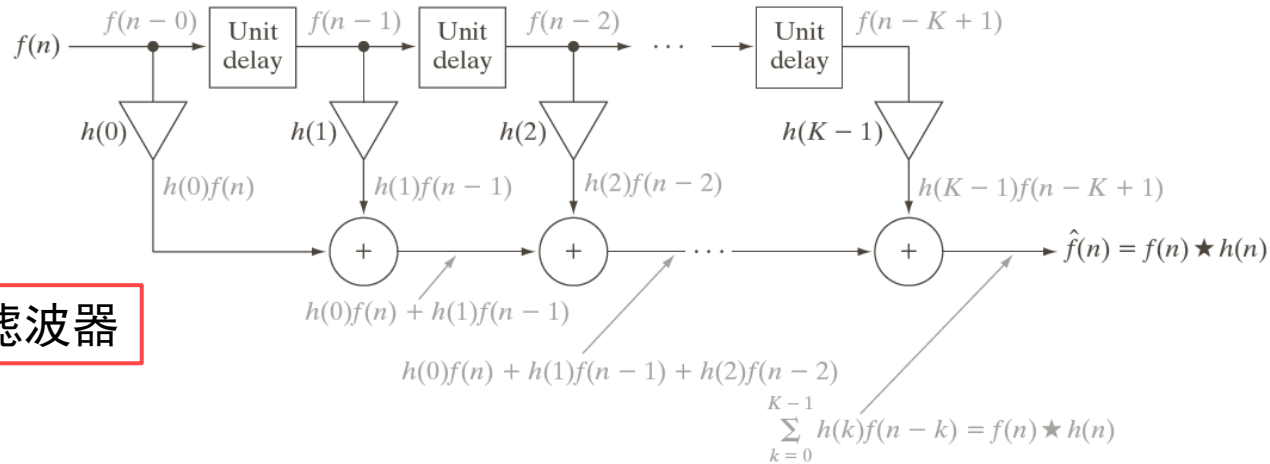
- 上图：近似金字塔，称为高斯金字塔
  - 因为构建金字塔时使用了高斯滤波器
- 分辨率越低，细节越少
  - 低分辨率适合分析大结构或图像整体内容
  - 高分辨率适合分析单个物体特性
- 下图：预测残差金字塔，通常称为拉普拉斯金字塔



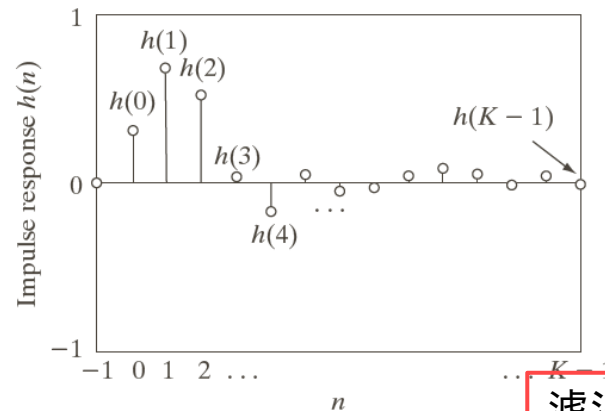
两种图像金字塔及它们的直方图

# 子带编码

- 在子带编码中，一副图像可分解为一组频带受限的分量，称为子带；子带可以重组在一起，无误差重建原始图像

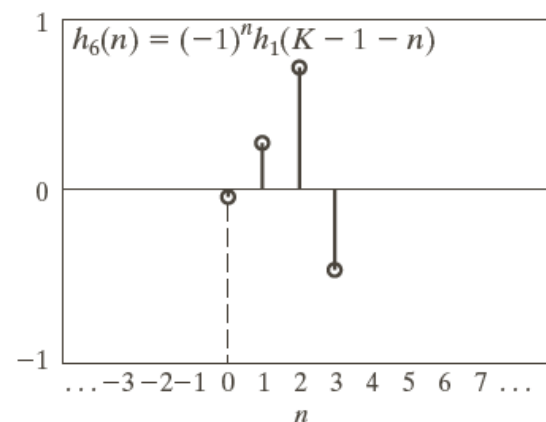
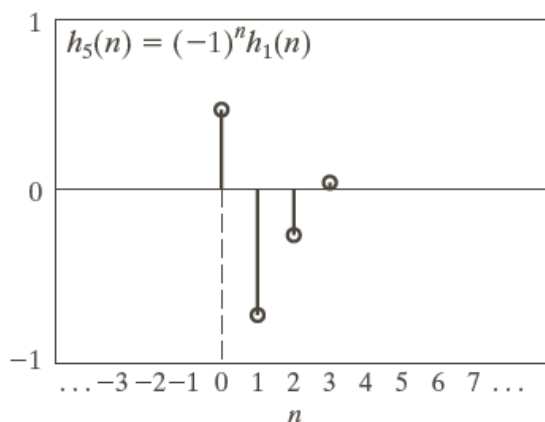
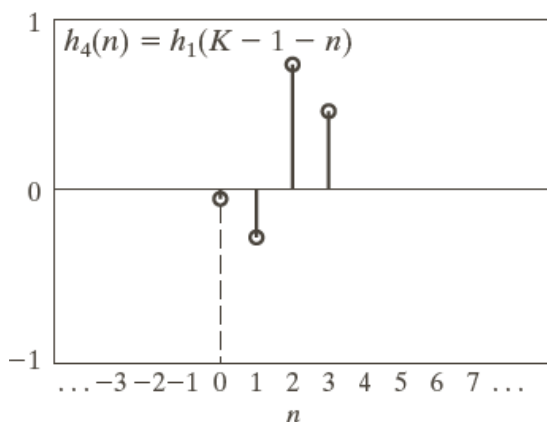
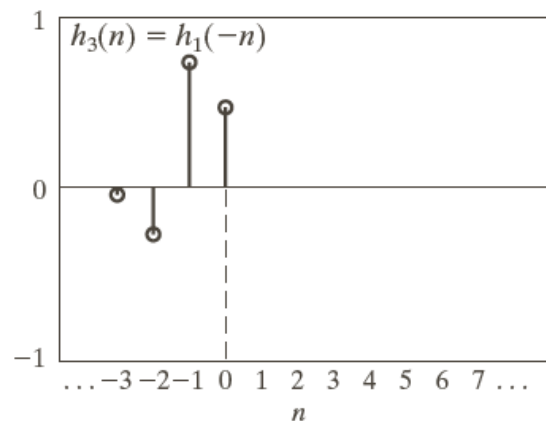
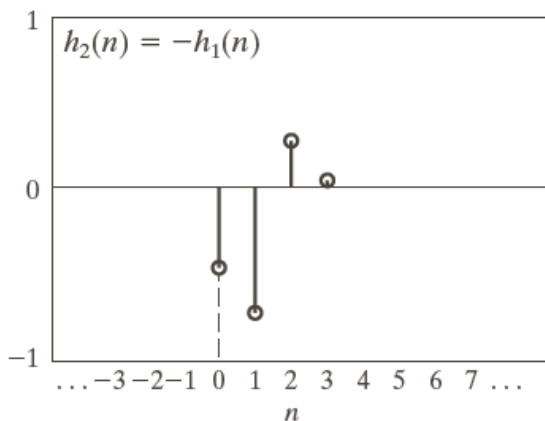
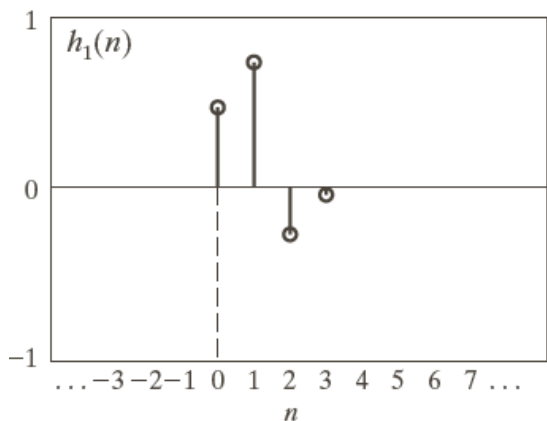


离散单位冲击



滤波器冲击响应

# 子带编码



6个功能相关的滤波器的冲击响应：

(a) 参考响应 (b) 符号反转 (c, d) 顺序反转（与延迟有关）

(e) 调制 (e) 顺序反转和调制

# 子带编码：两波段子带编码和解码

## 分析滤波器组和综合滤波器组串联

- 选择合适滤波器，实现完美重构

二带子带编码和解码系统

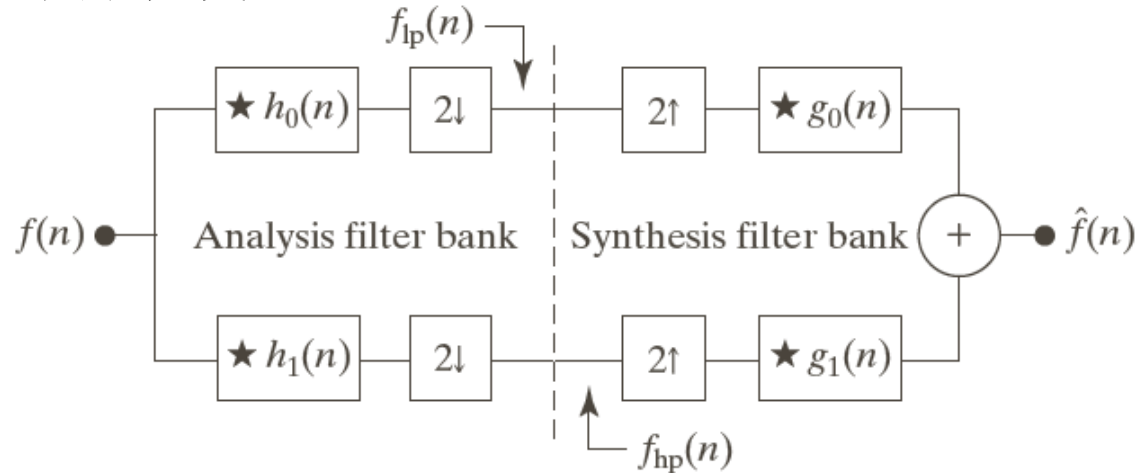
$$g_0(n) = (-1)^n h_1(n)$$

$$g_1(n) = (-1)^{n+1} h_0(n)$$

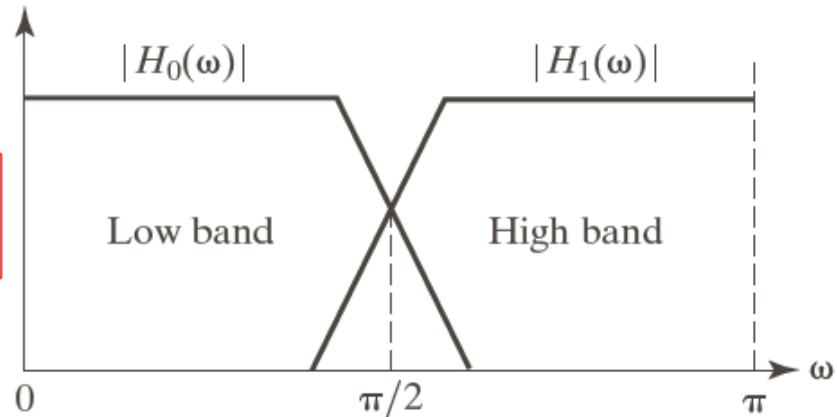
或

$$g_0(n) = (-1)^{n+1} h_1(n)$$

$$g_1(n) = (-1)^n h_0(n)$$

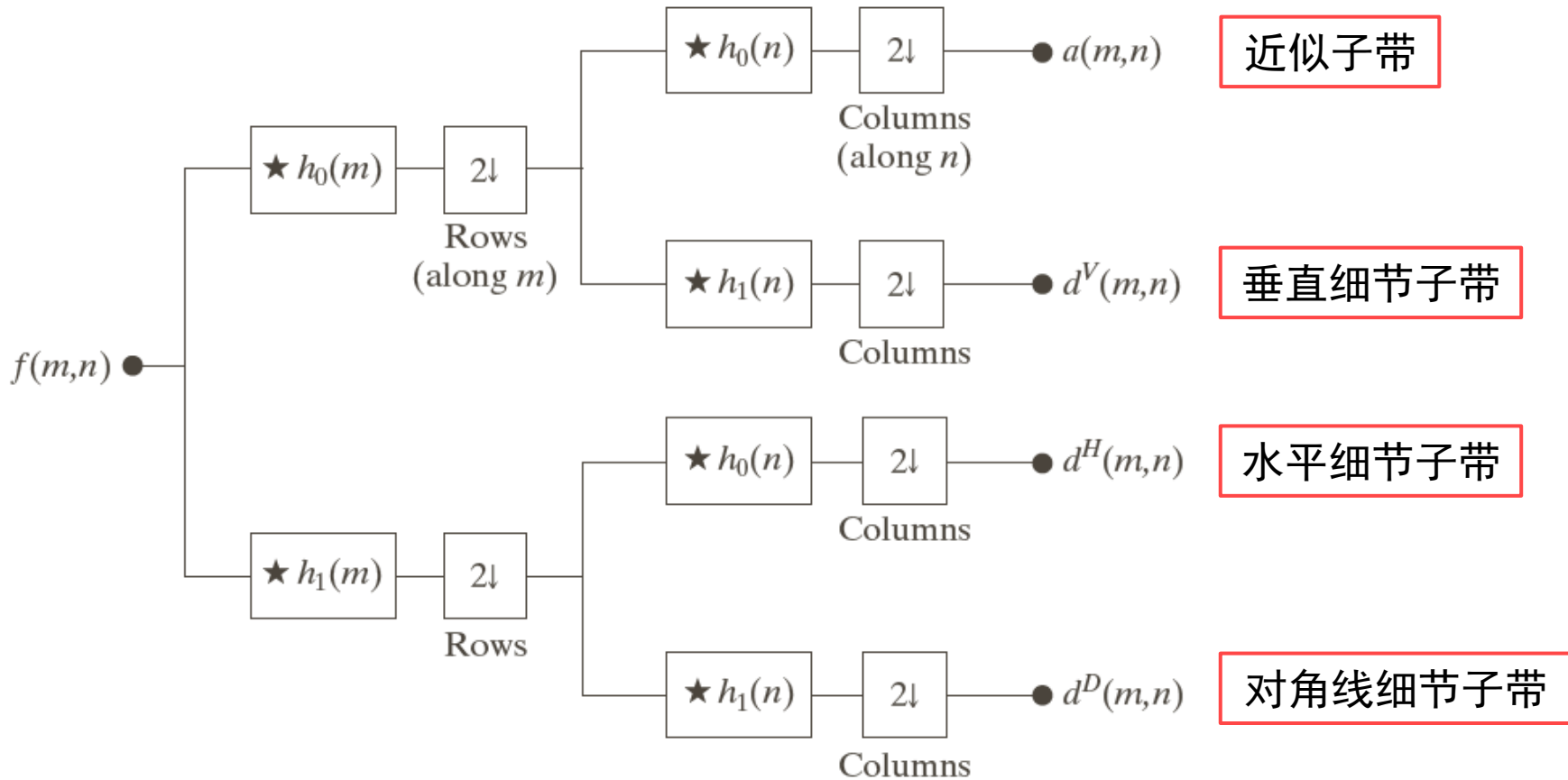


半波段滤波器，  
频谱可分



# 子带编码：二维子带编码

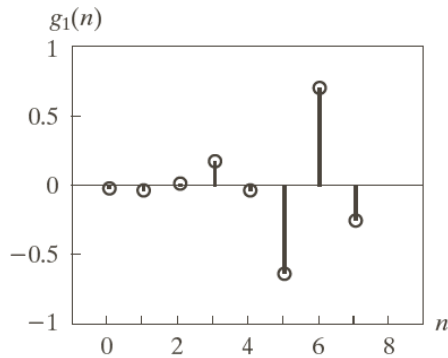
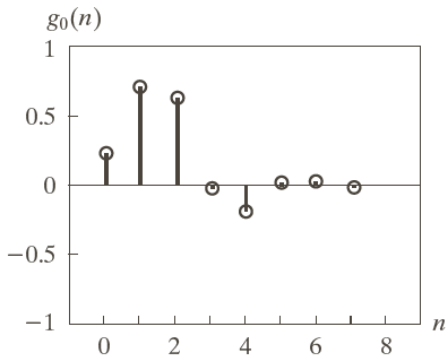
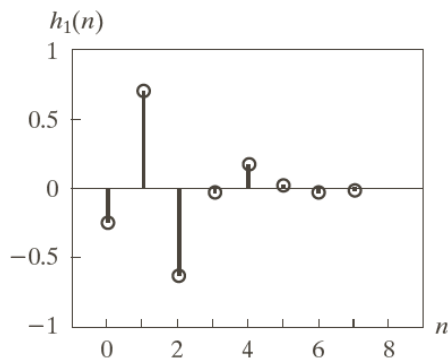
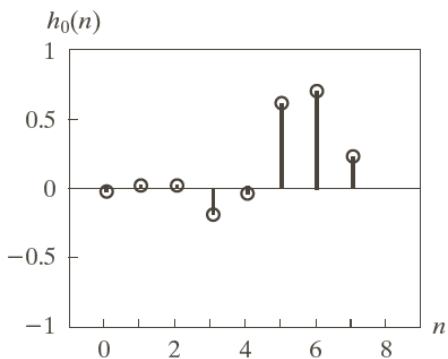
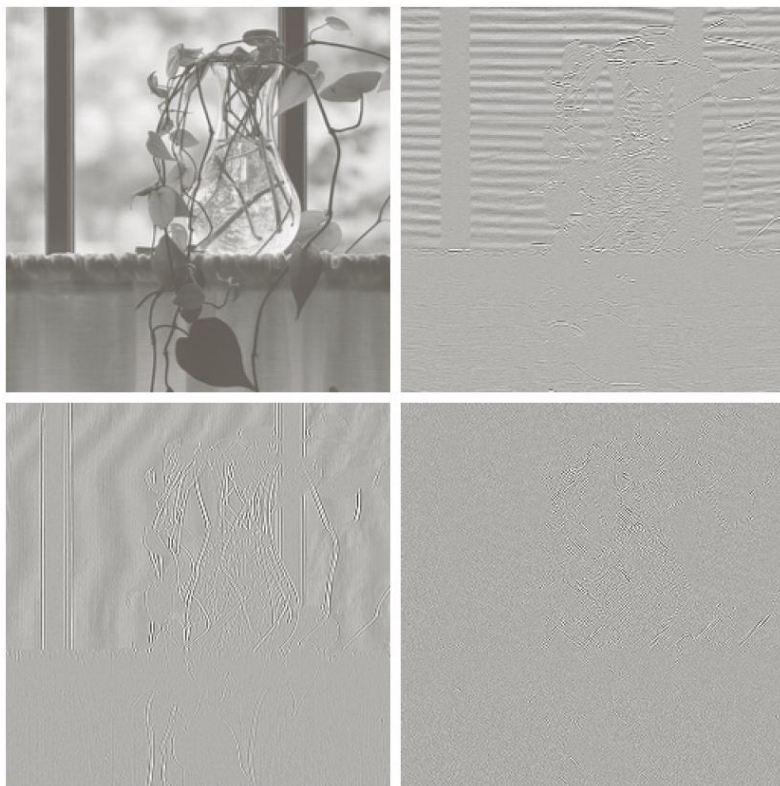
□ 将输入图像分解为四个子带图像



# 子带编码示例

$n$	$g_0(n)$
0	0.23037781
1	0.71484657
2	0.63088076
3	-0.02798376
4	-0.18703481
5	0.03084138
6	0.03288301
7	-0.01059740

4个8抽头Daubechies归一化  
正交滤波器的冲击响应  
(见公式7.1-14)



子带分离结果，4个子带分别是：  
(a) 近似子带      (b) 水平细节子带  
(c) 垂直细节子带    (d) 对角线细节子带



# 哈尔变换

## □ 哈尔 (Haar) 变换的矩阵表示

$$\mathbf{T} = \mathbf{H}\mathbf{F}\mathbf{H}^T$$

$\mathbf{H}$ :  $N \times N$  哈尔变换矩阵,  $\mathbf{F}$ :  $N \times N$  图像矩阵,  $\mathbf{T}$ :  $N \times N$  变换结果

## □ 哈尔基函数 ( $k = 2^p + q - 1$ )

$$h_0(z) = h_{00}(z) = \frac{1}{\sqrt{N}}, z \in [0,1]$$

$$h_k(z) = h_{pq}(z) = \frac{1}{\sqrt{N}} \begin{cases} z^{p/2}, & (q-1)/2^p \leq z < (q-0.5)/2^p \\ -z^{\frac{p}{2}}, & (q-0.5)/2^p \leq z < q/2^p \\ 0, & \text{其他}, z \in [0,1] \end{cases}$$

## □ 哈尔矩阵

$$H_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$$H_4 = \frac{1}{\sqrt{4}} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}$$

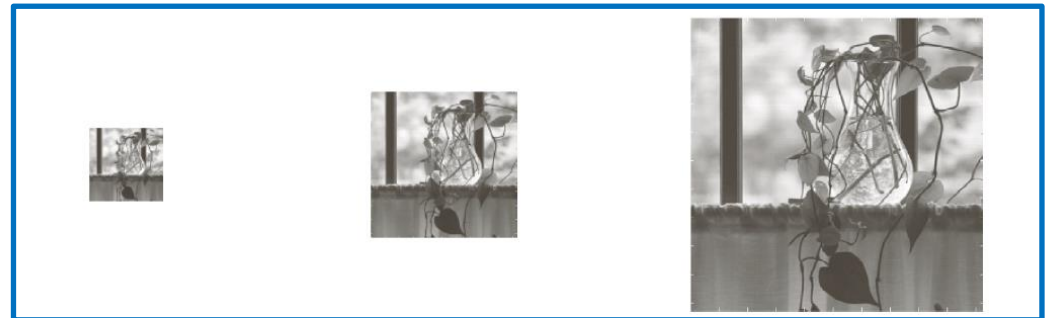
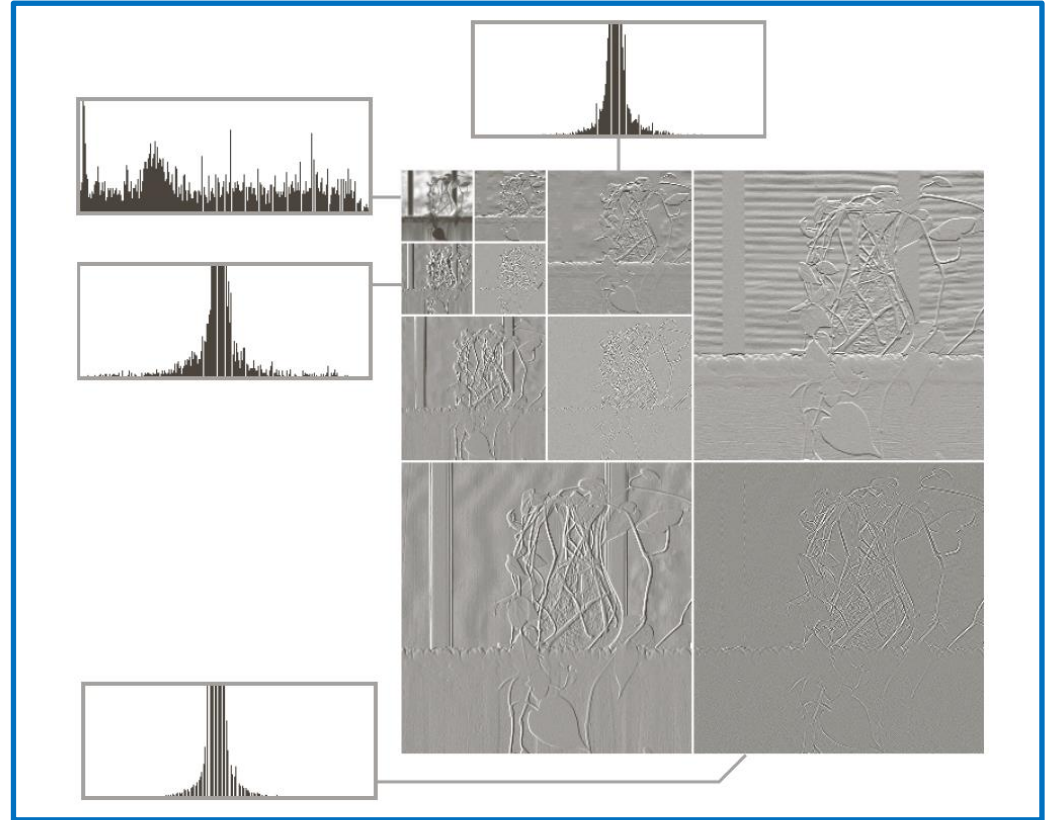


# 哈尔变换

$$H_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

$H_2$ 的行可用于定义一个2抽头完美重建滤波器组的分析滤波器 $h_0(n)$ 和 $h_1(n)$

每幅子图像描绘了原始图像中空间频率的一个特定频带



由 $H_2$ 哈尔基函数的离散小波变换得到三种不同分辨率的近似图像 ( $64 \times 64$ ,  $128 \times 128$ ,  $256 \times 256$ )



# 第7章 小波和多分辨率处理

---

7.1 背景

7.2 多分辨率展开

7.3 一维小波变换

7.4 快速小波变换

7.5 二维小波变换

7.6 小波包



## 7.2 多分辨率展开

- 在多分辨率分析中，**尺度函数**被用于建立一个函数或一幅图像的一系列近似，每个近似与其最近邻近似在分辨率方面都用基2来区分
- 使用称为**小波**的附加函数对相邻近似之间的差进行编码
- 基础概念
  - 级数展开
  - 尺度函数
  - 小波函数



# 级数展开

- 信号或函数  $f(x)$  通常能分解为一系列展开函数的线性组合

$$f(x) = \sum_k \alpha_k \varphi_k(x)$$

- 如果展开唯一，则称  $\varphi_k(x)$  为基函数
- 可展开的函数形成了一个函数空间，成为展开集合的闭合跨度：

$$V = \overline{\text{Span}\{\varphi_k(x)\}_k}$$

- 展开系数：

$$\alpha_k = \langle \tilde{\varphi}_k(x), f(x) \rangle = \int \tilde{\varphi}_k^*(x) f(x) dx$$



# 尺度函数

- 考虑由整数平移和实数二值尺度、平方可积函数 $\varphi(x)$ 组成的展开函数集合，即 $\{\varphi_{j,k}(x)\}$ ，其中

$$\varphi_{j,k}(x) = 2^{\frac{j}{2}} \varphi(2^j x - k)$$

$\varphi(x)$ 被称为尺度函数

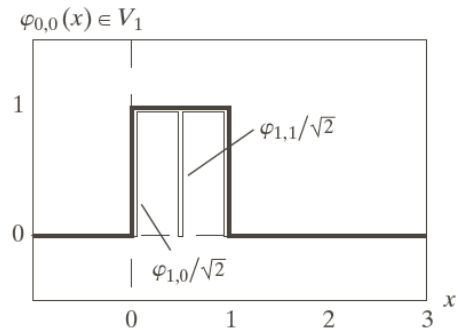
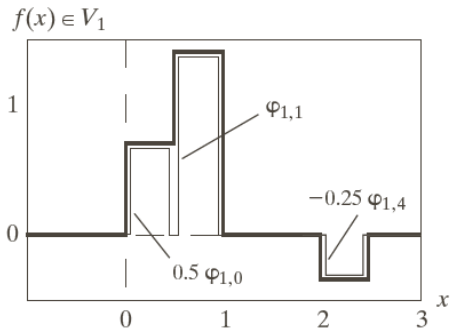
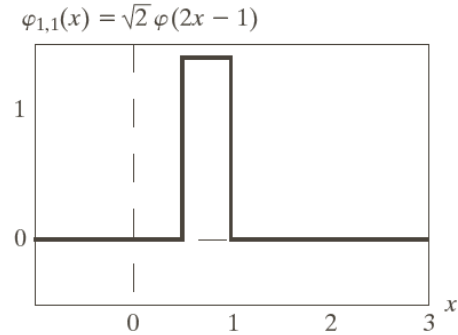
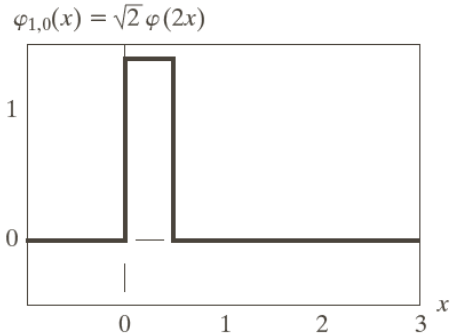
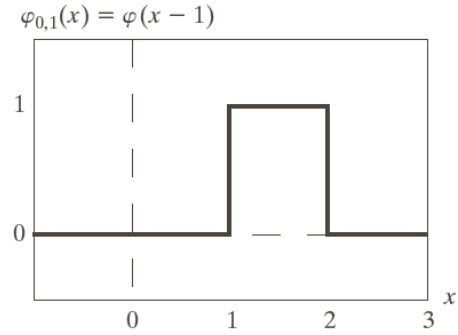
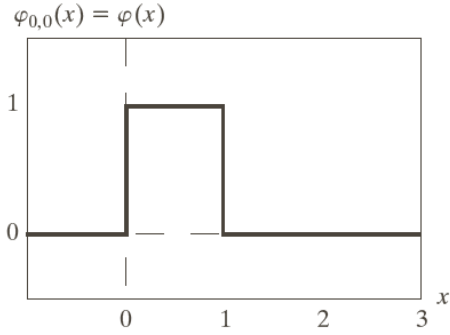
- 选择合适的 $\varphi(x)$ ，可使 $\varphi_{j,k}(x)$ 跨越 $L^2(\mathbf{R})$ ，即所有可度量的、平方可积函数的集合：

$$V_j = \overline{\text{Span}\{\varphi_{j,k}(x)\}_k}$$

- 增大 $j$ 会增大 $V_j$ 的大小，进而允许子空间中包含具有较小变换的变量或较细的细节函数
  - 随着 $j$ 的增大，用于表示子空间函数的 $\varphi_{j,k}(x)$ 会变窄， $x$ 有较小变化即可分开



# 尺度函数：哈尔尺度函数

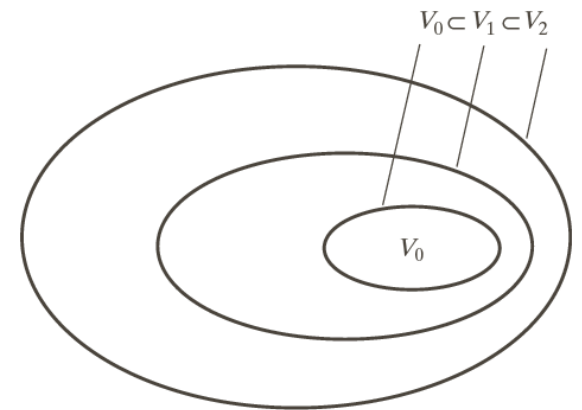


# 尺度函数

## □ 简单尺度函数满足多分辨率分析（MRA）4个基本要求

- MRA要求1：尺度函数对其整数平移是正交的
- MRA要求2：低尺度的尺度函数跨越的子空间，嵌套在高尺度跨越的尺度空间内，即

$$V_{-\infty} \subset \dots \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \dots \subset V_{\infty}$$



- MRA要求3：唯一对所有的 $V_j$ 通用的函数是 $f(x) = 0$ 
  - ✓ 没有信息的函数 $V_{-\infty} = \{0\}$
- MRA要求4：任何函数都可以按任意精度表示
  - ✓ 任意空间的展开函数，都可以由其本身的双倍分辨率副本建立





# 小波函数

- 定义小波集合 $\{\psi_{j,k}(x)\}$ :  $\psi_{j,k}(x) = 2^{\frac{j}{2}}\psi(2^j x - k)$
- 小波函数跨越的空间:

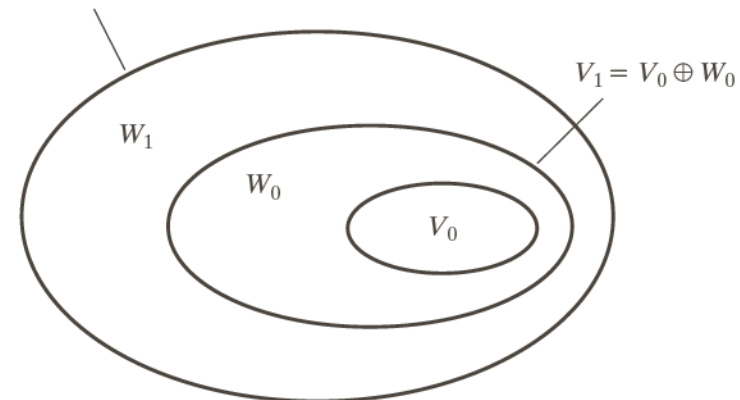
$$W_j = \overline{\text{Span}\{\psi_{j,k}(x)\}_k}$$

- 如果函数 $f(x) \in W_j$ , 则有:

$$f(x) = \sum_k \alpha_k \psi_{j,k}(x)$$

- 尺度函数和小波函数子空间关系

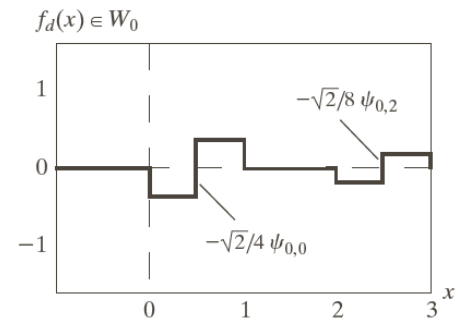
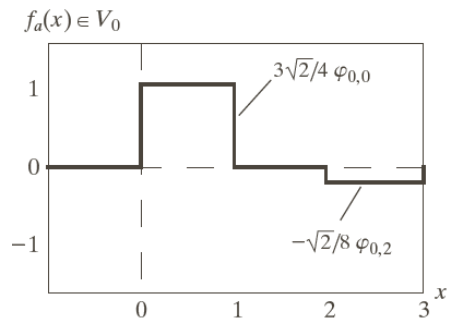
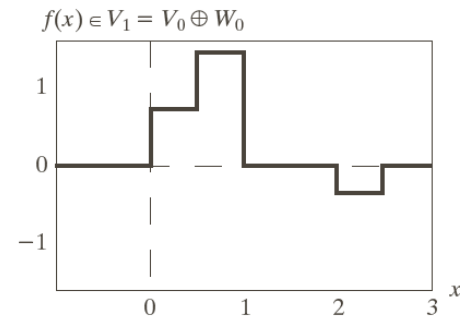
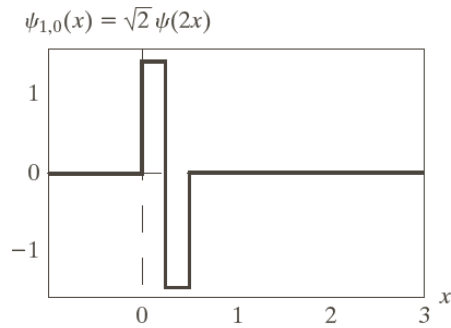
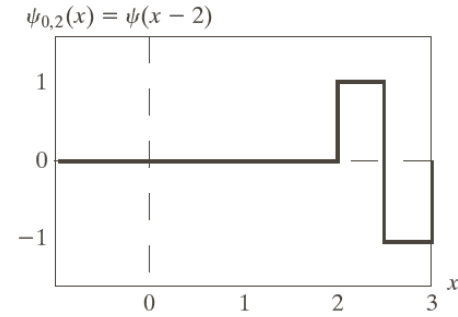
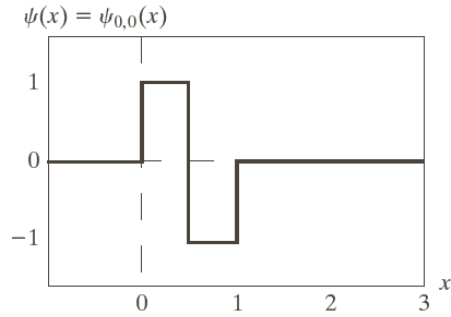
$$V_{j+1} = V_j \oplus W_j \quad V_2 = V_1 \oplus W_1 = V_0 \oplus W_0 \oplus W_1$$





# 小波函数

## □ 哈尔小波函数系数





# 第7章 小波和多分辨率处理

---

7.1 背景

7.2 多分辨率展开

7.3 一维小波变换

7.4 快速小波变换

7.5 二维小波变换

7.6 小波包

# 7.3 一维小波变换

## □ 三种小波变换

- 一般的小波级数展开  $\longleftrightarrow$  傅里叶级数展开
- 离散小波变换  $\longleftrightarrow$  离散傅里叶变换
- 连续小波变换  $\longleftrightarrow$  积分傅里叶变换

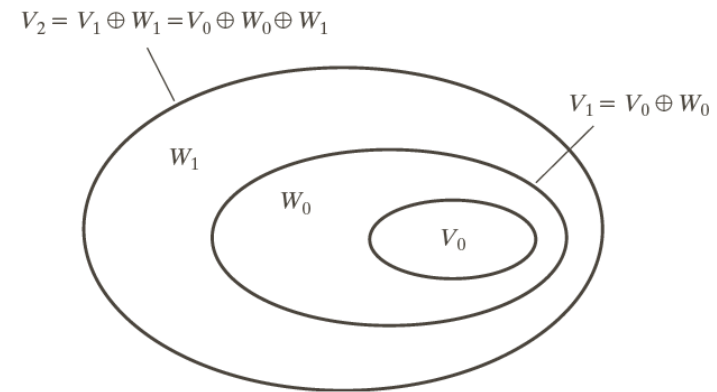
## □ 小波级数展开

$$f(x) = \sum_k c_{j_0}(k) \varphi_{j_0,k}(x) + \sum_{j=j_0}^{\infty} \sum_k d_j(k) \psi_{j,k}(x)$$

$c_{j_0}(k)$ : 近似或尺度系数;  $d_j(k)$ : 细节或小波系数

$$c_{j_0}(k) = \langle f(x), \varphi_{j_0,k}(x) \rangle = \int f(x) \varphi_{j_0,k}(x) dx$$

$$d_j(k) = \langle f(x), \psi_{j,k}(x) \rangle = \int f(x) \psi_{j,k}(x) dx$$





# 小波级数展开

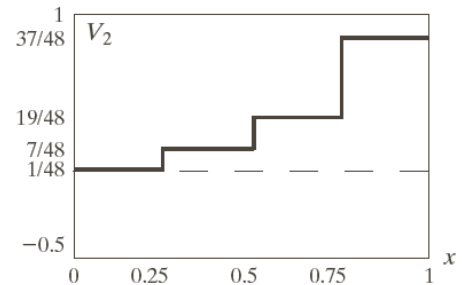
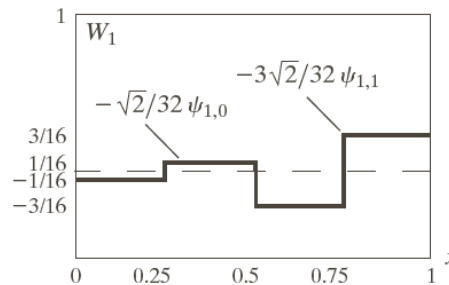
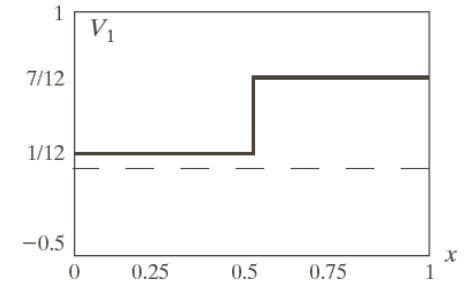
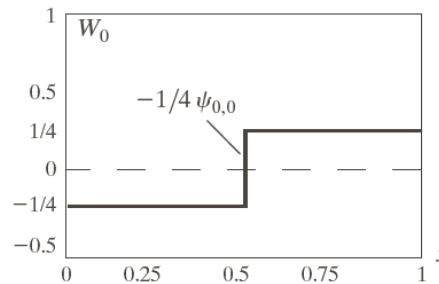
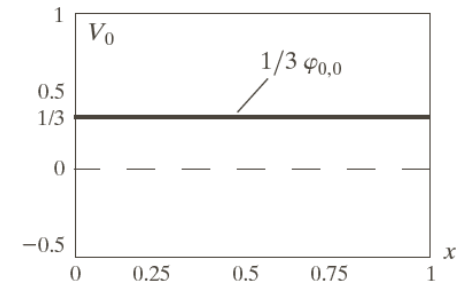
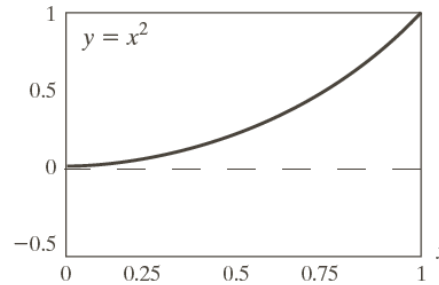
- 随着更高尺度的叠加，近似变得  
更接近测试函数的精确表示。
- 当  $j \rightarrow \infty$  时，可实现精确重构表示

$$y = \begin{cases} x^2, & 0 \leq x \leq 1 \\ 0, & \text{其他} \end{cases}$$

$$c_0(0) = \int_0^1 x^2 \varphi_{0,0}(x) dx = \int_0^1 x^2 dx = \frac{1}{3}$$

$$d_0(0) = \int_0^1 x^2 \psi_{0,0}(x) dx = -\frac{1}{4}$$

.....



使用哈尔小波的  $y = x^2$  的小波级数展开



# 离散小波变换

□ 如果待展开的函数是离散的（即数字序列），得到的系数就称之为离散小波变换（DWT）

□ 正向DWT系数

$$\text{近似系数: } W_{\varphi}(j_0, k) = \frac{1}{\sqrt{M}} \sum_n f(n) \varphi_{j_0, k}(n)$$

$$\text{细节系数: } W_{\psi}(j, k) = \frac{1}{\sqrt{M}} \sum_n f(n) \psi_{j, k}(n), j \geq j_0$$

□ 反向DWT:

$$f(n) = \frac{1}{\sqrt{M}} \sum_k W_{\varphi}(j_0, k) \varphi_{j_0, k}(n) + \frac{1}{\sqrt{M}} \sum_{j=j_0}^{\infty} \sum_k W_{\psi}(j, k) \psi_{j, k}(n)$$



# 连续小波变换

- 连续小波变换 (CWT) 将一个连续函数变换为两个变量 (平移和尺度) 的高冗余度函数
- 连续平方可积函数  $f(x)$  的连续小波变换与实数值小波  $\psi(x)$  的关系定义为

$$W_{\psi}(s, \tau) = \int_{-\infty}^{\infty} f(x) \psi_{s, \tau}(x) dx$$
$$\psi_{s, \tau}(x) = \frac{1}{\sqrt{s}} \psi\left(\frac{x - \tau}{s}\right)$$

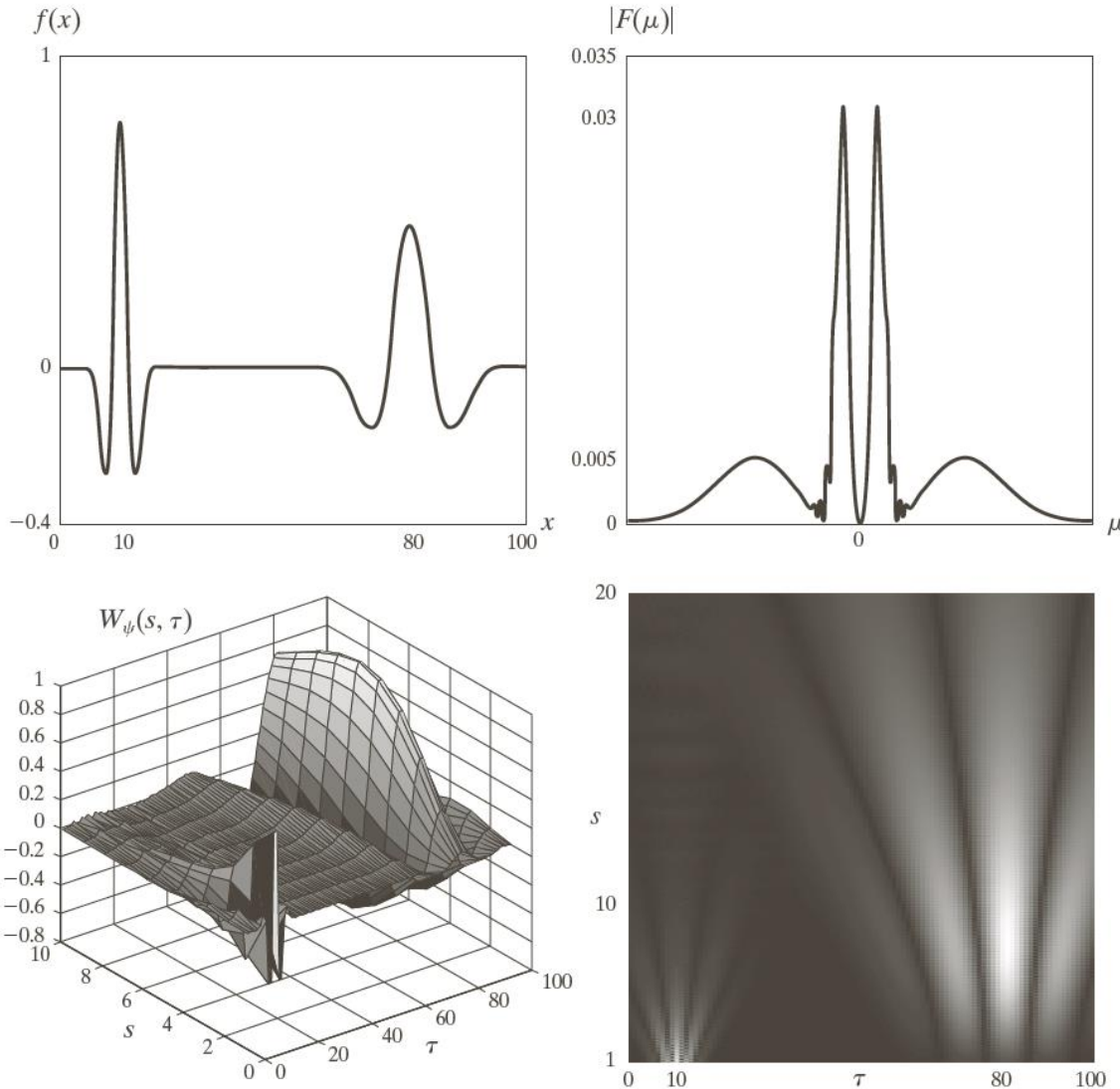
$s$ : 尺度参数  
 $\tau$ : 平移参数

- 给定  $W_{\psi}(s, \tau)$ , 连续小波反变换为

$$f(x) = \frac{1}{C_{\psi}} \int_0^{\infty} \int_{-\infty}^{\infty} W_{\psi}(s, \tau) \frac{\psi_{s, \tau}(x)}{s^2} d\tau ds$$
$$C_{\psi} = \int_{-\infty}^{\infty} \frac{|\psi(\mu)|^2}{|\mu|} d\mu$$



# 连续小波变换



a b  
c d

**FIGURE 7.16**  
The continuous wavelet transform (c and d) and Fourier spectrum (b) of a continuous 1-D function (a).

连续小波变换[(c)和(d)]和连续一维函数(a)的傅里叶谱(b)

$$f(x) = \psi_{1,10}(x) + \psi_{6,80}(x)$$



# 第7章 小波和多分辨率处理

---

7.1 背景

7.2 多分辨率展开

7.3 一维小波变换

7.4 **快速小波变换**

7.5 二维小波变换

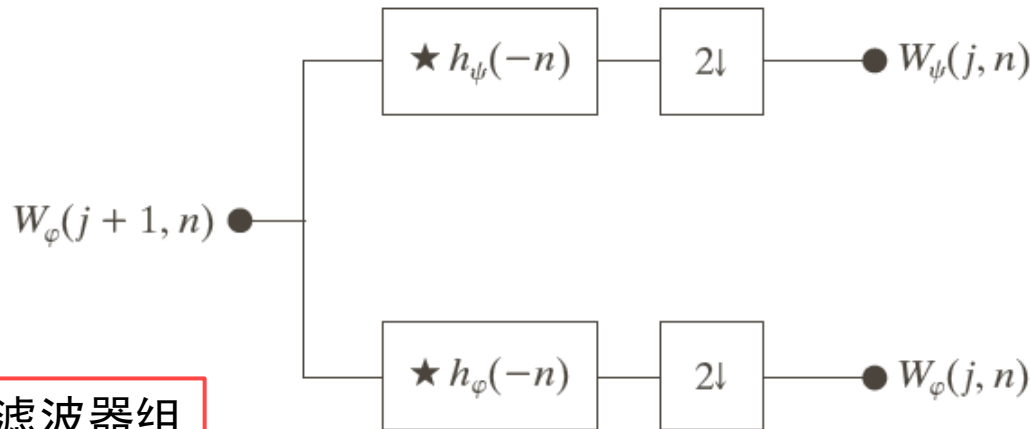
7.6 小波包



# 7.4 快速小波变换

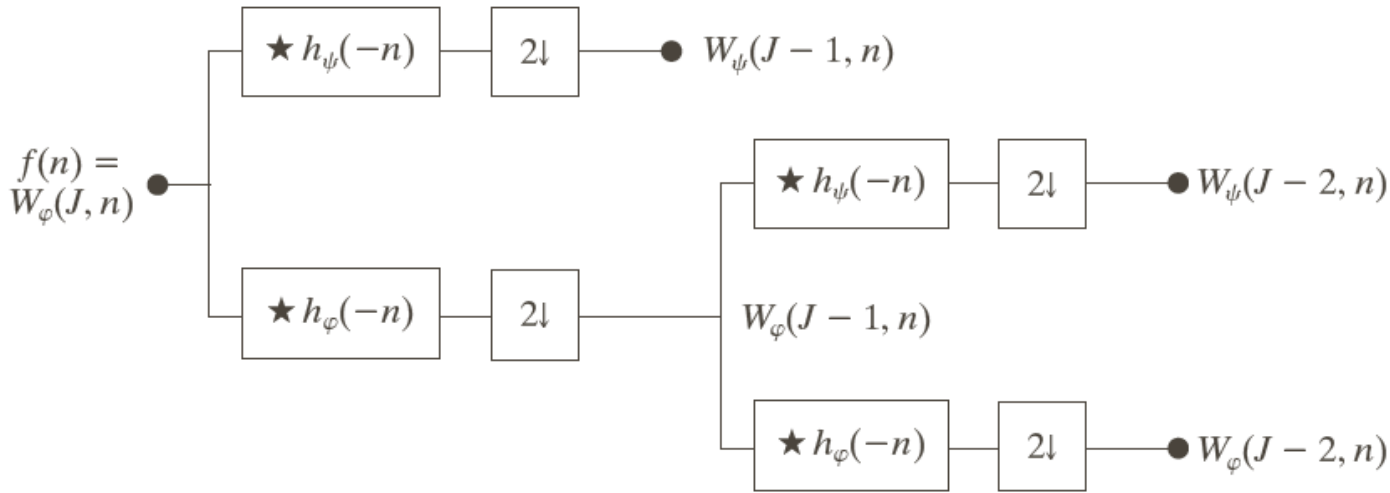
- 快速小波变换 (FWT) 是实现离散小波变换 (DWT) 的高效计算, 也称Mallat人字形算法
  - 类似于2子带的子带编码

$$W_\psi(j, k) = h_\psi(-n) \star W_\varphi(j + 1, n) \Big|_{n=2k, k \geq 0}$$
$$W_\varphi(j, k) = h_\varphi(-n) \star W_\varphi(j + 1, n) \Big|_{n=2k, k \geq 0}$$



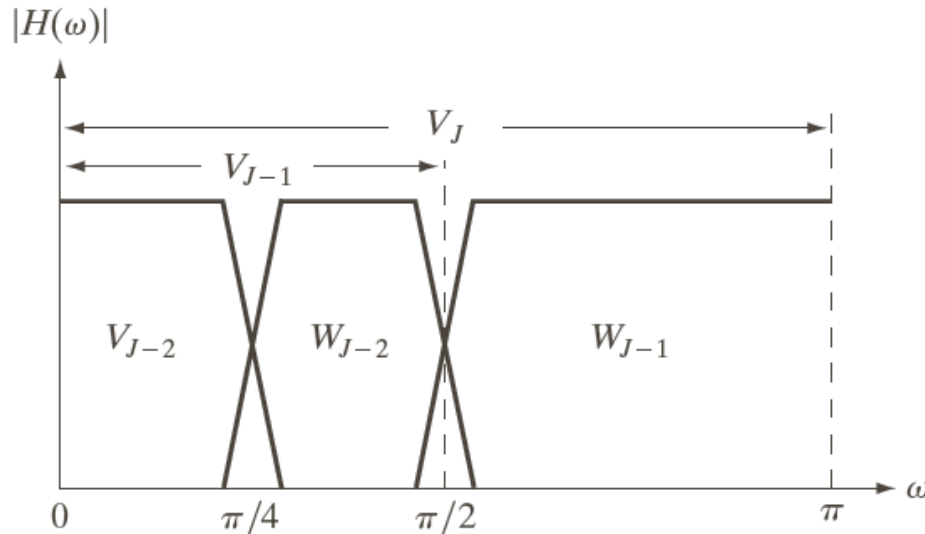
一个FWT分析滤波器组

# 7.4 快速小波变换



a  
b

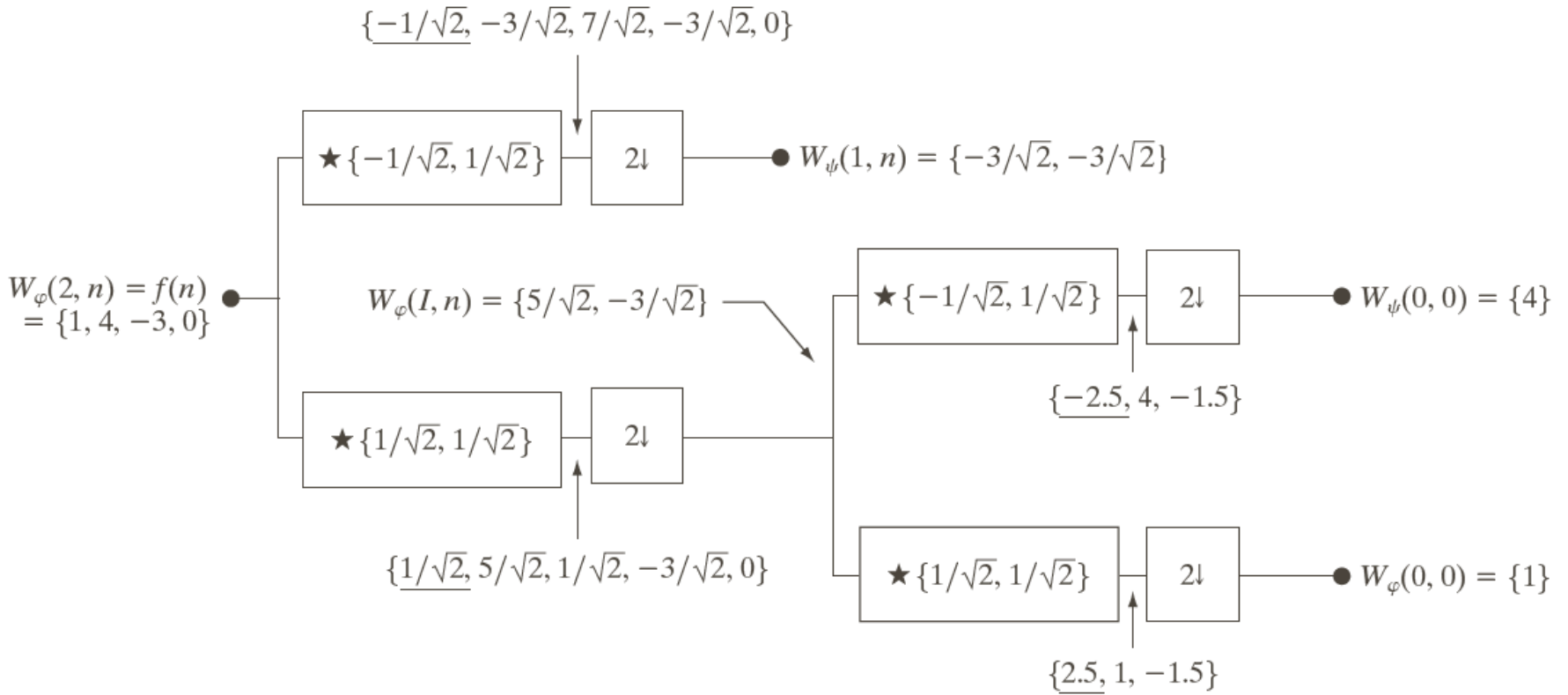
**FIGURE 7.18**  
(a) A two-stage or two-scale FWT analysis bank and (b) its frequency splitting characteristics.



(a) 一个二级或二尺度 FWT分析滤波器组  
(b) 其频谱分离特性



# 7.4 快速小波变换

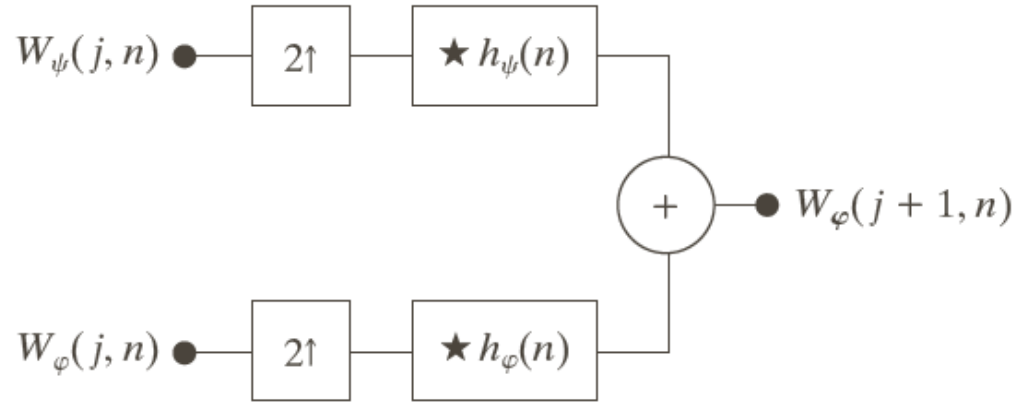


使用哈尔尺度和小波向量计算序列  $\{1, 4, -3, 0\}$  的一个二尺度快速小波变换

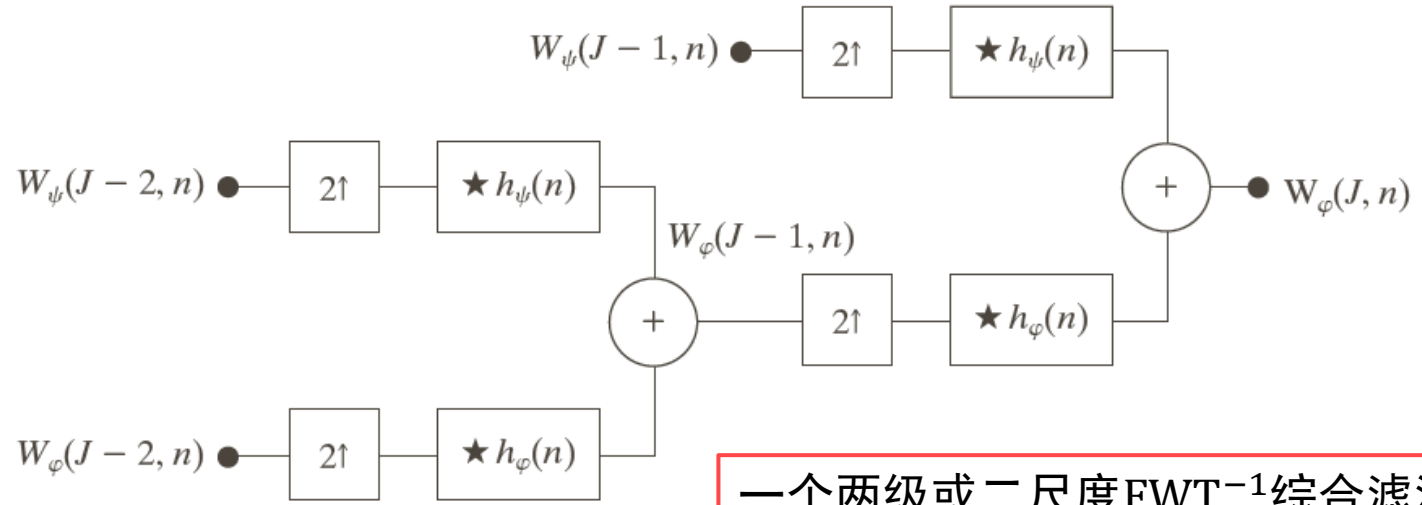


# 7.4 快速小波变换：反变换

FWT<sup>-1</sup>综合滤波器组



$$W_\phi(j+1, k) = h_\phi(k) \star W_\phi^{2\uparrow}(j, k) + h_\psi(k) \star W_\psi^{2\uparrow}(j, k) \Big|_{k \geq 0}$$

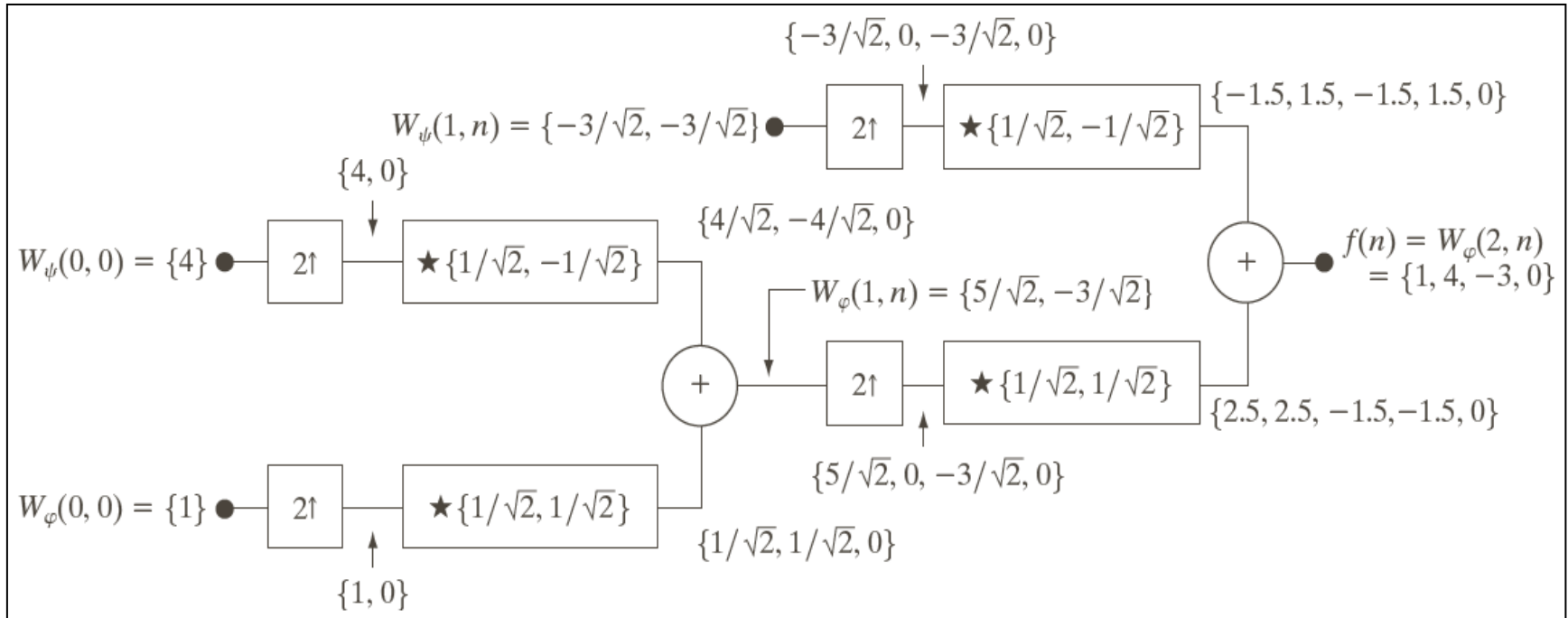


一个两级或二尺度FWT<sup>-1</sup>综合滤波器组



# 7.4 快速小波变换：反变换

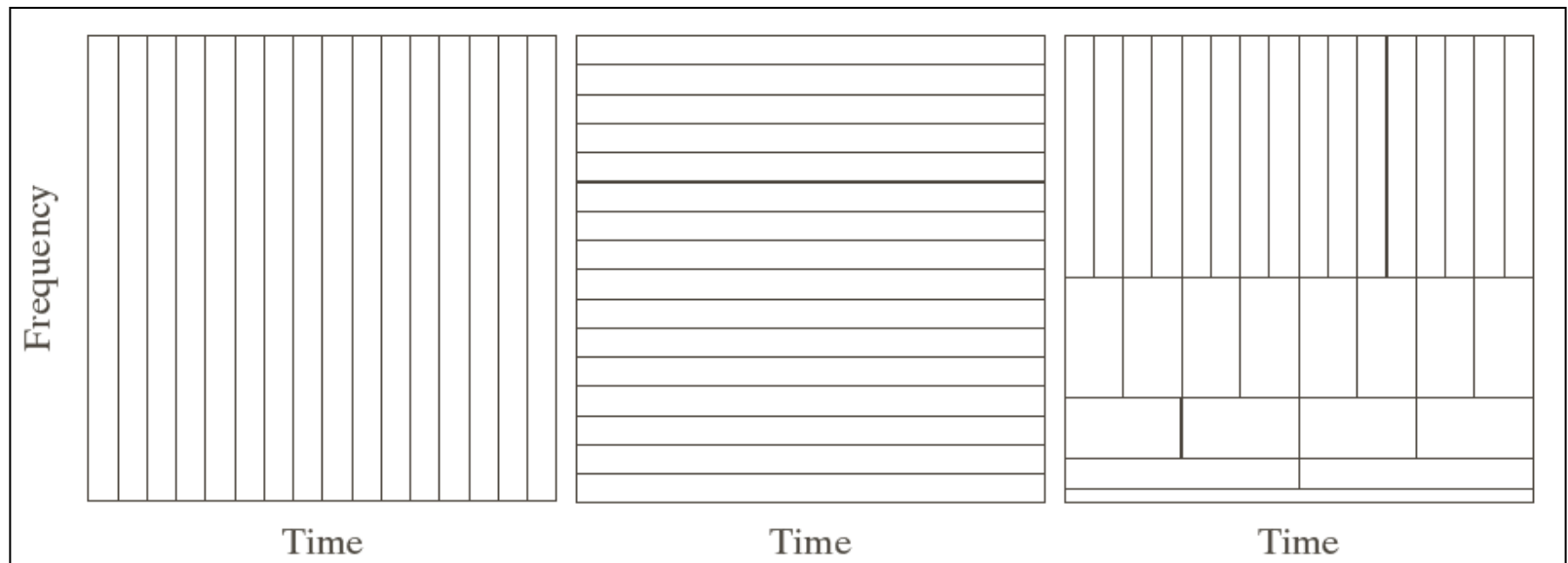
- 使用哈尔尺度和小波函数计算序列  $\{1, 4, -1.5, \sqrt{2}, -1, -1.52\sqrt{2}\}$  的二尺度快速小波反变换





# 7.4 快速小波变换

- 标准时域基指明了事件发生的时刻，但不提供频率信息
- 正弦基指出了发生较长时间的事件中出现的频率，但没有提供时间分辨率
- FWT的时间-频率片中时间和频率分辨率是变化的，但每个片的面积相同



(a) 冲激函数基

(a) 正弦函数(FFT)基

(c) FWT基



# 第7章 小波和多分辨率处理

---

7.1 背景

7.2 多分辨率展开

7.3 一维小波变换

7.4 快速小波变换

7.5 **二维小波变换**

7.6 小波包



# 7.5 二维小波变换

- 在二维情况下，需要1个二维尺度函数 $\varphi(x, y)$ 和3个二维小波 $\psi^H(x, y)$ ,  $\psi^V(x, y)$ 和 $\psi^D(x, y)$ 。每个二维小波都是两个一维函数的乘积

可分离尺度函数

$$\varphi(x, y) = \varphi(x)\varphi(y)$$

列方向变化（水平边缘）

$$\psi^H(x, y) = \psi(x)\varphi(y)$$

行方向变化（垂直边缘）

$$\psi^V(x, y) = \varphi(x)\psi(y)$$

对角线方向变化

$$\psi^D(x, y) = \psi(x)\psi(y)$$



# 7.5 二维小波变换

□ 大小为  $M \times N$  的图像  $f(x, y)$  的离散小波变换是

$$W_\varphi(j_0, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \varphi_{j_0, m, n}(x, y)$$

$$W_\psi^i(j, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \psi_{j, m, n}^i(x, y), i = \{H, V, D\}$$

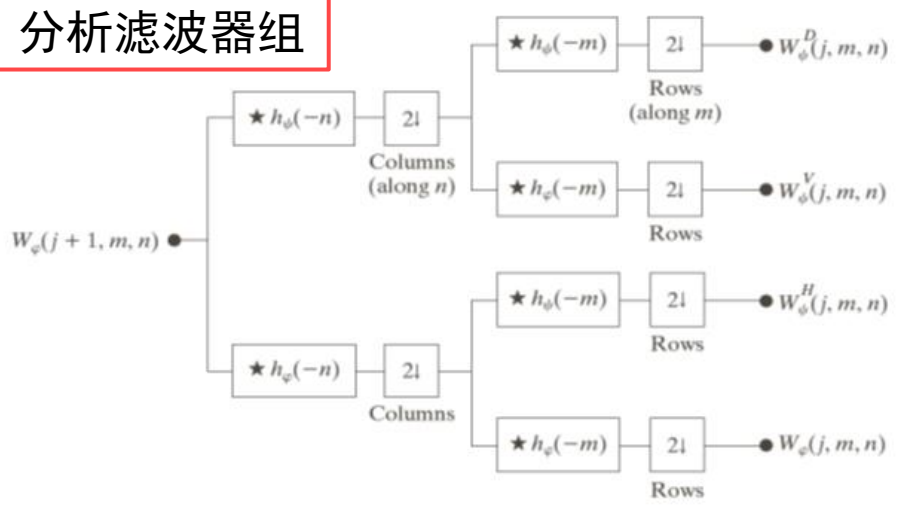
□  $f(x, y)$  离散小波反变换

$$\begin{aligned} f(x, y) = & \frac{1}{\sqrt{MN}} \sum_m \sum_n W_\psi(j_0, m, n) \varphi_{j_0, m, n}(x, y) \\ & + \frac{1}{\sqrt{MN}} \sum_{i=H, V, D} \sum_{j=j_0}^{\infty} \sum_m \sum_n W_\psi^i(j, m, n) \psi_{j, m, n}^i(x, y) \end{aligned}$$

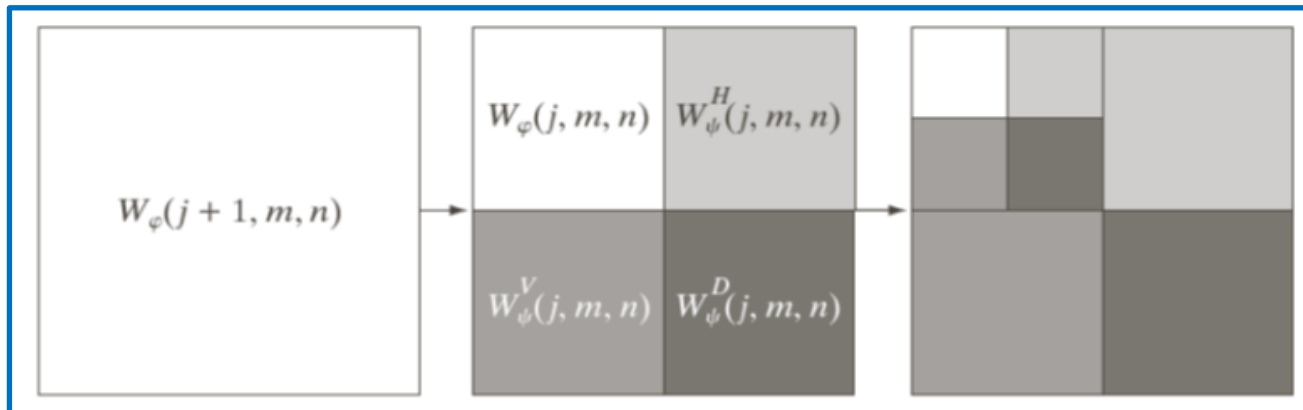
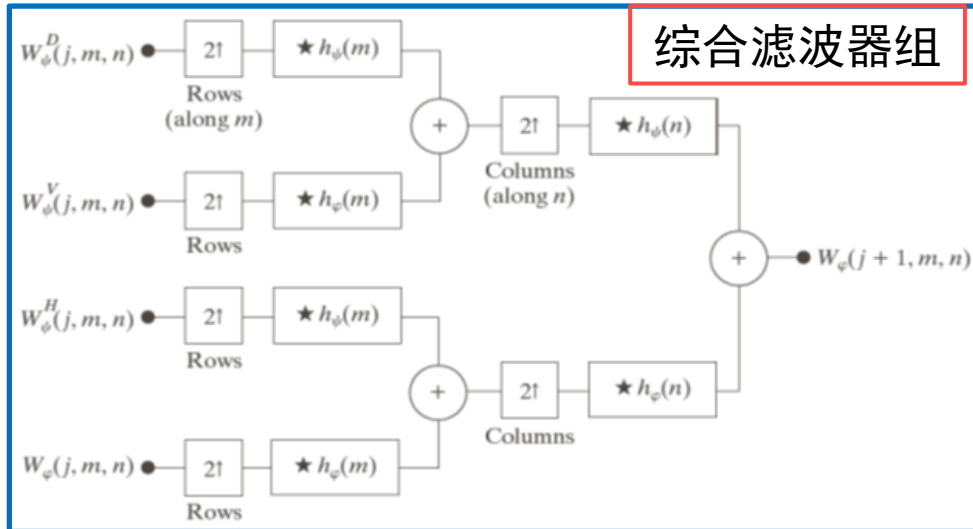
# 7.6 二维小波变换

## 二维快速小波变换

### 分析滤波器组

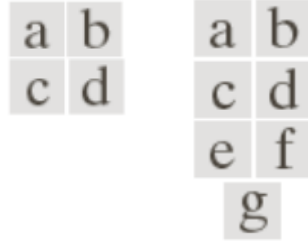
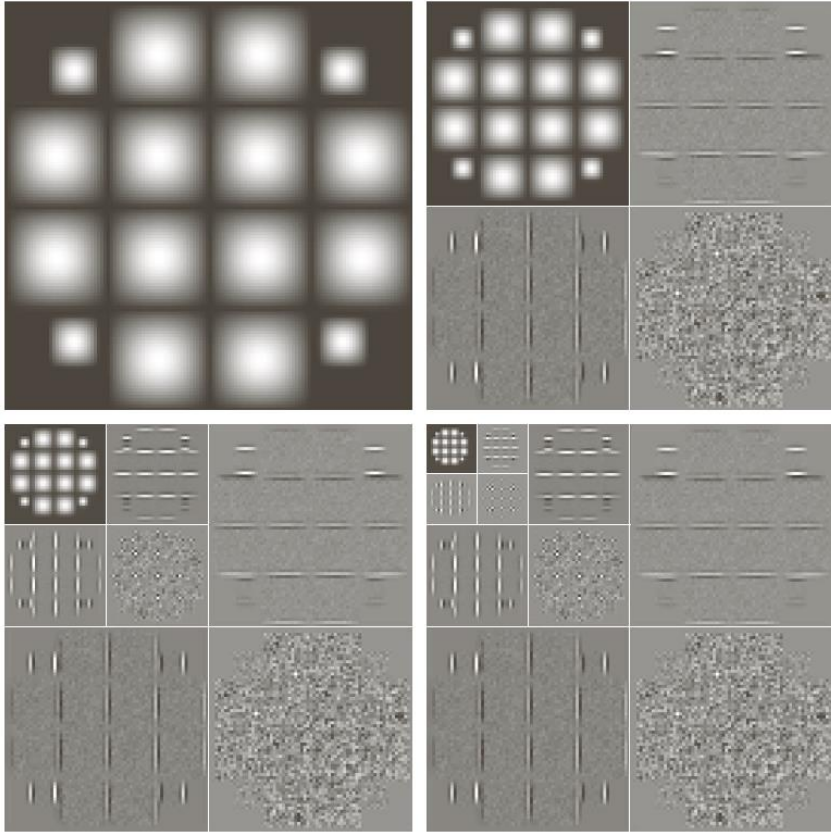


### 综合滤波器组

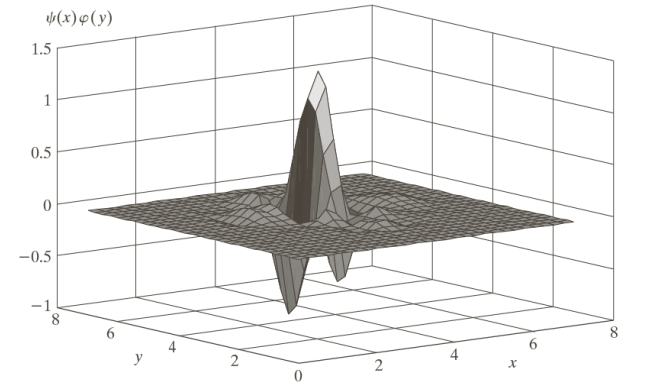
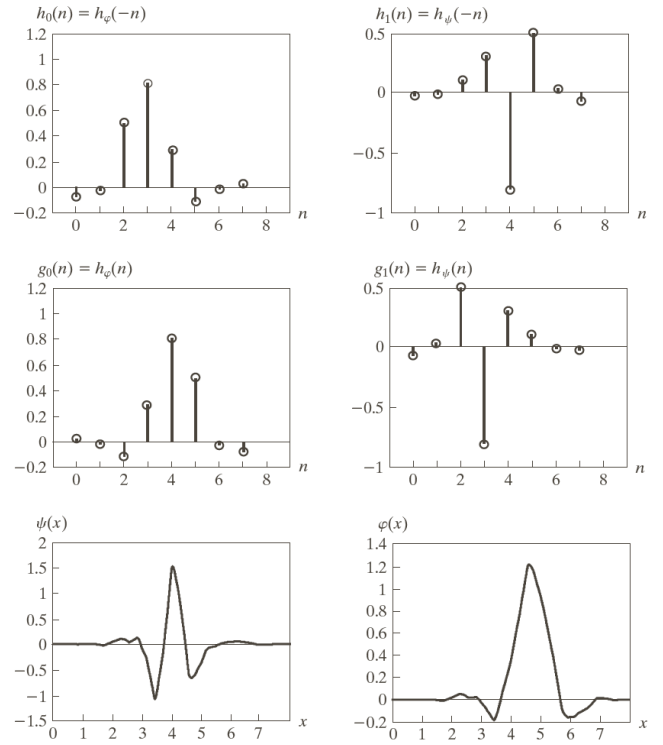


### 分解结果

# 7.6 二维小波变换

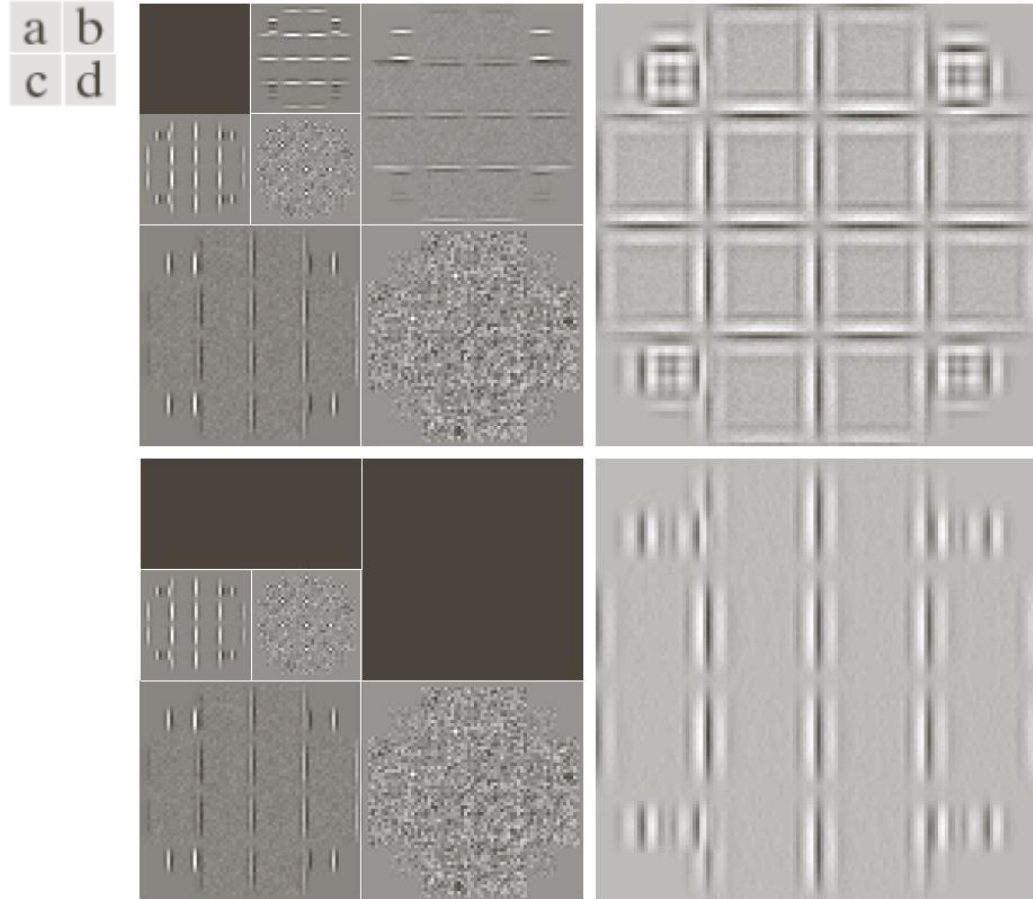


四阶对称小波：  
 (a)~(b) 重建滤波器；  
 (e) 一维小波；  
 (f) 一维尺度函数；  
 (g) 三个二维小波之一



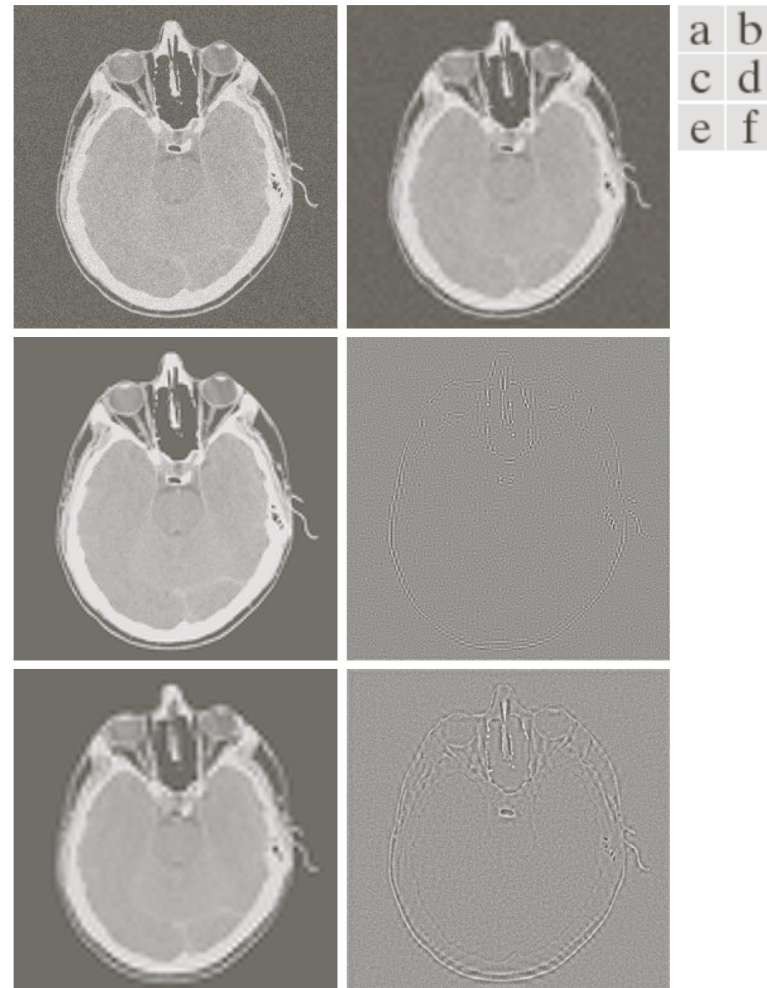
计算二维的三尺度FWT  
 (a) 原图像；(b) 一尺度FWT；  
 (c) 二尺度FWT；(d) 三尺度FWT

# 7.6 二维小波变换



针对**边缘检测**改进的DWT

(a) ~ (c) 删除所选系数的二尺度分解  
 (b) ~ (d) 相应的重建



为**噪声去除**修改DWT: (a) 人的头部噪声CT图像;  
 (b), (c)和(e)对细节系数进行阈值处理后的各种重建;  
 (d)和(f)在(c)和(e)重建过程中所删除的信息





# 第7章 小波和多分辨率处理

---

7.1 背景

7.2 多分辨率展开

7.3 一维小波变换

7.4 快速小波变换

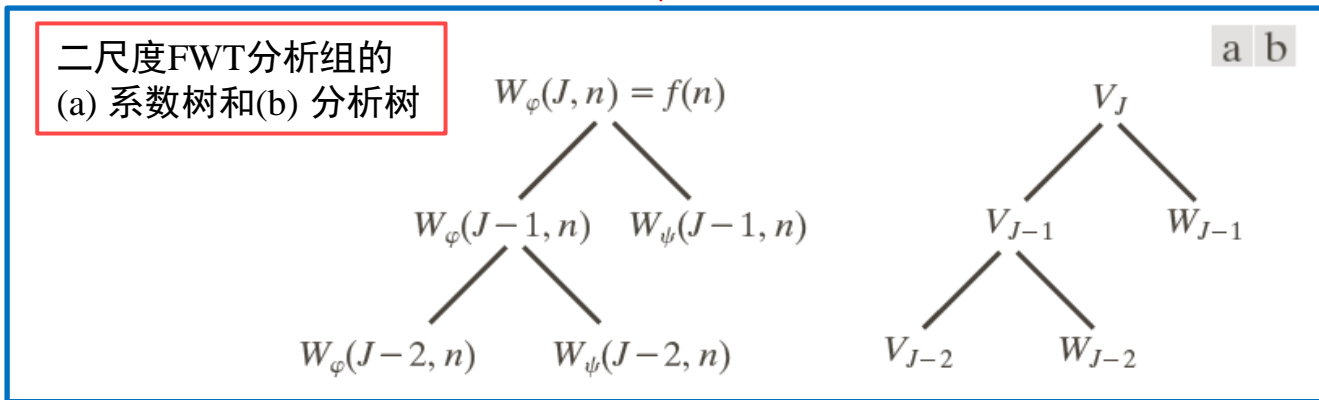
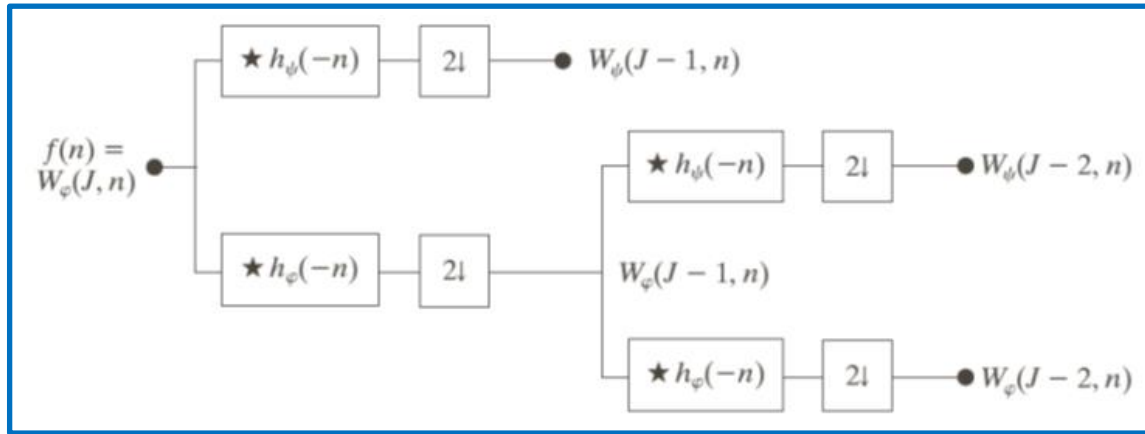
7.5 二维小波变换

7.6 **小波包**

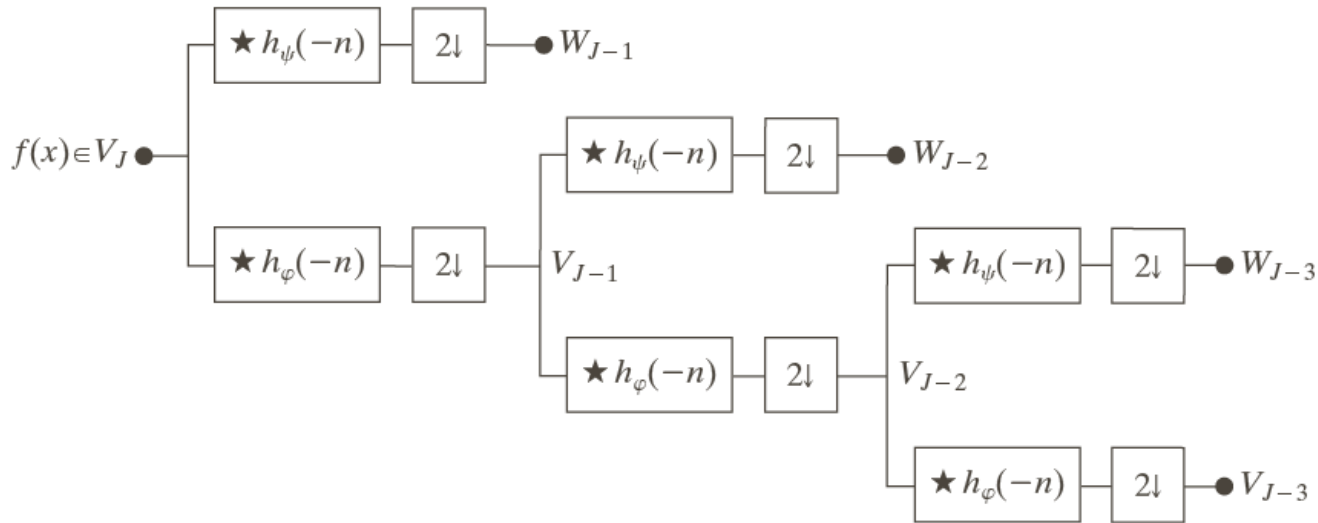
# 7.6 小波包

## □ 将小波分解想象为一个二叉树

- 根节点被赋予最高尺度的近似，叶节点继承变换的近似和细节

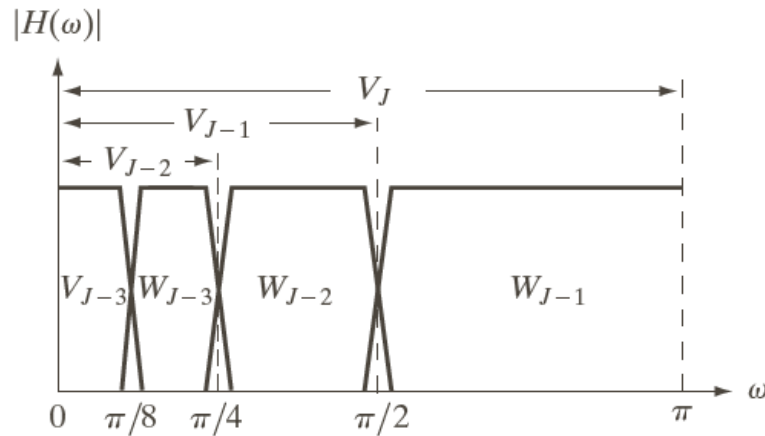
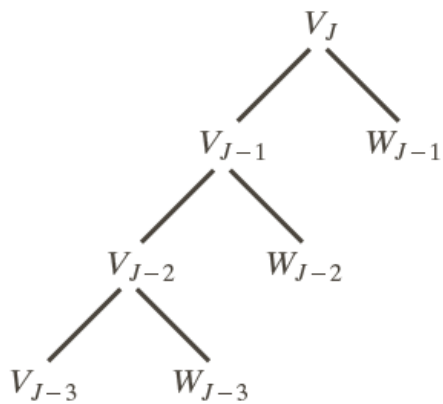


# 7.6 小波包



a  
b c

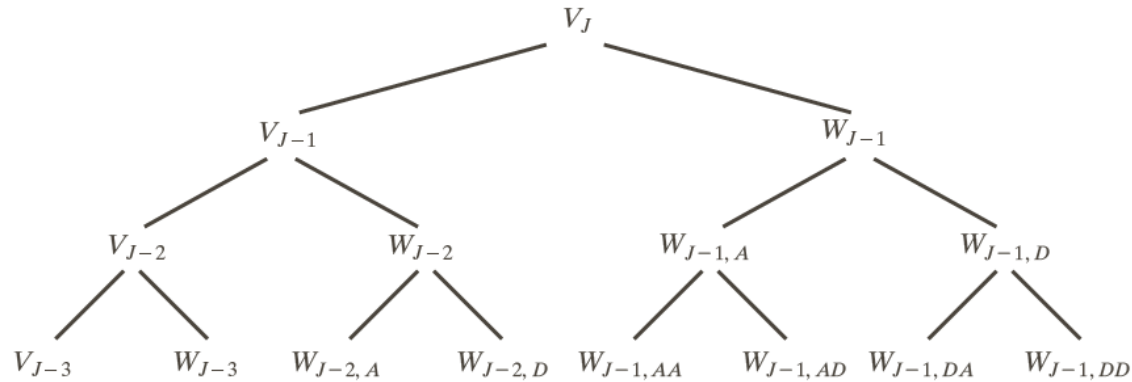
**FIGURE 7.30**  
A three-scale FWT filter bank:  
(a) block diagram;  
(b) decomposition space tree; and  
(c) spectrum splitting characteristics.



三尺度FWT分析组：  
(a) 方框图  
(b) 分解空间树  
(c) 频谱分离特性

# 7.6 小波包

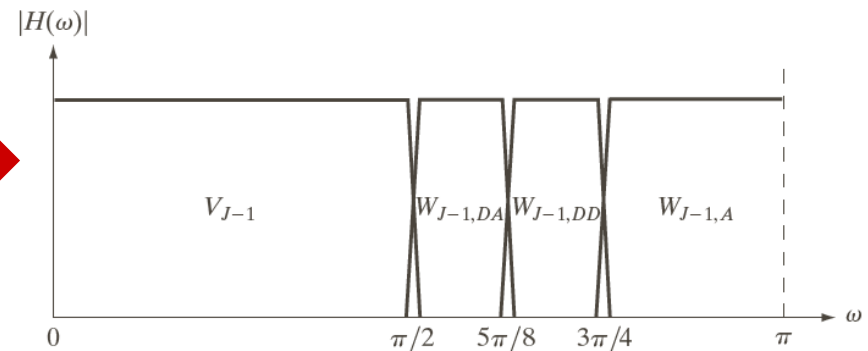
□ 在 $W_{J-1}$ 上附加近似滤波和细节滤波



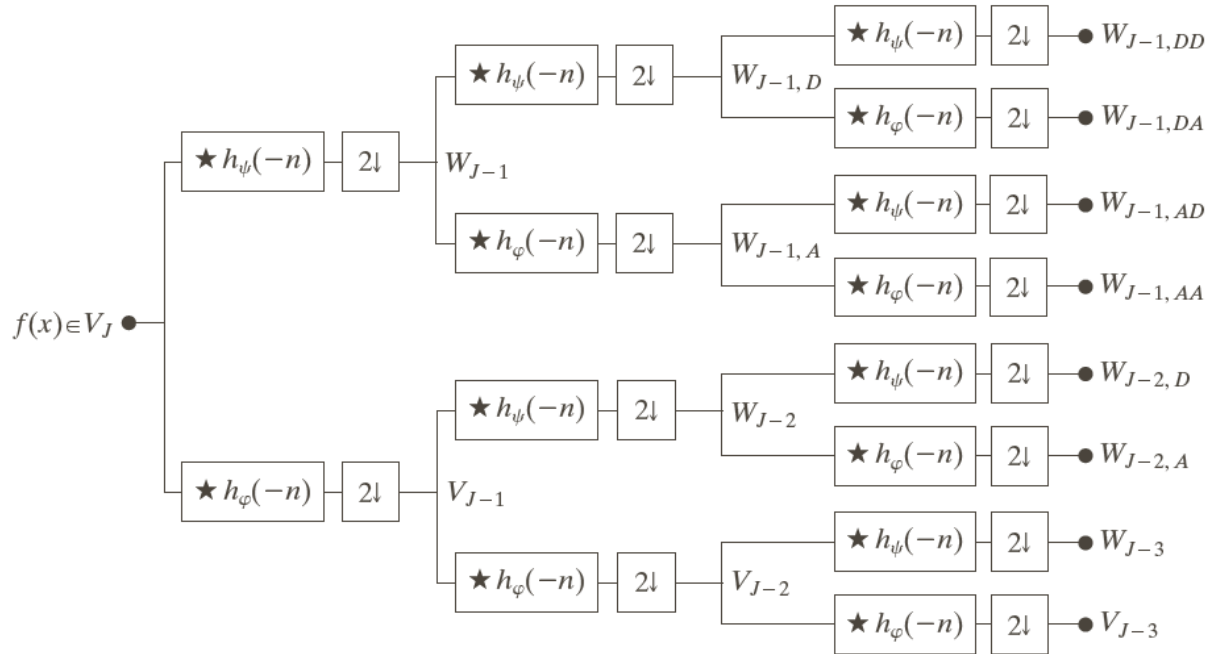
□ 上图中的小波包支持26中不同的分解，例如

$$V_J = V_{J-3} \oplus W_{J-3} \oplus W_{J-2,A} \oplus W_{J-2,D} \oplus W_{J-1,AA} \oplus W_{J-1,AD} \oplus W_{J-1,DA} \oplus W_{J-1,DD}$$

$$V_J = V_{J-1} \oplus W_{J-1,A} \oplus W_{J-1,DA} \oplus W_{J-1,DD} \quad \rightarrow$$

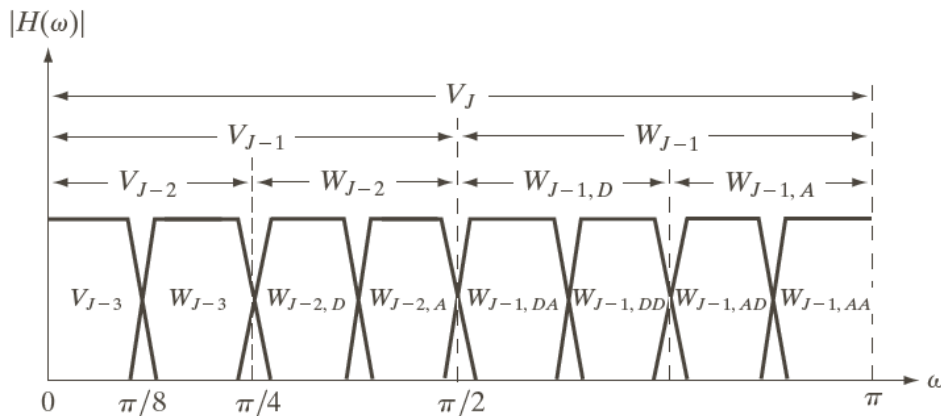


# 7.6 小波包



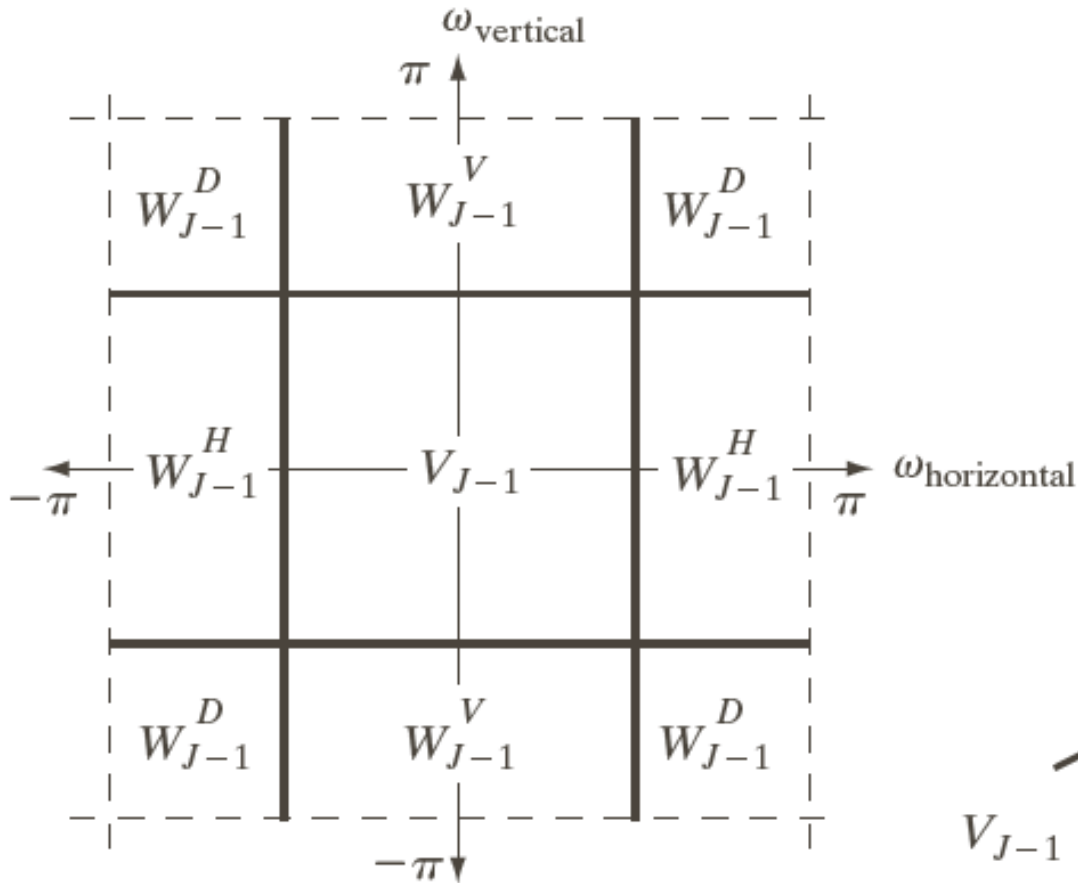
a  
b

**FIGURE 7.32**  
The (a) filter bank and (b) spectrum splitting characteristics of a three-scale full wavelet packet analysis tree.



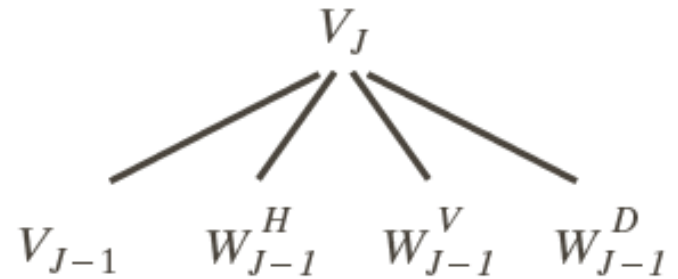
三尺度完全小波包分析树  
(a) 滤波器组  
(b) 频谱分离特性

# 7.6 小波包



a b

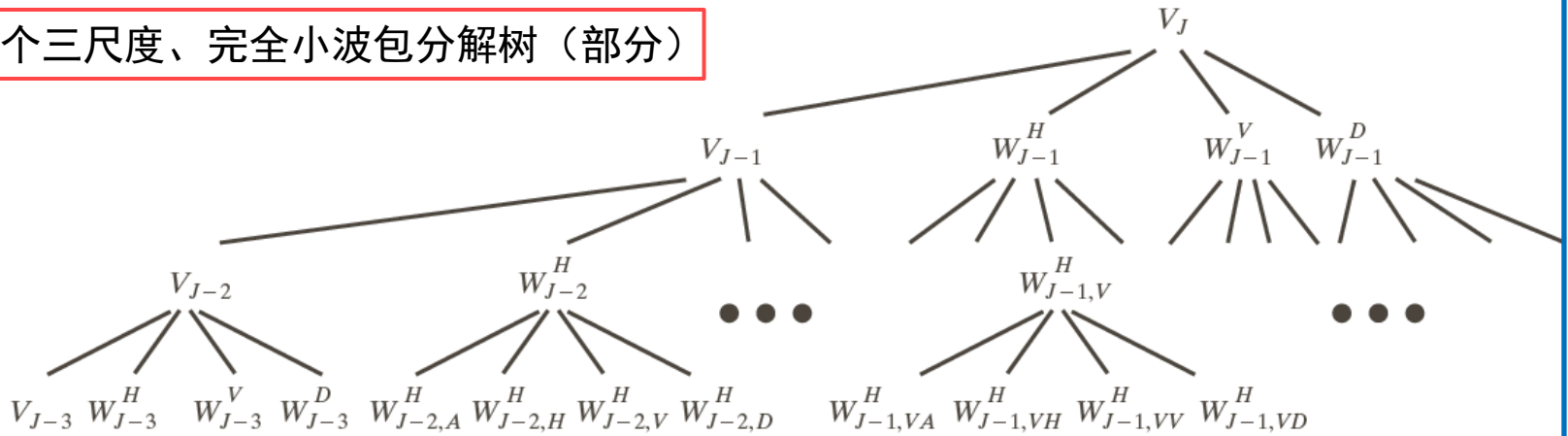
**FIGURE 7.34**  
The first decomposition of a two-dimensional FWT: (a) the spectrum and (b) the subspace analysis tree.



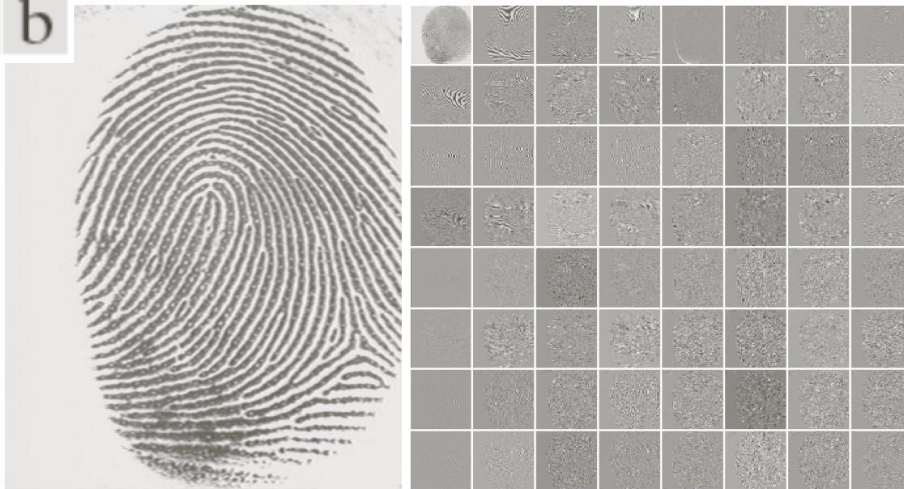
一个二维FWT的第一次分解 (a) 频谱 (b) 子空间分析树

# 7.6 小波包

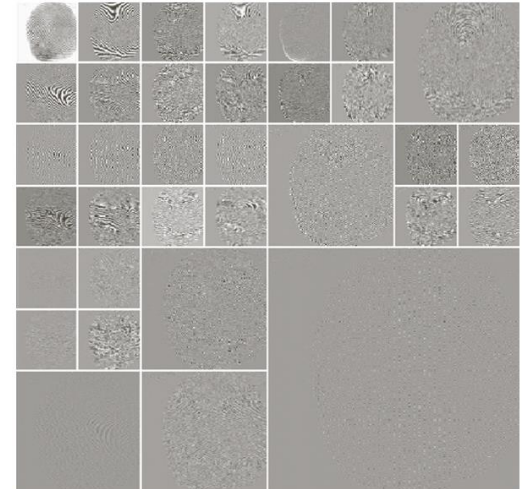
一个三尺度、完全小波包分解树（部分）



a b



(a) 一副扫描的指纹图像  
(b) 该图像的三尺度、全小波包分解



指纹的一种最佳小波分解



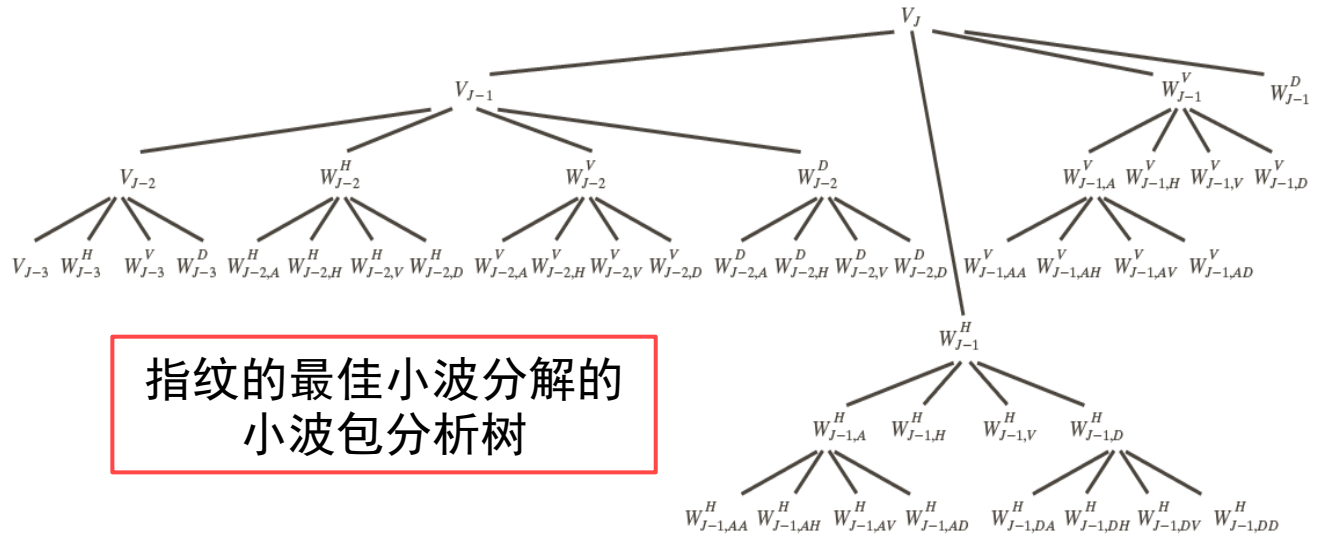
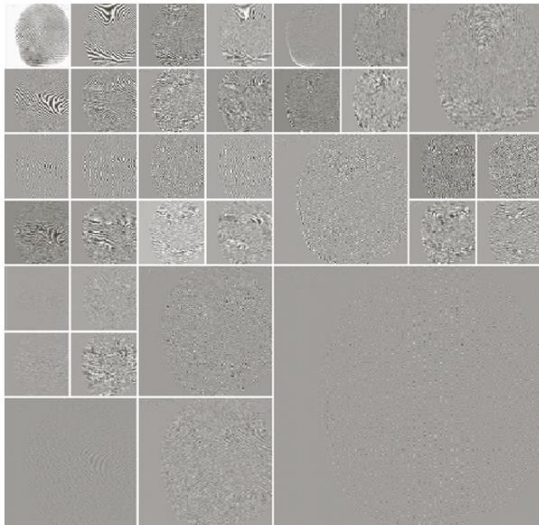
# 7.6 小波包： 指纹图像压缩

- 面向指纹图像压缩，定义代价函数，即

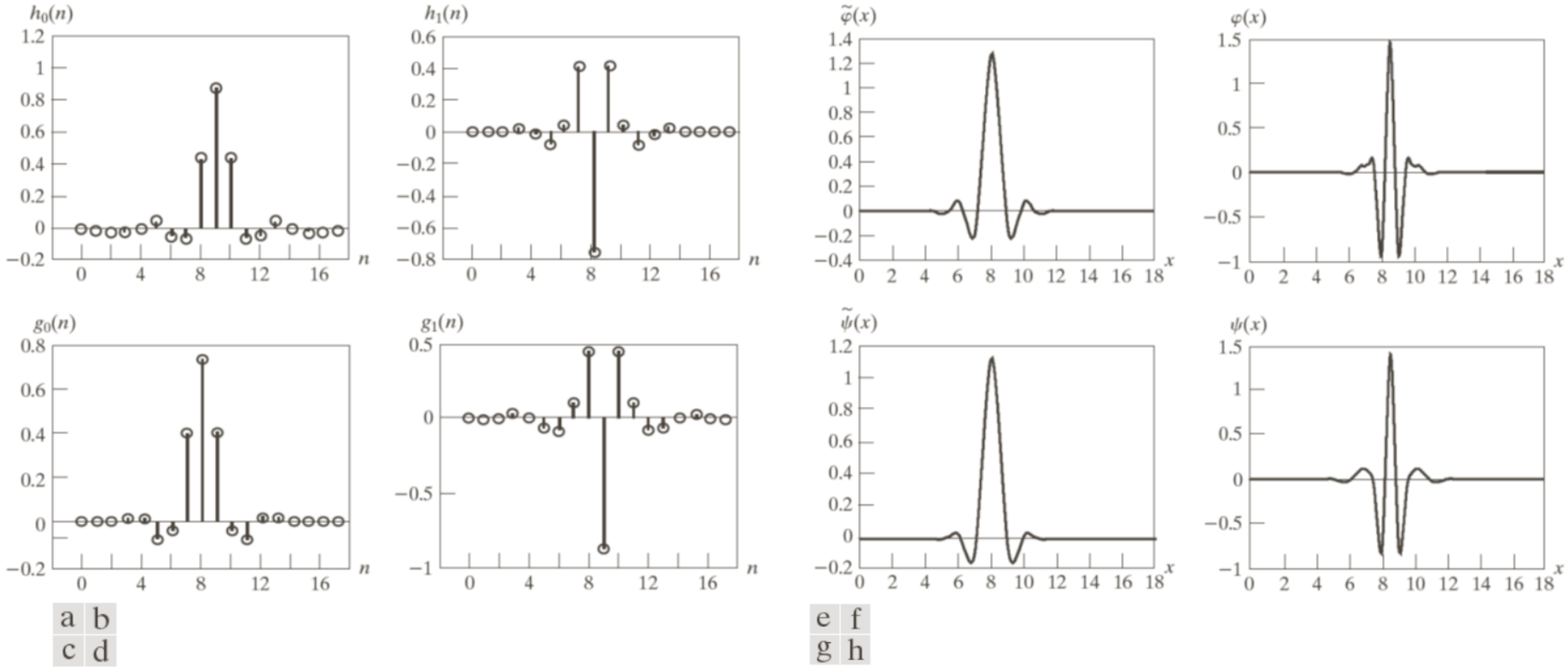
$$E(f) = \sum_{m,n} |f(m,n)|$$

- 父节点分解条件：子节点联合能量小于父节点能量，即

$$E_A + E_A + E_A + E_A < E_P$$



# 7.6 小波包



Cohen Daubechies Feauveau 双正交小波族成员：  
 (a)和(b)分解滤波器系数；(c)和(d)重建滤波器系数；  
 (e)~(h)双小波和尺度函数。



# 第8章 图像压缩

- 图像压缩是一种减少描绘一幅图像所需数据的技术和科学，它是数字图像处理领域最有用、商业上最成功的技术之一
  - 压缩的本质：用尽可能少的数据表达尽可能多的信息
  - 图像压缩必要性实例
    - ✓ 一两小时的标清电影，分辨率720×480，全彩色（24bit/pixel）
    - ✓ 如果不压缩，存储要224GB！
  
- Web网图像和分辨率数字摄像机图片要进行压缩，以节省存储空间和减少传输时间。同时，图像压缩在视频会议、遥感、文本和医学图像处理领域扮演重要角色



# 第8章 图像压缩

## □ 8.1 基础知识

- 编码冗余、空 冗余和时 冗余、不相关的信息 、图像信息的度 、保真度准则、图像压缩模型、图像格式容器和压缩标准

## □ 8.2 一些基本的压缩方法

- 夫曼编码
- 算术编码
- LZW编码
- 行程编码
- 基于符号的编码
- 比特平面 编码
- 块变换编码
- 预测编码
- 小波编码

## □ 8.3 数字图像水印

# 8.1 基础知识

## □ 相对数据冗余 $R$

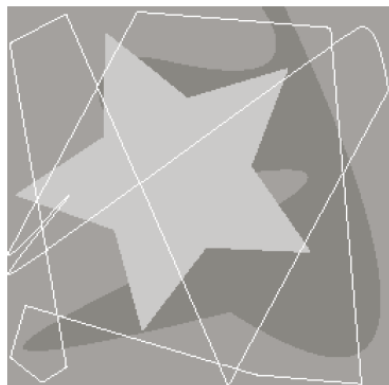
$$R = 1 - \frac{1}{C} = \frac{b - b'}{b'}$$

压缩率  $C = b/b'$

## □ 三种主要类型的数据冗余：

- 编码冗余：统计冗余
- 空 和 时 冗余
- 不相关的信息：感知冗余

编码冗余



空 冗余



无关信息



计算机产生的  $256 \times 256 \times 8$  比特图像



# 编码冗余

- $r_k$  表示一副  $M \times N$  图像的灰度,  $r_k$  出现概率  $p_r(r_k)$

$$p_r(r_k) = \frac{n_k}{MN}, k = 0, 1, 2, \dots, L - 1$$

$$L_{avg} = \sum_{k=0}^{L-1} l(r_k) p_r(r_k)$$

- 变 编码:  $L_{avg}=1.81$  bit
- 定 编码:  $L_{avg}=8$  bit
- $C=8/1.81=4.42$   $R=1-1/C=0.774$

$r_k$	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_{87} = 87$	0.25	01010111	8	01	2
$r_{128} = 128$	0.47	10000000	8	1	1
$r_{186} = 186$	0.25	11000100	8	000	3
$r_{255} = 255$	0.03	11111111	8	001	3
$r_k$ for $k \neq 87, 128, 186, 255$	0	—	8	—	0

**TABLE 8.1**  
Example of variable-length coding.

# 空 冗余和时 冗余

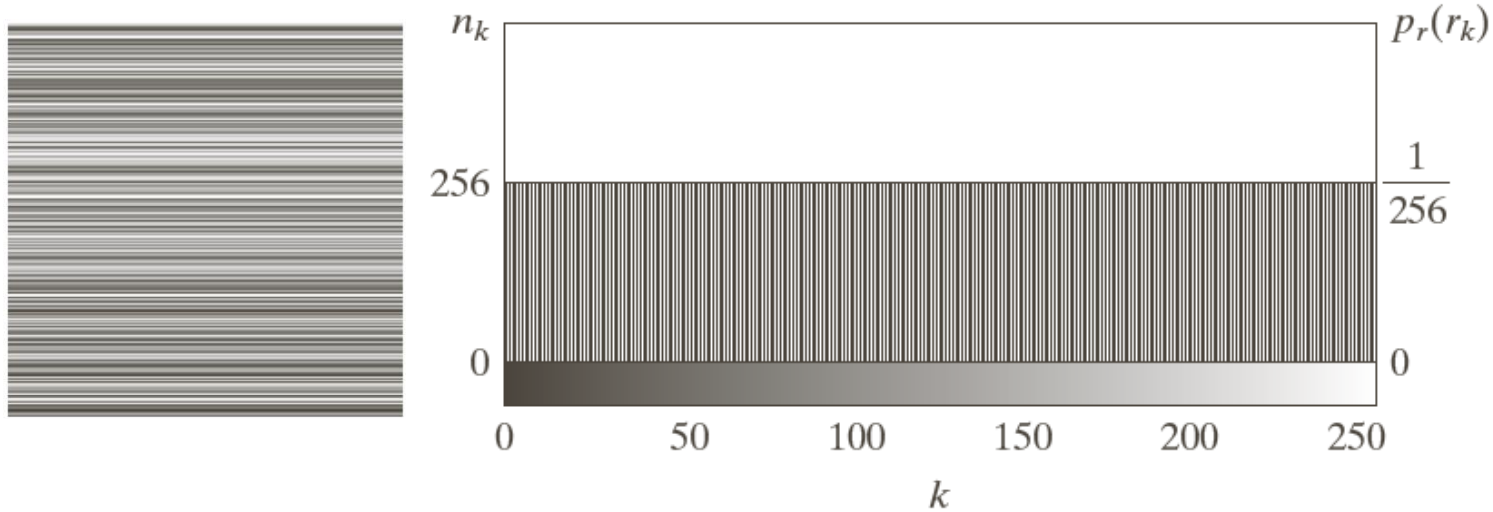


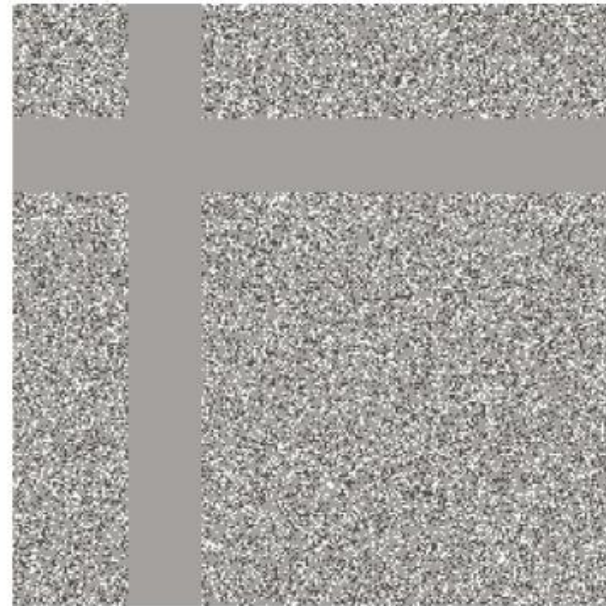
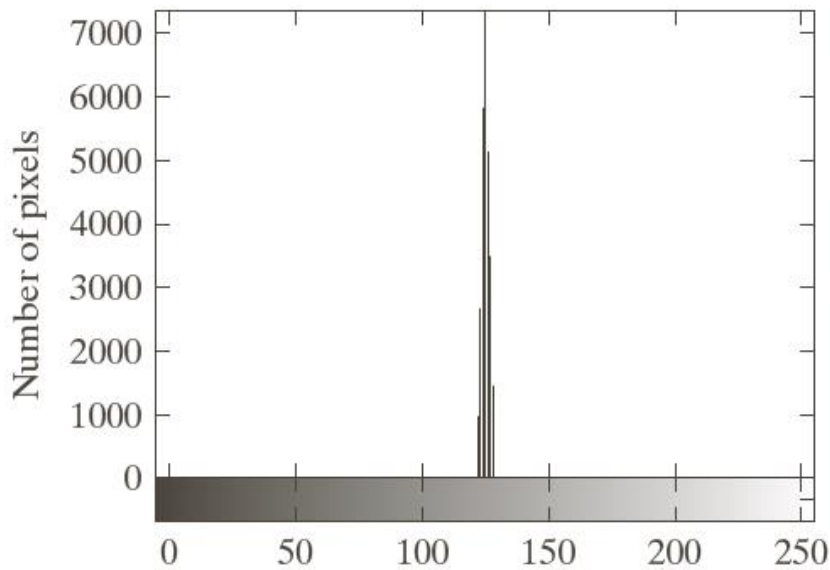
FIGURE 8.2 The intensity histogram of the image in Fig. 8.1(b).

- 空 相关（x和y方向）：多数像素可根据相 像素灰度进行合理使测
  - 行程编码：指定一个新的灰度开始点，以及该灰度持续像素的数
- 时 相关：当该图像是视 序列的一 分时



# 不相关信息

- 数字图像压缩中，被人类视觉系统忽略的信息或与图像与其的应用无关的信息 要被删



a b  
**FIGURE 8.3**  
(a) Histogram of the image in Fig. 8.1(c) and (b) a histogram equalized version of the image.



# 图像信息的度

- 概率为 $P(E)$ 的 机事件包含的信息

$$I(E) = -\log P(E)$$

- 信源的平均信息称为该信源的熵

$$H = -\sum_{j=1}^J P(a_j) \log P(a_j)$$

- 灰度信源的熵

$$\tilde{H} = \sum_{k=0}^{L-1} P_r(r_k) \log_2 p_r(r_k)$$

- 农第一定理（无噪声编码定理）

$$\lim_{n \rightarrow \infty} \left[ \frac{L_{avg,n}}{n} \right] = H$$



# 保真度准则：度 压缩质

- $f(x, y)$ 和 $\hat{f}(x, y)$ 之 误差 $e(x, y)$

$$e(x, y) = \hat{f}(x, y) - f(x, y)$$

- 平均误差的平方根

$$e_{rms} = \left[ \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]^2 \right]^{\frac{1}{2}}$$

- 均方信噪比

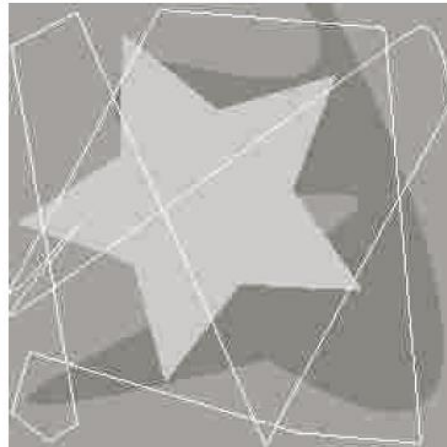
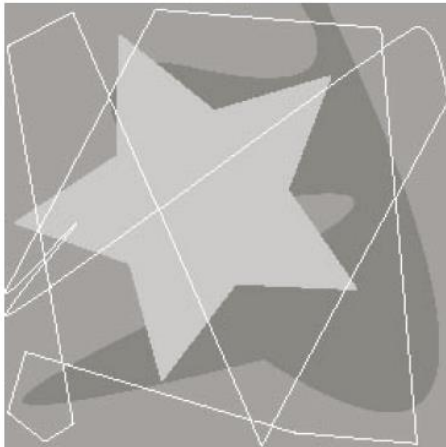
$$\text{SNR}_{ms} = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \hat{f}(x, y)^2}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]^2}$$

# 保真度准则

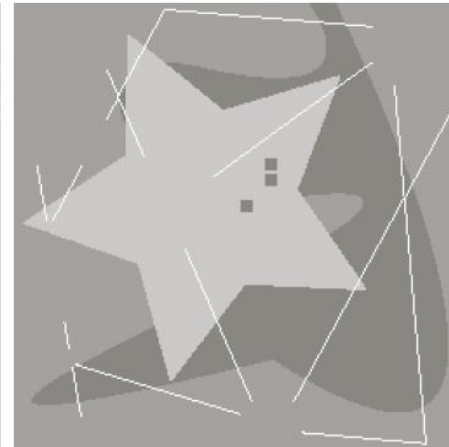
Value	Rating	Description
1	Excellent	An image of extremely high quality, as good as you could desire.
2	Fine	An image of high quality, providing enjoyable viewing. Interference is not objectionable.
3	Passable	An image of acceptable quality. Interference is not objectionable.
4	Marginal	An image of poor quality; you wish you could improve it. Interference is somewhat objectionable.
5	Inferior	A very poor image, but you could watch it. Objectionable interference is definitely present.
6	Unusable	An image so bad that you could not watch it.

电视 置研究组  
组织的等级尺度

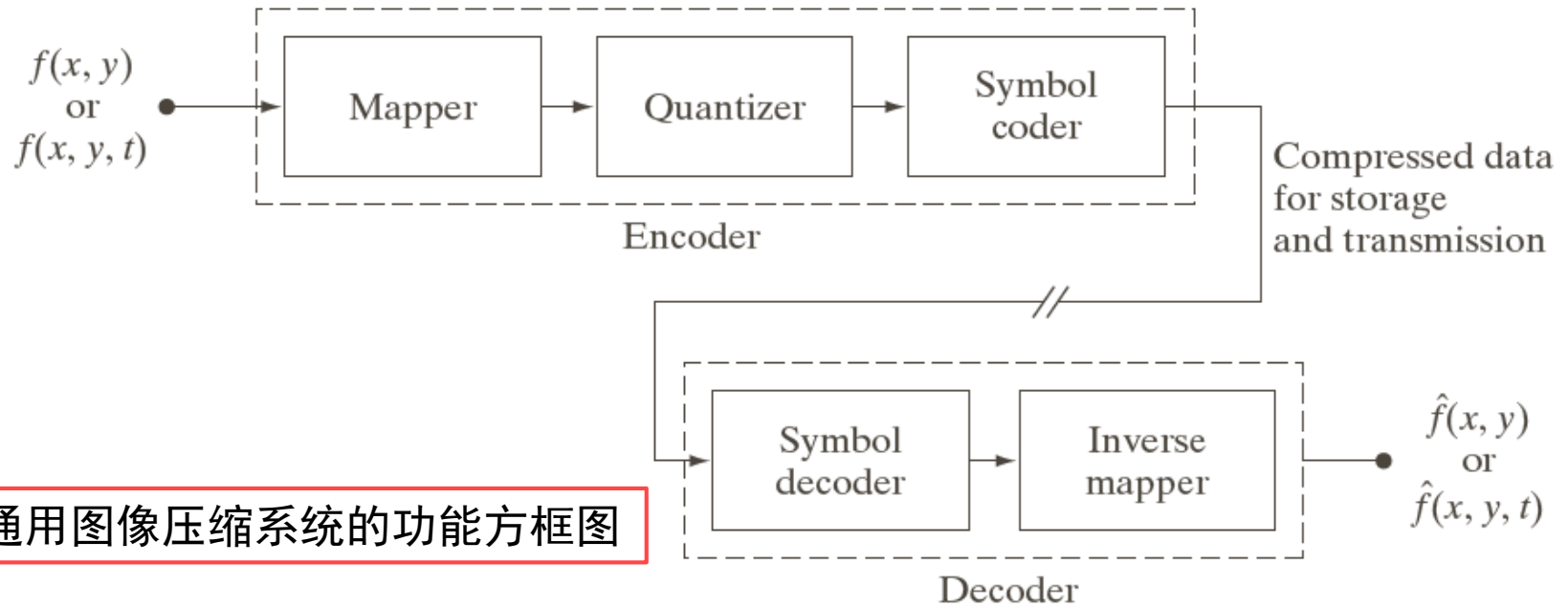
压缩 建的典型图像



人工生成的图像



# 图像压缩模型



一个通用图像压缩系统的功能方框图

## □ 编码或压缩过程

- 映射器： 低空 和时 冗余
- 量化器： 根据使设保真度准则，排 压缩表示的无关信息
- 符号编码器： 生成定 编码或变 编码，最小化编码冗余

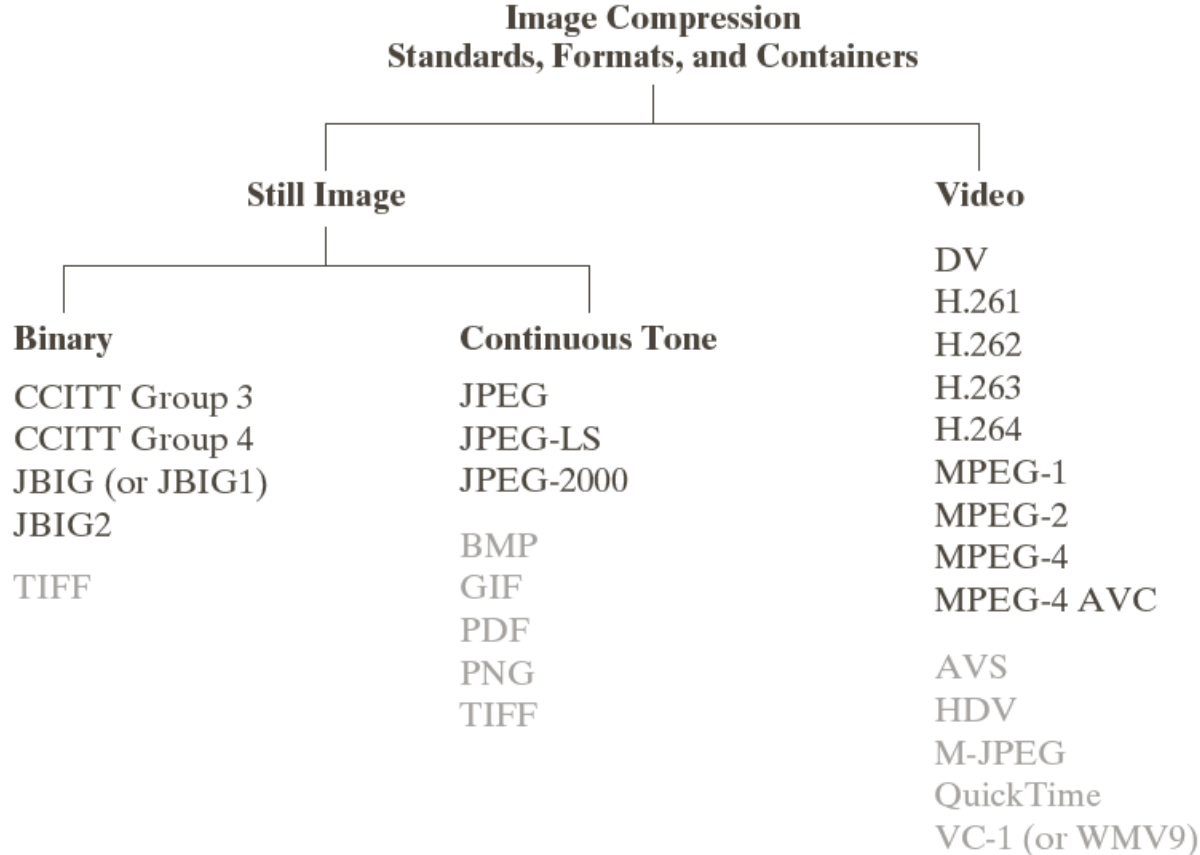
## □ 解码或解压缩过程

- 符号解码器和反映射器



# 图像格式、容器和压缩标准

- 文件格式：相当于 层的信息，定义了数据类型、压缩方法等
- 容器：相当于一个存放压缩以后的数据的东西
- 压缩方法：处在更底层，规定了数据的具体压缩的过程





## 8.2 一些基本的压缩方法

- 夫曼编码
- 算数编码
- LZW编码
- 行程编码
- 基于符号的编码
- 比特平面 编码
- 块变换编码
- 预测编码
- 小波编码





# 夫曼编码

夫曼编码步 ：

(1) 概率排序，缩减和简化信源符号数

Original source		Source reduction			
Symbol	Probability	1	2	3	4
$a_2$	0.4	0.4	0.4	0.4	0.6
$a_6$	0.3	0.3	0.3	0.3	0.4
$a_1$	0.1	0.1	0.2	0.3	
$a_4$	0.1	0.1	0.1		
$a_3$	0.06	0.1			
$a_5$	0.04				



# 夫曼编码

夫曼编码步 ：

(2) 对每个简化后的信源进行编码，从最小信源开始，直到 历原始信源

Original source			Source reduction							
Symbol	Probability	Code	1		2		3		4	
$a_2$	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
$a_6$	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
$a_1$	0.1	011	0.1	011	0.2	010	0.3	01		
$a_4$	0.1	0100	0.1	0100	0.1	011				
$a_3$	0.06	01010	0.1	0101						
$a_5$	0.04	01011								

平均 度

$$L_{avg} = 0.4 \times 1 + 0.3 \times 2 + 0.1 \times 3 + 0.1 \times 4 + 0.06 \times 5 + 0.04 \times 5 = 2.2$$



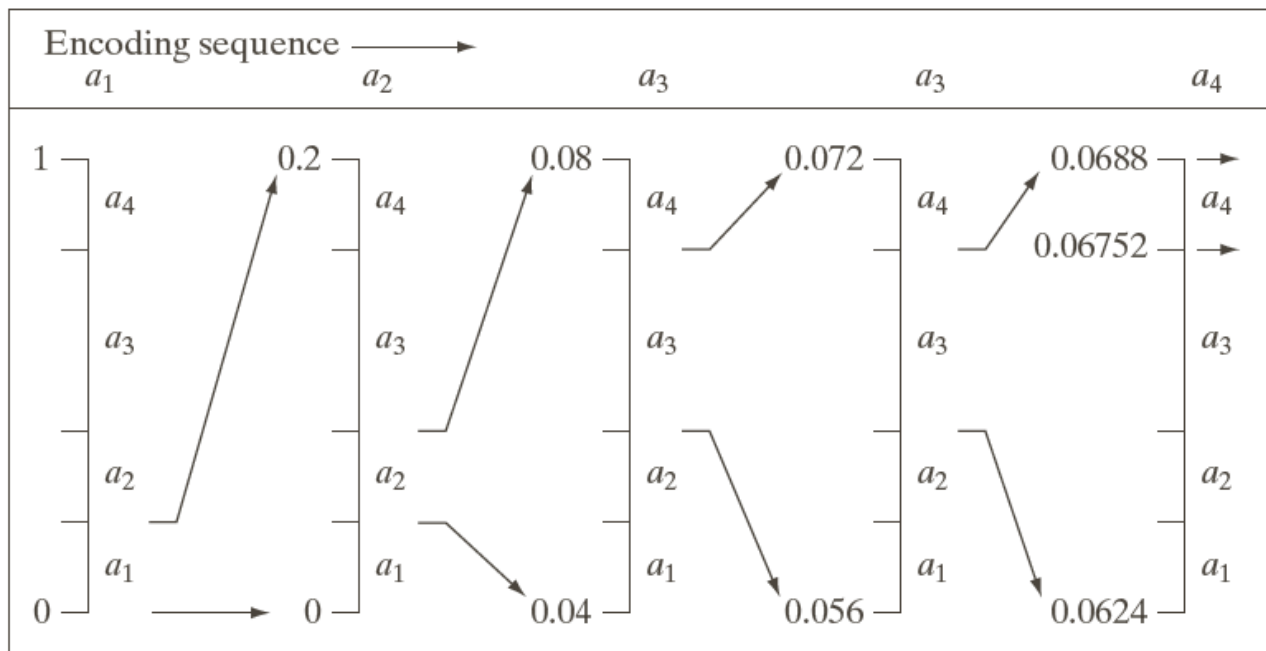
# 算术编码

- 将一个信源符号序列编码为一个介于0到1之间的实数
- 示例：对来自一个4符号信源的5符号序列进行编码

Source Symbol	Probability	Initial Subinterval
$a_1$	0.2	[0.0, 0.2)
$a_2$	0.2	[0.2, 0.4)
$a_3$	0.4	[0.4, 0.8)
$a_4$	0.2	[0.8, 1.0)

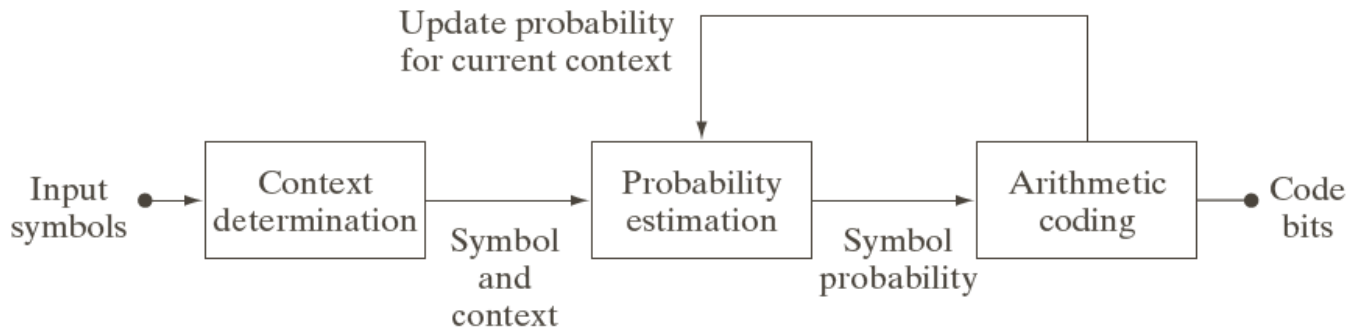
信源

算术编码过程



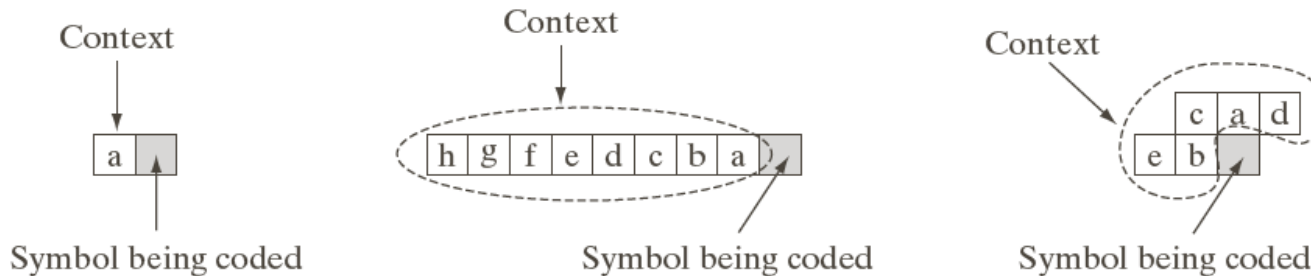
# 算术编码

## □ 自适应上下文相关的概率估计



a  
b c d

**FIGURE 8.13**  
(a) An adaptive, context-based arithmetic coding approach (often used for binary source symbols). (b)–(d) Three possible context models.



(a) 自适应上下文相关的算术编码方法（通常用于二值信源符号）  
(b)~(d) 三种可能的上下文模型



# LZW编码

- 无误差压缩图像中空 冗余的方法
- 将定 码字分 各变 信源符号序列

39 39 126 126  
 39 39 126 126  
 39 39 126 126  
 39 39 126 126

Dictionary Location	Entry
0	0
1	1
⋮	⋮
255	255
256	—
⋮	⋮
511	—

Currently Recognized Sequence	Pixel Being Processed	Encoded Output	Dictionary Location (Code Word)	Dictionary Entry
	39			
39	39	39	256	39-39
39	126	39	257	39-126
126	126	126	258	126-126
126	39	126	259	126-39
39	39			
39-39	126	256	260	39-39-126
126	126			
126-126	39	258	261	126-126-39
39	39			
39-39	126			
39-39-126	126	260	262	39-39-126-126
126	39			
126-39	39	259	263	126-39-39
39	126			
39-126	126	257	264	39-126-126
126	126	126		

LZW编码示例



# 行程编码

- 处理行（或列） **复灰度**的图像
- 行程对指定**一个新灰度的开始**和**具有该灰度的连续像素的数**
- 主要适用于压缩二值图像
- BMP文件格式的RLE是行程编码的一种形式
- 两种古老且应用最广的二值图像压缩标准CCITT Group3和4 用了行程编码



# 行程编码

## □ BMP文件格式的RLE

- 图像数据使用两种不同的模式来表示：编码和绝对值
- **编码**模式用两个字节RLE表示，第一个字节规定连续像素的数目，它有包含在第二个字节中的彩色索引
- **绝对值**模式中，第一个字节是0，第二个字节是4个可能的条件之一。
  - ✓ 当第二个字节是0或1时，就到达一行或一幅图像末尾
  - ✓ 当第二个字节是2，则下两个字节包含相对图像中一个新空 位置的无符号水平偏移或垂直偏移
  - ✓ 当第二个字节在3和255之 ，则它指定了 压缩像素的数

Second Byte Value	Condition
0	End of line
1	End of image
2	Move to a new position
3-255	Specify pixels individually

**TABLE 8.8**  
BMP absolute coding mode options. In this mode, the first byte of the BMP pair is 0.





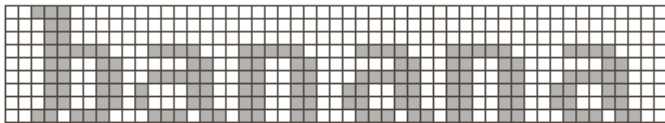
# 行程编码

## □ CCITT垂直模式编码示例

- CCITT Group3和4标准 用的二维压缩方法是逐行方法
- 对每个 到白或白到 的行程转换的位置，根据位于当前编码行的参考元素a0的位置来编码
- 已编码的前一行称为参考行，每幅新图像的第一行的参考行为一个虚构的白色行
- 所用的二维编码技术称为相对元素地址指派（READ）编码

# 基于符号的编码

- 图像被表示为多幅 繁发生的子图像的 合，称为符号
- 符号字典：符号的 合
- 图像以一个三元组  $\{(x_1, y_1, t_1), \{(x_2, y_2, t_2)\}, \dots\}$  的 合来编码，  $(x_i, y_i)$  规定图像中某个符号的位置，  $t_i$  表示该子图在字典中的地址

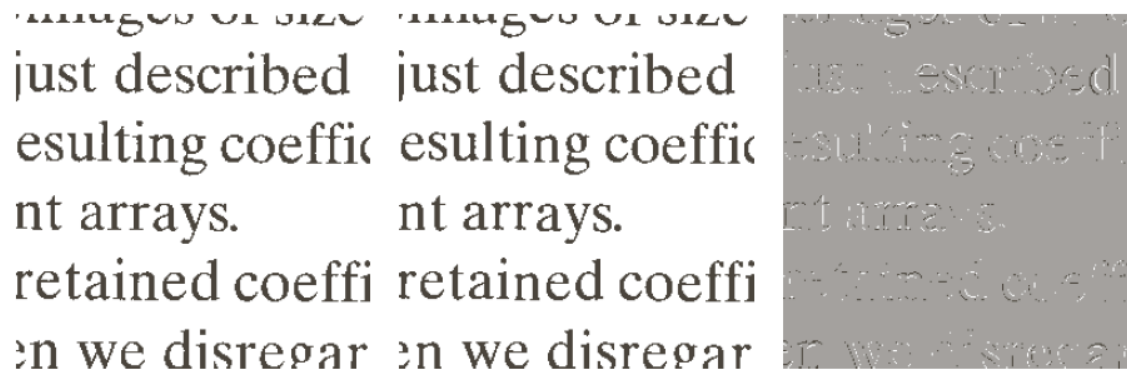


Token	Symbol	Triplet
0		(0, 2, 0) (3, 10, 1) (3, 18, 2)
1		(3, 26, 1) (3, 34, 2) (3, 42, 1)
2		

(a) 两灰度级文档 (b) 符号字典 (c) 文档中用于定位的三元组

# 基于符号的编码

- JBIG2是 对二级图像压缩的国 标准
- JBIG2包括无损压缩和感知无损压缩
- 感知无损指忽略字典位图和图像中相应字符特例之 的差别，但是不影响可读性



(a)无损压缩和 建 (b)感知无损压缩 (c)两种压缩程度的差值



# 比特平面 编码

□ 将一幅多级图像分解为一系列二值图像，然后使用二值压缩方法进行编码

□ 两种分解方法

■ 基2多 式

$$a_{m-1}2^{m-1} + a_{m-2}2^{m-2} + \dots + a_12^1 + a_02^0$$

■ m比特格 码表示图像

✓ 将m比特分解为格 码  $g_{m-1} \dots g_2 g_1 g_0$

$$g_i = a_i \oplus a_{i+1}, 0 \leq i \leq m-2$$

$$g_{m-1} = a_{m-1}$$

■ 特点：连续码字只有一个比特位不同

✓ 例如，127和128的格 码分别是01000000和11000000

# 比特平 编码

(a) 一幅256比特单色图像  
(b)-(h) 图(a)中图像的4个最 有效二进制编码和格 编码比特平





# 比特平 编码

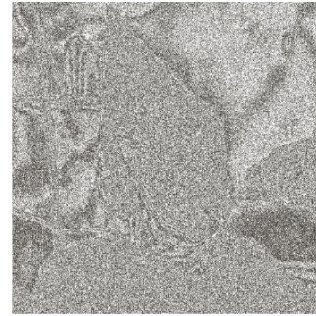
(a)-(h) 图(a)中图像的4个最低有效二进制编码和格 编码比特平



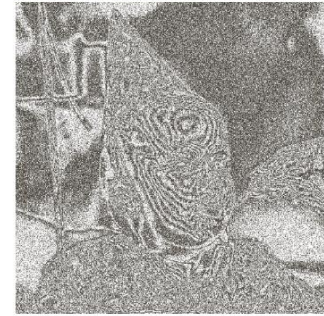
$a_3$



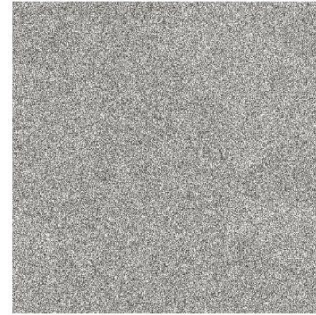
$g_3$



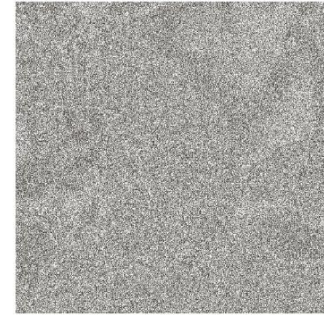
$a_2$



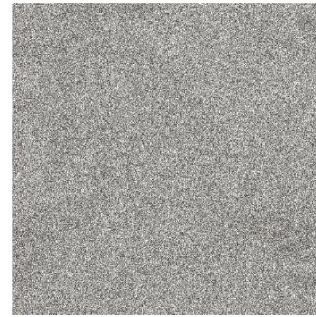
$g_2$



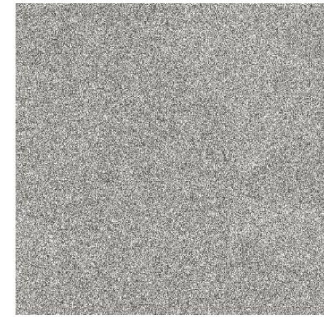
$a_1$



$g_1$



$a_0$



$g_0$



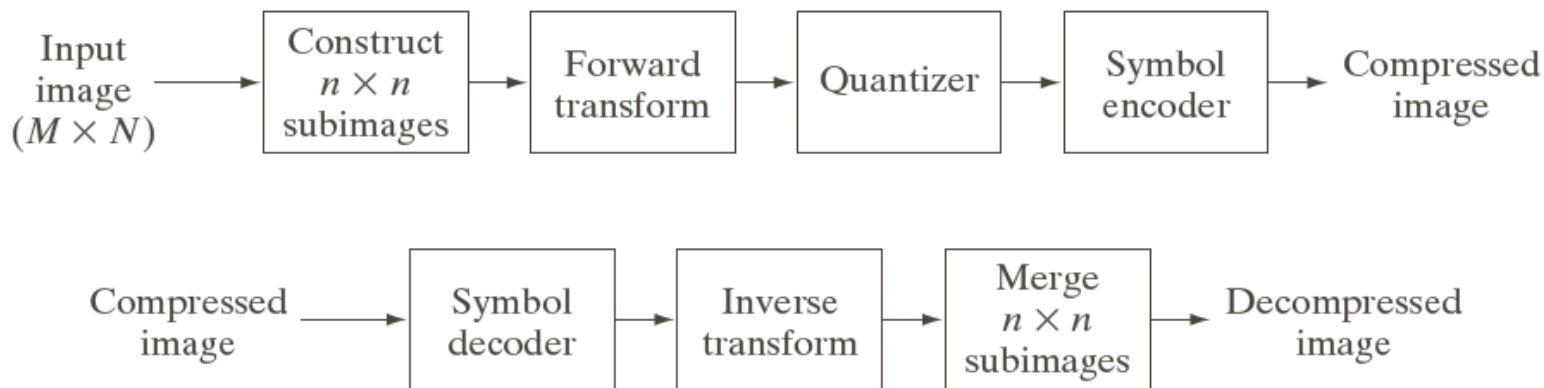
# 比特平 编码

Coefficient <i>m</i>	Binary Code (PDF bits)	Gray Code (PDF bits)	Compression Ratio
7	6,999	6,999	1.00
6	12,791	11,024	1.16
5	40,104	36,914	1.09
4	55,911	47,415	1.18
3	78,915	67,787	1.16
2	101,535	92,630	1.10
1	107,909	105,286	1.03
0	99,753	107,909	0.92

图像的二进制编码和格 编码比特平的  
的JBIG2无损编码结果



# 块变换编码



块变换编码系统

- 在给定的应用中，特定变换的选择取决于可容忍的 建误差大小和可用的计算资源
- 压缩是在对变换系数的 化期 实现的

正变换 
$$T(u, v) = \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} g(x, y) r(x, y, u, v)$$

反变换 
$$g(x, y) = \sum_{u=0}^{n-1} \sum_{v=0}^{n-1} T(u, v) s(x, y, u, v)$$

# 块变换编码

## □ 傅 叶变换核

$$r(x, y, u, v) = e^{-j2\pi(ux+vy)/n}$$

$$s(x, y, u, v) = \frac{1}{n^2} e^{j2\pi(ux+vy)/n}$$

## □ Walsh-Hadamard变换 (WHT)

$$r(x, y, u, v) = s(x, y, u, v) = \frac{1}{n} (-1)^{\sum_{i=0}^{m-1} [b_i(x)p_i(u) + b_i(y)p_i(v)]}$$

$$n = 2^m$$

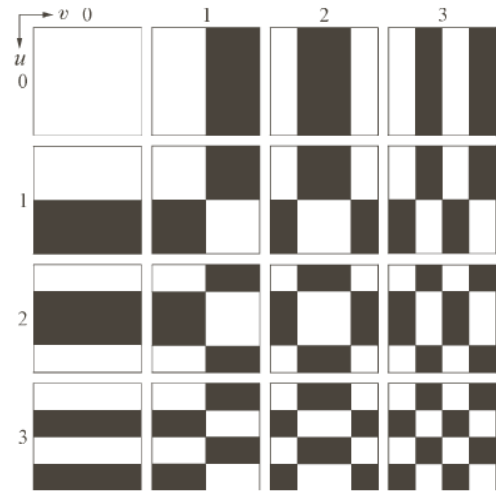
$b_k(z)$  是 $z$ 的二进制表示中的第 $k$ 位

$$p_0(u) = b_{m-1}(u)$$

$$p_1(u) = b_{m-1}(u) + b_{m-2}(u)$$

...

$$p_{m-1}(u) = b_1(u) + b_0(u)$$



$n=4$ 时的WHT的基函数



# 块变换编码

## □ 哈达玛变换的递推式

### □ $2^K \times 2^K$ 哈达玛递推式 :

$$H_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

### □ WHT的变换对:

$$\mathbf{F} = \mathbf{H} \mathbf{f} \mathbf{H}$$

$$\mathbf{f} = \mathbf{H} \mathbf{F} \mathbf{H}$$

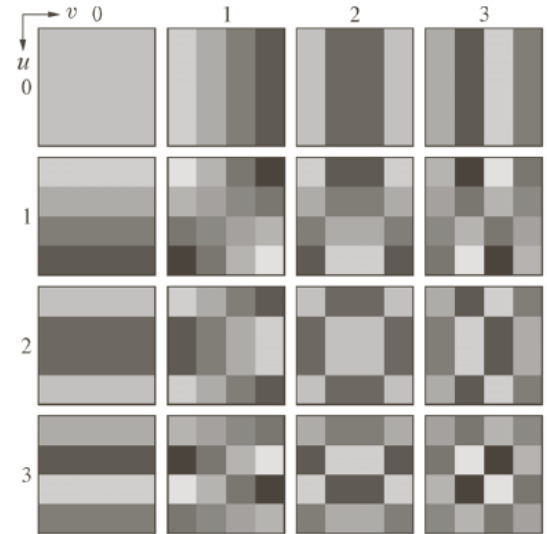
$$H_{2N} = \frac{1}{\sqrt{2}} \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix}$$

# 块变换编码

## □ 离散余弦变换 (DCT)

$$r(x, y, u, v) = s(x, y, u, v) = \alpha(u)\alpha(v) \cos \left[ \frac{(2x+1)u\pi}{2n} \right] \cos \left[ \frac{(2y+1)v\pi}{2n} \right]$$

$$\alpha(u) = \begin{cases} \sqrt{1/n}, u = 0 \\ \sqrt{2/n}, u = 1, 2, \dots, n-1 \end{cases}$$



n=4时的DCT的基函数

# 块变换编码

先分8\*8图像块，  
然后进行变换，  
对变换后最小的  
32个系数置0，  
利用剩下的32个  
大系数进行重构



重建误差：  
傅里叶：2.32；  
WHT：1.78；  
DCT：1.13

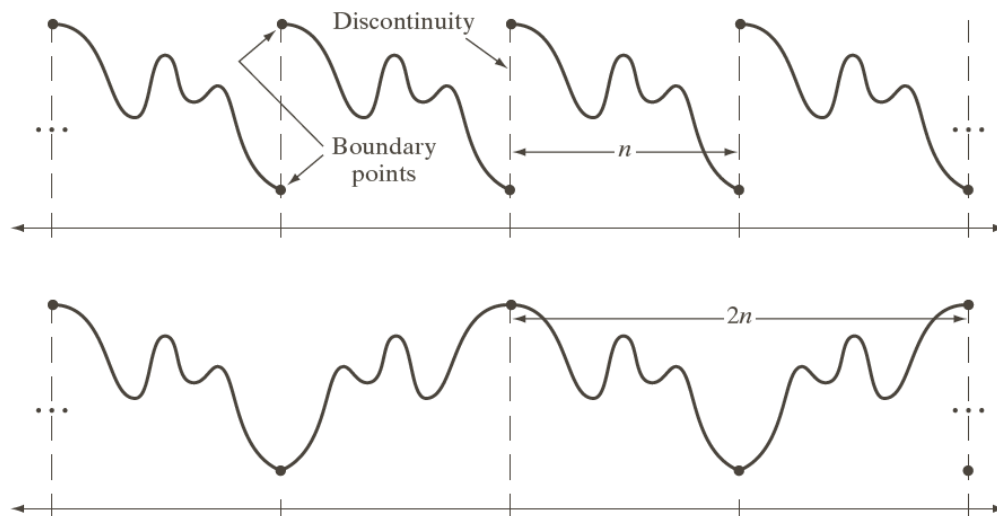
a	b	c
d	e	f

**FIGURE 8.24** Approximations of Fig. 8.9(a) using the (a) Fourier, (b) Walsh-Hadamard, and (c) cosine transforms, together with the corresponding scaled error images in (d)–(f).

# 块变换编码

## □ DCT的优点

- 用单片 成电路即可实现
- **可将最多信息装入最少系数中**
- 可使块效应最小化

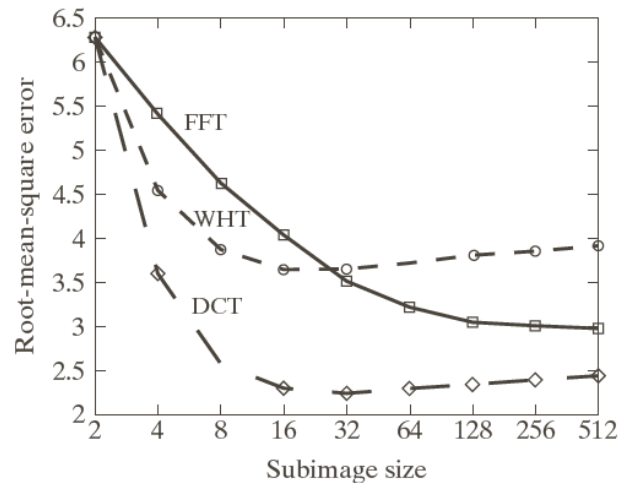


(a)一维DFT中 含的周期性 (b)一维DCT中 含的周期性

# 块变换编码

## □ 子图像尺寸的选择

- 子图像的尺寸是影响编码误差和计算复杂性的 要因素
- 一般压缩效率和计算复杂度 子图像尺寸的增加而增加
- 最常用的子图像尺寸为 $8 \times 8$ 和 $16 \times 16$

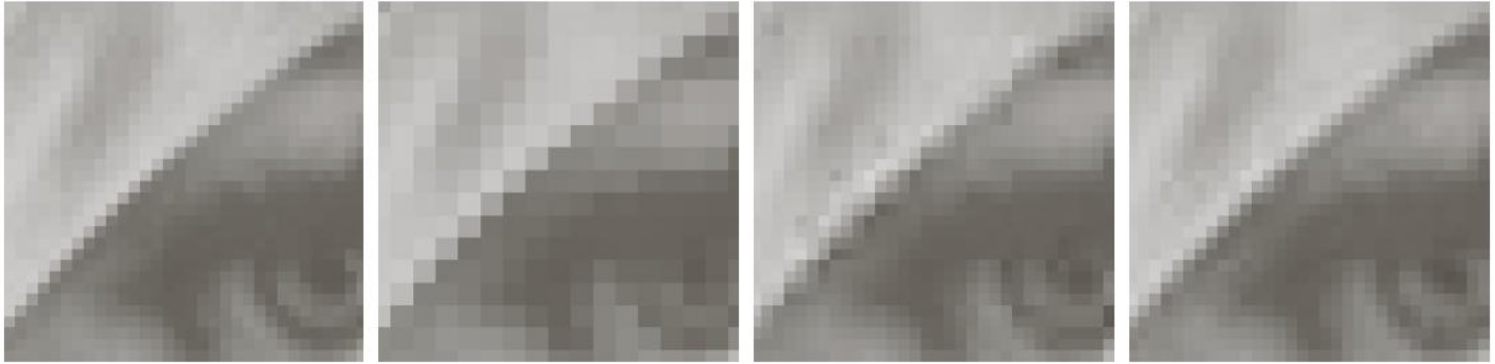


建误差与子图像尺寸的关系

( 去75%的变换系数后反变换 建图像)



# 块变换编码



使用25%DCT系数对原图的近似

(a)原图

(b)使用 $2 \times 2$ 大小的子图的近似

(c)使用 $4 \times 4$ 大小的子图的近似

(d)使用 $8 \times 8$ 大小的子图的近似

# 块变换编码

## □ 编码方式

### ■ 区域编码

- ✓ 保留的系数是根据最大方差进行选择

### ■ 值编码

- ✓ 保留的系数是根据最大幅度进行选择

## □ 比特分

- 对变换后的子图像的系数进行截取、量化和编码的过程



a b  
c d

**FIGURE 8.28**  
Approximations  
of Fig. 8.9(a) using  
12.5% of the  
 $8 \times 8$  DCT  
coefficients:  
(a)–(b) threshold  
coding results;  
(c)–(d) zonal  
coding results. The  
difference images  
are scaled by 4.

# 块变换编码

## □ 区域编码的实现

- 具有最大方差的变换系数被保留下来
- 区域取样可看做变换矩  $T(u,v)$  乘以区域模板的对应元素

1	1	1	1	1	0	0	0	8	7	6	4	3	2	1	0
1	1	1	1	0	0	0	0	7	6	5	4	3	2	1	0
1	1	1	0	0	0	0	0	6	5	4	3	3	1	1	0
1	1	0	0	0	0	0	0	4	4	3	3	2	1	0	0
1	0	0	0	0	0	0	0	3	3	3	2	1	1	0	0
0	0	0	0	0	0	0	0	2	2	1	1	1	0	0	0
0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	1	0	0	0	0	1	5	6	14	15	27	28
1	1	1	1	0	0	0	0	2	4	7	13	16	26	29	42
1	1	0	0	0	0	0	0	3	8	12	17	25	30	41	43
1	0	0	0	0	0	0	0	9	11	18	24	31	40	44	53
0	0	0	0	0	0	0	0	10	19	23	32	39	45	52	54
0	1	0	0	0	0	0	0	20	22	33	38	46	51	55	60
0	0	0	0	0	0	0	0	21	34	37	47	50	56	59	61
0	0	0	0	0	0	0	0	35	36	48	49	57	58	62	63

a b  
c d

**FIGURE 8.29**

A typical (a) zonal mask, (b) zonal bit allocation, (c) threshold mask, and (d) thresholded coefficient ordering sequence. Shading highlights the coefficients that are retained.



# 块变换编码

## □ 值编码的实现

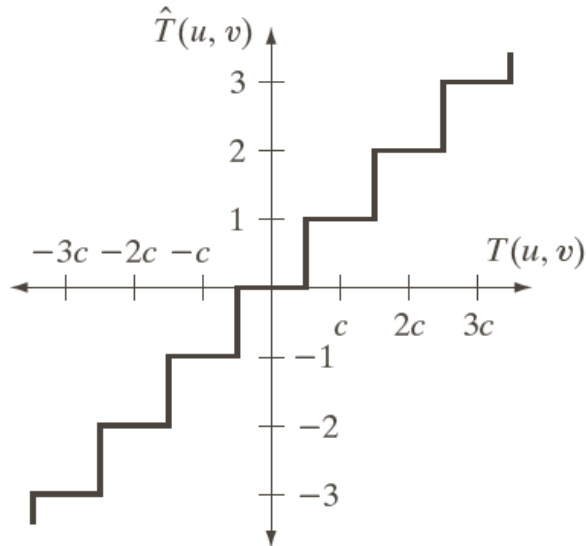
- 区域编码对所有子图像 用单个固定模板来实现
- 值编码在不同子图像保留系数的位置上具有自适应性
- 值模板函数的生成方法有三种
  - ✓ 对所有子图像 用单一的全局 值
    - 对不同图像的压缩水平不同, 取决于超过全局 值的系数数
  - ✓ 对每幅子图像 用不同的 值
    - 对每幅子图像丢掉相同数 的系数, 代码率恒定
  - ✓ 值 子图像内每个系数位置的变化而变化

$$\hat{T}(u,v) = \text{round}\left(\frac{T(u,v)}{Z(u,v)}\right) \quad \hat{T} = \hat{T}(u,v) Z(u,v)$$

$$Z = \begin{bmatrix} Z(0,0) & Z(0,1) & \dots & Z(0,n-1) \\ Z(1,0) & & \dots & \\ \vdots & & & \\ Z(n-1,0) & Z(n-1,1) & \dots & Z(n-1,n-1) \end{bmatrix} \quad \text{变换标准化 列}$$

# 块变换编码

## □ 值编码的实现



16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

a b

**FIGURE 8.30**  
 (a) A threshold coding quantization curve [see Eq. (8.2-29)]. (b) A typical normalization matrix.

# 块变换编码

## □ 值编码的实现



**FIGURE 8.31** Approximations of Fig. 8.9(a) using the DCT and normalization array of Fig. 8.30(b): (a)  $Z$ , (b)  $2Z$ , (c)  $4Z$ , (d)  $8Z$ , (e)  $16Z$ , and (f)  $32Z$ .

压缩比C: 12:1; 19:1; 30:1; 49:1; 85:1; 182:1  
相应的均方误差: 3.83, 4.93, 6.62, 9.35, 13.94和22.46



# 块变换编码

## □ JPEG标准

- 广泛使用的连续色调 静止帧压缩标准
- JPEG定义了三种编码系统
  - ✓ 有损的基本编码系统，以DCT为基础
  - ✓ 扩展的编码系统，用于更大压缩、更 精度或渐进式 建
  - ✓ 向可逆压缩的无损独立编码系统
- JPEG 序基线系统
  - ✓ 输入和输出数据精度为8比特， 化后的DCT值为11比特
  - ✓ 压缩步 ； DCT计算、 化、变 编码





# 块变换编码

## □ JPEG基本编码与解码

### ■ 8 × 8子图像

52	55	61	66	70	61	64	73
63	59	66	90	109	85	69	72
62	59	68	113	144	104	66	73
63	58	71	122	154	106	70	69
67	61	68	104	126	88	68	70
79	65	60	70	77	63	58	75
85	71	64	59	55	61	65	83
87	79	69	68	65	76	78	94

### ■ 移位128个灰度级

-76	-73	-67	-62	-58	-67	-64	-55
-65	-69	-62	-38	-19	-43	-59	-56
-66	-69	-60	-15	16	-24	-62	-55
-65	-70	-57	-6	26	-22	-58	-59
-61	-67	-60	-24	-2	-40	-60	-58
-49	-63	-68	-58	-51	-65	-70	-53
-43	-57	-64	-69	-73	-67	-63	-45
-41	-49	-59	-60	-63	-52	-50	-34



# 块变换编码

## □ JPEG基本编码与解码

### ■ 前向DCT

-415	-29	-62	25	55	-20	-1	3
7	-21	-62	9	11	-7	-6	6
-46	8	77	-25	-30	10	7	-5
-50	13	35	-15	-9	6	0	3
11	-8	-13	-2	-1	1	-4	1
-10	1	3	-3	-1	0	2	-1
-4	-1	2	-1	2	-3	1	-2
-1	-1	-1	-2	-1	-1	0	-1

### ■ 化、缩放和截断

-26	-3	-6	2	2	0	0	0
1	-2	-4	0	0	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	2	-1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0



# 块变换编码

## □ JPEG基本编码与解码

### ■ Z型扫描系数 排

[-26 -3 1 -3 -2 -6 2 -4 1 -4 1 1 5 0 2 0 0 -1 2 0 0 0 0 0 -1 -1 EOB]

### ■ 完整编码 列

1010110 0100 001 0100 0101 100001 0110 100011 001 100011 001  
001 100101 11100110 110110 0110 11110100 000 1010

### ■ 新生成的 化系数 列

-26	-3	-6	2	2	0	0	0
1	-2	-4	0	0	0	0	0
-3	1	5	-1	-1	0	0	0
-4	1	2	-1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0



# 块变换编码

## □ JPEG基本编码与解码

### ■ 反 化化

-416	-33	-60	32	48	0	0	0
12	-24	-56	0	0	0	0	0
-42	13	80	-24	-40	0	0	0
-56	17	44	-29	0	0	0	0
18	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

### ■ 反DCT变换

-70	-64	-61	-64	-69	-66	-58	-50
-72	-73	-61	-39	-30	-40	-54	-59
-68	-78	-58	-9	13	-12	-48	-64
-59	-77	-57	0	22	-13	-51	-60
-54	-75	-64	-23	-13	-44	-63	-56
-52	-71	-72	-54	-54	-71	-71	-54
-45	-59	-70	-68	-67	-67	-61	-50
-35	-47	-61	-66	-60	-48	-44	-44



# 块变换编码

## □ JPEG基本编码与解码

### ■ 移位128个灰度级

58	64	67	64	59	62	70	78
56	55	67	89	98	88	74	69
60	50	70	119	141	116	80	64
69	51	71	128	149	115	77	68
74	53	64	105	115	84	65	72
76	57	56	74	75	57	57	74
83	69	59	60	61	61	67	78
93	81	67	62	69	80	84	84

### ■ 原图与 建的差别

-6	-9	-6	2	11	-1	-6	-5
7	4	-1	1	11	-3	-5	3
2	9	-2	-6	-3	-12	-14	9
-6	7	0	-4	-5	-9	-7	1
-7	8	4	-1	6	4	3	-2
3	8	4	-4	2	6	1	1
2	2	5	-1	-6	0	-2	5
-6	-2	2	6	-4	-4	-6	10

# 块变换编码

## □ JPEG基本编码与解码



a b c  
d e f

**FIGURE 8.32** Two JPEG approximations of Fig. 8.9(a). Each row contains a result after compression and reconstruction, the scaled difference between the result and the original image, and a zoomed portion of the reconstructed image.

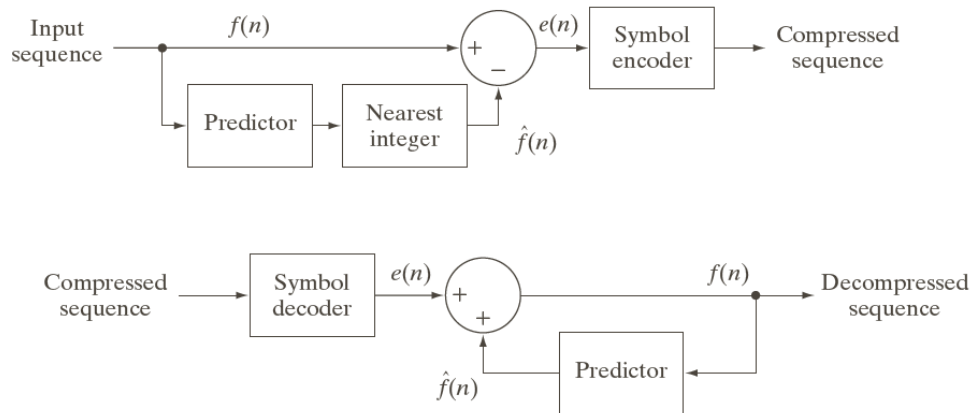
第一个结果压缩率25:1，第二个结果压缩率52:1

# 使测编码

## 概述

- 使测编码通过消 **紧** 像素在空 **和时** 上的冗余来实现
- 一个像素的新信息定义为实 **值** 和使测值的差值

## 无损使测编码



a  
b

**FIGURE 8.33**  
A lossless predictive coding model:  
(a) encoder;  
(b) decoder.

- 使测误差 
$$e(n) = f(n) - \hat{f}(n)$$

- 建 
$$f(n) = e(n) + \hat{f}(n)$$





# 使测编码

## □ 无损使测编码

- 多数情况下，使测可由m个样值进行线性组合

$$\hat{f}(n) = \text{round} \left[ \sum_{i=1}^m \alpha_i f(n-i) \right]$$

- 对一幅图像，使测值可以是来自当前扫描行（一维线性使测编码）、当前行和前几个扫描行
- 一维线性使测图像编码

$$\hat{f}(x, y) = \text{round} \left[ \sum_{i=1}^m \alpha_i f(x, y-i) \right]$$

# 无损测编码

## 无损测编码

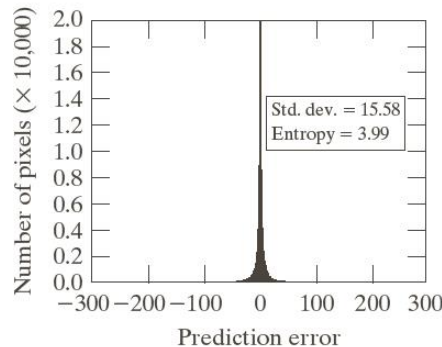
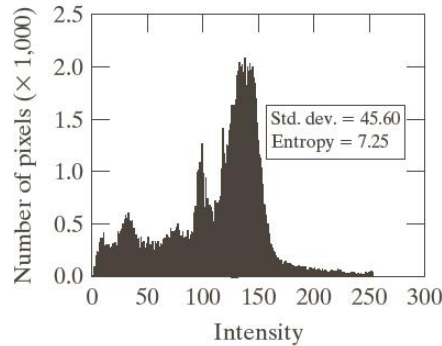
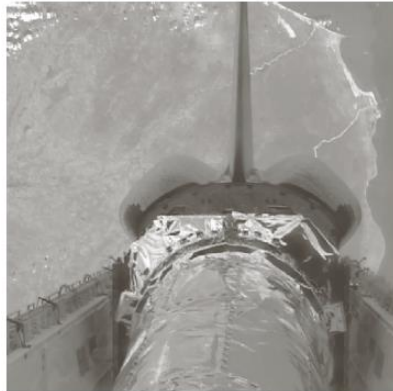
### 一维线性测图像编码示例1

$$\hat{f}(x, y) = \text{round} \left[ \sum_{i=1}^m \alpha_i f(x, y - i) \right] \quad m = 1$$

a b  
c d

**FIGURE 8.34**

(a) A view of the Earth from an orbiting space shuttle. (b) The intensity histogram of (a). (c) The prediction error image resulting from Eq. (8.2-34). (d) A histogram of the prediction error. (Original image courtesy of NASA.)



均值拉普拉斯分布

$$p_e(e) = \frac{1}{\sqrt{2}\sigma_e} e^{-\frac{\sqrt{2}|e|}{\sigma_e}}$$

# 使测编码

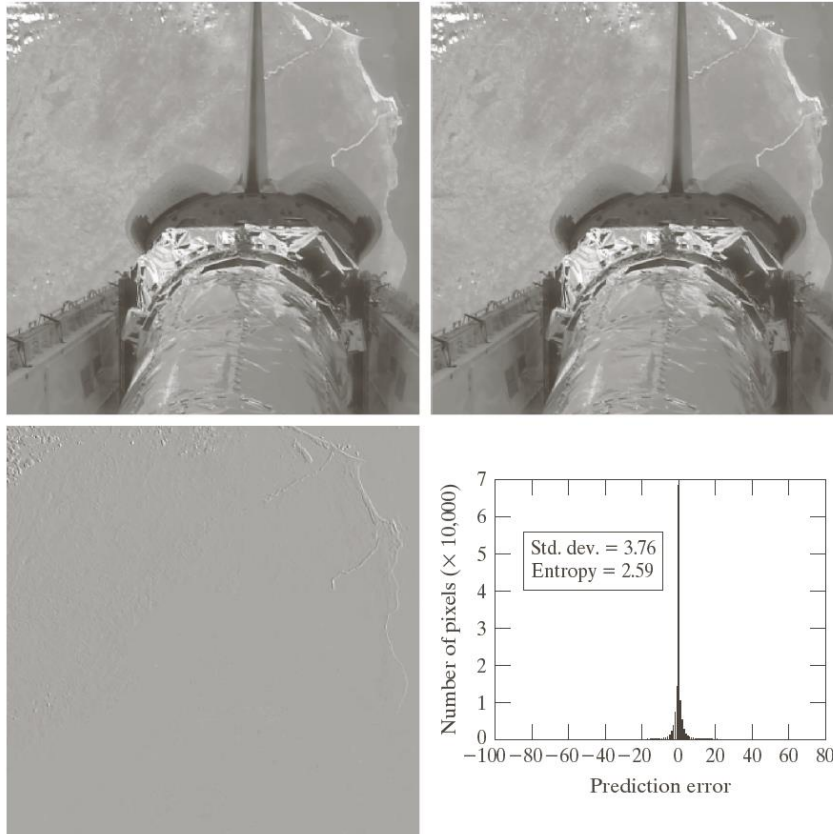
## □ 无损使测编码

### ■ 一维线性使测图像编码示例2

$$\hat{f}(x, y, t) = \text{round}[\alpha f(x, y, t - 1)]$$

a b  
c d

**FIGURE 8.35**  
(a) and (b) Two views of Earth from an orbiting space shuttle video. (c) The prediction error image resulting from Eq. (8.2-36). (d) A histogram of the prediction error.  
(Original images courtesy of NASA.)



帧内使测 vs 帧 使测  
使测误差：15.58 vs 3.76

均值拉普拉斯分布

$$p_e(e) = \frac{1}{\sqrt{2}\sigma_e} e^{-\frac{\sqrt{2}|e|}{\sigma_e}}$$

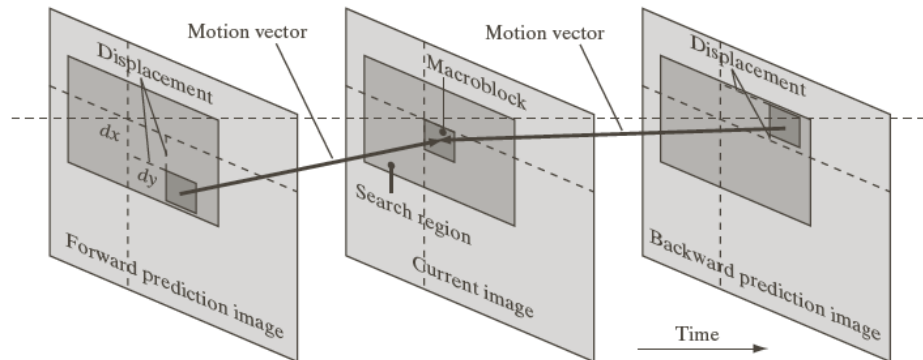
# 使测编码

## □ 运动补偿使测残差

- 视 序列相 帧相关性较强
- 使用运动补偿使测比空 使测更 效
- 运动补偿广泛用于视 编码中

## □ 运动补偿基本原理

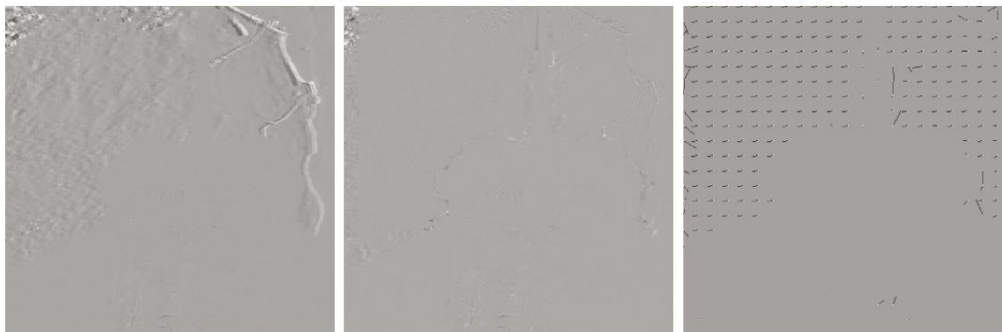
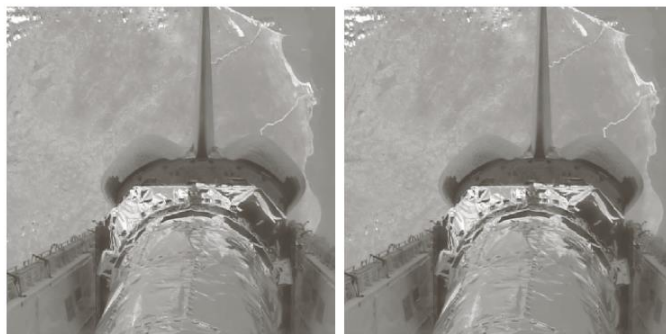
- 将视 帧分为互不 叠的子块，称为宏块
- 每个宏块在前一帧（称为参考帧）中找到最合适的块
- 使用运动矢 表示当前宏块在参考帧的位置
- 为提 运动搜索的精度，往往 用亚像素运动补偿



**FIGURE 8.36**  
Macroblock  
motion  
specification.

# 使测编码

## □ 运动补偿使测示例 1



c)使测残差：12.73灰  
度级；熵：4.17bit/pix；  
 $C=1.92$

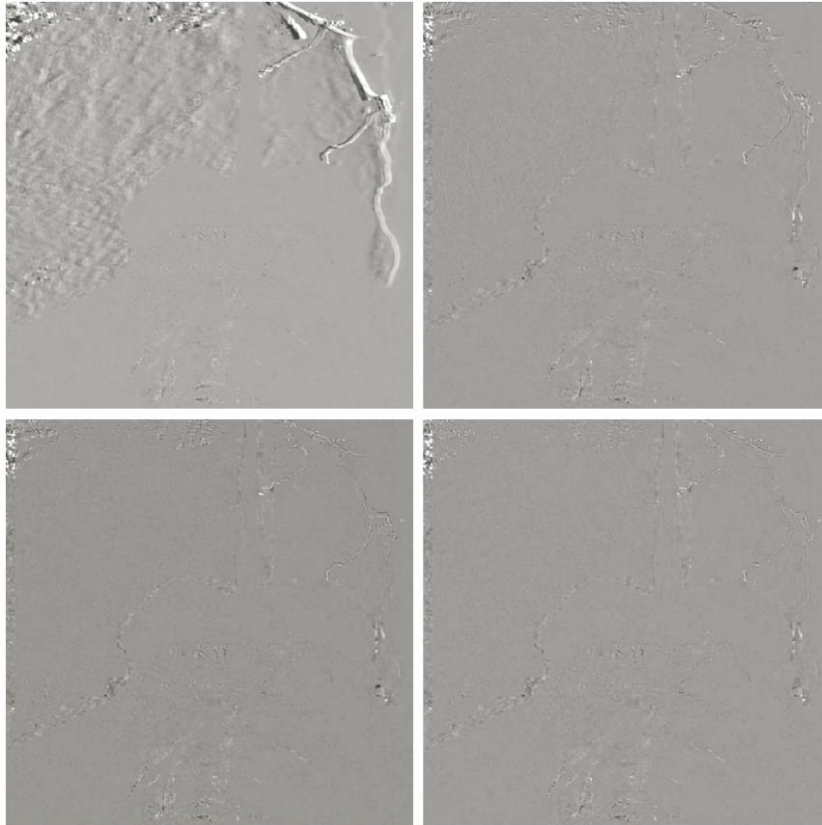
d)使测残差：5.62灰度  
级；熵：3.04bit/pix；  
 $C=2.63$

a b  
c d e

**FIGURE 8.37** (a) and (b) Two views of Earth that are thirteen frames apart in an orbiting space shuttle video. (c) A prediction error image without motion compensation. (d) The prediction residual with motion compensation. (e) The motion vectors associated with (d). The white dots in (d) represent the arrow heads of the motion vectors that are depicted. (Original images courtesy of NASA.)

# 使测编码

## □ 运动补偿使测示例 2



a b  
c d

**FIGURE 8.38**

Sub-pixel motion compensated prediction residuals: (a) without motion compensation; (b) single pixel precision; (c)  $\frac{1}{2}$  pixel precision; and (d)  $\frac{1}{4}$  pixel precision. (All prediction errors have been scaled to the full intensity range and then multiplied by 2 to increase their visibility.)



# 使测编码

- 运动估计的过程要求的计算 较大
- 只有编码端 要运动估计
- 视 压缩标准未包含运动估计，只在宏块维数、运动矢 精度等 设置约束

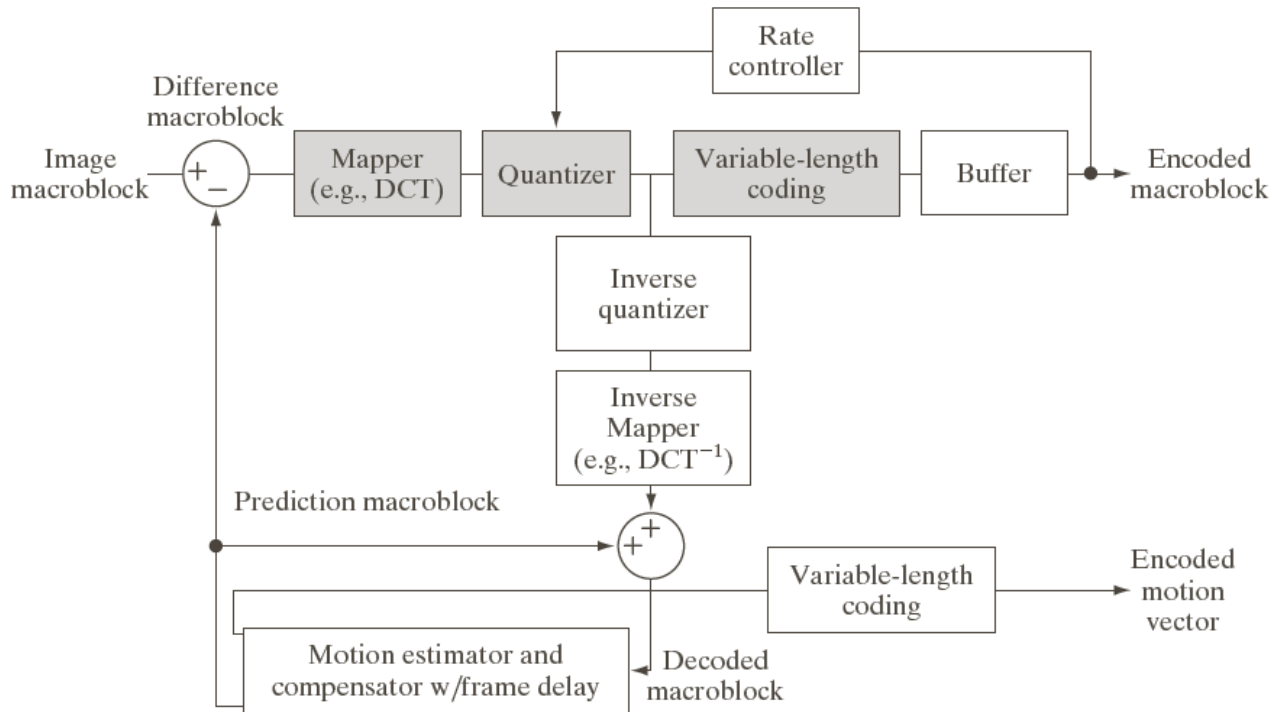
Parameter	H.261	MPEG-1	H.262 MPEG-2	H.263	MPEG-4	VC-1 WMV-9	H.264 MPEG-4 AVC
Motion vector precision	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
Macroblock sizes	$16 \times 16$	$16 \times 16$	$16 \times 16$ $16 \times 8$	$16 \times 16$ $8 \times 8$	$16 \times 16$ $8 \times 8$	$16 \times 16$ $8 \times 8$	$16 \times 16$ $16 \times 8$ $8 \times 16$ $8 \times 8$ $8 \times 4$ $4 \times 8$ $4 \times 4$
Transform	$8 \times 8$ DCT	$8 \times 8$ DCT	$8 \times 8$ DCT	$8 \times 8$ DCT	$8 \times 8$ DCT	$8 \times 8$ $8 \times 4$ $4 \times 8$ $4 \times 4$ Integer DCT	$4 \times 4$ $8 \times 8$ Integer
Interframe predictions	P	P,B	P,B	P,B	P,B	P,B	P,B
I-frame intra-predictions	No	No	No	No	No	No	Yes





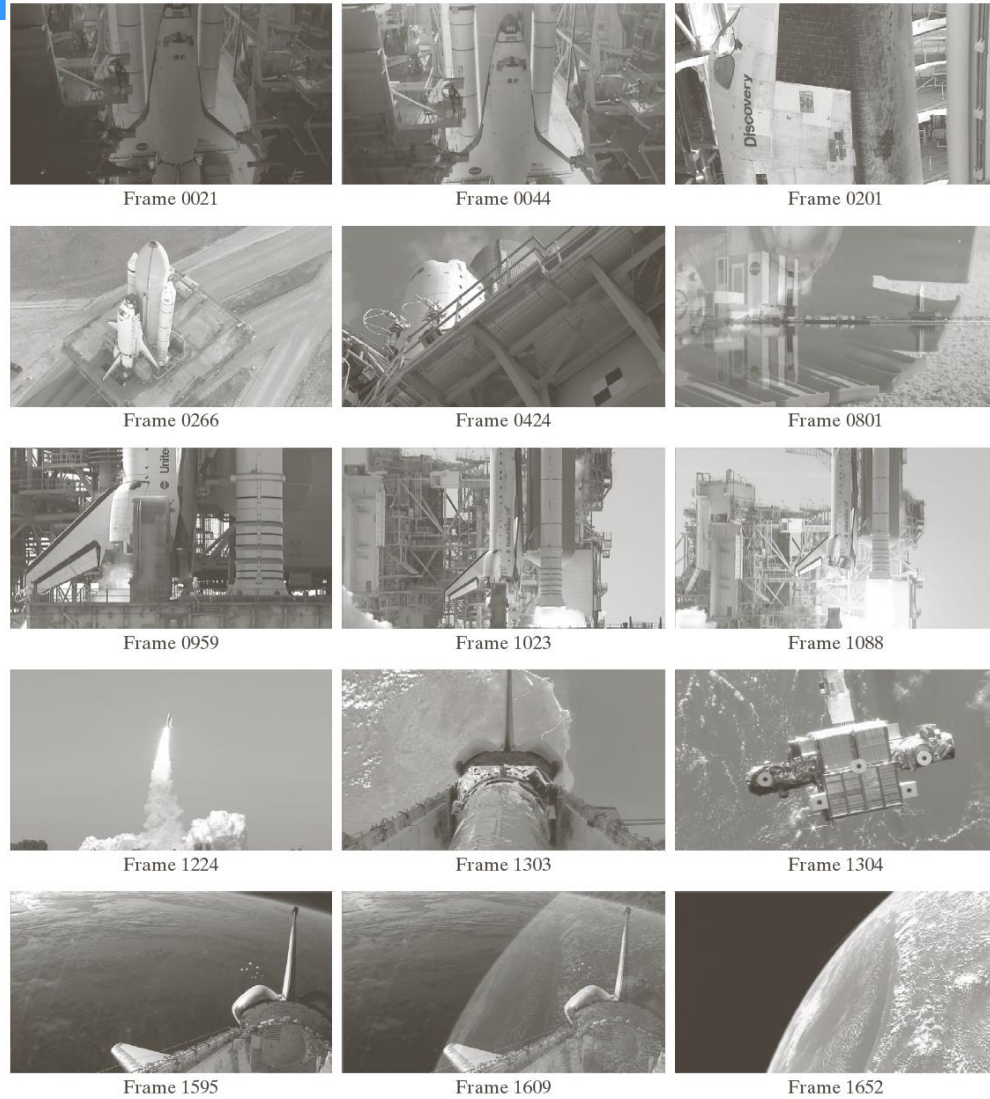
# 视频编码

## 典型的基于运动补偿的视频编码器



# 使测编码

## □ 视 压缩



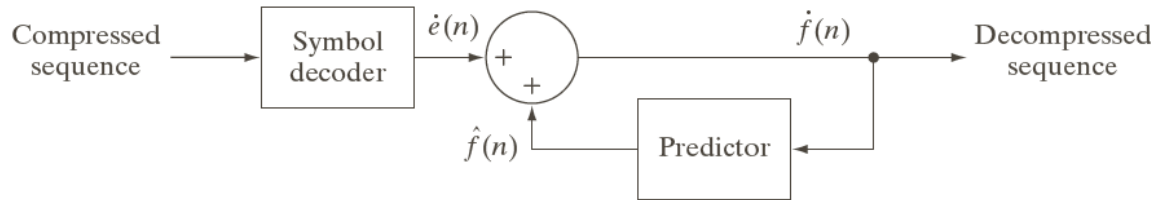
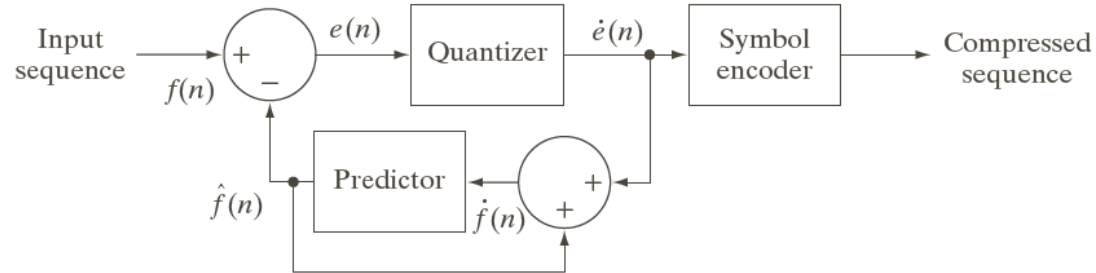
**FIGURE 8.40** Fifteen frames from an 1829-frame, 1-minute NASA video. The original video is in HD full color. (Courtesy of NASA.)



# 无损编码

## 有损无损编码

- 在反馈环中放置一个有损编码器的无损器
- 反馈环结构可以防止在解码器的输出处形成误差



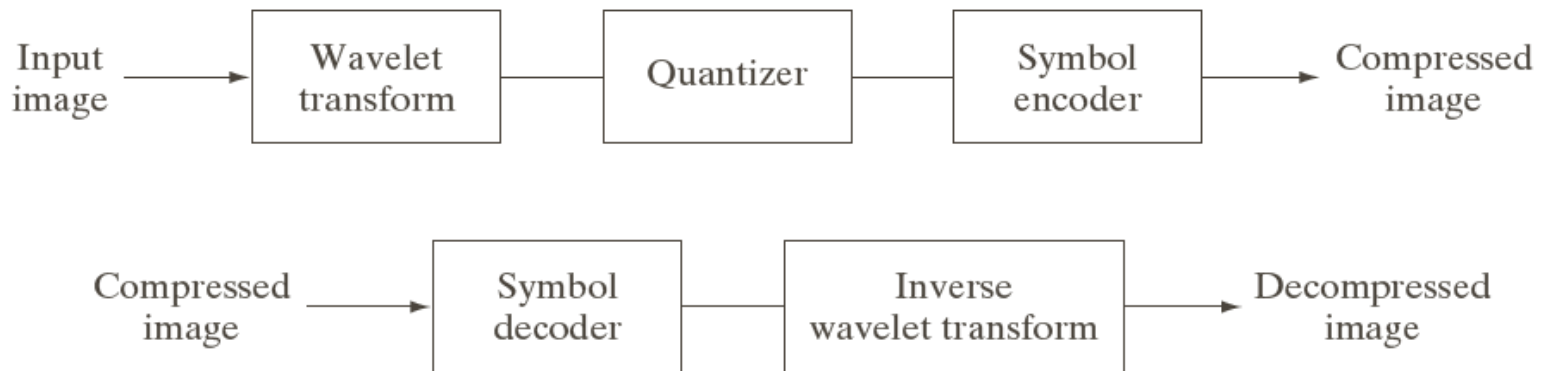
$$\dot{f}(n) = \dot{e}(n) + \hat{f}(n)$$



# 小波编码

## □ 概念

- 对图像像素解 相关的变换系数进行编码，比对原图像像素本身进行编码效率更
- 选择合适的变换基函数（小波函数），将大多数 要的视觉信息包装到少 系数中，剩下的系数可以被粗略 化甚至截取为0，而图像几乎没有失真



# 小波编码

## □ 小波基

### ■ 图像编码中广泛使用的小波有：

- ✓ Harr小波
- ✓ Daubechies小波
- ✓ 对称小波
- ✓ 双正交小波

**TABLE 8.13**

Wavelet transform filter taps and zeroed coefficients when truncating the transforms in Fig. 8.46 below 1.5.

a b  
c d

**FIGURE 8.46** Three-scale wavelet transforms of Fig. 8.9(a) with respect to (a) Haar wavelets, (b) Daubechies wavelets, (c) symlets, and (d) Cohen-Daubechies Feauveau biorthogonal wavelets.



小于1.5系数截短为0

Wavelet	Filter Taps (Scaling + Wavelet)	Zeroed Coefficients
Haar (see Ex. 7.10)	2 + 2	33.8%
Daubechies (see Fig. 7.8)	8 + 8	40.9%
Symlet (see Fig. 7.26)	8 + 8	41.2%
Biorthogonal (see Fig. 7.39)	17 + 11	42.1%



# 小波编码

## □ 分解级别选择

- 变换分解级别的数 是影响计算复杂性和 重建误差的 主要因素
- 正反变换计算操作次数 随分解级数的增加而增加

Decomposition Level (Scales or Filter Bank Iterations)	Approximation Coefficient Image	Truncated Coefficients (%)	Reconstruction Error (rms)
1	256 × 256	74.7%	3.27
2	128 × 128	91.7%	4.23
3	64 × 64	95.1%	4.54
4	32 × 32	95.6%	4.61
5	16 × 16	95.5%	4.63

**TABLE 8.14**  
Decomposition level impact on wavelet coding the 512 × 512 image of Fig. 8.9(a).

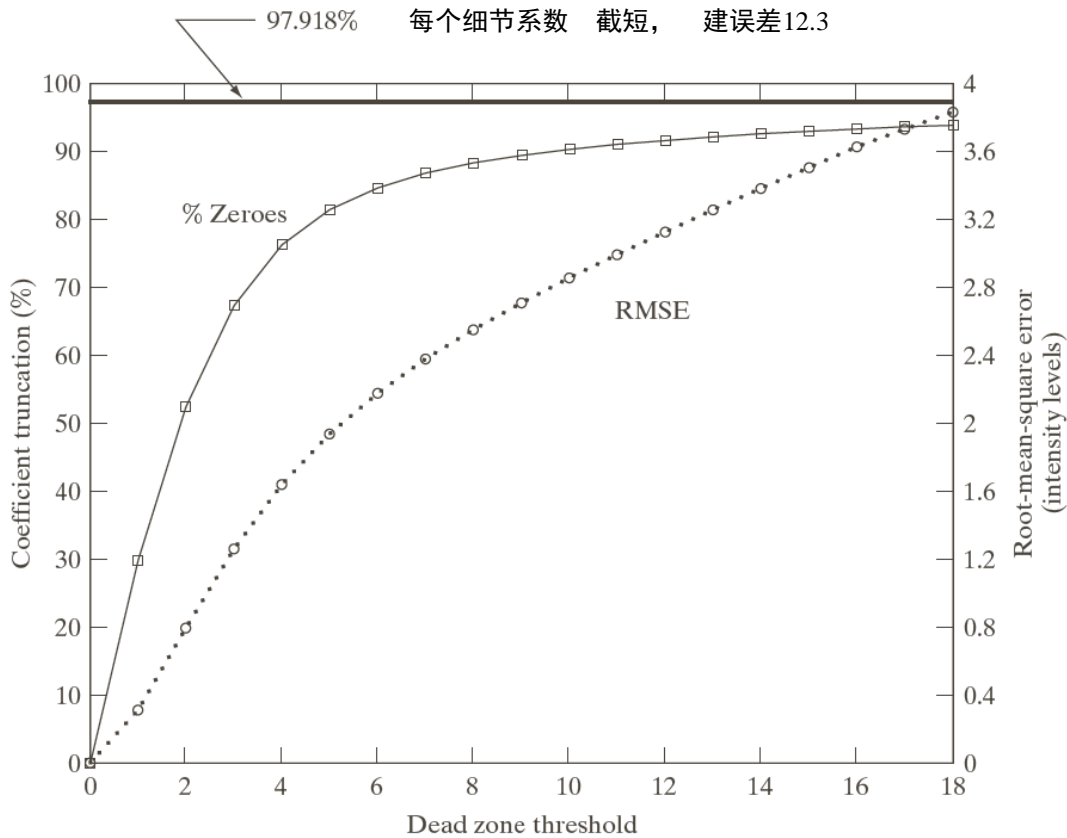
\* 用双正交小波和固定全局 值为25

## □ 量化器设计

- 系数 量化是影响小波编码压缩和 重建误差的最 主要因素
- 进一步改进
  - 引入以 为中心的较大 量化 ，称为死区
  - 从一个尺度到另一个尺度自适应调制 量化 的大小

# 小波编码

## 化器设计



**FIGURE 8.47** The impact of dead zone interval selection on wavelet coding.

死区尺寸增大时，被截断系数的数 同样增大





# 小波编码

## □ JPEG2000

- JPEG2000扩充了JPEG标准，在连续色调 止图像压缩和压缩数据的访 提供了更多灵活性
- JPEG2000编码步
  - 对编码的图像样本进行直流电平平移和去相关性
  - 将各个分 分成多个像块，即被单独处理的像素的矩 列
  - 计算每个像块分 的行和列的一维离散小波变换
    - 无损压缩以双正交、5-3系数尺度小波向 为基础
    - 有损压缩中， 用9-7系数尺度-小波向

$$Y(2n+1) = X(2n+1) + \alpha[X(2n) + X(2n+2)], i_0 - 3 \leq 2n+1 < i_1 + 3$$

$$Y(2n) = X(2n) + \beta[Y(2n-1) + Y(2n+1)], i_0 - 2 \leq 2n < i_1 + 2$$

$$Y(2n+1) = Y(2n+1) + \gamma[Y(2n) + Y(2n+2)], i_0 - 1 \leq 2n+1 < i_1 + 1$$

$$Y(2n) = Y(2n) + \delta[Y(2n-1) + Y(2n+1)], i_0 \leq 2n < i_1$$

$$Y(2n+1) = -K \square Y(2n+1), i_0 \leq 2n+1 < i_1$$

$$Y(2n) = Y(2n) / K, i_0 \leq 2n < i_1$$

提升和尺度操作



# 小波编码

## □ JPEG2000

### □ JPEG2000编码步

- 计算每个像块分 的行和列的一维离散小波变换

Filter Tap	Highpass Wavelet Coefficient	Lowpass Scaling Coefficient
0	-1.115087052456994	0.6029490182363579
±1	0.5912717631142470	0.2668641184428723
±2	0.05754352622849957	-0.07822326652898785
±3	-0.09127176311424948	-0.01686411844287495
±4	0	0.02674875741080976

**TABLE 8.15**

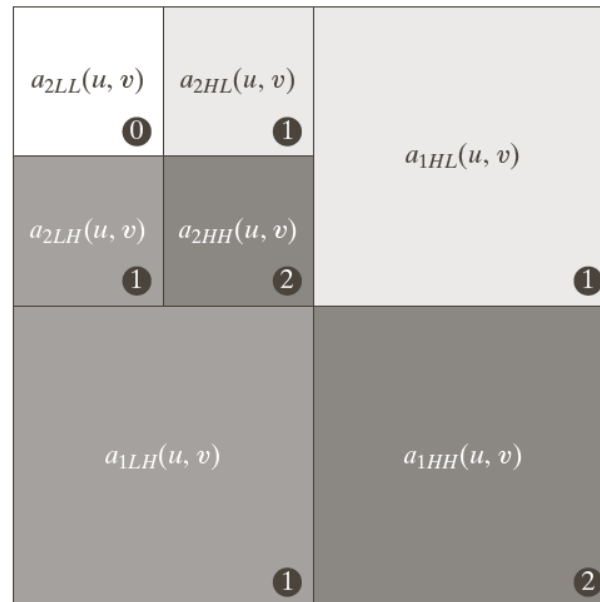
Impulse responses of the low- and highpass analysis filters for an irreversible 9-7 wavelet transform.

# 小波编码

## □ JPEG2000

### □ JPEG2000编码步

- 变换生成四个子带，分别是像块分 的低分辨率近似及分 的水平、垂直和对角 率特征。将后续迭代 制为前 的分解的近似系数，把该变换复N次，可得到N尺度小波变换



**FIGURE 8.48**  
JPEG 2000  
two-scale wavelet  
transform  
tile-component  
coefficient  
notation and  
analysis gain.



# 小波编码

## □ JPEG2000

### □ JPEG2000编码步

- 化，子带b的系数  $a_b(u,v)$  化为  $q_b(u,v)$

$$q_b(u,v) = \text{sign}[a_b(u,v)] \text{floor} \left[ \frac{|a_b(u,v)|}{\Delta_b} \right]$$

$$\Delta_b = 2^{R_b - g_b} \left( 1 + \frac{\mu_b}{2^{11}} \right)$$

- 每个像块分 的子带系数被排列为称为码块的矩形块，码块被单独编码，一次一个比特平

# 小波编码

## □ JPEG2000

从a)到d)压缩比C为25,  
52, 75, 105;

a) 误差: JPEG 5.4  
J2000 3.86;

b) 误差: JPEG 10.7  
J2000 5.77



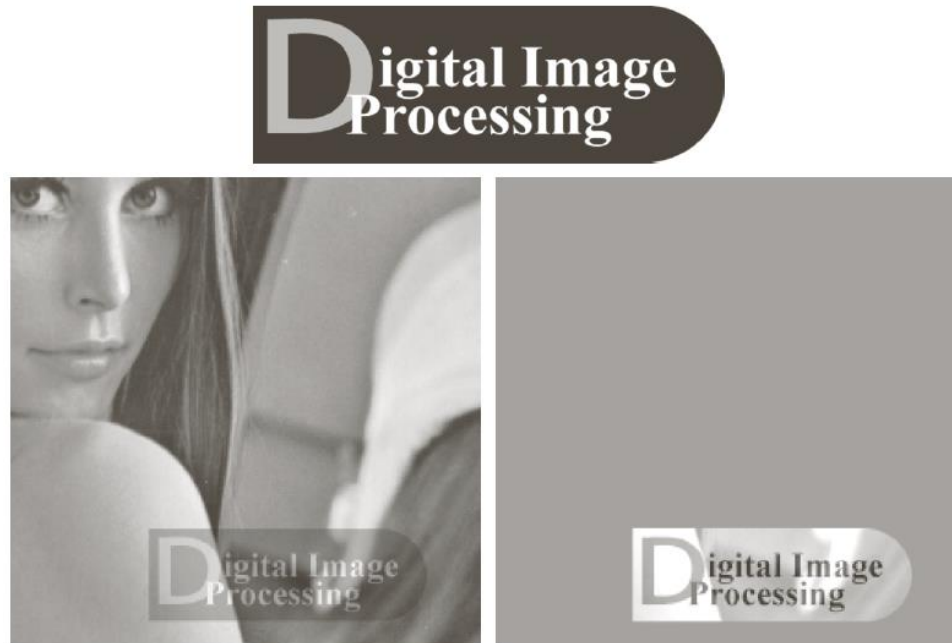
**FIGURE 8.49** Four JPEG-2000 approximations of Fig. 8.9(a). Each row contains a result after compression and reconstruction, the scaled difference between the result and the original image, and a zoomed portion of the reconstructed image. (Compare the results in rows 1 and 2 with the JPEG results in Fig. 8.32.)

# 数字图像水印

- 数字图像水印处理是把数据插入到一幅图像中的过程，从而形成对图像的保护。
- 简单的可见水印：

$$f_w = (1 - \alpha)f + \alpha w$$

其中  $f$  为原图， $w$  为水印



a  
b c

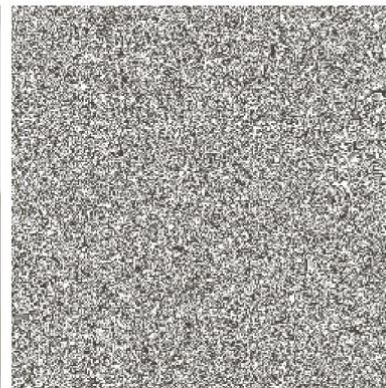
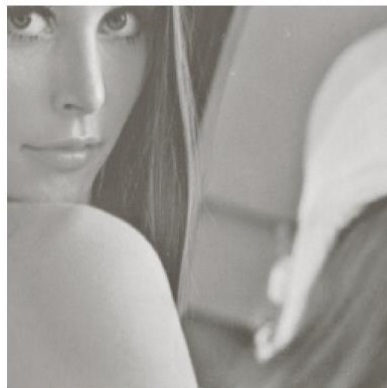
**FIGURE 8.50**  
A simple visible watermark:  
(a) watermark;  
(b) the watermarked image; and (c) the difference between the watermarked image and the original (non-watermarked) image.

# 数字图像水印

## □ 不可见水印

- 不可见水印裸眼是看不见的
- 使用合适的解码算法可以恢复原图
- 不可见水印是通过作为可见的冗余信息插入来保证的

$$f_w = 4 \left( \frac{f}{4} \right) + \frac{w}{64}$$



Digital Image Processing

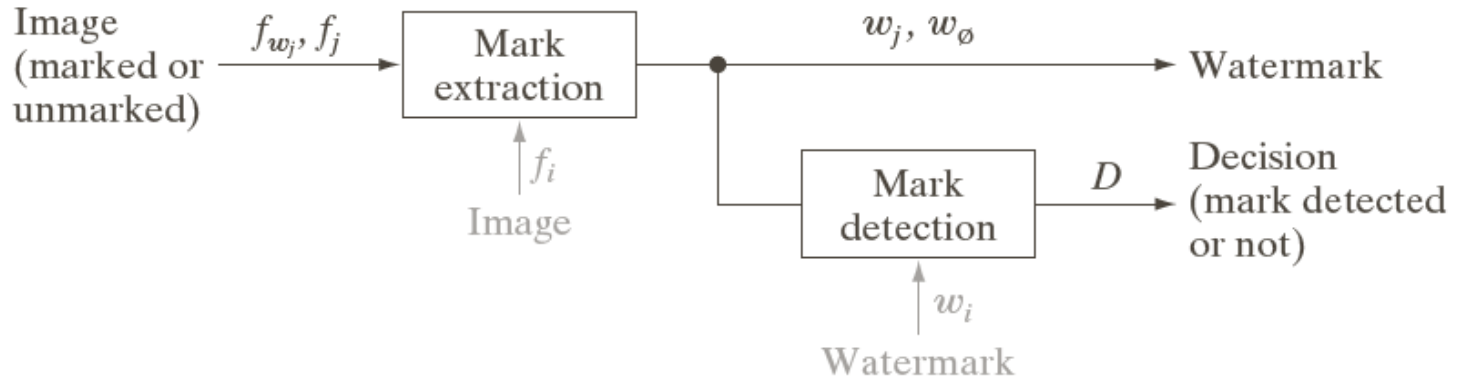
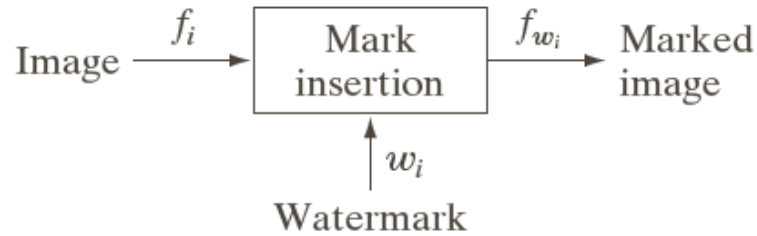
a b  
c d

**FIGURE 8.51** A simple invisible watermark: (a) watermarked image; (b) the extracted watermark; (c) the watermarked image after high quality JPEG compression and decompression; and (d) the extracted watermark from (c).



# 数字图像水印

## □ 典型的图像水印处理系统





# 数字图像水印

## □ 基于DCT的不可见的 棒水印

- ① 计算被加入水印图像的二维DCT
- ② 按幅值定位它的K个最大系数 $c_1, c_2, \dots, c_k$ .
- ③ 通过生成一个K元素伪 机数序列 $w_1, w_2, \dots, w_k$ 创建一个水印，  
这些伪 机数取自  $\mu = 0, \sigma^2 = 1$  的 斯分布
- ④ 使用下式将步 3得到的水印嵌入步 2的K个最大DCT系数中
$$c'_i = c_i * (1 + \alpha w_i), 1 \leq i \leq K$$
- ⑤ 计算步 4结果的反DCT

# 数字图像水印

## □ 基于DCT的不可见的 棒水印



a b  
c d

**FIGURE 8.53** (a) and (c) Two watermarked versions of Fig. 8.9(a); (b) and (d) the differences (scaled in intensity) between the watermarked versions and the unmarked image. These two images show the intensity contribution (although scaled dramatically) of the pseudo-random watermarks on the original image.



# 基于DCT的不可见的 棒水印的判断

□ 确定一幅特定图像是否是一幅事先加水印图像的副本，该图像具有水印  $w_1, w_2, \dots, w_k$  和DCT系数  $c_1, c_2, \dots, c_k$ .

- 计算 中图像的二维DCT
- 提取K个DCT系数，记为  $\hat{c}_1, \hat{c}_2, \dots, \hat{c}_K$ 。若 图像是事先加过水印的图像（未被修改），则对  $1 \leq i \leq K$  有  $\hat{c}_i = c'_i$ 。否则若它是加过水印图像的修改副本，则  $\hat{c}_i \approx c'_i$
- 使用下式计算  $\hat{\omega}_1, \hat{\omega}_2, \dots, \hat{\omega}_K$

$$\hat{\omega}_i = \frac{\hat{c}_i - c_i}{\alpha c_i}, 1 \leq i \leq K$$

- 使用相关系数度  $\hat{\omega}_1, \hat{\omega}_2, \dots, \hat{\omega}_K$  和  $\omega_1, \omega_2, \dots, \omega_K$  的相似度

$$\gamma = \frac{\sum_{i=1}^K (\hat{\omega}_i - \bar{\hat{\omega}})(\omega_i - \bar{\omega})}{\sqrt{\sum_{i=1}^K (\hat{\omega}_i - \bar{\hat{\omega}})^2 * \sum_{i=1}^K (\omega_i - \bar{\omega})^2}}, 1 \leq i \leq K$$



# 基于DCT的不可见的 棒水印的判断

- 确定一幅特定图像是否是一幅事先加水印图像的副本，该图像具有水印 $w_1, w_2, \dots, w_k$ 和DCT系数 $c_1, c_2, \dots, c_k$ .
- 将度 相似度  $\gamma$  和一个使定义的 值 $T$ 进行比较，进行二值检测判决。
  - ✓  $D=1$ 代表水印 $\omega_1, \omega_2, \dots, \omega_k$ 存在， $D=0$ 代表水印 $\omega_1, \omega_2, \dots, \omega_k$ 不存在

$$D = \begin{cases} 1, & \gamma \geq T \\ 0, & \text{other} \end{cases}$$

# 数字图像水印



**FIGURE 8.54** Attacks on the watermarked image in Fig. 8.53(a): (a) lossy JPEG compression and decompression with an rms error of 7 intensity levels; (b) lossy JPEG compression and decompression with an rms error of 10 intensity levels (note the blocking artifact); (c) smoothing by spatial filtering; (d) the addition of Gaussian noise; (e) histogram equalization; and (f) rotation. Each image is a modified version of the watermarked image in Fig. 8.53(a). After modification, they retain their watermarks to varying degrees, as indicated by the correlation coefficients below each image.



# 第9章 形态学图像处理

- 形态学一词通常表示生物学的一个分支，这里，我们使用同一词语表示数学形态学的内容
- 将数学形态学作为工具从图像中提取表达和描绘区域形状的有用图像分量，如边界、骨架和凸壳等。我们对预处理或后处理的形态学技术也感兴趣，比如形态学过滤、细化和修剪等。
- 下面几节我们将建立并说明数学形态学中的几个重要概念。这里介绍的许多概念可在 $n$ 维欧式空间 $E^n$ 中用公式表达，然而，我们的兴趣一开始是二值图像，这种图像的各个分量是 $Z^2$ 的元素。在9.6节，我们的讨论将扩展到灰度图像。





# 第9章 形态学图像处理

---

- 9.1 预备知识
- 9.2 腐蚀和膨胀
- 9.3 开操作与闭操作
- 9.4 击中与击不中变换
- 9.5 一些基本的形态学算法
- 9.6 灰度级形态学



# 第9章 形态学图像处理

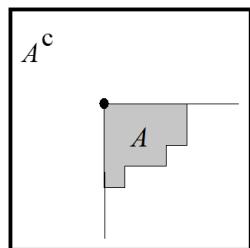
- 9.1 预备知识
- 9.2 腐蚀和膨胀
- 9.3 开操作与闭操作
- 9.4 击中与击不中变换
- 9.5 一些基本的形态学算法
- 9.6 灰度级形态学

# 9.1 预备知识

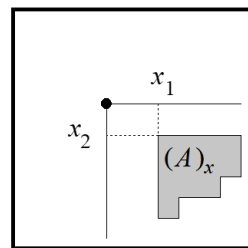
□ 数学形态学的语言是集合论。数学形态学中的集合表示图像中的对象。例如，在二值图像中，所有白色像素的集合是该图像的一个完整的形态学描述。问题集合是二维整数空间 $Z^2$ 的元素。

□ 基本集合定义：

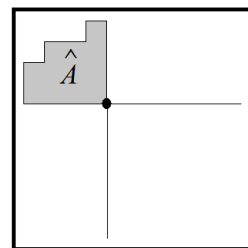
- 集合：用大写字母表示，空集记为 $\emptyset$
- 元素：用小写字母表示
- 子集： $A \subseteq B$
- 并集： $C = A \cup B$
- 交集： $D = A \cap B$
- 补集： $A^c = \{x | x \notin A\}$
- 位移： $(A)_x = \{y | y = a + x, a \in A\}$
- 映像： $\hat{A} = \{x | x = -a, a \in A\}$
- 差集： $A - B = \{x | x \in A, x \notin B\} = A \cap B^c$



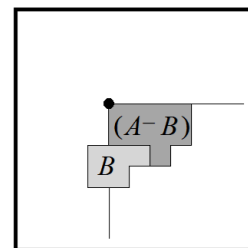
(a)



(b)



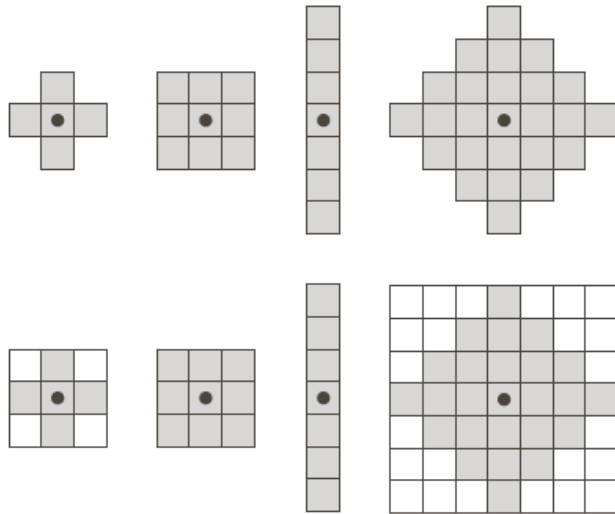
(c)



(d)

# 9.1 预备知识

- 形态学中集合的**反射**和**平移**广泛用来表达基于**结构元**（SE）的操作：研究一幅图像中感兴趣特性所用的小集合或子图像。
- 结构元：
  - 成员：每个涂阴影的方块
  - 原点：由一个黑点指出，当SE对称且未显示原点时，假定原点位于对称中心处
  - 矩形阵列：通过添加最小可能数量的背景元素形成一个矩形阵列来实现。

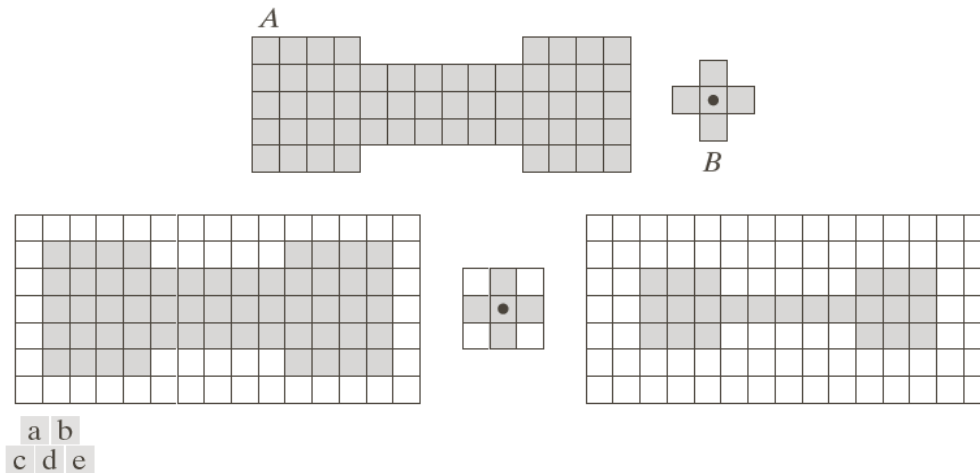


**FIGURE 9.2** First row: Examples of structuring elements. Second row: Structuring elements converted to rectangular arrays. The dots denote the centers of the SEs.

# 9.1 预备知识

## □ 形态学中如何使用结构元

- 用添加背景元素的办法把A也转换为一个矩形阵列。
- 当结构元原点位于原始集合的边界上时，背景边界要大到足以容纳整个结构元。



**FIGURE 9.3** (a) A set (each shaded square is a member of the set). (b) A structuring element. (c) The set padded with background elements to form a rectangular array and provide a background border. (d) Structuring element as a rectangular array. (e) Set processed by the structuring element.

- 定义一个用结构元B在集合A上的操作如下：通过让B在A上滑动，以便B的原点访问A的每个元素，来创建一个新集合。在B的每个原点位置，如果B完全被A包含，则将该位置标记为新集合的一个成员；否则，将该位置标记为非新集合的成员。最终结果是集合的**边界被腐蚀**。



# 第9章 形态学图像处理

9.1 预备知识

9.2 腐蚀和膨胀

9.2.1 腐蚀

9.2.2 膨胀

9.2.3 对偶性

9.3 开操作与闭操作

9.4 击中与击不中变换

9.5 一些基本的形态学算法

9.6 灰度级形态学

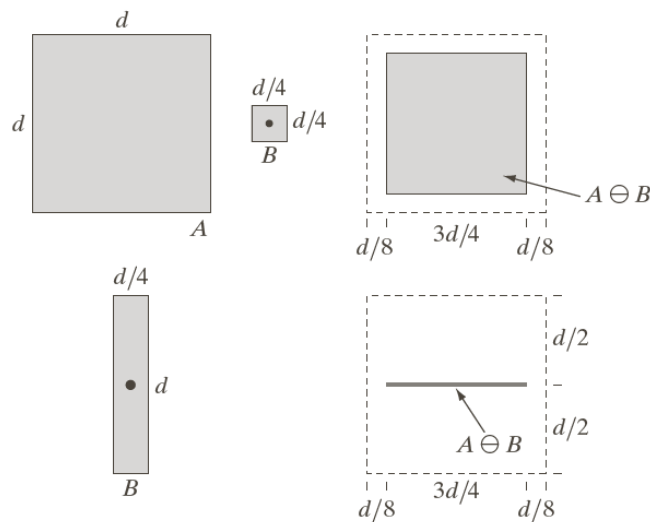
# 9.2.1 腐蚀

- 作为 $Z^2$ 中的集合A和B，表示为 $A \ominus B$ 的B对A的腐蚀定义为
 
$$A \ominus B = \{z | (B)_z \subseteq A\}$$
- 表面上，该式指出B对A的腐蚀是一个用 $z$ 平移的B包含在A中的所有的点 $z$ 的集合。
- 因为B必须包含在A中这一陈述等价于B不与背景共享任何公共元素，故可以将腐蚀表达为如下的等价形式：

$$A \ominus B = \{z | (B)_z \cap A^c = \emptyset\}$$

- 腐蚀的一个例子：

- B对A的腐蚀

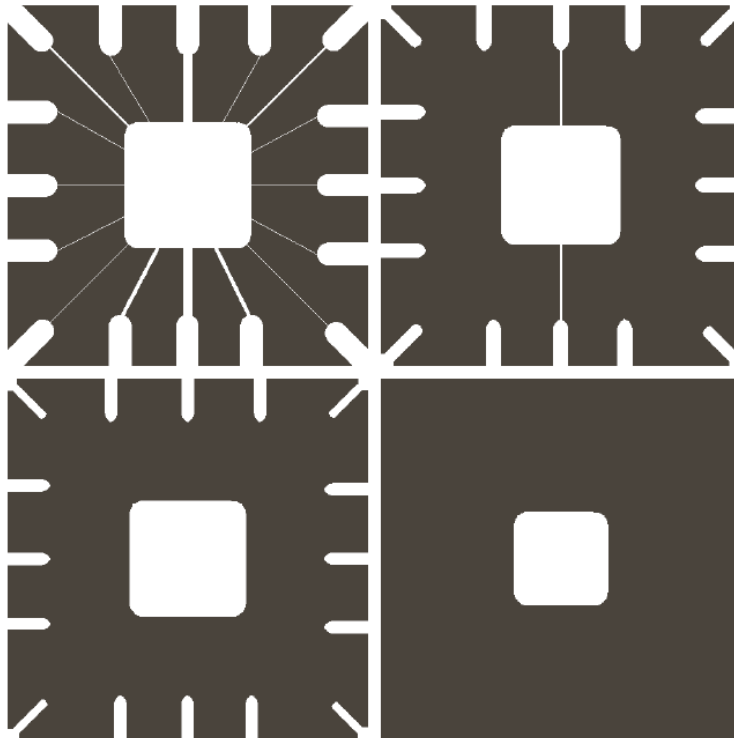


**FIGURE 9.4** (a) Set  $A$ . (b) Square structuring element,  $B$ . (c) Erosion of  $A$  by  $B$ , shown shaded. (d) Elongated structuring element. (e) Erosion of  $A$  by  $B$  using this element. The dotted border in (c) and (e) is the boundary of set  $A$ , shown only for reference.



# 9.2.1 腐蚀

- 使用腐蚀去除图像的某些部分
  - 假设我们希望去掉图中连接中心区域到边界焊接点的线
- 腐蚀缩小或细化了二值图像中的物体。
- 事实上，我们可以将腐蚀视为形态学滤波操作，这种操作把比结构元小的图像细节从图像中滤除



a b  
c d

**FIGURE 9.5** Using erosion to remove image components. (a) A  $486 \times 486$  binary image of a wire-bond mask. (b)–(d) Image eroded using square structuring elements of sizes  $11 \times 11$ ,  $15 \times 15$ , and  $45 \times 45$ , respectively. The elements of the SEs were all 1s.



# 第9章 形态学图像处理

- 9.1 预备知识
- 9.2 腐蚀和膨胀
  - 9.2.1 腐蚀
  - 9.2.2 膨胀
  - 9.2.3 对偶性
- 9.3 开操作与闭操作
- 9.4 击中与击不中变换
- 9.5 一些基本的形态学算法
- 9.6 灰度级形态学

# 9.2.2 膨胀

□ A和B是 $Z^2$ 中的集合，表示为A  $\oplus$  B的B对A的膨胀定义为

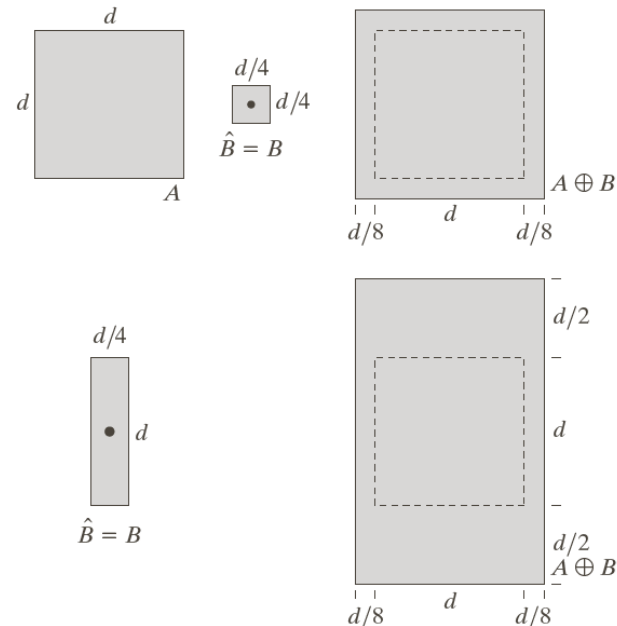
$$A \oplus B = \{z | (\hat{B})_z \cap A \neq \emptyset\}$$

□ 这个公式是以B关于它的原点的映像，且以z对映像进行平移为基础。

□ B对A的膨胀是所有位移z的集合，这样， $\hat{B}$ 和A至少有一个元素是重叠的，根据这种解释，膨胀可以等价

$$A \oplus B = \{z | [(\hat{B})_z \cap A] \subseteq A\}$$

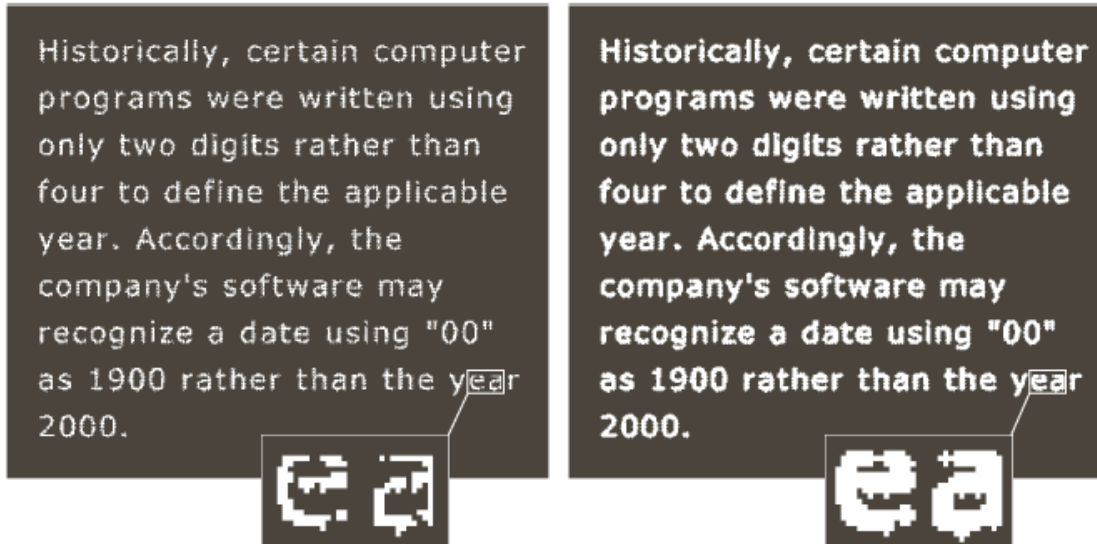
□ 腐蚀是一种收缩或细化操作，而膨胀则会“增长”或“粗化”二值图像中的物体。



**FIGURE 9.6**  
 (a) Set A.  
 (b) Square structuring element (the dot denotes the origin).  
 (c) Dilation of A by B, shown shaded.  
 (d) Elongated structuring element.  
 (e) Dilation of A using this element. The dotted border in (c) and (e) is the boundary of set A, shown only for reference

# 9.2.2 膨胀

- 最简单的膨胀应用之一是连接裂缝
  - 形态学方法较之用于连接断裂的低通滤波方法的一个直接优点是，形态学方法可在一幅二值图像中直接得到结果。
  - 另一方面，低通滤波方法则从一幅二值图像开始，生成一幅灰度图像，它需要一个阈值函数将灰度图像转换为二值图像。



a c  
b

**FIGURE 9.7**  
(a) Sample text of poor resolution with broken characters (see magnified view).  
(b) Structuring element.  
(c) Dilation of (a) by (b). Broken segments were joined.

0	1	0
1	1	1
0	1	0



# 第9章 形态学图像处理

9.1 预备知识

9.2 腐蚀和膨胀

9.2.1 腐蚀

9.2.2 膨胀

9.2.3 对偶性

9.3 开操作与闭操作

9.4 击中与击不中变换

9.5 一些基本的形态学算法

9.6 灰度级形态学



## 9.2.3 对偶性

- 膨胀和腐蚀彼此关于集合求补运算和反射运算是对偶的，即

$$(A \ominus B)^c = A^c \oplus \hat{B}$$

和

$$(A \oplus B)^c = A^c \ominus \hat{B}$$

- B对A的腐蚀是 $\hat{B}$ 对 $A^c$ 的膨胀的补集，反之亦然。
- 当结构元关于其原点对称时（通常如此），因为 $\hat{B} = B$ ，故对偶性特别有用。这样，可以用相同的结构元简单地使用B膨胀图像的背景（即膨胀 $A^c$ ），对该结果求补即可得到B对该幅图像的腐蚀。



# 第9章 形态学图像处理

- 9.1 预备知识
- 9.2 腐蚀和膨胀
- 9.3 开操作与闭操作**
- 9.4 击中与击不中变换
- 9.5 一些基本的形态学算法
- 9.6 灰度级形态学





## 9.3 开操作与闭操作

- 如我们所见，膨胀会扩大一幅图像的组成部分，而腐蚀则会缩小一幅图像中的组成部分

- 结构元B对集合A的**开操作**，表示为 $A \circ B$ ，定义如下：

$$A \circ B = (A \ominus B) \oplus B$$

因此，B对A的开操作就是B对A的腐蚀，紧接着用B对结果进行膨胀

- 类似地，用结构元B对集合A的**闭操作**，表示为 $A \cdot B$ ，定义如下：

$$A \cdot B = (A \oplus B) \ominus B$$

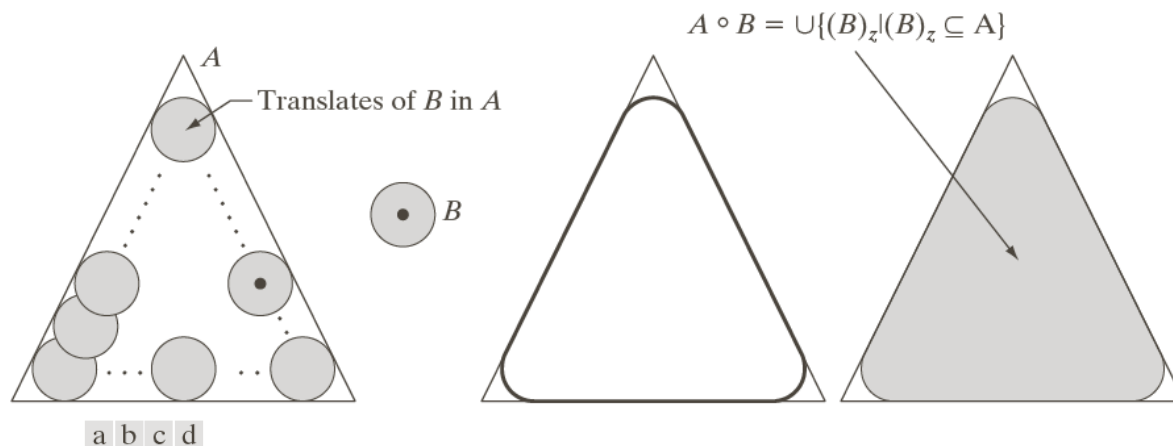
因此，B对集合A的闭操作就是用B对A膨胀，再用B对结果腐蚀

- **开操作**一般会平滑物体的轮廓、断开较窄的狭颈并消除较细的突出物。
- **闭操作**同样也会平滑轮廓的一部分，但与开操作相反，它通常会弥合较窄的间断和细长的沟壑，消除较小的孔洞，填补轮廓线中的断裂

# 9.3 开操作与闭操作：几何解释

- **开操作**有一个简单的几何解释。假设我们把结构元B视为一个“转球”。然后， $A \circ B$ 的边界由B中的点建立：当B在A的边界内侧滚动时，B所能到达的A的边界的最远点。
- 开操作的这种几何拟合特性导致了一个集合论公式，该公式表明B对A的开操作是通过拟合到A的B的所有平移的并集得到的。也就是，开操作可以表示为一个拟合处理：

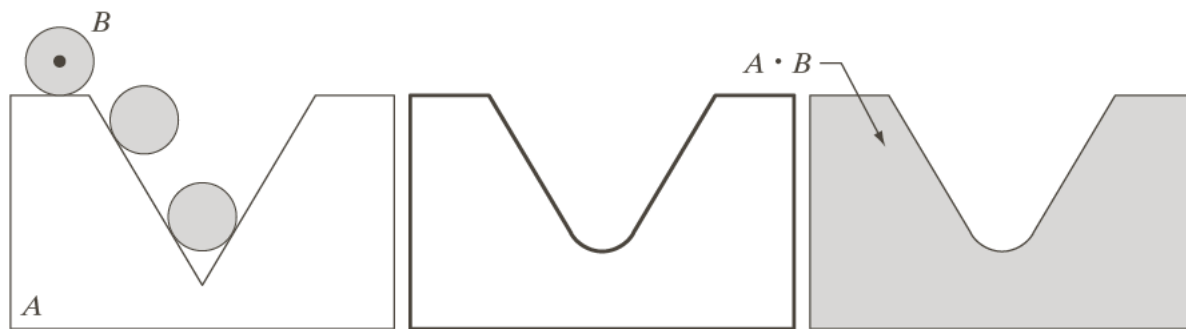
$$A \circ B = \bigcup \{(B)_z \mid (B)_z \subseteq A\}$$



**FIGURE 9.8** (a) Structuring element  $B$  “rolling” along the inner boundary of  $A$  (the dot indicates the origin of  $B$ ). (b) Structuring element. (c) The heavy line is the outer boundary of the opening. (d) Complete opening (shaded). We did not shade  $A$  in (a) for clarity.

## 9.3 开操作与闭操作：几何解释

- 除了我们现在是在边界的外侧滚动 $B$ （下图）之外，闭操作有类似的集合解释。
- 如下面所讨论的那样，开操作与闭操作彼此对偶，所以闭操作在边界外侧滚动球体是意料之中的事情。
- $B$ 在 $A$ 的外侧外切滚动一圈，所有 $(B)_z$ 的并集构成的内包络，即为 $A \cdot B$ 。



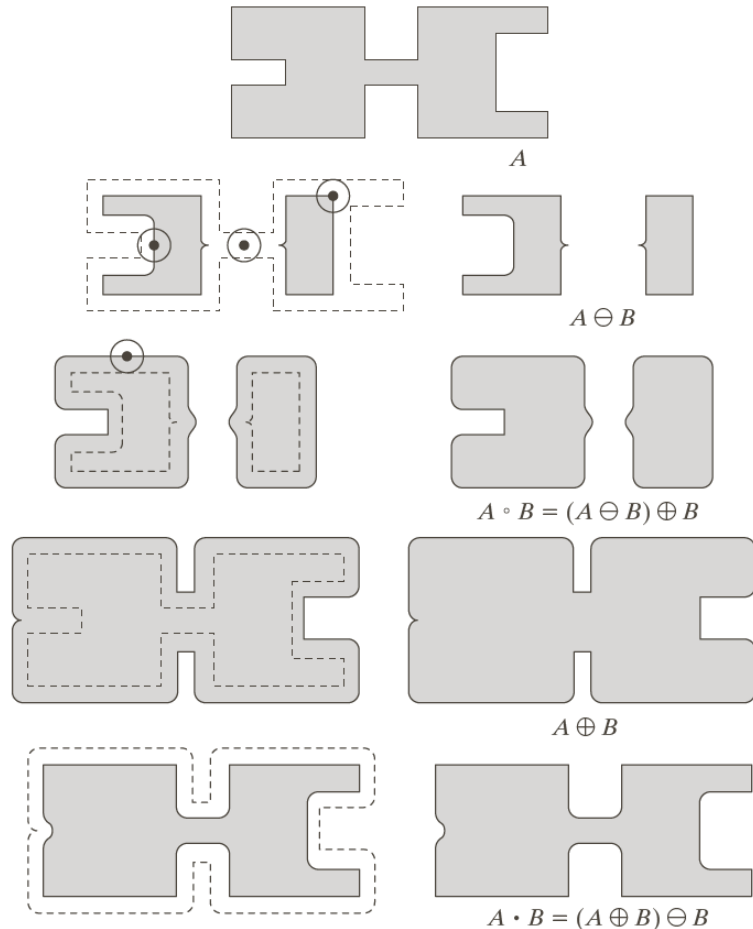
a b c

**FIGURE 9.9** (a) Structuring element  $B$  “rolling” on the outer boundary of set  $A$ . (b) The heavy line is the outer boundary of the closing. (c) Complete closing (shaded). We did not shade  $A$  in (a) for clarity.

# 9.3 开操作与闭操作

## □ 开操作与闭操作的简单说明

- 图c中两个主要部分之间的连接消失。连接部分的宽度与结构元的直径相比要细
- 图e中方向向外的角变圆，而方向向内的角则未受影响
- 图i中方向向内的角变圆，而方向向外的角则保持不变。在A的边界上，最左边的突入部分的尺寸明显减小，因为在这个位置上圆盘无法拟合。
- 还要注意使用圆盘形结构元对集合A进行开操作和闭操作所得到的物体的各个部分都变得平滑了



a  
b c  
d e  
f g  
h i

**FIGURE 9.10** Morphological opening and closing. The structuring element is the small circle shown in various positions in (b). The SE was not shaded here for clarity. The dark dot is the center of the structuring element.



## 9.3 开操作与闭操作

- 如同膨胀和腐蚀的情形那样，开操作和闭操作彼此关于集合求补和反射也是**对偶**的，即

$$(A \cdot B)^c = (A^c \circ \hat{B})$$

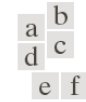
和

$$(A \circ B)^c = (A^c \cdot \hat{B})$$

- 开操作满足下列性质：
  - $A \circ B$ 是A的一个子集（子图像）
  - 如果C是D的一个子集，则 $C \circ B$ 是 $D \circ B$ 的一个子集
  - $(A \circ B) \circ B = A \circ B$
- 类似地，闭操作满足下列性质：
  - A是 $A \cdot B$ 的一个子集（子图像）
  - 如果C是D的一个子集，则 $C \cdot B$ 是 $D \cdot B$ 的一个子集
  - $(A \cdot B) \cdot B = A \cdot B$
- **注意：算子应用一次后，一个集合的多次开操作或闭操作没有影响**

# 9.3 开操作与闭操作

□ 针对形态学滤波使用开操作和闭操作



**FIGURE 9.11**

(a) Noisy image.  
 (b) Structuring element.  
 (c) Eroded image.  
 (d) Opening of  $A$ .  
 (e) Dilation of the opening.  
 (f) Closing of the opening.  
 (Original image courtesy of the National Institute of Standards and Technology.)



# 第9章 形态学图像处理

- 9.1 预备知识
- 9.2 腐蚀和膨胀
- 9.3 开操作与闭操作
- 9.4 击中与击不中变换
- 9.5 一些基本的形态学算法
- 9.6 灰度级形态学

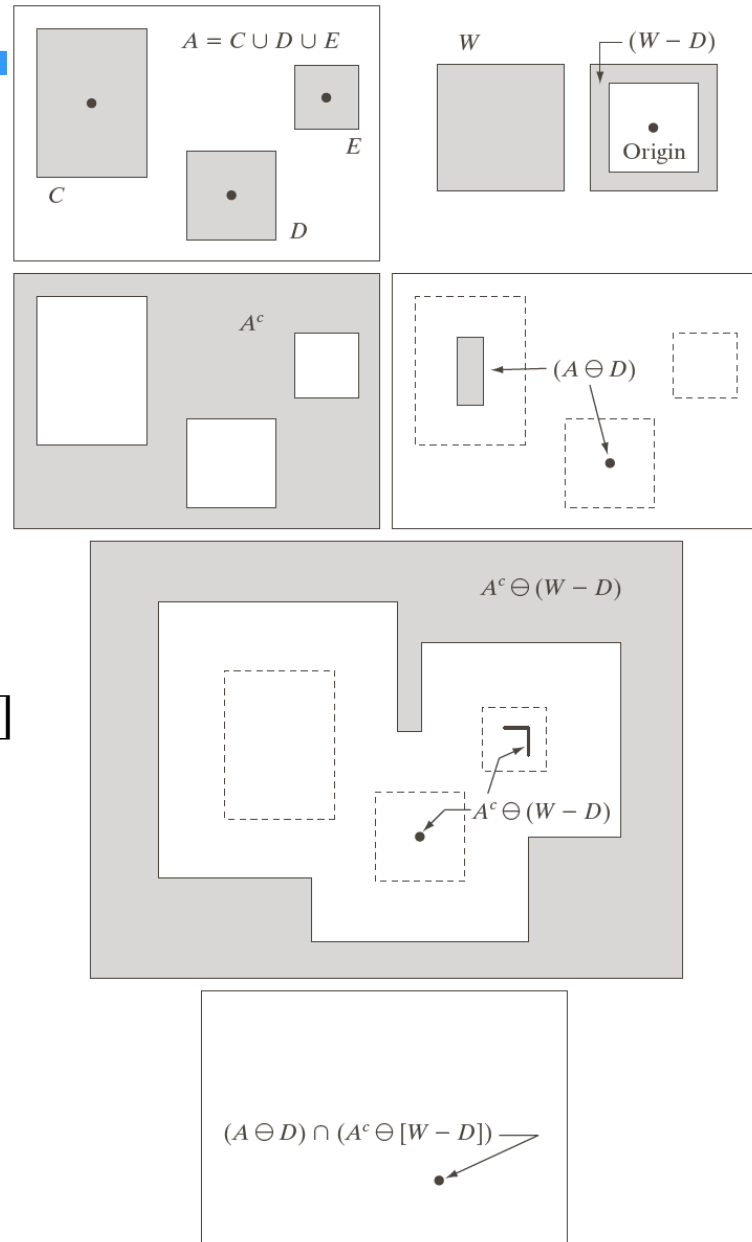


# 9.4 击中或击不中变换

□ 形态学击中或击不中变换是形状检测的基本工具。

□ 如果B表示由D及其背景组成的集合，则B在A中的匹配，表示为 $A \circledast B$ ，是

$$A \circledast B = (A \ominus D) \cap [A^c \ominus (W - D)]$$



**FIGURE 9.12**  
 (a) Set  $A$ . (b) A window,  $W$ , and the local background of  $D$  with respect to  $W$ ,  $(W - D)$ . (c) Complement of  $A$ . (d) Erosion of  $A$  by  $D$ . (e) Erosion of  $A^c$  by  $(W - D)$ . (f) Intersection of (d) and (e), showing the location of the origin of  $D$ , as desired. The dots indicate the origins of  $C$ ,  $D$ , and  $E$ .



## 9.4 击中或击不中变换

- 我们可以通过令  $B = (B_1, B_2)$  对这种表示法稍微做些推广，其中  $B_1$  是由与一个目标相联系的  $B$  的元素构成的集合， $B_2$  是由与相应背景相联系的  $B$  的元素构成的集合。根据前面的讨论， $B_1 = D$ ， $B_2 = (W - D)$ 。用这种表示方法，击中或击不中变换变为

$$A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2)$$

- 因此，集合  $A \circledast B$  包含了所有的原点，同时，在这些原点处， $B_1$  在  $A$  中找到了一个匹配（击中）， $B_2$  在  $A^c$  中也找到了一个匹配。使用腐蚀与膨胀的对偶关系，可将上式写为

$$A \circledast B = (A \ominus B_1) \cap (A \oplus \widehat{B_2})^c$$

- 使用与物体有关的结构元  $B_1$  和与背景有关的结构元  $B_2$  的原因，基于一个假设的定义——仅当两个或多个物体形成相脱离的集合时，这些物体才是可分的。
- 在某些应用中，我们可能对检测某个集合内由1和0组成的某些模式感兴趣，在这种情况下不需要背景。因此，击中或击不中就简化为简单的腐蚀。



# 第9章 形态学图像处理

- 9.1 预备知识
- 9.2 腐蚀和膨胀
- 9.3 开操作与闭操作
- 9.4 击中与击不中变换
- 9.5 一些基本的形态学算法**
- 9.6 灰度级形态学



# 9.5 一些基本的形态学算法

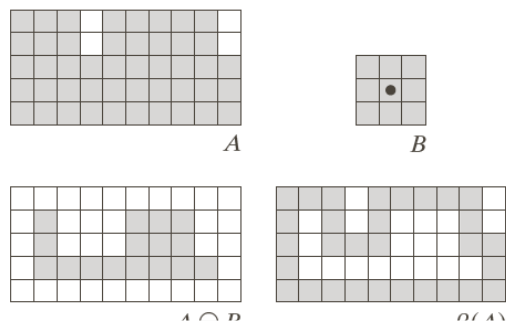
- 9.5.1 边界提取
- 9.5.2 孔洞填充
- 9.5.3 连通分量的提取
- 9.5.4 凸壳
- 9.5.5 细化
- 9.5.6 粗化
- 9.5.7 骨架
- 9.5.8 裁剪
- 9.5.9 形态学重建

# 9.5.1 边界提取

- 表示为 $\beta(A)$ 的集合A的边界可以通过先用B对A腐蚀，而后执行A和腐蚀的结果之间的差集得到，即

$$\beta(A) = A - (A \ominus B)$$

- 下图说明了边界提取的机理。这幅图像显示了一个简单的二值物体、一个结构元B和使用上式得到的结果



a b  
c d

**FIGURE 9.13** (a) Set  $A$ . (b) Structuring element  $B$ . (c)  $A$  eroded by  $B$ . (d) Boundary, given by the set difference between  $A$  and its erosion.



a b

**FIGURE 9.14** (a) A simple binary image, with 1s represented in white. (b) Result of using Eq. (9.5-1) with the structuring element in Fig. 9.13(b).



## 9.5.2 孔洞填充

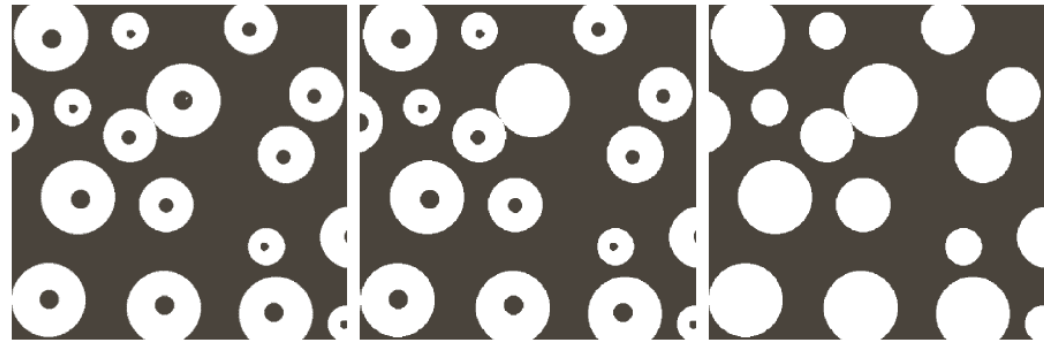
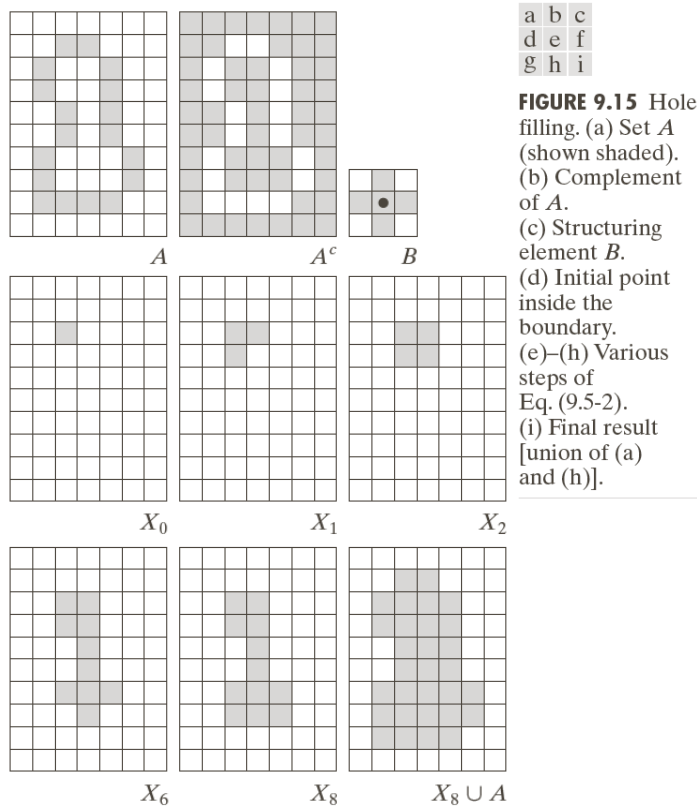
- 孔洞定义:由前景像素相连接的边界所包围的背景区域。
- 令A表示一个集合,其元素是8连通的边界,每个边界包围一个背景区域。当给定每个孔洞中的一个点后,目的就是用来填充所有的孔洞。
- 给定每个孔洞中对应于 $X_0$ 中的位置点(这一点已经置为1),我们从形成一个由0组成的阵列 $X_0$ 开始。然后,如下过程将用1填充所有孔洞:

$$X_k = (X_{k-1} \oplus B \cap A^c) \quad k = 1, 2, 3, \dots$$

- 如果 $X_k = X_{k-1}$ ,则算法在迭代的第k步结束。然后,集合 $X_k$ 包含所有被填充的孔洞。 $X_k$ 和A的并集包含所有填充的孔洞及这些孔洞的边界。
- 如果左边不加限制,那么膨胀将填充整个区域。然而,每一步中与 $A^c$ 的交集操作将把结果限制到感兴趣区域内,称为**条件膨胀**。

# 9.5.2 孔洞填充

□ 下图说明了孔洞填充的机理

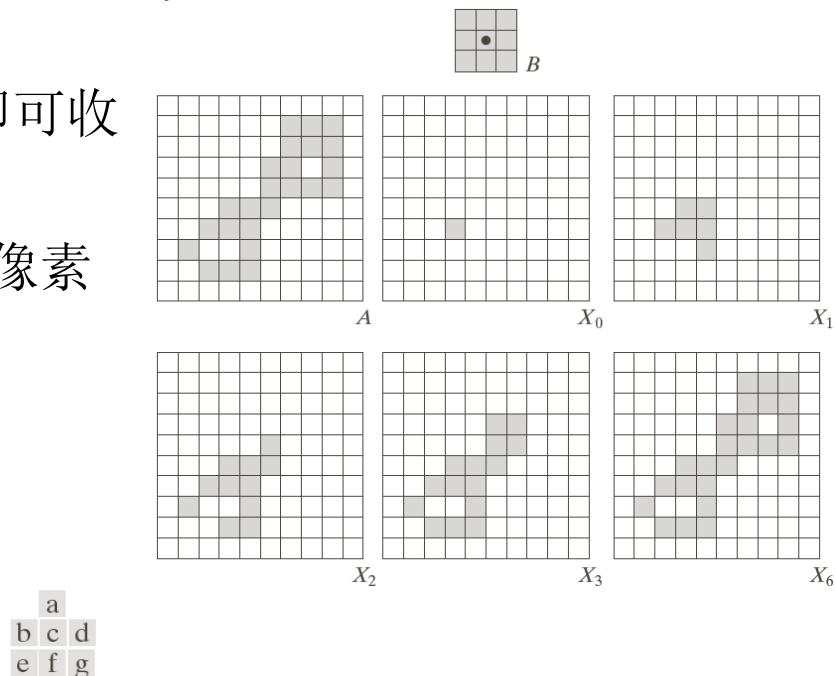


**FIGURE 9.16** (a) Binary image (the white dot inside one of the regions is the starting point for the hole-filling algorithm). (b) Result of filling that region. (c) Result of filling all holes.



# 9.5.3 连通分量的提取

- 令A是包含一个或多个连通分量的集合；并形成一一个阵列 $X_0$ ，除了在对应于A中每个连通分量的一个点的每个已知位置处我们已置为1外，该阵列的所有其他元素均为0。如下迭代过程可完成这一目的： $X_k = (X_{k-1} \oplus B \cap A) \quad k = 1, 2, 3, \dots$
- 当 $X_k = X_{k-1}$ 时，迭代过程结束， $X_k$ 包含输入图像中的所有连通分量。
- 右图说明了此机理，k=6时即可收敛。
- 注意，所用结构元的形状在像素间是基于8连通的



**FIGURE 9.17** Extracting connected components. (a) Structuring element. (b) Array containing a set with one connected component. (c) Initial array containing a 1 in the region of the connected component. (d)–(g) Various steps in the iteration of Eq. (9.5-3).

# 9.5.3 连通分量的提取

- 例：使用连通分量检测包装食品中的外来物
  - 这幅图中的最显著的特征是保留下来的点聚集为物体（骨头），而不是彼此孤立的毫无关系的点。我们通过提取图像中的连通分量来标记这些物体。
  - 15个连通分量有4个尺寸较大，这足以确定包含在原图像中的重要但不希望的物体



Connected component	No. of pixels in connected comp
01	11
02	9
03	9
04	39
05	133
06	1
07	1
08	743
09	7
10	11
11	11
12	9
13	9
14	674
15	85

a  
b  
c d

**FIGURE 9.18**

(a) X-ray image of chicken file with bone fragments.  
 (b) Thresholded image. (c) Image eroded with a  $5 \times 5$  structuring element of 1s.  
 (d) Number of pixels in the connected components of (c).  
 (Image courtesy of NTB Elektronische Geraete GmbH, Diepholz, Germany, [www.ntbxray.com](http://www.ntbxray.com).)



## 9.5.4 凸壳

- 如果在集合A内连接任意两个点的直线段都在A的内部，则称集合A是凸形的。任意集合S的凸壳H是包含S的最小凸集。差集H-S称为S的凸缺。下面介绍一种获得集合A的凸壳 $C(A)$ 的算法
- 令 $B^i, i = 1, 2, 3, 4$ 表示下图中的4个结构元。这个过程可通过执行下式实现：

$$X_k^i = (X_{k-1} \circledast B^i) \cup A \quad i = 1, 2, 3, 4 \text{ 和 } k = 1, 2, 3, \dots$$

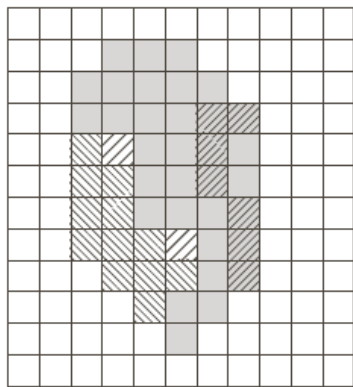
式中， $X_0^i = A$ 。当 $X_k^i = X_{k-1}^i$ 时，我们令 $D^i = X_k^i$ 。则A的凸壳为

$$C(A) = \bigcup_{i=1}^4 D^i$$

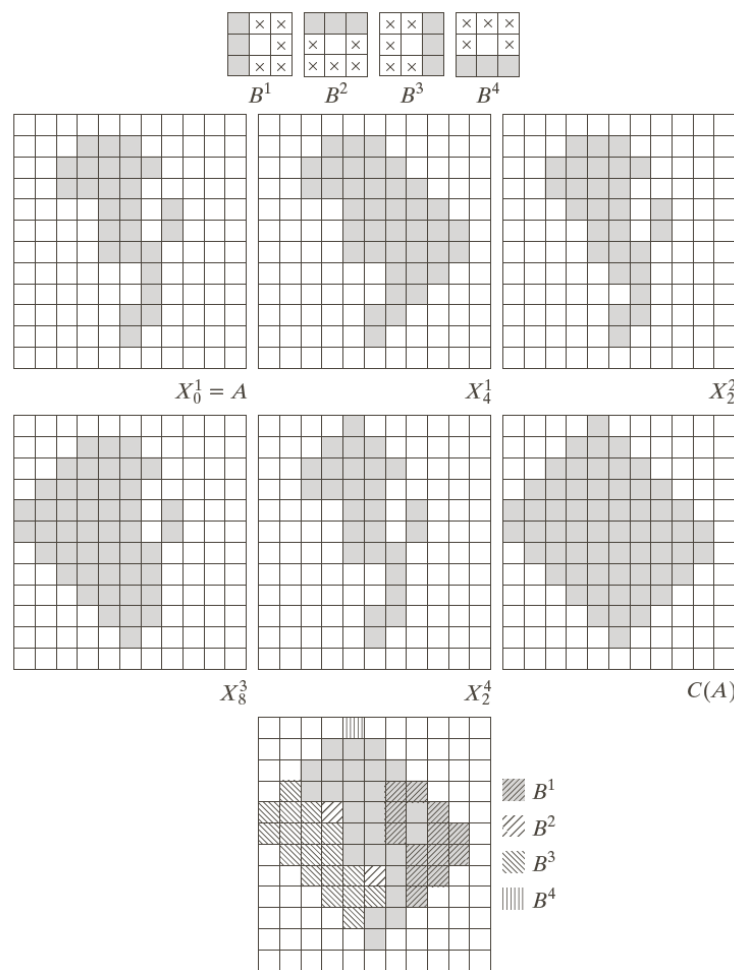
- 换句话说，该方法由反复使用 $B^1$ 对A做击中或击不中变换组成；不在发生变化时，执行与A的并集运算，结果为 $D^1$ 。这一过程使用 $B^2$ 重复（应用于A），直到不发生变化，如此往复。得到的4个D的并集组成了A的凸壳。
- 注意，我们使用不需要背景匹配的击中或击不中变换的简化实现

# 9.5.4 凸壳

- 右图说明了凸壳的过程
- 该过程的缺点是，凸壳可能超过确保凸性所需的最小尺寸。
- 减少这种影响的一种简单方法是限制生长，以便凸壳不会超过初始点集在水平和垂直方向上的尺寸，如下图：



**FIGURE 9.20** Result of limiting growth of the convex hull algorithm to the maximum dimensions of the original set of points along the vertical and horizontal directions.



**FIGURE 9.19** (a) Structuring elements. (b) Set A. (c)–(f) Results of convergence with the structuring elements shown in (a). (g) Convex hull. (h) Convex hull showing the contribution of each structuring element.



## 9.5.5 细化

- 结构元B对集合A的细化可表示为 $A \otimes B$ ，它可以根据击中或击不中变换来定义：

$$A \otimes B = A - (A \circledast B)$$

- 同前一节一样，我们仅对与结构元的模式匹配感兴趣，所以在击中或击不中变换中没有背景运算。
- 对称地细化A的一种更有用的表达方式是以结构元序列为基础的：

$$\{B\} = \{B^1, B^2, B^3, \dots, B^n\}$$

式中 $B^i$ 是 $B^{i-1}$ 旋转后的形式。

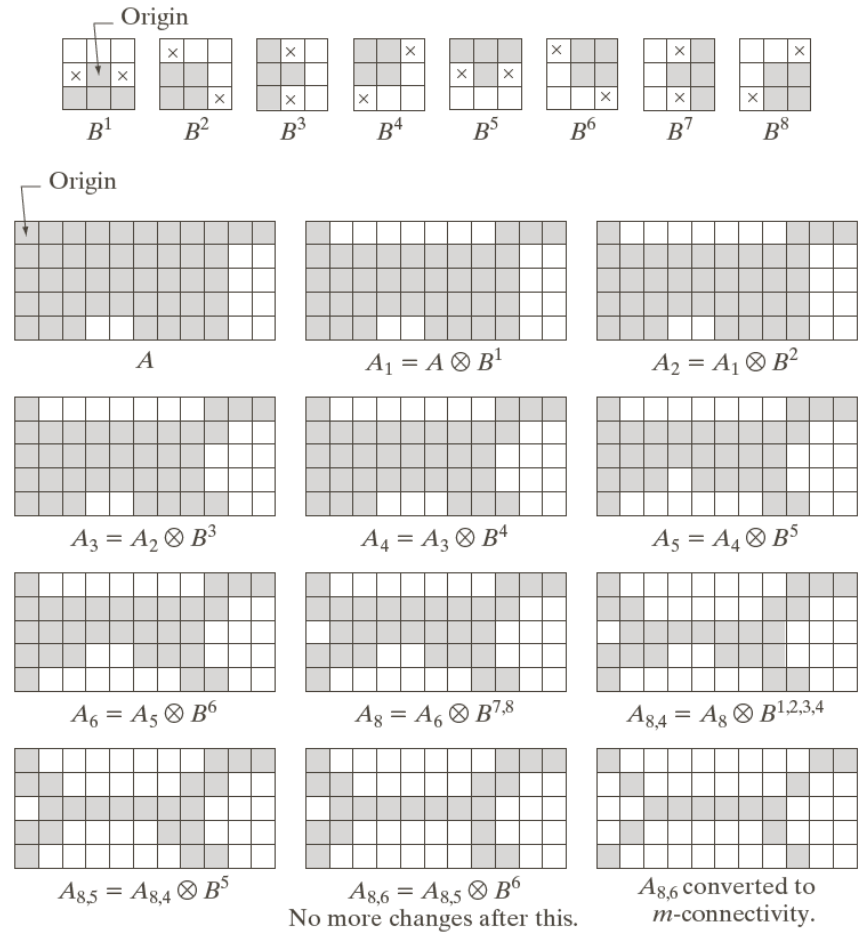
- 使用一个结构元序列将细化定义为

$$A \otimes \{B\} = ((\dots ((A \otimes B^1) \otimes B^2) \dots) \otimes B^n)$$

这种处理是A被 $B^1$ 细化一次，然后，得到的结果被 $B^2$ 细化一次，如此进行下去，直到A被 $B^n$ 细化一次。整个过程不断重复，直到得到的结果不再发生变化

# 9.5.5 细化

□ 下图说明了细化的过程



a
b c d
e f g
h i j
k l m

**FIGURE 9.21** (a) Sequence of rotated structuring elements used for thinning. (b) Set  $A$ . (c) Result of thinning with the first element. (d)–(i) Results of thinning with the next seven elements (there was no change between the seventh and eighth elements). (j) Result of using the first four elements again. (l) Result after convergence. (m) Conversion to  $m$ -connectivity.

# 9.5.6 粗化

- 粗化是细化的形态学对偶，定义如下：

$$A \odot B = AU(A * B)$$

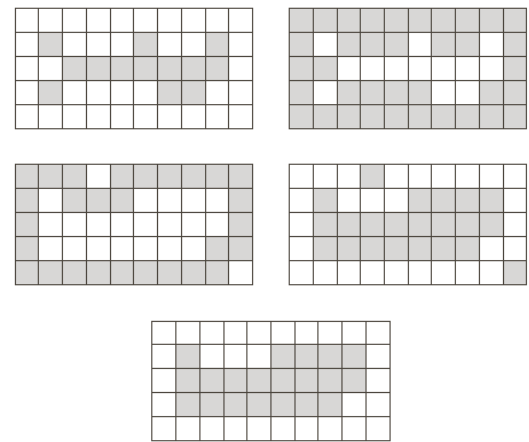
式中B是结构元，与细化一样，粗化处理也可以定义为一系列操作：

$$A \odot \{B\} = ((\dots ((A \odot B^1) \odot B^2) \dots) \odot B^n)$$

- 然而，针对粗化的分离算法在实际中很少用到，取而代之的过程是先对问题中集合的背景进行细化，而后对结果求补集。
- 换句话说，为粗化集合A，我们先形成  $C = A^c$ ，而后细化C，然后再求  $C^c$

- 下图说明了这一过程

- 由于依赖于A的性质，这个过程可能会产生某些断点。因此，通过这种方法的粗化处理通常会跟随一个消除断点的后处理

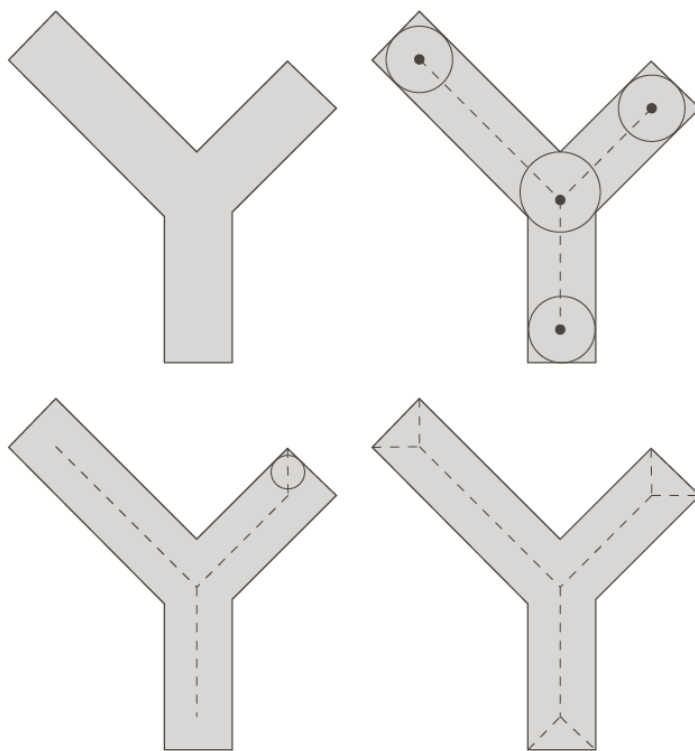


**FIGURE 9.22** (a) Set A. (b) Complement of A. (c) Result of thinning the complement of A. (d) Thickened set obtained by complementing (c). (e) Final result, with no disconnected points.



# 9.5.7 骨架

- 如图所示，集合A的骨架  $S(A)$  的概念很简单。由该图我们可以推出：
  - 如果  $z$  是  $S(A)$  的一个点，并且  $(D)_z$  是A内以  $z$  为中心的最大圆盘，则不存在包含  $(D)_z$  且位于A内的更大圆盘（不必以  $z$  为中心）。圆盘  $(D)_z$  称为最大圆盘
  - 圆盘  $(D)_z$  在两个或多个不同的位置与A的边界接触



a	b
c	d

**FIGURE 9.23**  
(a) Set  $A$ .  
(b) Various positions of maximum disks with centers on the skeleton of  $A$ .  
(c) Another maximum disk on a different segment of the skeleton of  $A$ .  
(d) Complete skeleton.



# 9.5.7 骨架

- A的骨架可以用腐蚀和开操作来表达，即骨架可以表示为

$$S(A) = \bigcup_{k=0}^K S_k(A)$$

其中， $S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B$

式中，B是一个结构元，而 $A \ominus kB$ 表示对A的连续k次腐蚀：

$$A \ominus kB = ((\dots ((A \ominus B) \ominus B) \ominus \dots) \ominus B)$$

K是A被腐蚀为空集前的最后一次迭代步骤。换句话说，

$$K = \max \{k | (A \ominus kB) \neq \emptyset\}$$

- $S(A)$ 可以作为骨架子集 $S_k(A)$ 的并集来得到，此外，还可证明使用下式可由这些子集来重建A：

$$A = \bigcup_{k=0}^K (S_k(A) \oplus kB)$$

式中 $(S_k(A) \oplus kB)$ 表示对 $S_k(A)$ 的k次连续膨胀，即

$$(S_k(A) \oplus kB) = ((\dots ((S_k(A) \oplus B) \oplus B) \oplus \dots) \oplus B)$$

# 9.5.7 骨架

## □ 计算简单图形的骨架

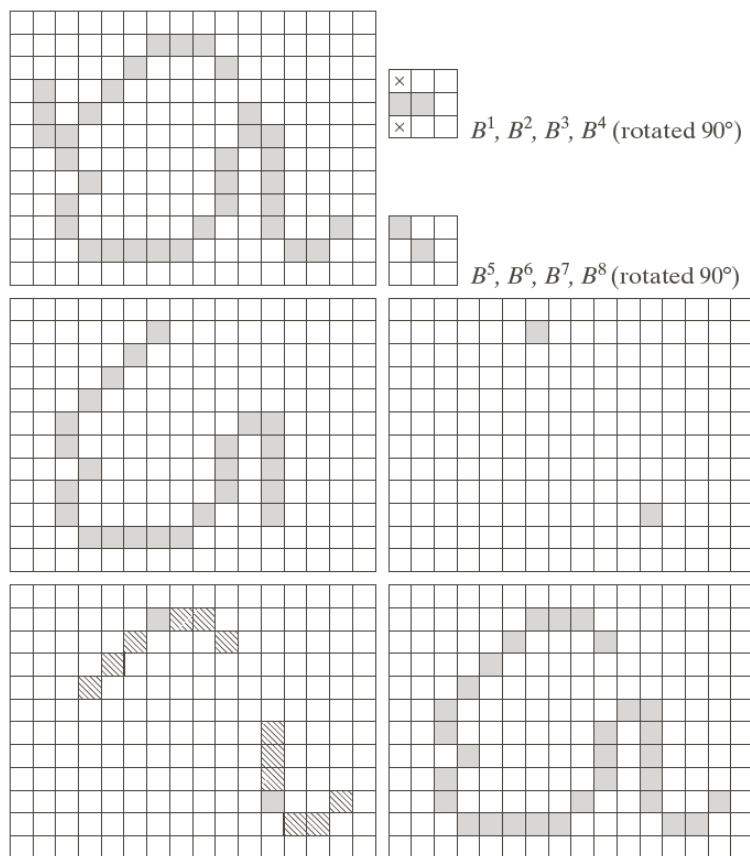
$k$	$A \ominus kB$	$(A \ominus kB) \circ B$	$S_k(A)$	$\bigcup_{k=0}^K S_k(A)$	$S_k(A) \oplus kB$	$\bigcup_{k=0}^K S_k(A) \oplus kB$
0						
1						
2						



**FIGURE 9.24**  
Implementation of Eqs. (9.5-11) through (9.5-15). The original set is at the top left, and its morphological skeleton is at the bottom of the fourth column. The reconstructed set is at the bottom of the sixth column.

# 9.5.8 裁剪

- 裁剪方法本质上是对细化和骨架算法的补充，因为这些过程会保留某些寄生成分，因而需要用后处理来清除这些寄生成分
- 在手写字符的自动识别中，通常使用的方法是分析每个字符的骨架形状。这些骨架通常带有许多“毛刺”（寄生成分）。



a b  
c  
d e  
f g

**FIGURE 9.25**  
 (a) Original image. (b) and (c) Structuring elements used for deleting end points. (d) Result of three cycles of thinning. (e) End points of (d). (f) Dilated end points conditioned on (a). (g) Pruned image.



# 9.5.8 裁剪

- 上图显示了手写字符“a”的骨架。该字符最左侧的寄生成分就是我们要消除的部分。使用一系列仅设计用来**检测端点**的结构元，对输入集合A进行**细化**可以得到期望的结果。令

$$X_1 = A \otimes \{B\}$$

式中， $\{B\}$ 表示结构元序列。连续对A应用三次，得到集合 $X_1$ 。

- 下一步是将字符“复原”为原来的形状，但要去掉寄生分支。首先需要求形成一个包含 $X_1$ 中所有端点的集合 $X_2$ ：

$$X_2 = \bigcup_{k=1}^8 (X_1 \circledast B^k)$$

式中， $B^k$ 是相同端点检测子。

- 下一步是用集合A作为限定器，对端点进行三次膨胀（**条件膨胀**）：

$$X_3 = (X_2 \oplus H) \cap A$$

最后， $X_3$ 和 $X_1$ 的并集就是我们想要的结果：

$$X_4 = X_1 \cup X_3$$



# 9.5.9 形态学重建

- 形态学重建涉及两幅图像和一个结构元。
  - 一幅图像是标记，它包含变换的起始点
  - 另一幅图像是模板，它用来约束该变换。结构元用来定义连续性
- 测地膨胀和腐蚀
  - 形态学重建的核心是测地膨胀和测地腐蚀这两个概念。
- 用腐蚀和膨胀的形态学重建
- 应用实例
  - 重建开操作
  - 填充孔洞
  - 边界清除

# 9.5.9 形态学重建

## □ 测地膨胀和腐蚀

- 令 $F$ 表示标记图像， $G$ 表示模板图像。假定两幅都是二值图像，且 $F \subseteq G$ 。令 $D_G^{(1)}(F)$ 表示大小为1的标记图像关于模板的**测地膨胀**，定义为

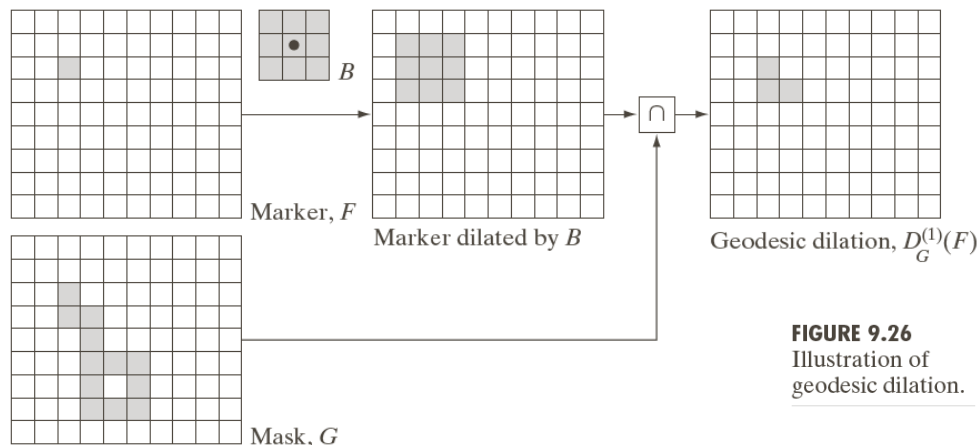
$$D_G^{(1)}(F) = (F \oplus B) \cap G$$

$F$ 关于 $G$ 的大小为 $n$ 的测地膨胀定义为

$$D_G^{(n)}(F) = D_G^{(1)} [D_G^{(n-1)}(F)]$$

式中， $D_G^{(0)}(F) = F$ 。在这个递推表达式中，集合求交在每一步中都执行。注意，交集算子保证模板 $G$ 将限制标记 $F$ 的生长（膨胀）。

- 下图显示了一个大小为1的测地膨胀的简单例子。



**FIGURE 9.26**  
Illustration of  
geodesic dilation.



# 9.5.9 形态学重建

## □ 测地膨胀和腐蚀

- 类似地，标记F关于模板G的大小为1的**测地腐蚀**定义为

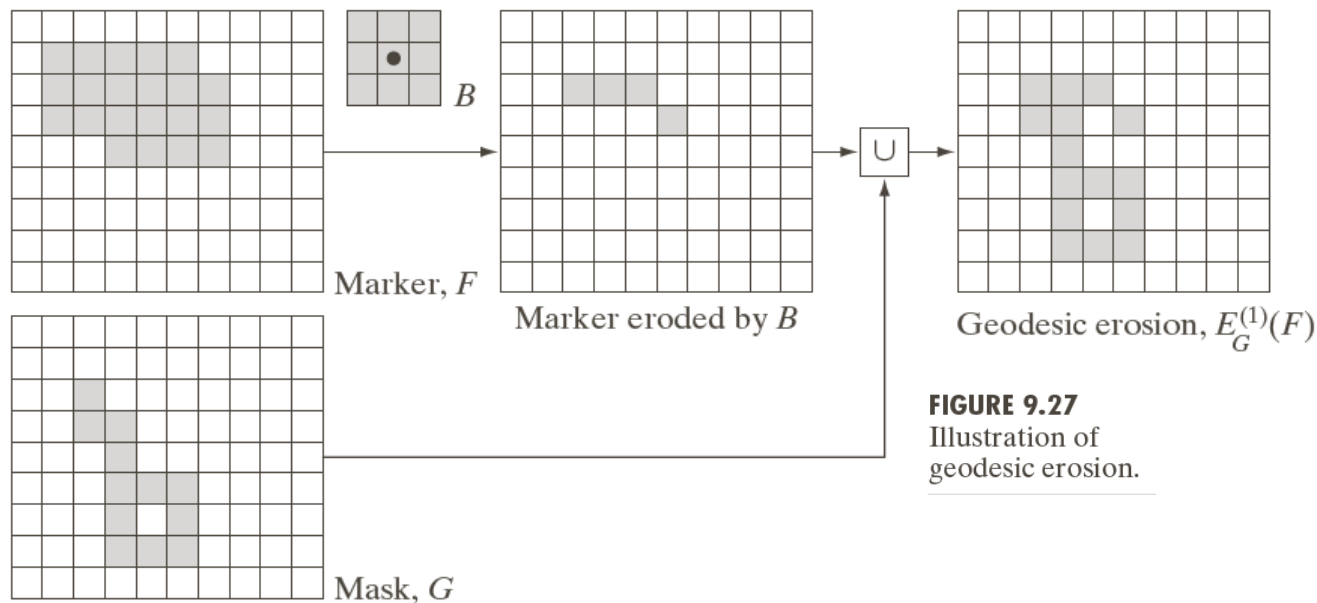
$$E_G^{(1)}(F) = (F \ominus B) \cup G$$

F关于G的大小为n的测地腐蚀定义为

$$E_G^{(n)}(F) = E_G^{(1)} [E_G^{(n-1)}(F)]$$

式中， $E_G^{(0)}(F) = F$ 。测地膨胀和测地腐蚀关于集合的补集对偶。

- 下图显示了一个大小为1的测地腐蚀的简单例子



**FIGURE 9.27**  
Illustration of  
geodesic erosion.

# 9.5.9 形态学重建

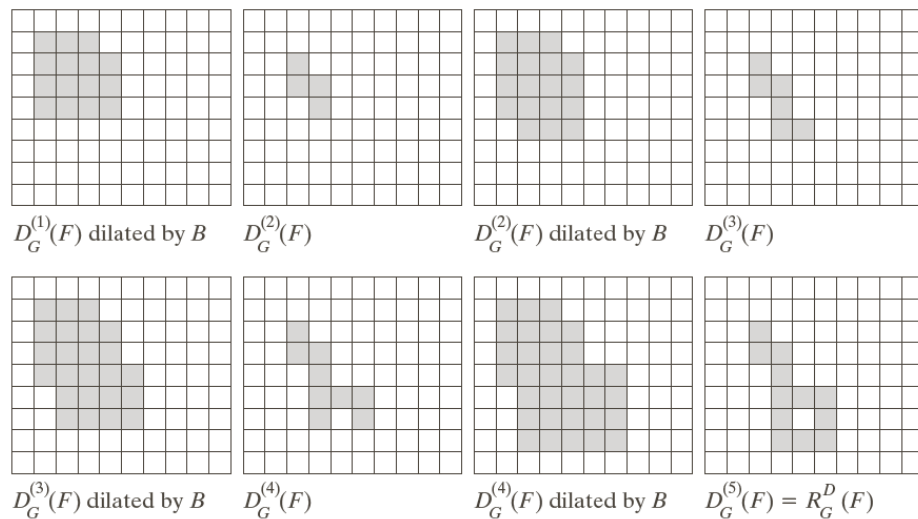
## □ 用腐蚀和膨胀的形态学重建

- 来自标记图像F对模板图像G的**膨胀形态学重建**表示为 $R_G^D(F)$ ，它被定义为F关于G的测地膨胀，**反复迭代直至达到稳定状态**；即

$$R_G^D(F) = D_G^{(k)}(F)$$

迭代k次，直至 $D_G^{(k)}(F) = D_G^{(k+1)}(F)$ 。

- 下图说明了使用膨胀的重建。得到 $D_G^{(1)}(F)$ 后，重建的下一步是膨胀该结果，然后用模板G与其相“与”得到 $D_G^{(2)}(F)$ ， $D_G^{(2)}(F)$ 的膨胀结果与目标G相“与”得到 $D_G^{(3)}(F)$ ，等等。重复这一过程，直至达到稳定



a	b	c	d
e	f	g	h

**FIGURE 9.28**  
Illustration of morphological reconstruction by dilation.  $F$ ,  $G$ ,  $B$  and  $D_G^{(1)}(F)$  are from Fig. 9.26.



# 9.5.9 形态学重建

## □ 用腐蚀和膨胀的形态学重建

- 依照类似的方法，模板图像G对标记图像F的**腐蚀形态学重建**表示为 $R_G^E(F)$ ，它被定义为F关于G的测地腐蚀，反复迭代直至达到稳定状态；即

$$R_G^E(F) = E_G^{(k)}(F)$$

迭代k次，直至 $E_G^{(k)}(F) = E_G^{(k+1)}(F)$ 。

- 膨胀和腐蚀形态学重建关于集合补集对偶

# 9.5.9 形态学重建

## □ 应用实例

- **重建开**操作：形态学开操作中，腐蚀会删除小的物体，而后续的膨胀会试图恢复遗留物体的形状。然而这种恢复的准确性高度依赖于物体的形状和所用结构元的相似性。
- 重建开操作可正确地**恢复腐蚀后所保留物体的形状**。图像F的大小为n的重建开操作，定义为来自F的大小为n的腐蚀的F的膨胀重建；即

$$O_R^{(n)}(F) = R_F^D[(F \ominus nB)]$$

式中， $(F \ominus nB)$ 表示B对F的n次腐蚀。

- 对于**重建闭**操作，可写出类似的式子

右图显示了重建开操作的一个例子，我们的兴趣是提取垂直笔画的长字符

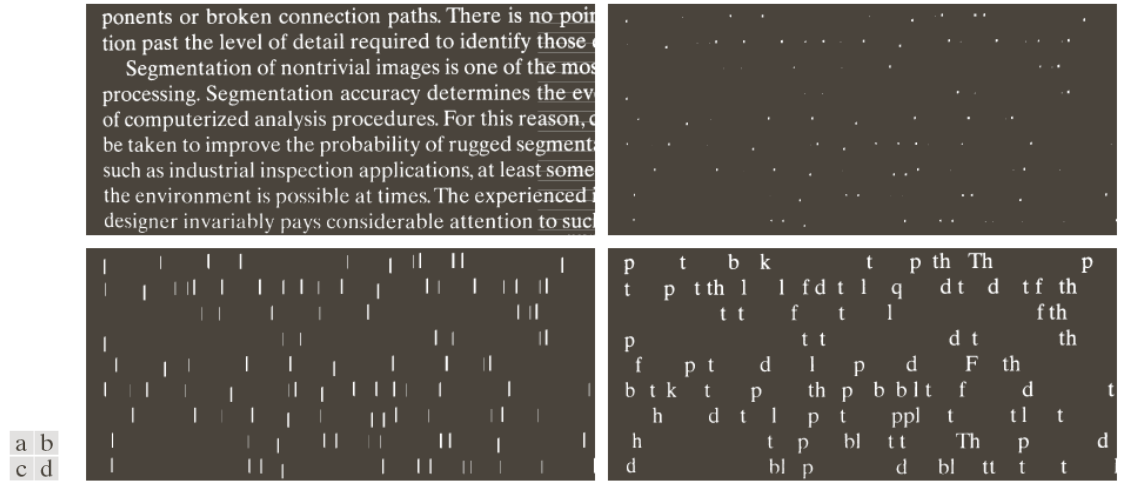


FIGURE 9.29 (a) Text image of size  $918 \times 2018$  pixels. The approximate average height of the tall characters is 50 pixels. (b) Erosion of (a) with a structuring element of size  $51 \times 1$  pixels. (c) Opening of (a) with the same structuring element, shown for reference. (d) Result of opening by reconstruction.

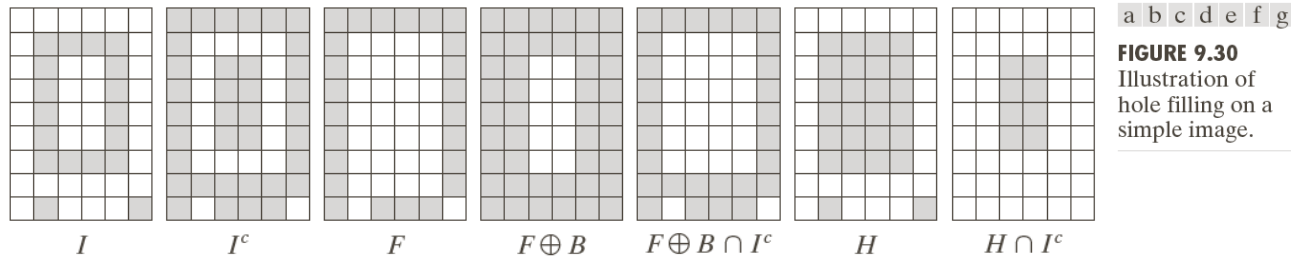
# 9.5.9 形态学重建

## 应用实例

- 填充孔洞：一个基于形态学重建的全自动化过程。令  $I(x, y)$  代表一幅二值图像，并假定我们形成了一幅标记图像  $F$ ，除了在该图像的边界位置为  $1-I$  之外，在其他位置均为0，即

$$F(x, y) = \begin{cases} 1 - I(x, y), & (x, y) \text{ 在 } I \text{ 的边界上} \\ 0, & \text{其他} \end{cases}$$

则  $H = [R_{I^c}^D(F)]^c$  是一幅等于  $I$  且所有孔洞都被填充的二值图像



**FIGURE 9.30**  
Illustration of hole filling on a simple image.

ponents or broken connection paths. There is no position past the level of detail required to identify those components. Segmentation of nontrivial images is one of the most difficult processing. Segmentation accuracy determines the evolution of computerized analysis procedures. For this reason, care should be taken to improve the probability of rugged segmentation, such as industrial inspection applications, at least some of the environment is possible at times. The experienced designer invariably pays considerable attention to such



ponents or broken connection paths. There is no position past the level of detail required to identify those components. Segmentation of nontrivial images is one of the most difficult processing. Segmentation accuracy determines the evolution of computerized analysis procedures. For this reason, care should be taken to improve the probability of rugged segmentation, such as industrial inspection applications, at least some of the environment is possible at times. The experienced designer invariably pays considerable attention to such

ponents or broken connection paths. There is no position past the level of detail required to identify those components. Segmentation of nontrivial images is one of the most difficult processing. Segmentation accuracy determines the evolution of computerized analysis procedures. For this reason, care should be taken to improve the probability of rugged segmentation, such as industrial inspection applications, at least some of the environment is possible at times. The experienced designer invariably pays considerable attention to such

a b  
c d

**FIGURE 9.31**  
(a) Text image of size  $918 \times 2018$  pixels. (b) Complement of (a) for use as a mask image. (c) Marker image. (d) Result of hole-filling using Eq. (9.5-29).

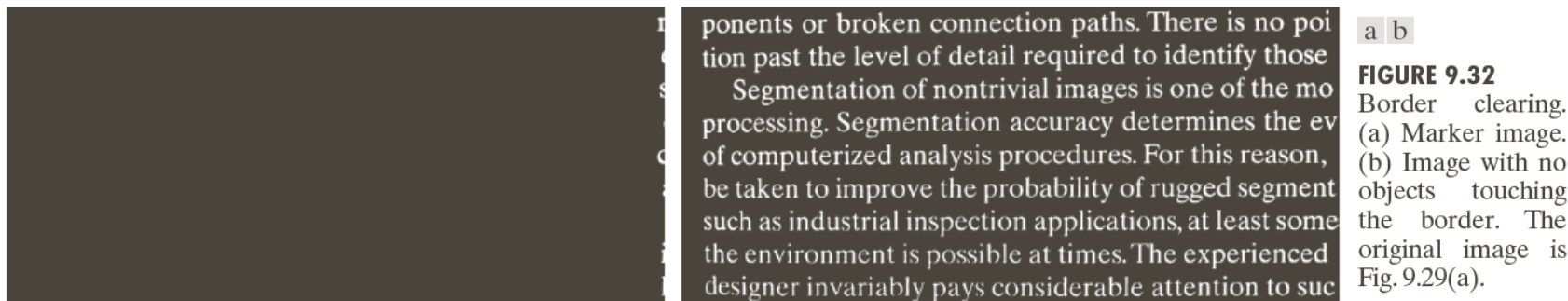
# 9.5.9 形态学重建

## □ 应用实例

- **边界清除：**针对后续形状分析的图像物体提取是自动图像处理的基本任务。删除接触边界的物体的算法是一个很有用的工具，因为：
  - (1) 它可以用于屏蔽图像，以便对进一步处理只保留完整的物体，
  - (2) 它可用作在视野中存在部分物体的一个信号。
- 现在提出一个基于形态学重建的边界清除过程。我们用原图像作为模板，并使用下面的标记图像：

$$F(x, y) = \begin{cases} I(x, y), & (x, y) \text{ 在 } I \text{ 的边界上} \\ 0, & \text{其他} \end{cases}$$

边界清除算法首先计算形态学重建  $R_I^D(F)$ ，然后计算差  $X = I - R_I^D(F)$  以得到一幅其中没有接触边界的物体的图像  $X$ 。



a b

**FIGURE 9.32**  
Border clearing. (a) Marker image. (b) Image with no objects touching the border. The original image is Fig. 9.29(a).



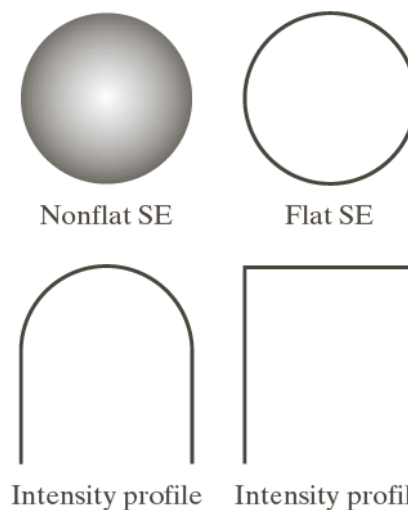
# 第9章 形态学图像处理

- 9.1 预备知识
- 9.2 腐蚀和膨胀
- 9.3 开操作与闭操作
- 9.4 击中与击不中变换
- 9.5 一些基本的形态学算法
- 9.6 灰度级形态学**



# 9.6 灰度级形态学

- 将把膨胀、腐蚀、开闭操作扩展到灰度级图像
  - 处理形如 $f(x, y)$ 和 $b(x, y)$ 的数字图像，其中 $f(x, y)$ 是一幅灰度级图像，而 $b(x, y)$ 是一个结构元。
  - $f$ 和 $b$ 是对每个坐标赋以灰度值（来自实数集合 $R$ 的一个实数）的函数
- 灰度级形态学中的结构元作为一个“探测器”以明确的特性检验一幅给定的图像。
  - 结构元分为两类：非平坦的和平坦的。
  - 实际中灰度级结构元并不常用。
- **本节中所有例子都以高度为1、对称的、平坦的结构元为基础，其原点位于中心处。**
- 结构元的反射的定义为
 
$$\hat{b}(x, y) = b(-x, -y)$$



a b  
c d

**FIGURE 9.34** Nonflat and flat structuring elements, and corresponding horizontal intensity profiles through their center. All examples in this section are based on flat SEs.



## 9.6.1 腐蚀和膨胀

- 当 $b$ 的原点位于 $(x, y)$ 处时，用一个平坦的结构元 $b$ 在 $(x, y)$ 处对图像 $f$ 的**腐蚀**定义为图像 $f$ 中与 $b$ 重合区域的最小值。公式为：

$$[f \ominus b](x, y) = \min_{(s, t) \in b} \{f(x + s, y + t)\}$$

- 为寻求 $b$ 对 $f$ 的腐蚀，我们把结构元的原点放在图像的每个像素的位置。在任何位置的腐蚀由从包含在与 $b$ 重合区域中的 $f$ 的所有值中选取的最小值决定。

- 类似地，当 $\hat{b}$ 的原点位于位置 $(x, y)$ 处时，平坦结构元 $b$ 在任何位置 $(x, y)$ 处对图像 $f$ 的**膨胀**定义为图像 $f$ 中与 $\hat{b}$ 重合区域的最大值即

$$[f \oplus b](x, y) = \max_{(s, t) \in b} \{f(x - s, y - t)\}$$

- 非平坦结构元具有随定义域变化的灰度级， $b_N$ 对图像 $f$ 的腐蚀定义：

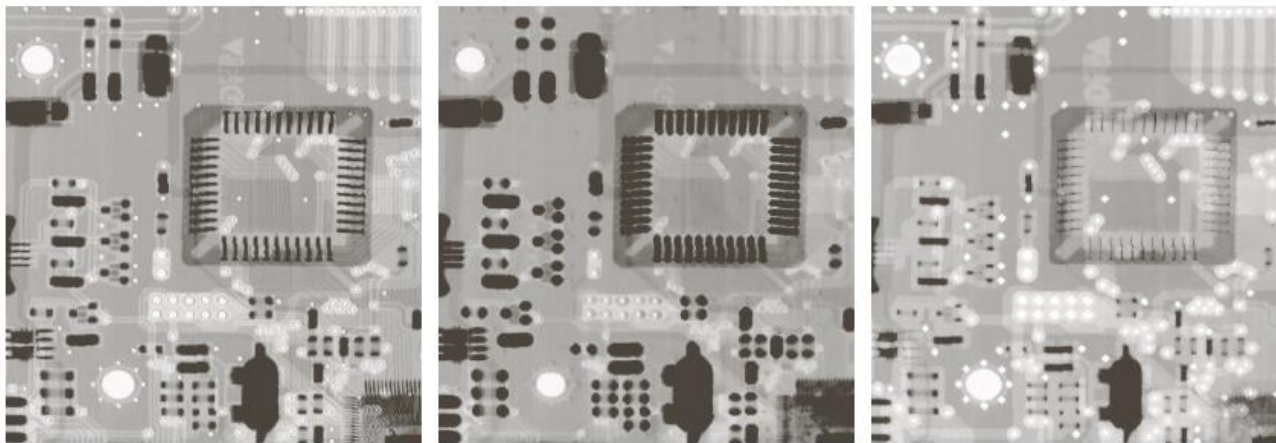
$$[f \ominus b_N](x, y) = \min_{(s, t) \in b_N} \{f(x + s, y + t) - b_N(s, t)\}$$

- 类似地，使用非平坦结构元的膨胀的定义如下：

$$[f \oplus b_N](x, y) = \max_{(s, t) \in b_N} \{f(x - s, y - t) + b_N(s, t)\}$$

## 9.6.1 腐蚀和膨胀

### □ 灰度级腐蚀和膨胀的说明



a b c

**FIGURE 9.35** (a) A gray-scale X-ray image of size  $448 \times 425$  pixels. (b) Erosion using a flat disk SE with a radius of two pixels. (c) Dilation using the same SE. (Original image courtesy of Lixi, Inc.)

### □ 如二值情况一样，腐蚀和膨胀是关于函数的补集和反射对偶的即

$$(f \ominus b)^c(x, y) = (f^c \oplus \hat{b})(x, y)$$

式中， $f^c = -f(x, y)$ 。类似地，

$$(f \oplus b)^c = (f^c \ominus \hat{b})$$

# 9.6.2 开操作和闭操作

- 灰度级图像的开操作和闭操作的表达式，与二值图像的对应操作具有相同的形式。结构元  $b$  对图像  $f$  的开操作表示为  $f \circ b$

$$f \circ b = (f \ominus b) \oplus b$$

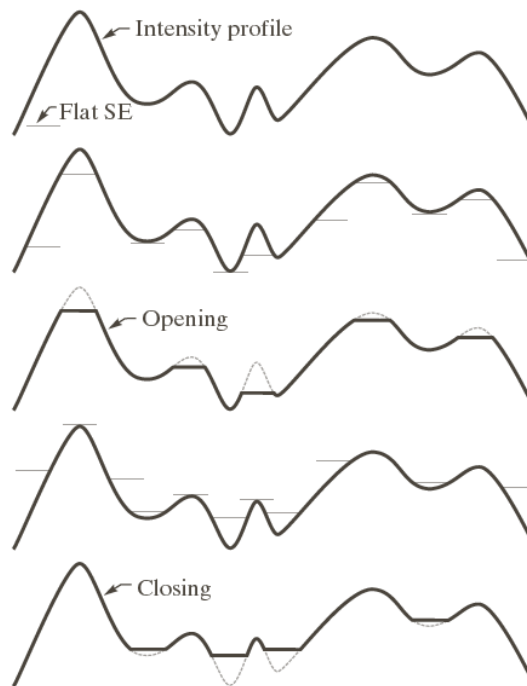
类似地， $b$  对  $f$  的闭操作表示为  $f \cdot b$ ，即  $f \cdot b = (f \oplus b) \ominus b$

- 开闭操作关于函数的补集和结构元的反射是**对偶**的：

$$(f \cdot b)^c = (f^c \circ \hat{b}) \text{ 和 } (f \circ b)^c = (f^c \cdot \hat{b})$$

- 几何解释

将一个图像函数视为一个三维表面， $b$  对  $f$  的开操作可从几何解释为：在  $b$  的每个原点位置，**开操作** 是当从  $f$  的下表面向上推动结构元时， $b$  的任何部分所达到的最高值。**闭操作** 通过寻找结构元的任何部分所能到达的最低点构建。

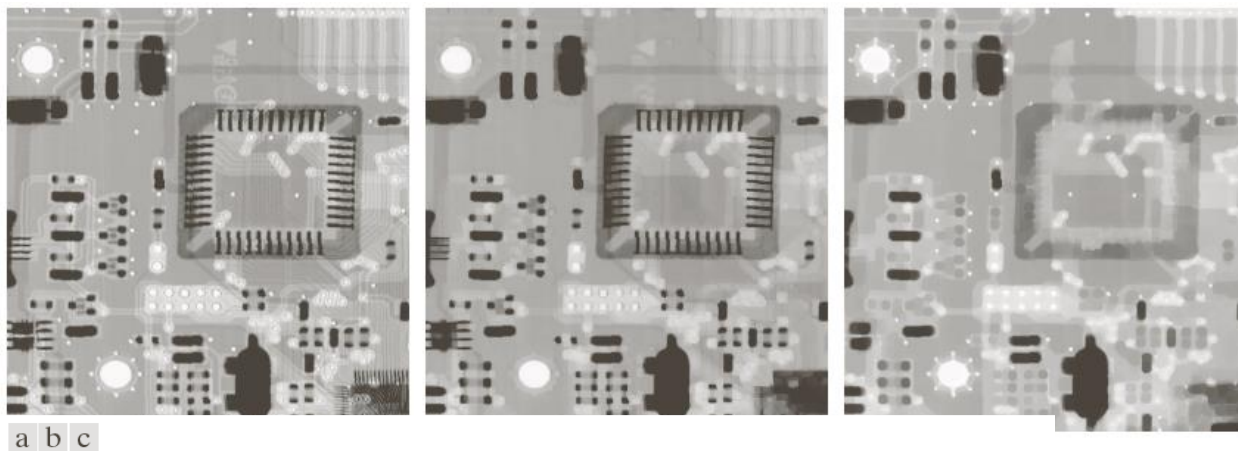


a  
b  
c  
d  
e

**FIGURE 9.36** Opening and closing in one dimension. (a) Original 1-D signal. (b) Flat structuring element pushed up underneath the signal. (c) Opening. (d) Flat structuring element pushed down along the top of the signal. (e) Closing.

## 9.6.2 开操作和闭操作

- 开操作用于去除**较小的明亮细节**，而保持**整体灰度级**和**较大的明亮特征**相对不变
- 闭操作**抑制暗细节**
- 灰度级开操作和闭操作的说明
  - 开操作：所有亮特征的灰度都降低了，降低的程度取决于这些特征相对于结构元的尺寸
  - 与腐蚀的结果不同，开操作对图像的暗特征影响可忽略不计
  - 闭操作：亮细节和背景相对来说未受影响，但削弱了暗特征，削弱的程度取决于这些特征相对于结构元的尺寸

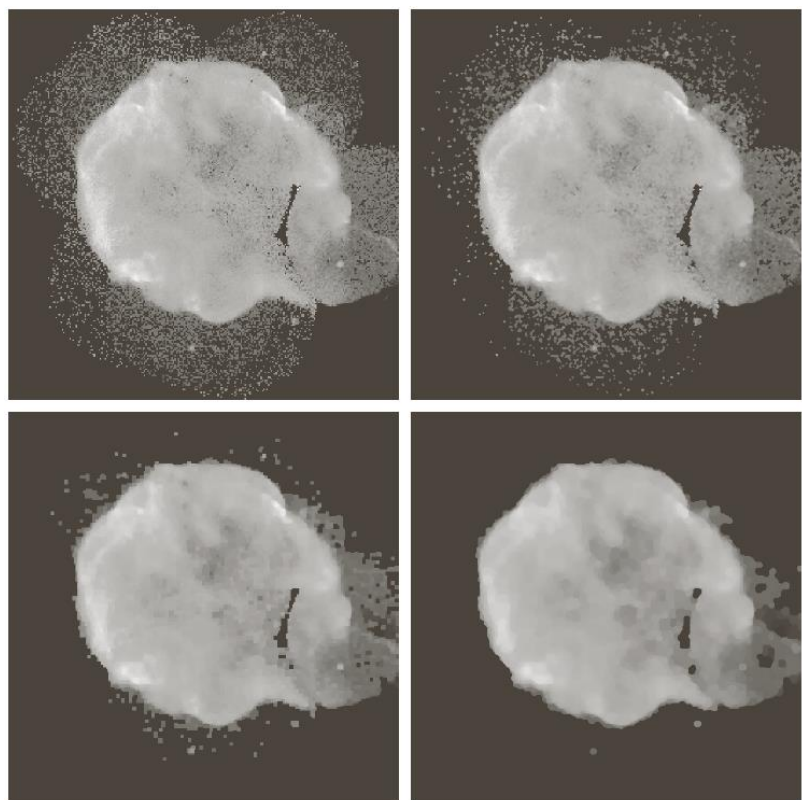


**FIGURE 9.37** (a) A gray-scale X-ray image of size  $448 \times 425$  pixels. (b) Opening using a disk SE with a radius of 3 pixels. (c) Closing using an SE of radius 5.

# 9.6.3 一些基本的灰度级形态学算法

## □ 形态学平滑

- 因为开操作抑制比结构元小的亮细节，而闭操作抑制暗细节，所以它们常常以形态滤波的形式结合起来**平滑图像**和**去除噪声**。
- 假设下图中心的亮区域是我们感兴趣的目标，而较小的分量是噪声。我们的目的是去除噪声。



a	b
c	d

**FIGURE 9.38**  
(a)  $566 \times 566$  image of the Cygnus Loop supernova, taken in the X-ray band by NASA's Hubble Telescope. (b)–(d) Results of performing opening and closing sequences on the original image with disk structuring elements of radii, 1, 3, and 5, respectively. (Original image courtesy of NASA.)



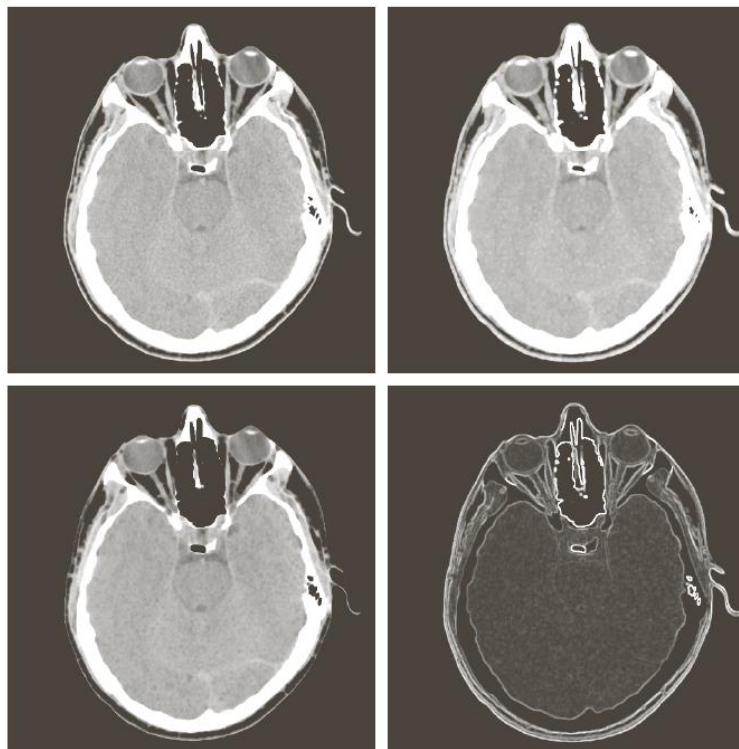
# 9.6.3 一些基本的灰度级形态学算法

## □ 形态学梯度

- 膨胀和腐蚀可与图像相减结合起来得到一幅图像的形态学梯度，由g定义：

$$g = (f \oplus b) - (f \ominus b)$$

- 膨胀粗化一幅图像中的区域，而腐蚀则细化它们。**膨胀和腐蚀之差强调区域间的边界**。同质区域不受影响，因此相减操作趋于消除同质区域。
- 最终结果是边缘被增强而同质区域的贡献则被抑制的图像，因此产生了“类似于微分”效果。



a b  
c d

**FIGURE 9.39**  
(a)  $512 \times 512$  image of a head CT scan.  
(b) Dilation.  
(c) Erosion.  
(d) Morphological gradient, computed as the difference between (b) and (c).  
(Original image courtesy of Dr. David R. Pickens, Vanderbilt University.)





## 9.6.3 一些基本的灰度级形态学算法

### □ 顶帽变换和底帽变换

- 图像相减与开操作和闭操作相结合，可产生所谓的**Top-hat**（**顶帽**）变换和**Bottom-hat**（**底帽**）变换。

- 灰度级图像 $f$ 的**顶帽变换**定义为 $f$ 减去其开操作：

$$T_{hat}(f) = f - (f \circ b)$$

类似地， $f$ 的**底帽变换**定义为 $f$ 的闭操作减去 $f$ ：

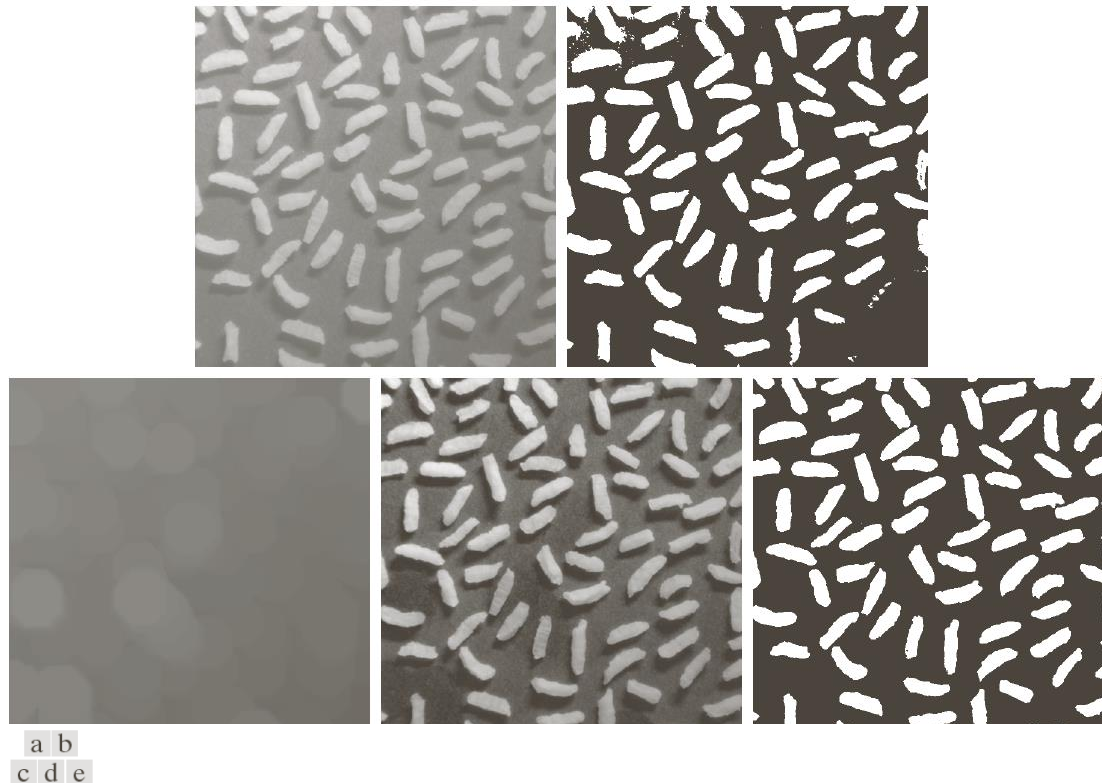
$$B_{hat}(f) = (f \cdot b) - f$$

- 这些变换的主要应用之一是：用一个结构元通过开操作或闭操作从一幅图像中删除物体；然后，差操作得到一幅仅保留已删除分量的图像。
- 顶帽变换用于暗背景上的亮物体，而底帽变换则用于相反的情况。由于这一原因，这些变换又称为**白顶帽变换**和**黑底帽变换**
- 顶帽变换的一个重要用途是**校正不均匀光照的影响**。合适（均匀）的光照在从背景中提取目标的处理中扮演核心的角色。

## 9.6.3 一些基本的灰度级形态学算法

### □ 顶帽变换和底帽变换

- 下图是在非均匀光照下得到的，如图像底部及最右侧的暗色区域就是明证。非均匀光照的最终结果导致了暗区域的分割错误（一些米粒未从背景中提取出来），且在图像的左上角，背景部分被错误地分类了

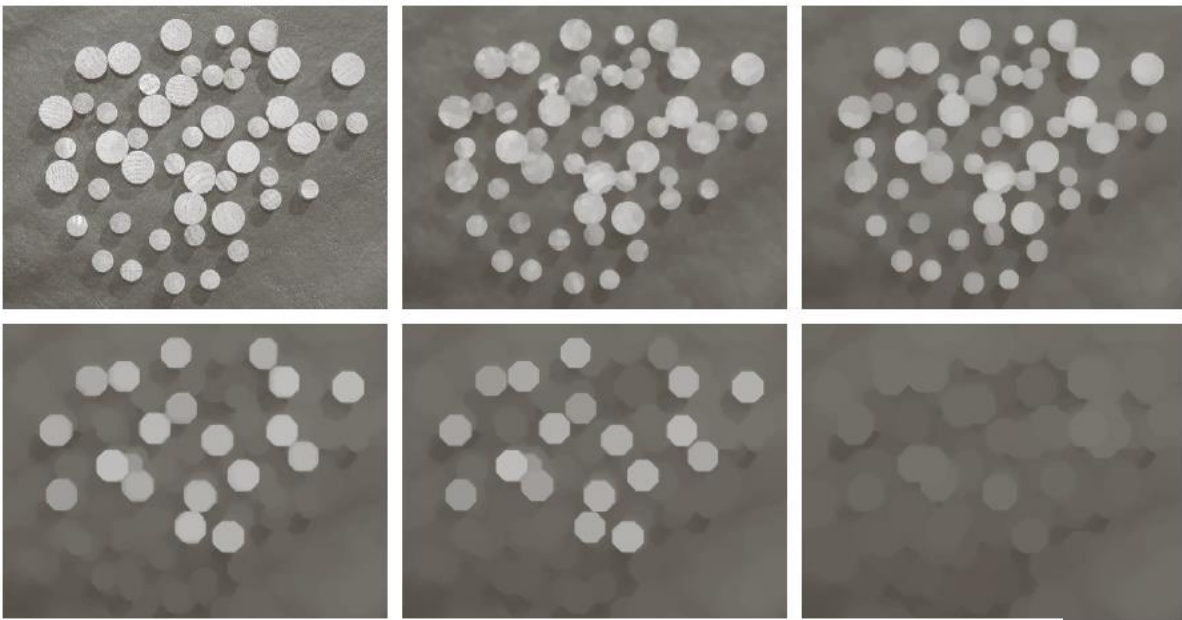


**FIGURE 9.40** Using the top-hat transformation for *shading correction*. (a) Original image of size  $600 \times 600$  pixels. (b) Thresholded image. (c) Image opened using a disk SE of radius 40. (d) Top-hat transformation (the image minus its opening). (e) Thresholded top-hat image.

# 9.6.3 一些基本的灰度级形态学算法

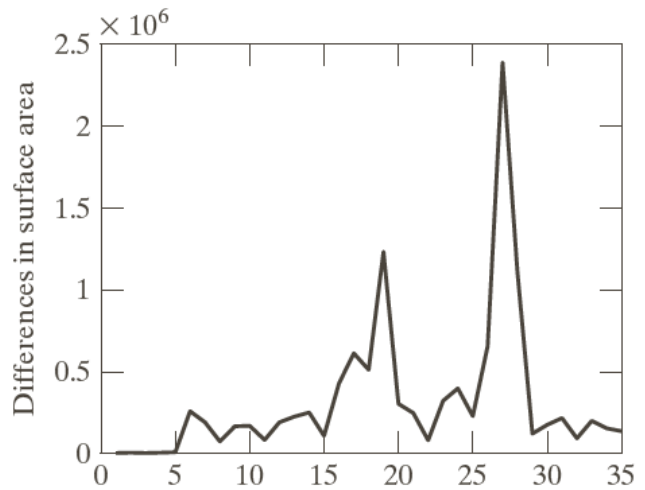
## □ 粒度测定

- 左图是两种不同大小的木钉图像
- 右图显示了该差值阵列的曲线，图中有两个明显的峰值，表面图像存在两种主要的物体尺寸



a b c  
d e f

**FIGURE 9.41** (a)  $531 \times 675$  image of wood dowels. (b) Smoothed image. (c)–(f) Openings of (b) with disks of radii equal to 10, 20, 25, and 30 pixels, respectively. (Original image courtesy of Dr. Steve Eddins, The MathWorks, Inc.)

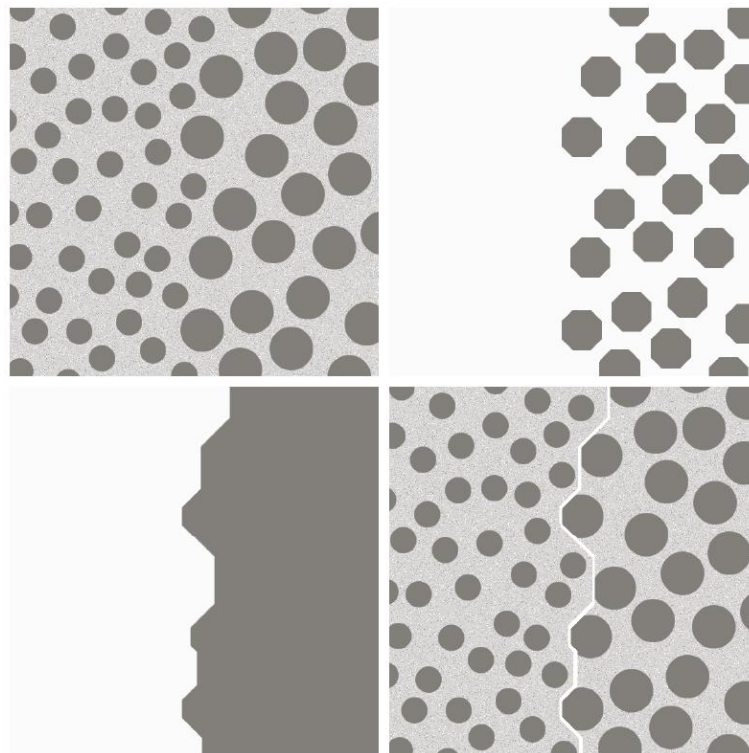


**FIGURE 9.42** Differences in surface area as a function of SE disk radius,  $r$ . The two peaks are indicative of two dominant particle sizes in the image.

## 9.6.3 一些基本的灰度级形态学算法

### □ 纹理分割

- 右图有两个纹理区域：右边大斑点组成的区域和左边小斑点组成的区域。目的是以纹理内容为基础找到两个区域的边界
- 对图执行闭操作后删除了小斑点
- 然后执行开操作后删除了大斑点间的亮间隔区域
- 再对图像执行形态学梯度操作，得到两个区域的边界
- 将边界叠加到原图像上



a b  
c d

**FIGURE 9.43**  
Textural segmentation. (a) A  $600 \times 600$  image consisting of two types of blobs. (b) Image with small blobs removed by closing (a). (c) Image with light patches between large blobs removed by opening (b). (d) Original image with boundary between the two regions in (c) superimposed. The boundary was obtained using a morphological gradient operation.



# 9.6.4 灰度级形态学重建

- 灰度级形态学重建基本上与二值图像所介绍的相同方法来定义。令 $f$ 和 $g$ 分别代表**标记图像**和**模板图像**。我们假设 $f$ 和 $g$ 是大小相同的灰度级图像，且 $f \leq g$ 。 $f$ 关于 $g$ 的大小为1的测地膨胀定义为：

$$D_g^{(1)}(f) = (f \oplus b) \wedge g$$

式中， $\wedge$ 代表点方式的最小算子。该式指出，大小为1的测地膨胀是先计算 $b$ 对 $f$ 的膨胀，**然后选择在每个 $(x, y)$ 点处该结果和 $g$ 间的最小者**。 $f$ 关于 $g$ 的大小为 $n$ 的测地膨胀定义为：

$$D_g^{(n)}(f) = D_g^{(1)}[D_g^{(n-1)}(f)]$$

并有 $D_g^{(0)}(f) = f$

类似地， $f$ 关于 $g$ 的大小为1的测地腐蚀定义为

$$E_g^{(1)}(f) = (f \ominus b) \vee g$$

式中， $\vee$ 代表点方式的最大算子。 $f$ 关于 $g$ 的大小为 $n$ 的测地腐蚀定义为

$$E_g^{(n)}(f) = E_g^{(1)}[E_g^{(n-1)}(f)]$$

并有 $E_g^{(0)}(f) = f$





# 9.6.4 灰度级形态学重建

- 灰度级标记图像 $f$ 对灰度模板图像 $g$ 的**膨胀形态学重建**，定义为 $f$ 关于 $g$ 的测地膨胀反复迭代，直至达到稳定；即

$$R_g^D(f) = D_g^{(k)}(f)$$

且 $k$ 应使 $D_g^{(k)}(f) = D_g^{(k+1)}(f)$ 。 $f$ 对 $g$ 的**腐蚀形态学重建**定义为

$$R_g^E(f) = E_g^{(k)}(f)$$

且 $k$ 应使 $E_g^{(k)}(f) = E_g^{(k+1)}(f)$ 。

- 如二值情况那样，灰度级图像重建的开操作首先腐蚀输入图像，并用它作为标记图像。图像 $f$ 的大小为 $n$ 的**重建开操作**定义为，先对 $f$ 进行大小为 $n$ 的腐蚀，再由 $f$ 的膨胀重建；即

$$O_R^{(n)}(f) = R_f^D[(f \ominus nb)]$$

式中， $(f \ominus nb)$ 表示 $b$ 对 $f$ 的 $n$ 次腐蚀。类似地，图像 $f$ 的大小为 $n$ 的**重建闭操作**定义为，先对 $f$ 进行大小为 $n$ 的膨胀，再由 $f$ 的腐蚀重建即

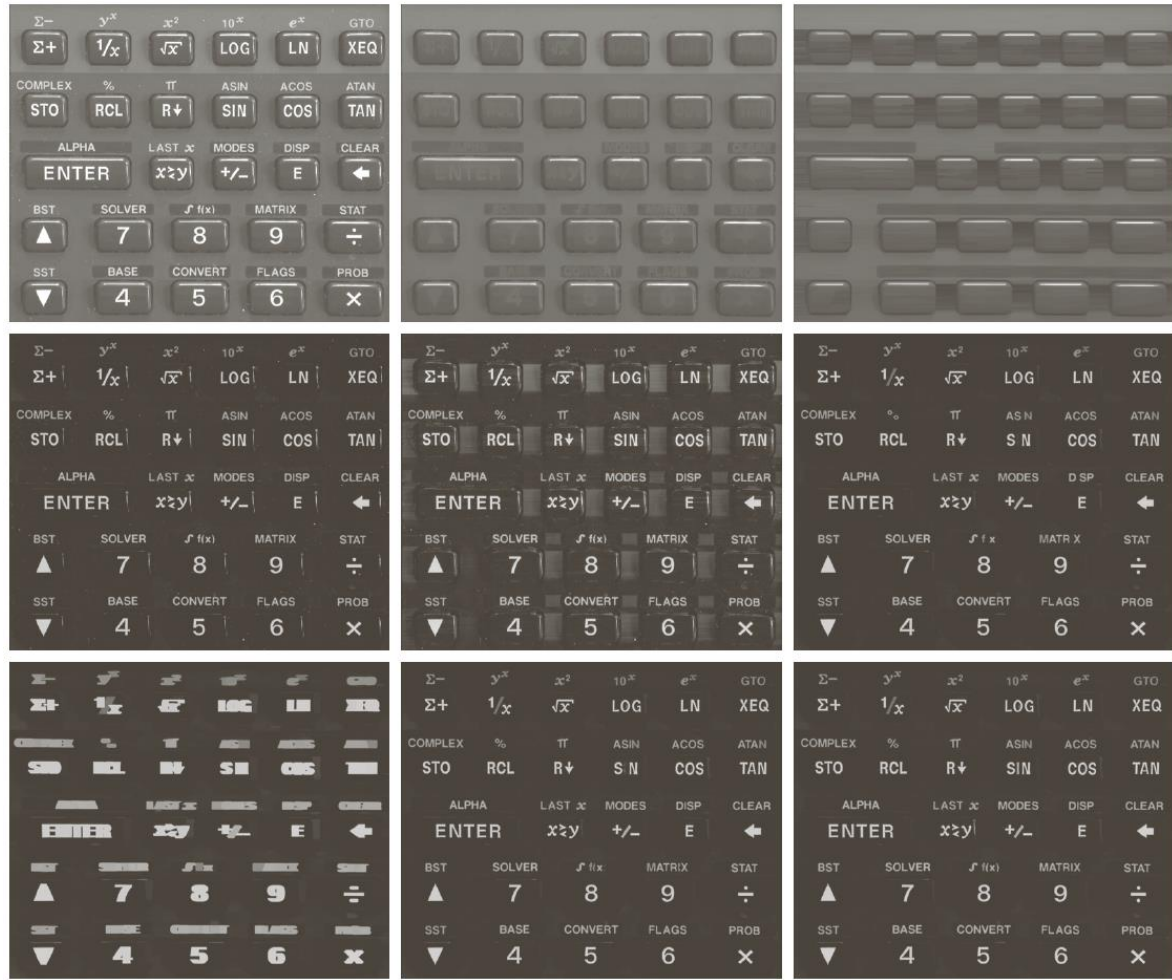
$$C_R^{(n)}(f) = R_f^E[(f \oplus nb)]$$

式中， $(f \oplus nb)$ 表示 $b$ 对 $f$ 的 $n$ 次膨胀。

# 9.6.4 灰度级形态学重建

- 最后，如下例所示，称为重建顶帽的一种有用的技术是**从一幅图像中减去其重建开操作**
- 使用形态学重建展平复杂的背景
  - 目的是仅留下恒定灰度背景上的正文

a	b	c
d	e	f
g	h	i



**FIGURE 9.44** (a) Original image of size 1134 × 1360 pixels. (b) Opening by reconstruction of (a) using a horizontal line 71 pixels long in the erosion. (c) Opening of (a) using the same line. (d) Top-hat by reconstruction. (e) Top-hat. (f) Opening by reconstruction of (d) using a horizontal line 11 pixels long. (g) Dilation of (f) using a horizontal line 21 pixels long. (h) Minimum of (d) and (g). (i) Final reconstruction result. (Images courtesy of Dr. Steve Eddins, The MathWorks, Inc.)





# 数字图象处理

---

中国科学技术大学  
电子工程与信息科学系

主讲教师：李厚强 ([lihq@ustc.edu.cn](mailto:lihq@ustc.edu.cn))  
周文罡 ([zhwg@ustc.edu.cn](mailto:zhwg@ustc.edu.cn))

助教：谢乔康 ([xieqiaok@mail.ustc.edu.cn](mailto:xieqiaok@mail.ustc.edu.cn))  
周 浩([zhouh156@mail.ustc.edu.cn](mailto:zhouh156@mail.ustc.edu.cn))



# 第10章 图像分割

- 分割将图像细分为构成它的子区域或物体。
  - 细分的程度取决于要解决的问题。
  - 在应用中，当感兴趣的物体或区域已经被检测出来时，就停止分割
  - 异常图像的分割是图像处理中最困难的任务之一
- 本章中的多数分割算法均基于灰度值的两个基本性质之一：**不连续性和相似性**。
  - 对于不连续性的灰度，方法是以灰度突变为基础分割一幅图像，比如图像的边缘。
  - 对于相似的灰度，主要方法是根据一组预定义的准则把一幅图像分割为相似的区域。阈值处理、区域生长、区域分裂和区域聚合都是这类方法的例子



# 第10章 图像分割

- 10.1 基础知识
- 10.2 点、线和边缘检测
- 10.3 阈值处理
- 10.4 基于区域的分割
- 10.5 用形态学分水岭的分割
- 10.6 分割中运动的应用



# 第10章 图像分割

- 10.1 基础知识
- 10.2 点、线和边缘检测
- 10.3 阈值处理
- 10.4 基于区域的分割
- 10.5 用形态学分水岭的分割
- 10.6 分割中运动的应用



# 10.1 基础知识

□ 令集合 $R$ 代表整个图像区域，对 $R$ 的分割可看做将 $R$ 分成若干个满足下述条件的非空的子集（子区域）

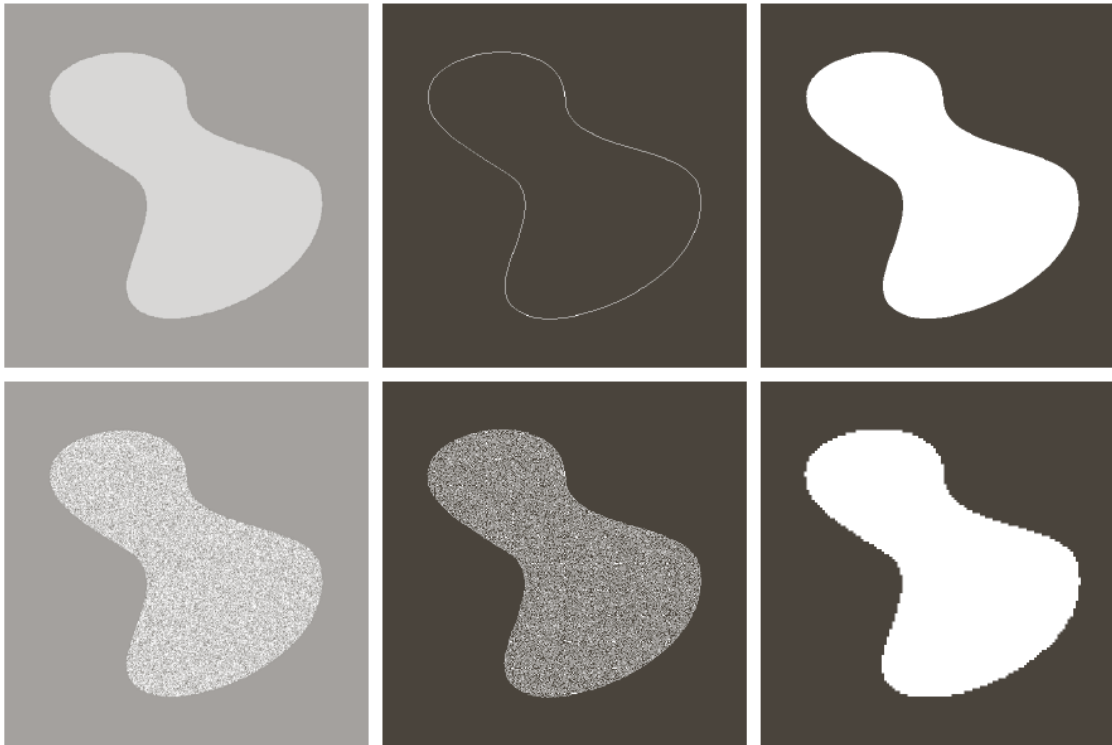
$R_1, R_2, \dots, R_n$ :

- 1、 $\bigcup_{i=1}^n R_i$
- 2、对 $i \neq j$ ，有 $R_i \cap R_j = \emptyset$
- 3、每个子区域 $R_1, R_2, \dots, R_n$ 是连通的
- 4、对于各个子区域，有均匀性测度度量 $P$ 为真。
- 5、对其中任意两个和两个以上相邻子区域之并，其均匀性测度度量 $P$ 为假。即：

$$P(R_i) = \text{TRUE} \text{ 且 } P(R_i \cup R_j) = \text{FALSE}$$

# 10.1 基础知识

□ 下面三幅图像说明了基于区域的分割



a b c  
d e f

**FIGURE 10.1** (a) Image containing a region of constant intensity. (b) Image showing the boundary of the inner region, obtained from intensity discontinuities. (c) Result of segmenting the image into two regions. (d) Image containing a textured region. (e) Result of edge computations. Note the large number of small edges that are connected to the original boundary, making it difficult to find a unique boundary using only edge information. (f) Result of segmentation based on region properties.



# 第10章 图像分割

- 10.1 基础知识
- 10.2 点、线和边缘检测
- 10.3 阈值处理
- 10.4 基于区域的分割
- 10.5 用形态学分水岭的分割
- 10.6 分割中运动的应用





# 10.2 点、线和边缘检测

- 本节介绍以**灰度局部剧烈变化检测**为基础的分割方法。
- 我们感兴趣的三种图像特征是孤立点、线和边缘
- 小节
  - 10.2.1 背景知识
  - 10.2.2 孤立点的检测
  - 10.2.3 线检测
  - 10.2.4 边缘模型
  - 10.2.5 基本边缘检测
  - 10.2.6 更先进的边缘检测技术
  - 10.2.7 边缘连接和边界检测

# 10.2.1 背景知识

- 就像局部平均平滑一幅图像那样，假设平均处理类似于积分，对于灰度的突变，局部变化可以用微分来检测。
- 由于变化非常短促，因此一阶微分和二阶微分特别适合

- 数字函数的导数可用差分定义

- 一维函数 $f(x)$ 在点 $x$ 处的导数的近似：数字差分

$$\frac{\partial f}{\partial x} = f'(x) = f(x + 1) - f(x)$$

- 关于点 $x$ 处的二阶导数的近似：

$$\frac{\partial^2 f}{\partial x^2} = f''(x) = f(x + 1) + f(x - 1) - 2f(x)$$

- 计算图像中每个像素位置处的一阶导数和二阶导数的另一种方法是使用空间滤波器：

$$R = w_1 z_1 + w_2 z_2 + \dots + w_9 z_9 = \sum_{k=1}^9 w_k z_k$$

$w_1$	$w_2$	$w_3$
$w_4$	$w_5$	$w_6$
$w_7$	$w_8$	$w_9$

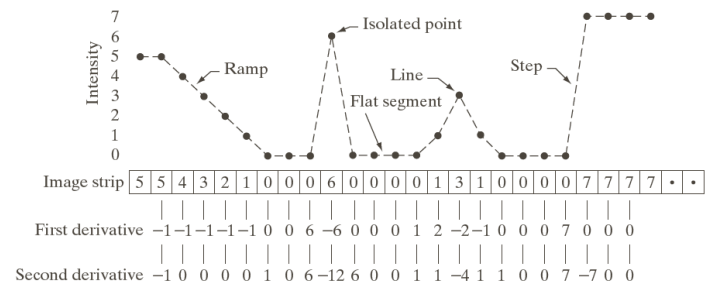
FIGURE 10.3  
A general  $3 \times 3$   
spatial filter mask.

# 10.2.1 背景知识

□ 基于一阶导数和二阶导数检测图像包含有不同的实心物体、一条线和单个噪声点（右图）

□ 结论：

- (1) 一阶导数通常在图像中产生较粗的边缘；
- (2) 二阶导数对精细细节，如细线、孤立点和噪声有较强的响应；
- (3) 二阶导数在灰度斜坡和灰度台阶过渡处会产生双边缘响应；
- (4) 二阶导数的符号可用于确定边缘的过渡是从亮到暗还是从暗到亮



a b  
c

**FIGURE 10.2** (a) Image. (b) Horizontal intensity profile through the center of the image, including the isolated noise point. (c) Simplified profile (the points are joined by dashes for clarity). The image strip corresponds to the intensity profile, and the numbers in the boxes are the intensity values of the dots shown in the profile. The derivatives were obtained using Eqs. (10.2-1) and (10.2-2).

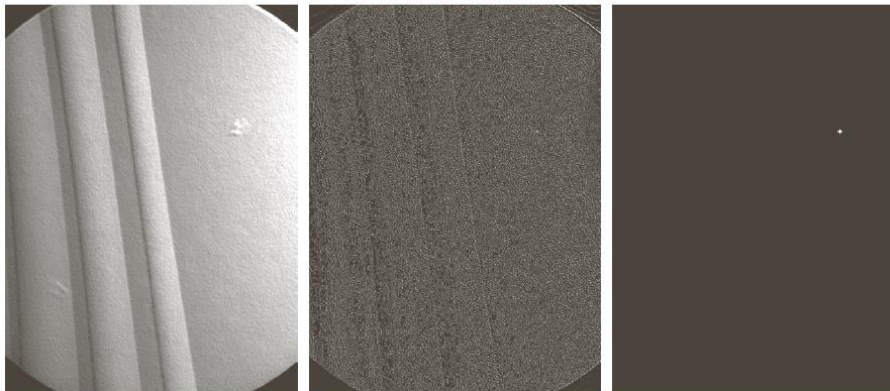
# 10.2.2 孤立点的检测

- 点的检测应以二阶导数为基础，即使用拉普拉斯算子：

$$\nabla^2 f(x, y) = f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1) - 4f(x, y)$$

- 同样，该算子可用模板来实现
- 如果在某个点处，该模板的响应的绝对值超过了指定的阈值，那么在 $(x, y)$ 处的点已被检测到。在输出图像中，这一点标注1

1	1	1
1	-8	1
1	1	1



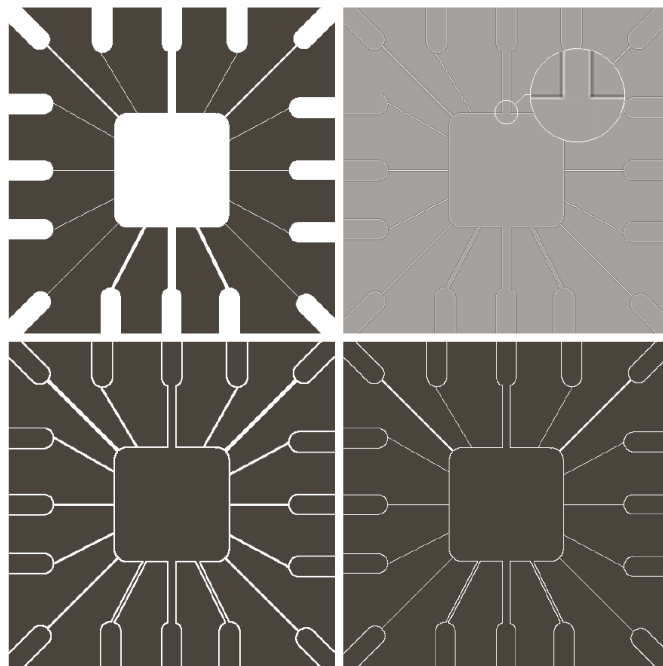
a  
b c d

FIGURE 10.4

(a) Point detection (Laplacian) mask. (b) X-ray image of turbine blade with a porosity. The porosity contains a single black pixel. (c) Result of convolving the mask with the image. (d) Result of using Eq. (10.2-8) showing a single point (the point was enlarged to make it easier to see). (Original image courtesy of X-TEK Systems, Ltd.)

## 10.2.3 线检测

- 对于线检测，二阶导数将导致更强的响应，产生比一阶导数更细的线。我们可以使用拉普拉斯模板，但要处理双线效应
- 例：



a b  
c d

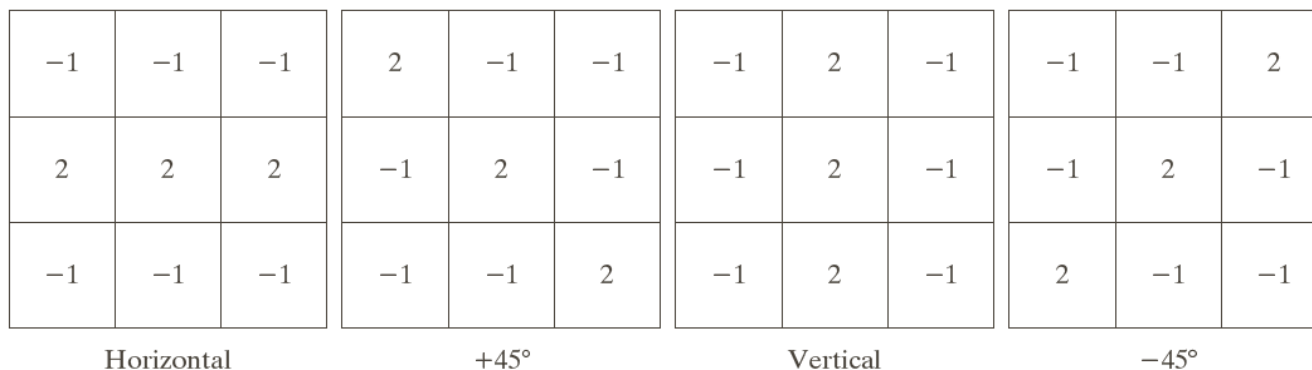
**FIGURE 10.5**

(a) Original image.  
(b) Laplacian image; the magnified section shows the positive/negative double-line effect characteristic of the Laplacian.  
(c) Absolute value of the Laplacian.  
(d) Positive values of the Laplacian.

- 当把 $3 \times 3$ 滤波器居中放在一条宽为5个像素的恒定灰度线上时，其响应将为零，这样就产生了刚才提到的双线效应

# 10.2.3 线检测

- 点检测的拉普拉斯算子是**各向同性**的，因此**其响应与方向无关**（相对于该 $3 \times 3$ 拉普拉斯目标的4个方向：垂直方向、水平方向和两个对角方向）。
- 通常，我们的兴趣在于检测**特定方向的线**。

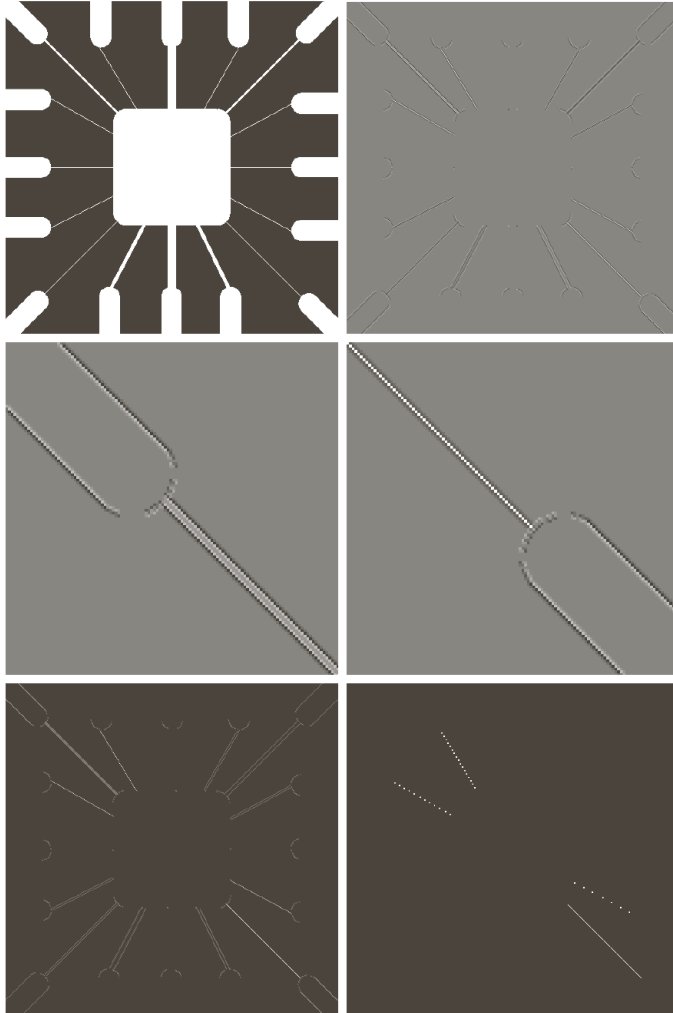


**FIGURE 10.6** Line detection masks. Angles are with respect to the axis system in Fig. 2.18(b).

- 第一个模板对于水平线有最佳响应；第二个模板对于45度方向的线有最佳响应；第三个模板对于垂直线有最佳响应；第四个模板对于-45度方向的线有最佳响应
- 每个模板的首选方向用一个比其他方向更大的系数（如2）加权。每个模板的系数之和为0，这表明**恒定灰度区域中的响应为0**

# 10.2.3 线检测

## □ 特定方向线的检测



a b  
c d  
e f

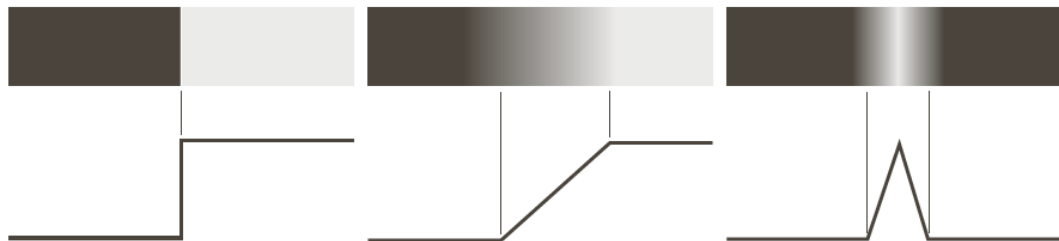
**FIGURE 10.7**

(a) Image of a wire-bond template. (b) Result of processing with the  $+45^\circ$  line detector mask in Fig. 10.6. (c) Zoomed view of the top left region of (b). (d) Zoomed view of the bottom right region of (b). (e) The image in (b) with all negative values set to zero. (f) All points (in white) whose values satisfied the condition  $g \geq T$ , where  $g$  is the image in (e). (The points in (f) were enlarged to make them easier to see.)



## 10.2.4 边缘模型

- 边缘检测是基于灰度突变来分割图像的常用方法，本节先介绍一些边缘建模的方法
- 边缘模型根据它们的灰度剖面来分类。有台阶边缘、斜坡边缘和“屋顶”边缘等
- 下图显示了这些边缘模型和相应的灰度剖面



a b c

**FIGURE 10.8**  
From left to right, models (ideal representations) of a step, a ramp, and a roof edge, and their corresponding intensity profiles.

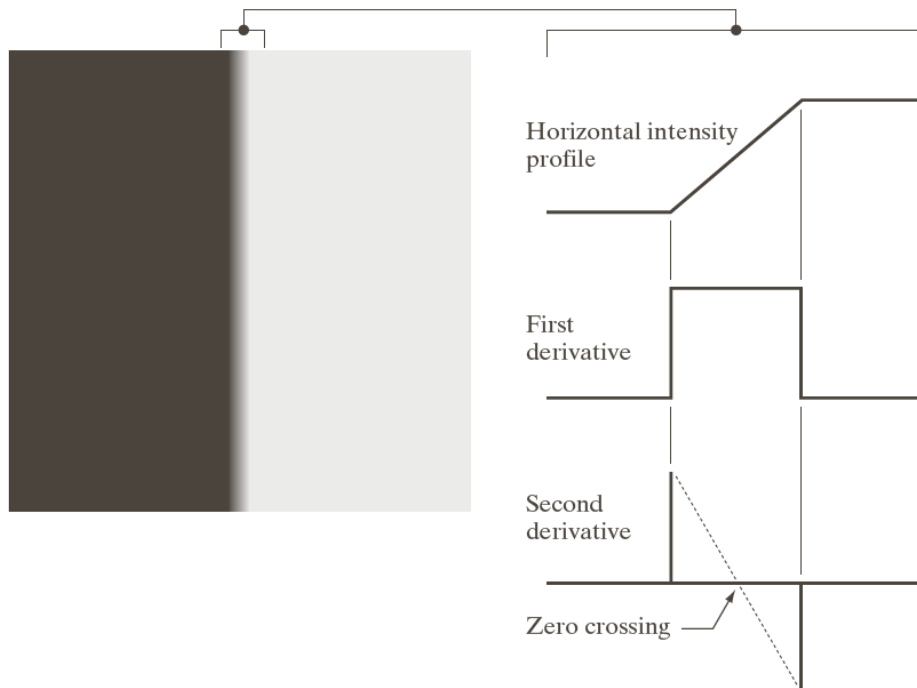
# 10.2.4 边缘模型

## □ 下图结论

- 一阶导数的**幅度**可用于检测图像中的某个点处是否存在一个边缘
- 二阶导数的**符号**用于确定一个边缘像素是位于该边缘的暗侧还是位于该边缘的亮侧

## □ 围绕一条边缘的二阶导数的两个附加性质：

- 对图像中的每条边缘，二阶导数生成两个值（一个不希望的特点）
- 二阶导数的**零交叉点**可用于定位粗边缘的中心



a b

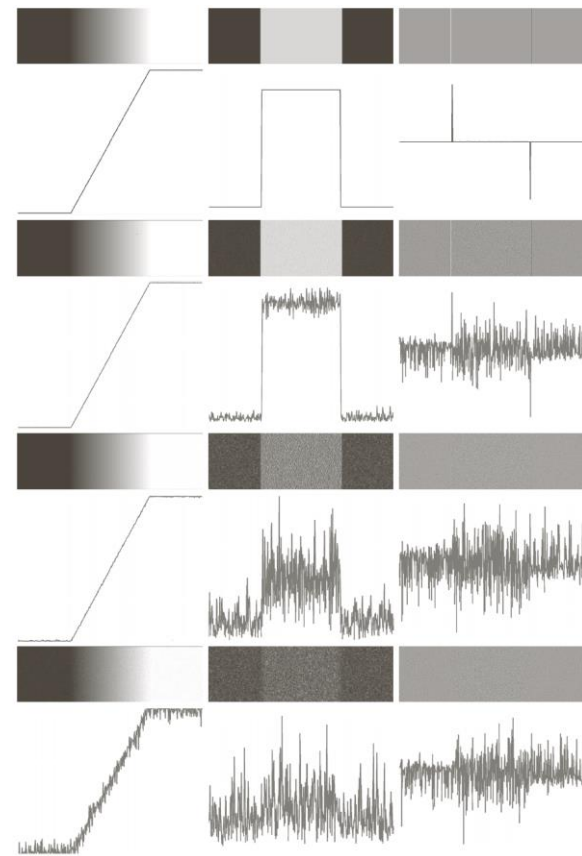
**FIGURE 10.10**

(a) Two regions of constant intensity separated by an ideal vertical ramp edge.

(b) Detail near the edge, showing a horizontal intensity profile, together with its first and second derivatives.

# 10.2.4 边缘模型

- 微弱的可见噪声严重影响检测边缘所用的两个关键导数
- 因此执行边缘检测的三个基本步骤是：
  - (1) 为降噪图像进行平滑处理
  - (2) 边缘点的检测。这是一个局部操作，从一幅图像中提取所有边缘点的潜在候选者
  - (3) 边缘定位。这一步的目的是从候选边缘点中选择组成边缘点集合的真实成员



**FIGURE 10.11** First column: Images and intensity profiles of a ramp edge corrupted by random Gaussian noise of zero mean and standard deviations of 0.0, 0.1, 1.0, and 10.0 intensity levels, respectively. Second column: First-derivative images and intensity profiles. Third column: Second-derivative images and intensity profiles.

# 10.2.5 基本边缘检测

## □ 图像梯度及其性质

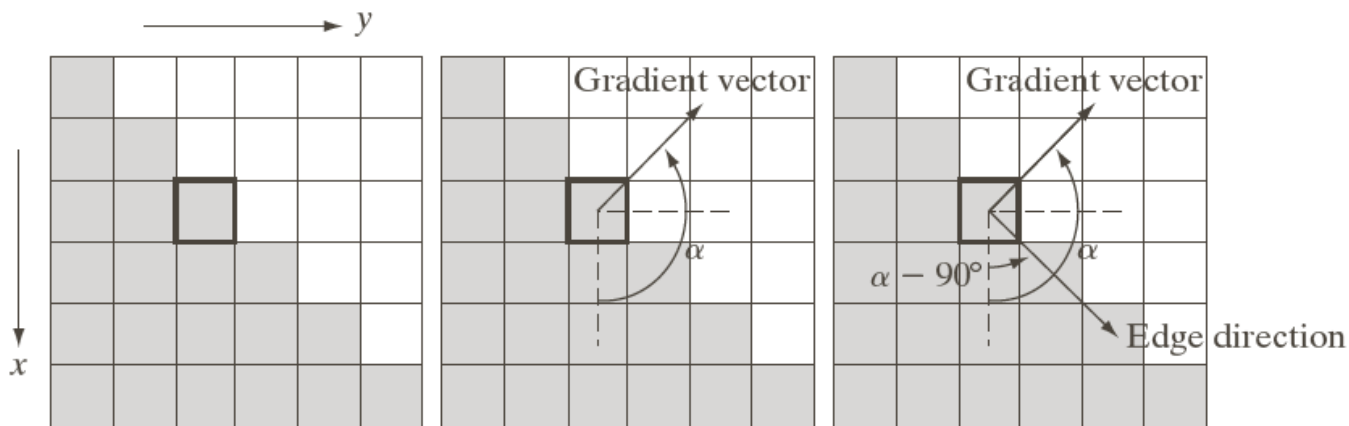
- 梯度向量、幅度和方向角

$$\nabla f(x, y) = [G_x \quad G_y]^T = \left[ \frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y} \right]^T$$

$$\text{mag}(\nabla f) = [G_x^2 + G_y^2]^{1/2}$$

$$\phi(x, y) = \arctan(G_y / G_x)$$

- 图像边缘方向与边缘上点的梯度向量正交



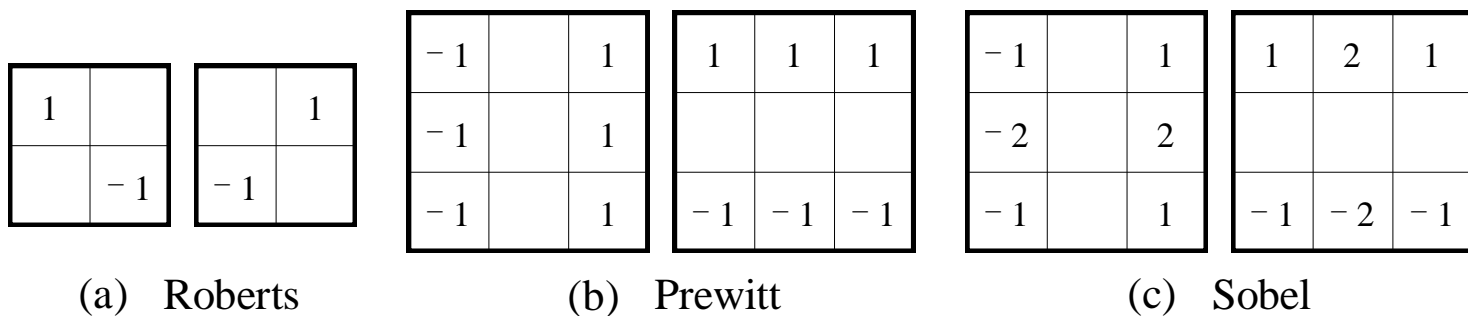
a b c

**FIGURE 10.12** Using the gradient to determine edge strength and direction at a point. Note that the edge is perpendicular to the direction of the gradient vector at the point where the gradient is computed. Each square in the figure represents one pixel.

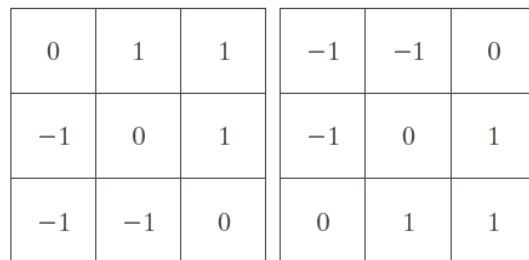
# 10.2.5 基本边缘检测

## □ 梯度算子

- 导数用模板对图像进行滤波来实现

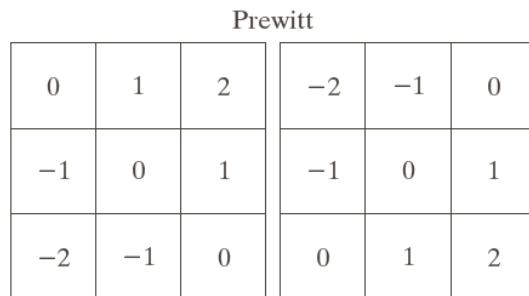


- 用绝对值来近似梯度的幅值



a b  
c d

**FIGURE 10.15**  
Prewitt and Sobel masks for detecting diagonal edges.



Sobel

# 10.2.5 基本边缘检测

## □ 梯度算子



a	b
c	d

**FIGURE 10.16**

(a) Original image of size  $834 \times 1114$  pixels, with intensity values scaled to the range  $[0, 1]$ .  
(b)  $|g_x|$ , the component of the gradient in the  $x$ -direction, obtained using the Sobel mask in Fig. 10.14(f) to filter the image.  
(c)  $|g_y|$ , obtained using the mask in Fig. 10.14(g).  
(d) The gradient image,  $|g_x| + |g_y|$ .

# 10.2.5 基本边缘检测

## □ 梯度算子

- 精细的细节在边缘检测中通常是不符合需要的，因为它往往表现为噪声，导致计算会增强这种噪声，使主要边缘的检测变得复杂。
- 减少精细细节的一个方法是对图像进行平滑处理，如下图



a b  
c d

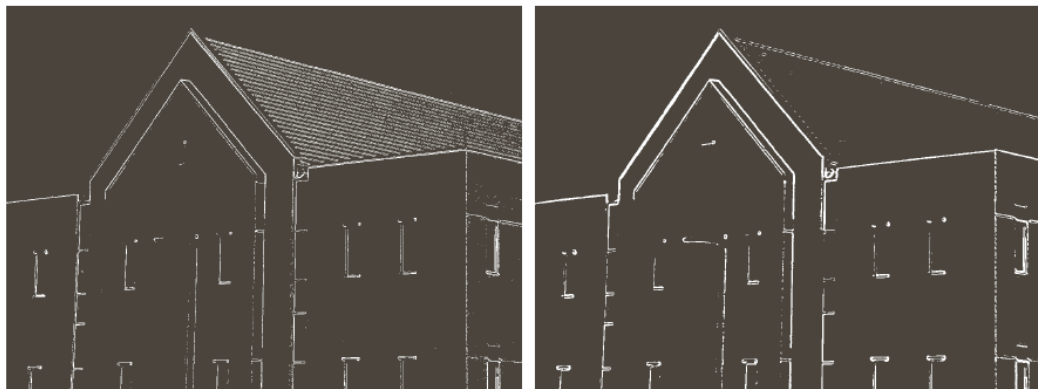
**FIGURE 10.18** Same sequence as in Fig. 10.16, but with the original image smoothed using a  $5 \times 5$  averaging filter prior to edge detection.



# 10.2.5 基本边缘检测

## □ 与阈值处理相结合的梯度

- 在计算梯度前对图像进行平滑处理，边缘检测可做更多的选择。实现相同基本目标的另一种方法是，对**梯度图像进行阈值处理**。
- 阈值处理后的图像边缘更少，并且图像中的边缘要清晰得多。另一方面，许多边缘被断开了
- 当目的是突出主要边缘并尽可能保持连接时，实践中通常既做平滑处理也做阈值处理



a b

**FIGURE 10.20** (a) Thresholded version of the image in Fig. 10.16(d), with the threshold selected as 33% of the highest value in the image; this threshold was just high enough to eliminate most of the brick edges in the gradient image. (b) Thresholded version of the image in Fig. 10.18(d), obtained using a threshold equal to 33% of the highest value in that image.

# 10.2.6 更先进的边缘检测

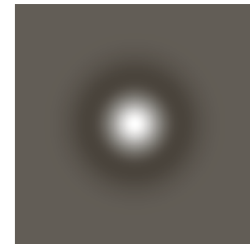
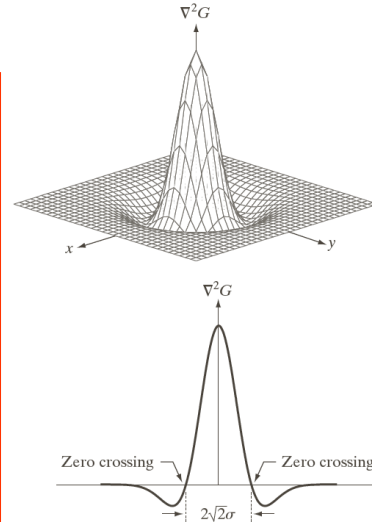
- 用于边缘检测的算子应有两个特点
  - 它应该是一个能计算图像中每一个点处的一阶导数或二阶导数的数字近似的微分算子
  - 它应能被“调整”以便在**任何期望的尺寸**上起作用，因此大算子也可用于检测模糊边缘，小算子可用于检测锐度集中的精细细节
- Marr-Hildreth边缘检测器（**马尔算子**）
  - 满足上述条件的算子是 $\nabla^2 h$ ：高斯拉普拉斯（LoG）
  - 下图显示了一个LoG的负函数的三维图、图像和剖面

$$h(x, y) = \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad g(x, y) = h(x, y) \otimes f(x, y)$$

$$\nabla_g^2 = \nabla^2[h(x, y) \otimes f(x, y)] = \nabla^2 h(x, y) \otimes f(x, y)$$

$$= \left(\frac{r^2 - \sigma^2}{\sigma^4}\right) \exp\left(-\frac{r^2}{2\sigma^2}\right) \otimes f(x, y)$$

$$\nabla^2 h = h''(r) = \left(\frac{r^2 - \sigma^2}{\sigma^4}\right) \exp\left(-\frac{r^2}{2\sigma^2}\right) \quad (LOG)$$



0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

a b  
c d

**FIGURE 10.21**  
 (a) Three-dimensional plot of the negative of the LoG. (b) Negative of the LoG displayed as an image. (c) Cross section of (a) showing zero crossings. (d)  $5 \times 5$  mask approximation to the shape in (a). The negative of this mask would be used in practice.

# 10.2.6 更先进的边缘检测

## □ Marr-Hildreth边缘检测器（马尔算子）

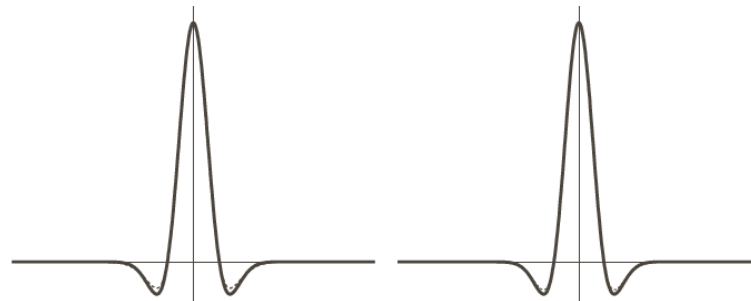
- (1) 算子的高斯部分会模糊图像，从而在尺寸上将结构的灰度（包括噪声）降低到远小于 $\sigma$ 的程度。
- (2) 关于 $\nabla^2$ ：尽管一阶导数可用于检测灰度突变，但它们是有方向的算子。另一方面，拉普拉斯有各向同性的优点，符合人的视觉系统特性

## □ Marr-Hildreth边缘检测算法可小结如下：

- 用一个2-D的高斯平滑模板与源图象卷积
- 计算卷积后图象的拉普拉斯值
- 检测拉普拉斯图象中的过零点作为边缘点

## □ 使用高斯差分（DoG）来近似LoG滤波器

$$\text{DoG}(x, y) = \frac{1}{2\pi\sigma_1^2} e^{-\frac{x^2+y^2}{2\sigma_1^2}} - \frac{1}{2\pi\sigma_2^2} e^{-\frac{x^2+y^2}{2\sigma_2^2}}$$

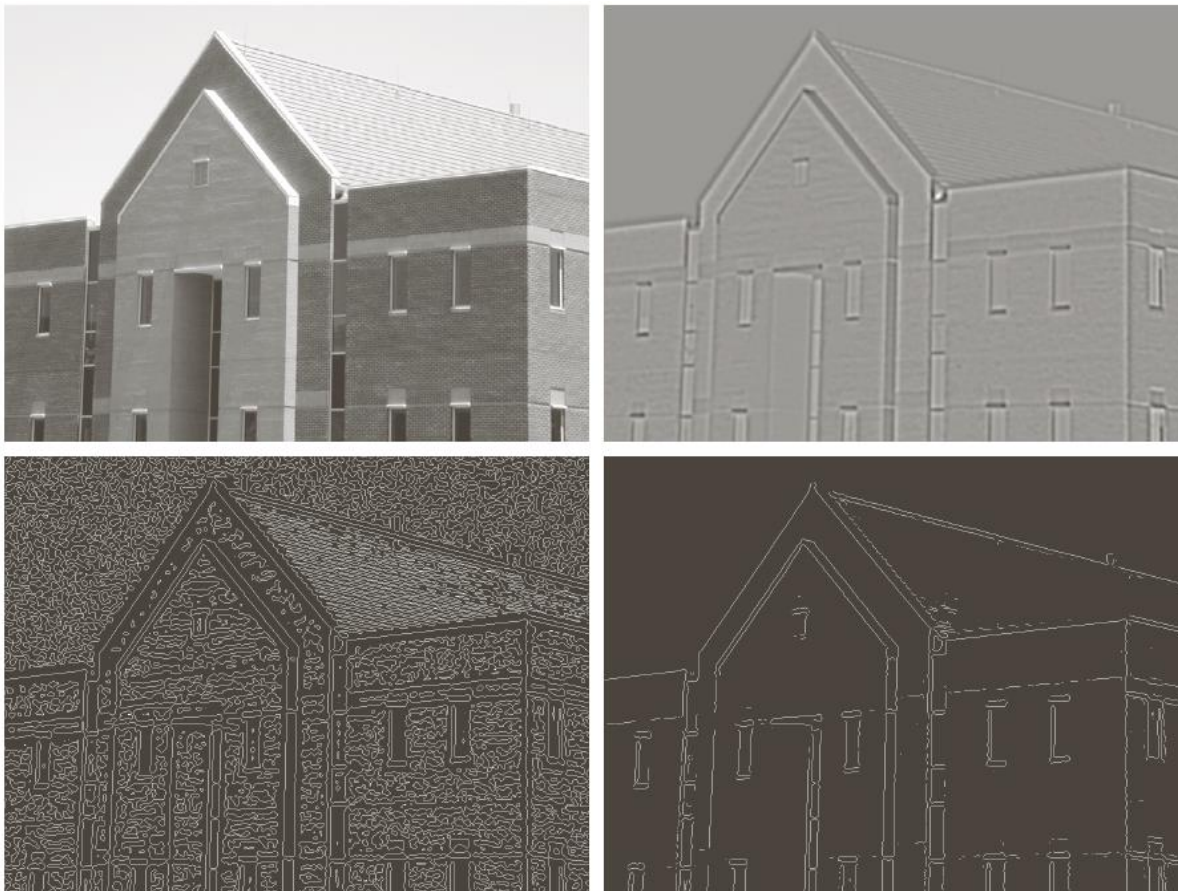


a b

**FIGURE 10.23**  
(a) Negatives of the LoG (solid) and DoG (dotted) profiles using a standard deviation ratio of 1.75:1.  
(b) Profiles obtained using a ratio of 1.6:1.

# 10.2.6 更先进的边缘检测

## □ Marr-Hildreth边缘检测算法说明



a b  
c d

**FIGURE 10.22**

(a) Original image of size  $834 \times 1114$  pixels, with intensity values scaled to the range  $[0, 1]$ . (b) Results of Steps 1 and 2 of the Marr-Hildreth algorithm using  $\sigma = 4$  and  $n = 25$ . (c) Zero crossings of (b) using a threshold of 0 (note the closed-loop edges). (d) Zero crossings found using a threshold equal to 4% of the maximum value of the image in (b). Note the thin edges.



# 10.2.6 更先进的边缘检测

## □ 坎尼边缘检测器

- 最优边缘检测算子，坎尼方法基于三个基本目标：
  - (1) 低错误率。既要少将真正的边缘丢失也要少将非边缘判为边缘
  - (2) 边缘点应被很好地定位。检测出的边缘应在真正的边界上
  - (3) 单一的边缘响应。得到的边界为单像素宽

## □ 坎尼边缘检测算法的基本步骤总结如下：

- (1) 用一个高斯滤波器平滑输入图像
- (2) 计算梯度幅值图像和角度图像
- (3) 对梯度幅值图像应用非最大抑制
- (4) 用双阈值处理和连接分析来检测并连接边缘

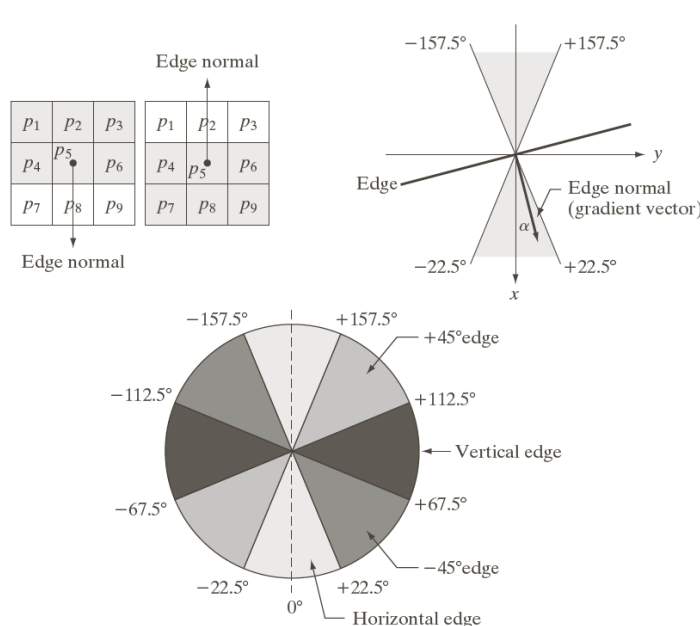
# 10.2.6 更先进的边缘检测

## □ 坎尼边缘检测器（高斯滤波平滑）

- 令  $f(x, y)$  表示输入图像， $G(x, y)$  表示高斯函数： $G(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$
- 用  $G$  和  $f$  的卷积形成一幅平滑后的图像  $f_s(x, y) = G(x, y) \star f(x, y)$
- 计算梯度幅度  $M(x, y) = \sqrt{g_x^2 + g_y^2}$  和方向  $\alpha(x, y) = \arctan[\frac{g_y}{g_x}]$

## □ 坎尼边缘检测器（非最大抑制）

- 把所有可能的边缘方向量化为四个方向：水平、垂直、 $+45^\circ$  和  $-45^\circ$



**FIGURE 10.24** (a) Two possible orientations of a horizontal edge (in gray) in a  $3 \times 3$  neighborhood. (b) Range of values (in gray) of  $\alpha$ , the direction angle of the edge normal, for a horizontal edge. (c) The angle ranges of the edge normals for the four types of edge directions in a  $3 \times 3$  neighborhood. Each edge direction has two ranges, shown in corresponding shades of gray.





# 10.2.6 更先进的边缘检测

## □ 坎尼边缘检测器（非最大抑制）

- 令 $d_1, d_2, d_3$ 和 $d_4$ 表示刚才讨论的 $3 \times 3$ 区域的四个基本边缘方向。对于 $\alpha(x, y)$ 中以每点 $(x, y)$ 为中心的 $3 \times 3$ 区域，有如下非最大抑制方案：
  - ✓ 寻找最接近 $\alpha(x, y)$ 的方向 $d_k$
  - ✓ 若 $M(x, y)$ 的值至少小于沿 $d_k$ 的两个邻居之一，则令 $g_N(x, y) = 0$ （抑制）；否则，令 $g_N(x, y) = M(x, y)$ ，这里 $g_N(x, y)$ 为非最大抑制后的图像。

## □ 坎尼边缘检测器（双阈值处理）

- 对 $g_N(x, y)$ 进行阈值处理，以便减少伪边缘点
- 坎尼算法使用两个阈值：一个低阈值 $T_L$ 和一个高阈值 $T_H$ 。坎尼建议，高阈值和低阈值的比率应为2:1或3:1
- 将阈值操作想象为创建两幅附加的图像
$$g_{NH}(x, y) = g_N(x, y) \geq T_H \quad g_{NL}(x, y) = g_N(x, y) \geq T_L$$
- 通过令 $g_{NL}(x, y) = g_{NL}(x, y) - g_{NH}(x, y)$ ，我们从 $g_{NL}(x, y)$ 中删除所有来自 $g_{NH}(x, y)$ 的非零像素。 $g_{NH}(x, y)$ 和 $g_{NL}(x, y)$ 中的非零像素可分别视为“强”和“弱”边缘像素
- 阈值处理后， $g_{NH}(x, y)$ 中的所有像素均被假设为有效的边缘像素，并被立即标记。取决于 $T_H$ 的值， $g_{NH}(x, y)$ 中的边缘通常会存在缝隙





## 10.2.6 更先进的边缘检测

- 坎尼边缘检测器（双阈值处理）
  - 较长的边缘用下列步骤形成：
    - (a) 在 $g_{NH}(x, y)$ 中定位下一个未被访问的边缘像素 $p$
    - (b) 在 $g_{NL}(x, y)$ 中将所有弱像素标记为有效边缘像素，用8连通的连接方法连接到 $p$
    - (c) 若 $g_{NH}(x, y)$ 中的所有非零像素已被访问，则调到步骤（d），否则返回步骤（a）
    - (d) 将 $g_{NL}(x, y)$ 中未标记为有效边缘像素的所有像素置零
- 最后，将来自 $g_{NL}(x, y)$ 的所有非零像素附加到 $g_{NH}(x, y)$ ，用坎尼算子形成最终的输出图像
- 尽管非最大抑制后的边缘比原来的梯度边缘细，但仍保持了比1像素粗的边缘。为得到1像素粗的边缘，通常要在最后一步执行一次边缘细化算法
- 坎尼算法改进性能所付出的代价是，它要比早些时候讨论的两种方法实现起来**更复杂，同时执行时间更长**。在某些应用中，成本和速度需求通常要求我们使用更简单的技术，主要是阈值梯度方法。
- 当关注的主要是边缘质量是，通常会使用Marr-Hildreth算法和坎尼算法，特别是坎尼算法

# 10.2.6 更先进的边缘检测

## □ 坎尼边缘检测方法说明



a b  
c d

**FIGURE 10.25**

(a) Original image of size  $834 \times 1114$  pixels, with intensity values scaled to the range  $[0, 1]$ .

(b) Thresholded gradient of smoothed image.

(c) Image obtained using the Marr-Hildreth algorithm.

(d) Image obtained using the Canny algorithm.

Note the significant improvement of the Canny image compared to the other two.



## 10.2.7 边缘连接和边界检测

- 理想情况下，边缘检测应该仅产生位于边缘上的像素集合。实际上，由于噪声、不均匀照明引起的**边缘间断**，以及其他引入灰度值虚假的不连续的影响，这些像素并不能完全描述边缘特性。
- 因此，一般是在边缘检测后紧跟连接算法，将边缘像素组合成有意义的边缘或区域边界。
- 本节讨论三种基本的边缘连接方法
  - 第一种方法需要有关局部区域中的边缘点（如一个 $3 \times 3$ 邻域）的知识；
  - 第二种方法要求区域边界上的点已知；
  - 第三种方法是处理整个边缘图像的全局方法。



# 10.2.7 边缘连接和边界检测

## □ 局部处理

- 在每个点  $(x, y)$  处的小邻域内分析像素的特点，根据预定的准则，将所有的相似点连接起来，以形成根据指定准则满足相同特性像素的一条边缘
- 一种特别适合于实时应用的简化方法由如下步骤组成：

- (1) 计算输入图像  $f(x, y)$  的梯度幅度阵列  $M(x, y)$  和角度阵列  $\alpha(x, y)$
- (2) 形成一幅二值图像  $g$ ，任何坐标对  $(x, y)$  处的值由下式给出：

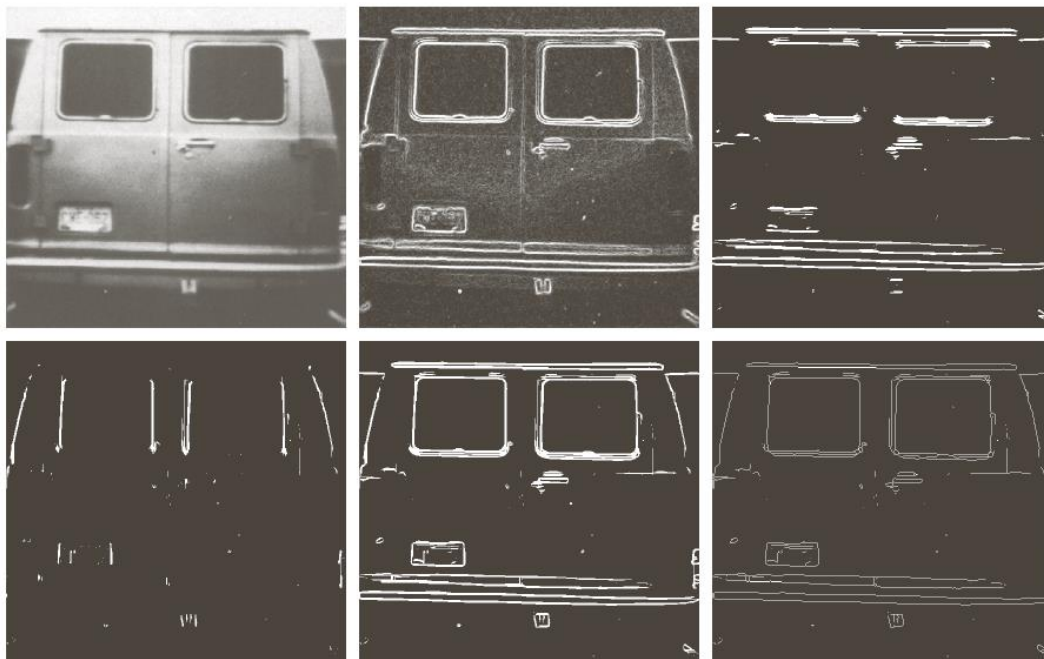
$$g(x, y) = \begin{cases} 1, & M(x, y) > T_M \text{ 且 } \alpha(x, y) = A \pm T_A \\ 0, & \text{其他} \end{cases}$$

式中， $T_M$  是一个阈值， $A$  是一个指定的角度方向， $\pm T_A$  定义了一个关于  $A$  的可接受方向“带宽”

- (3) 扫描  $g$  的行，并在不超过指定长度  $K$  的每一行中填充（置1）所有缝隙（0的集合）。注意，按照定义，缝隙一定要限制在一个1或多个1的两端。分别处理各行，它们之间没有记忆
- (4) 在任何其他方向  $\theta$  上检测缝隙，以该角度选择  $g$ ，并应用步骤3中的水平扫描过程。然后，将结果以  $-\theta$  旋转回来

# 10.2.7 边缘连接和边界检测

- 使用局部处理的边缘连接



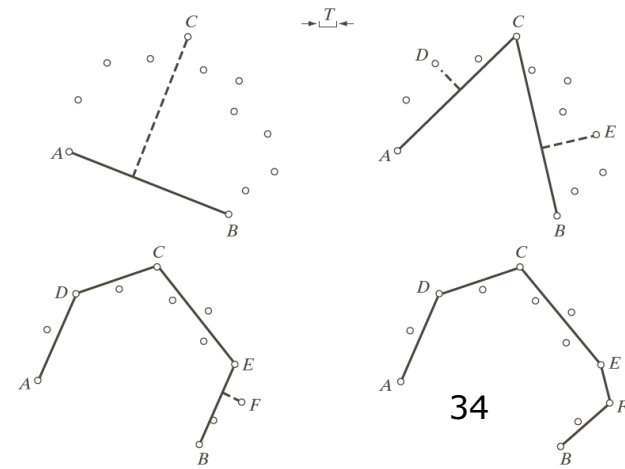
a	b	c
d	e	f

**FIGURE 10.27** (a) A  $534 \times 566$  image of the rear of a vehicle. (b) Gradient magnitude image. (c) Horizontally connected edge pixels. (d) Vertically connected edge pixels. (e) The logical OR of the two preceding images. (f) Final result obtained using morphological thinning. (Original image courtesy of Perceptics Corporation.)

# 10.2.7 边缘连接和边界检测

## □ 区域处理

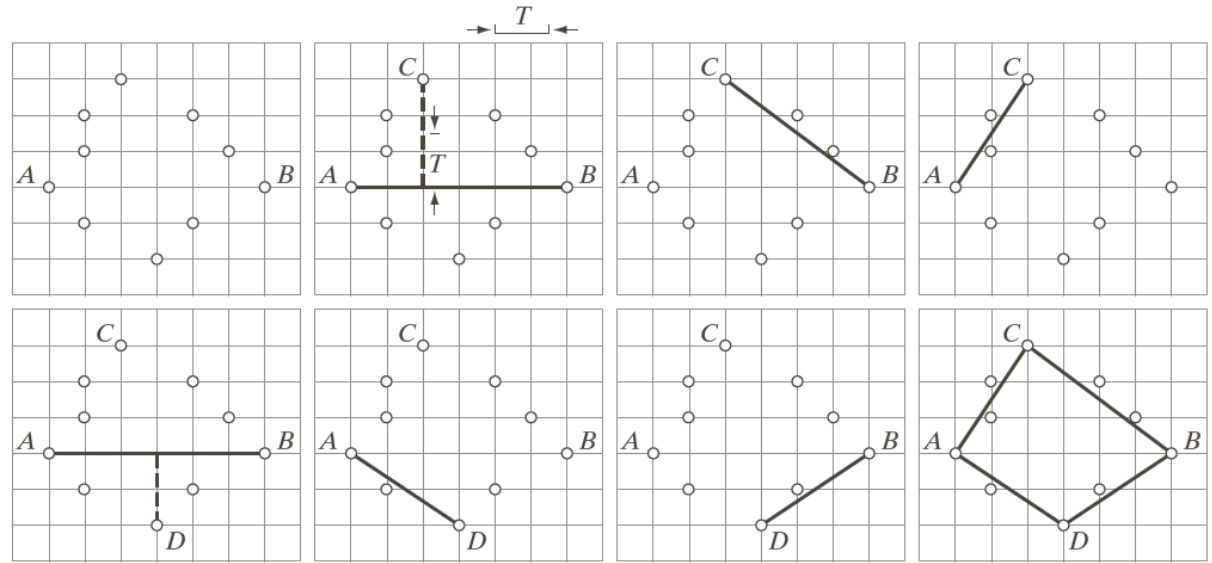
- 对这类处理的一种方法是函数近似，我们对已知点拟合一条二维曲线。我们的兴趣通常在于快速执行的技术，多边形近似尤其有吸引力
- 寻找一个多边形拟合开放和闭合曲线的算法描述如下：
  - ✓ 1、令P是一个已排序序列，显然，这些点是一幅二值图像中的1值点。指定两个起始点A和B。他们是多边形的两个起始顶点
  - ✓ 2、指定一个阈值T，以及两个空堆栈“开”（OPEN）和“闭”（CLOSED）
  - ✓ 3、若P中的点对应于一条闭合曲线，则把A放到“开”中，并把B放到“开”和“闭”中。若点对应于一条开放曲线，则把A放到“开”中，而把B放到“闭”中
  - ✓ 4、计算从“闭”中最后一个顶点到“开”中最后一个顶点的线的参数
  - ✓ 5、计算步骤4所得的直线至P中所有点的距离，序列把它们放到步骤4所得的两个顶点之间。选择具有最大距离 $D_{max}$ 的点 $V_{max}$ （解决任意性问题）
  - ✓ 若 $D_{max} > T$ ，则把 $V_{max}$ 作为一个新顶点放在“开”的末尾。转到步骤4
  - ✓ 否则，从“开”中移除最后一个顶点，并把它作为“闭”的最后一个顶点插入
  - ✓ 若“开”非空，转到步骤4
  - ✓ 否则，退出。“闭”中顶点就是拟合P中的点的多边形的顶点





# 10.2.7 边缘连接和边界检测

□ 用多边形近似连接边缘



a	b	c	d
e	f	g	h

**FIGURE 10.29** (a) A set of points in a clockwise path (the points labeled *A* and *B* were chosen as the starting vertices). (b) The distance from point *C* to the line passing through *A* and *B* is the largest of all the points between *A* and *B* and also passed the threshold test, so *C* is a new vertex. (d)–(g) Various stages of the algorithm. (h) The final vertices, shown connected with straight lines to form a polygon. Table 10.1 shows step-by-step details.

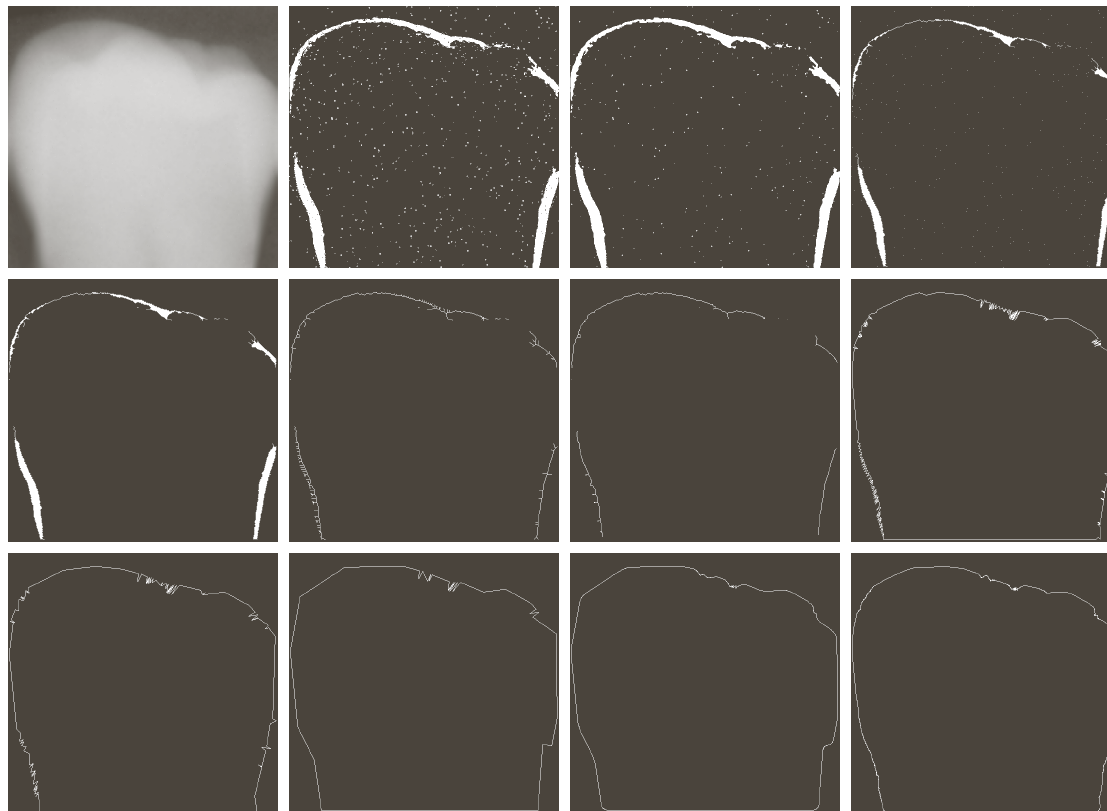
CLOSED	OPEN	Curve segment processed	Vertex generated
<i>B</i>	<i>B, A</i>	–	<i>A, B</i>
<i>B</i>	<i>B, A</i>	( <i>BA</i> )	<i>C</i>
<i>B</i>	<i>B, A, C</i>	( <i>BC</i> )	–
<i>B, C</i>	<i>B, A</i>	( <i>CA</i> )	–
<i>B, C, A</i>	<i>B</i>	( <i>AB</i> )	<i>D</i>
<i>B, C, A</i>	<i>B, D</i>	( <i>AD</i> )	–
<i>B, C, A, D</i>	<i>B</i>	( <i>DB</i> )	–
<i>B, C, A, D, B</i>	Empty	–	–

**TABLE 10.1**  
Step-by-step details of the mechanics in Example 10.11.



# 10.2.7 边缘连接和边界检测

## □ 一幅图像的边界的多边形拟合

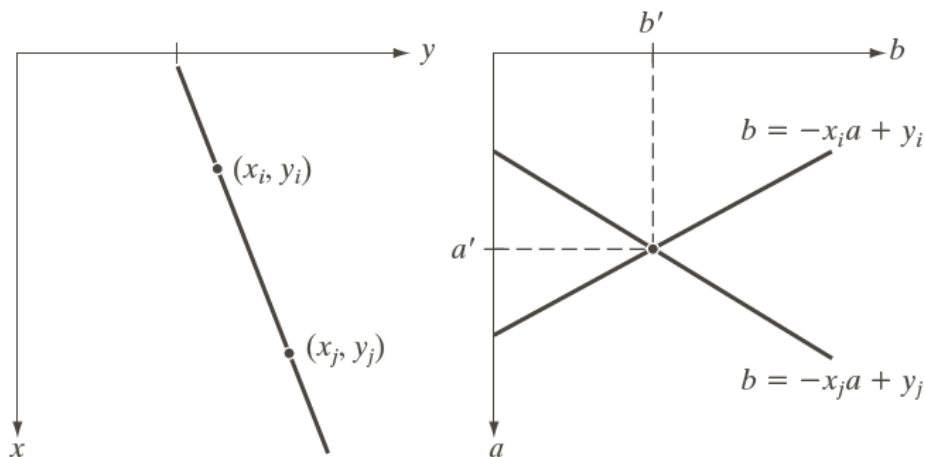


**FIGURE 10.30** (a) A  $550 \times 566$  X-ray image of a human tooth. (b) Gradient image. (c) Result of majority filtering. (d) Result of morphological shrinking. (e) Result of morphological cleaning. (f) Skeleton. (g) Spur reduction. (h)–(j) Polygonal fit using thresholds of approximately 0.5%, 1%, and 2% of image width ( $T = 3, 6, \text{ and } 12$ ). (k) Boundary in (j) smoothed with a 1-D averaging filter of size  $1 \times 31$  (approximately 5% of image width). (l) Boundary in (h) smoothed with the same filter.

# 10.2.7 边缘连接和边界检测

## □ 使用霍夫变换的全局处理

- 检测像素是否位于指定形状的曲线上。一旦检测到，这些曲线就会形成边缘或感兴趣的区域边界
- 例如：给定一幅图像中的 $n$ 个点，假设我们希望找到这些点中一个位于直线上的子集。一种可行的解决方法是：先找到所有由每对点确定的直线，然后寻找靠近特定直线的点的所有子集。这种方法涉及寻找  $\frac{n(n-1)}{2} \sim n^2$  条直线，然后对每个点，与所有直线执行  $\frac{(n)(n(n-1))}{2} \sim n^3$  次比较。这种方法计算量太大而没有应用价值
- 霍夫变换是一种替代方法：从图像的 $xy$ 平面到参数空间



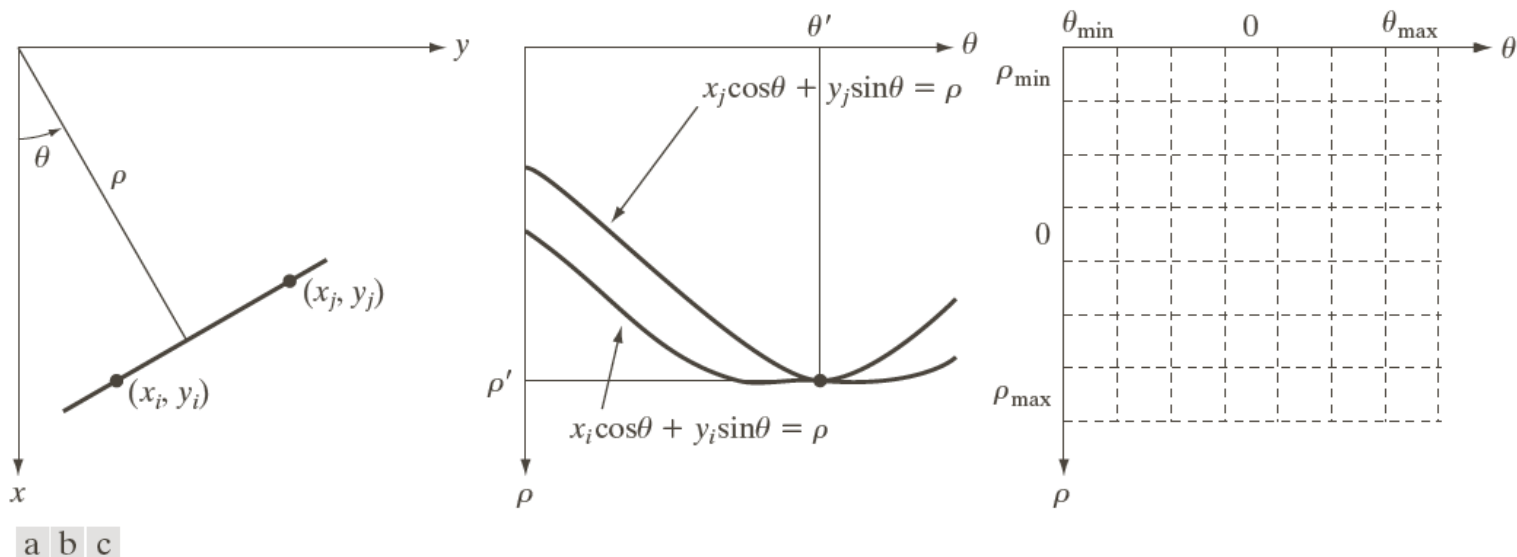
a b

**FIGURE 10.31**  
 (a)  $xy$ -plane.  
 (b) Parameter space.

# 10.2.7 边缘连接和边界检测

## □ 使用霍夫变换的全局处理

- 当直线逼近垂直方向时， $a$ （直线的斜率）会趋于无限大。
- 解决该方法之一，使用一条直线的法线表示： $x\cos\theta + y\sin\theta = \rho$



**FIGURE 10.32** (a)  $(\rho, \theta)$  parameterization of line in the  $xy$ -plane. (b) Sinusoidal curves in the  $\rho\theta$ -plane; the point of intersection  $(\rho', \theta')$  corresponds to the line passing through points  $(x_i, y_i)$  and  $(x_j, y_j)$  in the  $xy$ -plane. (c) Division of the  $\rho\theta$ -plane into accumulator cells.



## 10.2.7 边缘连接和边界检测

### □ 使用霍夫变换的全局处理

- 霍夫变换计算上的魅力在于可将 $\rho\theta$ 参数空间划分为所谓的累加单元，坐标 $(i, j)$ 处的单元具有累加值 $A(i, j)$ ，它对应于与参数空间坐标 $(\rho_i, \theta_i)$ 相关联的正方形。
- 最初，这些单元置为零。然后，对于 $xy$ 平面中的每个非背景点 $(x_k, y_k)$ ，令 $\theta$ 等于 $\theta$ 轴上每个允许的细分值，同时使用方程 $\rho = x_k \cos\theta + y_k \sin\theta$ 解出对于的 $\rho$ 。对得到的 $\rho$ 值进行四舍五入得到沿 $\rho$ 轴最接近的允许单元值。
- 若选择的一个 $\theta_p$ 值得到解 $\rho_q$ ，则令 $A(p, q) = A(p, q) + 1$ 。在这一过程结束后， $A(i, j)$ 中的值 $P$ 将意味着 $xy$ 平面中有 $P$ 个点位于直线 $x \cos\theta_j + y \sin\theta_j = \rho_i$ 上。 $P\theta$ 平面中的细分数量决定了这些点的共线精度。
- 可以证明，刚刚讨论的这种方法的计算次数与 $xy$ 平面中非背景点的数量 $n$ 呈线性关系

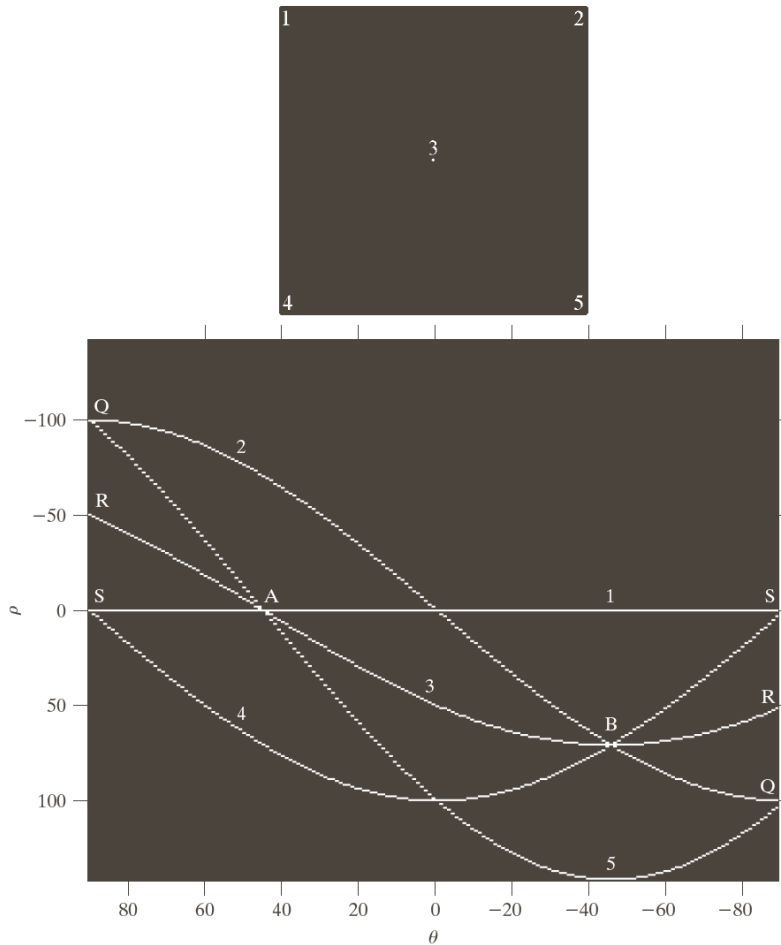
### □ 霍夫变换也适用于形如 $g(v, c) = 0$ 的任何函数，例如圆、椭圆

### □ 现在回到边缘连接问题。基于霍夫变换的一种连接方法如下：

- (1) 使用本节讨论的任何技术得到一幅二值图像
- (2) 指定 $\rho$ - $\theta$ 平面中的细分
- (3) 对像素高度集中的地方检验其累加单元的数量
- (4) 检验选中单元中像素间的关系

# 10.2.7 边缘连接和边界检测

## □ 基本霍夫变换性质的说明



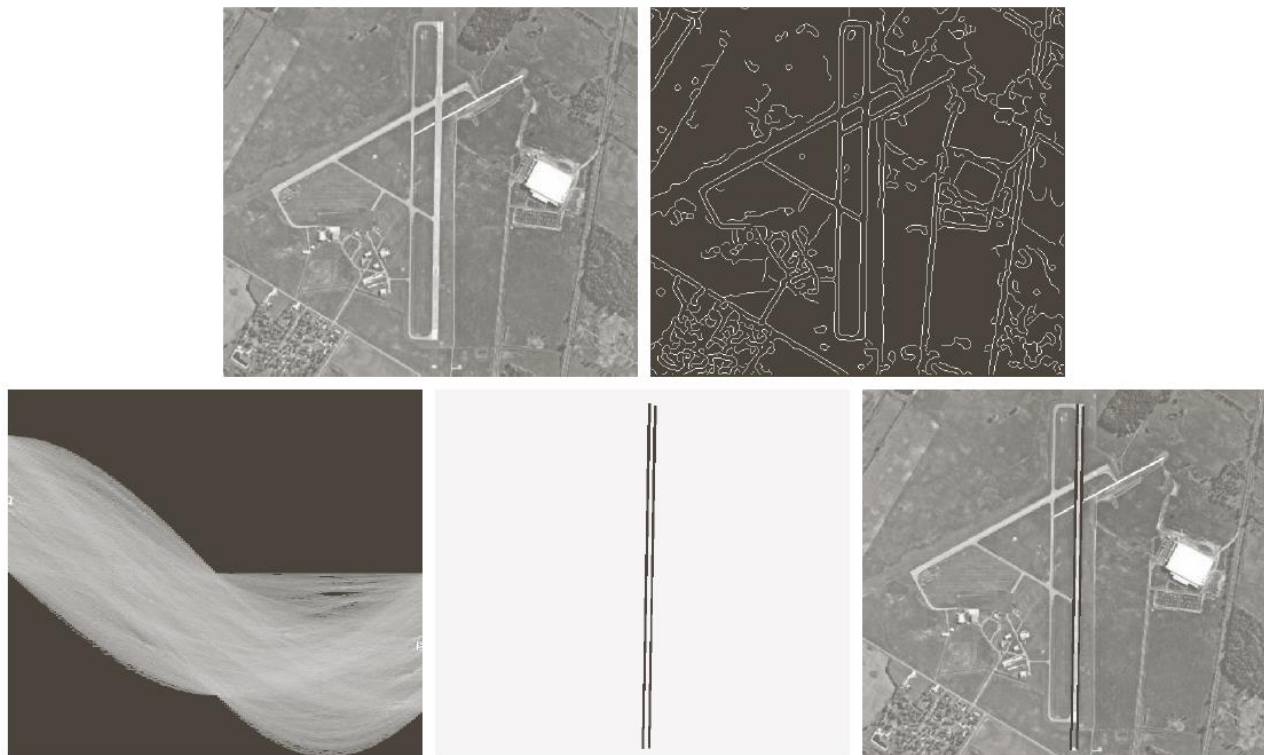
a  
b

**FIGURE 10.33**

(a) Image of size  $101 \times 101$  pixels, containing five points.  
(b) Corresponding parameter space. (The points in (a) were enlarged to make them easier to see.)

# 10.2.7 边缘连接和边界检测

- 使用霍夫变换连接边缘



a b  
c d e

**FIGURE 10.34** (a) A  $502 \times 564$  aerial image of an airport. (b) Edge image obtained using Canny's algorithm. (c) Hough parameter space (the boxes highlight the points associated with long vertical lines). (d) Lines in the image plane corresponding to the points highlighted by the boxes. (e) Lines superimposed on the original image.



# 第10章 图像分割

- 10.1 基础知识
- 10.2 点、线和边缘检测
- 10.3 阈值处理**
- 10.4 基于区域的分割
- 10.5 用形态学分水岭的分割
- 10.6 分割中运动的应用





# 10.3 阈值处理

- 由于阈值处理直观、实现简单且计算速度快，因此图像阈值处理在图像分割应用中处于核心地位。
- 在前面的章节中，我们采用首先寻找边缘线段，然后将这些线段连接为边界的方法来识别区域。本节讨论基于灰度值或灰度值的特性来将图像直接划分为区域的技术
- 小节
  - 10.3.1 基础知识
  - 10.3.2 基本的全局阈值处理
  - 10.3.3 用Otsu方法的最佳全局阈值处理
  - 10.3.4 用图像平滑改善全局阈值处理
  - 10.3.5 利用边缘改进全局阈值处理
  - 10.3.6 多阈值处理
  - 10.3.7 可变阈值处理
  - 10.3.8 多变量阈值处理

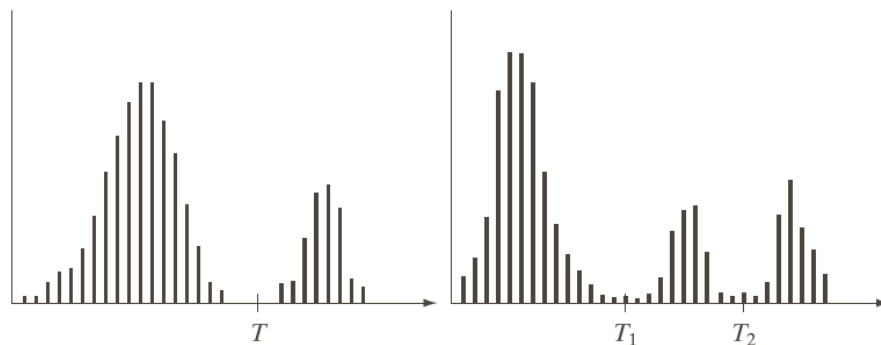
# 10.3.1 基础知识

## □ 灰度阈值处理基础

- 左图中的灰度直方图对应于图像 $f(x, y)$ ，该图像由暗色背景上的较亮物体组成。
- 从背景中提取物体的一种明显办法是选择一个将这些模式分开的阈值 $T$ ；然后， $f(x, y) > T$ 的任何点 $(x, y)$ 称为一个对象点；否则该点称为背景点。
- 换句话说，分割后图像 $g(x, y)$ 由下式给出：

$$g(x, y) = \begin{cases} 1 & \text{如 } f(x, y) > T \\ 0 & \text{如 } f(x, y) \leq T \end{cases}$$

- 当 $T$ 是一个适用于整个图像的常数时，称为全局阈值处理。当 $T$ 在一幅图像上改变时，使用可变阈值处理这一术语



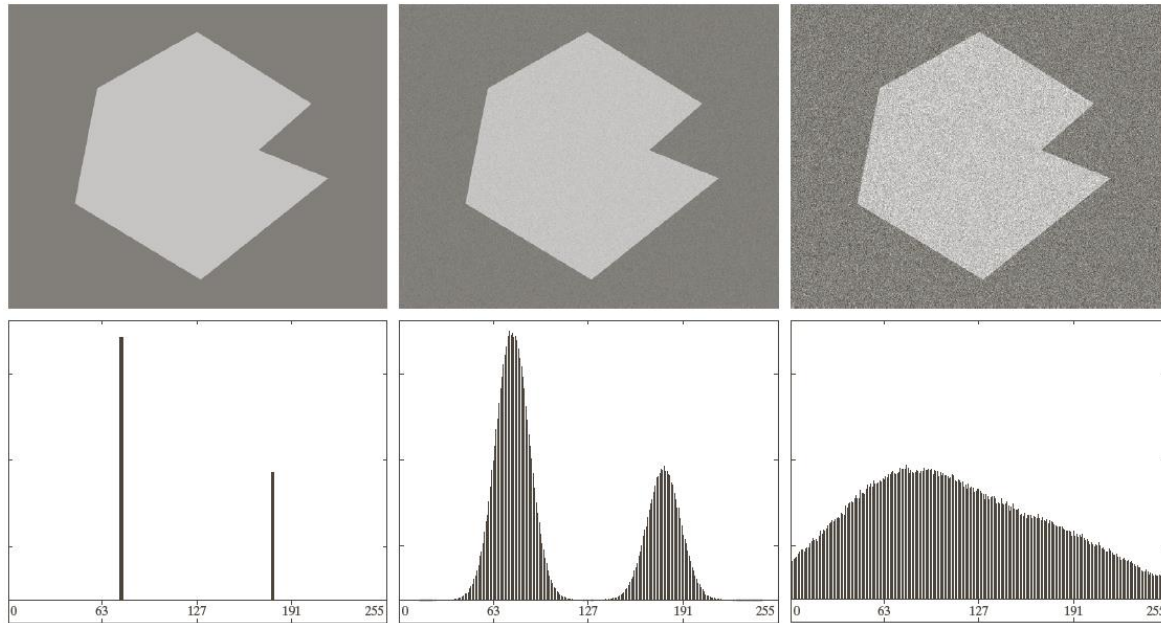
a b

**FIGURE 10.35**  
Intensity histograms that can be partitioned (a) by a single threshold, and (b) by dual thresholds.

- 右图显示了一个更为困难的阈值处理问题，它包含有三个支配模式的直方图。我们可以采用双阈值处理的方法

# 10.3.1 基础知识

## □ 图像阈值处理中噪声的作用



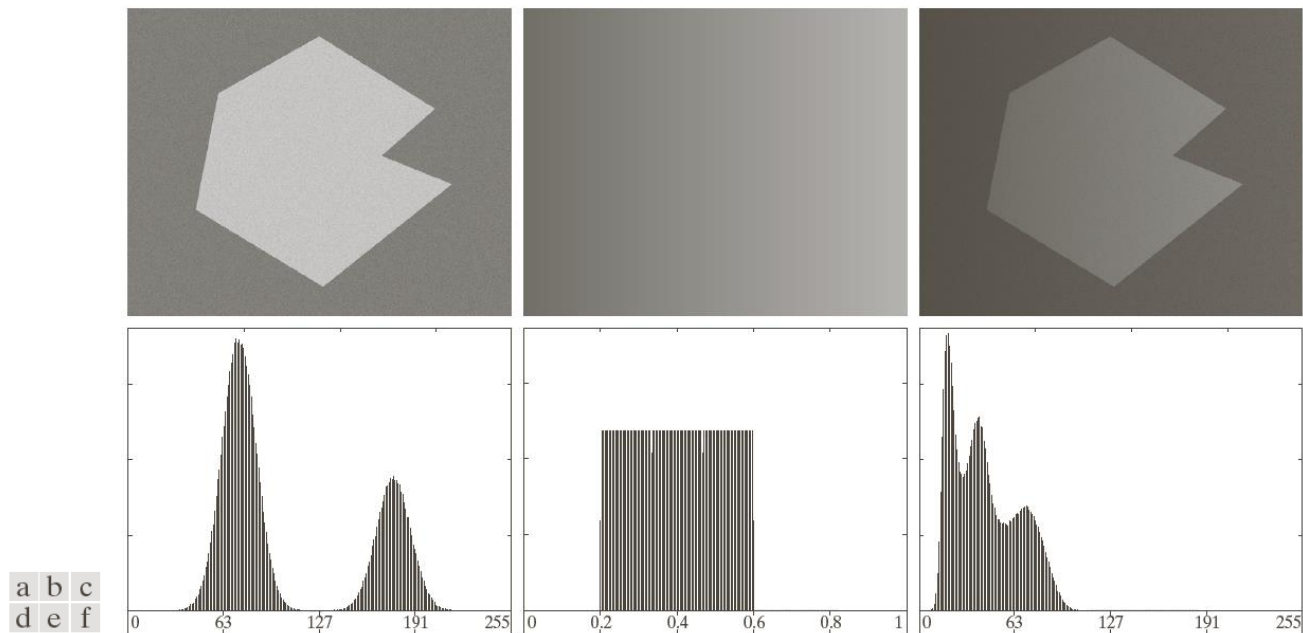
a b c  
d e f

**FIGURE 10.36** (a) Noiseless 8-bit image. (b) Image with additive Gaussian noise of mean 0 and standard deviation of 10 intensity levels. (c) Image with additive Gaussian noise of mean 0 and standard deviation of 50 intensity levels. (d)–(f) Corresponding histograms.

# 10.3.1 基础知识

## □ 光照和反射的作用

- 可变的灰度函数表示非均匀光照



**FIGURE 10.37** (a) Noisy image. (b) Intensity ramp in the range  $[0.2, 0.6]$ . (c) Product of (a) and (b). (d)–(f) Corresponding histograms.

## ■ 解决办法

- ✓ 直接校正这种阴影模式
- ✓ 通过顶帽变换处理来尝试校正全局阴影模式
- ✓ 使用可变阈值近似处理非均匀性

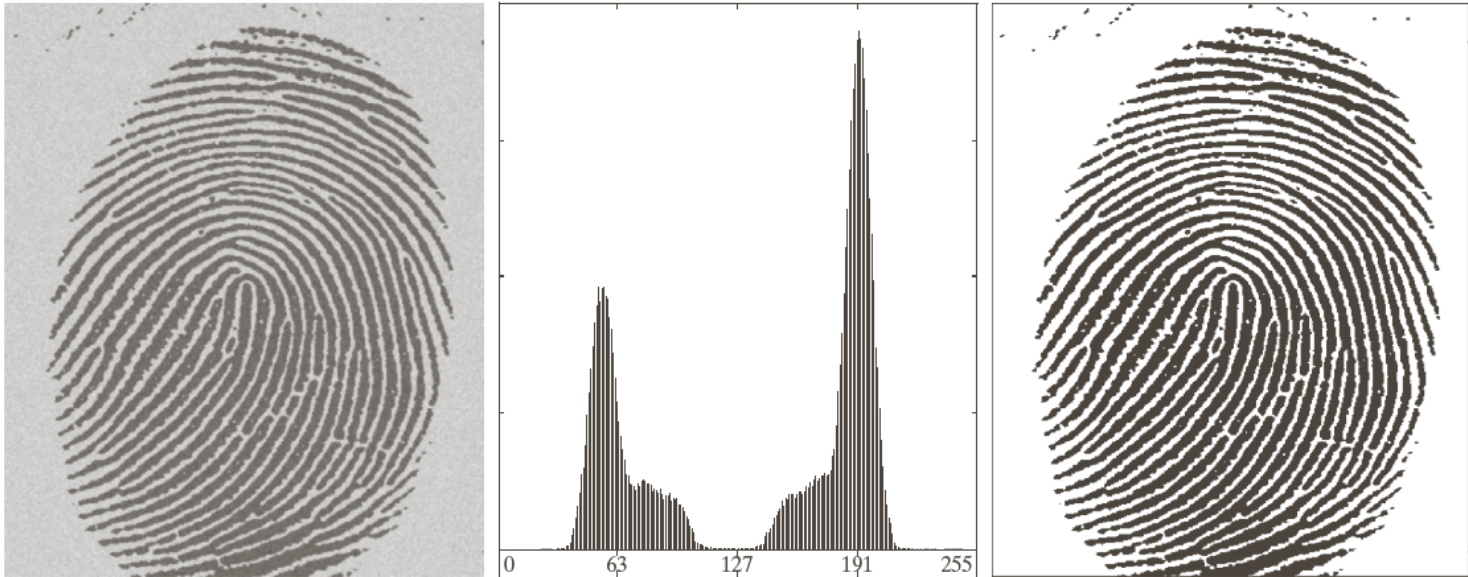


## 10.3.2 基本的全局阈值处理

- 当物体和背景像素的灰度分布十分明显时，可以用适用于整个图像的单个（全局）阈值。
- 在大多数应用中，通常图像之间有较大变化，即使全局阈值是一种合适的方法，也需要有能对每幅图像**自动估计阈值**的算法。
- 下面的迭代算法可用于这一目的（本质是k-means聚类，k=2）
  - （1）为全局阈值T选择一个初始估计值
  - （2）用单个阈值T分割该图像。这将产生两组像素： $G_1$ 由灰度值大于T的所有像素组成， $G_2$ 由所有小于等于T的像素组成
  - （3）对 $G_1$ 和 $G_2$ 的像素分别计算平均灰度值（均值） $m_1$ 和 $m_2$
  - （4）计算一个新的阈值：
$$T = \frac{1}{2}(m_1 + m_2)$$
  - （5）重复步骤（2）到步骤（4），直到连续迭代中的T值间的差小于一个预定义的参数 $\Delta T$ 为止
- 参数 $\Delta T$ 用于控制迭代的次数。通常， $\Delta T$ 越大，则算法执行的迭代次数少。图像的平均灰度对于T来说是较好的初始选择

# 10.3.2 基本的全局阈值处理

## □ 全局阈值处理



a b c

**FIGURE 10.38** (a) Noisy fingerprint. (b) Histogram. (c) Segmented result using a global threshold (the border was added for clarity). (Original courtesy of the National Institute of Standards and Technology.)



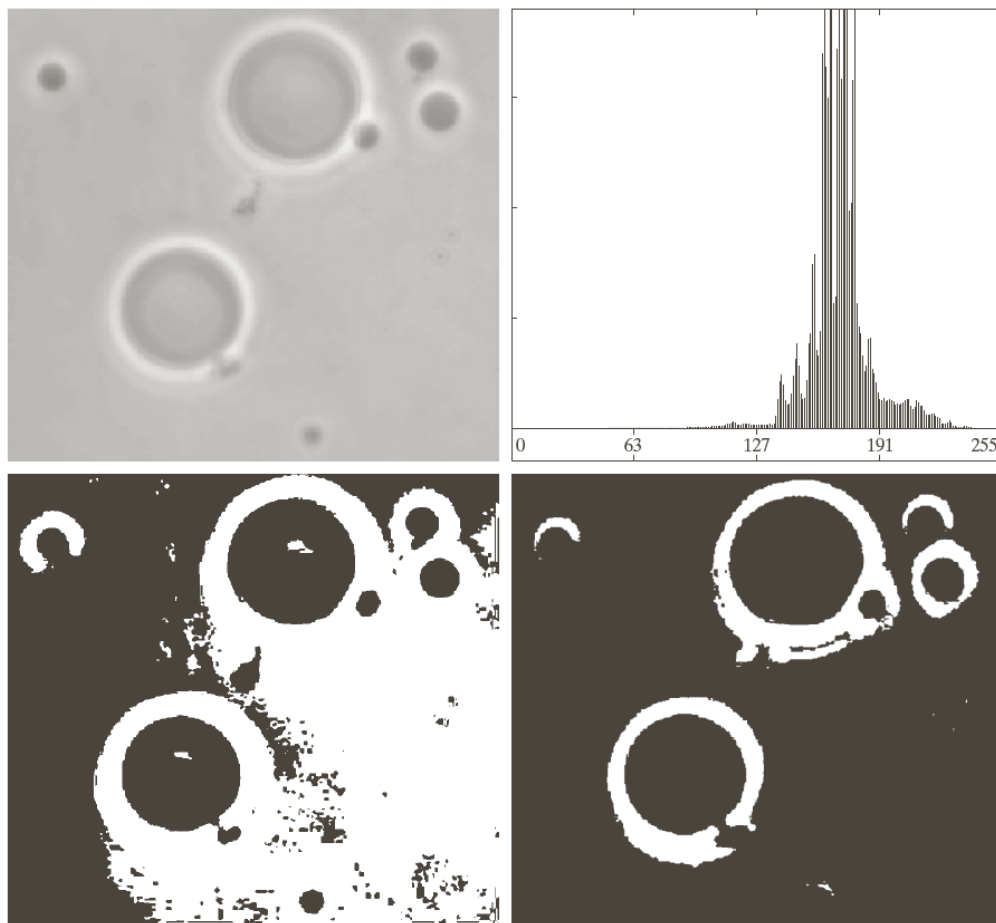
# 10.3.3 Otsu方法：最佳全局阈值处理

- 该方法在**类间方差最大**的情况下是最佳的；完全以在一幅图像的直方图上执行计算为基础。
- Otsu算法小节如下：
  - 计算输入图像的归一化直方图。使用 $p_i, i = 0, 1, 2, \dots, L - 1$ 表示该直方图的各个分量
  - 计算累积和： $P_1(k) = \sum_{i=0}^k p_i, (k = 0, 1, 2, \dots, L - 1), P_2(k) = 1 - P_1(k)$
  - 计算累积均值： $m(k) = \sum_{i=0}^k ip_i, 对于k = 0, 1, 2, \dots, L - 1,$
  - 计算全局灰度均值： $m_G = \sum_{i=0}^{L-1} ip_i$
  - 用 $\sigma_B^2(k) = \frac{[m_G P_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]}$ ，对于 $k = 0, 1, 2, \dots, L - 1$ ，计算类间方差 $\sigma_B^2(k)$
  - 得到Otsu阈值 $k^*$ ，即使得 $\sigma_B^2(k)$ 最大的k值。如果最大值不唯一，用相应检测到的各个最大值的k的平均得到 $k^*$
  - 在 $k = k^*$ 处计算 $\eta(k) = \frac{\sigma_B^2(k)}{\sigma_G^2}$ ，得到可分性测度 $\eta^*$



# 10.3.3 用Otsu方法的最佳全局阈值处理

## □ 使用Otsu方法的最佳全局阈值处理



a b  
c d

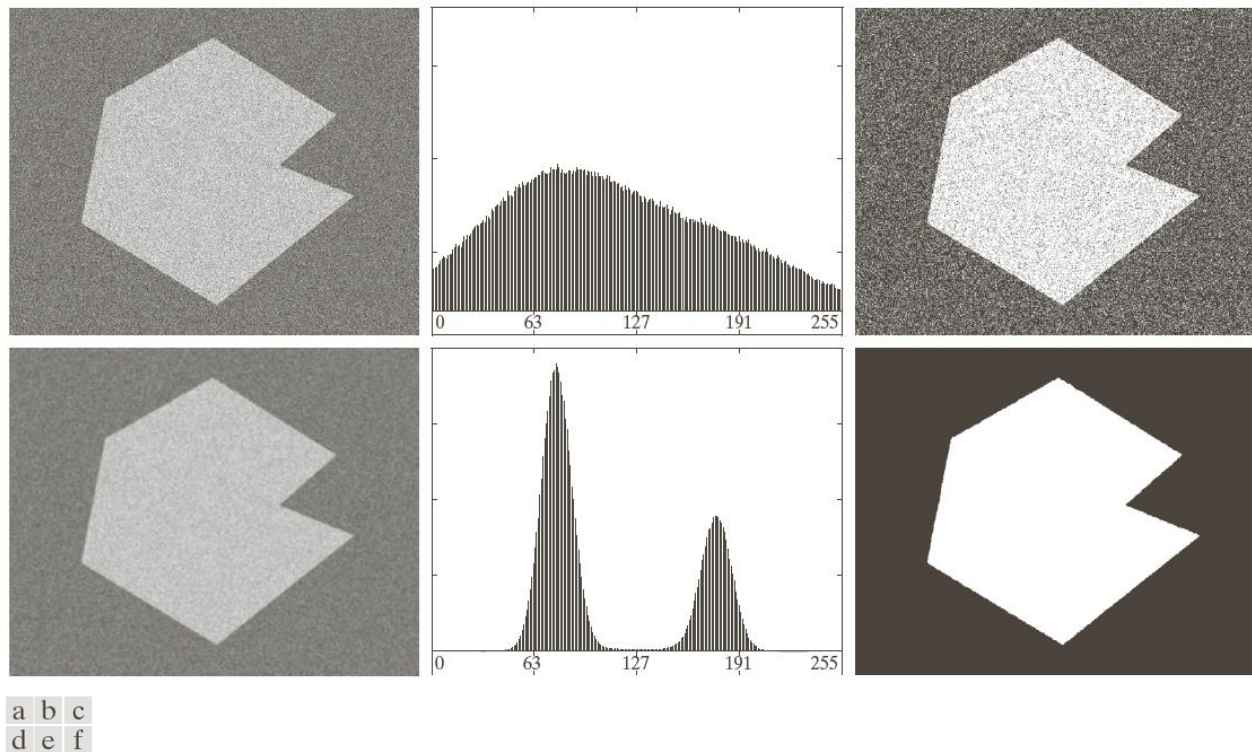
**FIGURE 10.39**

(a) Original image.  
(b) Histogram (high peaks were clipped to highlight details in the lower values).  
(c) Segmentation result using the basic global algorithm from Section 10.3.2.  
(d) Result obtained using Otsu's method. (Original image courtesy of Professor Daniel A. Hammer, the University of Pennsylvania.)

# 10.3.4 用图像平滑改善全局阈值处理

- 噪声会将简单的阈值处理问题变为不可解决的问题。当噪声不能在源头减少，并且阈值处理又是所选择的分割方法时，那么通常能增强性能的一种技术是，在阈值处理之前平滑图像

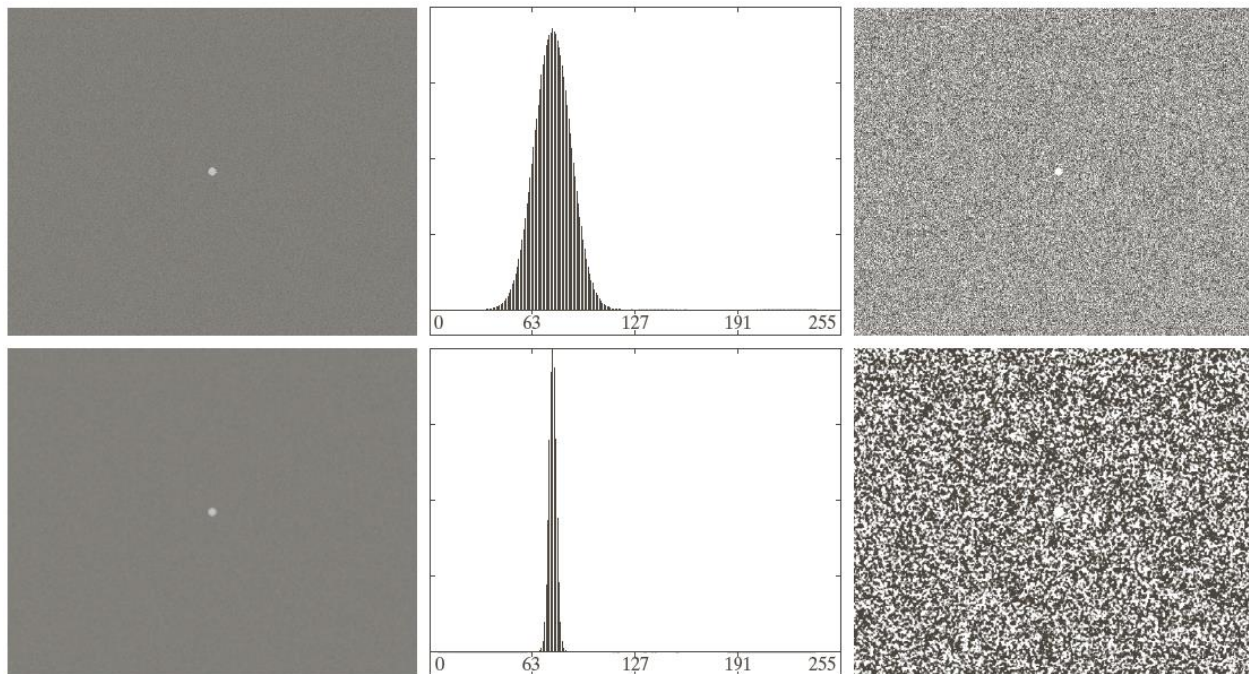
□ 例：



**FIGURE 10.40** (a) Noisy image from Fig. 10.36 and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a  $5 \times 5$  averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method.

# 10.3.4 用图像平滑改善全局阈值处理

- 考虑降低图中相对于背景的区域大小所带来的影响
  - 失败的原因可归于这样一个事实，即区域太小，以至于该区域对直方图的贡献与由噪声引起的灰度扩散相比无足轻重。



a b c  
d e f

**FIGURE 10.41** (a) Noisy image and (b) its histogram. (c) Result obtained using Otsu's method. (d) Noisy image smoothed using a  $5 \times 5$  averaging mask and (e) its histogram. (f) Result of thresholding using Otsu's method. Thresholding failed in both cases.

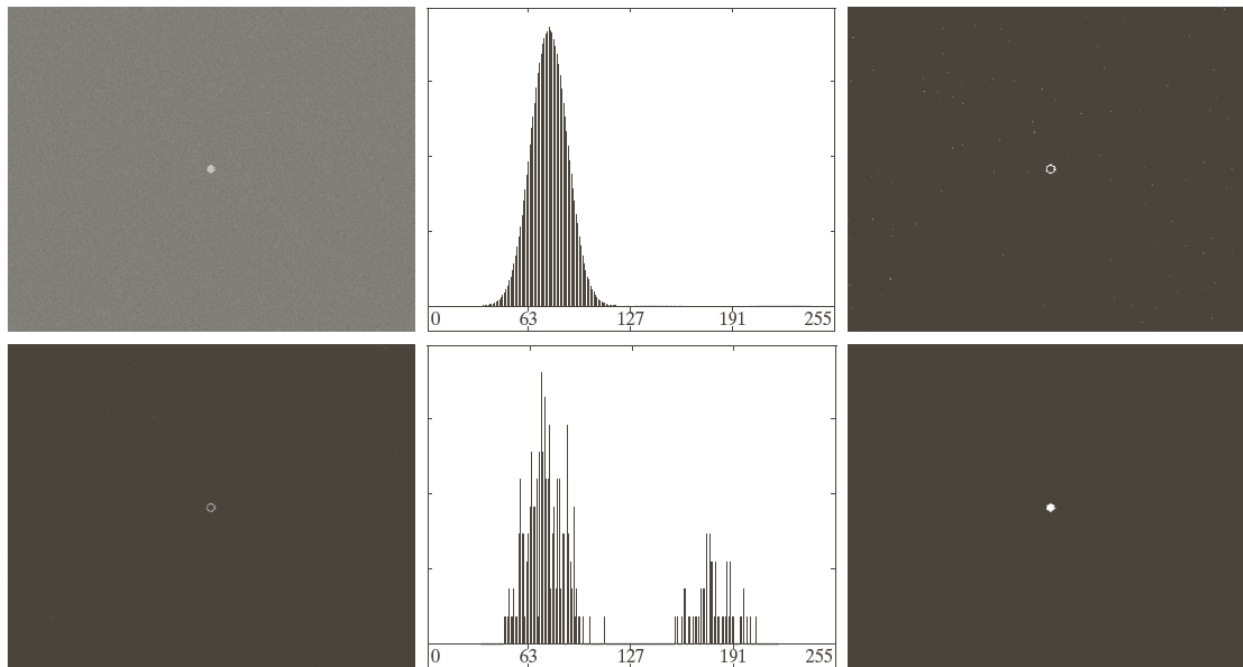


# 10.3.5 利用边缘改进全局阈值处理

- 如果直方图的波峰是高、窄、对称的，且被深波谷分开，则选取一个“较好”阈值的机会相当大。**改进直方图形状的一种方法是，仅考虑那些位于或靠近物体和背景间的边缘的像素。**该算法可小结如下，其中 $f(x, y)$ 是输入图像：
  - 1、采用10.2节讨论的任何一种方法来计算一幅**边缘图像**，无论是 $f(x, y)$ 梯度的幅度还是拉普拉斯的绝对值均可
  - 2、指定一个阈值 $T$
  - 3、用步骤2中的阈值对步骤1中的图像进行阈值处理，产生一幅二值图像  $g_T(x, y)$ 。在从 $f(x, y)$ 中选取对应于“强”边缘像素的下一步中，该图像用做一幅模板图像
  - 4、仅用 $f(x, y)$ 中对应于 $g_T(x, y)$ 中像素值为1的位置的像素计算直方图
  - 5、用步骤4中直方图全局地分割 $f(x, y)$ ，例如使用Otsu方法
  - 若将 $T$ 设为小于边缘图像的最小值的任何值， $g_T(x, y)$ 将由全1组成，这意味着 $f(x, y)$ 的所有像素将用于计算图像的直方图。习惯以百分比的形式来指定 $T$ 值，通常较高（90%），以便在计算中用梯度/拉普拉斯图像中的较少像素

# 10.3.5 利用边缘改进全局阈值处理

- 用以梯度为基础的边缘信息改进全局阈值处理



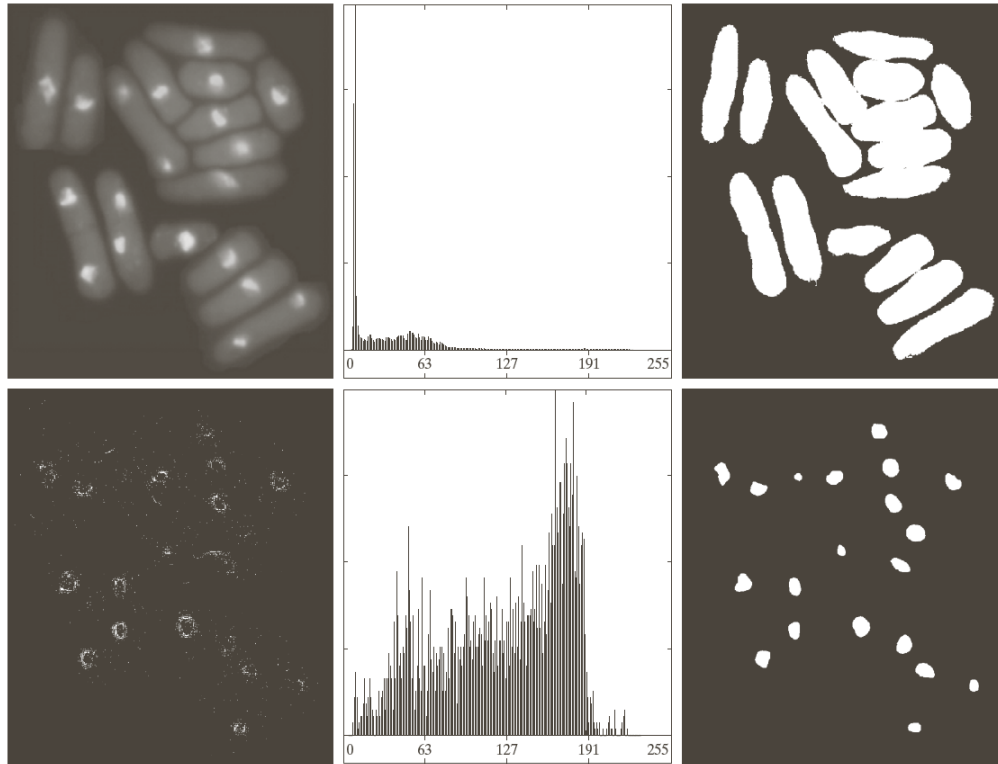
a b c  
d e f

**FIGURE 10.42** (a) Noisy image from Fig. 10.41(a) and (b) its histogram. (c) Gradient magnitude image thresholded at the 99.7 percentile. (d) Image formed as the product of (a) and (c). (e) Histogram of the nonzero pixels in the image in (d). (f) Result of segmenting image (a) with the Otsu threshold based on the histogram in (e). The threshold was 134, which is approximately midway between the peaks in this histogram.



# 10.3.5 利用边缘改进全局阈值处理

- 用拉普拉斯为基础的边缘信息改进全局阈值处理



a b c  
d e f

**FIGURE 10.43** (a) Image of yeast cells. (b) Histogram of (a). (c) Segmentation of (a) with Otsu's method using the histogram in (b). (d) Thresholded absolute Laplacian. (e) Histogram of the nonzero pixels in the product of (a) and (d). (f) Original image thresholded using Otsu's method based on the histogram in (e). (Original image courtesy of Professor Susan L. Forsburg, University of Southern California.)



# 10.3.6 多阈值处理

□ 迄今为止，我们关注的是用单个全局阈值对图像进行分割。我们可将10.3.3节介绍的阈值处理方法扩展到任意数量的阈值，因此以这种方法为基础的可分性测度也可以扩展到任意数量的分类

■ 在K个类 $C_1, C_2, \dots, C_K$ 的情况下，类间方差可归纳为下面的表达式：

$$\sigma_B^2 = \sum_{k=1}^K P_k (m_k - m_G)^2$$

式中， $P_k = \sum_{i \in C_k} P_i$ ， $m_k = \frac{1}{P_k} \sum_{i \in C_k} i P_i$ ， $m_G = \sum_{i=0}^{L-1} i p_i$

K类有K-1个阈值来分离，这些值 $k_1^*, k_2^*, \dots, k_{K-1}^*$ 是上式的最大值：

$$\sigma_B^2(k_1^*, k_2^*, \dots, k_{K-1}^*) = \max_{0 < k_1 < k_2 < \dots < k_{K-1} < L-1} \sigma_B^2(k_1, k_2, \dots, k_{K-1})$$

■ 对于由三个灰度间隔组成的三个类（两个阈值），最佳阈值是：

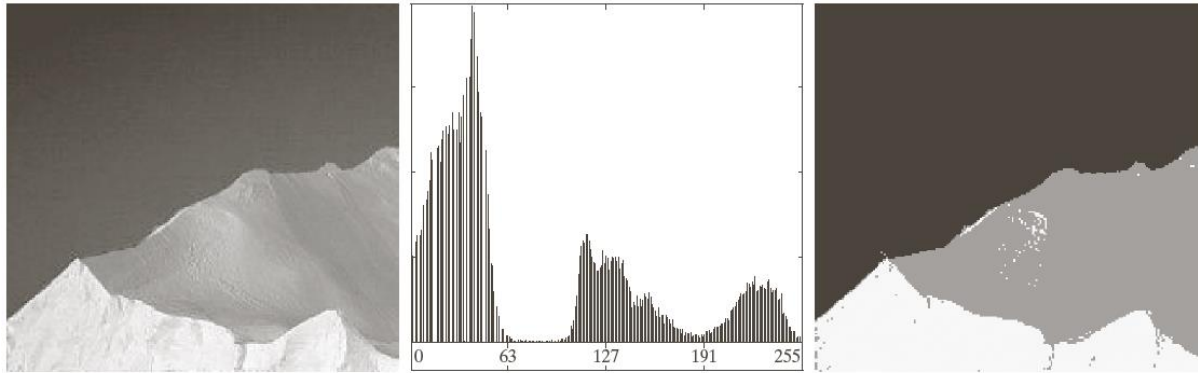
$$\sigma_B^2(k_1^*, k_2^*) = \max_{0 < k_1 < k_2 < L-1} \sigma_B^2(k_1, k_2)$$

该过程由选择第一个 $k_1$ 值开始（为1）。接着， $k_2$ 的所有值在大于 $k_1$ 和小于L-1的范围内增加。然后，将 $k_1$ 增大到其下一个值， $k_2$ 的所有值再次在大于 $k_1$ 的所有值范围内增加。重复该过程，直到 $k_1 = L - 3$ 为止。该结果是一个二维阵列 $\sigma_B^2(k_1, k_2)$ ，最后一步是在阵列中寻找最大值。对应于最大值的 $k_1$ 值和 $k_2$ 值就是最佳阈值 $k_1^*$ 和 $k_2^*$ 。



# 10.3.6 多阈值处理

## □ 多个全局阈值处理



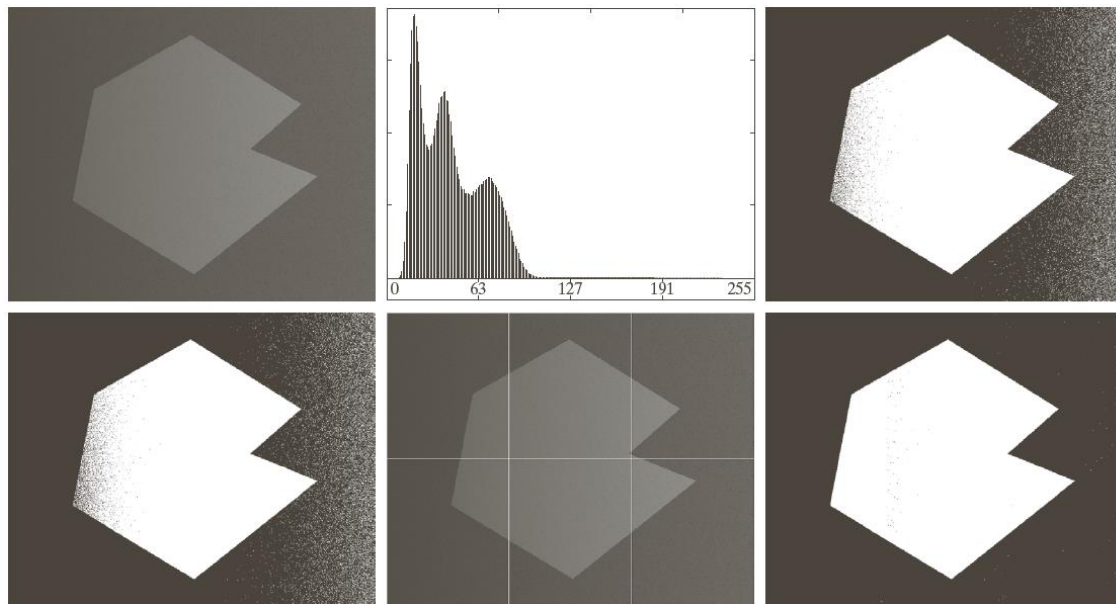
a b c

**FIGURE 10.45** (a) Image of iceberg. (b) Histogram. (c) Image segmented into three regions using dual Otsu thresholds. (Original image courtesy of NOAA.)

# 10.3.7 可变阈值处理

## □ 图像分块

- 可变阈值处理最简单的方法之一是，把一幅图像分成不重叠的矩形。
- 这种方法用于补偿光照和/或反射的不均匀性。选择的矩形要足够小，以便每个矩形的光照都近似是均匀的
- 例：



a b c  
d e f

**FIGURE 10.46** (a) Noisy, shaded image and (b) its histogram. (c) Segmentation of (a) using the iterative global algorithm from Section 10.3.2. (d) Result obtained using Otsu's method. (e) Image subdivided into six subimages. (f) Result of applying Otsu's method to each subimage individually.



# 10.3.7 可变阈值处理

## □ 基于局部图像特性的可变阈值处理

- 更为一般的方法是在一幅图像中的每点 $(x, y)$ 计算阈值，该阈值以一个或多个在 $(x, y)$ 邻域计算的特性为基础
- 令 $\sigma_{xy}$ 和 $m_{xy}$ 表示一幅图像中以坐标 $(x, y)$ 为中心的邻域 $S_{xy}$ 所包含的像素集合的标准差和均值。下面是可变局部阈值的通用形式：

$$T_{xy} = a\sigma_{xy} + bm_{xy}$$

式中， $a$ 和 $b$ 是非负常数，且 $T_{xy} = a\sigma_{xy} + bm_G$ 。分割后的图像计算如下：

$$g(x, y) = \begin{cases} 1, & f(x, y) > T_{xy} \\ 0, & f(x, y) \leq T_{xy} \end{cases}$$

该式对图像中的所有像素位置进行求值，并在每个点 $(x, y)$ 处使用邻域 $S_{xy}$ 中的像素计算不同的阈值

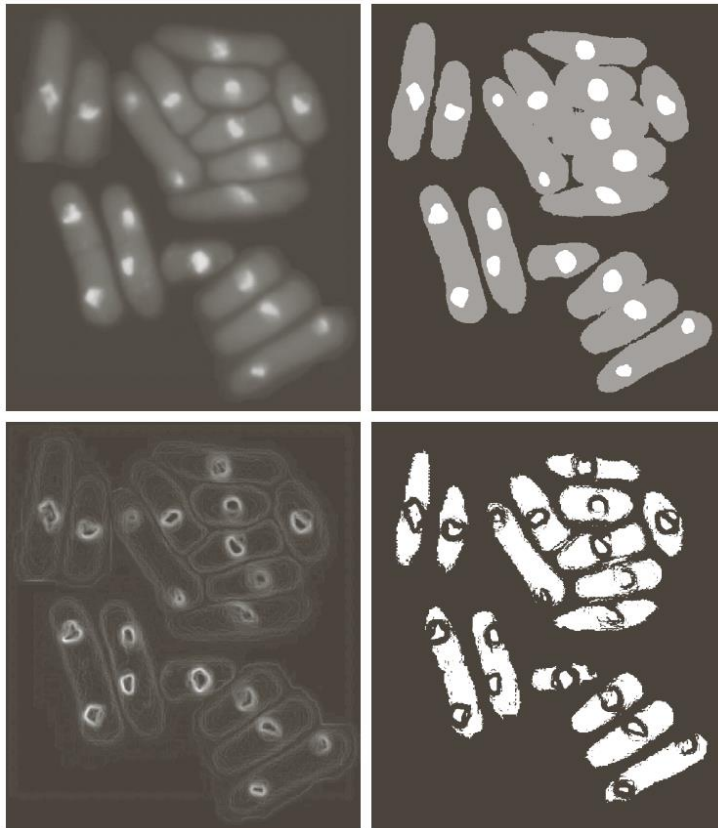
- 使用以 $(x, y)$ 处的邻域计算出的参数为基础的属性，有效的权值可显著增强局部阈值处理：

$$g(x, y) = \begin{cases} 1, & Q(\text{局部参数}) \text{为真} \\ 0, & Q(\text{局部参数}) \text{为假} \end{cases}$$

式中， $Q$ 是以邻域 $S_{xy}$ 中像素计算的参数为基础的一个属性。

# 10.3.7 可变阈值处理

## □ 基于局部图像特性的可变阈值处理



a	b
c	d

**FIGURE 10.48**

(a) Image from Fig. 10.43.

(b) Image segmented using the dual thresholding approach discussed in Section 10.3.6.

(c) Image of local standard deviations.

(d) Result obtained using local thresholding.



# 10.3.7 可变阈值处理

## □ 使用移动平均

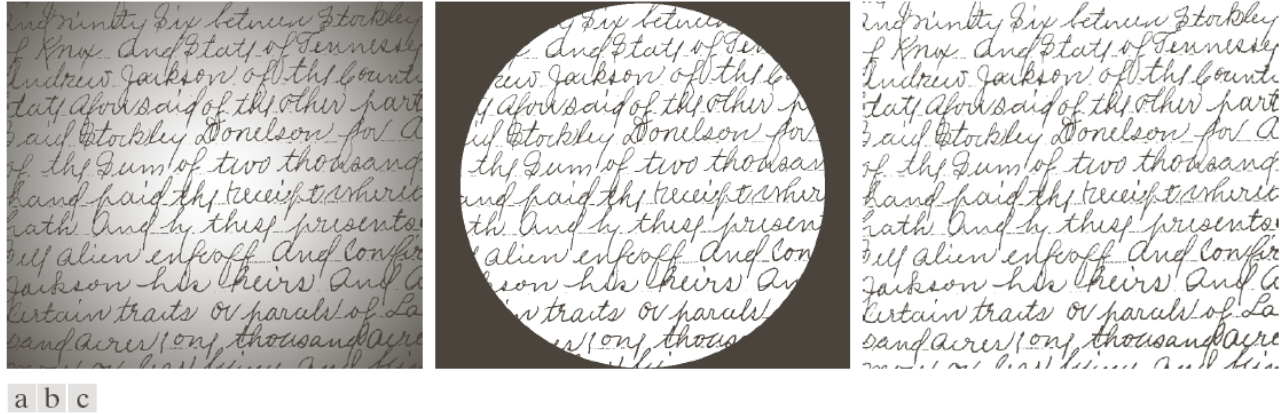
- 刚才讨论的局部阈值处理方法的一种特殊情形，是以一幅图像的**扫描行**计算**移动平均**为基础的。
- 通常，为减少光照偏差，扫描是以Z字形模式逐行执行的。令 $z_{k+1}$ 表示步骤 $k + 1$ 中扫描序列遇到的点的灰度。这个新点的移动平均由下式给出：

$$m(k + 1) = \frac{1}{n} \sum_{i=k+2-n}^{k+1} z_i = m(k) + \frac{1}{n} (z_{k+1} - z_{k+1-n})$$

- 式中， $n$ 是用于计算平均的点数， $m(1) = z_1/n$ 。这个初始值严格来说并不正确，因为单点的平均是该点自身的值。然而，第一次计算时，并不要求特殊的计算。从另一个角度看，如果图像的边界填充了 $n-1$ 个0，它就是我们将得到的那个值。
- 该算法仅初始化一次，而不在每行都初始化。因为对图像中的每个点都计算移动平均，因此用 $g(x,y) = \begin{cases} 1, & f(x,y) > T_{xy} \\ 0, & f(x,y) \leq T_{xy} \end{cases}$ 实现分割，其中 $T_{xy} = bm_{xy}$ ， $b$ 是常数， $m_{xy}$ 是在输入图像中的点 $(x,y)$ 处得到的移动平均

# 10.3.7 可变阈值处理

## □ 用移动平均的文档阈值处理



**FIGURE 10.49** (a) Text image corrupted by spot shading. (b) Result of global thresholding using Otsu's method. (c) Result of local thresholding using moving averages.



**FIGURE 10.50** (a) Text image corrupted by sinusoidal shading. (b) Result of global thresholding using Otsu's method. (c) Result of local thresholding using moving averages.





## 10.3.8 多变量阈值处理

- 迄今为止是基于单个变量（灰度）的阈值处理。某些情况下，传感器可产生多个可利用的变量来表征图像中的每个像素，这样就允许进行多变量阈值处理。例如，对彩色图像的阈值处理
- 多变量阈值处理可视为一种距离计算。
  - 假设我们想要从一幅彩色图像中提取具有指定颜色的所有区域。令 $a$ 表示，计算任意彩色点 $z$ 和平均彩色 $a$ 间的距离测度 $D(z, a)$ 。然后，按下式分割图像：

$$g = \begin{cases} 1, & D(z, a) < T \\ 0, & \text{其他} \end{cases}$$

- 其中， $T$ 是一个阈值。通常，如果 $z$ 是一个 $n$ 维向量，则距离可用欧式距离或马氏距离度量





# 第10章 图像分割

- 10.1 基础知识
- 10.2 点、线和边缘检测
- 10.3 阈值处理
- 10.4 基于区域的分割**
- 10.5 用形态学分水岭的分割
- 10.6 分割中运动的应用



# 10.4 基于区域的分割

- 如10.1节讨论那样，分割的目的是将一幅图像划分为多个区域。在10.2节中，我们基于灰度级的不连续性尝试寻找区域间的边界来解决这一问题；而在10.3节中，分割是通过以像素特性分布为基础
- 10.4.1 区域生长
- 10.4.2 区域分裂与聚合

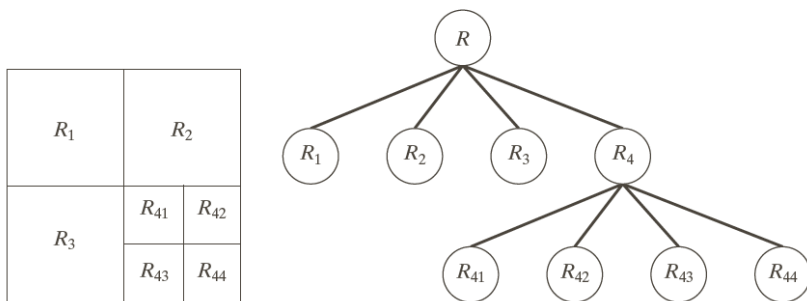


# 10.4.1 区域生长

- 区域生长是根据预先定义的生长准则，将像素或子区域组合为更大区域的过程。
- 基本方法是**从一组“种子”点开始**，将与种子预先定义的**性质相似**的那些领域像素添加到每个种子上，来形成这些生长区域。
  - **连通属性，终止规则**
- 令 $f(x, y)$ 表示一个输入图像； $S(x, y)$ 表示一个种子阵列，阵列中种子点位置处为1，其他为0； $Q$ 表示在每个位置 $(x, y)$ 处的属性。
- 假设阵列 $f$ 和 $S$ 的尺寸相同。基于8连接的一个基本区域生长算法：
  - 1、在 $S(x, y)$ 中寻找所有连通分量，并把每个连通分量腐蚀为一个像素；把找到的所有这种像素标记为1，把 $S$ 中的所有其他像素标记为0
  - 2、在坐标对 $(x, y)$ 处形成图像 $f_Q$ ：若输入图像在该坐标处满足给定的属性 $Q$ ，则令 $f_Q(x, y) = 1$ ，否则令 $f_Q(x, y) = 0$
  - 3、令 $g$ 是这样形成的图像：即把 $f_Q$ 中为8连通种子点的所有1值点，添加到 $S$ 中的每个种子点
  - 4、用不同的区域标记（如1,2,3,...）标记 $g$ 中的每个连通分量。这就是由区域生长得到的分割图像

# 10.4.2 区域分裂与聚合

- 上一节是从一组种子点来生长区域。另一种方法是首先将一幅图像细分为一组任意的不相交的区域，然后聚合和/或分裂这些区域
- 令 $R$ 表示整幅图像区域，并选择一个属性 $Q$ 。则区域分裂与聚合的算法过程如下：
  - 把满足 $Q(R_i) = FALSE$ 的任何区域 $R_i$ 分裂为4个不相交的象限区域
  - 不可能进一步分裂时，对满足条件 $Q(R_j \cup R_k) = TRUE$ 的任意两个邻接区域 $R_j$ 和 $R_k$ 进行聚合
  - 无法进一步聚合时，停止操作
- 习惯上要规定一个不能再进一步执行分裂的最小四象限区的尺寸

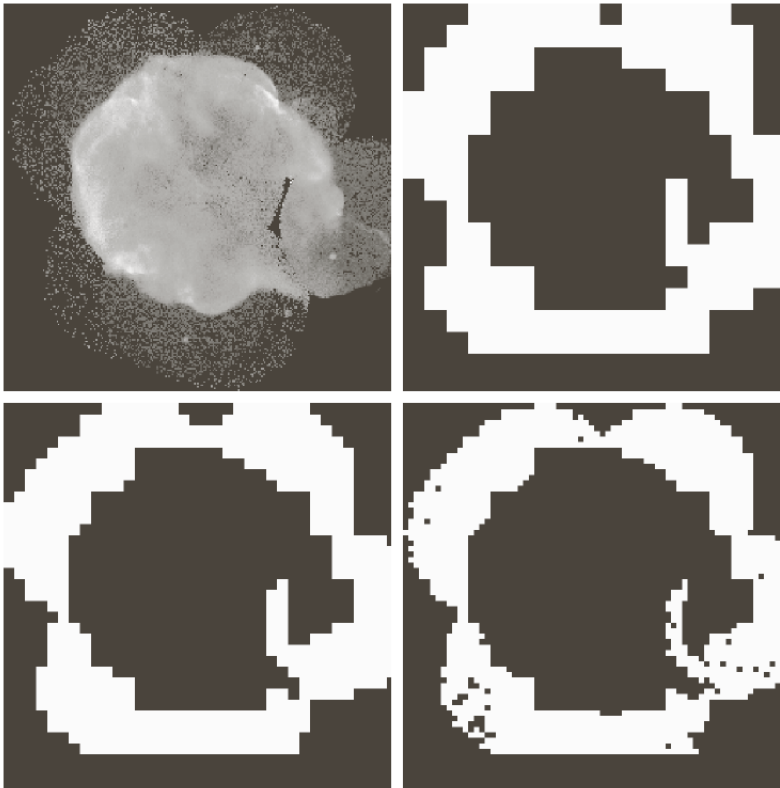


a b

**FIGURE 10.52**  
 (a) Partitioned image.  
 (b) Corresponding quadtree.  $R$  represents the entire image region.

# 10.4.2 区域分裂与聚合

## □ 用区域分裂和聚合进行分割



a b  
c d

**FIGURE 10.53**  
(a) Image of the Cygnus Loop supernova, taken in the X-ray band by NASA's Hubble Telescope. (b)–(d) Results of limiting the smallest allowed quadregion to sizes of  $32 \times 32$ ,  $16 \times 16$ , and  $8 \times 8$  pixels, respectively. (Original image courtesy of NASA.)

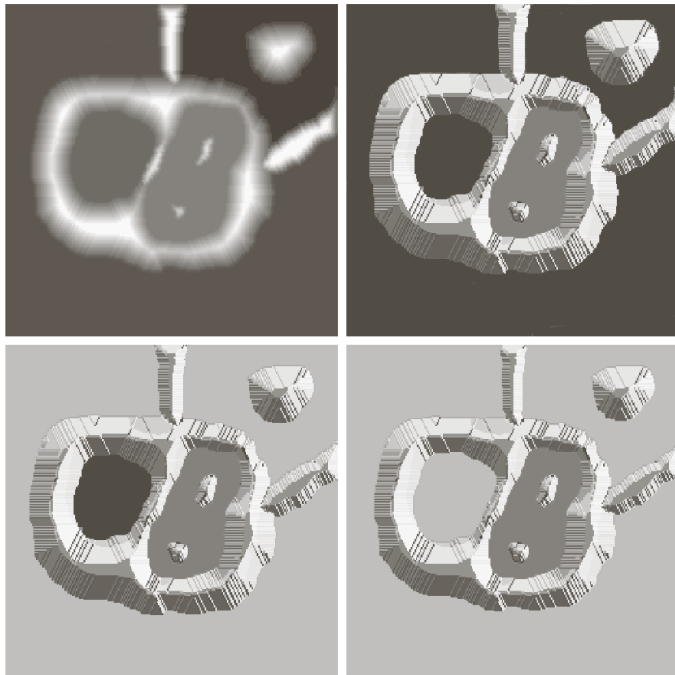


# 第10章 图像分割

- 10.1 基础知识
- 10.2 点、线和边缘检测
- 10.3 阈值处理
- 10.4 基于区域的分割
- 10.5 用形态学分水岭的分割**
- 10.6 分割中运动的应用

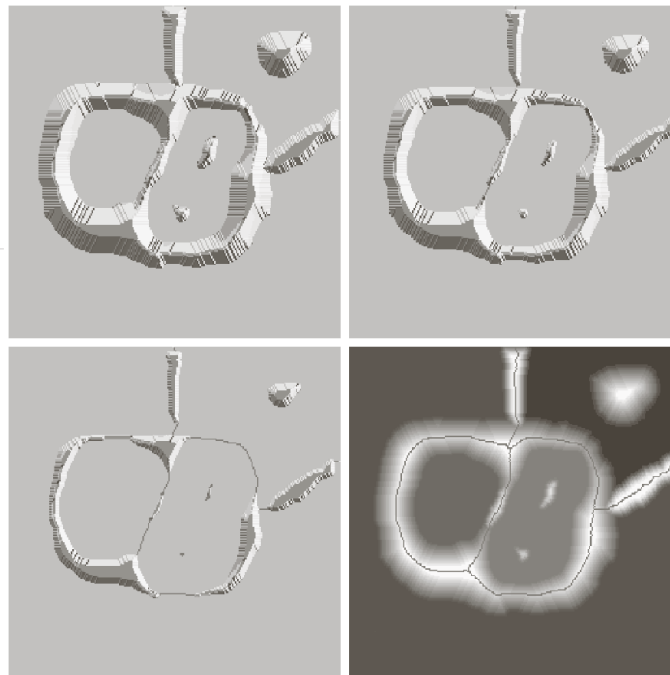
# 10.5.1 背景知识

- 分水岭也称分水线/水线，将图像看成三维地形表示
- 基本思想：
  - 假设在每个区域的最小值上打一个洞，并且让水通过洞以均匀的速率上升，从低到高淹没整个地形。当不同汇水盆地中上升的水聚集时，修建一个水坝来阻止这种聚合。水将达到在水线上只能见到各个水坝的顶部的程度。这些大坝的边界对应于分水岭的分割线



a b  
c d

**FIGURE 10.54**  
(a) Original image.  
(b) Topographic view. (c)–(d) Two stages of flooding.



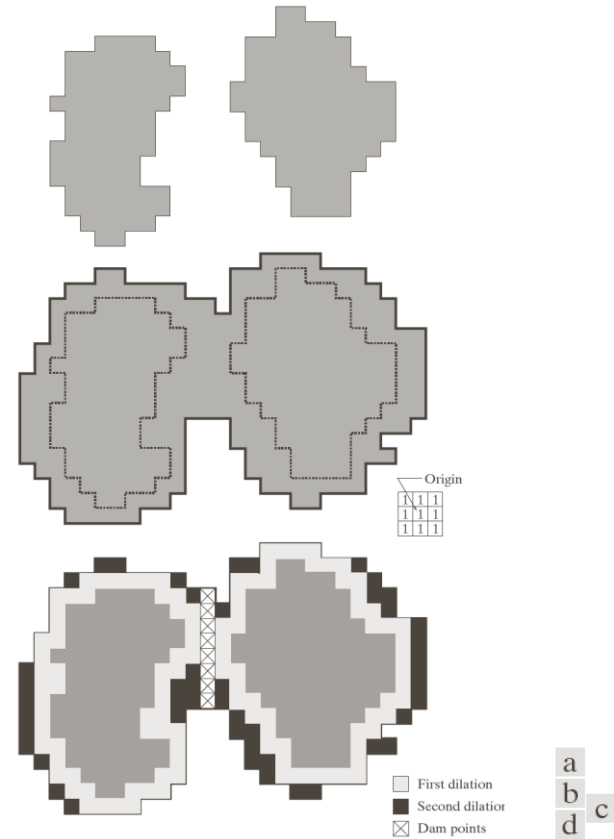
e f  
g h

**FIGURE 10.54**  
(Continued)  
(e) Result of further flooding.  
(f) Beginning of merging of water from two catchment basins (a short dam was built between them). (g) Longer dams. (h) Final watershed (segmentation) lines.  
(Courtesy of Dr. S. Beucher, CMM/École des Mines de Paris.)



# 10.5.2 水坝构建

- 使用形态学膨胀来构建水坝：
  - 令 $C[n - 1]$ 表示a图两个集合的并集
  - 令 $q$ 表示b图中这个连通分量
- 假设a图中的每个连通分量被c图所示结构元膨胀满足两个条件：
  - 膨胀必须受到 $q$ 约束（膨胀过程中，结构元的中心只能位于 $q$ 中的点处）
  - 不能对会引起正被膨胀集合聚合（变成单个连通分量）的那些点执行膨胀
- 很明显， $q$ 中只有满足上述两个条件的点，才能描绘出d图中由叉线所示的1像素宽的连通路徑



**FIGURE 10.55** (a) Two partially flooded catchment basins at stage  $n - 1$  of flooding. (b) Flooding at stage  $n$ , showing that water has spilled between basins. (c) Structuring element used for dilation. (d) Result of dilation and dam construction. 72

# 10.5.3 分水岭分割算法

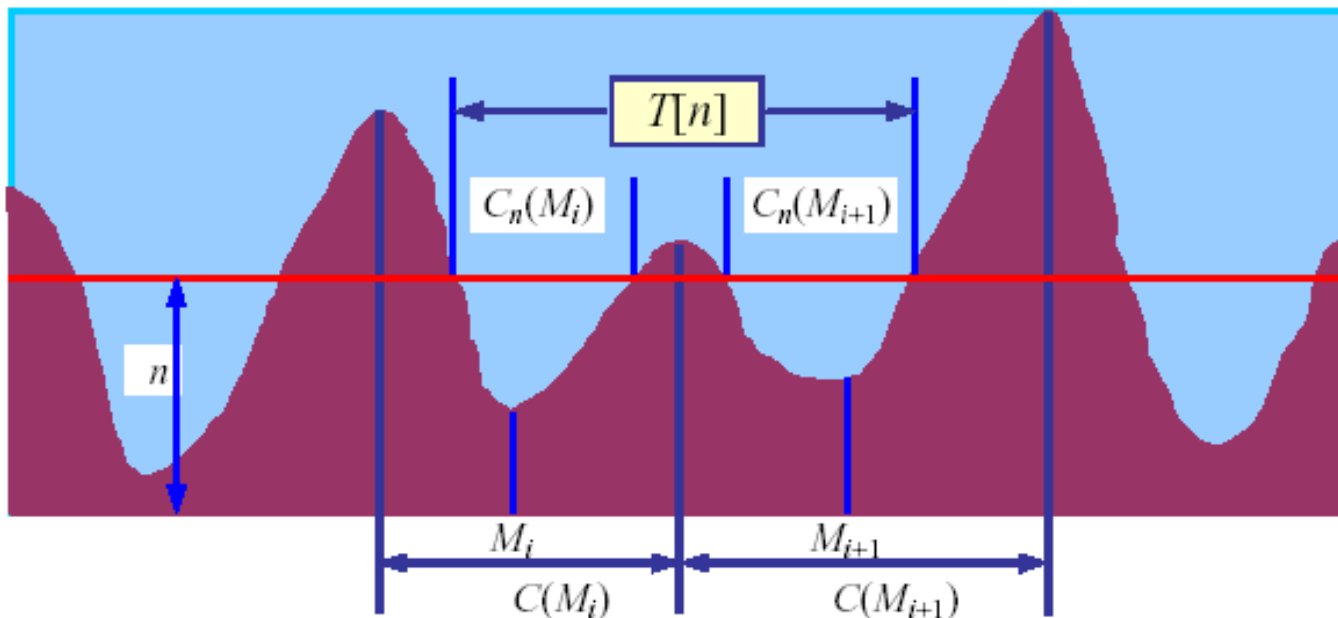
## □ 计算步骤:

- 待分割图象  $f(x, y)$ , 其梯度图象为  $g(x, y)$
- 用  $M_1, M_2, \dots, M_R$  表示  $g(x, y)$  中各局部极小值的象素位置,  $C(M_i)$  为与  $M_i$  对应的区域中的象素坐标集合。
- 用  $n$  表示当前灰度阈值,  $T[n]$  代表记为  $(u, v)$  的象素集合,  $g(u, v) < n$ ,

$$T[n] = \{(u, v) \mid g(u, v) < n\}$$

- 对  $M_i$  所在的区域, 其中满足条件的坐标集合  $C_n(M_i)$  可看作一幅二值图象

$$C_n(M_i) = C(M_i) \cap T[n]$$



# 10.5.3 分水岭分割算法

## □ 计算步骤:

- 用  $C[n]$  代表在灰度阈值为  $n$  时图象中所有满足条件的象素

$$C[n] = \bigcup_{i=1}^R C_n(M_i) \quad C[\max+1] = \bigcup_{i=1}^R C_{\max+1}(M_i)$$

- $C[\max + 1]$  将是所有区域的并集
- $C[n - 1]$  是  $C[n]$  的子集,  $C[n]$  是  $T[n]$  的子集, 所以  $C[n - 1]$  又是  $T[n]$  的子集
- 令  $S$  代表  $T[n]$  中的连通组元集合, 对每个连通组元  $s \in S[n]$ , 有3种可能性:
  - 1、  $s \cap C[n - 1]$  是1个空集, 则  $C[n]$  可由把连通组元  $s$  加到  $C[n - 1]$  中得到
  - 2、  $s \cap C[n - 1]$  里包含  $C[n - 1]$  中的一个连通组元, 则与上同
  - 3、  $s \cap C[n - 1]$  里包含  $C[n - 1]$  中一个以上的连通组元, 则需要在  $s$  中建分水岭

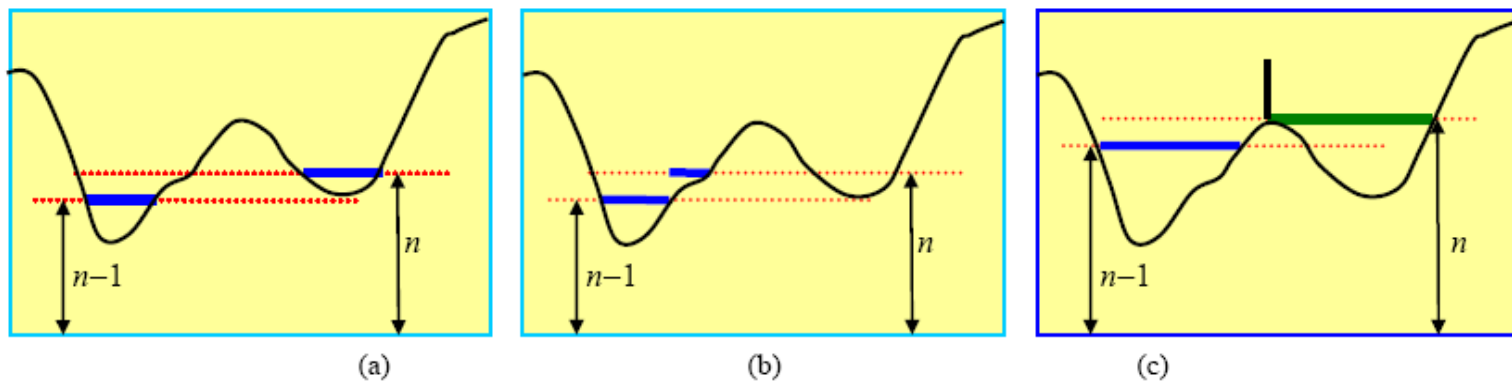
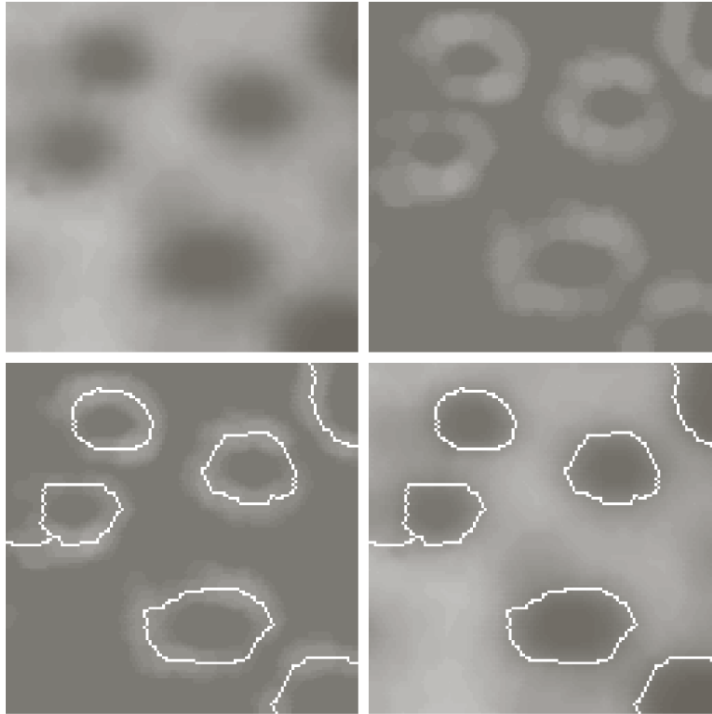


图 5.4.5 计算  $C_n(M_i)$  的示意图

# 10.5.3 分水岭分割算法

## □ 分水岭分割算法的说明

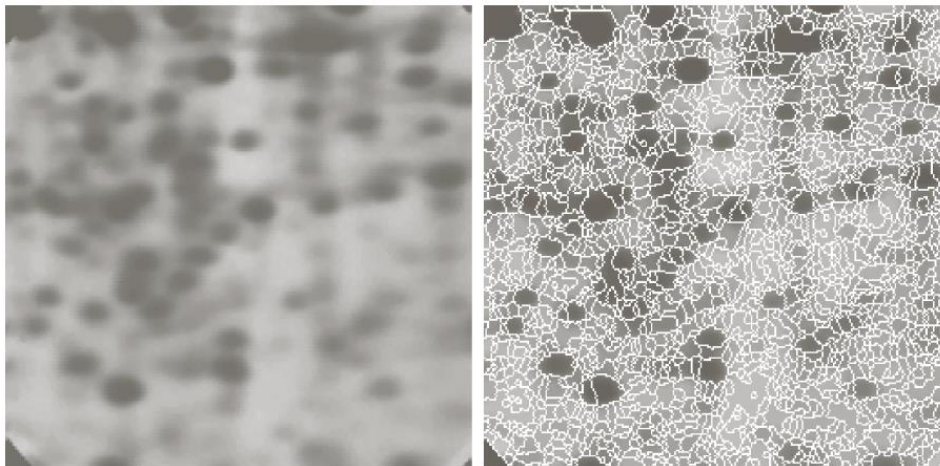


a b  
c d

**FIGURE 10.56**  
(a) Image of blobs.  
(b) Image gradient.  
(c) Watershed lines.  
(d) Watershed lines superimposed on original image.  
(Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

## 10.5.4 标记的使用

- 分水岭算法通常会由于噪声和梯度的其他局部不规则性造成过度分割，如图所示



a b

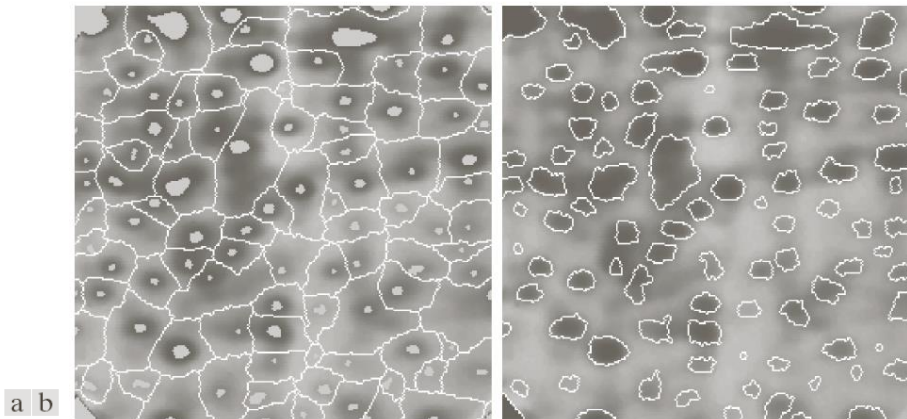
**FIGURE 10.57**

(a) Electrophoresis image. (b) Result of applying the watershed segmentation algorithm to the gradient image. Oversegmentation is evident. (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)

- 解决方案是，加入预处理步骤来限制允许存在的区域的数量
- 用于过度分割的一种方法基于标记。标记是指属于一幅图像的**连通分量**。与感兴趣物体想联系的标记称为**内部标记**，与背景相关联的标记称为**外部标记**。
- 选择标记的典型过程由两个主要步骤组成：（1）预处理；（2）定义标记必须满足的一个准则集合
- 考虑上图，用一个**平滑滤波器**对图像进行过滤是一种合适的方案<sup>6</sup>

## 10.5.4 标记的使用

- 假设我们将内部标记定义为：
  - (1) 被高“海拔”点包围的区域； (2) 区域中形成一个连通分量的那些点； (3) 连通分量中所有的点有相同的灰度值
- 图像经过平滑后，内部标记在a图中以浅灰色、斑点区域显示。分水线定义为外部标记。外部标记有效地将图像分割成了不同的区域，问题简化为将每个区域划分为两部分：单个物体及其背景
- 另一种方法是，求得平滑后图像的梯度，然后将算法限制在只对该特殊区域中包含这一标记的单一分水岭进行操作，如图b



**FIGURE 10.58** (a) Image showing internal markers (light gray regions) and external markers (watershed lines). (b) Result of segmentation. Note the improvement over Fig. 10.47(b). (Courtesy of Dr. S. Beucher, CMM/Ecole des Mines de Paris.)



# 第10章 图像分割

- 10.1 基础知识
- 10.2 点、线和边缘检测
- 10.3 阈值处理
- 10.4 基于区域的分割
- 10.5 用形态学分水岭的分割
- 10.6 分割中运动的应用**





# 10.6.1 空间域技术

## □ 基本方法

- 分别检测在时刻 $t_i$ 和 $t_j$ 获取的两帧图像 $f(x, y, t_i)$ 和 $f(x, y, t_j)$ 之间的变化的最简方法之一是，逐像素比较这两幅图像。形成的一幅差值图像定义为

$$d_{ij}(x, y) = \begin{cases} 1, & |f(x, y, t_i) - f(x, y, t_j)| > T \\ 0, & \text{其他} \end{cases}$$

- 实际上， $d_{ij}(x, y)$ 中的1值项可能由噪声造成，去除它们的简单办法是，在 $d_{ij}(x, y)$ 中构成由1组成的4连通或8连通区域，然后忽略1值项少于预定元素数量的任何区域

## □ 累积差值

- 考虑一个图像帧序列 $f(x, y, t_1), f(x, y, t_2), \dots, f(x, y, t_n)$ ，令 $f(x, y, t_1)$ 为参考图像。一幅累积差值图像（ADI）是通过将参考图像与序列中每幅后续图像进行比较形成的。当参考图像和序列中一幅图像之间在某个像素位置上出现一次差异时，累积图像中每个像素位置的计数器计数一次。
- 考虑如下累积差值图像的三种类型：绝对ADI、正ADI和负ADI。令 $R(x, y)$ 表示参考图像，对于任何 $k > 1$ ，且记住ADI的值是计数，我们对 $(x, y)$ 的所有相关值定义如下：

$$A_k(x, y) = \begin{cases} A_{k-1}(x, y) + 1, & |R(x, y) - f(x, y, k)| > T \\ A_{k-1}(x, y), & \text{其他} \end{cases}$$

# 10.6.1 空间域技术

## □ 累积差值

$$P_k(x, y) = \begin{cases} P_{k-1}(x, y) + 1, & |R(x, y) - f(x, y, k)| > T \\ P_{k-1}(x, y), & \text{其他} \end{cases}$$

$$N_k(x, y) = \begin{cases} N_{k-1}(x, y) + 1, & |R(x, y) - f(x, y, k)| > T \\ N_{k-1}(x, y), & \text{其他} \end{cases}$$

- 式中， $A_k(x, y)$ ， $P_k(x, y)$ 和 $N_k(x, y)$ 分别为遇到序列中的第 $k$ 幅图像后的绝对ADI、正ADI和负ADI
- 绝对、正和负的累积差异图像的计算



a b c

**FIGURE 10.59** ADIs of a rectangular object moving in a southeasterly direction. (a) Absolute ADI. (b) Positive ADI. (c) Negative ADI.

# 10.6.1 空间域技术

## □ 建立参考图像

- 前两节的技术成功的关键是，找到一幅可以与后续图像进行比较的参考图像
- 实际上，仅用固定元素得到一幅参考图像并不总是可行的，所以有必要根据一组包含一个或多个运动物体的图像来构建一幅参考图像。
- 生成参考图像的过程如下描述：考虑将图像序列中的第一幅图像作为参考图像。当一个非固定成分完全移出其参考帧的位置时，当前帧中对应背景可复制到最初被参考帧中物体所占据的位置。
- 当所有的移动物体均已完全移出原来的位置时，就创建了一幅仅包含固定成分的参考图像。通过监视正ADI的变化，可确定物体的位移
- 构造一幅参考图像



**FIGURE 10.60** Building a static reference image. (a) and (b) Two frames in a sequence. (c) Eastbound automobile subtracted from (a) and the background restored from the corresponding area in (b). (Jain and Jain.)



# 第11章 表示和描述

---

11.1 表示

11.2 边界描绘子

11.3 区域描绘子

11.4 使用主成分进行描绘

11.5 关系描绘子



# 第11章 表示和描述

## 11.1 表示

11.2 边界描绘子

11.3 区域描绘子

11.4 使用主成分进行描绘

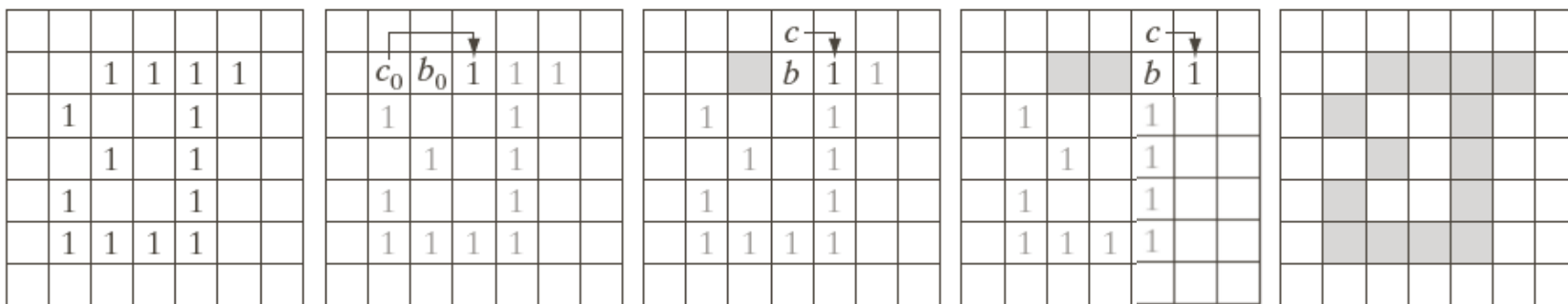
11.5 关系描绘子

# 11.1.1 边界追踪

□ 边界追踪：对一个区域上的边界点以逆时 或 时 方向排序

## 边界追踪步：

1. 令起始点  $b_0$  为图像中左上角标记为1的点， $c_0$  为  $b_0$  西侧的点。
2. 令  $b = b_1$  和  $c = c_1$ 。
3. 从  $c$  开始按 时 方向进行，令  $b$  的8个 点为  $n_1, n_2, \dots, n_8$ , 找到标记为1的第一个  $n_k$ 。
4. 复步 3和步 4, 直到  $b = b_0$  且找到的下一个边界点为  $b_1$ 。

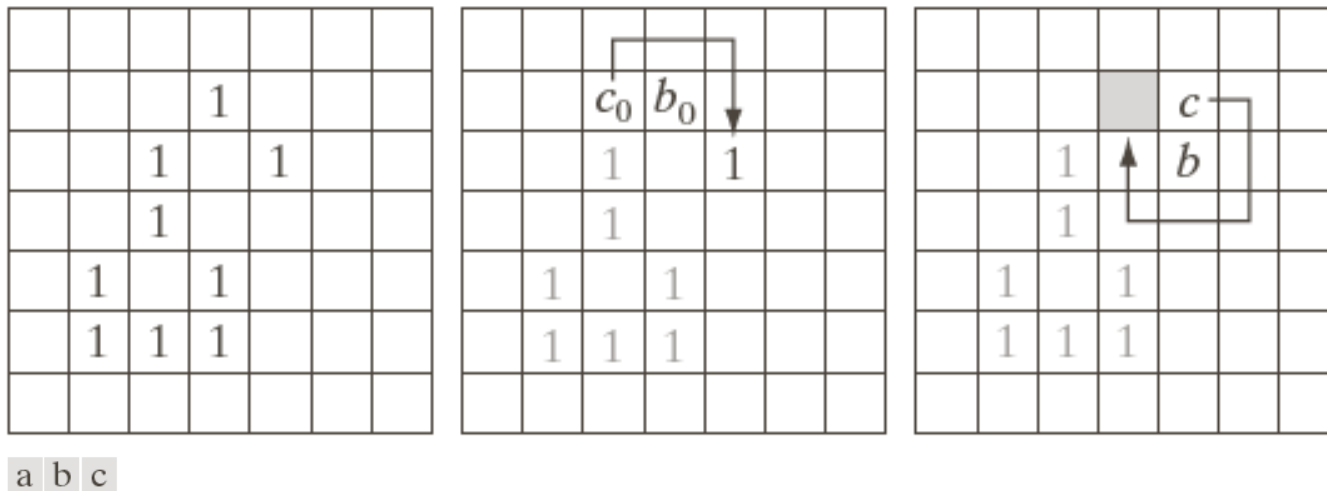


a b c d e

**FIGURE 11.1** Illustration of the first few steps in the boundary-following algorithm. The point to be processed next is labeled in black, the points yet to be processed are gray, and the points found by the algorithm are labeled as gray squares.

# 11.1.1 边界追踪

为了了解边界跟踪算法中规定的停止规则的必要性，考虑如下示例：



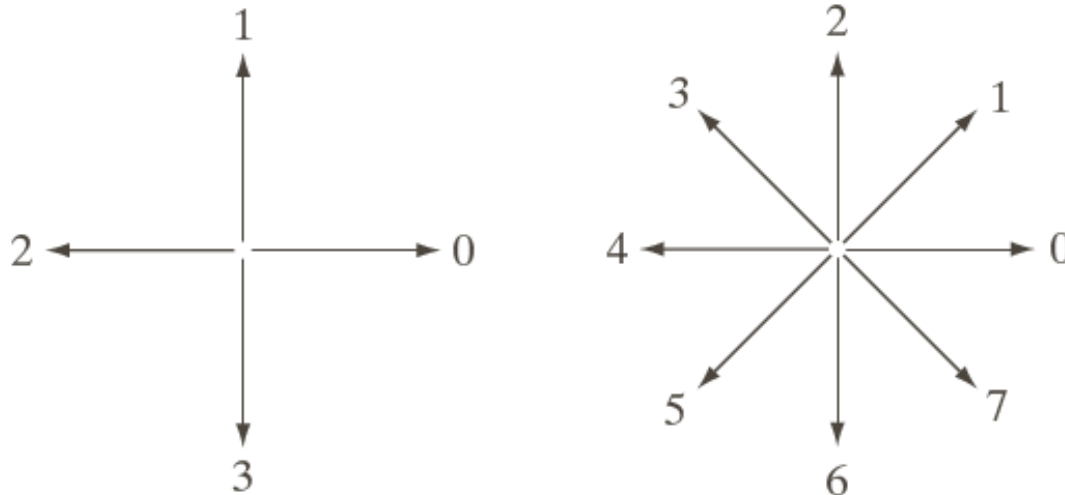
**FIGURE 11.2** Illustration of an erroneous result when the stopping rule is such that boundary-following stops when the starting point,  $b_0$ , is encountered again.

上述边界跟踪算法可以直接用于图像区域。



# 11.1.2 码

- 码用于表示由 次 接的具有指定 度和方向的直线段组成的边界
- 通常，这种表示基于这些线段的4连接或8连接
- 每个线段的方向使用一种数字编号方案编码
  - 佛 曼 码

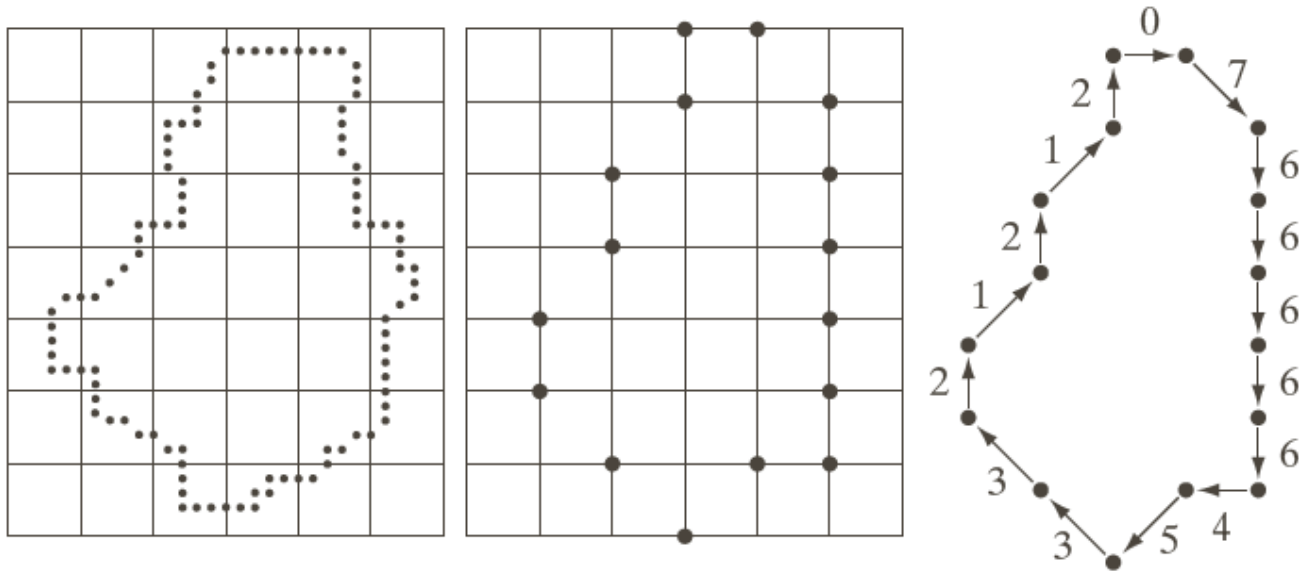


a b

**FIGURE 11.3**  
Direction numbers for (a) 4-directional chain code, and (b) 8-directional chain code.

# 11.1.2 码

- 码示例
  - 左：叠加有 取样网格的数字边界；
  - 中： 取样结果；
  - 右： 8方向 码边界。



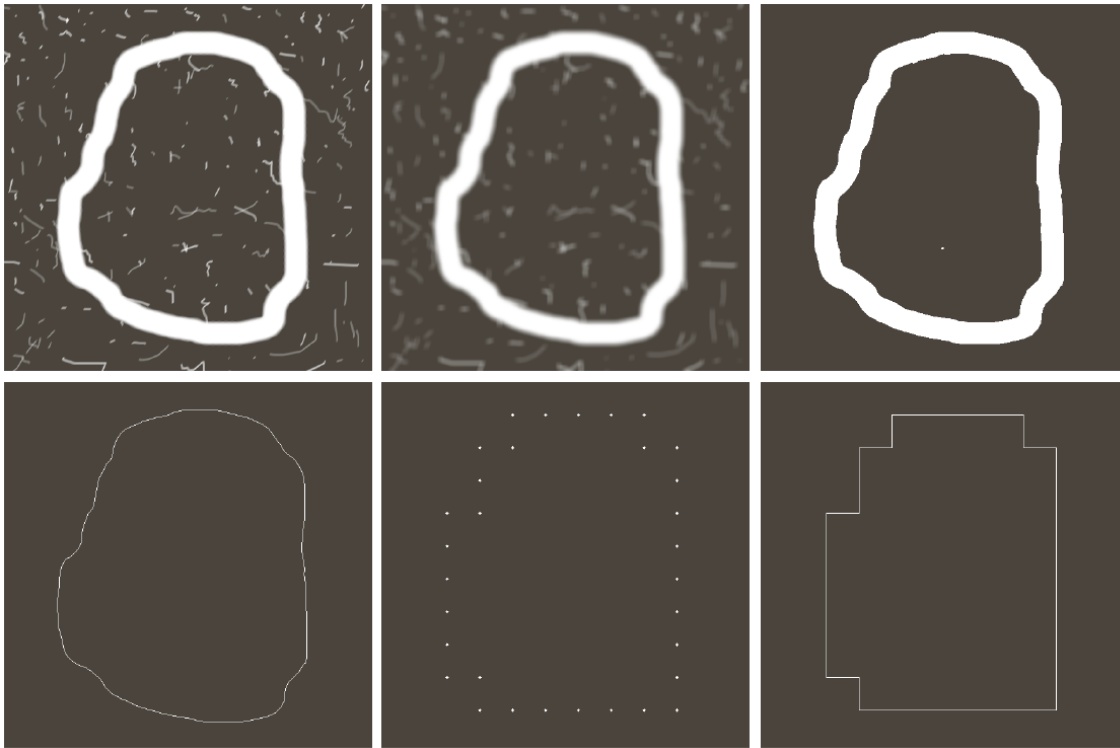
a b c

**FIGURE 11.4**  
(a) Digital boundary with resampling grid superimposed.  
(b) Result of resampling.  
(c) 8-directional chain-coded boundary.

- 码归一化
  - 起点归一化，旋转归一化，尺寸归一化

# 11.1.2 码

- 例：弗曼码及它的某些变化
  - 简单近似保留了原始边界的主要特征

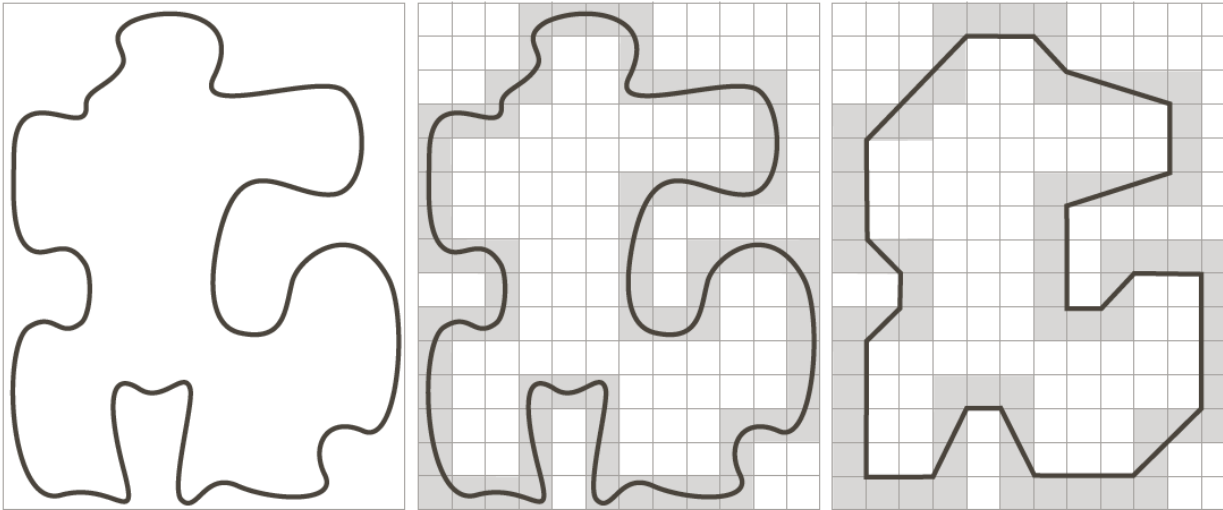


a	b	c
d	e	f

**FIGURE 11.5** (a) Noisy image. (b) Image smoothed with a  $9 \times 9$  averaging mask. (c) Smoothed image, thresholded using Otsu's method. (d) Longest outer boundary of (c). (e) Subsampled boundary (the points are shown enlarged for clarity). (f) Connected points from (e).

# 11.1.3 最小周 多边形近似

直观方法理解MPP：将连接单元的 合封入一条边界中(b)，将边界想象成橡皮条，橡皮条收缩会产生最小周 的多边形。

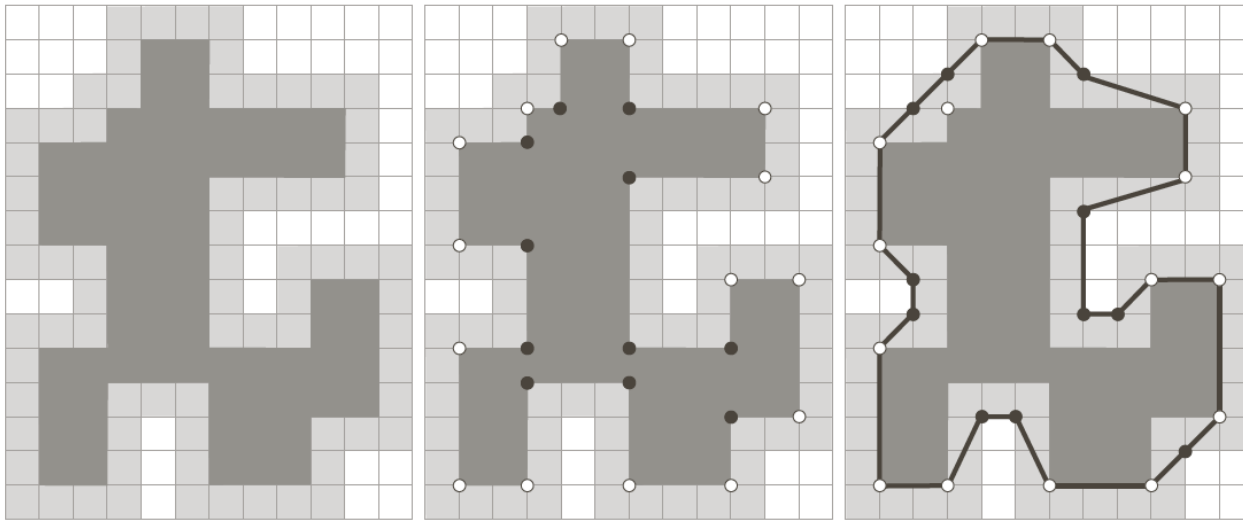


a b c

**FIGURE 11.6** (a) An object boundary (black curve). (b) Boundary enclosed by cells (in gray). (c) Minimum-perimeter polygon obtained by allowing the boundary to shrink. The vertices of the polygon are created by the corners of the inner and outer walls of the gray region.

# 11.1.3 最小周 多边形近似

MPP 的点要么与内墙中的凸点（白点）一致，要么与外墙中的凹点（黑点）一致。  
只有内墙的凸点和外墙的凹点才能成为 MPP 的点。



a b c

**FIGURE 11.7** (a) Region (dark gray) resulting from enclosing the original boundary by cells (see Fig. 11.6). (b) Convex (white dots) and concave (black dots) vertices obtained by following the boundary of the dark gray region in the counterclockwise direction. (c) Concave vertices (black dots) displaced to their diagonal mirror locations in the outer wall of the bounding region; the convex vertices are not changed. The MPP (black boundary) is superimposed for reference.

# 11.1.3 最小周 多边形近似

- MPP 算法示例
  - MPP的优点之一：边界平滑



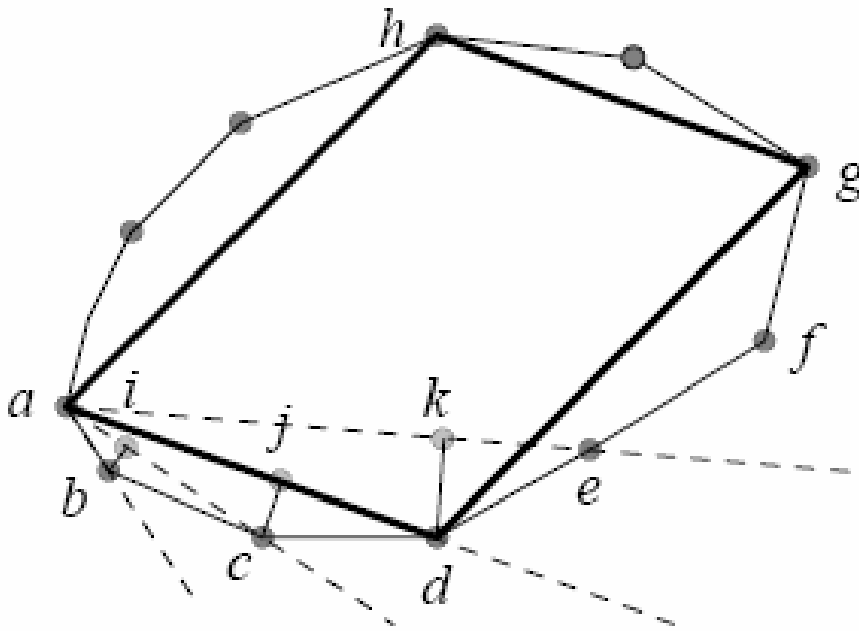
a	b	c
d	e	f
g	h	i

**FIGURE 11.8**

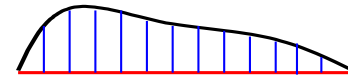
(a)  $566 \times 566$  binary image.  
 (b) 8-connected boundary.  
 (c) through (i), MMPs obtained using square cells of sizes 2, 3, 4, 6, 8, 16, and 32, respectively (the vertices were joined by straight lines for display). The number of boundary points in (b) is 1900. The numbers of vertices in (c) through (i) are 206, 160, 127, 92, 66, 32, and 13, respectively.

# 11.1.4 其他多边形近似

**聚合技术：**沿一条边界来聚合一些点，直到拟合这些聚合点的直线的最小均方误差超过某个使设的值。



误差计算

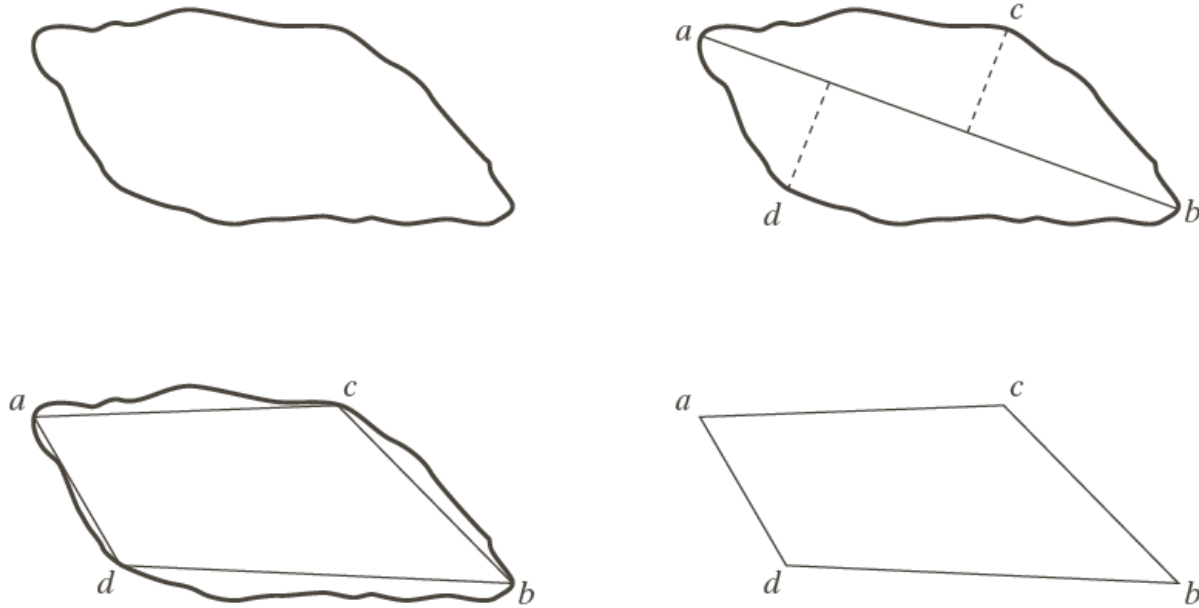


与起点有关  
贪婪算法



# 11.1.4 其他多边形近似

**分裂技术：**分裂边界线段的一种方法是将线段不断地细分成两分，直到满足规定的准则为止。

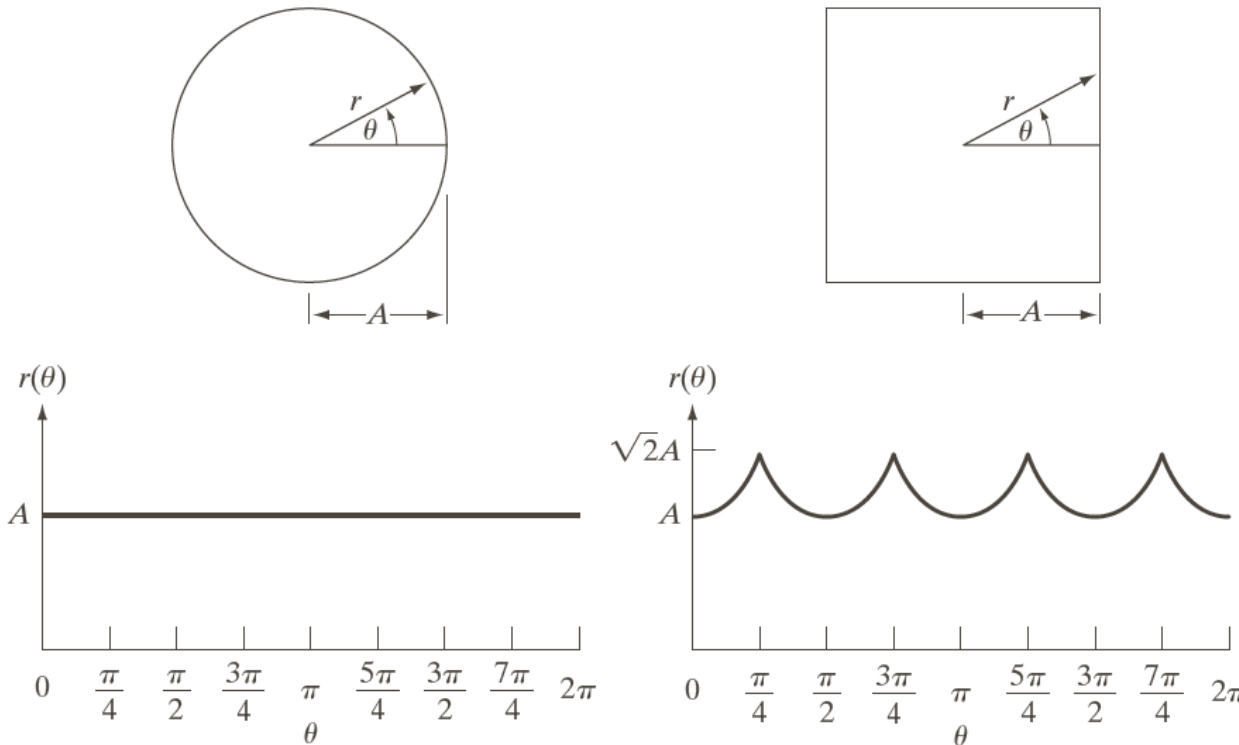


a	b
c	d

**FIGURE 11.9**  
(a) Original boundary.  
(b) Boundary divided into segments based on extreme points. (c) Joining of vertices. (d) Resulting polygon.

# 11.1.5 标记图

标记图是边界的一维函数表示，它可以使用各种方式来生成。一种最简单的方法是以角度函数的形式画出质心到边界的距离，如下：



a b

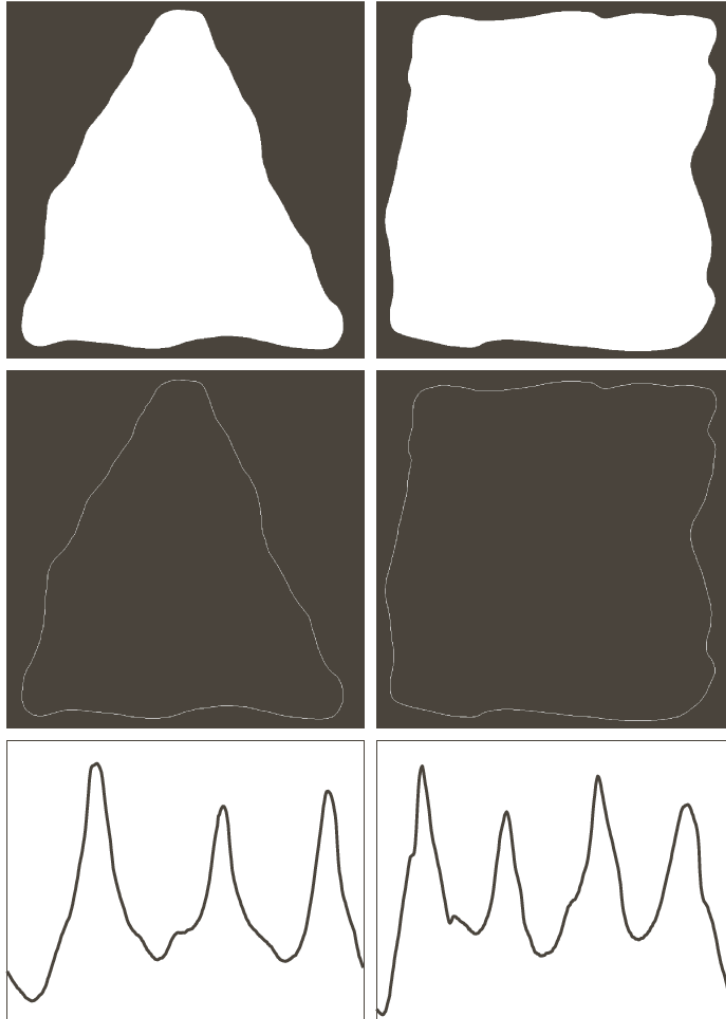
**FIGURE 11.10**

Distance-versus-angle signatures. In (a)  $r(\theta)$  is constant. In (b), the signature consists of repetitions of the pattern

$$r(\theta) = A \sec \theta \text{ for } 0 \leq \theta \leq \pi/4 \text{ and } r(\theta) = A \csc \theta \text{ for } \pi/4 < \theta \leq \pi/2.$$

# 11.1.5 标记图

两个简单物体的标记图：



a	b
c	d
e	f

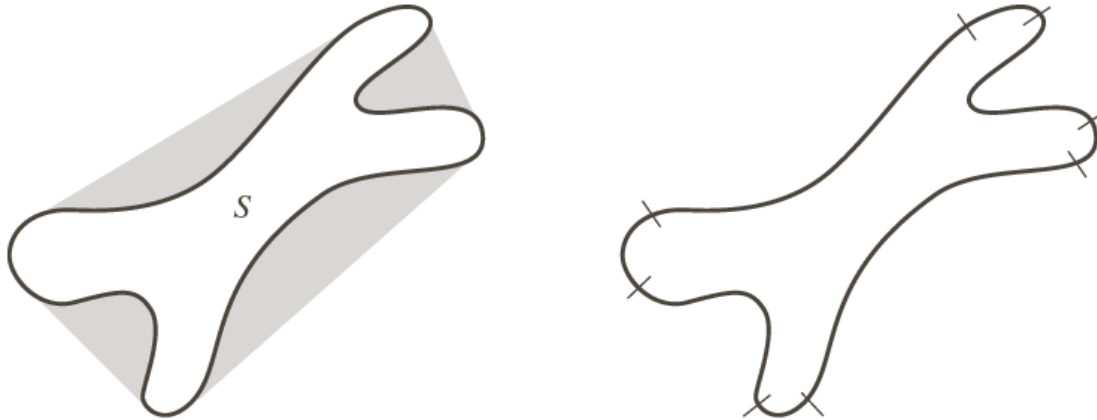
**FIGURE 11.11**

Two binary regions, their external boundaries, and their corresponding  $r(\theta)$  signatures. The horizontal axes in (e) and (f) correspond to angles from  $0^\circ$  to  $360^\circ$ , in increments of  $1^\circ$ .

# 11.1.6 边界线段

将边界分解成线段能 低边界的复杂性，从而简化描述过程。

图左显示了物体（ 合 $S$ ）及其凸缺（ 影区）。区域边界可以按照如下方式来分割：追踪  $S$  的轮廓，并标记进入或离开一个凸缺的转变点。



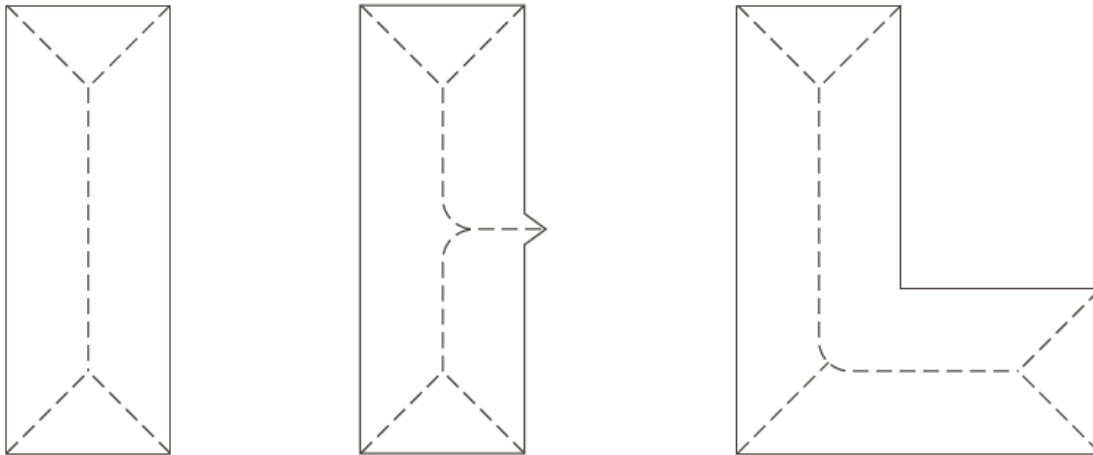
a b

**FIGURE 11.12**  
(a) A region,  $S$ , and its convex deficiency (shaded).  
(b) Partitioned boundary.

# 11.1.7 架

表示平面区域内结构形状的一种重要方法是将其简化为图形。这种简化可以通过一种细化(也成为骨架化)算法得到该区域的骨架来实现。

骨架可以由中轴变换 (MAT) 来定义：边界为  $B$  的区域  $R$ ，对  $R$  中的每个点  $p$ ，我们在  $B$  中找到与其最接近的点，如果  $p$  有多个这样的点，则认为  $p$  属于  $R$  的中轴 (骨架)。



a b c

**FIGURE 11.13**  
Medial axes  
(dashed) of three  
simple regions.



# 11.1.7 架

- 直接利用定义计算 架点，代价太大。
- 实 中可 用逐次消 边界点的迭代细化算法。
- 这个过程中，有3个 制条件 要满足：
  - (1) 不能删 端点
  - (2) 不能破坏连接性
  - (3) 不能导致区域的过度腐蚀



# 11.1.7 架

Step1 :

(1) 标记同时满足下列条件的边界点

i)  $2 \leq N(p_1) \leq 6$ ; ii)  $S(p_1) = 1$ ; iii)  $p_2 \cdot p_4 \cdot p_6 = 0$ ; iv)  $p_4 \cdot p_6 \cdot p_8 = 0$ ;

其中 $N(p_1)$ 是 $p_1$ 邻域的非零点数;

$S(p_1)$ 是从 $p_2$ 开始顺时针转一圈后, $0 \rightarrow 1$ 变化的次数。

(2) 当所有边界点都检验完毕后, 将标记点去掉。

$p_9$	$p_2$	$p_3$
$p_8$	$p_1$	$p_4$
$p_7$	$p_6$	$p_5$

Step2 :

(1) 标记同时满足下列条件的边界点

i)  $2 \leq N(p_1) \leq 6$ ; ii)  $S(p_1) = 1$ ; iii)  $p_2 \cdot p_4 \cdot p_8 = 0$ ; iv)  $p_2 \cdot p_6 \cdot p_8 = 0$ ;

(2) 当所有边界点都检验完毕后, 将标记点去掉。

Step3: 重复Setp1,2直到没有点满足标记条件



# 11.1.7 架

## □ 算法解

i)  $2 \leq N(p_1) \leq 6$ ; ii)  $S(p_1) = 1$ ; iii)  $p_2 \cdot p_4 \cdot p_6 = 0$ ; iv)  $p_4 \cdot p_6 \cdot p_8 = 0$ ;  
 i)  $2 \leq N(p_1) \leq 6$ ; ii)  $S(p_1) = 1$ ; iii)  $p_2 \cdot p_4 \cdot p_8 = 0$ ; iv)  $p_2 \cdot p_6 \cdot p_8 = 0$ ;

- 条件 i) 保证不消去线段端点和过多侵蚀区域。
- 条件 ii) 保证不中断原本连通的点。
- 条件 iii)和 iv)保证消去的点不是 架点

$p_9$	$p_2$	$p_3$
$p_8$	$p_1$	$p_4$
$p_7$	$p_6$	$p_5$

0	1	0
0	$p$	0
0	0	0

1	1	1
1	$p$	1
1	0	1

0	0	0
1	$p$	1
0	0	0

0	1	1
0	$p$	0
1	1	0

1	1	0
1	$p$	0
0	0	0

0	0	0
0	$p$	1
0	1	1



# 第11章 表示和描述

---

11.1 表示

11.2 边界描绘子

11.3 区域描绘子

11.4 使用主成分进行描绘

11.5 关系描绘子



# 11.2.1 简单的边界描绘子

- 边界的 度
- 边界 B 的直径

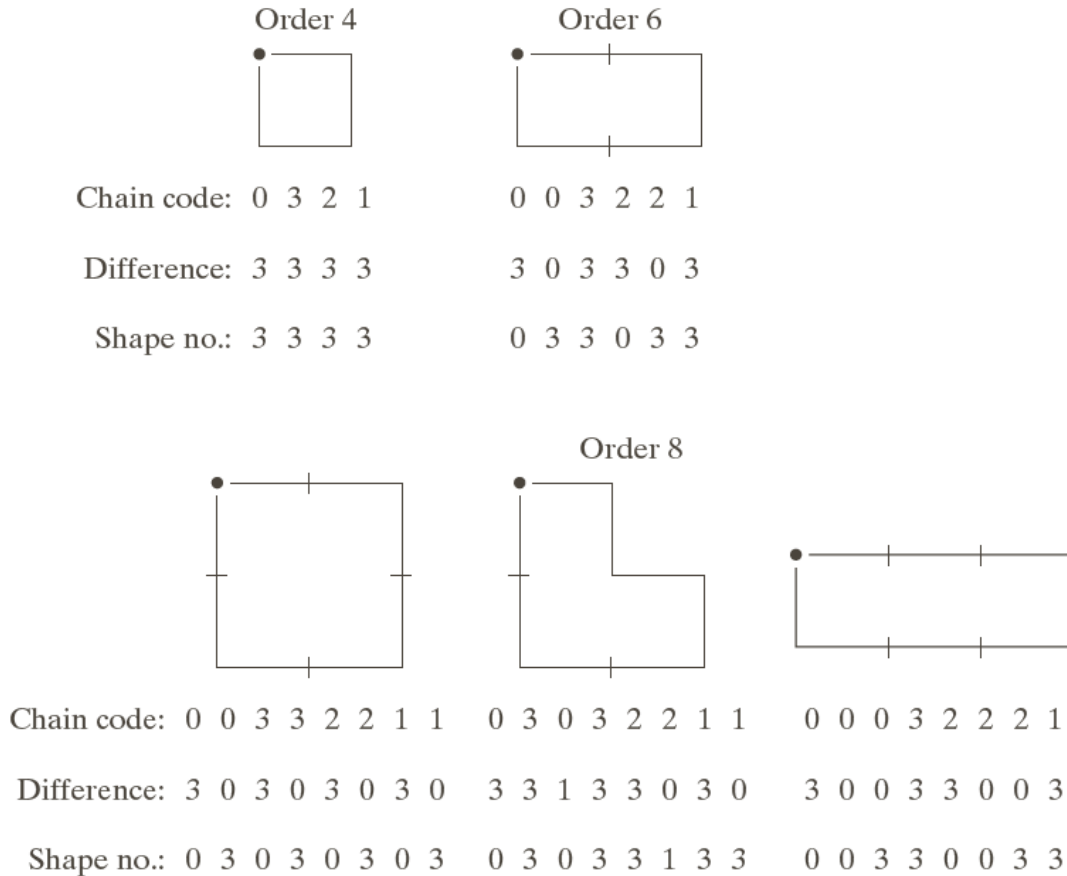
$$Diam(B) = \max_{i,j} [D(p_i, p_j)]$$

直径的值和连接组成该直径的两个端点的直线段 (边界的 轴) 的方向是边界的有用描绘子。

- 偏心率
  - 边界的短轴定义为与 轴垂直的直线。 轴与短轴之比称为边界的偏心率，偏心率也是一个有用的描绘子。
- 曲率
  - 斜率的变化率
  - 近似表达：用相 边界线段的斜率差作为这两条线段交点处的斜率

# 11.2.2 形状数

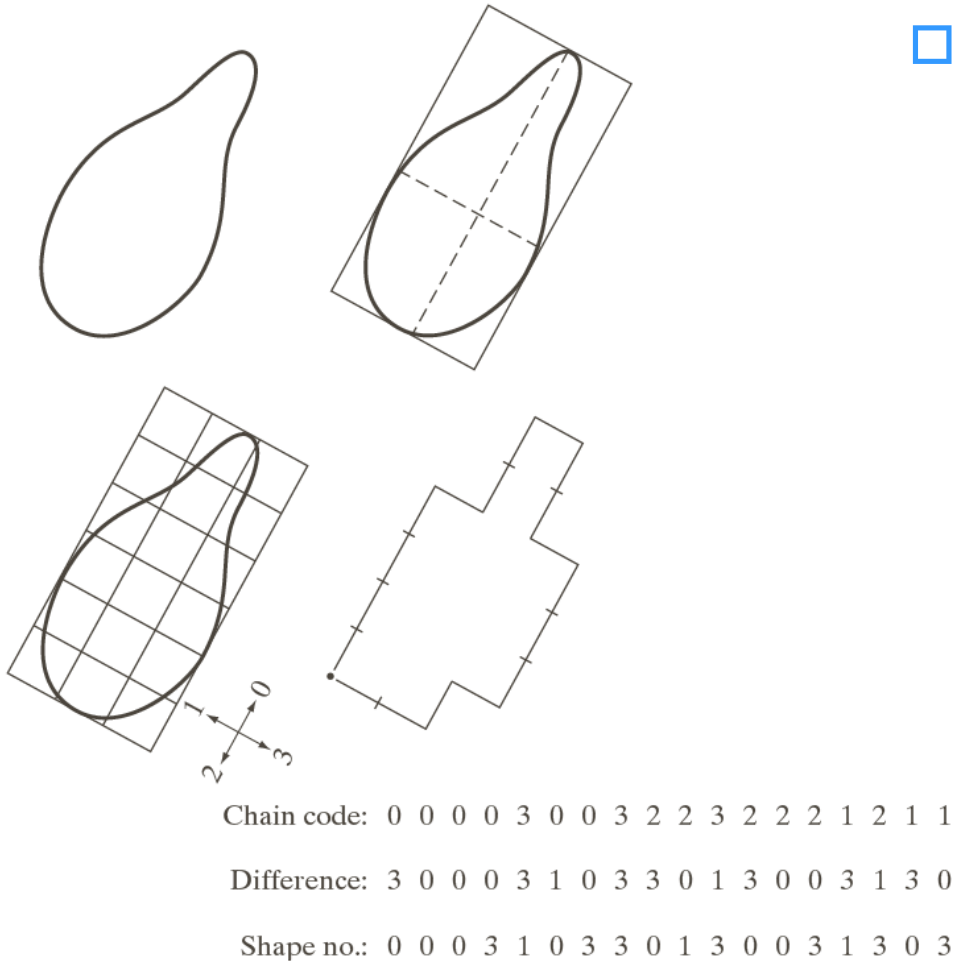
如下所示，为 4, 6, 8 的所有形状数，以及它们的链码表示、一次差分



**FIGURE 11.17**  
All shapes of order 4, 6, and 8. The directions are from Fig. 11.3(a), and the dot indicates the starting point.

# 11.2.2 形状数

- 归一化网格方向
- 获得指定 数N的形状数
  - 从所有满足(  $2(m+n)=N$  )矩形中，取 短轴比例与区域最接近的 一个。
  - 对外接矩形进行 $m \times n$ 网格划分，求出边界点（ 积50%以上包在边界内的正方形划入内 ）。
  - 求出 码、差分码以及形状数。



a	b
c	d

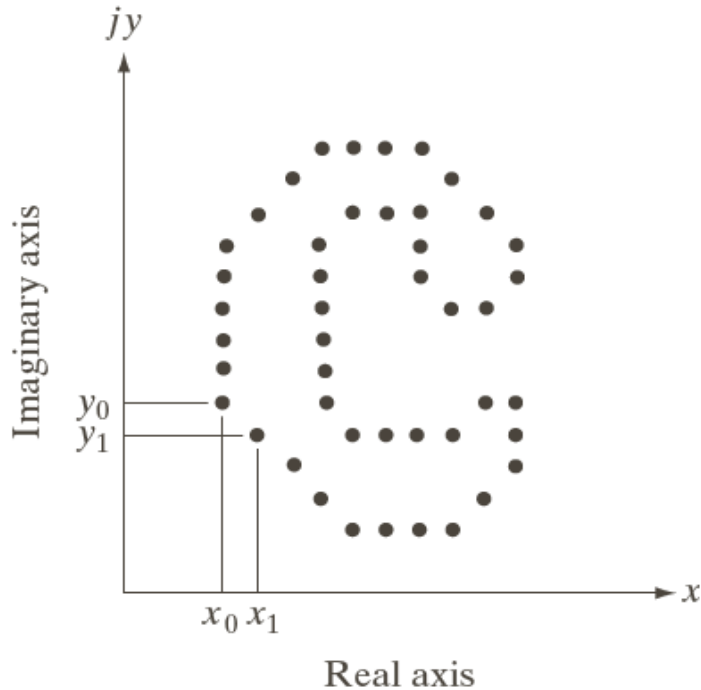
**FIGURE 11.18**  
Steps in the generation of a shape number.

# 11.2.3 傅 叶描绘子

- 对于平 内一个  $K$  点数字边界，如下图。边界的坐标序列可以近似为：

$$\hat{S}(k) = \frac{1}{K} \sum_{u=0}^{P-1} a(u) e^{j2\pi uk/P}$$

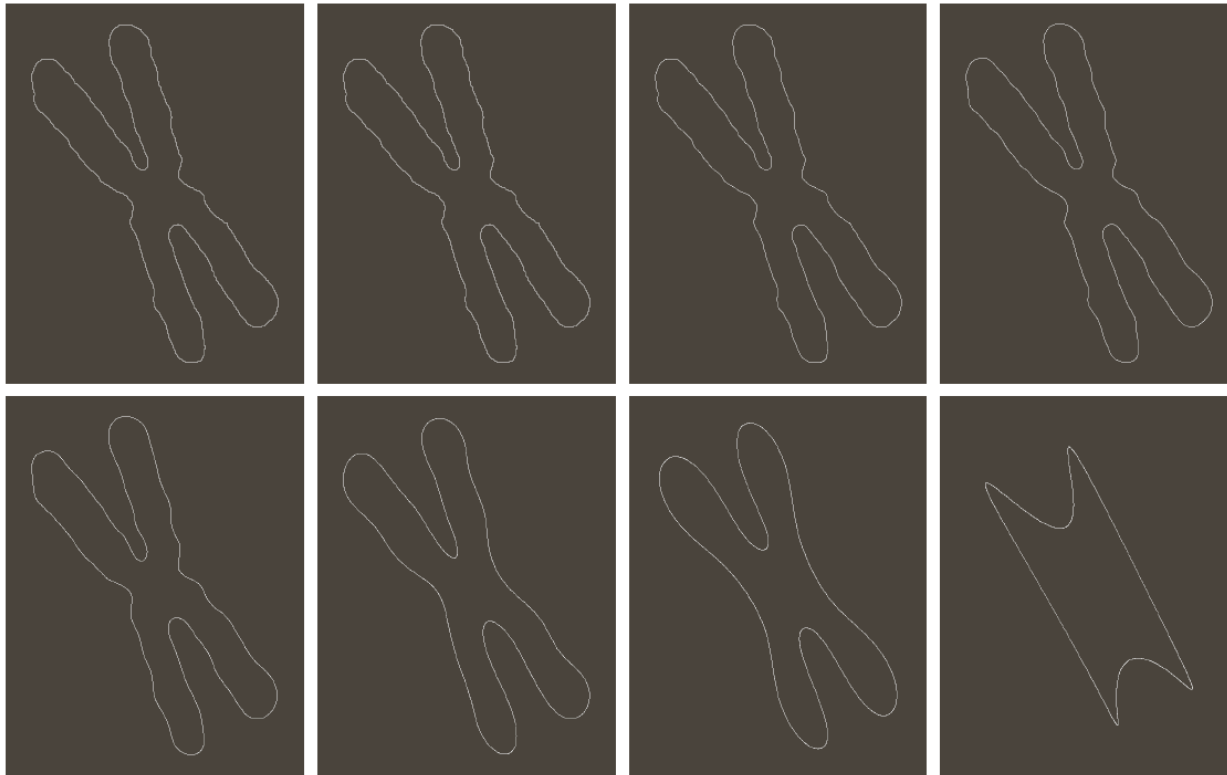
- 由傅 叶变换可知， 成分说明精细细节，而低 成分决定全局形状。少数的傅 叶描绘子就可以捕获边界的大体特征。



**FIGURE 11.19**  
A digital boundary and its representation as a complex sequence. The points  $(x_0, y_0)$  and  $(x_1, y_1)$  shown are (arbitrarily) the first two points in the sequence.

# 11.2.3 傅 叶描绘子

分别使用 1434, 286, 144, 72, 36, 18, 8个傅 叶描绘子 建染色体边界:



a	b	c	d
e	f	g	h

**FIGURE 11.20** (a) Boundary of human chromosome (2868 points). (b)–(h) Boundaries reconstructed using 1434, 286, 144, 72, 36, 18, and 8 Fourier descriptors, respectively. These numbers are approximately 50%, 10%, 5%, 2.5%, 1.25%, 0.63%, and 0.28% of 2868, respectively.





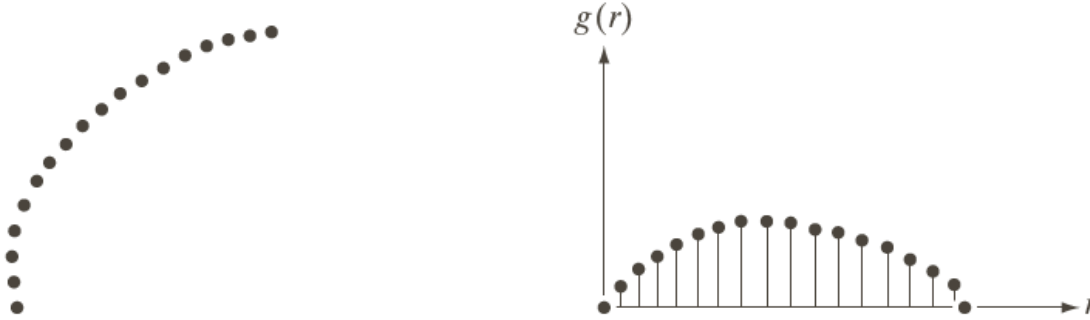
# 11.2.3 傅 叶描绘子

傅 叶描绘子的一些基本性质：

Transformation	Boundary	Fourier Descriptor
Identity	$s(k)$	$a(u)$
Rotation	$s_r(k) = s(k)e^{j\theta}$	$a_r(u) = a(u)e^{j\theta}$
Translation	$s_t(k) = s(k) + \Delta_{xy}$	$a_t(u) = a(u) + \Delta_{xy}\delta(u)$
Scaling	$s_s(k) = \alpha s(k)$	$a_s(u) = \alpha a(u)$
Starting point	$s_p(k) = s(k - k_0)$	$a_p(u) = a(u)e^{-j2\pi k_0 u/K}$

# 11.2.4 统计矩

- 边界线段的形状可使用统计矩来定 描述。
- 先，将线段的两个端点连接起来，然后旋转该直线线段，直至其为水平线段（如下）。将 $g$ 归一化为单位 积，可计算幅度  $g$  的矩。



a b

**FIGURE 11.21**  
(a) Boundary segment.  
(b) Representation as a 1-D function.

$$\mu_n(r) = \sum_{i=0}^{K-1} (r_i - m)^n g(r_i) \quad \text{其中,} \quad m = \sum_{i=0}^{K-1} r_i g(r_i)$$

- $\mu_n$ 与  $f(r)$  的形状有直接联系
  - $\mu_2$ 描述了曲线相对于均值的扩散程度
  - $\mu_3$ 描述了曲线相对于均值的对称性



# 第11章 表示和描述

11.1 表示

11.2 边界描绘子

11.3 区域描绘子

11.4 使用主成分进行描绘

11.5 关系描绘子



# 11.3.1 一些简单的区域描绘子

- **区域的 积** 定义为该区域中像素的数 。
- 区域的周 是其边界的 度。
- 这两个描绘子 繁用于度 一个区域的致密性，即  $(\text{周})^2 / \text{积}$  。
- 另一个致密性描绘子为**圆度率**，即一个区域的 积与具有相同周的一个圆（最致密形状）的 积之比，如下：

$$R_c = \frac{4\pi A}{P^2}$$

# 11.3.2 拓扑描绘子

- 对一幅二值图象A，可以定义两个欧拉数

(1) 4-连通欧拉数 $E_4(A)$

4-连通的目标个数减去8-连通的孔数

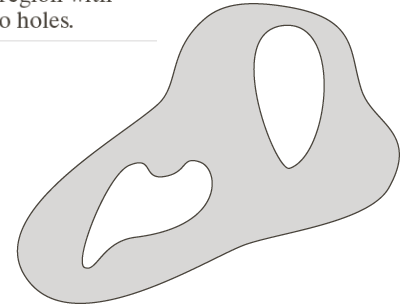
$$E_4(A) = C_4(A) - H_8(A)$$

(2) 8-连通欧拉数 $E_8(A)$

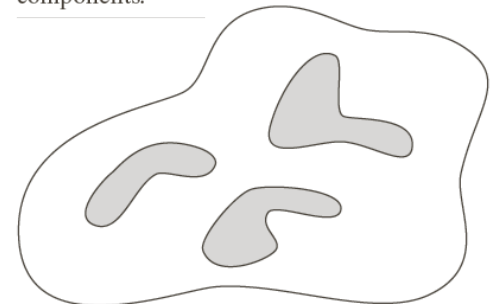
8-连通的目标个数减去4-连通的孔数

$$E_8(A) = C_8(A) - H_4(A)$$

**FIGURE 11.23**  
A region with  
two holes.



**FIGURE 11.24**  
A region with  
three connected  
components.



# 11.3.2 拓扑描绘子

## 多边形网

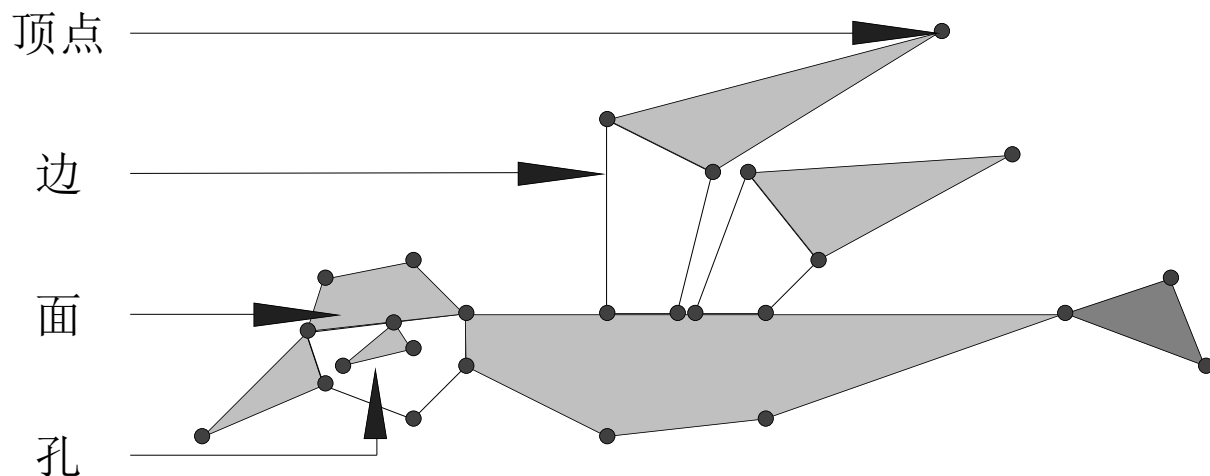
全由直线段（包围）构成的区域 合

欧拉公式  $V - Q + F = E = C - H$

$V$ : 点数

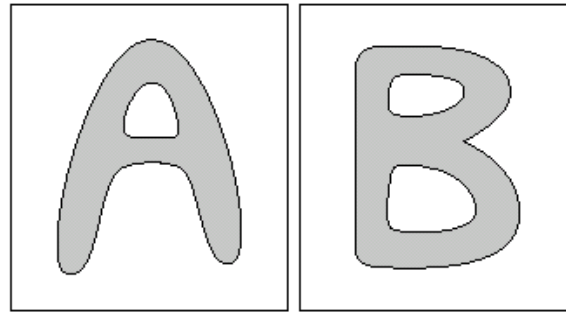
$Q$ : 边线数

$F$ : 数



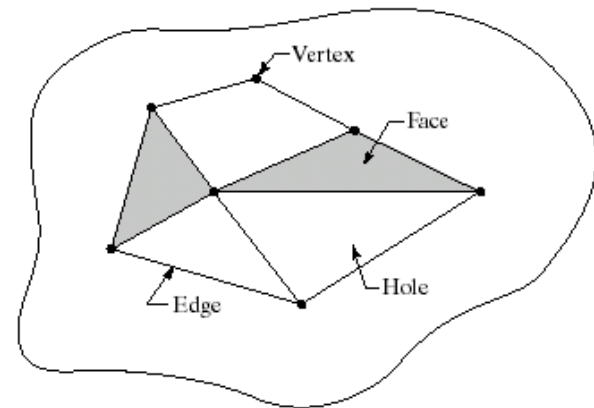
$V = 26, Q = 35, F = 7, C = 1, H = 3, E = -2$

# 11.3.2 拓扑描绘子



Euler number are 0 and -1 respectively

$$V - Q + F = E = C - H$$

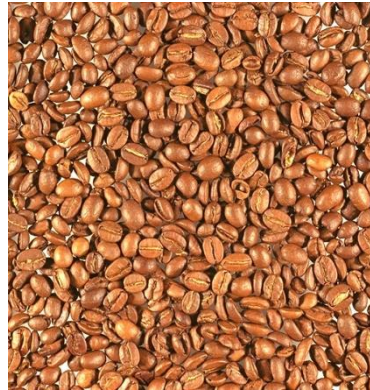


$$V = 7, Q = 11, F = 2, C = 1, H = 3, E = -2$$

# 11.3.3 纹理

## 什么是纹理

- 纹理与尺度有密切联系。任何物体的表面，如果一直放大下去进行观察的话一定会显现出纹理。
- 纹理具有区域性质的特点。对单个像素来说讨论纹理是没有意义的。纹理可用来辨识图象中的不同区域。







# 11.3.3 纹理

## 纹理分析的方法

### (1) 统计法

利用对图象灰度分布和关系的统计规则

### (2) 结构法

根据描述几何关系的放置/排列规则来描述纹理基元

### (3) 频谱法

根据傅里叶频谱的分布，特别是高能量窄脉冲来描述纹理的全局周期性质



# 11.3.3 纹理

## 统计法：灰度共生矩

- 思想

纹理是由灰度分布在空间位置上反复出现而形成  
纹理图像在图像空间中相隔某距离的两像素间会存在一  
定的灰度关系，即灰度的空间相关性  
共生矩阵方法用条件概率来反映纹理，是相邻像素的灰  
度相关性的表现。

- 方法

根据图像像素之间的位置关系（距离，方向），构造一  
种矩阵，作为纹理的描述  
矩阵的行坐标和列坐标表示不同的灰度，考察一对对象  
素出现的频度，以此作为矩阵中的元素



# 11.3.3 纹理

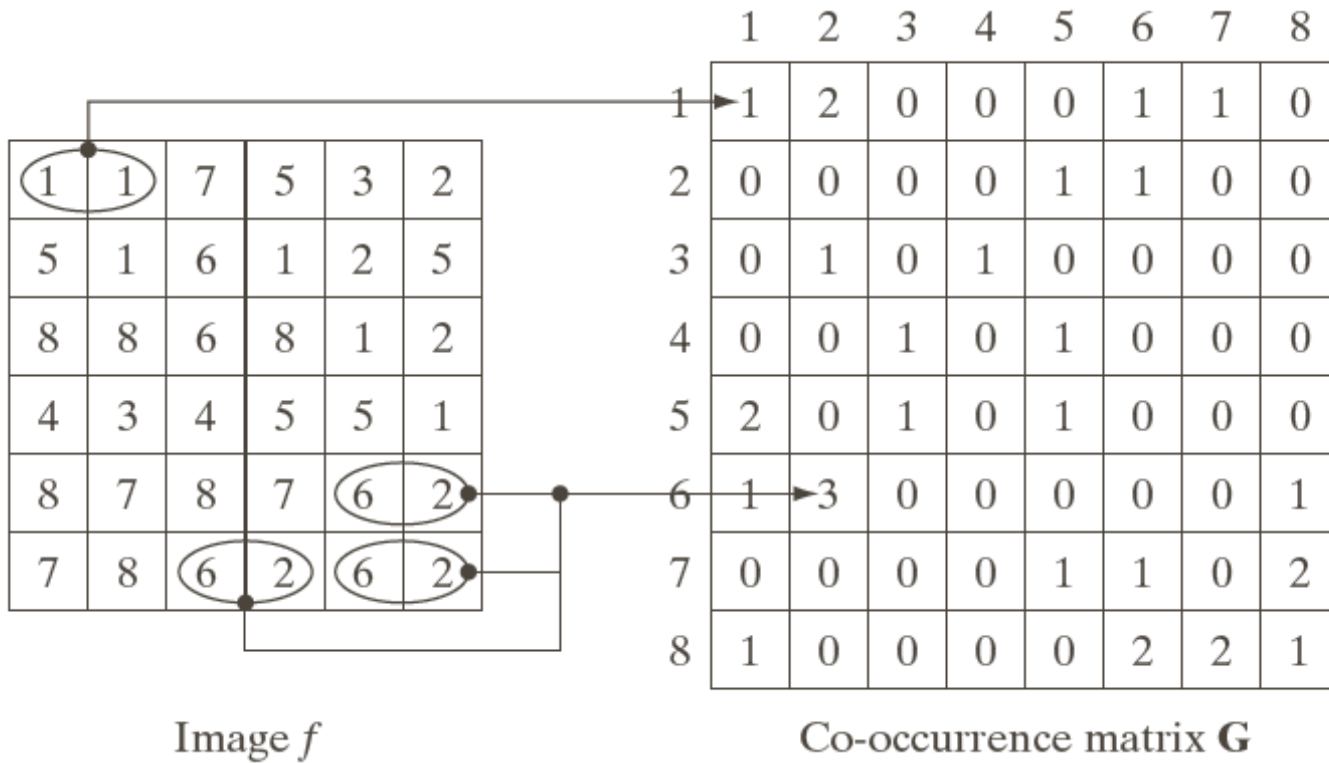
- 设  $S$  为目标区域  $R$  中具有特定空间联系（可由位置算子确定）的像素对的集合
- 共生矩阵  $\mathbf{P}$  中的元素（#代表数量）

$$p(g_1, g_2) = \frac{\#\{[(x_1, y_1), (x_2, y_2)] \in S \mid f(x_1, y_1) = g_1 \ \& \ f(x_2, y_2) = g_2\}}{\#S}$$

- 分子：具有某种空间关系、灰度值分别为  $g_1$  和  $g_2$  的像素对的个数
- 分母：像素对的总和个数

# 11.3.3 纹理

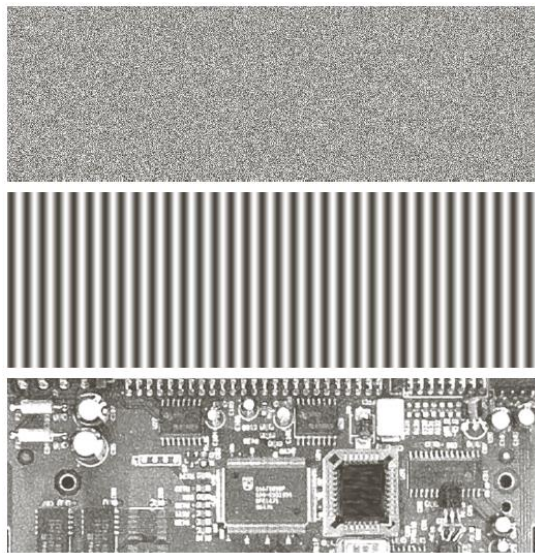
生成共生矩 的方式:



**FIGURE 11.29**  
How to generate a co-occurrence matrix.

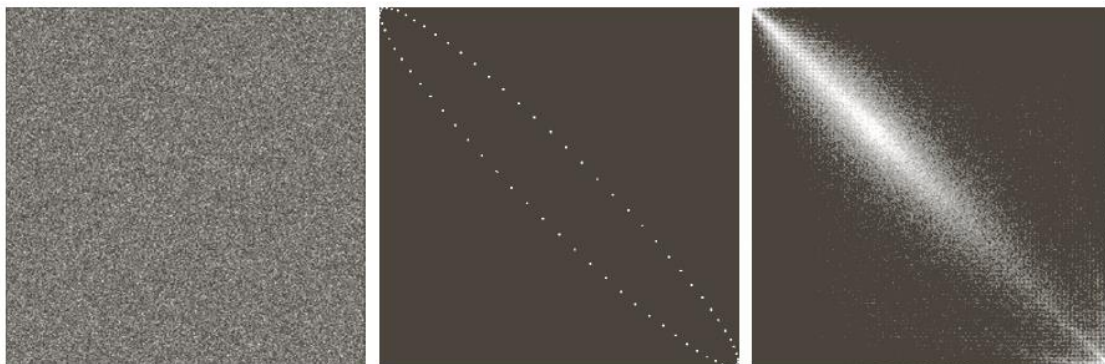
# 11.3.3 纹理

机模式，周期模式（正弦波），混合纹理模式图片（上图）及其共生矩（下图）



a  
b  
c

**FIGURE 11.30**  
Images whose pixels have (a) random, (b) periodic, and (c) mixed texture patterns. Each image is of size  $263 \times 800$  pixels.



a b c

**FIGURE 11.31**  
 $256 \times 256$  co-occurrence matrices,  $G_1$ ,  $G_2$ , and  $G_3$ , corresponding from left to right to the images in Fig. 11.30.



# 11.3.3 纹理

## 结构法：纹理基元和排列规则

结构法的基本思想：复杂的纹理可由一些简单的纹理基元（基本纹理元素）以一定的有规律的形式重复排列组合而成

- (1) 确定纹理基元
- (2) 建立排列规则

8个重写规则 (*a*: 模式, *b*: 向下, *c*: 向左)

- (1)  $S \rightarrow aA$  (变量*S*可用*aA*来替换)
- (2)  $S \rightarrow bA$  (变量*S*可用*bA*来替换)
- (3)  $S \rightarrow cA$  (变量*S*可用*cA*来替换)
- (4)  $A \rightarrow aS$  (变量*A*可用*aS*来替换)
- (5)  $A \rightarrow bS$  (变量*A*可用*bS*来替换)
- (6)  $A \rightarrow cS$  (变量*A*可用*cS*来替换)
- (7)  $A \rightarrow c$  (变量*A*可用常量*c*来替换)
- (8)  $S \rightarrow a$  (变量*S*可用常量*a*来替)

# 11.3.3 纹理

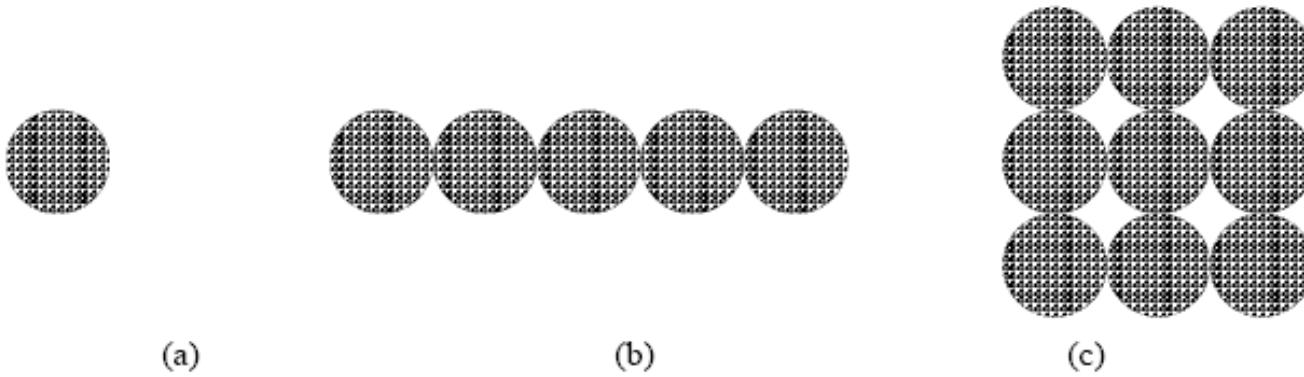


图 11.3.1 不同 2-D 模式的生成

**(b):** (1), (4), (1), (4), (8)

**(c):** (1), (4), (1), (5), (3), (6), (3), (4), (1), (4), (2), (6), (3), (6), (1), (4), (8)



# 11.3.3 纹理

## 傅里叶频谱

借助傅里叶频谱的频率特性来描述周期/近乎周期的2-D图象模式的方向性

- (1) 傅里叶频谱中突起的峰值对应纹理模式的主方向
- (2) 这些峰在频域平面的位置对应模式的基本周期
- (3) 利用滤波把周期性成分除去，用统计方法描述剩下的非周期性部分



# 11.3.3 纹理

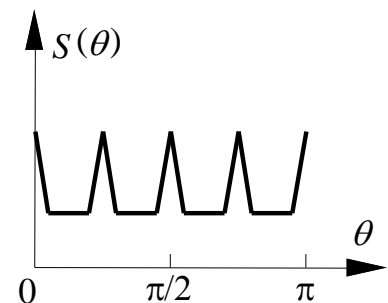
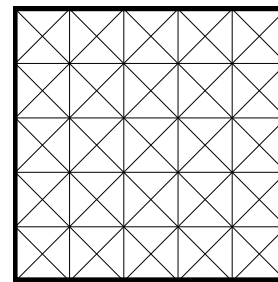
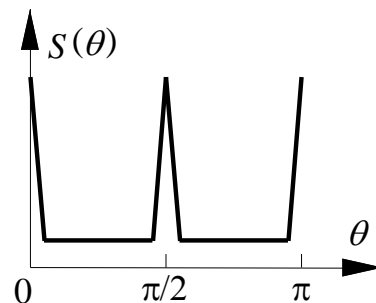
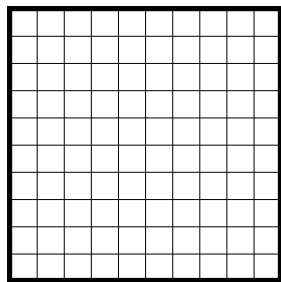
## 傅里叶频谱

极坐标系中频谱函数 $S(r, \theta)$

- 对每个确定的方向 $\theta$ ,  $S(r, \theta)$ 是1个1-D函数 $S_\theta(r)$
- 对每个确定的频率 $r$ ,  $S(r, \theta)$ 是1个1-D函数 $S_r(\theta)$

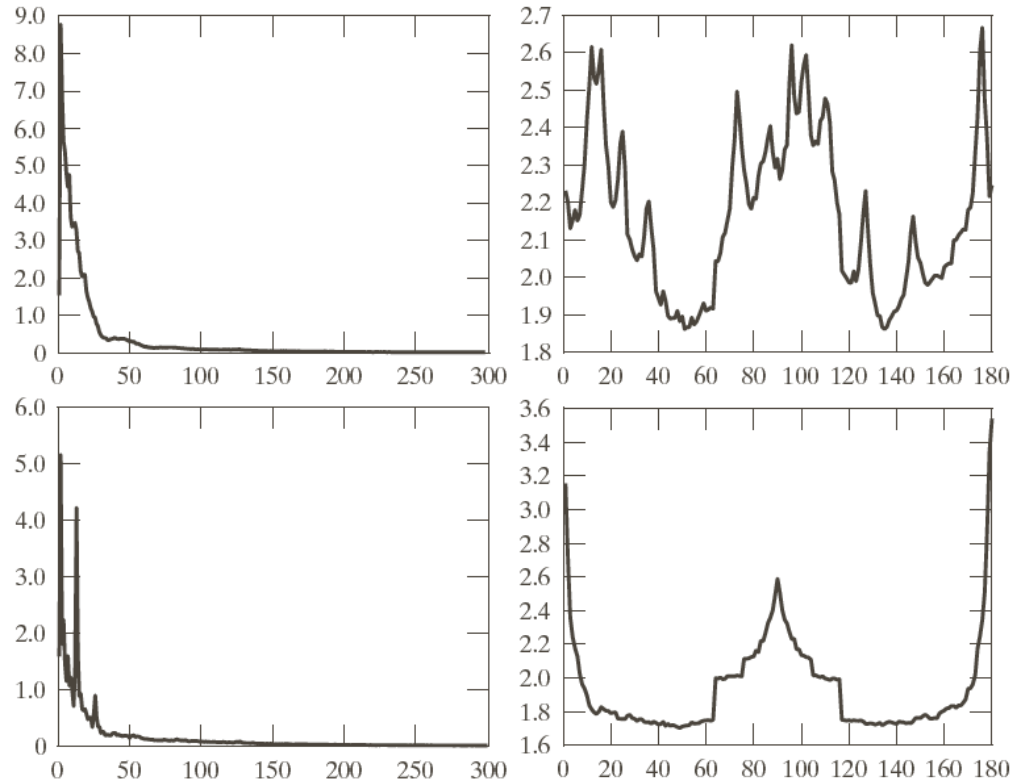
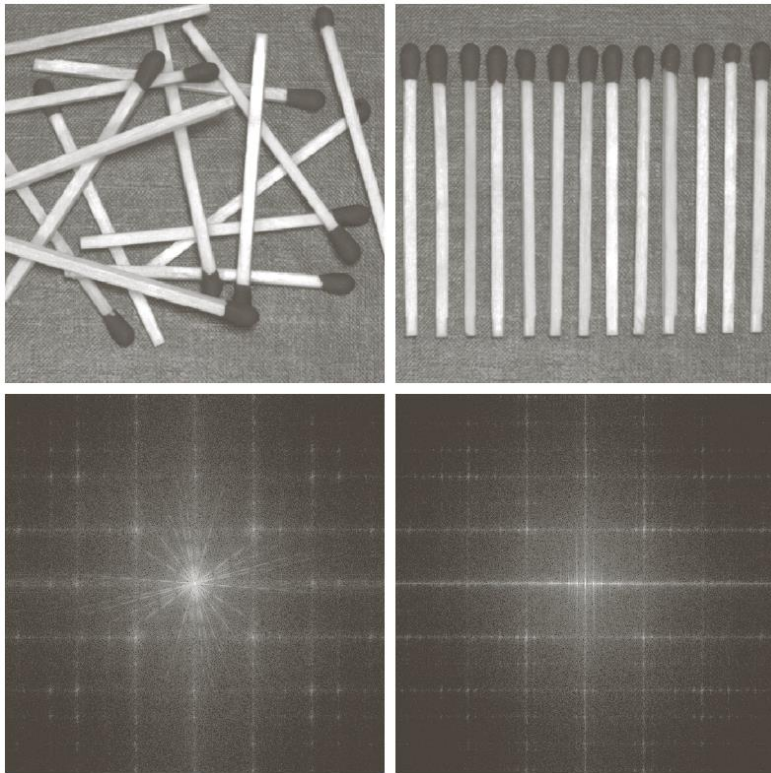
$$S(r) = \sum_{\theta=0}^{\pi} S_\theta(r)$$

$$S(\theta) = \sum_{r=1}^R S_r(\theta)$$



# 11.3.3 纹理

谱法分析纹理示例：





# 11.3.4 不变矩

**区域矩：**用所有属于区域内的点计算

$f(x, y)$ 的  $p + q$  矩

$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y)$$

$f(x, y)$ 的  $p + q$  中心矩

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y)$$

$f(x, y)$ 的归一化的中心矩

$$N_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma} \quad \text{其中 } \gamma = \frac{p+q}{2} + 1, \quad p+q = 2, 3, \dots$$



# 11.3.4 不变矩

平移、旋转、尺度不变矩

$$T_1 = N_{20} + N_{02} \quad T_2 = (N_{20} - N_{02})^2 + 4N_{11}^2$$

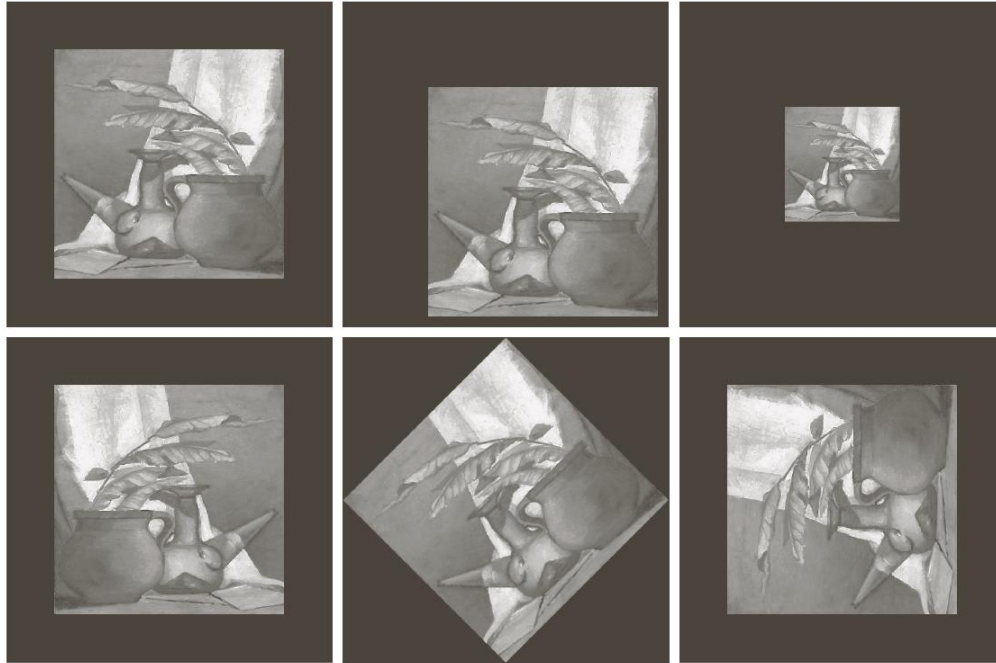
$$T_3 = (N_{30} - 3N_{12})^2 + (3N_{21} - N_{03})^2 \quad T_4 = (N_{30} + N_{12})^2 + (N_{21} + N_{03})^2$$

$$T_5 = (N_{30} - 3N_{12})(N_{30} + N_{12}) \left[ (N_{30} + N_{12})^2 - 3(N_{21} + N_{03})^2 \right] + \\ (3N_{21} - N_{03})(N_{21} + N_{03}) \left[ 3(N_{30} + N_{12})^2 - (N_{21} + N_{03})^2 \right]$$

$$T_6 = (N_{20} - N_{02}) \left[ (N_{30} + N_{12})^2 - (N_{21} + N_{03})^2 \right] + \\ 4N_{11}(N_{30} + N_{12})(N_{21} + N_{03})$$

$$T_7 = (3N_{21} - N_{03})(N_{30} + N_{12}) \left[ (N_{30} + N_{12})^2 - 3(N_{21} + N_{03})^2 \right] + \\ (3N_{12} - N_{30})(N_{21} + N_{03}) \left[ 3(N_{30} + N_{12})^2 - (N_{21} + N_{03})^2 \right]$$

# 11.3.4 不变矩



Moment Invariant	Original Image	Translated	Half Size	Mirrored	Rotated 45°	Rotated 90°
$\phi_1$	2.8662	2.8662	2.8664	2.8662	2.8661	2.8662
$\phi_2$	7.1265	7.1265	7.1257	7.1265	7.1266	7.1265
$\phi_3$	10.4109	10.4109	10.4047	10.4109	10.4115	10.4109
$\phi_4$	10.3742	10.3742	10.3719	10.3742	10.3742	10.3742
$\phi_5$	21.3674	21.3674	21.3924	21.3674	21.3663	21.3674
$\phi_6$	13.9417	13.9417	13.9383	13.9417	13.9417	13.9417
$\phi_7$	-20.7809	-20.7809	-20.7724	20.7809	-20.7813	-20.7809



# 第11章 表示和描述

---

11.1 表示

11.2 边界描绘子

11.3 区域描绘子

11.4 使用主成分进行描绘

11.5 关系描绘子



# 11.4 主成分描绘

- 向  $x$  的协方差矩： $C_x = E\{(x - m_x)(x - m_x)^T\}$
- 由于  $C_x$  是对称的，所以总可以找到一组  $n$  个**正交**的特征向。
  - 令  $A$  为一个矩，其行由  $C_x$  的特征向组成，特征向的排序方式为  $A$  的第一行对应于最大特征值的特征向，最后一行对应最小特征值的特征向。
- 假设将  $A$  用做将向  $x$  映射向  $y$  的一个变换矩（特林变换）：
$$y = A(x - m_x)$$
- 可以证明
  - $y$  的均值为向： $m_y = E\{y\} = 0$
  - $y$  的协方差矩  $C_y$  为对角矩：

$$C_y = AC_x A^T = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix}$$

其中  $\lambda_i$  为  $C_x$  的第  $i$  个特征向对应的特征值， $\lambda_i$  等于向  $y$  的第  $i$  维的方差， $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$



# 11.4 主成分描绘

- 任何向  $x$  能通过下式由其对应的  $y$  来恢复 ( $A^{-1} = A^T$ ):

$$x = A^T y + m_x$$

- 假设由最大的  $k$  个特征值对应的特征向 来形成矩  $A_k$ , 其矩大小为  $k \times n$ , 对应得到的向  $y$  为  $k$  维, 从而实现数据 维
- 使用  $A_k$  建向  $x$ , 得到:

$$\hat{x} = A_k^T y + m_x$$

- 可以证明,  $x$  和  $\hat{x}$  之 的均方误差由下式给出 ( $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ ):

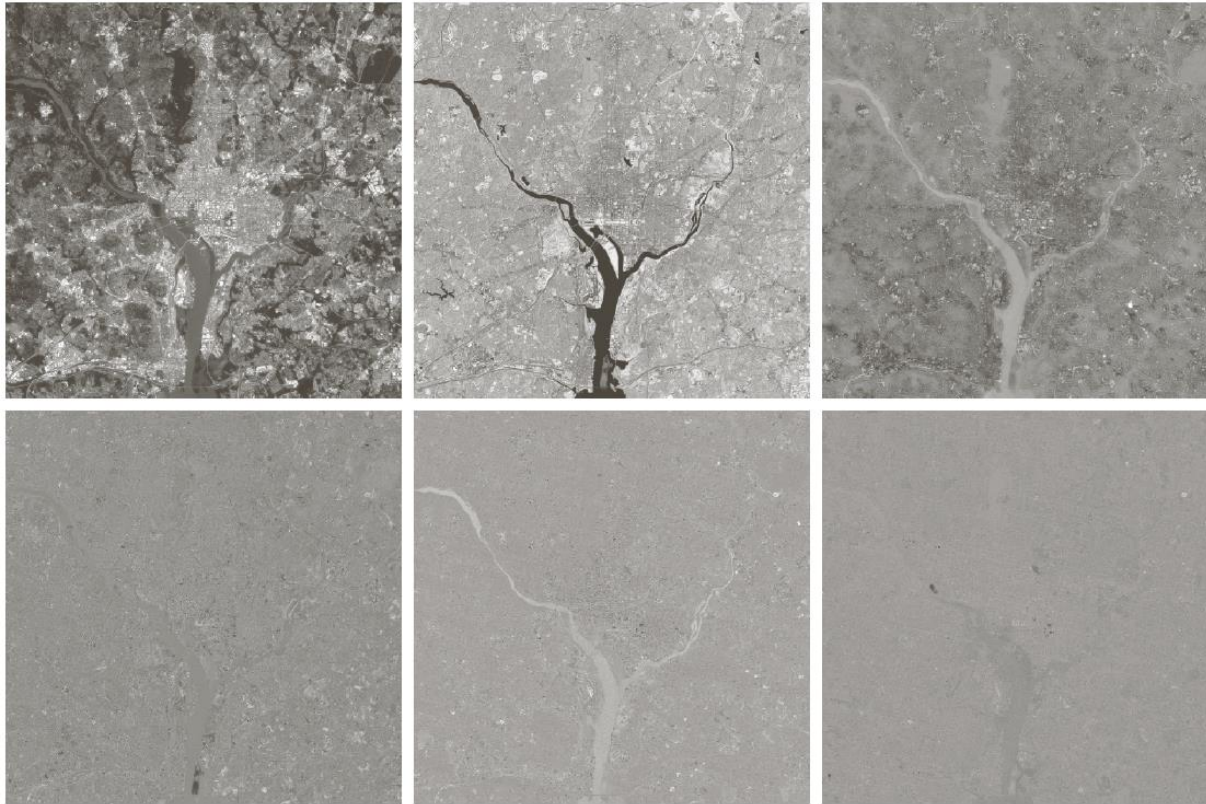
$$e_{ms} = \sum_{i=1}^n \lambda_i - \sum_{i=1}^k \lambda_i = \sum_{i=k+1}^n \lambda_i$$

- 将向  $x$  和其近似值  $\hat{x}$  之 的均方误差 至最小, 从这个意义上, 上述变换 ( 特林变换, 又称主成分变换) 是最优的



# 11.4 主成分描绘

- 主成分图像。前两个特征值远大于其他特征值，因此对比度细节的主要成分包含在前两幅图像中。

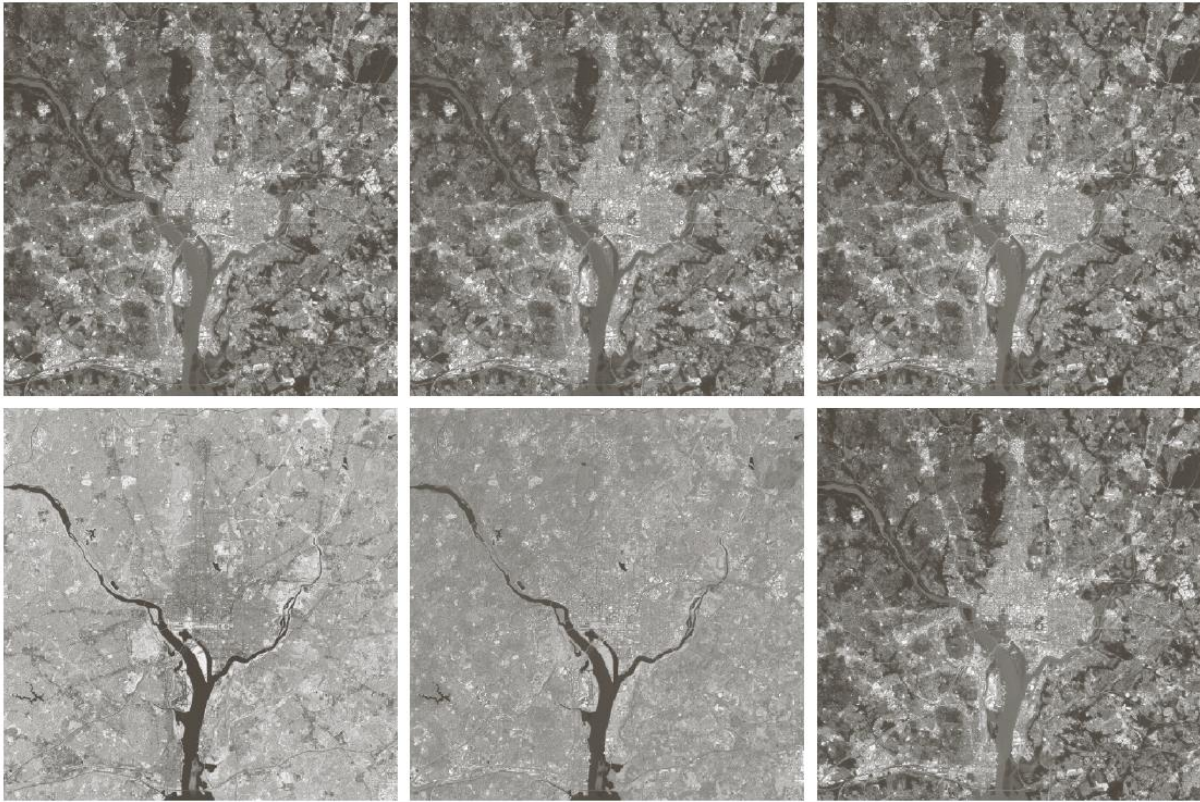


a	b	c
d	e	f

**FIGURE 11.40** The six principal component images obtained from vectors computed using Eq. (11.4-6). Vectors are converted to images by applying Fig. 11.39 in reverse.

# 11.4 主成分描绘

- 仅使用两幅主成分图像 建的多光谱图像：



a b c  
d e f

**FIGURE 11.41** Multispectral images reconstructed using only the two principal component images corresponding to the two principal component images with the largest eigenvalues (variance). Compare these images with the originals in Fig. 11.38.

# 11.4 主成分描绘

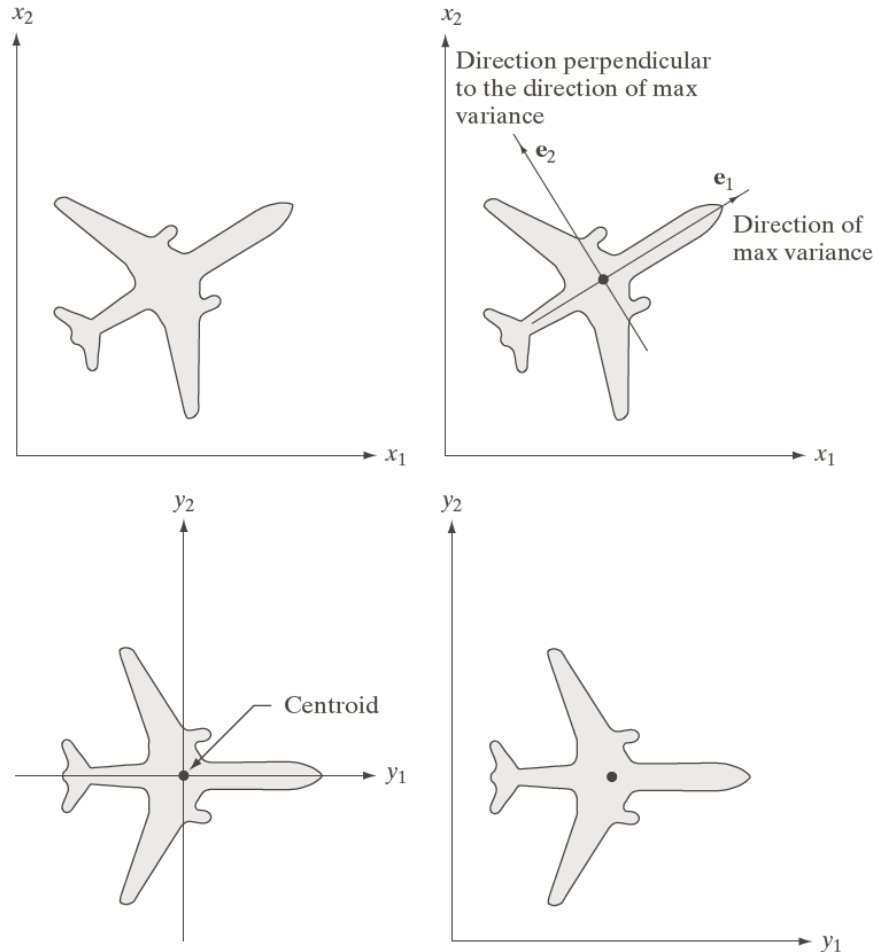
例子:

$$\mathbf{m}_x = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$

$$\mathbf{C}_x = \begin{bmatrix} 3.333 & 2.00 \\ 2.00 & 3.333 \end{bmatrix}$$

$$\mathbf{e}_1 = \begin{bmatrix} 0.707 \\ 0.707 \end{bmatrix}$$

$$\mathbf{e}_2 = \begin{bmatrix} -0.707 \\ 0.707 \end{bmatrix}$$



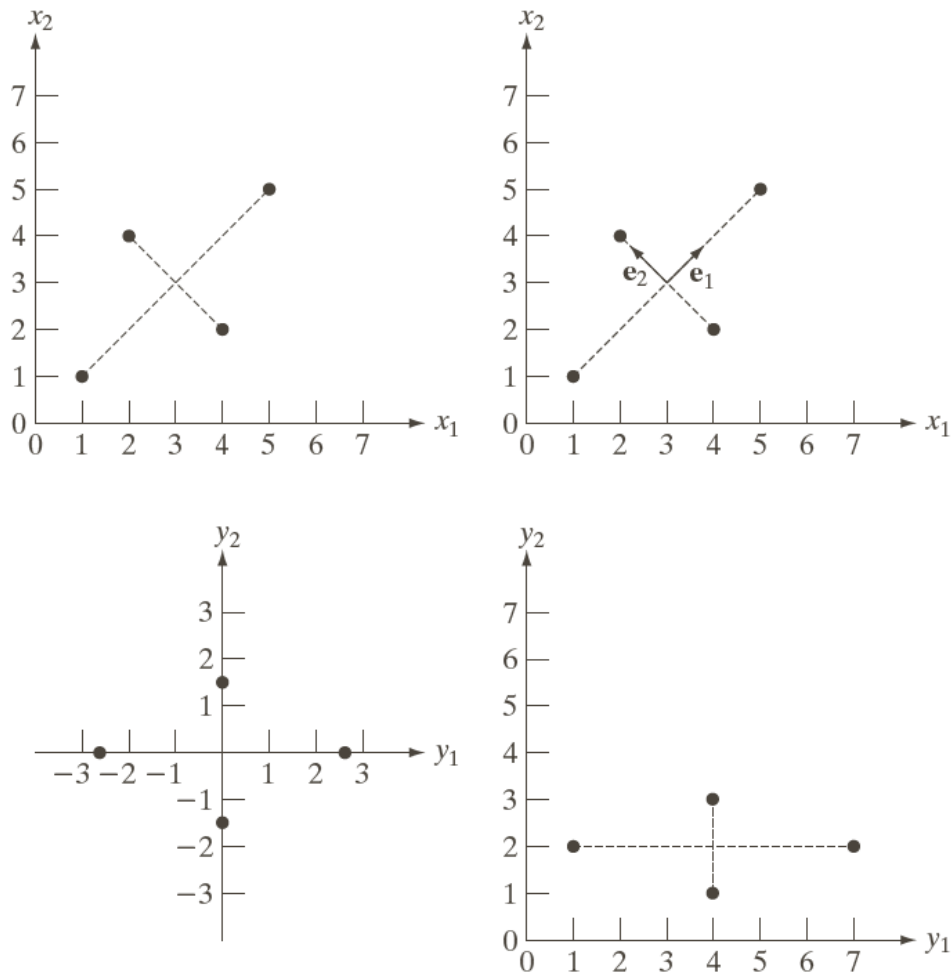
a	b
c	d

**FIGURE 11.43**

(a) An object.  
 (b) Object showing eigenvectors of its covariance matrix.  
 (c) Transformed object, obtained using Eq. (11.4-6).  
 (d) Object translated so that all its coordinate values are greater than 0.

# 11.4 主成分描绘

例子:



a b  
c d

**FIGURE 11.44**

A manual example.  
 (a) Original points.  
 (b) Eigenvectors of the covariance matrix of the points in (a).  
 (c) Transformed points obtained using Eq. (11.4-6).  
 (d) Points from (c), rounded and translated so that all coordinate values are integers greater than 0. The dashed lines are included to facilitate viewing. They are not part of the data.



# 第11章 表示和描述

---

11.1 表示

11.2 边界描绘子

11.3 区域描绘子

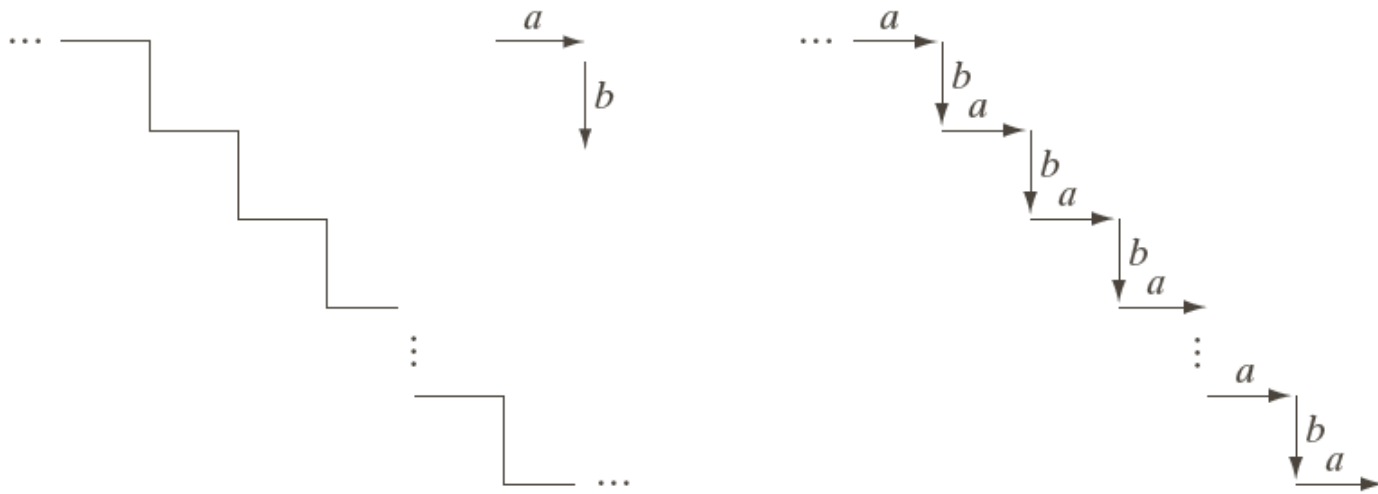
11.4 使用主成分进行描绘

11.5 关系描绘子

# 11.5 关系描绘子

关系描绘子主要的目的是以 **写规则** 的形式来获取边界或区域中的基本复模式。例如，对下图的 **梯结构** 用基本的递归单元进行 **写**：

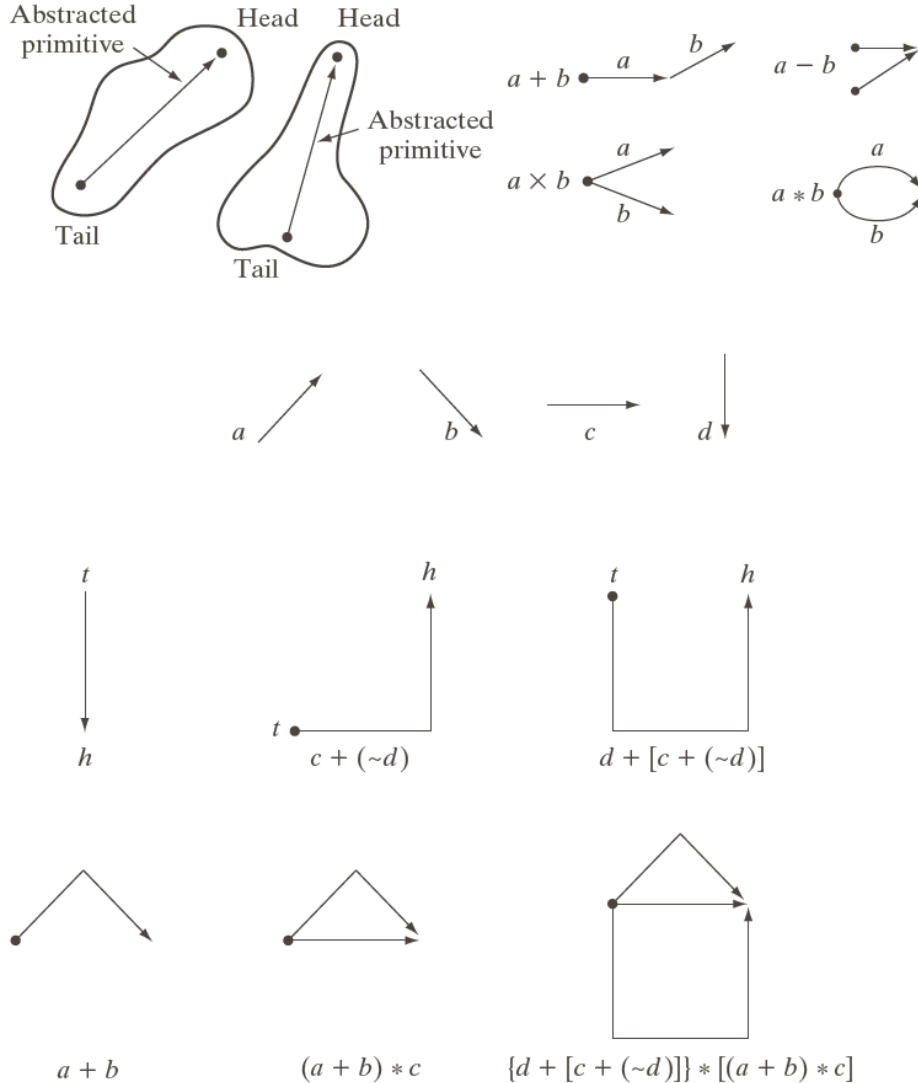
- (1)  $S \rightarrow aA,$
- (2)  $A \rightarrow bS,$  and
- (3)  $A \rightarrow b,$



a b  
**FIGURE 11.45**  
 (a) A simple staircase structure.  
 (b) Coded structure.



# 11.5 关系描绘子



使用有向线段来描绘图像的各个部分，除了用尾相连的方式外，有向线段也可以用其他方式来连接，如左图：

a b  
c  
d

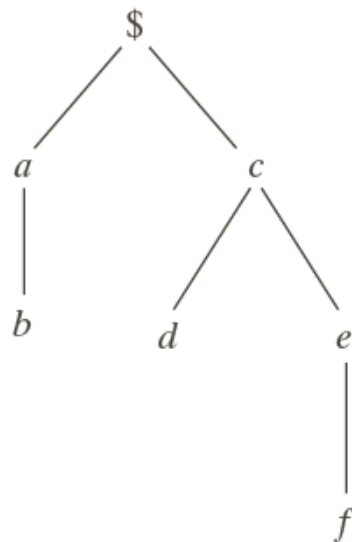
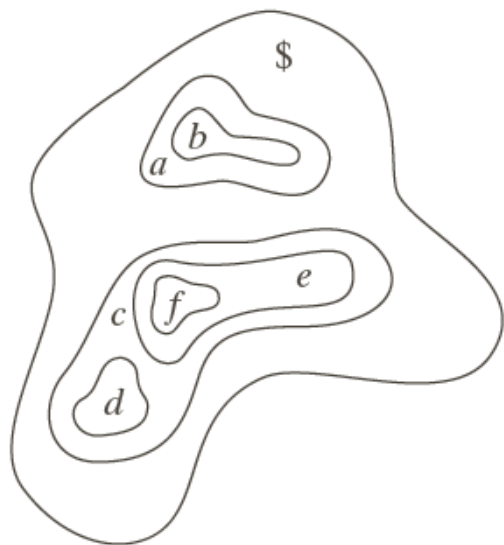
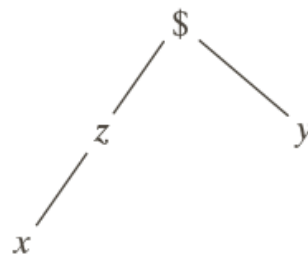
**FIGURE 11.48**  
 (a) Abstracted primitives.  
 (b) Operations among primitives.  
 (c) A set of specific primitives.  
 (d) Steps in building a structure.

# 11.5 关系描绘子

## 树描绘子

一棵树  $T$  是有着一个或多个节点的有 合：

1. 仅有一个表示根节点  $\$$
2. 余下的节点被分成不相交的 合  $T_1, \dots, T_m$ ，每个 合一 是一棵树，称为  $T$  的子树。



a b

**FIGURE 11.50**

(a) A simple composite region.  
 (b) Tree representation obtained by using the relationship "inside of."





# 第12章 目标识别

- 12.1 模式和模式类
- 12.2 基于决策理论方法的识别
  - 匹配
  - 最佳统计分类器
  - 神经网络
- 12.3 结构方法
  - 匹配形状数
  - 串匹配



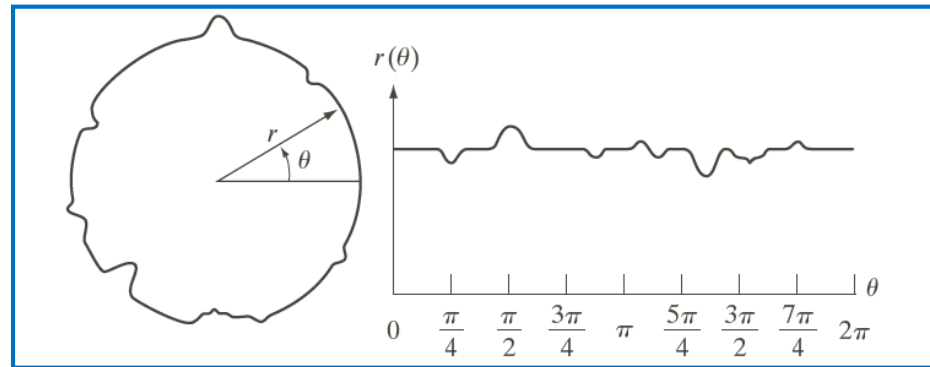
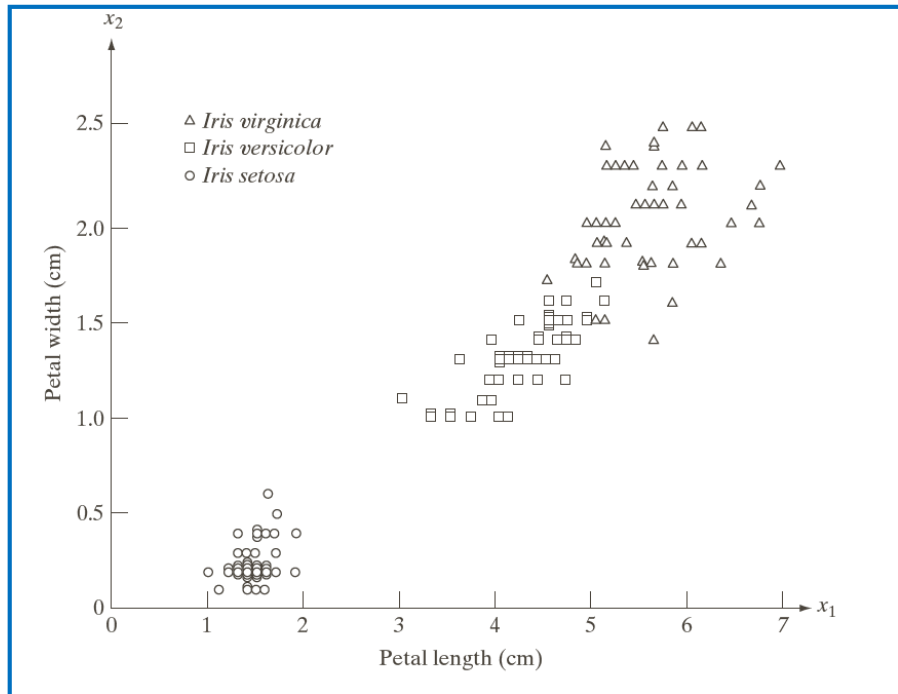
# 第12章 目标识别

- 12.1 模式和模式类
- 12.2 基于决策理论方法的识别
  - 匹配
  - 最佳统计分类器
  - 神经网络
- 12.3 结构方法
  - 匹配形状数
  - 串匹配

# 12.1 模式和模式类

## □ 模式向量

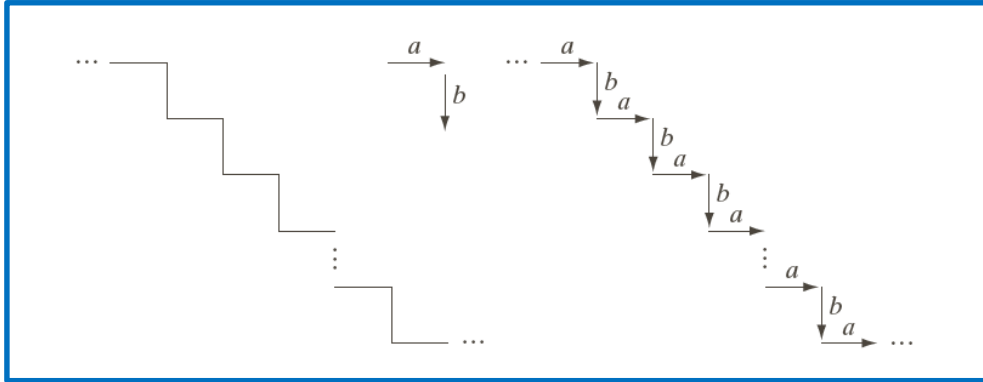
$$\mathbf{x} = (x_1, x_2, \dots, x_n)^T$$



一个带有噪声的目标及其相应的信号

由两个测度描述的三种鸢尾花

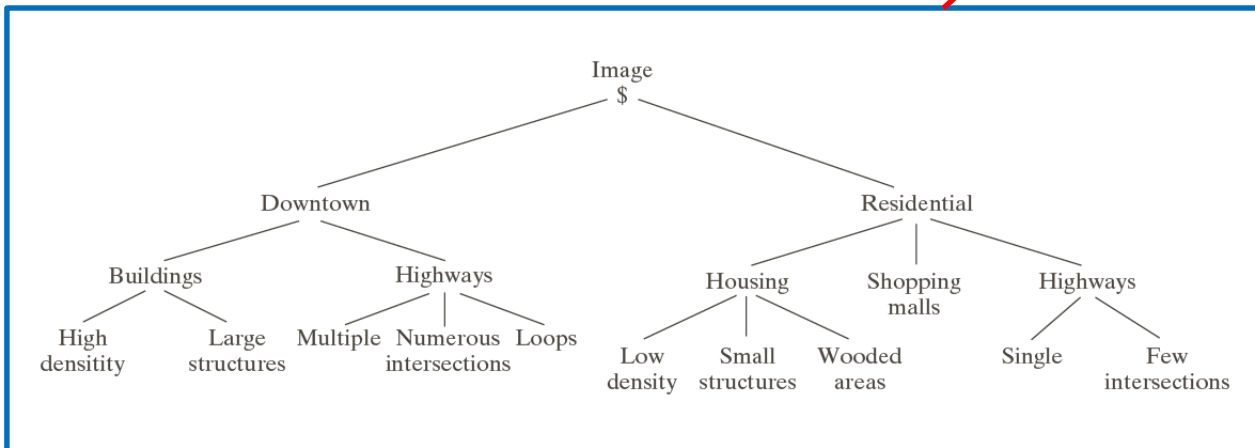
# 12.1 模式和模式类



(a) 阶梯结构 (b) 使用基元 $a$ 和 $b$ 对结构编码, 生成串描述...  $ababab$  ...



建筑物密集的城市中心区（华盛顿特区）和周围居民区的卫星图像（原图像由NASA提供）





# 第12章 目标识别

- 12.1 模式和模式类
- 12.2 基于决策理论方法的识别
  - 匹配
  - 最佳统计分类器
  - 神经网络
- 12.3 结构方法
  - 匹配形状数
  - 串匹配



# 12.2 基于决策理论方法的识别

- $W$ 个模式类 $\omega_1, \omega_2, \dots, \omega_W$ , 决策函数 $d_1(x), d_2(x), \dots, d_W(x)$

如果模式 $x$ 属于类 $\omega_i$ , 则

$$d_i(x) > d_j(x), \quad \forall j = 1, 2, \dots, W; j \neq i$$

- 决策边界

$$d_i(x) = d_j(x)$$



# 12.2.1 匹配

## □ 最小距离分类器

模式类的原型

$$\mathbf{m}_j = \frac{1}{N_j} \sum_{\mathbf{x} \in \omega_j} \mathbf{x}_j \quad j = 1, 2, \dots, W$$

距离测度

$$D_j(\mathbf{x}) = \|\mathbf{x} - \mathbf{m}_j\| \quad j = 1, 2, \dots, W$$

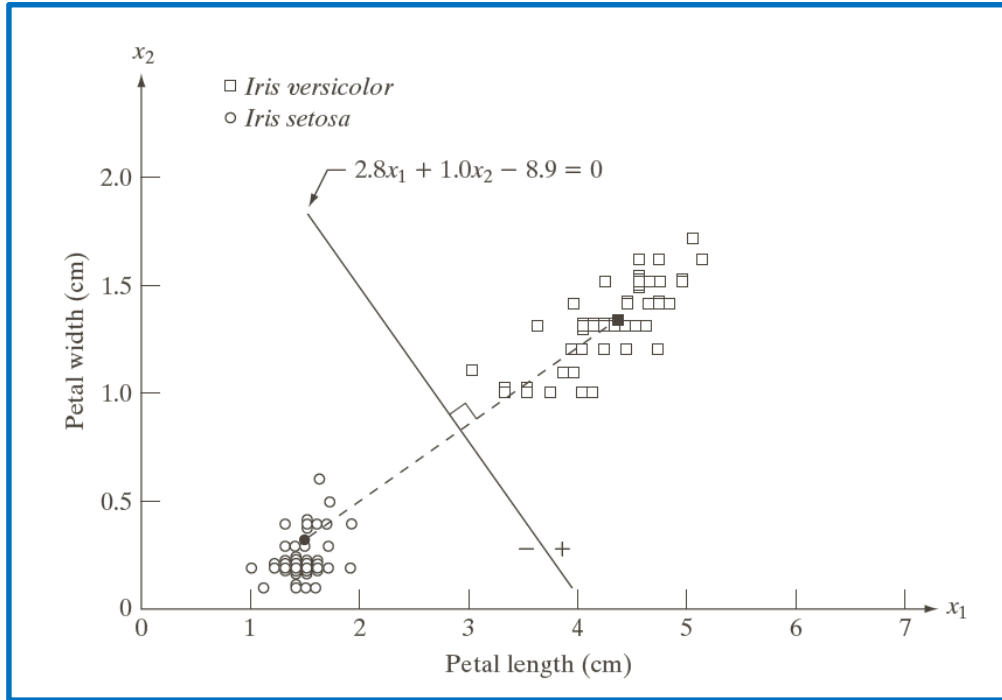
选择最小距离等价于

$$d_j(\mathbf{x}) = \mathbf{x}^T \mathbf{m}_j - \frac{1}{2} \mathbf{m}_j^T \mathbf{m}_j \quad j = 1, 2, \dots, W$$

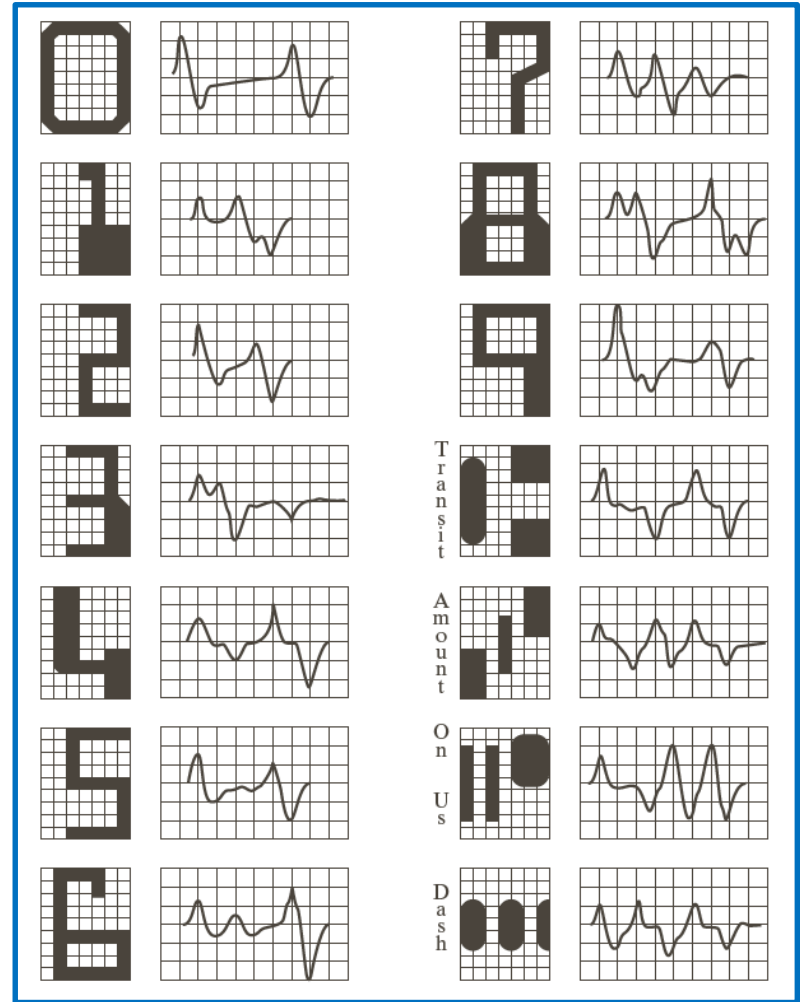
决策边界

$$\begin{aligned} d_{ij}(\mathbf{x}) &= d_i(\mathbf{x}) - d_j(\mathbf{x}) \\ &= \mathbf{x}^T (\mathbf{m}_i - \mathbf{m}_j) - \frac{1}{2} (\mathbf{m}_i - \mathbf{m}_j)^T (\mathbf{m}_i + \mathbf{m}_j) = 0 \end{aligned}$$

# 12.2.1 匹配



类 *Iris versicolor* 和类 *Iris setosa* 的最小距离分类器的决策边界



美国银行家协会E-138字符集和对应波形





# 12.2.1 匹配

## □ 相关匹配

相关表示

$$c(x, y) = \sum_s \sum_t w(s, t) f(x + s, y + t)$$

模板      图像

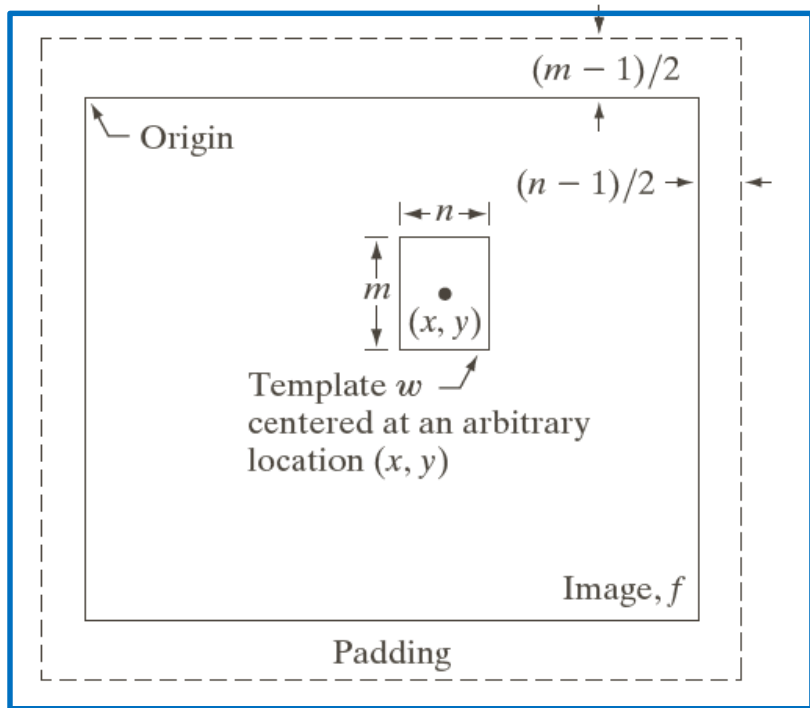
空间相关

$$f(x, y) \star w(x, y) \Leftrightarrow F^*(u, v)W(u, v)$$

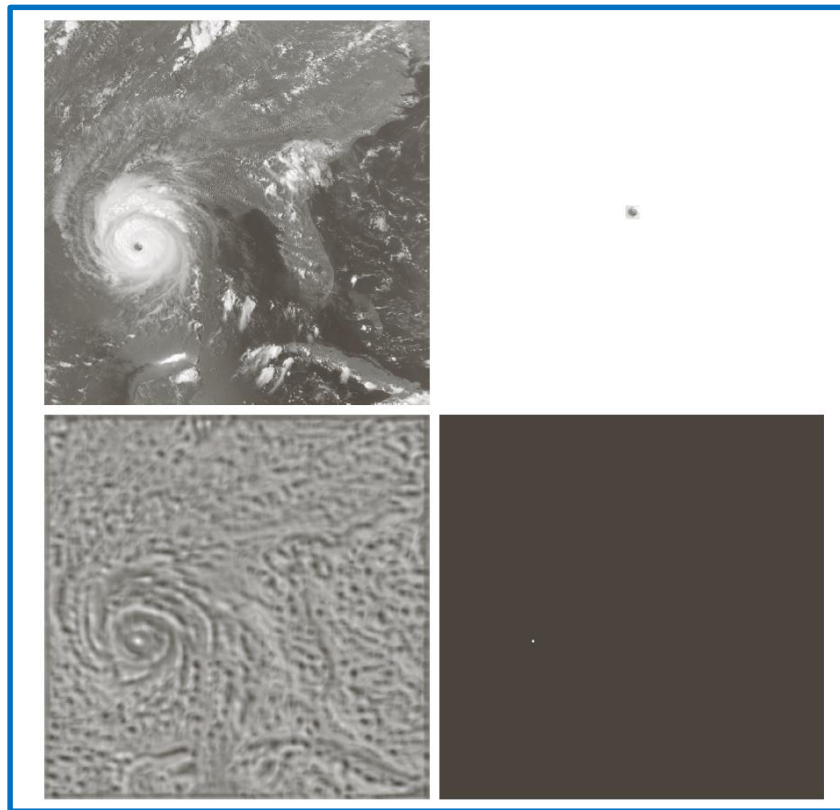
$$\gamma(x, y) = \frac{\sum_s \sum_t [w(s, t) - \bar{w}][f(x + s, y + t) - \overline{f_{xy}}]}{\left\{ \sum_s \sum_t [w(s, t) - \bar{w}]^2 \sum_s \sum_t [f(x + s, y + t) - \overline{f_{xy}}]^2 \right\}^{\frac{1}{2}}}$$

归一化相关系数

# 12.2.1 匹配



模板匹配的机理



(a) 飓风“安德鲁”的卫星图像，摄于1992年8月24日；(b) 飓风眼的模板；(c) 显示为图像的相关系数（注意最亮点）；(d) 匹配最好的位置。该点是单个像素，但其已被放大，以便于查看（原图像由NOAA提供）



# 12.2.2 最佳统计分类器

## □ 基础知识

- 模式 $\mathbf{x}$  被分为第 $k$  类的概率  $p(\omega_k|\mathbf{x})$ ，样本属于第 $k$  类但被判为第 $j$  类的风险  $L_{kj}$ ，则 $\mathbf{x}$  被分为第 $j$  类的平均损失：

$$r_j(\mathbf{x}) = \sum_{k=1}^W L_{kj} p(\omega_k|\mathbf{x}) = \frac{1}{p(\mathbf{x})} \sum_{k=1}^W L_{kj} p(\mathbf{x}|\omega_k) P(\omega_k)$$

去掉 $\frac{1}{p(\mathbf{x})}$ 不影响 $r_j(\mathbf{x})$ 的对不同 $j$  的相对大小排序

- 定义风险函数

$$L_{ij} = 1 - \delta_{ij}$$

- 更新后的平均损失函数为：

$$r_j(\mathbf{x}) = \sum_{k=1}^W (1 - \delta_{ij}) p(\mathbf{x}|\omega_k) P(\omega_k) = P(\mathbf{x}) - p(\mathbf{x}|\omega_j) P(\omega_j)$$

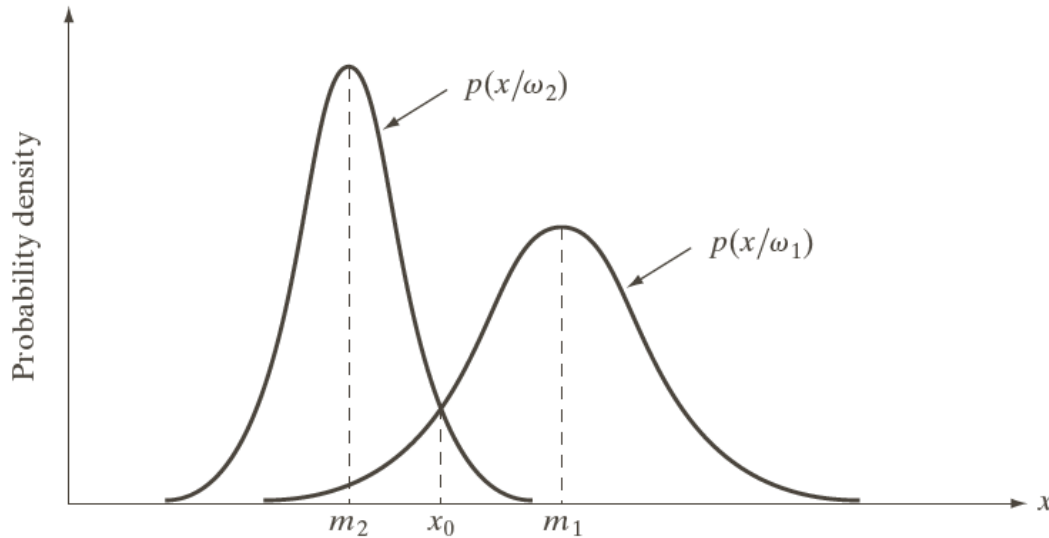


# 12.2.2 最佳统计分类器

## □ 高斯模式类的贝叶斯分类器

贝叶斯决策函数

$$d_j(x) = p(x|\omega_j)P(\omega_j)$$
$$= \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(x-m_j)^2}{2\sigma_j^2}} P(\omega_j) \quad j = 1,2$$



两个一维模式类的概率密度函数。如果两个类出现概率相等，那么所示的点 $x_0$ 就是决策边界



# 12.2.2 最佳统计分类器

## □ 高斯模式类的贝叶斯分类器

$n$ 维情形下,  $j$ 个  
模式类中的向量  
高斯密度

$$p(\mathbf{x}|\omega_j) = \frac{1}{(2\pi)^{n/2} |\mathbf{C}_j|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{m}_j)^T (\mathbf{x}-\mathbf{m}_j)}$$

$$\mathbf{m}_j = \frac{1}{N_j} \sum_{\mathbf{x} \in \omega_j} \mathbf{x}$$

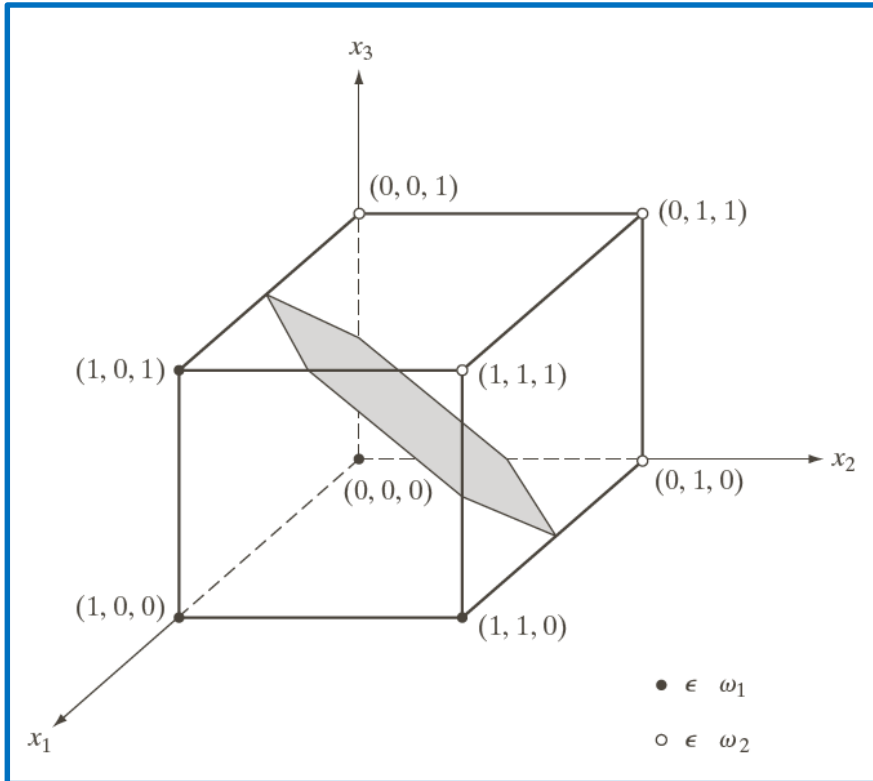
$$\mathbf{C}_j = \frac{1}{N_j} \sum_{\mathbf{x} \in \omega_j} \mathbf{x}\mathbf{x}^T - \mathbf{m}_j\mathbf{m}_j^T$$

线性决策函数  
(超平面)

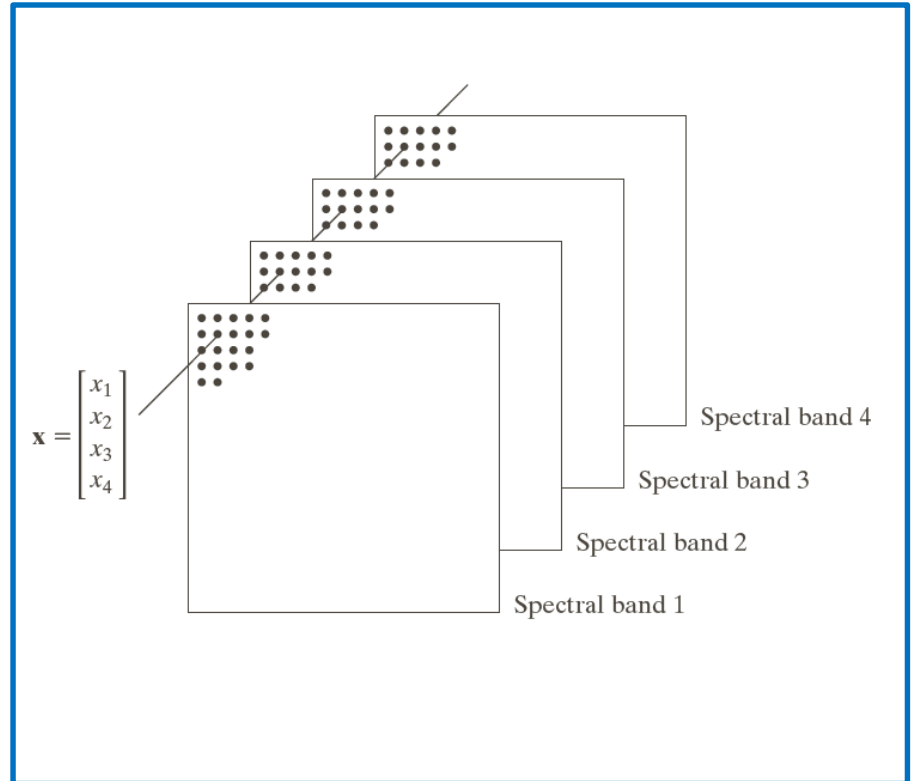
$$d_j(\mathbf{x}) = \ln P(\omega_j) + \mathbf{x}^T \mathbf{C}^{-1} \mathbf{m}_j - \frac{1}{2} \mathbf{m}_j^T \mathbf{C}^{-1} \mathbf{m}_j$$

# 12.2.2 最佳统计分类器

## □ 高斯模式类的贝叶斯分类器



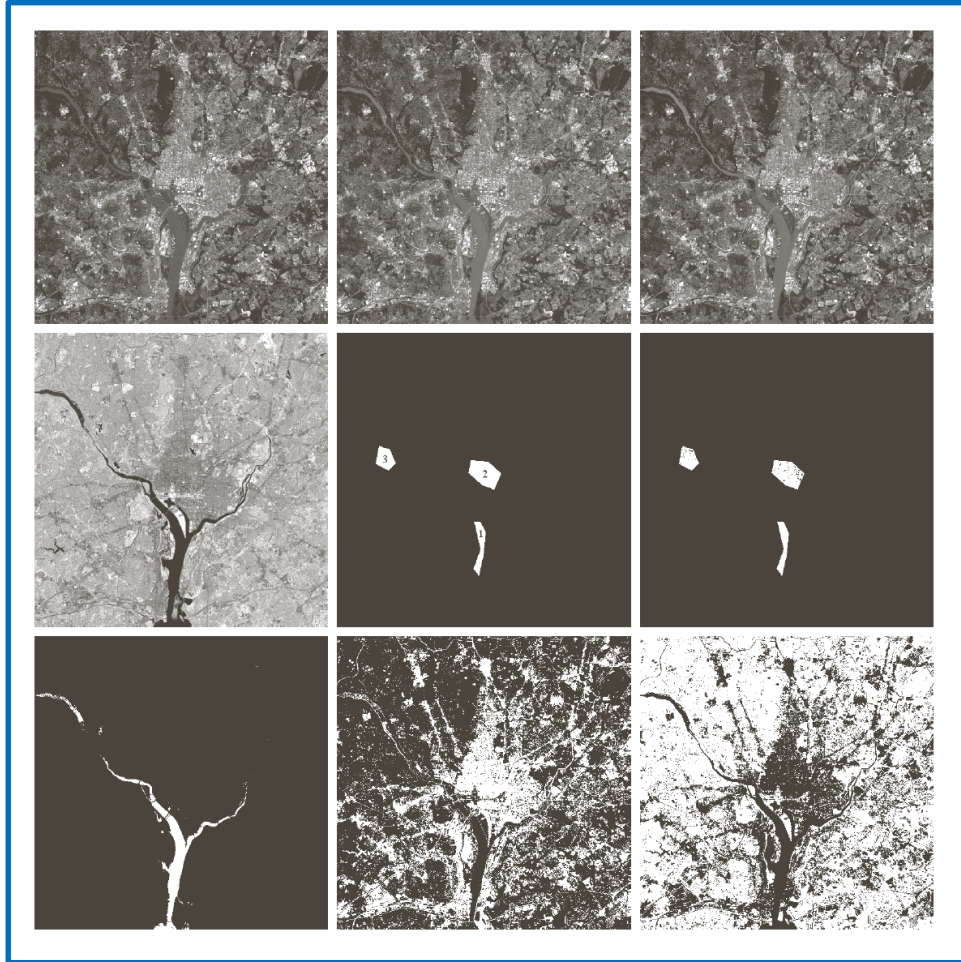
两个简单的模式类  
及其贝叶斯决策边  
界（阴影所示）



由多光谱扫描器生成的4  
幅数字图像经像素配准后，  
所形成的的模式向量

# 12.2.2 最佳统计分类器

## □ 高斯模式类的贝叶斯分类器

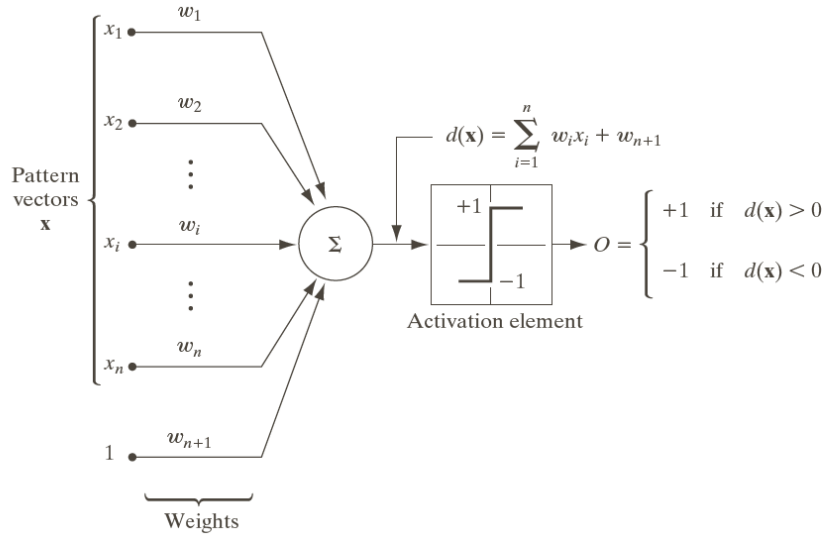


a b c  
d e f  
g h i

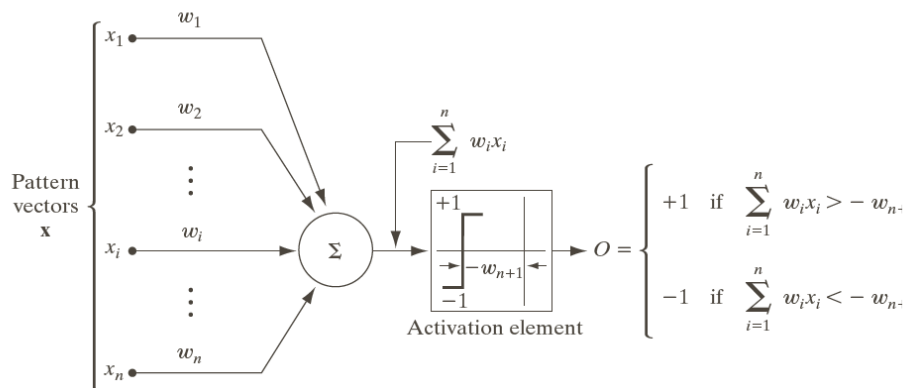
多光谱数据的贝叶斯分类：  
(a)~(d) 可见蓝光、可见绿光、可见红光和古今红外波长图像；  
(e) 显示(1)水体、(2)市区和(3)植被的样本区域的模板；  
(f) 分类结果。黑点表示为正确分类的点，其他（白）点是正确分类的点；  
(g) 分类为水体的所有图像像素（白色）；  
(h) 分类为市区的所有图像像素（白色）；  
(i) 分类为植被的所有图像像素（白色）

# 12.2.3 神经网络

## □ 两个模式类的感知机



$$d(x) = \sum_{i=1}^n w_i x_i + w_{n+1}$$



$$O = \begin{cases} +1, & \sum_{i=1}^n w_i x_i > -w_{n+1} \\ -1, & \sum_{i=1}^n w_i x_i < -w_{n+1} \end{cases}$$





# 12.2.3 神经网络

## □ 向量化表达

- 记  $\mathbf{y} = \{x_1, x_2, \dots, x_n, 1\}^T$ ,  $\mathbf{w} = \{w_1, w_2, \dots, w_n, w_{n+1}\}^T$ , 则判决函数:

$$d(\mathbf{y}) = \sum_{i=1}^{n+1} w_i y_i = \mathbf{w}^T \mathbf{y}$$

## □ 训练算法

- 线性可分的类:

如果  $\mathbf{y}(k) \in \omega_1$  且  $\mathbf{w}^T \mathbf{y}(k) \leq 0$ , 则

$$\mathbf{w}(k+1) = \mathbf{w}(k) + c\mathbf{y}(k)$$

相反, 如果  $\mathbf{y}(k) \in \omega_2$  且  $\mathbf{w}^T \mathbf{y}(k) \geq 0$ , 则

$$\mathbf{w}(k+1) = \mathbf{w}(k) - c\mathbf{y}(k)$$

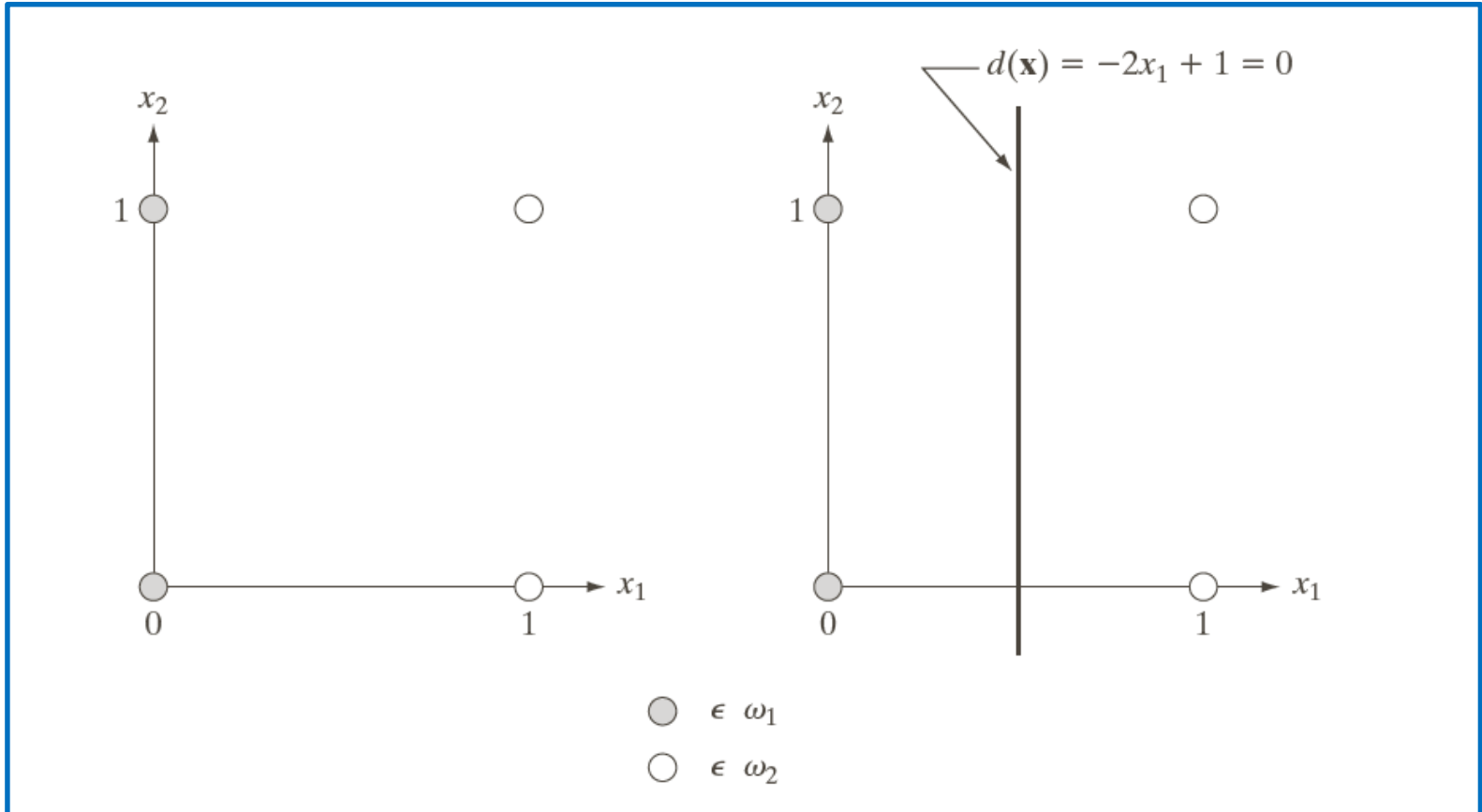
否则,  $\mathbf{w}(k)$  保持不变

- 线性不可分的类:

准则函数  $J(\mathbf{w}) = \frac{1}{2} (r - \mathbf{w}^T \mathbf{y})^2$

梯度下降算法  $\mathbf{w}(k+1) = \mathbf{w}(k) - \alpha \left[ \frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} \right]_{\mathbf{w}=\mathbf{w}(k)}$

# 12.2.3 神经网络



- (a) 属于两个类的模式;
- (b) 由训练确定的决策边界

# 12.2.3 神经网络

## □ 多层前馈神经网络

多层前馈神经网络模型。放大部分显示了整个网络中每个神经元的基本结构。

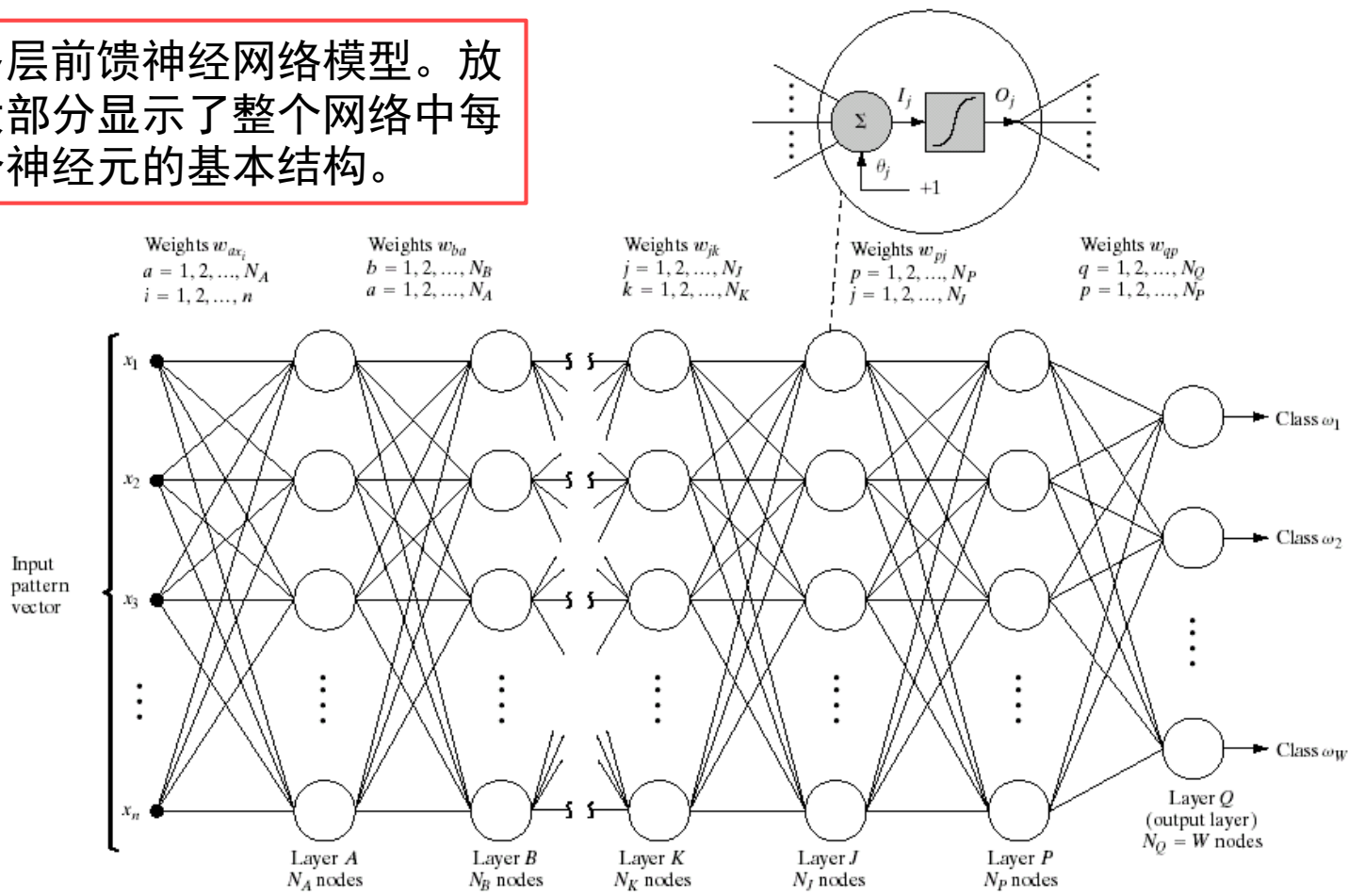


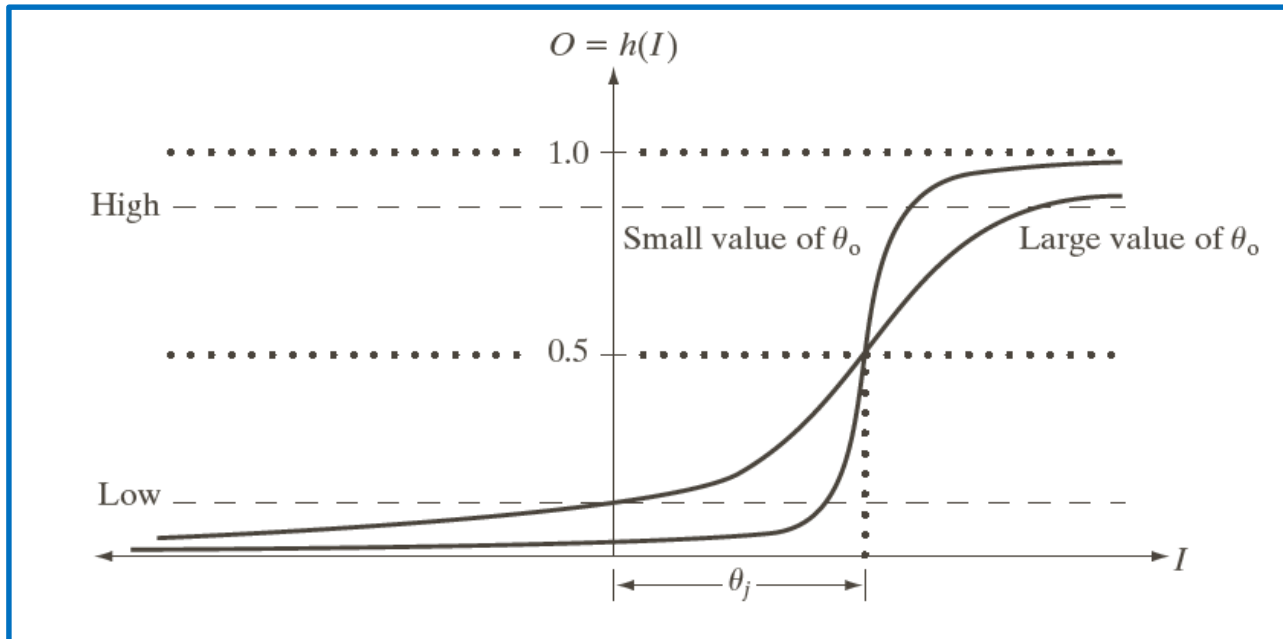
FIGURE 12.16 Multilayer feedforward neural network model. The blowup shows the basic structure of each neuron element throughout the network. The offset,  $\theta_j$ , is treated as just another weight.

# 12.2.3 神经网络

## □ 多层前馈神经网络

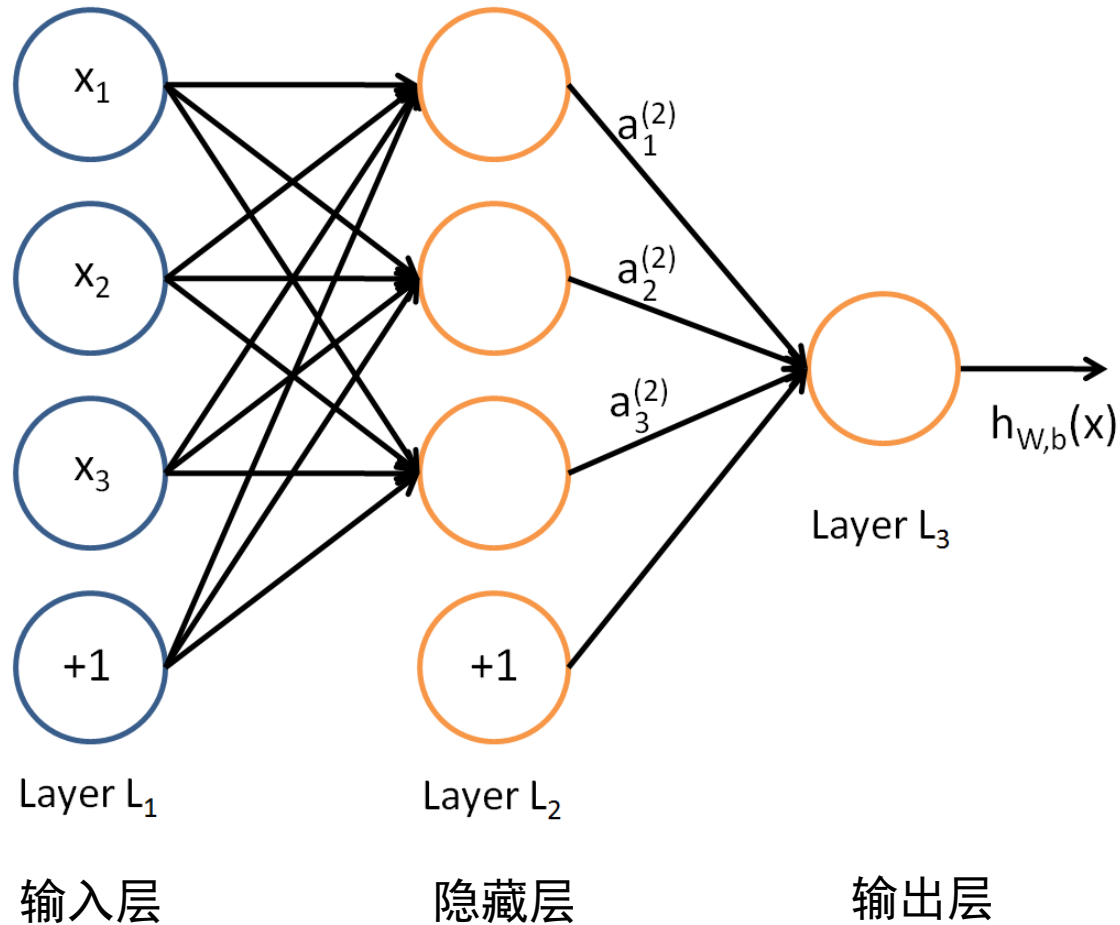
激活函数

$$h_j(I_j) = \frac{1}{1 + e^{-(I_j + \theta_j)/\theta_o}}$$



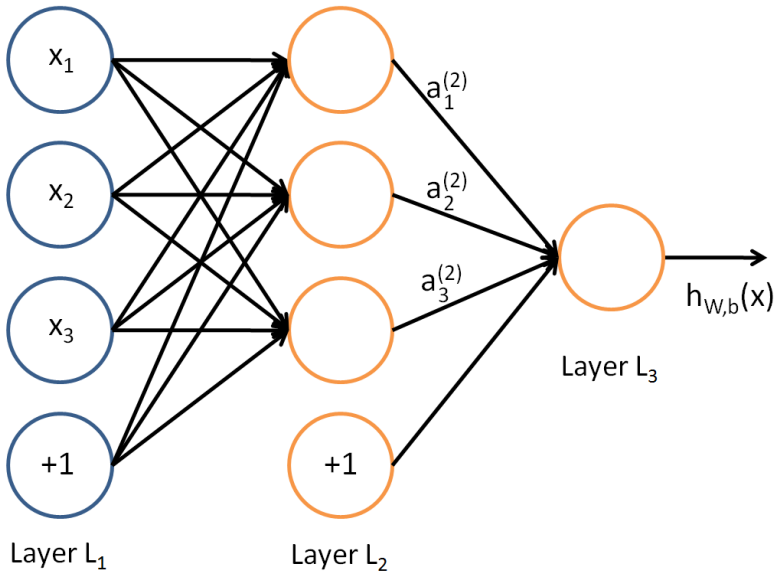
# Forward Network (前馈网络)

## □ 神经网络模型



# Forward Network (前馈网络)

## 神经网络模型



$a_i^{(l)}$ : 第 $l$ 层第 $i$ 单元的输出值

$z_i^{(l)}$ : 第 $l$ 层第 $i$ 单元的输入加权和

$$\begin{aligned} a_1^{(2)} &= f(z_1^{(2)}) \\ &= f(W_{11}^{(1)}x_1 + W_{12}^{(1)}x_2 + W_{13}^{(1)}x_3 + b_1^{(1)}) \end{aligned}$$

$$\begin{aligned} a_2^{(2)} &= f(z_2^{(2)}) \\ &= f(W_{21}^{(1)}x_1 + W_{22}^{(1)}x_2 + W_{23}^{(1)}x_3 + b_2^{(1)}) \end{aligned}$$

$$\begin{aligned} a_3^{(2)} &= f(z_3^{(2)}) \\ &= f(W_{31}^{(1)}x_1 + W_{32}^{(1)}x_2 + W_{33}^{(1)}x_3 + b_3^{(1)}) \end{aligned}$$

$$\begin{aligned} h_{W,b}(\mathbf{x}) &= a_1^{(3)} = f(z_1^{(3)}) \\ &= f(W_{11}^{(2)}a_1^{(2)} + W_{12}^{(2)}a_2^{(2)} + W_{13}^{(2)}a_3^{(2)} + b_1^{(2)}) \end{aligned}$$



# 训练前馈网络

## □ 误差后向传播算法（BP）

- 训练样本集  $\{(\mathbf{x}^{(1)}, y^{(1)}), \dots, (\mathbf{x}^{(m)}, y^{(m)})\}$
- 代价函数

对于单个样例  $(\mathbf{x}, y)$  的代价函数  $J(\mathbf{W}, b; \mathbf{x}, y) = \frac{1}{2} \|\mathbf{h}_{\mathbf{W}, b}(\mathbf{x}) - y\|^2$

对于训练集的整体代价函数

$$\begin{aligned} J(\mathbf{W}, b) &= \left[ \frac{1}{m} \sum_{i=1}^m J(\mathbf{W}, b; \mathbf{x}^{(i)}, y^{(i)}) \right] + \frac{\lambda}{2} \sum_{l=1}^{n_l-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (W_{ji}^{(l)})^2 \\ &= \left[ \frac{1}{m} \sum_{i=1}^m \left( \frac{1}{2} \|\mathbf{h}_{\mathbf{W}, b}(\mathbf{x}^{(i)}) - y^{(i)}\|^2 \right) \right] + \frac{\lambda}{2} \sum_{l=1}^{n_l-1} \sum_{i=1}^{s_l} \sum_{j=1}^{s_{l+1}} (W_{ji}^{(l)})^2 \end{aligned}$$

以上公式中第一项是一个**均方差项**，第二项是一个**正则化项**（也叫权重衰减weight decay项），其目的是减小权重的幅度，防止过度拟合。



# 训练前馈网络

## □ 后向传播算法

- 参数随机初始化
- 进行前馈传导计算，利用前向传导公式，得到 $L_2, L_3, \dots, L_{n_l}$ 的输出值；
- 对于第 $n_l$ 层（输出层）的每个输出单元 $i$ ，我们根据以下公式计算残差：

$$\begin{aligned}\delta_i^{(n_l)} &= \frac{\partial}{\partial z_i^{(n_l)}} J(\mathbf{W}, b; \mathbf{x}, y) = \frac{\partial}{\partial z_i^{(n_l)}} \frac{1}{2} \|y - h_{\mathbf{W}, b}(\mathbf{x})\|^2 \\ &= -(y - a_i^{(n_l)}) \cdot f'(z_i^{(n_l)})\end{aligned}$$





# 训练前馈网络

## □ 后向传播算法

- 对  $l = n_l - 1, n_l - 2, n_l - 3, \dots, 2$  的各个层，第  $l$  层的第  $i$  个节点的残差计算方法如下：

$$\delta_i^{(n_l-1)} = \frac{\partial}{\partial z_i^{(n_l-1)}} J(\mathbf{W}, b; \mathbf{x}, y) = \frac{\partial}{\partial z_i^{(n_l)}} J(\mathbf{W}, b; \mathbf{x}, y) \cdot \frac{\partial z_i^{(n_l)}}{\partial z_i^{(n_l-1)}}$$

$$= \delta_i^{(n_l)} \cdot \frac{\partial z_i^{(n_l)}}{\partial z_i^{(n_l-1)}} = \delta_i^{(n_l)} \cdot \frac{\partial}{\partial z_i^{(n_l-1)}} \sum_{j=1}^{s_{n_l}} W_{ji}^{(n_l-1)} f(z_i^{(n_l-1)})$$

$$= \left( \sum_{j=1}^{s_{n_l}} W_{ji}^{(n_l-1)} \delta_i^{(n_l)} \right) f'(z_i^{(n_l-1)})$$

$$\delta_i^{(l)} = \left( \sum_{j=1}^{s_{l+1}} W_{ji}^{(l)} \delta_i^{(l+1)} \right) f'(z_i^{(l)})$$



# 训练前馈网络

## □ 后向传播算法（BP）

- 计算我们需要的偏导数，计算方法如下：

$$\frac{\partial}{\partial W_{ij}^{(l)}} J(\mathbf{W}, b; \mathbf{x}, y) = a_j^{(l)} \delta_i^{(l+1)}$$

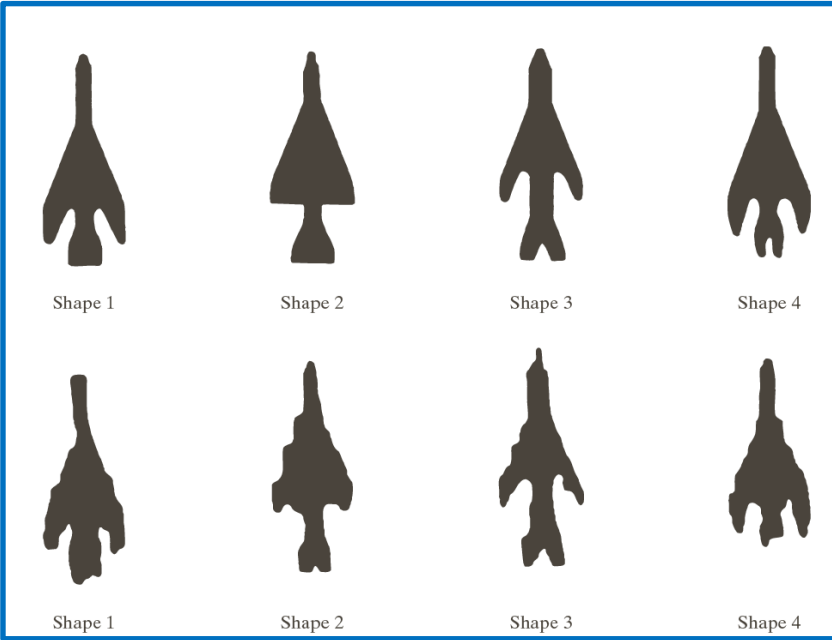
$$\frac{\partial}{\partial b_i^{(l)}} J(\mathbf{W}, b; \mathbf{x}, y) = \delta_i^{(l+1)}$$

- 更新参数：

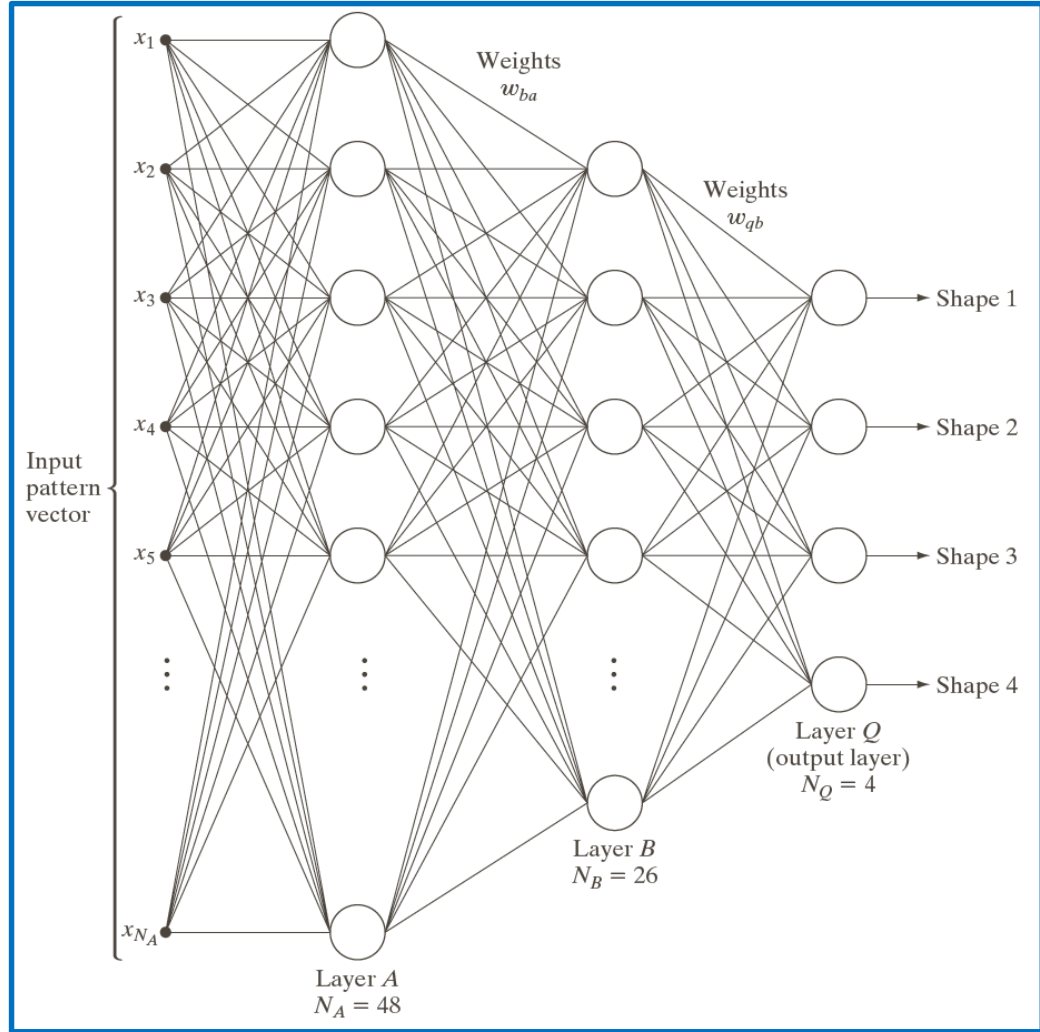
$$W_{ij}^{(l)} = W_{ij}^{(l)} - \alpha \left[ \frac{1}{m} a_j^{(l)} \delta_i^{(l+1)} + \lambda W_{ij}^{(l)} \right]$$

$$b_i^{(l)} = b_i^{(l)} - \frac{\alpha}{m} \delta_i^{(l+1)}$$

# 12.2.3 神经网络



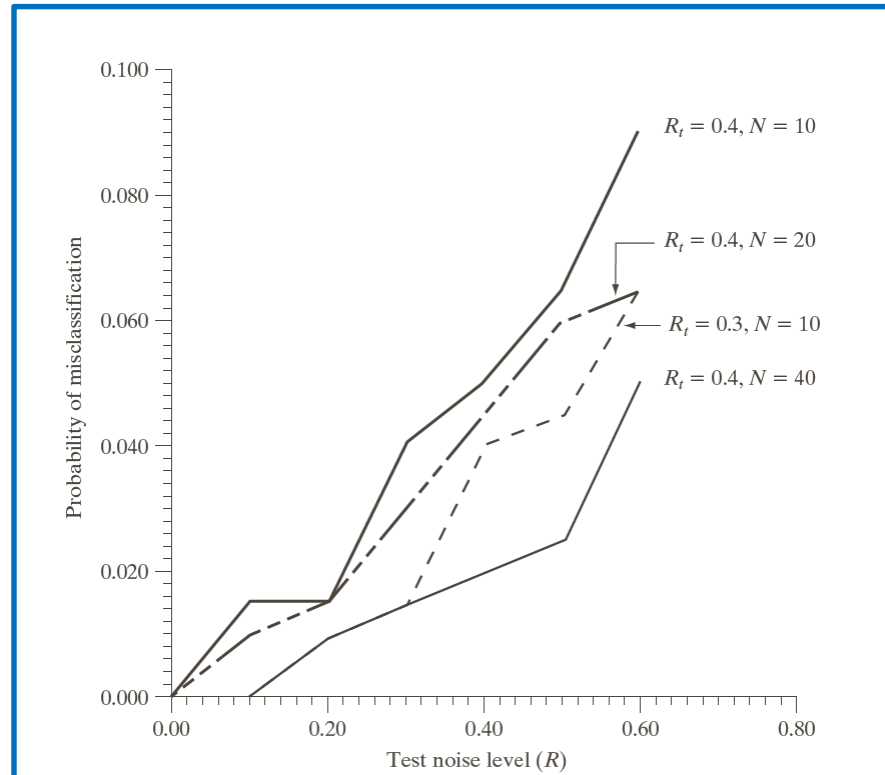
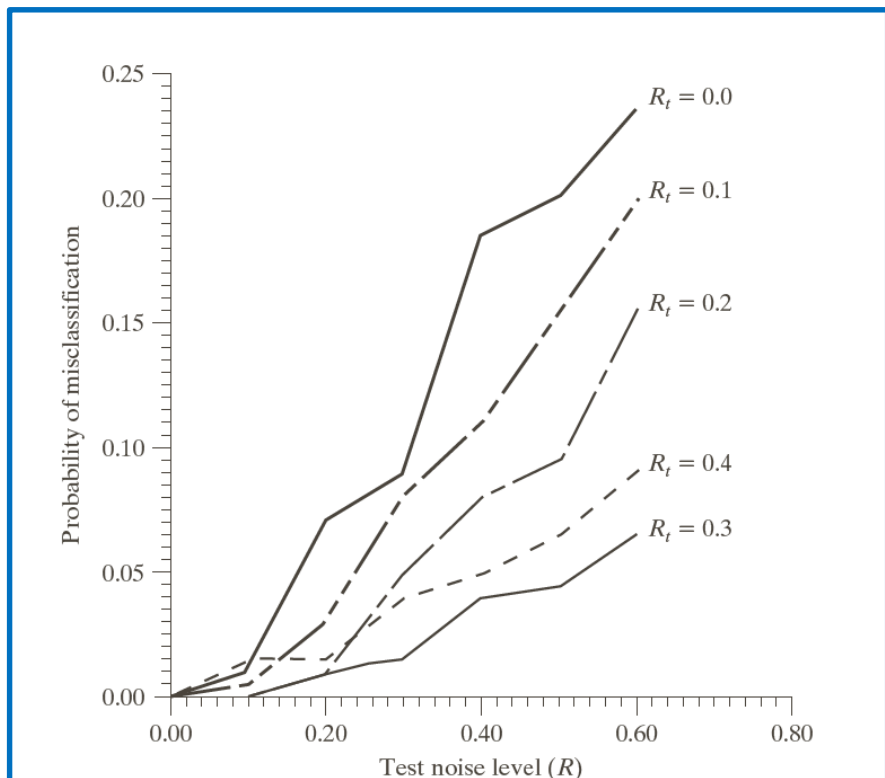
训练右图中的神经网络时所使用的  
(上) 参考图形和 (下) 典型的带  
噪声图形



用于识别左图所示形状三层神经网络



# 12.2.3 神经网络

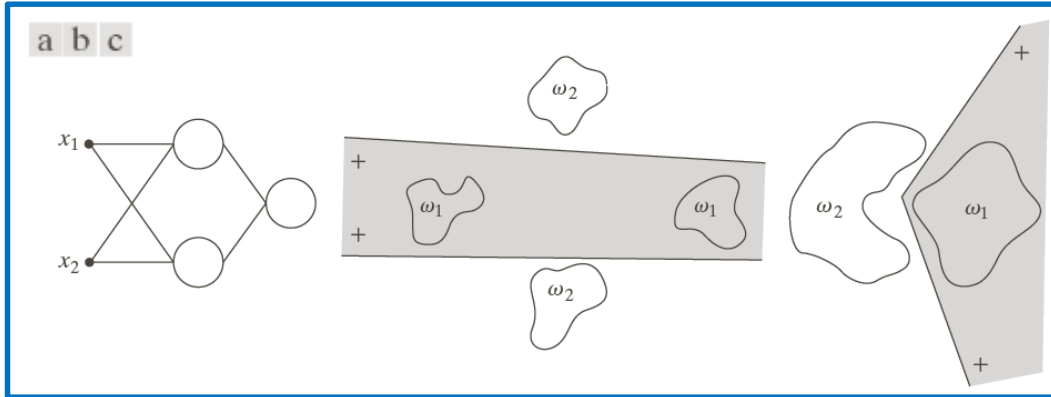


神经网络的性能是噪声水平的函数

增加训练模式的数量时,  $R_t = 0.4$ 时的性能改进 ( $R_t = 0.3$ 时的曲线仅作为参考)

# 12.2.3 神经网络

## □ 决策面的复杂性



(a) 一个双输入双层前馈神经网络；  
 (b) 和 (c) 可使用该网络实现的决策边界示例

Network structure	Type of decision region	Solution to exclusive-OR problem	Classes with meshed regions	Most general decision surface shapes
Single layer 	Single hyperplane			
Two layers 	Open or closed convex regions			
Three layers 	Arbitrary (complexity limited by the number of nodes)			

可以由单层和多层前馈网络带有一层或两层隐藏单元与两个输入形成的决策区域的类型



# 第12章 目标识别

- 12.1 模式和模式类
- 12.2 基于决策理论方法的识别
  - 匹配
  - 最佳统计分类器
  - 神经网络
- 12.3 结构方法
  - 匹配形状数
  - 串匹配



# 12.3 结构方法

## □ 匹配形状数

$$\begin{aligned} s_j(a) &= s_j(b), & i &= 4, 6, 8, \dots, k \\ s_j(a) &\neq s_j(b), & j &= k + 2, k + 4, \dots \end{aligned}$$

距离

$$D(a, b) = \frac{1}{k}$$

性质

$$\begin{aligned} D(a, b) &\geq 0 \\ D(a, b) &= 0, & \text{if } a &= b \\ D(a, c) &\leq \max[(D(a, b), D(b, c))] \end{aligned}$$

# 12.3 结构方法

## 匹配形状数

