

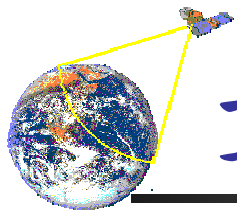
数字图像处理与分析

第八章 图像分析1

刘定生

中科院中国遥感卫星地面站

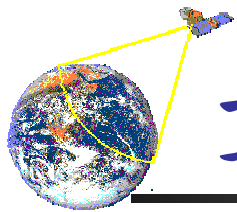
2004年春季学期



第八章图像分析1

■ 第八章 图像分析

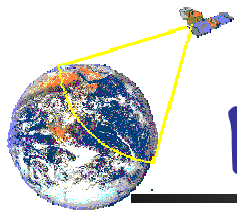
- 图像分析基本概念
- 图像分割
- 数学形态学图像处理
- 图像的表达与描述
- 图像的纹理分析



第八章图像分析1

■ 第八章 图像分析1

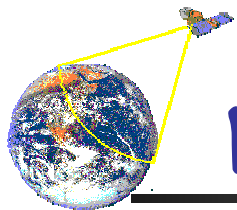
- 图像分析基本概念
- 图像分割
 - ✓ 图像分割引言
 - ✓ 边界分割法
 - ✓ 边缘连接法
 - ✓ Hough变换
 - ✓ 阈值分割法
 - ✓ 面向区域的分割
- 数学形态学图像处理



图像分析基本概念

■ 图像分析的概念

- 从图像中提取信息的技术
- 图象分析：“可以看作是一个信息提取过程，主要研究用自动或半自动的装置和系统，从图象中提取有用的数据、信息或度量，生成描述或表示”
- 也被称为图象理解、图象信息提取等，是当前研究比较活跃的领域
- 分析的任务是找到输入图象中感兴趣的内容，并进行相应的测量，形成对图象中感兴趣信息的合理描述
- 让计算机代替人，做更多的事情

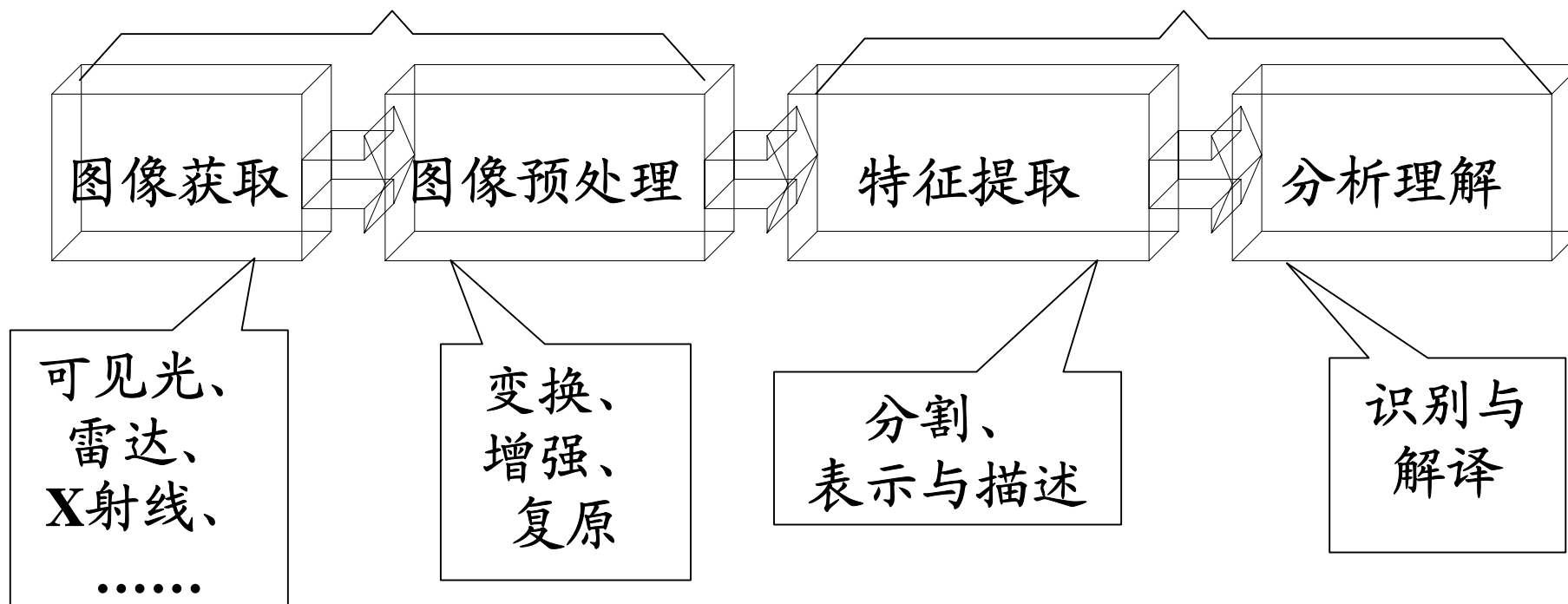


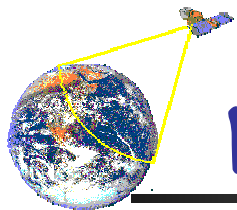
图像分析基本概念

■ 数字图像处理与分析系统基本框架

经典数字图像处理

数字图像分析

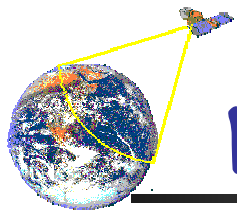




图像分析基本概念

■ 数字图像处理与分析研究内容

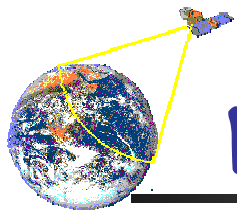
处理方法	分析方法
图象表示 图象变换, 图象增强, 图象复原, 图象重构, 图象编码,	图象边缘提取, 图象分割 图象形状描述, 图象纹理分析, 三维图象处理 运动图象分析 图象识别



图像分析基本概念

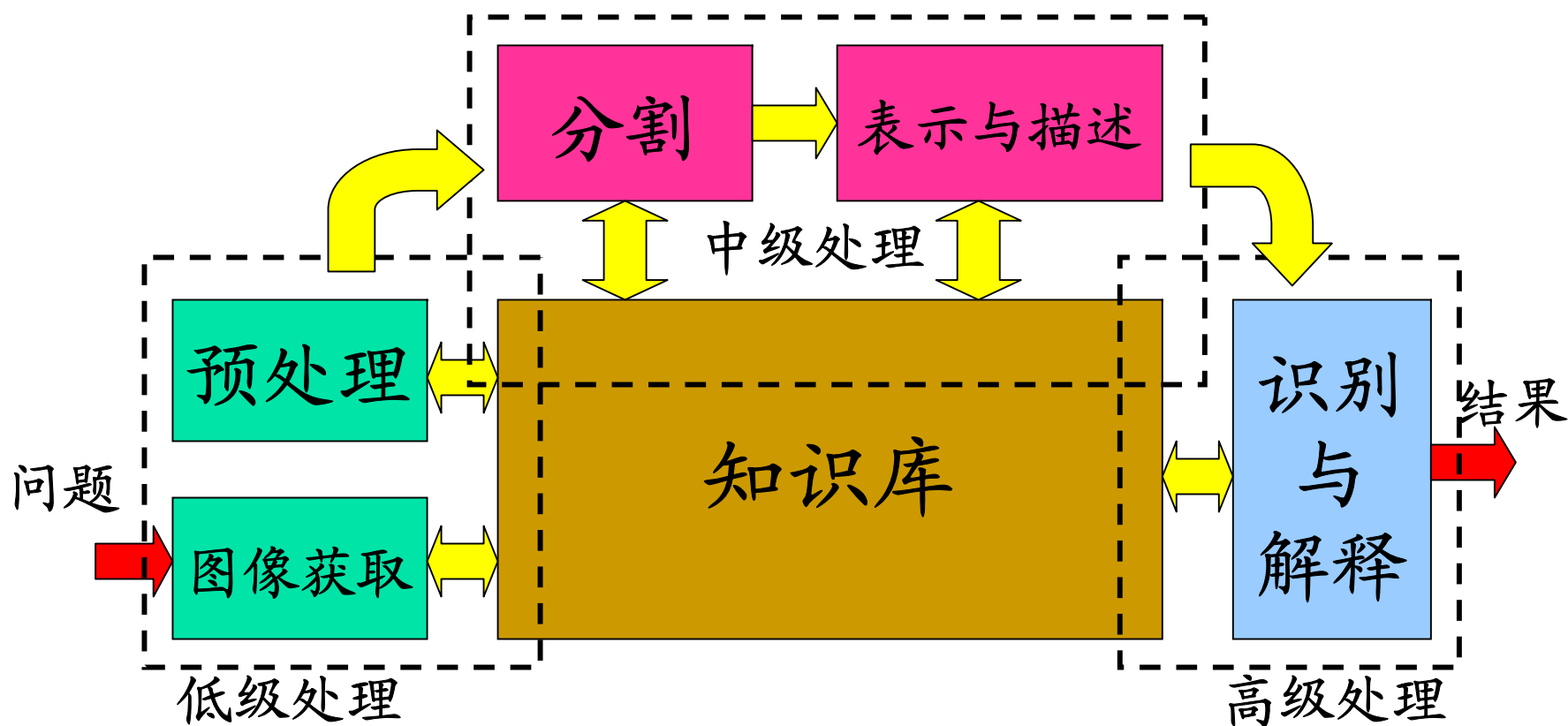
■ 数字图像处理与分析研究目标与方式的区别

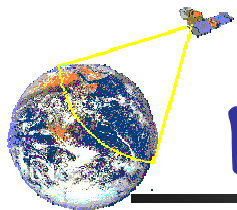
	图像处理	图象分析与理解
输出内容	图象	数值，符号
目的	主要是增强视觉效果 人看得更清楚	让计算机懂得图象内容
研究方法	理论上比较成熟	缺乏普适的方法， 具体问题具体分析 许多问题还很难解决
课程特点	理论推导	直观想像多
教材内容	大同小异	不同教材内容变化较大



图像分析基本概念

■ 数字图像处理与分析技术基本结构

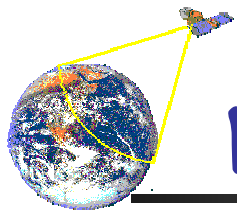




图像分析基本概念

■ 数字图像分析主要研究内容

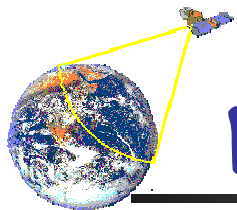
- 从图象中找到感兴趣的目标。景物内一般由目标组成，反映在图象中是众多的区域。为了把区域分开，我们要进行**图像分割**的研究
- 每个目标或区域由于某些特征的不同与其它区域区别开来，边缘、纹理、形状、颜色都是重要的特征。图象分割离不开**图象特征及其提取**方法的研究。
- 判断某种特征是否存在的判决也可以认为是一个识别的过程，判断每一个目标的类别也是一个识别的过程。高层次的分析理解更是大量需要使用识别技术，**模式识别**是一门专门的课程
- 对每个提取出来的目标还需要用能够有效的表示，这就需要研究**形状分析，表示与描述**方法



图像分割引言

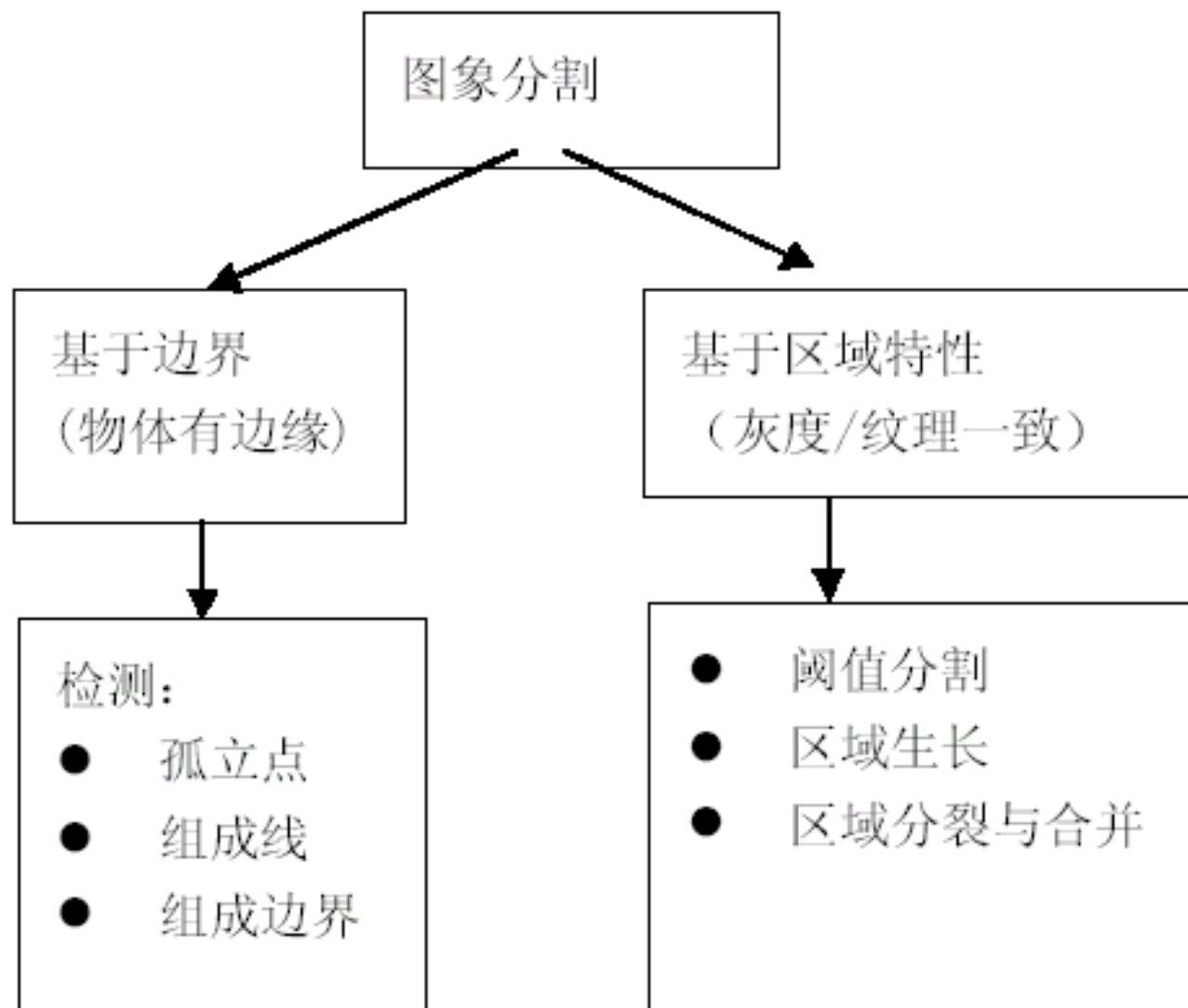
■ 图像分割技术的特征

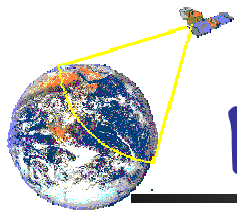
- 图象分割的任务是把图象分离成互不交叠的有意义的区域，以便于进一步的分析。它一般是图象分析的第一步
- 分开的区域一般是相对于图象中我们感兴趣的目标
- 图象处理中重要的基础环节，一般也是比较困难的环节
- 其它分析过程经常依赖于分割的结果，准确的分割影响甚至决定其它部分分析的准确程度
- 分割问题的困难在于图象数据的模糊和噪声的干扰
- 至今，还没有一个判断分割是否完全正确的准则，也没有一种标准的方法能够解决所有的分割问题。只有一些针对具体问题或要求满足一定条件的方法
- 分割得好坏必须从分割的效果来判断。实际景物中情况各异，具体问题具体分析，根据实际情况选择适合的方法
- 最基本的特征是图象的灰度值，彩色图象就是其颜色分量。边缘和纹理特征也是很常用的特征



图像分割引言

■ 比较常用的特征和方法



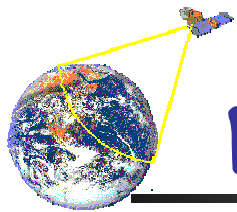


图像分割引言

■ 图像分割的概念

- 把图像分解成构成它的部件和对象的过程
- 有选择性地定位感兴趣对象在图像中的位置和范围



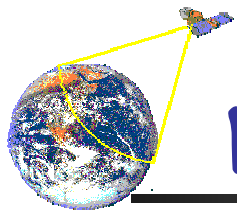


图像分割引言

■ 图像分割的基本思路

- 从简到难，逐级分割
- 控制背景环境，降低分割难度
- 把焦点放在增强感兴趣对象，缩小不相干图像成分的干扰上

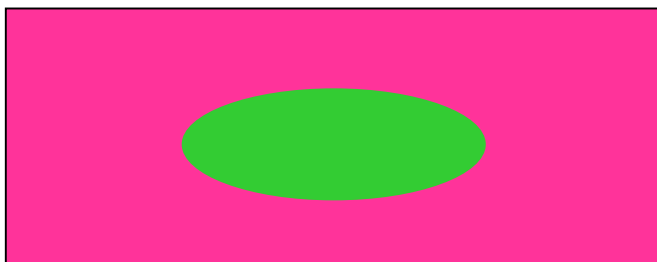


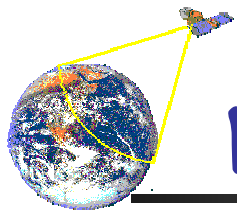


图像分割引言

■ 图像分割的基本策略

- 分割算法基于灰度值的两个基本特性：不连续性和相似性
- 检测图像像素灰度级的**不连续性**，找到点、线（宽度为1）、边（不定宽度）。先找边，后确定区域。

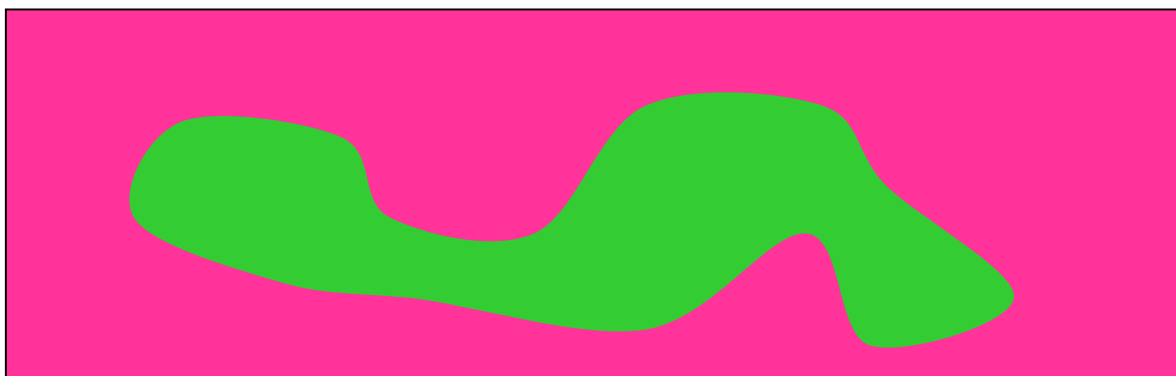


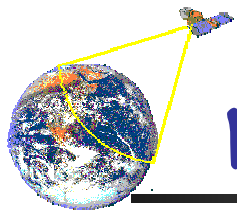


图像分割引言

■ 图像分割的基本策略

- 检测图像像素的灰度值的相似性，通过选择阈值，找到灰度值相似的区域，区域的外轮廓就是对象的边





图像分割:边界分割法

■ 点的检测

➤ 用空域的高通滤波器来检测孤立点

$$\begin{aligned}\text{例: } R &= (-1 * 8 * 8 + 128 * 8) / 9 \\ &= (120 * 8) / 9 = 960 / 9 = 106\end{aligned}$$

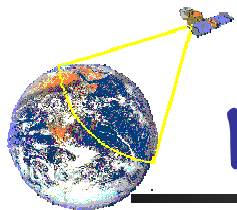
设：阈值：T = 64

图像

8	8	8
8	128	8
8	8	8

模板

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



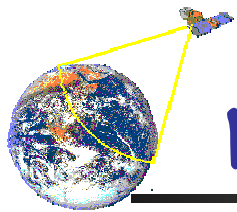
图像分割:边界分割法

■ 点的检测

- 如果 R 的值等于0, 说明当前检测点的灰度值与周围点相同
- 当 R 值足够大时, 说明该点值与周围的点非常不同, 是孤立点。通过阈值 T 来控制如 $T = 32、64、128$ 等

$|R| > T$ 便检测到一个孤立点

➤ 实例

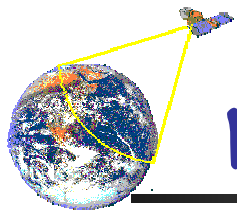


图像分割:边界分割法

■ 线的检测

- 通过比较典型模板的计算值, 确定一个点是否在某个方向的线上

R1	R2	R3	R4																																				
<table><tr><td>-1</td><td>-1</td><td>-1</td></tr><tr><td>2</td><td>2</td><td>2</td></tr><tr><td>-1</td><td>-1</td><td>-1</td></tr></table>	-1	-1	-1	2	2	2	-1	-1	-1	<table><tr><td>-1</td><td>-1</td><td>2</td></tr><tr><td>-1</td><td>2</td><td>-1</td></tr><tr><td>2</td><td>-1</td><td>-1</td></tr></table>	-1	-1	2	-1	2	-1	2	-1	-1	<table><tr><td>-1</td><td>2</td><td>-1</td></tr><tr><td>-1</td><td>2</td><td>-1</td></tr><tr><td>-1</td><td>2</td><td>-1</td></tr></table>	-1	2	-1	-1	2	-1	-1	2	-1	<table><tr><td>2</td><td>-1</td><td>-1</td></tr><tr><td>-1</td><td>2</td><td>-1</td></tr><tr><td>-1</td><td>-1</td><td>2</td></tr></table>	2	-1	-1	-1	2	-1	-1	-1	2
-1	-1	-1																																					
2	2	2																																					
-1	-1	-1																																					
-1	-1	2																																					
-1	2	-1																																					
2	-1	-1																																					
-1	2	-1																																					
-1	2	-1																																					
-1	2	-1																																					
2	-1	-1																																					
-1	2	-1																																					
-1	-1	2																																					
水平线	+45°线	垂直线	-45°线																																				



图像分割:边界分割法

■ 线的检测

1	1	1	1	1	1	1	1	1
5	5	5	5	5	5	5	5	5
1	1	1	1	1	1	1	1	1

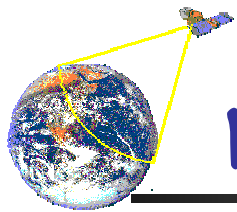
$$R_1 = -6 + 30 = 24$$

$$R_2 = -14 + 14 = 0$$

$$R_3 = -14 + 14 = 0$$

$$R_4 = -14 + 14 = 0$$

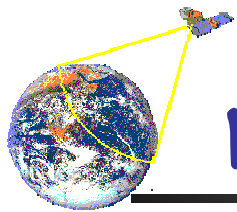
实例



图像分割:边界分割法

■ 边的检测

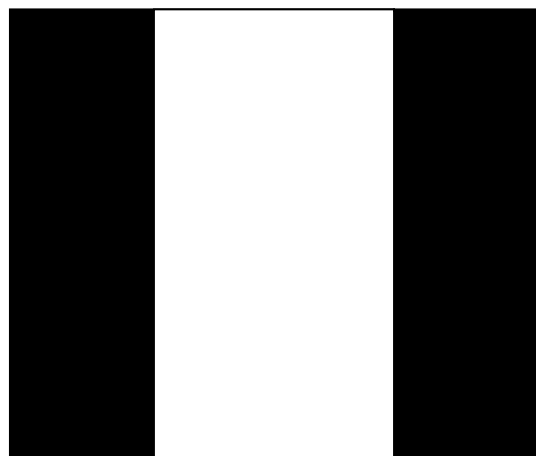
- 边界的定义：一段边是两个具有相对不同灰度值特性的区域的边界线
- 基本思想：计算局部微分算子
- 适用于：假定问题中的区域是非常类似的，两个区域之间的过渡，仅仅根据灰度的不连续性便可确定
- 不适用于：当假定不成立时，阈值分割技术一般来说比边缘检测更加实用



图像分割:边界分割法

■ 边的检测

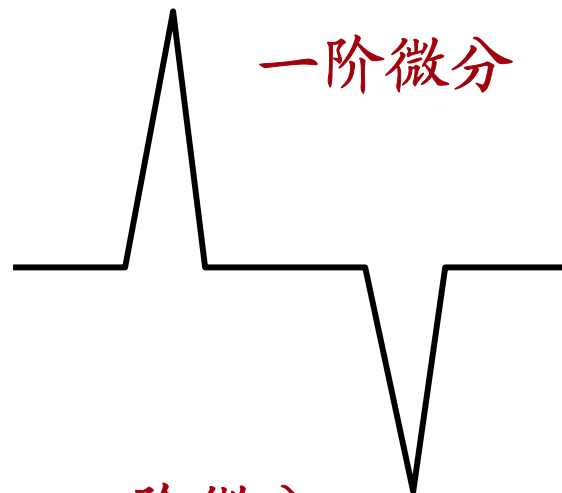
边界图像



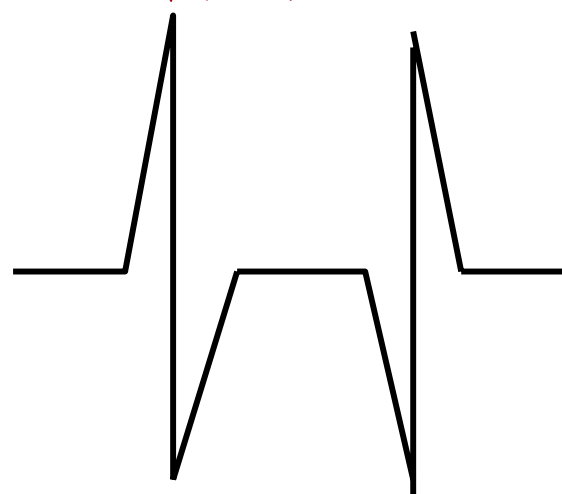
截面图

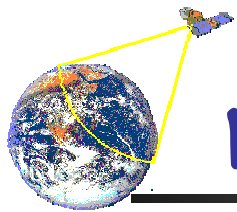


一阶微分



二阶微分





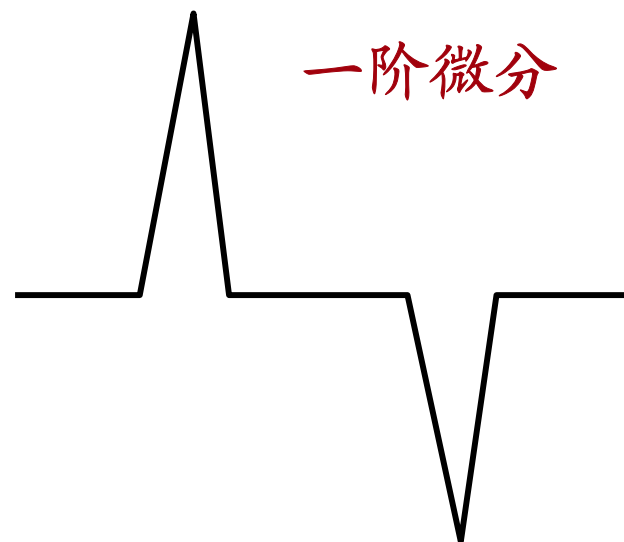
图像分割:边界分割法

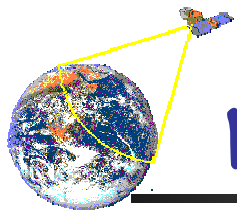
■ 边的检测

➤ 一阶微分: 用梯度算子来计算

✓ 特点: 对于亮的边, 边的变化起点是正的, 结束是负的。对于暗边, 结论相反。常数部分为零。

✓ 用途: 用于检测
图像中边的存在





图像分割:边界分割法

■ 边的检测

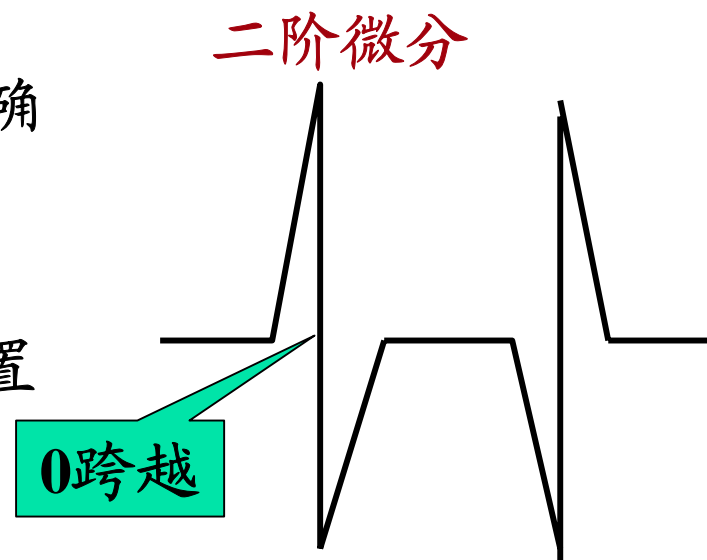
➤ 二阶微分: 通过拉普拉斯算子计算

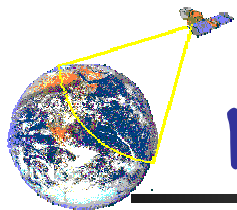
✓ 特点: 二阶微分在亮的一边是正的, 在暗的一边是负的。常数部分为零。

✓ 用途:

1) 二次导数的符号, 用于确定边上的像素是在亮的一边, 还是暗的一边。

2) 0跨越, 确定边的准确位置

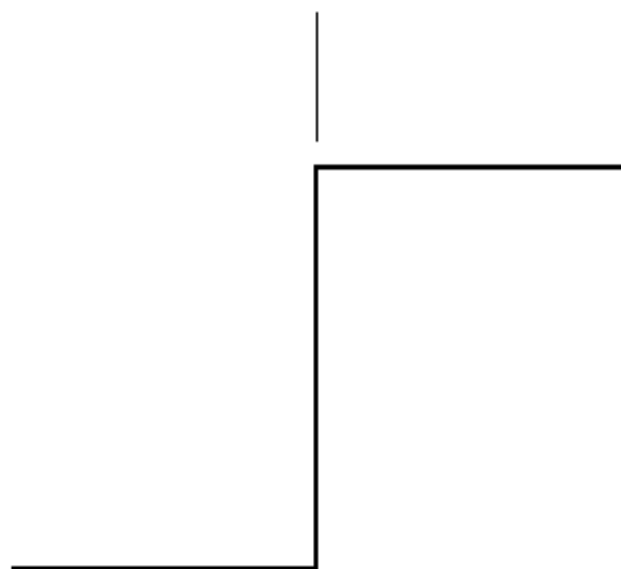




图像分割:边界分割法

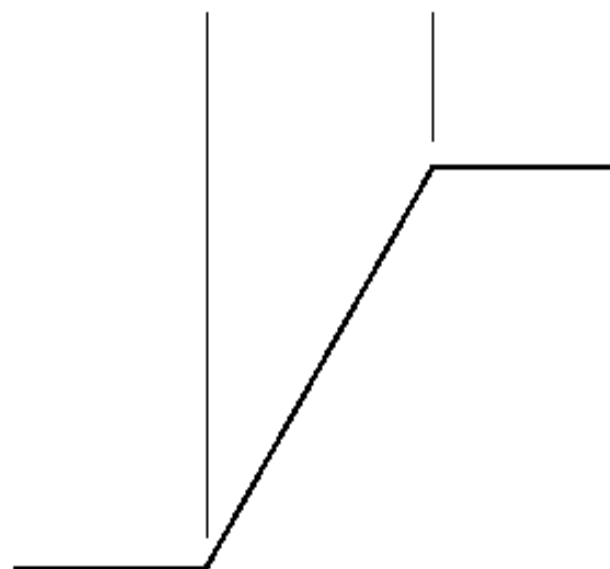
➤ 过渡性边缘

Model of an ideal digital edge



Gray-level profile
of a horizontal line
through the image

Model of a ramp digital edge

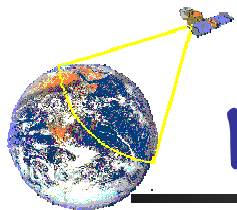


Gray-level profile
of a horizontal line
through the image

a b

FIGURE 10.5

(a) Model of an ideal digital edge.
(b) Model of a ramp edge. The slope of the ramp is proportional to the degree of blurring in the edge.



图像分割:边界分割法

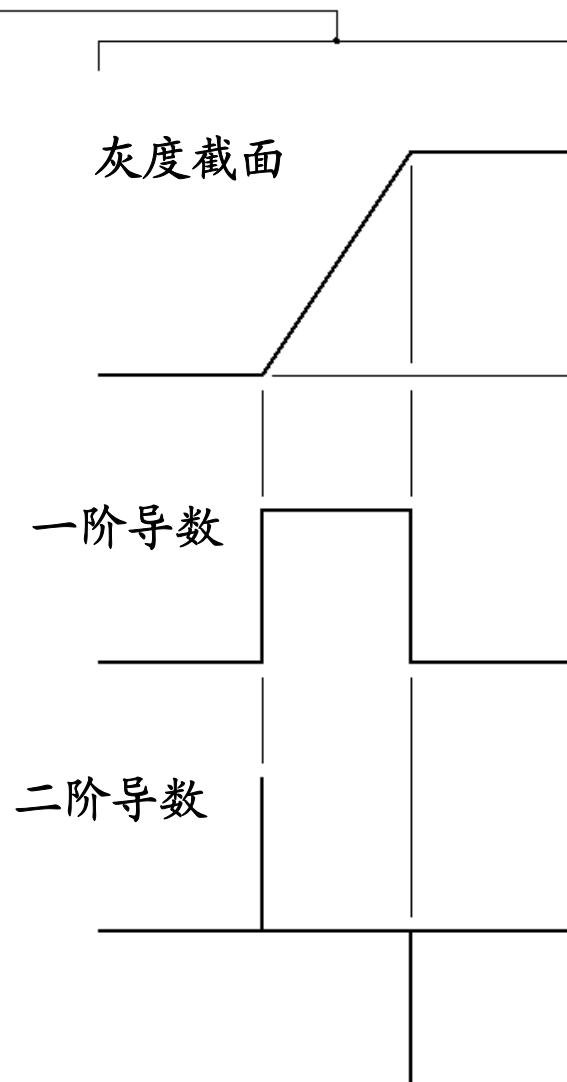
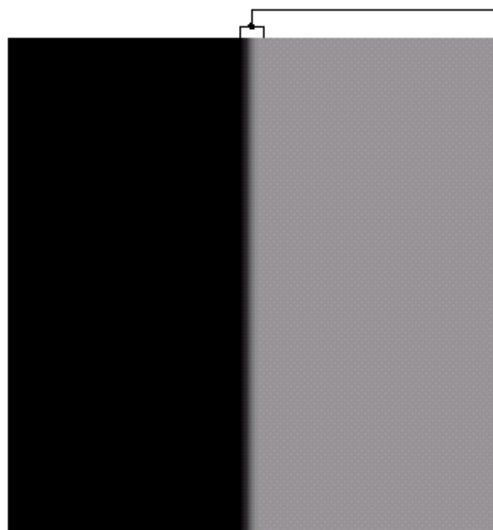
➤ 过渡性边缘的一阶微分和二阶微分检测

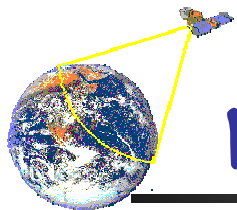
- ✓ 一阶导数检测
图像中像素点
是否为边缘

- ✓ 二阶导数判断该像素点在边缘
的哪一边

➤ 一阶与二阶微分算子边缘检测 受噪声干扰的影响

- ✓ 微分算子对噪声的敏感是边缘检测中的难题





图像分割:边界分割法

■ 边的检测

➤ 梯度算子

函数 $f(x,y)$ 在 (x,y) 处的梯度为一个向量:

向量指向坐标 (x,y) 处 f 的最大变化方向

计算这个向量的大小为:

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

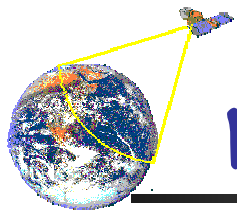
$$\nabla f(x, y) = \mathbf{mag}(\nabla \mathbf{f}) = \sqrt{G_x^2 + G_y^2} = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

近似为:

$$\nabla f(x, y) \approx |G_x| + |G_y|$$

梯度的方向角为:

$$\alpha(x, y) \approx \tan^{-1}\left(\frac{G_y}{G_x}\right)$$



图像分割:边界分割法

■ 边的检测

➤ 梯度算子

Sobel算子:

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

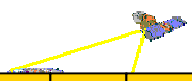
$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

$$G_x$$

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

$$G_y$$

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



0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	1	1	1	1	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

$$G_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

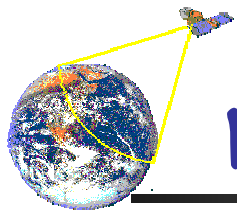
0	0	0	0	0	0	0	0
0	1	3	4	4	3	1	0
0	1	3	4	4	3	1	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	<u>1</u>	<u>3</u>	<u>4</u>	<u>4</u>	<u>3</u>	<u>1</u>	0
0	<u>1</u>	<u>3</u>	<u>4</u>	<u>4</u>	<u>3</u>	<u>1</u>	0
0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0
0	1	1	0	0	<u>1</u>	<u>1</u>	0
0	3	3	0	0	<u>3</u>	<u>3</u>	0
0	4	4	0	0	<u>4</u>	<u>4</u>	0
0	4	4	0	0	<u>4</u>	<u>4</u>	0
0	3	3	0	0	<u>3</u>	<u>3</u>	0
0	1	1	0	0	<u>1</u>	<u>1</u>	0
0	0	0	0	0	0	0	0

$$G_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

0	0	0	0	0	0	0	0
0	2	4	4	4	4	2	0
0	4	6	4	4	6	4	0
0	4	4	0	0	4	4	0
0	4	4	0	0	4	4	0
0	4	6	4	4	6	4	0
0	2	4	4	4	4	2	0
0	0	0	0	0	0	0	0

magnitude

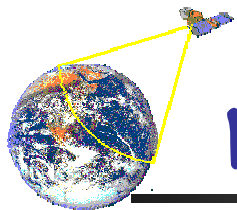


图像分割:边界分割法

■ 边的检测

➤ Sobel梯度算子的使用与分析

- ✓ 直接计算 G_x 、 G_y 可以检测到边的存在，以及从暗到亮，从亮到暗的变化
- ✓ 仅计算 $|G_x|$ ，产生最强的响应是正交于x轴的边；称之为水平梯度算子 G_x
- ✓ 仅计算 $|G_y|$ ，则产生最强的响应是正交于y轴的边；称之为垂直梯度算子 G_y
- ✓ 由于微分具有增强噪音的特性，平滑效果是Sobel算子特别引人注意的特性
- ✓ 实例



图像分割:边界分割法

■ 边的检测

➤ 其他梯度算子

Roberts

$$G_x = (z_9 - z_5), G_y = (z_8 - z_6)$$

$$\nabla f = \sqrt{[(z_9 - z_5)^2 + (z_8 - z_6)^2]}$$

$$\nabla f = |z_9 - z_5| + |z_8 - z_6|$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

-1	0	0	-1
0	1	1	0

Roberts

Prewitt

$$G_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

$$G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

$$\nabla f \cong |G_x| + |G_y|$$

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

Prewitt

Sobel

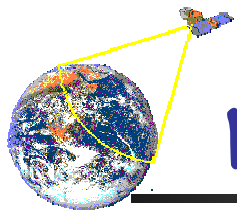
$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

$$\nabla f \cong |G_x| + |G_y|$$

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

Sobel



图像分割:边界分割法

■ 边的检测

➤ 其他梯度算子

专门检测对角线边缘的
Prewitt和Sobel算子

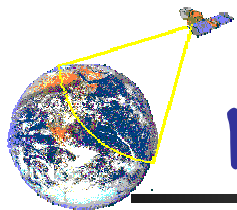
0	1	1	-1	-1	0
-1	0	1	-1	0	1
-1	-1	0	0	1	1

Prewitt

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

Sobel

Prewitt and Sobel masks for detecting diagonal edges.



图像分割:边界分割法

■ 边的检测

➤ 拉普拉斯

二维函数 $f(x,y)$ 的拉普拉斯是一个二阶的微分算子，
定义为：

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

可以用多种方式表示为数字形式。对于一个3x3的区域，经验上被推荐最多的形式有两种：

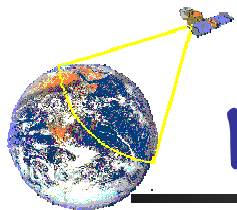
4邻域

$$\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8)$$

8邻域

$$\nabla^2 f = 8z_5 - \begin{pmatrix} z_1 + z_2 + z_3 + z_4 + \\ z_6 + z_7 + z_8 + z_9 \end{pmatrix}$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9



图像分割:边界分割法

■ 边的检测

➤ 拉普拉斯

- 定义数字形式拉普拉斯的基本要求是，作用于中心像素的系数是一个正数，而且其周围像素的系数为负数，系数之和必为0。

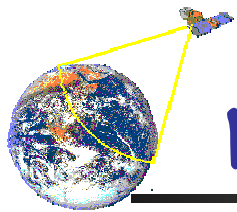
0	-1	0
-1	4	-1
0	-1	0

4-neighborhood

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

8-neighborhood

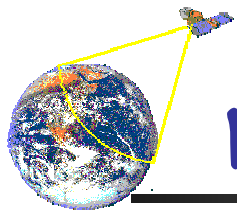
$$\sum_{i=1}^9 w_i = 0$$



图像分割:边界分割法

■ 边的检测

- 拉普拉斯算子的分析:
- 缺点:
 - ✓ 对噪声的敏感;
 - ✓ 会产生双边效果;
 - ✓ 不能检测出边的方向
- 应用: 拉普拉斯算子不直接用于边的检测, 通常只起第二位的角色;
 - ✓ 检测一个像素是在边的亮的一边还是暗的一边
 - ✓ 利用零跨越, 确定边的位置
- 通常应用拉普拉斯算子之前, 先对图像进行低通滤波。通常选用高斯低通滤波器与拉普拉斯算子合并, 形成一个单一的**高斯拉普拉斯算子 (LoG)**



图像分割:边界分割法

■ 边的检测

➤ 高斯拉普拉斯算子 (LoG)

$$h(r) = -\exp\left[\frac{-r^2}{2\sigma^2}\right]$$

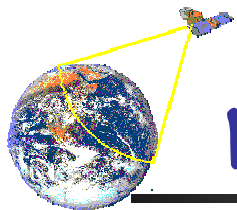
Gaussian function

$$h'(r) = \frac{r}{\sigma^2} \exp\left[\frac{-r^2}{2\sigma^2}\right]$$

$$\begin{aligned}\nabla^2 h(r) &= h''(r) = -\frac{1}{\sigma^2} \left(\frac{r^2}{\sigma^2} - 1 \right) \exp\left[\frac{-r^2}{2\sigma^2}\right] \\ &= -\left(\frac{r^2 - \sigma^2}{\sigma^4} \right) \exp\left[\frac{-r^2}{2\sigma^2}\right]\end{aligned}$$

Laplacian of
Gaussian function
(LoG)

Mexican hat
function



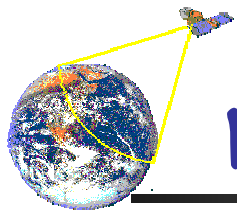
图像分割:边界分割法

■ 边的检测——高斯拉普拉斯算子 (LoG)

$$\begin{bmatrix}
 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 \\
 0 & 0 & -1 & -1 & -1 & -2 & -3 & -3 & -3 & -3 & -3 & -2 & -1 & -1 & -1 & 0 \\
 0 & 0 & -1 & -1 & -2 & -3 & -3 & -3 & -3 & -3 & -3 & -3 & -2 & -1 & -1 & 0 \\
 0 & -1 & -1 & -2 & -3 & -3 & -3 & -2 & -3 & -2 & -3 & -3 & -3 & -2 & -1 & -1 \\
 0 & -1 & -2 & -3 & -3 & -3 & -3 & 0 & 2 & 4 & 2 & -3 & -3 & -3 & -2 & -1 \\
 -1 & -1 & -3 & -3 & -3 & 0 & 4 & 10 & 12 & 10 & 4 & 0 & -3 & -3 & -3 & -1 \\
 -1 & -1 & -3 & -3 & -2 & 2 & 10 & 18 & 21 & 18 & 10 & 2 & -2 & -3 & -3 & -1 \\
 -1 & -1 & -3 & -3 & -3 & 4 & 12 & 21 & 24 & 21 & 12 & 4 & -3 & -3 & -3 & -1 \\
 -1 & -1 & -3 & -3 & -2 & 2 & 10 & 18 & 21 & 18 & 10 & 2 & -2 & -3 & -3 & -1 \\
 -1 & -1 & -3 & -3 & -3 & 0 & 4 & 10 & 12 & 10 & 4 & 0 & -3 & -3 & -3 & -1 \\
 0 & -1 & -2 & -3 & -3 & -3 & -3 & 0 & 2 & 4 & 2 & -3 & -3 & -3 & -2 & -1 \\
 0 & -1 & -1 & -2 & -3 & -3 & -3 & -2 & -3 & -2 & -3 & -3 & -3 & -2 & -1 & -1 \\
 0 & 0 & -1 & -1 & -2 & -3 & -3 & -3 & -3 & -3 & -3 & -3 & -2 & -1 & -1 & 0 \\
 0 & 0 & -1 & -1 & -1 & -2 & -3 & -3 & -3 & -3 & -3 & -2 & -1 & -1 & -1 & 0 \\
 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 & 0
 \end{bmatrix}
 \begin{bmatrix}
 0 & 0 & -1 & 0 & 0 \\
 0 & -1 & -2 & -1 & 0 \\
 -1 & -2 & 16 & -2 & -1 \\
 0 & -1 & -2 & -1 & 0 \\
 0 & 0 & -1 & 0 & 0
 \end{bmatrix}$$

5×5

17×17



图像分割:边界分割法

■ 边的检测——高斯拉普拉斯算子 (LoG)

- 算子的实质是对高斯滤波平滑图像进行拉普拉斯边缘提取
- 上述处理步骤亦可相互交换——运算顺序无关性

$$\nabla^2 [h(r) * f(x, y)] = \nabla^2 h(r) * f(x, y)$$

- 利用LoG算子处理图像后, 需再进行后续的边界判定处理——过零边界处理

- ✓ 用一个2x2的窗口, 在图像中移动

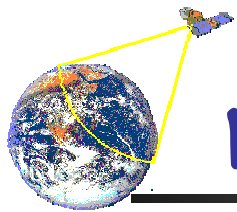
- ❖ 如果某一个像素与其他像素发生极性改变, 则将该像素设定为边界点

- ❖ 如果2x2窗口中没有极性改变, 则该窗口没有边界

➤ 应用实例

<0	>0
>0	>0

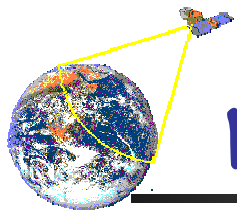
>0	>0
>0	>0



图像分割:边缘连接法

■ 边缘连接法

- 由于噪音的原因，边界的特征很少能够被完整地描述，在亮度不一致的地方会中断。
- 因此典型的边检测算法后面总要跟随着连接过程和其它边界检测过程，用来归整边像素，成为有意义的边。
- ✓ 局部处理法
- ✓ Hough变换



图像分割:边缘连接法

■ 局部连接处理

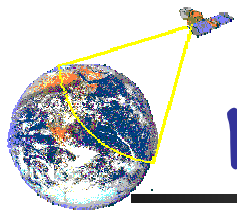
➤ 连接处理的时机和目的:

时机: 对做过边界检测的图像进行

目的: 连接间断的边

➤ 连接处理的原理:

用比较梯度算子的响应强度和梯度方向确定两个点是否同属一条边



图像分割:边缘连接法

■ 局部连接处理

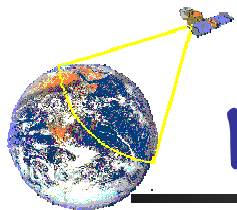
➤ 连接处理的原理:

➤ 通过比较梯度, 确定两个点的连接性:

对于点 (x', y') , 判断其是否与邻域内的边界点 (x, y) 相似, 当:

$$|\nabla f(x, y) - \nabla f(x', y')| \leq T$$

其中 T 是一个非负的阈值



图像分割:边缘连接法

■ 局部连接处理

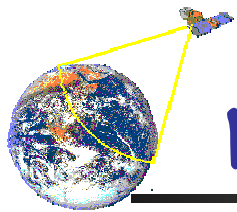
➤ 连接处理的原理:

➤ 比较梯度向量的方向角

对于点 (x',y') , 判断其是否与邻域内的边界点 (x,y) 的方向角相似, 当:

$$|\alpha(x,y) - \alpha(x',y')| \leq A$$

其中A是一个角度阈值

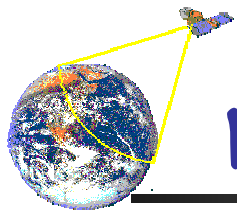


图像分割:边缘连接法

■ 局部连接处理

➤ 连接处理的原理:

当梯度值和方向角都是相似的, 则点 (x', y') , 与边点界 (x, y) 是连接的



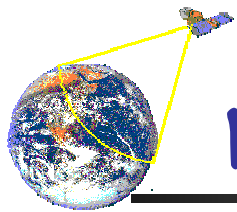
图像分割:边缘连接法

■ 局部连接处理

➤ 局部连接算法描述:

- 1) 设定A、T的阈值大小, 确定邻域的大小
- 2) 对图像上每一个像素的邻域点进行分析, 判断是否需要连接。
- 3) 记录像素连接的情况, 另开一个空间, 给不同的边以不同的标记。
- 4) 最后, 删除孤立线段, 连接断开的线段。

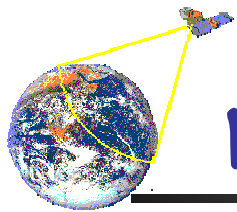
➤ 实例



图像分割：Hough变换

■ Hough变换

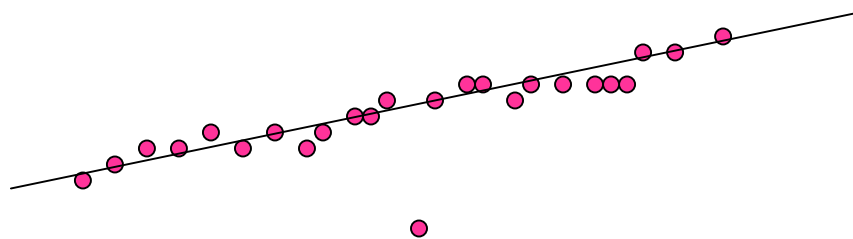
- 问题的提出
- Hough变换的基本思想
- 算法实现
- Hough变换的扩展



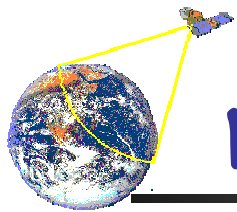
图像分割：Hough变换

■ Hough变换问题的提出

- 大量边沿算法，产生的可能是一系列相互孤立的边沿点集
- 在找出边界点集之后，需要连接，形成完整的边界图形描述



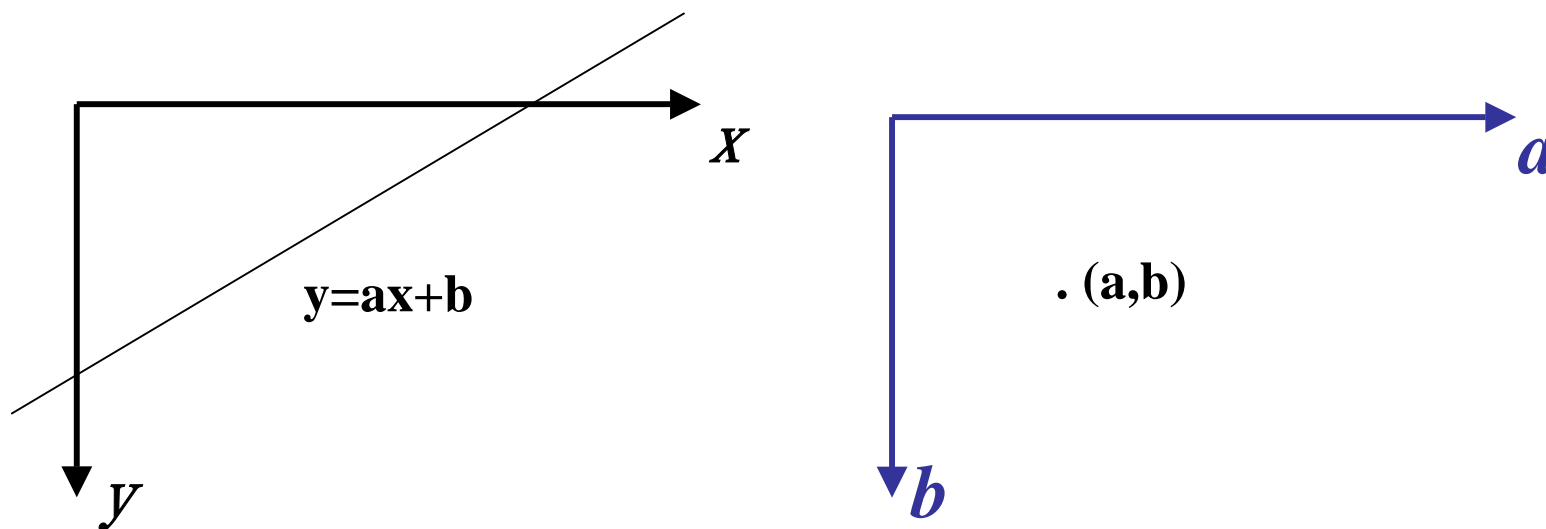
- 数学模型：对于边界上的 n 个点的点集，需找出共线的点集和直线方程

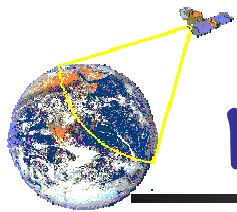


图像分割：Hough变换

■ Hough变换的基本思想

- 对于任意两点的直线方程： $y = ax + b$ ，以参数 a, b 作为坐标，构造一个 (a, b) 的平面，从而可将 (x, y) 平面的直线变换为 (a, b) 平面的一个点：





图像分割：Hough变换

■ Hough变换的基本思想

- xy 平面上的任意一点，对应参数 ab 平面上都有一条直线
- 如果点 (x_1, y_1) 与点 (x_2, y_2) 共线，那么这两点在参数 ab 平面上的直线将有一个交点

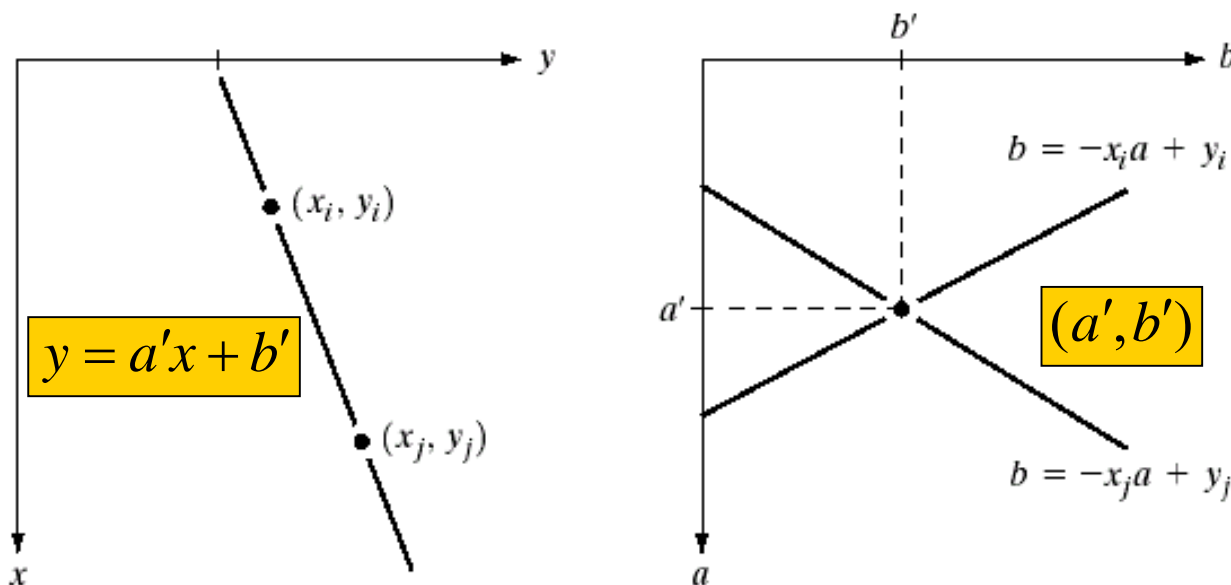
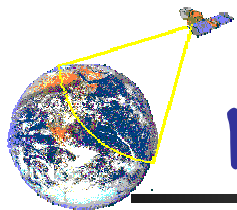


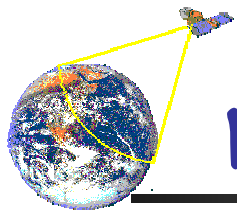
FIGURE 10.17
(a) xy -plane.
(b) Parameter space.



图像分割：Hough变换

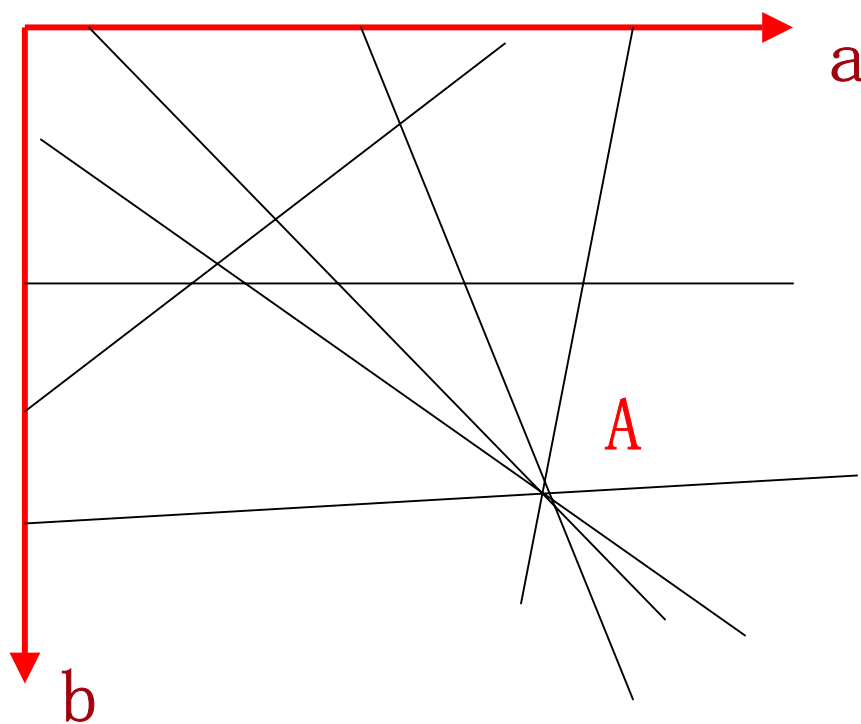
■ Hough变换的基本思想

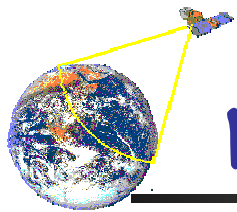
- 过 xy 平面一个点 (x,y) 的所有直线，构成参数 ab 平面上的
一条直线
- 在参数 ab 平面上相交直线最多的点，对应的 xy 平面上的
直线就是我们的解



图像分割：Hough变换

■ Hough变换的基本思想





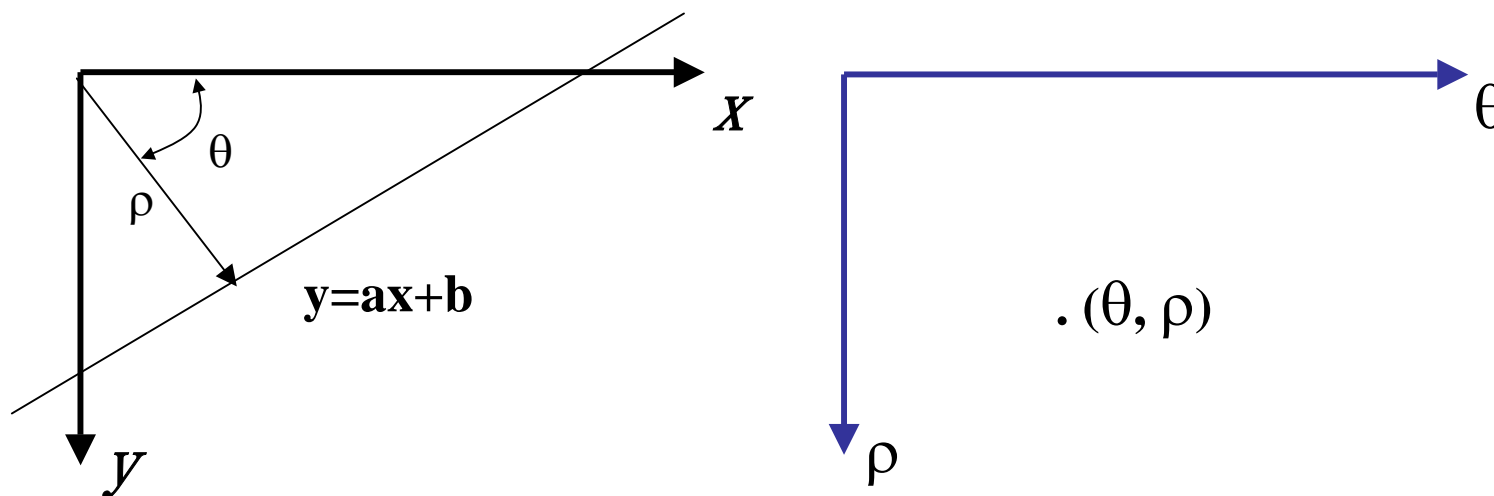
图像分割：Hough变换

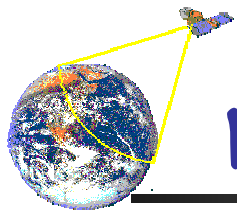
■ Hough变换算法实现

- 由于直线方程中参数 a 对应垂直直线时为无穷大，通常采用直线的极坐标方程：

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

- 形成 (θ, ρ) 参数平面





图像分割：Hough变换

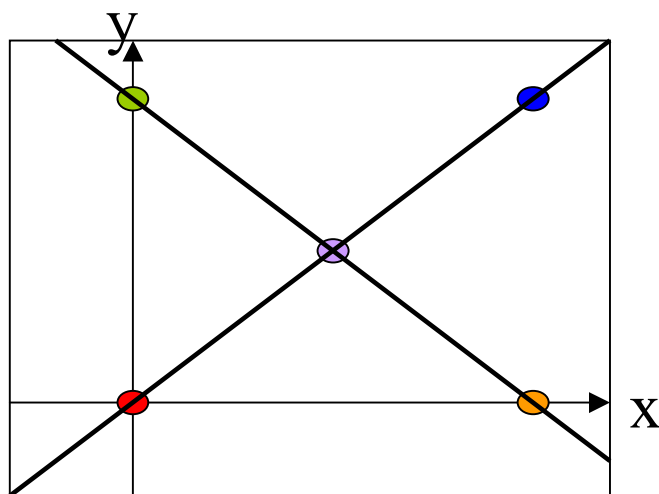
■ Hough变换算法实现

➤ xy 平面任意一点，对应 (θ, ρ) 平面上的一条正弦型曲线

$$(0,0) \Rightarrow \rho = 0$$

$$(2,0) \Rightarrow \rho = 2 \cos \theta$$

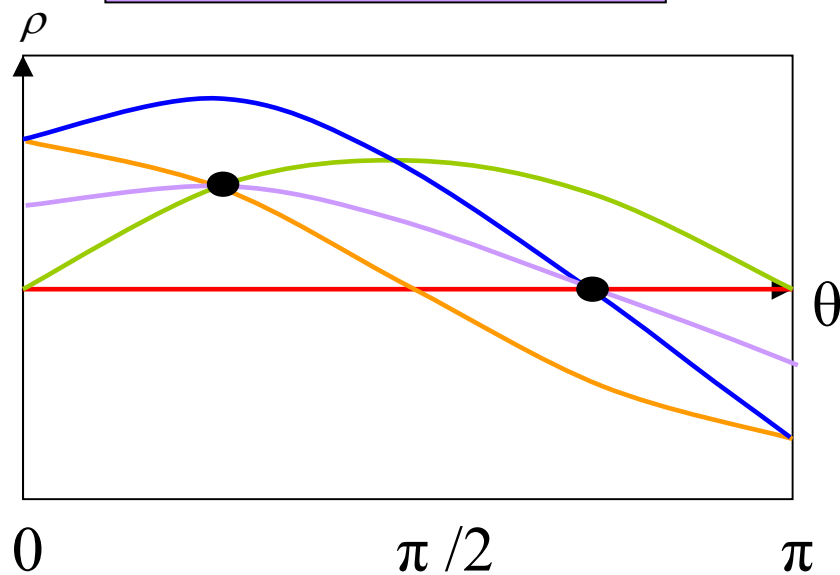
$$(0,2) \Rightarrow \rho = 2 \sin \theta$$

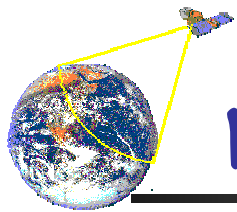


$$\rho = x \cos \theta + y \sin \theta$$

$$(2,2) \Rightarrow \rho = 2 \cos \theta + 2 \sin \theta$$

$$(1,1) \Rightarrow \rho = \cos \theta + \sin \theta$$

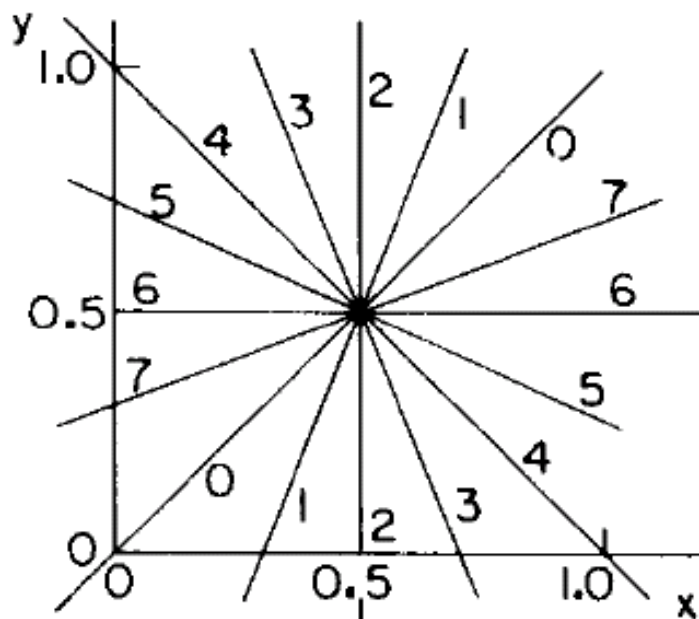




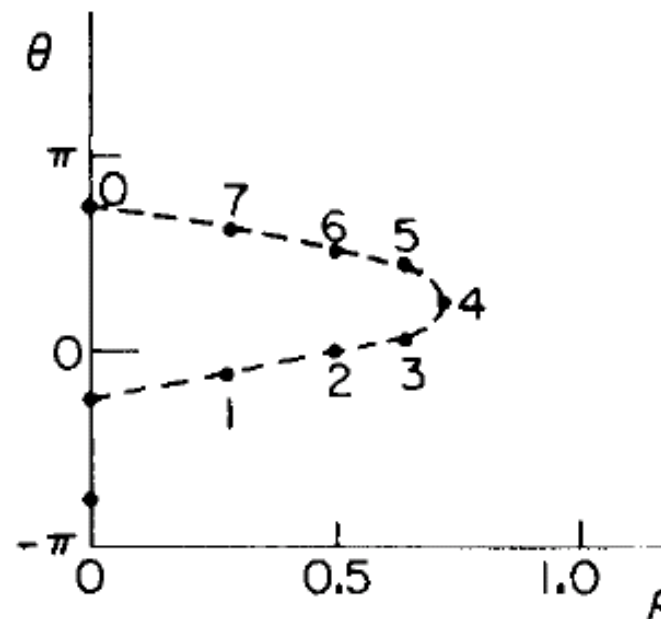
图像分割：Hough变换

■ Hough变换算法实现

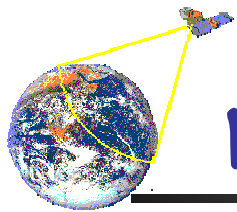
- xy 平面相交的一组直线，对应 (θ, ρ) 平面上的
一条曲线



(c) Family of lines, common point



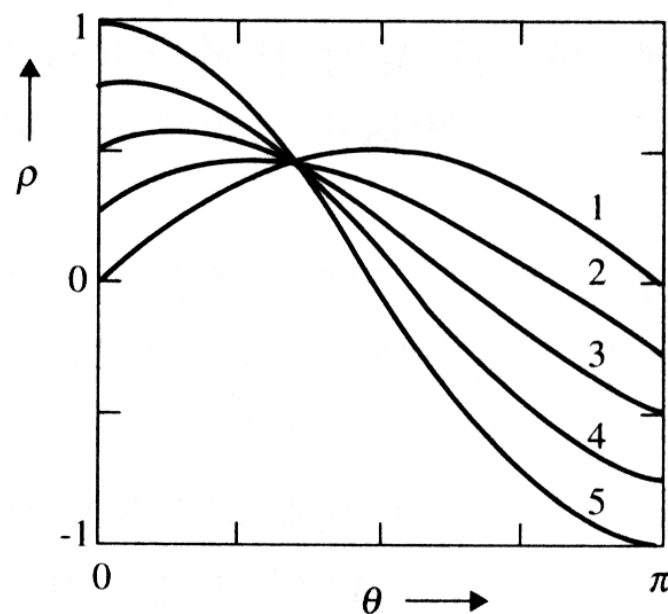
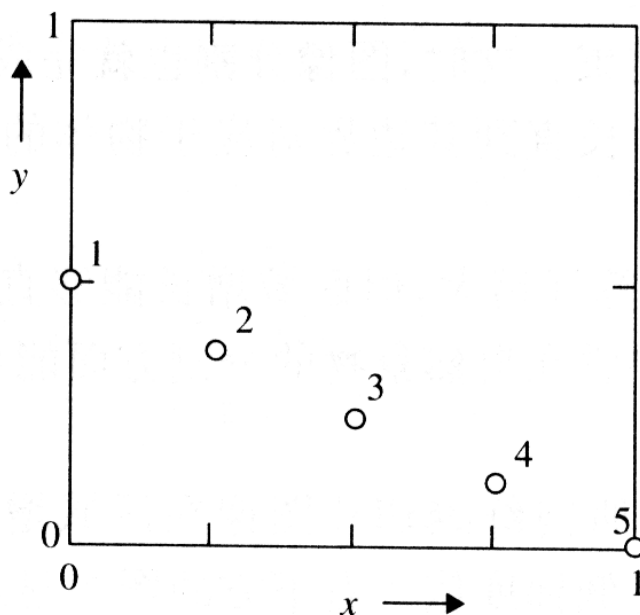
(d) Hough transform of (c)

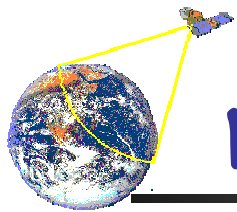


图像分割：Hough变换

■ Hough变换算法实现

- 若 (x,y) 平面上有一组点共线，则对应 (θ,ρ) 平面上各点的曲线相交于同一点





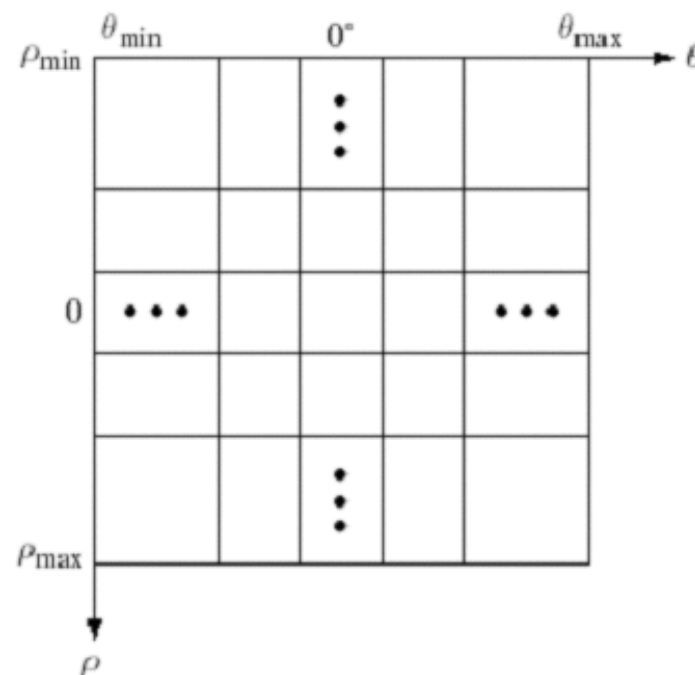
图像分割：Hough变换

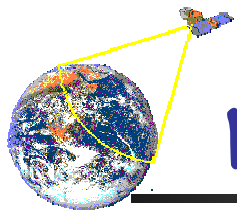
■ Hough变换检测步骤

- 对 (x,y) 平面各点 $\{(x_1,y_1), (x_2,y_2), \dots, (x_n,y_n)\}$, 形成 (θ, ρ) 平面上对应各条曲线的参数方程:

$$x_i \cos \theta + y_i \sin \theta = \rho \quad i=1,2,\dots,n$$

- 在 (θ, ρ) 平面上绘制各条曲线
- 对 θ, ρ 参数进行适当量化, 形成冗余度单元
- 凡位于同一单元内的各点, 看作为位于同一直线上
- 实例

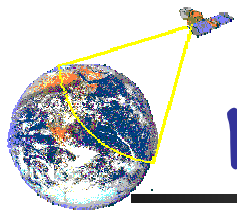




图像分割：Hough变换

■ Hough变换检测直线步骤

- 取 θ, ρ 合理的范围，并网格化，形成一个二维数组，称为累加数组，数组的每个值对应空间的一个矩形块
- 累加数组的每个元素置0
- (x, y) 空间上的每一点，做Hough变换，对应的空间曲线经过的每个网格，其数组值加1
- 找出数组中的局部极大值，其 (θ, ρ) ，值就是要检测的直线
- Hough变换应用实例



图像分割：Hough变换

■ Hough变换的扩展

➤ Hough变换不只对直线，也可以用于圆：

$$(x - c_1)^2 + (y - c_2)^2 = c_3^2$$

➤ 这时需要三个参数的参数空间 c_1, c_2, c_3

➤ 进一步扩展，可检测已知参数的物体形状

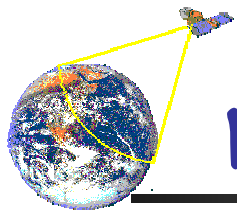
■ Hough变换的特征

➤ 图像空间的特定形状，在参数空间聚集成一个点

➤ 将困难的全局检测问题，转变为峰值探测问题

➤ 计算结果具有鲁棒性（**Robustness**）

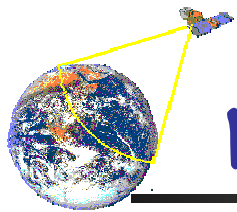
➤ 计算量大



图像分割:阈值分割法

■ 图像分割——阈值分割法

- 阈值分割法的基本思想
- 通过交互方式得到阈值
- 通过直方图得到阈值
- 通过边界特性选择阈值
- 简单全局阈值分割
- 分割连通区域
- 基于多个变量的阈值



图像分割:阈值分割法

■ 阈值分割法

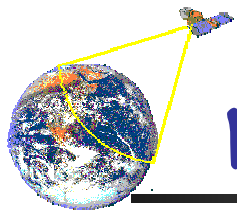
➤ 阈值分割法的基本思想:

- ✓ 确定一个合适的阈值T
- ✓ 将大于等于阈值的像素作为物体或前景，小于阈值的像素作为背景，生成一个二值图像

If $f(x,y) \geq T$ set 255

Else set 0

0	0	255
0	255	255
255	255	255



图像分割:阈值分割法

■ 阈值分割法

➤ 阈值分割基本模型

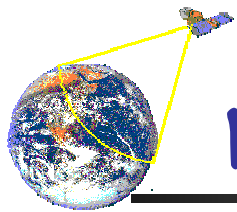
✓ 阈值分割作为一种图像变换，其输入为灰度图像 f ，输出为二进制图像 g ：

$$g(x, y) = \begin{cases} 1 & f(x, y) \geq T \quad (\text{object}) \\ 0 & f(x, y) < T \quad (\text{background}) \end{cases}$$

0	0	1
0	1	1
1	1	1

✓ 通用变换模型为：

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



图像分割:阈值分割法

■ 阈值分割法

➤ 阈值分割基本模型

✓ 基本变换模型类别:

❖ 全局阈值——对全图采用单一阈值

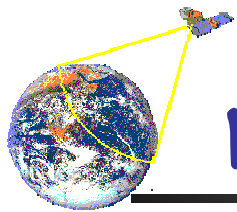
$$T = T(f(x, y))$$

❖ 局部阈值——不同局部采用不同阈值

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

❖ 动态或自适应阈值——每像素点阈值随像素特性而变化

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

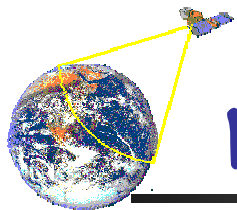


图像分割:阈值分割法

■ 阈值分割法

➤ 阈值分割法的特点:

- ✓ 简单，处理方便
- ✓ 适用于物体与背景有较强对比的情况，重要的是背景或物体的灰度比较单一，确定合适的阈值是关键
- ✓ 这种方法总可以得到封闭且连通区域的边界
- ✓ 噪声、前景或背景灰度变化范围大时将难以应用



图像分割:阈值分割法

■ 通过交互方式得到阈值

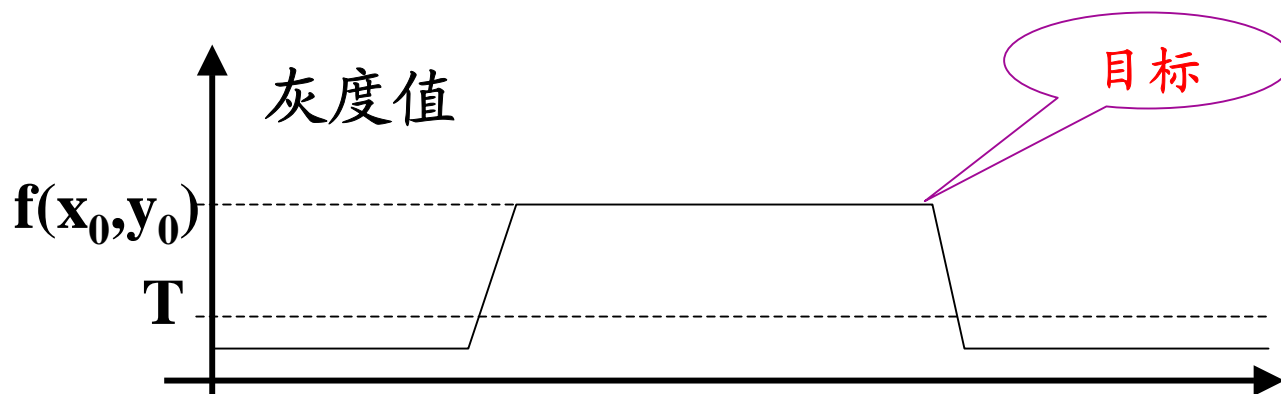
➤ 基本思想:

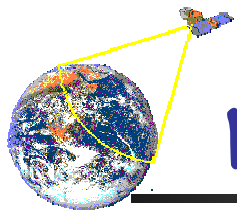
- ✓ 在交互方式下, 得到对象 (或背景) 的灰度值, 比得到阈值 T 容易得多。

假设: 对象的灰度值(也称样点值)为 $f(x_0, y_0)$, 且:

$$T = f(x_0, y_0) - R$$

其中 R 是容忍度, 可通过试探获得。





图像分割:阈值分割法

■ 通过交互方式得到阈值

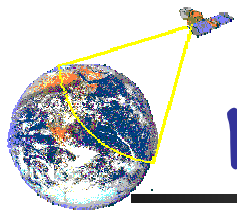
➤ 实施方法:

(1) 通过光标获得样点值 $f(x_0, y_0)$

(2) 选取容忍度 R

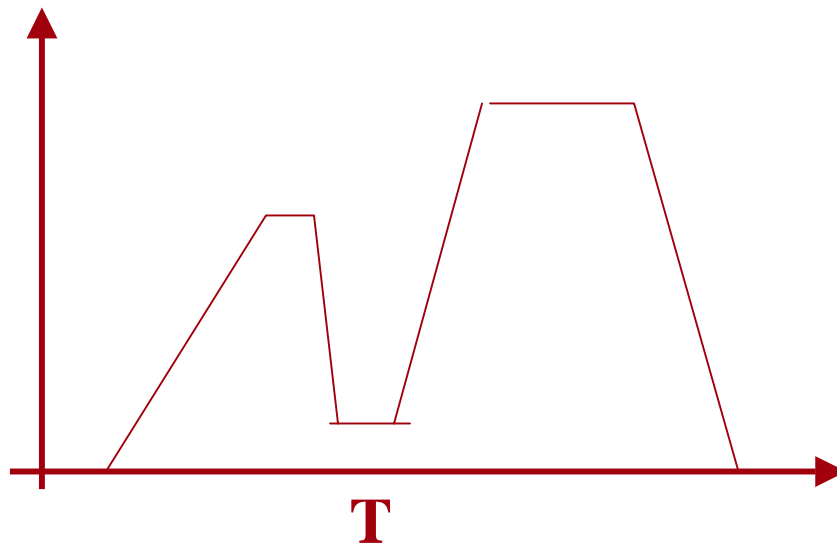
(3) **if $|f(x, y) - f(x_0, y_0)| \leq R$ set 255**
else set 0

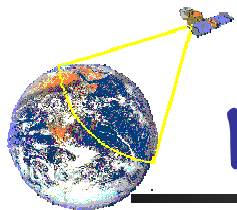
➤ 实例



图像分割:阈值分割法

- 通过直方图得到阈值——基本思想
边界上的点的灰度值出现次数较少





图像分割:阈值分割法

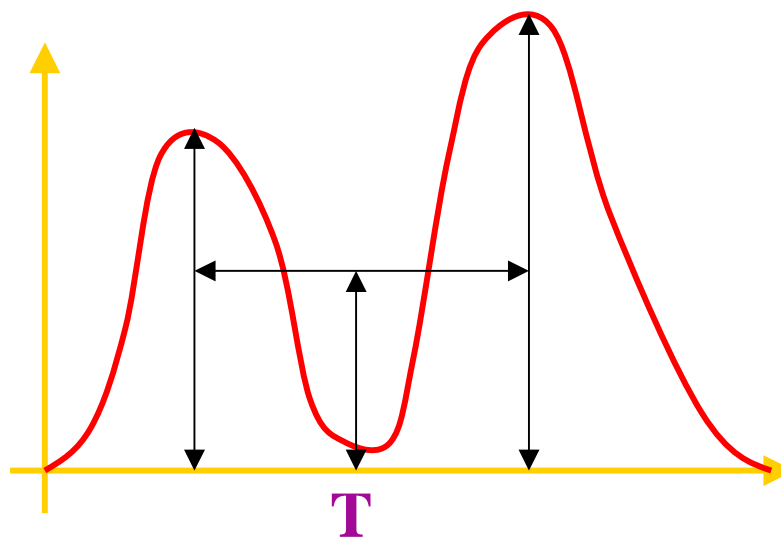
■ 通过直方图得到阈值—直接法

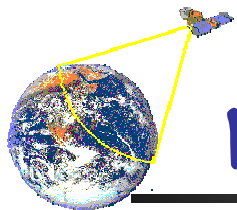
➤ 取值方法:

✓ 取直方图谷底, 以最小值的灰度值为阈值 T

➤ 缺点: 会受到噪音的干扰, 最小值不是预期的阈值, 而偏离期望的值;

➤ 改进: 取两个峰值之间某个固定位置, 如中间位置上。由于峰值代表的是目标与背景的典型值, 一般情况下, 比选谷底更可靠, 可排除噪音的干扰





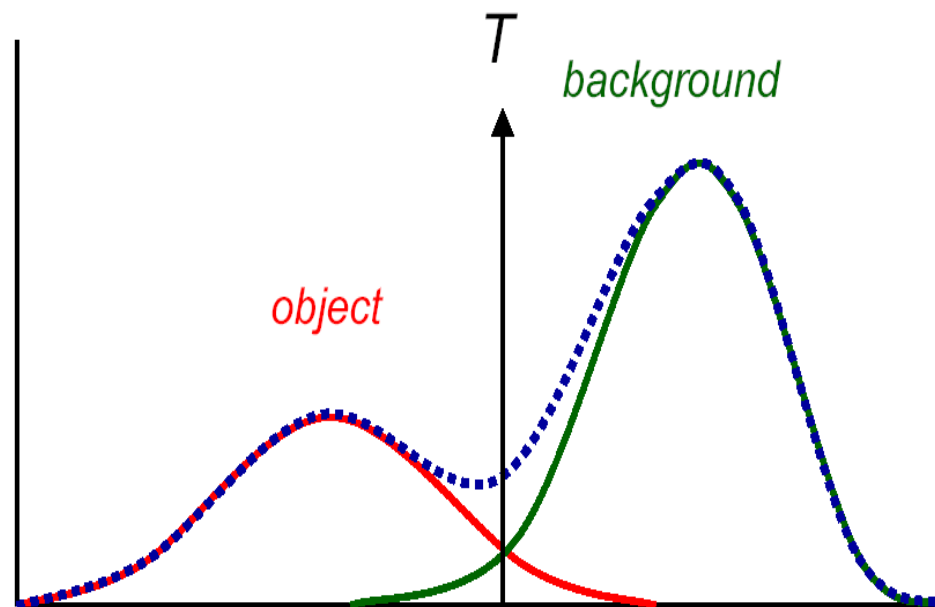
图像分割:阈值分割法

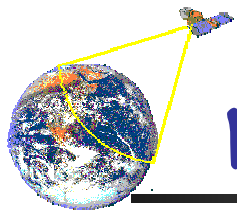
■ 通过直方图得到阈值—Isodata门限算法

- ✓ 选定一个初始门限 T_0
- ✓ 计算 T_0 左右两边的灰度平均值
- ✓ $T_1 = (T_0 \text{左右两边灰度平均值的均值})$
- ✓ 重复计算直至:

$$T_n = T_{n-1}$$

或 T_n 与 T_{n-1} 之差小于事先定义的常数 T_0 。

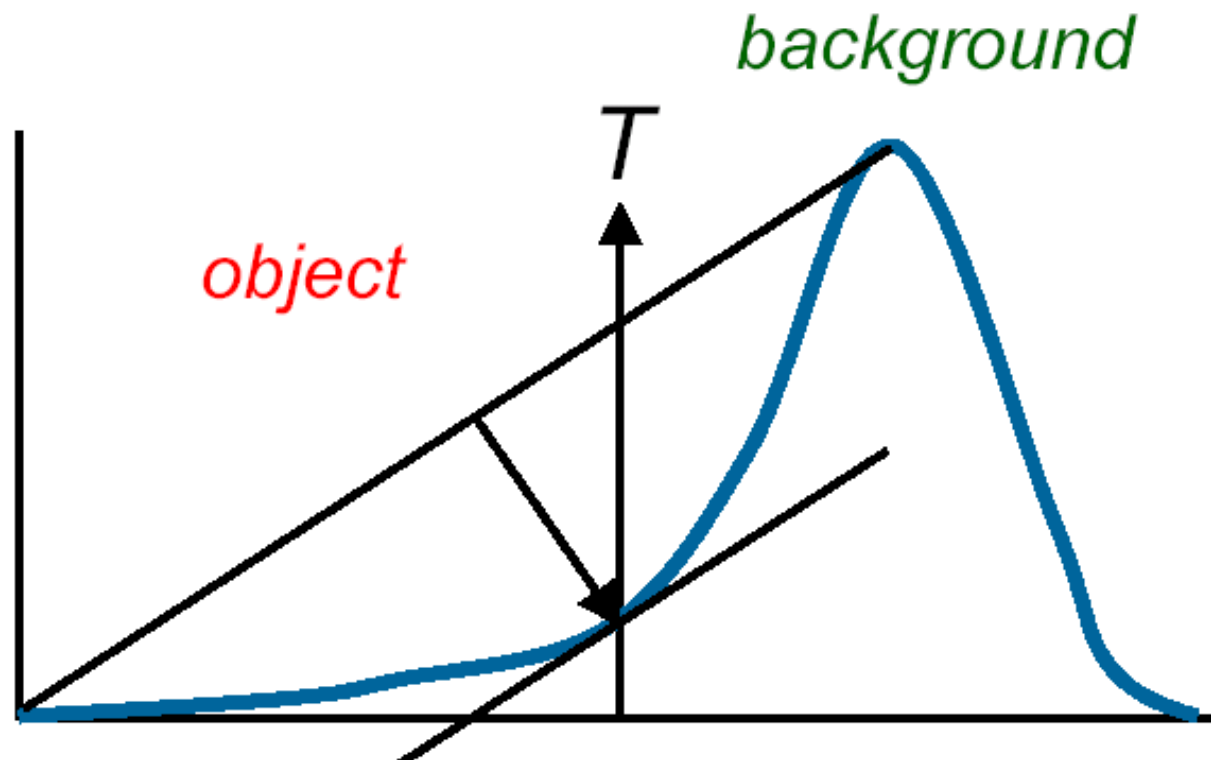


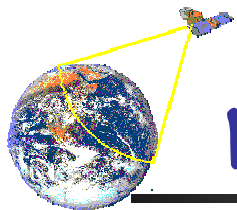


图像分割:阈值分割法

■ 通过直方图得到阈值—弓弦法

➤ 图像中存在大量背景和小的目标时





图像分割:阈值分割法

■ 通过直方图得到阈值——理论最佳全局门限

- “最佳”的准则——最小平均分割误差
- 假定一幅图像中仅包含两个主要的灰度级区域——背景和目标；从概率角度，其直方图可看作为这两种灰度级概率密度函数的迭加或混合

$$p(z) = P_1 p_1(z) + P_2 p_2(z)$$

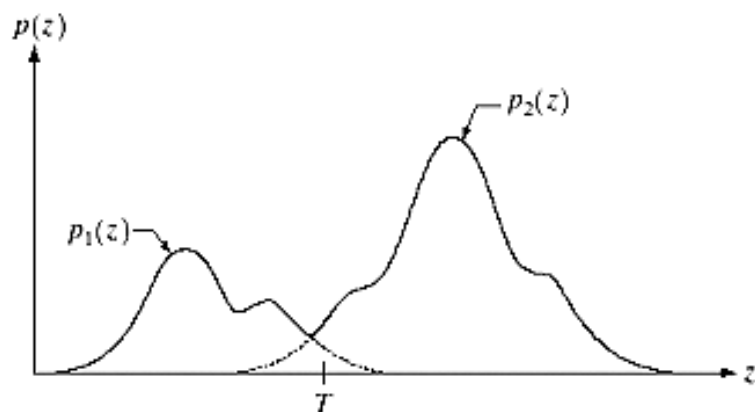
- 假定图像中任何像素不是属于背景就是属于目标，可有

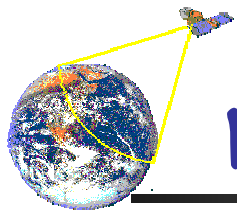
$$P_1 + P_2 = 1$$

- 设定一个门限 T ，按照 T 将像素分为背景与目标，由此产生的错误分类概率为：

$$E_1(T) = \int_{-\infty}^T p_2(z) dz$$

$$E_2(T) = \int_T^{\infty} p_1(z) dz$$





图像分割:阈值分割法

■ 通过直方图得到阈值——理论最佳全局门限

➤ 总体错误分类概率:

$$E(T) = P_2 E_1(T) + P_1 E_2(T)$$

➤ 使上式对T求极小, 可得出:

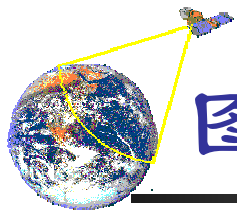
$$P_1 p_1(T) = P_2 p_2(T)$$

➤ 通常假定目标和背景的概率密度为高斯函数, 可得到最佳门限T的表达式为:

$$AT^2 + BT + C = 0$$

$$A = \sigma_1^2 - \sigma_2^2 \quad B = 2(\mu_1 \sigma_2^2 - \mu_2 \sigma_1^2)$$

$$C = \mu_2^2 \sigma_1^2 - \mu_1^2 \sigma_2^2 + 2\sigma_1^2 \sigma_2^2 \ln(\sigma_2 P_1 / \sigma_1 P_2)$$

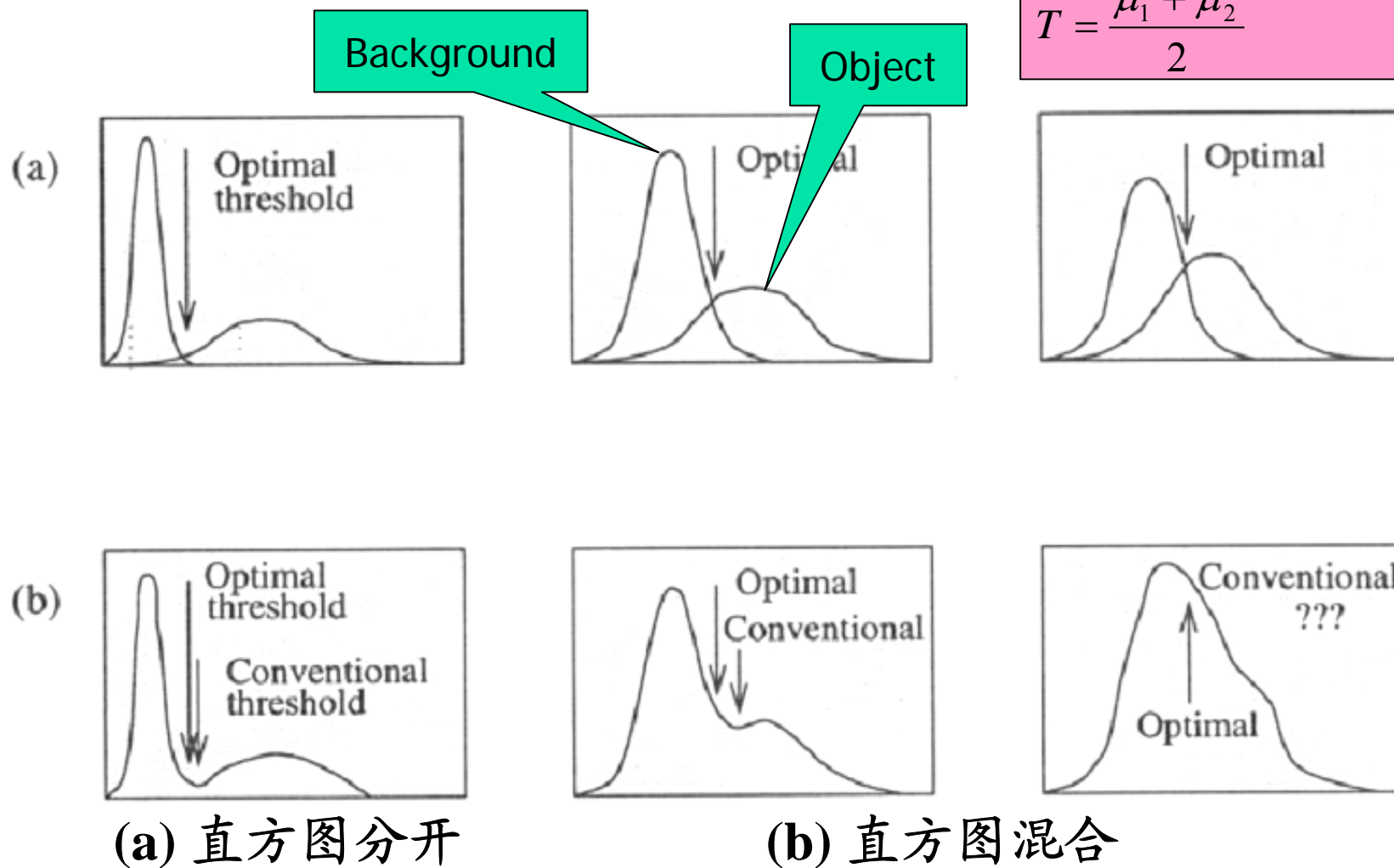


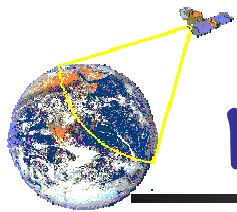
图像分割: 阈值分割法

- 通过直方图得到阈值——理论最佳全局门限

If $\sigma_1^2 = \sigma_2^2 = \sigma^2$ and $P_1 = P_2$

$$T = \frac{\mu_1 + \mu_2}{2}$$





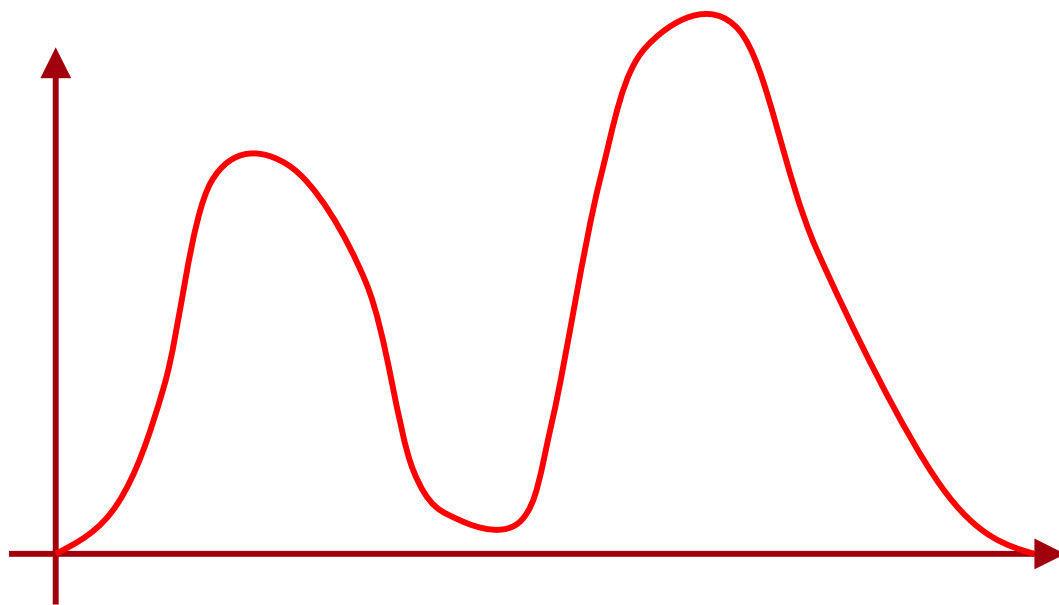
图像分割:阈值分割法

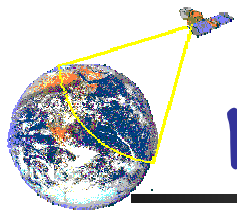
■ 通过直方图得到阈值

➤ 对噪音的处理

对直方图进行平滑处理

➤ 实例



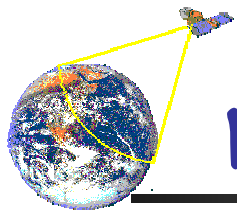


图像分割:阈值分割法

■ 通过边界特性选择阈值

➤ 基本思想:

- ✓ 如果直方图的各个波峰很高、很窄和对称，且被很深的波谷分开时，有利于选择阈值。
- ✓ 为了改善直方图的波峰形状，我们只把区域边缘的像素绘入直方图，而不考虑区域中间的像素。
- ✓ 用微分算子，处理图像，使图像只剩下边界中心两边的值。



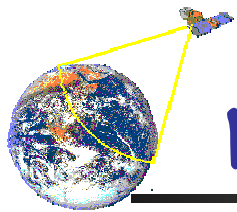
图像分割:阈值分割法

■ 通过边界特性选择阈值

➤ 基本思想:

✓ 这种方法有以下优点:

- 1) 在前景和背景所占区域面积差别很大时, 不会造成一个灰度级的波峰过高, 而另一个过低
- 2) 边缘上的点在区域内还是区域外的概率是相等的, 因此可以增加波峰的对称性
- 3) 基于梯度和拉普拉斯算子选择的像素, 可以增加波峰的高度



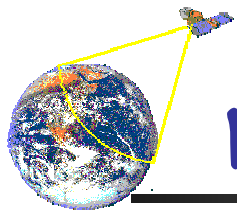
图像分割:阈值分割法

■ 通过边界特性选择阈值

➤ 算法的实现:

- 1) 对图像进行梯度计算, 得到梯度图像。
- 2) 得到梯度值最大的那一部分 (比如**10%**) 的像素直方图
- 3) 通过直方图的谷底, 得到阈值**T**。

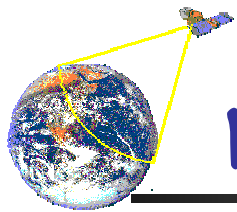
➤ 如果用拉普拉斯算子, 不通过直方图, 可直接得到阈值, 方法是使用拉普拉斯算子过滤图像, 将**0**跨越点对应的灰度值为阈值**T**。



图像分割:阈值分割法

■ 全局阈值分割

- 基本思想: 用前述方法获得阈值 T , 对全图采用该阈值, 产生一个二值图, 区分出前景对象和背景
- 算法实现:
 - ✓ 规定一个阈值 T , 逐行扫描图像。
 - ✓ 凡灰度级大于 T 的, 颜色置为255; 凡灰度级小于 T 的, 颜色置为0。
- 适用场合: 明度图像是可以控制的情况, 例如用于工业监测系统中。



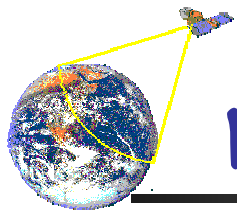
图像分割:阈值分割法

■ 局部阈值分割

- 将图像实现区分为不同的子图
- 对每一个子图分别选定一个门限值进行分割

■ 自适应阈值分割

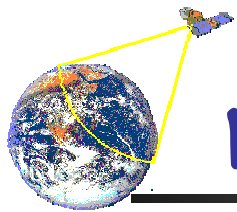
- 按照图像中各像素的局部特征，自适应确定该像素的阈值，进而按照阈值进行分类



图像分割:阈值分割法

■ 分割连通区域

- 基本思想: 用前述方法获得阈值 T , 并产生一个二值图, 区分出单独的连通前景对象和背景区域
- 算法实现:
 - ✓ 规定一个阈值 T , 上下左右4个方向进行逐行扫描图像
 - ✓ 凡灰度级大于 T 的, 颜色置为255; 凡灰度级小于 T 的, 颜色置为0。

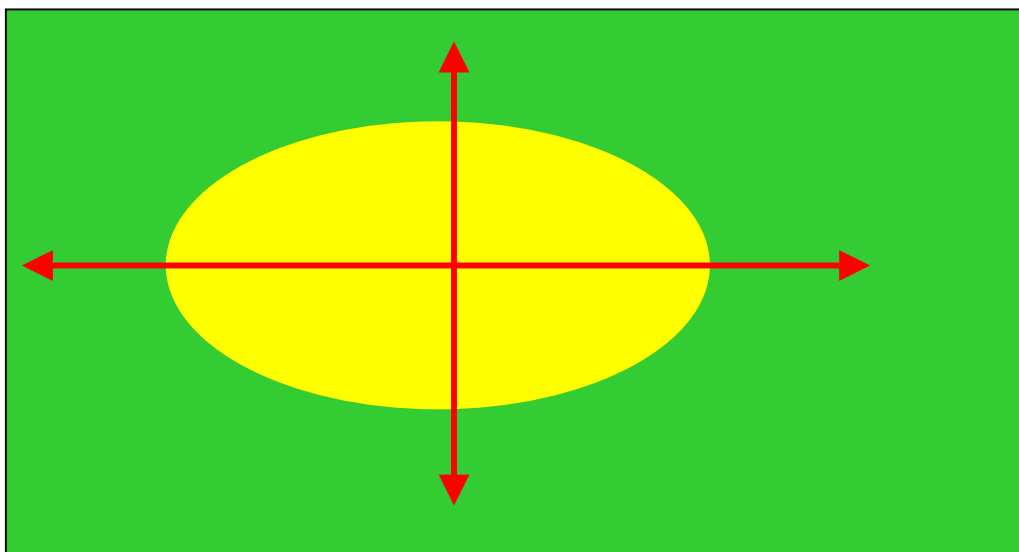


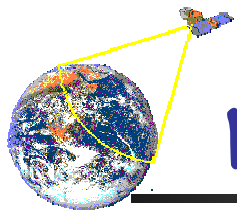
图像分割:阈值分割法

■ 分割连通区域

➤ 适用场合: 印前等。

先左后右, 先上半部分、后下半部分

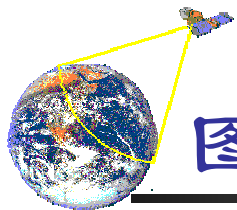




图像分割:阈值分割法

■ 基于多个变量的阈值

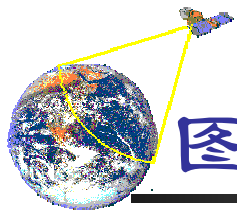
- 目的: 针对有多个分量的图像, 如**RGB**、**CMYK**、**HIS**、多光谱图像, 解决其目标与背景的分割问题
- 基本思想: 把前面的方法扩展到多维空间, 则寻找波谷的过程, 变为寻找点簇的过程。
- 算法实现:
 - ✓ 各维分量波谷之间进行逻辑与运算, 从波谷重合的点, 得到实际的阈值**T**。



图像分割:面向区域的分割

■ 图像分割—面向区域的分割

- 基本概念
- 通过像素集合的区域增长
- 区域分裂与合并



图像分割:面向区域的分割

■ 面向区域的分割

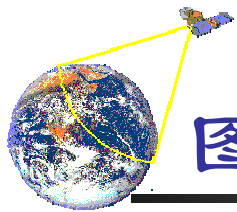
➤ 基本概念

✓ 目标: 将区域**R**划分为若干个子区域**R₁, R₂, ..., R_n**, 这些子区域满足5个条件:

1) 完备性: $\bigcup_{i=1}^n R_i = R$

2) 连通性: 每个**R_i**都是一个连通区域

3) 独立性: 对于任意 $i \neq j$, $R_i \cap R_j = \Phi$



图像分割:面向区域的分割

■ 面向区域的分割

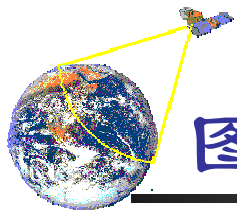
➤ 基本概念

4) 单一性: 每个区域内的灰度级相等,

$$P(R_i) = \text{TRUE}, i = 1, 2, \dots, n$$

5) 互斥性: 任两个区域的灰度级不等,

$$P(R_i \cup R_j) = \text{FALSE}, i \neq j$$



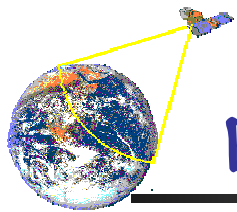
图像分割:面向区域的分割

■ 面向区域的分割

➤ 通过像素集合的区域增长

✓ 算法实现:

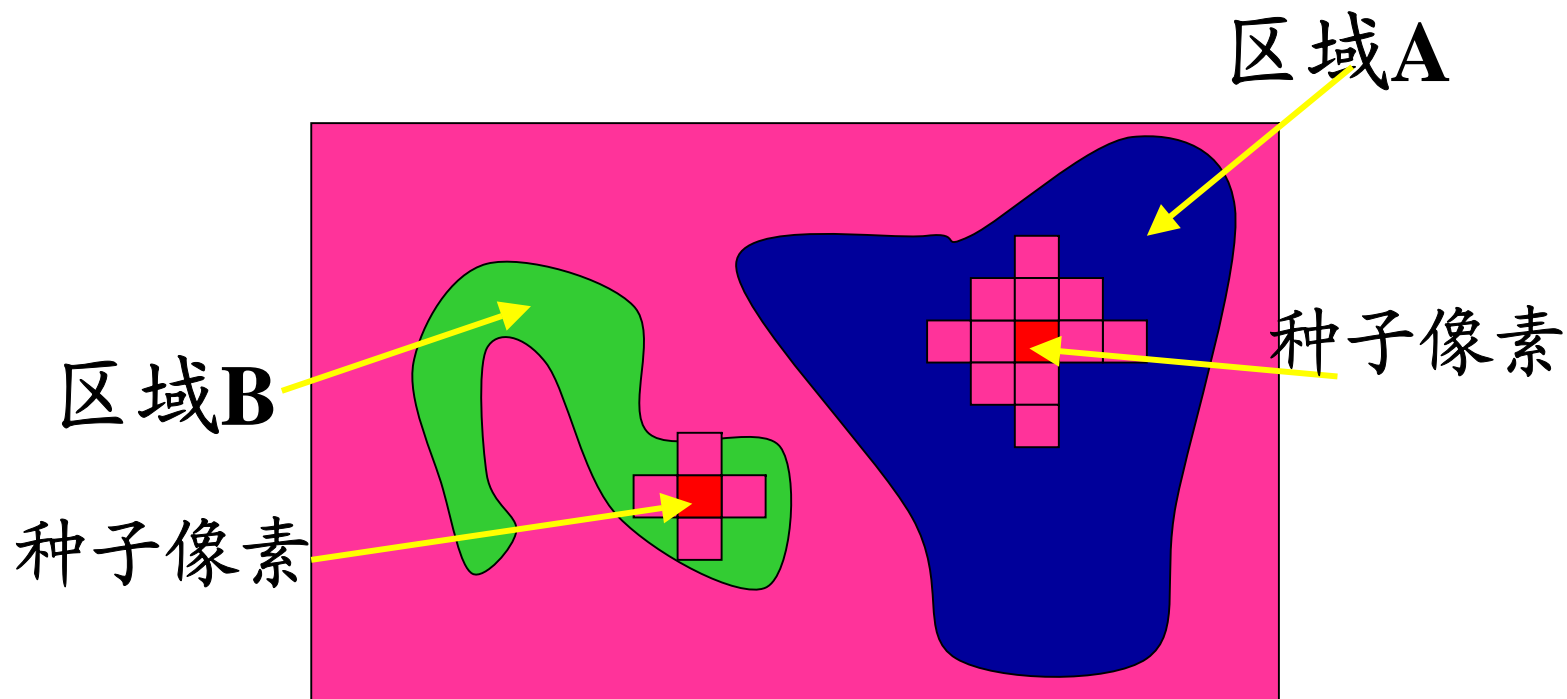
- 1) 根据图像的不同应用选择一个或一组种子, 它或者是最亮或最暗的点, 或者是位于点簇中心的点。
- 2) 选择一个描述符 (条件)
- 3) 从该种子开始向外扩张, 首先把种子像素加入集合, 然后不断将与集合中各个像素连通、且满足描述符的像素加入集合
- 4) 上一过程进行到不再有满足条件的新结点加入集合为止。



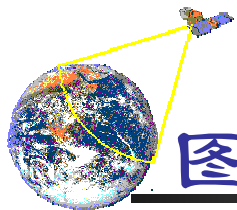
图像分割:面向区域的分割

➤ 通过像素集合的区域增长

✓ 算法实现:



➤ 实例



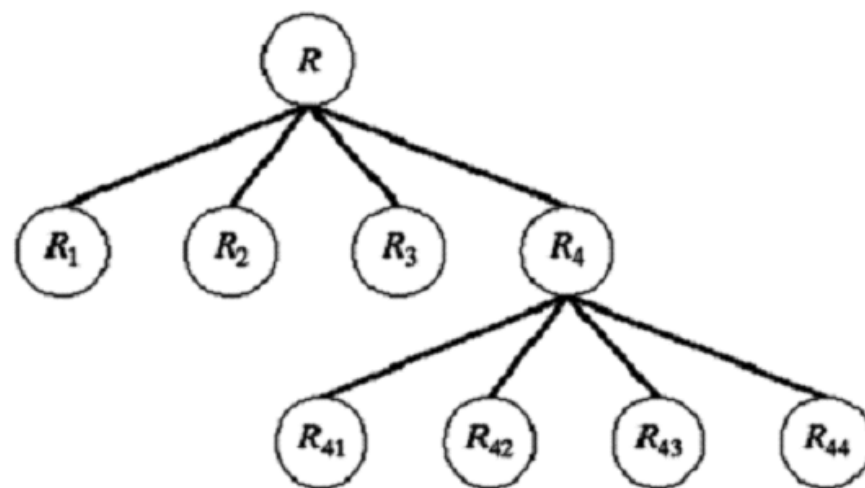
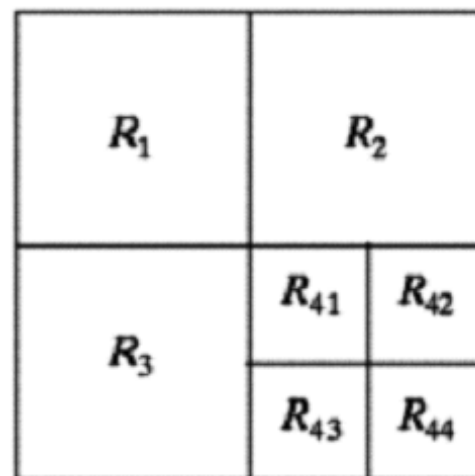
图像分割:面向区域的分割

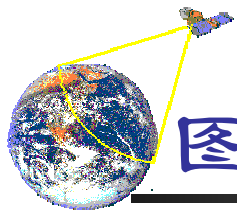
■ 面向区域的分割

➤ 区域分裂与合并

✓ 算法实现:

- 1) 对于图像中灰度级不同的区域, 均分为四个子区域。
- 2) 如果相邻的子区域所有像素的灰度级相同, 则将其合并。
- 3) 反复进行上两步操作, 直至不再有新的分裂与合并为止。





图像分割:面向区域的分割

➤ 区域分裂与合并

✓ 算法实现: 实际应用中还可作以下修改:

$P(R_i)$ 的定义为:

1) 区域内多于80%的像素满足不等式

$$|z_j - m_i| \leq 2 \sigma_i,$$

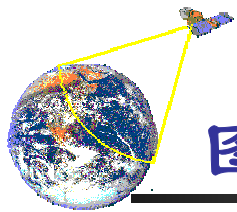
其中: z_j 是区域 R_i 中第 j 个点的灰度级,

m_i 是该区域的平均灰度级,

σ_i 是区域内灰度级的标准方差。

2) 当 $P(R_i)=\text{TRUE}$ 时, 将区域内所有像素的灰度级置为 m_i 。

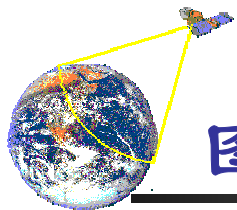
实例



图像分割:数学形态学图像处理

■ 图像分割—数学形态学图像处理

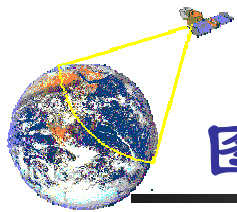
- 基本概念
- 腐蚀与膨胀
- 开—闭运算
- 变体
- 形态学图像处理应用
- 灰度图像的形态学处理



图像分割:数学形态学图像处理

■ 数学形态学图像处理

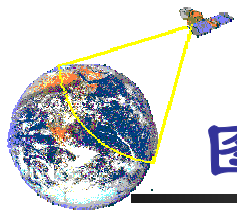
- **形态学**(Morphology)是生物学的一个分支,常用来处理动物和植物的形状与结构
- **数学形态学**(Mathematical Morphology)是分析几何形状和结构的数学方法,它建立在集合代数的基础上,是用集合论方法定量描述集合结构的学科。1985年之后,数学形态学逐渐成为分析图像几何特征的工具
- 数学形态学是一门建立在严格数学理论基础上的科学,是一门综合了多学科交叉知识的交叉科学。其理论基础颇为艰深,但基本观念却较为简单
- **数学形态学包括一组基本的形态学运算符**:腐蚀、膨胀、开、闭等。运用这些算子及其组合来进行图像形状和结构的分析及处理
- **形态学的理论基础是集合论**。在图像处理中形态学的集合代表着黑白和灰度图像的形状,如黑白图像中的所以黑像素点组成了此图像的完全描述。通常我们选择图像中感兴趣的目标图像区域像素集合来进行形态学变换



图像分割:数学形态学图像处理

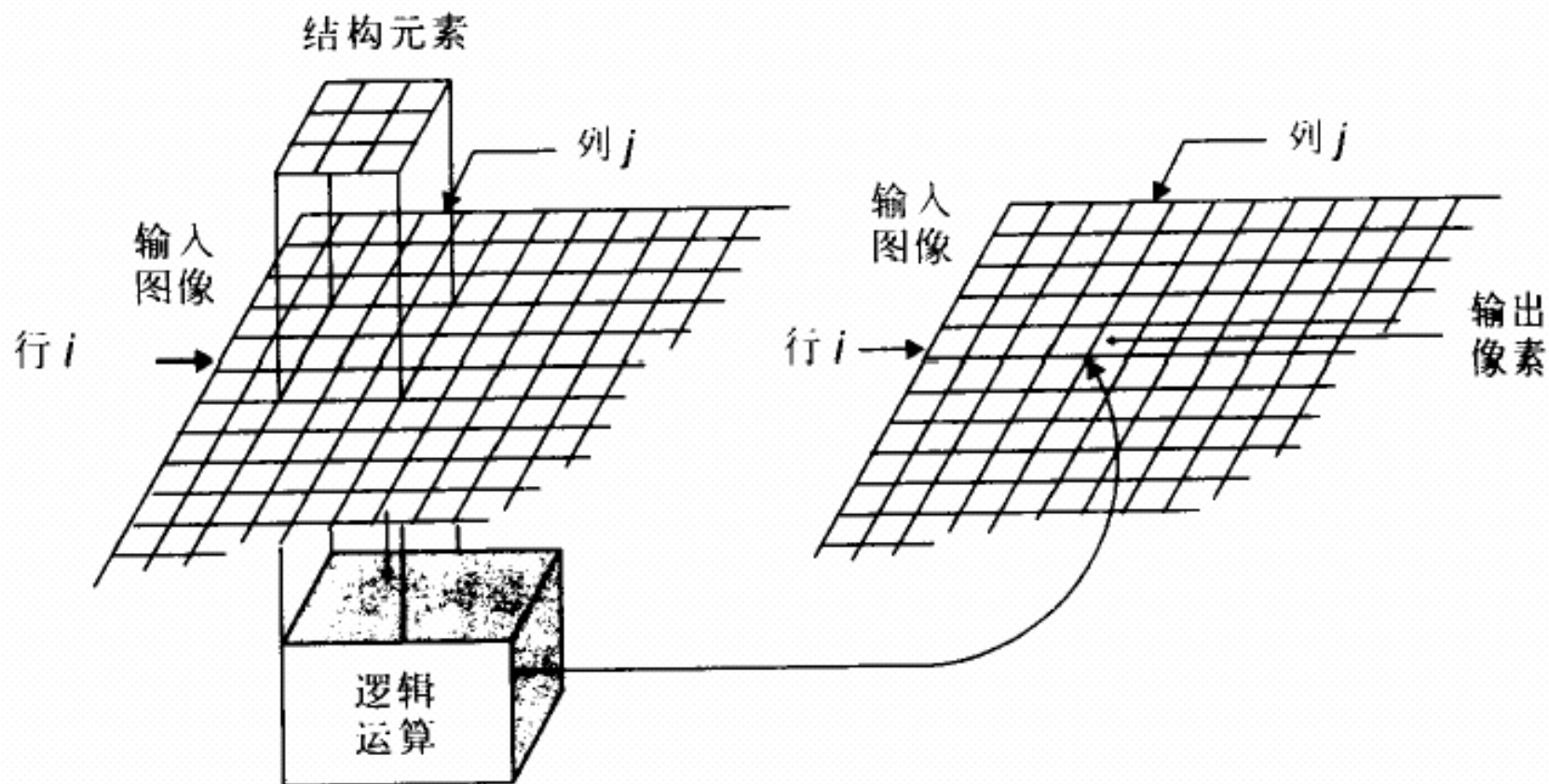
■ 基本概念

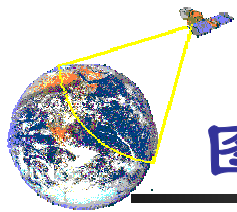
- 形态学: 从图像出发研究目标的结构和拓扑
- 结构元素: 任意大小, 包含任意0、1组合的一个区域。在结构元素上可任意指定一点作为原点
- 形态学图像处理: 结构元素与图像进行逻辑运算, 产生新的图像的图像处理方法
- 结构元素与图像的作用: 类似于卷积, 但用逻辑运算代替乘加运算。逻辑运算的结果保存在输出图像对应点的位置
- 形态学处理的效果取决于: 结构单元的大小、内容、逻辑运算的性质



图像分割:数学形态学图像处理

■ 基本概念



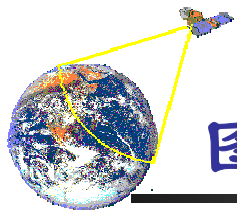


图像分割:数学形态学图像处理

■ 基本概念

➤ 集合概念上的二值图像:

- ✓ 二值图像**B**和结构元素**S**是定义在笛卡儿网格上的集合，网格中值为1的点是集合的元素
- ✓ 当结构元素的原点移到点 (x,y) 时，记为 S_{xy}
- ✓ 为简单起见，结构元素为 3×3 ，且全都为1，在这种限制下，决定输出结果的是逻辑运算



图像分割:数学形态学图像处理

■ 基本概念

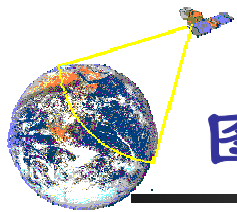
结构元素S

1	1	1
1	1	1
1	1	1

结构元素 S_{xy}

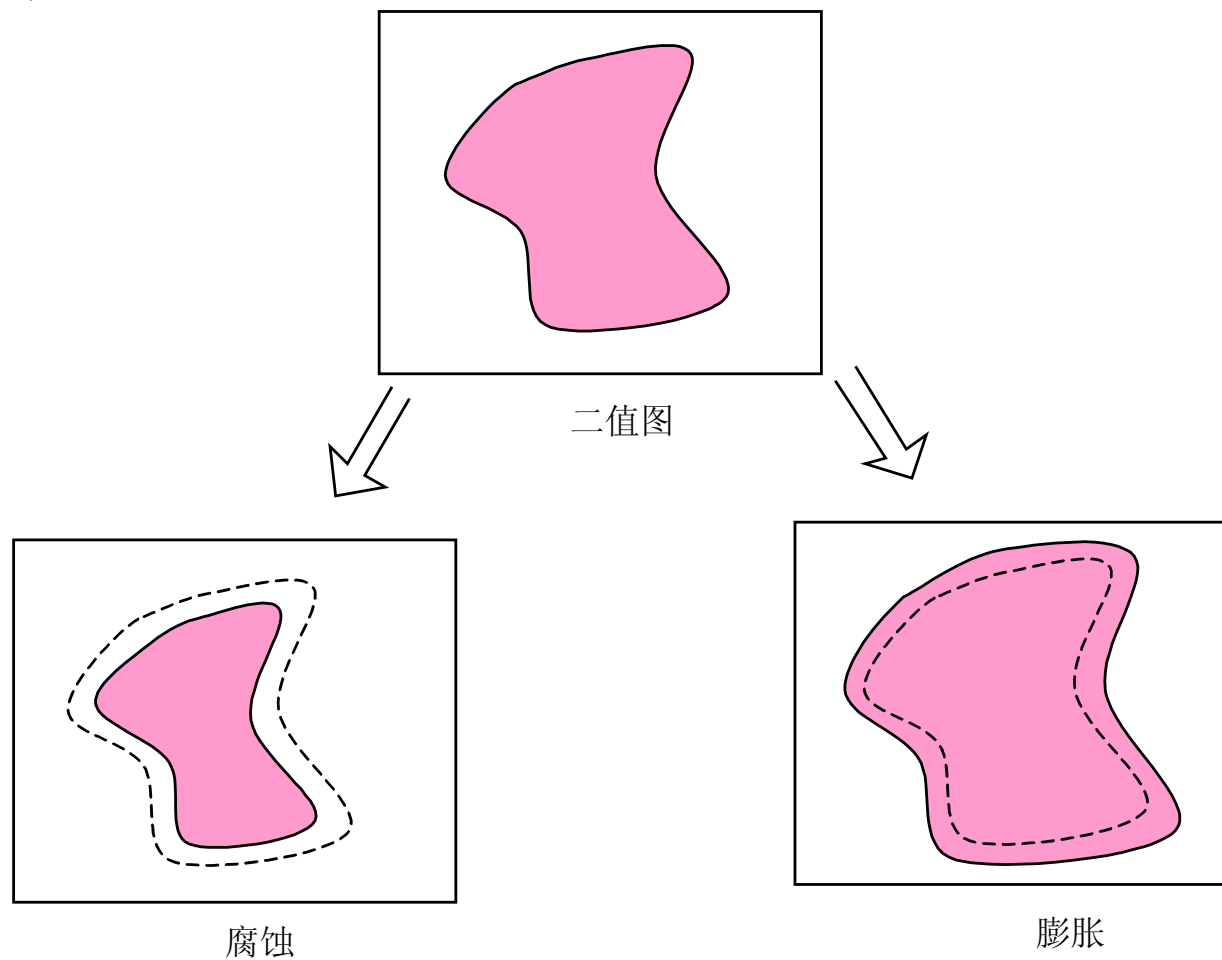
1	1	1				
1	1	1	1	0	0	1
1	1	1	1	1	1	1
	0	0	1	1	1	0

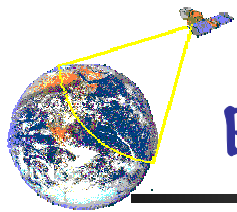
图像B



图像分割:数学形态学图像处理

■ 腐蚀与膨胀





图像分割:数学形态学图像处理

■ 腐蚀与膨胀

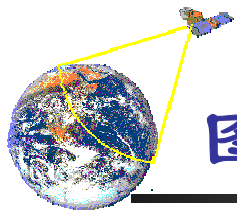
➤ 腐蚀(Erosion)

✓ 定义: $E = B \ominus S = \{x, y \mid S_{xy} \subseteq B\}$

✓ 意义: 当结构元素 S 的原点移动到 (x, y) 位置, 如果 S 完全包含在 X 中, 则新图象上该点为1, 否则为0

✓ 算法:

- ❖ 用结构元素, 扫描图像的每一个像素
- ❖ 用结构元素与其覆盖的二值图像做“与”操作
- ❖ 如果都为1, 结果图像的该像素为1。否则为0。

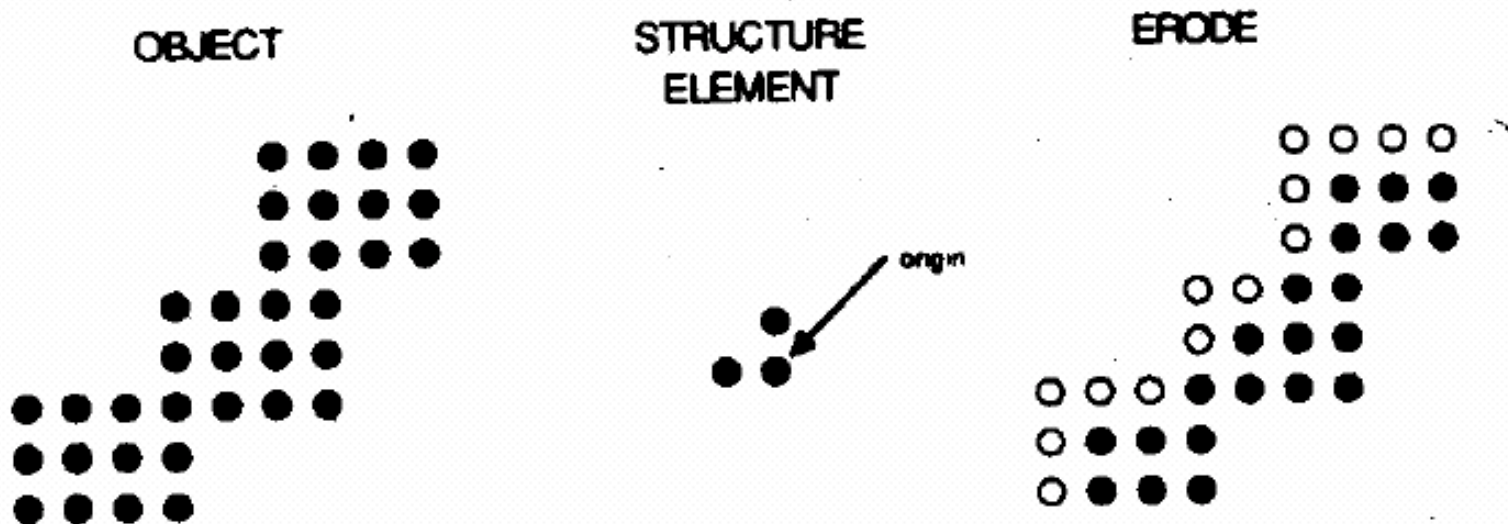


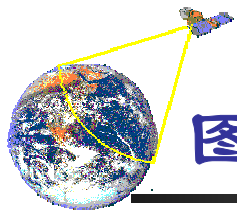
图像分割:数学形态学图像处理

■ 腐蚀与膨胀

➤ 腐蚀的作用

- ✓ 用 3×3 的结构元素时, 使物体的边界沿周边减少一个像素
- ✓ 去掉小于结构元素的物体, 选择不同大小的结构元素, 可以去掉大小不同的物体
- ✓ 如果两物体之间有细小的连通, 当结构元素足够大时, 可以将物体分开
- ✓ 不同的结构元素, 可导致不同的结果



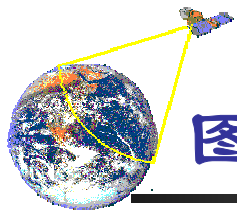


图像分割:数学形态学图像处理

■ 腐蚀与膨胀

➤ 膨胀(Dilation)

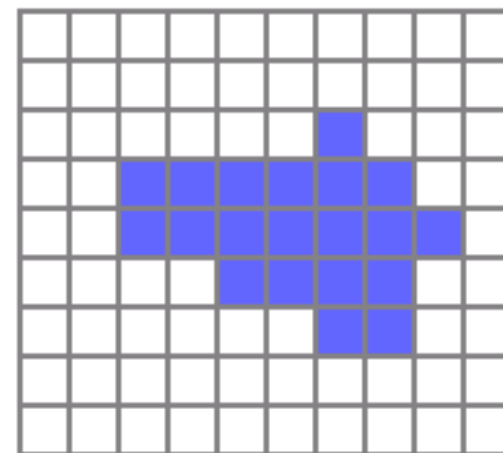
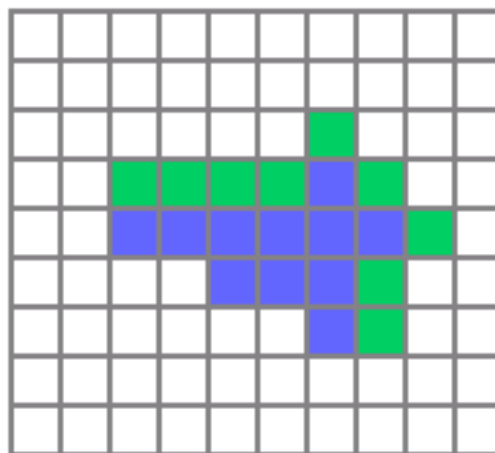
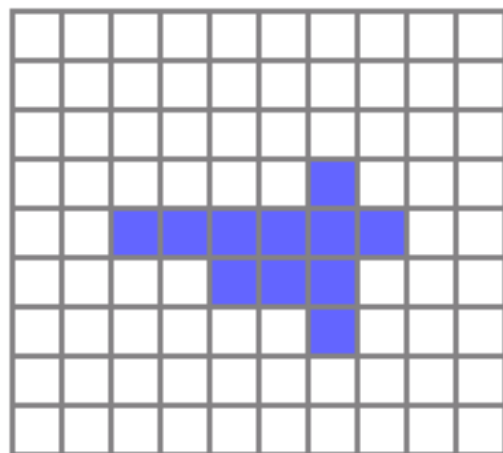
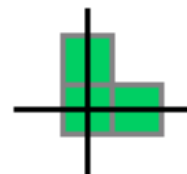
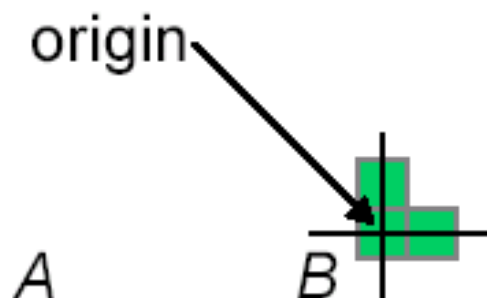
- ✓ 定义: $E = B \oplus S = \{x, y \mid S_{xy} \cap B \neq \Phi\}$
- ✓ 意义: 当结构元素 S 的原点移动到 (x, y) 位置, 如果 S 与物体 X 有任何一点同时为1, 则新图象上相应点为1; 如果 S 与 X 完全没有相交, 新图象上点为0
- ✓ 算法:
 - ❖ 用结构元素, 扫描图像的每一个像素
 - ❖ 用结构元素与其覆盖的二值图像做“与”操作
 - ❖ 如果都为0, 结果图像的该像素为0。否则为1

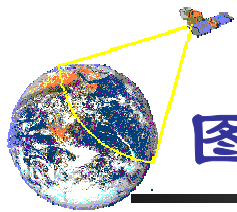


图像分割:数学形态学图像处理

- Dilation of set A by structuring element B

$$\delta_B A = A \oplus B = \bigcup_{\alpha \in A} B_\alpha$$



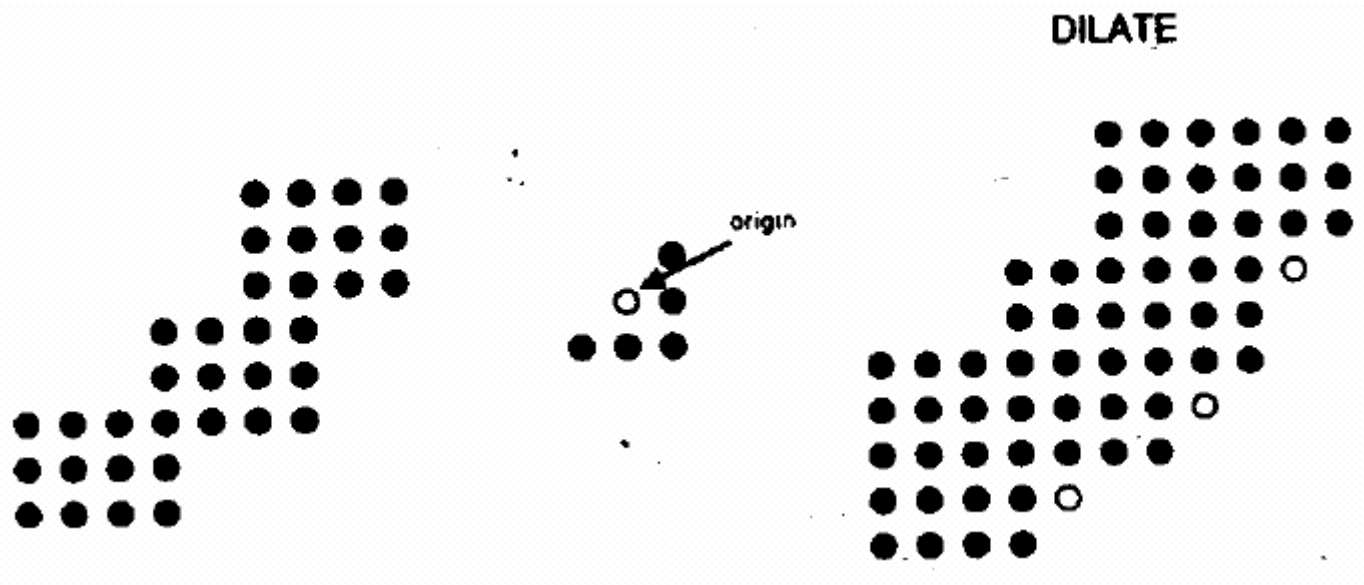


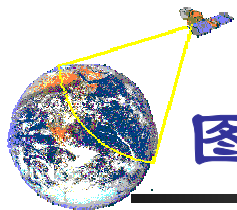
图像分割:数学形态学图像处理

■ 腐蚀与膨胀

➤ 膨胀的作用

- ✓ 用 3×3 的结构元素时, 使物体的边界沿周边增加一个像素
- ✓ 把图象周围的背景点合并到物体中。如果两个物体距离比较近,
通过膨胀可能连通在一起
- ✓ 对于填补图象分割后物体中的空洞十分有用

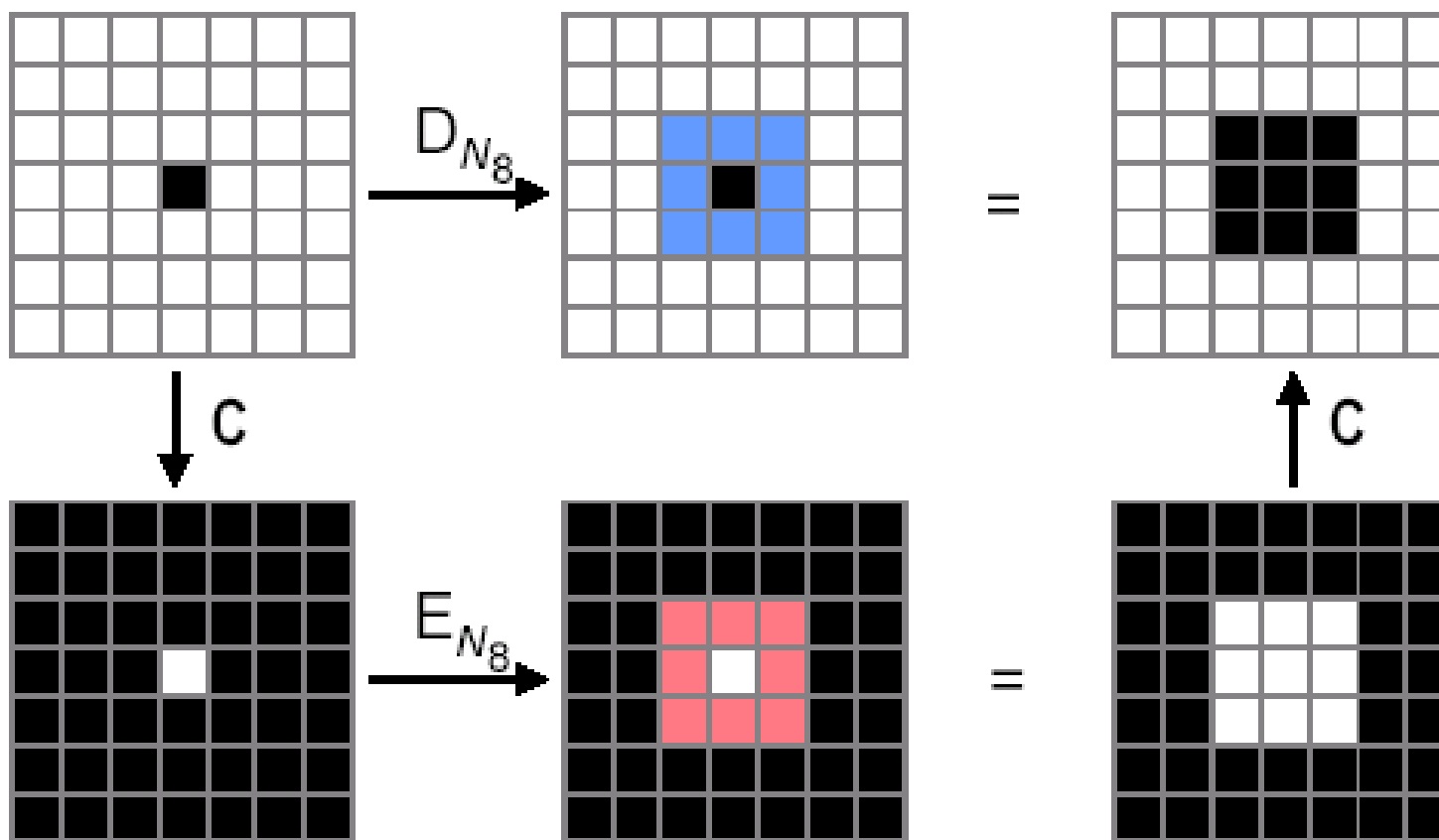


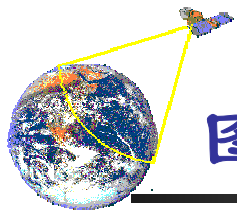


图像分割:数学形态学图像处理

■ 腐蚀与膨胀

➤ 腐蚀与膨胀的对偶性





图像分割:数学形态学图像处理

■ 开-闭运算

➤ 开运算(Opening)

✓ 思路: 先腐蚀, 再膨胀

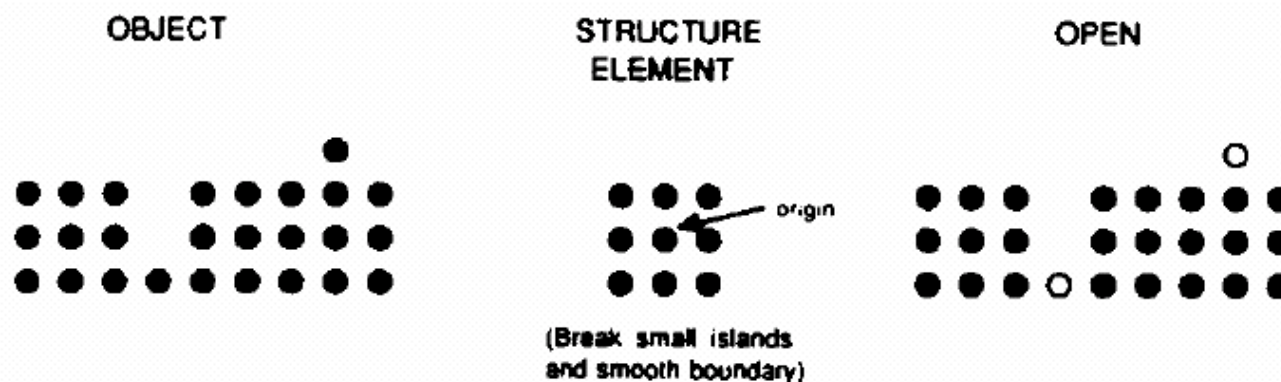
✓ 定义: $B \circ S = (B \ominus S) \oplus S$

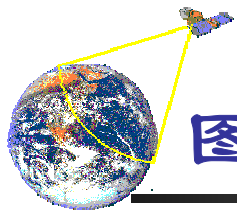
✓ 结果:

❖ 消除细小对象

❖ 在细小粘连处分离对象

❖ 在不改变形状的前提下, 平滑对象的边缘





图像分割:数学形态学图像处理

■ 开-闭运算

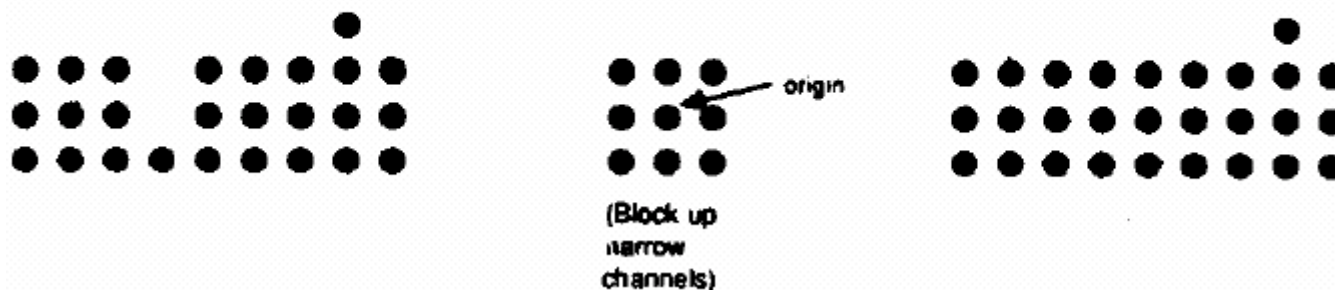
➤ 闭运算 (Closing)

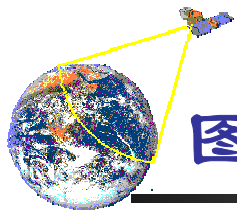
✓ 思路: 先膨胀、再腐蚀

✓ 定义: $B \bullet S = (B \oplus S) \ominus S$

✓ 结果:

- ❖ 填充对象内细小空洞。
- ❖ 连接邻近对象
- ❖ 在不明显改变面积前提下, 平滑对象的边缘

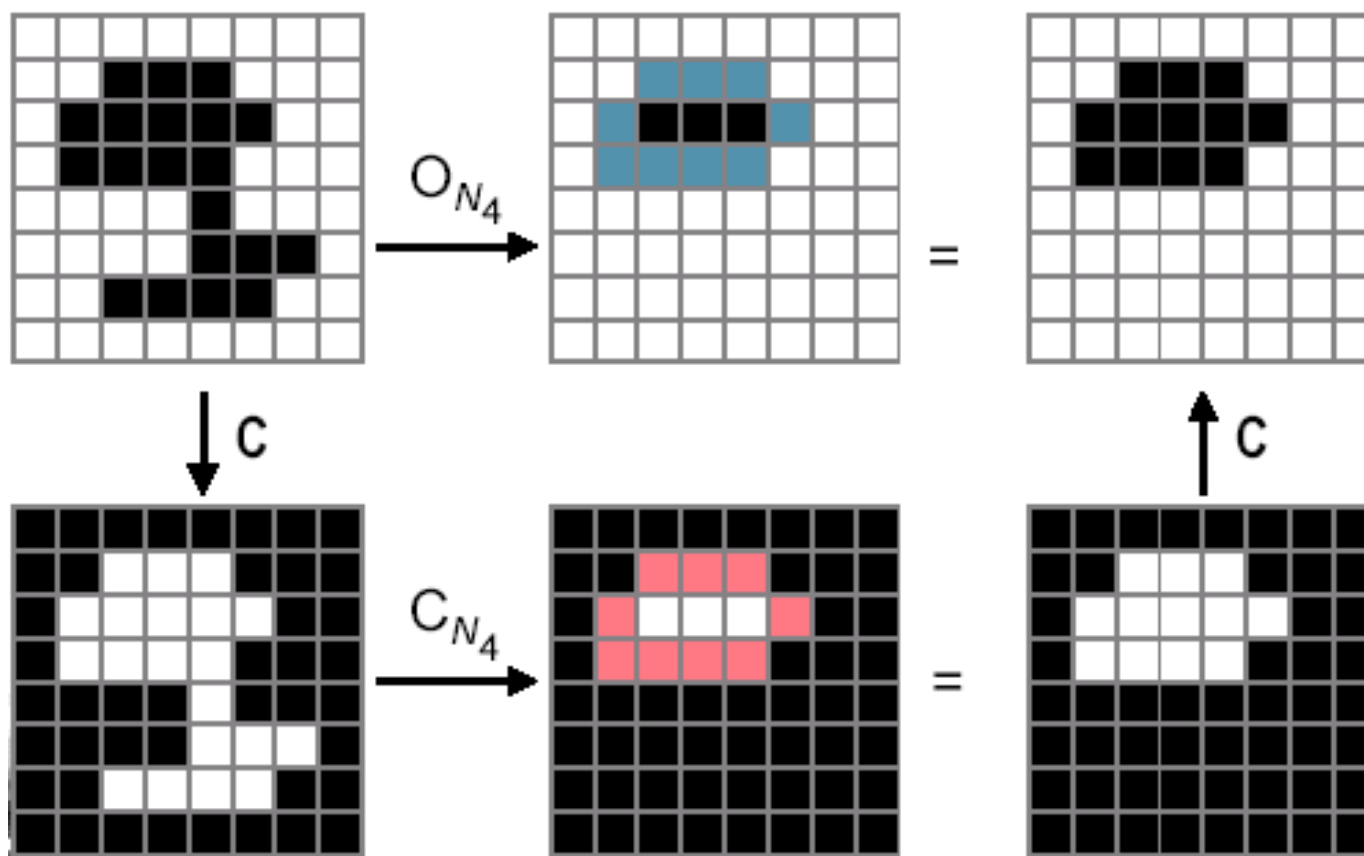


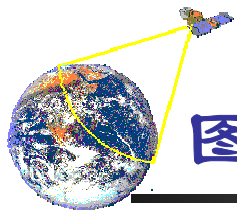


图像分割:数学形态学图像处理

■ 开-闭运算

➤ 开运算与闭运算的对偶性



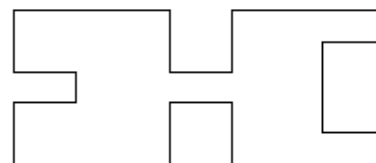


图像分割:数学形态学图像处理

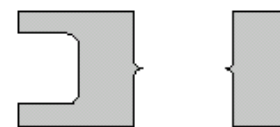
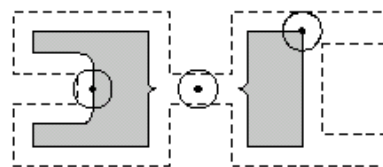
a
b c
d e
f g
h i

FIGURE 9.10

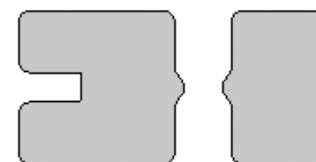
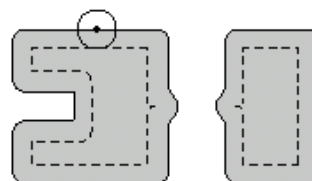
Morphological opening and closing. The structuring element is the small circle shown in various positions in (b). The dark dot is the center of the structuring element.



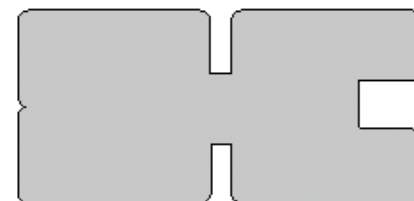
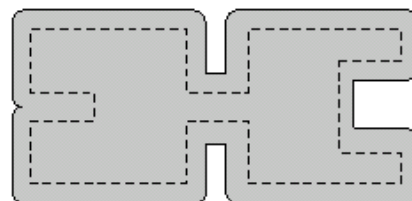
A



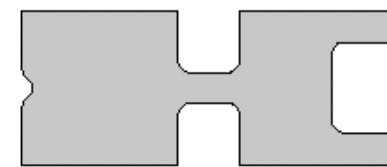
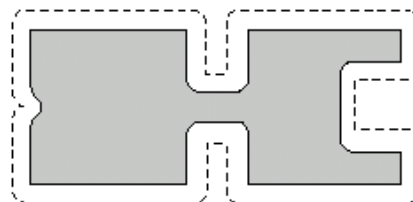
$A \ominus B$



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

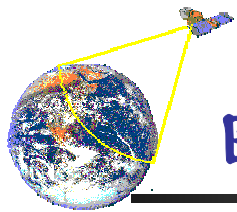


$A \oplus B$



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

■ 综合实例

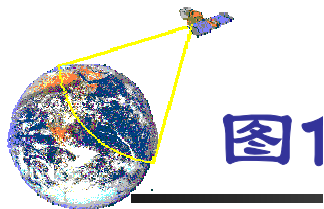


图像分割:数学形态学图像处理

■ 变体

➤ 细化

- ✓ 结果:在不破坏连通性的前提下, 细化图像。
- ✓ 算法实现:
 - 1) 做腐蚀操作, 但不立刻删除像素, 只打标记
 - 2) 将不破坏连通性的标记点删掉。
 - 3) 重复执行, 将产生细化结果



■ 变体

➤ 细化

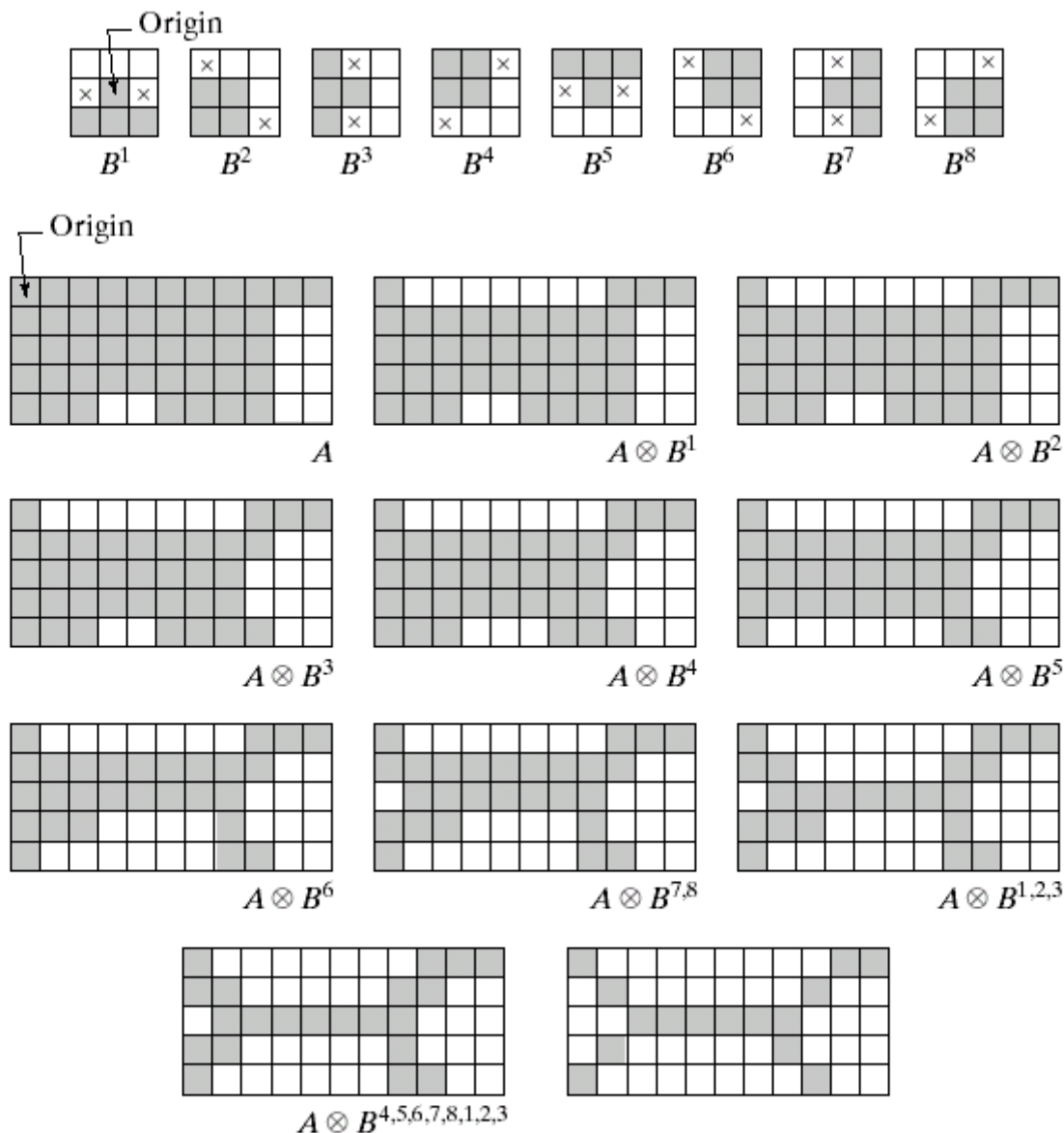
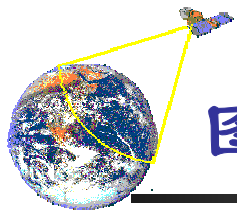


FIGURE 9.21 (a) Sequence of rotated structuring elements used for thinning. (b) Set A . (c) Result of thinning with the first element. (d)–(i) Results of thinning with the next seven elements (there was no change between the seventh and eighth elements). (j) Result of using the first element again (there were no changes for the next two elements). (k) Result after convergence. (l) Conversion to m -connectivity.

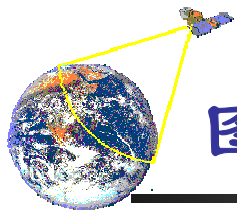


图像分割:数学形态学图像处理

■ 变体

➤ 粗化

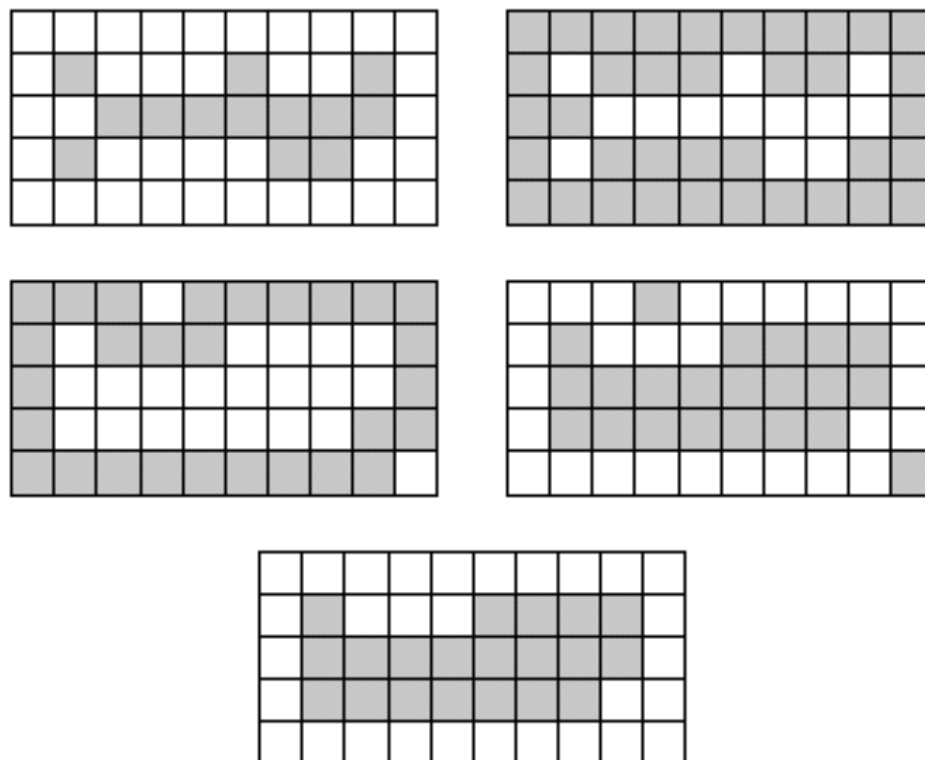
- ✓ 结果: 在不合并对象的前提下, 粗化图像。
- ✓ 算法实现:
 - 1) 做膨胀操作, 但不立刻添加像素, 只打标记
 - 2) 将不产生对象合并的标记点添加进来。
 - 3) 重复执行, 将产生粗化结果
- ✓ 另一方案: 将图像求反, 执行细化, 结果再求反



图像分割:数学形态学图像处理

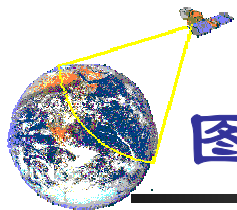
■ 变体

➤ 粗化



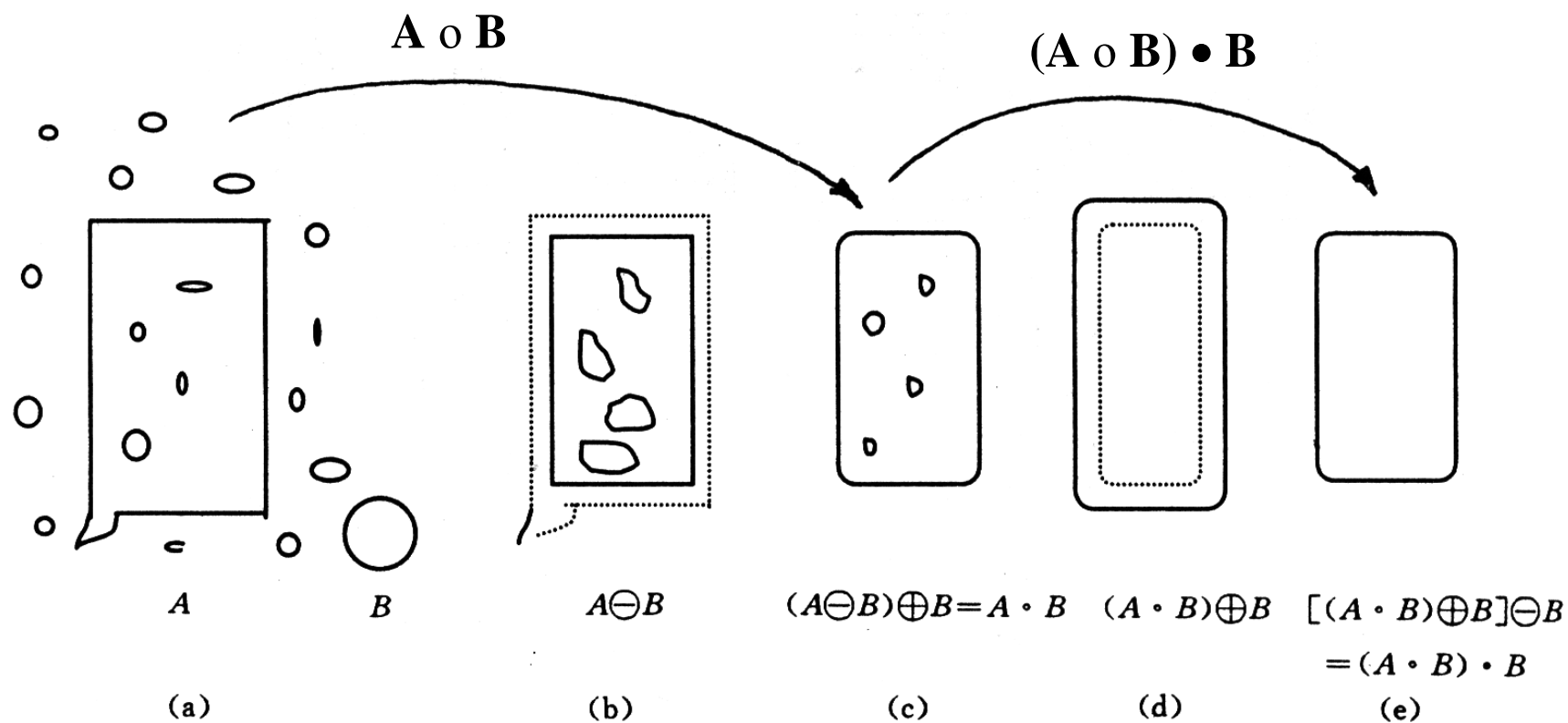
a	b
c	d
e	

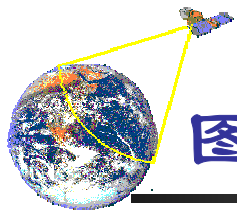
FIGURE 9.22 (a) Set A . (b) Complement of A . (c) Result of thinning the complement of A . (d) Thickened set obtained by complementing (c). (e) Final result, with no disconnected points.



图像分割:数学形态学图像处理

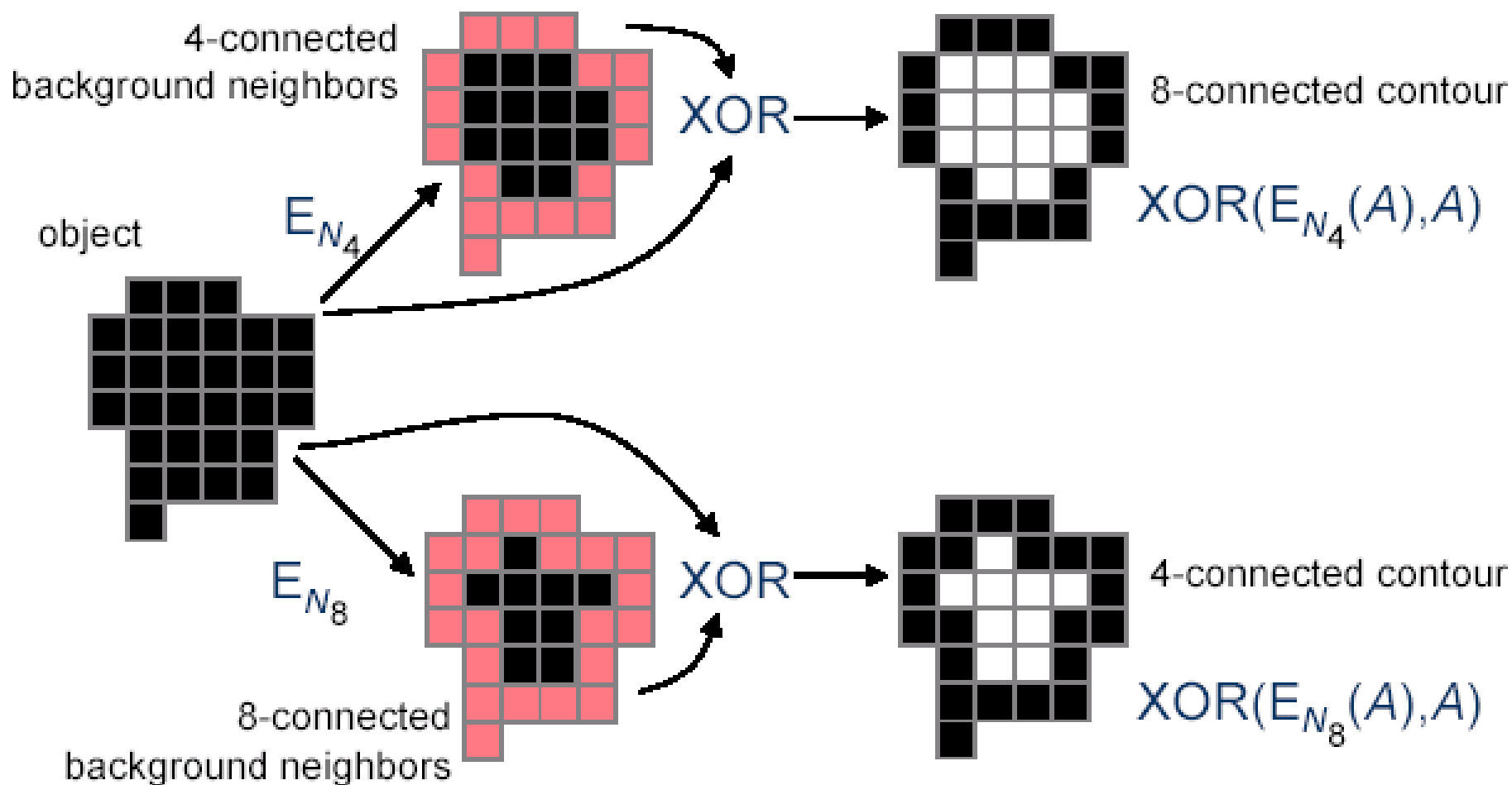
■ 形态学图像处理应用——形态学滤波

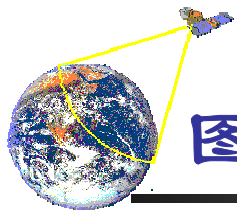




图像分割:数学形态学图像处理

■ 形态学图像处理应用——边界提取

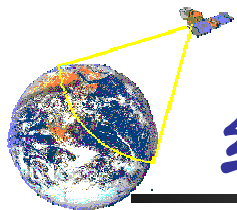




图像分割:数学形态学图像处理

■ 灰度图像的形态学处理

- 严格意义上的数学形态学仅处理二值图像
- 在二值图像形态学处理技术的基础上扩充形成灰度图像的形态学处理
- 类似二值图像的形态学处理，可定义灰度图像形态学处理的基本运算腐蚀、膨胀、开运算、闭运算等
- 灰度图像的形态学处理具有特征：
 - ✓ 如果所有结构元素为正，膨胀使图像变亮；腐蚀使图像变暗
 - ✓ 通过选择结构单元的值与形状，膨胀使黑色细节减少或去除，腐蚀使亮细节减弱或去除
 - ✓ 开运算常用于去除小的亮点，闭运算去除小的暗点
- 实例



结束

第八章1 结束