



## 第七章 图像分割

---



### 本章内容

---

- 一：概述
- 二：边缘检测
- 三：门限化分割
- 四：基于区域的分割



## 本章内容

### 一：概述

### 二：边缘检测

### 三：门限化分割

### 四：基于区域的分割

#### 1. 图像分割概述

#### 2. 图像分割方法分类

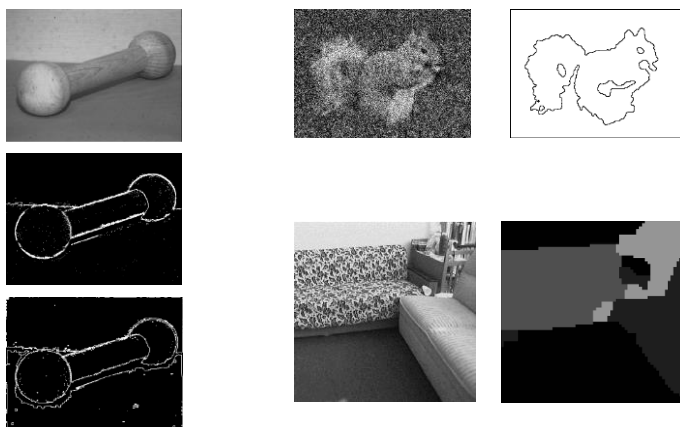
#### 3. 图像分割应用



## 1. 概述

- 在对图像的研究和应用中，人们往往仅对图像中的某些部分感兴趣，这些部分被称为**目标**或**前景**。
- 为了辨别和分析目标，需要将有关区域分离提取出来，在此基础上对目标进一步利用，如进行特征提取和测量。
- **图像分割**就是指把图像分成各具特性的区域并提取出感兴趣目标的技术和过程。
- 特征可以是**灰度**、**颜色**、**纹理**等，目标可以对应单个区域，也可以对应多个区域。

## 1. 概述（续1）



## 1. 概述（续2）

- 图像分割是计算视觉和图像理解中的最基本问题之一。它也是该领域国际学术界公认的将会**长期存在的、最困难的问题之一**
- 图像分割之所以困难的一个重要原因是其并不完全属于图像特征提取问题，它还涉及到各种图像特征的**知觉组织**
- 从一般意义上来说，只有对图像内容的**彻底理解**，才能产生完美的分割



## 1. 概述（续3）

- 在图像分割领域还没有出现对任意图像都可以分割的算法，需要根据问题的不同设计和采用不同的算法，还可能要考虑时空复杂度的可接受性。



## 2. 图像分割方法分类

- 图像分割的依据：
  - ✓ 同一区域内像素的相似性
  - ✓ 不同区域的边界上像素的不连续性
- 根据灰度的不连续性和相似性，分成两类：
  - ✓ 边缘检测法：利用区域间之灰度不连续性，确定区域的边界或边缘的位置。
  - ✓ 区域生成法：利用区域内灰度的相似性，将象素分成若干相似的区域。

## 2. 图像分割方法分类（续1）



边缘检测法



区域生成法

## 3. 图像分割应用

- 图像分割的应用：
  - ✓ 机器阅读理解
  - ✓ **OCR**录入
  - ✓ 遥感图像自动识别
  - ✓ 在线产品检测
  - ✓ 医学图像样本统计
  - ✓ 医学图像测量
  - ✓ 图像编码
  - ✓ 图像配准的预处理

# 本章内容

一：概述

二：边缘检测

三：门限化分割

四：基于区域的分割

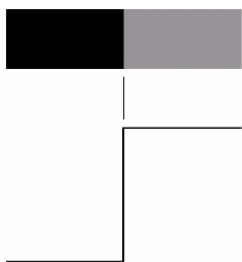
1. 边缘检测概述

2. 微分算子法

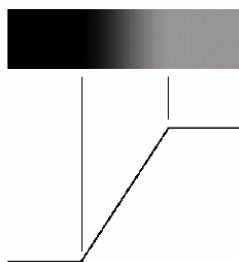
3. 边缘连接

## 1. 边缘检测概述

- 定义：边缘定义为图像局部特性的不连续性（相邻区域之交界）。
- 种类：大致分为阶跃状和屋顶状



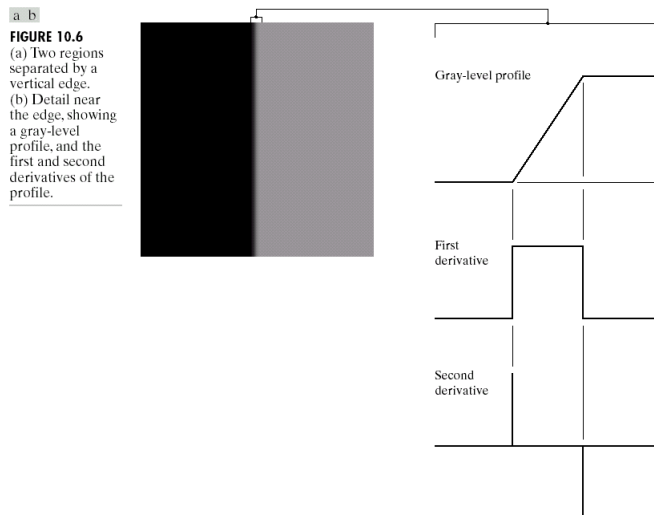
阶跃状边缘



屋顶状边缘

## 1. 边缘检测概述（续1）

- 边缘和导数的关系：



## 1. 边缘检测概述（续2）

- 边缘特点

- ✓ 局部特性不连续性；
- ✓ 边缘位置的微分特性；
- ✓ 方向性（沿边缘方向灰度缓（不）变，垂直方向突变）。

- 边缘检测用途

- ✓ 将图像中各不同区域的边缘（边界）检测出来，以达到图像分割的目的。

## 2. 微分算子法

- 1) 梯度算子 (一阶导数)
- 2) **Laplacian**算子 (二阶导数)
- 3) 点、线检测

## 2. 微分算子法 (续1)

- 1) 梯度算子 (一阶导数)

$$\nabla \vec{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \end{bmatrix}^T$$

$$\nabla f = \text{mag}(\nabla \vec{f}) = \left[ G_x^2 + G_y^2 \right]^{1/2} = \left[ \left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2 \right]^{1/2}$$

$$\alpha(x, y) = \arctan \frac{G_y}{G_x}$$



## 2. 微分算子法（续2）

### ■ 常用的梯度算子模板

-1	0	0	-1
0	1	1	0

Roberts

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

Prewitt

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

Prewitt

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

Sobel

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

Sobel

## 2. 微分算子法（续3）

### ■ 结论：

- ✓ **Sobel**算子是三种算子中效果较好的算子。
- ✓ 当边缘过渡比较尖锐且图像中噪声比较小时，梯度算子效果较好。



## 2. 微分算子法（续4）

### 2) Laplacian算子 (二阶导数)

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



## 2. 微分算子法（续5）

■ Laplacian算子也可以借助模板实现：

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

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

## 2. 微分算子法（续6）

### ■ Laplacian算子的缺点：

- ✓ 对噪声敏感
- ✓ 不能提供方向信息
- ✓ 会产生双边缘

因此，**Laplacian**算子一般不直接用于边缘检测

## 2. 微分算子法（续7）

### 3) 点、线检测

#### ● 点检测

使用如图所示的模板，如果  $|R| \geq T$ ，则在模板中心位置检测到一个点

其中，**T**是阈值，**R**是模板计算值

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

- ✓ 基本思想：如果一个孤立点与它周围的点不同，则可以使用上述模板进行检测。
- ✓ 注意：如果模板响应为0，则表示在灰度级为常数的区域

## 2. 微分算子法（续8）

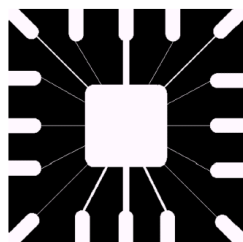
### • 线检测

✓ 4个线检测模板

-1	-1	-1	-1	-1	2	-1	2	-1	2	-1	-1
2	2	2	-1	2	-1	-1	2	-1	-1	2	-1
-1	-1	-1	2	-1	-1	-1	2	-1	-1	-1	2
Horizontal			+45°			Vertical			-45°		

## 2. 微分算子法（续9）

二值电路接线模板，  
寻找方向为-45度一个  
像素宽度的线条



**FIGURE 10.4**  
Illustration of line  
detection.  
(a) Binary wire-  
bond mask.  
(b) Absolute  
value of result  
after processing  
with -45° line  
detector.  
(c) Result of  
thresholding  
image (b).

使用-45度模  
板得到结果  
的绝对值



对图b门限化  
后的结果

### 3. 边缘连接

- 为什么需要边缘连接？
  - ✓ 由于噪声、照明等产生边缘间断，使得一组像素难以完整形成边缘
  - ✓ 因此，在边缘检测算法后，使用连接过程将间断的边缘像素组合成完整边缘

### 3. 边缘连接（续1）

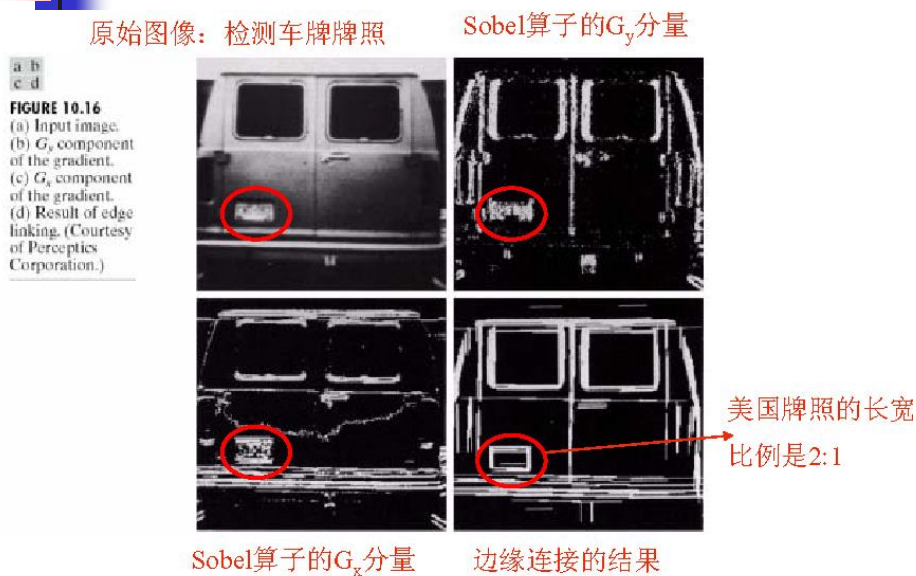
- 如何实现边缘连接？

梯度相似法

- 可连接的两像素点 $(x,y)$ 和 $(s,t)$ ，其梯度（幅度和方向）具有相似性，即：

$$\begin{cases} |\nabla f(x,y) - \nabla f(s,t)| \leq T \\ |\nabla \alpha(x,y) - \nabla \alpha(s,t)| \leq A \end{cases}$$

### 3. 边缘连接（续2）



### 本章内容

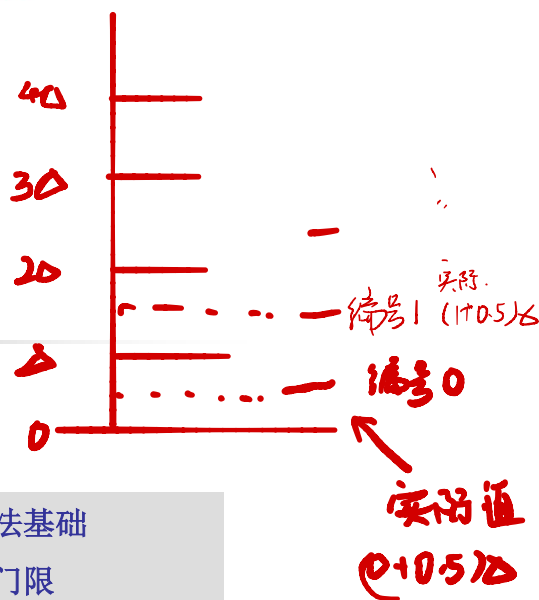
一：概述

二：边缘检测

三：门限化分割

四：基于区域的分割

1. 门限法基础
2. 全局门限
3. 自适应门限

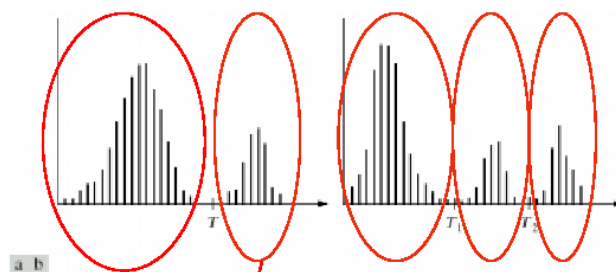


## 1. 门限法基础

■ 通过取灰度门限对图像像素进行分类，该方法基于：

- 1) 同一区域由灰度值相近的像素点组成；
- 2) 目标物和背景、不同目标物之间的灰度值有明显差异，可通过取门限区分。

## 1. 门限法基础（续1）



**FIGURE 10.26** (a) Gray-level histograms that can be partitioned by (a) a single threshold, and (b) multiple thresholds.

暗的背景:  $f(x,y) \leq T$

亮的对象:  $f(x,y) > T$

暗的背景:  $f(x,y) \leq T_1$

亮的一个对象:  $T_1 < f(x,y) \leq T_2$

亮的另一个对象:  $f(x,y) > T_2$

## 1. 门限法基础（续2）

- ✓ 阈值处理操作

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

$f(x, y)$  是点  $(x, y)$  的灰度级， $p(x, y)$  表示该点的局部性质，如以  $(x, y)$  为中心的邻域的平均灰度级

- ✓ 阈值处理后的图像  $g(x, y)$  定义为

$$g(x, y) = \begin{cases} 1 & f(x, y) > T \\ 0 & f(x, y) \leq T \end{cases}$$

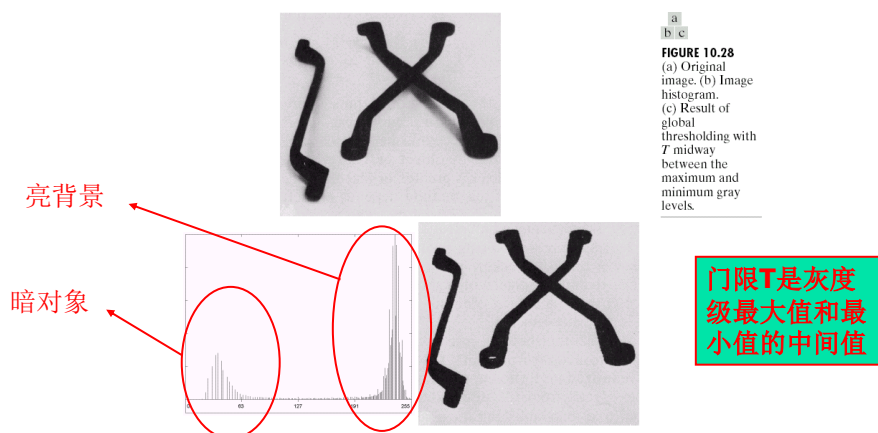
## 1. 门限法基础（续3）

- 门限法分割适合于物体与背景有较强对比度的图像。
- 门限法分割计算简单。



## 2. 全局门限

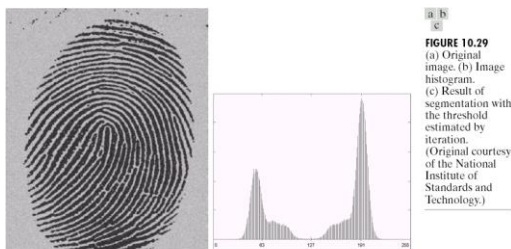
- 对图像中的每一个像素采用同一个门限进行判决。



## 2. 全局门限（续1）

- 计算基本全局阈值算法
  1. 选择一个 $T$ 的初始估计值
  2. 用 $T$ 分割图像，生成两组像素： $G_1$ 由所有灰度值大于 $T$ 的像素组成，而 $G_2$ 由所有灰度值小于或等于 $T$ 的像素组成
  3. 对区域 $G_1$ 和 $G_2$ 中的所有像素计算平均灰度值 $\mu_1$ 和 $\mu_2$
  4. 计算新的阈值  $T = \frac{1}{2}(\mu_1 + \mu_2)$
  5. 重复步骤2到4，直到逐次迭代所得的 $T$ 值之差小于事先定义的参数 $T_0$

## 2. 全局门限（续2）



迭代算法的应用  
在初始时使用平  
均灰度级

基本全局阈值算法  
处理的结果  
 $T_0=0$ ，3次迭代得  
到值为125.4  
最后确定 $T=125$

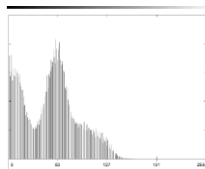
## 3. 自适应门限

- ✓ 单一全局阈值存在的问题
  - 不均匀亮度图像无法有效分割
- ✓ 方法
  - 将图像进一步细分为子图像，并对不同的子图像使用不同的阈值处理
  - 解决的关键问题：如何将图像进行细分和如何为得到的子图像估计阈值
  - 自适应阈值：取决于像素在子图像中的位置

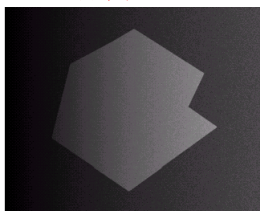


### 3. 自适应门限（续1）

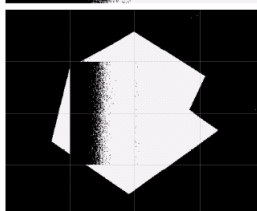
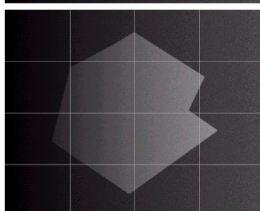
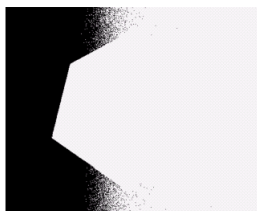
直方图



原图



一个全局门限处理后的结果，门限为直方图的波谷



分割为子图像：4等分后再4等分 自适应阈值处理结果



## 本章内容

一：概述

二：边缘检测

三：门限化分割

四：基于区域的分割

1. 基本公式

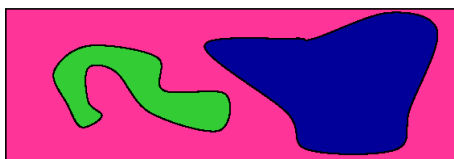
2. 区域生长

3. 区域分裂合并

## 1. 基本公式

- 令集合 $\mathbf{R}$ 代表整个图象区域，对 $\mathbf{R}$ 的分割应满足：

- (1) 完备性：  $\bigcup_{i=1}^n R_i = R$
- (2) 连通性： 每个  $R_i$  都是一个连通的区域
- (3) 独立性：  $\forall i \neq j, R_i \cap R_j = \emptyset$
- (4) 单一性：  $P(R_i) = True$
- (5) 互斥性：  $\forall i \neq j, P(R_i \cup R_j) = False$



## 2. 区域生长

- 基本思想：

将相似象素结合起来构成区域

- 基本步骤：

- (1) 选择区域的种子象素
- (2) 确定将相邻象素包括进来的准则
- (3) 制定生长停止的规则

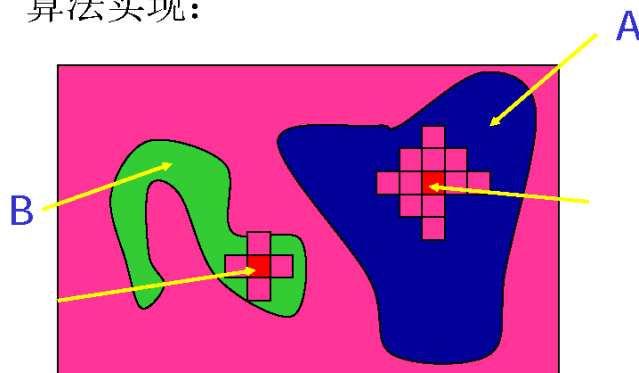
## 2. 区域生长

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结果集合，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

## 2. 区域生长（续1）

✓ 算法实现:



### 3. 区域分裂合并

#### ■ 主要步骤

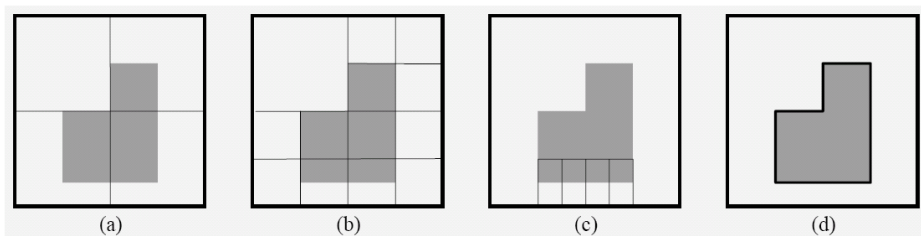
先把图象分成任意大小且不重叠的区域然后再（根据准则）合并或分裂这些区域（迭代进行直到实现分割）

示例（四叉树）：

分裂

分裂

合并



### 3. 区域分裂合并

- ◆ 令 $R$ 代表整个图象区域， $P$ 代表逻辑谓词
  - ◆ 把 $R$ 连续地分裂成越来越小的 $1/4$ 的正方形子区域 $R_i$ ，并且始终使 $P(R_i) = \text{TRUE}$
- (1) 对任一个区域 $R_i$ ，如果 $P(R_i) = \text{FALSE}$ ，就将其分裂成不重叠的四等分
  - (2) 对相邻的两个区域 $R_i$ 和 $R_j$ ，如果 $P(R_i \cup R_j) = \text{TRUE}$ ，就将它们合并起来
  - (3) 如果进一步的分裂或合并都不可能了，则结束

### 3. 区域分裂合并

算法实现:

- 1) 对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

### 3. 区域分裂合并（续1）

✓ 算法实现:

