

数字图像处理

(Digital Image Processing)

核心知识点精炼笔记 10

东南大学计算机科学与工程学院
09023419 严伟琛

授课教师：薛澄
参考教材：拉斐尔·冈萨雷斯 (Rafael C. Gonzalez) 第四版
2026 年 1 月 15 日

目录

10 图像分割 (Image Segmentation)	2
10.1 基础知识	2
10.1.1 分割的基本性质	2
10.1.2 基本定义	2
10.1.3 分割的分类	2
10.2 点、线和边缘检测	3
10.2.1 背景知识 (Background)	3
10.2.2 孤立点的检测 (Detection of Isolated Points)	3
10.2.3 线检测 (Line Detection)	3
10.2.4 边缘模型 (Edge Models)	4
10.2.5 基本边缘检测 (Basic Edge Detection)	4
10.2.6 高级边缘检测器 (Advanced Edge Detection)	4
10.3 阈值处理	5
10.3.1 基础知识	5
10.3.2 Otsu 算法 (大津法/最大类间方差法)	5
10.3.3 自适应阈值 (Variable Thresholding)	6
10.3.4 例题 23: 基于噪声统计特性的阈值选择	6
10.4 基于区域的分割	7
10.4.1 区域生长 (Region Growing)	7
10.4.2 区域分裂与聚合 (Region Splitting and Merging)	7
10.4.3 基于聚类的分割方法 (Clustering Methods)	8
10.5 形态学分水岭分割	8
10.5.1 基本概念	8
10.5.2 算法物理过程: 模拟浸没 (Immersion)	8
10.5.3 梯度图像的应用	8
10.5.4 过度分割问题与标记符限制 (Markers)	8
10.5.5 典型应用流程	9
10.6 第十章核心重点问题总结	9

10 图像分割 (Image Segmentation)

图像分割是将图像细分为构成它的子区域或物体的过程。分割的深度取决于要解决的具体问题。当感兴趣的物体或区域被检测出来时，分割过程即可停止。

10.1 基础知识

图像分割是图像处理中最重要的任务之一，也是最困难的任务之一。

10.1.1 分割的基本性质

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。

- **不连续性 (Discontinuity)**: 以灰度的突变为基础分割图像。主要应用是检测图像的边缘、点和线。
- **相似性 (Similarity)**: 根据一组预定义的准则（如灰度值范围、纹理、颜色等），将图像分割为相似的区域。阈值处理、区域生长、区域分裂与聚合都是这类方法的例子。

10.1.2 基本定义

令 R 表示整幅图像区域。我们可将图像分割看作是把 R 分成 n 个子区域 R_1, R_2, \dots, R_n 的过程，这些区域需满足以下条件：

- $\bigcup_{i=1}^n R_i = R$ (分割必须是完全的，覆盖整幅图像)。
- R_i 是一个连通区域， $i = 1, 2, \dots, n$ (每个区域内部是连通的)。
- $R_i \cap R_j = \emptyset$ 对于所有的 $i \neq j$ (区域之间不相交)。
- $P(R_i) = \text{TRUE}$ 对于 $i = 1, 2, \dots, n$ (每个区域内的像素满足某种相似性属性 P)。
- $P(R_i \cup R_j) = \text{FALSE}$ 对于任何相邻区域 R_i 和 R_j (相邻区域的属性必须不同)。

10.1.3 分割的分类

根据分割目标的类型，通常可以分为：

- **目标或前景 (Object/Foreground)**: 图像中的感兴趣区域。
- **背景 (Background)**: 图像中的其余部分。

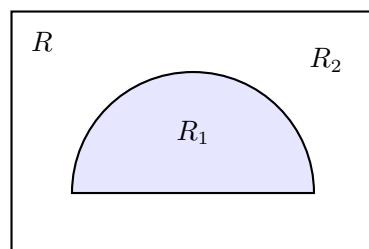


图 10.1 图像分割示意图：将 R 分割为目标 R_1 与背景 R_2

10.2 点、线和边缘检测

这一节主要研究如何利用灰度的突变（不连续性）来定位图像中的特征结构。

10.2.1 背景知识 (Background)

图像的不连续性通常通过一阶和二阶导数来检测：

- **一阶导数:** $\frac{\partial f}{\partial x} = f(x+1) - f(x)$ 。
 - 对灰度阶跃和斜坡敏感，在恒定区域为零。
- **二阶导数:** $\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$ 。
 - **各向同性 (Isotropic):** 二阶导数算子（如拉普拉斯算子）是旋转不变的。这意味着算子的响应与图像中不连续性的方向无关，能够均匀地提取各个方向的细节。
 - **零交叉 (Zero-crossing):** 二阶导数在斜坡过渡的中点会穿过零轴，这为精确定位边缘提供了物理基础。

10.2.2 孤立点的检测 (Detection of Isolated Points)

- **原理:** 由于拉普拉斯算子是各向同性的，它对孤立点的响应最强。
- **算子掩模:** 常用 3×3 的拉普拉斯掩模。

标准拉普拉斯 (90° 旋转)	扩展拉普拉斯 (45° 各向同性)
$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

- **检测准则:** 计算响应值 R 。若 $|R| > T$ (T 为预设非负阈值)，则判定该点为孤立点。由于其各向同性，无论噪声点在何处，其响应计算方式均一致。

10.2.3 线检测 (Line Detection)

线检测考虑特定方向的突变，与点检测不同，其掩模具有方向选择性。

表 1: 常见线检测算子掩模

水平 (Horizontal)	+45° 方向	垂直 (Vertical)	-45° 方向
$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix}$	$\begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$

- **检测逻辑:** 对每个像素应用上述四个掩模。如果 $|R_i| > |R_j|$ (对于所有 $j \neq i$) 且 $|R_i| > T$ ，则该点属于方向 i 的线。

10.2.4 边缘模型 (Edge Models)

边缘检测的性能受边缘类型影响，主要模型包括：

- **阶跃边缘 (Step Edge)**: 灰度级在 1 像素距离内发生跳变。
- **斜坡边缘 (Ramp Edge)**: 灰度在一定宽度内逐渐变化。一阶导数产生“宽脉冲”响应，二阶导数产生正负峰值对。
- **屋顶边缘 (Roof Edge)**: 穿过区域的线条，其宽度取决于线的粗细。

10.2.5 基本边缘检测 (Basic Edge Detection)

1. 执行边缘检测的三个基本步骤 一个完整的边缘检测器通常包含以下三个关键过程：

1. **图像平滑 (Smoothing)**: 噪声是灰度的不连续，导数会将其极大放大。因此，第一步必须使用低通滤波器（如高斯滤波）抑制噪声。
 2. **边缘增强 (Enhancement)**: 通过计算梯度幅值（一阶导）或拉普拉斯响应（二阶导），在局部产生显著的灰度变化响应。
 3. **边缘检测 (Detection)**: 通过阈值处理，将增强后的像素点从背景中分离出来，形成最终的二值边缘图。
2. **梯度算子 (Gradient Operators)** 通过计算向量 $\nabla f = [g_x, g_y]^T$ 的幅值 $M(x, y)$ 来寻找边缘。

表 2: 常用梯度检测算子掩模

算子名称	g_x (水平响应)	g_y (垂直响应)	特点简述
Roberts	$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$	交叉差分，计算最快，但对噪声极其敏感。
Prewitt	$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	均值平滑思想，抗噪性强于 Roberts。
Sobel	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	工程最常用。通过给中心像素加权，兼顾了差分与平滑，抗噪性能优秀。

10.2.6 高级边缘检测器 (Advanced Edge Detection)

1. Marr-Hildreth 检测器 (LoG)

- **步骤**: 先高斯平滑，再求拉普拉斯。
- **特点**: 利用 3.5σ 作为掩模尺寸。它在不同尺寸下能捕捉不同粒度的边缘，并通过 ** 零交叉点 ** 来精确确定边缘。

2. Canny 边缘检测器 (Canny Edge Detector) 公认最优算子，核心步骤如下：

1. 高斯滤波：平滑图像。
2. 计算梯度：求取幅值 $M(x, y)$ 和方向 $\alpha(x, y)$ 。
3. 非极大值抑制 (NMS)：在梯度方向上搜索局部最大值，将粗边缘压细为单像素宽。
4. 滞后阈值处理：使用高低两个阈值。高阈值用于寻找“强边缘”；低阈值用于寻找与强边缘连接的“弱边缘”，从而补齐边缘断裂。

10.3 阈值处理

阈值处理由于其计算简单、性能稳定，在许多应用中处于核心地位。其基本目的是通过一个准则自动选出将目标与背景分离的灰度值。

10.3.1 基础知识

设原始图像为 $f(x, y)$ ，选定阈值 T ，分割后的二值图像 $g(x, y)$ 定义为：

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

其中，1 代表目标物体，0 代表背景。

- **单阈值**：只有一个阈值 T 。
- **多阈值**：使用多个阈值将图像划分为多个灰度区间。
- **影响因素**：峰值的分离程度、图像噪声、光照均匀性以及物体的相对尺寸。

10.3.2 Otsu 算法 (大津法/最大类间方差法)

Otsu 算法是自动选取最优阈值的经典方法。其核心思想是：**最优阈值应使分割后的两类（背景与目标）之间的方差最大**。

(1) **符号定义** 设图像大小为 $M \times N$ ，灰度级为 $[0, L - 1]$ 。 n_i 为灰度级 i 的像素数，归一化直方图为 $p_i = n_i / (M \times N)$ 。假设我们将像素按阈值 k 分为两类： C_1 (灰度级 $[0, k]$) 和 C_2 (灰度级 $[k + 1, L - 1]$)。

(2) 类概率与类均值

- **类概率** (出现频数): $\omega_1(k) = \sum_{i=0}^k p_i$, $\omega_2(k) = \sum_{i=k+1}^{L-1} p_i = 1 - \omega_1(k)$
- **类均值**: $m_1(k) = \frac{1}{\omega_1(k)} \sum_{i=0}^k i \cdot p_i$, $m_2(k) = \frac{1}{\omega_2(k)} \sum_{i=k+1}^{L-1} i \cdot p_i$
- **全图均值**: $m_G = \omega_1 m_1 + \omega_2 m_2$

(3) 目标函数: 类间方差 类间方差 $\sigma_B^2(k)$ 定义为:

$$\sigma_B^2(k) = \omega_1(k)(m_1(k) - m_G)^2 + \omega_2(k)(m_2(k) - m_G)^2$$

简化后的等价公式:

$$\sigma_B^2(k) = \frac{[m_G\omega_1(k) - m(k)]^2}{\omega_1(k)[1 - \omega_1(k)]}$$

其中 $m(k) = \sum_{i=0}^k i \cdot p_i$ 是截止到 k 的累积均值。

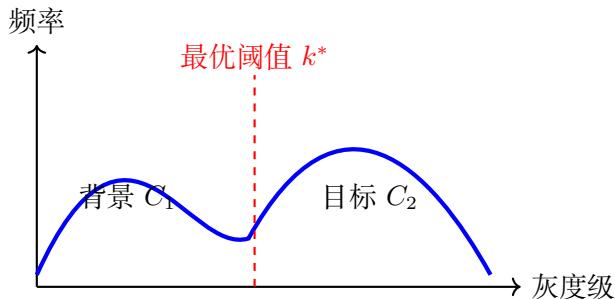
(4) 算法步骤

1. 计算原始图像的归一化直方图 p_i 。
2. 计算累积均值 $m(k)$ 和全图平均灰度 m_G 。
3. 遍历 $k \in [0, L - 1]$, 计算每个 k 对应的类间方差 $\sigma_B^2(k)$ 。
4. 选取使 $\sigma_B^2(k)$ 达到最大值的 k^* 作为最优阈值: $k^* = \arg \max \sigma_B^2(k)$ 。

10.3.3 自适应阈值 (Variable Thresholding)

当图像光照不均匀或背景复杂时, 全局阈值 (如 Otsu) 往往失效。

- 局部阈值: 将图像分成若干子图像, 对每个子图像独立计算阈值。
- 移动平均法: 计算像素周围一定范围内 (如扫描线方向) 的移动平均值作为当前点的阈值。该方法对扫描打印文档的二值化非常有效。



10.3.4 例题 23: 基于噪声统计特性的阈值选择

题目描述 已知图像中的目标区域和背景区域在灰度区间 $[0, 255]$ 上的平均灰度分别为 $\mu_{obj} = 170$ 和 $\mu_{bg} = 60$ 。图像被均值为 0、标准差 $\sigma = 10$ 的高斯噪声污染。请提出一种正确分割率达 90% 以上的阈值分割方法。

解题思路与分析

1. **概率分布分析:** 受高斯噪声污染后, 目标和背景的灰度分布分别服从高斯分布 $N(170, 10^2)$ 和 $N(60, 10^2)$ 。
2. **阈值选取:** 由于两个分布的方差相等 ($\sigma_{obj} = \sigma_{bg} = 10$), 根据贝叶斯最小误差准则, 最优阈值 T 应选在两个均值的中点:

$$T = \frac{\mu_{obj} + \mu_{bg}}{2} = \frac{170 + 60}{2} = 115$$

3. 正确率验证：回顾高斯分布的特性：

- 约 68.3% 的面积在 $\mu \pm \sigma$ 范围内。
- 约 95.4% 的面积在 $\mu \pm 2\sigma$ 范围内。
- 约 99.7% 的面积在 $\mu \pm 3\sigma$ 范围内。

对于背景区域（均值 60），阈值 $T = 115$ 距离均值的距离为：

$$d = T - \mu_{bg} = 115 - 60 = 55 = 5.5\sigma$$

对于目标区域（均值 170），阈值 $T = 115$ 距离均值的距离为：

$$d = \mu_{obj} - T = 170 - 115 = 55 = 5.5\sigma$$

由于 $5.5\sigma \gg 3\sigma$ ，根据 3σ 原则，此时错误分割的概率（即背景像素大于 115 或目标像素小于 115 的概率）远小于 0.3%。

4. 结论：设定全局阈值 $T = 115$ 。由于该阈值距离两个类别的均值均超过 5σ ，其正确分割率将远高于 99.9%，完全满足题目要求的 90% 以上。

10.4 基于区域的分割

与基于边缘检测寻找“不连续性”不同，基于区域的分割是根据像素的“相似性”将图像划分为不同的区域。

10.4.1 区域生长 (Region Growing)

区域生长是从一组“种子点”开始，根据预定义的准则将邻域像素合并到种子所在区域的过程。

- 核心要素：

1. 种子点的选择：根据先验知识（如灰度级范围）自动提取，或由人工手动选取。
2. 相似性准则：通常基于灰度差、颜色、纹理等属性。例如：若待加入像素与种子的灰度差小于 T ，则归为一类。
3. 连通性准则：通常使用 8-连通或 4-连通。

- 停止规则：当不再有满足相似性与连通性准则的像素时，区域生长停止。

10.4.2 区域分裂与聚合 (Region Splitting and Merging)

该算法利用四叉树 (Quadtree) 结构，通过不断分解和合并来满足区域的一致性要求。

算法逻辑 令 R 表示整幅图像区域， Q 是反映区域内像素相似性的属性准则。

1. 分裂：若 $Q(R_i) = \text{FALSE}$ （区域内部不一致），则将 R_i 分裂为 4 个不相交的象限区域。
2. 聚合：当无法进一步分裂时，检查相邻区域 R_j 和 R_k 。若 $Q(R_j \cup R_k) = \text{TRUE}$ （合并后依然一致），则将其聚合。
3. 最小尺寸：通常规定分裂的最小尺寸（如 4×4 ）以避免过度分割。

10.4.3 基于聚类的分割方法 (Clustering Methods)

将阈值分割概念推广到多维特征空间，最常用的是 **K-均值 (K-means) 聚类**。

1. **初始化**: 在特征空间中随机选取 K 个聚类中心。
2. **分配**: 计算每个像素到各个中心的距离，将其归入距离最近的类。
3. **更新**: 根据现有的分类结果重新计算各类的重心作为新的聚类中心。
4. **迭代**: 重复上述步骤直到中心位置不再变化。

10.5 形态学分水岭分割

分水岭算法是一种基于拓扑理论的分割方法。它将图像看作地理拓扑表面，其中像素的灰度值代表海拔高度。

10.5.1 基本概念

- **区域最小值**: 地形图中的局部最低点（盆地底部）。
- **集水盆地 (Catchment Basins)**: 如果雨水落在某个区域最终都流向同一个局部最小值，该区域称为集水盆地。
- **分水岭/脊线 (Watershed lines)**: 雨水流向不同最小值的边界线，即我们要寻找的分割线。

10.5.2 算法物理过程: 模拟浸没 (Immersion)

想象在每个局部最小值处打洞，并将整个地形表面慢慢浸入水中：

1. 水会从最小值处上升，逐渐充满集水盆地。
2. 当来自两个不同盆地的水即将汇合时，建立一座“大坝”来阻止它们混合。
3. 继续浸没过程，直到水面到达最高点。
4. 最终保留下来的所有大坝就是图像的**分水岭线**。

10.5.3 梯度图像的应用

在实际分割中，我们通常不直接对原图进行分水岭变换，而是对**梯度图像**进行操作。

- **原因**: 原始图像的梯度图在物体边界处响应最高（脊线），而在物体内部（灰度均匀区域）响应最低（盆地）。
- **效果**: 这样做可以将物体内部作为盆地，将物体的边缘作为分水岭。

10.5.4 过度分割问题与标记符限制 (Markers)

直接应用分水岭算法常会因为噪声和细微的灰度波动产生数以千计的局部最小值，导致**过度分割 (Over-segmentation)**。

标记符限制的分水岭算法：为了克服过度分割，引入“标记符”来引导分割过程：

1. **内部标记符**：属于感兴趣物体的像素区域。
2. **外部标记符**：属于背景的像素区域。

通过强制修改梯度图像，使其只在标记符所在位置拥有局部最小值，从而确保算法只在感兴趣的区域生成分水岭。

10.5.5 典型应用流程

一个标准的形态学分水岭分割通常包含以下步骤：

1. **预处理**：使用平滑滤波器减少噪声。
2. **梯度计算**：计算图像的梯度幅值（如使用 Sobel 算子）。
3. **标记符提取**：利用阈值处理或形态学操作（如开闭运算、重建）提取内部和外部标记符。
4. **强制修改**：修改梯度图，使得只有标记符位置是局部最小值。
5. **分水岭变换**：在修改后的梯度图上运行算法，得到最终边界。

10.6 第十章核心重点问题总结

1. **图像分割的两大基本性质是什么？**
 - **不连续性**：基于灰度的剧烈变化检测边缘、点、线。
 - **相似性**：基于预定义的准则（如灰度、纹理）将像素归类为区域。
2. **为什么 Sobel 算子比 Roberts 算子更常用？** Sobel 算子在计算差分的同时，在垂直于差分方向上引入了平均操作，这赋予了它一定的平滑（抑制噪声）能力。Roberts 算子虽然简单，但对噪声极其敏感。
3. **简述 Canny 边缘检测算法的核心步骤及其优越性。** Canny 算法是目前公认的性能最优的边缘检测算子之一，其步骤如下：
 - **高斯平滑**：降低噪声干扰。
 - **计算梯度**：获取图像幅值 $M(x, y)$ 和方向 $\alpha(x, y)$ 。
 - **非极大值抑制 (NMS)**：沿着梯度方向检查，仅保留局部幅值最大的像素。这一步实现了边缘的精确正确定位，将粗边缘压细。
 - **滞后阈值处理**：设置高低两个阈值。高阈值用于确定强边缘，低阈值用于连接与强边缘相邻的弱边缘。
 - **优越性**：它实现了低错误率（检测出的边缘真实存在）、高定位精度（检测到的边缘与真实边缘中心重合）以及单像素响应（消除虚假边缘）。
4. **简述 Otsu 算法（大津法）的核心原理。** 其核心是寻找一个最优阈值 k ，使得分割后的两个类别（背景和目标）之间的类间方差达到最大。类间方差越大，意味着背景和目标的灰度区分度越高。

5. 霍夫变换 (Hough Transform) 如何解决直线检测问题？它利用对偶性，将图像空间中的点变换为参数空间中的曲线（正弦曲线）。图像空间中多个点共线，对应参数空间中多条曲线相交于一点。通过寻找参数空间中的局部峰值（交点）即可检测直线。
6. 区域分裂聚合算法中，四叉树 (Quadtree) 的作用是什么？四叉树提供了一种系统化的空间搜索机制。它将不满足一致性条件的图像块不断四等分（分裂），并检查相邻块是否满足一致性以决定是否合并，从而保证了分割结果在数学描述上的严谨性。
7. 分水岭算法如何处理过度分割问题？主要通过标记符 (Markers)。通过预先定义的内部和外部标记符强制修改梯度图像的最小值分布，限制分水岭的生成位置，从而避免噪声导致的碎片化分割，只提取感兴趣物体的轮廓。