第八章：图像分割（一）

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)
李　礼 (lil1@ustc.edu.cn)
胡　洋 (eeyhu@ustc.edu.cn)

图像分割

- 图像分割定义
- 阈值分割
- 区域生长法
- 分裂合并方法
- 分水岭算法
- 聚类分割算法
- 主动轮廓分割
- Graph Cut

图像分割

- □ 图像分割定义
- □ 阈值分割
- □ 区域生长法
- □ 分裂合并方法
- □ 分水岭算法
- □ 聚类分割算法
- □ 主动轮廓分割
- □ Graph Cut

图像分割的定义

- 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）$R_1, R_2, \cdots, R_n$：

1. $\bigcup_{i=1}^{n} R_i = R$
2. 对$\forall i \neq j$,有$R_i \cap R_j = \emptyset$
3. 每个子区域R_i都是连通的；
4. 对于各个子区域，有均匀性测度度量P为真；但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：

$$P(R_i) = \text{TRUE}, \text{且 } P(R_i \cup R_j) = \text{FALSE}$$

图像分割

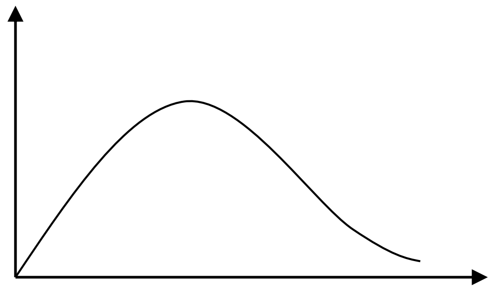
- 图像分割定义
- 阈值分割
- 区域生长法
- 分裂合并方法
- 分水岭算法
- 聚类分割算法
- 主动轮廓分割
- Graph Cut

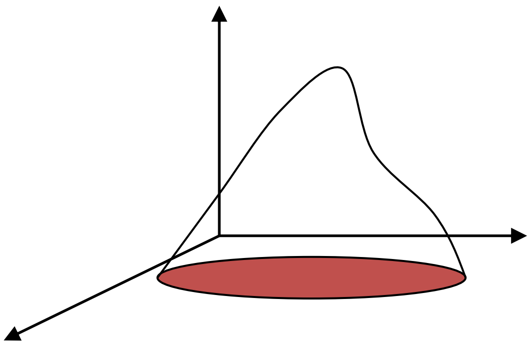
原理和分类

- 阈值分割法一般基于如下图像模型：
  - 图像由具有单峰测度(灰度, 颜色, 纹理)分布的目标和背景组成
  - 目标或背景内部的相邻像素间的测度值高度相关
  - 目标和背景交界处两边的像素测度值差异大
- 满足上述条件的图像的直方图是双峰的，可用阈值法进行较好的分割

原理和分类

- 取阈值法是以图像直方图为依据，选定阈值，再逐个对像素作判决。
- 图像直方图可以是：
  - 单个特征的一维直方图
    - ✓ 如灰度直方图。
  - 多个特征的多维直方图
    - ✓ 如两个波段组成的二维直方图。

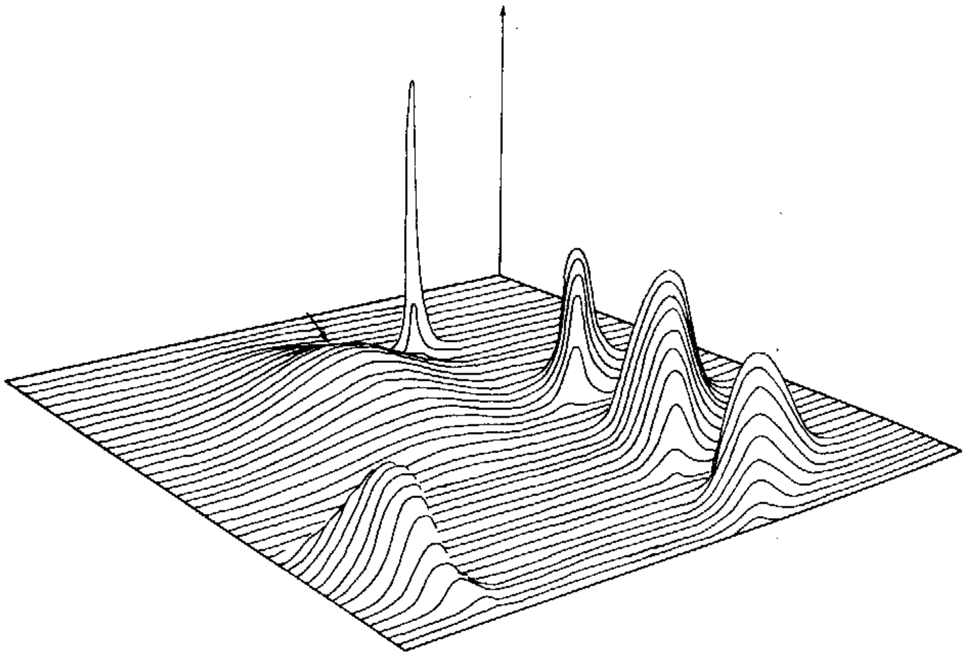




原理和分类

- 图像的特征直方图可以是：
  - 灰度直方图
  - 梯度直方图
  - 纹理直方图

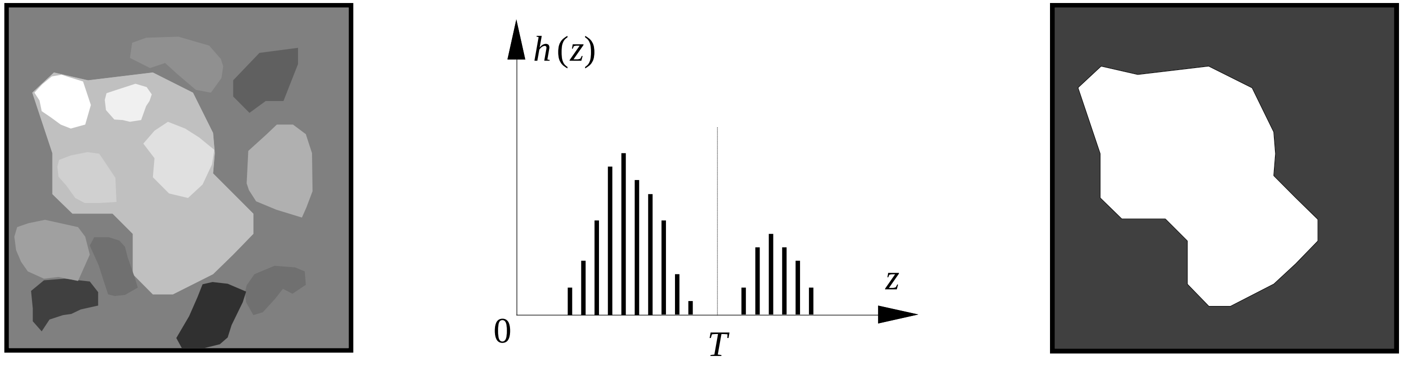
  ……
- 多个特征组成多维直方图



原理和分类

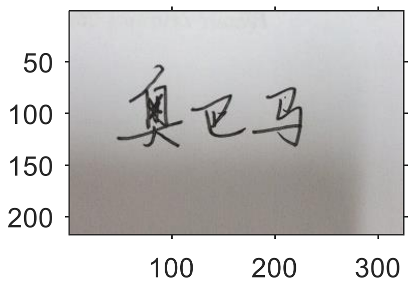
- 单阈值分割图像
- 对灰度图（取值在g_{min}和g_{max}之间）确定一个灰度阈值T（$g_{min} < T < g_{max}$）

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

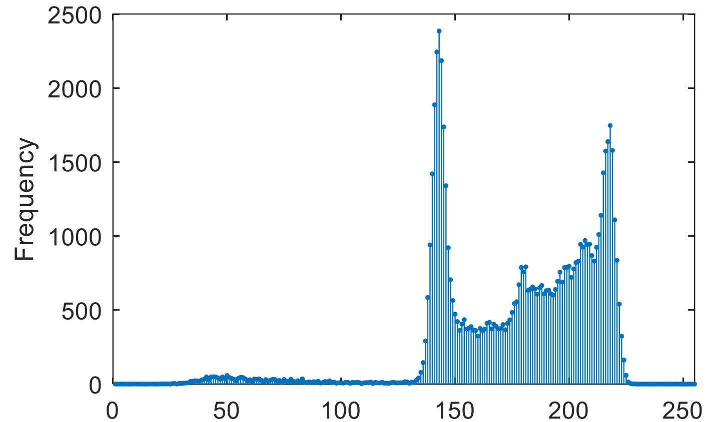


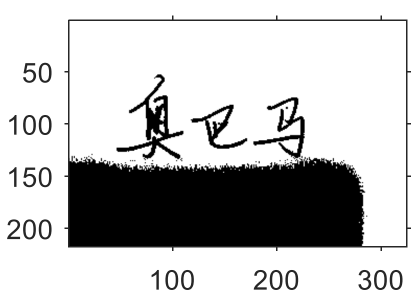
原理和分类

□ 实例：生成电子签名

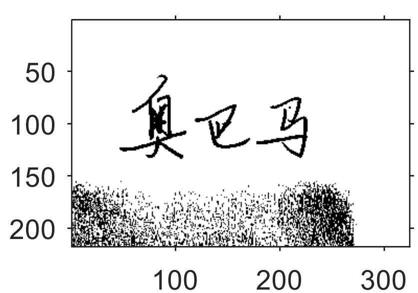


f

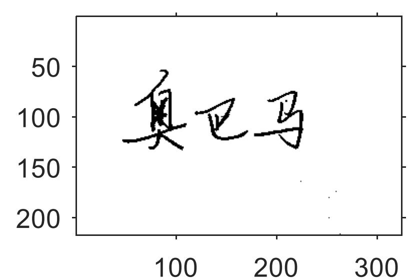




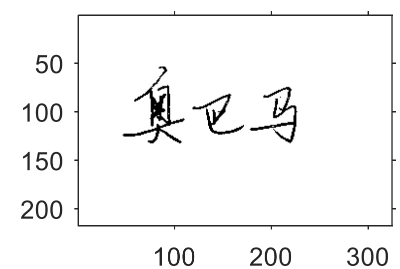
$f > 160$



$f > 140$



$f > 130$

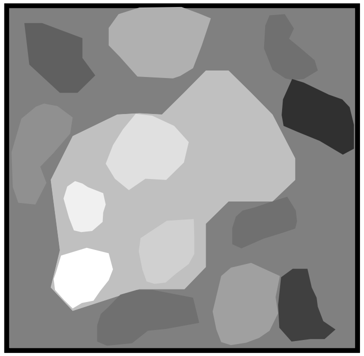


$f > 100$

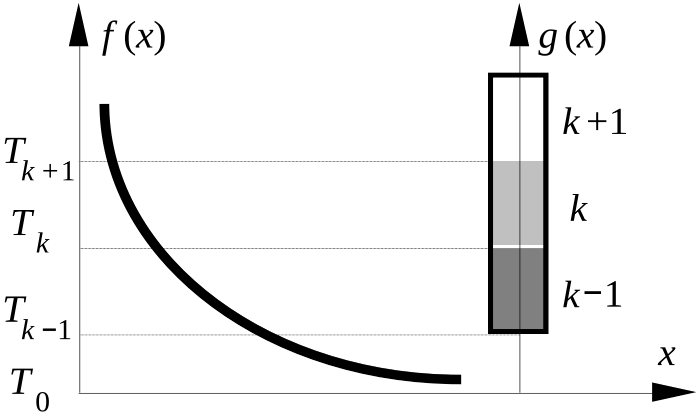
原理和分类

- 多阈值分割图像
  - 确定一系列分割阈值

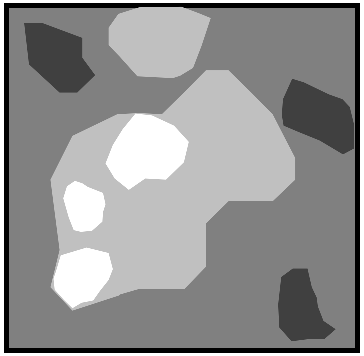
$$g(x,y)=k \qquad 如 \quad T_k < f(x,y) \leqslant T_{k+1} \qquad k=0,1,2,\cdots,K$$



(a)



(b)



(c)

原理和分类

- 阈值分割方法分类

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

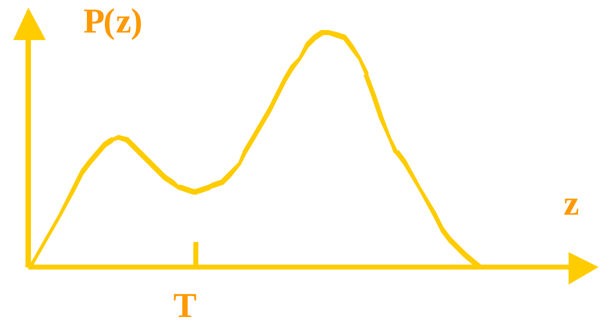
1. 依赖像素的（全局）阈值方法：
   - 仅根据 $f(x, y)$来选取阈值
2. 依赖区域的（局部）阈值方法：
   - 根据 $f(x, y)$和$p(x, y)$来选取阈值，$p(x, y)$表示像素(x, y)的邻域局部性质
3. 依赖坐标的（动态）阈值方法：
   - 除根$f(x, y)$和$p(x, y)$来选取，还与x, y有关

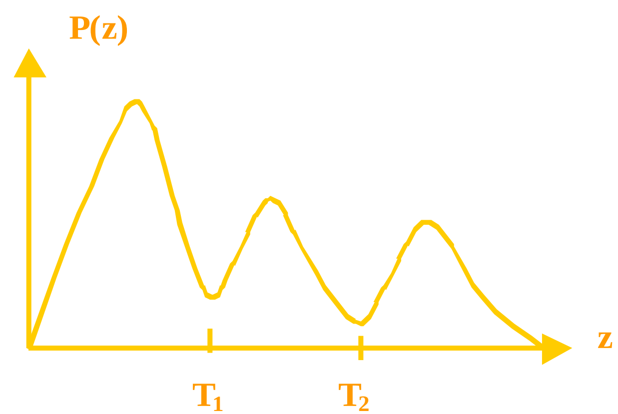
- 将前两种阈值也称为固定阈值

依赖象素的阈值选取

适用于直方图为双峰或多峰的图象

- 单一阈值
  - 取一阈值，把图像分成目标和背景两部分
- 多个阈值
  - 取几个阈值将图象分成若干个目标和背景几部分
- 阈值宜取在双峰或多峰直方图的谷点。





极小值点阈值：$\dfrac{\partial h(z)}{\partial z}=0$和$\dfrac{\partial^2 h(z)}{\partial z^2}>0$

最优阈值的选取

可以证明，在对象的分布近似为正态分布时，阈值选取在直方图的谷点，分割误差最小。

$$p_1(z)=\frac{1}{\sqrt{2\pi}\sigma_1}\exp[-\frac{(z-\mu_1)^2}{2{\sigma_1}^2}] \qquad E_1(T)=\int_T^{\infty}p_1(z)dz$$

$$p_2(z)=\frac{1}{\sqrt{2\pi}\sigma_2}\exp[-\frac{(z-\mu_2)^2}{2{\sigma_2}^2}] \qquad E_2(T)=\int_{-\infty}^{T}p_2(z)dz$$

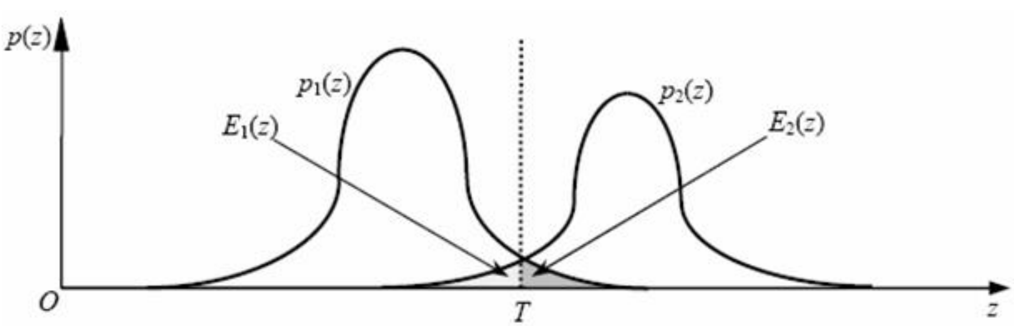
$$E(T)=P_1E_1(T)+P_2E_2(T)$$

（P_1，P_2是先验概率）

为求使该误差最小的阈值，令 $\frac{\partial E(T)}{\partial T}=0\Rightarrow P_1\times p_1(T)=P_2\times p_2(T)$

当$\sigma_1=\sigma_2=\sigma$时

$$T=\frac{\mu_1+\mu_2}{2}+\frac{\sigma^2}{\mu_1-\mu_2}\ln(\frac{P_2}{P_1})$$



p(z)的参数可根据最小均方的方法借助直方图得到

$$e_{ms}=\frac{1}{n}\sum_{i=1}^{n}[p(z_i)-h(z_i)]^2$$

Otsu's 方法 (1979)

□ 基本思想

- ■ 假设图像像素可分为两类，选择分割阈值，使得类内方差最小。

□ 形式化表达

- ■ 给定一个图像*I*，计算其归一化的直方图*P*(*i*)
- ■ 选择阈值*t*，将像素分为两类，每类概率：

$$q_1(t) = \sum_{i=1}^{t} P(i) \qquad q_2(t) = \sum_{i=t+1}^{T} P(i)$$

- ■ 每类像素的灰度均值：

$$\mu_1(t) = \sum_{i=1}^{t} \frac{i \cdot P(i)}{q_1(t)} \qquad \mu_2(t) = \sum_{i=t+1}^{T} \frac{i \cdot P(i)}{q_2(t)}$$

- Otsu, N., "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 9, No. 1, 1979, pp. 62-66.

Otsu's 方法

- 形式化表达

  每类像素的灰度方差：

$$\sigma_1^2(t)=\sum_{i=1}^{t}[i-\mu_1(t)]^2\frac{P(i)}{q_1(t)}\qquad\sigma_2^2(t)=\sum_{i=t+1}^{T}[i-\mu_2(t)]^2\frac{P(i)}{q_2(t)}$$

- 加权类内方差：$\sigma_w^2(t)=q_1(t)\sigma_1^2(t)+q_2(t)\sigma_2^2(t)$

- 最优阈值选择：$\tau=\arg\min_t \sigma_w^2(t)$

- 二值化图像分割：$B(i,j)=\begin{cases}1 & I(i,j)\ge\tau\\ 0 & I(i,j)<\tau\end{cases}$

依赖象素的阈值选取

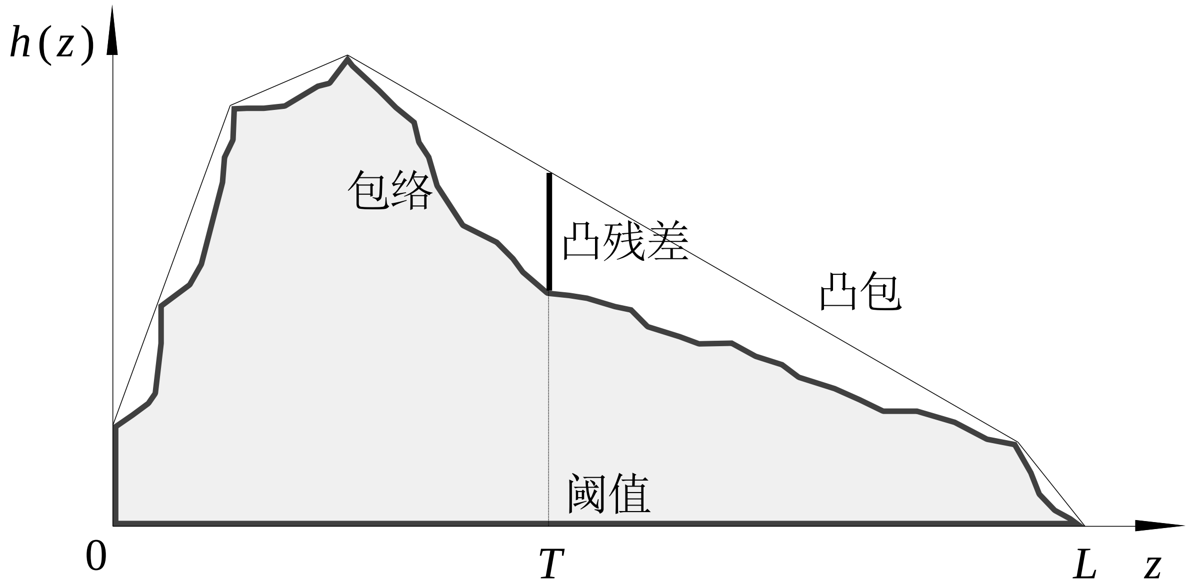
- 由直方图凹凸性确定的阈值

  直方图的一个峰淹没在另一个峰旁的缓坡里
- 直方图的包络

  →区域凸包

  →最大凸残差

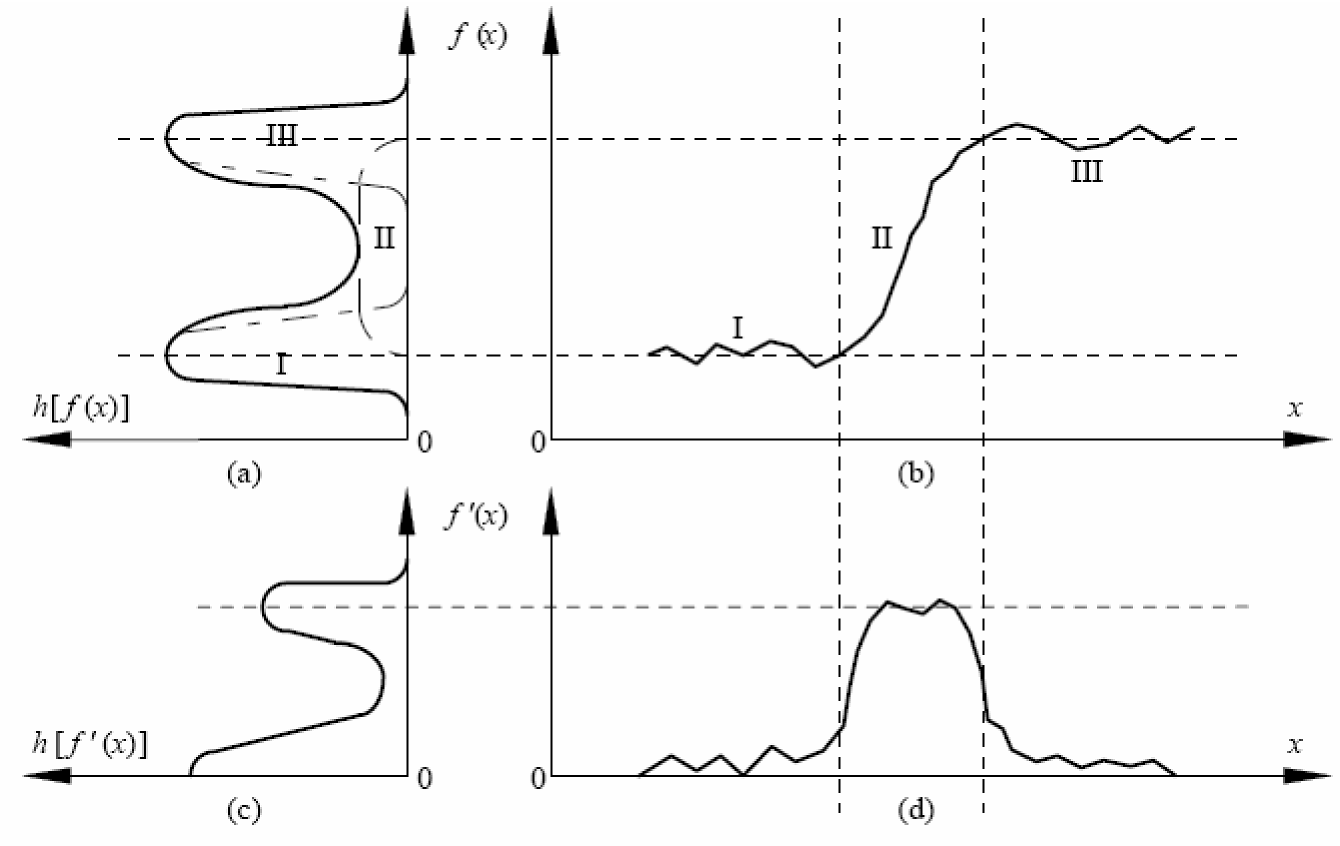
  →分割阈值



依赖区域的阈值选取

- 直方图变换
  - **仅利用像素灰度可能出现的问题：**
    - ✓ 灰度直方图的谷被填充
- 借助邻域性质变换原来的直方图
  1. 获得低梯度值像素的直方图
     - ✓ **峰之间的谷比原直方图深**
  2. 获得高梯度值像素的直方图
     - ✓ **峰由原直方图的谷转化而来**

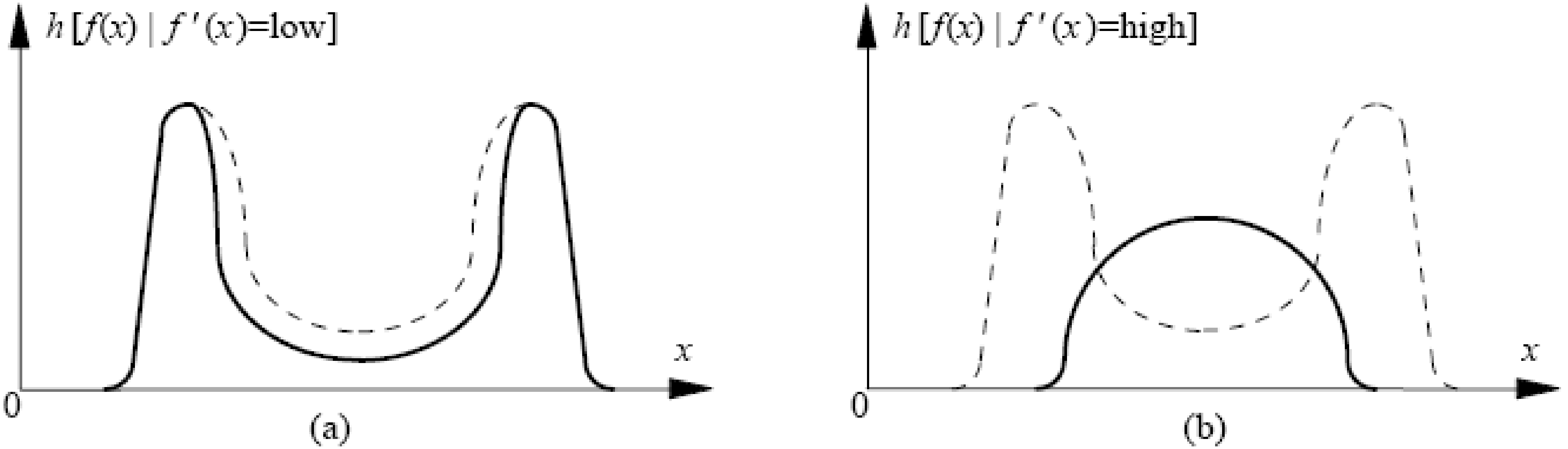
依赖区域的阈值选取



边缘及梯度的直方图

依赖区域的阈值选取

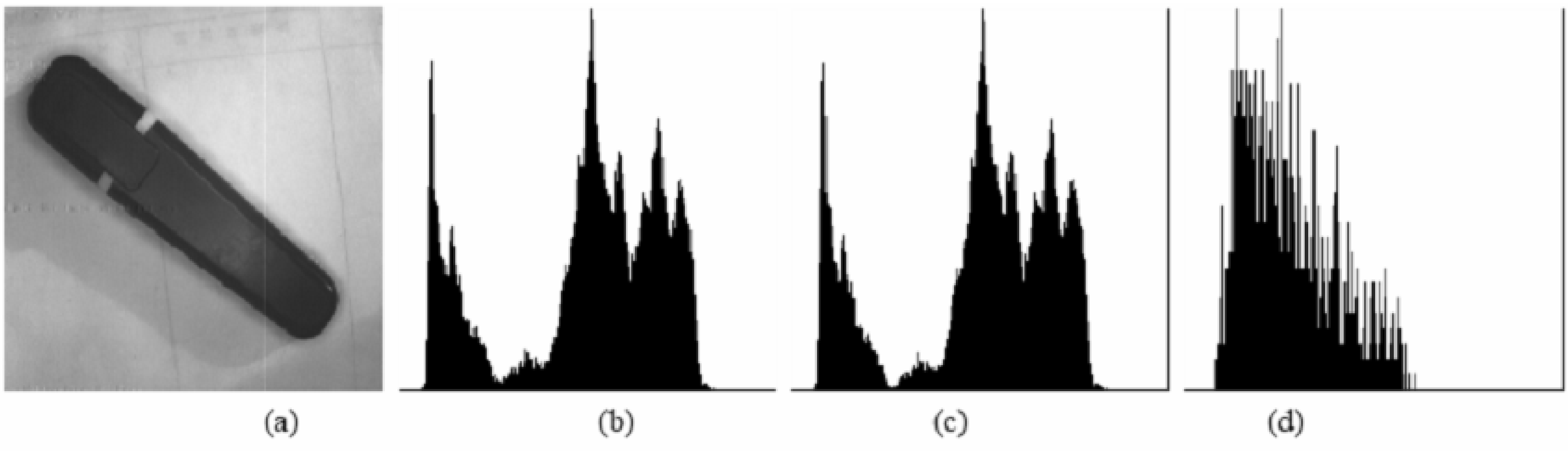
- 直方图变换
  - 加权直方图：给低梯度的像素更大的权重 $1/(1+g)^2$
  - 加权直方图：给高梯度的像素更大的权重



(a) (b)

图 4.4.6 变换直方图示例

变换直方图实例

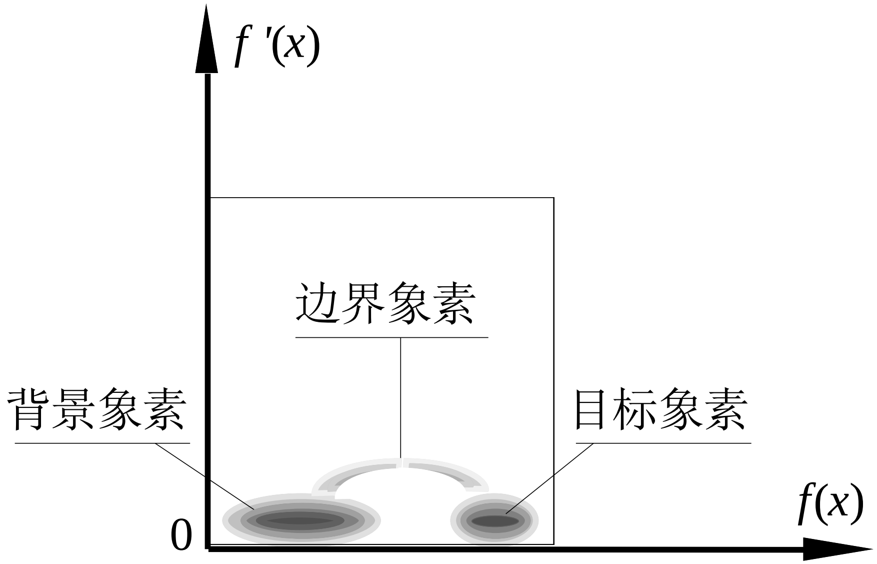


(a) 原图像；(b) 直方图；(c) 具有低梯度像素的直方图；(d)具有高梯度像素的直方图

依赖区域的阈值选取

- 直方图变换实际上可以通过灰度-梯度散射图并计算其对灰度值轴的不同权重的投影而得到

2-D直方图



2-D图象

1个轴是灰度值轴

1个轴是梯度值轴

示例

目标，背景聚类

依赖区域的阈值选取

- 灰度-梯度散射图

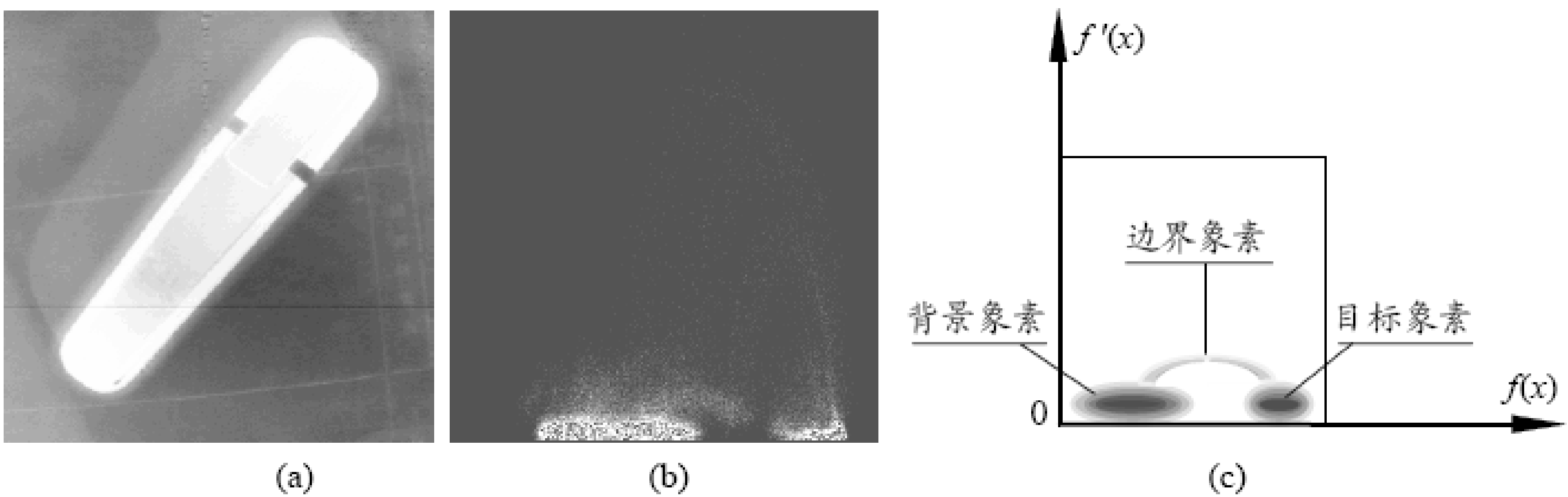
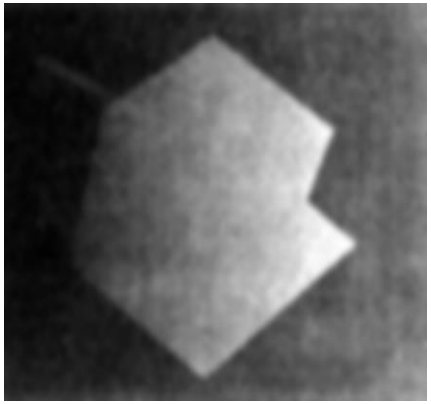
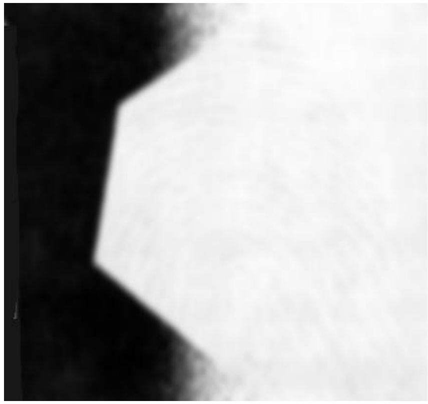


图 4.4.8 灰度和梯度散射图

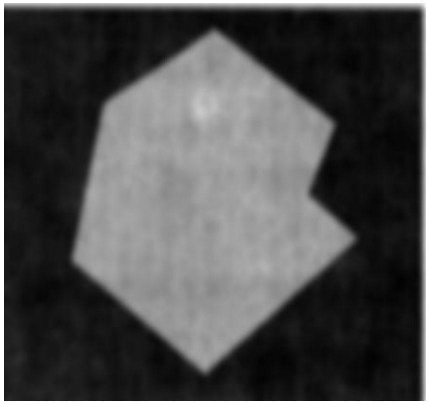
依赖坐标的阈值选取

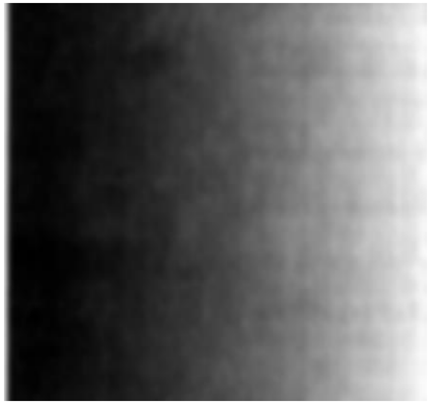
- 全局阈值不能兼顾图像各处的情况
  - 解决方案：用与坐标相关的一系列阈值来对图像分割

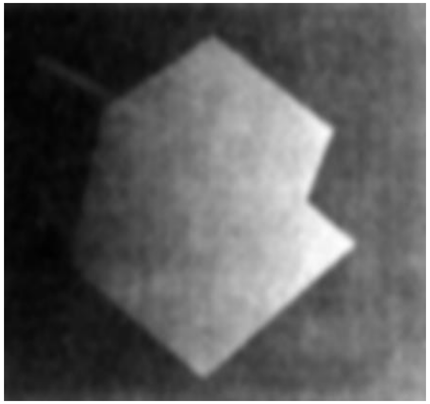
 

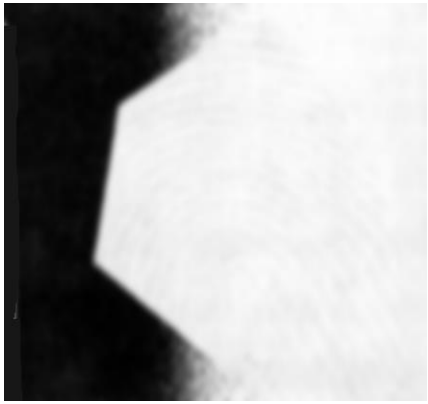
- 基本思想/思路：
  1. 将图像分解成一系列子图像
  2. 对每个子图像计算一个阈值
  3. 用这些子图像阈值对每个像素的阈值进行插值
  4. 用插值结果（阈值曲面）进行分割

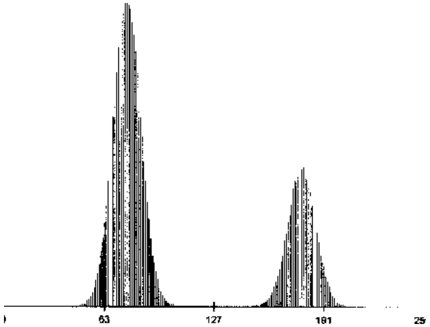
光照不均匀对分割的影响

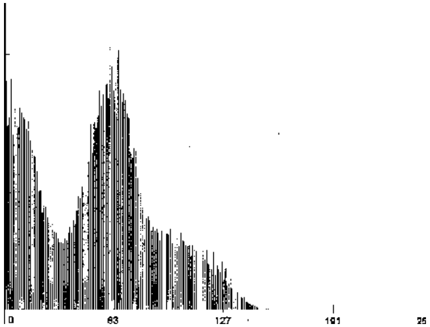






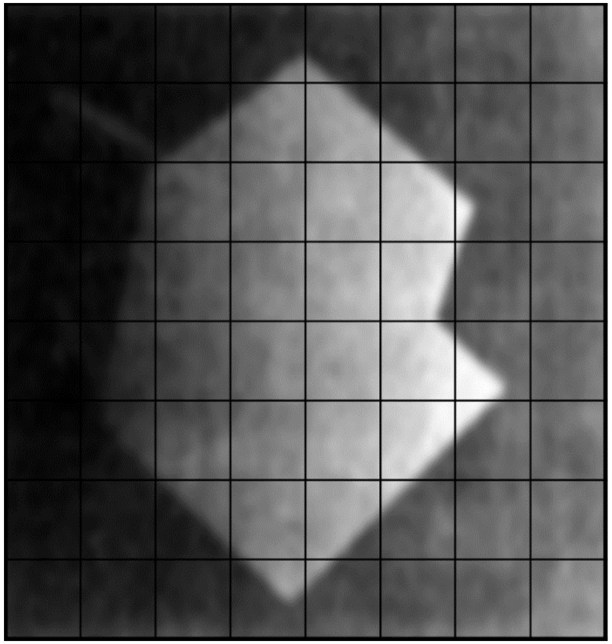




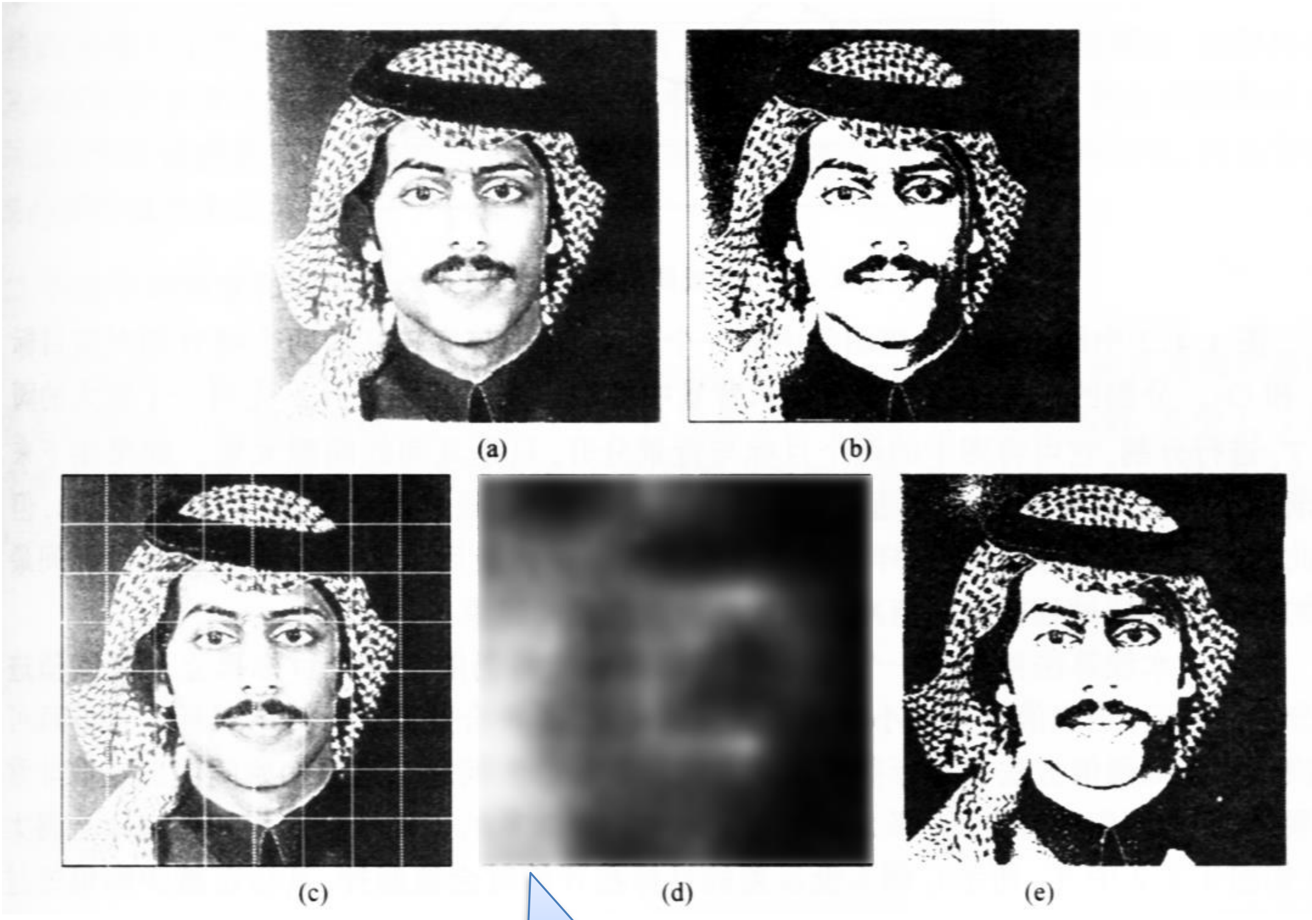


变化阈值法

1. 将整幅图像分成一系列互相之间有50%重叠的子图像
2. 做出每个子图像的直方图
3. 检测各个子图像的直方图是否为双峰的。如是，则采用最优阈值法确定一个阈值；否则，不进行处理
4. 根据对直方图为双峰的子图像得到的阈值通过插值得到所有子图像的阈值
5. 根据各子图像的阈值，再通过插值得到所有像素的阈值，然后对图像进行分割



分块取阈值一例



阈值图像

图像分割

- 图像分割定义
- 阈值分割
- 区域生长法
- 分裂合并方法
- 分水岭算法
- 聚类分割算法
- 主动轮廓分割
- Graph Cut

区域生长法

- 基本思想：
  - 将相似像素结合起来构成区域

- 基本步骤：
  1. 选择区域的种子像素
  2. 确定将相邻像素包括进来的准则
  3. 将新像素作为种子继续区域生长
  4. 制定生长停止的规则

- 讨论：
  1. 种子像素的选取
  2. 生长准则依赖应用

区域生长法

□ 生长示例

1. 根据直方图选取聚类中心的象素为种子
2. 根据与种子象素灰度差（<*T*）判断是否生长
3. 根据图象边缘确定生长何时终结

原始图

1	0	4	7	5
1	0	4	7	7
0	**1**	5	**5**	5
3	0	5	6	5
3	3	5	6	4

$T = 3$

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

$T = 2$

1	1	5	7	5
1	1	5	7	7
1	1	5	5	5
3	1	5	5	5
3	3	5	5	5

$T = 7$

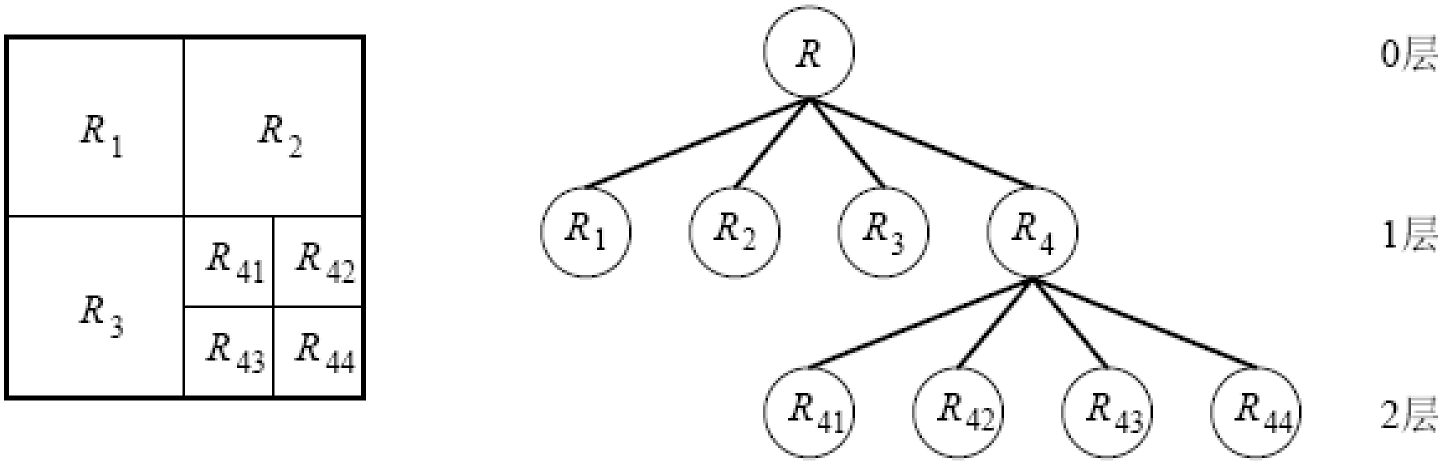
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

图像分割

- 图像分割定义
- 阈值分割
- 区域生长法
- 分裂合并方法
- 分水岭算法
- 聚类分割算法
- 主动轮廓分割
- Graph Cut

分裂合并

- 主要步骤
  - 先把图像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



图像的四叉树表达法

分裂合并

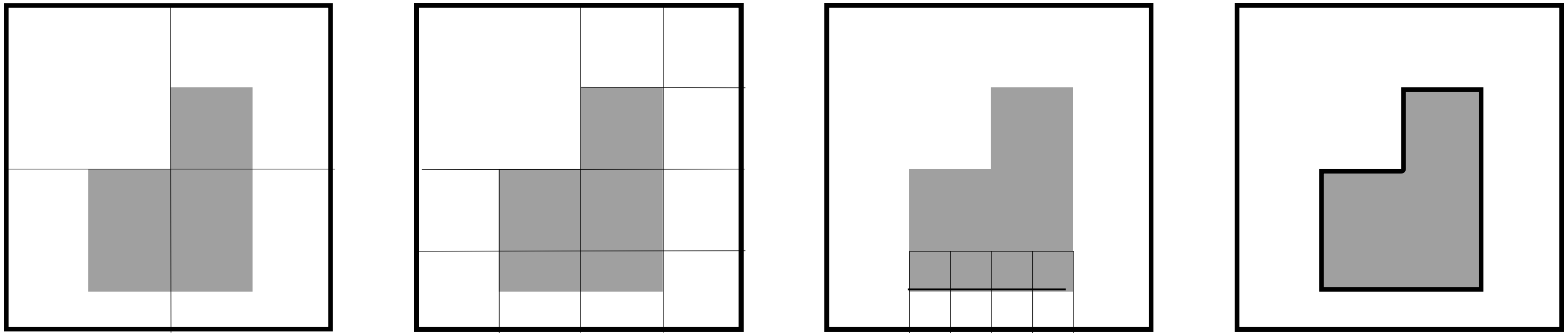
- 令R代表整个图像区域，P代表逻辑谓词
- 把R连续地分裂成越来越小的1/4的正方形子区域Ri，并且始终使$P(R_i) = \text{TRUE}$
  1. 对任一个区域R_i，如果$P(R_i) = \text{FALSE}$，就将其分裂成不重叠的四等分
  2. 对相邻的两个区域R_i和R_j，如果$P(R_i \cup R_j) = \text{TRUE}$，就将它们合并起来
  3. 如果进一步的分裂或合并都不可能了，则结束

分裂合并

1. 确定均匀性测度，构造四叉树结构
2. 选择初始分割层（一般为中间某一层）
3. 分裂处理
   - 从中间层开始，计算各块均匀性测度。对于均匀性测度为假的那些块，一分为四，重新编码。重复进行，直到各块的均匀性测度为真。
4. 合并处理
   - 从同一中间层开始，测试同属于一个父节点的四块，如果它们之和的均匀性测度为真，则合并这四块为一块。重复进行，直至不再存在可以合并的那些块。
5. 组合处理
   - 使用该数据编码判断位置，对相邻的大小不一，或者虽然大小一样，但不能合并为一个父节点的区域，进行均匀性测度测试，合并均匀测度度量为真的一对区域。反复重复这一运算，直到不再存在可以合并的区域。

分裂合并

示例（四叉树）： 分裂 分裂 合并



(a) (b) (c) (d)