



图像处理与计算机视觉

第十一讲：图像分割

任课老师：寇旗旗

TEL:18852146321 QQ:137156449

中国矿业大学计算机科学与技术学院

Image

MDL

MML

ERL1

ERL2



#134052



PR=0.718



PR=0.670



PR=0.759



PR=0.763



#169012



PR=0.781



PR=0.800



PR=0.933



PR=0.933



#253036



PR=0.710



PR=0.607



PR=0.897



PR=0.897



#249061



PR=0.865



PR=0.758

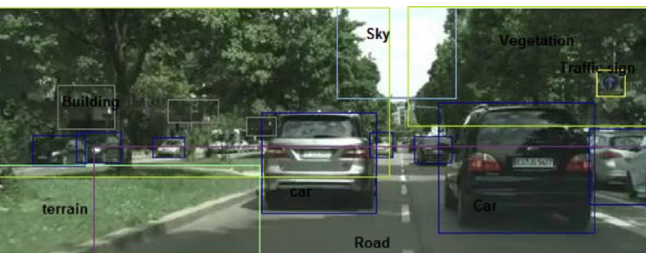


PR=0.951



PR=0.953

第11章 图像分割



第11章 图像分割



问题的提出:





第11章 图像分割

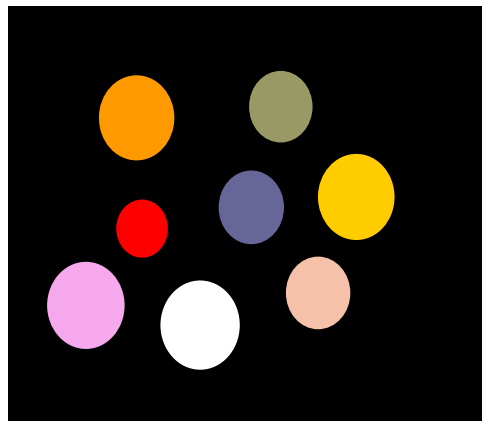
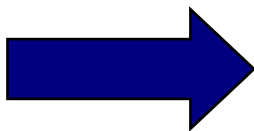
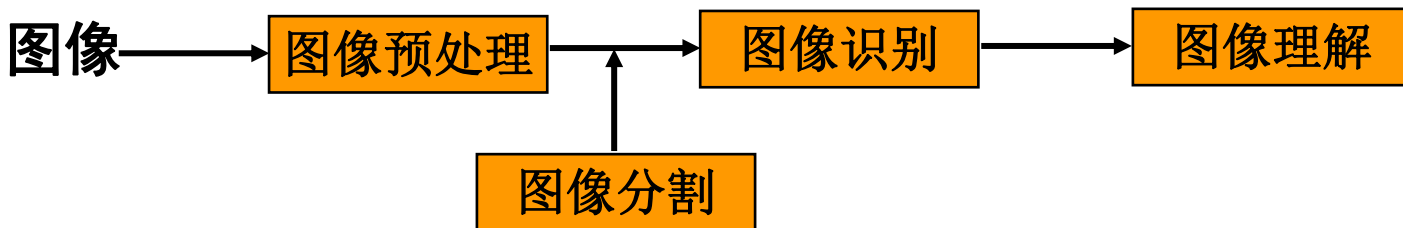


问题的提出:

- 在对图像的研究和应用中，人们往往仅对图像中的某些目标感兴趣，这些目标通常对应图像中具有特定性质的区域。
- 图像分割（Image Segmentation）是指把一幅图像分成不同的具有特定性质区域的图像处理技术，将这些区域分离提取出来，以便进一步提取特征和理解。
- 根据灰度、颜色、纹理和形状等特征把图像划分成若干互不交迭的区域，并使这些特征在同一区域内呈现出相似性，而在不同区域间呈现出明显的差异性。
- 有选择性地定位感兴趣对象在图像中的位置和范围。
- 如何实现图像分割？

第11章 图像分割

图像分割：是图像识别和图像理解的基本前提步骤





主要内容



- 11.1 阈值分割
- 11.2 边界分割
- 11.3 区域分割
- 11.4 基于聚类的图像分割
- 11.5 分水岭分割
- 11.6 综合实例



11.1 阈值分割



■ 阈值分割原理

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

其中， $f(x,y)$ 为原始图像， $g(x,y)$ 为结果图像（二值）， T 为阈值。

显然，阈值的选取决定了二值化效果的好坏。



11.1 阈值分割



- 阈值化概念
 - 上阈值化：灰度值大于等于阈值的所有像素作为前景像素，其余像素作为背景像素
 - 下阈值化：灰度值小于等于阈值的所有像素作为前景像素
 - 内阈值化：确定一个较小的阈值和一个较大的阈值，灰度值介于二者之间的像素作为前景像素
 - 外阈值化：灰度值介于小阈值和大阈值之外的像素作为前景像素



11.1 阈值分割



11.1.1 基于灰度直方图的阈值选择

11.1.2 基于模式分类思路的阈值选择

11.1.3 其他阈值分割方法

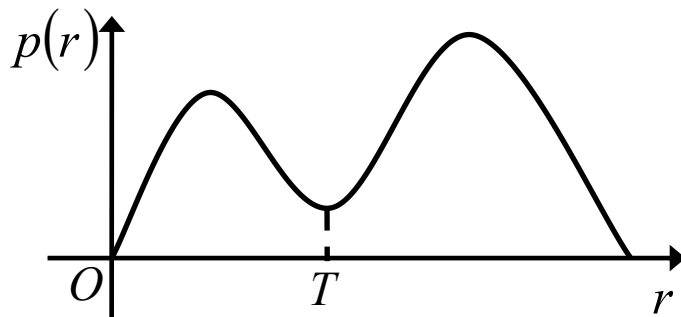


11.1.1 基于灰度直方图的阈值选择

阈值分割

(1) 原理

- 若图像的灰度直方图为**双峰分布**，表明图像的内容大致为两部分，分别为灰度分布的两个山峰的附近。



- 选择阈值为两峰间的谷底点对应灰度值

适用于图像中前景与背景灰度差别明显，且各占一定比例的情形，是一种**特殊**的方法。若图像整体直方图不具有双峰或多峰特性，可以考虑局部范围内应用。

11.1.1 基于灰度直方图的阈值选择

阈值分割

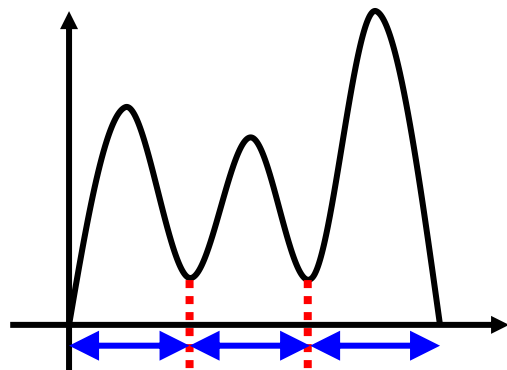
(2) 实例

- 一幅图像的直方图如图所示，根据直方图对其进行分割，分成几个区域？阈值点是什么？

- 分成3个区域，阈值点为3峰之间的谷点对应灰度级

- 或内阈值化，灰度值介于二者之间的像素作为前景像素

- 或外阈值化：灰度值介于小阈值和大阈值之外的像素作为前景像素





11.1.1 基于灰度直方图的阈值选择

阈值分割

(3) 例程

实现基于双峰分布的直方图选择阈值，分割图像

■ 分析

- 重点在于找到直方图的峰和波谷，但直方图通常是不平滑的，因此，首先要平滑直方图，再去搜索峰和谷。
- 程序设计中，将直方图中相邻三个灰度的频数相加求平均作为中间灰度对应的频数，不断平滑直方图，直至成为双峰分布。
- 也可以采用其他方法确定峰谷。



11.1.1 基于灰度直方图的阈值选择

阈值分割

(3) 例程

■ 程序

见教材【例10.1】

11.1.1 基于灰度直方图的阈值选择

阈值分割

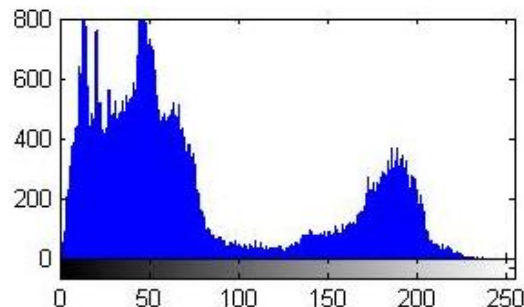
(3) 例程

■ 效果

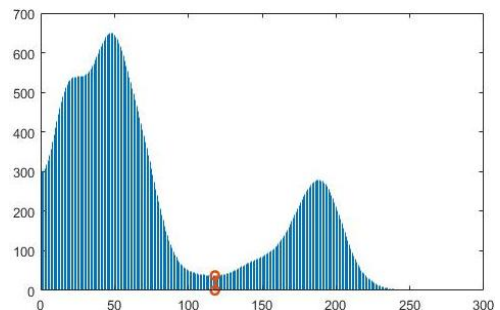
原图



分割图
 $T=118$



灰度直方图



平滑后的直
方图和波谷



11.1.2 基于模式分类思路的阈值选择

阈值分割

(1) 原理

- 认为像素值（或像素特征值）为待分类的数据，寻找合适的阈值，把数据分为不同类别，从而实现图像分割。
- 把所有的像素分为两组（类），属于“同一类别”的对象具有较大的一致性，“不同类别”的对象具有较大的差异性。
- 如何衡量同类的一致性和类间的差异性，采用不同的衡量方法对应不同的算法



11.1.2 基于模式分类思路的阈值选择

阈值分割

(2) 最大类间方差法 (OTSU)

■ 准则

各级灰度出现的概率为： $p_i = \frac{n_i}{M \times N}, i = 0, 1, 2, \dots, L-1$

两类像素在图像中的分布概率： $p_O = \sum_{i=0}^T p_i$ $p_B = \sum_{i=T+1}^{L-1} p_i$

两类像素值均值和总体灰度均值：

$$\mu_O = \frac{1}{p_O} \sum_{i=0}^T i \times p_i \quad \mu_B = \frac{1}{p_B} \sum_{i=T+1}^{L-1} i \times p_i \quad \mu = p_O \times \mu_O + p_B \times \mu_B$$



11.1.2 基于模式分类思路的阈值选择

阈值分割

(2) 最大类间方差法 (OTSU)

■ 准则

$$\text{两类方差: } \sigma_o^2 = \frac{1}{p_o} \sum_{i=0}^T p_i (i - \mu_o)^2 \quad \sigma_B^2 = \frac{1}{p_B} \sum_{i=T+1}^{L-1} p_i (i - \mu_B)^2$$

类内方差和类间方差:

$$\sigma_{in}^2 = p_o \cdot \sigma_o^2 + p_B \cdot \sigma_B^2$$

$$\sigma_1^2 = p_o \times (\mu_o - \mu)^2 + p_B \times (\mu_B - \mu)^2$$

使得类内方差最小或类间方差最大、或者类内和类间方差比值最小的阈值 T 为最佳阈值。



11.1.2 基于模式分类思路的阈值选择

阈值分割

(2) 最大类间方差法 (OTSU)

■ 函数

- ❑ $LEVEL = \text{graythresh}(I)$: 采用OTSU方法计算图像I的全局最佳阈值LEVEL。
- ❑ $BW = \text{im2bw}(I, LEVEL)$: 采用阈值LEVEL实现灰度图像I的二值化。
- ❑ $BW = \text{imbinarize}(I)$: 采用基于OTSU方法的全局阈值实现灰度图像I的二值化。
- ❑ $BW = \text{imbinarize}(I, METHOD)$: 采用METHOD指定的方法获取阈值实现灰度图像I的二值化



11.1.2 基于模式分类思路的阈值选择

阈值分割

(2) 最大类间方差法 (OTSU)

■ 程序

```
Image=rgb2gray(imread('lotus1.jpg'));  
figure,imshow(Image),title('原始图像');  
T=graythresh(Image);  
result=im2bw(Image,T);  
figure,imshow(result),title('OTSU方法二值化图像');
```


11.1.2 基于模式分类思路的阈值选择

阈值分割

(2) 最大类间方差法 (OTSU)

■ 效果



原图



分割图，阈值 $T=109$

11.1.2 基于模式分类思路的阈值选择

阈值分割

(2) 最大类间方差法 (OTSU)

■ 效果



原图



阈值 $T=80$



11.1.2 基于模式分类思路的阈值选择

阈值分割

(2) 最大类间方差法 (OTSU)

■ 效果

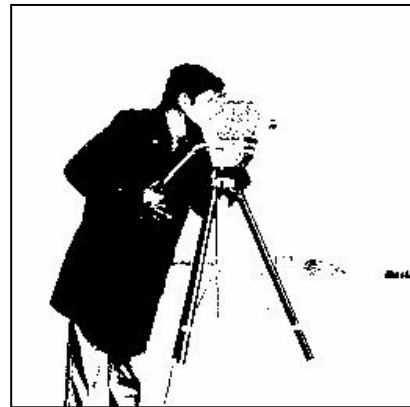


原图



Th=82

类内方差=24.4



Th=31

类内方差=29.7



11.1.2 基于模式分类思路的阈值选择

阈值分割

(3) 最大熵法

■ 准则

熵作为分类的标准：若两类的平均熵之和为最大时，从图像中获得最大信息量，对应阈值是最佳阈值。

$$H_o(T) = - \sum_{i=0}^T \frac{p_i}{p_o} \log \frac{p_i}{p_o}$$

$$H_B(T) = - \sum_{i=T+1}^{L-1} \frac{p_i}{p_B} \log \frac{p_i}{p_B}$$

$$J(T) = H_o(T) + H_B(T)$$

11.1.2 基于模式分类思路的阈值选择

阈值分割

(3) 最大熵法

■ 例程

见教材【例10.3】



原图



阈值 $T=120$

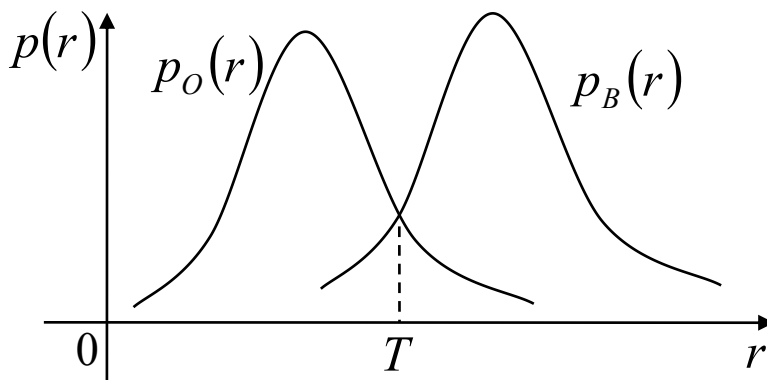


11.1.2 基于模式分类思路的阈值选择

阈值分割

(4) 最小误差法

■ 准则 分类的错误率最小时对应的阈值为最佳阈值



目标和背景的概率密度分布

目标均值和标准差： μ_O σ_O

背景均值和标准差： μ_B σ_B

正态分布概率密度：

$$p_O(r) = \frac{1}{\sqrt{2\pi}\sigma_O} e^{[-(r-\mu_O)^2/2\sigma_O^2]}$$

$$p_B(r) = \frac{1}{\sqrt{2\pi}\sigma_B} e^{[-(r-\mu_B)^2/2\sigma_B^2]}$$



11.1.2 基于模式分类思路的阈值选择

阈值分割

(4) 最小误差法

背景误判为目标的概率： $\varepsilon_B(T) = \int_{-\infty}^T p_B(r) dr$

目标误判为背景的概率： $\varepsilon_o(T) = \int_T^{+\infty} p_o(r) dr = 1 - \int_{-\infty}^T p_o(r) dr$

误判的概率： $J(T) = \alpha \varepsilon_o(T) + (1 - \alpha) \varepsilon_B(T)$

α 为目标占图像的比例。 $J(T)$ 最小时对应 T 为最佳阈值

求解极值点，可得： $(1 - \alpha) p_B(r) - \alpha p_o(r) = 0$

需要已知目标在图像中所占比例，且目标和背景的灰度概率密度符合正态分布，因此，往往需要用已知的正态分布来拟合直方图的分布，实现较为复杂。

11.1.2 基于模式分类思路的阈值选择

阈值分割

(4) 最小误差法

■ 例程

程序见教材【例10.4】，默认目标和背景服从正态分布，仅估计了参数，简化了求解



原图



阈值 $T=111$



11.1.3 其他阈值分割方法

阈值分割

(1) 基于迭代运算的阈值选择

- **基本思路**：先选择一个阈值作为初始值，然后进行迭代运算，按照某种策略不断改进阈值，直到满足给定的准则为止。
- **关键技术**：阈值改进策略的选择，应能使算法快速收敛且每次迭代产生的新阈值优于上一次的阈值。



11.1.3 其他阈值分割方法

阈值分割

(1) 基于迭代运算的阈值选择

■ 一种常用的基于迭代运算的阈值分割算法

□ 求出图像中的最小和最大灰度值 r_1 和 r_2 ，令
阈值初值为： $T^0 = \frac{r_1 + r_2}{2}$

□ 根据阈值 T^k 将图像分割成背景和目标两部分，
求出两部分的平均灰度值 r_B 和 r_O

$$r_O = \frac{\sum_{f(x,y) < T^k} f(x,y)}{N_O}$$

$$r_B = \frac{\sum_{f(x,y) \geq T^k} f(x,y)}{N_B}$$



11.1.3其他阈值分割方法

阈值分割

(1) 基于迭代运算的阈值选择

- 求出新的阈值

$$T^{k+1} = \frac{r_B + r_O}{2}$$

- 如果 $T^k = T^{k+1}$ ，则结束，否则 k 增加1，转入第二步



11.1.3其他阈值分割方法

阈值分割

(1) 基于迭代运算的阈值选择

■ 程序

```
Image=im2double(rgb2gray(imread('lotus1.jpg')));  
T=(max(Image(:))+min(Image(:)))/2;  
equal=false;  
while ~equal  
    NewT=(mean(Image(Image>=T))  
           +mean(Image(Image<T)))/2;  
    equal=abs(NewT-T)<1/256;  
    T=NewT;  
end  
result=im2bw(Image,T);  
figure,imshow(result),title('迭代方法二值化图像');
```


11.1.3 其他阈值分割方法

阈值分割

(1) 基于迭代运算的阈值选择

■ 效果



原图



阈值 $T=109.7$



11.1.3 其他阈值分割方法

阈值分割

(2) 基于模糊理论的阈值选择

- **基本思路**：模糊度表示一个模糊集的模糊程度，模糊熵是一种度量模糊度的数量指标，用**模糊熵作为目标函数**，求解最佳阈值。
- 图中的每一点对于目标和背景均有一定的隶属程度，隶属度函数为

$$\mu_f(f_{xy}) = \begin{cases} \frac{1}{1 + |f_{xy} - \mu_o|/C} & f_{xy} \leq T \\ \frac{1}{1 + |f_{xy} - \mu_B|/C} & f_{xy} > T \end{cases} \quad \begin{array}{l} C \text{ 是一个常数, 保证} \\ \mu_f(f_{xy}) \in [0.5, 1] \end{array}$$



11.1.3 其他阈值分割方法

阈值分割

(2) 基于模糊理论的阈值选择

■ 模糊熵

$$H(f) = \frac{1}{MN \ln 2} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} S(\mu_f(f_{xy}))$$

Shannon函数:
$$S(k) = \begin{cases} -k \ln k - (1-k) \ln(1-k) & k \in (0,1) \\ 0 & k = 0,1 \end{cases}$$

模糊熵取最小值时对应的阈值为最佳阈值

11.1.3 其他阈值分割方法

阈值分割

(2) 基于模糊理论的阈值选择

■ 例程

程序见教材【例10.6】



原图



阈值 $T=135$



11.2 边界分割



- 边界分割：通过检测区域的边界轮廓来实现图像分割的方法
- 分割过程：
 - 边界检测：通过各种边缘检测算子从图像中抽取边缘线段
 - 边界改良：执行各种改良边缘的处理：如边界细化，边缘闭合等
 - 边界跟踪：跟踪边缘形成边界曲线



11.2 边界分割



11.2.1 基于梯度的边界闭合

11.2.2 Hough变换

11.2.3 边界跟踪



11.2.1 基于梯度的边界闭合

边界分割

(1) 边界改良

- 目标的部分边界与相邻部分背景相近或相同时，提取出的目标区域**边界线会出现断点、不连续或分段连续**等情况；有噪声干扰时，也会使**轮廓线断开**。
- 要提取目标区域时，应使不连续边界闭合
- 方法多种多样：Hough变换、基于梯度的边界闭合技术、数学形态学等



11.2.1 基于梯度的边界闭合

边界分割

(2) 基于梯度的边界闭合

两像素 (x_1, y_1) 、 (x_2, y_2) 互为邻点，梯度幅度和方向满足

$$|G(x_1, y_1) - G(x_2, y_2)| \leq T$$

$$|d(x_1, y_1) - d(x_2, y_2)| \leq D$$

将两像素连接起来。对所有边缘像素进行同样的操作，则有希望得到闭合的边界。



11.2.2 Hough变换

边界分割

(1) 概述

- 霍夫变换 (Hough transform) 是检测图像中直线和曲线的一种方法。
- 核心思想：
 - 建立一种点—线对偶关系，将图像从图像空间变换到参数空间，**确定曲线的参数**，进而确定图像中的曲线。
 - 若边界线形状已知，通过检测图像中离散的边界点，确定曲线参数，在图像空间重绘边界曲线，进而改良边界。



11.2.2 Hough变换

边界分割

(2) 检测直线

■ 原理

以截距式方程为例： $y = kx + b$

□ 一条确定的直线对应一组确定的数据 k 、 b

xy 空间一条确定的直线对应参数空间的一个点

□ 直线变形为关于 k 和 b 的直线： $b = -xk + y$

参数空间的一条直线对应 xy 空间的一个点

综上所述， xy 空间一条直线上的 n 个点，对应参数 kb 空间经过一个公共点的 n 条直线

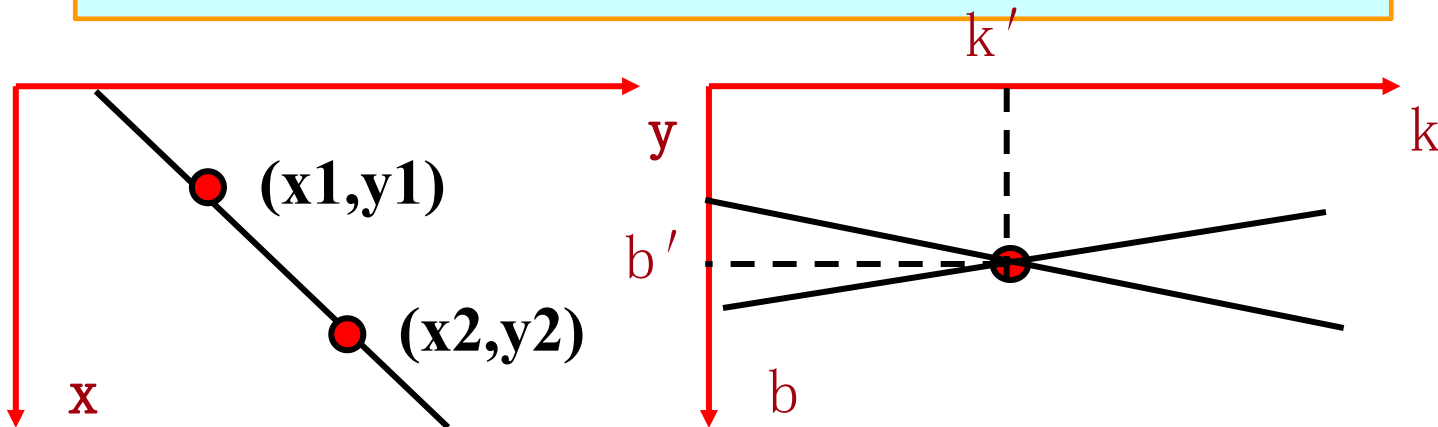
11.2.2 Hough变换

边界分割

(2) 检测直线

■ 原理

对于原图中每一点，在参数空间确定一条直线，经过直线最多的点为原图中直线的参数





11.2.2 Hough变换

边界分割

(2) 检测直线

■ 算法

- 假设原图像已经处理为二值边缘图像，扫描图中的每一个像素点：
 - ◆ 背景点，不作任何处理
 - ◆ 目标点，确定直线： $b = -xk + y$ ，参数空间上的对应直线上所有的值累加1
- 循环扫描所有点，重复上述操作
- 参数空间上累计值最大的点 (k^*, b^*) 为所求直线参数，按照该参数绘制直线

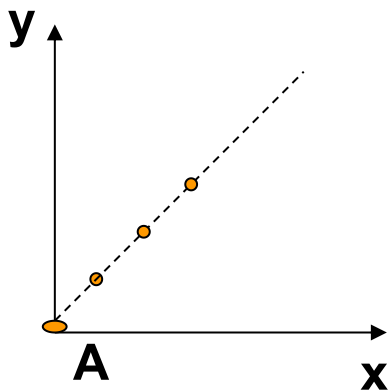


11.2.2 Hough变换

例子说明

边界分割

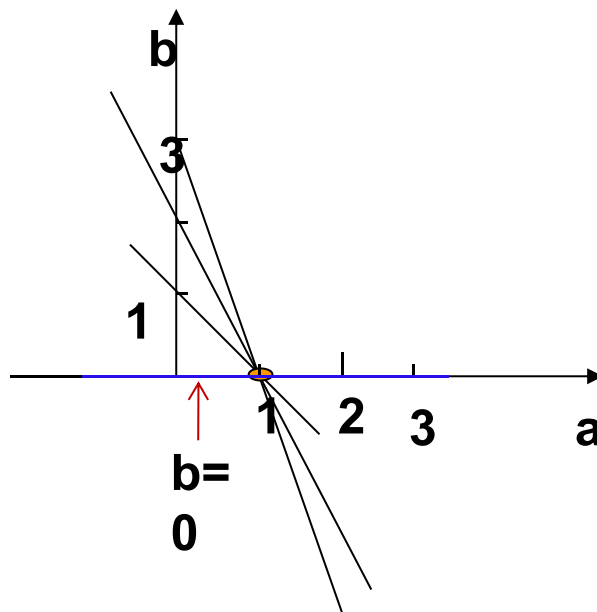
(2) 检测直线



- A. (0,0)
- B. (1,1)
- C. (2,2)
- D. (3,3)

$$y = ax + b,$$

$$b = -xa + y$$



$$b=0;$$

$$b=-a+1;$$

$$b=-2a+2;$$

$$b=-3a+3$$



11.2.2 Hough变换

边界分割

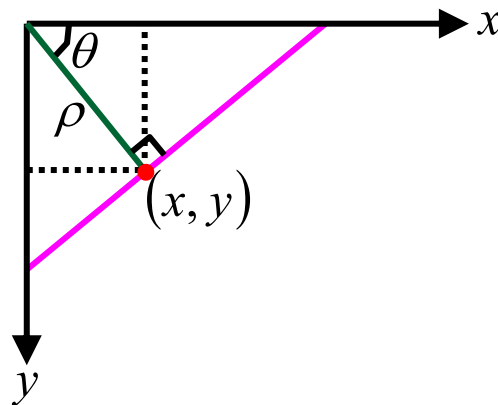
(2) 检测直线

■ 极坐标形式

- 直线方程 $y=kx+b$ 对垂直线不起作用，采用极坐标形式

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

$$\begin{cases} x = \rho \cos \theta \\ y = \rho \sin \theta \end{cases}$$





11.2.2 Hough变换

边界分割

(2) 检测直线

■ 极坐标形式

- xy 空间一条确定的直线对应 $\rho\theta$ 参数空间的一个点
- 参数空间的一条正弦曲线对应 xy 空间的一个点
- xy 空间一条直线上的 n 个点，对应 $\rho\theta$ 参数空间经过一个公共点的 n 条正弦曲线
- 对于原图中每一点，在参数空间确定一条正弦曲线，经过曲线最多的点为原图中直线的参数



11.2.2 Hough变换

边界分割

(2) 检测直线

■ 函数

- $[H, THETA, RHO] = \text{hough}(BW)$
- $[H, THETA, RHO] = \text{hough}(BW, PARAM1, VAL1, PARAM2, VAL2)$: 对输入图像BW进行Hough变换
- $LINES = \text{houghlines}(BW, THETA, RHO, PEAKS)$
- $LINES = \text{houghlines}(..., PARAM1, VAL1, PARAM2, VAL2)$
根据Hough变换的结果提取图像BW中的线段
- $PEAKS = \text{houghpeaks}(H, NUMPEAKS)$
- $PEAKS = \text{houghpeaks}(..., PARAM1, VAL1, PARAM2, VAL2)$
提取Hough变换后参数平面的峰值点



11.2.2 Hough变换

边界分割

(2) 检测直线

■ 例程

```
Image=rgb2gray(imread('houghsource.bmp'));  
bw=edge(Image,'canny');  
figure,imshow(bw);  
[h,t,r]=hough(bw,'RhoResolution',0.5,  
               'ThetaResolution',0.5);  
figure,imshow(imadjust(mat2gray(h)),'XData',t,  
               'YData',r,'InitialMagnification','fit');  
xlabel('\theta'),ylabel('\rho');  
axis on,axis normal,hold on;  
P=houghpeaks(h,2);  x=t(P(:,2));  y=r(P(:,1));  
plot(x,y,'s','color','r');
```



11.2.2 Hough变换

边界分割

(2) 检测直线

■ 例程

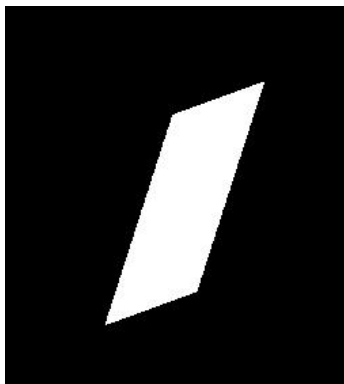
```
lines=houghlines(bw,t,r,P,'FillGap',5,'Minlength',7);  
figure,imshow(Image);  
hold on;  
max_len=0;  
for i=1:length(lines)  
    xy=[lines(i).point1;lines(i).point2];  
    plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','g');  
    plot(xy(1,1),xy(1,2),'x','LineWidth',2,'Color','y');  
    plot(xy(2,1),xy(2,2),'x','LineWidth',2,'Color','r');  
end
```

11.2.2 Hough变换

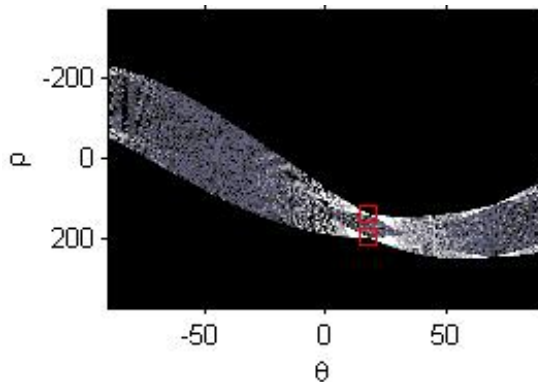
边界分割

(2) 检测直线

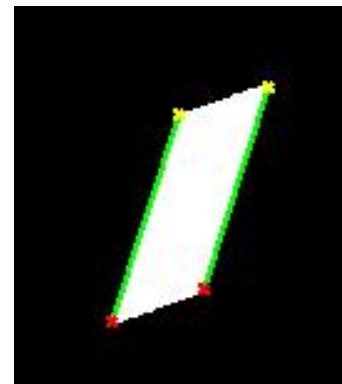
■ 例程



原始图像



$\rho\theta$ 参数空间



检测结果



11.2.2 Hough变换

边界分割

(3) 检测圆

■ 原理

$$\text{圆方程 } (x-a)^2 + (y-b)^2 = r^2$$

- xy 空间一个圆对应三维参数空间一个点 (a,b,r)
- xy 空间圆上一个点 (x,y) 对应参数空间一条曲线
- xy 空间圆上 n 个点对应参数空间 n 条相交于一点的曲线

对于原图中每一点，在参数空间确定一条曲线，经过曲线最多的点为原图中圆的参数



11.2.2 Hough变换

边界分割

(3) 检测圆

■ 算法

- 假设原图像已经处理为二值边缘图像，扫描图中的每一个像素点：
 - ◆ 背景点，不作任何处理
 - ◆ 目标点，确定曲线：参数空间上的对应曲线上所有的值累加1
- 循环扫描所有点
- 参数空间上累计值为最大的点 (a^*, b^*, r^*) 为所求圆参数
- 按照该参数与原图像同等大小的空白图像上绘制圆

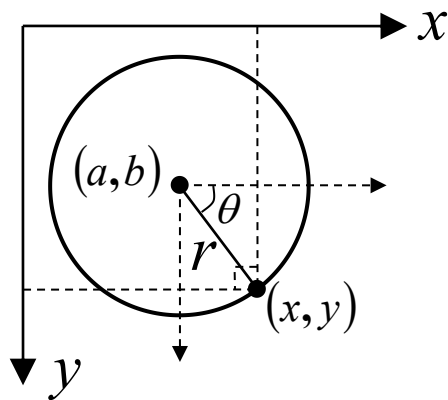
11.2.2 Hough变换

边界分割

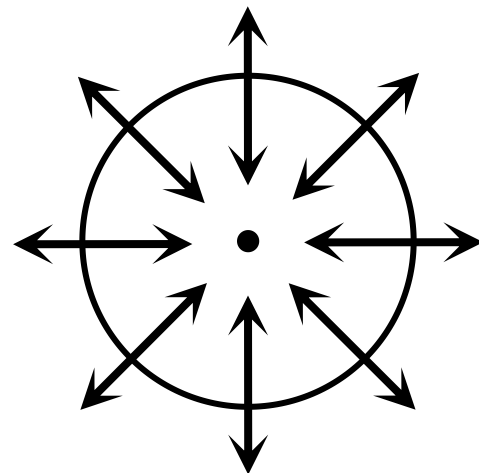
(3) 检测圆

■ 简化运算

三维参数空间，计算量大，可以采样其他形式，如极坐标式，进行进一步简化。



$$\begin{cases} x = a + r \cos \theta \\ y = b + r \sin \theta \end{cases}$$





11.2.2 Hough变换

边界分割

(4) 检测任意曲线

Hough变换可以推广到具有解析形式 $f(x,a)=0$ 的任意曲线， x 表示图像点， a 表示参数向量

■ 过程

- 初始化参数空间 $A[a]$
- 对每个边缘像素 x 确定 a ，使得 $f(x,a)=0$ ，并令 $A[a] += 1$
- A 的局部最大值对应图像中曲线参数



11.2.3边界跟踪

边界分割

(1) 跟踪方法

- 根据某些严格的“**探测准则**”找出目标物体轮廓上的像素，即确定边界的起始搜索点；
- 再根据一定的“**跟踪准则**”找出目标物体上的其他像素，直到符合跟踪终止条件。
- **由二维图像变为一维的点序列**



11.2.3边界跟踪

边界分割

(2) 函数

- **B = bwboundaries(BW):** 搜索二值图像BW的外边界和内边界。
- **B = bwboundaries(BW,CONN,OPTIONS)**
- **[B,L,N,A] = bwboundaries(...)**
- **B = bwtraceboundary(BW,P,FSTEP):** 跟踪二值图像BW中目标轮廓



11.2.3边界跟踪

边界分割

(3) 例程

读取一幅灰度图像，对其进行阈值分割，并对分割的二值图像进行边界跟踪

```
Image=im2bw(imread('algae.jpg'));  
Image=1-Image;  
[B,L]=bwboundaries(Image);  
figure,imshow(L),title('划分的区域');  
hold on;  
for i=1:length(B)  
    boundary=B{i};  
    plot(boundary(:,2),boundary(:,1),'r','LineWidth',2);  
end
```



11.2.3边界跟踪

边界分割

(3) 例程



原始图像

边界跟踪



11.3 区域分割



一般认为，**同一个区域内的像素点具有某种相似性**，如灰度、颜色、纹理等，区域分割即是根据特定区域与其他背景区域特性上的不同来进行图像分割的技术。

10.3.1 区域生长

10.3.3 区域分裂

10.3.2 区域合并

10.3.4 区域合并分裂



11.3.1 区域生长

区域分割

(1) 原理

■ 设计思路

拟把图像划分成一系列区域，确定每个区域区别于其他区域的特征，由此生成相似性判据；从图像某个像素开始，判断其应该属于哪个区域，使区域逐渐变大，直到被比较的像素与区域像素具有显著差异为止。



11.3.1 区域生长

区域分割

(1) 原理

■ 实现方法

在每个要分割的区域内确定一个**种子点**，判断种子像素周围邻域**是否有与种子像素相似的像素**，若有将新的像素包含在区域内，并作为新的种子继续生长，直到没有满足条件的像素点时**停止生长**。



11.3.1 区域生长

区域分割

(1) 原理

■ 关键技术

□ 种子点的选取

选择待提取区域的**具有代表性的点**，可以是单个像素，也可以是包括若干个像素的子区域，根据具体问题，利用先验知识来选择。

□ 生长准则的确定（相似性准则）

一般根据图像的特点，采用与种子点的距离度量（彩色、灰度、梯度等量之间的距离）。

□ 区域停止生长的条件

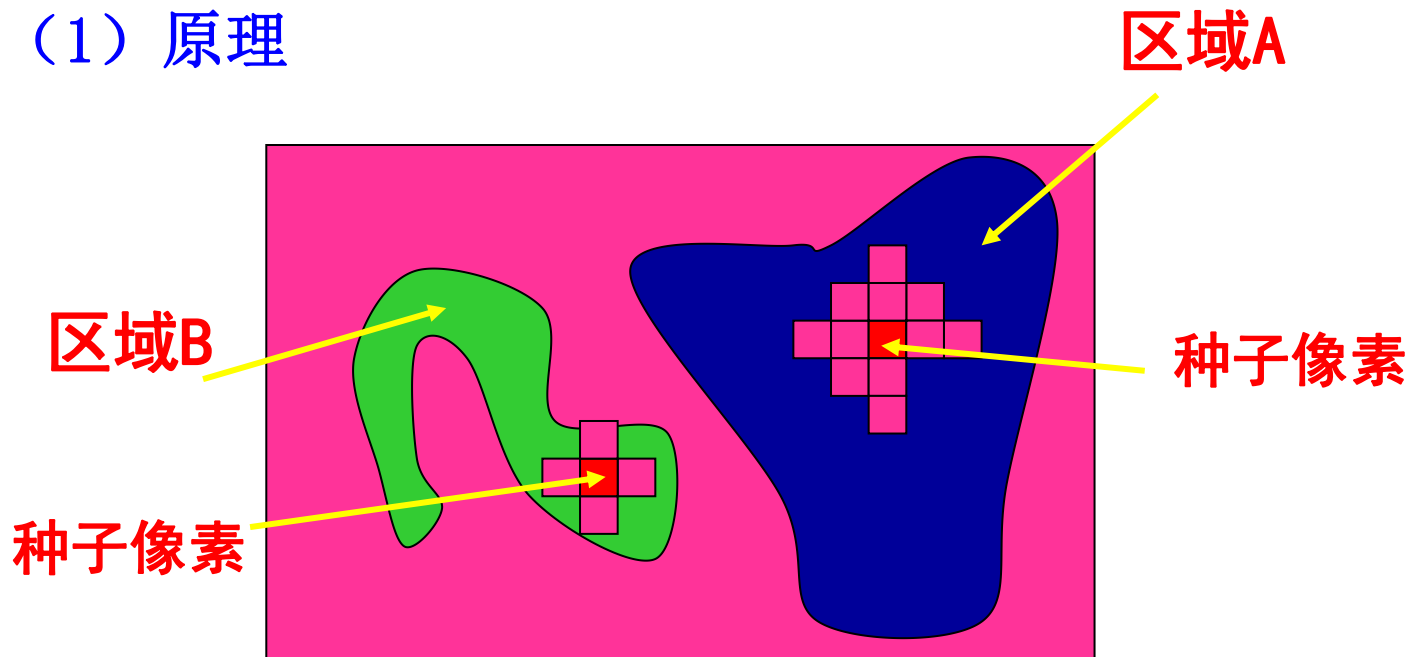
区域大小、迭代次数或区域饱和等条件。



11.3.1 区域生长

区域分割

(1) 原理





11.3.1 区域生长

区域分割

(2) 示例

(1) 种子选择准则：最亮的点；

(2) 相似性准则：新加入像素值与已生长的区域的平均值之差的绝对值小于 2（不等于），且为4连通；

(3) 终止准则：没有像素加入。

5	5	8	6
4	8	9	7
2	2	8	3
3	3	3	3



11.3.1 区域生长

区域分割

(2) 示例

第二步

5	5	8	6
4	8	9	7
2	2	8	3
3	3	3	3



11.3.1 区域生长

区域分割

(2) 示例

第三步

5	5	8	6
4	8	9	7
2	2	8	3
3	3	3	3

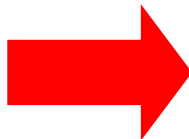


11.3.1 区域生长

区域分割

(2) 示例

1	0	4	6	5	1
1	0	4	6	6	2
0	1	5	5	5	1
0	0	5	6	5	0
0	0	1	6	0	1
1	0	1	2	1	1



1	1	2	2	2	1
1	1	2	2	2	1
1	1	2	2	2	1
1	1	2	2	2	1
1	1	1	2	1	1
1	1	1	1	1	1

种子点为：(2, 2)

相似性准则：灰度值差小于2

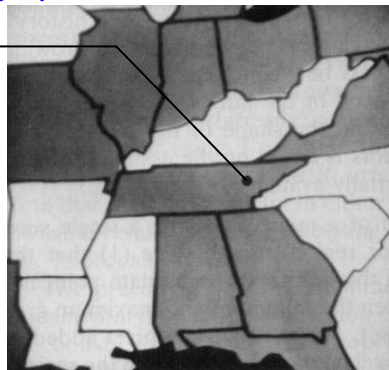
邻域选择：4邻域

11.3.1 区域生长

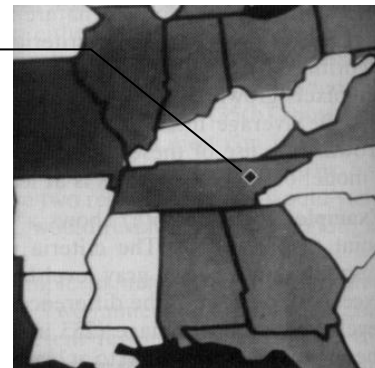
区域分割

(2) 示例

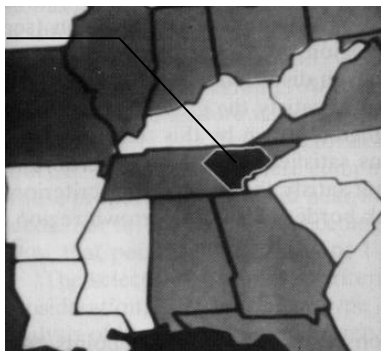
原始图
像及种
子像素
点



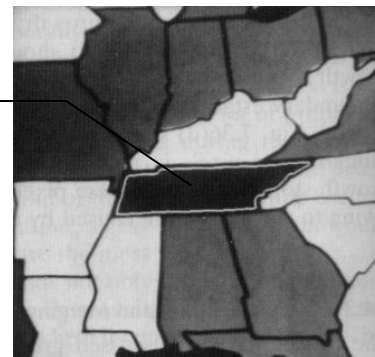
开始增
长阶段
的结果



中间结
果



最后结
果



11.3.1 区域生长

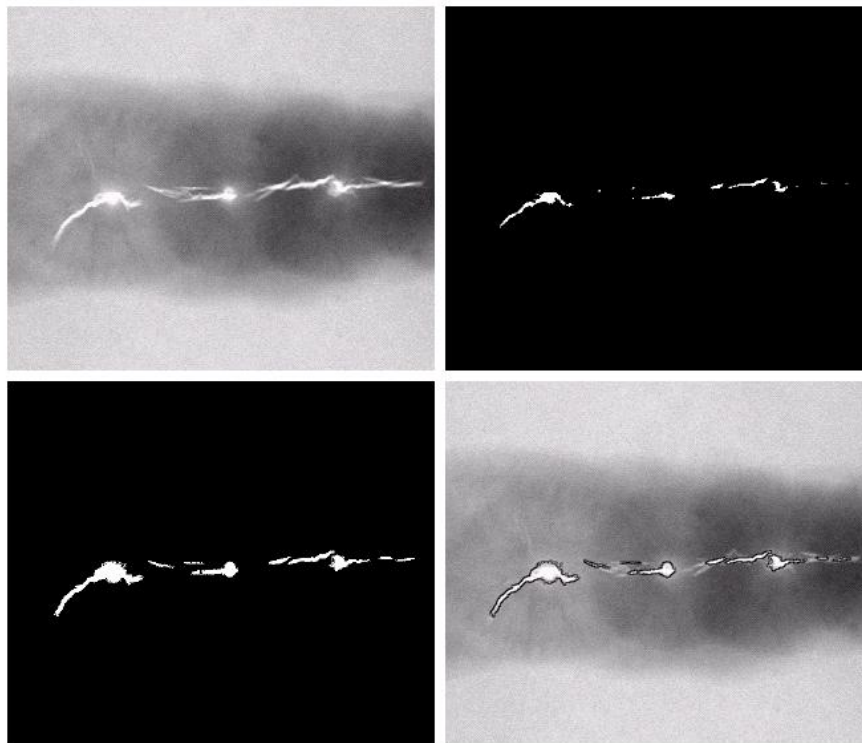
区域分割

(2) 示例

a b
c d

FIGURE 10.40

(a) Image showing defective welds. (b) Seed points. (c) Result of region growing. (d) Boundaries of segmented defective welds (in black). (Original image courtesy of X-TEK Systems, Ltd.).



11.3.1 区域生长

区域分割

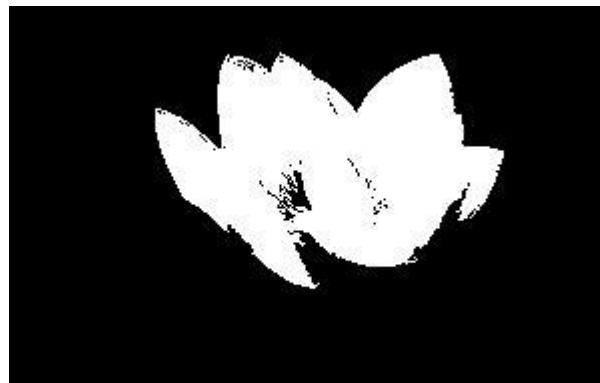
(3) 例程

对图像进行区域生长。交互式选取种子，生长准则采用“待测像素点与区域的平均灰度差小于40”，8邻域范围生长，停止生长条件为区域饱和。

程序见教材【例10.10】



原图



区域生长



11.3.2 区域合并

区域分割

(1) 原理

假设图像已经分为若干个小区域，合并具有相似性的相邻区域。

极端的情况，可以认为每个像素为一个小区域，把具有相似性的像素合并到一个区域内。

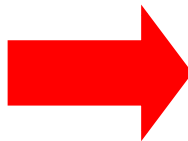


11.3.2 区域合并

区域分割

(2) 示例

1	0	4	6	5	1
1	0	4	6	6	2
0	1	5	5	5	1
0	0	5	6	5	0
0	0	1	6	0	1
1	0	1	2	1	1



1	0	4	6	5	1
1	0	4	6	6	2
0	1	5	5	5	1
0	0	5	6	5	0
0	0	1	6	0	1
1	0	1	2	1	1



11.3.2 区域合并

区域分割

(3) 例程

通过区域合并将示例中的小图像分割为两个区域

■ 设计思路

- 初始化：每个像素为一个小区域；相似性准则采用：相邻区域灰度均值差 ≤ 2 ；左上角第一个点设为区域1，其余为0，表示未标记。
- 第一次扫描图像：从左到右，从上到下，判断每一点与其左上、上、左邻点的灰度距离，三个距离中最小的若符合合并规则，将对应邻点的标记赋予当前点；若没有相似的点，则赋予当前点新的标记。



11.3.2 区域合并

区域分割

(3) 例程

■ 设计思路

- 再次扫描图像：若某一像素点上、左邻点标记不一致，但当前点和其中一个邻点标记一致，则判断两个区域是否是同一个，若是，则将两个区域标记修改为较小的一个，即区域合并。



11.3.2 区域合并

区域分割

(3) 例程

- 程序

见教材【例10.11】

- 效果

同示例

一种自下而上的方法，某些区域一旦合并，即使与后来的区域相似性并不好，也无法去除。



11.3.3 区域分裂

区域分割

(1) 原理

认为整幅图像是一个完整区域，检验整幅区域是否具有**一致性**，不具有时，分裂为几个小区域；然后再检测小区域的一致性，不具有时进一步分裂；重复这个过程直到每个区域都具有**一致性**。

每个区域可以具有不同的一致性

通常在分裂区域时，采用**一分为4**的方法



11.3.3 区域分裂

区域分割

(2) 示例

1	1	0	1	1	0	0	1
0	1	2	0	1	1	1	0
0	0	6	7	1	0	0	1
1	6	7	5	6	7	1	1
0	7	6	6	6	0	1	1
0	7	6	5	7	1	0	0
1	1	0	1	1	1	1	0
0	1	1	1	1	1	0	1

- 初始化及准则、方法的确定
 - 区域内最大灰度值与最小灰度值之差 ≤ 2
 - 采用一分为四的分裂方法

分裂

①				②			
1	1	0	1	1	0	0	1
0	1	2	0	1	1	1	0
0	0	6	7	1	0	0	1
1	6	7	5	6	7	1	1
0	7	6	6	6	0	1	1
0	7	6	5	7	1	0	0
1	1	0	1	1	1	1	0
0	1	1	1	1	1	0	1
③				④			



11.3.3 区域分裂

区域分割

(2) 示例

对四个小区域分别计算最大与最小灰度差，与阈值2比较，每个区域均需分裂

1	1	0	1	1	0	0	1
0	1	2	0	1	1	1	0
0	0	6	7	1	0	0	1
1	6	7	5	6	7	1	1
0	7	6	6	6	0	1	1
0	7	6	5	7	1	0	0
1	1	0	1	1	1	1	0
0	1	1	1	1	1	0	1

依此类推，直至所有的区域都不能再分裂

1	1	0	1	1	0	0	1
0	1	2	0	1	1	1	0
0	0	6	7	1	0	0	1
1	6	7	5	6	7	1	1
0	7	6	6	6	0	1	1
0	7	6	5	7	1	0	0
1	1	0	1	1	1	1	0
0	1	1	1	1	1	0	1



11.3.3 区域分裂

区域分割

(3) 例程

■ 函数

S = qtdecomp(I)

S = qtdecomp(I,THRESHOLD)

S = qtdecomp(I,THRESHOLD,MINDIM)

**S = qtdecomp(I,THRESHOLD,
[MINDIM MAXDIM])**

S = qtdecomp(I,FUN)



11.3.3 区域分裂

区域分割

(3) 例程

■ 程序

```
Image=imread('cameraman.jpg');  
S=qtdecomp(Image,0.27);  
blocks=repmat(uint8(0),size(S));  
for dim=[256 128 64 32 16 8 4 2 1]  
    numblocks=length(find(S==dim));  
    if(numblocks>0)  
        values=repmat(uint8(1),[dim dim numblocks]);  
        values(2:dim,2:dim,:)=0;  
        blocks=qtsetblk(blocks,S,dim,values);  
    end  
end
```

11.3.3 区域分裂

区域分割

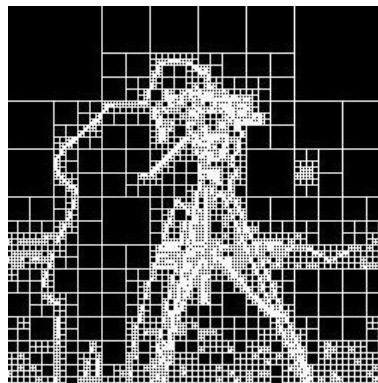
(3) 例程

■ 程序

```
blocks(end,1:end)=1;    blocks(1:end,end)=1;  
imshow(Image);          figure,imshow(blocks,[]);
```



原始图像



四叉树分解



11.3.4 区域分裂合并

区域分割

(1) 原理

结合合并、分裂方法，将原图分成若干个子块，检测子块是否具有一致性，不具有则分裂该子块，若某些子块具有相似性，合并这些子块



11.3.4 区域分裂合并

区域分割

(2) 步骤

- 将原图分为四个相等的子块，计算子块区域是否具有**一致性**（例如灰度均值或方差）
- 如果子块不具有**一致性**（例如：方差大于设定的**阈值**）分裂该块
- 对不需要分裂的子块进行比较，具有相似性的子块**合并**
- 重复上述过程，直到不再需要分裂或合并



11.3.4 区域分裂合并

区域分割

(3) 示例

$$f = \begin{pmatrix} 1 & 3 & 7 & 8 \\ 1 & 2 & 8 & 9 \\ 2 & 1 & 2 & 7 \\ 3 & 8 & 8 & 9 \end{pmatrix} \xrightarrow{\text{green arrow}} \begin{matrix} f_{11} = \begin{pmatrix} 1 & 3 \\ 1 & 2 \end{pmatrix}, f_{12} = \begin{pmatrix} 7 & 8 \\ 8 & 9 \end{pmatrix} \\ f_{13} = \begin{pmatrix} 2 & 1 \\ 3 & 8 \end{pmatrix}, f_{14} = \begin{pmatrix} 2 & 7 \\ 8 & 9 \end{pmatrix} \end{matrix}$$

$$\mu_{11} = 1.75, \sigma_{11}^2 = 0.6875$$

$$\mu_{12} = 8, \sigma_{12}^2 = 0.5$$

$$\mu_{13} = 3.5, \sigma_{13}^2 = 7.25$$

$$\mu_{14} = 6.5, \sigma_{14}^2 = 7.25$$

$\sigma_{Th} = 1$

1	3	7	8
1	2	8	9
2	1	2	7
3	8	8	9

$\mu_{Th} = 2$

1	3	7	8
1	2	8	9
2	1	2	7
3	8	8	9

11.3.4区域分裂合并

区域分割

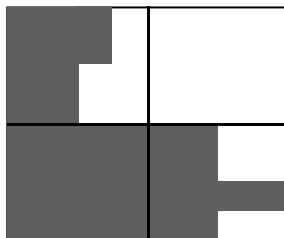
(3) 示例



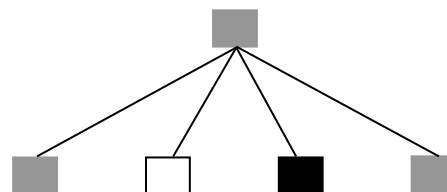
(a) 原始图像

1	2
3	4

分裂序号



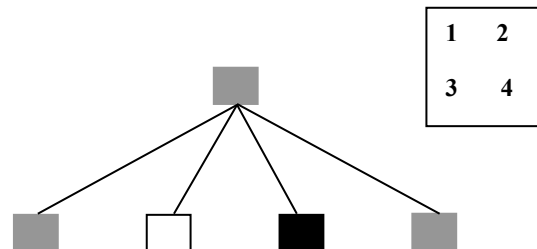
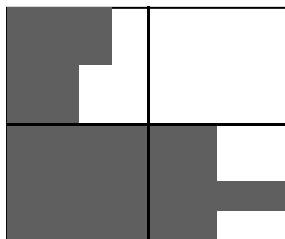
(b) 第一次分裂成4个区域



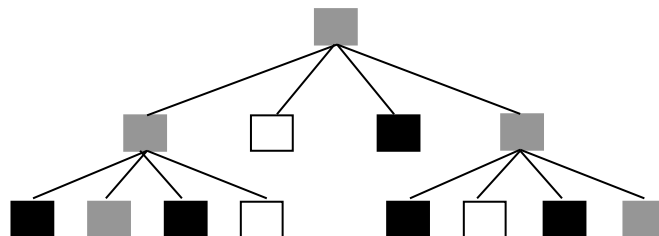
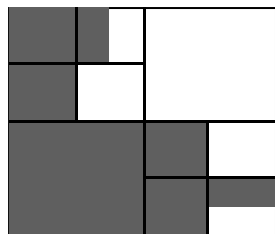
11.3.4区域分裂合并

区域分割

(3) 示例



(b) 第一次分裂成4个区域

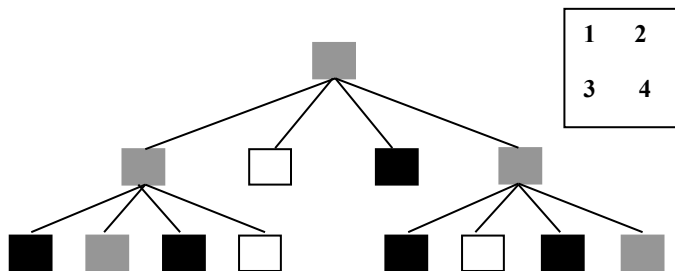
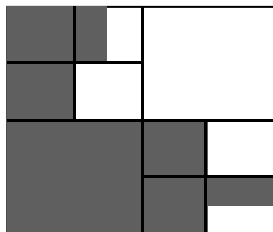


(c) 第二次分裂成十个区域

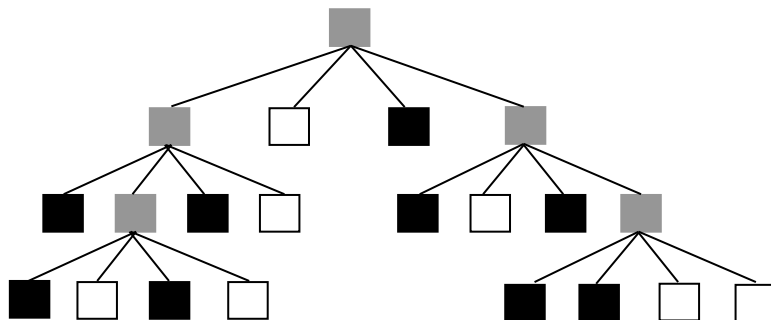
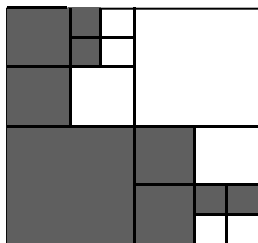
11.3.4区域分裂合并

区域分割

(3) 示例



(c) 第二次分裂成十个区域



(d) 最后一次分裂成十六个区域



11.3.4 区域分裂合并

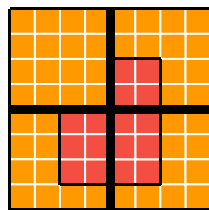
区域分割

对于灰度图象的一些可以选择的分裂-合并准则：

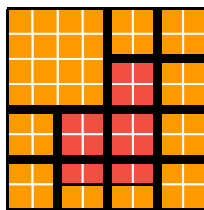
- (1) **同一区域**中最大灰度值与最小灰度值之差或方差小于某选定的阈值；
- (2) **两个区域**的平均灰度值之差及方差小于某个选定的阈值；
- (3) **两个区域**的灰度分布函数之差小于某个选定的阈值；
- (4) **两个区域**的某种图像统计特征值的差小于等于某个阈值。

11.3.4 区域分裂合并

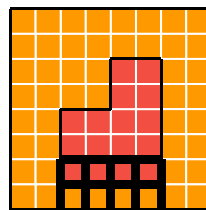
区域分割



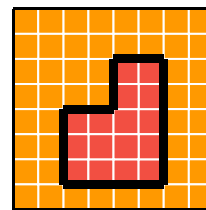
(a)



(b)



(c)



(d)

图中红色区域为目标，其它区域为背景，它们都具有常数灰度值

对整个图像 R ， $P(R)=FALSE$ ，（ $P(R)=TRUE$ 代表在 R 中的所有象素都具有相同的灰度值），所以先将其分裂成如图(a)所示的四个正方形区域，由于左上角区域满足 P ，所以不必继续分裂，其它三个区域继续分裂而得到(b)，此时除包括目标下部的两个子区域外，其它区域都可分别按目标和背景合并。对下面的两个子区域继续分裂可得到(c)，因为此时所有区域都已满足 P ，所以最后一次合并可得到(d)的分割结果。



11.4 基于聚类的图像分割



(1) 原理

把图像分割看做**对像素进行分类**的问题，把像素表示成特征空间的点，采用聚类算法把这些点划分为不同类别，对应原图则是实现对像素的分组，分组后利用“连通成分标记”找到连通区域。



11.4 基于聚类的图像分割



(2) 关键技术

- 如何把像素表示成特征空间中的点

用向量来代表像素或像素周围邻域，向量的元素为与像素相关的特征，根据图像的具体情况，判断待分割区域的共性来设计。

- 聚类方法



11.4 基于聚类的图像分割



(3) K均值聚类

■ 原理

首先确定 K 个初始聚类中心，然后根据各类样本到聚类中心的距离平方和最小的准则，不断调整聚类中心，直到聚类合理。



11.4 基于聚类的图像分割



(3) K均值聚类

■ 步骤

- 任选 K 个初始聚类中心
- 逐一将样本按最小距离原则分配给 K 个聚类中心

若 $\|x - \mu_j(m)\| < \|x - \mu_i(m)\|, i = 1, 2, \dots, K, i \neq j$ 则 $x \in \omega_j(m)$

$\omega_j(m)$ 为第 m 次迭代时，聚类中心为 $\mu_j(m)$ 的聚类域

- 计算新的聚类中心

$$\mu_i(m+1) = \frac{1}{N_i} \sum_{x \in \omega_i(m)} x \quad i = 1, 2, \dots, K$$



11.4 基于聚类的图像分割



(3) K均值聚类

■ 步骤

□ 判断算法是否收敛

若 $\mu_i(m+1) = \mu_i(m)$ $i = 1, 2, \dots, K$ 算法收敛

否则，重新分配，进行下一次迭代。



11.4 基于聚类的图像分割



(3) K均值聚类

■ 函数

[IDX,C] = kmeans(X,K);

**[IDX,C,SUMD,D] =kmeans(..., 'PARAM1',val1,
'PARAM2',val2, ...)**



11.4 基于聚类的图像分割



(3) K均值聚类

■ 例程

对一幅苹果图像，利用色彩信息，实现聚类分割

```
Image=imread('fruit.jpg');
```

```
figure,imshow(Image);
```

```
hsv=rgb2hsv(Image);
```

```
h=hsv(:,:,1);
```

```
h(h>330/360)=0;
```

```
training=h(:);
```



11.4 基于聚类的图像分割



(3) K均值聚类

■ 例程

```
startdata = [0;60/360;120/360;180/360;  
             240/360;300/360];  
[IDX,C]= kmeans(training,6,'Start',startdata);  
idbw = (IDX == 1);  
template = reshape(idbw, size(h));  
figure,imshow(template),title('K均值聚类分割');
```

11.4 基于聚类的图像分割

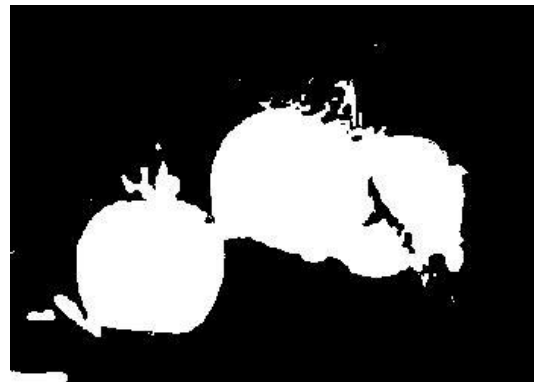


(3) K均值聚类

■ 例程



原始图像



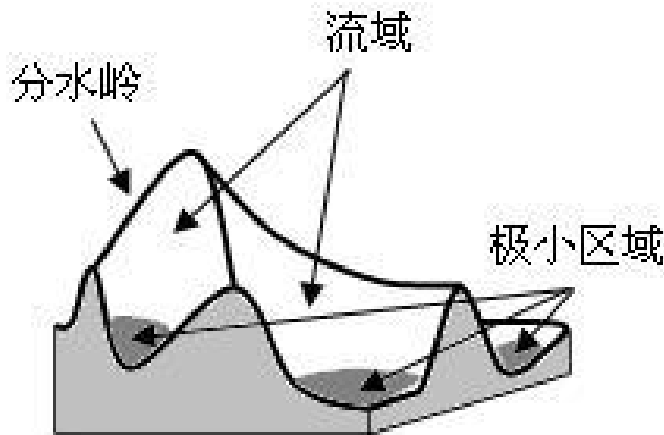
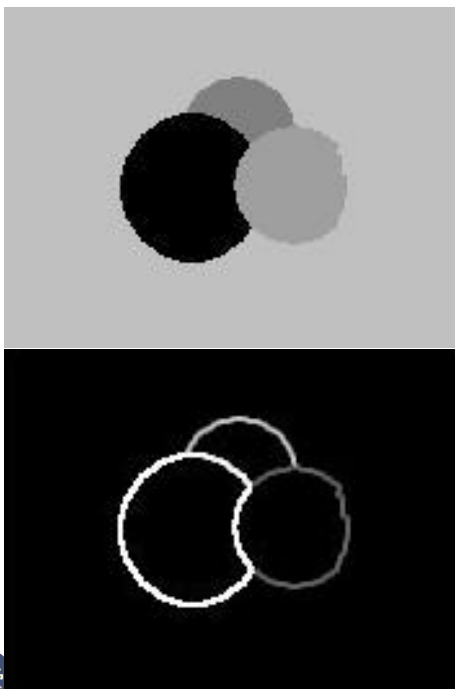
K均值聚类分
割

11.5 分水岭分割



(1) 原理

■ 地形图与梯度图像



梯度图像中各区域内部对应极小区域，边界对应高灰度，即**分水岭**。确定分水岭，即确定区域边界，实现分割



11.5 分水岭分割



(2) 算法

■ 涨水法

设水从谷底上涌，水位逐渐升高。若水位高过山岭，不同流域的水就会汇合。在不同流域中的水面将要汇合到一起时，在中间筑起一道堤坝，阻止水汇合，且堤坝高度随着水面上升而增高。当所有山峰都被淹没时，露出水面的只剩下堤坝，且将整个平面分成了若干个区域



11.5 分水岭分割



(2) 算法

■ 相关定义

- 梯度图像 $g(x, y)$
- 梯度图像中的极小区域 M_1, M_2, \dots, M_r
- 流域 $C(M_i)$
- 谷底和山峰 $\min g(x, y) \max g(x, y)$
- 涨水从谷底开始，单灰值递加，第 n 步时水深为 n ，定义： $T(n) = \{(x, y) | g(x, y) < n\}$
- 水深为 n 时，流域所对应水平面区域
$$C_n(M_i) = C(M_i) \cap T(n)$$
- 第 n 步流域溢流部分的并 $C(n)$

11.5 分水岭分割

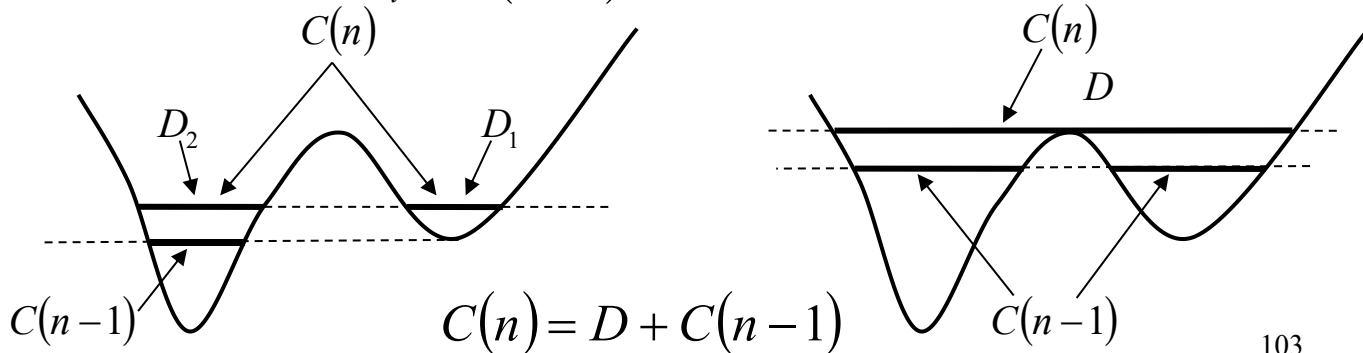


(2) 算法

■ $C(n)$ 的递归计算

- 初始: $C(\min+1) = T(\min+1)$
- $n = n + 1$: 确定 $T(n)$ 中的连通成分 D_i , 直至山峰

$D_i \cap C(n-1)$ 有三种情况



不加堤坝

加堤坝





11.5 分水岭分割



(2) 算法

■ 步骤

- 计算梯度图像及其值的最小 min 和最大 max
- 初始化 $n = min + 1$, 并标识出目前的极小区域
$$C(min + 1) = T(min + 1) : \{g(x, y) < min + 1\}$$
- $n = n + 1$, 确定 $T(n)$ 中的连通成分: $D_i, i = 1, 2, \dots$
求 $D_i \cap C(n - 1)$, 判断属于哪种情况, 确定 $C(n)$
- 重复第三步, 直到 $C(max + 1)$

11.5 分水岭分割

(3) 示例

对 f 进行分水岭分割： $f =$

- 采用Prewitt梯度算子计算梯度图像

3	3	3	1	1	1
3	1	3	1	1	1
3	3	3	1	1	1
1	1	1	3	3	3
1	1	1	3	1	3
1	1	1	3	3	3

$$g = \begin{bmatrix} 4 & 2 & 6 & 6 & 0 & 0 \\ 2 & 0 & 4 & 6 & 0 & 0 \\ 6 & 4 & 0 & 4 & 6 & 6 \\ 6 & 6 & 4 & 0 & 4 & 6 \\ 0 & 0 & 6 & 4 & 0 & 2 \\ 0 & 0 & 6 & 6 & 2 & 4 \end{bmatrix}$$

- 求梯度图像最大最小值

$$\min = 0, \max = 6$$

涨水过程中： $n=1 \sim 7$



11.5 分水岭分割



(3) 示例

■ $n=1$

$$T(1) = \{(x, y) \mid g(x, y) < 1\} \quad C(1) = T(1)$$

极小区域三个: M_1, M_2, M_3

■ $n=2$

$$T(2) = \{(x, y) \mid g(x, y) < 2\}$$

三个连通成分 $D_i, i = 1, 2, 3$

$$D_1 \cap C(1) = C_1(M_1) \quad D_2 \cap C(1) = C_1(M_2)$$

$$D_3 \cap C(1) = C_1(M_3)$$

$$C(2) = C(1) + D_{i=1,2,3} = C(1)$$

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

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



11.5 分水岭分割



(3) 示例

■ $n=3$

$$T(1) = \{(x, y) \mid g(x, y) < 3\}$$

三个连通成分 $D_i, i = 1, 2, 3$

$$D_1 \cap C(2) = C_2(M_1) \quad D_2 \cap C(2) = C_2(M_2)$$

$$D_3 \cap C(2) = C_2(M_3) \quad C(3) = C(2) + D_{i=1,2,3}$$

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

■ $n=4$

结果同 $n=3$



11.5 分水岭分割



(3) 示例

■ $n=5$

$$T(1) = \{(x, y) \mid g(x, y) < 5\}$$

一个连通成分D

$$D \cap C(4) = C_4(M_1) \cup C_4(M_2) \cup C_4(M_3)$$

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

属于第三种情况，三个极小区域即将连通
在D中加堤坝，黑色底纹的点，其余阴影点为C(5)

■ $n=6$

结果同 $n=5$

11.5 分水岭分割



(3) 示例

■ $n=7$

$$T(1) = \{(x, y) \mid g(x, y) < 7\}$$

一个连通成分D

$$D \cap C(6) = C_6(M_1) \cup C_6(M_2) \cup C_6(M_3)$$

至此，所有流域均被淹没，只剩下分水岭露于水面上，分割完成，把最后分割出来的区域依次用编号1、2、3表示，分水岭用0表示，则分割结果为：

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

2	2	2	0	1	1
2	2	2	0	1	1
2	2	2	0	0	0
0	0	0	2	2	2
3	3	0	2	2	2
3	3	0	2	2	2



11.5 分水岭分割



(4) 例程

■ 函数

$L = \text{watershed}(A)$: 对矩阵 A 进行分水岭区域标识, 生成标识矩阵 L 。

$L = \text{watershed}(A, \text{CONN})$



11.5 分水岭分割



(4) 例程

■ 程序

```
image=im2double(rgb2gray(imread('bricks.jpg')));  
figure,imshow(image),title('原图');  
hv=fspecial('prewitt');  
hh=hv.';  
gv=abs(imfilter(image,hv,'replicate'));  
gh=abs(imfilter(image,hh,'replicate'));  
% g=sqrt(gv.^2+gh.^2);  
g=abs(gv)+abs(gh);  
figure,imshow(g),title('梯度图像');
```

11.5 分水岭分割

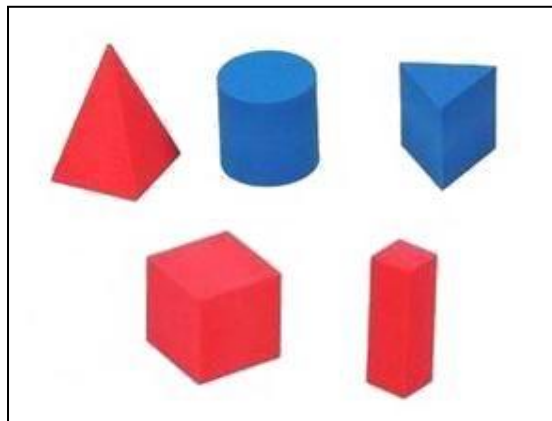


(4) 例程

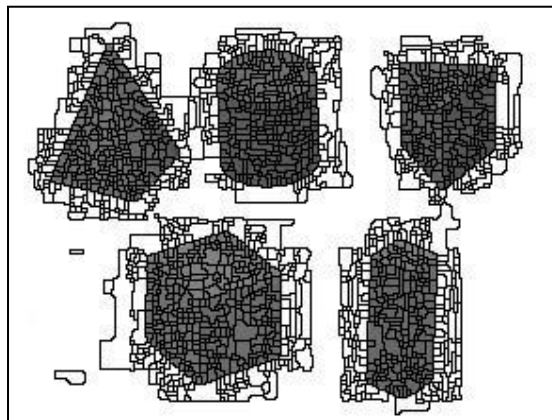
■ 程序

```
L=watershed(g);  
wr=L==0;  
figure,imshow(wr);  
title('分水岭');  
image(wr)=0;  
figure,imshow(image);  
title('分割结果');
```

产生过分割现象



原始图像



分割结果



11.5 分水岭分割



(5) 改进

- 过分割现象产生原因：梯度噪声、量化误差及目标内部细密纹理的影响，导致许多局部的“谷底”和“山峰”，形成很多小区域。
- 减少过分割：在分割前、后加入预处理和后处理步骤：
 - 采用滤波，以减弱噪声干扰，滤除小目标即目标中的细节；
 - 增强图像中的轮廓；
 - 合并一些较小的区域等。

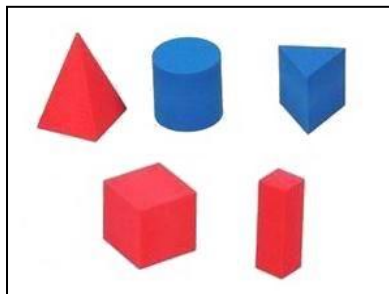
11.5 分水岭分割



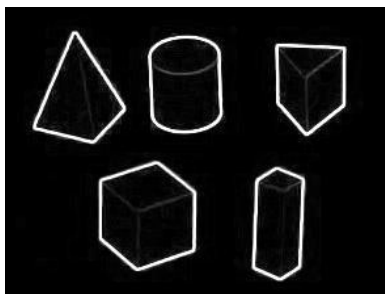
(5) 改进

■ 教材例【10.18】改善过分割策略：

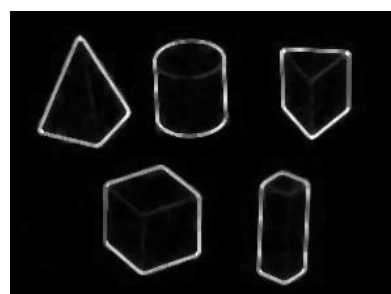
- 对梯度图像进行中值滤波，减少极小区域；
- 在分水岭分割后，采用区域合并方法，将邻近且灰度近似的区域合并起来。



原图



梯度图像

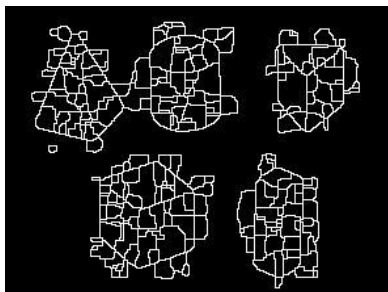


中值滤波

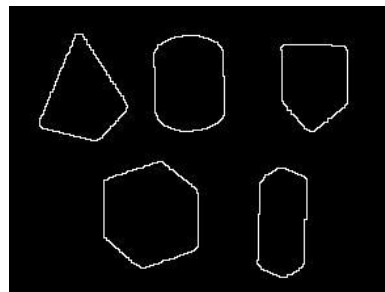
11.5 分水岭分割



(5) 改进



分水岭分割



区域合并



11.6 综合实例



对答题卡图像进行分割

(1) 设计思路

- 要求：将答题卡分割成不同区域。采用所学基础处理方法实现题目要求。
- 操作：
 - 几何校正；
 - 裁切；
 - 上下区域分割；
 - 信息区和答题区分割。

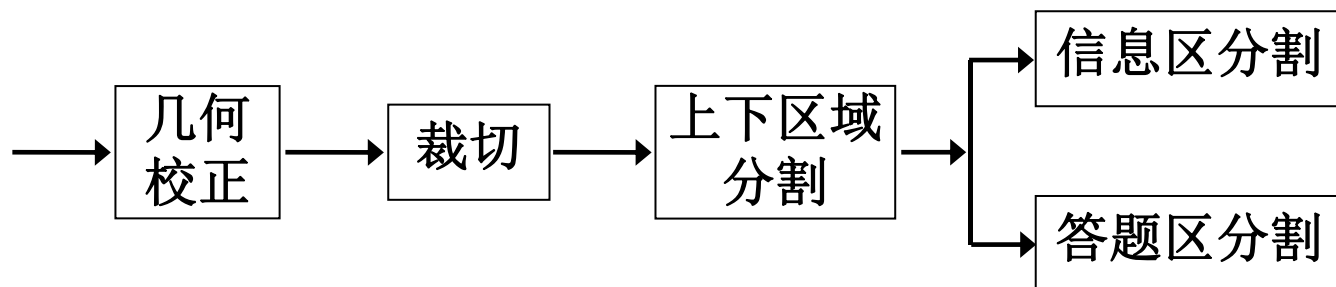


11.6 综合实例



(2) 各模块设计

■ 主程序



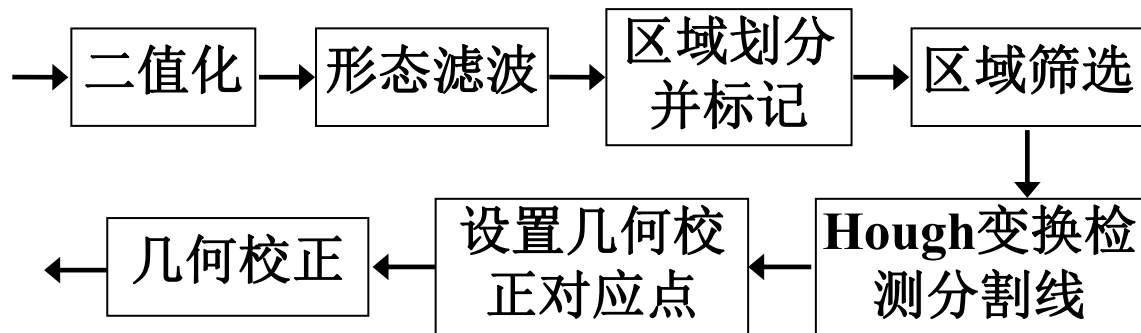


11.6 综合实例



(2) 各模块设计

■ 几何校正



11.6 综合实例



(2) 各模块设计

原始图像



几何校正



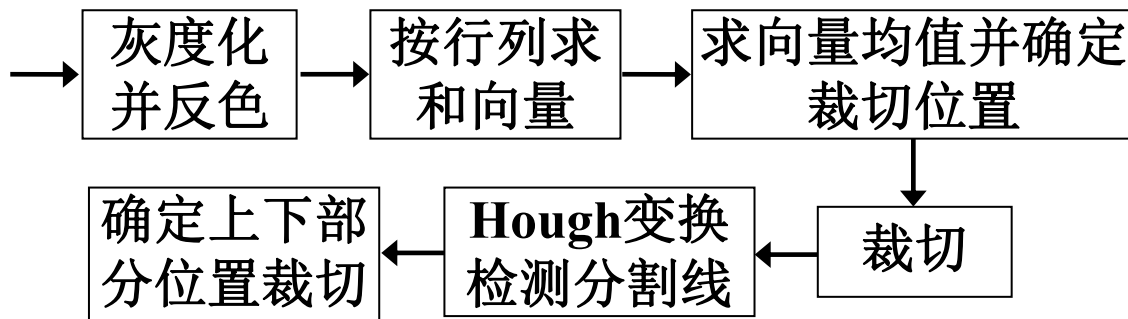


11.6 综合实例



(2) 各模块设计

■ 裁切



(2) 各模块设计

模拟答题卡		准考证号										科 目
姓名	李 同	考 号	0	1	2	3	4	5	6	7	8	政治 语文 数学 物理 化学 英语 历史 地理 生物
			9	0	1	2	3	4	5	6	7	
			1	2	3	4	5	6	7	8	9	
			2	3	4	5	6	7	8	9	0	
			3	4	5	6	7	8	9	0	1	
			4	5	6	7	8	9	0	1	2	
			5	6	7	8	9	0	1	2	3	
			6	7	8	9	0	1	2	3	4	
			7	8	9	0	1	2	3	4	5	
			8	9	0	1	2	3	4	5	6	
			9	0	1	2	3	4	5	6	7	
			0	1	2	3	4	5	6	7	8	
			1	2	3	4	5	6	7	8	9	
			2	3	4	5	6	7	8	9	0	
			3	4	5	6	7	8	9	0	1	
			4	5	6	7	8	9	0	1	2	
			5	6	7	8	9	0	1	2	3	
			6	7	8	9	0	1	2	3	4	
			7	8	9	0	1	2	3	4	5	
			8	9	0	1	2	3	4	5	6	
			9	0	1	2	3	4	5	6	7	
			0	1	2	3	4	5	6	7	8	
			1	2	3	4	5	6	7	8	9	
			2	3	4	5	6	7	8	9	0	
			3	4	5	6	7	8	9	0	1	
			4	5	6	7	8	9	0	1	2	
			5	6	7	8	9	0	1	2		

中考模拟机读答题卡

姓名 张	准考证号 B	准考证号										姓名
		0	1	2	3	4	5	6	7	8	9	
英语 听力 笔试	五选填读 100											数学
	精读读法 100											英语 听力 笔试

裁切出的上半部

[illegible][illegible]

裁切出的下半部

11.6 综合实例



(2) 各模块设计

■ 信息区分割

通过检测边缘来进行分割：canny边缘检测、边缘滤波、边界修复和区域定位四个步骤。

模拟答题卡

姓名: 李同	准考证号: 0123456789	科目: 语文
正确填涂: <input checked="" type="checkbox"/>	错误填涂: <input type="checkbox"/>	缺考标记: <input type="checkbox"/>
填涂要求: <input checked="" type="checkbox"/>	错误填涂: <input type="checkbox"/>	缺考标记: <input type="checkbox"/>

中考模拟机读答题卡

姓名: <input type="text"/>	准考证号: <input type="text"/>	科目: <input type="text"/>
正确填涂: <input checked="" type="checkbox"/>	错误填涂: <input type="checkbox"/>	缺考标记: <input type="checkbox"/>
填涂要求: <input checked="" type="checkbox"/>	错误填涂: <input type="checkbox"/>	缺考标记: <input type="checkbox"/>

中考模拟机读答题卡

姓名: 张三	准考证号: 0123456789	科目: 数学
正确填涂: <input checked="" type="checkbox"/>	错误填涂: <input type="checkbox"/>	缺考标记: <input type="checkbox"/>
填涂要求: <input checked="" type="checkbox"/>	错误填涂: <input type="checkbox"/>	缺考标记: <input type="checkbox"/>

模拟答题卡

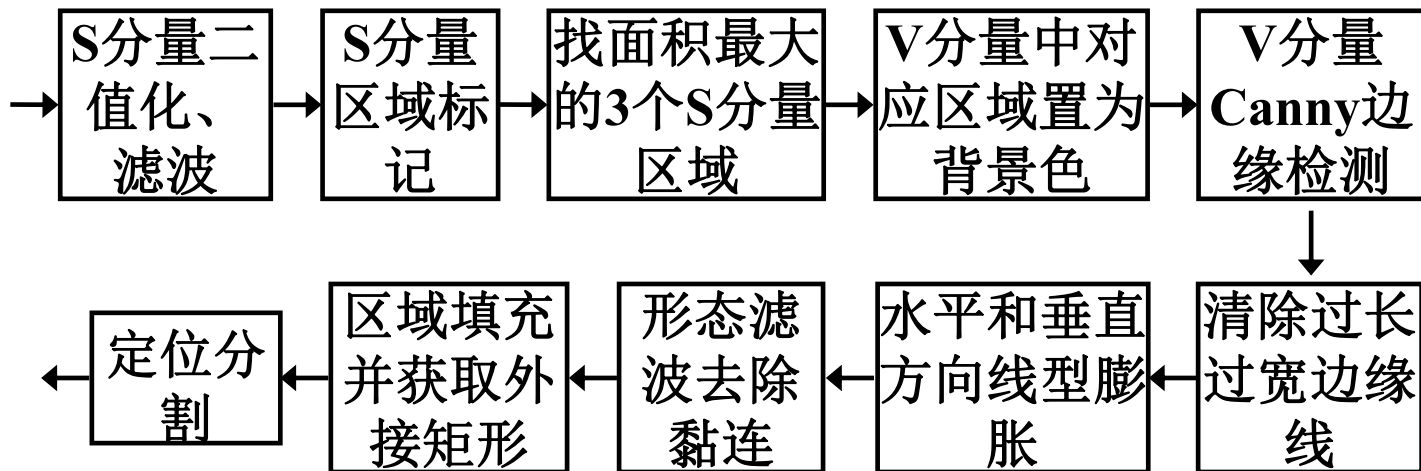
姓名: <input type="text"/>	准考证号: <input type="text"/>	科目: <input type="text"/>
正确填涂: <input checked="" type="checkbox"/>	错误填涂: <input type="checkbox"/>	缺考标记: <input type="checkbox"/>
填涂要求: <input checked="" type="checkbox"/>	错误填涂: <input type="checkbox"/>	缺考标记: <input type="checkbox"/>

11.6 综合实例



(2) 各模块设计

■ 答题区分割

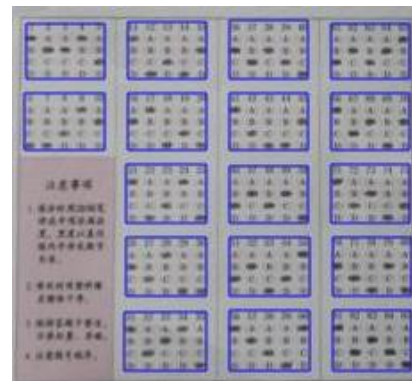
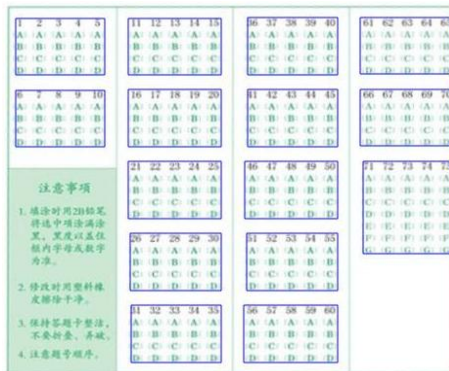


11.6 综合实例



(2) 各模块设计

■ 答题区分割





11.6 综合实例



(3) 分析

- 方案综合利用了灰度化、二值化、形态滤波、Hough变换、几何校正、边缘检测、边界修复、色彩检测、特征检测等技术
- 不足之处：程序中用到的阈值，实现了根据图像信息按比例自动选择，但比例的确定采用了固定值；仅对于具有两条黑色分割线的答题卡适用等。
- 所设计方案与实际答题卡自动机器阅读原理不一致。

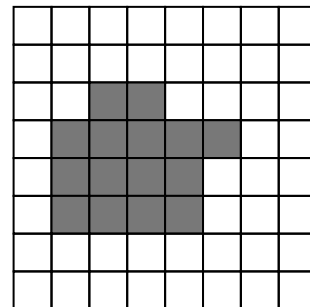
计算、思考



- 11.1 绘制下图直方图，并用峰谷法求出二值化的阈值。

$$f = \begin{bmatrix} 1 & 5 & 25 & 10 & 20 & 20 \\ 1 & 7 & 25 & 10 & 10 & 9 \\ 3 & 7 & 10 & 10 & 2 & 6 \\ 1 & 0 & 8 & 7 & 2 & 1 \\ 1 & 1 & 6 & 50 & 2 & 2 \\ 2 & 3 & 9 & 7 & 2 & 0 \end{bmatrix}$$

- 11.2 试按区域分裂合并方法分割右图，给出分割的各个步骤图。





计算、思考



- 11.3利用边缘检测实现分割，常常会有一些短小或不连续的曲线，用什么样的处理方法可以消除这些干扰？
- 11.4如何利用数学形态学算法削弱分水岭算法分割时所产生的过分割？
- 11.5设方差阈值为1，均值阈值为2，试用区域分裂合并方法对下图进行分割

$$f = \begin{bmatrix} 1 & 3 & 7 & 8 \\ 1 & 2 & 8 & 9 \\ 2 & 1 & 2 & 7 \\ 3 & 8 & 8 & 9 \end{bmatrix}$$



编程实践



- 11.6编写程序实现基于边界的图像分割，可以选择不同的边界改良算法。
- 11.7编写程序实现基于区域生长的图像分割。
- 11.8编写程序实现基于Hough变换检测圆。
- 11.9尝试在分水岭分割算法中加入相关预处理、后处理步骤，编程实现并查看分割效果。