图像分割

陈小潘
计算机与信息工程学院

目录

- 图像分割的概念
- 基于边缘检测的图像分割
- 基于阈值的图像分割
- 基于跟踪的图像分割
- 基于区域的图像分割

7.1 图像分割的概念

图像分割（Image Segmentation）

- 指根据图像的灰度、颜色、纹理和形状等特征，将图像划分成若干个互不交迭的区域，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的过程。

- 图像分割也是一种标记过程，即给属于同一区域的某种相似性特征的像素赋予相同的编号或颜色。从本质上说是图像分割将各像素进行分类的过程。分类所依据的特性可以是像素的灰度值、颜色或多谱特性、空间特性和纹理特性等。

7.1 图像分割的概念

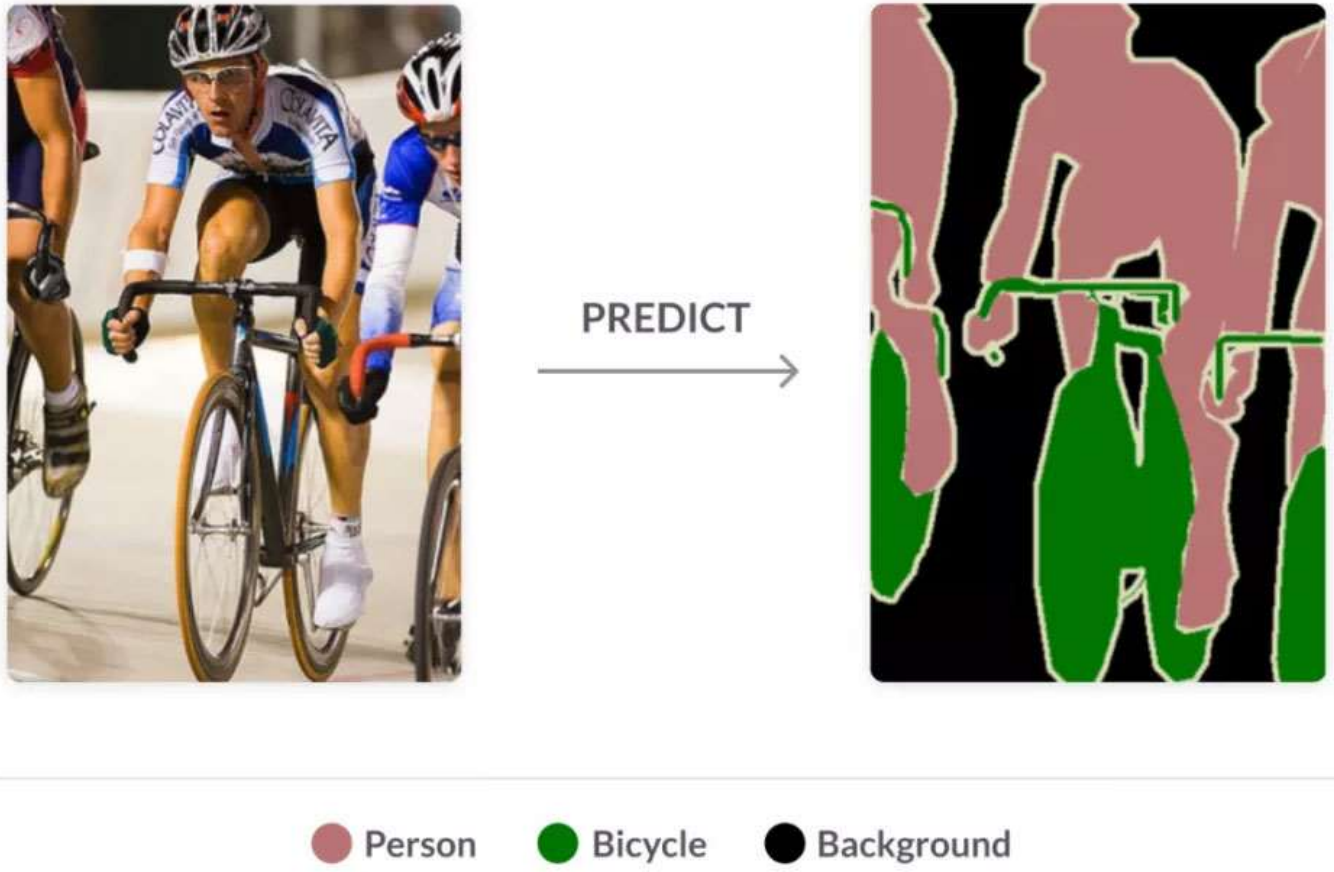
图像分割的目的是简化或改变图像的表示形式，以便在分割成的相关区域中**提取目标**，并进而根据目标的特征或结构信息对其进行分类和识别。

图像分割属于图像分析的范畴，是从图像处理到图像分析的标志性步骤。

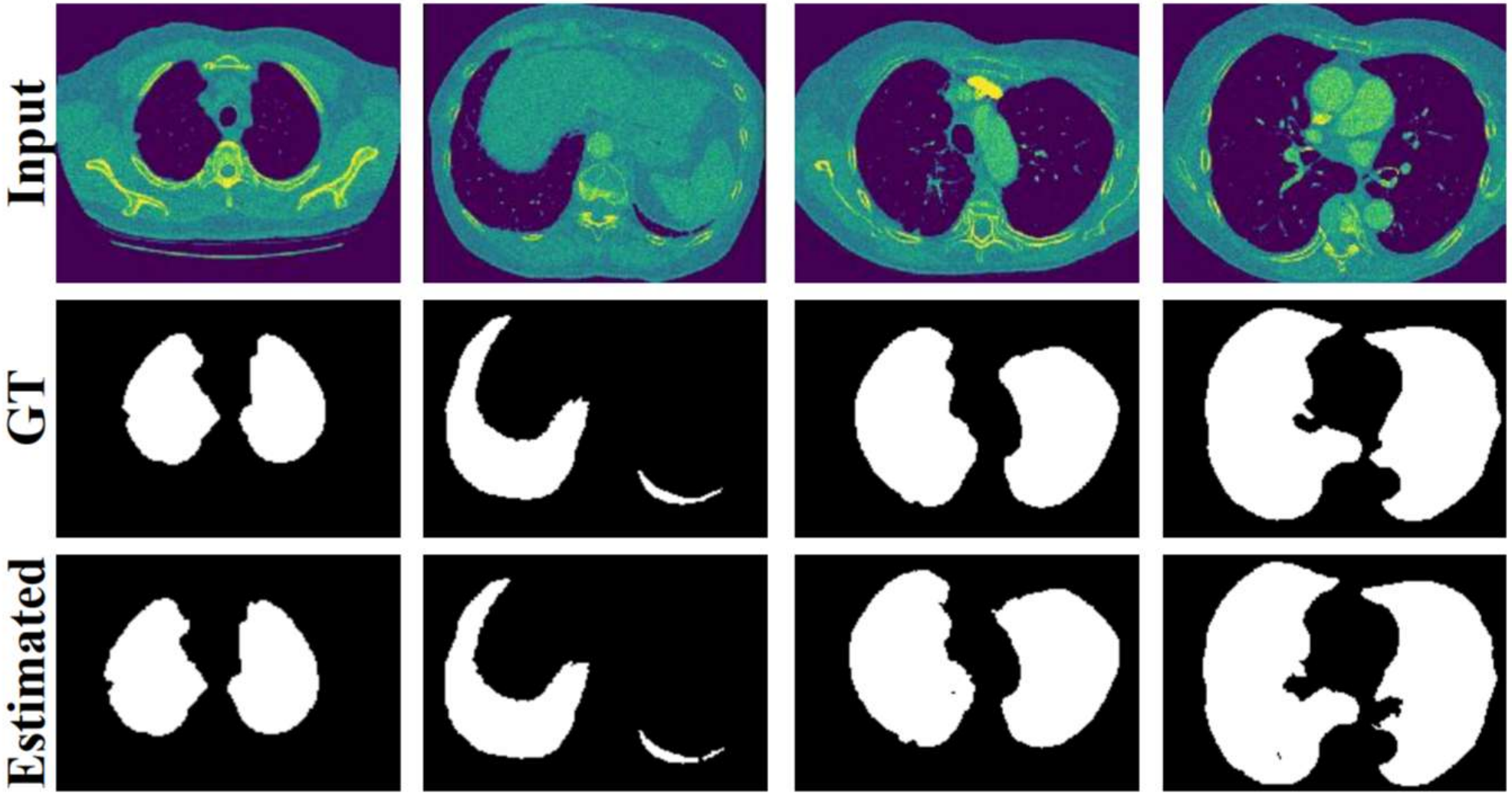
例如：

- ✓ 确定航空照片中的森林、耕地、城市区域等，首先需要将这些部分在图像上分割出来；
- ✓ 辨认文件中的文字，需要首先将这些文字先从文件中分选出来；
- ✓ 用图像分割可以识别和标定细胞的显微照片中的染色体

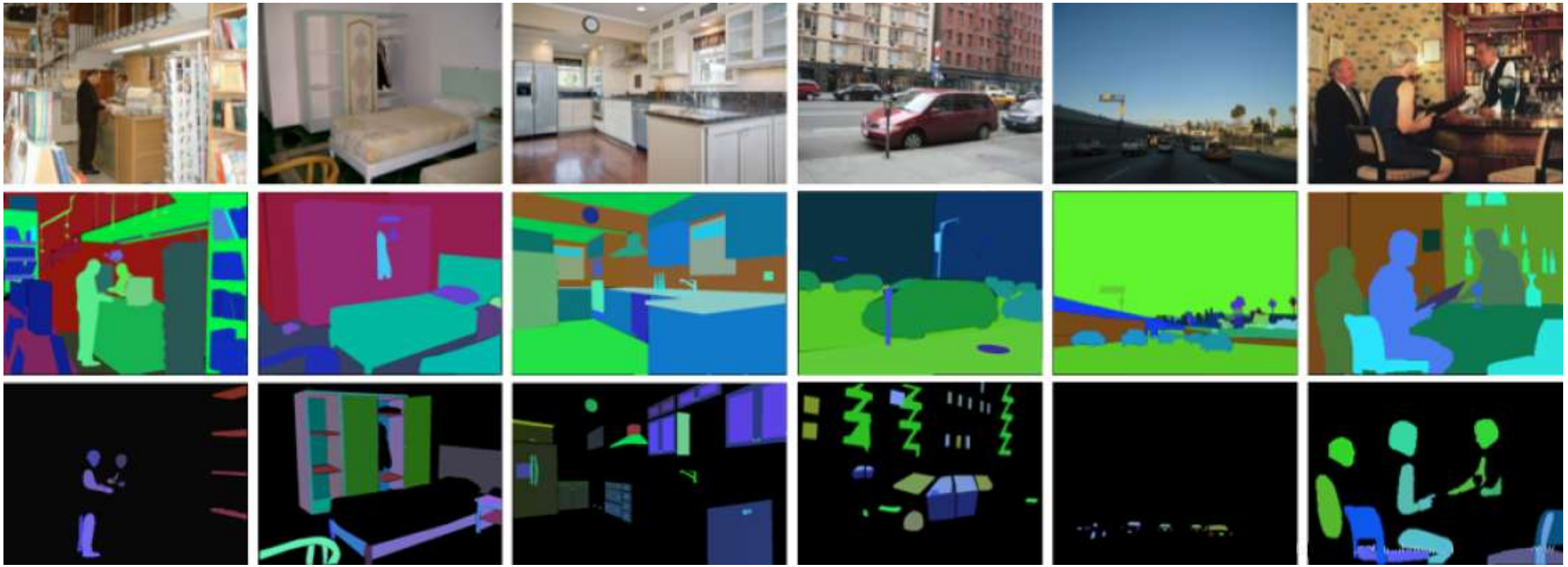
7.1　图像分割的概念



7.1　图像分割的概念



7.1　图像分割的概念



7.1　图像分割的概念

图像分割的意义

- **将图像空间按照一定的要求分成一些“有意义”的区域**
- 图像分割是模式识别、图像理解等技术的基础。

7.1　图像分割的概念

1、目标、前景和背景

人们在对图像进行研究和应用中，往往会对图像中某些部分感兴趣，这些部分一般称为目标或前景，而其它部分称为背景。

目标和背景是相对的概念，感兴趣的区域就认为是目标，反之就认为是背景。若想从一幅图像中“提取”物体，可以设法用专门的方法标出属于该物体的点，如将物体上的点标为“1”，而将背景点标为“0”，通过分割以后，可得一幅二值图像。

7.1 图像分割的概念

2. 图像分割的定义

设R代表整个图像区域，对R的分割可看作将R分成若干个满足以下5个条件的非空子集(子区域)R_1，R_2，…，R_n。

（1）$\bigcup_{i=1}^{n} R_i = R$。即分割成的所有子区域的并应能构成原来的区域R。[完备性]

（2）对于所有的i和j及i≠j，有$R_i \cap R_j = \phi$。即分割成的各子区域互不重叠。[独立性]

（3）对于i=1,2,…,n；有$P(R_i)$=TRUE。即分割得到的属于同一区域的像素应具有某些相同的特性。【单一性，P 为相似性准则 】

（4）对于i≠j，有$P(R_i \cup R_j)$=FALSE。即分割得到的属于不同区域的像素应具有不同的性质或特征。【互斥性】

（5）对于i=1,2,…,n；R_i是连通的区域。即同一子区域内的像素应当是连通的。【连通性】

7.1　图像分割的概念

好的图像分割的特征

- 分割出来的各区域对某种性质（例如灰度，纹理而言）具有相似性，区域内部是连通的且没有过多小孔；
- 区域边界是明确的；
- 相邻区域对分割所依据的性质有明显的差异。

7.1　图像分割的概念

3、灰度图像的分割

图像分割的依据是认为图像中各区域在灰度、颜色和纹理方面具有不同的特性。

由于受图像传感器技术和彩色图像需要传输巨大量数据的限制，各种遥感和军事侦察应用大都是灰度图像。

7.1　图像分割的概念

3、灰度图像的分割

灰度图像分割的依据是基于相邻像素灰度值的不连续性和相似性。同一区域内部的像素一般具有灰度相似性，而在不同区域之间的边界上一般具有灰度不连续性。

灰度图像的各种分割算法可据此分为：

- 利用区域间灰度不连续的基于边界的图像分割算法。
- 利用区域内灰度相似性的基于区域的图像分割算法。

7.1　图像分割的概念

图像分割方法分类

- 基于边缘检测的方法
- 基于区域生成的方法

7.1 图像分割的概念

基于边缘检测的方法

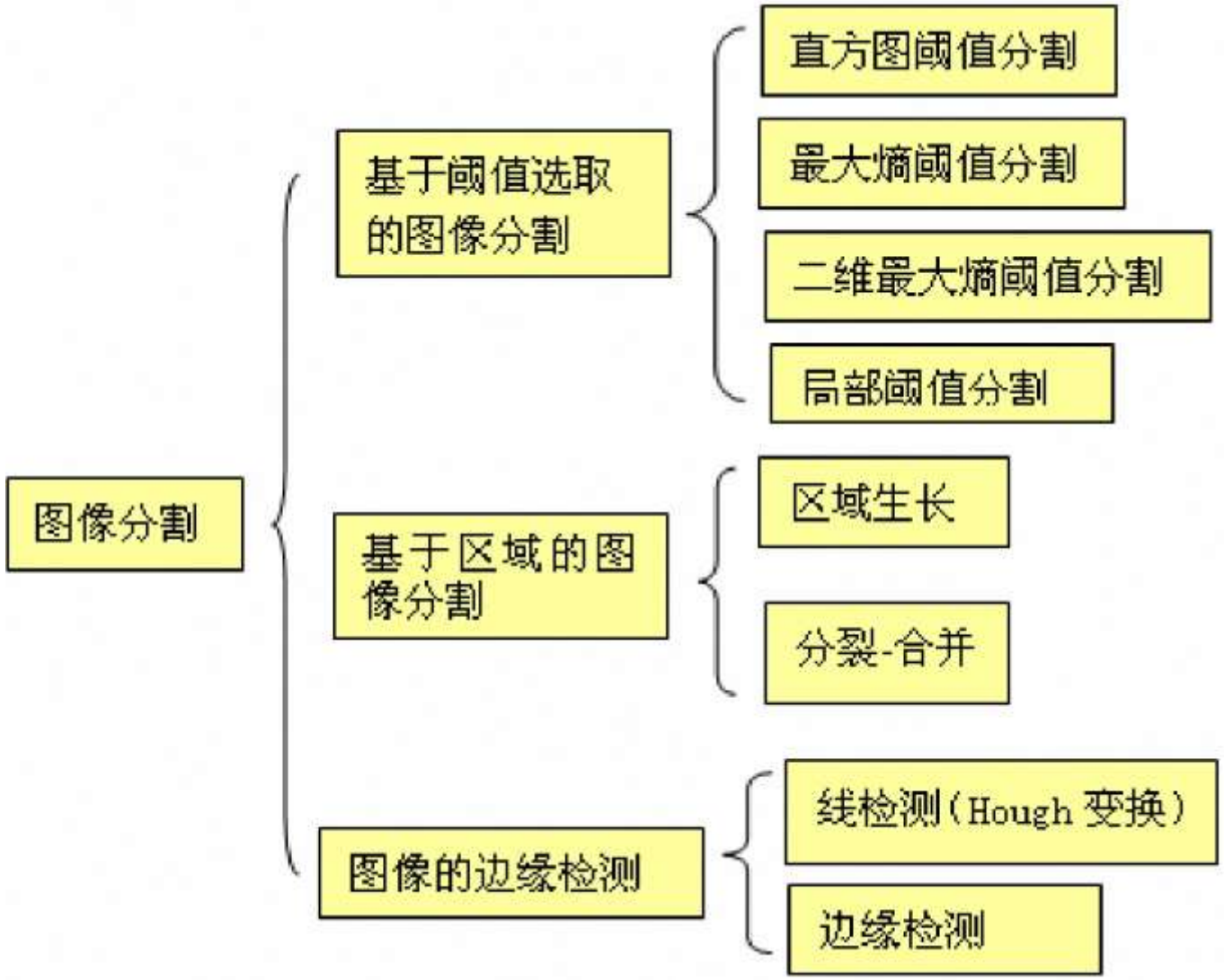
- 首先检出局部特性的不连续性，再将它们连成边界，这些边界将图像分成不同的区域，从而分割出各个区域。
- 常用边缘检测方法有基于边缘检测的图像分割、基于阈值选取的图像分割。

7.1 图像分割的概念

■ 基于区域生成的方法

- 为找出图像的边缘信息，将像素分成不同的区域，根据相应的区域特性在图像中找出与其相似的部分并进行处理。
- 常用的方法有区域生长、分裂-合并分割方法。

7.1 图像分割的概念


图像分割
基于阈值选取的图像分割
直方图阈值分割
最大熵阈值分割
二维最大熵阈值分割
局部阈值分割
基于区域的图像分割
区域生长
分裂-合并
图像的边缘检测
线检测（Hough 变换）
边缘检测

7.1 图像分割的概念

图像分割的重要应用领域：

- **医学成像**：在医学领域，图像分割用于分析磁共振成像（MRI）、计算机断层扫描（CT）等医学图像。它帮助医生识别和量化病变组织，如肿瘤，从而在疾病诊断、治疗规划和病情监测方面发挥关键作用。
- **自动驾驶汽车**：在自动驾驶技术中，图像分割用于从车载相机捕获的图像中识别道路、行人、车辆和其他障碍物，这对于路径规划和碰撞预防至关重要。
- **遥感影像分析**：在遥感应用中，图像分割用于处理来自卫星或航空摄影的图像，以识别地表特征，如土地覆盖、水体、林地等，对环境监测、城市规划和农业管理等领域有重要应用。

7.1 图像分割的概念

图像分割的重要应用领域

- **机器人视觉**：在机器人技术中，图像分割帮助机器人理解其周围的环境，从而在导航、物体识别和操控任务中发挥作用。
- **安全监控**：在安全和监控领域，图像分割可以用于人群监控、异常行为检测、车辆识别等，提高监控系统的效率和准确性。
- **图像编辑和特效**：在图像处理和电影制作领域，图像分割用于背景替换、特效添加等，使创意内容制作变得更加高效和逼真。
- **工业检测**：在工业生产中，图像分割用于自动检测产品缺陷，如裂纹、磨损或形状不一致，以保证产品质量。

目录

- 图像分割的概念
- **基于边缘检测的图像分割**
- 基于阈值的图像分割
- 基于跟踪的图像分割
- 基于区域的图像分割

7.2.1 图像边缘的概念

1、图像边缘的概念

图像边缘是指图像中两个相邻区域的边界线上连续的像素点的集合。图像中灰度发生突变或不连续的区域，即两个具有相对不同灰度值特性区域的边界线。

进一步讲，图像的边缘是指图像中的灰度、颜色、纹理等特征发生空间突变的像素的集合。

- 边缘有方向和幅度两个特性。一般认为沿边缘走向的灰度变化较为平缓，而垂直于边缘走向的灰度变化剧烈。
- 边缘处的灰度不连续常可利用求导数方便地检测到，所以一般常用一阶导数和二阶导数来检测边缘。

7.2.1 图像边缘的概念

数字图像的边缘检测是图像分割、目标区域识别、区域形状提取等图像分析领域十分重要的基础，也是图像识别中提取图像特征的重要属性。

- 在进行图像理解和分析时，第一步往往就是边缘检测， 由于边缘广泛存在于目标与目标、物体与背景、区域与区域（含不同色彩）之间，它是图像分割所依赖的重要特征。
- 边缘检测已成为机器视觉研究领域最活跃的课题之一，在工程应用中占有十分重要的地位。

7.2.1 图像边缘的概念

边缘处的灰度不连续常可利用求导数方便地检测到，所以一般常用一阶导数和二阶导数来检测边缘。

梯度算子是一阶导数算子。对一个连续函数$f(x, y)$，它在位置(x, y)的梯度可表示为一个矢量：$\nabla f(x,y) = [G_x \quad G_y]^{\mathrm{T}} = \left[\frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y}\right]^{\mathrm{T}}$

幅度（也常直接简称为梯度）和方向角分别为

$$\mathrm{mag}(\nabla f) = \|\nabla f\| = \left[G_x^2 + G_y^2\right]^{1/2}$$

$$\phi(x,y) = \arctan\left(G_y / G_x\right)$$

7.2.1　图像边缘的概念

2、图像边缘的特点

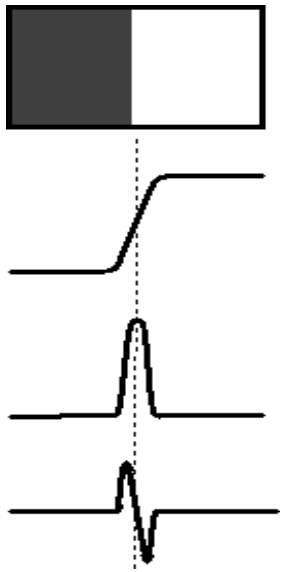
图像边缘的特点是两侧的灰度在通过边缘时将发生某种显著的变化

（a）边缘的理想阶跃截面　　（b）实际中的边缘阶跃截面

图7.2 图像中的边缘的截面示意图

7.2.1 图像边缘的概念

2、图像边缘的特点



图像边缘

剖面

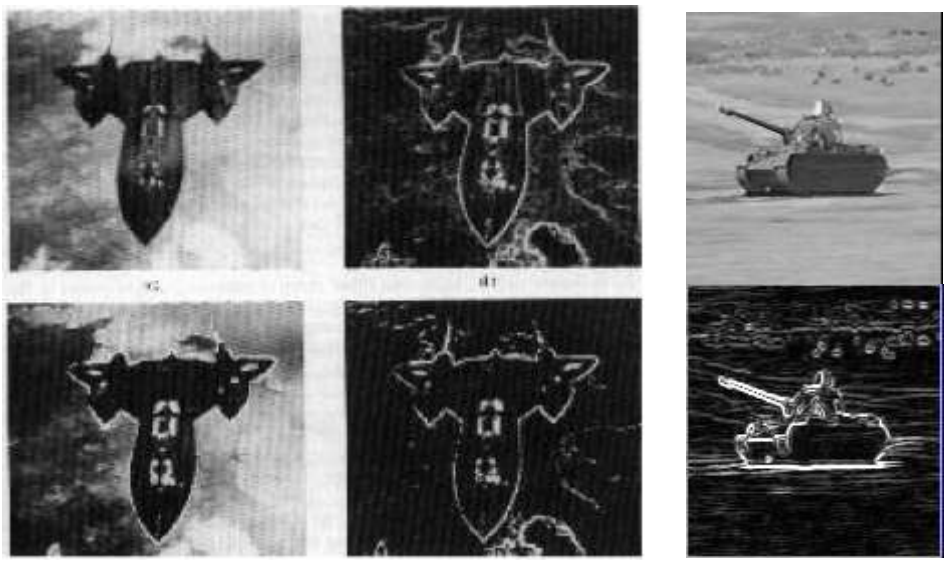
一阶导数在边缘处取极值

二阶导数在边缘处有过零点（零交叉）

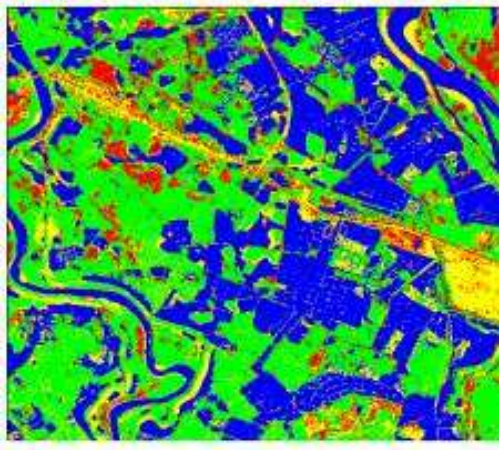
7.2.1　图像边缘的概念

3、基于边缘检测(Edge Detection)的图像分割方法的基本思路

基本思路是先确定图像中的边缘像素，然后将这些边缘像素连接在一起构成所需的边界。



(a)边缘区分的空中炸弹区域



(b)边缘区分的水体，旱地，林地等

7.2.2 Hough变换

基本思想：

Hough变换是图像处理中从图像中识别几何形状的基本方法之一。Hough变换的基本原理在于利用点与线的对偶性，将原始图像空间的给定的曲线通过曲线表达形式变为参数空间的一个点。这样就将原始图像中给定曲线的检测问题转化为寻找参数空间中的峰值问题。也即把检测整体特性转化为检测局部特性。比如直线、椭圆、圆、弧线等。

霍夫变换于1962年由Paul Hough首次提出，后于1972年由Richard Duda和Peter Hart推广使用，经典霍夫变换用来检测图像中的直线，后来霍夫变换扩展到任意形状物体的识别，多为圆和椭圆。

7.2.2 Hough变换

1. Hough变换的基本原理

设在图像空间X-Y中，所有过点(x,y)的直线都满足方程：

$$y = px + q \qquad (7.1)$$

将其改写成：

$$q = -px + y \qquad (7.2)$$

这时，p和q可看作是变量，x和y可看作是参数。式（7.2）就可表示参数空间P-Q中过点(p,q)的一条直线。

7.2.2 Hough变换

1. Hough变换的基本原理

一般地，对于任意的i和j，设图像空间X-Y中同时过点(x_i,y_i)和点(x_j,y_j)的直线方程分别为：

$$y_i = px_i + q \quad （7.3）$$

$$y_j = px_j + q \quad （7.4）$$

则其对应的在参数空间P-Q中，同时过点（p,q）的直线分别为:

$$q = -px_i + y_i \quad （7.5）$$

$$q = -px_j + y_j \quad （7.6）$$

7.2.2 Hough变换

- 由于一条确定的直线对应一组确定的参数p，q，因此X-Y空间一条确定的直线对应一组确定的数据p，q。
- P-Q空间一条确定的直线对应一组确定的数据x、y（因为x，y为其参数），因此P-Q空间一条确定的直线对应XY空间一个点。
- X-Y空间一条直线上的n个点，对应参数P-Q空间经过一个公共点的n条直线

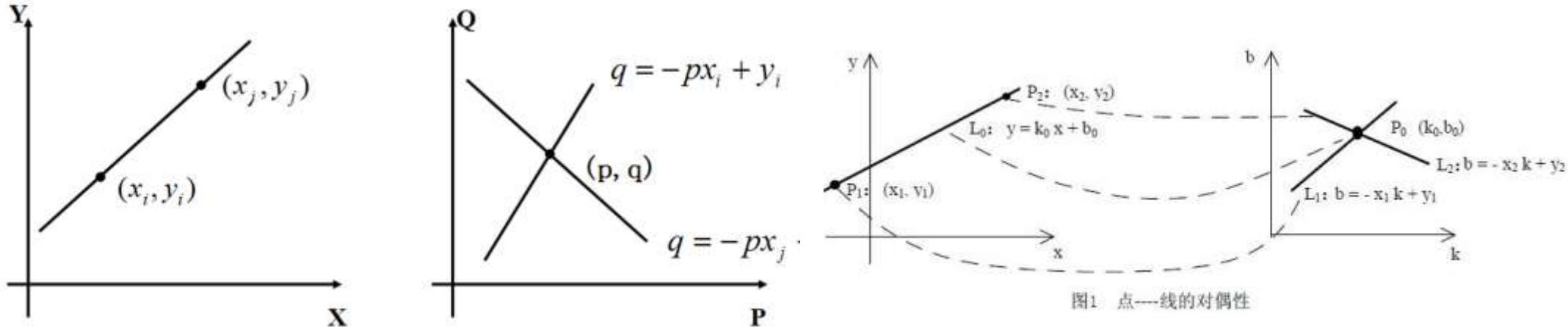


图1 点----线的对偶性

图7.4 图像空间直线与参数空间点的对偶性

7.2.2 Hough变换

1. Hough变换的基本原理

将上述结论推广到更一般的情况：如果图像空间X-Y中的有一条边界线为直线段，其上有n个点，那么这些点对应参数空间P-Q上的一个由n条直线组成的直线簇，且所有这些直线相交于同一点。检测出这个交点，即可得到图像中该直线的参数，进而确定这条直线。

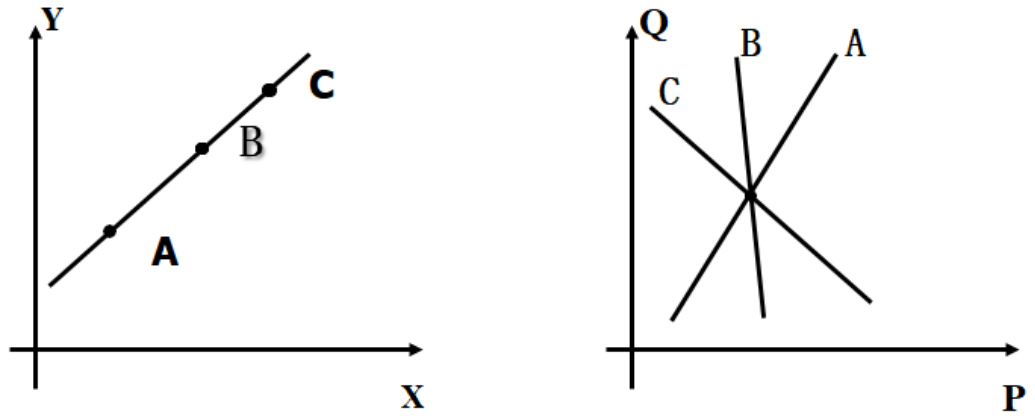


图7.5 图像空间的一条直线与参数空间的直线簇交点对应示例

7.2.2 Hough变换

1. Hough变换的基本原理

简而言之，Hough变换思想为：在原始图像坐标系下的一个点对应了参数坐标系中的一条直线，同样参数坐标系的一条直线对应了原始坐标系下的一个点，然后，原始坐标系下呈现直线的所有点，它们的斜率和截距是相同的，所以它们在参数坐标系下对应于同一个点。这样在将原始坐标系下的各个点投影到参数坐标系下之后，看参数坐标系下有没有聚集点，这样的聚集点就对应了原始坐标系下的直线。

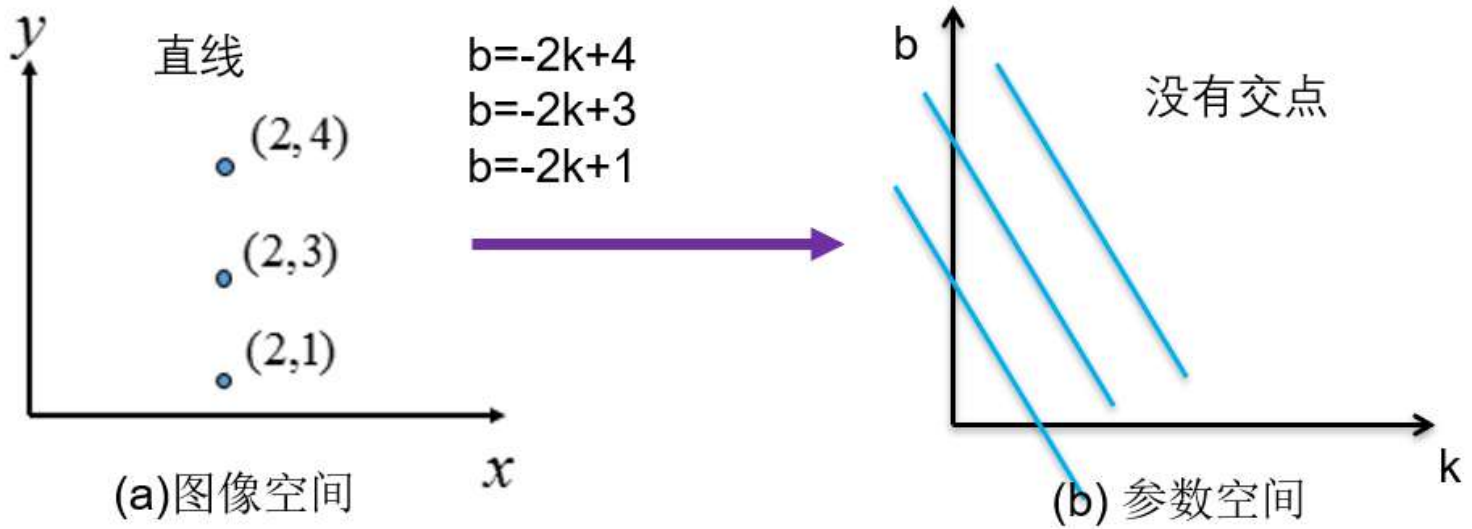
参数空间n条直线交点的检测方法：

对于原图中的每个点，在参数空间确定一条直线，即该直线所经过点的值累加1，经过直线最多的点（累加值最大的点）为原图中直线的参数。

7.2.2 Hough变换

2. 经典Hough变换

在实际中， 当用式（7.1）$y = px + q$ 表示的直线方程接近竖直（根据斜率计算公式，分母为0，所以也即当该直线的斜率接近无穷大）或为垂直时（也可以理解为$y = px + q$对垂直线不起作用），则会由于参数空间中p和q的值接近无穷大或为无穷大而无法表示，实际应用中，是采用参数方程$\rho = x\,sin\,\theta + y\,cos\,\theta$来表示。



7.2.2 Hough变换

2. 经典Hough变换

检测直线的Hough变换一般使用含极坐标参数的直线表示形式，如图7.6。

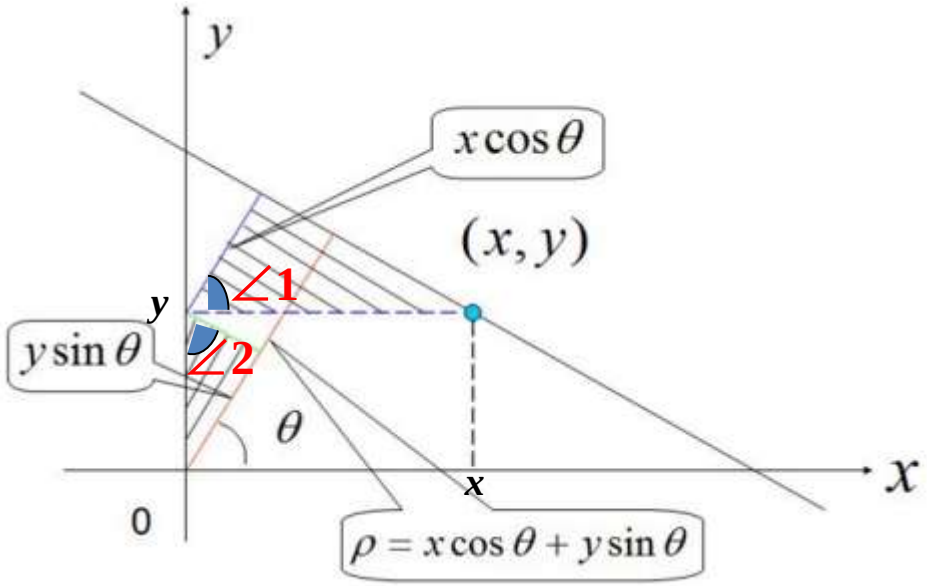


图7.6 图中蓝色为θ

7.2.2 Hough变换

2. 经典Hough变换

图7.6描述了在X-Y坐标中相交于x轴与y轴的直线L和用极坐标ρ-θ中的极径ρ（从极点o出发并垂直于直线L的线段）与极角θ表示直线L的关系。

分析：（1）根据图中各线段之间的平行关系，∠1大小为θ，所以∠1的邻边为x。{$\cos\theta$ =邻边/斜边x}

（2）∠2的大小为θ，所以∠2的对边为$y\sin\theta$。{$\sin\theta$ =对边/斜边y}

所以，极径ρ为：$\rho = x\,sin\,\theta + y\,cos\,\theta$ （7.7）

$$\text{也即：}\ y = \frac{-cos\theta}{sin\theta}x + \frac{\rho}{sin\theta} \quad (7.8)$$

显然，式（7.8）即是X-Y坐标中的直线方程——式（7.1）在极坐标ρ-θ中的表示方式。

$y = px + q$ （7.1）

7.2.2 Hough变换

2. 经典Hough变换

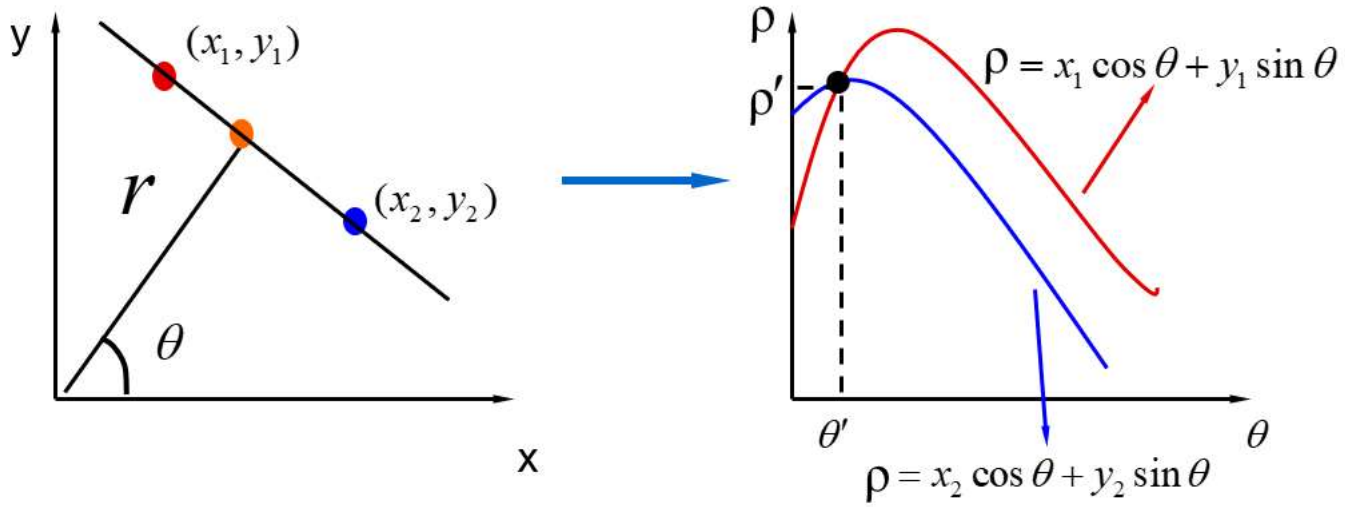
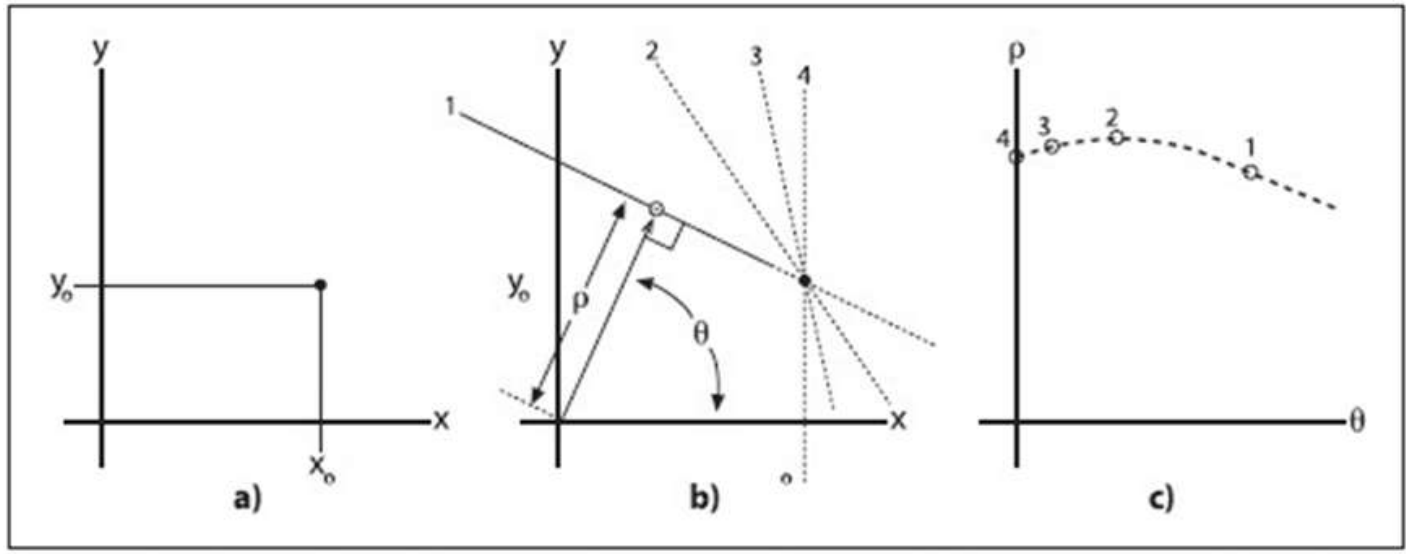


图 图像空间到极坐标参数空间的转换过程

7.2.2 Hough变换

2. 经典Hough变换



（a）所示为原始的图像空间中一个点；（b）所示为直角坐标系当中为过同一四条直线；（c）所示为这四条直线在极坐标参数空间可以表示为四个点

7.2.2 Hough变换

2. 经典Hough变换

在式（7.7）和式（7.8）的意义下，

- 图像空间X-Y中的一条确定的直线对应极坐标空间θ-ρ的一个点
- θ-ρ空间的一条正弦曲线对应X-Y空间的一个点
- X-Y空间一条直线上的n个点对应极坐标空间θ-ρ中经过一个公共点的n条正弦曲线
- 对于原图中的每个点，在参数空间确定一条正弦曲线，即该曲线所经过点的值累加1，经过曲线最多的点（累加值最大的点）为原图中直线的参数。

7.2.2 Hough变换

2. 经典Hough变换

在极坐标参数空间变换下，图像空间中一条直线就和参数空间下一组曲线的**交点相对应**，也就是说，图像空间中一点对应参数空间中一条曲线。这样就将图像空间中直线的检测转换成极坐标参数空间中曲线交点的检测。

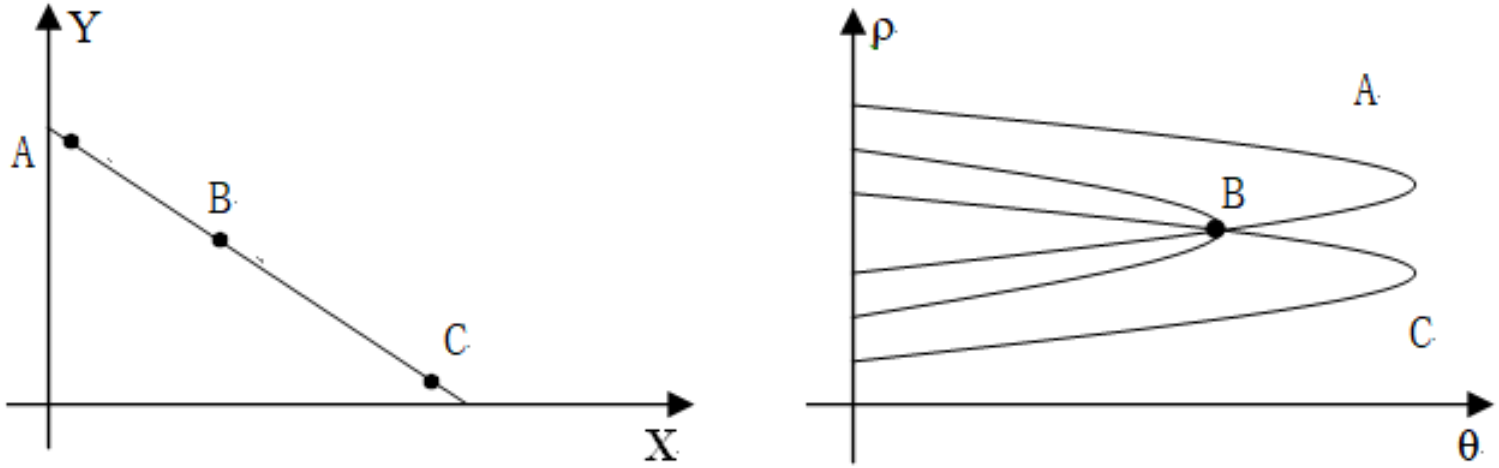


图7.7 图像空间的一条直线与极坐标空间的曲线簇交点对应示例

7.2.2 Hough变换

3. 用Hough变换提取图像中直线的方法

实际上图像空间中的直线可能并不是完全直的，不同部位的像素值粗细也可能不同，所以在实现时，要根据精度要求将参数空间ρ-θ离散化成一个累加器阵列，也即将参数空间细分成一个个网格阵列（将每个网格近似看作是一点，认为通过每个网格的曲线近似相交于该网格对应的“点”），其中的每一个格子对应一个累加器（用于记录相交于该点的曲线数），如图7.8所示。

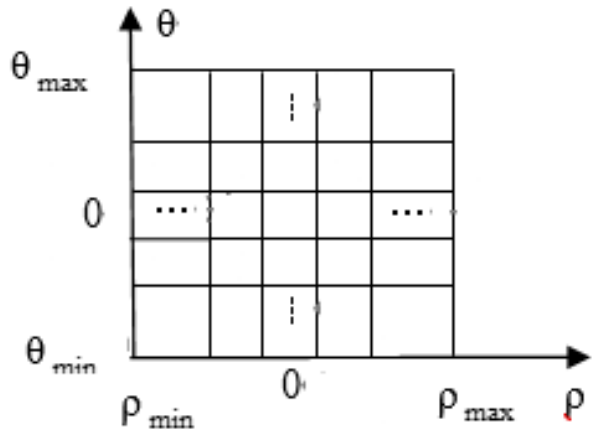


图7.8 将ρ-θ平面细分成网格阵列

7.2.2 Hough变换

3. 用Hough变换提取图像中直线的方法

初始时累加器阵列中每个值都被置为0，将图像空间中每一点都转换成参数空间中的一条曲线，曲线经过一个格子则该位置的累加器值+1，由于通过同一个格子所对应的点近似认为共线，所以每个格子上累加器的值就代表了图像空间中有多少个点共线，该点的极坐标就代表了所共直线的信息。

这样，如果图像空间中包含若干直线，则在参数空间中就会有同样数量的格子出现局部极大值，通过检测这些极大直就可以检测出各个直线。

7.2.2 Hough变换

Eg10_6.m

```
clc,clear,close all;
Image=rgb2gray(imread('houghsource.bmp'));
figure,imshow(Image);
bw=edge(Image,'canny');      %Canny算子边缘检测得到二值边缘图像
figure,imshow(bw);
[h,t,r]=hough(bw,'RhoResolution',0.5,'ThetaResolution',0.5);  %Hough变换
figure,imshow(imadjust(mat2gray(h)),‘XData’,t,‘YData’,r,‘InitialMagnificatio
n’,‘fit’);        %显示Hough变换矩阵
xlabel('\theta'),ylabel('\rho');
axis on,axis normal,hold on;
```

7.2.2 Hough变换

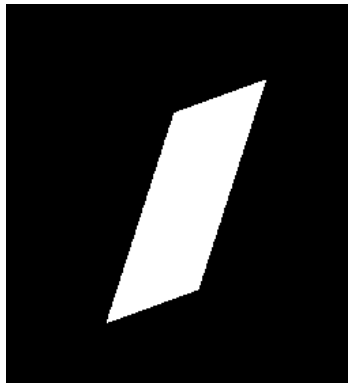
Eg10_6.m

```
P=houghpeaks(h,2);
x=t(P(:,2)); y=r(P(:,1));
plot(x,y,'s','color','r');              %获取并标出参数平面上的峰值点
lines=houghlines(bw,t,r,P,'FillGap',5,'Minlength',7);%检测图像中的直线度
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');     %用绿色标注直线段
   plot(xy(1,1),xy(1,2),'x','LineWidth',2,'Color','y');
   plot(xy(2,1),xy(2,2),'x','LineWidth',2,'Color','r');  %标注直线段端点
```

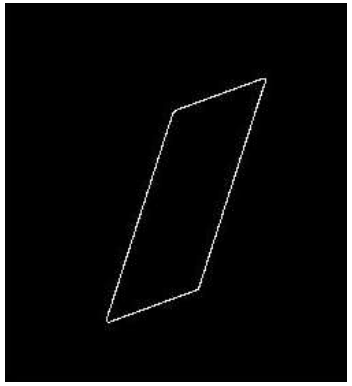
peaks = houghpeaks（H，numpeaks）：在由霍夫函数生成的霍夫变换矩阵**H**中定位峰。**numpeaks**指定要识别的最大峰数。该函数返回峰的矩阵，其中包含峰的行和列

7.2.2 Hough变换

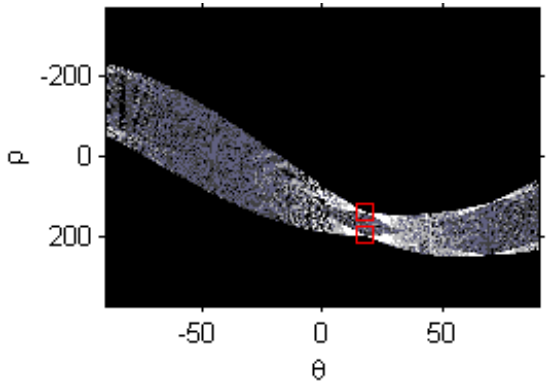
Hough变换检测直线



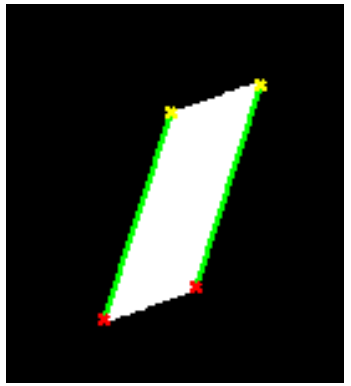
原始图像



边界图像



ρ-θ参数空间



检测结果

7.2.2 Hough变换

Hough检测圆

原理

$$圆方程(x-a)^2+(y-b)^2=r^2$$

式中(a,b)为圆心坐标，r为圆的半径。

- *x-y*空间中一个圆对应三维参数空间一个点（*a*,*b*,*r*)
- *x-y*空间中圆上一个点(*x*,*y*)对应参数空间一条曲线
- *x-y*空间中圆上*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。

7.2.2 Hough变换

Hough检测圆

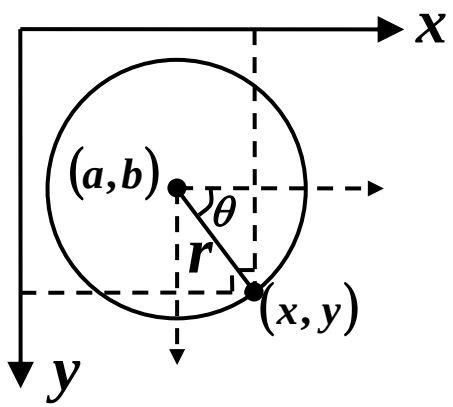
假设原图像已经处理为二值边缘图像，循环扫描图像中的每一个点。对于每一点在(a,b,r)参数空间确定一条曲线，即参数空间上的对应曲线经过的所有(a,b,r)点的值累加1。参数空间上累计值最大的点(a^*,b^*,r^*)为所求圆参数，按照该参数在与原图像同等大小的空白图像上绘制圆。

7.2.2 Hough变换

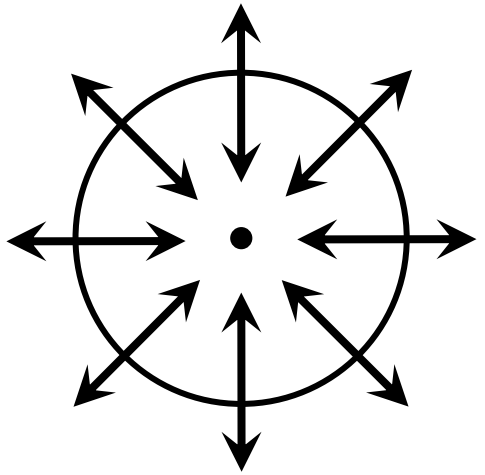
Hough检测圆

■ 简化运算

三维参数空间，计算量大，算法复杂度增加。可以采用极坐标形式，进行进一步简化。通过获取边界点的梯度，根据指向圆内的梯度，可以求出圆心的位置。因为仅计算圆的半径，所以计算量减小。



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



7.2.2 Hough变换

Hough检测圆实例

```
clear,clc,close all;
Image=imread('coins.png');
subplot(131),imshow(Image);title('(a)原始图像','FontSize',12)
[centers,radii,metric]=imfindcircles(Image,[15 30]);%检测半径在[15,30]的圆
subplot(132),imshow(Image);title('(b)检测的所有圆','FontSize',12)
viscircles(centers,radii,'Color','b');          %用蓝色线绘制所有的圆
centersStrong5=centers(1:5,:);      %获取圆度做大的5个圆的圆心
radiiStrong5=radii(1:5);
metricStrong5=metric(1:5);
subplot(133),imshow(Image);title('(c)圆度最大的5个圆','FontSize',12)
viscircles(centersStrong5, radiiStrong5,'Color','g','LineStyle',':');
```

7.2.2 Hough变换

使用圆形Hough变换查找圆

- centers = imfindcircles（A，radius）在图像A中找到半径近似等于要求半径的圆。

centers是一个两列的矩阵，其中包含图像中圆心的（x，y）坐标。

- [centers，radii] = imfindcircles（A，radiusRange）查找半径在radiusRange指定范围内的圆。

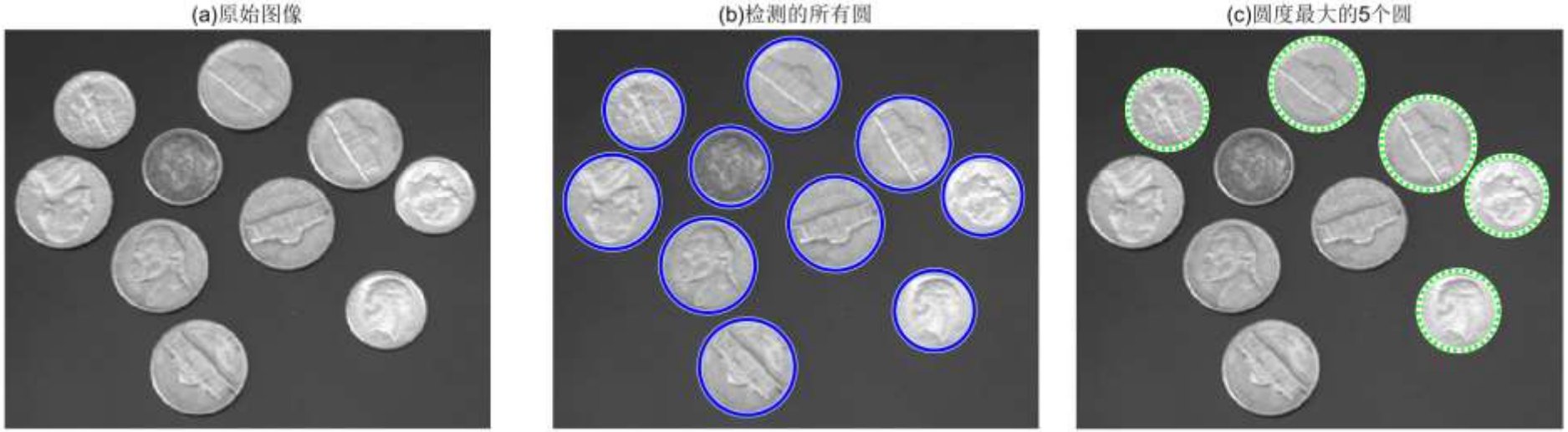
附加输出参数radii包含与中心中每个圆心相对应的估计半径。

- [centers，radii，metric] = imfindcircles（A，radiusRange）还返回一个列向量metric，

其中包含每个圆的累加器阵列峰值的大小（降序排列）。 中心行和半径行对应于公制行。

7.2.2 Hough变换

- **Hough检测圆实例**



目录

- 图像分割的概念
- 基于边缘检测的图像分割
- **基于阈值的图像分割**
- 基于跟踪的图像分割
- 基于区域的图像分割

7.3 基于阈值的图像分割

基于阈值的图像分割适用于那些物体（前景）与背景在灰度上有较大差异的图像分割问题。

严格来讲，基于阈值的图像分割方法是一种区域分割技术。

若图像中目标和背景具有不同的灰度集合且两个灰度集合可用一个灰度级阈值T进行分割。这样就可以用阈值分割灰度级的方法在图像中分割出目标区域与背景区域，这种方法称为灰度阈值分割方法。

7.3.1　基于阈值的分割方法

1. 阈值化分割方法

当图像由较亮的物体和较暗的背景组成，且物体与背景的灰度有较大差异时，该图像的灰度直方图会呈现出类似于图7.10所示的两个峰值的情况。

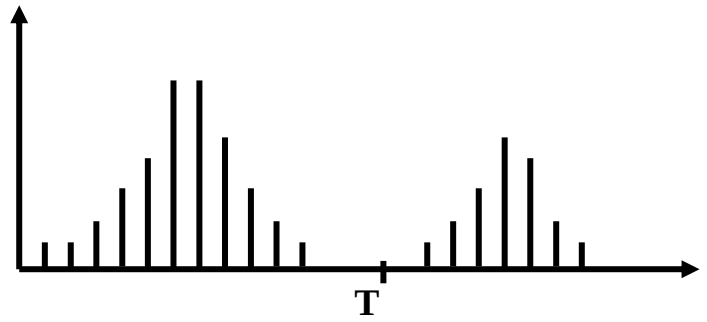


图7.10 基于单一阈值分割的灰度直方图

7.3.1 基于阈值的分割方法

1. 阈值化分割方法（续）

提取物体/目标的方法是通过选取位于两个峰值中间的谷底对应的灰度值T作为灰度阈值，可区分目标和背景。

从暗的背景上分割出亮的物体：

$$g(x,y)=\begin{cases}1 & f(x,y)\geq T\\ 0 & f(x,y)<T\end{cases} \qquad (7.9)$$

从亮的背景上分割出暗的物体：

$$g(x,y)=\begin{cases}1 & f(x,y)\leq T\\ 0 & f(x,y)>T\end{cases} \qquad (7.10)$$

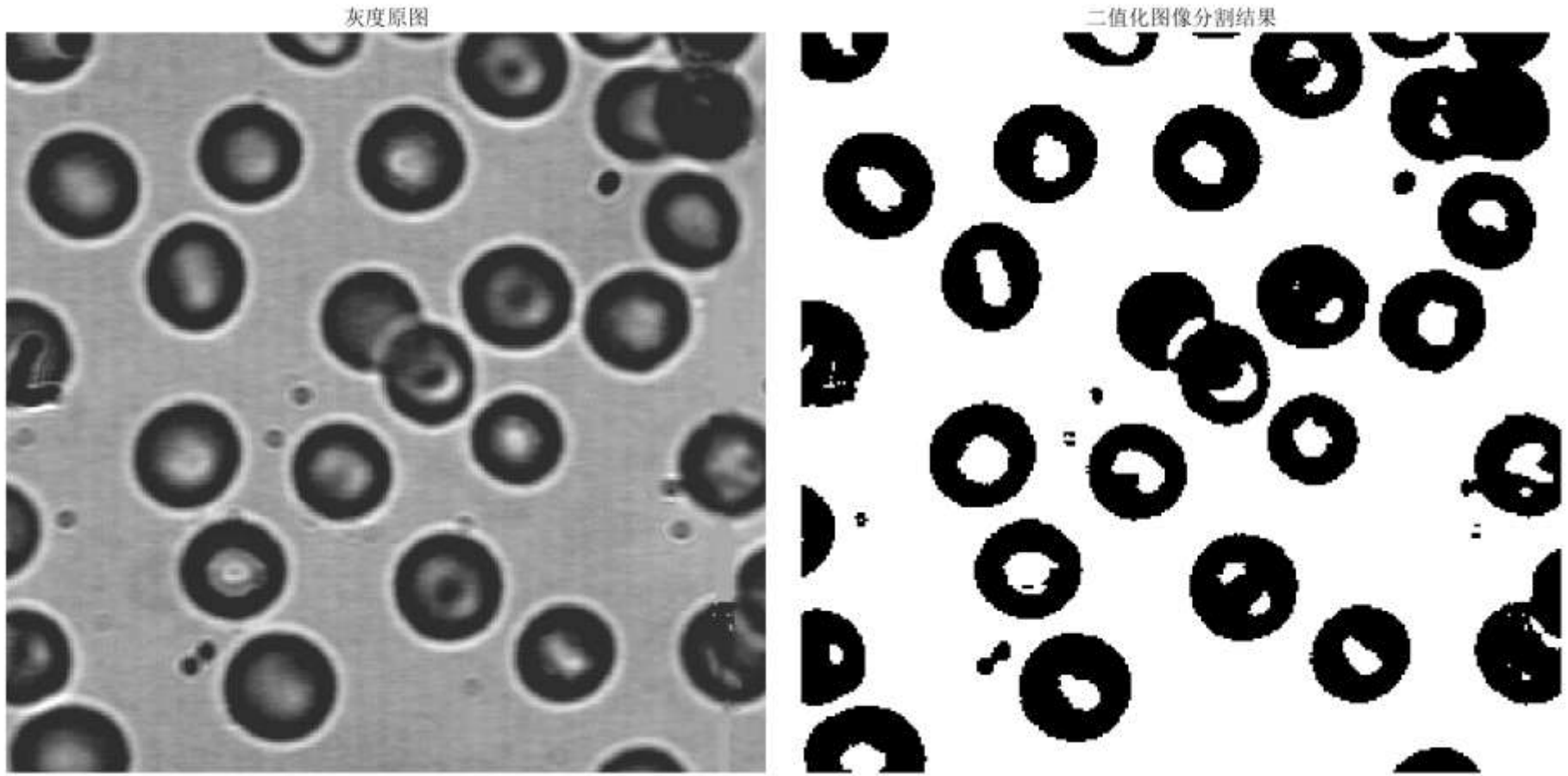
7.3.1 基于阈值的分割方法

例 利用阈值化方法分割图像的MATLAB程序。

```
clc; clear all; close all;
img0=imread('blood1.tif');
[h,w]=size(img0);
Th=graythresh(img0); %给灰度图像找一个合适的(归一化)阈值
Th1 = uint8(256*Th); %阈值的8位数值
img1 = img0; %给结果图像赋初值
for i=1:h    %求该图像的二值图像
      for j=1:w
        if img0(i,j) >= Th1
          img1(i,j) = 255;
        else
          img1(i,j) = 0;
        end
   end
end
 subplot(1,2,1);imshow(img0),title('灰度原图','FontSize',12);
 subplot(1,2,2);imshow(img1),title('二值化图像分割结果','FontSize',12);
```

7.3.1　基于阈值的分割方法

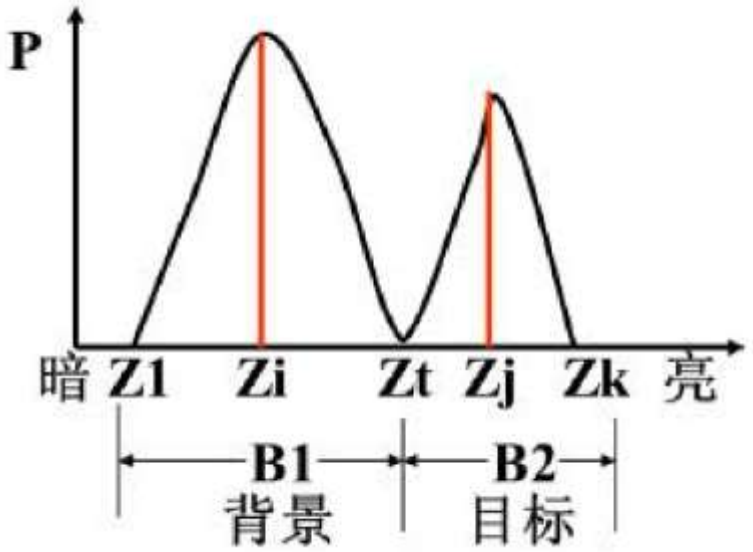
例　利用阈值化方法分割图像的MATLAB程



7.3.1 基于阈值的分割方法

- 直方图阈值的双峰法

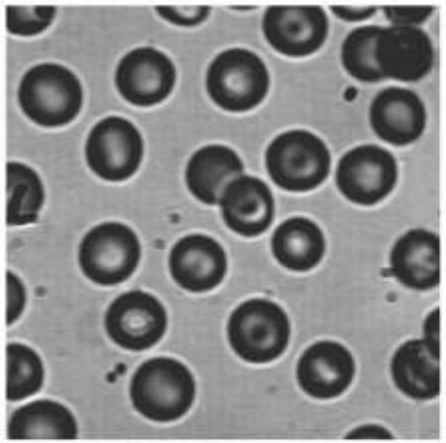
当灰度图像中画面比较简单且对象物的灰度分布比较有规律：背景和对象物在图像的灰度直方图上各自形成一个波峰。



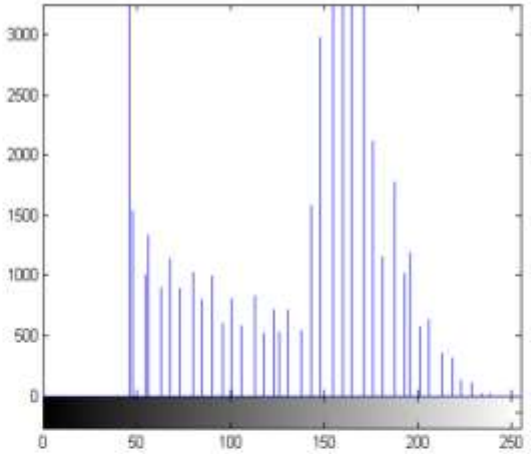
7.3.1 基于阈值的分割方法

- 直方图阈值的双峰法
  - ✓ 由于每两个波峰间形成一个低谷，因而选择双峰间低谷处所对应的灰度值为阈值，可将两个区域分离。
  - ✓ 我们将这种通过选取直方图阈值来分割目标和背景的方法称为直方图阈值双峰法。

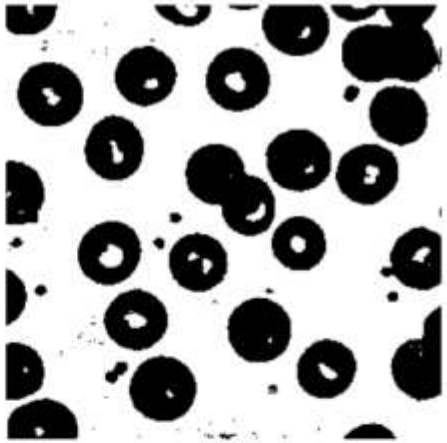
7.3.1 基于阈值的分割方法



细胞原灰度图像

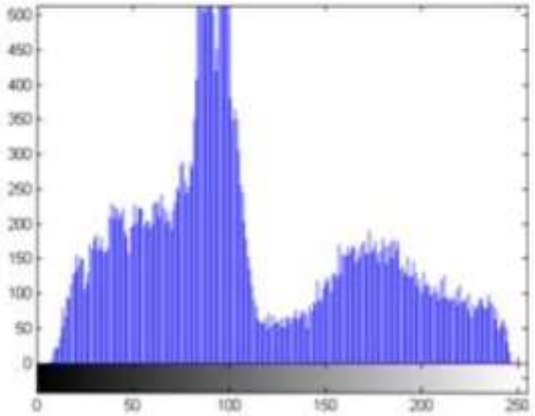


图像直方图



$T = 140$时阈值分割后的图像

原灰度图像



图像直方图

$T = 130$时阈值分割后的图像

7.3.1 基于阈值的分割方法

【例】实现基于双峰分布的直方图选择阈值，分割图像

- 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。
- 例如，将直方图中相邻3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。

7.3.1 基于阈值的分割方法

【例】实现基于双峰分布的直方图选择阈值，分割图像

```
%基于双峰分布的直方图选择阈值分割图像
clear,clc,close all;
Image=rgb2gray(imread('lotus.jpg'));
subplot(2,2,1);imshow(Image),title('(a)原始图像');
subplot(2,2,2);imhist(Image);title('(b)原始图像的直方图');
hist1=imhist(Image);
hist2=hist1;
iter=0; %迭代次数，限制循环次数
while 1
   [is,peak]=Bimodal(hist1);%判断是否为双峰分布直方图，是的话则找到峰
   if is==0  %对非双峰分布直方图进行平滑
      hist2(1)=(hist1(1)*2+hist1(2))/3;
      for j=2:255
         hist2(j)=(hist1(j-1)+hist1(j)+hist1(j+1))/3;%对相邻3个点求平均以平滑直方图
      end
      hist2(256)=(hist1(255)+hist1(256)*2)/3;
      hist1=hist2;
      iter=iter+1;
      if iter>1000
         break;
      end
   else
      break;
   end
end
```

7.3.1 基于阈值的分割方法

【例】实现基于双峰分布的直方图选择阈值，分割图像

```
[trough,pos]=min(hist1(peak(1):peak(2)));%找双峰间的波谷
thresh=pos+peak(1);%波谷对应的灰度
subplot(2,2,3);stem(1:256,hist1,'Marker','none');title('(c)平滑后的直方图和波谷');
hold on
stem([thresh,thresh],[0,trough],'Linewidth',2);
hold off
result=zeros(size(Image));
result(Image>thresh)=1;%阈值化
str1=strcat('(d)基于双峰直方图的阈值化,T=',num2str(thresh));
subplot(2,2,4);imshow(result),title(str1);
imwrite(result,'bilotus1.jpg');
```

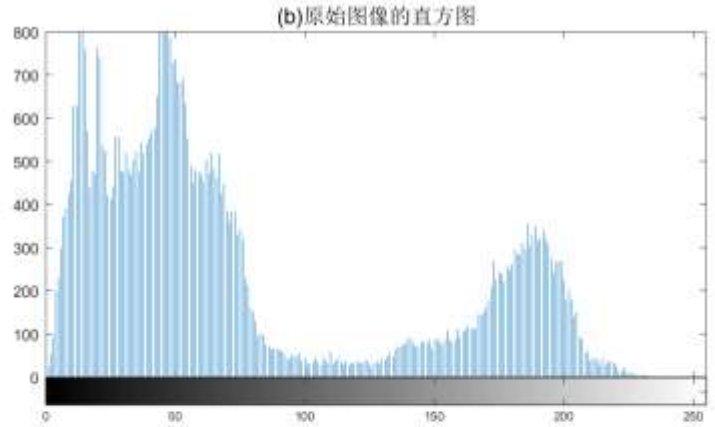
7.3.1 基于阈值的分割方法

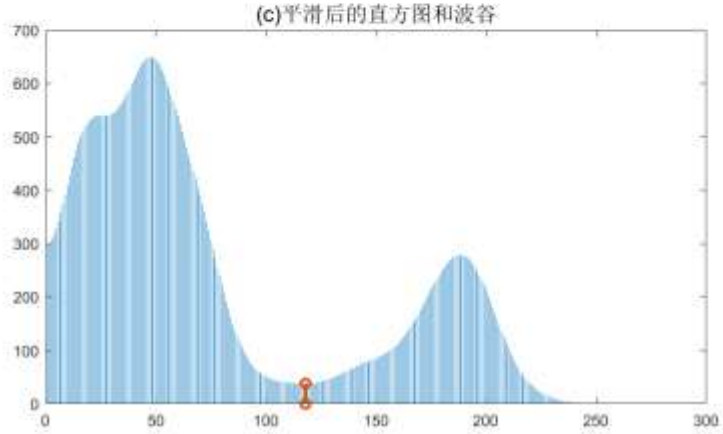
【例】实现基于双峰分布的直方图选择阈值，分割图像

```
function [is,peak]=Bimodal(histgram)
    count=0;
    for j=2:255
        if histgram(j-1)<histgram(j) && histgram(j+1)<histgram(j)
            count=count+1;
            peak(count)=j;%记录峰所在的位置
            if count>2
                is=0;
                return;
            end
        end
    end
    if count==2
        is=1;
    else
        is=0;
    end
end
```

7.3.1 基于阈值的分割方法

【例】实现基于双峰分布的直方图选择阈值，分割图像





7.3.1 基于阈值的分割方法

1. 阈值化分割方法（续）

当在较暗的背景上有2个较亮的物体，且有如下的直方图和约定时：

（1）$f(x,y) \le T_1$，　为背景(序号为1)；

（2）$T_1 < f(x,y) \le T_2$，为物体甲(序号为2)；

（3）$T_2 < f(x,y)$，为物体乙(序号为0)。

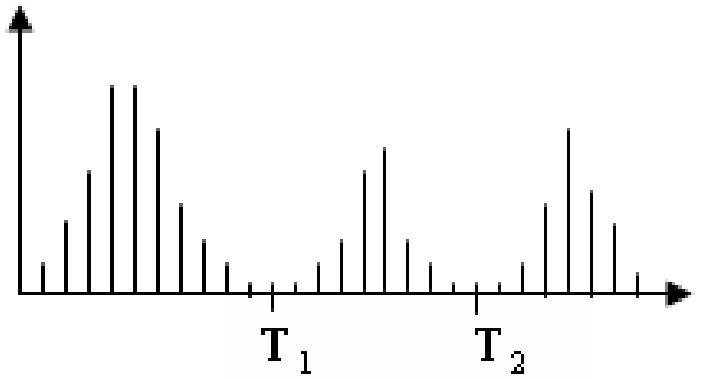


图7.12 基于多阈值分割的灰度直方图

7.3.1 基于阈值的分割方法

1. 阈值化分割方法（续）

更一般的多个阈值的情况为：

$$g(x,y)=\begin{cases}1 & f(x,y)\le T_1\\ k & T_{k-1}<f(x,y)\le T_k\\ k+1 & T_k<f(x,y)\end{cases} \qquad (7.11)$$

7.3.1 基于阈值的分割方法

2. 半阈值化分割方法

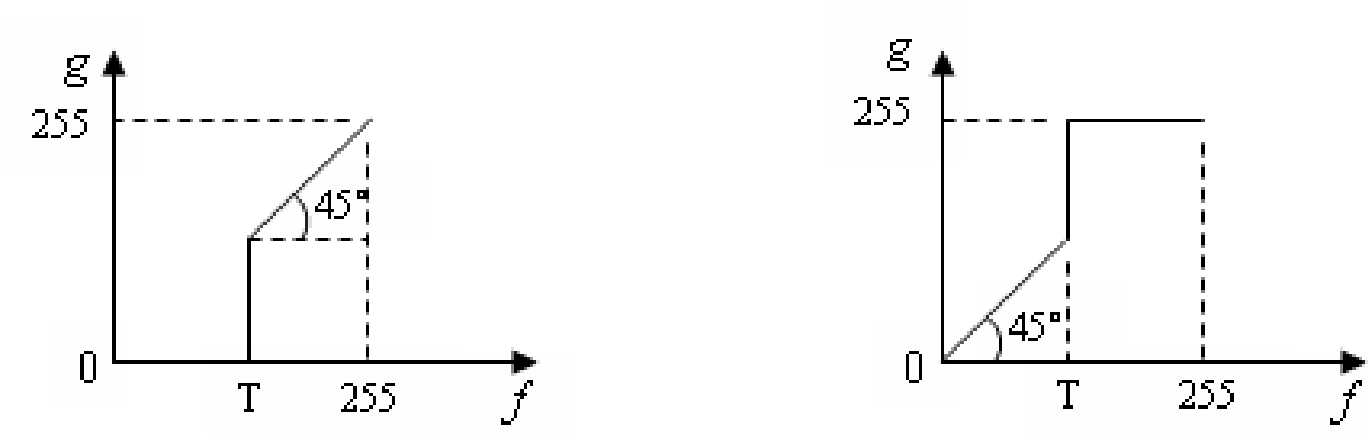
半阈值化方法是将比阈值大的亮像素的灰度级保持不变，而将比阈值小的暗像素变为黑色；或将比阈值小的暗像素的灰度级保持不变，而将比阈值大的亮像素变为白色。利用半阈值化方法分割后的图像可定义为：

$$g(x,y)=\begin{cases} f(x,y) & f(x,y)\geq T \\ 0 & f(x,y)<T \end{cases} \qquad (7.12)$$

$$g(x,y)=\begin{cases} f(x,y) & f(x,y)\leq T \\ 255 & f(x,y)>T \end{cases} \qquad (7.13)$$

7.3.1　基于阈值的分割方法

2.半阈值化分割方法（续）



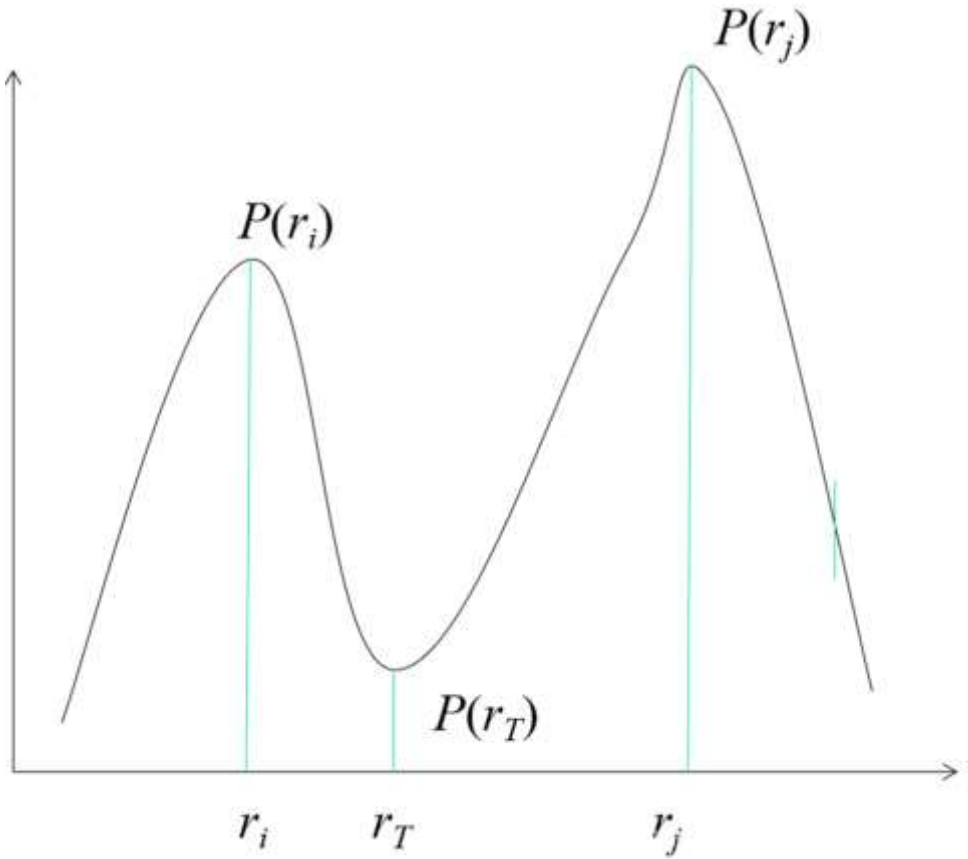
（a）式（7.12）的图示　（b）式（7.13）的图示

图7.13 半阈值化的图示

7.3.2 阈值选取方法

1. 利用极大值和极小值寻找谷底及其阈值

首先求图像 $f(x,y)$ 的直方图 $P(f)$，在该直方图具有双峰特征的情况下，接着在直方图$P(f)$上找出两个局部极大值（或最大值），并设这两个局部极大值的位置分别为r_i和r_j；然后找出位于r_i和r_j之间的具有最小值$P(r_T)$的点r_T，也即对于直方图上所有满足$r_i \le r \le r_j$的位置点，有$P(r_T) \le P(r)$。



7.3.2 阈值选取方法

1. 利用极大值和极小值寻找谷底及其阈值（续）

接下来可以进一步测定直方图双极性的强弱，也即计算：

$$K_T = \frac{P(r_T)}{\min\left(P(r_i), P(r_j)\right)} \qquad (7.14)$$

当K_T的值较小，说明谷低峰高，认为在该直方图的两个峰值之间存在一个有用的阈值K_T，并可以将其作为进行基于阈值的图像分割方法的一个可用的阈值；

当K_T的值较大，特别是接近1时，说明谷和峰的高度比较接近，说明该直方图的双极性较弱，这时若利用阈值K_T进行基于阈值的图像分割，显然很难得到满意的分割效果。

7.3.2　阈值选取方法

2. 全局阈值的选取

如果图像中的背景具有同一灰度值，或背景的灰度值在整个图像中可几乎看作接近于某一恒定值；而图像中的物体为另一确定的灰度值，或明显可几乎看作接近于另一恒定值，如图(a)；或与背景有明显的灰度级区别，如图(b)，则可使用一个固定的全局阈值一般会取得较好的分割效果。

(a)物体和背景分别具有恒定灰度值　(b)物体与背景有明显的灰度区别

图7.14　全阈值选取示例

7.3.2 阈值选取方法

3. 类二值图像的阈值选取

- 类二值图像是指在图像处理中，通过特定的算法将图像转换为仅包含两种像素值（通常是0和1，或黑和白）的图像。
- 当图像可看作是一幅类二值图像，并且大约已知该类二值图像灰度分布的百分比时，就可通过试探的方法选取阈值，直到阈值化后的图像的效果达到最佳为止。
- 比如，分离一个文字约占印刷全页20%的文本图像中的文本时，就可以选择一个与其对应的灰度阈值（也即使灰度值小于阈值的像素点数达到约占总像素点数的20%），并通过多次更改阈值的阈值化试探过程，达到最佳分割效果。

7.3.2 阈值选取方法

4. 迭代式阈值的选取

基本思路是：首先根据图像中物体的灰度分布情况，选取一个近似阈值作为初始阈值，一个比较好的方法就是将图像的灰度均值作为初始阈值；然后通过分割图像和修改阈值的迭代过程来获得认可的最佳阈值。

7.3.2 阈值选取方法

4. 迭代式阈值的选取

迭代式阈值选取过程可描述为：

① 选取一个初始阈值T；

② 利用阈值T将给定图像分割成两组图像，记为R_1和R_2；

③ 计算R_1和R_2均值μ_1和μ_2；

④ 选择新的阈值T，且$T = \frac{\mu_1+\mu_2}{2}$

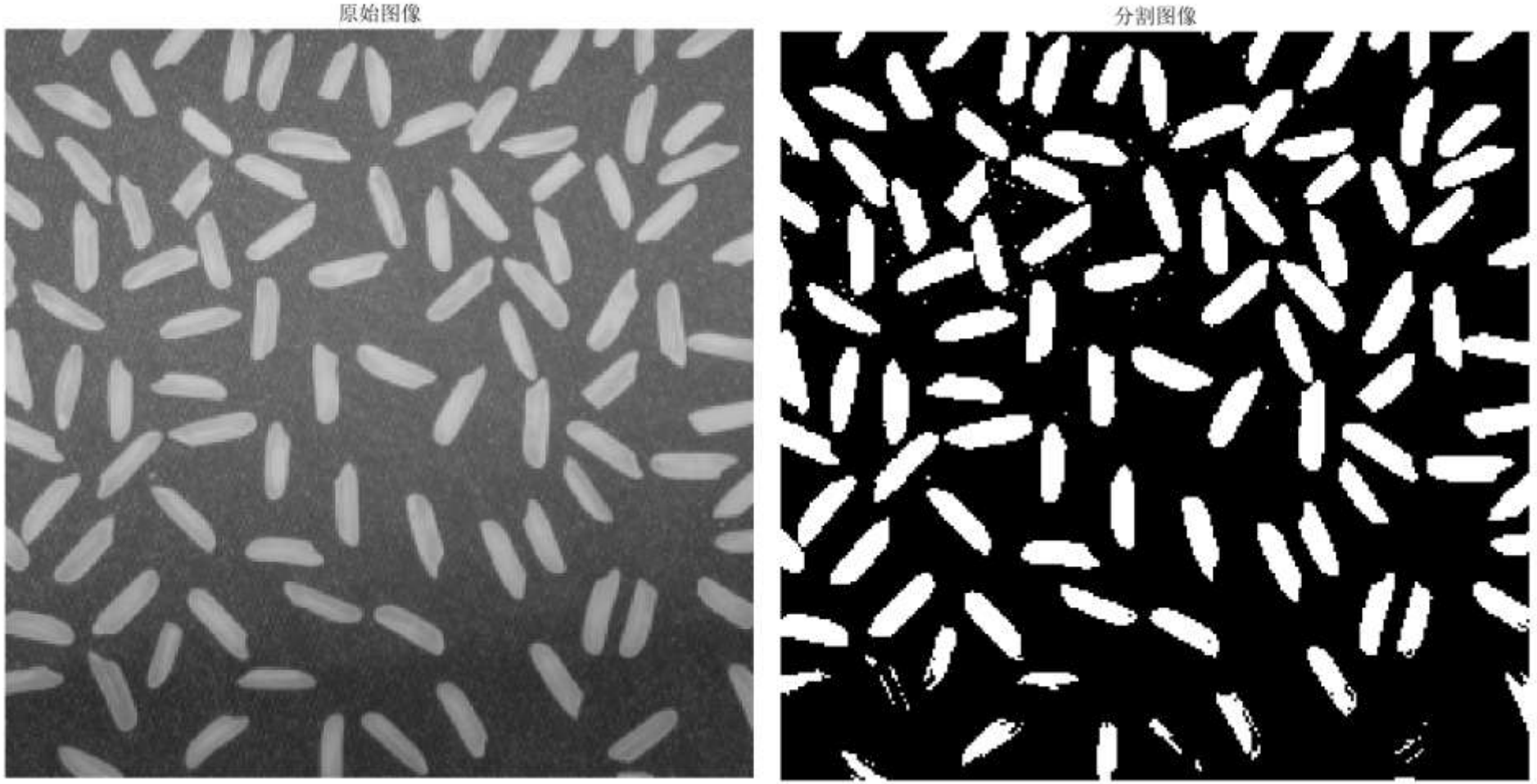
⑤ 重复第②至④步，直至R_1和R_2的均值μ_1和μ_2不再变化为止，也即直至两次计算所得的T值（前一次的T值pre_T与当前的T值now_T）之差（diff_T=abs(pre_T – now_T），十分接近为止（理想情况是等于0，但实际中只要0≤diff_T≤1即可）。

7.3.2 阈值选取方法

diedaiyuzhi.m

```
%迭代阈值选取分割
A = imread('rice.png');
subplot(1,2,1);imshow(A);title('原始图像','FontSize',12);
T = mean2(A);%取均值作为初始阈值
done = false ;%定义跳出循环的理由
i= 0;
   %while循环进行迭代
while ~done
   r1 = find(A<=T);%小于阈值的部分
   r2 = find(A>T);%大于阈值的部分
   Tnew = (mean(A(r1)) + mean(A(r2))) / 2;%计算分割后两部分的阈值均值的均值
   done = abs(Tnew - T)<1;%判断迭代是否收敛
   T= Tnew;%如不收敛,则将分割后的均值的均值作为新的阈值进行循环计算
   i = i+1;
end
A(r1) = 0;%将小于阈值的部分赋值为0
A(r2) = 1;%将大于阈值的部分赋值为1这两步是将图像转换成二值图像
subplot(1,2,2);imshow(A,[]);title('分割图像','FontSize',12);
```

7.3.2　阈值选取方法



7.3.2　阈值选取方法

例3：diedaiyuzhi2

```
clear,clc,close all;
Image=im2double(rgb2gray(imread('lotus.jpg')));
subplot(1,2,1);imshow(Image),title('(a)原始图像','FontSize',14);
T=(max(Image(:))+min(Image(:)))/2;
equal=false;
while ~equal
    rb=find(Image>=T);
    ro=find(Image<T);
    NewT=(mean(Image(rb))+mean(Image(ro)))/2;
    equal=abs(NewT-T)<1/256;
    T=NewT;
end
result=imbinarize(Image,T);
subplot(1,2,2);imshow(result),title('(b)迭代方法二值化图像 ','FontSize',14);
% imwrite(result,'lotus1diedai.jpg');
```

7.3.2　阈值选取方法

例3：

(a)原始图像

(b)迭代方法二值化图像

7.3.2　阈值选取方法

例2：　diedaiyuzhi3.m

```
%利用迭代式阀值选取方法分制图像的MATLAB程序
clc; clear all;
close all;
img0 = imread('flower.png');
[h,w]= size(img0);
img1 = img0;
%初始化
avg= 0.0;%图像的平均值
for i = 1:h
   for j= 1:w
      avg =avg + double(img0(i,j));
   end
end
Thd = avg./h./w;
%按初始阈值Thd将图像像素分为R1和R2两组,分别计算R1和R2的均值u1和u2
preThd = Thd;
nowThd=Thd;
diffThd=2;
```

7.3.2 阈值选取方法

例2： diedaiyuzhi3.m

```
while(diffThd>1.0)
   preThd = nowThd;
   u1=0.0;    u1_num=0;
   u2 =0.0;    u2_num =0;
   for i = 1:h
      for j= 1:w
         if double( img0(i, j))< preThd
            u1 =u1 + double(img0(i,j));
            u1_num=u1_num+1;
         else
            u2 =u2 + double(img0(i,j));
            u2_num=u2_num+1;
         end
      end
   end
   nowThd =(u1./u1_num+u2./u2_num)./2;
   diffThd = abs(preThd - nowThd);
end
```

```
%根据最终的分割阀值nowThd 二值化(分割)图像
for i= 1:h
   for j= 1:w
      if double(img0(i,j))< nowThd
         img1(i,j)=0;
      else
         img1(i,j)= 255;
      end
   end
end
%显示原图像
subplot(1,2,1);imshow(img0);title('原图像');
subplot(1,2,2);imshow(img1),title('分割结果');%分割结果图像
```

7.3.2 阈值选取方法

7.3.2 阈值选取方法

5.基于模式分类思路的阈值选择

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，将数据分为不同类别，从而实现图像分割。

将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

7.3.2 阈值选取方法

5.基于模式分类思路的阈值选择

- 大津法（OTSU）是一种确定图像二值化分割阈值的算法，由日本学者大津于1979年提出。
- 从大津法的原理上来讲，该方法又称作最大类间方差法，因为按照大津法求得的阈值进行图像二值化分割后，前景与背景图像的类间方差最大。
- 大津法被认为是图像分割中阈值选取的最佳算法，计算简单，不受图像亮度和对比度的影响，因此在数字图像处理上得到了广泛的应用。

7.3.2 阈值选取方法

5.基于模式分类思路的阈值选择

- 最大类间方差法（OTSU）

各级灰度出现的概率为：

$$p_i = \frac{n_i}{M \times N}, i = 0,1,2,\cdots,L-1$$

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：

$$\mu_O = \frac{1}{p_O}\sum_{i=0}^{T} i \times p_i$$

$$\mu_B = \frac{1}{p_B}\sum_{i=T+1}^{L-1} i \times p_i$$

$$\mu = p_O \times \mu_O + p_B \times \mu_B$$

7.3.2 阈值选取方法

5.基于模式分类思路的阈值选择

- 最大类间方差法（OTSU）

两类方差：

$$\sigma_O^2 = \frac{1}{p_O}\sum_{i=0}^{T} p_i(i-\mu_O)^2 \qquad \sigma_B^2 = \frac{1}{p_B}\sum_{i=T+1}^{L-1} p_i(i-\mu_B)^2$$

类内方差和类间方差：

$$\sigma_{in}^2 = p_O \cdot \sigma_O^2 + p_B \cdot \sigma_B^2$$

$$\sigma_b^2 = p_O \times (\mu_O - \mu)^2 + p_B \times (\mu_B - \mu)^2$$

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。

7.3.2 阈值选取方法

5.基于模式分类思路的阈值选择

- **最大类间方差法（OTSU）**

应用：求图像全局阈值的最佳方法，适用于大部分需要求图像全局阈值的场合。

优点：计算简单快速，不受图像亮度和对比度的影响。

缺点：对图像噪声敏感；只能针对单一目标分割；当目标和背景大小比例悬殊、类间方差函数可能呈现双峰或者多峰时效果不好。

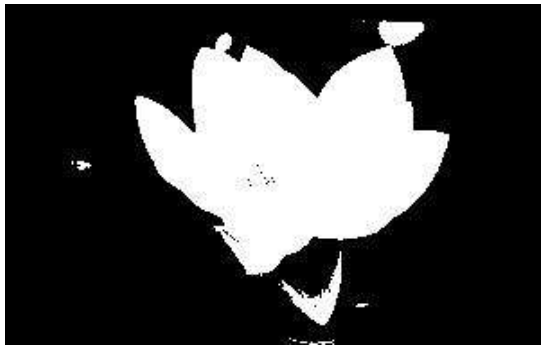
7.3.2　阈值选取方法

5.基于模式分类思路的阈值选择

最大类间方差法（OTSU）

```
Image=rgb2gray(imread('lotus1.jpg'));
figure,imshow(Image),title('原始图像');
T=graythresh(Image);%使用最大类间方差法找到图片的一个合适的阈值。
result=imbinarize(Image,T);
figure,imshow(result),title('OTSU方法二值化图像 ');
```

原始图像



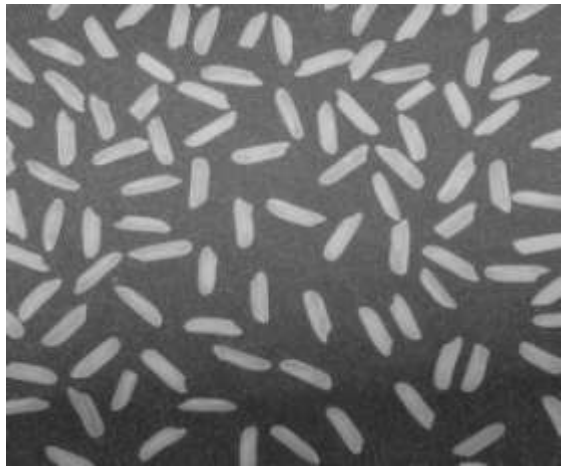
OTSU方法二值化图像

7.3.2 阈值选取方法

5.基于模式分类思路的阈值选择

最大类间方差法（OTSU）

```
I = imread('rice.png');            %读取图像
subplot(1,2,1);imshow(I); title('原始图像');          %显示原始图像
O = graythresh(I)                  %计算得到最大类间方差的阈值
BW1 = im2bw(I,O);                  %使用得到的最大类间方差得到的阈值进行图像分割
subplot(1,2,2);imshow(BW1);    title('OTSU方法二值化图像'); %显示阈值分割后的图像
```



目录

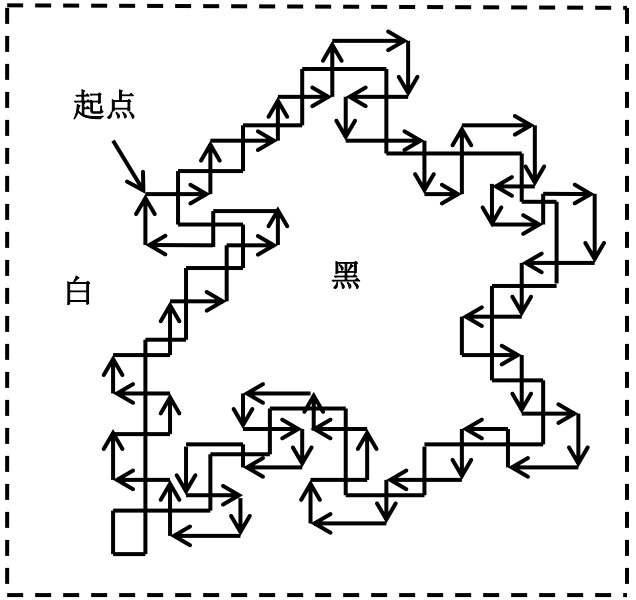
- 图像分割的概念
- 基于边缘检测的图像分割
- 基于阈值的图像分割
- **基于跟踪的图像分割**
- 基于区域的图像分割

7.4 基于跟踪的图像分割

基于跟踪的图像分割方法：先通过对图像上的点的简便运算，来检测出可能存在的物体上的点，然后在检测到的点的基础上通过跟踪运算来检测物体的边缘轮廓的一种图像分割方法。

这种方法的特点是跟踪计算不需要在每个图像点上都进行，只需要在已检测到的点和正在跟踪的物体的边缘轮廓延伸点上进行即可。

7.4.1　轮廓跟踪法



1、轮廓跟踪图像分割算法

(1)在靠近边缘处任取一起始点，然后按照每次只前进一步，步距为一个像素的原则开始跟踪。

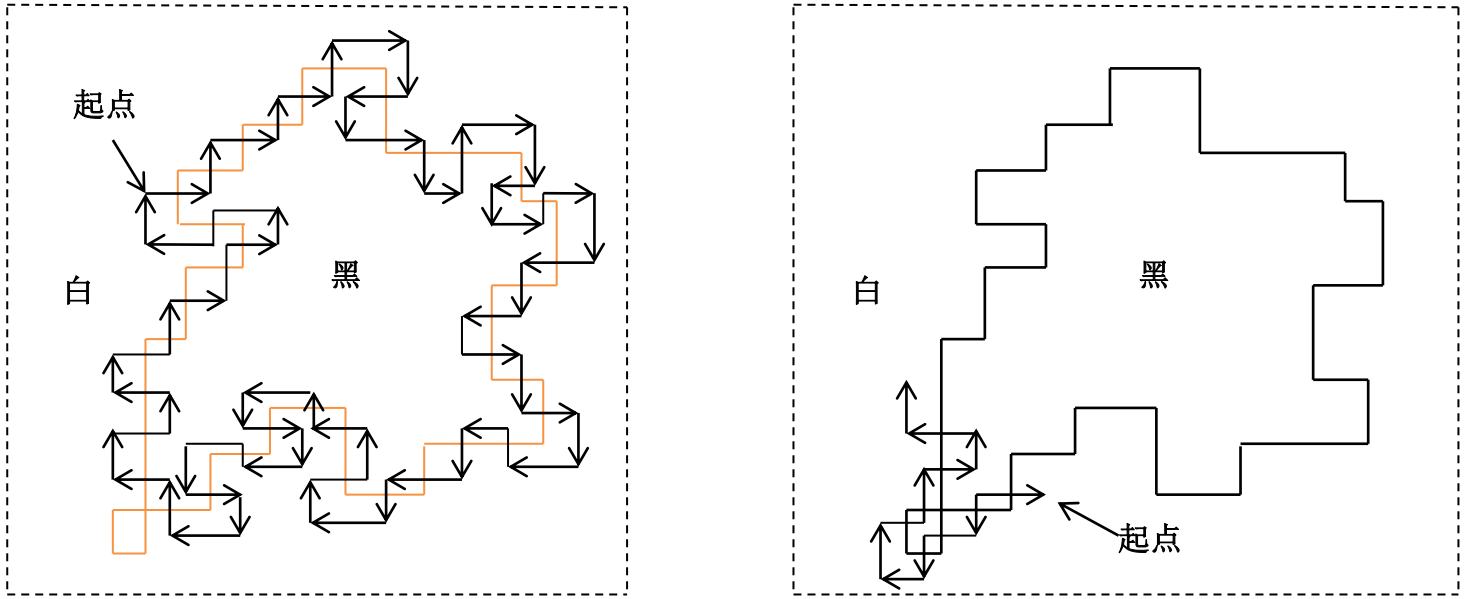
(2)当跟踪中的某步是由白区进入黑区时，以后各步向左转，直到穿出黑区为止。

(3)当跟踪中的某步是由黑区进入白区时，以后各步向右转，直到穿出白区为止。

(4)当围绕目标边界循环跟踪一周回到起点时，则所跟踪的轨迹便是目标的轮廓；否则，应继续按(2)和(3)的原则进行跟踪。

7.4.1　轮廓跟踪法

设图像是由黑色物体和白色背景组成的二值图像。轮廓跟踪的目的是找出目标的边缘轮廓，如图7.16示。



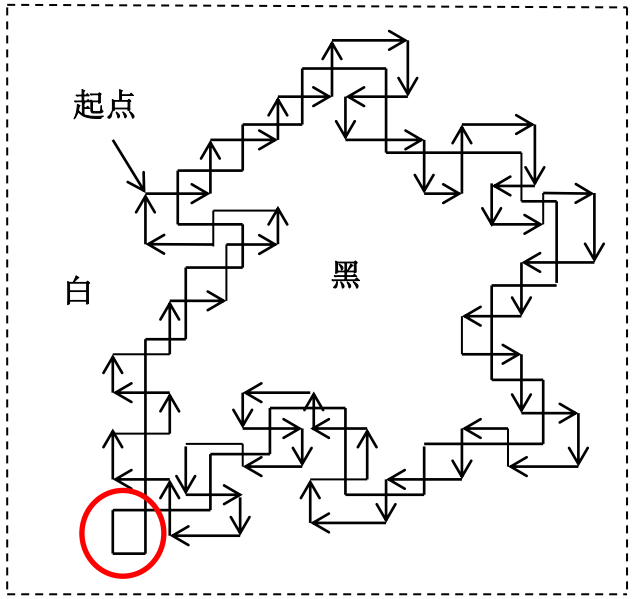
（a）轮廓跟踪过程　（b）利用不同起点跟踪小凸部分

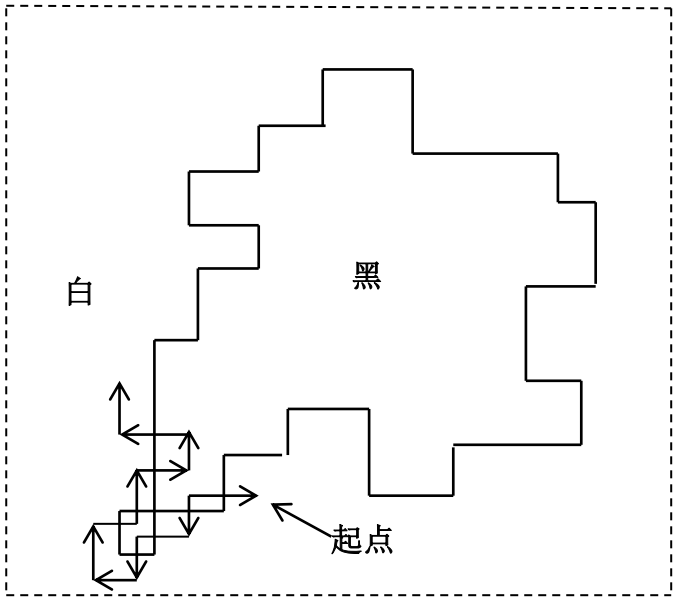
图7.16 轮廓跟踪法示例

7.4.1　轮廓跟踪法

2、在轮廓跟踪中需要注意的问题

（1）在目标中的某些小凸部分可能因被迂回过去而被漏掉，如图7.16(a)左下部所示。避免这种情况的常用方法是选取不同的多个起始点（如图7.16(b)）进行多次重复跟踪，然后将相同的跟踪轨迹选作为目标轮廓。





7.4.1　轮廓跟踪法

2、在轮廓跟踪中需要注意的问题

（2）由于这种跟踪方法可形象地看作是一个爬虫在爬行，所以又称为“爬虫跟踪法”。当出现围绕某个局部的闭合小区域重复爬行而回不到起点时，就出现了爬虫掉进陷阱的情况。

防止爬虫掉进陷阱的一种方法是让爬虫具有记忆能力，当爬行中发现在走重复的路径时，便退回原起始点A，并重新选择起始点和爬行方向重新进行轮廓跟踪。

7.4.1　轮廓跟踪法

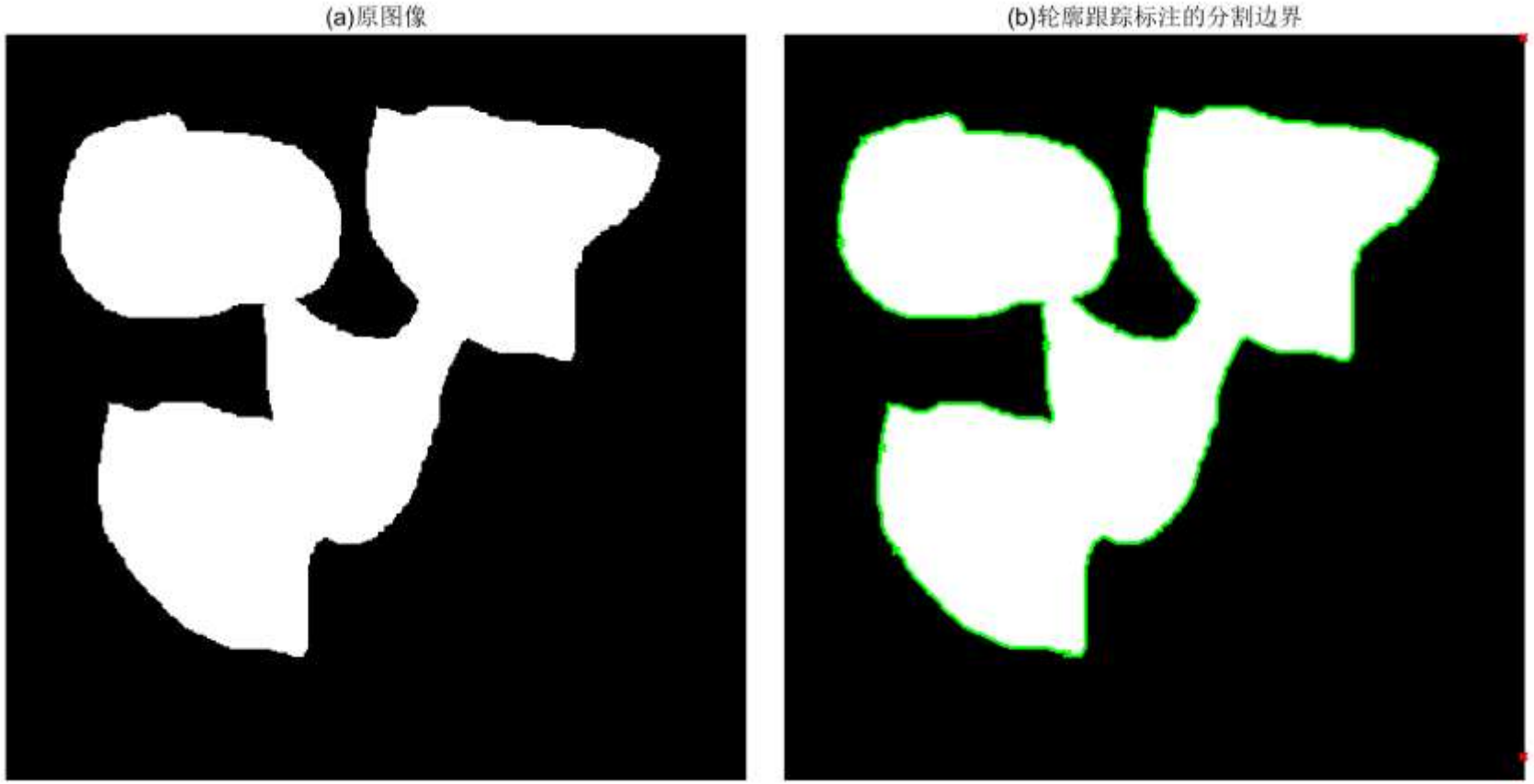
```
%轮廓跟踪
clc;
clear all;
close all;
img0= imread('round.jpg');
subplot(1,2,1);
imshow(img0);title('(a)原图像','FontSize',14);%显示原图像
threshold=graythresh( img0);%计算灰度图像转换为二值图像的门限
f_W= im2bw(img0,threshold);%将灰度图像转化为二值图像
subplot(1,2,2);imshow(f_W);title('(b)轮廓跟踪标注的分割边界','FontSize',14);
num_points=inf;%设定边界轮廓的总像素点个数，默认值为inf
[h,w1]= size(f_W);%图像矩阵的行和列号
```

7.4.1 轮廓跟踪法

```
for h=2:50:h%以(垂直向下的)行步长50快速找边界轮廓点
    for w=1:w1
        if f_W(h,w)
            %如果像素点为1，则找到了边缘起点
            break;
        end
    end
%在二值图像中追踪对象边界，非零像素代表目标，零像素为背景。
    contour = bwtraceboundary(f_W,[h,w],'W',8,num_points,'counterclockwise');%检测是否为边界
    if(~isempty(contour))
        %如果非空为真，说明是边界轮廓点
        plot(contour(:,2),contour(:,1),'g','Linewidth',2);%画出边界
        hold on;
        plot(w,h,'gx','LineWidth',2);%画出边界起点
    else
        hold on;
        plot(w,h,'rx','LineWidth',2);
    end
end
```

7.4.1 轮廓跟踪法

下图是利用轮廓跟踪方法进行图像分割的验证结果图例。



7.4.1 轮廓跟踪法

bwtraceboundary是MATLAB中用于在二值图像中跟踪对象边界的函数。该函数通过顺序查找边缘点来跟踪边界，适用于具有不同像素值的区域，但每个区域内的像素值是相同的情况。

- B = bwtraceboundary(BW, P, fstep)

  BW：二值图像，非零像素属于对象，零值像素构成背景。

  P：起始点的行和列坐标，格式为 [row column]。

  fstep：初始搜索方向，可以是 "N", "NE", "E", "SE", "S", "SW", "W", "NW" 之一。

  B是返回的边界像素的行号和列号，即[row,col]。

- B = bwtraceboundary(BW, P, fstep, conn) %**带连通性参数的用法**

  conn：像素连通性，可以是 4 或 8。4 表示像素的边缘接触，8 表示像素的边缘或角接触。

  带最大边界像素数和方向参数的用法

- B = bwtraceboundary(BW, P, fstep, conn, m, dir)

  m：要提取的最大边界像素数，默认为 Inf。

  dir：跟踪边界的方向，可以是 "clockwise" 或 "counterclockwise"。

7.4.2　光栅跟踪法

基本思想：

先利用检测准则确定接受对象点，然后根据已有的接受对象点和跟踪准则确定新的接受对象点，最后将所有标记为1且相邻的对象点连接起来就得到了检测到的细曲线。

7.4.2 光栅跟踪法

- 需要事先确定检测阈值d、跟踪阈值t，且要求d>t。
- 检测准则：对图像逐行扫描，将每一行中灰度值大于或等于检测阈值d 的所有点（称为接受对象点）记为1。
- **跟踪准则**：设位于第i行的点(i，j)为接受对象点，如果位于第i+1行上的相邻点(i+1，j-1)、(i+1，j)和(i+1，j+1)的灰度值大于或等于跟踪阈值t，就将其确定为新的接受对象点，并记为1。

7.4.2　光栅跟踪法

光栅跟踪图像分割算法：

（1）确定检测阈值d和跟踪阈值t，且要求d>t；

（2）用检测阈值d逐行对图像进行扫描，依次将灰度值大于或等于检测阈值d的点的位置记为1；

（3）逐行扫描图像，若图像中的（i,j）点为接受对象点，则在第i+1行上找点(i,j)的邻点：(i+1,j-1)、(i+1,j)、(i+1,j+1)并将其中灰度值大于或等于跟踪阈值t的邻点确定为新的接受对象点，将相应位置记为1；

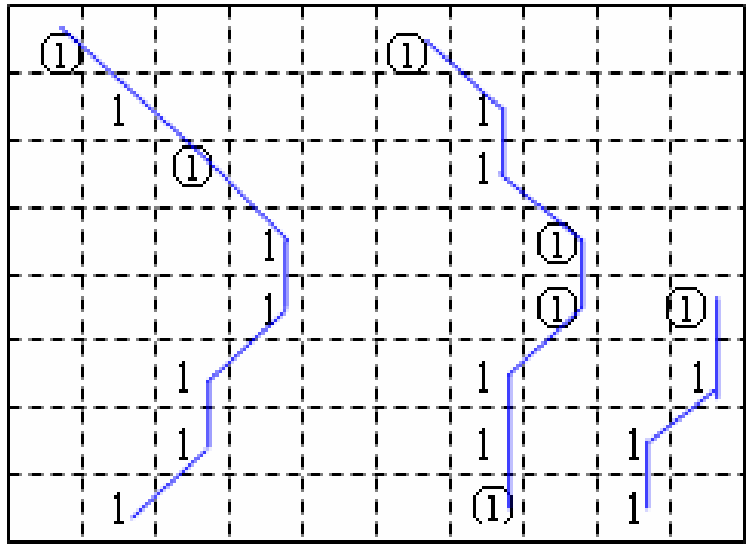
（4）重复步骤(3)，直至图像中除最末一行以外的所有接受点扫描完为止。

7.4.2　光栅跟踪法

例7.5：　d=7，t=4

9	2	5			8	4	5		3
	5		3			6	4	2	
3		9	1	1		5		5	
6			5		4		7	2	
	3		6	2			9	3	7
4		6		3		5	3		5
2		5		2		6	2	6	
6	4		3		2	7	3	4	2

图7.18　例7.5的原图像

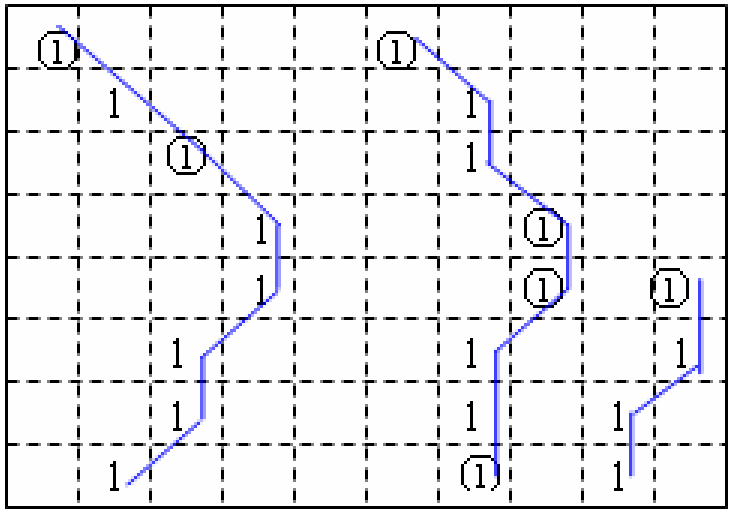


（a）1解题过程和检测结果

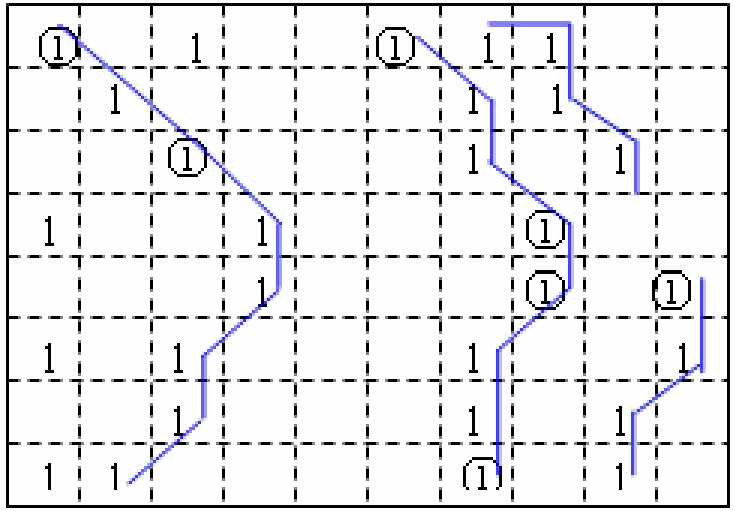
图7.19　例7.5利用光栅跟踪法检测边缘结果图示

7.4.2　光栅跟踪法

例7.5：　d=7，t=4



（a）1解题过程和检测结果



（b）直接取阈值为4时的检测结果

图7.19　例7.5利用光栅跟踪法检测边缘结果图示

7.4.2 光栅跟踪法

注意：检测准则和跟踪准则的判定可以不是灰度值，而是其它反映图像局部特征的量，比如可以是对比度、梯度等。相邻点的定义也可以将范围选得大一些。

目录

- 图像分割的概念
- 基于边缘检测的图像分割
- 基于阈值的图像分割
- 基于跟踪的图像分割
- 基于区域的图像分割

7.5 基于区域的图像分割

基于区域的图像分割是根据图像的灰度、纹理、颜色和图像像素统计特征的均匀性等图像的空间局部特征，将图像中的像素划归到各个物体或区域中，进而将图像分割成若干个不同区域的一种分割方法。

7.5.1 区域生长法

区域生长法（region growing）的基本思想是根据事先定义的相似性准则，将图像中满足相似性准则的像素或子区域聚合成更大区域的过程。

区域生长法从将一幅图像分成许多小区域开始。这些初始的区域可能是小的邻域甚至是单个像素。在每个区域中，对经过适当定义的能反映一个物体内成员隶属程度的性质(度量)进行计算。用于区分不同物体内像素的性质(度量)包括平均灰度值，纹理，或颜色信息。

7.5.1 区域生长法

区域生长法第一步是赋给每个区域一组参数，这些参数的值能够反映区域属于哪个物体。接下来，对相邻区域的所有边界进行考查，相邻区域的平均度量之间的差异是计算边界强度的一个尺度。如果给定边界两侧的度量差异明显，那么这个边界很强，反之则弱。强边界允许继续存在，而弱边界被消除，相邻区域被合并。

7.5.1 区域生长法

区域生长法是一个迭代过程，每一步重新计算被扩大区域的物体成员隶属关系并消除弱边界，当没有消除的弱边界时，区域合并过程结束。这时，图像分割也就完成。检查这个过程会使人感觉这是一个物体内部的区域不断增长直到其边界对应于物体的真正边界为止的过程。

区域增长算法比一些较简单技术的计算开销大，但区域增长能够直接和同时利用图像的若干种性质来决定最终边界的位置。也许它在自然景物的分割方面能显示最佳性能，因为在这种情况下得不到足够的先验知识。

7.5.1 区域生长法

区域生长的基本方法

- 首先，在每个需要分割的区域中找一个“种子”像素作为生长的起点，
- 然后，将种子像素周围邻域中与种子像素有相同或相似性质的像素合并到种子像素所在的区域中，
- 接着，以合并成的区域中的所有像素作为新的种子像素继续上面的相似性判别与合并过程，直到再没有满足相似性条件的像素可被合并进来为止。
- 这样就使得满足相似性条件的像素就组成（生长成）了一个区域。

7.5.1 区域生长法

区域生长法的三个关键条件的确定：

（1）选择和确定一组能正确代表所需区域的种子像素

一般原则为：

① 接近聚类中心的像素可作为种子像素。例如，图像直方图中像素最多且处在聚类中心的像素；

② 红外图像目标检测中最亮的像素可作为种子像素；

③ 按位置要求确定种子像素；

④ 根据某种经验确定种子像素。

7.5.1 区域生长法

区域生长法的三个关键条件的确定：

(2) 确定在生长过程中能将相邻像素合并进来的相似性准则。主要有：

① 当图像是彩色图像时，可以各颜色为准则，并考虑图像的连通性和邻近性；

② 待检测像素点的灰度值与已合并成的区域中所有像素点的平均灰度值满足某种相似性标准，比如灰度值差小于某个值；

③ 待检测点与已合并成的区域构成的新区域符合某个大小尺寸或形状要求等。

7.5.1 区域生长法

区域生长法的三个关键条件的确定：

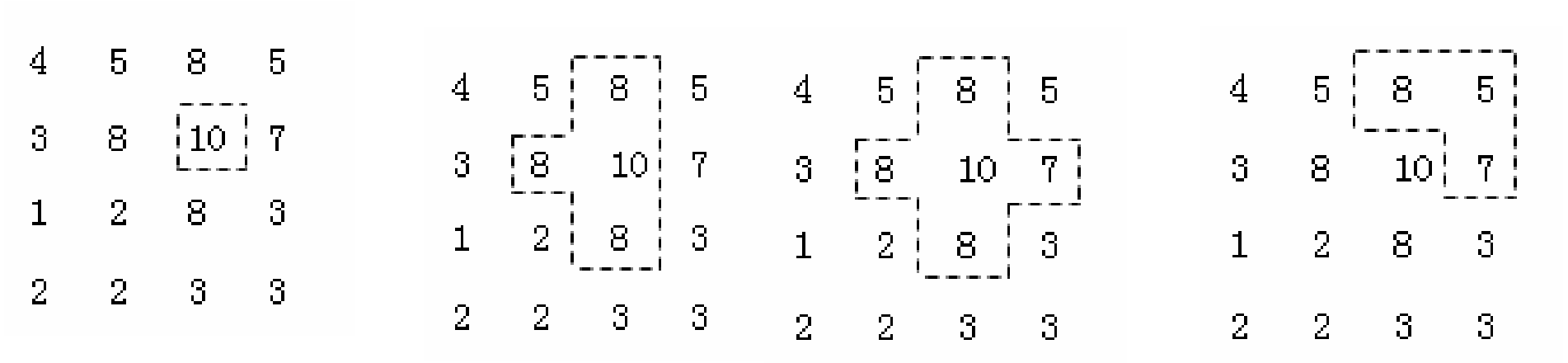
(3) 确定终止生长过程的条件或规则

① 一般的停止生长准则是生长过程进行到没有满足生长准则的像素时为止；

② 其它与生长区域需要的尺寸、形状等全局特性有关的准则。

7.5.1 区域生长法

例7.6 设有原始图像如下图（a）所示，以灰度值最大者为种子像素，以相邻像素与组成物体（种子像素所在区域）的所有像素的平均灰度值之差小于等于2为相似性准则，完成区域生长操作，并对分割结果进行说明。



（a） （b） （c） （d）

b：平均灰度值 7.5=(8+8+8+10)/4

c：平均灰度值 7.2 **d：以5为种子像素**

7.5.1 区域生长法

例7.7 如图7.21(a)为原图像，其中有2个种子像素如虚线框所示。生长准则是：当四周相邻像素与种子像素区域的灰度值差的绝对值小于或等于某个门限T时，将相邻像素合并到种子像素所在的区域。

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

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

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7.21 区域生长示例2

当T=1时，区域生长结果如图7.21（b）所示；

当T=8时，区域生长结果如图7.21（c）所示。

7.5.1　区域生长法

```
clc;
clear all;
close all;
I=imread('coins.png');%读取图像
if isinteger(I)
   I=im2double(I);%将uint类型转换成double类型
end
subplot(1,3,1);imshow(I);title('(a)原图像','FontSize',14);%显示原始图像
[y,x]=getpts;%选取种子点，只能选择一个，选择后按Enter键
x1=round(x);
y1=round(y);
seed=I(x1,y1);
J=zeros(M,N);
J(x1,y1)=1;
sum=seed;
suit=1;
count=1;
threshold=0.15;
```

7.5.1 区域生长法

```
while count>0
   s=0;%记录判断一点周围8个点时，符合条件的新点的灰度值之和
   count=0;
   for i=1:M
      for j=1:N
         if J(i,j)==1%判断此点是否为目标点，下面判断该点的邻域点是否越界
            if(i-1)>0 & (i+1)<(M+1) & (j-1)>0 & (j+1)<(N+1)
               for u= -1:1%判断点周围8个点是否符合生长规则
                  for v= -1:1
                     if J(i+u,j+v)==0 && abs(I(i+u,j+v)-seed)<=threshold&&1/(1+1/15*abs(I(i+u,j+v)-seed))>0.8%判断符合尚未标记，
且满足条件的点
                        J(i+u, j+v)=1;%将满足条件的点其在J中对应位置设置为白
                        count=count+1;
                        s=s+I(i+u,j+v);%此点的灰度值加入s中
                     end
                  end
               end
            end
         end
      end
   end
```

7.5.1　区域生长法

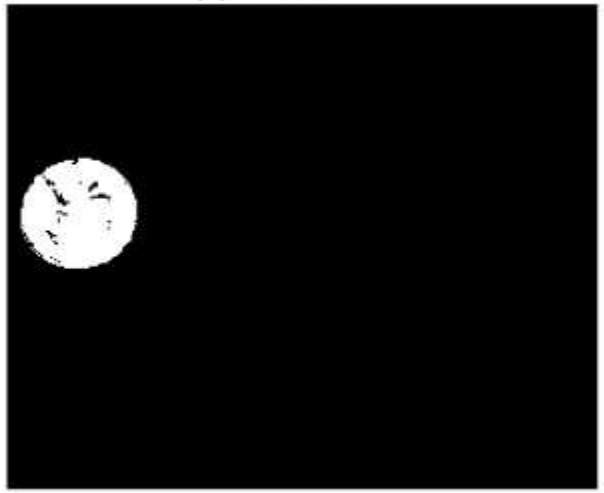
```
suit=suit+count;%将count加入符合点数计数器中
    sum=sum+s;%将s加入符合点的灰度值总和中
    seed=sum/suit;%计算新的灰度平均值
end
subplot(1,3,2);imshow(J);title('(b)区域生长结果','FontSize',14);%显示区域生长结果图
A=I.*J;
subplot(1,3,3);imshow(A);title('(b)区域生长结果与原始图像点乘结果','FontSize',14);
```

7.5.1 区域生长法

下图是利用区域生长方法进行图像分割的验证结果图例，选择1个种子。

(a)原图像

(b)区域生长结果



(b)区域生长结果与原始图像点乘结果

7.5.2 区域分裂合并法

1.区域分裂法

如果区域的特性差别比较大，即不满足一致性准则时，需要采用区域分裂法。分裂过程是从图像的最大区域开始，一般情况下，是从整幅图像开始，区域分裂要注意的两大问题是：

1）确定分裂准则(一致性准则)。

2）确定分类方法，即如何分裂区域，使得分裂后的子区域的特性尽可能都满足一致性准则。

如果用一个阈值T(x)运算来表示区域的一致性准则，其算法步骤如下:

a.形成初始区域。

b.对图像的每个区域R_1，计算$T(R_i)$，如果$T(R_i)$=False，则沿着某一合适的边界分类区域。

c.重复步骤b，当没有区域需要分裂时，算法结束。

7.5.2 区域分裂合并法

2. 区域合并法

区域分裂算法存在的问题是，将整体区域分裂成不同子区域，每个子区域由于区域分类规则不同，导致其区域分割算法结果不同，可能存在某些相连子区域满足一致性准则。因此，需要采用区域合并的方法将这些子区域再合并在一起。

这里假设同样的一致性阈值规则T(x)，进行区域合并，其算法过程如下：

1）获得上节分割区域。

2）对图像中相邻的区域，计算是否满足一致性阈值规则T(x)，满足则合并为一个区域。

3）重复步骤2)，直到没有区域可以合并，算法结束。

7.5.2　分裂合并法

3.分裂-合并分割方法：

根据事先确定的分裂合并准则,也即区域特征一致性的测度。该方法从整个图像出发，根据图像中各区域的不一致性，将图像或区域分裂成新的子区域；

同时，可查找相邻区域有没有相似的特征，当相邻子区域满足一致性特征时，将它们合并成一个较大区域，直至所有区域不再满足分裂和合并的条件为止。

7.5.2　分裂合并法

1、图像四叉树

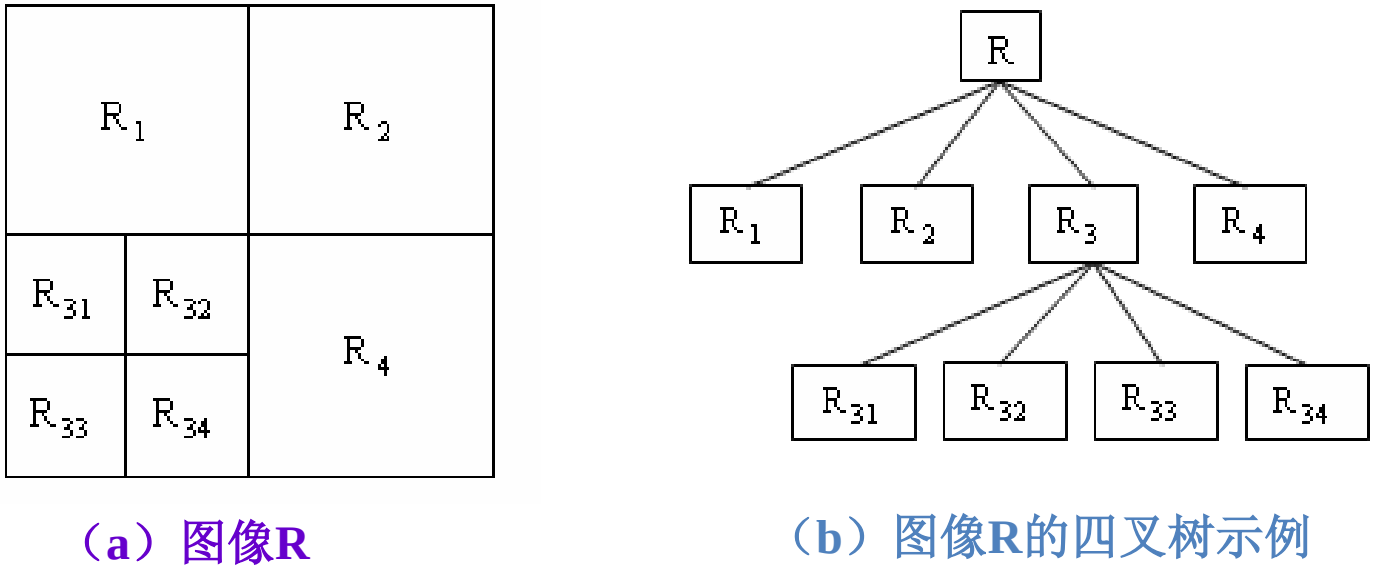
当整个图像或图像中的中某个区域的特征不一致时，就将该图像（区域）分裂成大小相同的4个象限区域，其编号依次为1（左上角）、2（右上角）、3（左下角）、4（右下角）；

并依序分别根据已经分裂得到的新区域的特征是否一致，将特征不一致的区域进一步分成大小相同的4个更小的象限区域；

如此不断继续分割下去，直至所有的更小区域的特征一致为止。

7.5.2 分裂合并法

1、图像四叉树



（a）图像R　　（b）图像R的四叉树示例

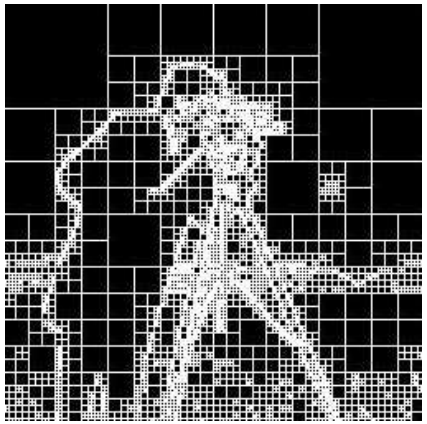
图7.23 图像的四叉树表示

7.5.2 分裂合并法

```
%四叉树分解
clear,clc,close all;
Image=imread('cameraman.tif');
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
blocks(end,1:end)=1;
blocks(1:end,end)=1;
imshow(Image);
figure,imshow(blocks,[]);
```

7.5.2　分裂合并法

原始图像



四叉树分解

7.5.2 分裂合并法

分裂-合并分割法过程

(1)设R_0表示整幅图像，用R_i(i=1，2，3，4)表示原图像分割成的图像区域；并假设当同一区域 R_i中的所有像素满足某一相似度测量准则(认为它们具有相同的性质)时，$P(R_i)$=TRUE，否则$P(R_i)$=FALSE

(2)对于任何的区域R_i，当$P(R_i)$=TRUE时，该区域及该分支不再进一步分裂。

(3)对于任何的区域R_i，当$P(R_i)$=FALSE时，将该区域分成大小相同的4个更小的象限区域R_{i1}、R_{i2}、R_{i3}、R_{i4}。

7.5.2 分裂合并法

2、分裂-合并分割法

(4)在每一次将R_i分成更小的区域R_{i1}、R_{i2}、R_{i3}、R_{i4}后，如果它们中相互相邻的两区域R_{ij}与R_{ik}有$P(R_{ij})$=TRUE 和 $P(R_{ik})$=TRUE，则将它们合并成新区域。如果它们中的区域R_{ij}同时使$P(R_{ij})$=TRUE 和$p(R_{(i-1)k})$=TRUE(i-1 ≥1)成立，或同时使$p(R_{ij})$=TRUE和 $p(R_{(i-2)k})$=TRUE(i-2≥1)成立；则将R_{ij}和$R_{(i-1)k}$合并成新区域，或将R_{ij}和$R_{(i-2)k}$合并成新区域。

7.5.2 分裂合并法

2、分裂-合并分割法

【例7.9】利用分裂-合并法进行图像分割，原图像如图 7.24(a)所示。

解：

(1)将原图像分裂成4个像限区域 R_i(i=1,2,3,4)；为方便起见，后续将其描述成(1,2,3,4)，也即仅用R_i的角标数字描述，如图 7.24(b)所示。

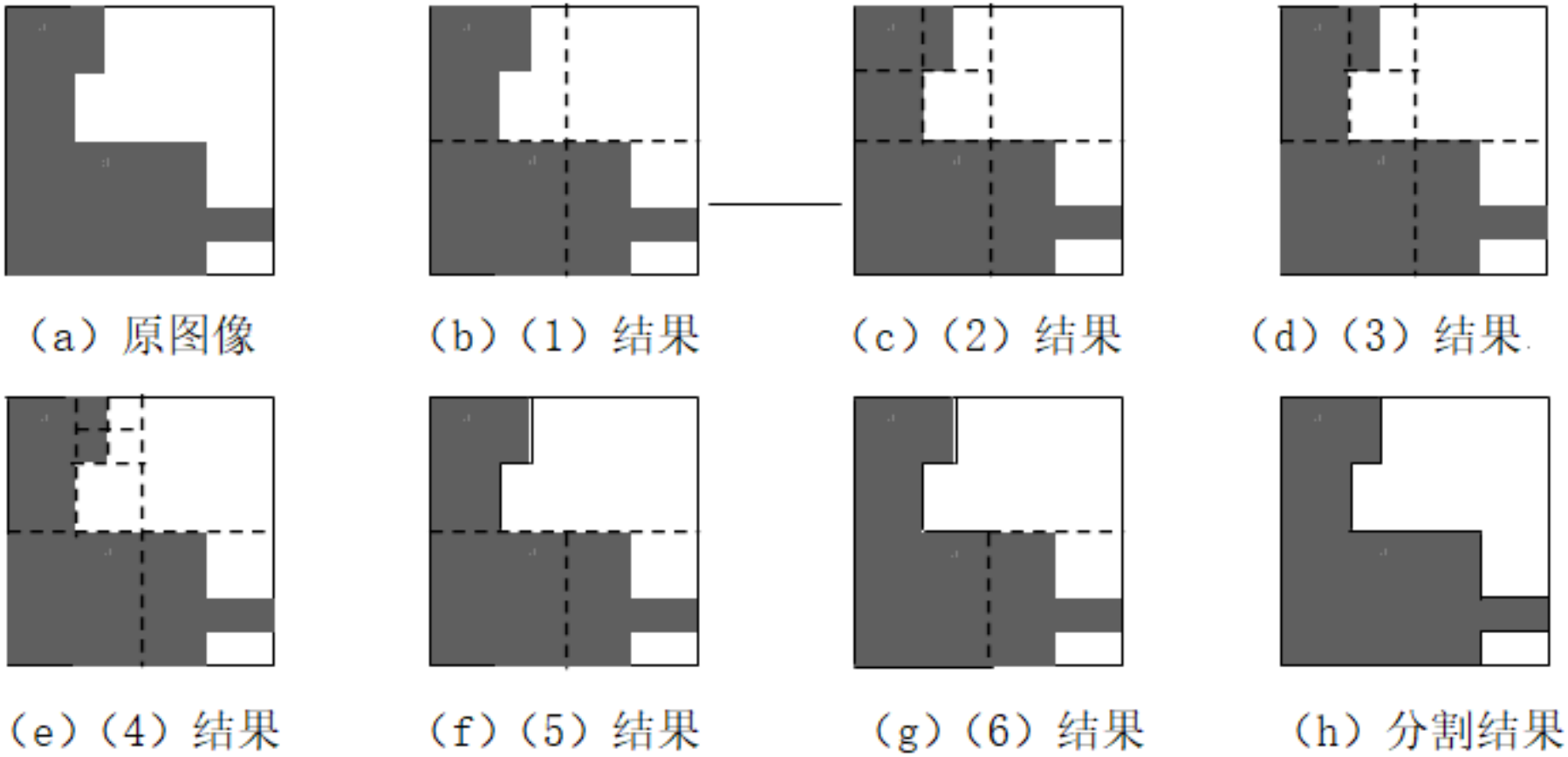
(2)由于图 7.24(b)的区域1使$P(R_1)$=FALSE，进一步将区域1分裂成(11，12，13，14)如图 7.24(c)所示。

(3)由于区域1中的相邻区域R_{11}和R_{13}使$P(R_{11})\cap P(R_{13})$=TRUE 成立，所以合并R_{11}和R_{13}结果如图 7.24(d)所示。

(4)由于图 7.24(d)中区域 12 使得 $P(R_{12})$=FALSE，进一步将区域 12 分裂成(121，122，123，124)如图7.24(e)所示。

7.5.2　分裂合并法

2、分裂-合并分割法



（a）原图像　（b）（1）结果　（c）（2）结果　（d）（3）结果

（e）（4）结果　（f）（5）结果　（g）（6）结果　（h）分割结果

7.5.2 分裂合并法

2、分裂-合并分割法

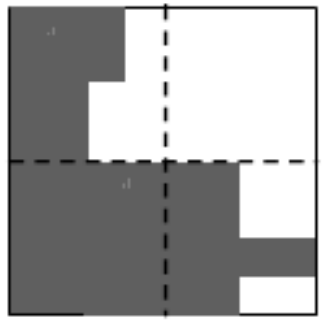
(5)由于区域 12 中的相邻区域R_{121}和R_{123}使$P(R_{121}) \cap P(R_{123})$=TRUE 成立，所以合并$R_{121}$和$R_{123}$，同时，由于成立$P(R_1) \cap P(R_{121}) \cap P(R_{123}$=TRUE，所以将$R_{121}$和$R_{123}$合并到$R_{11}$和$R_{13}$的连片区域，并将该连片区与记为$R_{1\text{-}}$，结果如图 7.24(f)所示。

(6)由于在图7.24(f)中，区域1中的$R_{1\text{-}}$与区域3 相邻，且成立$P(R_1) \cap P(R_3)$=TRUE所以合并$R_{1\text{-}}$和R_3，并将其新连片的区域记为$R_{(1,3)\text{-}}$，结果如图7.24(g)所示。

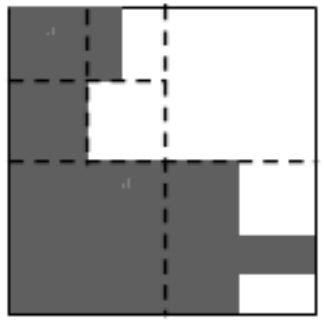
(7)同理，由于$P(R_3)$=FALSE，可将其分裂成R_{31}、R_{32}、R_{33}、R_{34}；然后根据$P(R_{(1.3)\text{-}}) \cap P(R_{31}) \cap P(R_{33}$）=TRUE，将$R_{31}$和$R_{33}$合并到区域$R_{(1,3)\text{-}}$形成更大的分割区域；进而将$R_{341}$和$R_{342}$合并近来，就可得到最终的分割结果如图7.24(h)所示。

7.5.2 分裂合并法

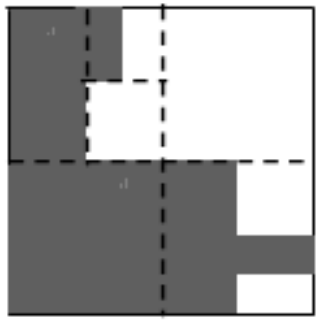
（a）原图像



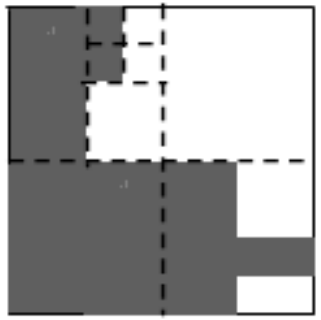
（b）（1）结果



（c）（2）结果



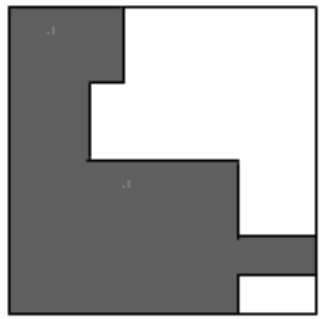
（d）（3）结果



（e）（4）结果

（f）（5）结果

（g）（6）结果



（h）分割结果

7.5.2　分裂合并法

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。

图像分割综合应用

（1）读入图像

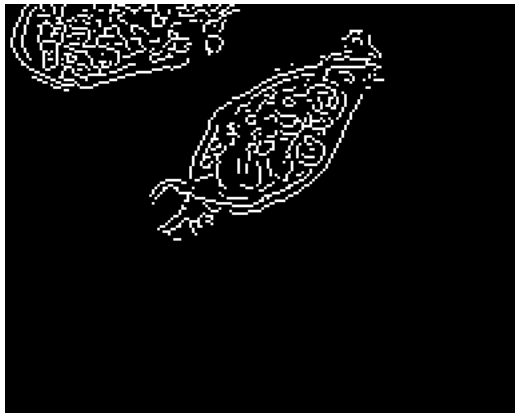
```
I = imread('cell.tif');
figure,imshow(I);
```

(a)原始图像

（2）检测完整的细胞

```
BWs0 = edge(I,'sobel');
figure,imshow(BWs0);
BWs = edge(I,'sobel',(graythresh(I)*.1));
figure,imshow(BWs);
```

(b) sobel算子提取边缘



(c) 利用graythresh函数获得最佳阈值后再用sobel算子提取边缘

图像分割综合应用

（3）填补缝隙

虽然edge函数提取了图像的大概轮廓，但是边缘线存在断裂的情况，没有完整而精确地描绘出细胞的轮廓，在这里，可以通过strel函数利用线性的结构函数对边缘进行膨胀操作，填补边缘缝隙。

```
se90 = strel('line',3,90);
se0 = strel('line',3,0);
```

（4）膨胀操作

用imdilate函数对图像进行膨胀操作，膨胀结果如图(c)所示。

```
BWsDil = imdilate(BWs, [se90 se0]);
figure,imshow(BWsDil);
```

图像分割综合应用

（5）填充

膨胀后的灰度图精确显示了细胞的外围轮廓，但是在细胞内部还有一些孔隙。可以利用imfill函数对这些空袭进行填充，填充结果如图(d)所示。

```
BWDfill = imfill(BWsDil,'holes');
figure,imshow(BWDfill);
```

（c）膨胀操作

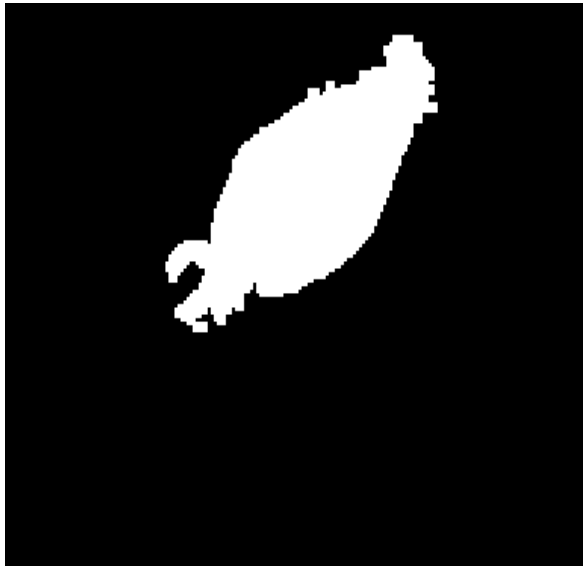
(d) 填充

图像分割综合应用

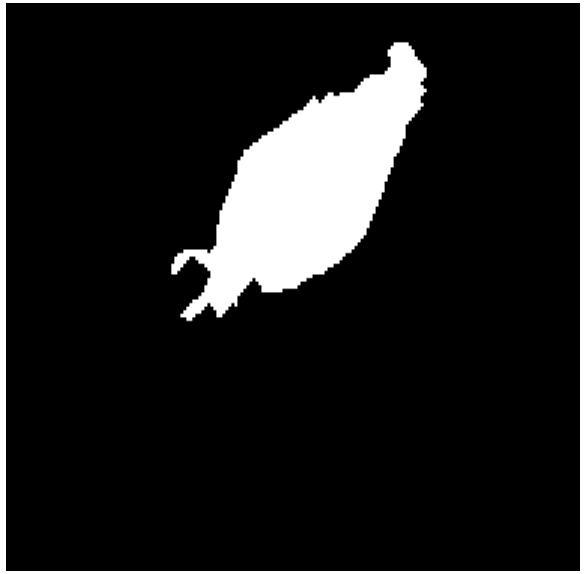
（6）移除与边界连通的目标

至此，可以对感兴趣的细胞进行成功分割，但是画面上还有其他物体，可以通过imclearborder来清楚与边界连通的物体，得到如图(e)所示的分割结果。

```
BWnobord = imclearborder(BWDfill,4);
figure,imshow(BWnobord);
```



图（e）分割



图(f) 平滑

图像分割综合应用

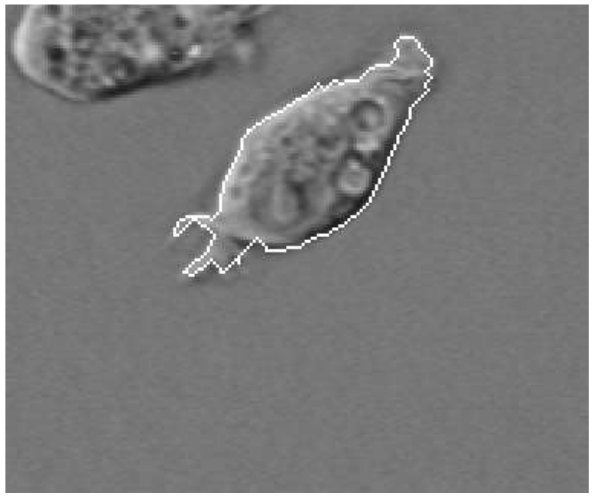
（7）平滑

对于分割结果，边缘不是很光滑，需要利用菱形结构元素对图像进行平滑处理，得到如图(f)所示的平滑处理图。

```
seD =strel('diamond',1); % Think:Why choose 'diamond'as strel?
BWfinal = imerode(BWnobord,seD);
BWfinal = imerode(BWfinal,seD);
figure,imshow(BWfinal);
```

然后在原图上以轮廓线标出细胞的轮廓，至此，细胞的检测就完成了，如图(g)所示的标识结果。

```
BWoutline = bwperim(BWfinal);
Segout = I;
Segout(BWoutline) = 255;
figure,imshow(Segout);
```



图(g)分割结果标记

THE END

Thank You.