第7章　图像分割

（Image Segmentation）

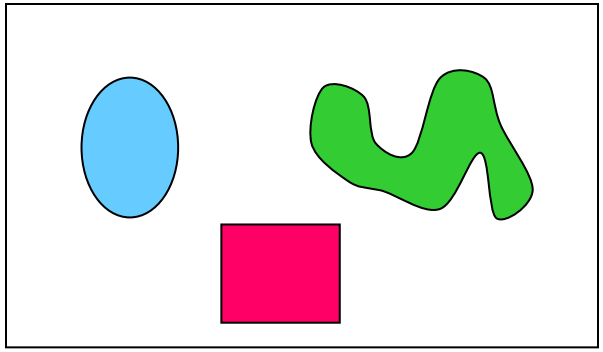
7.1 概述和分类

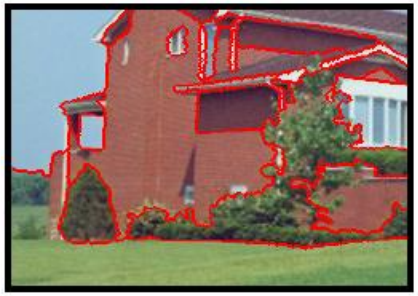
◆ 图像分割

图像分割是把图像分成若干个有意义的区域的处理技术。

是指根据灰度、彩色、空间纹理、几何形状等特征把图像划分成若干个互不相交的区域，使得这些特征在同一区域内，表现出一致性或相似性，而在不同区域间表现出明显的不同。

简单地讲，就是在一幅图像中，把目标从背景中分离出来，以便于进一步处理。



图像分割

借助集合概念进行定义：

令R代表整个图像区域，对R的分割可看做将R分成若干个满足以下条件的非空子集(子区域)$R_1, R_2, R_3, \ldots, R_n$:

① $\bigcup_{i=1}^{N} R_i = R$

即全部子区域的总和（并集）应能包括图像中全部像素。说明分割必须彻底，每一个像点必须处在一个子区域中。

② 对所有的 i 和 j，$i \neq j$，有 $R_i \bigcap R_j = \phi$，ϕ 为空集

在分割结果中，各个子区域互不重叠（*交集为零*），即一个像素不能同时属于两个区域。

③ $P(R_i) = \text{TRUE} \quad \text{for } i = 1, 2, 3, \cdots, N$

$P(R_i)$为作用于图中所有像素的相似性逻辑程度。

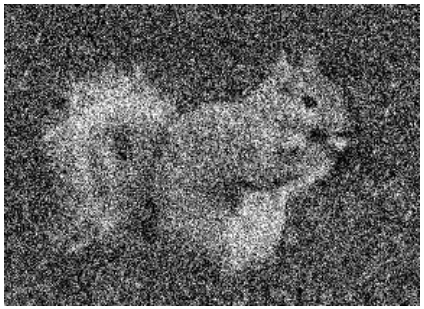
在分割后得到的属于同一个区域中的像素应该具有某些相似性。

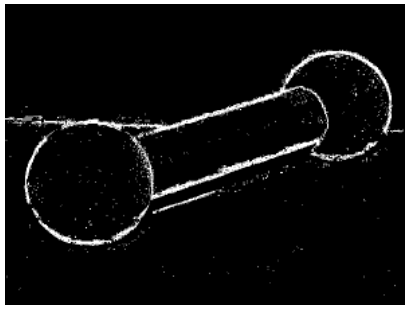
④ $P(R_i \bigcup R_j) = \text{FALSE} \quad \text{for } i \neq j$

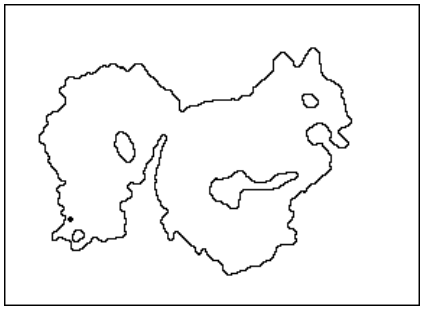
在分割结果中属于不同区域的像素应该具有一些不同的特性。也就是说，在测试准则的意义上，区域R_i和R_j是不同的区域。

⑤ R_i是连通的区域，对 $i = 1, 2, 3, \cdots, N$

要求同一个子区域内的像素应当是连通的。

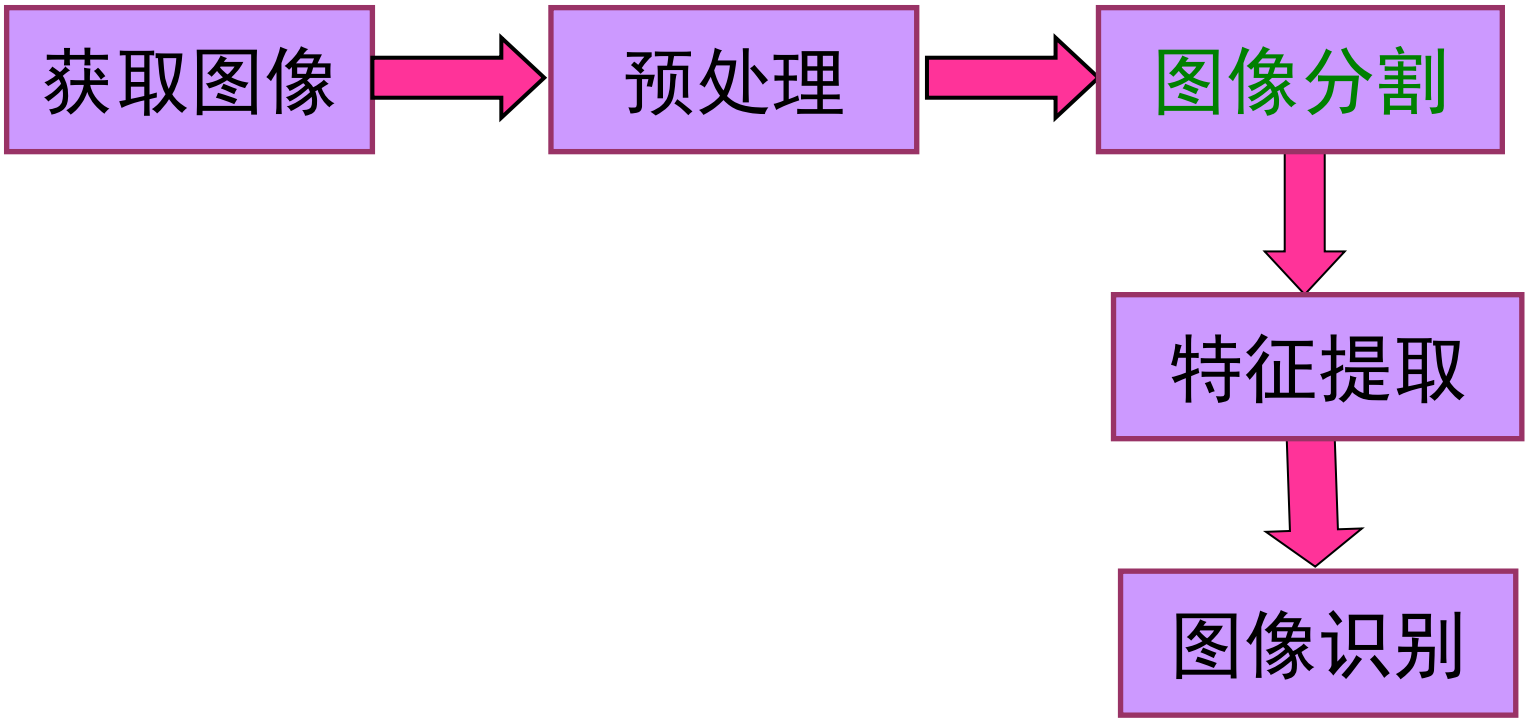






实例

图像分析系统的基本构成:



Basic Descrip.

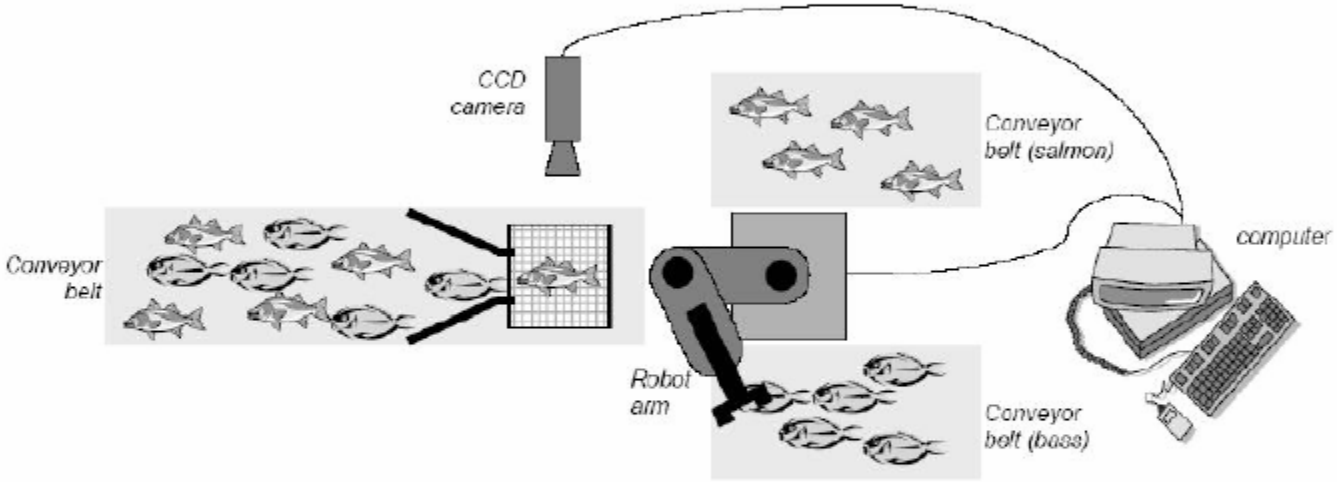
Task: A fish processing plan wants to automate the process of sorting incoming fish according to species (salmon or sea bass) on a conveyor belt using optical sensing.

经验人士告诉我们：鲑鱼（salmon, 即　　大马哈鱼）和鲈鱼（sea base）：鲈鱼一般要比鲑鱼长；鲈鱼要比鲑鱼宽；鲈鱼要比鲑鱼白；鲈鱼的眼睛要比鲑鱼黑；……

如何合理采纳这些经验来进行自动分类？

方案

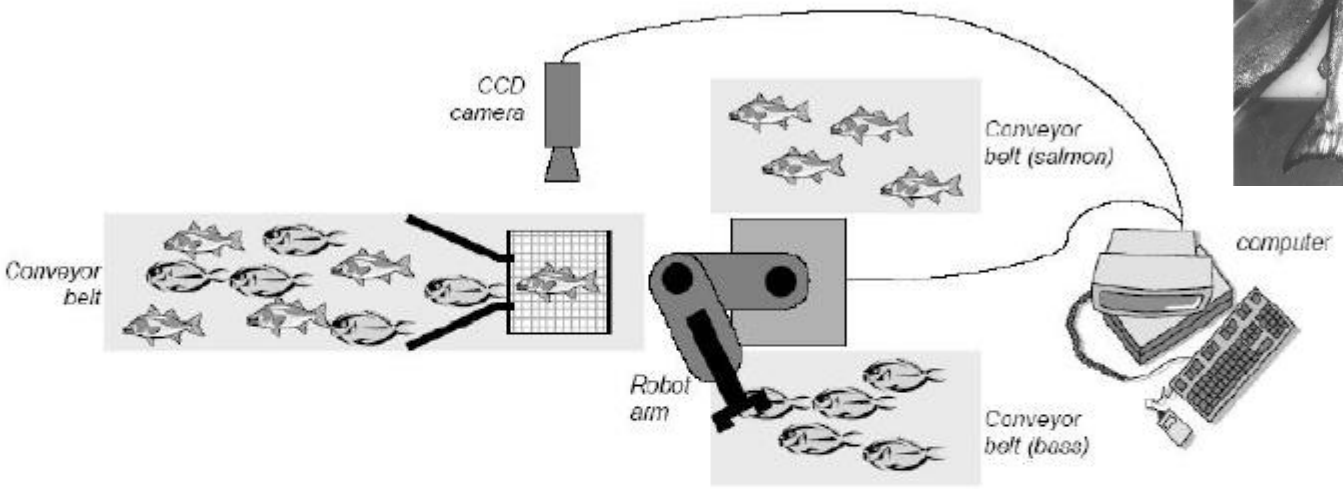
安装摄像头，拍摄传送带上的鱼，分割图像，自动测量鱼体长、宽、颜色、眼睛，据此判断，鱼体长些的是鲈鱼，鱼体宽些的是鲈鱼，颜色白些的是鲈鱼，眼睛黑些的是鲈鱼，否则为鲑鱼。



The automation system consists of

1. a conveyor belt for incoming products
2. two conveyor belts for sorted products
3. a pick-and-place robotic arm
4. a vision system with an overhead CCD camera
5. a computer to analyze images and control the robot arm

方案 Focus



关注问题：

1. 图像采集质量如何？ 图像预处理
2. 图像分割质量如何？ 图像分割
3. 长、宽、鱼体颜色、眼睛颜色如何测量？ 特征提取
4. 如何判别？（即长度大于多少是鲈鱼，等等）……

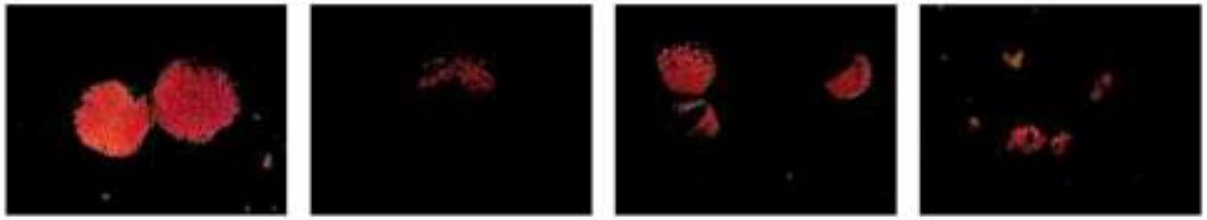
分类器的设计、分类决策

在图像分析过程中，一般首先要对所给的图像进行分割，再对分割的区域作适当的描述，然后才能对图像作某种分析。 可见图像分割是图像分析前的一个重要处理步骤。

图像分割是图像识别的前提，图像分割质量的好坏直接影响后续图像处理的效果。

a. Test images

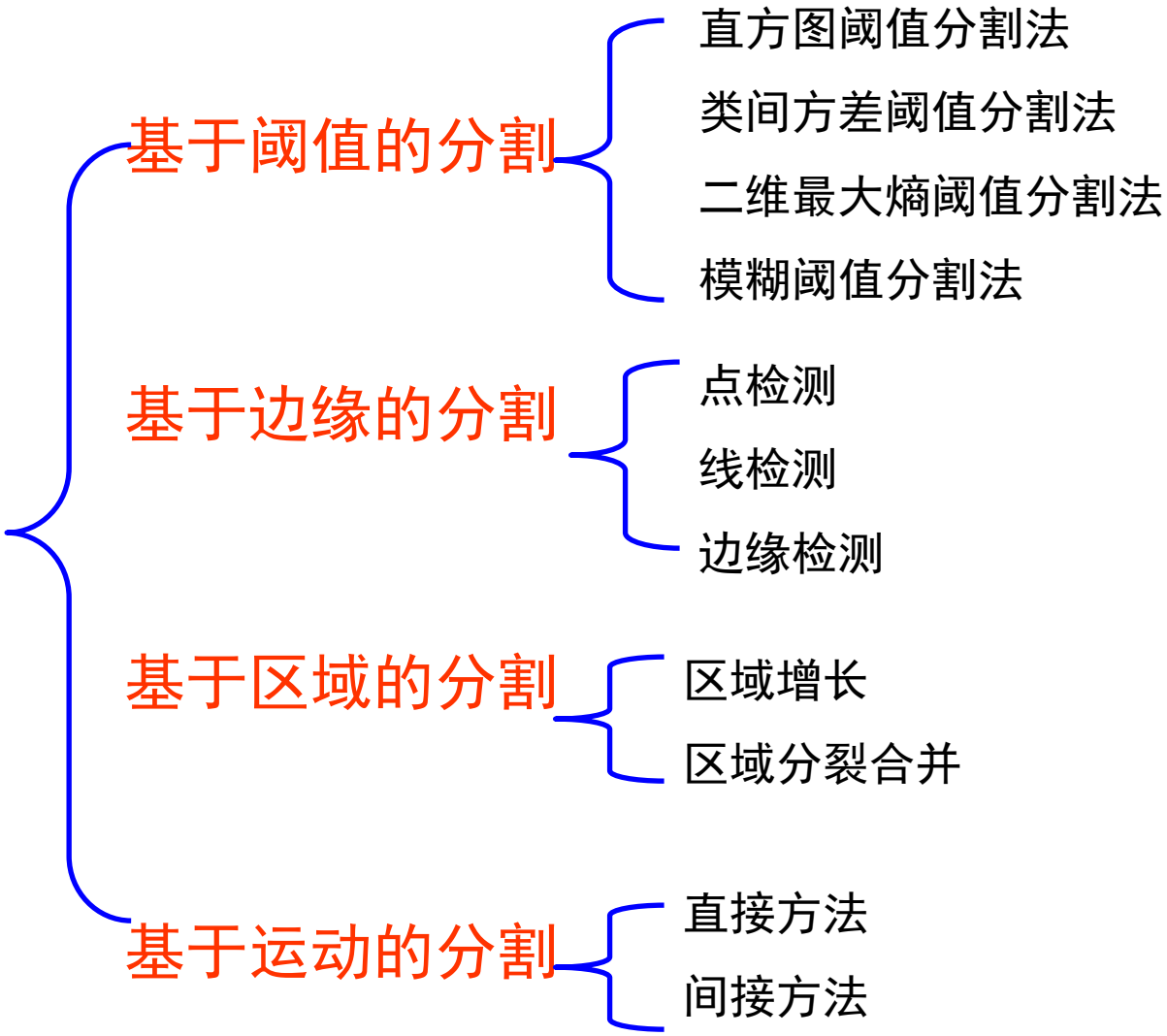
b. K-means



c. Otsu

果园环境下的杨梅图像的分割

图像分割的分类：


基于阈值的分割
直方图阈值分割法
类间方差阈值分割法
二维最大熵阈值分割法
模糊阈值分割法
基于边缘的分割
点检测
线检测
边缘检测
基于区域的分割
区域增长
区域分裂合并
基于运动的分割
直接方法
间接方法

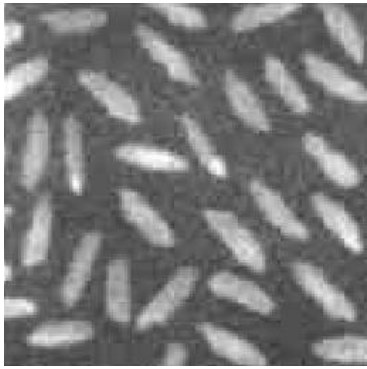
7.2 基于阈值的分割

若图像中要提取的目标物与其背景在灰度特性上存在差异，可把图像视为具有不同灰度级的两类区域（目标和背景）的组合，选取一个合适的阈值，以确定图像中每一个像素点应该属于目标还是背景区域，从而产生相应的二值图像。

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

或

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



另外，还可以将阈值设置为一个灰度范围[t1, t2]，凡是灰度在范围内的像素都变为1，否则皆变为0，即

$$g(x,y)=\begin{cases}1 & t1\le f(x,y)\le t2\\ 0 & \text{其它}\end{cases}$$

某种特殊情况下，高于阈值t的像素保持原灰度级，其它像素都变为0，称为半阈值法，分割后的图像可表示为：

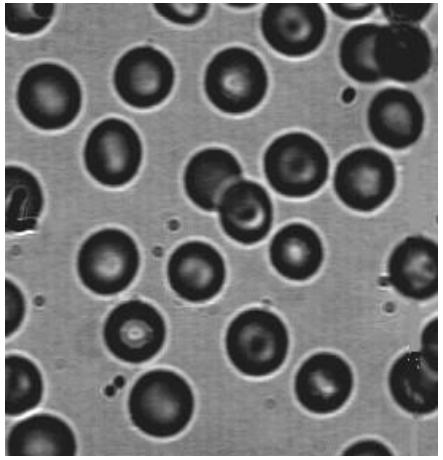
$$g(x,y)=\begin{cases}f(x,y) & f(x,y)\ge t\\ 0 & \text{其它}\end{cases}$$

阈值分割图像的基本原理，可用下式表示：

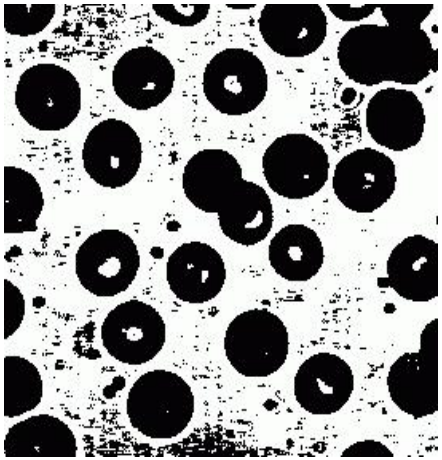
$$g(x,y)=\begin{cases} Z_E & f(x,y)\in Z \\ Z_B & \text{其它} \end{cases}$$

Z为阈值，是图像f(x, y)灰度级范围内的任一个灰度级集合，Z_E和Z_B为任意选定的目标和背景灰度级。

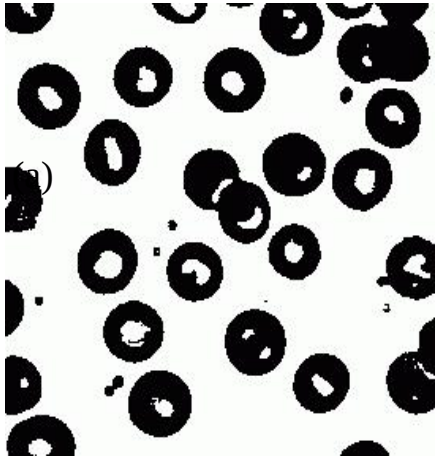
阈值的选取是阈值分割技术的关键，如果过**高**，则过多的目标点被误**归为背景**；如果阈值过**低**，则会出现相反的情况。



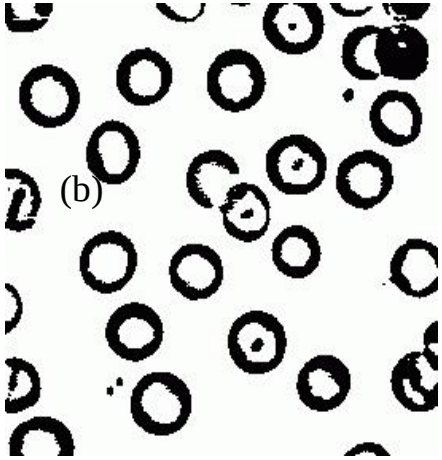
原始图像



阈值T=130

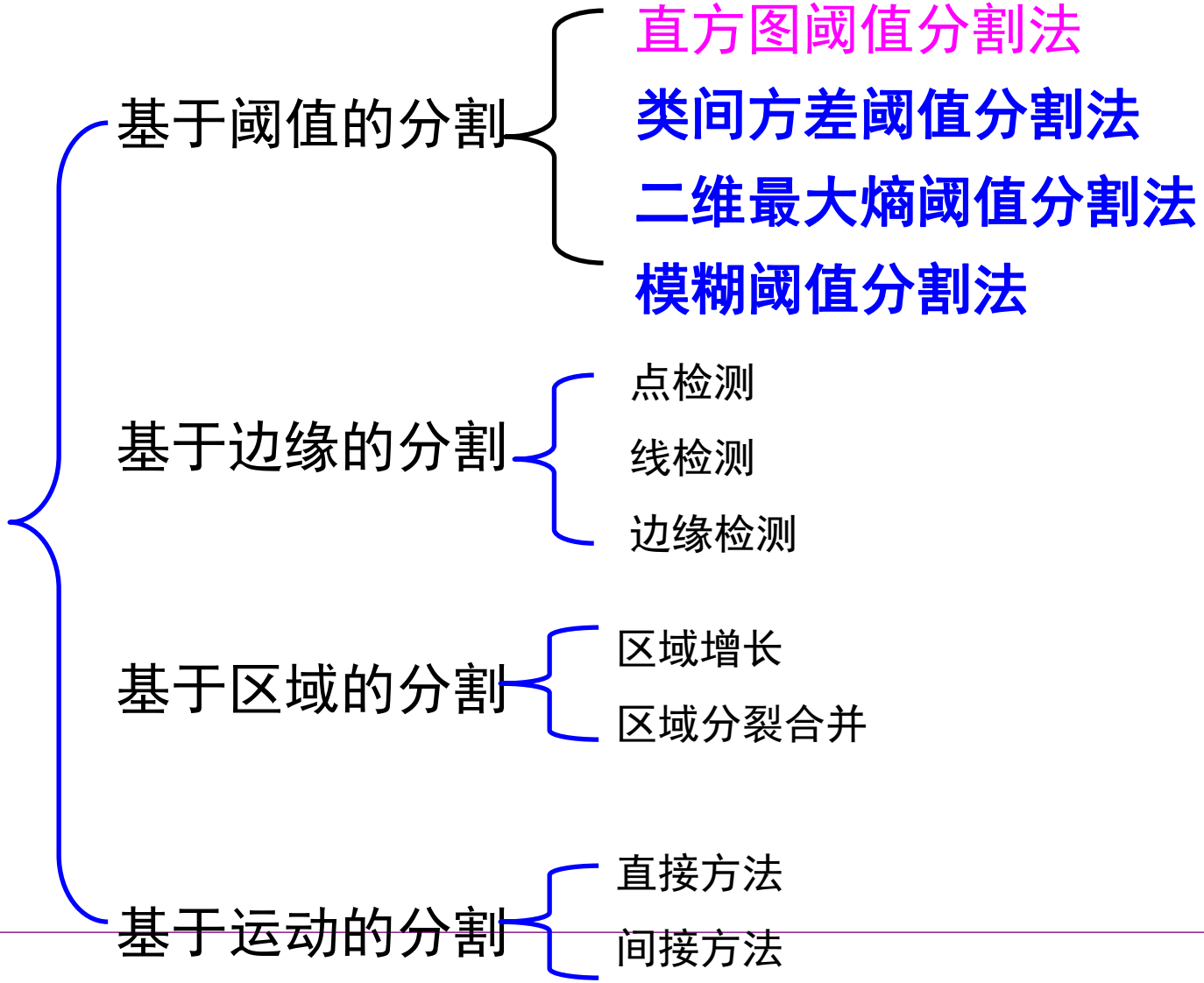


阈值T=90



阈值T=40

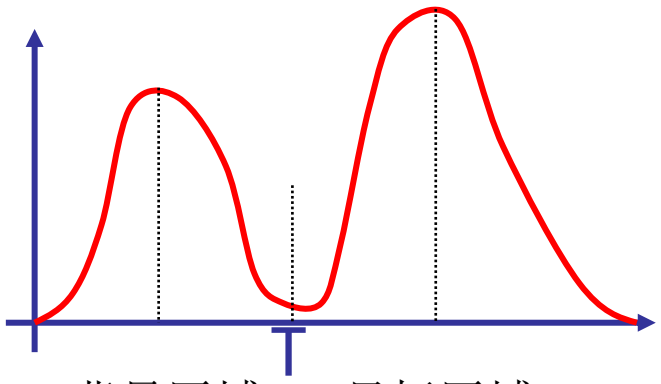
阈值选取方法

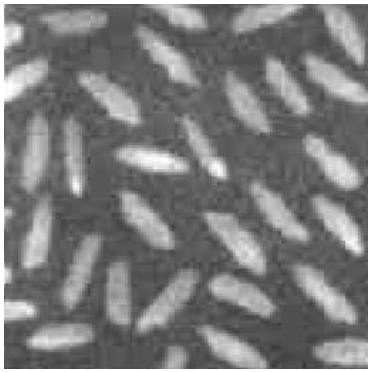

基于阈值的分割
直方图阈值分割法
类间方差阈值分割法
二维最大熵阈值分割法
模糊阈值分割法
基于边缘的分割
点检测
线检测
边缘检测
基于区域的分割
区域增长
区域分裂合并
基于运动的分割
直接方法
间接方法

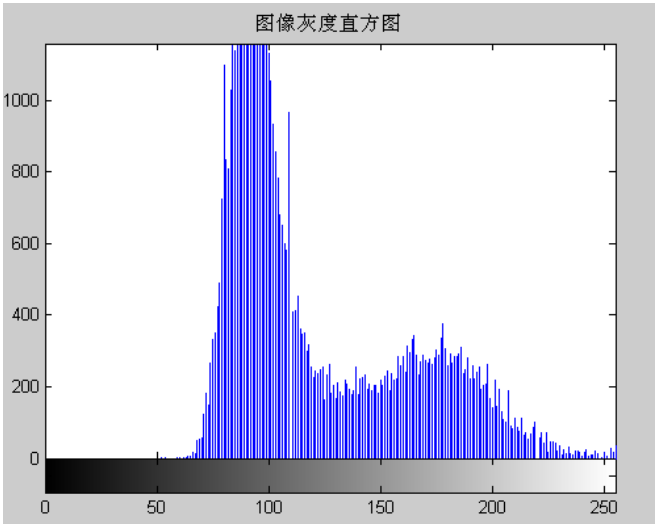
7.2.1 直方图阈值分割

1. 简单直方图分割法：

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时，**背景和对象物在图像的灰度直方图上各自形成一个波峰**，因而选择双峰间的低谷处所对应的灰度值为阈值，可将两个区域分离。







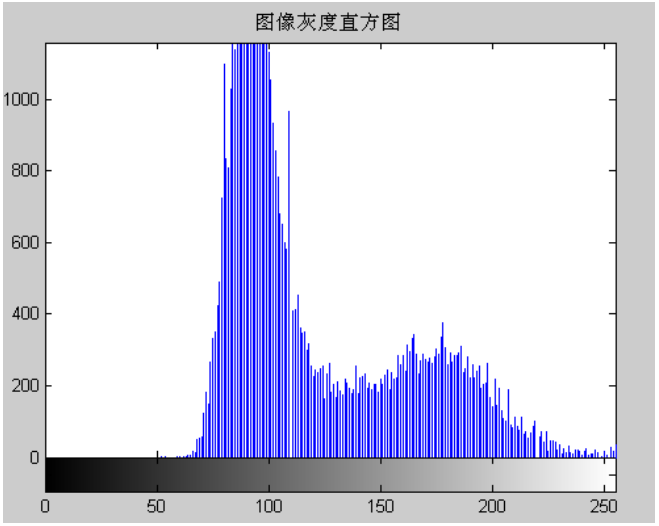
%直方图阈值分割图像

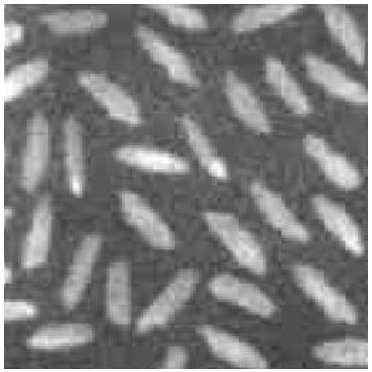
imgnew=im2bw(img1,130/255);

% 图像二值化，根据130/255确定的阈值，划分目标与背景，img1为灰度图像

figure;imshow(imgnew);

title('分割后图像');





2. 最佳阈值：

- 所谓最佳阈值是指使图像中目标物与背景的分割错误最小的阈值。该方法来源于模式识别中的贝叶斯最小误差分类方法。

7.2.2 类间方差阈值分割

- 基于阈值的分割
  - **直方图阈值分割法**
  - 类间方差阈值分割法
  - **二维最大熵阈值分割法**
  - **模糊阈值分割法**
- 基于边缘的分割
- 基于区域的分割
- 基于运动的分割

类间方差阈值法

类间方差阈值法又称最大类间方差法，是由Otsu于1978年提出的一种动态阈值法，此算法利用图像的灰度直方图，以目标均值和背景均值之间的方差最大为依据，动态的确定分割阈值，是在分析最小二乘法的基础上推导所得。该算法计算简单、较为稳定，一直广为使用。

具体算法：

设原始灰度图像灰度级为L，灰度级为i的像素点数为n_i，则图像的全部像素数为

$$N=n_0+n_1+\cdots+n_{L-1}$$

归一化直方图，则

$$p_i=n_i/N \qquad \sum_{i=0}^{L-1} p_i = 1$$

按灰度级用阈值t划分为两类：C_0=(0,1,..t) 和 C_1=(t+1,t+2,…L-1)， 因此，C_0类和C_1类的出现概率及均值分别如下：

$$w_0 = P_r(C_0) = \sum_{i=0}^{t} p_i = w(t)$$

$$w_1 = P_r(C_1) = \sum_{i=t+1}^{L-1} p_i = 1 - w(t)$$

$$\mu_0 = \sum_{i=0}^{t} ip_i / w_0 = \mu(t) / w(t)$$

$$\mu_1 = \sum_{i=t+1}^{L-1} ip_i / w_1 = \frac{\mu_T(t) - \mu(t)}{1 - w(t)}$$

其中：

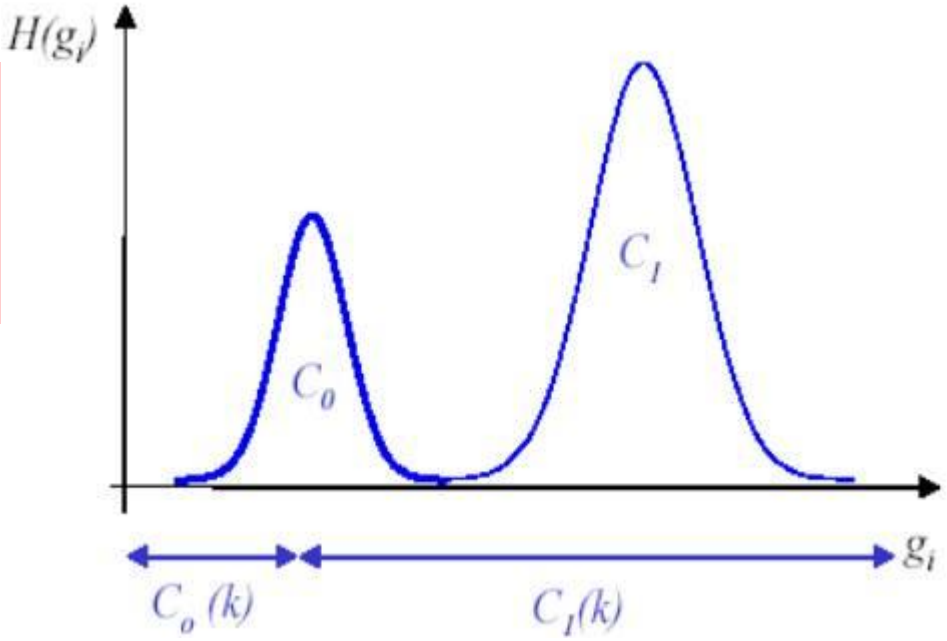
$$\mu(t) = \sum_{i=0}^{t} ip_i$$

$$\mu_T = \mu(L-1) = \sum_{i=0}^{L-1} ip_i$$

对任何t值，下式都能成立：

$$w_0\mu_0 + w_1\mu_1 = \mu_T$$

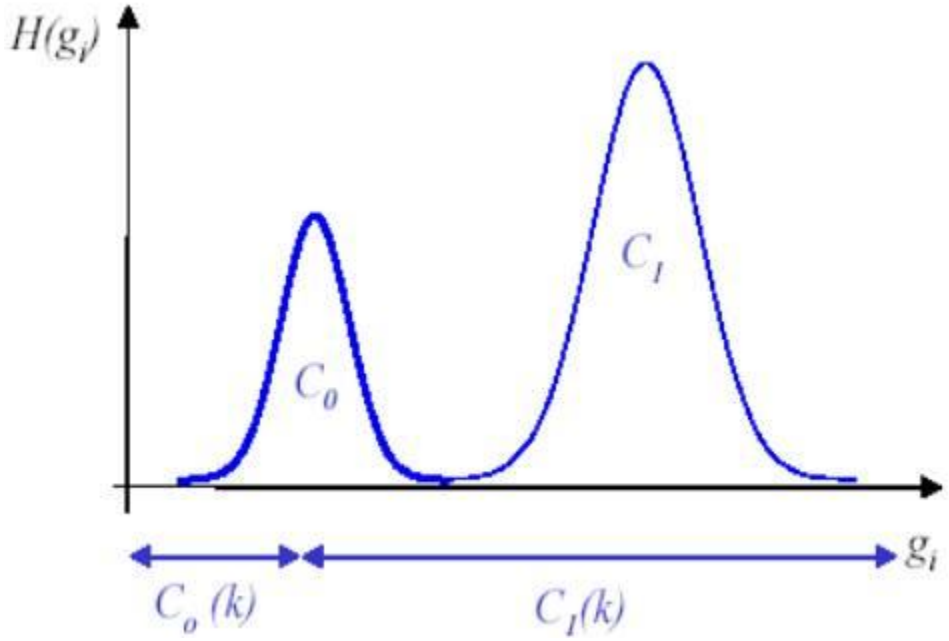
$$w_0 + w_1 = 1$$



C_0和C_1类的方差可由下式求得：

$$\sigma_0^2 = \sum_{i=0}^{t} (i-\mu_0)^2 \, p_i / w_0$$

$$\sigma_1^2 = \sum_{i=t+1}^{L-1} (i-\mu_1)^2 \, p_i / w_1$$



定义类内方差为：

$$\sigma_w^2 = w_0\sigma_0^2 + w_1\sigma_1^2$$

类间方差为：

$$\sigma_B^2 = w_0(\mu_0-\mu_T)^2 + w_1(\mu_1-\mu_T)^2 = w_0 w_1(\mu_1-\mu_0)^2$$

总体方差为：

$$\sigma_T^2 = \sigma_B^2 + \sigma_w^2$$

引入关于t的等价判决准则：

$$\lambda(t)=\frac{\sigma_B^2}{\sigma_w^2}$$

类间/类内

$$\eta(t)=\frac{\sigma_B^2}{\sigma_T^2}$$

$$\kappa(t)=\frac{\sigma_T^2}{\sigma_w^2}$$

三个准则是等效的，把使C0,C1两类得到最佳分离的t值作为最佳阈值，因此，将λ(t)、η(t)、κ(t)定义为最大判决准则。

由于σ_w^2是基于二阶统计特性，而σ_B^2是基于一阶统计特性，它们都是阈值t的函数，而σ_T^2与t值无关，因此三个准则中η(t)最为简单，因此选其作为准则，可得到最佳阈值t*

$$t^*=Arg\max_{0\le t\le L-1}\eta(t)$$

graythresh函数

函数功能：使用最大类间方差法找到图片的一个合适的阈值（threshold）。

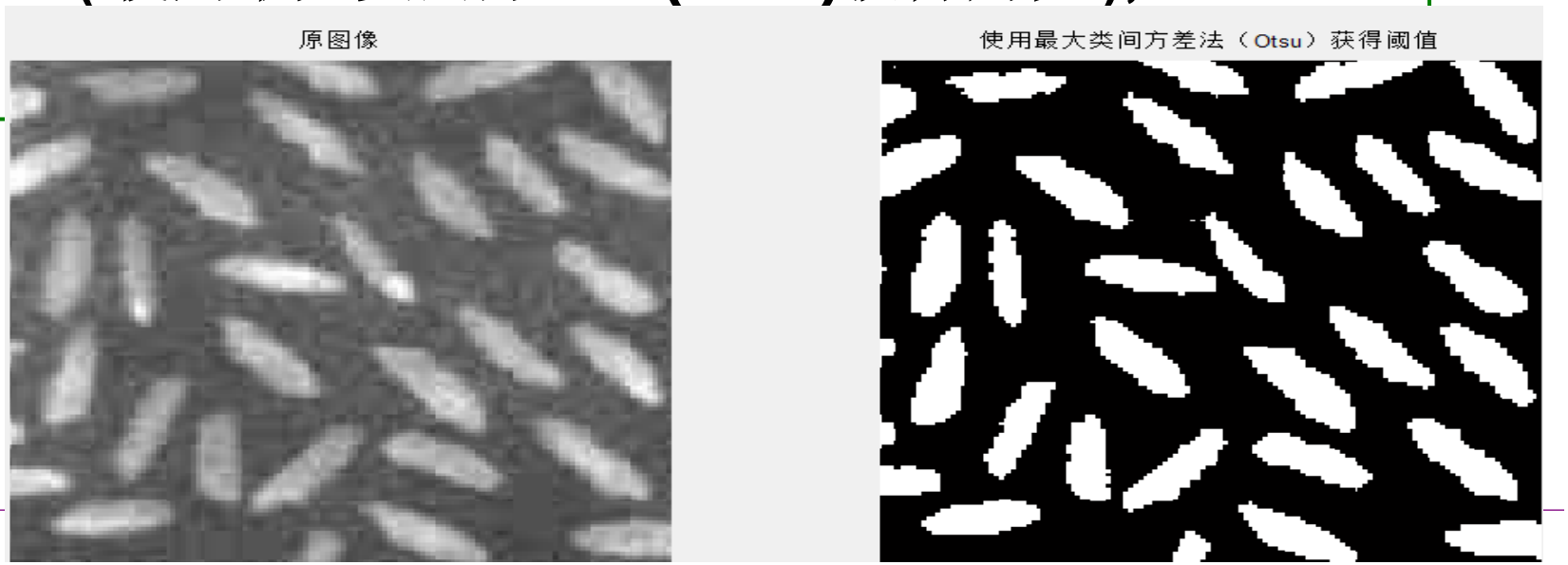
在使用im2bw函数将灰度图像转换为二值图像时，需要设定一个阈值，这个函数可以帮助我们获得一个合适的阈值。

调用格式：

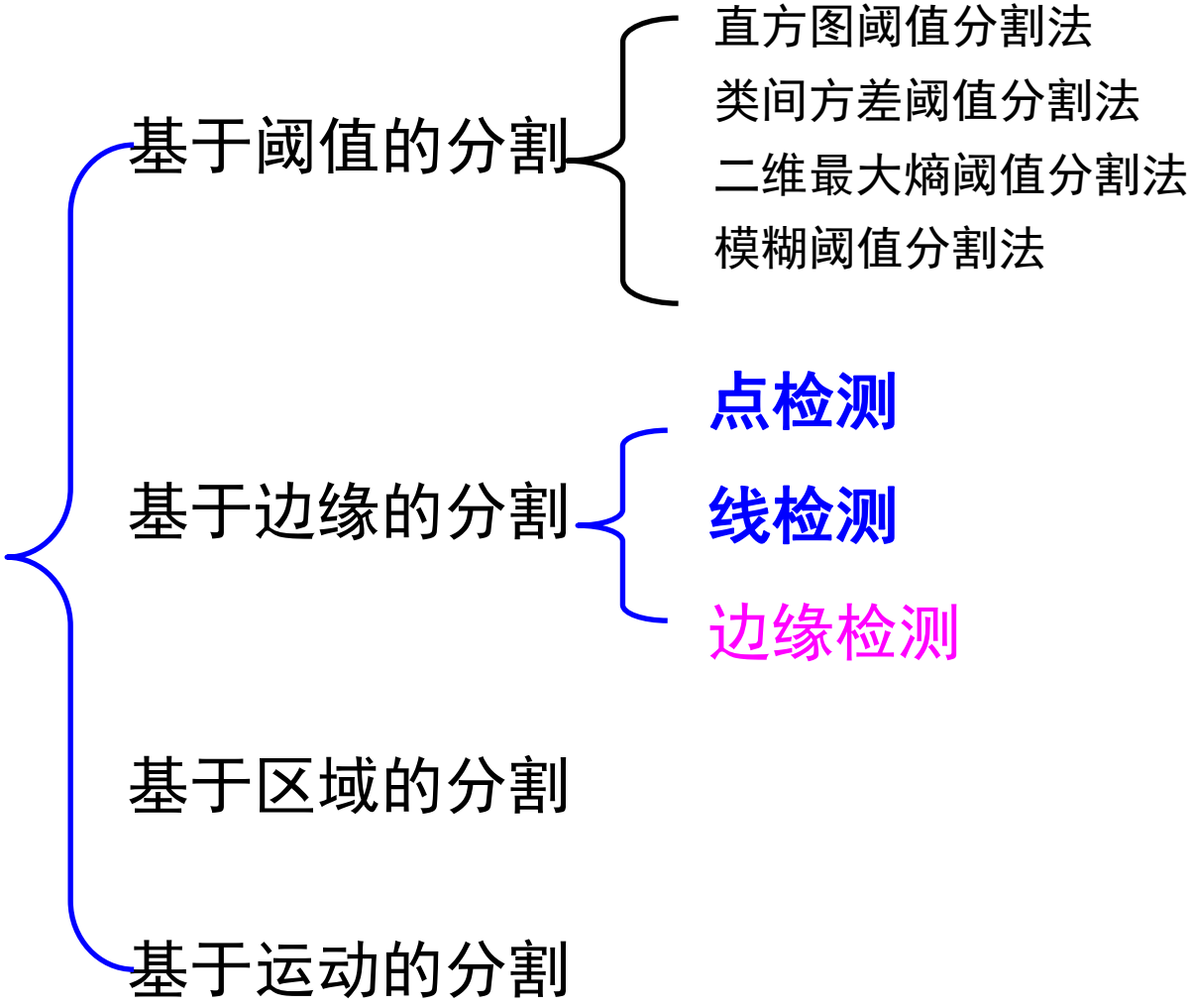
```
level = graythresh(I)
```

通过计算获得输入图像的阈值，这个阈值在[0, 1]范围内。该阈值可以传递给im2bw完成灰度图像转换为二值图像的操作。

```
img1 = imread('rice.bmp');
level = graythresh(img1);
imgbw = im2bw(img1,level);
imshow(imgbw);
title('使用最大类间方差法(Otsu)获得阈值');
```

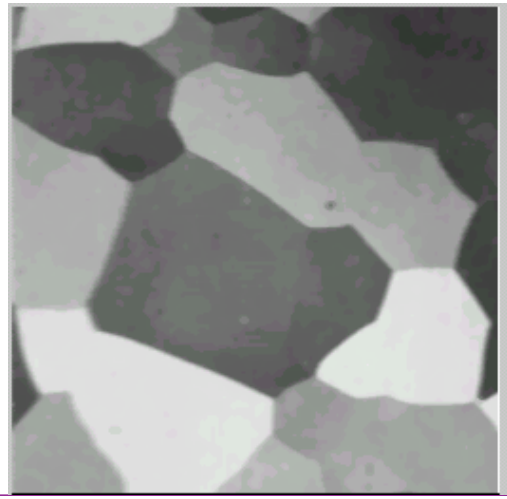


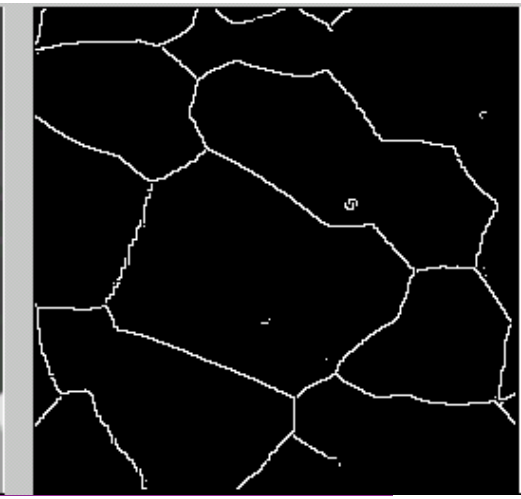
7. 3. 基于边缘的分割


基于阈值的分割
直方图阈值分割法
类间方差阈值分割法
二维最大熵阈值分割法
模糊阈值分割法
基于边缘的分割
点检测
线检测
边缘检测
基于区域的分割
基于运动的分割

图像的边缘是图像最基本的特征之一。所谓边缘是图像局部特性的不连续性。灰度或结构等信息的突变处称为边缘。例如：灰度级的突变、颜色的突变、纹理结构的突变等。

边缘是一个区域的结束，也是另一个区域的开始。它是图像分割依赖的重要依据。

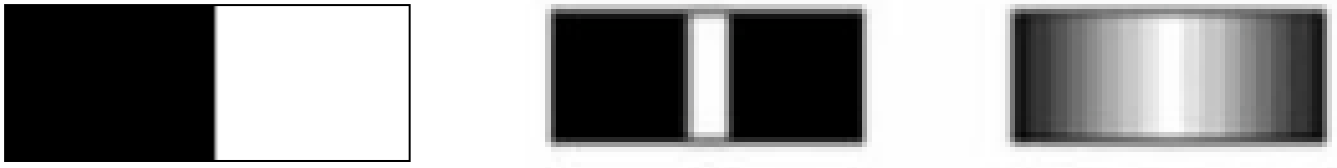




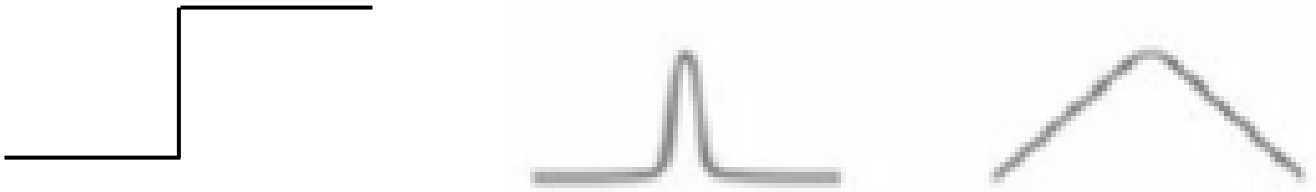
边缘的分类

- 阶跃状
- 脉冲状
- 屋顶状

图像：

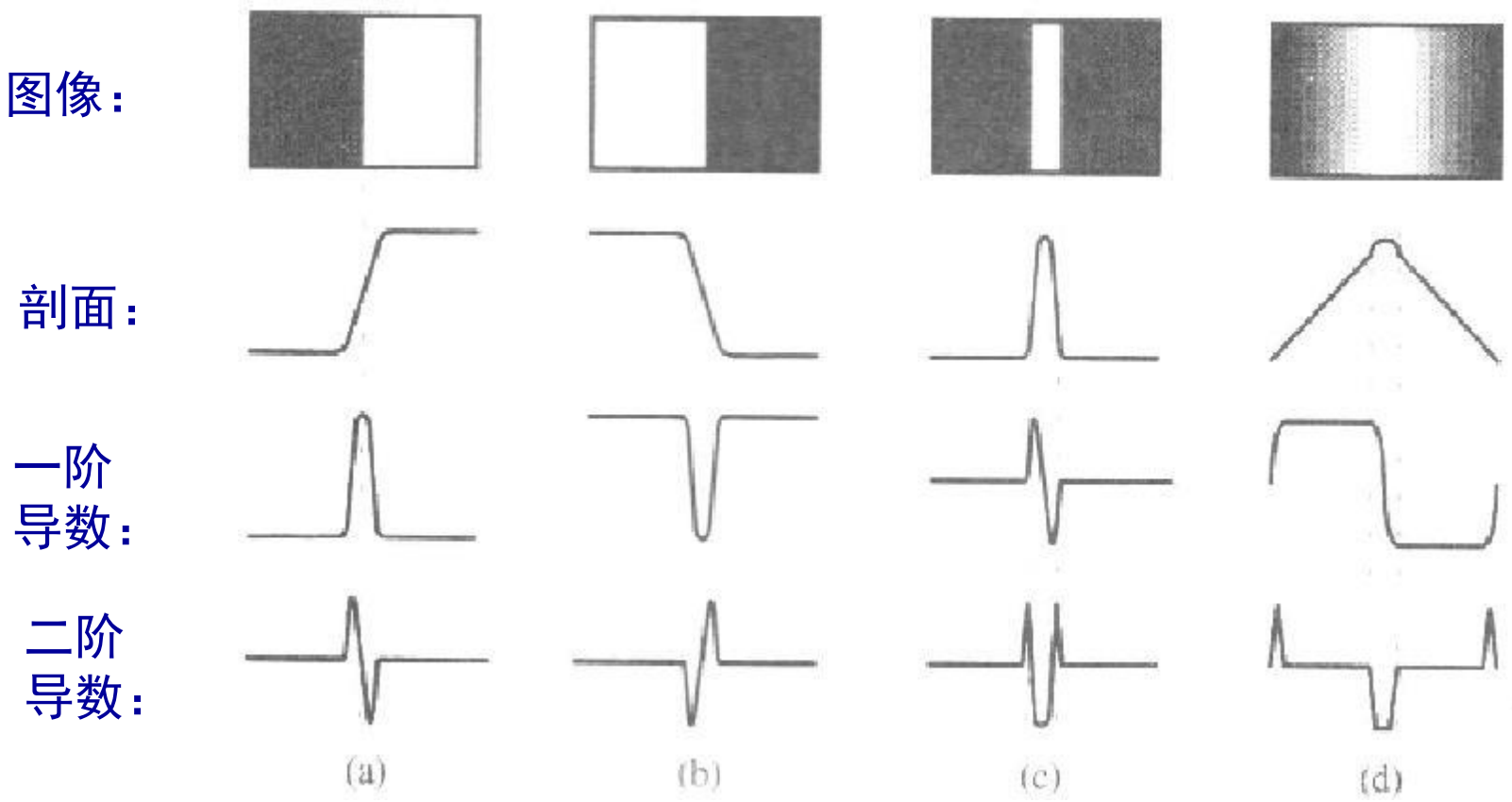


剖面：



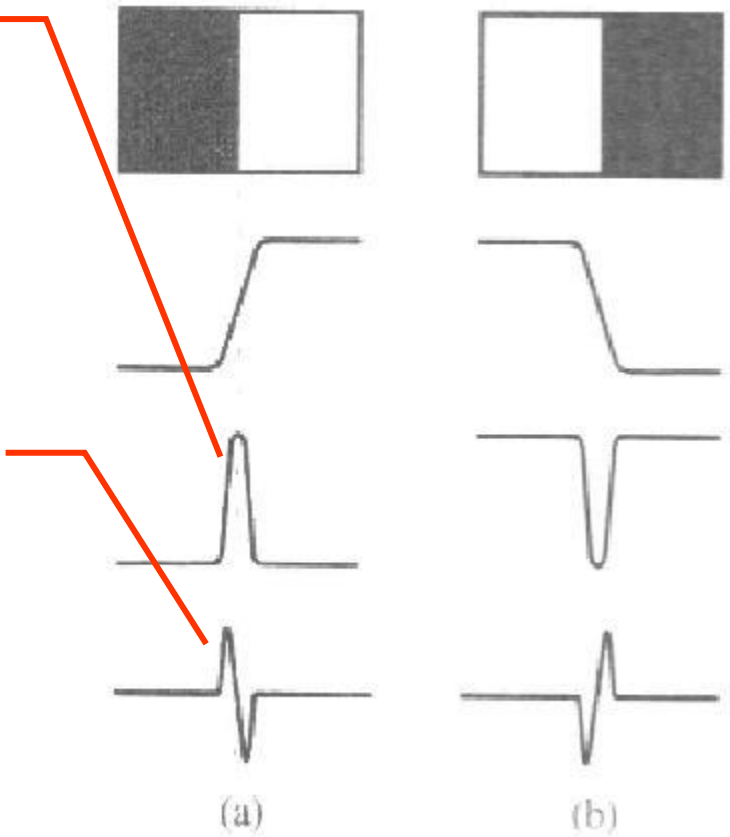
阶跃状　　脉冲状　　屋顶状

各种边缘其一阶、二阶导数特点



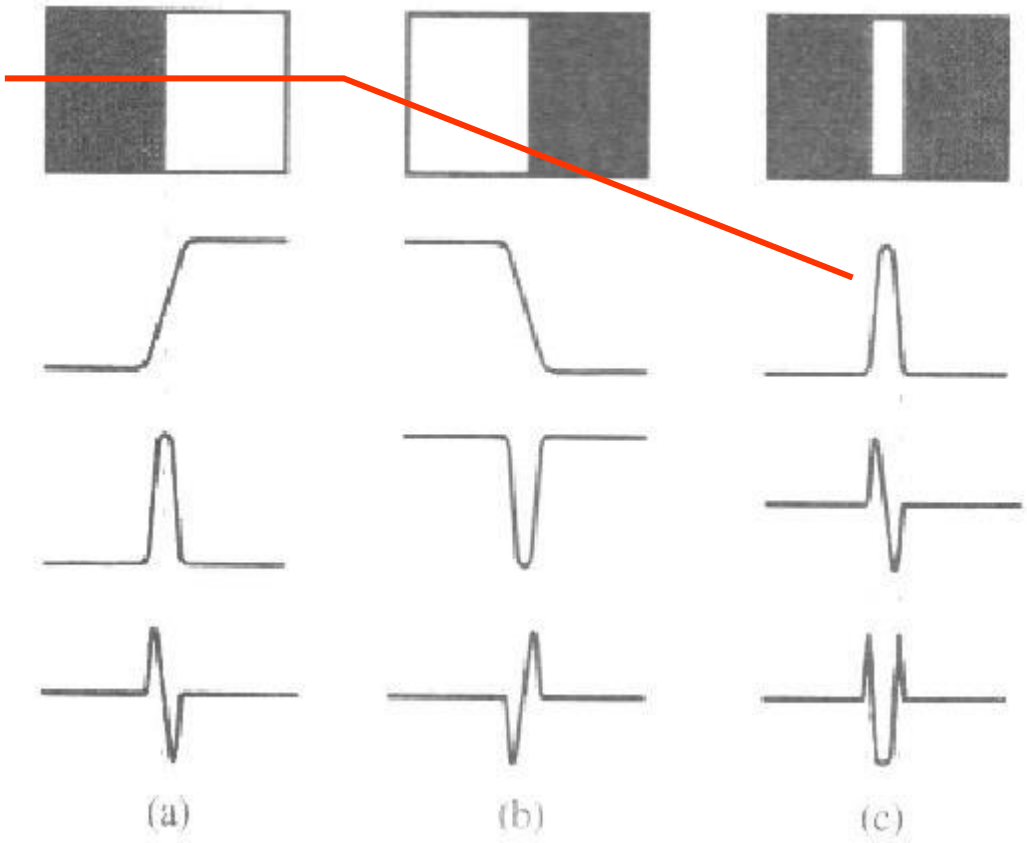
一阶导数在图像由暗变明的位置处有1个向上的阶跃，而其它位置都为0，这表明可用一阶导数的幅度值来检测边缘的存在，幅度峰值一般对应边缘位置。

二阶导数在一阶导数的阶跃上升区有1个向上的脉冲，而在一阶导数的阶跃下降区有1个向下的脉冲，在这两个脉冲之间有1个过0点，它的位置正对应原图像中边缘的位置，所以可用二阶导数的过0点检测边缘位置，而用二阶导数在过0点附近的符号确定边缘像素在图像边缘的暗区或明区。



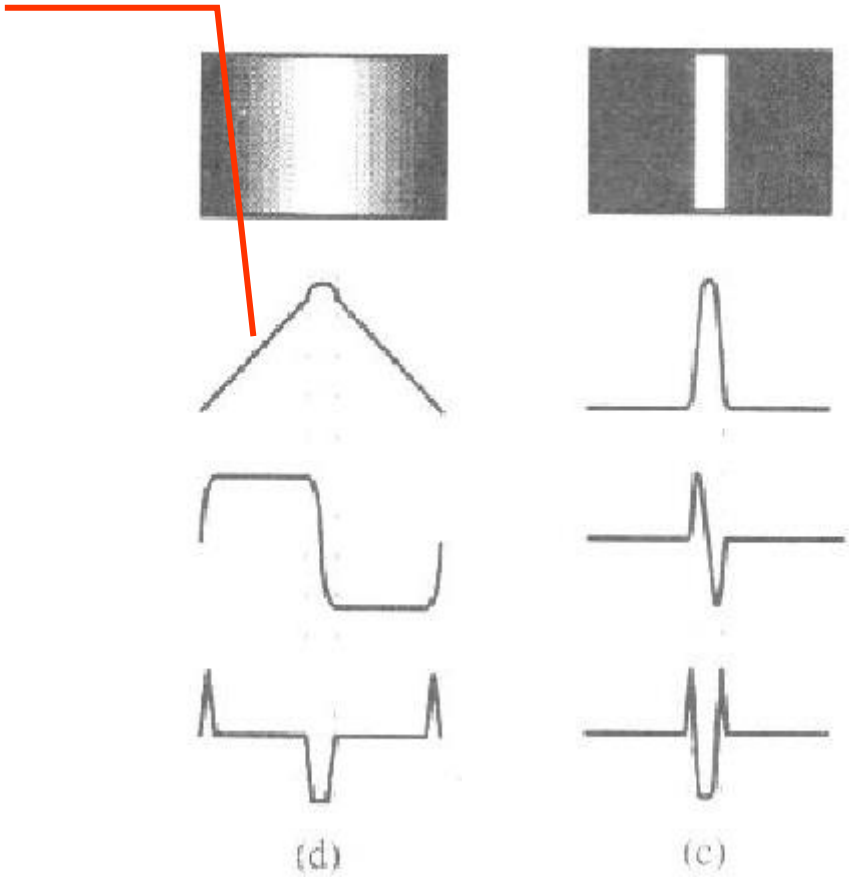
阶跃状

对(c)而言，脉冲状的剖面边缘与(a)的一阶导数形状相同，所以(c)的一阶导数形状与(a)的二阶导数形状相同，而它的2个二阶导数过0点正好分别对应脉冲的上升沿和下降沿，**通过检测脉冲剖面的2个二阶导数过0点就可确定脉冲的范围。**



阶跃状　　阶跃状　　脉冲状

对(d)而言，屋顶状边缘的剖面可看作是将脉冲边缘底部展开得到，所以它的一阶导数是将(c)脉冲剖面的一阶导数的上升沿和下降沿展开得到的，而它的二阶导数是将脉冲剖面二阶导数的上升沿和下降沿拉开得到的，**通过检测屋顶状边缘剖面的一阶导数过0点，可以确定屋顶位置。**



屋顶状　　脉冲状

边缘检测的实质： 是采用某种算法来提取图像中图像与背景的交界线。图像灰度的变化情况可以用图像灰度分布的梯度来反映，因此可以用局部图像微分技术来获得边缘检测算子。

边缘检测算子

可用一阶、二阶局部微分算子来检测图像中的边缘。下面是几种常用的微分算子。

- Marr算子
- Canny算子
- Roberts算子
- Prewitt算子
- Sobel算子
- Laplacian算子

梯度算子

二维图像 x 和 y 方向的导数变化用梯度表示。

$$\nabla f(x,y)=\begin{bmatrix}\dfrac{\partial f(x,y)}{\partial x}\\ \dfrac{\partial f(x,y)}{\partial y}\end{bmatrix}$$

梯度方向是变化最快的方向，方向角度：

$$\theta_g=\tan^{-1}\left(f_y/f_x\right)$$

该方向的变化率，

$$g(x,y)=\left|\nabla f(x,y)\right|=\sqrt{\left(\frac{\partial f(x,y)}{\partial x}\right)^2+\left(\frac{\partial f(x,y)}{\partial y}\right)^2}$$

g 为梯度算子，也经常被简化为：

$$g(x,y)=\left|f_x\right|+\left|f_y\right|$$

数字图像，用差分代替微分，则：

$$f_x(x,y)=f(x,y)-f(x-1,y)\quad f_y(x,y)=f(x,y)-f(x,y-1)$$

Roberts算子

Robert梯度采用的是对角方向相邻两像素之差，即

$$\Delta_x f(x,y) = f(x,y) - f(x-1,y-1)$$
$$\Delta_y f(x,y) = f(x-1,y) - f(x,y-1)$$

-1	0
0	1

0	-1
1	0

Roberts模板

- 特点：对噪声敏感，但效果较梯度算子略好。

Prewitt算子

- 模板：

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

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

- 特点：在检测边缘的同时，能抑止噪声的影响.

Sobel算子

- 模板

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

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

- 特点：对4邻域采用带权方法计算差分；能进一步抑止噪声；但检测的边缘较宽。

Laplacian算子

- 定义：

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

Laplacian 算子是二阶微分算子，可以证明，它具有各向同性，即与坐标轴方向无关，坐标轴旋转后结果不变。

- 离散形式：

$$\nabla^2 f(x,y) = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

- 模板：可以用多种方式表示。定义数字形式的拉普拉斯的基本要求是：作用于中心像素的系数是一个负数，而且其周围像素的系数为正数，系数之和必为0。对于一个3x3的区域，经验上被推荐最多的形式是：

0	1	0
1	-4	1
0	1	0

- 拉普拉斯算子的分析：
  - 优点：
    - 各向同性、线性和位移不变的；
    - 对细线和孤立点检测效果较好。
  - 缺点：
    - 对噪音的敏感，对噪声有双倍加强作用；
    - 不能检测出边缘的方向；
    - 常产生双像素的边缘。

由于以上原因，拉普拉斯算子很少直接用于边缘检测，而主要用于已知边缘像素后，确定该像素是在图像的暗区或明区一边。

LOG（Laplacian-Gauss）算子(或Marr)

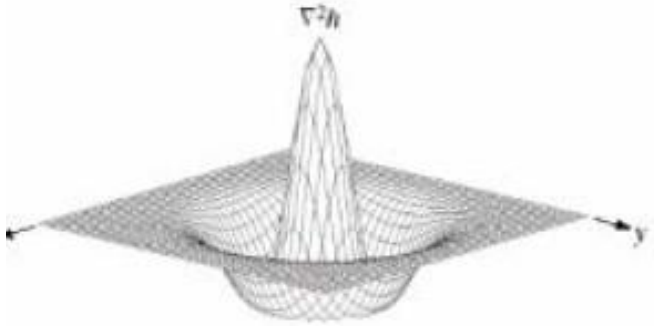
将**Gaussian**滤波器和**Laplacian**边缘检测结合在一起，形成了**LoG (Laplacian of Gaussian)**算法。即先用高斯函数对图像进行平滑，然后再用拉普拉斯算子进行运算，形成**Laplacian-Gauss**算法，它使用一个墨西哥草帽函数形式。

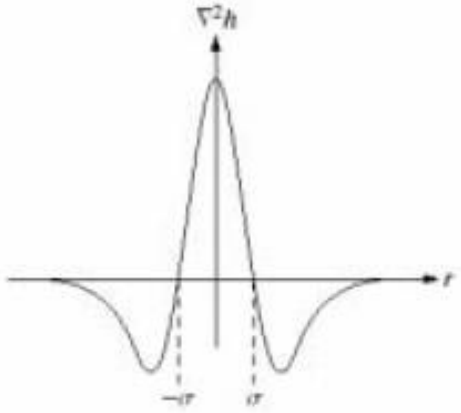
$$LOG(x,y)=\left(\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}\right)\frac{1}{2\pi\sigma^2}\exp\left(-\frac{(x^2+y^2)}{2\sigma^2}\right)$$

$$=\frac{-1}{2\pi\sigma^4}\left(2-\left(\frac{x^2+y^2}{\sigma^2}\right)\right)\exp\left(-\frac{(x^2+y^2)}{2\sigma^2}\right)$$

LoG 函数的三维曲线、横截面

三维曲线





横截面

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

模板

近似的5×5模板：一个正的中心项，周围是一个相邻的负值区域，并被一个零值的外部区域包围。系数的总和为零

◈ Canny（坎尼）算子

基本原理：

采用二维高斯函数的任一方向上的一阶方向导数为噪声滤波器，通过与图像**f(x,y)**卷积进行滤波；然后对滤波后的图像寻找图像梯度的局部极大值，以确定图像边缘。

边缘检测

原图像

$|G_x|$，x方向上的梯度分量，水平细节非常清楚

$|G_y|$，y方向上的梯度分量，垂直细节非常清楚

梯度图像$|G_x|+|G_y|$，水平和垂直细节都非常清楚

边缘检测的编程实现

edge函数可以提供各算子对边缘的检测：

BW=edge(I,’method’)

BW=edge(I,’method’,thresh)

I为输入图像，edge函数对灰度图像I进行边缘检测，返回与I同样的二值图像BW。

method是表示选用的方法（算子）类型，可以选择的method有Sobel、Prewitt、Roberts.、LOG、Canny等。

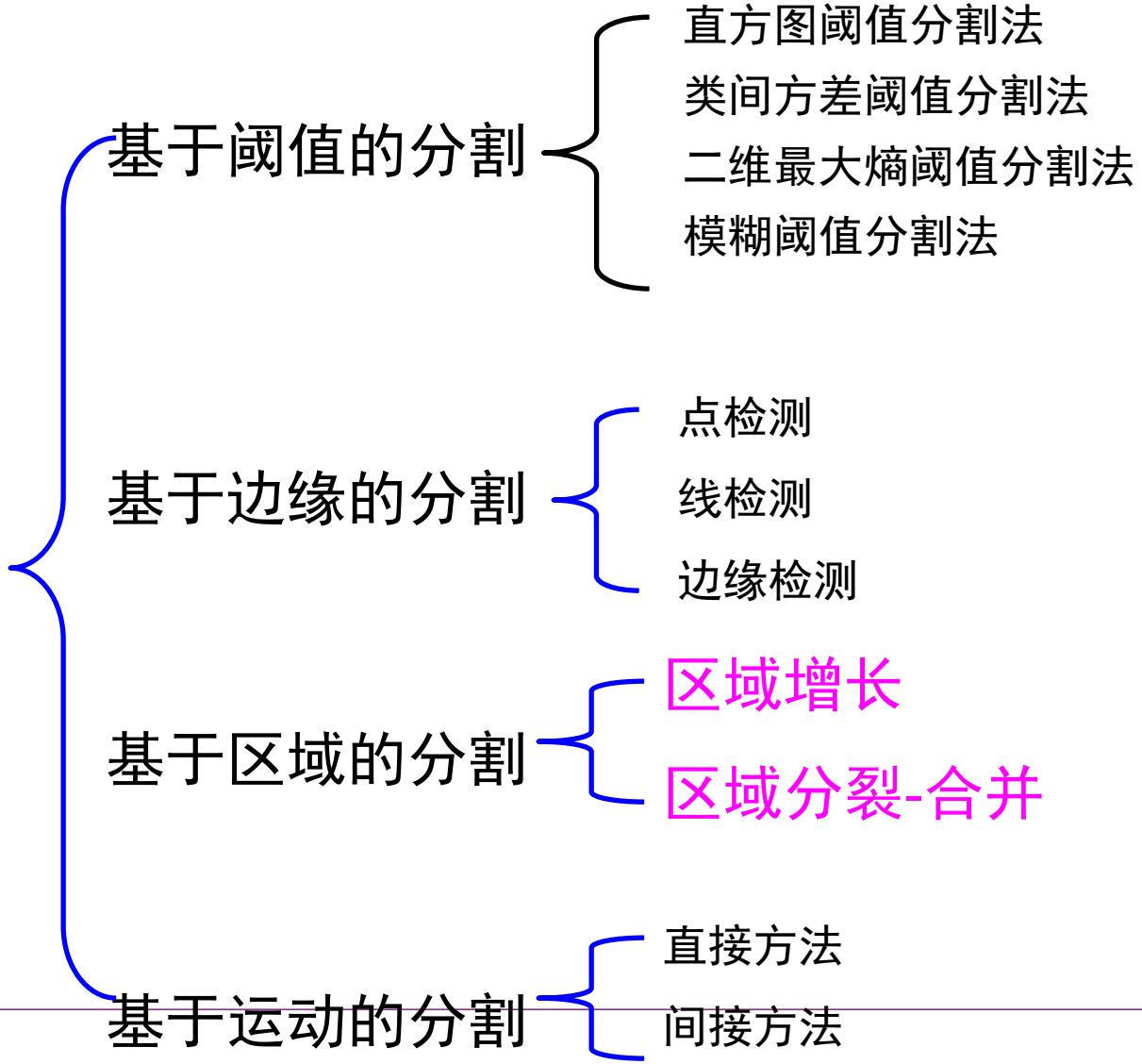
```
Y=edge(img1,‘sobel’,0.1);
        %用Sobel算子进行边缘检测，判别阈值为0.1
figure; imshow(Y);
title('用Sobel算子边缘检测');
```

采用各种边缘检测算子得到的边缘图像效果：

Log算子阈值为0.01, 其它为0.1。

比较几种算法分边缘检测结果，从图中可以看出Canny算子提取边缘较完整，其边缘连续性很好，效果优于其它算子。

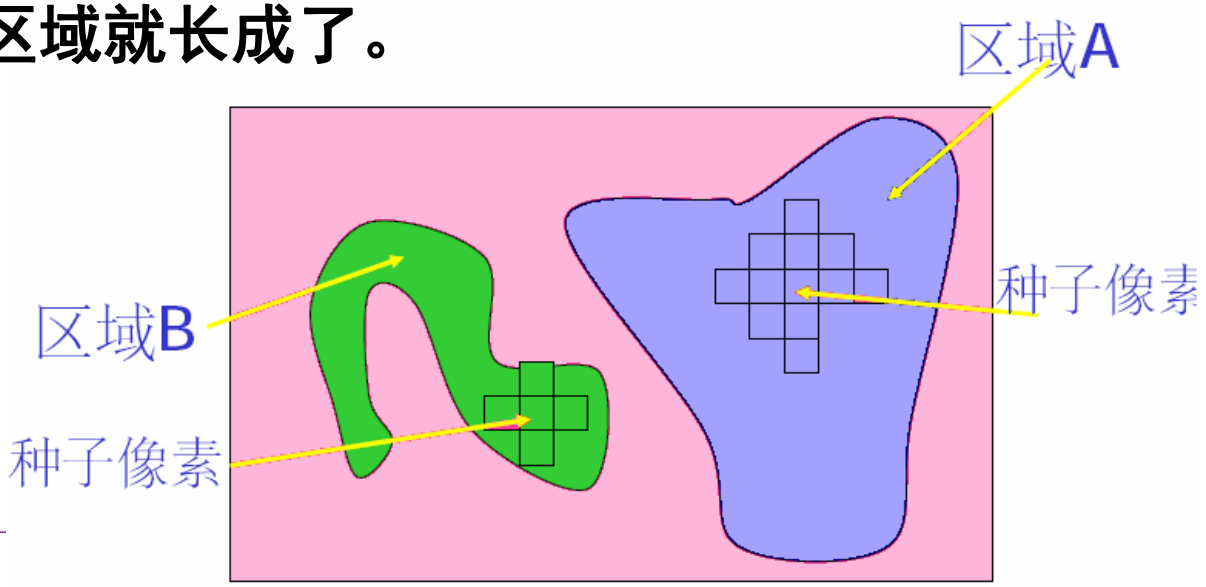
7.4 基于区域的分割


基于阈值的分割
直方图阈值分割法
类间方差阈值分割法
二维最大熵阈值分割法
模糊阈值分割法
基于边缘的分割
点检测
线检测
边缘检测
基于区域的分割
区域增长
区域分裂-合并
基于运动的分割
直接方法
间接方法

7.4.1 区域生长

基本思想—将具有相似性质的像素集合起来构成区域。

具体步骤—先对每个需要分割的区域找一个**种子像素**作为生长起点，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(根据某种事先确定的**生长或相似准则**来判定)**合并**到种子像素所在的区域中。将这些新像素当做新的种子像素继续进行上面的过程，直到再没有满足条件的像素可被包括进来，这样一个区域就长成了。



如图给出已知种子点 区域生长的一个示例：

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

给出需要分割的图像，设已知两个种子像素，现在进行区域生长。

生长规则：

如果像素与种子像素灰度值差小于阈值T，则将像素包括进种子像素所在的区域。

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

(a)

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

(b)

1	1	5	7	5
1	1	5	7	7
1	1	5	5	5
2	1	5	5	5
2	2	5	5	5

(c)

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

(d)

图(b)给出T=3时区域生长的结果，整幅图被较好的分成2个区域。

图(c)给出T=2时区域生长的结果，有些像素无法判定。

图(d)给出T=6时区域生长的结果，整幅图都被分成1个区域。

阈值T的选择非常重要

从上面的例子可以看出，在实际应用区域生长法时需要解决三个问题：

（1）选择或确定一组能正确代表所需区域的种子像素；

（2）确定在生长过程中能将相邻像素包括进来的准则；

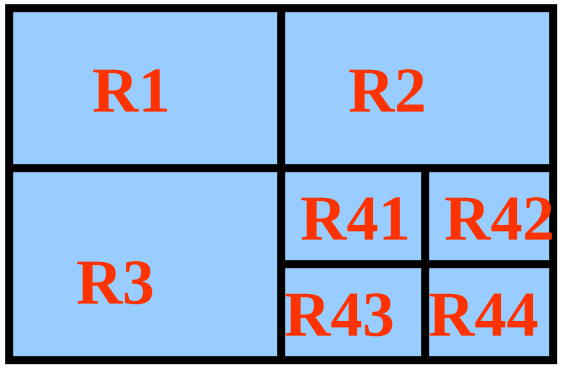
（3）制定让生长过程停止的条件或规则。

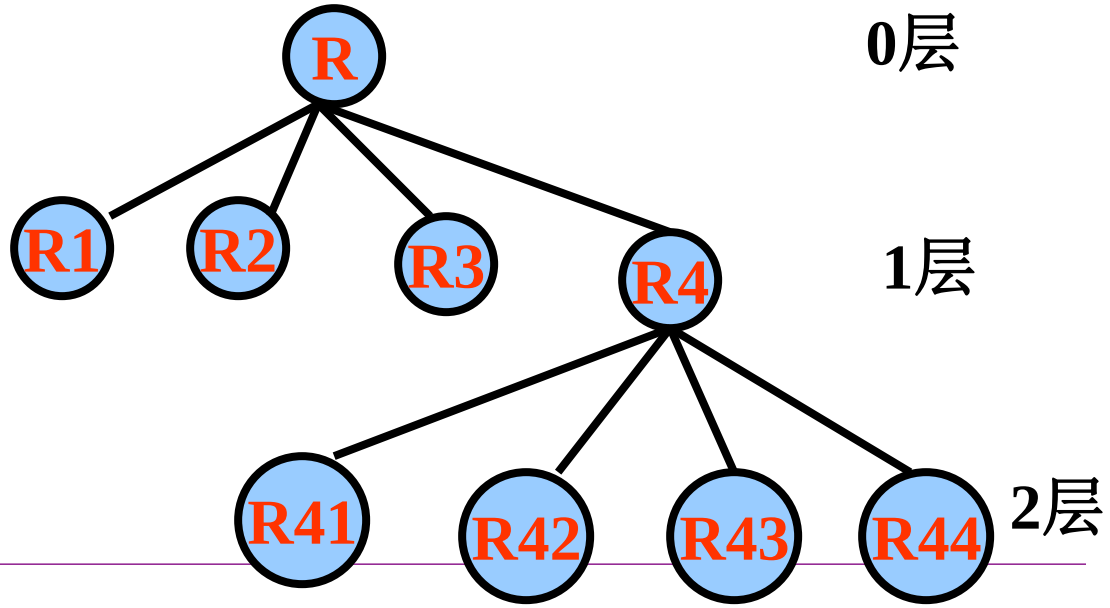
7.4.2 区域分裂-合并

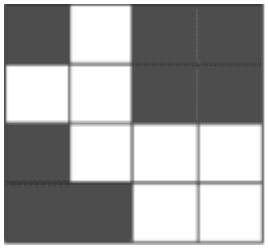
- 先将图像分割成一系列任意不相交的区域，然后将他们进行合并或分裂得到各个区域。
- 在这类方法中，最常用的方法是四叉树分解法.

利用图像四叉树表达方式的简单分裂合并算法

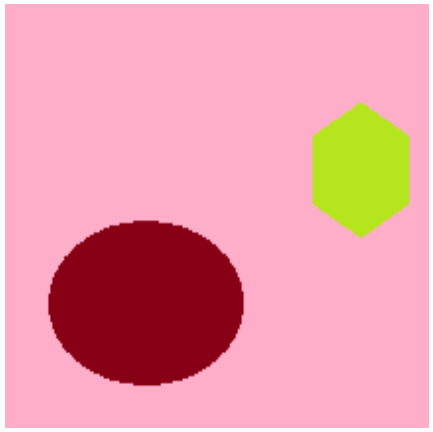
设R代表整个正方形图像区域，P代表逻辑谓词。从最高层开始，把R连续分裂成越来越小的1/4的正方形子区域Ri，并且始终使P(Ri)=TRUE。 即：如果P(R) =FALSE,那么就将图像分成四等分。如果P(Ri)=FALSE，那么就将Ri分成四等分，如此类推，直到Ri为单个像素。







图像1



图像2

- 如果仅使用分裂，最后得到的分割结果可能包含具有相同性质的相邻区域。为此，可在分裂的同时进行区域合并。
- 合并规则：

  只要两个相邻的区域R_i、R_j，具有一致性，则可以将Ri和R_j进行合并。

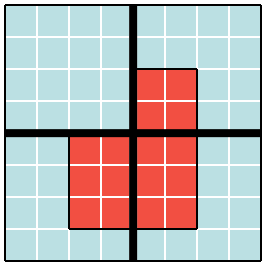
- **分裂合并算法步骤：**

（1）对任一区域Ri，如果P(Ri)=FALSE，就将其分裂成不重叠的四等分；

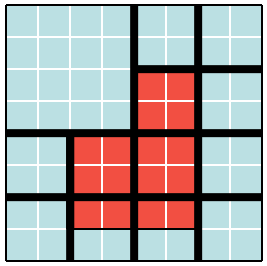
（2）对相邻的两个区域Ri和Rj（它们可以大小不同，即不在同一层），如果条件P(Ri∪Rj)=TRUE，就将它们合并；

（3）如果进一步的分裂或合并都不可能，则结束。

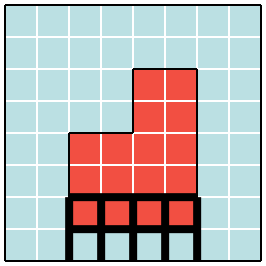
示例



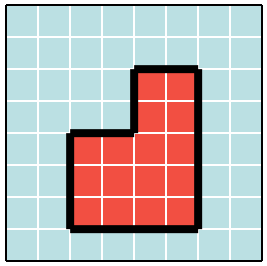
(a)



(b)



(c)



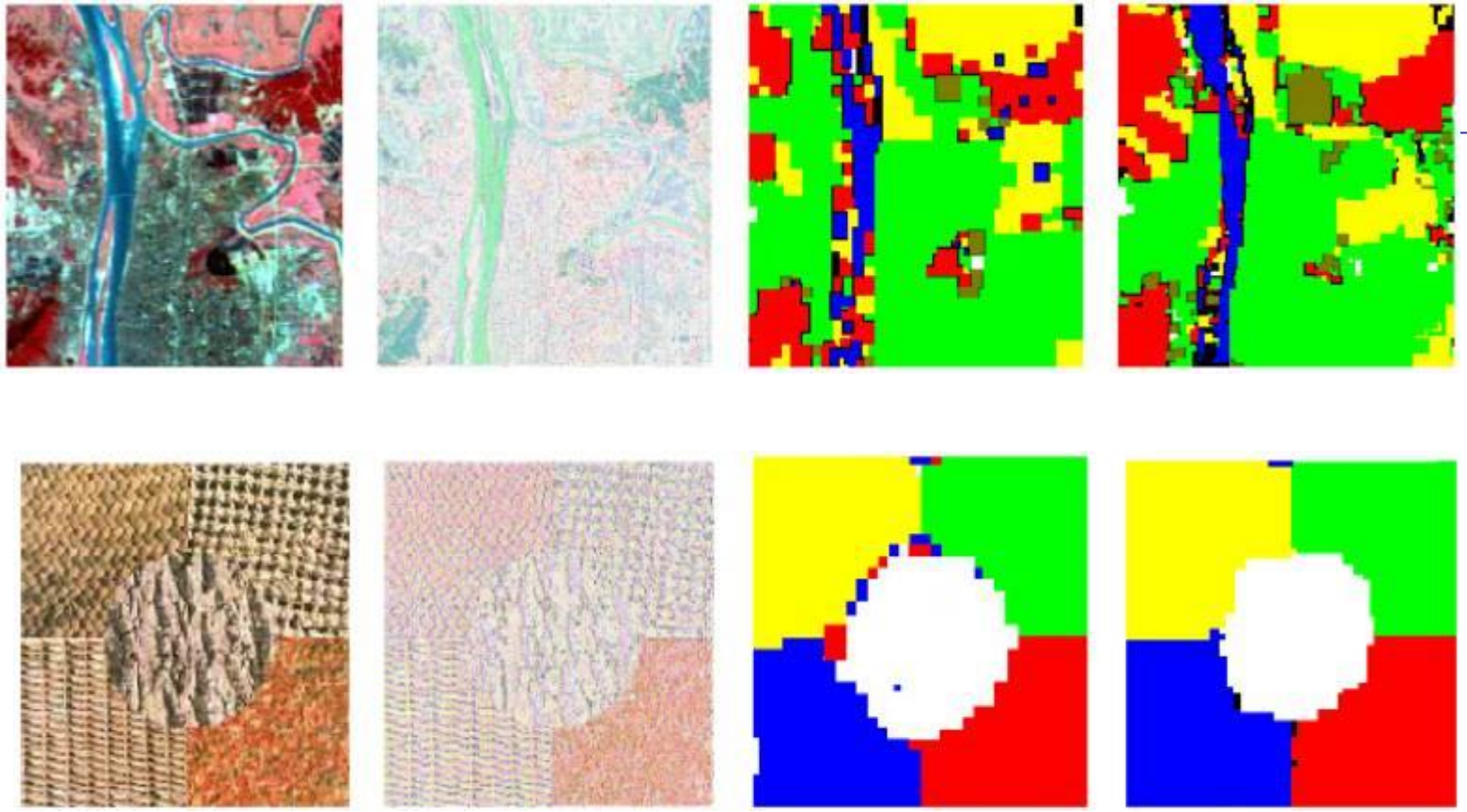
(d)

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中的所有像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。



(a)原始图像　(b)预处理后结果　(c)分裂结果　(d)分裂合并结果

Image Segmentation

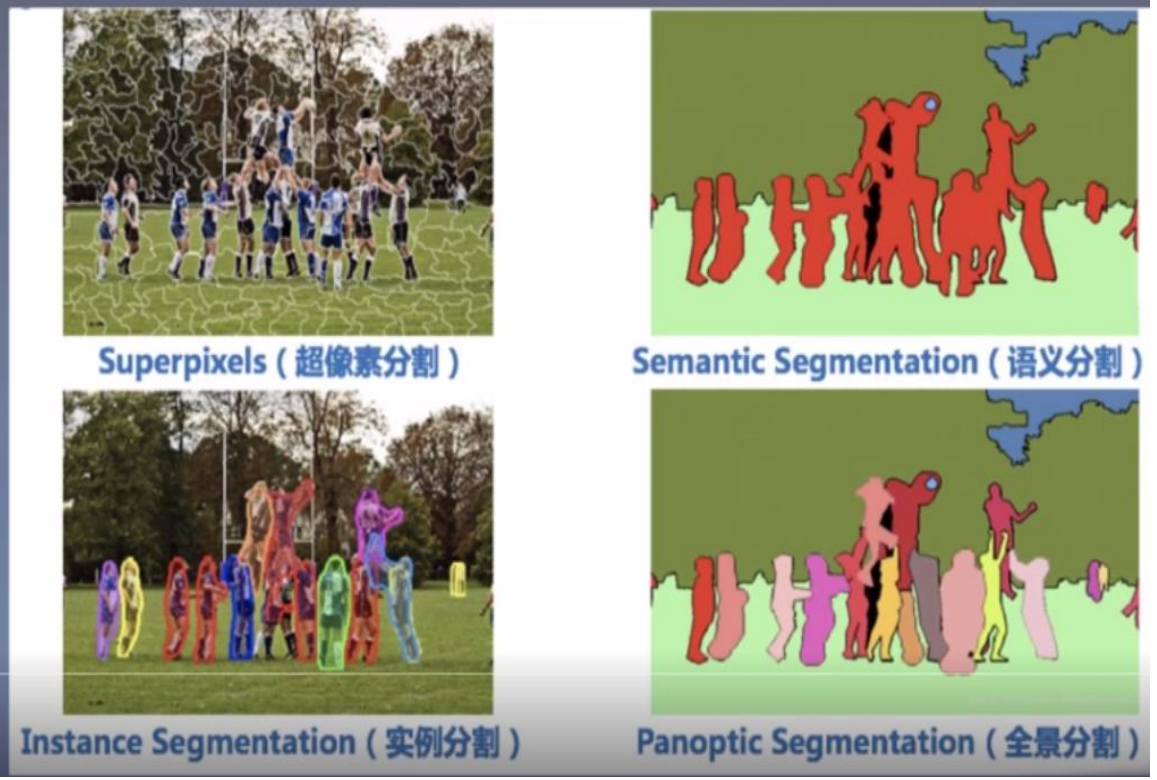
图像分割分类：

1.**超像素分割**：少量超像素代替大量像素，常用于图像预处理

2. **语义分割**：逐像素分类，无法区分个体

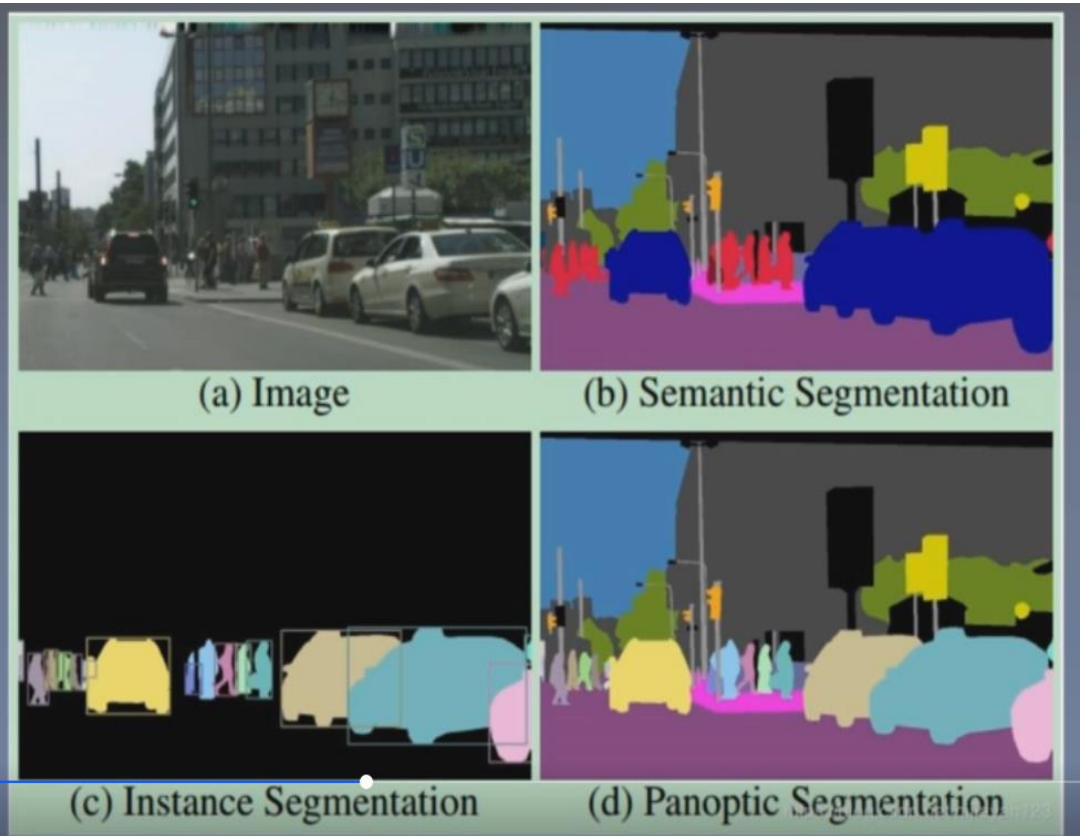
3. **实例分割**：对个体目标进行分割，像素级目标检测

4. **全景分割**：语义分割结合实例分割



实例分割是目标检测和语义分割的结合，
实例分割只对图像中的object进行检测，并对检测到的object进行分割，而全景分割是对图中的所有物体包括背景都要进行检测和分割。

- 语义分割
- 实例分割
- 全景分割

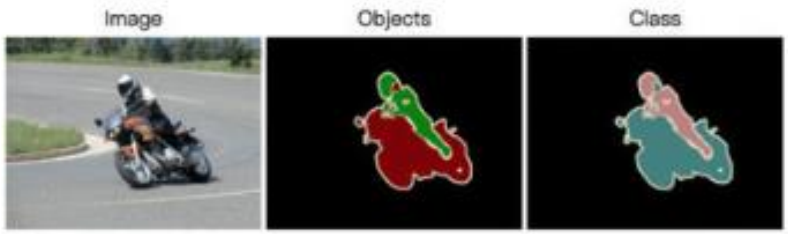


图像分割常用的数据集：

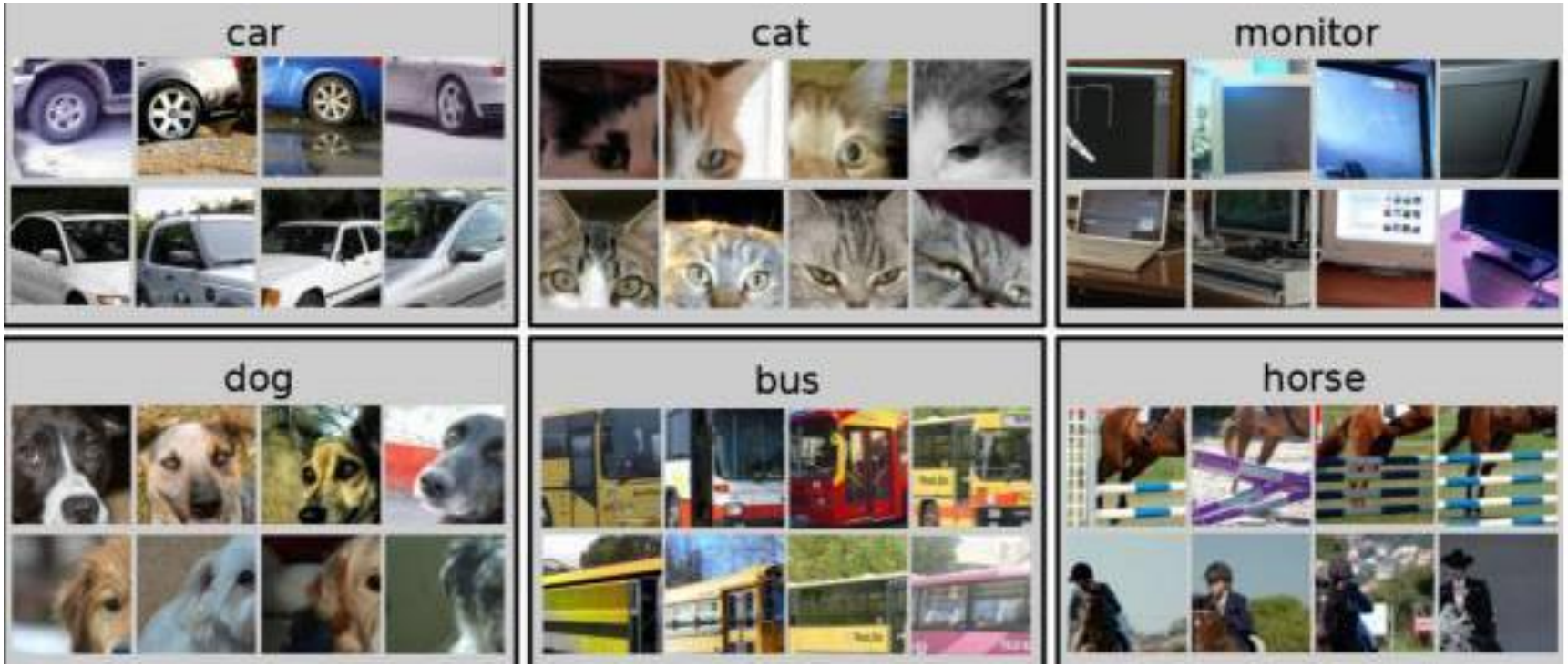
Pascal VOC； MS COCO ； CityScapes； CamVid

Pascal VOC

- 20个类别
- 6929张标注图片
- Both class-level and instance-level



这是一个比较老牌的数据集，它提供 20 个类别，包括，人，车等。有 6929 张标注图片，提供了类别层面的标注和个体层面的标注，也就是说既可以做语义分割，只区分是不是车；也可以做个体分割，区分有几辆车，把不同的车标记出来。


car
cat
monitor
dog
bus
horse

MS COCO

- 80类
- 33万张图片，超过20万张有标注
- 150万个物体的个体
- Instance-level

目前为止有语义分割的最大数据集，提供的类别有 80 类，有超过 33 万张图片，其中 20 万张有标注，整个数据集中个体的数目超过 150 万个，最新的一些论文都会在 MS COCO 数据集上做实验，因为它的难度最大，挑战新最高。

Dataset examples

ImageNet

ImageNet 是一个计算机视觉系统识别项目， 是目前世界上图像识别最大的数据库。

Imagenet数据集有1400多万幅图片，涵盖2万多个类别；其中有超过百万的图片有明确的类别标注和图像中物体位置的标注，具体信息如下：

1）Total number of non-empty synsets: 21841

2）Total number of images: 14,197,122

3）Number of images with bounding box annotations: 1,034,908

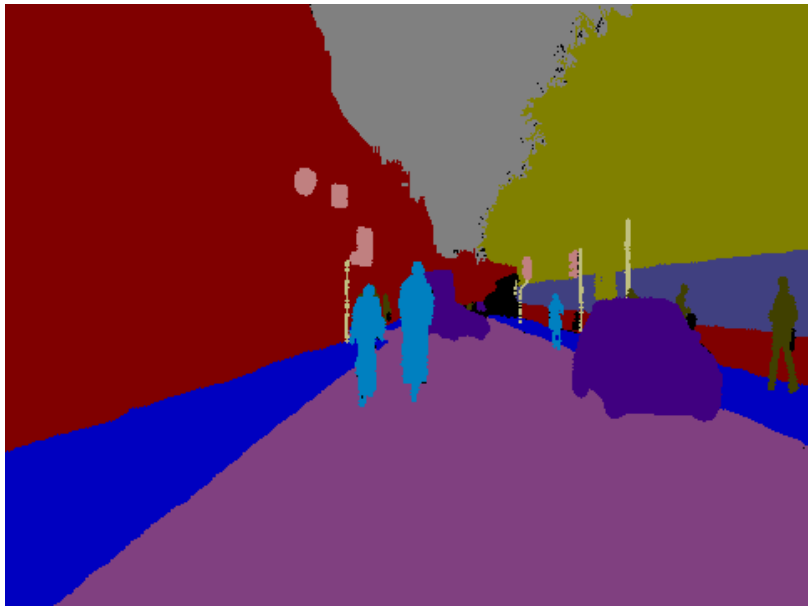
4）Number of synsets with SIFT features: 1000

5）Number of images with SIFT features: 1.2 million

Imagenet数据集是目前深度学习图像领域应用得非常多的一个领域，关于图像分类、定位、检测等研究工作大多基于此数据集展开。

CamVid　城市道路场景的数据集

原图



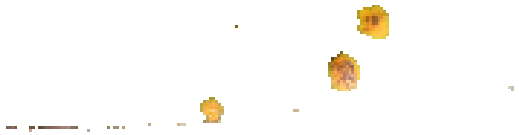
标签图

分割实例：

原始图像

裁剪后

分割结果(方法1)



分割结果(方法2)