

图像处理和分析技术

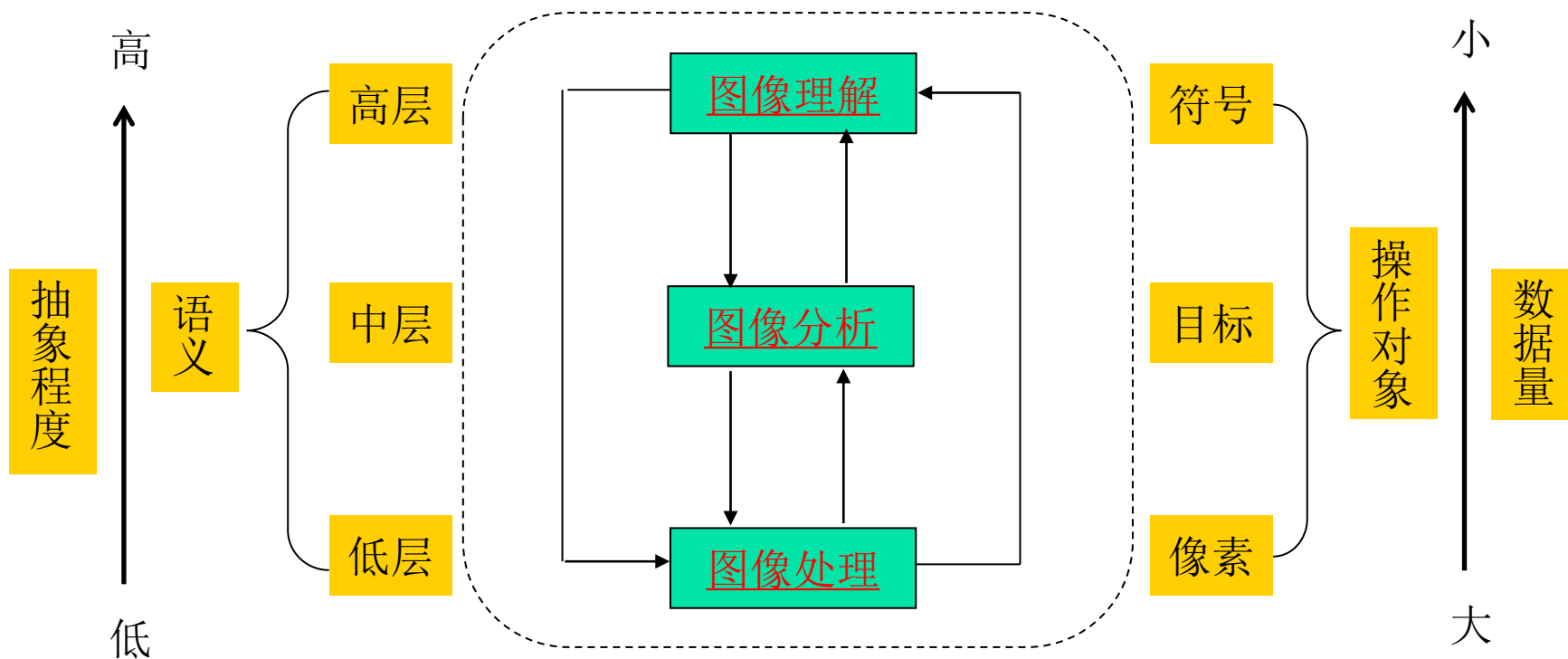


第五章 图像分割技术

主讲：李子印
中国计量大学

5.1 图像分割定义和方法分类

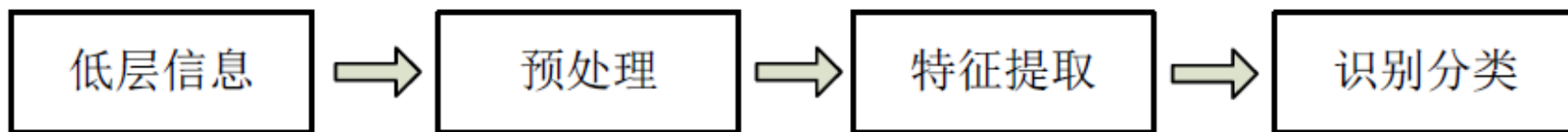
5.1.1 研究背景





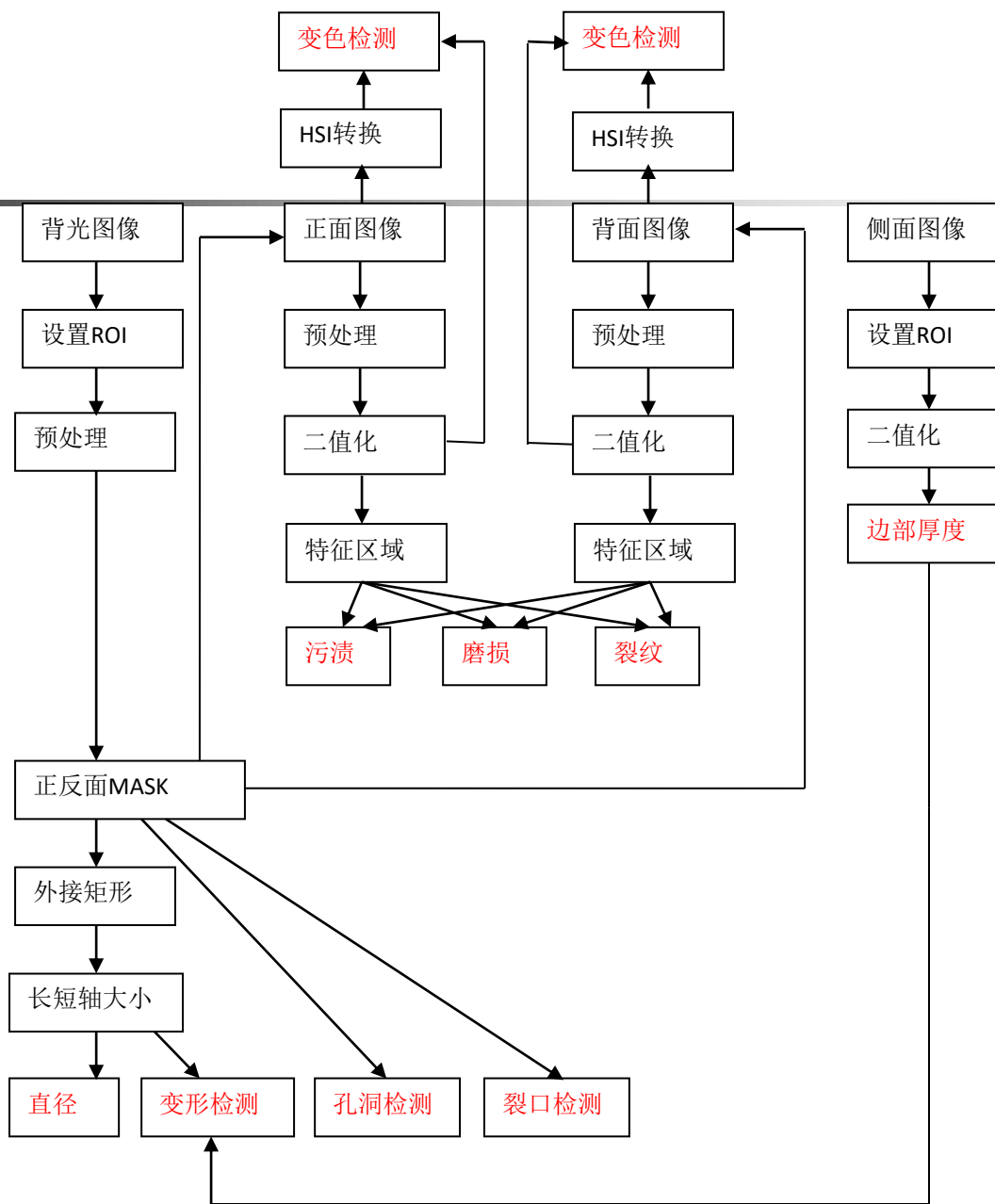
5.1 图像分割定义和方法分类

5.1.1 研究背景



传统图像识别的流程

5.1 图像分割定义和方法分类



5.1 图像分割定义和方法分类

FirstStepMulti

检测完毕

测试

建立模板

标定





类型	检测	标准	百分比
直径(毫米)	24.89	25.00	-0.43
重量(克)	6.98	6.05	-1.12
厚度(毫米)	1.78	1.85	-4.05
正面颜色	21.31	35.60	
反面颜色(单)	34.07		7.04
反面颜色(多)	88.15		14.07
边缘颜色	15.28	35.60	
边缘(单)	22.03		4.56
边缘(多)	44.15		9.12

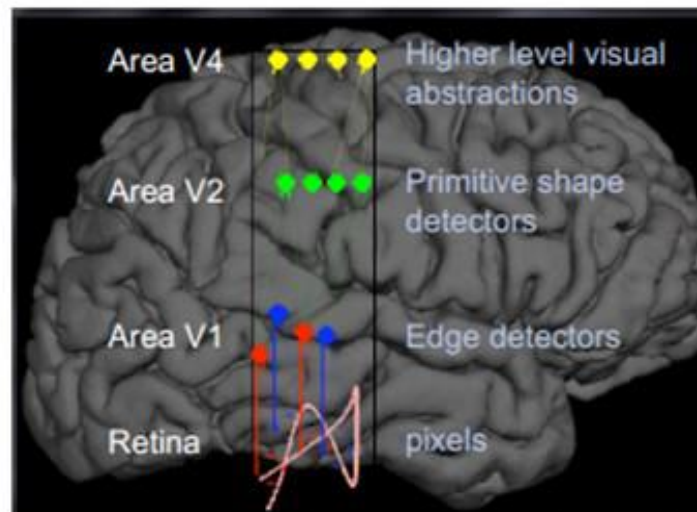
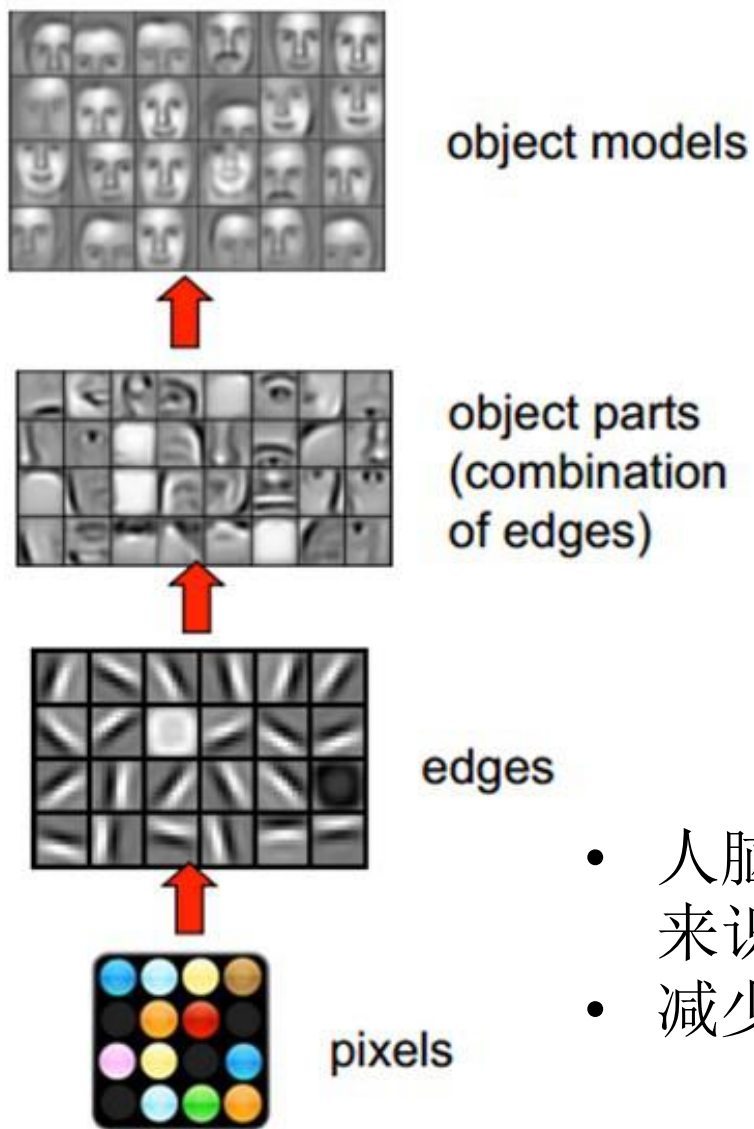
序号	币种	结果	缺陷类型	检测时间	检测地点	检测人员
31	五角	不合格	边缘缺陷	2018.11.19.09:01:18	地球	张三
32	一元	合格	无缺陷	2018.11.19.09:01:33	地球	张三
33	一元	不合格	边缘缺陷	2018.11.19.09:02:00	地球	张三
34	五角	合格	无缺陷	2018.11.19.09:02:24	地球	张三
35	五角	合格	无缺陷	2018.11.19.09:03:30	地球	张三
36	五角	不合格	边缘缺陷	2018.11.19.09:04:13	地球	张三
37	一元	合格	无缺陷	2018.11.19.09:04:43	地球	张三
38	一元	不合格	边缘缺陷	2018.11.19.09:05:02	地球	张三

不合格

5.1

图像分割定义和方法分类

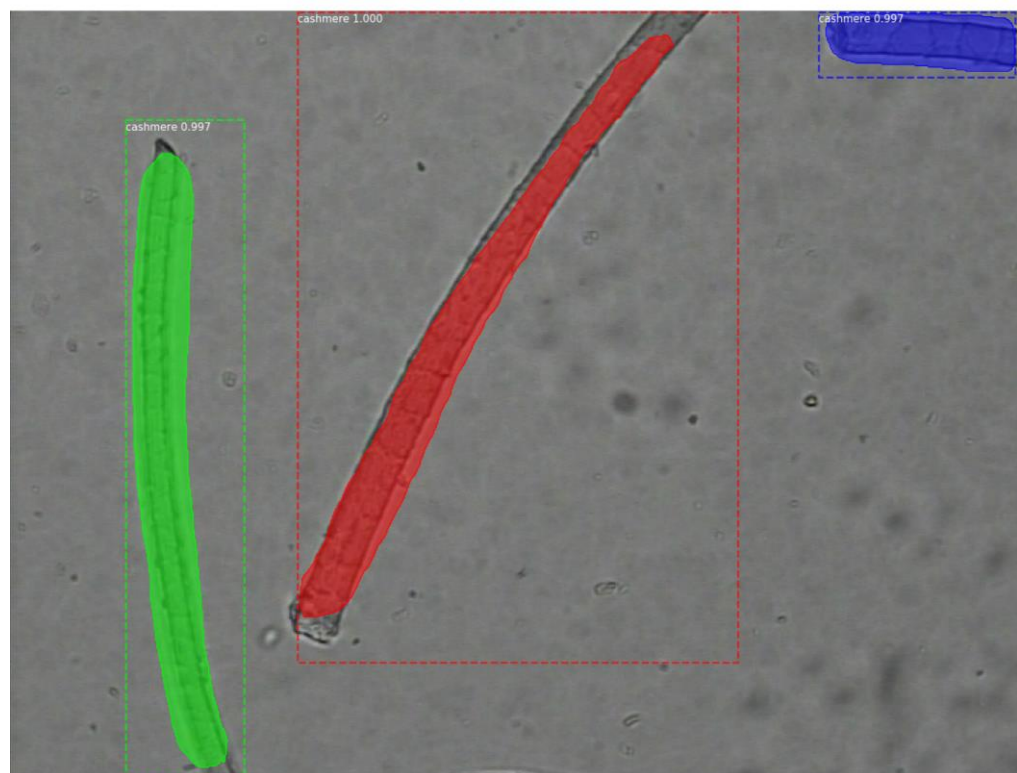
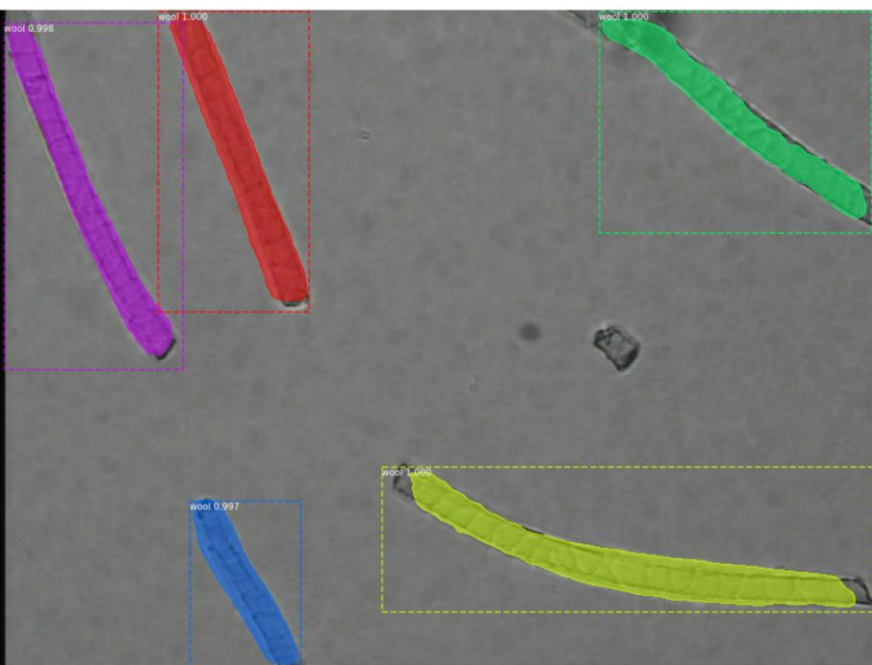
深度学习识别的流程



- 人脑是通过分级的、多层网络模型来识别
- 减少数据量，保留物体的有用信息

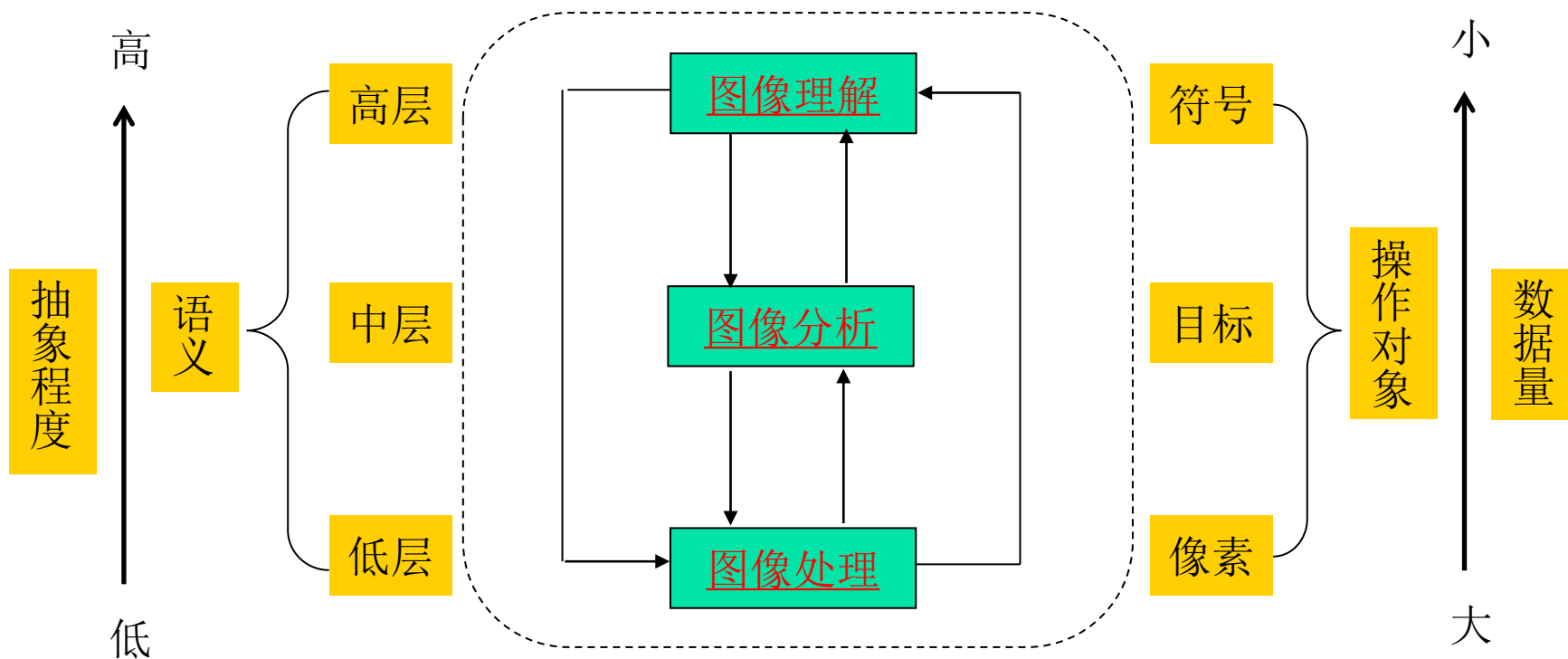
5.1 图像分割定义和方法分类

5.1.1 研究背景



5.1 图像分割定义和方法分类

5.1.1 研究背景



5.1 图像分割定义和方法分类

5.1.1 研究背景





5.1 图像分割定义和方法分类

5.1.1 研究背景

- 人们往往仅对图像中的某些部分感兴趣（ROI）；
- 为了辨识和分析目标，需要将ROI分离提取出来；
- 然后，对ROI进一步进行分析和理解；
- ROI一般对应图像中特定的、具有独特性质的区域；

5.1 图像分割定义和方法分类

5.1.1 研究背景



CBSR

presents



CBSR

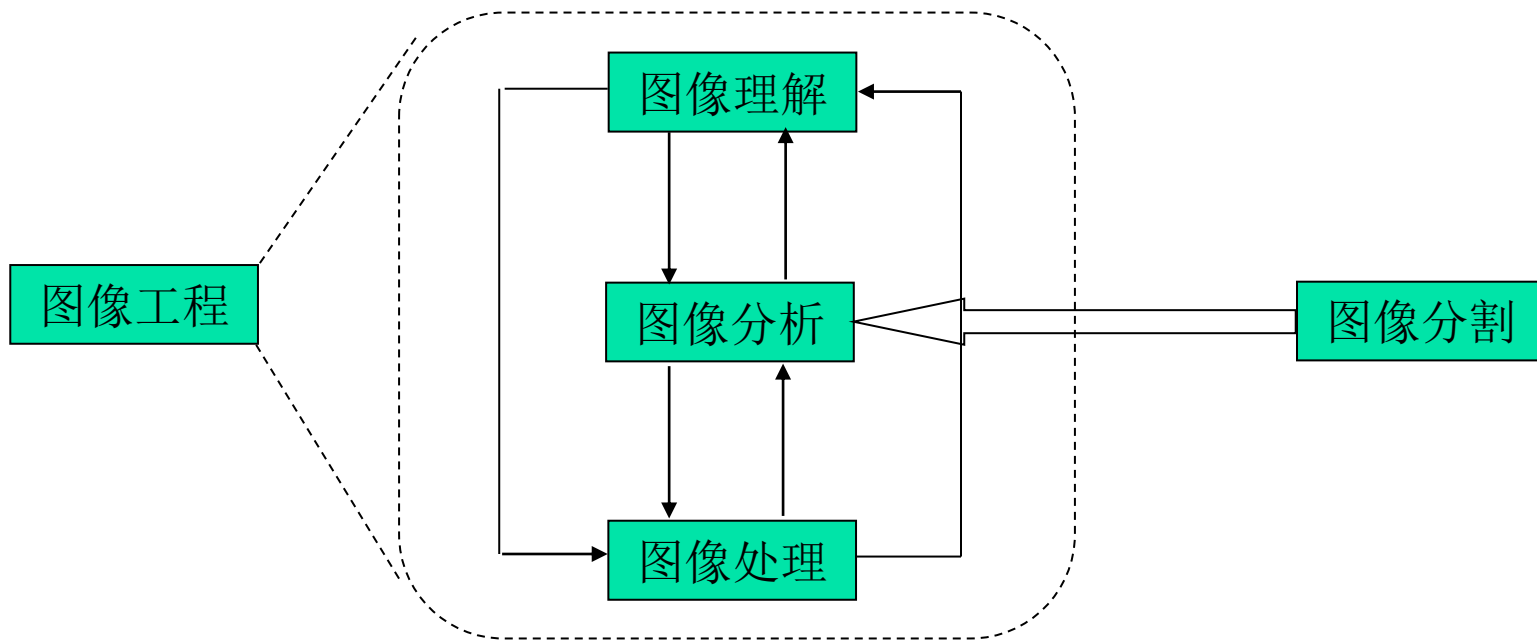
presents

5.1

图像分割定义和方法分类

5.1.1 研究背景

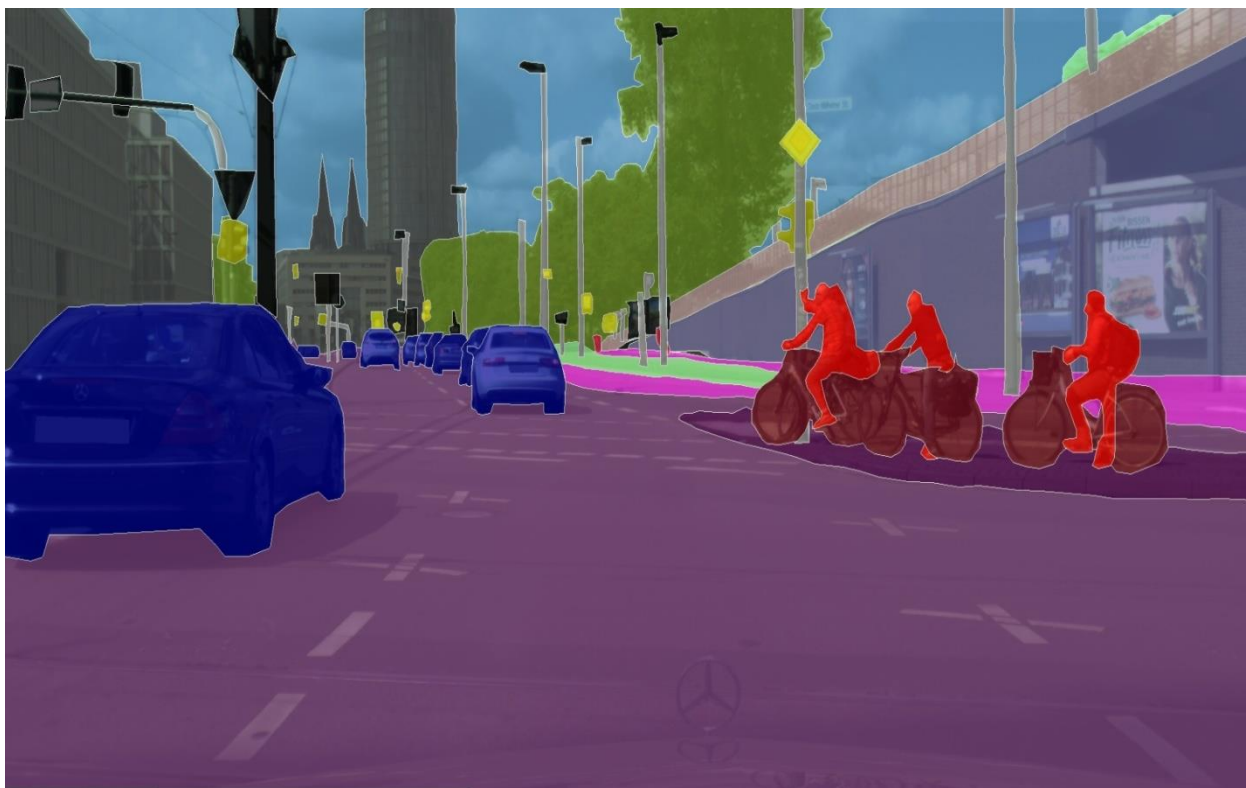
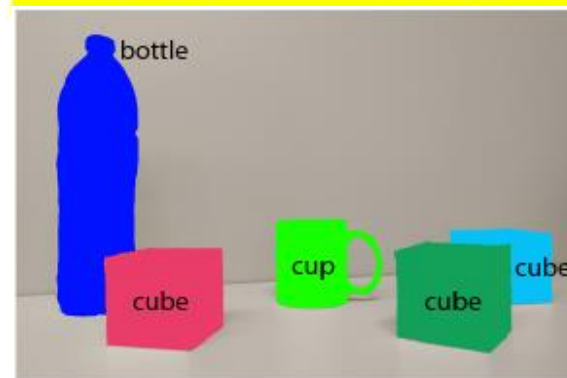
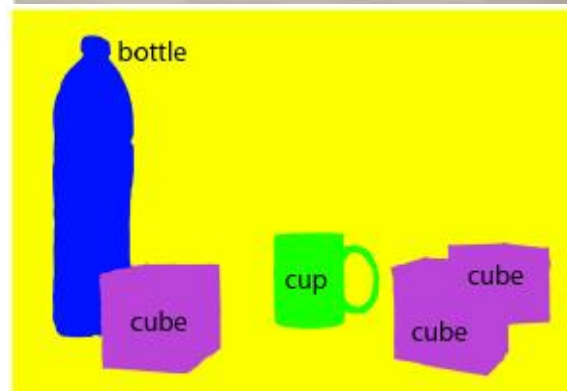
- ✓ 图像分割的研究已有几十年的历史，但还未建立起统一的理论体系，是图像研究中最困难的部分之一。
- ✓ 图像分割是当前图像处理技术的研究热点。



5.1 图像分割定义和方法分类

5.1.1 研究背景

语义分割是当前一个研究热点





5.1

图像分割定义和方法分类

5.1.2 定义

灰度、颜色、纹理

图像分割就是把图像分成各具特性的区域并提取出感兴趣目标的技术和过程。

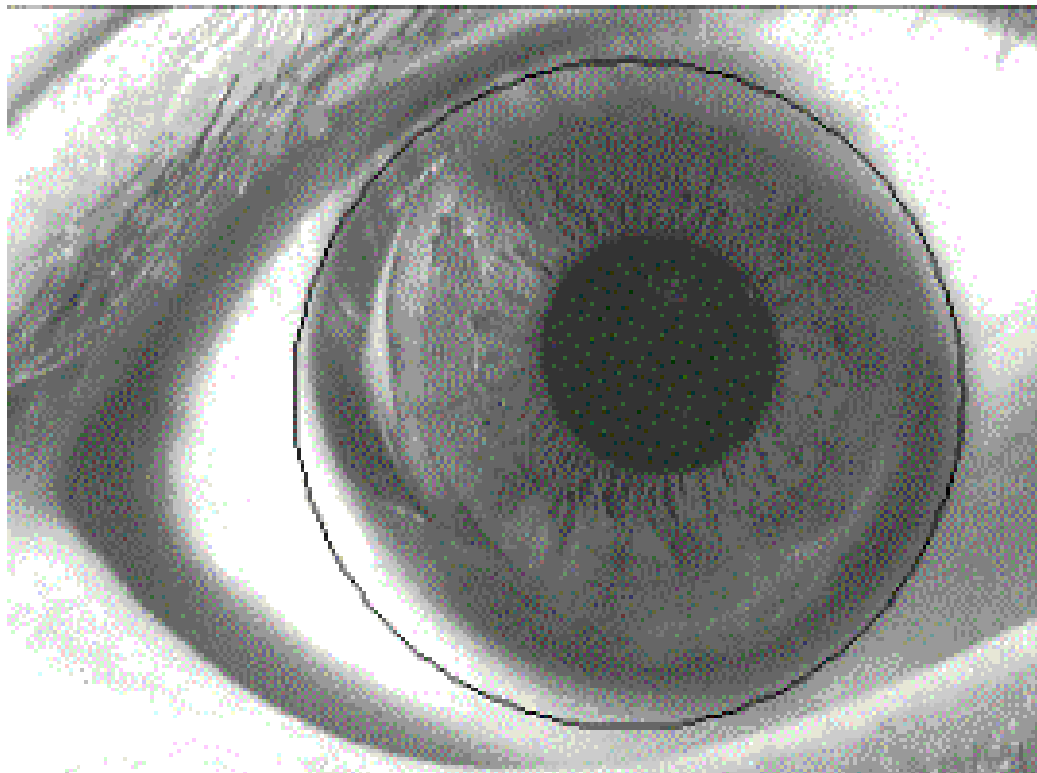
对应单个区域
和多个区域

5.1 图像分割定义和方法分类

5.1.2 定义



虹膜识别





5.1 图像分割定义和方法分类

5.1.2 定义

图象分割的严格定义

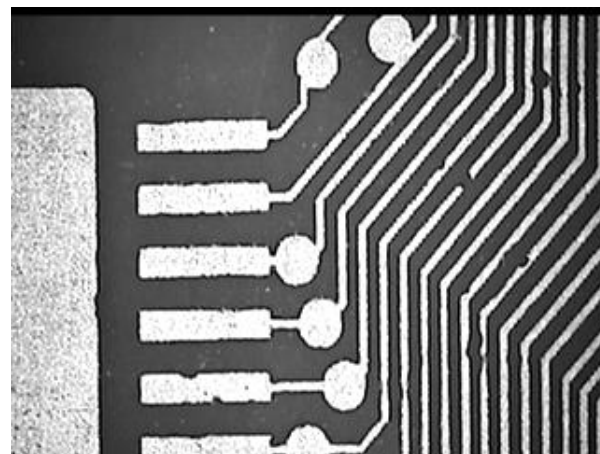
- (1) $\bigcup_{i=1}^n R_i = R$
- (2) 对所有的 i 和 j , $i \neq j$, 有 $R_i \cap R_j = \emptyset$
- (3) 对 $i = 1, 2, \dots, n$, 有 $P(R_i) = \text{TRUE}$
- (4) 对 $i \neq j$, 有 $P(R_i \cup R_j) = \text{FALSE}$
- (5) 对 $i = 1, 2, \dots, n$, R_i 是连通的区域

5.1 图像分割定义和方法分类

5.1.3 方法分类

不连续性和相似性

- ▶ 利用区域间灰度不连续性的基于边界的算法;
- ▶ 利用区域内灰度相似性的基于区域的算法。

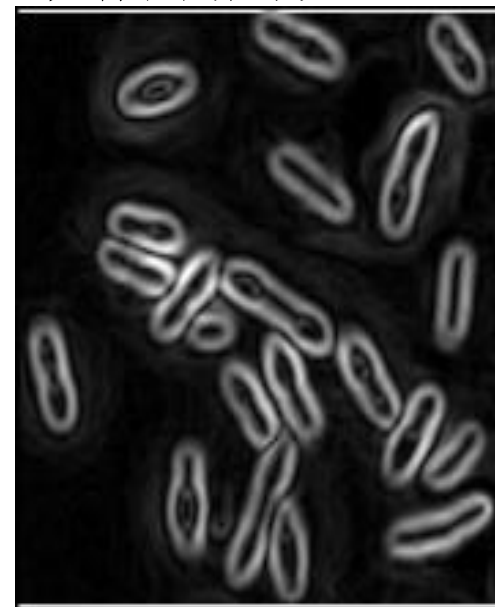
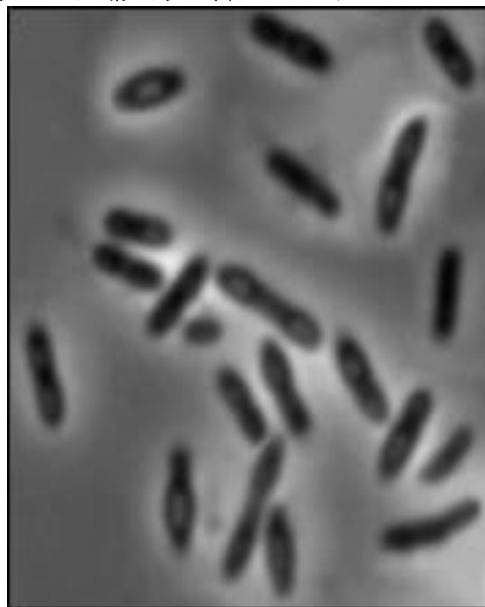


5.1

图像分割定义和方法分类

5.1.3 方法分类 分割过程中的处理策略

- 并行算法：所有判断和决定都可独立地和同时地做出；
 - 串行算法：早期的处理结果可被其后的处理过程所利用；
- 计算时间和抗噪声能力比较





5.1

图像分割定义和方法分类

5.1.3 方法分类

分类表

分类	边界（不连续性）区域（相似性）	
并行处理	并行边界类	并行区域类
串行处理	串行边界类	串行区域类

- ① 并行边界类
- ② 串行边界类
- ③ 并行区域类
- ④ 串行区域类

5.2 边缘检测

5.2.1 基本原理

- 是所有基于边界的分割方法的第一步；
- 边缘是灰度值不连续的结果；
- 可利用求导数方便地检测到；

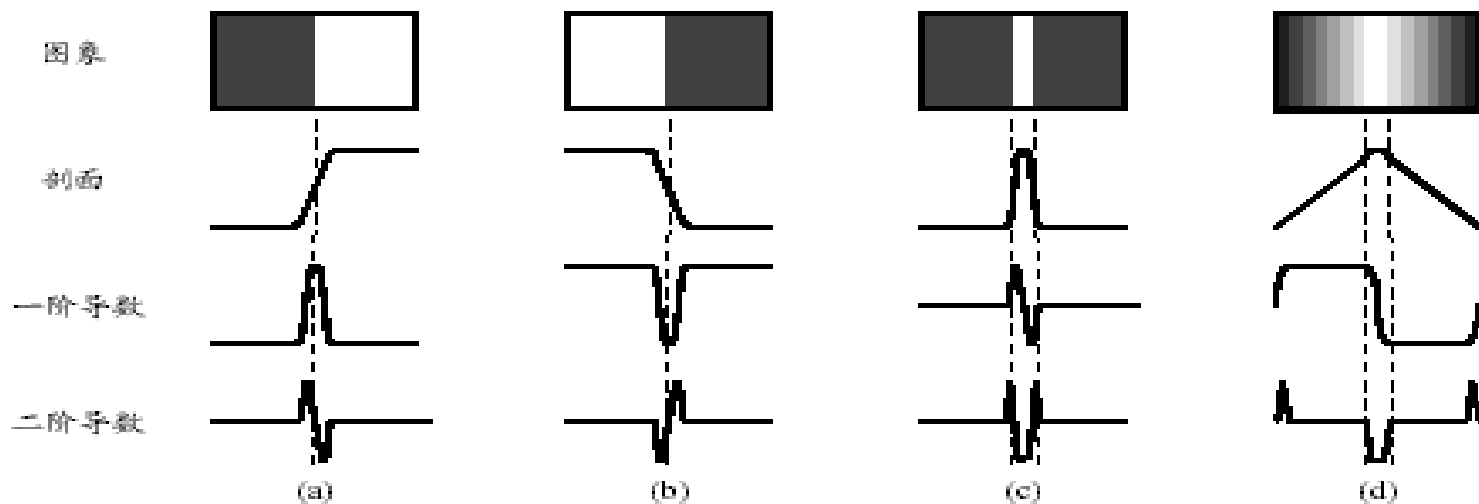


5.2 边缘检测

5.2.1 基本原理

常见的边缘剖面：

- 阶梯状；
- 脉冲状；
- 屋顶状；

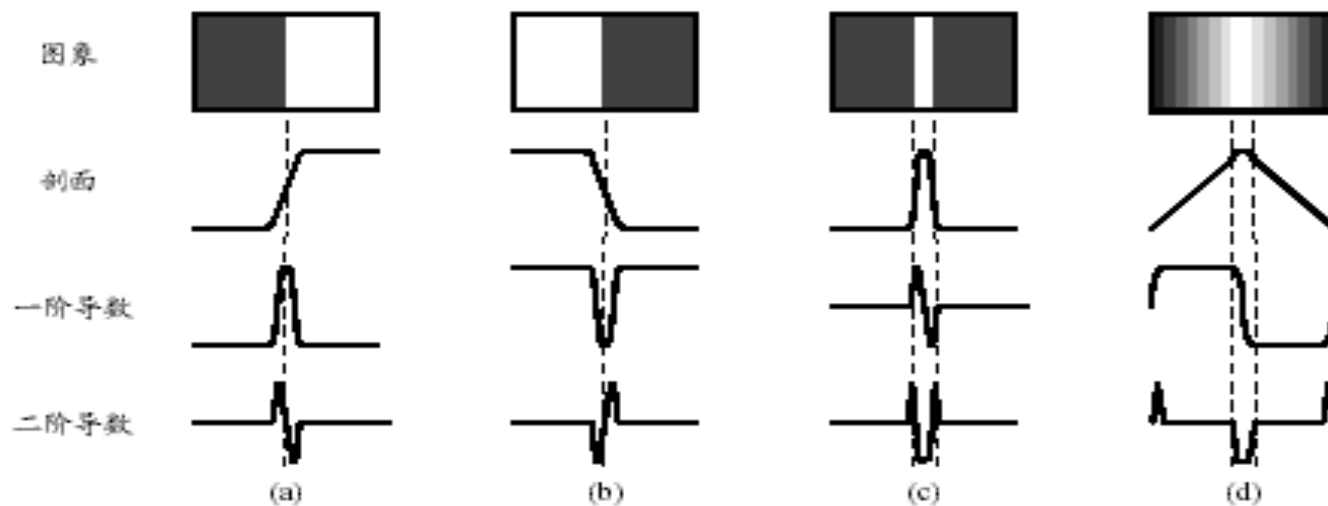


边缘和导数

5.2 边缘检测

5.2.1 基本原理

- 图像中边缘的检测可借助空域微分算子通过卷积完成；
- 利用差分来近似微分来进行；



边缘和导数

5.2 边缘检测

5.2.2 梯度算子-梯度与方向梯度图的计算

$$\nabla f(x, y) = [G_x \quad G_y]^T = \left[\frac{\partial f}{\partial x} \quad \frac{\partial f}{\partial y} \right]^T$$
$$\phi(x, y) = \arctan(G_y / G_x)$$

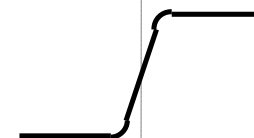
$$f'_x(x, y) = \begin{cases} 1 & |G_x| > T \\ 0 & \text{其它} \end{cases}$$

$$f'_y(x, y) = \begin{cases} 1 & |G_y| > T \\ 0 & \text{其它} \end{cases}$$

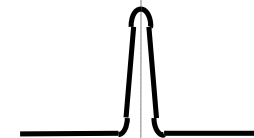
图象



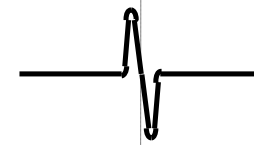
剖面



一阶导数



二阶导数





5.2 边缘检测

5.2.2 梯度算子-梯度幅度图的计算

$$|\nabla f_{(2)}| = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{\frac{1}{2}}$$

$$|\nabla f_{(1)}| = \left| \frac{\partial f}{\partial x} \right| + \left| \frac{\partial f}{\partial y} \right|$$

$$|\nabla f_{(\infty)}| = \max \left\{ \left| \frac{\partial f}{\partial x} \right|, \left| \frac{\partial f}{\partial y} \right| \right\}$$

$$f'(x, y) = \begin{cases} 1 & |\nabla f| > T \\ 0 & \text{其它} \end{cases}$$

5.2 边缘检测

5.2.2 梯度算子

利用模板（与图象进行）卷积

1	
	-1

	1
-1	

(a) Roberts

-1		1
-1		1
-1		1

1	1	1
-1	-1	-1

(b) Prewitt

-1		1
-2		2
-1		1

1	2	1
-1	-2	-1

(c) Sobel

5.2 边缘检测

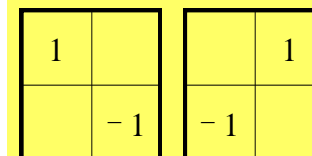
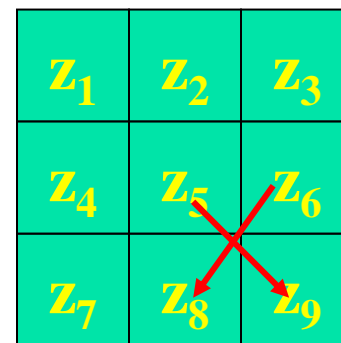
5.2.2 梯度算子

Roberts交叉梯度算子

- 由两个模板组成，第一个求得梯度的第一项 $G_x = z_5 - z_9$ ，第二个求得梯度的第二项 $G_y = z_6 - z_8$ ；
- 得到方向梯度图。

$$f'_x(x, y) = \begin{cases} 1 & |G_x| > T \\ 0 & \text{其它} \end{cases}$$

$$f'_y(x, y) = \begin{cases} 1 & |G_y| > T \\ 0 & \text{其它} \end{cases}$$



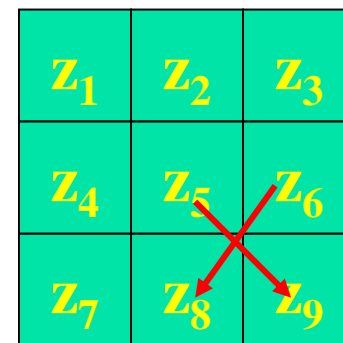
(a) Roberts

5.2 边缘检测

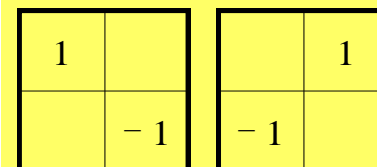
5.2.2 梯度算子

Roberts交叉梯度算子

$$\nabla f \approx |z_5 - z_9| + |z_6 - z_8|$$



$$f'(x, y) = \begin{cases} 1 & |\nabla f| > T' \\ 0 & \text{其它} \end{cases}$$



(a) Roberts

5.2 边缘检测

5.2.2 梯度算子

Prewitt梯度算子

- $G_x = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$,
 $G_y = (z_1 + z_2 + z_3) - (z_7 + z_8 + z_9)$;
- 得到方向梯度图。

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$$f'_x(x, y) = \begin{cases} 1 & |G_x| > T \\ 0 & \text{其它} \end{cases}$$

$$f'_y(x, y) = \begin{cases} 1 & |G_y| > T \\ 0 & \text{其它} \end{cases}$$

-1		1
-1		1
-1		1

1	1	1
-1	-1	-1

(b) Prewitt

5.2 边缘检测

5.2.2 梯度算子

Prewitt梯度算子

$$\nabla f \approx |(z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)| \\ + |(z_1 + z_2 + z_3) - (z_7 + z_8 + z_9)|$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$$f'(x, y) = \begin{cases} 1 & |\nabla f| > T' \\ 0 & \text{其它} \end{cases}$$

-1		1	1	1	1
-1		1			
-1		1	-1	-1	-1

(b) Prewitt

5.2 边缘检测

5.2.2 梯度算子

Soble梯度算子

- $G_x = (z_3 + 2z_6 + z_9) - (z_1 + z_4 + z_7)$,
 $G_y = (z_1 + 2z_2 + z_3) - (z_7 + z_8 + z_9)$;
- 得到方向梯度图。

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$$f'_x(x, y) = \begin{cases} 1 & |G_x| > T \\ 0 & \text{其它} \end{cases}$$

$$f'_y(x, y) = \begin{cases} 1 & |G_y| > T \\ 0 & \text{其它} \end{cases}$$

-1		1	1	2	1
-2		2			
-1		1	-1	-2	-1

(c) Sobel

5.2 边缘检测

5.2.2 梯度算子

Soble梯度算子

$$\nabla f \approx |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)| \\ + |(z_1 + 2z_2 + z_3) - (z_7 + 2z_8 + z_9)|$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

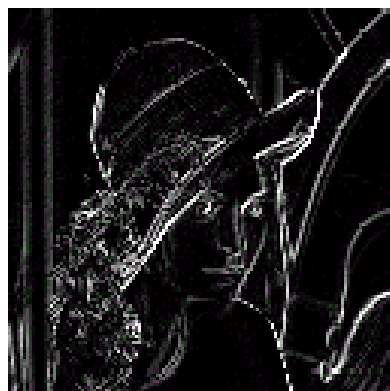
$$f'(x, y) = \begin{cases} 1 & |\nabla f| > T' \\ 0 & \text{其它} \end{cases}$$

-1		1	1	2	1
-2		2			
-1		1	-1	-2	-1

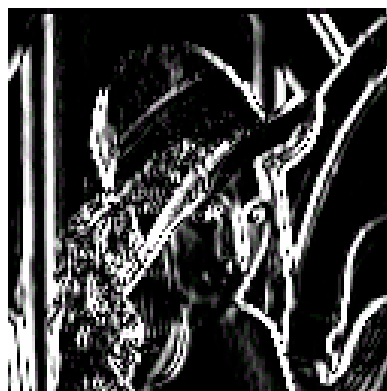
(c) Sobel

5.2 边缘检测

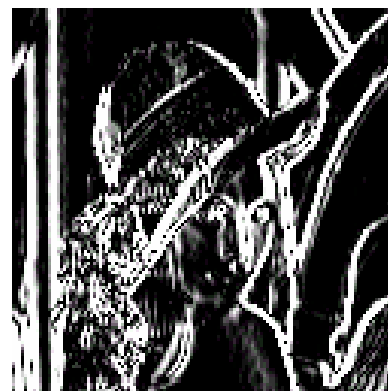
5.2.2 梯度算子



(a)



(b)



(c)

方向梯度图比较

5.2 边缘检测

5.2.2 梯度算子

利用模板（与图象进行）卷积

1	
	-1

	1
-1	

(a) Roberts

-1		1
-1		1
-1		1

1	1	1
-1	-1	-1

(b) Prewitt

-1		1
-2		2
-1		1

1	2	1
-1	-2	-1

(c) Sobel

模板比较

- ① 边缘粗细； ② 方向性

5.2 边缘检测

5.2.2 梯度算子



Roberts边缘检测效果图
例

5.2 边缘检测

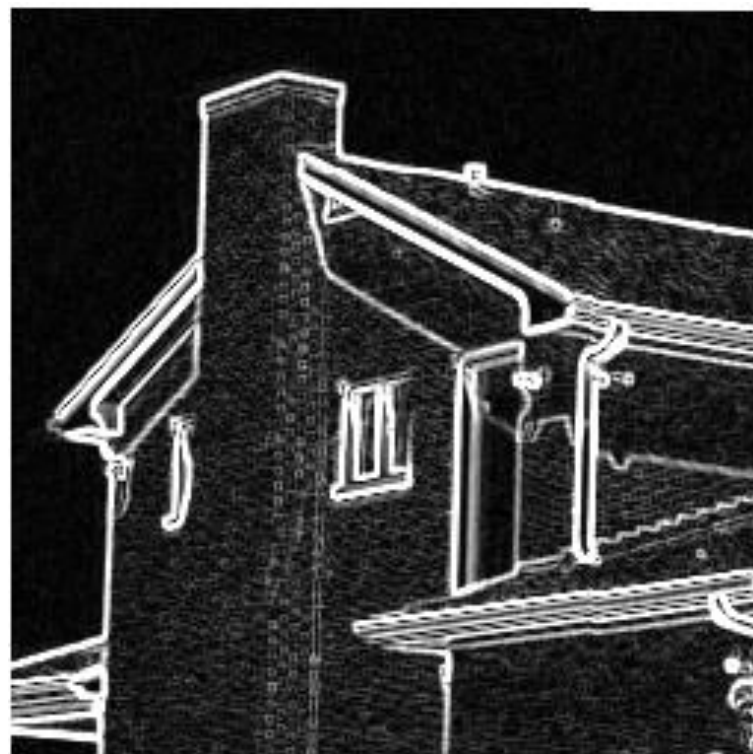
5.2.2 梯度算子



Prewitt边缘检测效果图例

5.2 边缘检测

5.2.2 梯度算子



Sobel边缘检测效果图例



5.2 边缘检测

5.2.2 梯度算子

$$|\nabla f_{(2)}| = \left[\left(\frac{\partial f}{\partial x} \right)^2 + \left(\frac{\partial f}{\partial y} \right)^2 \right]^{1/2}$$

$$|\nabla f_{(1)}| = \left| \frac{\partial f}{\partial x} \right| + \left| \frac{\partial f}{\partial y} \right|$$

$$|\nabla f_{(\infty)}| = \max \left\{ \left| \frac{\partial f}{\partial x} \right|, \left| \frac{\partial f}{\partial y} \right| \right\}$$

5.2 边缘检测

5.2.2 梯度算子



(a)



(b)



(c)

图 6.2.4 不同模梯度幅度图示例

以2为范数的梯度比以1和 ∞ 为范数的梯度更灵敏一些。

5.2 边缘检测

5.2.3 拉普拉斯算子

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

0	-1	0
-1	4	-1
0	-1	0

(a)

-1	-1	-1
-1	8	-1
-1	-1	-1

(b)

拉普拉斯算子的模板

- 对应中心像素的系数为正；
- 对应中心像素的邻近像素的系数为负数；
- 所有系数的和为零。

5.2 边缘检测

5.2.3 拉普拉斯算子

- 拉普拉斯算子的特点
- ① 对图像中的噪声特别敏感;
- ② 常产生双像素宽的边缘;



(a)

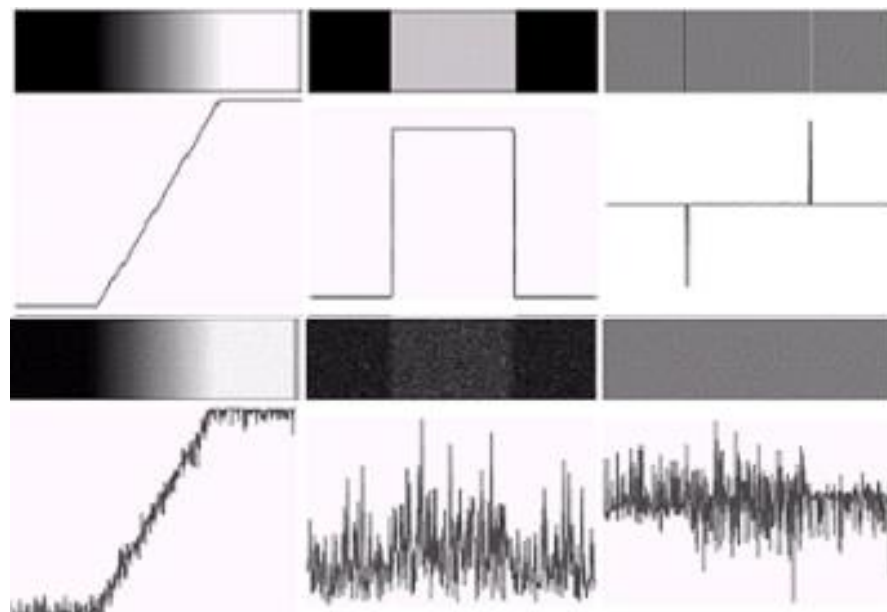


(b)

用拉普拉斯算子的模板检测出的边缘点

5.2 边缘检测

- 边缘检测中经常碰到的问题是：
 - 图像中存在太多的细节。比如，前面例子中的砖墙
 - 图像受到噪声的干扰，不能准确的检测边缘
- 解决的一个方法是在边缘检测之前对图像进行平滑





5.2 边缘检测

5.2.3 拉普拉斯算子

马尔算子

- (1) 用一个2-D的高斯平滑模板与源图象卷积
- (2) 计算卷积后图象的拉普拉斯值
- (3) 检测拉普拉斯图象中的过零点作为边缘点



5.3 轮廓跟踪和图搜索

5.3.1 基本原理

- 检测出边缘后还需将它们连接以构成闭合的轮廓；
- 一般可采用先检测可能的边缘点再串行跟踪的方法；
- 另外，也可将边缘检测和轮廓跟踪互相结合、顺序进行；



5.3 轮廓跟踪和图搜索

5.3.2 轮廓跟踪

- **边缘点连接**，是由（梯度图中）一个边缘点出发，依次搜索并连接相邻边缘点从而逐步检测出轮廓的方法；
- **为了保持轮廓的光滑性**，在搜索时每确定一个新的轮廓点都要考虑先前已得到的轮廓点；
- **为了克服噪声造成的边缘点之间的不连通**，要充分保持梯度图已有的信息。



5.3

轮廓跟踪和图搜索

5.3.2 轮廓跟踪

轮廓跟踪步骤:

- 确定搜索起点;
- 确定数据结构和搜索机理;
- 确定终止条件。

5.3 轮廓跟踪和图搜索

5.3.2 轮廓跟踪

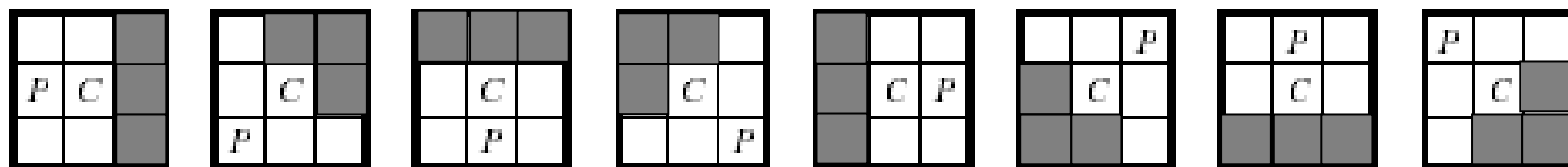


图 6.3.1 八种搜索窗口

- 对噪声较小的图像效果较好；
- 可确定出目标具有最大梯度的轮廓；
- 图像中噪声较大时，会偏离正确轮廓或失踪。



5.3 轮廓跟踪和图搜索

5.3.3 图搜索法

- 噪声较大时仍有比较好的效果；
- 借助状态空间搜索来寻求全局最优的轮廓；
- 将轮廓点和轮廓段用图结构表示，通过在图中搜索对应最小代价的通道来寻找闭合轮廓。

5.3 轮廓跟踪和图搜索

5.3.3 图搜索法

将轮廓点和轮廓段用图表示

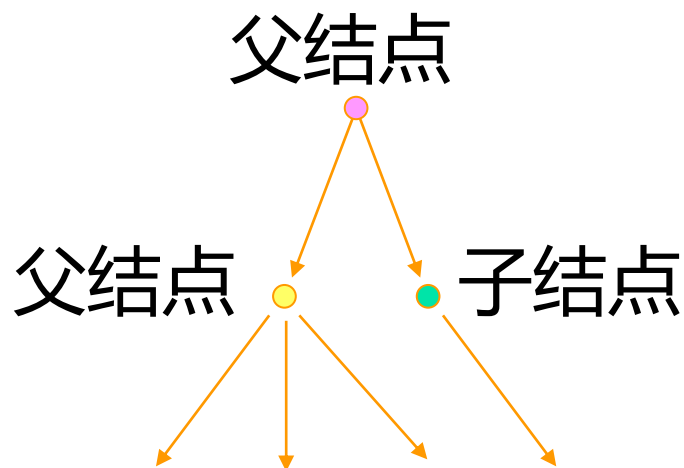
图: $G = \{N, A\}$

结点集 $\{n_1, \dots\}$

结点对集 $\{(n_i, n_j)\}$ - 弧

通路代价

$$C = \sum_{i=2}^K c(n_{i-1}, n_i)$$

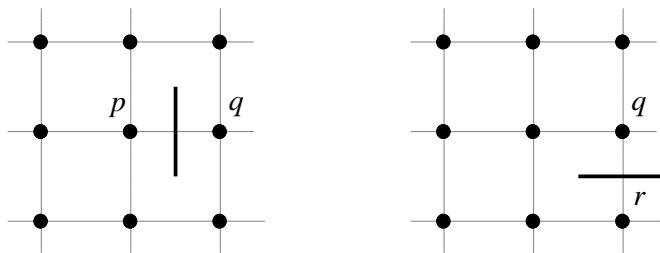


5.3

轮廓跟踪和图搜索

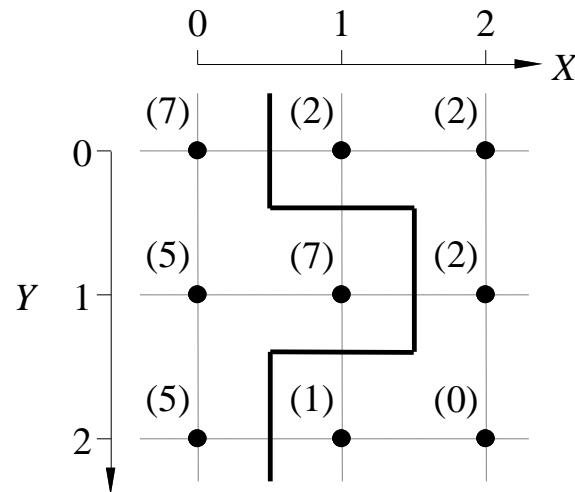
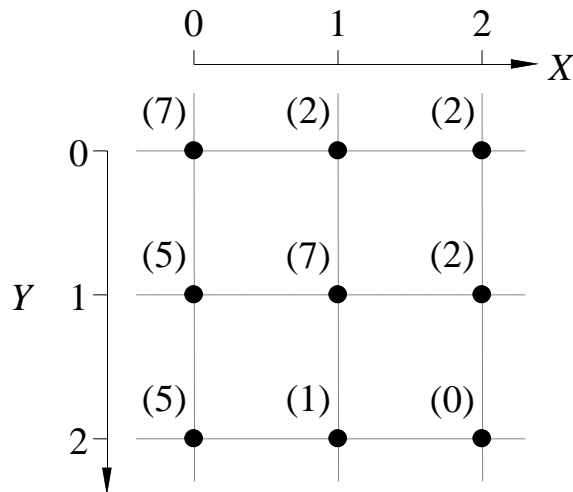
5.3.3 图搜索法

边缘元素



代价函数

$$c(p, q) = H - [f(p) - f(q)]$$



5.3 轮廓跟踪和图搜索

5.3.3 图搜索法

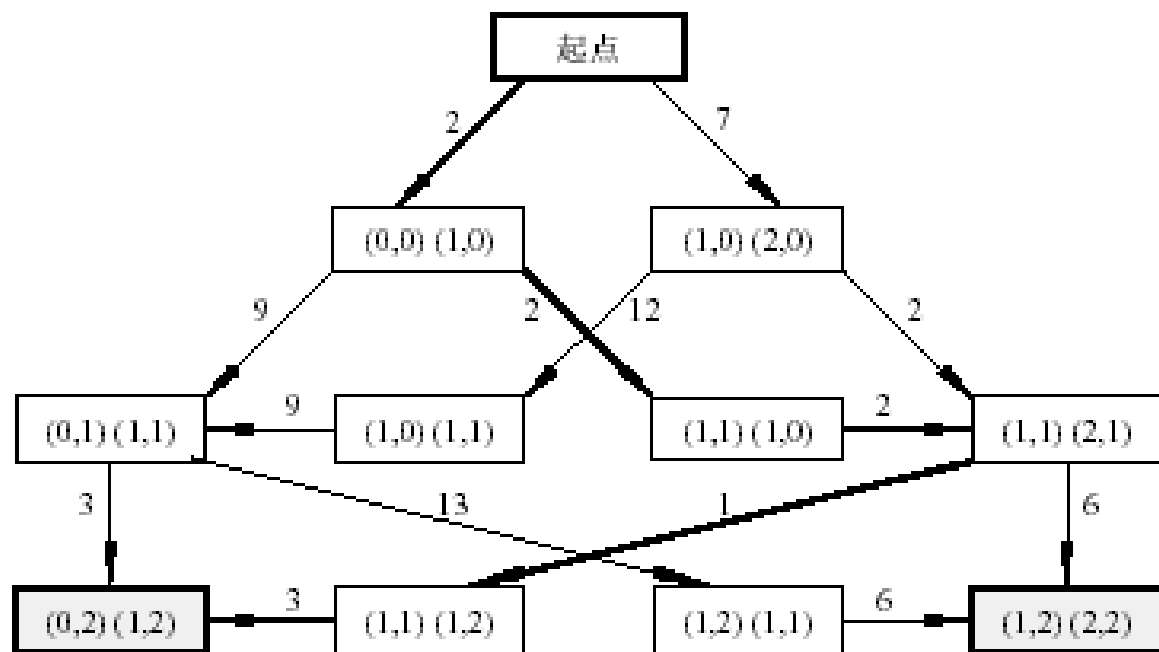
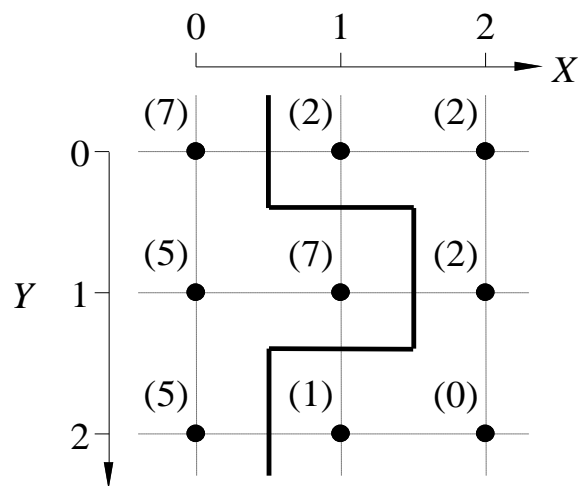


图 6.3.4 用于检测轮廓的搜索图

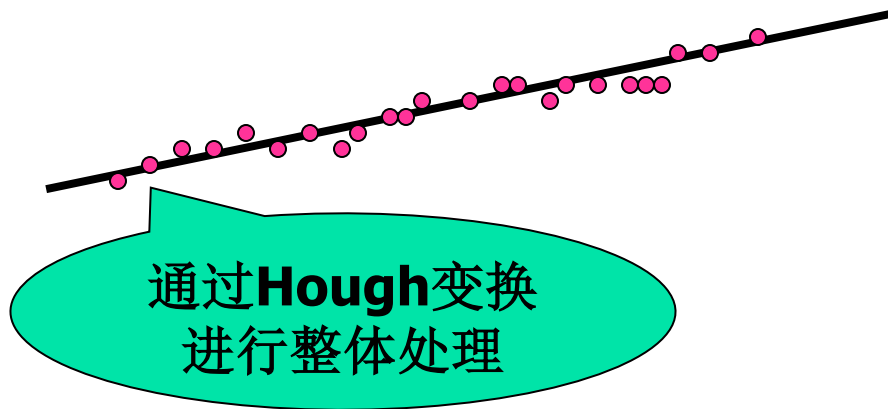


5.4 哈夫变换

5.4.1 基本原理

问题的提出

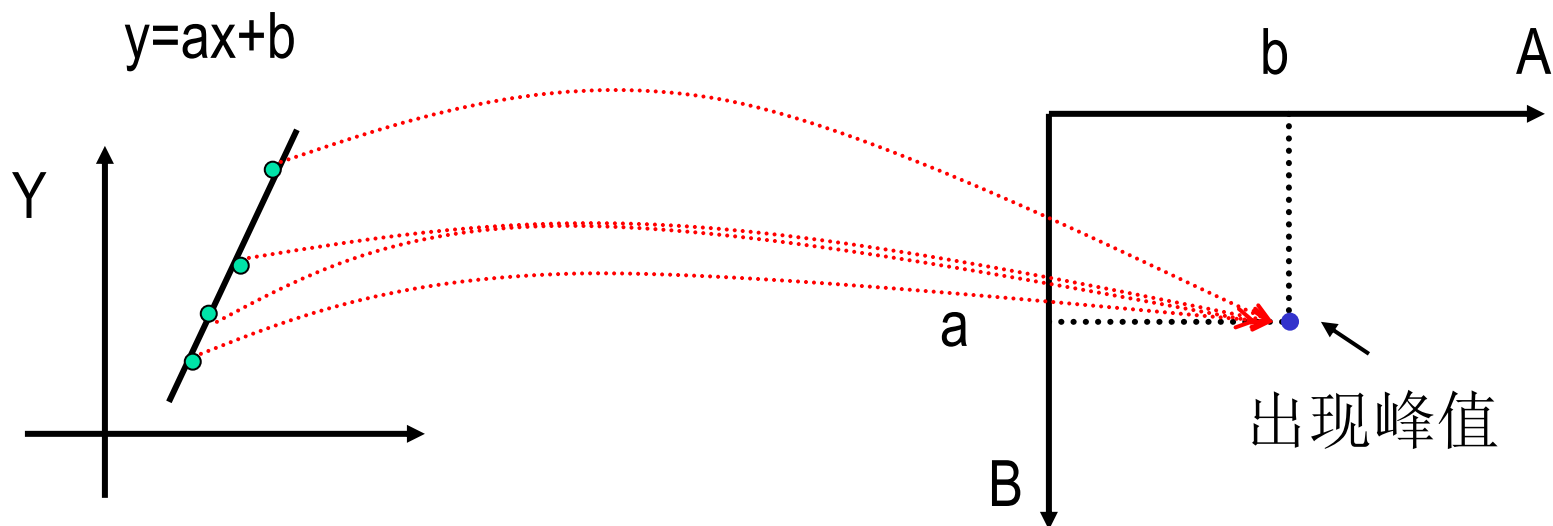
在找出边界点集之后，需要连接，形成完整的边界图形描述。



5.4 哈夫变换

5.4.1 基本原理

- 设图像空间中有一个目标，其轮廓可用代数方程表示；
- 方程中既有图像空间坐标的变量，也有属于参数空间的变量；
- 哈夫变换是图像空间和参数空间之间的一种变换。





5.4 哈夫变换

5.4.1 基本原理

作用：1.利用图像全局特性将目标边缘像素连接起来组成轮廓；2.对图像中已知形状的目标进行检测。

特点：受噪声和边界间断的影响较小，比较稳健。

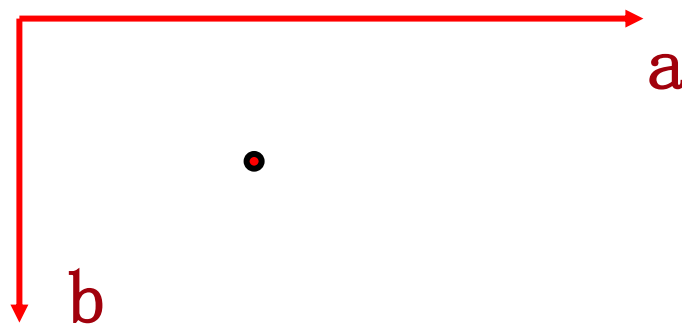
寻找一种从区域边界到参数空间的变换，用大多数边界点满足的对应的参数来描述这个区域的边界。



5.4 哈夫变换

5.4.2 点-线对偶性

对于边界上的 n 个点的点集，找出共线的点集和直线方程。设任意两点的直线方程： $y = ax + b$ ，构造一个参数 a ， b 的平面。



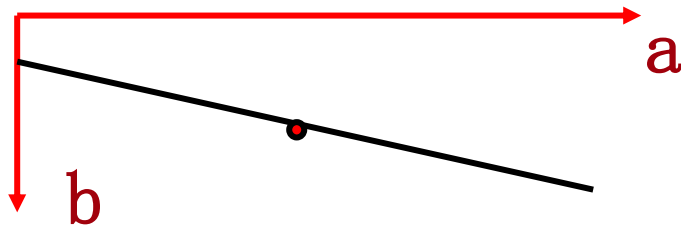
5.4 哈夫变换

5.4.2 点-线对偶性

A、 xy 平面上的任意一条直线 $y = ax + b$ ，对应参数 ab 平面上都有一个点。



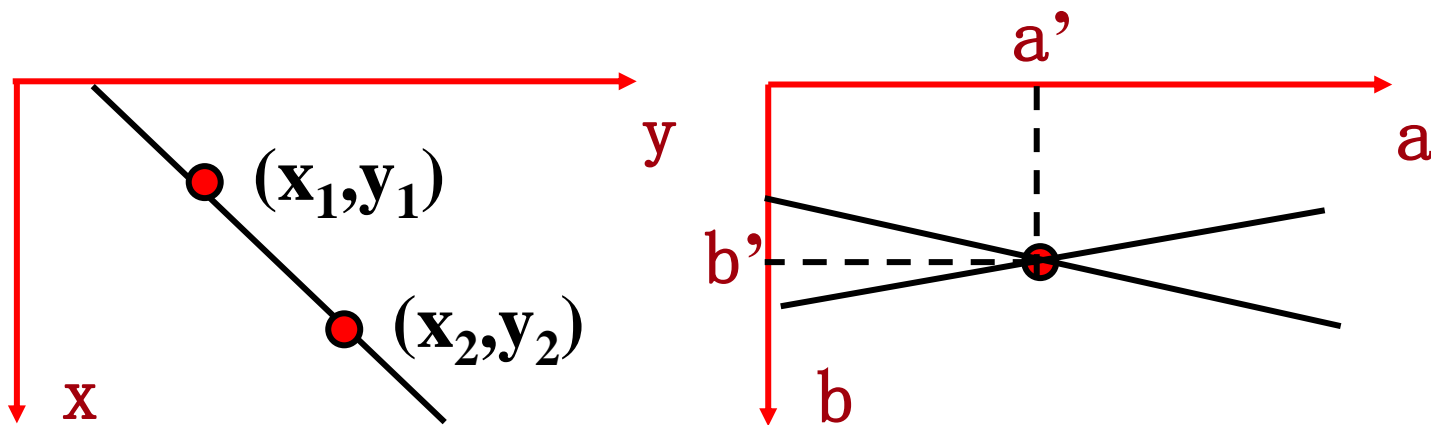
B、 xy 平面中的一个点 (x,y) ，对应参数 ab 平面上的一条直线。



5.4 哈夫变换

5.4.2 点-线对偶性

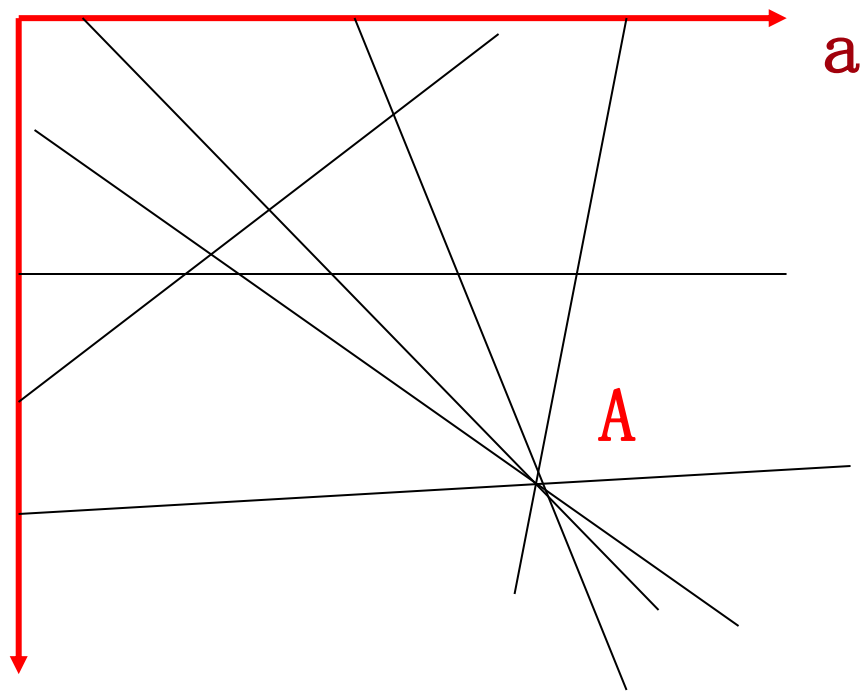
C、如果点 (x_1, y_1) 与点 (x_2, y_2) 共线，那么这两点在参数 ab 平面上的直线将有一个交点。



5.4 哈夫变换

5.4.2 点-线对偶性

D、在参数 ab 平面上相交直线最多的点，对应的 xy 平面上的直线就是我们的解。





5.4 哈夫变换

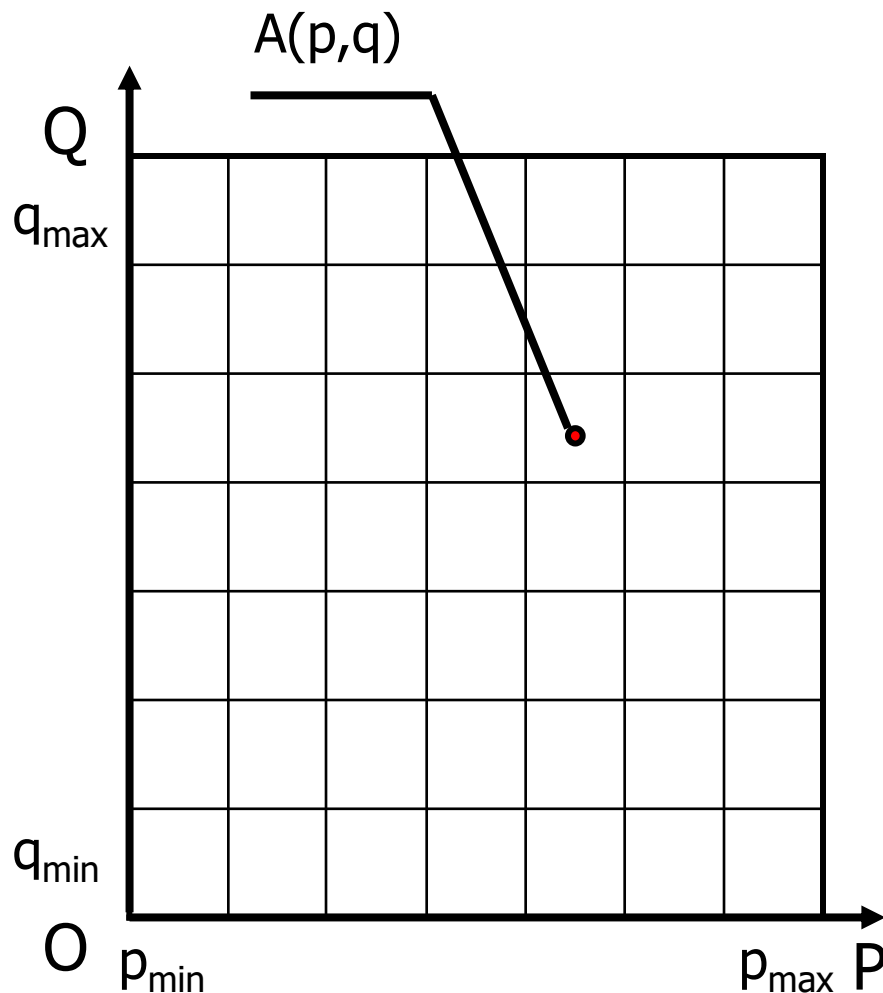
5.4.2 点-线对偶性

结论：在图像空间中共线的点对应参数空间里相交的线；反过来，参数空间中相交于同一个点的所有直线在图像空间里都有共线的点与之对应。

5.4 哈夫变换

5.4.3 计算步骤与示例

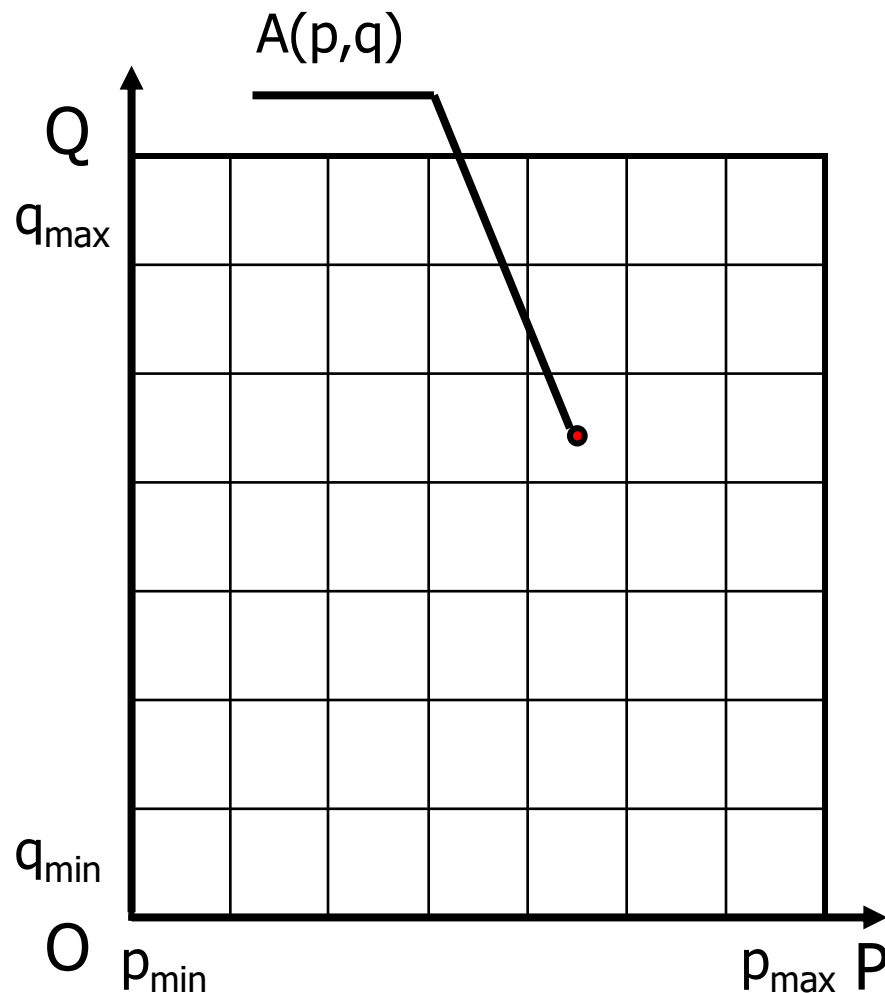
- 对参数空间中参数 p 和 q 的可能取值范围进行量化，根据量化结果构造一个累加数组
 $A(p_{\min}:p_{\max}, q_{\min}:q_{\max})$ ，并初始化为零；
- 对每个XY空间中的给定点让 p 取遍所有可能值，计算出 q ，根据 p 和 q 值累加 A : $A(p, q) = A(p, q) + 1$ ；
- 根据累加后 A 中最大值所对应的 p 和 q ，定出XY中的一个直线。 A 中最大值代表了在此直线上给定点的数目。



5.4 哈夫变换

5.4.3 计算步骤与示例

- 可用来检测满足解析式 $f(x,c)=0$ 形式的各类曲线并把曲线上的点连接起来。
- x 为坐标矢量， c 为系数矢量。





5.4 哈夫变换

5.4.4 举例-圆形的检测

$$(x-a)^2 + (y-b)^2 = r^2$$

- 在参数空间建立3-D的累加数组A(a,b,r)；
- 对每个XY空间中的给定点让a和b取遍所有可能值，计算出r，累加A:
 $A(a,b,r) = A(a,b,r) + 1$;
- 根据累加后A中最大值所对应的a，b和r，定出XY中的一个圆形。A中最大值代表了在此圆形上点的数目。



5.5 阈值分割

5.5.1 基本原理

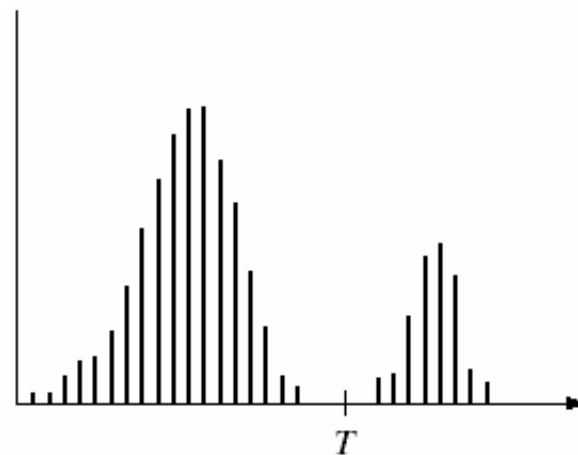
思路：根据图像中要提取的目标物与背景在灰度特性上的差异，确定一个阈值，将图像二值化，即将目标从背景中分离出来。

5.5 閾値分割

5.5.1 基本原理

(1) 単閾値分割

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





5.5 阈值分割

5.5.1 基本原理

什么样的图像适合用单阈值分割算法（图像模型）：

- 图像由具有单峰灰度分布的目标和背景组成；
 - 灰度直方图两个分布大小接近；
- 在目标或背景内部的相邻像素的灰度值是高度相关的；
 - 两个分布均方差足够小；
- 在目标和背景交界处两边的像素具有较大的灰度差。
 - 两个分布均值相距足够远。



5.5 阈值分割

5.5.1 基本原理

阈值分割步骤： \Rightarrow ③ 并行区域类

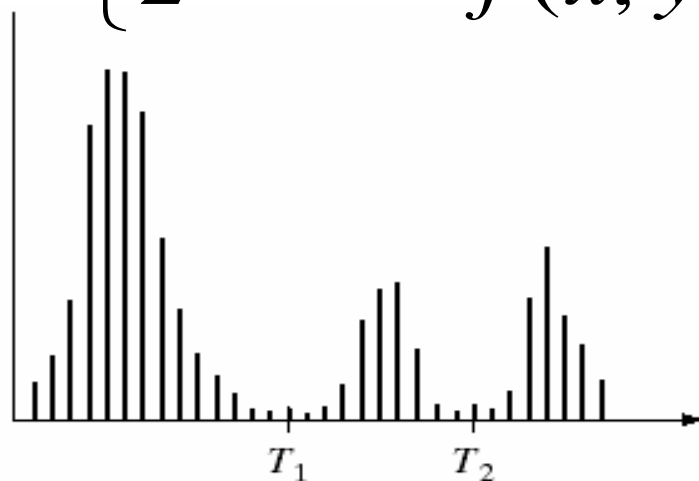
- (1) 确定阈值（关键/难点）
- (2) 根据阈值对像素分类

5.5 閾値分割

5.5.1 基本原理

(2) 多閾値分割

$$g(x, y) = \begin{cases} 0 & f(x, y) \leq T_1 \\ 1 & T_1 < f(x, y) \leq T_2 \\ 2 & f(x, y) > T_2 \end{cases}$$

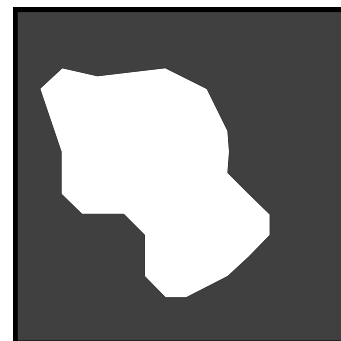
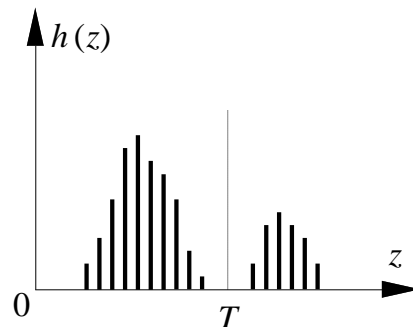
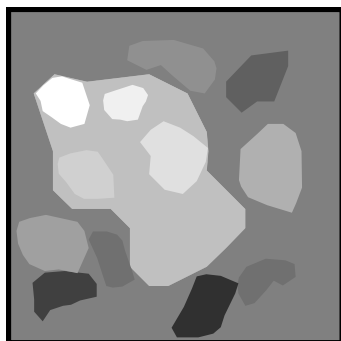


5.5 阈值分割

5.5.2 单阈值分割举例

对灰度图（取值在 g_{\min} 和 g_{\max} 之间）确定一个灰度阈值 T （ $g_{\min} < T < g_{\max}$ ）

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

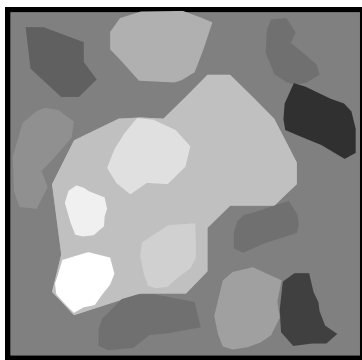


5.5 閾值分割

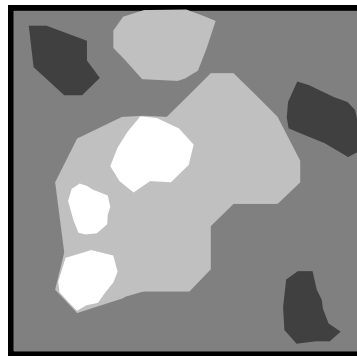
5.5.2 多閾值分割举例

➤ 确定一系列分割閾值

$$g(x, y) = k \quad \text{如} \quad T_k < f(x, y) \leq T_{k+1} \quad k = 0, 1, 2, \dots, K$$



(a)



(c)



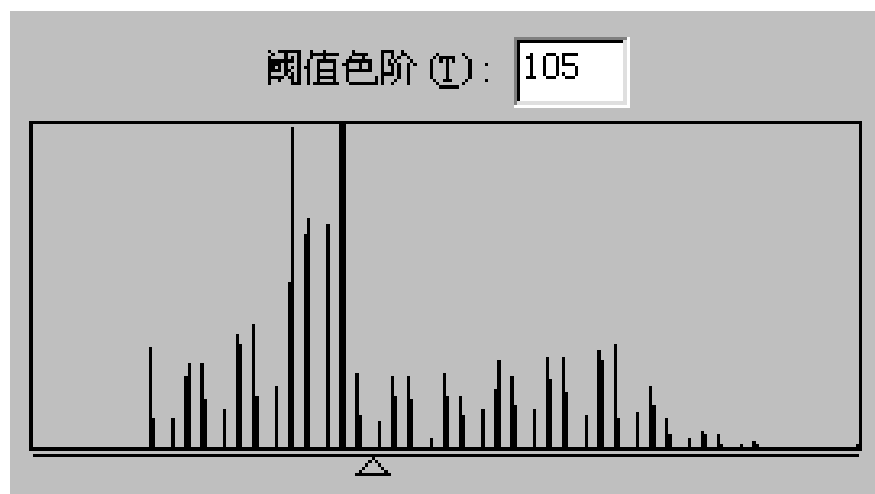
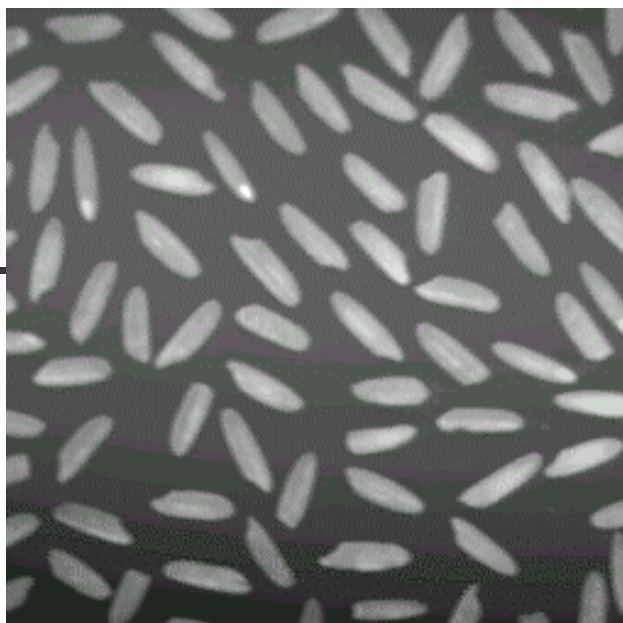
5.5 阈值分割

5.5.3 极小值点阈值

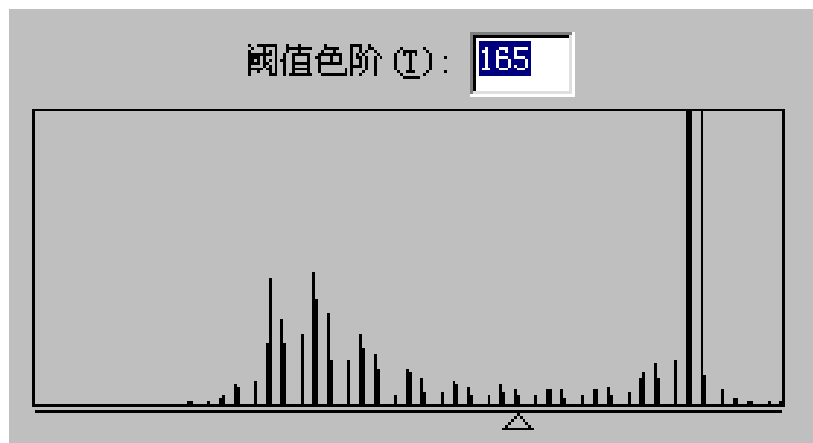
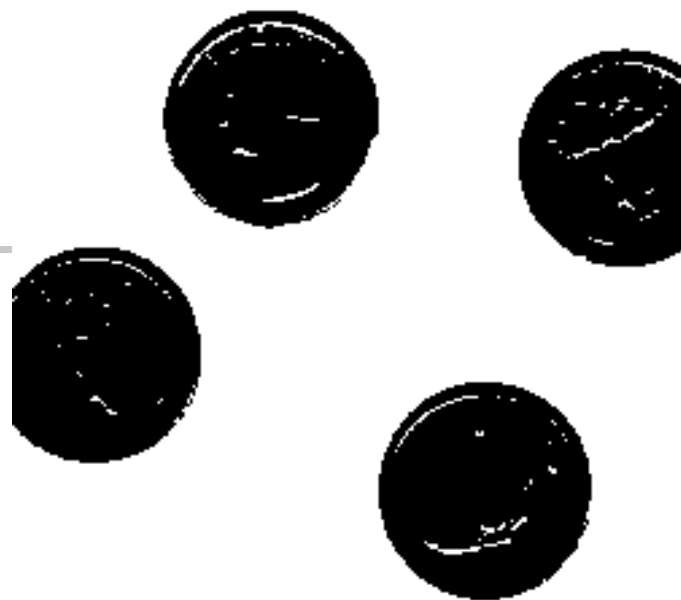
- 将直方图的包络看成1条曲线；
- 借助求曲线的极小值的方法选取直方图的谷；

$$\frac{\partial h(z)}{\partial z} = 0$$

$$\frac{\partial^2 h(z)}{\partial z^2} > 0$$



由直方图确定阈值进行分割 (1)



由直方图确定阈值进行分割 (2)



5.5 阈值分割

5.5.4 最优阈值

- 有时目标和背景之间有部分交错，用1个全局阈值不能将它们绝对分开；
- 为了减小误分割的概率，常使用最优阈值分割；

5.5 阈值分割

5.5.4 最优阈值

- 设 μ_1 和 μ_2 分别是背景和目标区域的平均灰度值；
- 设 σ_1 和 σ_2 分别是背景和目标区域关于均值的均方差；
- 设 P_1 和 P_2 分别是背景和目标区域灰度值的先验概率。

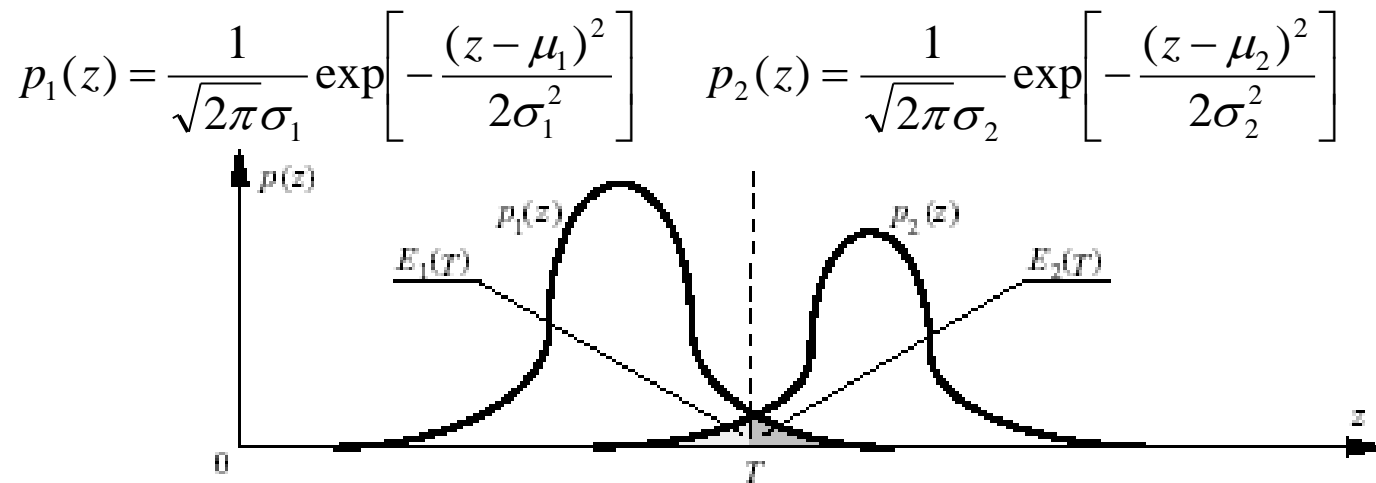


图 6.4.2 最优阈值选取示例

5.5 阈值分割

5.5.4 最优阈值

错误地将1个目标像素划分为背景的概率为：
$$E_1(T) = \int_{-\infty}^T p_2(z) dz$$

错误地将1个背景像素划分为目标的概率为：
$$E_2(T) = \int_T^{\infty} p_1(z) dz$$

总的误差概率为：
$$E(T) = P_2 E_1(T) + P_1 E_2(T)$$

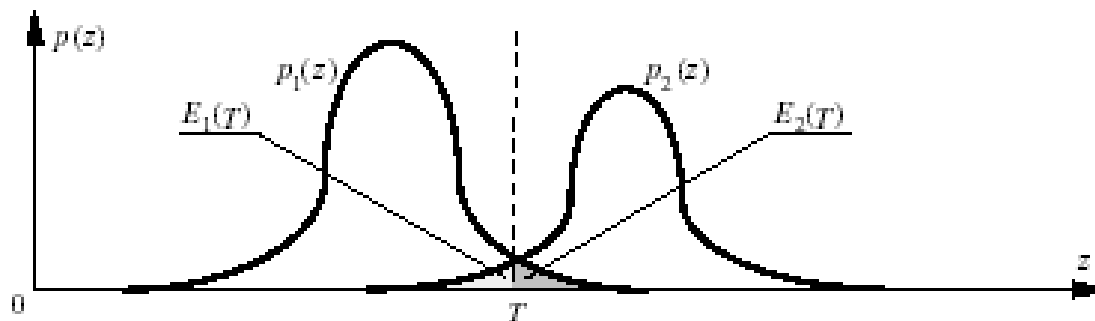


图 6.4.2 最优阈值选取示例

5.5 阈值分割

5.5.4 最优阈值

为求得使该误差最小的阈值，将 $E(T)$ 对 T 求导并令导数为零

$$P_1 p_1(T) = P_2 p_2(T)$$

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

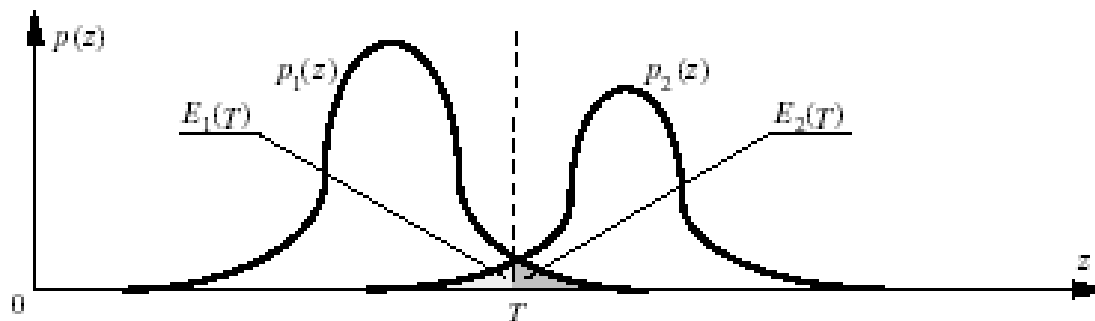


图 6.4.2 最优阈值选取示例



5.5 閾値分割

5.5.5 迭代閾値

$$T_{i+1} = \frac{1}{2} \left\{ \frac{\sum_{k=0}^{T_i} h_k \cdot k}{\sum_{K=0}^{T_i} h_k} + \frac{\sum_{k=T_i+1}^{L-1} h_k \cdot k}{\sum_{k=T_i+1}^{L-1} h_k} \right\}$$

- ✓ 初始閾値 T_0 取图像灰度范围的中值；
- ✓ 迭代一直进行到 $T_{i+1} = T_i$ 结束，取结束时的 T_i ；
- ✓ 迭代閾値从路径规划的角度看也是一种最优閾値。

5.5 阈值分割

5.5.5 迭代阈值

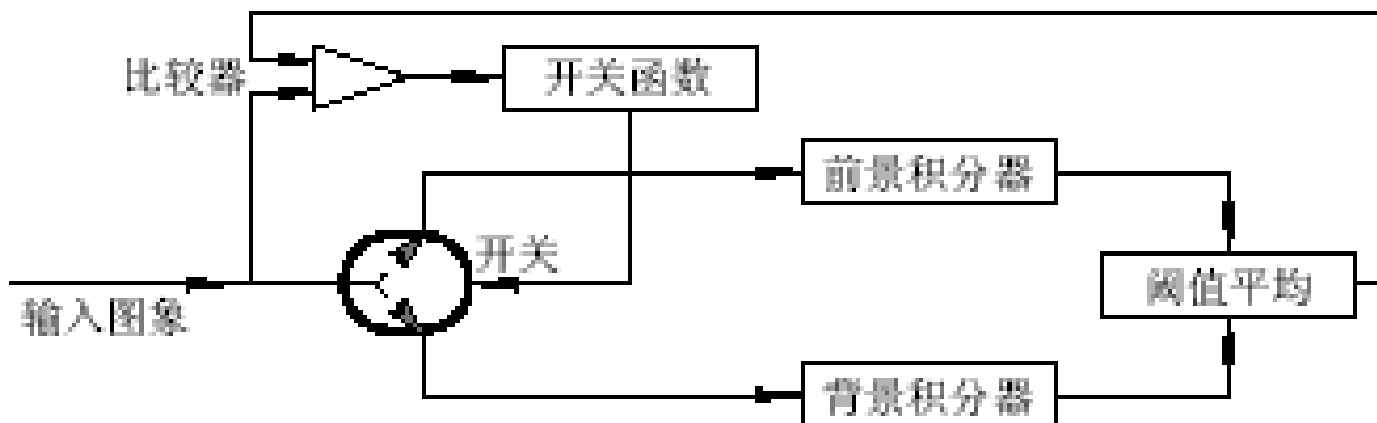


图 6.4.3 迭代选择法的流程和实现

- ✓ 迭代阈值从路径规划的角度看也是一种最优阈值。



5.5 阈值分割

5.5.6 最大类间方差法——大津方法（Otsu）

- ✓ Otsu一直被认为是阈值自动选取方法的最优方法。
- ✓ 基本思想：
 - 设定一个阈值 k ，将图像分成两组。
 - 变动 k 的取值使得两组的类间方差最大，此时该值 K 为所求分割阈值。
- ✓ 计算简单，受图像对比度与亮度变化的影响小。



5.5 阈值分割

5.5.6 最大类间方差法——大津方法 (Otsu)

- ✓ $M \times N$ 大小的图像 $f(x, y)$ ，灰度级取值为 $[0 \sim m-1]$ （通常为 $[0, 255]$ ）。记 $p(k)$ 为灰度 k 的概率，则有：

$$p(k) = \frac{1}{MN} \sum_{f(i,j)=k} 1$$

- ✓ 设用灰度值 t 为阈值分割出的目标与背景，则：

- 目标部分比例：
$$w_0(t) = \sum_{0 \leq i \leq t} p(i)$$

- 目标部分点数：
$$N_0 = MN \sum_{0 \leq i \leq t} p(i)$$



5.5 阈值分割

5.5.6 最大类间方差法——大津方法 (Otsu)

- 背景部分比例: $w_1(t) = \sum_{t < i \leq m-1} p(i)$
- 背景部分点数: $N_1 = MN \sum_{t < i \leq m-1} p(i)$
- 目标均值: $\mu_0(t) = \sum_{0 \leq i \leq t} ip(i) / w_0(t)$
- 背景均值: $\mu_1(t) = \sum_{t < i \leq m-1} ip(i) / w_1(t)$
- 总均值: $\mu = w_0(t)\mu_0(t) + w_1(t)\mu_1(t)$



5.5 阈值分割

5.5.6 最大类间方差法——大津方法 (Otsu)

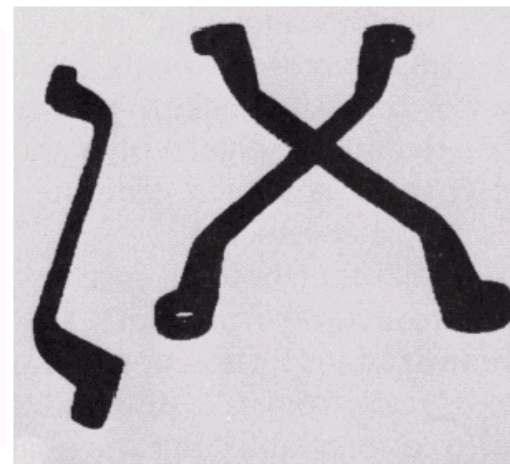
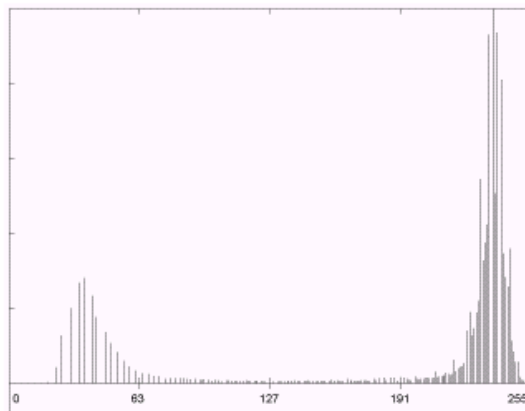
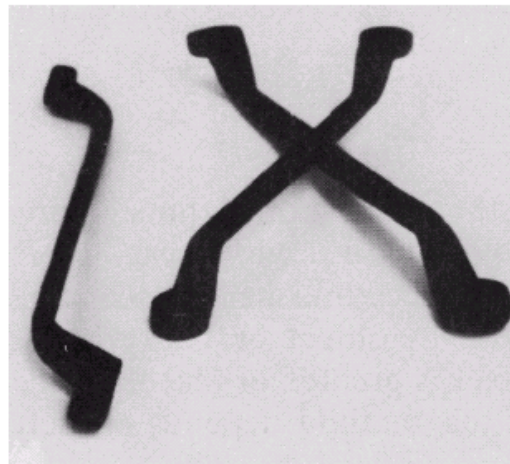
✓ 图像最佳阈值g

$$g = \text{Max}[w_0(t)(\mu_0(t) - \mu)^2 + w_1(t)(\mu_1(t) - \mu)^2]$$

- 右边括号内实际上就是类间方差值。
- 当部分目标错分为背景或部分背景错分为目标都会导致两部分差别变小。
- 使类间方差最大的分割意味着错分概率最小。

5.5 閾值分割

5.5.7 效果举例

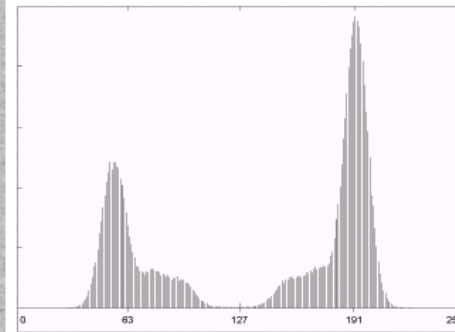


a
b c

FIGURE 10.28
(a) Original image. (b) Image histogram. (c) Result of global thresholding with T midway between the maximum and minimum gray levels.

5.5 阈值分割

5.5.7 效果举例



a b
c

FIGURE 10.29

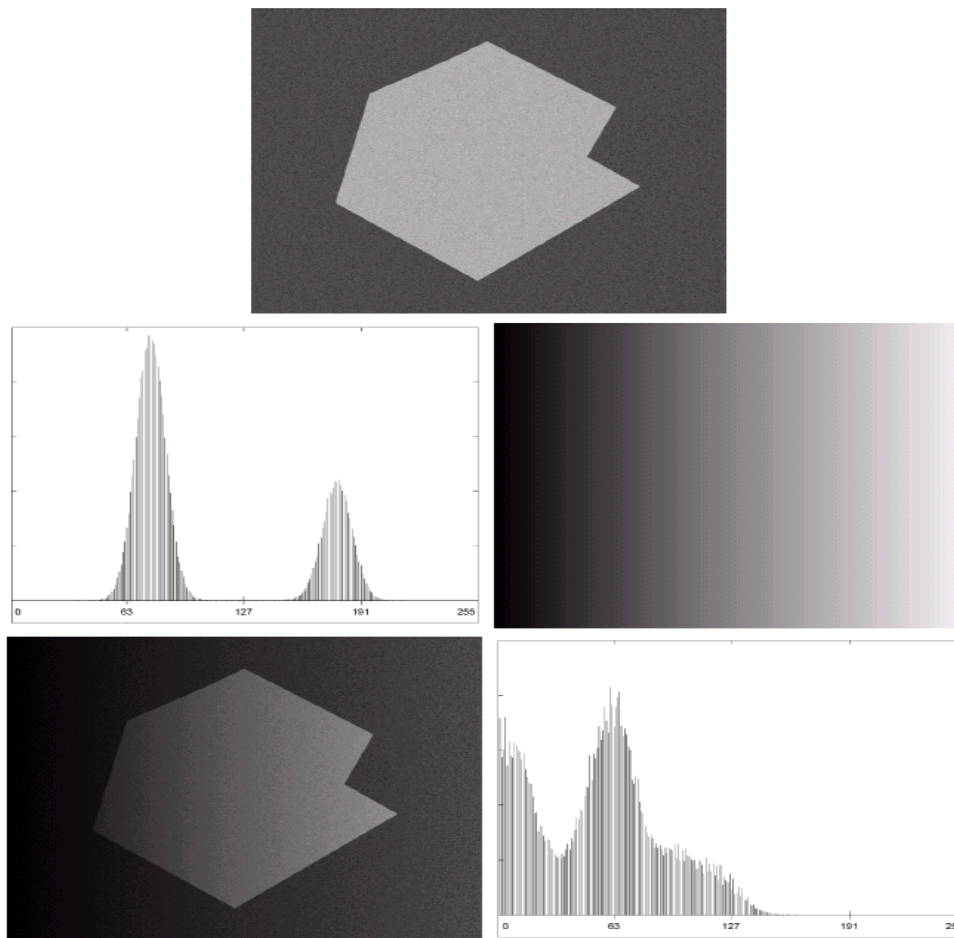
(a) Original image. (b) Image histogram. (c) Result of segmentation with the threshold estimated by iteration. (Original courtesy of the National Institute of Standards and Technology.)



5.5 阈值分割

5.5.7 改进算法举例-自适应阈值

不均匀亮度等造成单一全局门限无法有效分割。一种改进措施是将图像进一步细分为不同的子图像，并对不同的子图像使用不同的门限进行分割。



a
b c
d e

FIGURE 10.27
(a) Computer generated reflectance function.
(b) Histogram of reflectance function.
(c) Computer generated illumination function.
(d) Product of (a) and (c).
(e) Histogram of product image.

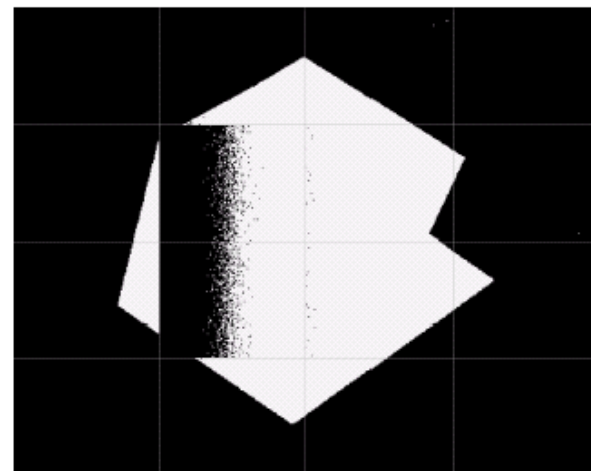
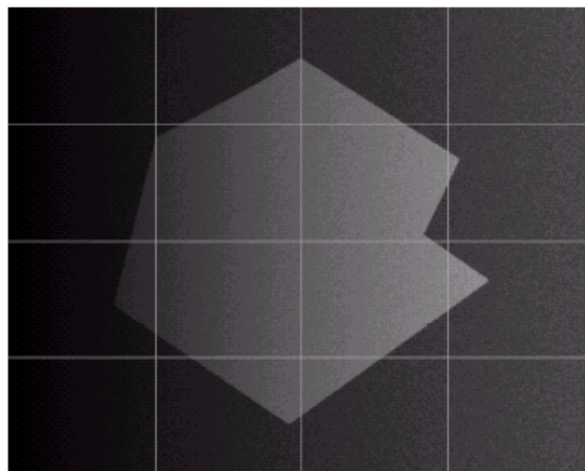
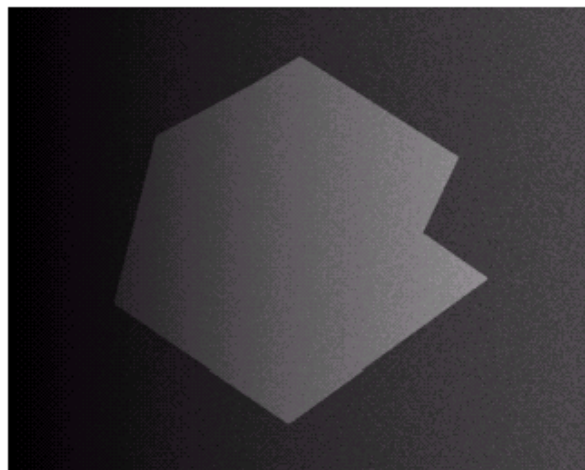
5.5 阈值分割

5.5.7 改进算法举例-自适应阈值

a	b
c	d

FIGURE 10.30

(a) Original image. (b) Result of global thresholding. (c) Image subdivided into individual subimages. (d) Result of adaptive thresholding.



你有什么其它的改进方法？



5.6 基于变换直方图选取阈值

5.6.1 算法背景

- 图像受到噪声等影响可能使直方图中原本分离的峰之间的谷被填充；
- 可利用一些像素邻域的局部性质来解决这类问题。



5.6 基于变换直方图选取阈值

5.6.2 直方图变换

- 基本思想：利用一些像素领域的局部性质来变换原来的直方图以得到**1**个新的直方图；
- 目标和背景内部的像素具有较低的梯度值，而它们边界上的像素具有较高的梯度值。
- 新的直方图可分为两类：
 - ①具有低梯度值像素的直方图；
 - ②具有高梯度值像素的直方图。

5.6 基于变换直方图选取阈值

5.6.2 直方图变换

- 做出仅具有低梯度值的像素的直方图；
- 计算一个加权的直方图；
- 峰基本不变而谷变深，峰谷差距加大

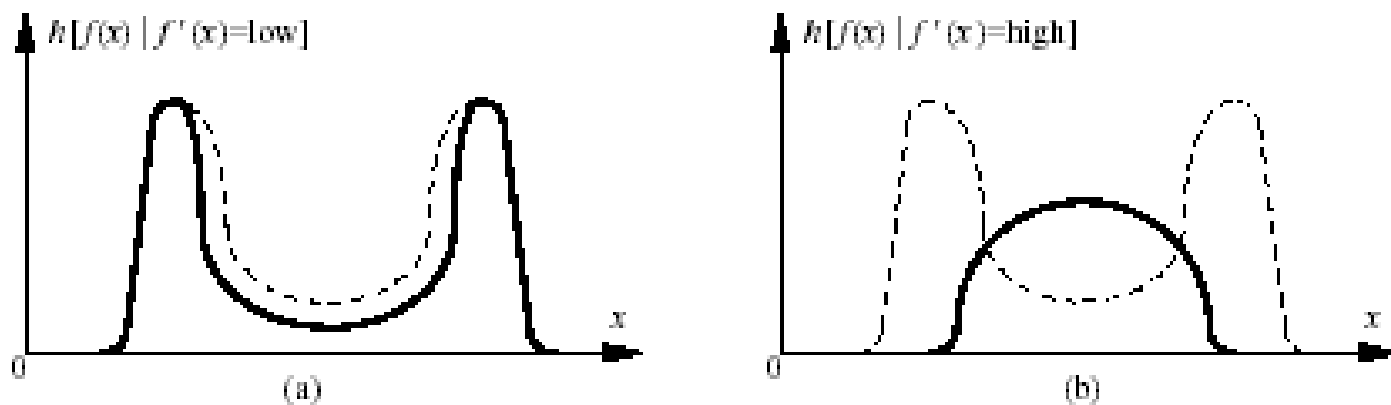


图 6.5.1 变换直方图示例

5.6 基于变换直方图选取阈值

5.6.2 直方图变换

- 做出仅具有高梯度值的像素的直方图；
- 计算一个加权的直方图；
- 对应目标和背景的边界像素灰度级处有1个峰。

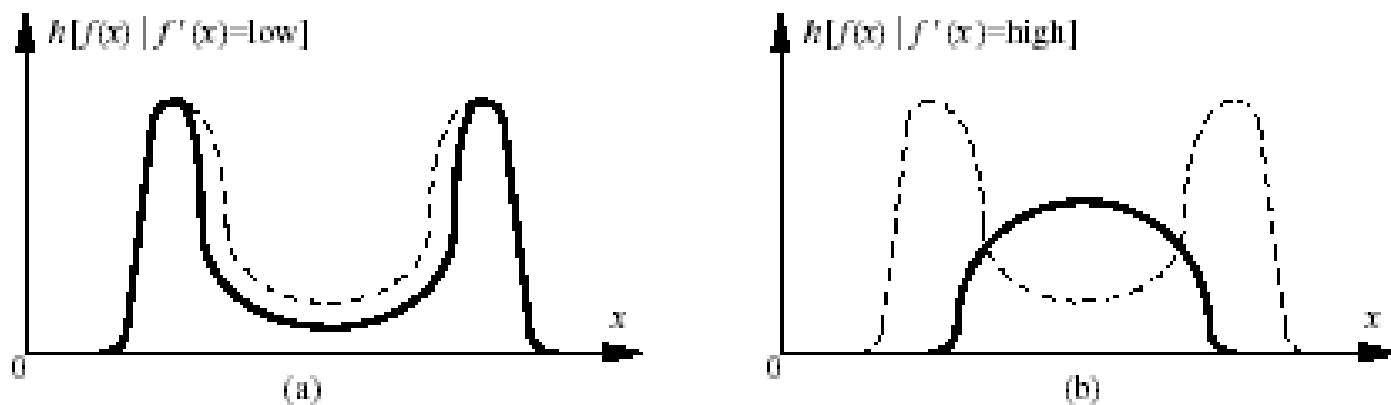


图 6.5.1 变换直方图示例

5.6 基于变换直方图选取阈值

5.6.2 直方图变换

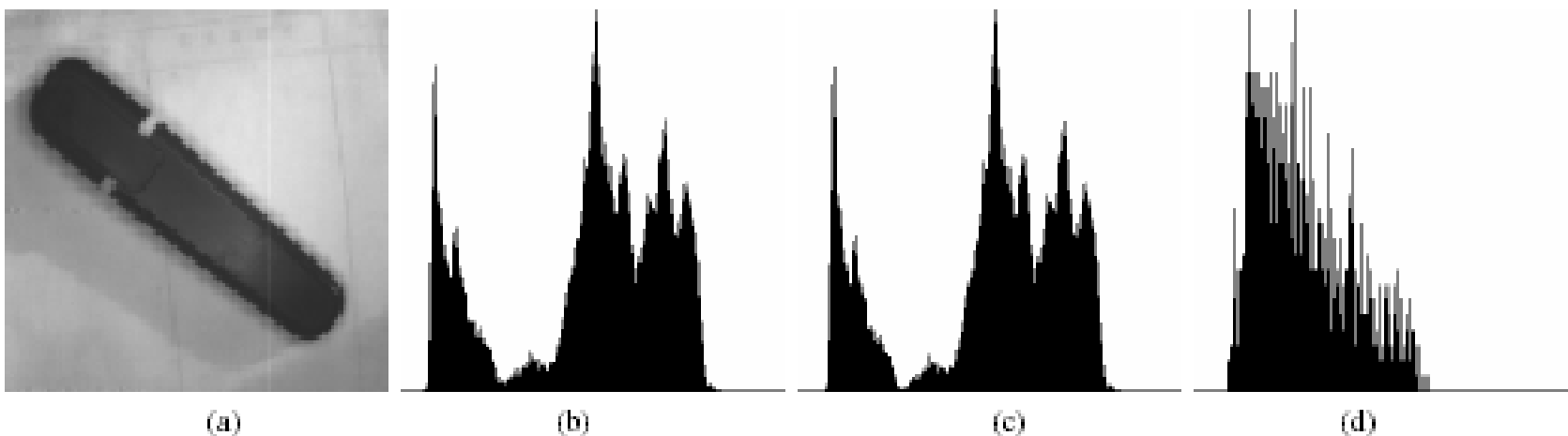


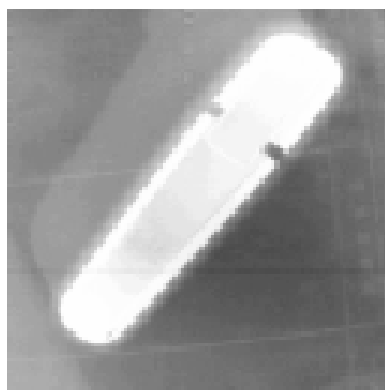
图 6.5.2 变换直方图实例

- 1) 低梯度像素直方图:峰不变、谷变深
- 2) 高梯度像素直方图:谷变峰、易检测

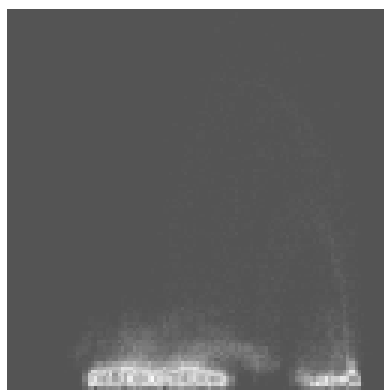
5.6 基于变换直方图选取阈值

5.6.3 灰度-梯度散射图

- 横轴是灰度值轴;
- 纵轴是梯度值轴;
- 统计值是同时具有某个灰度值和梯度值的像素个数;



(a)



(b)

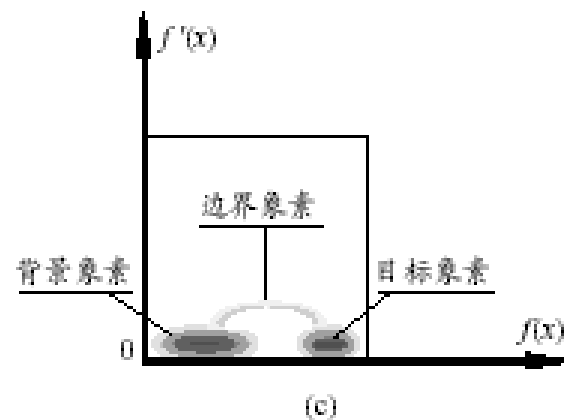
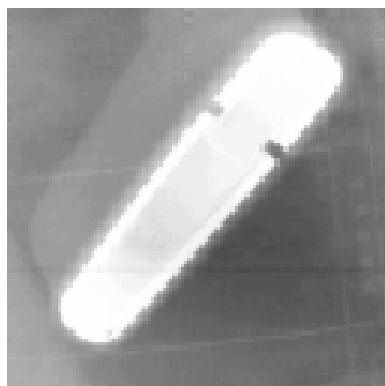


图 6.5.3 灰度和梯度散射图

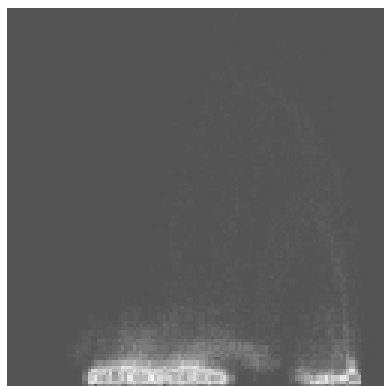
5.6 基于变换直方图选取阈值

5.6.3 灰度-梯度散射图

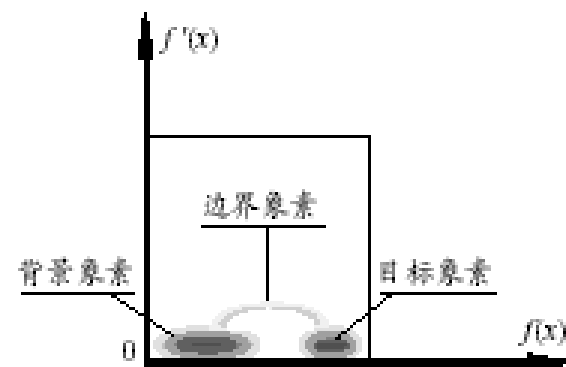
- 可结合使用灰度阈值和梯度阈值来把他们分开



(a)



(b)



(c)

图 6.5.3 灰度和梯度散射图

5.7 基于过渡区的阈值

5.7.1 过渡区的概念

- 边界是有宽度的；
- 边界将不同区域分割开来——边界；
- 面积不为零——区域。



先计算过渡区，再选取阈值进行分割。



5.7 基于过渡区的阈值

5.7.2 过渡区和有效平均梯度(EAG)

$$EAG = \frac{TG}{TP}$$

$$TG = \sum_{i,j \in Z} g(i, j)$$

$$TP = \sum_{i,j \in Z} p(i, j)$$

$$p(i, j) = \begin{cases} 1 & g(i, j) > 0 \\ 0 & g(i, j) = 0 \end{cases}$$

过渡区可借助对图像有效平均梯度的计算和对图像灰度的剪切操作来确定



5.7 基于过渡区的阈值

5.7.2 过渡区和有效平均梯度(EAG)

为了减少各种干扰的影响，定义以下特殊的剪切变换——把被剪切的部分设成剪切值。

$$\text{高端剪切} \quad f_{high}(i, j) = \begin{cases} L & f(i, j) \geq L \\ f(i, j) & f(i, j) < L \end{cases}$$

$$\text{低端剪切} \quad f_{low}(i, j) = \begin{cases} f(i, j) & f(i, j) > L \\ L & f(i, j) \leq L \end{cases}$$

对剪切后的图像求梯度，则其梯度函数必然与L有关： $EAG_{high}(L)$ 和 $EAG_{low}(L)$ 。

5.7 基于过渡区的阈值

5.7.3 EAG的极值点和过渡区边界

典型的 $EAG_{high}(L)$ 和 $EAG_{low}(L)$ 曲线都是单峰曲线。

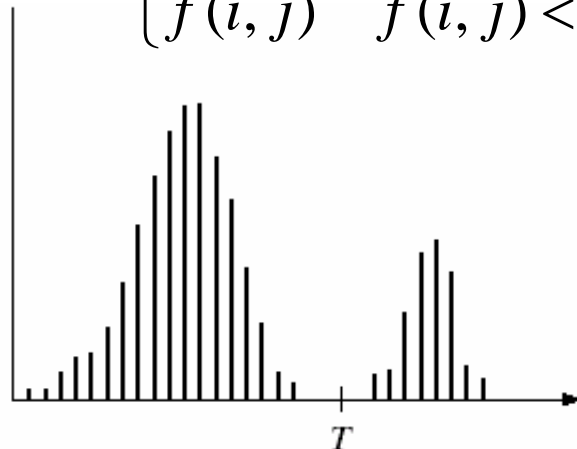
$$EAG_{low}(L) = \frac{TG_{low}(L)}{TP_{low}(L)}$$

$TG_{low}(L)$ 和 $TP_{low}(L)$ 都随着 L 的增加而减小

当 L 从0开始增加，
 $TP_{low}(L)$ 下降较快—>
 $TG_{low}(L)$ 下降较快

$$f_{low}(i, j) = \begin{cases} f(i, j) & f(i, j) > L \\ L & f(i, j) \leq L \end{cases}$$

$$f_{high}(i, j) = \begin{cases} L & f(i, j) \geq L \\ f(i, j) & f(i, j) < L \end{cases}$$



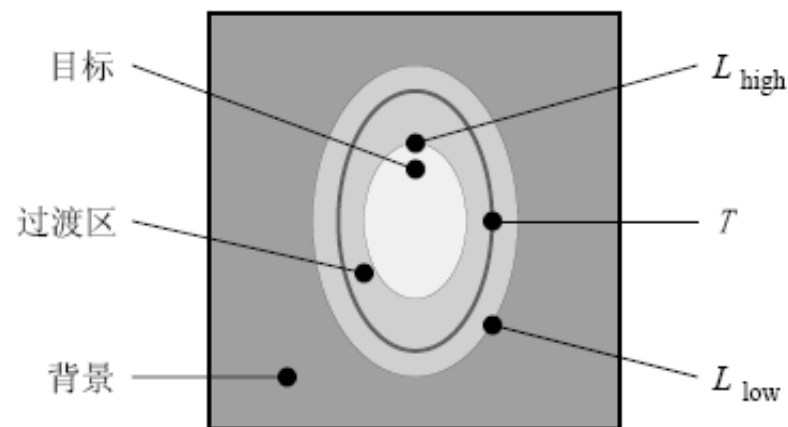
5.7 基于过渡区的阈值

5.7.3 EAG的极值点和过渡区边界

设 $EAG_{high}(L)$ 和 $EAG_{low}(L)$ 曲线的极值点分别为 L_{high} 和 L_{low} 。

$$L_{high} = \arg \left\{ \max_L [EAG_{high}(L)] \right\}$$

$$L_{low} = \arg \left\{ \max_L [EAG_{low}(L)] \right\}$$



过渡区示例

L_{high} 和 L_{low} 在灰度上限定了过渡区的范围。



5.7 基于过渡区的阈值

5.7.3 EAG的极值点和过渡区边界

极值点有3个重要性质：

- 对每个过渡区， L_{high} 和 L_{low} 总是存在并且各只存在一个；
- L_{high} 和 L_{low} 所对应的灰度值都具有明显的像素特性区别能力；
- 同一过渡区， L_{high} 不会比 L_{low} 小。



5.7 基于过渡区的阈值

5.7.4 阈值选取

- 边界处于过渡区中，可借助过渡区帮助选取阈值；
 - 过渡区包含的像素一般位于目标和背景区域之间。
- 过渡区像素的平均灰度值或过渡区内像素直方图的极值；
 - 也可确定多个阈值进行多阈值分割。



5.8 空间聚类

5.8.1 基本原理

- 将图像中的元素按照它们的特征值用对应的特征点表示；
- 将特征点聚集成对应不同区域的类团；
- 映射回原图像空间以得到分割的结果。



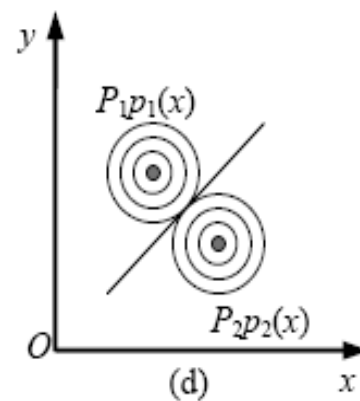
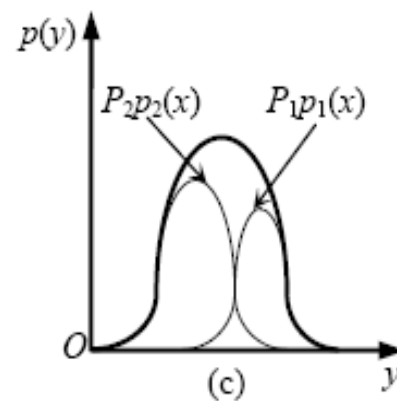
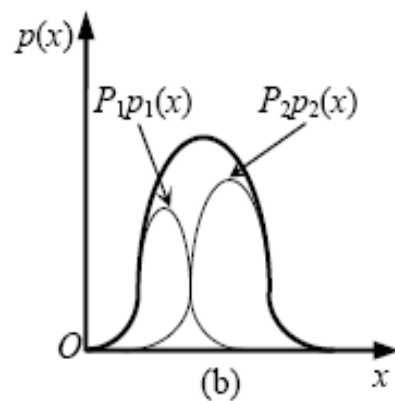
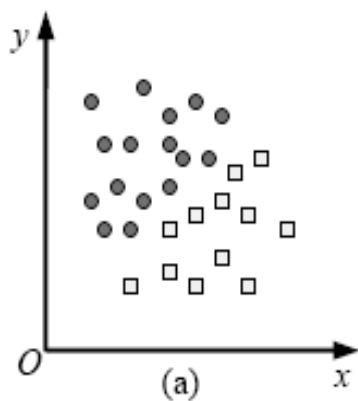
5.8 空间聚类

5.8.1 基本原理

- 空间聚类可看做是对阈值分割概念的推广；
在利用直方图的阈值分割中；
在利用灰度-梯度散射图分割的方法中；
- 空间聚类属于全局的方法，比基于边界的方法更抗噪；
- 但常会产生不连通的分割区域。

5.8 空间聚类

5.8.2 空间聚类举例





5.8 空间聚类

5.8.3 K -均值聚类

令 $x = (x_1, x_2)$ 代表一个特征空间的坐标, $g(x)$ 代表在该位置的特征值, K -均值法是要最小化如下指标:

$$E = \sum_{j=1}^K \sum_{x \in Q_j^{(i)}} \|g(x) - \mu_j^{(i+1)}\|^2$$

$Q_j^{(i)}$ 代表在第 i 次迭代后赋给类 j 的特征点集合, μ_j 表示第 j 类的均值:



5.8 空间聚类

5.8.3 K -均值聚类

- (1) 任意选 K 个初始类均值 $\mu_1^{(1)}, \mu_2^{(1)}, \dots, \mu_K^{(1)}$,
- (2) 特征点赋类

$$\mathbf{x} \in Q_l^{(i)} \quad \text{如果} \quad \left\| g(\mathbf{x}) - \mu_l^{(i)} \right\| < \left\| g(\mathbf{x}) - \mu_j^{(i)} \right\|$$

- (3) 更新类均值

$$\mu_j^{(i+1)} = \frac{1}{N_j} \sum_{\mathbf{x} \in Q_j^{(i)}} g(\mathbf{x})$$

- (4) 判断算法收敛



5.9 串行区域分割

5.9.1 串行区域分割技术

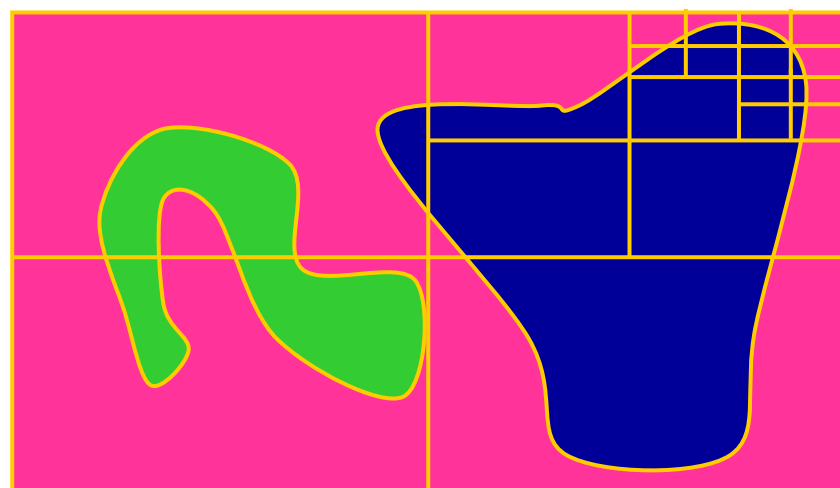
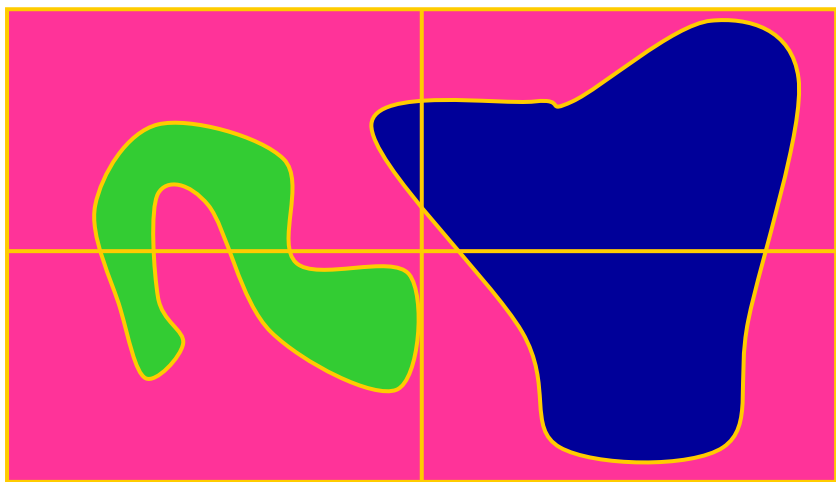
- 定义：
- 两种基本形式：一种是从全图出发，逐渐分割至所需的分割区域，例如分裂合并方法；另一种是从单个像素出发，逐渐合并以形成所需的分割区域，例如区域生长方法。

5.9 串行区域分割

5.9.2 分裂合并法

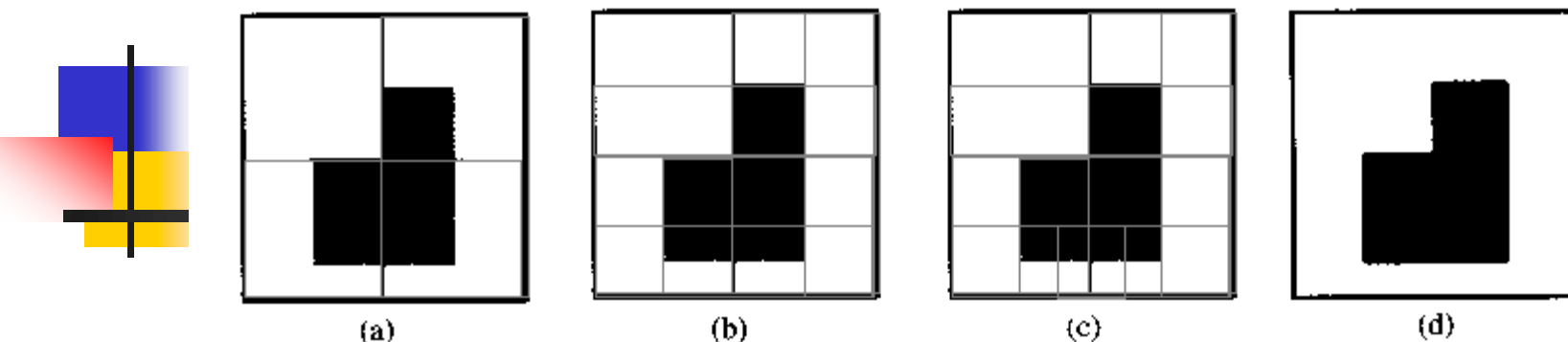
算法实现：

- 1) 对图像中灰度级不同（不相似）的区域，均分为四个子区域；
- 2) 如果相邻的子区域所有像素的灰度级相同（相似），则将其合并；
- 3) 反复进行上两步操作，直至不再有新的分裂与合并为止。



5.9 串行区域分割

5.9.2 分裂合并法



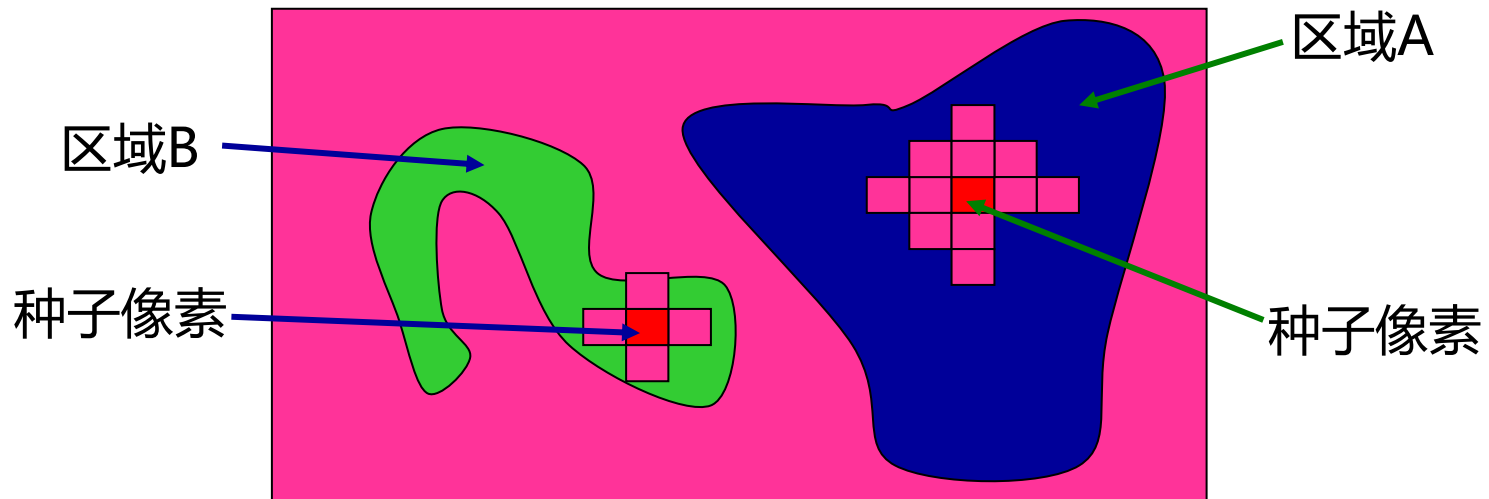
分裂合并法分割图象图解

我们可总结前面所述的基本分裂合并算法步骤如下：

- (1) 对任 1 个区域 R_i ，如果 $P(R_i) = \text{FALSE}$ 就将其分裂成不重叠的 4 等分；
- (2) 对相邻的 2 个区域 R_i 和 R_j ，如果 $P(R_i \cup R_j) = \text{TRUE}$ ，就将它们合并起来；
- (3) 如果进一步的分裂或合并都不可能了，则结束。

5.9 串行区域分割

5.9.3 区域生长



5.9 串行区域分割



5.9.3 基本思想

- 区域生长的基本思想是将具有相似性质的像素集合起来构成区域。

5.9 串行区域分割

5.9.3 基本步骤

- 对每个需要分割的区域找一个种子像素作为生长的起点；
- 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。将这些新像素当作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。

5.9 串行区域分割

5.9.3 需要解决的问题

- 选择或确定一组能正确代表所需区域的种子像素；
- 确定在生长过程中能将相邻像素包括进来的准则；
- 制定让生长过程停止的条件或规则。

5.9 串行区域分割

5.9.3 基于区域灰度差的区域生长方法

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

原始图

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

$T = 3$

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

$T = 2$

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

$T = 6$

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

对区域生长起点的选择有较大依赖型

5.9 串行区域分割

5.9.3 改进的区域生长方法

(1) 设灰度差的阈值为零，用上述方法进行区域扩张，合并灰度相同的像素；

(2) 求出所有邻接区域之间的平均灰度差，合并具有最小灰度差的邻接区域；

(3) 设定终止准则，反复进行步骤（2）中的操作，将区域依次合并直到终止条件满足为止。

当图像中存在缓慢变化的区域时，该方法有可能会将不同区域逐步合并而产生错误。



小节

5.1 图像分割定义和方法分类

5.2 边缘检测

5.3 轮廓跟踪和图搜索

5.4 Hough变换

5.5 阈值分割

5.6 基于变换直方图选取阈值

5.7 基于过渡区的阈值

5.8 空间聚类

5.9 串行区域分割