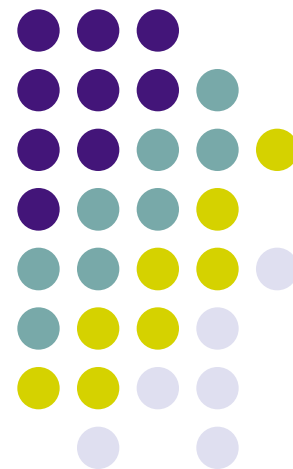
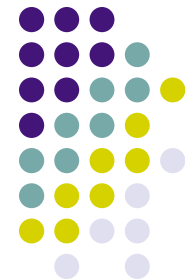


# 数字图像处理

## 第十三讲 图像分割



# 图像分割



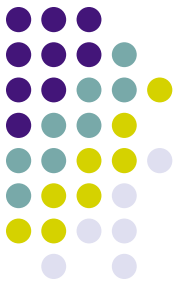
- 序言
- 区域分割技术
- 梯度分割技术
- 边界跟踪和区域生长
- 要点总结

# 序言



- 问题：之前的图像处理都假定输入和输出都是图像。然而在有些应用中，输出不要求是个图像，它只需要输出图像的属性、特征、模式等进行处理判断
  - 这方面的内容也被称为**图像分析**
- **图像分割**是图像分析方面的一个主要步骤

# 图像分割



- 图像分割：将图像细分为构成它的子区域或对象
- 分割的程度取决于要解决的问题
  - 当感兴趣的对象被分离出来时，就停止分割
  - 更多的细节继续分割没有意义
- 一般来讲，图像分割是问题相关的

# 图像分割



- 问题：图像分割依靠什么特征？
- 两个基本特征
  - 不连续性：边缘、边界；
  - 相似性：分割图像具有相似的区域，例如圆形，方形等；

# 分割算法分类



- 问题：如何进行算法分类？
- 基本原则：
  - (1) 每个算法都能归到某一类；
  - (2) 所有算法都能包含在各类中；
  - (3) 同一类中的算法具有某些相同的性质；
  - (4) 不同类中的算法具有某些不同的性质；
  - (5) 类别划分具有不可替代性；

# 分割算法分类



- 主要有两类
  - 区域分割技术：各像素归到不同物体或区域中；
  - 梯度分割技术：确定区域间的边界/边缘，并把像素连接在一起构成所需边界或区域；



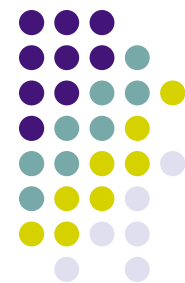
- 图像分割目前在研究的问题
  - (1) 针对应用场景，引入合适的假设，研究相应的分割算法；
  - (2) 根据分割结果的评价准则，对各种分割算法的性能进行刻画和比较；
  - (3) 对分割评价准则的研究

**图像分割: IMAGE SEGMENTATION**

**边缘检测: EDGE DETECTION**

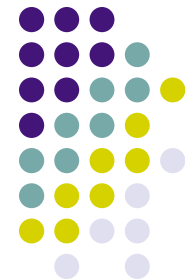


# 要点



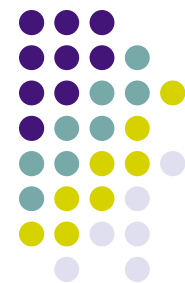
- 什么是图像分割？
- 图像分割是否有最好的算法？为什么？
- 图像可以分割主要依据图像什么特征？
- 图像分割主要分为哪两类？
- 目前图像分割的前沿问题

# 图像分割



- 序言
- 区域分割技术
- 梯度分割技术
- 边界跟踪和区域生长
- 要点总结

# 区域分割技术



- 问题：如何划分区域？
- 解法：阈值化分割
- 两个基本步骤
  - 1) 确定需要的分割阈值（算法的关键）
  - 2) 将像素值与分割阈值相比较以划分像素

# 区域分割技术



- 问题：如何分割前景与背景？
- 阈值化分割
  - 比如：大于某阈值为前景，小于某阈值为背景
- 关键问题：阈值如何确定？

# 区域分割技术



- 主要有如下做法：
  - 基于各像素值的阈值
  - 基于区域性质的阈值（略）
  - 基于坐标位置的阈值
  - 基于目标形状的阈值

# 区域分割技术



- 这类方法对图像的基本假设
  - 假设图像是由具有单峰灰度分布的目标和背景组成。
  - 在单阈值情况下：

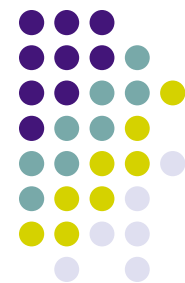
$$g(x, y) = \begin{cases} 1 & \text{如 } f(x, y) > T \\ 0 & \text{如 } f(x, y) \leq T \end{cases}$$

# 区域分割技术



- 阈值化算法很多，比如
  - 根据分割过程是否需要人工干预，分为交互的与自动的
  - 根据阈值的不同作用范围，分为全局的与局部的
  - 根据阈值与灰度分布的关系，分为基于灰度分布的一阶统计和基于灰度分布的二阶统计
  - 根据算法的处理策略，分为迭代的与非迭代的
  - 根据算法是否需要分割估计，分为有监督的和无监督的

# 区域分割技术



- 从阈值确定的因素来看，阈值算法的因素主要包括如下方面：

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

其中 $f(x, y)$ 代表 $(x, y)$ 点的灰度值；

$p(x, y)$ 代表 $(x, y)$ 点邻域的某种局部性质；

$s(x, y)$ 代表 $(x, y)$ 点所属目标形状。



# 区域分割技术



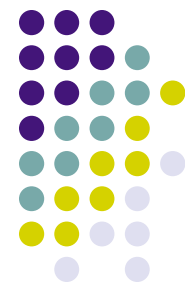
- (1) 基于各像素值的阈值。阈值根据  $f(x,y)$  来选取；
- (2) 基于区域性质的阈值。阈值根据  $f(x,y)$  和  $p(x,y)$  来选取；
- (3) 基于坐标位置的阈值。阈值根据  $f(x,y)$ 、 $p(x,y)$  和  $x,y$  来选取；
- (4) 基于目标形状的阈值。阈值根据  $f(x,y)$ 、 $x,y$  和  $s(x,y)$  来选取。

# 基于各像素值的阈值



- 主要有两类技术
  - (1) 极小值点阈值
  - (2) 最优阈值

# 极小值点阈值



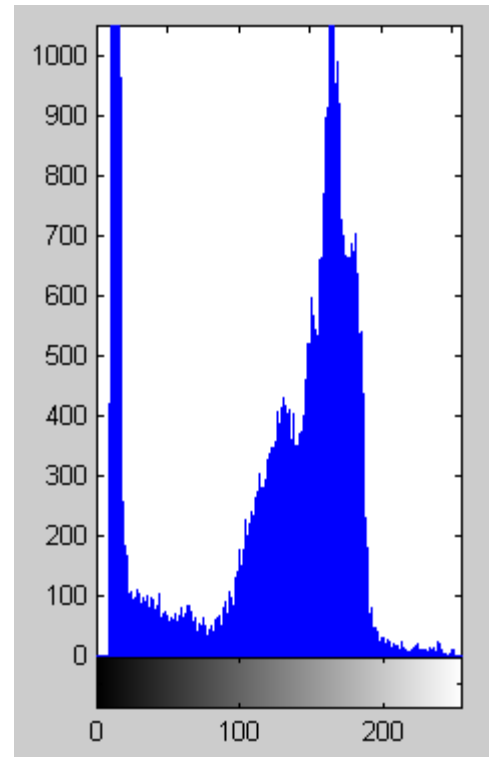
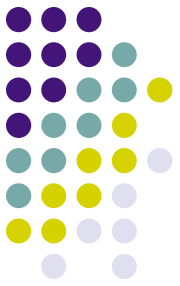
- 思想：将直方图包络曲线的极小值作为阈值

$$\frac{\partial h(d)}{\partial d} = 0 \text{ 和 } \frac{\partial^2 h(d)}{\partial d^2} > 0$$

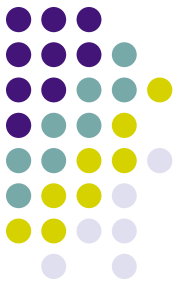
$h(d)$ 是图像的直方图

- 如果前景背景的灰度级都比较集中，且两者灰度级有明显的差距，这种方法比较适用

# 极小值点阈值



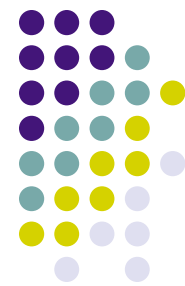
# 极小值点阈值



效果图



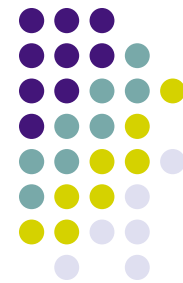
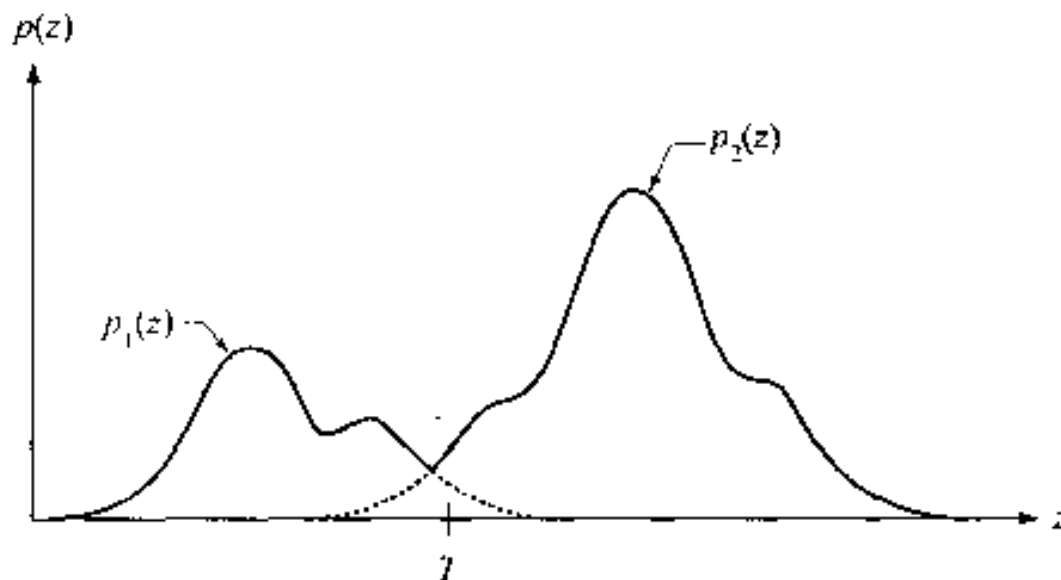
# 最优阈值



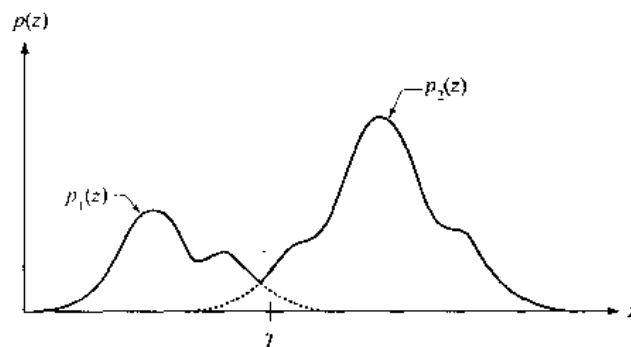
- 问题：之前极小值点阈值假定图像前景与背景的灰度没有交叠。若图像与背景的灰度有部分交错，这时采用阈值化分割进行分割会产生一定的误差。
- 方法：尽可能减少误分割

# 最优阈值

- 假设背景和目标的灰度符合某种分布



# 最优阈值



- 第一类错误

$$E_1(T) = \int_{-\infty}^T p_2(z) dz$$

- 第二类错误

$$E_2(T) = \int_T^{\infty} p_1(z) dz$$

- 总错误

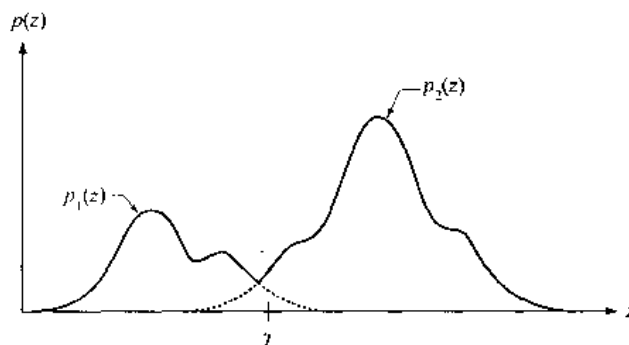
$$E(T) = P_2 E_1(T) + P_1 E_2(T)$$

- 其中

$$P_1 + P_2 = 1$$



# 最优阈值



$$E(T) = P_2 E_1(T) + P_1 E_2(T)$$

- 求解：对 $E(T)$ 求导，最优解满足等式：

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

# 最优阈值



- 特别的，假定前景和背景都符合高斯分布

$$p(z) = \frac{P_1}{\sqrt{2\pi}\sigma_1} e^{-\frac{(z-\mu_1)^2}{2\sigma_1^2}} + \frac{P_2}{\sqrt{2\pi}\sigma_2} e^{-\frac{(z-\mu_2)^2}{2\sigma_2^2}}$$

- 假定方差相等

$$\sigma^2 = \sigma_1^2 = \sigma_2^2$$

# 最优阈值



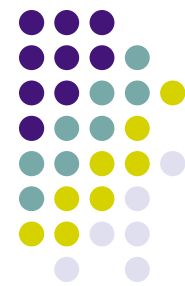
- 根据最优阈值的条件：

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

- 求解可得：

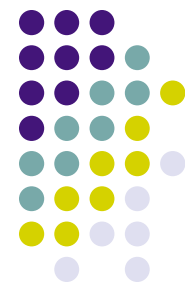
$$T = \frac{\mu_1 + \mu_2}{2} + \frac{\sigma^2}{\mu_1 - \mu_2} \ln\left(\frac{P_2}{P_1}\right)$$

# 基于区域性质的阈值（略）



- 思想：不仅考虑了各像素本身灰度值而且利用了各像素邻域内像素灰度关系；
- 典型的算法有
  - 双阈值法：先用一个全局阈值进行全局分割，然后用一个局部阈值对每一个局部结果进行精化
  - 直方图变换：通过区域划分将一个直方图转变为多个直方图分别进行分割；

# 基于坐标位置的阈值



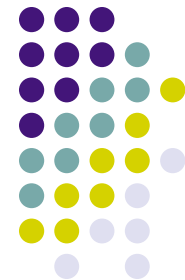
- 问题：之前假定用一个全局阈值对图像进行分割，阈值的选择与坐标位置无关。然而如果图像存在光照不均匀等因素，不能使用一个固定的全局阈值。
- 基于坐标位置的阈值：阈值是坐标的函数。
  - 通常又称为动态阈值或自适应阈值法。

# 基于坐标位置的阈值： 举例



左下方的阈值与全局的阈值不同

# 基于坐标位置的阈值



- 介绍两种算法
  - 阈值插值法
  - 分水岭算法

# 基于坐标位置的阈值



- 阈值插值法

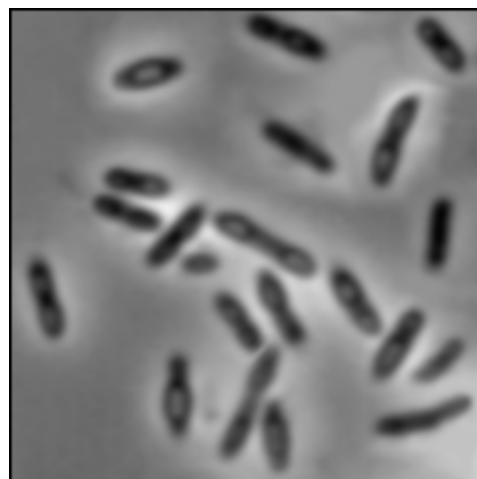
- Step1:将整幅图像分成一系列互相有50%重叠的子图像;
- Step2:得到每个子图像的直方图;
- Step3:检测各子图像是否为双峰, 若是则采用最优阈值法, 否则不进行处理;
- Step4:根据对直方图为双峰的子图像得到的阈值, 通过插值得到所有子图像的插值;
- Step5:根据各子图像的阈值, 然后对图像进行分割。



# 基于坐标位置的阈值



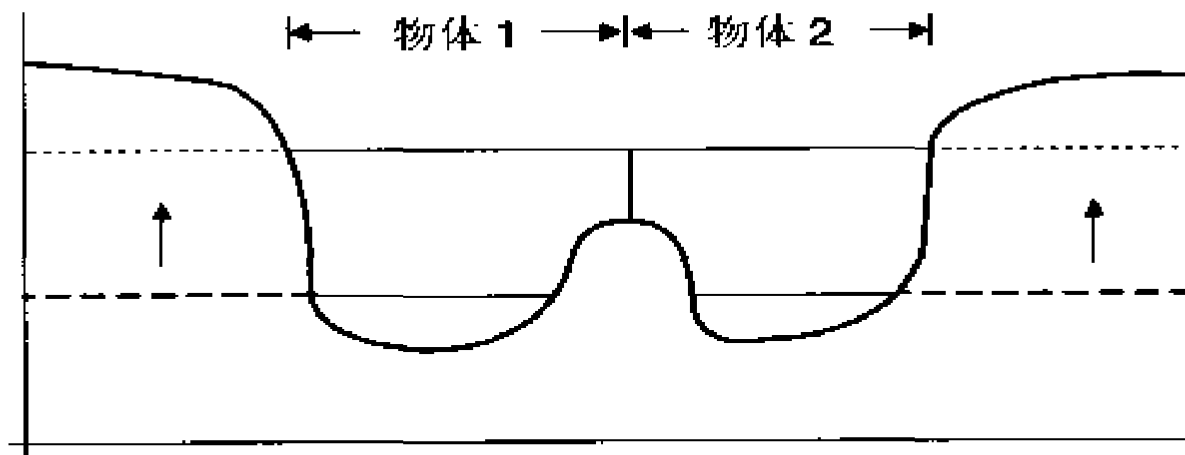
- 水线阈值算法（分水岭算法）
  - Watershed算法是一种特殊的自适应阈值迭代算法。
  - 目的：将两个物体目标从背景中提取出来并互相分开。



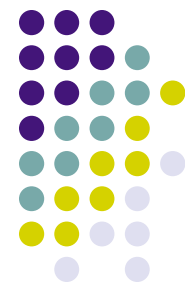
# 基于坐标位置的阈值



- 水线阈值算法（分水岭算法）



# 分水岭算法



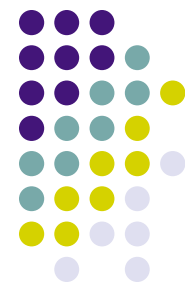
- Step1:在一个低灰度级上阈值进行二值化，将目标正确的分离出，但边界偏向物体内部；
- Step2:每次将阈值增加一个灰度级，物体的边界随之向外扩展；
- Step3:重复Step2，直至两个物体的边界相互接触。此时可确定物体的边界。

# 基于目标形状的阈值



- 问题：在许多重要的情况下，需要寻找某一形状的物体。
- 形状物体阈值确定的4种方法
  - 在直方图的局部极小值选取阈值 $T$ ；
  - 依据 $h$ -等价在轮廓函数中的转折点选取 $T$ ；
  - 依据 $p$ -等价在轮廓函数中的转折点选取 $T$ ；
  - 选择使平均边界梯度最大的 $T$ ；

# 基于目标形状的阈值



- P-等价：记一个点的周长函数为 $P(T)$ 。若两个点具有相同的 $P(T)$ ，称为这两个点是p-等价的；
- h-等价：记一个点的阈值面积函数为 $A(T)$ 。若两个点具有相同的 $A(T)$ ，称为这两个点是h-等价的。

# 基于目标形状的阈值



- 直方图和轮廓

圆形目标面积函数的导数为

$$\frac{\Delta A}{\Delta r} = \frac{\pi(r + \Delta r)^2 - \pi r^2}{\Delta r} = 2\pi r$$

物体直方图可由面积函数的导数表示为

$$H(D) = \lim_{D \rightarrow 0} \frac{\Delta A}{\Delta D} = \frac{dA/dr}{dD/dr} = \frac{2\pi r}{\frac{dB_p(r)}{dr}}$$

# 基于目标形状的阈值



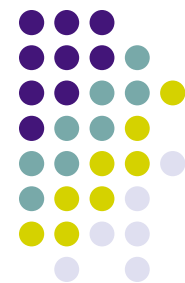
- 由面积函数导出轮廓函数

$$R(T) = \left[ \frac{1}{\pi} A(T) \right]^{\frac{1}{2}} = \left[ \frac{1}{\pi} \int_T^{\infty} H(D) dD \right]^{\frac{1}{2}}$$

- 由周长函数导出的轮廓函数

$$R(T) = \frac{1}{2\pi} P(T)$$

# 基于目标形状的阈值



- 平均边界梯度
  - 原理：面积可以由周长乘以径长逼近。

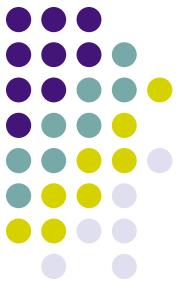
不规则物体边界梯度定义为

$$|\nabla B| = \lim_{\Delta D \rightarrow 0} \frac{\Delta D}{\Delta r} = \lim_{\Delta D \rightarrow 0} \frac{\Delta D}{\frac{\Delta A}{p(D)}} = \frac{p(D)}{H(D)}$$

平均边界梯度等于周长函数和直方图的比值

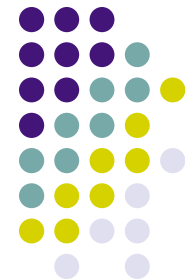


# 要点



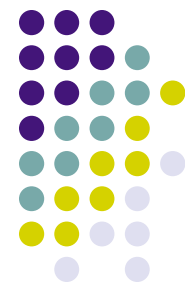
- 区域划分一类主流技术是阈值化
- 根据阈值确定因素的不同，阈值化主要有哪些方法？
- 极小值阈值化的原理和不足
- 熟悉最优阈值的原理和基本计算
- 熟悉基于坐标的阈值化方法

# 图像分割



- 序言
- 区域分割技术
- 梯度分割技术
- 边界跟踪和区域生长
- 要点总结

# 梯度分割技术



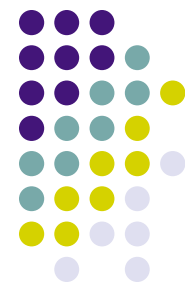
- 基本想法：通过寻找边界进行图像分割。
- 四大类：
  - 1) 微分算子边缘检测
  - 2) 边界闭合
  - 3) 边缘拟合
  - 4) Hough变换

# 微分算子边缘检测



- 梯度算子
  - 梯度算子
  - 拉普拉斯算子
  - Marr算子
  - Canny算子
- 参见边缘检测内容

# 边界闭合



- 问题：在有噪声时，边缘算子得到的边缘往往是孤立的或分段不连续的。且有些是真正的边界像素，有些是噪声点。
- 目的：为使图像中不同区域分开，需要将边缘像素连接起来组成区域的封闭边界。
- 方法：
  - 1) 利用边缘像素梯度的相似性；
  - 2) 利用形态学的闭运算。（参见Ch8）

# 边界闭合



## ● 利用边缘像素梯度的相似性

*Step1*: 用梯度算子得到梯度幅度和梯度方向;

*Step2*: 若像素 $(s, t)$ 在像素 $(x, y)$ 的邻域内, 且满足

$$|\nabla f(x, y) - \nabla f(s, t)| \leq T$$

*Step3*: 则将像素 $(s, t)$ 和像素 $(x, y)$ 连接起来。

**注意:** 当邻域较小时, 则不是所有边界都能闭合。此方法对复杂图像效果较差。

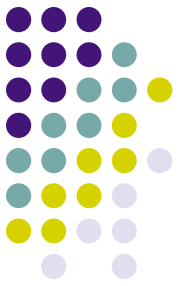


# 边缘拟合



- 问题：若边缘点很稀疏，则需要用某个解析函数如分段线性或高阶样条曲线来拟合边缘。这称为边缘拟合。
- 方法：常根据图像一小块区域建立拟合模型。
- 常用方法有：
  - (1) 迭代端点拟合的分段线性方法；
  - (2) 灰度渐变边缘模型；
  - (3) 灰度阶跃边缘模型等

# 迭代端点拟合 (iterative endpoint fitting)



- 方法步骤:

*Step1*: 确定一个散布在两个特定边缘点 $A$ 和 $B$ 之间的边缘点集合 $P$ ;

*Step2*: 首先从 $A$ 到 $B$ 引一直线, 然后计算其他边缘点到该直线的垂直距离;

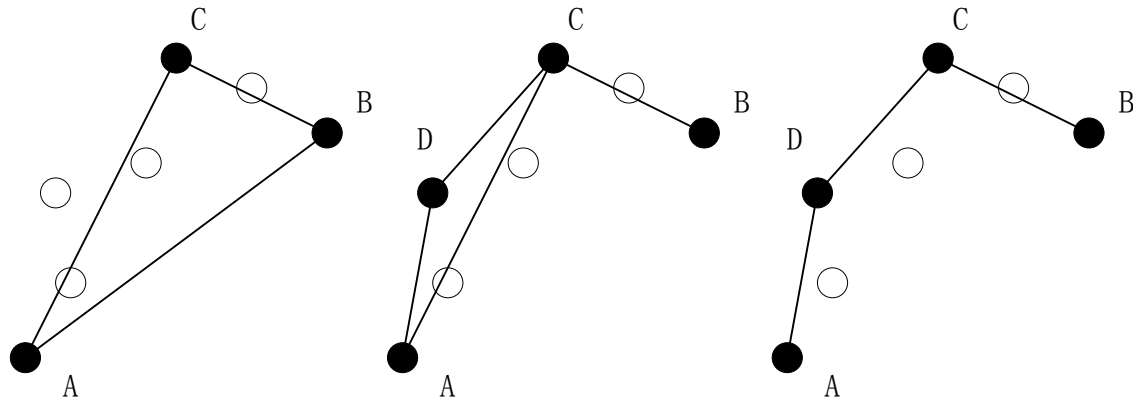
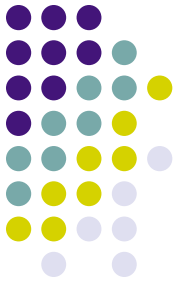
*Step3*: 将距离最大的点标记为 $C$ , 然后将 $P$ 分为包含 $A$ 和 $C$ 之间的边缘点集合 $P_1$ 和包含 $C$ 和 $B$ 之间的边缘点集合 $P_2$ ;

*Step4*: 重复*Step2*和*Step3*, 直到边缘点集合 $P_i$ 的点到线距离小于某固定距离。

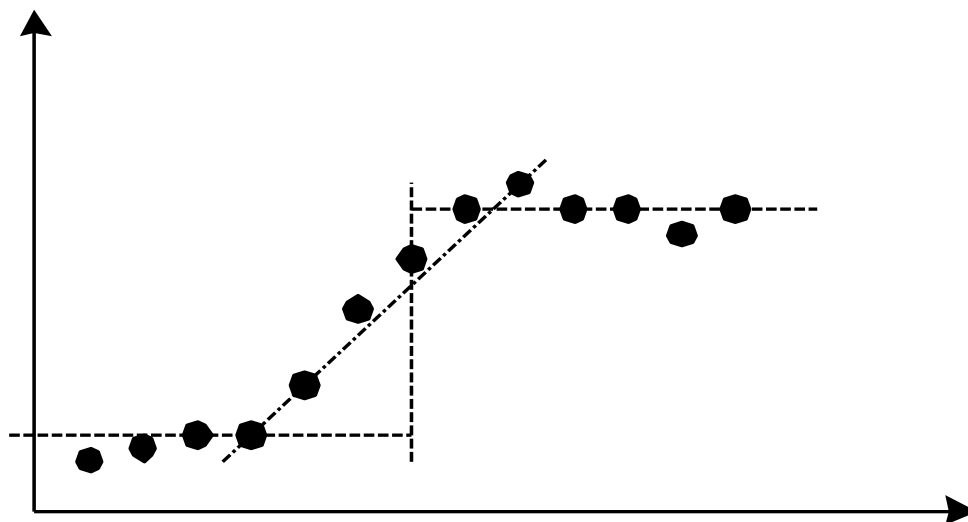
*Step5*: 对每个边缘点集合采用直线拟合。



# 迭代端点拟合



# 灰度渐变边缘模型



# 灰度渐变边缘模型



Step1: 将一幅图象表示成一系列相邻但不相交的子区域集合 $Q$ ;

Step2: 对每一个区域建立一个斜面方程模型:

$$H(x, y) = ax + by + c$$

Step3: 用 $H(x, y)$ 拟合 $g(x, y)$ , 则最小平方拟合误差为

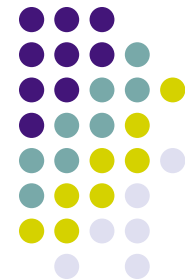
$$E = \sum_Q [ax + by + c - g(x, y)]^2$$

Step4: 如果 $Q$ 是 $3 \times 3$ 的子图象, 则拟合即Prewitt模板

$$a = \frac{1}{2} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \otimes g(x, y) \quad b = \frac{1}{2} \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \otimes g(x, y)$$

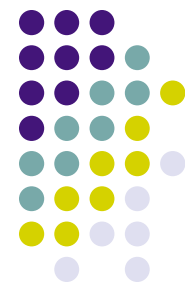
$$c = \frac{1}{9} \sum_{3 \times 3} g(x, y)$$

# 灰度渐变边缘模型



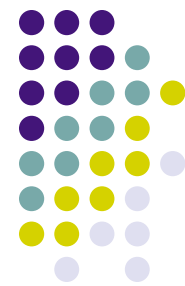
*Step5*: 将一幅图象表示成一系列斜面集合后, 图象的边缘即不同斜面的交界。则边缘问题转换为如何确定不同斜面及求它们交线的问题。

# Hough变换



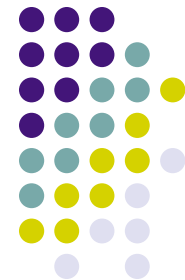
- 问题：检测图像中某些给定形状的曲线并用参数方程描绘，并希望较少受曲线中断点的影响。
- 思想：Hough变换是对图像进行坐标变换，使之在另一个坐标空间的特定位置出现峰值
  - 原理：点-线的对偶性 (duality)

# Hough变换



- 更具体的问题描述：给定 $n$ 个点，找到这些点位于多少条直线上，每条直线上都有哪些点？
- 直接做法：两两连线确定直线的参数后，对结果加以综合来得到
- 时间复杂度： $O(N^2)$ 条直线，每个点跟每条直线比对， $O(N^3)$ 复杂度

# Hough变换



*Step1*: 图象空间 $XY$ 中, 所有过点 $(x, y)$ 的直线满足

$$y = px + q$$

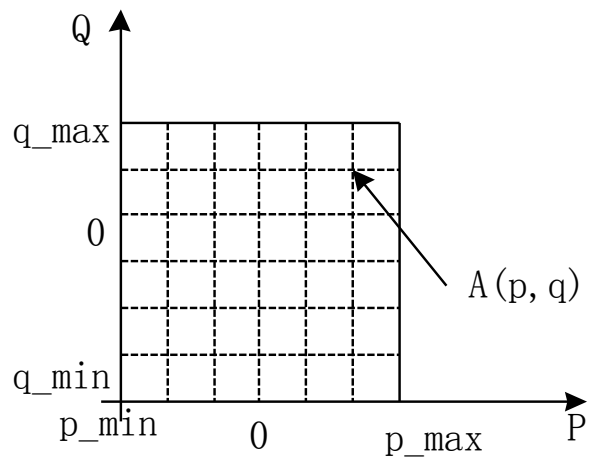
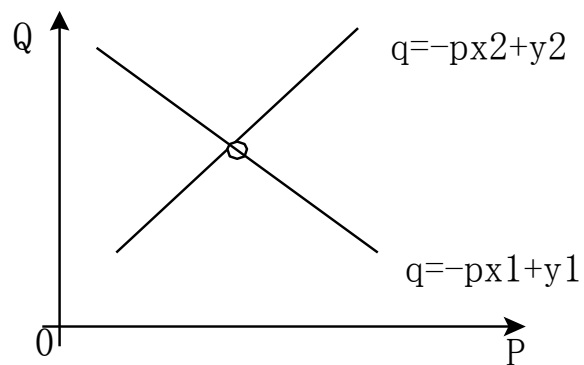
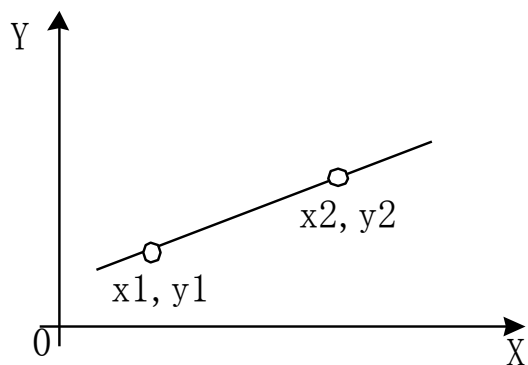
*Step2*: 这个通用的直线方程, 转换到参数空间即

$$q = -px + y$$

*Step3*: 图象空间 $XY$ 中, 过点 $(x_1, y_1)$ 和 $(x_2, y_2)$ 的直线转换到参数空间必相交在点 $(p', q')$ 上。

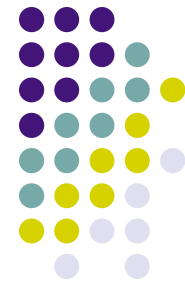
因此图象空间上共线的点必对应应在参数空间里相交的线。

# Hough变换





# Hough变换



## ● 计算方法

*Step1*: 在参数空间建立一个二维数组 $A(p, q)$ , 初始化为0

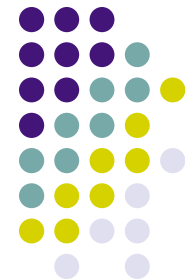
其中 $p \in [p_{\min}, p_{\max}]$ ,  $q \in [q_{\min}, q_{\max}]$ ;

*Step2*: 对于图象中每一个边缘点, 让 $p$ 遍取 $P$ 轴所有值, 则相应得到每一个 $q$ 值, 令 $A(p, q) = A(p, q) + 1$ ;

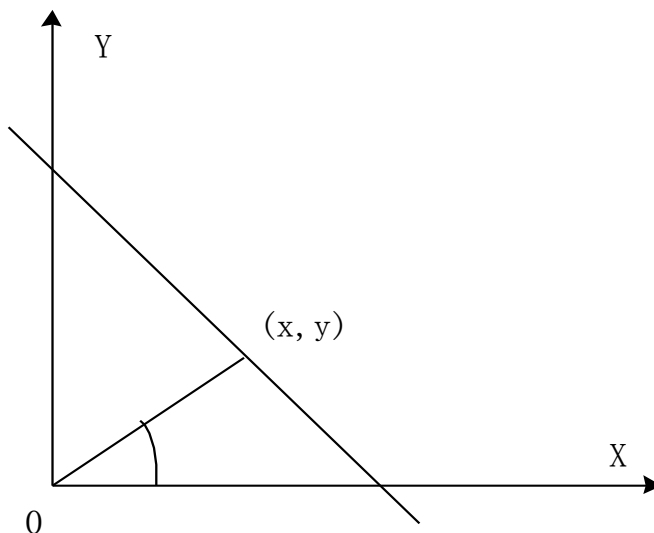
*Step3*: 对于图象中每一个边缘点, 重复*Step2*;

*Step4*: 根据 $A(p, q)$ 即知道有多少点共线, 且 $(p, q)$ 给出直线参数。

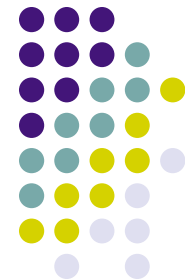
# Hough变换的扩展



- 问题：在直线斜率接近垂直时，累加数组变得很大，效率低下。
- 解决方法：极坐标法



# Hough变换的扩展



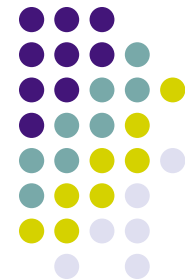
*Step1*:直线的极坐标方程为, 为参数空间 $\Lambda\Theta$ 的正弦曲线

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

*Step2*:则原来的点-直线对偶转换成点-正弦曲线对偶性;

*Step3*:余下步骤同基本计算方法, 唯一区别是 $\Lambda\Theta$ 取值是有限区间。

# 要点



- 熟悉梯度分割技术的基本原理及主流算法
- 熟悉Hough变换

# 图像分割



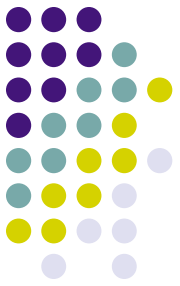
- 序言
- 区域分割技术
- 梯度分割技术
- 边界跟踪和区域生长
- 要点总结

# 边界跟踪



- 问题：要求目标轮廓边界细、连续无间断、准确。
- 边界跟踪（boundary tracking）或边缘点连接（edge point linking）的思想是：由图像梯度出发，依次搜索并连接相邻边缘点从而逐步检测出边界。

# 边界跟踪的三个基本步骤



- 基本步骤：
  - 1.确定作为搜索起点的边缘点；
  - 2.确定和采用一种合适的搜索机理，在已发现的边界点基础上确定新的边界点；
  - 3.确定搜索终结的准则或终止条件（如边界闭合或回到起点）。则满足条件时，结束搜索。

# 边界跟踪



- 直接实现

*Step1*: 给定一幅只有一个目标的图象，先计算其梯度图；

*Step2*: 从梯度图中选取梯度最大的点作为边界跟踪的第一个起点 $P$ ；

*Step3*: 在 $P$ 的8-邻域中选梯度最大的点作为第二个边界点 $C$ ；

*Step4*: 以 $C$ 为中心，根据 $P$ 的位置，在阴影中选下一个边界点 $C'$ ；

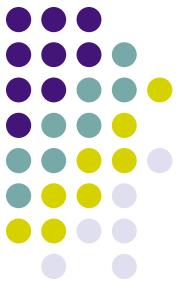
*Step5*: 以 $C'$ 为中心，根据 $C$ 的位置，重复*Step4*；

*Step6*: 直至满足终止条件。

- 此方法适用于噪声较小图像



# 边界跟踪



P	C	

	C	
P		

	C	
	P	

	C	
		P

	C	P

		P
	C	

	P	
	C	

P		
	C	

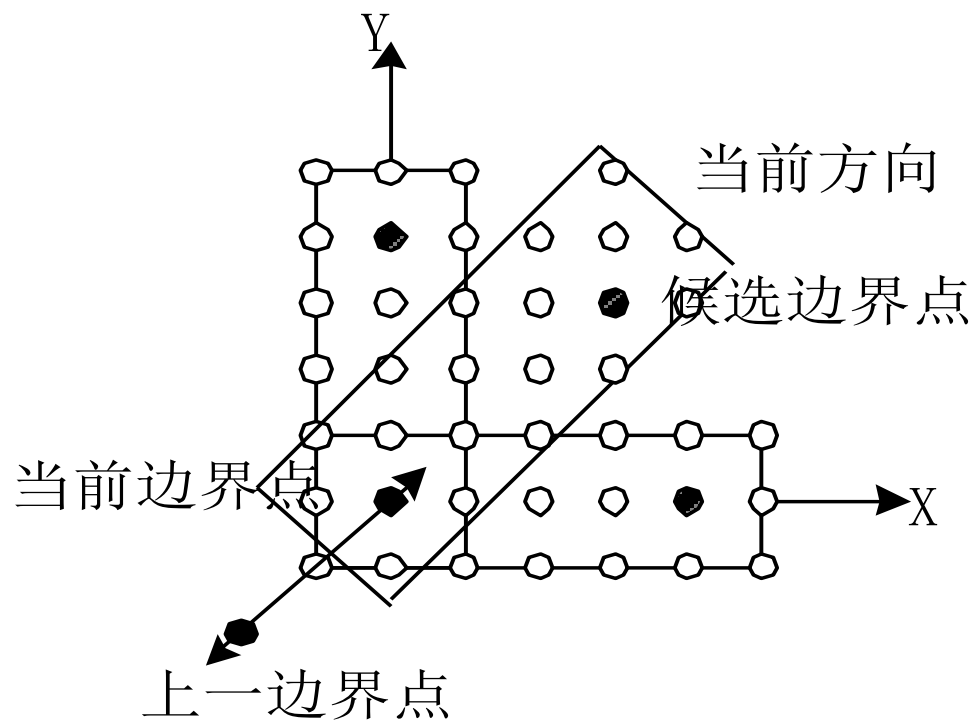
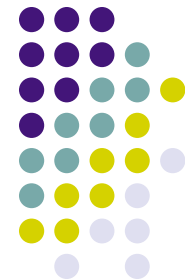
# 跟踪虫技术 (Tracking bug)



- Step1*: 定义虫，通常是一个矩形形状的具有相同权值的窗口；
- Step2*: 以当前边界点作为窗口的轴心；
- Step3*: 以上一边界点和当前边界点确定当前边界方向；
- Step4*: 窗口在当前边界方向左右  $-\theta$  :  $+\theta$  角度内旋转，在每一个窗口内确定平均梯度及对应梯度点；
- Step5*: 确定具有最高平均梯度的窗口，对应梯度点作为下一边界；
- Step6*: 重复 *Step5*，直到终止条件满足。

窗口越大，对梯度的平滑作用越强，也越抗噪声。

# 跟踪虫技术 (Tracking bug)



# 区域生长



- 问题：如何从单个像素出发，逐渐合并像素点以形成所需的分割区域；
- 思想：将具有相似性质的像素集合起来构成区域。
- 与边界跟踪不同，边界跟踪基于梯度得到的是边界；而区域生长基于灰度得到的是区域。

# 区域生长



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

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

T=3

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

T=1

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

T=6

# 区域生长



- 区域生长应解决三个问题：
  - (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够将相邻像素包括进来的准则；
  - (3) 确定在生长过程中停止的条件或准则。

# 区域生长



- 确定种子像素
  - 可以将灰度呈现聚类的情况，则将聚类中心的像素取为种子像素。
- 基于区域灰度差的生长准则
  - Step1*: 对图象进行逐行扫描，找出尚没有归属的像素；
  - Step2*: 以该像素为中心检查它邻域，如果灰度差小于预先确定的阈值，则将它合并；
  - Step3*: 以新合并的像素为中心，重复*Step2*；
  - Step4*: 重复*Step1*，直到所有行。

# 区域生长



- 基于区域内灰度分布统计性质的生长准则

*Step1*: 将图象分成互不重叠的小区域;

*Step2*: 比较相邻区域的灰度累积分布函数 $CDF$ ,  
根据灰度分布的相似性进行区域合并;

*Step3*: 直到终止准则满足。

- 基于区域形状的生长准则



# 区域生长



*Method1*: 将图象分割成灰度基本固定的区域,  
设两邻接区域周长分别为 $P_1$ 和 $P_2$ , 且共同  
边界上两侧灰度小于阈值的长度设为 $L$ ,  
若满足以下准则, 则合并区域。

$$\frac{L}{\min(P_1, P_2)} > T$$

*Method2*: 将图象分割成灰度基本固定的区域,  
设两邻接区域共同边界长度为 $A$ , 且共同  
边界上两侧灰度小于阈值的长度设为 $L$ ,  
若满足以下准则, 则合并区域。

$$\frac{L}{A} > T$$

# 要点总结



- 图像阈值化分割的统一模型和分类；
- 了解和掌握几种常用的阈值化分割算法；
- 了解什么是边界闭合，其主要解决方法是什么；
- 了解什么是边缘拟合，其主要解决方法是什么；
- 了解Hough变换的思想，计算方法和极坐标方法；
- 了解边界跟踪思想，和跟踪虫技术；
- 了解边缘生长思想，和主要生长准则。