图像分割

数字图像处理第8讲

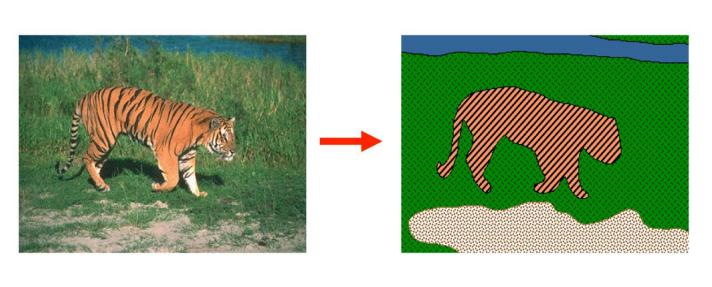
内容

- 图像分割概述
- 基于阈值的方法
- 基于边缘的方法
- 基于区域的方法

什么是图像分割

- 将图像中分割成不相交的区域
- 区域具有某种意义上的一致性
 - 颜色、亮度、纹理、内容(语义) ...





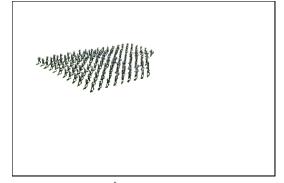


图像理解的一个前期步骤

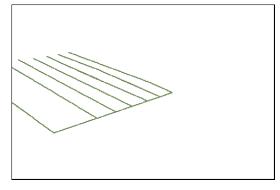
• 分割的目的是让图像更容易被分析和理解



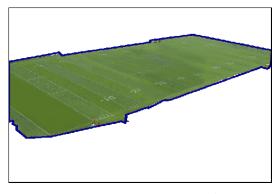
input image



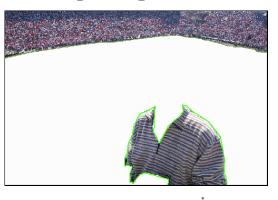
point process



curve process



a color region



texture regions



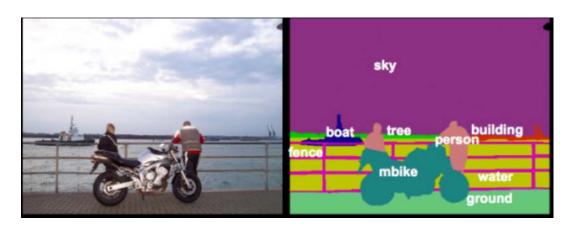
objects

怎样才是一个好的分割

- 达到语义级
- Semantic Object

- 分割依赖于低层视觉,同时又 依赖于高层视觉
- 从一般意义上来说,只有对图像内容的彻底理解,才能产生完美的分割(分割VS理解)



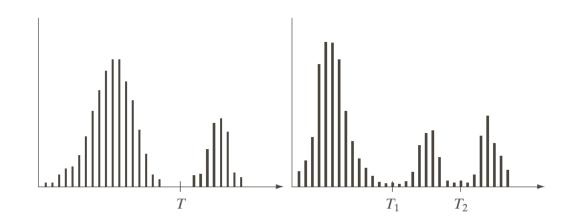


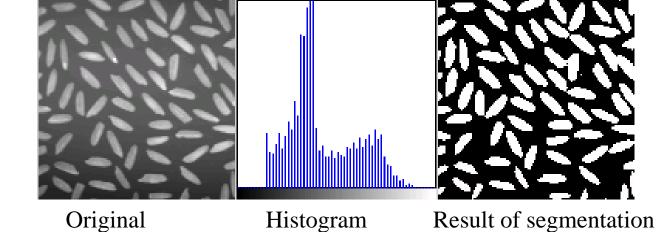
内容

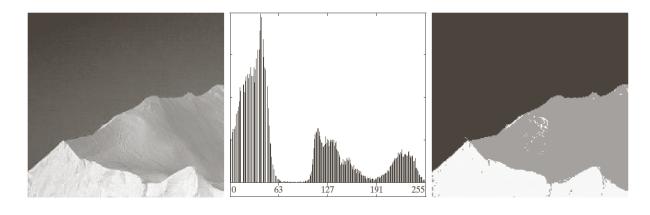
- 图像分割概述
- 基于阈值的方法
- 基于边缘的方法
- 基于区域的方法

阈值处理 Threshold

- 通过设定阈值来分割图像
 - 利用图像中目标像素和背景像素 在灰度值上的差异
 - 设定阈值来分割前景/背景
 - 以图像直方图来指导
 - 单阈值分割 / 多阈值分割

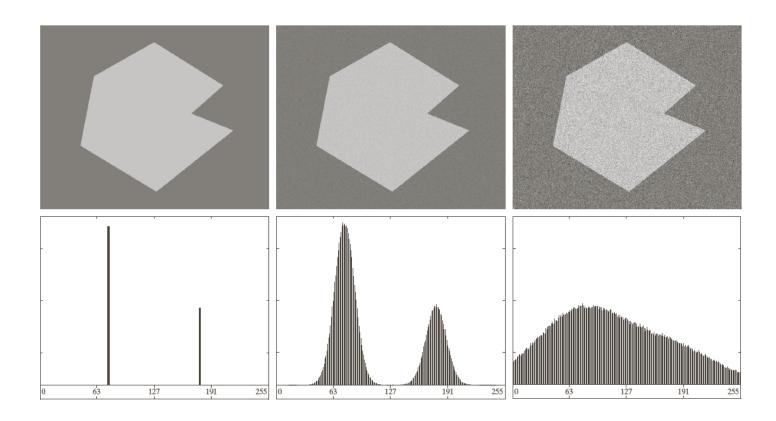




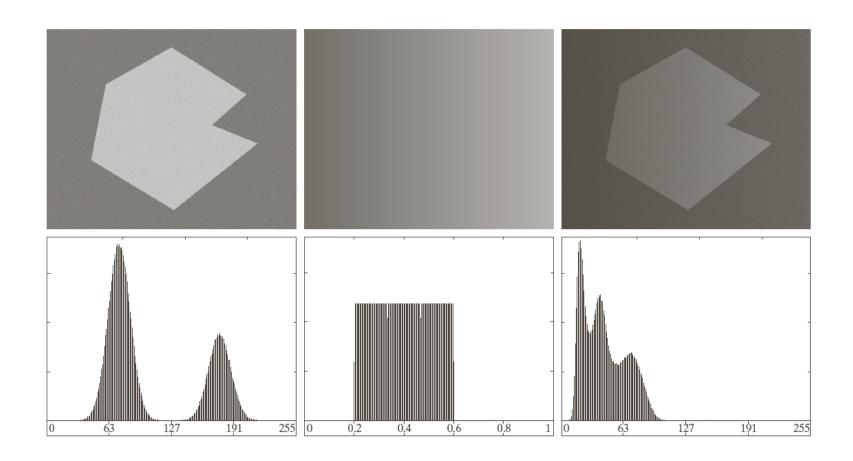


$$g(x,y) = \begin{cases} 1 & f(x,y) > T \\ 0 & f(x,y) \le T \end{cases} \qquad g(x,y) = \begin{cases} a, & f(x,y) > T_2 \\ b, & T_1 < f(x,y) \le T_2 \\ c, & f(x,y) \le T_1 \end{cases}$$

噪声影响

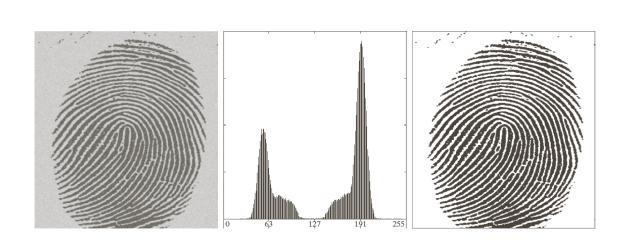


光照和反射



自动确定阈值的算法

- 全局阈值分割
 - 初始全局阈值T为图像的平均灰度值
 - 分割图像,产生两组像素集合: ≥T 和 <T
 - 分别计算两组像素的平均灰度值m1和m2
 - 计算一个新的阈值: *T* = (m1+ m2) / 2
 - 重复上述步骤, 直到T变化很小时停止



指纹图像分割:初始设为图像的平均值,通过3次迭代自动找到T=125



Otsu法 大津法

- 看作一个分类问题
- 用一个阈值将像素值分成两类
- 最小化类内方差等价于最大化类间方差

$$\sigma_W^2 = \omega_0 \sigma_0^2 + \omega_1 \sigma_1^2$$
 类内总方差
$$\sigma_B^2 = \omega_0 (\mu_0 - \mu_T)^2 + \omega_1 (\mu_1 - \mu_T)^2$$
 类间总方差
$$= \omega_0 \omega_1 (\mu_1 - \mu_0)^2$$

• 使用穷举法选择阈值,最大化 类间总方差

$$egin{align} \omega_0(t) &= \sum_{i=0}^{t-1} p(i) & \mu_0(t) &= rac{\sum_{i=0}^{t-1} i p(i)}{\omega_0(t)} \ \omega_1(t) &= \sum_{i=t}^{L-1} p(i) & \mu_1(t) &= rac{\sum_{i=t}^{L-1} i p(i)}{\omega_1(t)} \ \mu_T &= \sum_{i=0}^{L-1} i p(i) \ \omega_0 \mu_0 + \omega_1 \mu_1 &= \mu_T \ \end{pmatrix}$$

 $\omega_0 + \omega_1 = 1$



Otsu法 大津法

$\omega_0(t) = \sum_{i=0}^{t-1} p(i) \,\,\, \mu_0(t) = rac{\sum_{i=0}^{t-1} i p(i)}{\omega_0(t)}$

Algorithm

- $\omega_1(t) = \sum_{i=t}^{L-1} p(i) \,\,\, \mu_1(t) = rac{\sum_{i=t}^{L-1} i p(i)}{\omega_1(t)}$
- 1. Compute histogram and probabilities of each intensity level

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

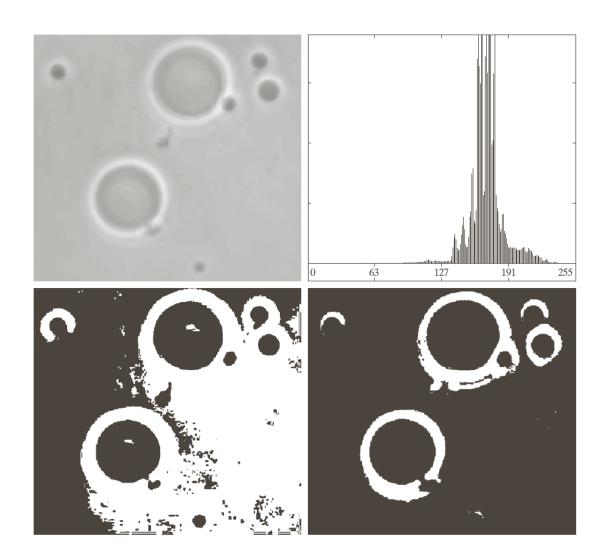
- 2. Set up initial $\omega_i(0)$ and $\mu_i(0)$
- 3. Step through all possible thresholds $t=1,\dots$ maximum intensity
 - 1. Update ω_i and μ_i
 - 2. Compute $\sigma_b^2(t)$
- 4. Desired threshold corresponds to the maximum $\sigma_b^2(t)$

$$\sigma_W^2 = \omega_0 \sigma_0^2 + \omega_1 \sigma_1^2$$

$$\sigma_B^2 = \omega_0 (\mu_0 - \mu_T)^2 + \omega_1 (\mu_1 - \mu_T)^2$$

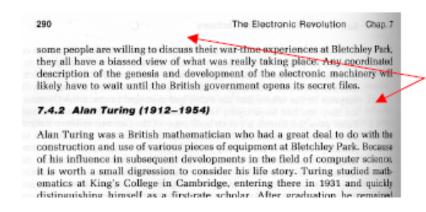
$$= \omega_0 \omega_1 (\mu_1 - \mu_0)^2$$

全局法和Ostu法对比



阈值分割小结

- 实现简单,运算复杂度低
- 目标与背景灰度值有较大差异时有效
- 对噪声、光照敏感
 - 不均匀亮度图像
- 复杂场景图像效果不佳



光照不均匀,全 局阈值不好取



改进

• 对噪声进行平滑

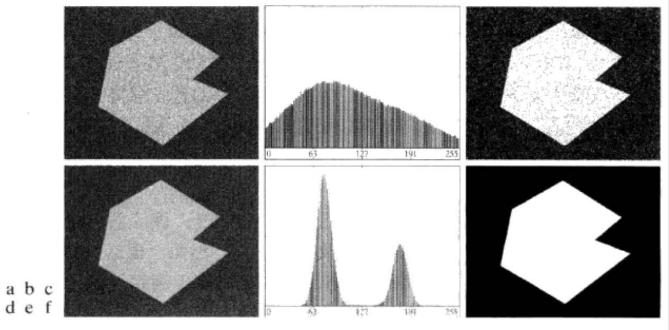
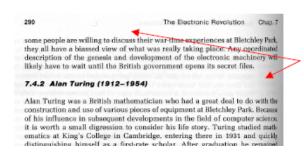


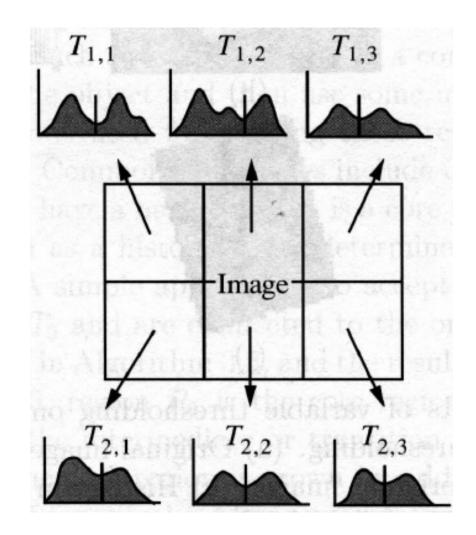
图 10.40 (a)来自图 10.36 的噪声图像; (b)该图像的直方图; (c)用 Otsu 方法得到的结果; (d)用一个大小为 5×5 的均值模板平滑噪声图像的结果; (e)结果图像的直方图; (f)用 Otsu 方法阈值处理后的结果

改进2

- 单一全局阈值不一定适用
 - 不均匀亮度图像



光照不均匀,全 局阈值不好取



内容

- 图像分割概述
- 基于阈值的方法
- 基于边缘的方法
- 基于区域的方法

图像边缘 Edge

- 图像中像素值突变的位置
- 闭合的边缘(边界、轮廓)可以用来分割图像











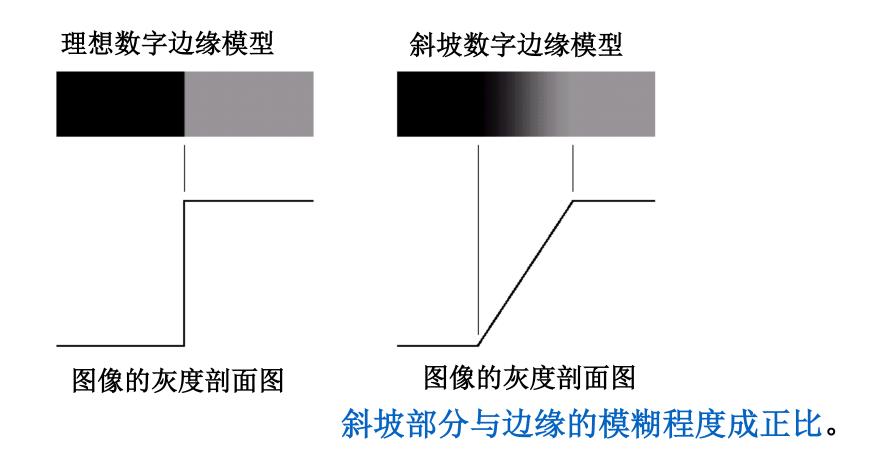






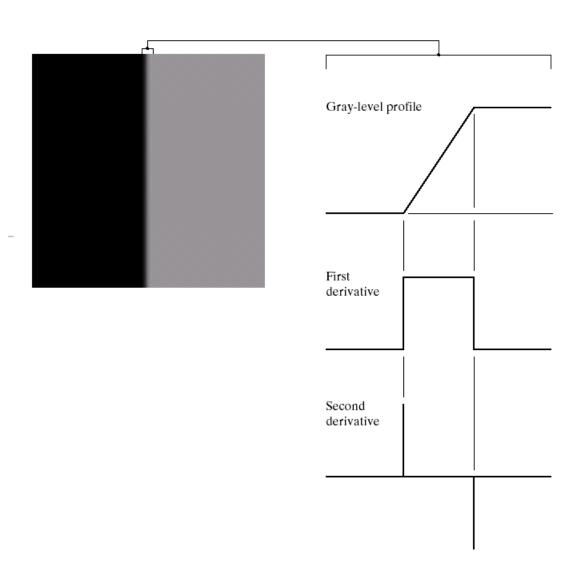


边缘模型



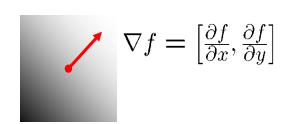
边缘的导数特性

- 一阶导数在边缘上是正的
 - 其它区域为零
- 二阶导数在边缘上
 - 从暗到边缘:正
 - 从边缘到亮: 负
 - 在边缘上: 0
- 过零的特性
 - 从正到负有一个过零的特性
 - 对于边缘定位有用





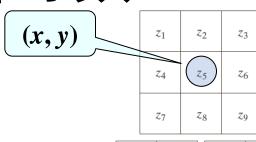
边缘检测-一阶导数



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

$$g = \left| \frac{\partial f}{\partial x} \right| + \left| \frac{\partial f}{\partial y} \right|$$

Better noise-suppression



-1	0	0	-1
0	1	1	0

Roberts

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

Prewitt

_	·1	-2	-1	-1	0	1
()	0	0	-2	0	2
	L	2	1	-1	0	1

Sobel

● 水平垂直差分法

$$\begin{cases} g_x = z_8 - z_5 \\ g_y = z_6 - z_5 \end{cases}$$

Robert Gradient

$$\begin{cases} g_x = z_9 - z_5 \\ g_y = z_8 - z_6 \end{cases}$$

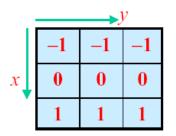
Prewitt 算子:

$$\begin{cases} g_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3) \\ g_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7) \end{cases}$$

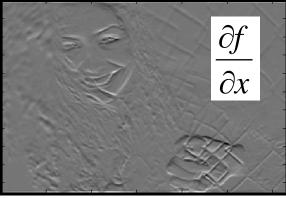
• Sobel 算子:

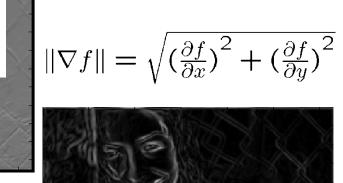
$$\begin{cases} g_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \\ g_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \end{cases}$$

梯度图像

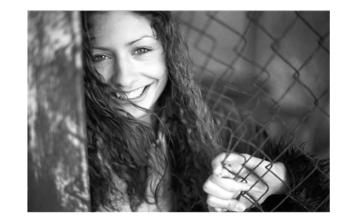


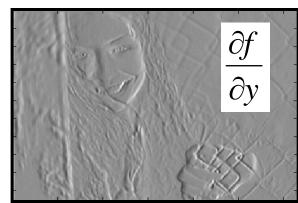
		<u>y</u>					
	-1	0	1				
x	-1	0	1				
	-1	0	1				
_			-				



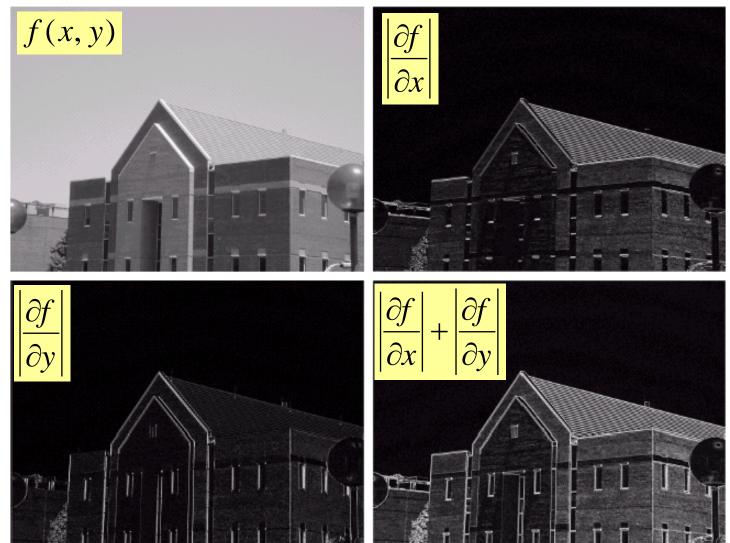




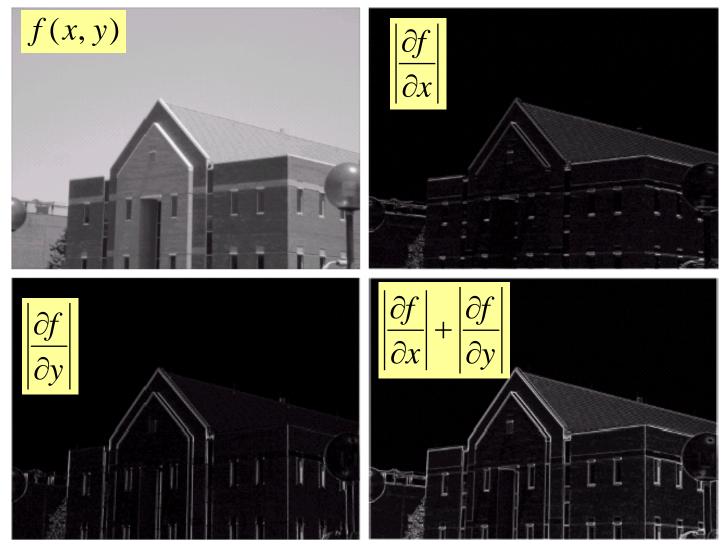




梯度图像



先平滑

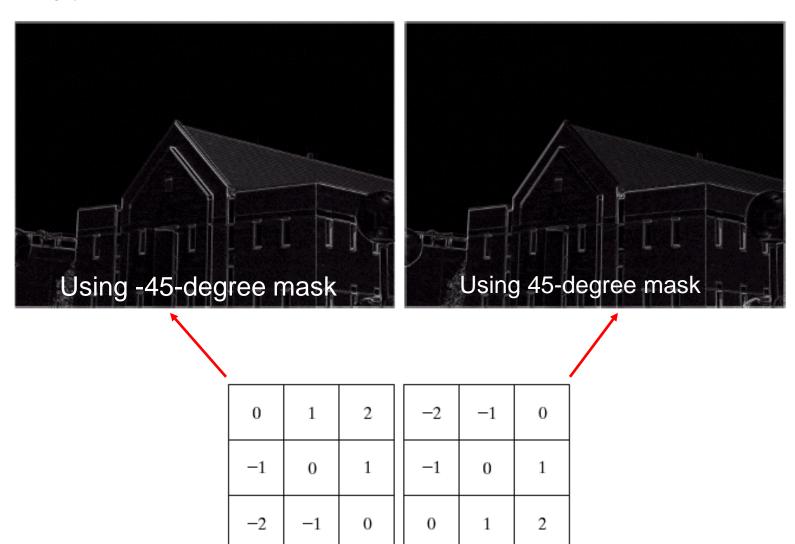


smoothed by a 5x5 moving average mask first.

对梯度图像加阈值



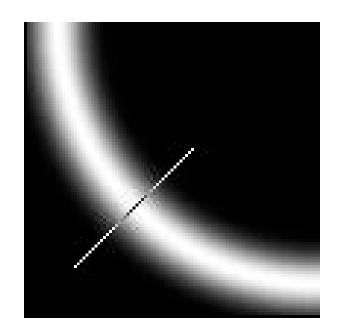
梯度图像

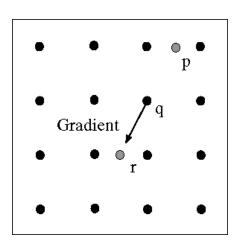


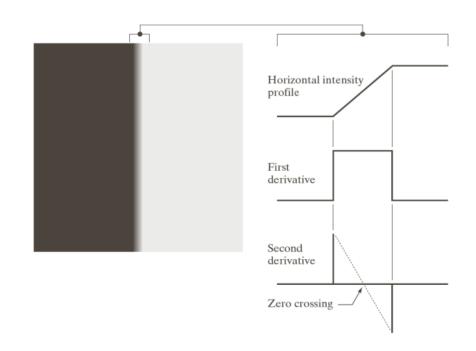
26

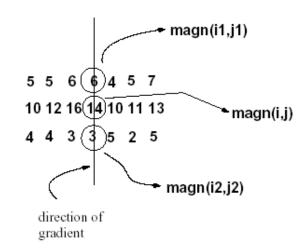
非最大值抑制

- 一阶导数得到的边缘过粗
- Non-maximum suppression
 - 只保留梯度方向梯度值最大的那个点
 - 细化边缘









非最大值抑制





梯度图像 细化后

非最大值抑制





边缘检测-二阶导数

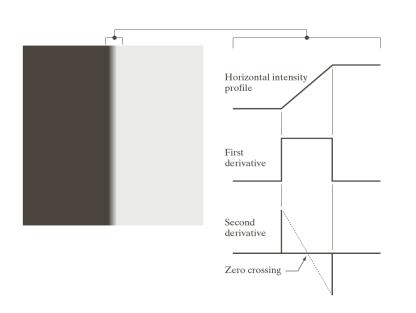
• 拉普拉斯算子

Laplacian
$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial^2 x} + \frac{\partial^2 f(x,y)}{\partial^2 y}$$

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

- •二阶导数的0交叉点
 - 用来精确定位边缘位置

-						
	0	1	0	1	1	1
	1	-4	1	1	-8	1
	0	1	0	1	1	1
	0	-1	0	-1	-1	-1
	-1	4	-1	-1	8	-1
	0	-1	0	-1	-1	-1





拉普拉斯算子

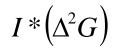




拉普拉斯算子

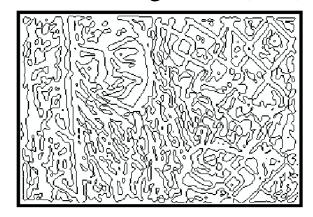
I







Zero crossings of $I^*(\Delta^2 G)$

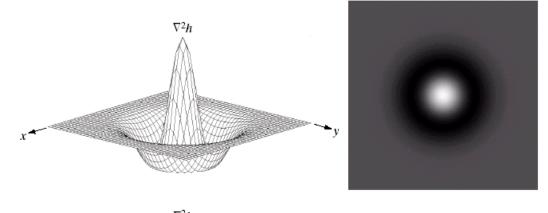


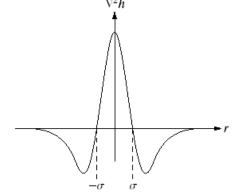
高斯-拉普拉斯(LOG)算子

- 噪声点对边缘检测有较大的影响
- 把高斯平滑和拉普拉斯结合起来
- 先平滑掉噪声,再进行边缘检测
- Laplacian of Gaussian

• (LOG)
$$G(x, y) = -e^{\frac{-(x^2 + y^2)}{2\sigma^2}}$$

$$\nabla^2 G(x, y) = -\left[\frac{x^2 + y^2 - \sigma^2}{\sigma^4}\right] e^{-\left(\frac{x^2 + y^2}{2\sigma^2}\right)}$$



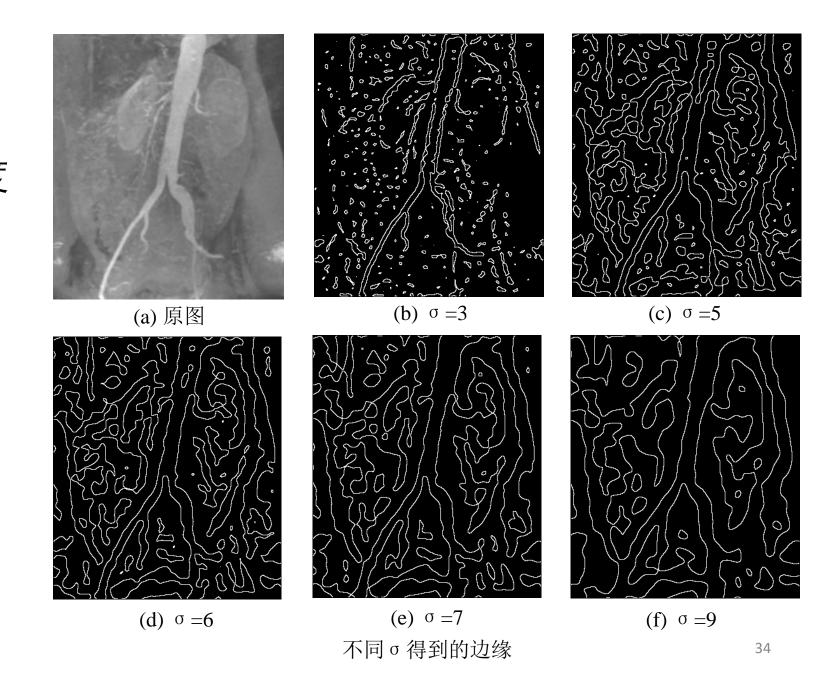


0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0



尺度调节

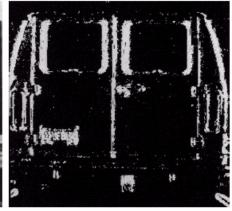
• 调整到想要的尺度



边界闭合

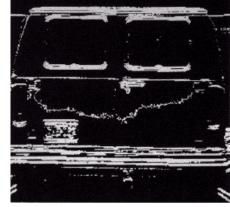
- 边缘像素很可能孤立或分段连续
- 将边缘连接起来组成区域的封闭边界
- 边缘像素连接基础: 相似性
 - 梯度的幅度 $|\nabla f(x,y) \nabla f(s,t)| \le T$
 - 梯度的方向 $|\phi(x,y)-\phi(s,t)| \le A$
- 连接每个边缘点邻域内的相似点





(a)原图

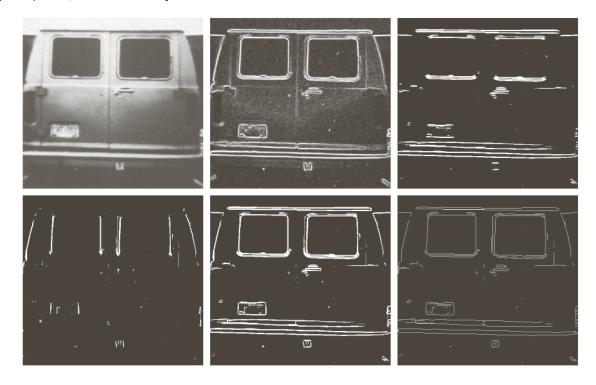
(b)水平梯度分量



(c)垂直梯度分量 (d)边缘连接结果 边缘连接的结果

边界闭合

• 车牌位置定位,美国牌照2:1

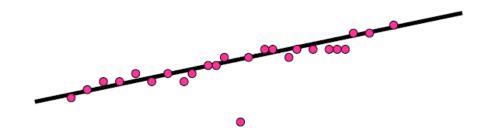


用于检测车牌位置: (a) 汽车尾部(b) 梯度图像(c) 水平连接的边缘(d) 垂直连接的边缘(e) 两幅图像的或 OR(f) 细化后的结果



基于Hough变换的全局处理

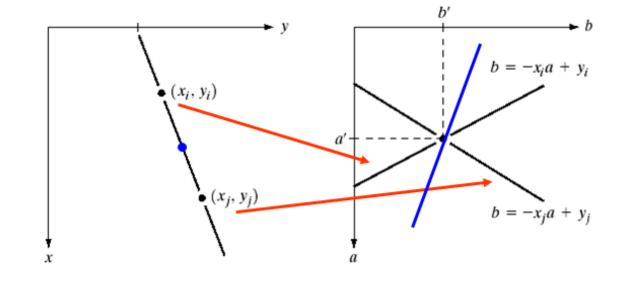
- 根据预定义的全局特性来连接边缘
 - •特定形状,比如直线
- 过 (x_i, y_i) 点有无数条直线 $y_i = a x_i + b$
- 在a, b空间(霍夫空间)表示
 - 图像空间中过特定点(x_i, y_i)的所有直线
 - 表示为霍夫空间中的一条直线 $b = -x_i a + y_i$
 - 该直线上任意一点表示过(x_i, y_i)的一条直线

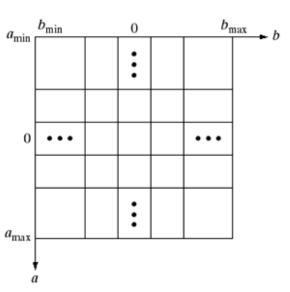




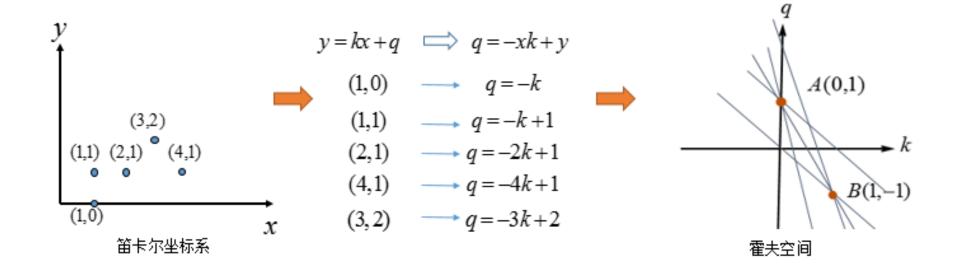
Hough变换

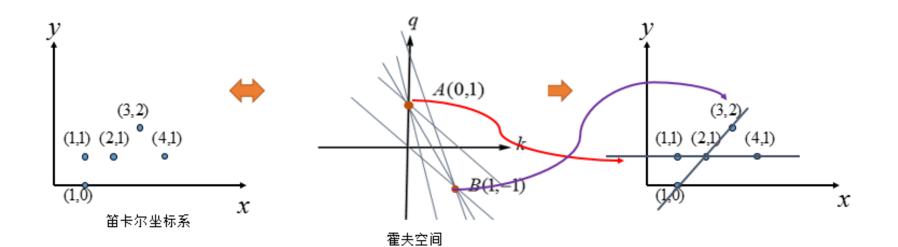
- 考虑有另一个点 (x_j, y_j)
- 找一条直线(a',b')穿过这两个点
- 过 (x_i, y_i) 和 (x_j, y_j) 的一条直线
- 应该是霍夫空间中, 以下两条直线的交点
 - $b = -x_i a + y_i$
 - $b = -x_i a + y_i$
- 画出所有边缘点的参数空间直线
- 离散化参数空间,统计交点多的位置,得到主要直线
- 斜率a的范围过大,不方便离散化统计





交点多,对应于共线的点多





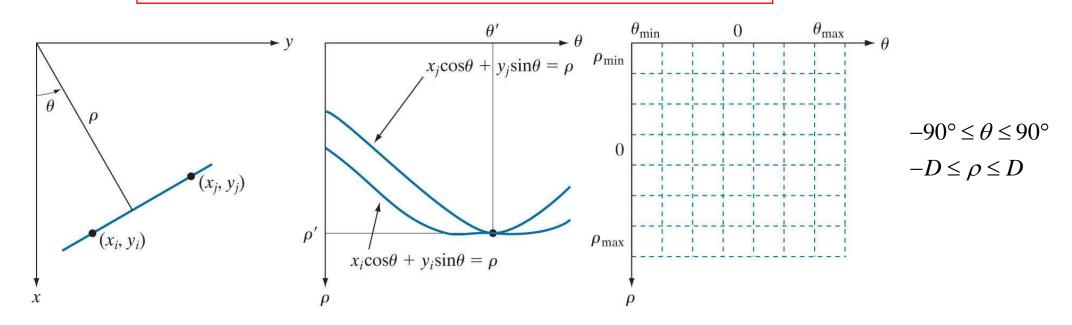


Hough变换

• 换成极坐标参数空间

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

ρ直线到原点的最短距离, θ 是角度

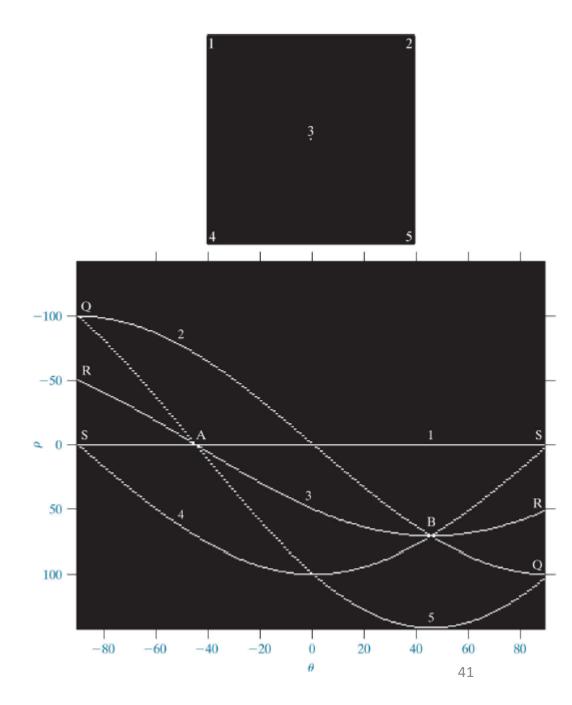


过固定的点所有直线在参数空间里是一条正弦曲线

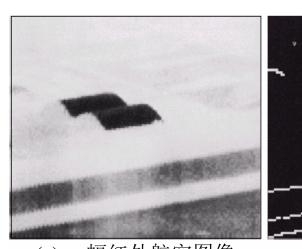
Hough变换

- 将p, θ参数空间离散化
- 对每个边缘点 (x_i, y_i) ,根据 $x\cos\theta + y\sin\theta = \rho$
- 对θ的每个离散值, 求出对应的ρ
- 累计到离散区间中A (ρ, θ)
- 最终的A(ρ , θ)每个值确定了有几个点同在某条直线上

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



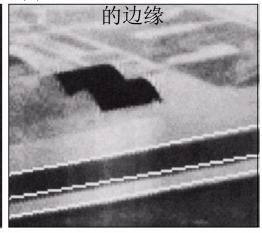
用Hough变换连接边缘



(a) 一幅红外航空图像

(b) 利用Sobel算子检测到





(c) 霍夫变换结果 (d) 检测 霍夫变换的应用实例

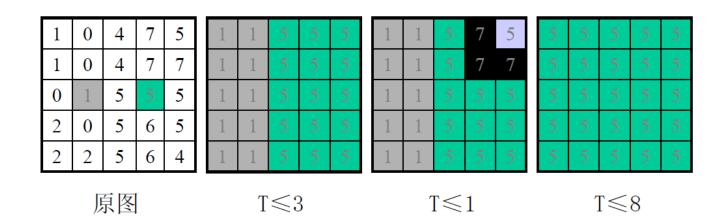
(d) 检测到的三条直线

内容

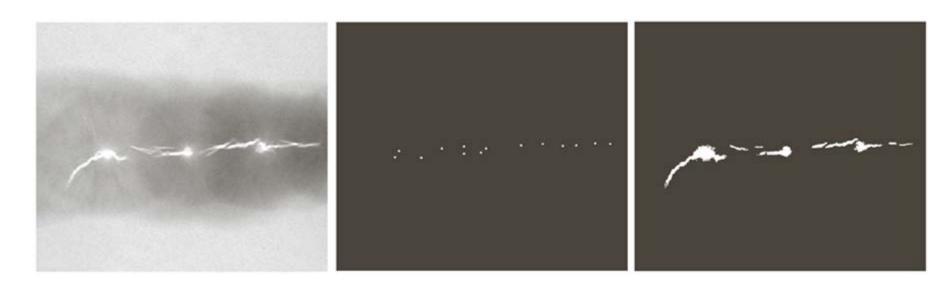
- 图像分割概述
- 基于阈值的方法
- 基于边缘的方法
- 基于区域的方法
 - 区域生长
 - 分裂与聚合

区域生长

- 将相似性质的像素聚合起来实现分割
- 实现步骤
 - 找一些像素点作为种子
 - 设定相似准则,将和种子连通的相似像素合并
 - 将相似像素点作为新的种子, 重复上述过程
 - 满足一定条件停止: 比如区域不再变化



区域生长举例



(a) 缺陷焊缝X光图像

(b) 初始的种子点

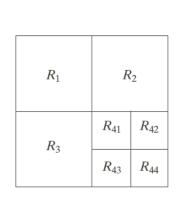
(c) 区域生长的结果

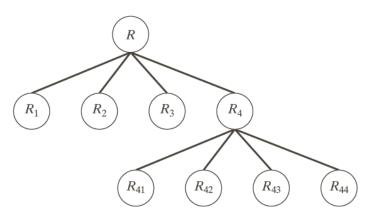
内容

- 图像分割概述
- 基于阈值的方法
- 基于边缘的方法
- 基于区域的方法
 - 区域生长
 - 分裂与聚合

分裂与聚合

- 按照某个准则将区域不断分裂和聚合进行分割
- 算法步骤
 - 设定某个准则,比如区域方差小于10
 - 如该准则在当前区域不满足,则将其区域分裂
 - 如不能进一步分裂,满足准则的相邻区域聚合
 - 如无法进一步聚合,则算法终止





分裂与聚合

$$Q = \begin{cases} TRUE & \sigma > a , \ 0 < m < b \\ FALSE & other \end{cases}$$

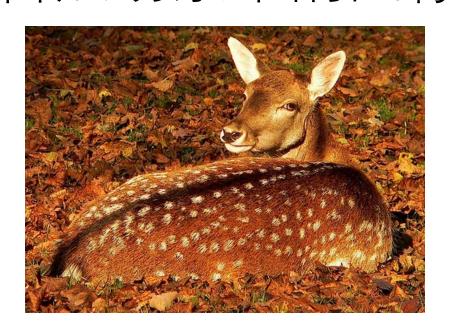
- 分割出图片中的环状行星带
- 行星带所在区域的像素均值和方差有其特性



原图

最小四象限尺寸为 32x32, 16x16, 8x8的分割结果

自然场景中的图像分割是困难的













作业

- •实验8: 分割算法实验
 - 实现OTSU算法
 - 实现自动确定阈值的算法