图像分割

Outline

Topic

图像分割

- 分割的目的: 将图像划分为不同区域
- 方法
  - 根据区域间灰度不连续搜寻区域之间的边界 (间断检测、边缘连接和边界检测)
  - 以像素性质的分布进行阈值处理,(阈值处理)
  - 直接搜寻区域进行分割,(基于区域的分割)

图像分割

- 在对图像的研究和应用中，人们往往仅对图像中的某些部分感兴趣，这些部分一般称为目标或前景
- 为了辨识和分析目标，需要将有关区域分离提取出来，在此基础上对目标进一步利用，如进行特征提取和测量
- 图像分割就是指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
- 特性可以是灰度、颜色、纹理等，目标可以对应单个区域，也可以对应多个区域
- 图像分割算法是基于亮度值的不连续性和相似性
  - 不连续性是基于亮度的不连续变化分割图像，如图像的边缘
  - 根据制定的准则将图像分割为相似的区域，如阈值处理、区域生长、区域分离和聚合

图像分割

- 间断检测
- 边缘连接和边界检测
- 阈值处理
- 基于区域的分割
- 分割中运动的应用

Topic

间断检测

- 间断检测类型
  - 点检测
  - 线检测
  - 边缘检测
- 寻找间断的一般方法: 模板检测

点检测

- 使用如下所示的模板，如果 $|R| \geq T$ 则在模板中心位置检测到一个点。其中，T 是阈值，R 是模板计算值

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

- 基本思想：如果一个孤立点与它周围的点不同，则可以使用上述模板进行检测。
- 注意：如果模板响应为 0，则表示在灰度级为常数的区域

线检测

- 4 个线检测模板

$$\begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix} \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

- 第一个模板对水平线有最大响应
- 第二个模板对 45° 方向线有最大响应
- 第三个模板对垂直线有最大响应
- 第四个模板对 -45° 方向线有最大响应

线检测

- 用 R_1, R_2, R_3, R_4 分别代表 $0°, 45°, 90°, -45°$ 方向线的模板响应, 在图像中心的点, 如果 $|R_i| > |R_j|, j \neq i$ 则此点被认为与在模板 i 方向上的线更相关
- 在灰度恒定的区域, 上述 4 个模板的响应为零

边缘检测

- 边缘：一组相连的像素集合，这些像素位于两个区域的边界上
- 检测方法：一阶导数和二阶导数

边缘与导数

- 一阶导数可用于检测图像中的一个点是否在边缘上
- 二阶导数可以判断一个边缘像素是在边缘亮的一边还是暗的一边
- 一条连接二阶导数正值和负值的虚构直线将在边缘中点附近穿过零点
- 一阶导数使用梯度算子，二阶导数使用拉普拉斯算子

梯度算子

图像 $f(x,y)$ 在位置 (x,y) 的梯度定义为:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

且有:

$$|\nabla f| = (G_x^2 + G_y^2)^{\frac{1}{2}}$$

$$\alpha(x,y) = \arctan(\frac{G_y}{G_x})$$

Roberts 交叉梯度算子

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

Prewitt 梯度算子

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

Sobel 梯度算子

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

梯度算子特点

- Prewitt 和 Sobel 算子是计算数字梯度时最常用的算子
- Prewitt 模板比 Sobel 模板简单, 但 Sobel 模板能够有效抑制噪声

拉普拉斯算子

图像函数的拉普拉斯变换定义为

$$\begin{aligned}\Delta f &= \nabla^2 f \\ &= \nabla \cdot \nabla f \\ &= \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}\end{aligned}$$

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & 0 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

拉普拉斯算子特点

- 缺点:
  - 拉普拉斯算子对噪声具有敏感性
  - 拉普拉斯算子的幅值产生双边缘
  - 拉普拉斯算子不能检测边缘的方向
- 优点:
  - 可以利用零交叉的性质进行边缘定位
  - 可以确定一个像素是在边缘暗的一边还是亮的一边

高斯型的拉普拉斯算子 (LoG)

$$h(r) = e^{-\frac{r^2}{2\sigma^2}}$$

$$r^2 = x^2 + y^2$$

$$\Delta h(r) = -\frac{r^2 - \sigma^2}{\sigma^4} e^{-\frac{r^2}{2\sigma^2}}$$

$$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix}$$

高斯型拉普拉斯算子特点

- 高斯型函数的目的是对图像进行平滑处理
- 拉普拉斯算子的目的是提供一幅用零交叉确定边缘位置的图像
- 平滑处理减少了噪声的影响
- 与梯度算子相比
  - 缺点
    - 边缘由许多闭合环的零交叉点决定
    - 零交叉点的计算比较复杂
  - 优点
    - 零交叉点图像中的边缘比梯度边缘细
    - 抑制噪声的能力和反干扰性能
  - 梯度算子具有更多的应用

Topic

边缘连接

- 由于噪声、照明等产生边缘间断，使得一组像素难以完整形成边缘
- 在边缘检测算法后，使用连接过程将间断的边缘像素组合成完整边缘

局部处理

- 分析图像中每个边缘点 (x, y) 的一个邻域内的像素，根据某种准则将相似点进行连接，由满足该准则的像素连接形成边缘
- 如何确定边缘像素的相似性
  - 边缘像素梯度算子的响应强度
  - 边缘像素梯度算子的方向

如何确定边缘像素的相似性

- 邻域像素梯度算子的响应强度

$$||\nabla f(x_1, y_1)| - |\nabla f(x_2, y_2)|| \leq E$$

- 邻域像素梯度算子的方向

$$|\alpha(x_1, y_1) - \alpha(x_2, y_2)| \leq A$$

通过 Hough 变换进行整体处理

- Hough 变换的基本思想
- 算法实现
- Hough 变换的扩展

Hough 变换的基本思想

- 对于边界上的 n 个点的点集, 找出共线的点集和直线方程。
- 对于任意两点的直线方程: $y = ax + b$, 构造一个参数 a, b 的平面, 从而有如下结论:
- $x - y$ 平面上的任意一条直线 $y = ax + b$, 对应在参数 $a - b$ 平面上都有一个点 (a, b)
- 过 $x - y$ 平面一个点 (x, y) 的所有直线, 构成参数 (a, b) 平面上的一条直线
- 如果点 $(x1, y1)$ 与点 $(x2, y2)$ 共线, 那么这两点在参数 $a - b$ 平面上的直线将有一个交点
- 在参数 $a - b$ 平面求解相交直线最多的点, 即可求得对应 $x - y$ 平面上的直线

Hough 变换算法实现

- 极坐标形式: $x\cos\theta + y\sin\theta = \rho$
- 参数平面为 θ, ρ
- 使用交点累加器, 或交点统计直方图, 找出相交线段最多的参数空间的点
- 找出该点对应的 xy 平面的直线线段

Hough 变换圆检测

$$(x-c_1)^2 + (y-c_2)^2 = c_3^2$$

Topic

阈值处理

- 阈值处理操作
- 基本全局阈值
- 基本自适应阈值
- 最佳全局和自适应阈值
- 通过边界特性选择阈值
- 基于不同变量的阈值

阈值处理操作

- $f(x, y)$ 是点 (x, y) 的灰度级, $p(x, y)$ 表示该点的局部性质, 如以 (x, y) 为中心的邻域的平均灰度级

$$T = T(x, y, p(x, y), f(x, y))$$

- 阈值处理后的图像 $g(x, y)$ 定义为

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

- 标记为 1 的像素对应于对象, 标记为 0 的像素对应于背景
- 当 T 仅取决于 f(x,y), 阈值称为全局的
- 当 T 取决于 f(x,y) 和 p(x,y), 阈值是局部的
- 当 T 取决于空间坐标 x 和 y, 阈值就是动态的或自适应的

基本全局阈值算法

- 选择一个 T 的初始估计值
- 重复以下步骤, 直到逐次迭代所得的 T 值之差小于事先定义的参数 T_0
- 用 T 分割图像, 生成两组像素:
  - G_1 由所有灰度值大于 T 的像素组成,
  - G_2 由所有灰度值小于或等于 T 的像素组成
- 对区域 G_1 和 G_2 中的所有像素计算平均灰度值 $_1$ 和 $_2$
- 计算新的阈值 $T = \frac{1}{2}(\mu_1 + \mu_2)$

基本自适应阈值

- 单一全局阈值存在的问题
  - 不均匀亮度图像无法有效分割
- 方法
  - 将图像进一步细分为子图像，并对不同的子图像使用不同的阈值处理
  - 关键问题:
    - 如何将图像进行细分
    - 如何为得到的子图像估计阈值
  - 自适应阈值: 取决于像素在子图像中的位置

最佳全局和自适应阈值

- 假设一幅图像仅包含两个主要的灰度级区域。令 z 表示灰度级值，则两个灰度区域的直方图可以看作它们概率密度函数 (PDF) 的估计 $p(z)$
- $p(z)$ 是两个密度的和或混合。一个是图像中亮区域的密度，另一个是暗区域的密度
- 如果 $p(z)$ 已知或假设，则它能够确定一个最佳阈值 (具有最低的误差) 将图像分割为两个可区分的区域

最佳全局和自适应阈值

假设 2 个 *PDF* 中较大的一个对应背景的灰度级, 较小的一个描述了图像中对象的灰度级, 则混合 *PDF* 是

$$p(z) = P_1 p_1(z) P_2 p_2(z)$$

其中 P_1 是属于对象像素的概率, P_2 是属于背景像素的概率, 假设图像只包括对象和背景, 则

$$P_1 + P_2 = 1$$

最佳全局和自适应阈值

- 在区间 $[a, b]$ 内取值的随机变量的概率是它的概率密度函数从 a 到 b 的积分, 即在这两个上下限之间 PDF 曲线围住的面积
- 将一个背景点当作对象点进行分类时, 错误发生的概率为:

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

这是在曲线 $p_2(z)$ 下方位于阈值左边区域的面积

- 将一个对象点当作背景点进行分类错误发生的概率为

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

这是在曲线 $p_1(z)$ 下方位于阈值右边区域的面积

最佳全局和自适应阈值

- 出错率的整体概率是

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

- 为了找到出错最少的阈值, 使用莱布尼兹法则把 $E(T)$ 对 T 求微分并令结果等于 0, 得到

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

上式解出 T, 即为最佳阈值

- 如果 $P_1 = P_2$, 则最佳阈值位于曲线 $p_1(z)$ 和 $p_2(z)$ 的交点处

最佳全局和自适应阈值

- 高斯密度可以用两个参数均值和方差描述

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

- 出错最少的阈值 T 的解

$$AT^2 + BT + C = 0$$

$$A = \sigma_1^2 - \sigma_2^2$$

$$B = 2(\mu_1\sigma_2^2 - \mu_2\sigma_1^2)$$

$$C = \sigma_1^2\mu_2^2 - \sigma_2^2\mu_1^2 + 2\sigma_1^2\sigma_2^2 \ln\frac{\sigma_2 P_1}{\sigma_1 P_2}$$

- 如果方差相等 $\sigma = \sigma_1 = \sigma_2$，则得到单一的阈值

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

通过边界特性选择阈值

- 如果直方图的各个波峰很高、很窄、对称，且被很深的波谷分开时，有利于选择阈值
- 为了改善直方图的波峰形状，可只把区域边缘的像素绘入直方图，而不考虑区域中间的像素。用微分算子处理图像，使图像只剩下边界中心两边的值
- 优点：
  - 在前景和背景所占区域面积差别很大时，不会造成一个灰度级的波峰过高，而另一个过低
  - 边缘上的点在区域内还是区域外的概率是相等的，因此可以增加波峰的对称性
  - 基于梯度和拉普拉斯算子选择的像素，可以增加波峰的高度

边界特性阈值算法实现:

- 对图像进行梯度计算, 得到梯度图像。
- 得到梯度值最大的那一部分 (比如 10%) 的像素直方图
- 通过直方图的谷底, 得到阈值 T
- 如果用拉普拉斯算子, 不通过直方图, 直接得到阈值, 方法是使用拉普拉斯算子过滤图像, 将 0 穿越点对应的灰度值为阈值 T

基于不同变量的阈值

- 在某些情况下, 传感器可以产生不止一个在图像中描述每一个像素的可利用的变量, 因此, 允许进行多谱段阈值处理
- 例如一幅有 3 个变量的图像 (RGB 分量), 每个像素有 16 种可能的灰度级, 构成 16×16×16 种灰度级 (网格, 立方体)
- 阈值处理就是在三维空间内寻找点的聚簇的过程。如在直方图中找到有效点簇 K, 可以对 RGB 分量值接近某一个簇的像素赋予一个任意值 (如白色的值), 对其它像素赋予另一个值 (如黑色的值)
- 彩色图像处理中的色调和饱和度易于图像分割

Topic

基于区域的分割

- 基本公式
- 区域生长
- 区域分离与合并

基本概念

- 目标: 将区域 R 划分为若干个子区域 $R_1, R_2, \cdots, R_n$，这些子区域满足 5 个条件:
  - 完备性: $\bigcup_{i=1}^{n} R_i = R$
  - 连通性: 每个 R_i 都是一个连通区域
  - 独立性:$R_i \cap R_j = \emptyset, i \neq j$
  - 单一性: 每个区域内像素满足同样的性质（如：灰度级相等）, $P(R_i) = TRUE, i = 1, 2, \cdots, n$
  - 互斥性: 任两个区域像素的性质不同（如：灰度级不等）, $P(R_i \cup R_j) = FALSE, i \neq j$

区域增长的算法实现:

- 根据图像的不同应用选择一个或一组种子, 它或者是最亮或最暗的点, 或者是位于点簇中心的点
- 选择一个描述符 (条件)
- 从该种子开始向外扩张, 首先把种子像素加入结果集合, 然后不断将与集合中各个像素连通、且满足描述符的像素加入集合
- 以上过程进行到不再有满足条件的新结点加入集合为止

区域分裂与合并算法实现:

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

区域分裂与合并算法实现:

- $P(R_i)$ 可定义为: 区域内多于 80% 的像素满足不等式

$$|z_j - m_i| \leq 2_i$$

其中:

  - z_j 是区域 R_i 中第 j 个点的灰度级,
  - m_i 是该区域的平均灰度级,
  - $_i$ 是区域的灰度级的标准方差。

- 当 $P(R_i) = TRUE$ 时, 将区域内所有像素的灰度级置为 m_i

Topic

分割中运动的应用

- 使用两帧图像 $f(x, y, t_i)$ 和 $f(x, y, t_j)$ 相减的办法，形成差值图像

$$d_{i,j}(x, y) = \begin{cases} 1, & |f(x, y, t_i) - f(x, y, t_j)| > T \\ 0, & \text{其它} \end{cases}$$

- 在动态图像处理过程中，$d_{i,j}(x, y)$ 中值为 1 的像素被认为是对象运动的结果
- 考虑图像帧序列 $f(x, y, t_1), f(x, y, t_2), \cdots, f(x, y, t_n)$，并令 $f(x, y, t_1)$ 为基本图像，一幅累积差异图像 (ADI) 由基准图像和图像序列的后续图像对比得到

空间技术

令 R(x,y) 表示基准图像, 绝对 ADI, 正 ADI 和负 ADI 定义如下:

$$A_k(x,y) = \begin{cases} A_{k-1}(x,y)+1, & |R(x,y)-f(x,y,k)| > T \\ A_{k-1}(x,y), & \text{其它} \end{cases}$$

$$P_k(x,y) = \begin{cases} P_{k-1}(x,y)+1, & |R(x,y)-f(x,y,k)| > T \\ P_{k-1}(x,y), & \text{其它} \end{cases}$$

$$N_k(x,y) = \begin{cases} N_{k-1}(x,y)+1, & |R(x,y)-f(x,y,k)| < -T \\ N_{k-1}(x,y), & \text{其它} \end{cases}$$