形态学图像处理

文件版本	修改日期/作者	备注内容
V0.1	2019.10.31/AlimyBreak	首次编写该文档,卡在**区域填充**的连通分析理论

Ref:

1. <<精通Matlab数字图像处理与识别>> 张铮等编著 2013.04.01第一版 Page220

一. 预备知识

- 数学形态学，主要应用是从图像中**提取**对于表达和描绘区域形状**有意义**的图像分量，使后续的识别工作能够抓住目标对象最本质(最具区分能力-most discriminative)的形状特征，如**边界**和**连通区域**；同时**像细化**、**像素化**和**修剪毛刺**等技术也常常应用于图像的预处理和后处理中，成为图像增强技术的有力补充.
- 形态学是借助**集合论**的语言来描述.
    - **结构元素**(structure element):设有两副图像A和S,若A是被处理的对象，而S是用来处理A的，则称S为结构元素。结构元素通常都是一些比较小的图像，A与S的关系类似滤波中**图像**和**模板**的关系.

二. 二值图像中的基本形态学运算

- **前景认定习惯**：默认情况下白色的(二值图像中的灰度值为1的像素，或灰度图像中灰度值为255的像素)是前景(物体)，黑色的为背景.

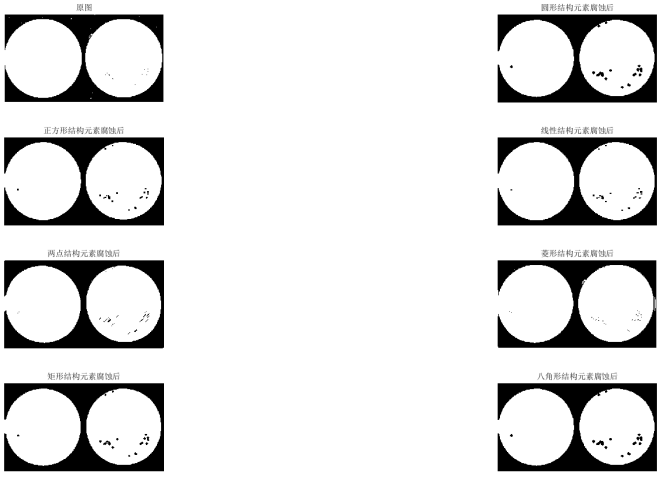
2.1 腐蚀操作

- 理论基础
    - 对于Z^2上的元素集合A和S,使用S对A进行腐蚀操作，记做$A \ominus S$,形式化地定义

$$A \ominus S = \{z \mid (S)_z \subseteq A\} \tag{1}$$

    - 让原本位于图像原点的结构元素S在整个Z^2平面上移动,如果当S的原点平移至z点时能完全包含在A中,则所有这样z点构成的集合即为S对A的腐蚀图像.
    - 可以通过设置S的属性来筛选出满足条件的所有z点，一般来说所有z点构成的集合元素会二值图像中的物体轮廓的图像元素构成的集合元素少一些.
- matlab 函数及用例
    - $erode.m$

- 效果：



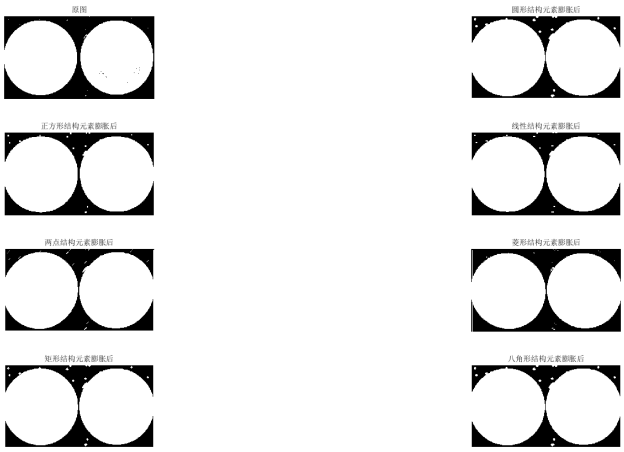
- halcon 函数及用例(略)
- opencv函数及用例(略)
- 腐蚀的作用:
  - 腐蚀能消融物体的边界，腐蚀结果与图像本身和结构元素有关
    - 若物体整体上大于结构元素，腐蚀的结果是使得物体"瘦"一圈，这一圈到底是多大由结构元素决定；
    - 若物理本身小于结构元素，则在腐蚀后图像中的物体将完全消失；
    - 若物体仅有部分区域小于结构元素（在细小的连通），则腐蚀后物体会在细小的连通处**断裂**，分离为两部分.

2.2 膨胀操作

- 理论基础
  - 对Z^2上元素的集合A和S,使用S对A进行膨胀,记作$A \oplus S$,形式化地定义为

$$A \oplus S = \{(\hat{S})_z \cap A \neq \varnothing\} \tag{2}$$

  - 设想有原本位于图像原点的结构元素S,让S在整个Z^2平面上移动，当其自身原点平移至z点时，S相对于其自身的原点的映像$\hat{S}$和A有公共交集，即$\hat{S}$和A至少有一个像素是重叠的，则所有这样的z点构成的集合为S对A的膨胀图像.
  - 实际上膨胀和腐蚀对于集合求补和反射运算是彼此对偶的，与前景认定习惯相反的图片进行膨胀对应的就是与前景认定习惯相同的图片进行腐蚀一样操作，区别是像素需要取反.
- MATLAB函数及用例
  - $delilate.m$
  - 效果：



- Halcon函数及用例(略)
- opencv函数及用例(略)
- 膨胀的作用:
  - 和反射相反，**膨胀能使物体边界扩大**；具体的膨胀结果与图像本身和结构元素的形状有关，膨胀常用于将图像中原本断裂开来的同一物体桥接起来。
  - 对图像进行二值化之后，很容易使得一个连接的物体断裂为两个部分，而这会给后续的图像分析造成困扰（如要基于连通区域的分析统计物体的个数），此时就可借助膨胀桥接断裂的缝隙.

2.3 开运算

- 开运算和闭运算都是由腐蚀和膨胀复合而成的，**开运算是先腐蚀后膨胀**，而**闭运算是先膨胀后腐蚀。**
- 理论基础:
  - 使用结构元素S对A进行开运算，记作$A \circ S$，可表示为

$$A \circ S = (A \ominus S) \oplus S \tag{3}$$

  - 一般来说，开运算使图像的轮廓变得光滑，断开狭窄的连接和消除毛刺。
  - 查看imopen中的MATLAB源代码可以发现，就是先腐蚀后膨胀

```
outputImage = imdilate(imerode(inputImage,se,packopt,M),se,packopt,M);
```

2.4 闭运算

- 开运算和闭运算都是由腐蚀和膨胀复合而成的，**开运算是先腐蚀后膨胀**，而**闭运算是先膨胀后腐蚀。**
- 理论基础:
  - 使用结构元素S对A进行闭运算，记作$A \bullet S$,可表示为

$$A \bullet S = (A \oplus S) \ominus S \tag{4}$$

  - 查看imclose的MATLAB源代码可以发现，就是先膨胀后腐蚀

```
outputImage = imerode(imdilate(inputImage,se,packopt,M),se,packopt,M);
```

- 开闭运算也是对偶的，但是对于某图像应用多次开或闭运算和只进行一次运算的效果相同,即

$$(A \circ B) \circ B = A \circ B, (A \bullet B) \bullet B = A \bullet B \tag{5}$$

三. 二值图像中形态学应用

- 一些经典的形态学应用，都是基本运算(腐蚀、膨胀、开运算和闭运算)的特定次序组合，并采用一些特殊的结构元素来实现的。
  - 击中与击不中变换
  - 边界提取与跟踪
  - 区域填充
  - 连通分量提取及其实现
  - 细化算法
  - 像素化算法

3.1 击中与击不中变换及其实现

- 击中与击不中变换是一种**形状检测的基本工具**，常用于图像中特定形状的精准定位。
- 理论基础:
  - 记作

$$A \otimes S = (A \ominus S_1) \cap (A^c \ominus S_2) \tag{6}$$

  - 其中$S_1 \cap S_2 = \varnothing$且$S = S_1 \cup S_2$
  - 实际上S_1代表我们感兴趣的物体(要检测的形状)对应的集合，而S_2为S中背景部分对应的集合.
  - 首先用我们感兴趣的物体S_1去腐蚀图像A,得到的结果是使S_1完全包含于A中前景部分时中心点位置的集合U_1,**可以将U_1看做S_1在A中所有匹配点的集合**.
  - 为了在A中精确地定位S_1而排除掉那些仅仅包含S_1但不同于S_1的物体或区域，有必要引入和S_1相关的背景部分S_2,一般来说S_2是S_1周围包络着S_1的背景部分，S_1和S_2并在一起组成了S.
  - $A^c \ominus S_2$正是计算图像A的背景A^c与S的背景部分S_2的腐蚀，得到的结果U_2是使S的背景部分S_2完全包含于A^c时S中心位置的集合.
  - U_1和U_2的交集是这样一些p点的集合：当S中心位于p时，S的前景(物体)部分S_1和A中的某个前景完全重合，而S的背景部分也这A的某个背景部分完全重合，而S_1又是包络在S_2其中的，从而保证了我们感兴趣物体S_1在图像A的p点处找到了一个精准匹配.
- Matlab实现及例子
  - $bwhitmiss\_my.m$
  - $bwhitmiss\_matlab.m$
- 对于结构元素S,我们感兴趣的物体S_1之外的S_2不能选得太宽，因为使得S包含背景S_1的外轮廓，以便在图像中能够找到准确的完全匹配位置。从这个意义而言，物体S_1周围有一个像素款的背景环绕就足够了，S_2选择的太大，可能导致计算出来的结果为空集.
- 根据$(A \ominus B)^c = A^c \oplus \hat{B}$和 (6),可以得到(书上的公式写错了)

$$A \otimes S = (A \ominus S_1) \cap (A^c \ominus S_2) \tag{7}$$
$$= (A \ominus S_1) \cap (A \oplus \hat{S}_2)^c \tag{8}$$
$$= (A \ominus S_1) - (A \oplus \hat{S}_2) \tag{9}$$

3.2 边界提取与跟踪及其实现

- 轮廓是对物体形状的有力描述，对于图像分析和识别。通过**边界提取算法**可以得到物体的边界轮廓；而**边界跟踪算法**在提取边界的同事还能依次记录下**边界像素的位置信息**.

3.2.1 边界提取

- 要在二值图像中提取物体的边界，容易想到的一个方法是将所有物体内部的点删除(置为背景色)。
- 可以逐行扫描原图像，若发现一个前景点的8个邻域点都是前景点，则该点为内部点，在目标图像中将它删除。
- 实际上就是相当于采用一个3×3的结构元素对原图像进行腐蚀，使得只有那些8个点邻域都是前景点的内部点被保留，再用原图像减去腐蚀后图像，恰好删除了这些内部点，留下了边界像素.
- matlab程序和实例
  - $getEdge.m$

3.2.2 边界跟踪

- 为了**依次记录边界上下的各个像素**，边界跟踪：首先按照某种扫描规则找到目标物体边界上的一个像素，而后以这该像素点为起始点，根据某种顺序(如顺时针或逆时针)依次找出物体边界上的其他元素，直到又回到起始点，完成整条边界的跟踪.
- 我们可以按照从左到右、从上到下的顺序扫描图像，这样会首先找到目标物体最左上方的边界点P_0,可以采用逆时针的方法对周围可能的8个点进行搜索.

3.2 区域填充

- 区域填充可以看做**边界提取的反过程**，它是在**边界已知**的情况下得到边界包围的整个区域的形态学技术.
- 连通域概念：
  - 两个像素连通的两个条件是：
    - 两个像素的位置是否相邻
    - 两个像素的灰度值是否满足特定的**相似性准则**(同时满足某种条件，比如在某个集合内或者相等)
  - 令V是用于定义连通性的灰度值集合,例如$V = x \mid 0 < x < 125$
    - **4连通**：对于灰度值在V集合中的像素p和q,如果p在q的**4邻域**($N4(p)$中，那么称像素p和q是4连通的.
    - **8连通**：对于灰度值在V集合中的像素p和q,如果p在q的**8邻域**($N8(p)$中，那么称像素p和q是8连通的.
    - **m连通(混合连通)**:对于灰度值在V集合中的像素p和q,
      - q在p的4邻域中，或者

        - q在p的D邻域中，并且p的4邻域和q的4邻域的交集是空的（即没有灰度值在V集合中的像素点）
        - 那么称这两个像素是m连通的，即4连通和D连通的混合连通.
        - ???? 什么是D连通
- 问题描述:已知某一8连通边界和边界内部的某个点，要求从该店开始填充整个边界包围的区域，这一过程被称为“种子填充”，填充的开始点被称为“种子”.