

寻找分叉点过程详解

邱欣欣

2015 年 3 月 19 日

1. 简介

文档主要介绍寻找分叉点的主要过程，这部分程序集中在 `Registration/points_init.m` 中，程序输入为二值图像 `bw`，输出为其分叉点。

2. 过程

1. 求取图像中每个像素的 8 邻域内像素值为 1 的像素个数，并记录在 `countmap` 中；提取其中值大于等于 3 的像素，因为只有这样的像素点才可能是分叉点。
2. 按 8 连通寻找连通区域 (`bwlabel` 函数)，并标记了每个连通区域的标签值 (标签的值为 1、2、`num` (连通区域的个数))。
3. 根据每个连通区域内的像素个数 (`regionprops` 函数)，将连通区域划分为简单连通区域 (像素个数 < 6) 和复杂连通区域 (像素个数 ≥ 6) 两种情况。

(a) 简单连通区域形状一般为以下情况 (图1)，这种情况下直接求取该连通区域的中心处 (`regionprops` 函数) 作为分叉点 (红色像素处)：

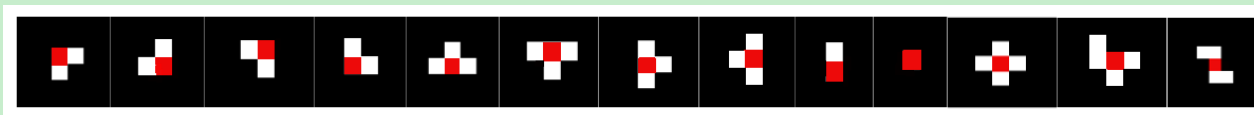


图 1: 简单情况

(b) 复杂连通区域需要进行三步判断：

- i. 当该连通区域像素数为 6 时，在多种情况中先判断是否为图 a 情况 (图2)：否则则判断是否为图 c 情况，其他则为图 b 等情况。图 a 处理方法与简单连通区域方法相同，只选择中心处作为分叉点 (考虑到实际中应该只有一个分叉点)。而图 b、c 及其他的情况下，需要进行连通区域的进一步分离，但两者处理方法不同：图 c 情况需要求得该

连通区域两端的点，再利用该点的 8 邻域像素确定要分割的两个区域；图 b 情况则通过连通区域的像素数按像素序号简单分成两部分（图 c 不能这样处理的原因是对 c 的情况处理会有错误）。图 d 则是一些其他可能出现的情况（这种情况下图 d 第 5 幅图处理方法与图 a 相同，其他与图 b 处理情况相同）。

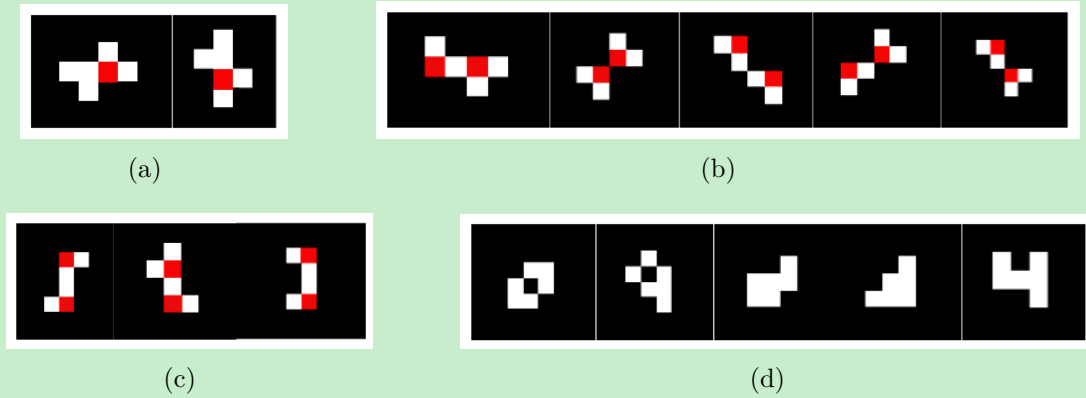


图 2: 第一步判断

- ii. 当连通区域内有一个或以上像素其周围 8 邻域内有 ≥ 5 个像素值为 1，则认为该连通区域表示的为多像素聚集的情况，如下面所示的几种情况（图3），此时与简单连通区域处理方法相同，只选择中心处作为分叉点：

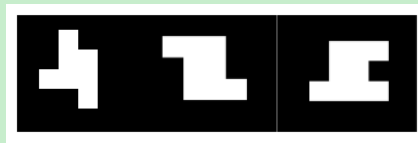


图 3: 第二步判断

- iii. 排除上述两种情况，将剩下的连通区域按像素数划分为两部分，分别求其中心处的点作为分叉点。如下面的几种情况（图4）：

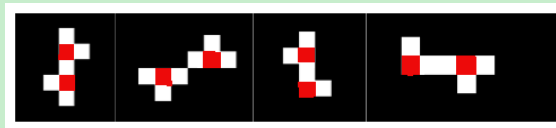


图 4: 第三步判断

3. 其他

1. 当函数寻找的中心处像素值为 0（即没有像素点存在）时，则给该点赋值为 1。