

相似点区段负责制的改进区域生长法及其应用^①

祝森林, 庞 全

(杭州电子科技大学 生物医学工程与仪器研究所, 杭州 310018)

摘 要: 区域生长法被广泛应用于对特定目标的选取, 但其生长速度缓慢的特点又限制了其在某些领域的应用. 为此, 提出一种基于相似点区段负责制的像素点进出堆栈管理方法, 有效的提高了区域生长的效率. 在此基础上, 结合链码法边界跟踪对生长过程中区域内部出现的孔洞实现区域补偿, 建立了针对链状藻细胞分析统计的有效应用系统.

关键词: 区域生长; 链码法; 边界跟踪; 区域补偿; 链状藻

Improved Region Growing Method Based on "A Similar Points is Responsible for a Zone" and Its Applications

ZHU Sen-Lin, PANG Quan

(Institute of Biomedical Engineering and Instrument, Hangzhou Dianzi University, Hangzhou 310018, China)

Abstract: Region growing algorithm is widely used in segmentation of special objects, but the slowly growing speed limits its application in certain area. In this issue, a new method is proposed to improve efficiency of Region growing. This method is based on a mechanism that a certain level zone is described by a single similar point. Only a region with many interior holes can be obtained with this method. Combined with Chain-code edge tracking algorithm, these interior holes can be filled completely and finally get regional compensation. With these two methods, an efficiency application system is established to analyze cells of chain-like algae.

Key words: region growing; chain-code; edge tracking; regional compensation; chain-like algae

1 引言

区域生长法在应用时主要围绕三个方面进行, 即种子点的选取、生长规则的设置、和终止条件的确定. 区域生长法在分割目标时直接针对目标, 不会受图像中其他不相关区域信息的影响, 在一些像素较为均匀且纹理信息较少的目标分割中得到了很好的应用. 但当图像目标的像素较高时, 由于生长过程中像素点不断地进出堆栈, 传统的区域生长法的速度受到很大的影响.

传统的区域填充递归算法^[1,2]较为简单、直观, 但其程序本身的多层递归以及堆栈的频繁操作使得算法本身的效率变低. 区域填充扫描线算法^[3,4]对传统的区域填充进行改进, 避免递归算法因多层递归、反复进

出堆栈操作而造成的时间及内存空间的消耗. 其特点是每次压入相似区段的两个端点来达到效率的提高.

本文提出进一步改进的策略, 即建立一种“相似点区段负责制”来管理相似点的进出堆栈. 这种方法只需利用一个相似点来代替其某一邻域的相似点, 从而能最大限度的减少相似点进出堆栈的次数, 比传统的区域生长算法在效率上有了很大的改善. 在此基础上, 结合链码法轮廓跟踪等技术, 建立了针对链状藻细胞分析统计的专用系统, 有效解决了这类复杂细胞的分析统计问题.

2 基于相似点区段负责制的区域生长算法

考虑一幅 R、G、B 三个通道图像, 设每一个像素

^① 基金项目: 国家科技部支撑计划(2009BAB47B06)

收稿时间: 2012-01-06; 收到修改稿时间: 2012-02-26

点 $u(x, y)$ 所对应的各通道值为 u_r 、 u_g 、 u_b ，对任意两个像素点 $u_1(x, y)$ 与 $u_2(x, y)$ 规定其容差为：

$$\varepsilon(u_1, u_2) = \text{Max}(|u_{1r} - u_{2r}|, |u_{1g} - u_{2g}|, |u_{1b} - u_{2b}|) \quad (1)$$

若某一像素点 $u(x, y)$ 与种子点 $u_0(x, y)$ 的容差 $\varepsilon(u, u_0) \leq \delta$ 则认为这一点属于与种子点颜色相近的区域，据此设集合

$$Q = \{u(x, y) | \varepsilon(u, u_0) \leq \delta\} \quad (2)$$

其中 δ 为设定的容差大小。

引入一组操作器：标记器 **Mark**、添加器 **Push**、移除器 **Pop**。其中 $\text{Mark}[u(x, y)]$ 表示对像素点 $u(x, y)$ 标记， $\text{Push}[u(x, y)]$ 表示将像素点 $u(x, y)$ 追加到存放像素点的容器 **C** 的尾部， $\text{Pop}[u(x, y)]$ 表示将像素点从容器 **C** 的尾部移除，由此定义集合 **R**、**P** 如下。

$$R = \{u(x, y) | u(x, y) \text{ be } \text{mayked}\} \quad (3)$$

$$P = \{u(x, y) | u(x, y) \text{ in } C\} \quad (4)$$

规定如果 $\varepsilon(u, u_0) \leq \delta$ ，并且 $u(x, y) \in \sim R$ ，则 $u(x, y)$ 是“合格点”，否则 $u(x, y)$ 为“不合格点”；“合格点”组成的区域称为“合格区”，“不合格点”组成的区域称为“非合格区”。

本文采用“相似点区段负责制”来代替端点压堆栈，其基本思路是：相似点沿右(左)方向遍历的同时将该行上下两行中由非合格区进入合格区的第一个合格点放入容器 **C** 的尾部，那么所放入的每一个合格点都可以代表该合格点所在行的一个邻域，如图 1 所示。图中黑色矩形区(包括灰色方格)代表不合格区，**a**、**b**、**c** 分别代表当前行从左往右遍历所处的几个区段的第一个相似点，★为向右遍历当前行上下两行由非合格区进入合格区的第一个相似点，区段 0、2、3 为当前行上一行合格区，区段 1、4 为当前行下一行合格区。

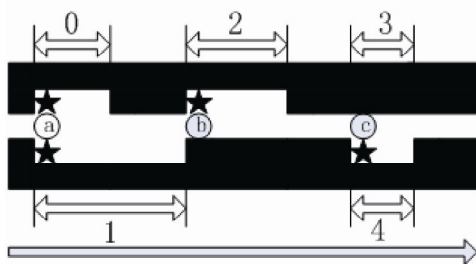


图 1 相似点区段负责制像素进出堆栈示意图

如图 1 所示，当从左往右遍历到达 **a** 时，将 **a** 上

下方相似点压入堆栈，遍历至 **b** 与 **c** 时分别将 **b** 上方相似点与 **c** 下方相似点压堆栈，后面以此类推，当遍历到某一点不合格时，当前区段遍历结束。当前区段遍历完成后弹出堆栈顶元素，以所弹出元素为基准开始新区段的遍历。这样利用“相似点区段负责制”所压入的每一个像素点都可以表征其所在区段的信息，无需将区段中每一个像素点压堆栈，大大地减少了进出堆栈次数，从而提高区域生长的速度。

例如对一幅分辨率为 2048×2048 的区域完全均匀的图片，如果用传统的方法进行区域生长，需要进出堆栈次数约 2048×2048 次，而本文方法只需 2048 次，即每一行压入一个相似点即可。

采用图 2 所示的 3000×4000 的螺旋图片和图 3 所示的 890×764 的着色细胞图片进行试验，分别采用传统的区域生长法和本文所提出的“相似点区段负责制”快速区域生长法进行测试，效率对比如表 1 所示。

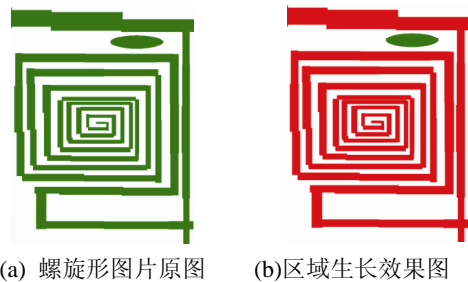


图 2 螺旋形图像试验

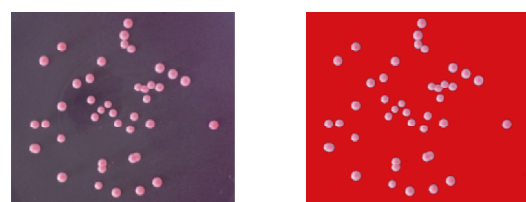


图 3 染色细胞图像试验

表 1 传统方法与本文方法效率对比

图片类型	所用方法	进/出堆栈 (次)	耗时 (ms)
螺旋形图片 3000×4000	传统方法	5075447	2480
	本文方法	19469	205
染色细胞 890×764	传统方法	629236	344
	本文方法	2334	63

3 算法流程及实现

本文建立的相似点区段负责制算法的具体实现如下:

- ① 初始化操作器, 选取种子点压入堆栈.
- ② 判断如果 $P \neq \emptyset$, 弹出堆栈顶端元素作为种子点的新位置, 先从新位置开始往左遍历, 遇到非合格点时开始从新位置往右遍历, 再遇到非合格点时结束当前区段遍历. 在左右两边扫描过程中若遇到合格点则标记, 同时将当前扫描区段上下两行由非合格区进入合格区的第一个合格点压入堆栈.
- ③ 重复步骤②, 直到 $P \neq \emptyset$, 结束区域生长.

算法流程图如图 4 所示.

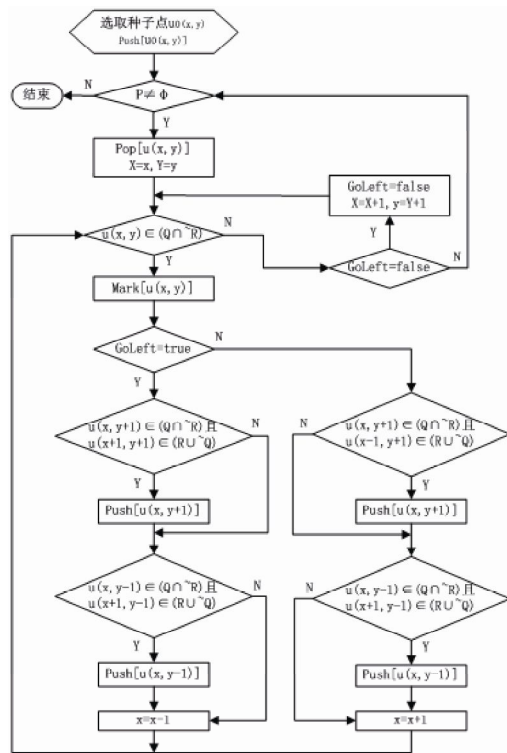
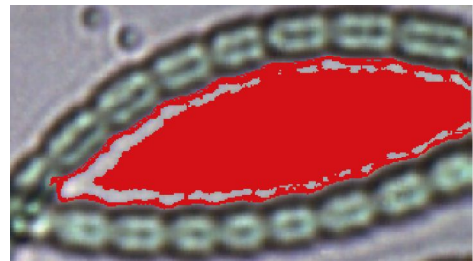


图 4 “相似点区段负责制”的区域生长流程图

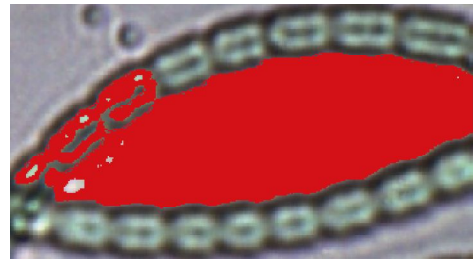
4 生长区域补偿

对种子点进行区域生长时, 遇到背景不均匀的情况, 生长后的区域内部可能存在孔洞, 如果加大容差背景虽能得到完全填充但可能出现过填充的现象, 如图 5 所示.

要将图 5 所示的链状藻准确的分割出来, 须要将这些孔洞填补上. 由于已生长区域的边界点均已被标记, 所以可以根据边界线来将其内部连通区域再次填充以达到补偿的目的.



(a)填充不足, 内部存在空洞



(b)加大容差后, 出现过填充

图 5 背景不均匀情况下的区域生长

对已填充区域的边界采取八链码跟踪^[5], 即根据八方向编码而产生的链码表来获取中心像素邻近的像素最终达到跟踪边界的目的. 八方向链码如图 6 所示, 对应的八方向偏移向量表如表 6 所示.

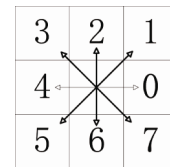


图 6 八方向链码, 中间代表当前像素点位置

表 2 八方向偏移向量表

链码	0	1	2	3	4	5	6	7
X	1	1	0	-1	-1	-1	0	1
Y	0	1	1	1	0	-1	-1	-1

已标记区域的边界跟踪: 利用区域边界点的连通性搜索区域轮廓, 从某一边界点出发, 沿着某一个方向不断搜索下一个轮廓点, 然后根据找到的轮廓点继续搜寻下一轮廓点, 最终得到区域的完整轮廓. 这一过程的关键在于对下一轮廓点的搜索. 为了保证在搜索的过程中不出现重复搜索而导致搜索停滞或搜索无法终止, 需使搜索路径始终沿着轮廓的一个方向进行, 本文采取顺时针搜索邻域的方法保证搜索沿轮廓的正方向进行, 如图 7 所示.

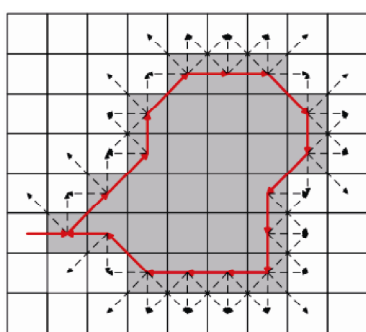


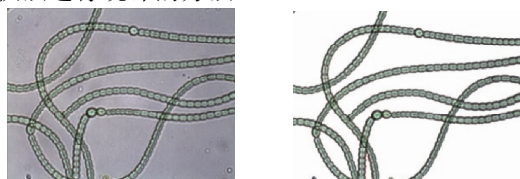
图7 轮廓跟踪正方向搜索示意图

边界跟踪步骤:

- ① 首先从所标记区域任意选取一个边界点作为当前边界点, 设上一边界点为区域外一点;
- ② 获得上一边界点到当前边界点的链码, 将链码加4反向再减1, 并以此链码为起点对8邻域进行搜索;
- ③ 搜索到下一边界点后返回当前边界点到下一边界点的链码, 更新当前边界点;
- ④ 重复步骤2和3直到当前边界点与1中所选的起始边界点重合为止。

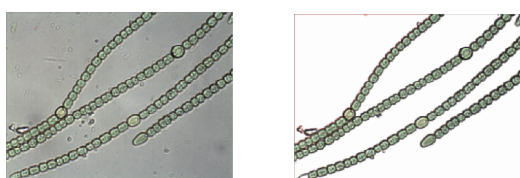
5 链状藻细胞分析个数统计应用系统

链状藻细胞很小, 从而需要采用高分辨率图像观察. 按国家藻类检定规程, 对藻类的检定需要通过100个视野图片的分析统计. 高分辨率和上百张图像的处理, 对算法提出快速高效的要求. 此外, 链状藻内部纹理丰富、又相互交叉导致周围孔洞较多, 直接分割较为困难. 为此, 本文采用剔除背景、保留目标, 并结合面积法进行统计的方法.



(a)链状藻样本 (b)剔除背景后

图8 本文算法的试验效果



(a)链状藻样本 (b)剔除背景后

图9 本文算法的试验效果

图8和图9分别显示了本文建立的系统的分割效果. 其中图(a)显示了链状藻图片, 图(b)显示了采用本文算法剔除背景后的效果.

由前两步处理可以精确地得到背景区域 S_B , 设整幅图像面积为 S_P 则链状藻的总面积 S_T 为 $S_T = S_P - S_B$, 然后随机选取几个样本细胞测其面积分别为 S_1, S_2, \dots, S_k , 平均样本面积 $S_A = \sum_{i=1}^k S_i$, 则

$$\text{细胞的个数 } count = \frac{S_T}{S_A}$$

对图8和图9所示链状藻, 先通过人工观察获得其包含细胞的精确个数. 然后采用本文建立的系统进行自动统计. 将自动统计数据与人工精确计数进行对比, 准确率如表3所示.

表3 人工统计与本文方法统计对比

链状藻	人工统计	本文方法	准确率
图5-1	178	162	91.0%
图5-2	133	126	94.7%

6 结语

本文提出一种基于相似点区段负责制的像素点进出堆栈管理, 有效提高区域生长算法的速度. 同时, 结合链码法边界跟踪等技术, 建立了针对链状藻细胞分析统计的应用系统, 有效性解决了这类复杂细胞的统计问题.

参考文献

- 1 唐荣锡,汪嘉业,彭群生,等.计算机图形学教程.北京:科学出版社,1990.
- 2 Rogers DF. Procedure Elements for Computer Graphics. New York: McGraw-Hill, 1985.
- 3 柳朝阳,李叔梁.压入区段端点的区域填充扫描线算法.计算机辅助设计与图形学学报,1996,8(6):415-419.
- 4 任继成,刘慎权.区域填充扫描线算法的改进.计算机辅助设计与图形学学报,1998,10(6):481-486.
- 5 Freeman H. on the encoding of arbitrary geometric configurations. IRE Trans., 1961,10:260-268.