

基于栅格的任意复杂区域自动填充算法

邱国清

(闽南师范大学计算机学院, 福建 漳州 363000)

摘 要: 基于栅格与等间距平行线原理, 设计出一种适用于任意复杂区域完全自动化填充算法。首先, 将整个填充区域栅格化。其次, 绘制一组等间距平行线, 计算每条平行线经过多边形区域内栅格的行列序号值。最后, 根据计算出栅格的行列序号值, 采用自主循环方法对每个栅格单元填充, 最终实现整个区域的自动化填充。通过自主设计的应用程序验证多组数据表示该算法能快速自动填充, 对实验中出现的难点做了详细分析。

关 键 词: 栅格; 等间距平行线; 区域填充; 算法复杂度; 自动化填充

中图分类号: TP 399

DOI: 10.11996/JG.j.2095-302X.2018030419

文献标识码: A

文章编号: 2095-302X(2018)03-0419-05

Automatic Filling Algorithm of Arbitrary Complex Region Based on Grid

QIU Guoqing

(Computer College, Minnan Normal University, Zhangzhou Fujian 363000, China)

Abstract: A fully automated filling algorithm for any complex area is designed based on the principle of grid and equal-spaced parallel lines. Firstly, we turn the whole area filling into grid. Secondly, we draw a series of equidistant parallel lines and calculate row column ordinal value of each parallel line going through the polygon grid. Thirdly, each grid cell is filled with the autonomous cycle method, according to the calculation of the row and column number value of the grid. Finally, the entire area is automatically filled. The verification of multiple sets of data through an independently designed application program indicates that the algorithm can realize filling quickly and automatically, and a detailed analysis is made on the technical difficulties that occur in the experiment.

Keywords: grid; parallel lines; area filling; algorithm complexity; automatic filling

区域填充是指给出一个区域的边界, 在边界范围内对所有像素单元赋予指定的像素值。经典的区域填充算法递归种子填充和扫描线种子填充及其改进算法能快速实现填充, 但区域填充的全自动化已经成为区域填充领域的主要研究方向。递归种子算法需要预先设置填充胚, 以此为基础不断搜索新的填充胚, 这种方法虽然使用简单, 但无法通过狭窄区域, 不能填满整个区域, 当有多个对象需要填充时, 种子点的选择非常困难^[1], 种子填充算法

效率比较低的原因是大量的出入栈操作^[2]。扫描线填充则需要判断大量像素点的值, 填充效率不佳, 主要用来填充比较简单的标准多边形区域, 比如圆、椭圆以及其他一些简单的多边形, 其对轮廓线的形状有一定要求^[3]。随着图形学技术的发展, 区域填充的全自动化和算法的通用性已经成为衡量区域填充算法的主要参数。基于栅格和等间距平行线原理的全自动区域填充算法在实现自动填充和通用性方面做了具体分析和研究。

收稿日期: 2017-08-10; 定稿日期: 2017-09-12

基金项目: 福建省教育厅中青年骨干教师科研项目(JAT160290)

第一作者: 邱国清(1975-), 男, 江西临川人, 讲师, 硕士。主要研究方向为图形编码处理。E-mail: qiugq02@163.com

1 区域填充算法原理

1.1 等间距平行线基本原理

(1) 旋转坐标系, 使新坐标 Y 轴与等间距平行线相互平行且多边形轮廓点均位于第一象限, 设等间距平行线与原坐标系的 y 轴之间夹角为 θ ($-90^\circ \leq \theta \leq 90^\circ$), 新坐标系下轮廓点的转换计算公式为

$$x'_i = x_i \cos \theta + y_i \sin \theta + H \quad (1)$$

$$y'_i = -x_i \sin \theta + y_i \cos \theta + T \quad (2)$$

(2) 求新坐标系里曲面轮廓点横坐标的最小值和最大值, 取轮廓点中横坐标最小值, d 为等间距平行线的间距, 首先选定一条平行线开始推算, 该平行线与新坐标系 Y 轴之间的距离称为 a 值, 也就是第一条平行线的横坐标, 计算公式为^[4]

$$a = \left\lceil \frac{x'_{\min} - 0.012}{d} \right\rceil + d \quad (3)$$

其中, $\lceil \rceil$ 表示取整符号。

计算平行线与轮廓各条边的交点, 如果 $(a - x'_i)(a - x'_{i+1}) < 0$, 则表明有交点。

(3) 根据斜率公式计算交点坐标值, 即

$$\left(a, y'_i + \frac{(y'_{i+1} - y'_i)(a - x'_i)}{x'_{i+1} - x'_i} \right) \quad (4)$$

(4) 根据交点坐标计算该条平行线穿过的栅格单元的行列序号, 即

$$\begin{aligned} \text{列序号} &= \left\lceil \frac{a}{\text{网格边长}} \right\rceil - 1 \\ \text{行序号} &= \left\lceil \frac{\text{交点起始点纵坐标}}{\text{网格边长}} \right\rceil - 1 \end{aligned} \quad (5)$$

其中, $\lceil \rceil$ 表示取整符号

$$a = a + d \quad (6)$$

如果 $a > x'_{\max}$ 时停止运算, 否则从第(3)步继续循环。

1.2 奇异点的处理

当平行线通过多边形顶点的时候会出现交点个数异常, 这种交点计数也会出现二义性的情况, 就是奇异点问题。正确处理好奇异点问题, 是算法获得成功的关键。可以归纳出两点: ①当角点处的两条轮廓边位于剖面线的两侧时, 应计一个交点为正确。②当角点处的两条轮廓边位于剖面线的同侧时, 应不计交点或者索性计两个交点为正确。要实现上述两个要求, 可以采用左闭右开的原则, 即在

一条轮廓边线段的两个端点中, 一端采用闭区间而另一端采用开区间的处理原则。结果为: 两轮廓边位于剖面线两侧, 则角点处的两个端点必一取一舍; 两轮廓边位于剖面线同侧, 则角点处的两个端点必同取或同舍。

1.3 区域栅格化

栅格数据是最直观的空间数据结构, 是指将二维平面划分为大小一致、紧密相连的网格阵列, 每个网格作为一个像元, 每个像元由行列号确定其位置, 且具有表示实体属性类型或值的编码值。网格质量的好坏直接影响到数值计算的精度^[5]。代表像素的网格通常为正方形、矩形或者等边三角形等。网格边长决定了栅格数据的精度。通过平面上行距、列距固定的点内插值是获取栅格数据的常用方法。

1.4 算法流程图

基于栅格与等间距平行线原理算法中, 首先依次采集区域边界顶点的坐标, 找出其中横坐标的最小值作为等间距平行线的初始值, 同时按设定的精度值将整个区域栅格化, 把对区域的填充转化为对区域内所有栅格单元的填充, 栅格单元划分的越细, 填充效果越明显, 引入一组等间距平行线, 计算每条平行线与所有边界的交点, 根据交点的横坐标或纵坐标的差值来确定该条平行线穿过的单元个数及行列序号, 因为栅格大小均是一致的, 所以可以很快计算出所有单元的坐标, 整个过程完全由计算机程序自动完成, 具体的算法流程如图 1 所示。

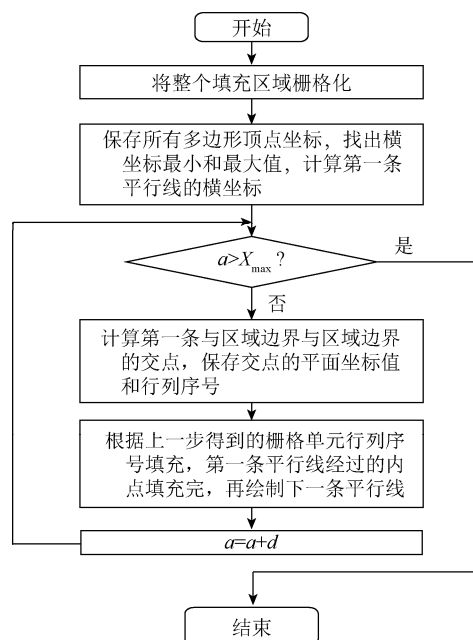


图 1 填充算法流程图

2 数据验证

2.1 不规则复杂区域填充

采集多边形区域所有顶点坐标值绘制填充区域,如图2(a)所示,该区域是一个完全不规则的图形,有多个凹进和凸起部分,对于这样的图形填充,很多专业的绘图软件都比较难实现全自动填充。依照图1,将整个区域栅格化,从图2(b)中可以看到,所有的边界均落在栅格单元中。根据等间距平行线

绘制算法,计算每一条平行线与边界是否存在交点以及交点的坐标,根据交点的坐标可以确定该条平行线在区域中的起点和终点的行列序号;同时根据平行线起点和终点纵坐标的差值计算出该条平行线所经过的栅格单元个数及对应的行列序号;最后根据每个栅格单元行列序号填充,循环依次就完成一条平行线所经过的区域内栅格单元的填充,再执行下一条平行线的绘制,最终的效果如图3所示。

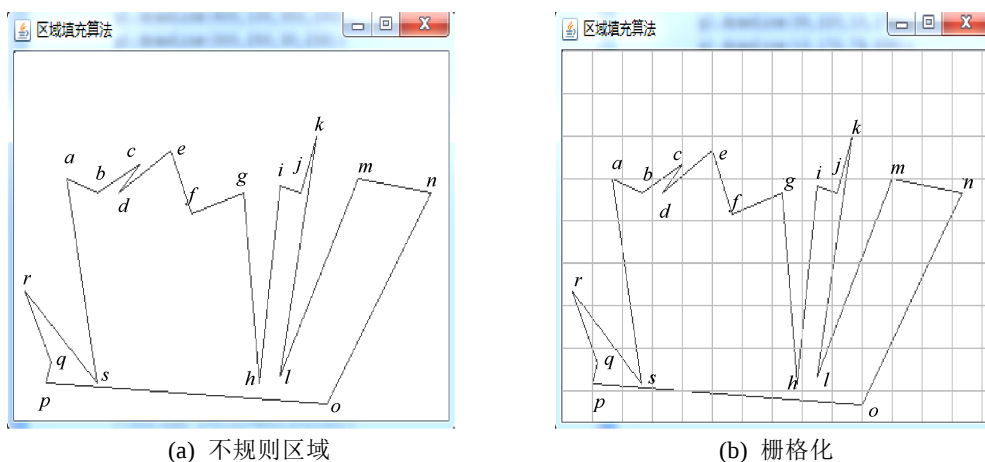


图2 不规则复杂区域的栅格化

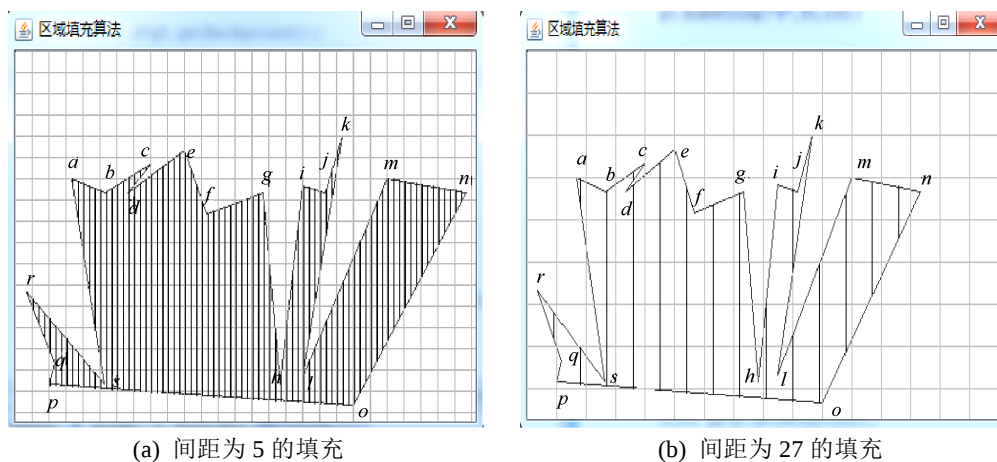


图3 不规则复杂区域的栅格与等间距平行线填充

图3表示的是不规则复杂区域的等间距平行线填充,图3(a)栅格边长为15,平行线间距为5,图3(b)栅格边长为30,平行线间距为27,可以看到每一条平行线均穿过至少一列栅格单元,通过对比可以看出,网格密度越小,平行线间距越细,填充效果越好。以图3(b)为例,每条平行线穿过区域内栅格单元的行列序号见表1。

采用栅格化与等间距平行线填充算法得到的

填充效果取决于栅格的大小,栅格划分越细密,得到的填充效果越好,但计算量也变大。为了做到栅格的每一列或每一行都至少能有一条平行线穿过,约定等间距平行线的间距值不能大于栅格单元的边长,否则会出现若干列或若干行没有被平行线穿过,这样该列或该行的栅格单元就无法被填充,图4是栅格边长为15,等间距平行线间距为8时的填充效果。

表 1 平行线与边界的交点及经过内点的行列序号表

等间距平行线	项目		
	二维坐标值	行列号	栅格单元数目
第 1 条	(27,186),(27,204)	(6,0)	1
第 2 条	(54,91),(54,110),(54,211),(54,237)	(3,1) , (7,1)	2
第 3 条	(81,100),(81,238)	(3,2),(4,2),(5,2),(6,2),(7,2)	5
第 4 条	(108,86),(108,92),(108,96),(108,240)	(2,3),(3,3),(4,3),(5,3),(6,3),(7,3)	6
第 5 条	(135,79),(135,241)	(2,4),(3,4),(4,4),(5,4),(6,4),(7,4)	6
第 6 条	(162,97),(162,243)	(3,5),(4,5),(5,5),(6,5),(7,5)	5
第 7 条	(189,110),(189,244)	(3,6),(4,6),(5,6),(6,6),(7,6)	5
第 8 条	(216,102),(216,246)	(4,7),(5,7),(6,7),(7,7)	4
第 9 条	(243,179),(243,247)	(6,8),(7,8)	2
第 10 条	(270,98),(270,157),(270,202),(270,249)	(3,9),(4,9),(5,9),(7,9),(8,9)	5
第 11 条	(297,152),(297,250)	(5,10),(6,10),(7,10),(8,10)	4
第 12 条	(324,102),(324,214)	(3,11),(4,11),(5,11),(6,11)	4
第 13 条	(351,93),(351,173)	(3,12),(4,12),(5,12)	3
第 14 条	(378,96),(378,133)	(3,13),(4,13)	2

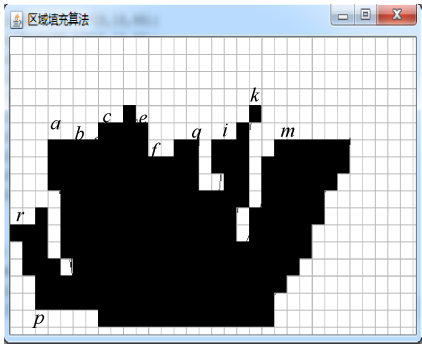


图 4 栅格与等间距平行线算法填充效果

2.2 区域填充与其他填充算法的对比

从区域填充的全自动化和算法的普遍有效性对比，扫描线算法和种子填充算法常常不能同时满足这两方面的要求。扫描线种子算法是指区域内同值相邻像素在水平方向的组合，一段扫描线的中间只能有一种像素，需要存放每条扫描线格填充区域段右端点作为种子点。递归种子算法需要在区域内设定一个种子点而且种子点反复进出堆栈，消耗大量的时间和内存，改进的扫描线种子算法需要找出

种子点所在的区域，这些经典的填充算法在算法的适用性和填充自动化方面的无法做到最大效率。基于栅格化和等间距平行线填充算法通过将区域栅格化并引入一组等间距的平行线，计算每条平行线所经过区域内栅格单元的行列序号并对每个单元一一填充，不需要设置种子点，实现了区域填充的自动化和算法的普遍性，该算法的循环次数等于等间距平行线和多边形边界数目中的最大值，3 种算法比较的对比结果见表 2。

计算复杂度包括空间复杂度和时间复杂度。空间复杂度一般指存储量的问题，时间复杂度则是指计算的工作量问题^[6]。以图 2(b)为例，通过比较可以看出，采用扫描种子算法的存储空间为 58 个点，栅格与等间距平行线算法存储空间为 58 个点，但两种算法的区别在于前者需要把每条扫描线的右端点作为种子胚放入堆栈，采用四邻法的递归种子算法的存储空间为 $2^2+2^4+2^6+\cdots+2^{2n}$ ， n 表示填充次数，经过计算存储空间大约为 72 个点，其中多个点反复进出堆栈。

表 2 算法对比表

算法	项目			
	设置种子点	算法结构	耗费时间	算法适用
扫描线种子	是	双重嵌套循环	$T(n1)+T(n2)$	边界定义的区域
递归种子	是	四邻法或八邻法	不确定	细小区域填充
栅格与等间距平行线	否	双重并列循环	$T(n)$	任意复杂区域

2.3 技术难点的分析

(1) 如果计算平行线所经过区域内栅格单元的行列序号。基于栅格化与等间距平行线填充算法的关键在于程序如何自主搜寻并保存每一条平行线穿过的栅格单元并记录行列序号,为下一步的填充提供数据。根据平行线与边界的交点,计算纵坐标方向的距离,除以栅格单元的边长就等于该条平行线穿过栅格单元的个数,交点的起点和终点代表该条平行线首尾两个栅格单元的平面坐标。

```
for (int j=0; j<tx.length(); j=j+2)
    g.fillRect ((int)(tx[j]/15)*15, (int) (ty[j]/15)* 15,15,
    (ty[j+1]/15-ty[j]/15+1)*15)
```

// tx[j]/15 表示列序号, ty[j]/15 表示行序号, 15 表示栅格单元边长, ty[j+1]-ty[j]+1 表示该条平行线穿过栅格单元个数

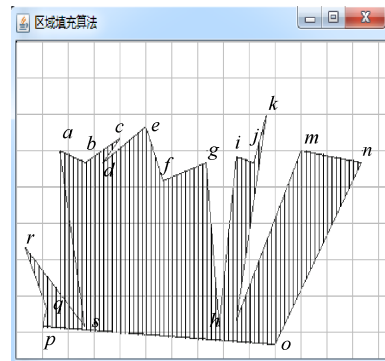
(2) 奇异点的处理。基于栅格与等间距平行线填充算法在绘制等间距平行线时难免会出现奇异点,解决的方式主要是通过稍微增大或者减小平行线的间距值,这样就可以解决。

(3) 如何解决凹进与凸起部分填充问题。对于任意复杂的区域包含多个凸起和凹进部分,当某一条平行线同时穿过多个不规则部位时,偶尔会出现平行线在穿过与自身垂直方向连续多个凹进或凸起区域时,既穿过区域内的栅格单元也穿过了一些区域外不应该被填充的单元,如图 5(a)所示,由于计算机系统有时难以分辨过于狭小的凹进或凸起部分,在图中等间距平行线是垂直方向, sab、rsa、qrs、pqr 连续 4 个水平方向的凹进和凸起,其与等间距平行线方向基本垂直。为了避免出现类似问题,可以稍微调整区域顶点坐标的采集顺序,原来顶点采集顺序为 abcdefghigklmnopqrsa,此时可调整为 sabcdefghijklmnopqrs,调整后此类问题就可以解决,如图 5(b)所示。

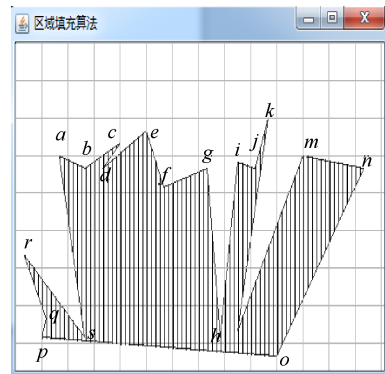
3 结束语

基于栅格化与等间距平行线区域填充算法适用于复杂的区域填充,特别是对于凹进和凸起的区域填充更适合,有很好地填充自动化和算法的适用

性,采用的是网格和等间距平行线计算区域内栅格单元的行列序号,对非常复杂的区域经一次处理即可完成全部的填充。



(a) 采集顺序调整前



(b) 采集顺序调整后

图5 消除多余的区域填充

参考文献

- [1] 陈优广, 顾国庆, 王玲. 一种基于缝隙码的区域填充算法[J]. 中国图象图形学报, 2007, 12(11): 2086-2092.
- [2] 张正峰, 马少飞, 李玮. 新的种子点区域填充算法[J]. 计算机工程与应用, 2009, 45(6): 201-202.
- [3] 柳稼航, 方涛, 杨建峰. 适用于复杂区域的全自动填充方法[J]. 计算机工程, 2008, 34(4): 238-240.
- [4] 闫浩文, 杨树文, 孙建国, 等. 计算机地图制图原理与算法基础[M]. 北京: 科学出版社, 2007: 132-134.
- [5] 刘喜康, 张建海, 殷荣刚, 等. 一种基于逻辑栅格的网格自动生成算法[J]. 计算机工程, 2012, 38(21): 268-271.
- [6] 于海燕, 蔡鸿明, 何援军. 图学计算基础[J]. 图学学报, 2013, 34(6): 1-4.