

8-13-2020

Block Algorithm for Geometric Division of Planar Region Based on Vector and Raster Interblended

Guojun Chen

College of Computer and Communication Engineering, China University of Petroleum, Qingdao 266580, China;

Jingyi Liu

College of Computer and Communication Engineering, China University of Petroleum, Qingdao 266580, China;

Yingying Huang

College of Computer and Communication Engineering, China University of Petroleum, Qingdao 266580, China;

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the [Artificial Intelligence and Robotics Commons](#), [Computer Engineering Commons](#), [Numerical Analysis and Scientific Computing Commons](#), [Operations Research, Systems Engineering and Industrial Engineering Commons](#), and the [Systems Science Commons](#)

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Block Algorithm for Geometric Division of Planar Region Based on Vector and Raster Interblended

Abstract

Abstract: *Using the vector method to maintain the accuracy of the data and the raster method to decrease the search scope, an algorithm was proposed for geometrics division of planar region based on vector and raster interblended. Special treatments for the raster filling polygon of "dead-zone" and the continuous multi-island polygon to adaptively dealt with the common polygon and the polygon with island, as well as the local individual polygons constructing and the global geometric division of planar region. In order to improve the time efficiency, geometric division of planar region used the block division process. Geometric division for sub-block was done, and *polygon seamless contiguity was done by the partition table that set up in segmentation*. Experiments show that the new algorithm can correctly generate the nested relationship between polygons and polygons with "dead-zone", and the polygon seamless contiguity algorithm provides support for parallel processing between blocks.*

Keywords

polygon construction, geometric division of planar region, seamless contiguity, topology

Recommended Citation

Chen Guojun, Liu Jingyi, Huang Yingying. Block Algorithm for Geometric Division of Planar Region Based on Vector and Raster Interblended[J]. Journal of System Simulation, 2016, 28(10): 2460-2466.

矢量与栅格结合的分块平面区域几何划分算法

陈国军, 刘婧怡, 黄莹莹

(中国石油大学(华东) 计算机与通信工程学院, 青岛 266580)

摘要: 利用矢量法保持数据精度以及栅格法缩小搜索范围, 建立矢量与栅格结合的平面区域几何划分算法。针对栅格填充“死区”多边形和带连续多岛屿多边形进行特殊处理, 自适应区分和处理普通多边形与带岛屿多边形, 同时能自适应处理局部单个多边形构建与全局平面区域几何划分。为提高算法时间效率, 平面区域几何划分采用分块处理。先将子块几何划分, 再进行多边形无缝拼接, 利用分割时建立的关系表对多边形进行配对和拼接。实验表明, 新算法能正确生成具有“死区”特征的多边形和多边形间的嵌套关系; 分块拼接算法为块间并行处理提供支撑。

关键词: 多边形构建; 平面区域几何划分; 无缝拼接; 拓扑

中图分类号: TP242.3 文献标识码: A 文章编号: 1004-731X(2016)10-2460-07

Block Algorithm for Geometric Division of Planar Region Based on Vector and Raster Interblended

Chen Guojun, Liu Jingyi, Huang Yingying

(College of Computer and Communication Engineering, China University of Petroleum, Qingdao 266580, China)

Abstract: Using the vector method to maintain the accuracy of the data and the raster method to decrease the search scope, an algorithm was proposed for geometrics division of planar region based on vector and raster interblended. Special treatments for the raster filling polygon of "dead-zone" and the continuous multi-island polygon to adaptively dealt with the common polygon and the polygon with island, as well as the local individual polygons constructing and the global geometric division of planar region. In order to improve the time efficiency, geometric division of planar region used the block division process. Geometric division for sub-block was done, and *polygon seamless contiguity* was done by the partition table that set up in segmentation. Experiments show that the new algorithm can correctly generate the nested relationship between polygons and polygons with "dead-zone", and the polygon seamless contiguity algorithm provides support for parallel processing between blocks.

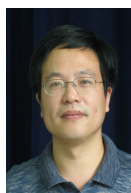
Keywords: polygon construction; geometric division of planar region; seamless contiguity; topology

引言

平面区域几何划分在地理信息系统、地球物理学、计算机图形学以及计算几何等领域有着广

泛的应用。计算机图形学中平面区域几何划分是指 n 条折线 $E=\{E_1, E_2, \dots, E_n\}$ 将一个任意形状的平面区域 $G=\{V, E\}$ 划分成 m 个子区域 G_1, G_2, \dots, G_m 。

20 世纪 70 年代, 自动构建拓扑关系的基本方法已初步生成, Burrough^[1]概括了自动拓扑构建的基本原理。目前, 多边形拓扑关系自动构建主要有两种方法, 一种是基于矢量图形的弧段跟踪算法, 王浩等^[2]提出利用地理信息系统中的空间



收稿日期: 2016-05-10 修回日期: 2016-07-11;
基金项目: 国家 863 计划主题项目子课题 (2015AA016403); 虚拟现实技术与系统国家重点实验室 (北京航空航天大学) 开放基金 (BUAA-VR-15KF-13);
作者简介: 陈国军(1968-), 男, 江苏, 博士, 副教授, 研究方向为图形图像处理, 虚拟现实。

<http://www.china-simulation.com>

• 2460 •

拓扑关系建立思想进行几何划分。闫浩文等^[3]采用基于方位角和最小角法快速构建拓扑关系, 但该算法会产生重复和无效多边形, 消除无效多边形操作增加了程序时间复杂度。梁晓文等^[4]提出一种基于夹角变化趋势来判断左转右转的算法, 实现多边形的快速生成, 但存在无效搜索和回退搜索。申传庆等^[5]采用了矢量外积的高效算法, 避免了无效搜索和回退搜索, 提高了时间效率。另一种是基于栅格图像的种子填充算法, 徐庆荣等^[6]利用栅格填充法建立多边形文件, 艾延华等^[7-8]建立了基于不规则三角网 TIN 结构的形式数据模型 FDTM, 并运用 FDTM 模型, 建立基于不规则三角形的种子填充法构建多边形算法。

基于矢量图形的弧段跟踪算法输出结果精确, 但需要频繁的坐标排序、角度计算等操作, 而且在对折线增删改操作后需要对整个文件重新进行拓扑结构表的构建, 实时性不高; 基于栅格图像的种子填充算法原理和数据结构简单, 实时效果好, 但如果多边形组织的数据基础是矢量弧段, 栅格填充法构建多边形则会在栅格-矢量转换过程中破坏空间数据精度, 产生数据质量问题。结合两者优点, 王结臣等^[9-11]提出建立图幅多边形与弧段所属关系的栅格算法和梯形面片数据模型算法。但其与 FDTM 模型一样, 需要对研究区域内的全部弧段构建多边形, 对所有弧段上的点一次性建立梯形网, 如果只针对局部建立多边形, 则参与特定多边形构建的预选弧段的确定带有盲目性。

1 矢量与栅格结合的平面区域几何划分算法

基于矢量图形的弧段跟踪算法中多边形所属线段的搜索是从起始线段开始的, 然后按一定方向和规则进行追踪, 需对节点的所有线段进行判断。基于栅格图像的种子填充算法中所属线段的搜索是从起始像元开始的, 然后进行八邻域内的追踪并计算出顶点, 需对多边形的所有像元进行连续追踪。事实上, 可通过栅格填充形成的区域边界得出

多边形的边界线段集, 有效减少线段搜索范围, 再利用该线段集进行线段追踪构建多边形。利用矢量法保持数据精度, 利用栅格法缩小搜索范围, 建立矢量与栅格结合的平面区域几何划分算法。

1.1 矢量与栅格结合的多边形构建

平面区域 G 是由若干个子区域组成的, 因此可以通过局部单个多边形构建算法拓展到全局平面区域几何划分算法, 从而能自适应处理局部单个多边形构建与全局平面区域几何划分。以局部单个多边形构建为例, 算法基本步骤为: (1) 线段预处理; (2) 矢量数据栅格化; (3) 种子填充; (4) 栅格数据矢量化求出顶点序列。

预处理时, 区域 G 中折线集 E 内部求交, 如图 1(a)所示, 得到线段集 $GL=\{GL_1, GL_2, \dots, GL_i, \dots\}$ ($i \in \mathbb{N}^*$), 如图 1(b)所示。矢量数据栅格化时, 每个像元存入其所属线段信息。垂直搜索 G 内空白栅格作为种子点, 填充后求得填充的边界序列 B , $B=\{L_1, L_2, \dots, L_i, \dots, L_N\}$, $L_i=\{V_j, V_{j+1}\}$ ($i, j \in \mathbb{N}^* \cap i \in [1, N]$), N 为序列 B 的长度, 如图 1(c)所示。

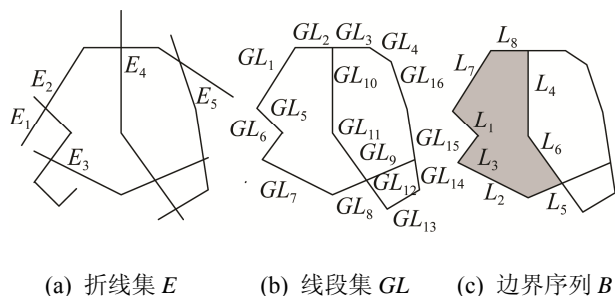


图 1 折线集、线段集与边界序列

此时, B 中的线段不能顺序构成多边形的边界, 需要对 B 中的线段进行追踪, 查找包含同一点的两条线段, 提取多边形顶点序列, 算法如下:

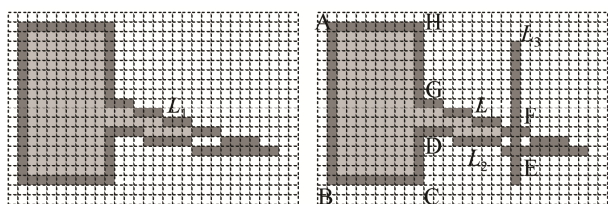
- ①将 L_1 存入顶点序列 V , $V=\{V_1, V_2\}$, 序列 B 中删除 L_1 , 令 $P=V_2$;
- ②查找序列 B , 求得 B 中包含 P 的 $L_i=\{P, V_x\}$, $V_x=(V_j \cap !P) \cup (V_{j+1} \cap !P)$, 将 V_x 存入顶点序列 V , 令 $P=V_x$, 并在序列 B 中删除 L_i ;
- ③重复②, 直到 $B=\phi$ 。

若在查找序列 B 时, 未找到 B 中包含 P 的 L_i , 则表明 B 中丢失了部分线段信息, 其原因是矢量数据栅格化后形成死区。

1.2 栅格填充“死区”多边形处理

在矢量栅格化时, 若 GL 中两条线段的栅格像元部分重叠或相邻, 则种子填充无法到达最外层像元处, 无法到达的区域称之为栅格填充的“死区”。

若第三条线段在“死区”内与造成“死区”的线段相交, 则会出现丢失线段信息的情况。如图 2(a) 所示, 线段 L_1 与 L_2 矢量栅格化后形成的夹角微小, 在栅格图像中出现了栅格填充的“死区”(圈内), 种子填充无法到达“死区”, 所以若 L_3 与 L_1 、 L_2 相交于点 F 、 E , 如图 2(b) 所示, 则种子填充后丢失了线段 EF 的信息, 形成了错误的序列 B 。



(a) “死区” (b) 线段于“死区”相交

图 2 栅格填充“死区”

若 B 中包含某一顶点的线段条数小于 2, 则表明在该点上出现了因栅格填充的“死区”所造成的线段信息丢失的情况, 此时 $B=\{L_1, L_2, \dots, L_i, \dots\}$, $L_i=\{V_j, V_{j+1}\} (i \in N^* \wedge i \neq \text{某个常数})$ 。采用交点交线表辅助 B 中的线段追踪过程, 从而恢复丢失的线段信息, 提取多边形顶点序列。预处理时, 生成交点交线表 $C=\{\{P_1, CL_1\}, \{P_2, CL_2\}, \dots, \{P_i, CL_i\}, \dots\} (i \in N^*)$, P_i 为线段交点, P_i 所对应的交线集为 CL_i , $CL_i=\{CLL_1, CLL_2, \dots, CLL_j, \dots\}$, $CLL_j=\{P_i, CV_j\} (j \in N^*)$, 具体辅助追踪算法如下:

①将 L_1 存入顶点序列 V , $V=\{V_1, V_2\}$, 序列 B 中删除 L_1 , 令 $P=V_2$; ②查找序列 B , 若求得 B 中包含 P 的 $L_i=\{P, V_x\}$, $V_x=(V_j \cap !P) \cup (V_{j+1} \cap !P)$, 将 V_x 存入顶点序列 V , 令 $P=V_x$, 并在序列 B 中删除 L_i ;

若求得 B 中包含 P 的 $L_i=\phi$, 则查找交点交线表 C 中交点 P 所对应的交线集, 设 M 为序列 V 的长度, 若 $(CV_j \in B) \cap (CV_j \neq V_{M-2})$, 则将 CV_j 存入顶点序列 V , $P=CV_j$; ③重复②, 直到 $B=\phi$ 。

1.3 带连续多岛屿多边形处理

带岛屿多边形中存在一种特殊的多边形: 带连续多岛屿多边形, 利用基于栅格图像的算法在岛屿的边界像元追踪过程中, 由于追踪路径不唯一, 会发生错误, 因此需要将其形成独立岛屿后再进行操作, 新增加的独立岛屿栅格图像会造成内存浪费。以下给出带连续多岛屿多边形定义和构建方法。

设 P_1 为外环多边形, $P_2, \dots, P_n (n > 1)$ 均为内环多边形, 若 $P_i (i > 2 \wedge i \in N^*)$ 中存在顶点的度数大于 2, 没有公共顶点的所有边均不相交, 则 P_1, P_2, \dots, P_n 构成的多边形称为带连续多岛屿多边形(亦称带连续多岛多边形, 带连续多孔多边形), 如图 3 所示。图 3 中, P_1 为外环多边形, $P_2 \sim P_4$ 为连续岛屿, P_5 为独立岛屿。

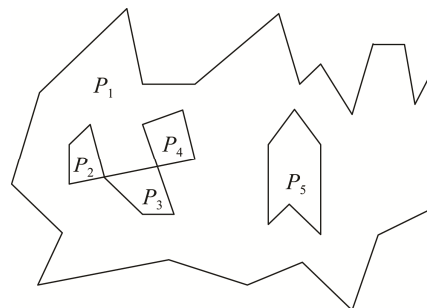


图 3 带连续多岛屿多边形

带连续多岛屿多边形利用上述算法形成的序列 B , 包含了杂糅的多边形外环与内环信息, 因此需要利用岛屿嵌套信息将外环与内环信息分离开。

垂直搜索空白栅格作为新种子点, 求得多边形边界序列 $O=\{B_1, B_2, \dots, B_i, \dots\}$, $B_i=P_i (i \in N^*)$, 令 OP_i 表示多边形 P_i 的外环多边形序列。利用 1.2 中的辅助追踪算法对 O 进行多次追踪直至 $O=\phi$, 生成顶点序列集 $VSV=\{VS_1, VS_2, \dots, VS_i, \dots\} (i \in N^*)$, VS 中包含有外环多边形和内环多边形, 因为已填充像元包含有岛屿嵌套信息, 因此可以利用其对 VS 中的多

多边形进行包含关系判断, 得出包含已填充像元的多边形为外环多边形, 其余为内环多边形。内环多边形顶点序列 $VS_i = \{V_1, V_2, \dots, V_m \dots V_n \dots V_1\}$ ($m \in \mathbb{N}^*$) ($n \in \mathbb{N}^*$), 若 $V_m = V_n$, 则该内环多边形为连续岛屿, 该多边形为带连续多岛屿多边形, 将 VS_i 中的岛屿信息分开, 可得独立岛屿 $I_1 = \{V_m \dots V_n\}$, $I_2 = \{V_1, V_2, \dots, V_m \dots V_1\}$, $I_1 \cap I_2 = \{V_m\}$ 。算法有效地求得完整独立的内外环多边形序列, 如图 4 和表 1。

图 4 中, $P_1 = \{OP_1, OP_2, P_4, OP_5\}$, $P_2 = \{OP_2, P_3\}$, $P_5 = \{OP_5, P_6\}$ 。

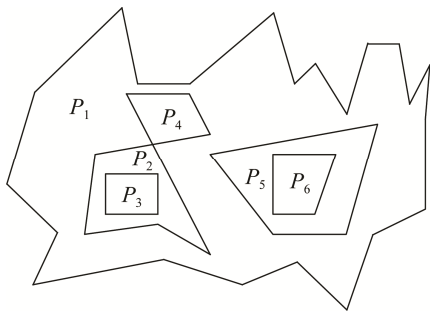


图 4 多层嵌套多边形

表 1 岛屿嵌套关系表

多边形	外环多边形	岛屿集
P_1	OP_1	OP_5, P_4, OP_2
P_2	OP_2	P_3
P_5	OP_5	P_6

2 几何划分的分块处理

分块机制是降低单次数据处理量的常用策略, 当数据量增大时, 算法追踪时间也随之增大, 数据的分块有利于算法时间效率的提高。分块后需要对数据进行无缝拼接, 以保证原区域的几何划分结果正确。王卉^[12]在其博士论文中提出用多边形的交、并运算来实现面状要素的无缝数据生产, 张志友^[13]在其硕士论文中提出了一种基于瓦片分割的矢量数据的合并机制, 两者都致力于已知面状要素的无缝数据生产, 而这里需要对未知的面状要素进行分割来输出面状要素, 之后再行拼接, 因此无法利用已有的操作获得无缝多边形拼接数据, 需要结合几何划分算法的特点设计多边形无缝拼接算法。

2.1 无缝拼接算法过程

一条切割边可将原始区域分成两块, 分割后的区域内折线与切割边相交生成新的多边形, 新生成的多边形有两种: 单拼接边多边形与多拼接边多边形。与切割边相交形成的交线条数为 1 的多边形称为单拼接边多边形, 与切割边相交形成的交线条数大于 1 的多边形称为多拼接边多边形。

分割时建立拼接线段表保存切割边与子区域的关系, 种子填充后建立顶点多边形包含关系表保存切割点与各子区域多边形的关系; 拼接时利用拼接线段表和顶点多边形包含关系表计算出需要拼接的多边形并进行拼接, 将参加拼接多边形的序号成对加入已拼接多边形表。以下以两两分割和拼接为例进行算法说明。

切割边 L 将区域 G 分成 A、B 两区域, L 与区域内线段相交产生的交点序列 $CP = \{CP_1, CP_2, \dots, CP_i \dots\}$ ($i \in \mathbb{N}^*$), 如图 5。 $CP = \{a, b, c, d, e, f\}$, 建立拼接线段表如表 2。分别在区域 A 和 B 中选取空白栅格作为种子点进行填充, 建立顶点多边形包含关系表, 结合拼接线段表, 计算出填充后得到的多边形的拼接边的数目 N_i ($i \in \mathbb{N}^*$), 并将 $N_i \geq 1$ 的多边形对应存入顶点多边形包含关系表, 如表 3。 $N_i = 1$ 为单拼接边多边形, $N_i > 1$ 为多拼接边多边形。

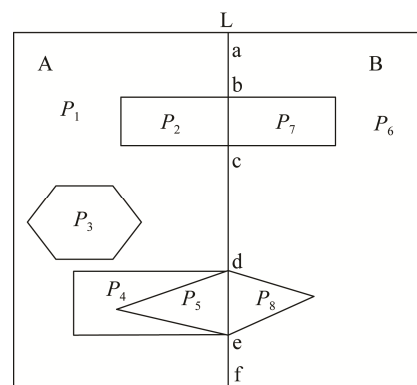


图 5 区域分割图

表 2 拼接线段表

切割边 L	切割边 L 分割后产生的区域	交点序列 CP
$\{a, f\}$	A、B	$\{a, b, c, d, e, f\}$

表 3 顶点多边形包含关系表

交点	属于区域 A 的多边形序列 pA	属于区域 B 的多边形序列 pB
a	P_1	P_6
b	P_1, P_2	P_6, P_7
c	P_1, P_2	P_6, P_7
d	P_1, P_5	P_6, P_8
e	P_1, P_5	P_6, P_8
f	P_1	P_6

根据顶点多边形包含关系表进行拼接: 顺序搜索顶点多边形包含关系表, 从同一顶点的 A、B 两个区域的多边形序列 pA 与 pB 中各选一个多边形进行配对, 再判断是否合并, 若合并, 则合并完成后从顶点多边形包含关系表中删除。令 $NUM_A=pA.size$, $NUM_B=pB.size$ 。

多边形配对: ①若 $NUM_A=1 \cap NUM_B=1$: 都选, 组成一对。若两个均 $N_i > 1$, 选择后加入多拼接边多边形拼接序列; ②若 $NUM_A=1 \cap NUM_B=2$: 若 pB 中的多边形 $N_i=1$, 则一定要选(优先选择单拼接边多边形), 与 pA 组成一对; ③若 $NUM_A=2 \cap NUM_B=1$: 若 pA 中的多边形 $N_i=1$, 则一定要选(优先选择单拼接边多边形), 与 pB 组成一对; ④若 $NUM_A=2 \cap NUM_B=2$: 参照多拼接边多边形拼接序列, 组成两对。

判断是否合并: 若两个多边形中存在一个多边形 $N_i=1$, 则合并, 将参加拼接多边形的序号加入已拼接多边形序列, 合并后 N_i-1 , 然后将 $N_i=0$ 的多边形加入已合并序列, 并将已合并序列中的多边形从顶点多边形包含关系表中删除; 若两个多边形都 $N_i > 1$, 则 N_i-1 , 不合并。

两个多边形合并:

多边形 PA 与 PB 的顶点序列为 $PA=\{A_1, A_2 \dots A_m, A_{m+1} \dots A_1\} (m \in \mathbb{N}^*)$, $PB=\{B_1, B_2 \dots B_n, B_{n+1} \dots B_1\} (n \in \mathbb{N}^*)$, $A_m A_{m+1}$ 、 $B_n B_{n+1}$ 为切割边, $A_m=B_{n+1}$, $A_{m+1}=B_n$, 如图 6 所示。①从 PA 的起始点 A_1 出发, 并存入顶点序列 V , $V=\{A_1\}$ 。②判断 PA 的下一个顶点是否是切割点(A_m), 若是则找到 PB 中的对应的切割点(B_{n+1}), 判断 PB 的下一个顶点(序列循环搜索)是否是切割点(B_n), 若是则找到 PA 中的对应

的切割点(A_{m+1})后, 判断 PA 的下一个顶点是否是起始点 A_1 , 若是则输出顶点序列 V ; 上述判断条件若不满足则将顶点存入顶点序列 V 中。

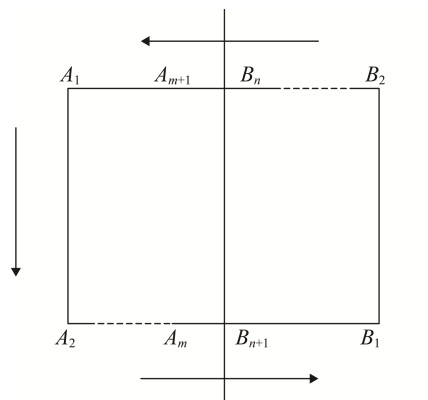


图 6 顶点序列生成图

2.2 无效拼接的删除

分割后的区域内折线与切割边相交可能会产生新的带岛屿多边形, 新形成的带岛屿多边形内的岛屿有两种: 单拼接边岛屿与多拼接边岛屿。与切割边相交形成的交线条数为 1 的内环多边形称为单拼接边岛屿, 大于 1 的称为多拼接边岛屿。

拼接后产生的多边形并非几何划分的子区域, 称之为无效拼接。当多拼接边岛屿内的单拼接边岛屿与多拼接边多边形进行拼接时, 就会产生无效拼接。如图 7 所示, P_3 、 P_{10} 为多拼接边岛屿内的单拼接边岛屿, P_3 与 P_6 、 P_1 与 P_{10} 进行拼接后, 会形成无效拼接多边形。通过查找岛屿嵌套关系表与已拼接多边形序列, 找出无效拼接多边形, 将其删除。以下以两两拼接为例描述算法。

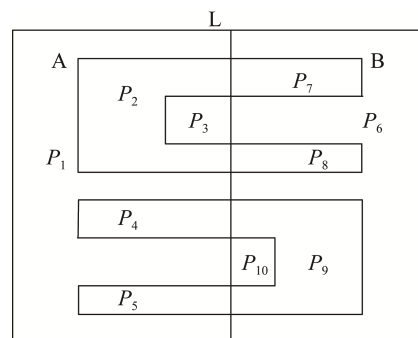


图 7 无效拼接多边形

A 区域中的多边形 $AP=\{AP_1, AP_2 \cdots AP_i \cdots\}$ ($i \in \mathbb{N}^*$), B 区域中的多边形 $BP=\{BP_1, BP_2 \cdots BP_j \cdots\}$ ($j \in \mathbb{N}^*$). 查找 A、B 两区域的岛屿嵌套关系表, $AP=\{P_1, P_2, P_3, P_4, P_5\}$, $BP=\{P_6, P_7, P_8, P_9, P_{10}\}$, $P_1=\{OP_1, OP_2, P_4, P_5\}$, $P_2=\{OP_2, P_3\}$, $P_6=\{OP_6, P_7, P_8, OP_9\}$, $P_9=\{OP_9, P_{10}\}$.

分别找出区域 A 岛屿嵌套关系表中的多拼接边岛屿内(P_2)的单拼接边岛屿 $AP_i(P_3)$, 在已拼接多边形序列中找出与 $AP_i(P_3)$ 拼接的多边形 $BP_j(P_6)$, 若 BP_j 为多拼接边岛屿, 则为无效拼接多边形, 删除; 同理, 对区域 B 的岛屿嵌套关系表进行操作。

3 算法正确性

定理 1: 种子填充后, 形成正确边界序列 $B=\{L_1, L_2 \cdots L_i \cdots L_N\}$, N 为序列 B 的长度, 当 $N \geq 3$ ($N \in \mathbb{N}^*$) 时, 算法输出局部多边形顶点正确。

证明: (1) 当 $N=3$ 时, 多边形边界为 3, 设为三角形 ABC (如图 8(a)), 栅格矢量化后 $B=\{L_1, L_2, L_3\}$, $L_1=\{A, B\}$, $L_2=\{B, C\}$, $L_3=\{A, C\}$, 根据算法: ①从 L_1 出发, $V=\{A, B\}$, 从 B 中删除 L_1 , $B=\{L_2, L_3\}$, 令 $P=B$; ②求得 B 中包含 P 的为 L_2 , $V=\{A, B, C\}$, 从 B 中删除 L_2 , $B=\{L_3\}$, 令 $P=C$; ③求得 B 中包含 P 的为 L_3 , $V=\{A, B, C, A\}$, 从 B 中删除 L_3 , $B=\phi$. 追踪完毕, 输出结果 V 准确。定理 1 在 $N=3$ 时成立。(2) 假设 $N=m$ 时, 定理 1 成立, 即有 m 边形 (如图 8(b)) $B=\{L_1, L_2, L_3 \cdots L_i\}$, 顶点输出序列为 $\{A, B, C, \cdots N, A\}$ 。那么当 $N=m+1$ 时, $m+1$ 边形在 m 边形的基础上增加点 Q (如图 8(c)), $B=\{L_1, L_2, L_3 \cdots L_i, L_{i+1}\}$, $L_i=\{N, Q\}$, $L_{i+1}=\{Q, A\}$, ①根据 $N=m$ 时, 可得算法进行到 L_{i-1} 时 $V=\{A, B, C \cdots N\}$, 从 B 中删除 L_{i-1} , $B=\{L_i, L_{i+1}\}$, 令 $P=N$; ②求得 B 中包含 P 的为 L_i , $V=\{A, B, C \cdots N, Q\}$, 从 B 中删除 L_i , $B=\{L_{i+1}\}$, 令 $P=Q$; ③求得 B 中包含 P 的为 L_{i+1} , $V=\{A, B, C \cdots N, Q, A\}$, 从 B 中删除 L_{i+1} , $B=\phi$, 追踪完毕, 输出结果 V 准确。定理 1 在 $N=m+1$ 时成立。

由数学归纳法, 定理 1 证毕。

注: B 为边界序列, B 为多边形顶点。

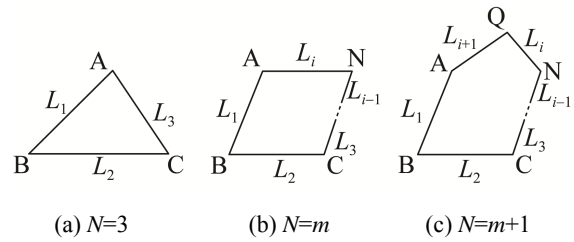


图 8 N 边形 ($N \geq 3$)

定理 2: 输入区域 G 和折线序列 E , 几何划分算法输出子区域 $\{G_1, G_2, \cdots G_i \cdots G_m\}$ ($i, m \in \mathbb{N}^*$), $G_i \cap G_j \in \{\phi, E\}$ ($i \neq j$), $G_1 \cup G_2 \cup \cdots \cup G_m = G$ 。

证明: 区域 G 和折线序列 E 栅格化后形成子区域 $\{G_1, G_2, \cdots G_i \cdots G_m\}$, 令 Q 为子区域的栅格数, 则子区域有两种: $Q > 1$ 和 $Q = 1$ 。① $Q > 1$ 时, 垂直搜索空白栅格进行种子填充, 本文采用的是四连通种子填充算法, 因此填充后的子区域必定为连通区域, 即 $G_i \cap G_j \in \{\phi, E\}$ ($i \neq j$); ② $Q = 1$ 时, 对子区域进行局部放大, 放大后仍采用四连通种子填充算法, 因此 $G_i \cap G_j \in \{\phi, E\}$ ($i \neq j$)。栅格化后对所有栅格进行不重复的垂直搜索, 生成 $Q > 1$ 类的所有子区域且 G_i 唯一; 搜索 $Q = 1$ 的栅格, 生成此类的所有子区域且 G_i 唯一; 则 $G_1 \cup G_2 \cup \cdots \cup G_m$ 唯一且包含所有子区域, $G_1 \cup G_2 \cup \cdots \cup G_m = G$ 。定理 2 证毕。

4 实验结果及分析

实验环境为 Intel(R) Core(TM) i5-2410M CPU、6.00 GB 内存, 实验采用计算机随机生成线段, 如图 9(a), 子区域包含有栅格填充“死区”多边形以及带连续多岛屿多边形, 实验结果如图 9(b)。分块后各子块几何划分串行执行, 统计时间为串行时间。

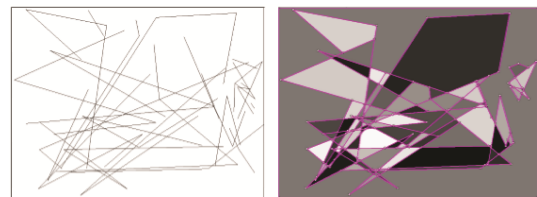


图 9 测试实例

表 4 中, P 表示子区域个数, T 表示不分割时平面区域几何划分算法的时间。表 5 中 JT1 表示两两分割后平面区域几何划分算法的时间, 其分割后子块所用总栅格数量与不分割时相等, JT2 表示两两拼接的时间, JT 表示分割和拼接的总时间。

表 4 不同栅格数量下算法执行时间

P/个	1 000×1 000	2 000×2 000	5 000×5 000
	T/s	T/s	T/s
1 298	0.091	0.352	2.114
2 306	0.094	0.367	2.129
4 902	0.102	0.369	2.132
9 986	0.112	0.382	2.166
19 602	0.129	0.405	2.190

表 5 1 000×500 栅格下不分割与分割情况下算法执行时间

P/个	T/s	JT1/s	JT2/s	JT/s
1 298	0.047	0.049	0.000 001	0.049 001
2 306	0.05	0.053	0.000 001	0.053 001
4 902	0.055	0.063	0.000 001	0.063 001
9 986	0.065	0.086	0.001	0.087
19 602	0.084	0.126	0.001	0.127

表 4 和表 5 表明: (1) 算法时间主要与划分栅格单元的数量有关, 多边形的数量对其影响很小; (2) 两两分割后平面区域几何划分算法的时间约与不分割时相等, 并且拼接算法花费的时间极小, 因此可为块间并行处理提供支撑; (3) 矢量与栅格结合算法的时间效率优于文献[5,9]中的矢量或栅格的多边形自动生成算法。

5 结论

利用矢量与栅格结合进行平面区域几何划分, 与基于矢量图形的弧段跟踪算法和基于栅格图像的种子填充算法比较, 具有以下几个特点: (1) 既可以交互式给出内点, 求出局部单个多边形, 也可以进行自动平面区域几何划分, 求出全局多边形。(2) 结合矢量栅格的优点, 与种子填充扩展同步求出封闭区域线段, 避免了无关线段的搜索, 缩小线

段追踪范围。(3) 特殊处理栅格填充“死区”多边形以及带连续多岛屿多边形。(4) 拼接效率高, 有利于算法的并行。

分析与实验表明, 新算法在特殊情况处理、时间效率等方面有一定优势。下一步工作主要为子块与子块之间的并行处理, 并运用“动态均分”达到负载均衡。

参考文献:

- [1] P A Burrough. Principles of Geographical Information Systems for Land Resources Assessment (Monographs on Soil and Resources Survey) [M]. UK: Oxford University, 1986.
- [2] 王浩, 姚长利, 严红平, 等. 平面区域几何划分的拓扑算法研究 [J]. 计算机应用与软件, 2008, 25(12): 12-14, 42.
- [3] 闫浩文, 杨维芳, 陈全功, 等. 基于方位角计算的拓扑多边形自动构建快速算法 [J]. 中国图象图形学报, 2000, 5(7): 563-567.
- [4] 梁晓文, 刘宗岐, 陈宜金, 等. 基于夹角变化趋势的多边形自动搜索和生成算法 [J]. 中国图象图形学报, 2005, 10(6): 785-789.
- [5] 申传庆, 唐新明, 史绍雨, 等. 多边形自动生成的一种改进算法 [J]. 测绘科学, 2012, 37(2): 105-106, 109.
- [6] 徐庆荣. 用栅格填充法建立多边形文件 [J]. 武汉测绘科技大学学报, 1989, 14(4): 42-46.
- [7] 艾廷华, 郭仁忠. 支持地图综合的面状目标约束 Delaunay 三角网剖分 [J]. 武汉测绘科技大学学报, 2000, 25(1): 35-41.
- [8] 艾廷华, 陈涛. 基于三角网的“种子法”多边形生成 [J]. 武汉大学学报(信息科学版), 2004, 29(1): 14-19.
- [9] 王杰臣. 多边形拓扑关系构建的栅格算法 [J]. 测绘学报, 2002, 31(3): 249-254.
- [10] 王结臣, 崔璨, 陈刚, 等. 梯形面片数据模型及其空间运算应用 [J]. 测绘科学技术学报, 2011, 28(2): 141-145.
- [11] 崔璨, 王结臣. 一种基于梯形剖分的多边形布尔运算方法 [J]. 测绘学报, 2011, 40(1): 104-110.
- [12] 王卉. 无缝 GIS 相关理论与技术的研究 [D]. 郑州: 中国人民解放军信息工程大学, 2004.
- [13] 张志友. 基于瓦片的矢量图幅的分割与合并 [D]. 大连: 大连理工大学, 2010.