

一种平面散乱点集的自适应空间划分算法

王结臣¹ 张 辉¹ 吴文周¹ 王 豹¹

(1 南京大学地理信息科学系, 南京市汉口路 25 号, 210093)

摘 要: 针对平面散乱点集空间划分问题, 提出了一种基于栅格统计的自适应空间划分算法。以栅格场为辅助手段为散乱点集建立空间索引, 即判断各点与栅格的归属关系; 统计各个栅格内包含点的数量; 以栅格为基本统计单元对空间进行划分。划分过程中借助了二叉树结构, 同时引入迭代次数作为划分终止的参数。该方法可灵活地将点集划分为数据量相对均衡的若干组, 且各组的范围较合理。实验与分析表明, 该算法具有较高的计算效率, 也不需占用太多额外的存储空间。

关键词: 散乱点集; 自适应划分; 算法; 地理信息系统

中图分类号: P208

平面散乱点集是 GIS 应用与 GIS 空间分析中常见的处理对象, 如凸包生成、狄罗尼三角网构建、泰森多边形生成、空间数据内插、基于离散点建立 DEM 等。由于新型空间信息获取技术的发展, 点集规模越来越大, 很多空间分析方法的效率不能满足应用需求。为了提高空间数据处理效率, 一些空间分析算法中采用对点集进行空间划分或分治处理的策略, 如构建 Delaunay 三角网、Voronoi 多边形、最小凸包等^[1-8]。近年来, 随着并行计算技术的发展, 一些应用分组或分治策略的算法, 由于它们较容易采用并行计算中的“数据分组”策略, 因此, 有一部分算法很容易实现并行化, 如 Delaunay 三角网并行构建算法、Voronoi 图并行生成算法、平面散乱点集凸包并行算法等^[7-13]。由此可见, 点集空间划分是空间数据处理过程中的共用、常用过程, 探索实用、有效的划分方法; 同时, 空间数据划分方法与策略也是并行空间数据库设计的关键^[14], 具有一定的应用价值。

对散乱点集的空间划分, 目前专门研究其划分方法的成果相对较少, 更多地体现在具体的空间数据分析与处理问题中。Lee 和 Schachter 提出了单向划分的方法, 按照点集的 X 值进行单向排序、分割, 方法实现简单, 但容易形成狭长条带^[15]; Katajainen 和 Koppinen 按照二叉树方式

对点集进行空间划分, 划分子块的形状较整齐, 但二叉树叶子节点内的点分布不均匀^[16]; 胡金星等采用了如下方法: 按照先 X 方向、后 Y 方向的顺序进行点集的排序, 并进行精细分割, 如此反复进行, 保证了点分布的均匀性, 但反复排序降低了算法的时间效率^[17]; 为了更自由地划分空间, 且划分后各子块内点的数量较一致、划分后形态较优, 同时考虑到算法的执行效率, 本文设计了一种基于栅格统计和二叉树的自适应空间划分方法, 以期更好地应用于类似的空间分析算法中。

1 算法原理

点集自适应空间划分就是根据原始散乱点集的规模和空间分布特征将点集灵活地划分为若干子集, 各子集内点的数量应大致相同, 且各子集的空间形态较优。参考相关空间分析算法和其他划分方法中的自适应划分思路, 同时考虑到栅格辅助的方法能有效地提高空间数据分析效率, 借鉴文献^[18]中把栅格作为空间数据索引、快速检索空间数据的思路, 本文设计的方法如下。

图 1 为平面空间的散乱点集, 为对散乱点集建立空间索引, 需建立辅助栅格场, 首先把全图划分为 $M \times N$ 的栅格, 再对点集进行遍历, 计算每个点所属栅格的行列号, 并统计每个栅格内的点

数量,作为辅助计数。当散乱点恰好落在栅格线上时,其行(列)号取值统一设置为上(右)侧相邻栅格的行(列)号。由于点的分布并不均匀,所以每个栅格单元可能不包含点,也可能包含一个或多个点。

为了对点集进行划分,不妨先考虑如何对点集进行“对分”,因为更细的划分可以在折半对分基础上完成。如果范围“扁”,为了让划分后形态较好,选择在 X 方向“对分”,即针对列进行检索,检索前 K 列内的总点数是否达到点集数量的一

半,一旦达到,则划分。由于事先统计了每个栅格内点的数量,故随着列数的增加可较快速地动态更新前 K 列的总点数,不需对点集进行重新遍历。对上面的子集,采用同样的思路继续进行划分,直到达到用户所需要的块数为止。整个点集划分过程实际上是对栅格进行“空间划分”,再把栅格划分的结果,理解为点集划分的结果(图 2)。这种划分是对点集的粗糙划分,但粗糙程度可人为控制,当栅格单元足够小时,基本可看作精确划分。

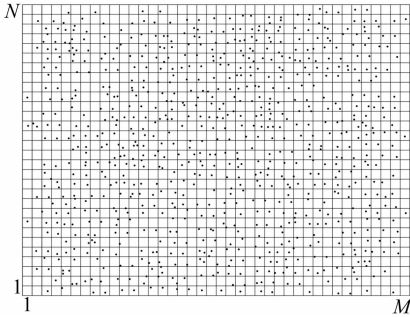


图 1 辅助栅格场建立

Fig. 1 Constructing Aided-grids

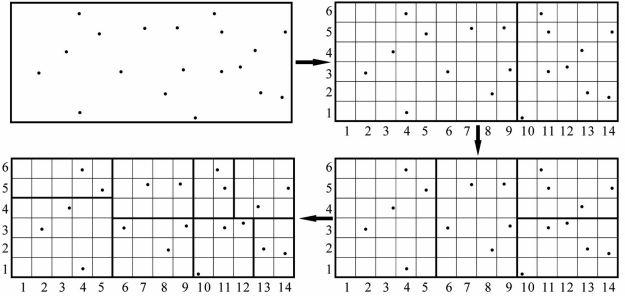


图 2 栅格辅助的自适应划分

Fig. 2 Self-adaptive Partition Based on Grids

2 算法实现步骤与方法

自适应划分过程相当于不断地对空间进行二叉划分,直到子块数量满足要求。在具体实现时,不妨采用二叉树数据结构进行相关信息的存储,在此基础上采用类似于构建二叉树的方法进行空间划分。假设给定平面散乱点集 $S = \{p_i = (x_i, y_i) | i = 1, \dots, n\}$,其中 p_i 表示点集 S 中的第 i 个数据点, n 表示 S 中数据点的个数,算法的目标就是根据数据点集的分布特征对空间点集进行划分,最终将数据划分为 $S = W_1 \cup W_2 \dots \cup W_m$,其中 W_i 为原始点集的第 i 个子块。为了得到最终分裂的状态,只需存储当前二叉树中的所有叶子节点,每一个叶子节点存储一个子块的相关信息。

2.1 数据结构

算法采用节点结构存储离散点,记录下各点的 x、y 坐标;并用辅助栅格场结构记录每一个栅格单元内点的总数及各点的序号;分裂得到的二叉树叶子节点采用链队列数据结构来存储,叶子节点记录子块的起始、终止行列号,以及子块内包含的点数和子块当前的分裂次数等信息。相关数据结构用 C 语言描述为如下:

```
struct PNT{ //点的数据结构
    Double x, y;}; //点的坐标值
```

```
struct GridField{ //辅助栅格场数据结构
    int pNum; //记录栅格内包含的离散点个数
    int * PointId;}; //记录栅格内包含的离散点的序号
struct BinaryTree{ //子块的数据结构
    int nStartLine, nEndLine; //记录块的起始、终止行号
    int nStartCol, nEndCol; //记录块的起始、终止列号
    int npNum; //记录块内包含的离散点个数
    int nDivision;}; //记录块当前的分裂次数
struct Queue{ //存储最终分块的链队列结构
    BinaryTree * treenode; //节点子块
    Queue * next;}; //链队列指针
```

2.2 点集预处理

以原始点集为基础生成辅助栅格场:构造原始点集的最小外包矩形,对最小外包矩形进行网格划分,保证每一个网格单元为正方形或近似正方形,这样划分的细密网格,如果散列点分布均匀,则可保证每个网格内的点数几乎一样。

完成栅格场构建后,需判定点集中各点与网格单元的包含关系。对点集中的离散点逐个进行判定,确定其所属的网格单元,根据最小外包矩形的顶点坐标、网格单元的大小以及各个点的坐标值可以很容易确定各个判定点所属网格单元的行列号,同时将离散点的序号存储在栅格场数据结构中,并统计网格中当前离散点的个数。当所有点判定结束后,可以确定每一个网格中包含点的总数,为后续的栅格统计作准备。

2.3 自适应分块

对点集进行预处理后,需对点集进行自适应空间划分。即基于栅格统计对辅助栅格场进行空间划分,这一过程是本算法的主体部分,其流程如图3所示,主要包括以下3个步骤。

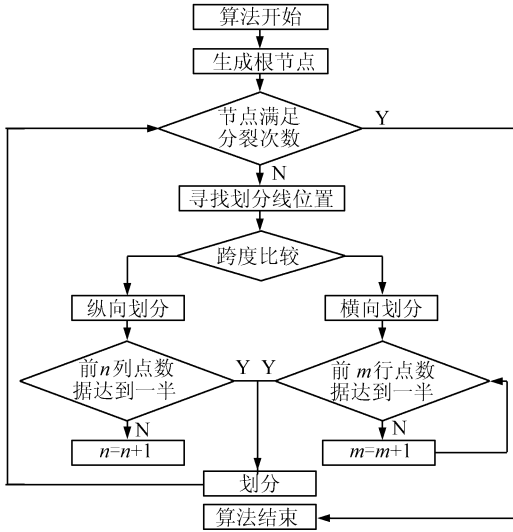


图3 自适应分块流程图

Fig. 3 Flow Chart of Self-adaptive Partition

1) 初始化生成根节点,此时根节点记录块的起始、终止行列号等于原始栅格场的起始、终止行列号,记录块中点的个数为原始离散点集中点的总数,当前的分裂次数为0,同时将根节点存储到队列链表中。

2) 依次从队列中取出一个节点元素,判定该节点元素的分裂次数是否等于设定的分裂次数。若等于,取出下一个节点元素继续进行判定,直到取完最后一个节点元素,划分过程终止;否则,转到步骤3)。

3) 对当前叶子结点进行分裂,也即对矩形范围进行划分,首先判定分块的方向是沿x轴还是

y轴:计算节点元素在x轴方向的跨度与y轴方向的跨度,并比较它们的大小。若在x轴方向的跨度大于y轴方向的跨度,则沿垂直于x轴方向划分;反之,沿垂直于y轴方向划分。将划分后产生的两个新节点添加到队列链表尾部,同时将原节点从队列链表中删除,完成后转到步骤2)。

分裂叶子节点也即确定划分线的位置,并存储新节点的相关信息。这里仅以沿垂直于x轴方向划分为例介绍划分的具体方法:设取出的叶子节点所代表的块如图4(a)所示,首先沿x轴逐列扫描辅助栅格场,用nCurrentTotal记录从最左列开始扫描到当前列所包含的离散点数量,用nextColTotal记录当前列的下一列所包含的离散点个数。若 $|pNumTotal - 2 * nCurrentTotal| < |pNumTotal - 2 * (nCurrentTotal + nextColTotal)|$,则当前列为找到的使之左右两块点数近似相等的列(图4(b));然后将该叶子节点分裂成两个新的节点,设置新节点的起始、终止行列号及其内包含的点数,并将新节点的分裂次数在原节点分裂次数的基础上加1。

3 实验与分析

为测试上述算法的划分效果及时间效率,笔者进行了若干实验。图5(a)为以中国水系图为基础经离散化后获得的实验点集,用该算法对实验数据进行自适应划分,将辅助栅格场行列数的初始设置值为 200×200 ,分裂次数设为8。从划分结果中可以看出,点密集的区域划分的子块相对较小,点稀疏的区域划分的子块相对较大,体现了自适应的划分思想,且划分后各个子块的形状也较合理,达到了划分的目的。为进一步观察划分效果,将划分结果进行局部放大,如图5(b)。

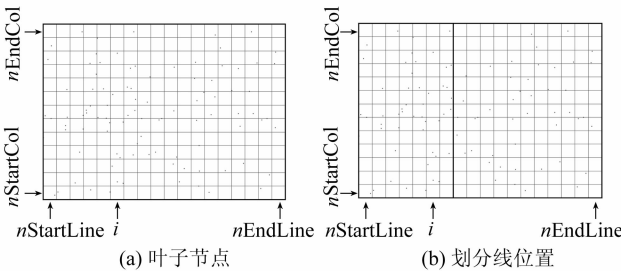


图4 节点分裂示意图

Fig. 4 Split of the Leaf-node

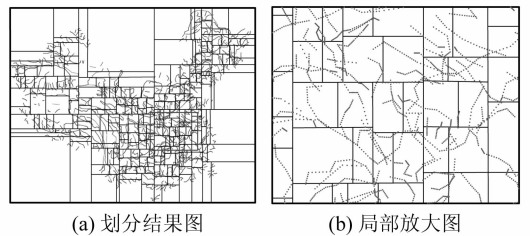


图5 测试实例

Fig. 5 Test Sample

测试过程中发现,当辅助栅格场的行列数设置为不同值时,划分的结果存在一定差异,这表明划分效果与辅助栅格场行列数的设置有关。行列

数越大,每个栅格单元越小,包含的点数也相对越少,由于各个子块是由若干栅格单元组成,故栅格单元越小,划分越精细。但随着栅格场行列数的

进一步增加,划分效果改善并不明显,且算法的时间效率会下降,因此栅格场的行列数设置也并非越大越好,需兼顾划分效果和算法的时间效率。当测试点集的数量分别为 10 万、50 万及 100 万时,分裂次数设为 12,在不同栅格场行列数情况下自适应空间划分的执行时间(Genuine Intel(R)

处理器,2.53 GHz)见表 1。

为进一步评估点集数量与自适应划分执行时间之间的关系,本文对数量为 10 万、20 万、50 万、100 万、200 万、500 万的点集进行了测试,将栅格场行列数固定为 200×200,分裂次数设为 12。测试结果见表 2。

表 1 执行时间与栅格场行列数设置的关系

Tab. 1 Relationship Between Process Time with Rows and Columns of Raster-field

点集数量 /个	栅格场行列数设置/(行×列)及分组算法执行时间/s							
	50×50	100×100	150×150	200×200	300×300	400×400	500×500	1 000×1 000
10 万	0.047	0.079	0.094	0.110	0.141	0.157	0.172	0.266
50 万	0.125	0.140	0.156	0.172	0.203	0.234	0.266	0.438
100 万	0.172	0.187	0.203	0.219	0.250	0.297	0.344	0.562

表 2 处理时间与集合中点的数量的关系

Tab. 2 Relationship Between Process Time with Point Number in Point-set

集合中点集数量 /万个	10	20	50	100	200	500
处理时间/s	0.110	0.140	0.172	0.219	0.313	0.625

从表中看出,该算法时间效率较高,将 500 万点划分为 2^{12} 即 4 096 组时耗时仅为 0.625 s,相对于很多空间分析算法而言,耗费时间几乎可以忽略;把分组作为这类空间分析算法的预处理部分,对很多空间数据处理过程而言,仅消耗了很少的时间,但往往可为后续处理节约大量时间。

4 结 语

随着空间数据采集技术的快速发展,GIS 应用中空间数据量日益庞大,对空间数据处理与分析的效率提出了更高要求,进行空间数据划分以提高后续空间数据处理与空间分析的效率具有理论和现实意义。同时,随着计算机软硬件的并行化技术快速发展,空间数据划分方法有望进一步应用于相关并行空间分析算法中。本文针对 GIS 中常见的点集数据,在分析现有的一些划分方法后,提出了一种基于栅格辅助的自适应空间划分方法,通过实例验证了该方法的可行性,且划分效果较理想;测试与分析表明,该算法的计算效率亦相对较高。借鉴本文方法,可望进一步研究线、面等其他类型要素集合的空间划分。

参 考 文 献

[1] Dwyer R A. A Faster Divide-and-conquer Algorithm for Constructing Delaunay Triangulations[J]. *Algorithmica*, 1987, 2(2): 137-151
 [2] Chen M B, Chuang T R, Wu J J. Parallel Divide-

and-conquer Scheme for 2D Delaunay Triangulation [J]. *Concurrency and Computation: Practice and Experience*, 2006, 18(12): 1 595-1 612
 [3] 李根, 邹志文, 鞠时光. 结合二叉树和 Graham 扫描技术的高效 Delaunay 三角网构建算法[J]. *计算机应用研究*, 2010, 27(3): 894-896
 [4] Setter O, Sharir M, Halperin D. Constructing Two-dimensional Voronoi Diagrams via Divide-and-conquer of Envelopes in Space[J]. *Transactions on Computational Science IX*, 2010, 6 290: 1-27
 [5] Aichholzer O, Aigner W, Aurenhammer F, et al. Divide-and-conquer for Voronoi Diagrams Revisited [J]. *Computational Geometry*, 2010, 43(8): 688-699
 [6] 张忠武, 吴信才. 平面海量散乱点集凸壳算法[J]. *计算机工程*, 2009, 35(9): 43-45
 [7] 余杰, 吕品, 郑昌文. Delaunay 三角网构建方法比较研究[J]. *中国图像图形学报*, 2010, 15(8): 1 158-1 167
 [8] 张三元, 马利庄. 平面散乱点集凸包并行算法[J]. *浙江大学学报(工学版)*, 1999, 33(04): 432-435
 [9] Guan Q F, Clarke K C. A General-purpose Parallel Raster Processing Programming Library Test Application Using a Geographic Cellular Automata Model[J]. *International Journal of Geographical Information Science*, 2010, 24(5): 695-722
 [10] 范刚龙, 陈媛媛. 海量数据 Delaunay 三角网并行构建及实时更新算法[J]. *武汉理工大学学报*, 2010, 32(21): 123-126
 [11] Kohout J, Kolingerova I, Zara J. Practically Oriented Parallel Delaunay Triangulation in E^2 for Computers with Shared Memory [J]. *Computers & Graphics*, 2004, 28(5): 703-718
 [12] Subrata R, Zomaya A Y. Some Results on the Computation of Voronoi Diagrams on a Mesh with Multiple Broadcasting [J]. *Journal of Parallel and*

- Distributed Computing, 2003, 63(12): 1 300-1 314
- [13] 郝小柱, 胡祥云, 戴光明, 等. 平面点集凸包的并行算法研究[J]. 计算机应用, 2005, 25(10): 2 462-2 464
- [14] 赵春宇, 孟令奎, 林志勇. 一种面向并行空间数据库的数据划分算法研究[J]. 武汉大学学报·信息科学版, 2006, 31(11): 962-965
- [15] Lee D T, Schachter B J. Two Algorithm for Constructing a Delaunay Triangulation[J]. International Journal of Computer and Information Science, 1980, 9(3): 219-242
- [16] Katajainen J, Koppinen M. Constructing Delaunay Triangulations by Merging Buckets in Quad Tree Order[J]. Fundamenta Informaticae XI, 1988, 11(3): 275-288
- [17] 胡金星, 马照亭, 吴焕萍, 等. 基于格网划分的海量数据 Delaunay 三角剖分[J]. 测绘学报, 2004, 33(2): 163-167
- [18] 王结臣, 陈焱明. 一种栅格辅助的平面点集最小凸包生成算法[J]. 武汉大学学报·信息科学版, 2010, 35(4): 403-406

第一作者简介: 王结臣, 博士, 教授, 现主要从事 GIS 理论与应用研究。

E-mail: hotmailwang@ yahoo. com. cn

An Algorithm for Self-adaptive Partition of Scattered Points in Planar Domain

WANG Jiechen¹ ZHANG Hui¹ WU Wenzhou¹ WANG Bao¹

(1 Department of Geographic Information Science, Nanjing University, 25 Hankou Road, Nanjing 210093, China)

Abstract: This paper presents a method to partition the scattered points in planar domain based on grid-statistics and adaptive binary-partition. The basic idea of the method is as follows. First, build a grid field which can cover over all planar points to create the spatial index of the points. Then, figure out the position of every point in the grid field, such as the row and column place in the field, meanwhile, reserve the number of points in each grid. Finally, set the number of iterations as the parameter of terminating to partition the scattered points with proposed rules. This method can divide the data point-set into several groups flexibly and the result is relatively balanced. Besides, the computational efficient of the algorithm is relatively high and not too much extra storage space is needed. The method proposed in this paper can be used in the existing spatial analysis algorithms using partitioning-strategy and can be further applied to the related parallel algorithms. It also provides reference for adaptive partition of other types of spatial elements.

Key words: scattered points; self-adaptive partition; algorithm; GIS

About the first author: WANG Jiechen, Ph. D, professor, majors in the theory and application of GIS.

E-mail: hotmailwang@ yahoo. com. cn

欢迎订阅 2013 年《武汉大学学报·信息科学版》

《武汉大学学报·信息科学版》即原《武汉测绘科技大学学报》,是以测绘为主的专业学术期刊。其办刊宗旨是:立足测绘科学前沿,面向国际测量界,通过发表具有创新性和重大研究价值的测绘理论成果,展示中国测绘研究的最高水平,引导测绘学术研究的方向。本刊为中国中文核心期刊, EI 核心刊源期刊。是国家优秀科技期刊和中国高校精品科技期刊,并获中国国家期刊奖,入选中国期刊方阵。

本刊主要栏目有院士论坛、学术论文、科技新闻等,内容涉及摄影测量与遥感、大地测量与物理大地测量、工程测量、地图学、图形图像学、地球动力学、地理信息系统、全球定位系统等。收录本刊论文的著名国际检索机构包括 EI、CAS、PUB 等,其中 EI 收录率达 100%,其影响因子长期名列中国高校学报前列。本刊读者对象为测绘及相关专业的科研人员、教师、研究生等。

本刊为月刊,国内外公开发行,邮发代号 38-317,国外代号 MO1555。A4 开本,128 面,定价 10 元/册,每月 5 日出版。漏订的读者可以与编辑部联系补订。