基于图模型的多边形自动并行构建算法*

陈占龙[1]，张丁文[1,2]，吴 亮[1]
(1. 中国地质大学 信息工程学院，武汉 430075；2. 教育部地理信息系统软件及其应用工程中心，武汉 430074)

摘 要：目前 GIS 基础算法并行化成为高性能 GIS 进一步深入的前提，作为 GIS 空间分析基础算法的重点，有必要对多边形构建提出一种自动并行算法。为此，提出基于图模型的多边形自动并行构建算法。该算法根据图模型中有向闭合环的特点对一组线段的集合进行多边形构建，能有效提高多边形构建的自动化程度。将搜索、排序等耗时较多的操作进行并行化处理，能有效减少全局搜索次数及整体排序和逻辑操作时间。实验表明，在对大规模线性数据生成区域时，该算法能有效地实现效率提升，达到良好的效果。

关键词：图模型；多边形化；并行；自动
中图分类号：TP301.6 **文献标志码**：A **文章编号**：1001-3695(2012)05-1634-03
doi:10.3969/j.issn.1001-3695.2012.05.008

Automatic parallel polygonization algorithm based on graph model

CHEN Zhan-long[1], ZHANG Ding-wen[1,2], WU Liang[1]
(1. *Faculty of Information Engineering, China University of Geosciences, Wuhan* 430075, *China*; 2. *Engineer Research Center for GIS Software & Application of Ministry of Education, Wuhan* 430074, *China*)

Abstract: Polygonization is always the emphasis in the basic algorithms of GIS spatial analysis. And the parallel processing of basic algorithms is the prerequisite for further progress of high performance GIS. For the computing requirements of parallel GIS, this paper put forward a parallel polygonization algorithm based on the graph model. The algorithm could effectively increase the degree of automation for polygonization, through constructed polygons according to the characteristics of graph model with directed rings. And it significantly reduced the number of global scan operations and the time of permutation and element-wise operations. The experimental result shows this algorithm has good ceffect for efficiency improvement when determining polygons formed by a large-scale line set.

Key words: graph model; polygonization; parallel; automatic

空间数据由空间对象构成，而空间对象由点、线、区域、矩形、平面、体积及包括时间的高维数据组成。空间数据的例子包括了城市、河流、道路、县城、州、作物覆盖面、山脉等横跨的区域的定位。这种数据对环境监测、太空、城市规划、资源管理及 GIS 的应用十分有效[1,2]。

前人在串行环境下对多边形的构建算法进行了很多研究，如闫浩文等人[3]于 2000 年提出了基于方位角计算的拓扑多边形自动构建快速算法。2005 年李大军等人[4]在此基础上又作出了改进。2002 年王杰臣[5]提出多边形拓扑关系构建的栅格算法。罗芳等人[6]则在 2004 年提出了闭合坐标链多边形数据的拓扑关系快速构建算法。陈占龙等人[7]在 2010 年提出了简单要素模型下基于单调链和 STR 树的多边形叠置分析算法[7]。而在并行环境下进行多边形构建主要有 Bestul 的研究[8]，在通用扫描模型[9]子集即数据并行 SAM 模型的环境下，他提出了若干的四叉树变种及怎样利用它们去构建多边形，同时展示了对多边形怎样在数据并行四叉树上进行一定次数的操作。在通用扫描模型的环境下，Lindenbaum 和 Hoel 等人[10,11]提出利用 bucket PMR 四叉树、R 树和 R + 树进行数据并行的多边形构建。

1 并行计算模型与线段求交点

本文使用的扫描模型[12]，对能在任意长变量(一维数组)的数据上操作的基本操作集合而言，被定义为并行计算的通用模型。三种基本操作(查找、向量积和排列)主要用来生成等长的结果向量，查找操作[13]需要一个结合算符⊕，一个输入向量$[a_0,a_1,\cdots,a_{n-1}]$，及返回一个结果向量$[a_0,(a_0\oplus a_1),\cdots,(a_0\oplus a_1\oplus\cdots\oplus a_{n-1})]$。

一个多边形至少由三个点组成，并且每一个点都与两条边毗邻。其生成过程不能凭人的简单的视觉感观，而需要通过良好的逻辑来对图作进一步多边形化。

如图 1 所示，平面上给定两个多边形 $A(A_1,A_2,\cdots,A_5)$ 与 $B(B_1,B_2,\cdots,B_4)$，它们的顶点按顺时针排列，多边形 A 和 B 分别组成平面图 P_1 和 P_2，其中 P_1 由多边形 A 的边界(A_1，$A_2,\cdots,A_5$)构成的边组成，P_2 由多边形 B 的边界($B_1,B_2,\cdots,B_4$)构成的边组成。确定多边形 A 和 B 的交，记为 $A\cap B$。$A\cap B=\{q\mid q\in A\&\&q\in B\}$，$A\cap B$ 的边界由 A 的边界和 B 的边界交错组合而成，A 边与 B 边的交点恰好是 $A\cap B$ 边界序列中 A 边与 B 边的交错之处。对 A 的每条边检查 B 的所有边，判断它们是

收稿日期：2011-10-10；**修回日期**：2011-10-19 **基金项目**：中央高校基本科研业务费专项资金资助项目(CUGL090251)；国家“十二五”科技支撑计划资助项目(2011BAH06B04)；国家教育部地理信息系统软件及其应用工程研究中心开放课题(20111109)

作者简介：陈占龙(1980-)，男，河北无极人，讲师，博士，主要研究方向为空间分析算法、高性能计算、地理信息系统软件开发与应用(chenzhanlong2005@126.com)；张丁文(1987-)，女，硕士研究生，主要研究方向为多核计算、并行计算、空间分析；吴亮(1976-)，男，副教授，博士，主要研究方向为网格计算、分布式计算、GIS 开发与应用.

否相交,若有交点则记录下来,求交的步骤如下:

a)通过边顶点的 x 值筛选不可能相交的线段,计算多边形 A 的边 $C_i = \overline{A_iA_{i+1}}$ 和多边形 B 的边 $D_i = \overline{B_iB_{i+1}}$ 的位置关系。若 C_i 在 D_i 左边,计算 C_{i+1} 在 D_i 的位置关系;若 C_i 在 D_i 右边,计算 C_i 在 D_{i+1} 的位置关系。

b)判断 C_i 和 D_i 是否有顶点重合,若有重合,记录线段 C_i 和 D_i。

c)通过边顶点的 y 值二次筛选不可能相交线段。若边 C_i 的最小 y 值 y_{pmin} 大于 D_i 的最大 y 值 y_{qmax},边 C_i 的最大 y 值 y_{pmax} 大于 D_i 的最小 y 值 y_{qmin},则两边不相交。

d)对 a)和 b)留下的可能相交线段求交点,根据线段的不同位置情况来计算交点。比较不同情况的交点方法如下:计算出点相对于线段的关系,再根据交点存在关系矩阵决定交点。

求 C_i 和 D_i 的交点算法
begin
求点 A_i、A_{i+1} 相对于 D_i 的关系
求点 B_i、B_{i+1} 相对于 C_i 的关系
if 点 A_i、A_{i+1} 不在边 D_i 上,and 分布在 D_i 两边
　begin
　if 点 B_i、B_{i+1} 不在边 C_i 上,and 分布在 C_i 两边
　　计算并记录交点
　else if B_i 在 C_i 上 and B_{i+1} 在 C_i 内侧
　　交点 $= B_i$
　else if B_{i+1} 在 C_i 上 and B_i 在 C_i 内侧
　　交点 $= B_{i+1}$
　end
else if A_i 在 D_i 内侧或外侧 and A_{i+1} 在 D_i 上
　交点 $= A_{i+1}$
else if A_{i+1} 在 D_i 内侧 and A_i 在 D_i 上
　交点 $= A_i$
end

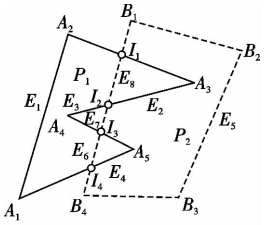

图1　相交的任意多边形

2　并行构建

在进行多边形构建以前,首先需要将输入的线生成图模型。按以下步骤添加边:将节点 0 和 1 添加到图中,再添加边 l_0 到图中,同时添加有向边 v_0、v_1,v_0 方向与原始边相同,v_1 方向与其相反,这样称 l_0 为父边,v_0、v_1 为子边,并且 v_0 和 v_1 互为兄弟边,其方向相反,添加第一条边完毕。依此添加余下的边,生成图后如图 2 所示。

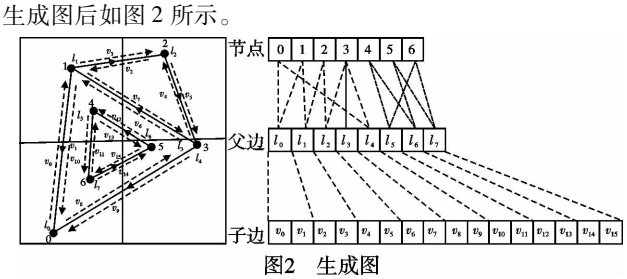

图2　生成图

2.1　记录边与边之间的关系

多边形上的点按照一定的顺序存储,对于构成多边形的边也需要按一定的顺序记录。本文在此处根据有向边的方向角大小对同一个节点的出边进行排序。以某有向边的起始节点为原点,水平方向为 X 轴,垂直方向为 Y 轴(与直角平面坐标系相同),以 X 轴正向为起始方向,逆时针旋转到由起始节点和此边第二个节点所构成的线段处的夹角即为该有向边的方向角。图 2 中节点 1 的各条出边的方向角,对其出边进行排列,结果为:v_3-v_7-v_1,放大后的效果如图 3 所示。

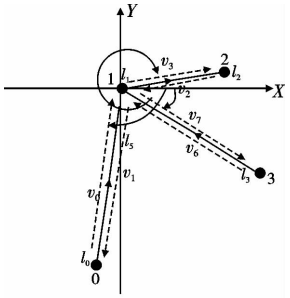

图3　节点1出边的方向角

对有 m 个节点的图的出边进行排序时,采取分组排序算法。为保证同一组内的节点在空间上互相邻近,图 4 中先对节点进行全局空间划分,然后根据空间位置对节点排序,再对排序后的节点进行分组就能在总体上保证组内的节点是空间邻接的。设有 n 个处理器($n \geqslant m$),根据处理器个数对节点分组后,节点的个数 $z = (z_1, z_2, \cdots, z_n)$,考虑到负载均衡,每一组的节点个数不一定均等,但总体数据量均衡,且同一组的节点在空间上相互邻近。

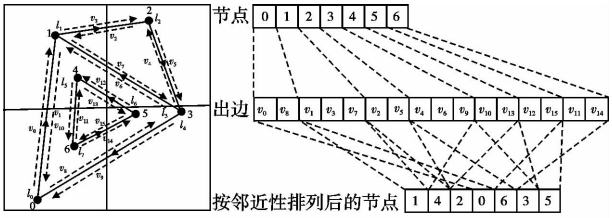

图4　按邻近性排列后的节点

对每个节点的出边按照方向角进行排序后,对于每个节点,边的存储关系主要为:对于节点 i,根据排序后的出边顺序取其出边 v_p 的兄弟边 v_q,设置 v_q 的下一边为围绕节点 i 排序的下一条出边 v_m,取 v_m 的兄弟边 v_n,依此设置边的存储关系,直到回到边 v_p。因此图 4 的有向边的存储关系如表 1 所示。

表 1　各子边的拓扑

编号	边的存储关系
1	$v_0 \rightarrow v_7 \rightarrow v_9 \rightarrow v_0$
2	$v_1 \rightarrow v_8 \rightarrow v_4 \rightarrow v_2 \rightarrow v_1$
3	$v_2 \rightarrow v_7 \rightarrow v_4 \rightarrow v_2$
4	$v_7 \rightarrow v_9 \rightarrow v_0 \rightarrow v_7$
5	$v_1 \rightarrow v_8 \rightarrow v_6 \rightarrow v_1$
6	$v_{10} \rightarrow v_{14} \rightarrow v_{12} \rightarrow v_{10}$
7	$v_{11} \rightarrow v_{13} \rightarrow v_{15} \rightarrow v_{11}$

2.2　结果多边形的计算

标记 label,根据边的存储关系将子边分为对应的几组输入向量,每组向量中的子边是唯一的,组间也不存在重复,每组向量由向量组的编号进行标记。根据组与处理器的个数,将向量组分配到各个处理器中。最后按照组编号,并行地执行向量积操作对各组中的子边进行标记。从 v_0 开始,$v_0 \rightarrow v_3 \rightarrow v_5 \rightarrow v_9$,标记这些有向边的 label 为 1;$v_1$ 开始,$v_1 \rightarrow v_8 \rightarrow v_6$ 标记这些有向边的 label 为 2,对于 v_2,$v_2 \rightarrow v_7 \rightarrow v_4$ 标志为 3;$v_{10} \rightarrow v_{14} \rightarrow v_{12}$ 标志为 4。标志完结果如图 5 所示。

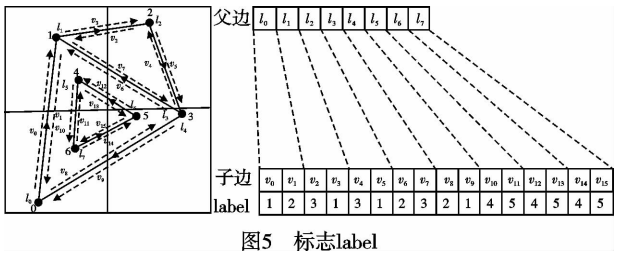

图5 标志label

标记每条子边的目的是得到特殊标记线,特殊标记线指的是属于同一条父边且具有相同 label 的子边,并将这些特殊标记线删除。删除标记线主要有三个步骤,都是并行扫描模型中的基本操作:a)向前独占扫描;b)逻辑减;c)排序。

2.3 生成有效环

根据边的存储关系生成环,闭合的环才是有效环,判断是否闭合主要是判断环的首边和尾边是否相同,首尾边相同的环即是闭合环。一个多边形至少有两个环,因此需要进行判断的环的数目至少是多边形数目的两倍,这一判断过程耗时较长。在并行构建中,根据可用处理器数目,将所有环分配到各个处理器中,每个处理器处理一组相邻接的环,并行操作使得判断时间可以大为缩减。最后再根据每组的处理结果,保留有效环,删除无效环。图 2 共生成六个环,执行判断操作后留下的环都是有效的,生成的有效环如图 6 所示。

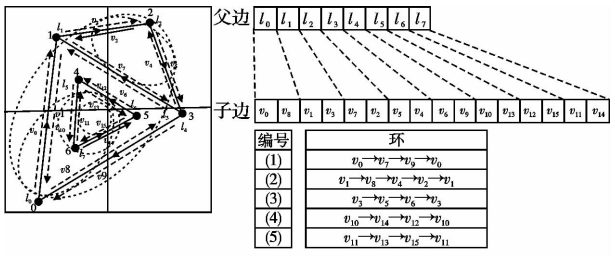

图6 生成有效环

构建多边形过程的最后一步是,根据有效环的的方向来判断该环是壳或者是洞,环的方向是逆时针就是洞,顺时针即是壳。此判断过程与判断有效环一样需要耗费大量时间,因此使用与上述判断同样的方法,根据处理器个数将环分组,然后对所有环组进行并发判断。上述有效环最后生成的洞和壳如表 2 所示

表2 洞和壳

壳	洞
$v_0 \rightarrow v_7 \rightarrow v_9 \rightarrow v_0$	$v_1 \rightarrow v_8 \rightarrow v_4 \rightarrow v_2 \rightarrow v_1$
$v_3 \rightarrow v_5 \rightarrow v_6 \rightarrow v_3$	$v_{11} \rightarrow v_{13} \rightarrow v_{15} \rightarrow v_{11}$
$v_{10} \rightarrow v_{14} \rightarrow v_{12} \rightarrow v_{10}$	

3 实验及性能分析

在保持正确性的前提下,多边形并行构建的最终目的是对一定规模的空间数据进行多边形构建运算时,运算速度可以得到显著的提高。表 3 描述了在不同的核个数下,对规模为 93 664个对象的线数据进行并行多边形构建的运算时间。

表3 不同核数量下的实验结果

核数	时间/s	线的个数	成区个数
1	30	93 664	34 123
2	22	93 664	34 123
4	16	93 664	34 123
6	18	93 664	34 123
8	18	93 664	34 123

通过改进计算的某一部分,所得到的性能提升可以用 Amdahl 定律定量地反映出来。Amdahl 定律定义了采用某种优化或者并行方案所取得的加速比。

$$\text{并行加速比(speedup)} = \frac{\text{并行前整个任务的运行时间}}{\text{并行后整个任务的运行时间}}$$

通过加速比公式,可以得到实验结果加速比的折线图如图 7 所示。

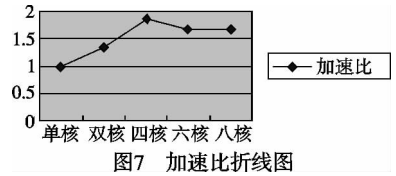

图7 加速比折线图

由折线图可以看到,在四核的情况下加速比是最高的,达到了 1.875,而在六核和八核的情况下,加速比降低到 1.66,这说明了并行中使用的核越多,额外开销也越多。但在总体上并行加速比是相当不错的,达到了提高效率的目的。因此,可以认为在并行条件下进行多边形构建是提高效率的有效途径。

4 结束语

本文提出了一种基于图模型进行多边形自动构建的并行算法,使得空间分析的基层运算效率得到提升。本文算法是将运算中对查询、排列等消耗时间的基本操作并行化以提高运行效率,在此基础上考虑空间数据分布的邻近性及并行节点间的负载均衡。为了求证以上并行构建的可行性和有效性,本文进行了测试实验,由实验结果可知基于图模型的多边形自动构建并行算法在效率上达到了预期效果,证明这一算法是可行的。未来的研究方向主要是增强扩展性和提高加速比,同时实现一些其他操作,如空间相交。

参考文献:

[1] GUTING R H, PAPADIAS D. Advances in spatial databases[C]// Proc of the 6th International Symposium on SSD. 1999.

[2] SCHOLL M. Advances in spatial databases[C]//Proc of the 5th International Symposium on SSD. 1997.

[3] 闫浩文,陈全功. 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] 李大军,刘波,赵宝贵,等. 拓扑多边形自动构建的一种改进算法[J]. 计算机工程和应用,2005,41(16):80-82.

[5] 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.

[6] 罗芳,艾廷华,王洪. 闭合坐标链多边形数据的拓扑关系快速构建[J]. 武汉大学学报:信息科学版,2004,29(6):558-561.

[7] 陈占龙,吴信才,吴亮. 基于单调链和 STR 树的简单要素模型多边形叠置分析算法[J]. 测绘学报,2010,39(1):102-108.

[8] BESTUL T. Parallel paradigms and practices for spatial data[D]. College Park: University of Maryland, 1992.

[9] BLELLOCH G E. Scans as primitive parallel operations[J]. IEEE Trans on Computers, 1989, 38 (11): 1526-1538.

[10] LINDENBAUM M, SAMET H, HJALTASON G R. A probabilistic analysis of trie-based sorting of large collections of line segments, CS-TR-3455[R]. College Park: University of Maryland, 1995.

[11] HOEL E G, SAMET H. Data-parallel polygonization[J]. Parallel Computing, 2003, 29(10): 1381-1401.

[12] BLELLOCH G E, LITTLE J J. Parallel solutions to geometric problems on the scan model of computation, AIM-952[R]. Cambridge, MA: MIT, 1988.

[13] SCHWARTZ J T. Ultracomputers[J]. ACM Trans on Programming Languages and Systems, 1980, 2(4): 484-521.