

批量遥感影像湖泊提取后的矢量拼接策略问题

沈占锋¹ 李均力² 夏列钢¹ 吴 炜³ 骆剑承¹

1 中国科学院遥感与数字地球研究所,北京,100101

2 中国科学院新疆生态与地理研究所,新疆 乌鲁木齐,830001

3 浙江工业大学计算机科学与技术学院,浙江 杭州,310023

摘要:在遥感影像湖泊提取结果进行矢量化之后,需要对多景影像的湖泊提取矢量数据进行拼接并形成湖泊信息专题图,不同的拼接策略直接决定区域湖泊数据拼接的效率。结合湖泊提取结果的矢量数据拼接与专题图制作问题,在分析矢量数据拼接原理与过程的基础上,首先分析了两景矢量数据的拼接方法与最佳策略,重点研究了面向大区域多景矢量数据的拼接策略,提出了两种实用的高效矢量数据拼接方法,即基于任务队列与基于间隔选择的策略,分析了不同策略的特点与适用情况。实际应用中,可以根据实际情况进行选择,较大程度地提高矢量数据拼接的效率。

关键词:矢量拼接;湖泊;策略;任务队列;间隔选择

中图分类号:P237.3;P208 **文献标志码:**A

在进行遥感影像的湖泊提取时,首先需要选择研究区某一时相的遥感影像,并基于一定的湖泊提取算法进行影像的湖泊提取,可以采用基于水体指数的“全域-局部”多层次信息提取与迭代模型实现区域湖泊的高精度自动提取^[1-2]。在湖泊提取结果矢量化之后,需要采取一定策略将湖泊提取的矢量结果进行拼接,形成该区域的湖泊信息专题图。不同的拼接方法可能导致不同的处理效率,特别是在对大区域实现遥感湖泊提取与快速制图时,策略的选择变得尤为重要。

目前针对拼接方面的研究虽然很多,但是专门针对矢量数据拼接方面的文献却很少,而且主要集中在地图分幅情况下的矢量拼接^[3-5]。ArcGIS 提供了矢量拼接工具,但其对于大区域的矢量数据拼接效率不高,且当数据量很大时容易出现数据处理错误的情况。本文结合湖泊提取结果的矢量数据拼接与专题图制作问题,在分析矢量数据拼接原理及过程的基础上,提出了针对两景矢量数据的拼接策略与实现方法,针对区域多景数据拼接的问题提出了两种实用化的数据拼接策略,并重点分析了两种策略的特点与适用情况,给出了其实现原理与流程。通过实验分析可知,两种方法均能够很大程度上提高拼接的效率,并各

自有着不同的适用情况,实际应用中可以根据实际情况选择不同策略,从而在更大程度上提高数据拼接的效率。

1 两景矢量数据的拼接方法及策略

1.1 湖泊提取过程与拼接问题

图 1 为基于遥感影像进行湖泊自动提取的流程图。在进行影像预处理(必要的辐射纠正、几何纠正)之后,以景为单位进行湖泊信息的自动提取,采用“全域-局部”的多层次信息提取模型进行算法实现。在进行遥感影像湖泊提取结果的矢量化之后,形成与各景影像相对应的湖泊矢量化提取结果(尽管矢量数据不具有“景”的概念,但为方便起见,在本文中称此遥感影像提取的湖泊矢量也为该“景”矢量)。多景矢量进行拼接时,应根据实际需要需要进行矢量数据的重投影工作,特别是在进行较大区域的湖泊提取过程中,为了减少对湖泊面积的影响,可以采用某种等面积投影方式实现(如 Albers 等面积投影)^[6]。投影完成后,利用一定的矢量数据拼接策略进行多景矢量拼接并形成湖泊专题图。

图 1 中,之所以在影像提取湖泊结果的基础上

收稿日期:2013-07-15

项目来源:国家 863 计划资助项目(2013AA12A401);国家科技支撑计划资助项目(2012BAH33B01,2011BAH06B02);NSFC-新疆联合基金资助项目(U1178302)。

第一作者:沈占锋,博士,研究员,主要从事遥感影像信息提取与分析、高性能影像计算等研究。E-mail: shenzf@irsa.ac.cn

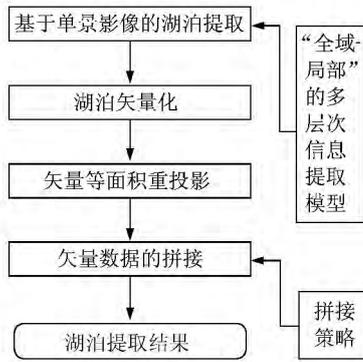


图 1 遥感影像湖泊提取流程及湖泊专题图成图过程

Fig. 1 Procedure of Lakes Extraction from Remote Sensing Images and the Generation of Thematic Map

先进行湖泊的矢量化拼接,而不是先进行栅格数据的拼接之后再矢量化,原因主要有两点:一是基于精度方面的考虑。在影像提取湖泊之后直接进行矢量化,基本没有精度方面的损失,矢量化结果同原始影像上的湖泊完全对应,再对相应的矢量数据进行等面积投影时,也可以认为矢量数据的等面积投影没有改变相应的湖泊的面积等属性(由投影性质决定)^[7],拼接之后相应的湖泊提取结果仍然具有面积可测量性;如果先拼接影像再提取湖泊,影像拼接过程中可能发生影像像素的重采样而导致湖泊像元属性发生变化,并最终导致提取湖泊矢量面积发生了变化,不符合精度要求。二是考虑计算机的处理能力,当区域范围较大时,进行大区域的影像拼接是不现实的,需要很大的系统开销(CPU、内存等),后续的矢量化过程也将由于数据量过大而变得困难。

图 2 示意了两景影像的湖泊提取结果在拼接时可能的几种情况。图 2 中右侧红色部分表示第一景的湖泊提取结果,左侧绿色多边形为第 2 景的湖泊提取结果(图 2 示意了右侧压左侧时的情况)。图 2 中,第 1 景和第 2 景进行拼接时,存在以下几种情况:①图 2 中的湖泊 a 仅存在于第 2 景,因此合并结果中该湖泊也直接来自第 2 景数据;②湖泊 b 在第 1 景中提取了半个湖泊,但在第 2 景中提取的为全湖,因此在没有外界干扰的情况下(如云、雪或影像时相、质量等),可以直接采用第 2 景的提取结果,也可以采用两景的拼接结果;③对于图 2 中的湖泊 c,由于在两景中均存在相应的完整提取的湖泊,可以综合两景影像的时相、成像质量与条件、提取湖泊的大小等因素决定采用哪一景的结果;④图 2 中的湖泊 d,由于在两景影像中的提取结果均为半个湖泊,因此必须对两景影像提取结果进行矢量拼接才能完成整个湖

泊的提取过程;同时,相应的湖泊的属性也将由这两景的共同属性合成(如该湖泊的时相需标记两景影像的时相,同理,其水体指数、面积、周长等属性),相应的拼接算法可由 GEOS 类库 GEOS_C. DLL 中提供的拼接函数实现。

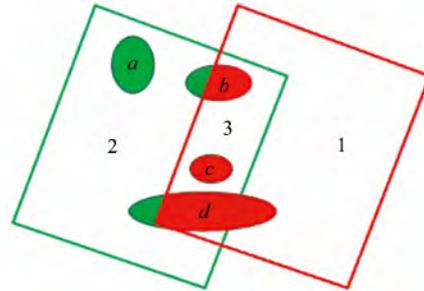


图 2 两景湖泊提取结果矢量拼接中的几种情况
Fig. 2 Several Cases for the Extracted Lakes Union from Two Scene Images

同样,当一个较大湖泊存在跨多景影像时(类似于图 2 中湖泊 d),需要综合这几景的矢量数据,拼接后方能得到最终的该湖泊提取结果,这种情况则更复杂。

1.2 矢量多边形的 Union 算法实现策略

GEOS 发布的类库 GEOS_C. DLL 提供了两个关于拼接的函数,能够实现矢量数据的高效拼接功能,其函数原型分别为 $OGRGeometry * Union(OGRGeometry *)$ 及 $OGRGeometry * UnionCascaded()$ 。其中函数体 $Union()$ 的功能是实现两个几何体的拼接,而 $UnionCascaded()$ 则是实现一个几何体内部的级联拼接。对于本文的湖泊提取与拼接来说,几何体 $OGRGeometry *$ 实际上指向为提取出的湖泊矢量多边形。图 3 示意了一种稍复杂的湖泊拼接情况。

在图 3 中,一个湖泊被两景影像分为几个部分,在第 1 景中为两个独立的湖泊,在第 2 景中为一个湖泊。只有在两景影像进行拼接时,才能判断出第 1 景的两个湖泊实际上同属一个湖泊(并需要在合并完成后更新相应的湖泊属性)。在采用 $Union()$ 函数对这两景湖泊的拼接过程中,如果 $OGRGeometry *$ 指向的为图 3 中第 1 景的湖泊 a (表示为 $OGRPolygon * a$),而其参数 $OGRGeometry *$ 指向的为第 2 景的湖泊 c ($OGRPolygon * c$),则在进行完该拼接任务后,还必须将其结果在这两景中进行进一步的搜索,若有同该结果相交的多边形,再继续拼接(比如本例中的 $OGRPolygon * b$),直到没有相交的为止。分析 $Union()$ 的实现过程可知,要进行两景数据中的两个矢量多边形联合,需要遍历这两景

中的所有多边形并进行判断,这会大大降低算法的效率。解决思路有两种:一是在两景数据中分别重新构造各景数据对应的 $OGRMultiPolygon^*$,各景中原来的 $OGRPolygon^*$ 分别加入到新的对象中去,其中在第 1 景中的 $OGRMultiPolygon^*$ 实际上为 $OGRPolygon^* a$ 及 $OGRPolygon^* c$ 的集合,第 2 景中的 $OGRMultiPolygon^*$ 实际上仅包含 $OGRPolygon^* b$ 。而在进行 $Union()$ 函数调用时,对应的两个几何体(一个为参数,另一个为函数主体)分别指向这两个 $OGRMultiPolygon^*$,因此在采用函数体 $Union()$ 进行多边形拼接时,对应的两个 $OGRGeometry^*$ 就可以实现对应的所有湖泊的拼接。二是将上述两景中的所有矢量多边形加入到一个新的 $OGRMultiPolygon^*$ 中,再采用 GEOS 提供的函数体 $UnionCascaded()$ 进行自身的级联 $Union$ 实现。

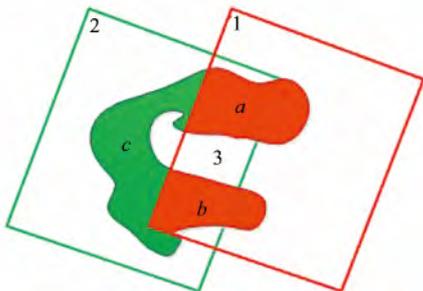


图 3 矢量拼接过程中的 Union 次序与策略

Fig. 3 Sequence and Policy for Two Vector Data Union

比较上述两种方法,二者的共同之处是都需要额外构造新的 $OGRMultiPolygon^*$,将待处理的矢量多边形的 $OGRPolygon^*$ 对象存储到新的 $OGRMultiPolygon^*$ 中,再将新的 $OGRMultiPolygon^*$ 传给 $Union()$ 或 $UnionCascaded()$,由函数体判断不同对象之间的关系并进行拼接;二者的不同之处在于函数体 $UnionCascaded()$ 通过内部封装的多边形间的关系判断机制来决定相互之间是否需要进行拼接(实际上也有一个内部的判断过程),函数效率相对较高。因此,推荐采用此函数进行矢量数据的拼接,实际实验中也发现此函数体的执行效率比 $Union()$ 的效率略高。

1.3 基于 UnionCascaded 的两景数据拼接策略

在采用 $UnionCascaded()$ 函数实现多个湖泊多边形内部的级联拼接时,尽管函数内部进行了各个多边形间关系的判断,但如果可以先确定某景内的湖泊多边形同最终的湖泊拼接结果的关系,并先对其进行归类,能减少算法内部需要判断的次数,并进一步地提高效率,采用的方法如图 4 所示。

如图 4 所示,对于两景数据的拼接,首先计算

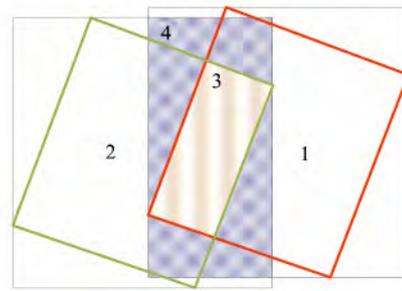


图 4 基于 UnionCascaded 的两景矢量数据拼接方法

Fig. 4 Two Scene Vector Data Method Based on Function Union Cascaded

出两景数据的公共区域,再遍历每景数据中的矢量多边形,判断其与此区域是否相交,如果相交,则将其归入待处理的 $OGRMultiPolygon^*$ 中;如果不相交,则直接将其归入拼接结果。理想的结果中,应该计算的公共区域为图 4 中区域 3,但实际应用中,最好采用图 4 中区域 4 示意的外接矩形进行计算($OGRGeometry::getEnvelope()$ 与 $OGRLayer::getExtent()$)。这是因为,采用外接矩形方法(图中区域 4)虽然可能加入一些不必要的矢量多边形(同其他多边形不相交),但这种方法(采用 $OGREnvelope::Intersect()$)判断的速度非常快,而基于矢量多边形的关系判断(采用 $OGRGeometry::Intersect()$)则效率相对较低,因此这种方法最终的效率仍高于采用多边形直接判断的方式,也能够一定程度上提高算法的效率。算法相应的 C++ 伪代码如图 5 所示。

```

1  OGRMultiPolygon to_union, result
2  vector<OGREnvelope> co_env
3  for layer(i) in two vector data
4  compute env[i] = layer(i).getExtent()
5  co_env.push_back(env[i])
6  for feature(j) in two vector data
7  if(feature(j).getEnvelope().Intersect(co_env))
8  to_union.addGeometry(feature(j))
9  else
10 result.addGeometry(feature(j))
11 to_union.UnionCascaded()
12 result.addGeometry(to_union)

```

图 5 基于 UnionCascaded 实现两景矢量数据拼接伪代码

Fig. 5 Pseudo Code of Union Two Vector Data by Union Cascaded

图 5 中的斜体为 OGR 提供的函数,其原理是在第 4 行获取 2 景数据的范围并在第 5 行计算其共同区域后,在第 7 行判断 2 景中的所有 feature 的外接矩形是否与此共同区域相交。如果有,则需要进行 $Union()$ (第 8 行);如果没有,则直

接加入结果集(第 10 行),在第 11 行采用级联函数 UnionCascaded()进行拼接之后,将其结果加入最终的结果集,程序结束。

2 多景矢量数据拼接中的策略问题

当一个区域由多景影像构成时,需要在上述算法的基础上,分别对该区域的多景湖泊提取结果进行拼接,获得湖泊提取结果。而相应的实现途径仍然是多景数据中进行两两拼接,这里首先以中亚地区巴尔喀什湖的 4 景数据拼接为例来讨论多景矢量数据拼接过程中的顺序与策略问题。

2.1 四景矢量拼接过程中的 Union 策略

图 6 示意了由 4 景 Landsat 影像提取的巴尔喀什湖的主体湖区及拼接方案。

图 6 中,巴尔喀什湖的主体湖区由 4 景 Landsat 影像提取并拼接后构成,4 景影像分别为 2009~2010 年的 8、9 月份影像,属于研究巴尔喀什湖水体的较好期影像,4 景影像中分别含有水体 16 329、1 989、7 126、4 984 个矢量特征,拼接后的 2010 年的整个巴尔喀什湖主体水体面积约 1.66 万 km²。在对每景影像进行湖泊提取后,分别采用不同拼接顺序实现整个湖区的拼接,其拼接效率的测试结果如表 1 所示。

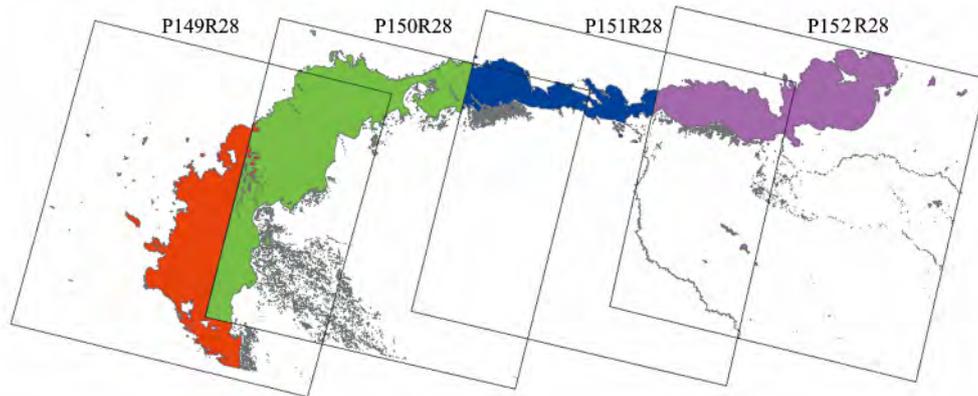


图 6 构成巴尔喀什湖的 4 景 Landsat 影像湖泊提取与拼接示意图

Fig. 6 Union Policy and Effect of Four Scene Vector Data for Balkhash Lake

表 1 巴尔喀什湖的 4 景湖泊提取结果的 3 种拼接方案效率对比表

Tab. 1 Efficiency Compare of Four-scene Landsat Extracted Vector Data Union of Balkhash Lake

编号	拼接策略	S1/s	S2/s	S3/s	最终耗时(串行)/s	最终耗时(并行)/s
方案一	(P149R28, P150R28)→S1, (P151R28, P152R28)→S2, (S1, S2)→S3	16.65	102.33	19.30	138.28	121.63
方案二	(P149R28, P151R28)→S1, (P150R28, P152R28)→S2, (S1, S2)→S3	10.40	5.55	181.02	196.97	191.42
方案三	(P149R28, P152R28)→S1, (P150R28, P151R28)→S2, (S1, S2)→S3	6.21	18.56	172.71	197.48	191.27

表 1 是采用 3 种不同的拼接顺序实现 4 景提取湖泊矢量数据拼接的效率对比表。从表 1 中可以看出,拼接策略 1 的方法最优,其实现策略中的 S1 为最左侧两景进行拼接,最右侧两景进行拼接,最后再对两个拼接结果进行拼接。在此过程中,由于 S1 过程与 S2 过程互不相关,可以采用并行计算的方式实现(可采用计算机的不同 CPU 核并行计算),能够进一步提高算法的效率,因此实际上第一种策略的串行耗时为 138.28 s,而如果采用并行方式进行处理的话则需要 121.63 s。方案二、方案三分别是拼接的顺序不同的测试结果,效率不如方案一。

进一步分析以上三种方案的差别。由于可以

采用并行计算的方式实现先期的两景数据拼接^[8,9],再对其相应的结果进行进一步的拼接,因此,在总体计算量相当的前提下,最佳方案应当尽可能地减少步骤 S3 的计算量(耗时),并尽可能地将步骤 S1 与 S2 进行平均,这是制定区域多景数据拼接策略的前提。从上面三种方案中也可以看出,方案一的步骤 S3 仅有一个公共小区域需要处理(即 P150R28 与 P151R28 的公共区域),可以认为公共区域的大小与其拼接的计算量成正比,而方案二的步骤 S3 需要处理 3 个公共区域,方案三的步骤 S3 需要处理两个公共区域,因此方案一是最优的。

2.2 基于任务队列的多景矢量拼接策略

根据以上数据拼接策略的制定原则进行推广,对于一个较大区域的湖泊来说,拼接过程类似 § 2.1 中多个拼接过程的共同完成。假设区域中有 n 景影像的湖泊提取结果需要拼接,按上述步骤将其分为 $\lfloor \frac{n+1}{2} \rfloor$ 份,分别进行两两拼接,再将其拼接结果进一步两两拼接,直到最后。过程类似于二叉树的实现过程,其树高为 $n \log_2 n$,如图 7 所示。每层内的拼接任务可以是并行的。下一层的任务需要依赖上一层任务的完成情况,如图 7 中的 $((1,2), (3,4))$ 任务必须要等待任务 $(1,2)$ 与任务 $(3,4)$ 完成后方可启动,因此还存在一个等待与轮循判断的过程。

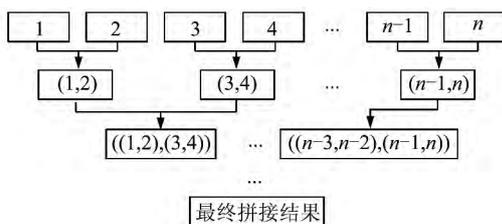


图 7 基于任务队列的多景矢量拼接策略流程

Fig. 7 Multi-scene Vector Union Method Based on Task Queue Method

实际应用中,可以按上述方法生成一个任务队列,再判断当前可用的计算资源,利用多进程或多线程的方式进行拼接任务的分配。这种方法能够充分运用当前计算机的计算资源,但任务启动的前提是该任务依赖的两个拼接任务已经完成。采用 `GetExitCodeThread()` 与 `GetExitCodeProcess()` 分别进行线程与进程进度的判断,如果完成,则继续分配任务;否则,轮循任务队列中的下一任务或等待,直到最后。

这种策略中还有一个关键性问题,就是任务队列的生成策略。类似于 § 2.1 所述,该队列的生成原则就是尽可能地减少后续步骤的计算量(耗时),并尽可能地将前一步骤的两个任务量进行平均。实际队列生成时,可以假定两景数据间的公共区域大小与该两景的拼接计算量成正比,首先计算研究区中相邻两景数据的公共区域,再将所有重叠区域按面积大小进行降序排列并加入待拼接队列,图 7 中的各层任务排序方案均按这种方式进行,直到拼接完成。实践与相关实验也证明这种策略是可选方案中的最佳策略。

2.3 基于间隔选择的多景矢量拼接策略

§ 2.2 中的基于任务队列的拼接策略的最大优点是能够充分利用当前计算节点的计算资源,

并采用并行任务队列的方式进行任务轮循分配。但该方法需要尽可能地平均一个拼接节点的两个分支上的计算量,以达到尽可能的任务均衡;同时,此策略将会产生一个较长的任务队列,队列在计算机上的任务安排及各任务的最优次序问题较难确定,且后续拼接步骤还需要等待前期拼接步骤完成,一定程度上也造成效率损失,因此本文在此基础上又提出了另一种基于间隔选择的拼接策略,如图 8 所示。

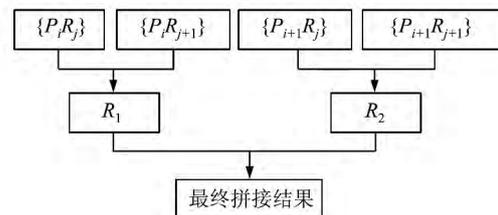


图 8 基于间隔选择的多景矢量拼接流程

Fig. 8 Multi-scene Vector Union Method Based on Interval Selection Method

对于按 Path 与 Row 编号的影像数据的矢量化结果而言,可以根据其空间关系将区域所有的待拼接矢量数据分成 4 组,如图 9 所示。首先,找到区域中的最小 Path 与最小 Row 的一景,记为 $P_i R_j$,然后将该景数据归至组 $\{P_i R_j\}$ 中,再按间隔选择的原则将 $P_{i+2} R_{j+2}, P_{i+4} R_{j+4}, P_{i+6} R_{j+6}, \dots$ 均归到此组;同理将 $P_i R_{j+1}, P_{i+2} R_{j+1}, P_i R_{j+3}, P_{i+2} R_{j+3}, \dots$ 归到组 $\{P_i R_{j+1}\}$ 中,再按图 8 两层拼接即可。

同基于任务队列的拼接策略相比,这种策略最大的优点是构造简单、步骤少且易于实现,而且不需要过多构造任务队列也不再需要过多的任务判断与等待,由于每一组数据(对应一个 `OGRMultiPolygon*`)内各景之间没有公共区域重叠,可由相应的数据直接加入,几乎不具有任何计算量(类似于一个 Merge 过程)。缺点是由于步骤少,仅第一层可以应用并行方法进行处理,如果计算机中有多个计算资源(如超过 2 个计算核或 2 个计算节点),无法有效利用。而且该方法也存在各组之间公共区域过多而导致的效率损失问题。

3 实验与分析

采用上述方法对北美阿拉斯加地区的一个 4×4 的 16 景影像的湖泊提取结果进行了拼接实验,如图 9 所示。在进行这些湖泊提取结果的拼接时有很多跨多景的半湖存在,图 9 右侧上下图

分别示意了一个湖泊存在于多景数据的情况,通过景间拼接完成整个湖泊的提取过程。

分别采用上述两种策略对该区域 16 影像数据进行拼接,对基于任务队列的策略来说,算法通过遍历该 16 景数据间的重叠区域判断并生成 15 个相应的拼接规则,相应的规则如下: $(M, N) \rightarrow S1$, $(A, B) \rightarrow S2$, $(P, O) \rightarrow S3$, $(C, D) \rightarrow S4$, $(E, F) \rightarrow S5$, $(I, J) \rightarrow S6$, $(K, L) \rightarrow S7$, $(G, H) \rightarrow S8$, $(S1, S3) \rightarrow S9$, $(S2, S4) \rightarrow S10$, $(S5, S6) \rightarrow S11$, $(S7, S8) \rightarrow S12$, $(S9, S10) \rightarrow S13$, $(S11, S12) \rightarrow S14$, $(S13, S14) \rightarrow \text{Result}$ 。对于基于间隔

选择策略来说,其实现方法比较简单。首先,基于间隔法将上述 16 景数据分为 4 组(图 9 中的红、绿、蓝、粉色),分别将各组构建一个 OGRMultiPolygon* 数据并生成 $S1, S2, S3, S4$,再采用以下规则: $(S1, S2) \rightarrow S5$, $(S3, S4) \rightarrow S6$, $(S5, S6) \rightarrow \text{Result}$ 。这种策略中的两景间的拼接方法均采用 § 2.3 中所述方法。

进一步地,采用类似的实验方法对阿拉斯加地区的 $3 \times 3, 4 \times 4, 5 \times 5, 6 \times 6$ 等分别进行实验,实验结果效率对比如表 2 所示。

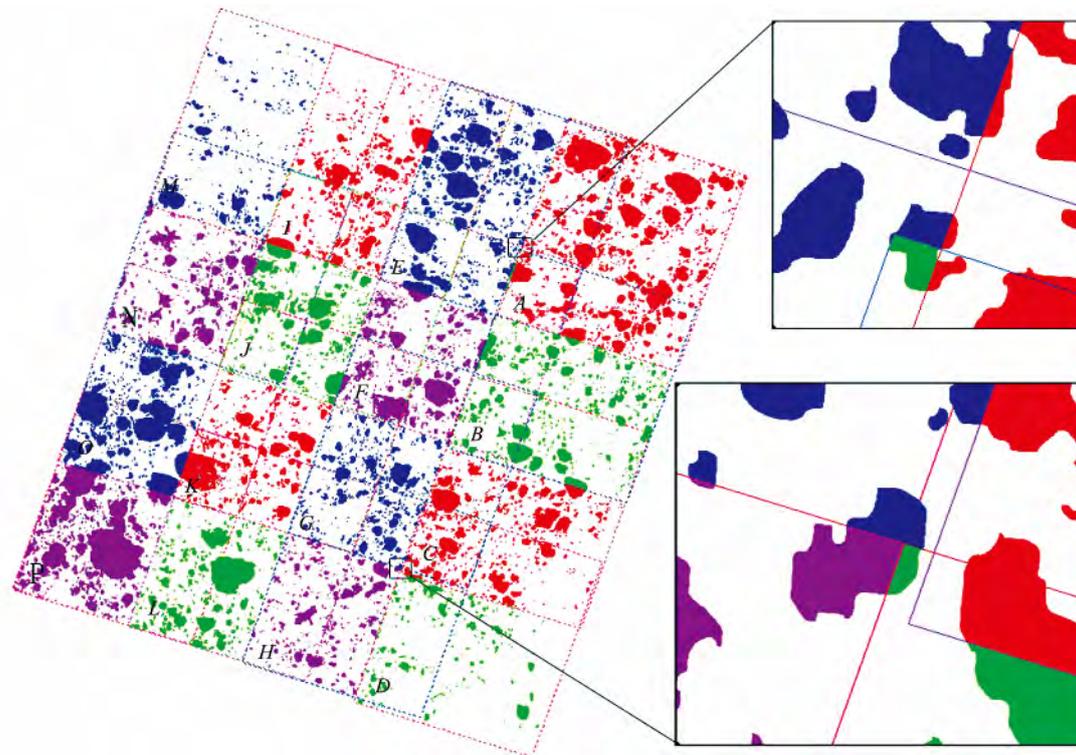


图 9 阿拉斯加地区 16 景 Landsat 提取湖泊结果的拼接示意图

Fig. 9 Union Effect of Sixteen Vector Data Extracted from Landsat of Alaska Region

表 2 两种不同策略在阿拉斯加地区拼接实验速度对比/s

Tab. 2 Speed Compare of the Two Policies in Alaska Region Applications/s

环境	3×3		4×4		5×5		6×6	
	任务队列	间隔选择	任务队列	间隔选择	任务队列	间隔选择	任务队列	间隔选择
E1(双核)	85.29	75.66	132.21	114.64	230.76	327.87	405.90	487.32
E2(四核)	67.03	62.29	90.53	134.64	170.12	275.22	315.25	412.77
E3(八核)	47.97	44.25	64.71	87.96	127.16	192.15	227.64	290.16

表 2 中,分别在三个计算机上进行了实验,其中 E1 为双核计算机, E2 为四核计算机, E3 为八核计算机。分析各组耗时时间可知,仅在 3×3 的数据测试中,间隔选择策略稍优于任务队列方式;其他情况下,当数据量较大时,基于任务队列的策略均优于基于间隔选择的策略。进一步拼接可以发现,当数据量较少时,间隔选择策略可略优于任

务队列策略(包括 $2 \times 3, 2 \times 4, 3 \times 4$ 等);但当数据量更大时,任务队列方式的效率更优一些;在实际数据处理时可根据相应的情况选择不同的策略进行实现。

分析这两种策略的原理,即可得出二者差别的主要原因:基于任务队列的方式在任务队列制定时较为灵活,能够将合并任务较重的任务放到

前面先计算;且由于大区域拼接时任务较多,可以更好地应用计算机的多核资源进行任务的并行处理,避开后续处理的大计算量;而间隔选择的原理较为简单,却能避免过长的任务队列,队列也将有一定的计算量损耗,而且不需要进行额外的任务等待,数据量不太大时具有一定优势;但当数据量更大时,由于这种策略的每一步的计算量都相对大些(实际上是将前面的合并任务推迟到后面的合并步骤),每景同其周围的几景之间的公共区域被“放大”了,而且无法同时应用更多的计算机资源(如多核),其优势就不再存在。

进一步地,将本文算法策略同商业软件 ArcGIS 的拼接相比,ArcGIS 提供的 Union 功能并没有真正实现几何体的拼接,而是将不同几何体之

间的重叠区域进行了去除,如图 10(b)所示,其结果是将巴尔喀什湖的主体湖区分成了若干份(如图 10(b)中标记①的分界线),这不符合我们湖泊拼接的要求,也无法对其进行属性赋值等操作。采用 ArcGIS 实现本文湖泊拼接时,需要首先将不同来源的矢量合并(Merge)至一个图层,再在此图层内进行 Union 操作,这样会浪费一个步骤并更加耗时,在遇到大数据量时容易出错。进行 4 景以上的 Landsat 影像数据提取后的湖泊结果的合并过程中,本文效率已经高于 ArcGIS 的效率,且数据量越大,本文算法的优势就越明显。而另一组实验中,当对 31 景以上的 Landsat 影像提取后的湖泊结果合并时,ArcGIS 在拼接过程中出现了算法问题而无法拼接,而本文算法工作正常。



图 10 本文方法及 ArcGIS 的 Union 方法针对巴尔喀什湖的拼接对比

Fig. 10 Union Effect Compare of this Paper with the Union() Function of ArcGIS

4 结 语

针对区域湖泊自动提取后的湖泊矢量多边形的拼接任务,本文在给出了两景矢量数据拼接策略的基础上,提出了面向矢量多边形高效拼接的两种拼接策略,即基于任务队列的拼接策略与基于间隔选择的策略,并在中亚及北美阿拉斯加地区的湖泊拼接中进行了实验。实验证明,基于任务队列的策略在数据量较大时能够充分发挥该策略并行的优势;而基于间隔选择的策略在数据量较少时具有较高的效率。这两种策略均能够应用并行方法提高拼接任务的效率,实际应用中可根据不同情况进行策略选择。同时,本文提出的针对矢量多边形的拼接方法也可推广至基于遥感影像进行多种专题信息提取与影像分类的结果拼接过程中。

参 考 文 献

[1] Li Junli, Sheng Yongwei, Luo Jiancheng. Automatic Extraction of Himalayan Glacial Lakes with Remote Sensing[J]. *Journal of Remote Sensing*,

2011, 15(1): 29-43(李均力,盛永伟,骆剑承. 喜马拉雅山地区冰湖信息的遥感自动化提取[J]. *遥感学报*, 2011, 15(1): 29-43)

[2] Luo Jiancheng, Sheng Yongwei, Shen Zhanfeng, et al. Automatic and High-precise Extraction for Water Information from Multispectral Images with the Step-by-step Iterative Transformation Mechanism [J]. *Journal of Remote Sensing*, 2009, 13(4): 610-615 (骆剑承,盛永伟,沈占锋,等. 分步迭代的多光谱遥感水体信息高精度自动提取[J]. *遥感学报*, 2009, 13(4): 610-615)

[3] Xiang Yimin, Wang Leqiu, Chen Juhua. Realization of Engineering Drawing Vectorization and Connection Technology[J]. *ACTA Scientiarum Naturalium Universitatis Sunyatsse*, 2003, 42(2): 113-115 (项益民,王乐球,陈炬桦. 工程图矢量化和拼接技术的实现[J]. *中山大学学报(自然科学版)*, 2003, 42(2): 113-115)

[4] Zhang Chengcai, Ji Guanghui, Chen Junbo, et al. A Method of Logical Join of Multi-sheet Digital Maps Based on ArcGIS and AutoCAD[J]. *Journal of Zhengzhou University (Engineering Science)*, 2007, 28(1): 118-121(张成才,季广辉,陈俊博,等. 一种基于 ArcGIS 和 AutoCAD 的数字地图拼接方

- 法[J]. 郑州大学学报(工学版), 2007, 28(1): 118-121)
- [5] Zhang Ying, Wang Rongfeng, Liao Xuejun. Research on Automatic Jointing of Massive Spatial Line Vectors[J]. *Journal of Computer Applications*, 2009, 29(12): 222-225 (张赢, 汪荣峰, 廖学军. 海量空间线矢量自动拼接研究[J]. 计算机应用, 2009, 29(12): 222-225)
- [6] Chai Huaqi, Yuan Ning, Zhao Xuejun. The Arithmetic Research of Digital Map Speediness Manage in Geography Information System Based on WGS-84 [J]. *Journal of Northwest University (Natural Science Edition)*, 2008, 38(2): 111-115 (柴华奇, 袁宁, 赵学军. 基于 WGS-84 的 GIS 数字地图快速处理算法[J]. 西北大学学报(自然科学版), 2008, 38(2): 111-115)
- [7] Rosa D, Plonka G. Uniform Spherical Grids via Equal Area Projection from the Cube to the Sphere [J]. *Journal of Computational and Applied Mathematics*, 2011, 236: 1 033-1 041
- [8] Shen Zhanfeng, Luo Jiancheng, Chen Qiuxiao, et al. Data Partition Policy of High-resolution Remotely Sensed Image Parallel Processing[J]. *Journal of Harbin Institute of Technology*, 2006, 38(11): 1 968-1 973 (沈占锋, 骆剑承, 陈秋晓, 等. 高分辨率遥感影像并行处理数据分配策略研究[J]. 哈尔滨工业大学学报, 2006, 38(11): 1 968-1 973)
- [9] Shen Z F, Luo J C, Wu W. A New Approach to Improve the Cluster-based Parallel Processing Efficiency of High-Resolution Remotely Sensed Image [J]. *Journal of the Indian Society of Remote Sensing*, 2012, 40(3): 357-370

Vector Data Union Policy of Lakes Extracted from Multi Remote Sensing Images

SHEN Zhanfeng¹ LI Junli² XIA Liegang¹ WU Wei³ LUO Jiancheng¹

¹ Institute of Remote Sensing and Digital Earth, CAS, Beijing 100101, China

² Xinjiang Institute of Ecology and Geography, CAS, Urumqi 830011, China

³ College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China

Abstract: Efficiency is an important issue that must be considered in the union process of the regional vector data. After the vectorization of the remote sensing images of lakes extraction result, the extracted lakes from multi-scene images should be merged to a lakes thematic map, and different union policies will directly determine the efficiency of the regional lake data union. Focused on the key technique of vector data union and lake thematic map production, this paper firstly analyzes the premium union policy for two scene vector data, then two different union policies for multi scene vector data are proposed, that is, the policy based on task queue and the policy based on interval selection. Furthermore, the characteristics of both policies are then analyzed and we can select different policies for the different situations so as to gain more efficiency.

Key words: vector data union; lakes; policy; policy based on task queue; policy based on interval selection

First author: SHEN Zhanfeng, PhD, professor, specializes in the remote sensing image information extraction and analysis, high-performance image computing. E-mail: shenzf@irsa.ac.cn

Foundation support: The National High Technology Research and Development Program of China (863 Program), No. 2013AA12A401; the National Key Technology Support Program, Nos. 2012BAH33B01, 2011BAH06B02; NSFC United Foundation, No. U1178302.