

基于 ArcGIS Engine 9.2 的图幅自动接边算法的实现

王永哲¹, 金永宝², 刘烜³ (1. 中南大学信息物理工程学院测绘与国土信息工程系, 湖南长沙 410083; 2. 湖北省长江武汉航道局航道测绘管理处, 湖北武汉 430014; 3. 浙江省杭州领图信息科技有限公司, 浙江杭州 311200)

摘要 随着信息技术的发展, 地理信息的存储介质发生了很大变化, 数字地形图作为计算机可识别存储介质上的有序集合, 仍沿用最初的分幅方式。这种基于分幅、相互分离的数字地形图存在以下弊端: 与无分割的实际地理空间实体不一致, 无法适应地理信息的完整性需求; 不便于数据库的分布式管理; 地形图的叠幅使得信息重复获取与存储, 产生裂隙和数据冗余, 致使资源空间在使用上浪费。基于 ArcGIS Engine 9.2 (相对于以前版本, 功能大大得到扩展) 的图幅自动化接边, 可以实现数字地形图无缝拼接, 解决了存在的弊端。

关键词 ArcGIS Engine; 图幅接边; 算法分析

中图分类号 S127 文献标识码 A 文章编号 0517 - 6611(2008) 34 - 15263 - 03

Algorithm Research for Automatic Edge Matching of Digital Maps Based on ArcGIS Engine 9.2

WANG Yong-zhe et al (School of Info-physics and Geomatics Engineering, Department of Geomatics, Central South University, Changsha, Hunan 410083)

Abstract The advantage of Geography information storage medium had very sweeping change along with the development of the information technology. The digital relief map as an aggregate could be distinguished by the computer, and continued to use the initial divide up way. These digital map's disadvantages which based on minute were as follow: It was inconsistent with the non-division actual geography space entity, also was unable to adapt the geography information demand completely. It was inconvenient to manage the database distribution. The repeatedly obtainment and storages of topographic fold characteristic caused the crevasse and the data redundancy. Using the automatic edge match of digital maps, matching the digital maps non-divisionally could be realized, these disadvantages could be improved.

Key words ArcGIS engine 9.2; Digital maps edge matching; Algorithms analysis

地理信息系统^[1-2]中的很多图形信息^[3], 空间地理数据都来源于地形图, 纸质地形图又是一种基本的空间数据源, 分幅的图幅设计方法人为地将连续的地表空间划分成若干个子空间, 分别分布于不同的图幅中, 地形图的物理分散性导致数字化后的逻辑分散性。为很好地解决这些问题, 图幅间各种图元的连接问题不可避免地产生: 在实际生活中, 人们往往将连续完整的等高线进行分割, 因此在 DEM 的制作及图属互查中, 分幅数字化后的等高线拼接问题, 就成了地理信息系统中基础数据处理方面的一个必须要解决的问题; 在数据的存储和管理上, 按图幅以文件形式来组织和存储空间几何数据, 造成不同的图幅之间存在缝隙。为此, 笔者对基于 ArcGIS Engine 9.2 的图幅自动接边算法的实现进行了研究。

1 研究意义

如果能够很好地实现图幅自动化接边的算法, 解决好上述问题, 则有着相当重要的意义: 在尽可能少的用户干预下实现接边工作的自动化, 使数字化制图人员通过操作计算机规则被计算机接受, 就可以实现图幅接边工作的自动化, 将会使处理效率大幅度提高; 在数据处理方面, 通过良好的算法, 将数据处理过程中的容差或处理过程的不一致而产生的缝隙删除, 从而减少边界处的图形误差, 提高基础数据质量; 表现出来的结果更符合用户的需求, 用户提出需要某区域的基础数据愿望, 系统自动提取该地区的图幅, 自动完成水平和垂直合并工作, 使后续工作更加易于扩展, 而不是仅仅限定在某一范围。

2 自动接边中的关键问题

2.1 地物匹配及地貌特征线的匹配 地物查找的内容应与接边相交的地物, 为保证查找及拼接的准确性, 查找是以

图层为单位依据空间属性进行的, 不同图层中的地物不进行接边, 只有相同图层中的同类要素地物才进行匹配。地貌特征线也需要在同一图层中时才可进行匹配。

2.2 属性的变化 经过接边处理后, 地物地貌的属性必然会发生变化, 面积、长度等是自动相加的, 保留累加值, 原先两个面积删除, 为更好地表现这些变化, 可考虑增加属性字段, 将接边前后发生改变的属性进行标注, 从而反映出这种变化。其他的属性保留一个即可, 因为属性是一样的, 只是因为图幅不能拼接的原因, 出现了 2 个相同的属性。

2.3 误差的处理 由于数字化本身存在误差, 从而使得本该拼接在一起的地物及地貌特征线拼接出现偏差。当相邻图幅在接边处的对应要素间距离小于 0.3 mm 时, 可不考虑误差影响, 直接进行接边; 当距离在 0.3 ~ 0.6 mm 时, 2 要素各自移动 1/2 距离; 当距离大于 0.6 mm 时, 则按一般制图原则接边, 作记录。

3 ArcGIS Engine 9.2 介绍

3.1 简要介绍^[4-6] ESRI 的 ArcGIS Engine 是一个用于建立自定义独立地理信息系统(GIS)应用程序的平台, 支持多种应用程序接口(Application Program Interfaces, APIs), 拥有许多高级 GIS 功能, 而且构建在工业标准基础之上。

ArcGIS Engine 是开发人员用于建立自定义应用程序的嵌入式 GIS 组件的一个完整类库。开发人员可以使用 ArcGIS Engine 将 GIS 功能嵌入到现有的应用程序中。ArcGIS Engine 包括 2 个产品: Engine 开发包, 是组件、APIs 和工具的集合, 是创建自定义的 GIS 和制图应用的工具包。Engine 运行时, 是为了运行自定义的 Engine 应用的可分发的 ArcObjects。

3.2 新增功能 可以说, ESRI 公司的 ArcGIS 9.2 系列产品相对 9.1 来说变化是很大的, 无论对开发人员还是用户来说, 这次变化使得应用都更加方便合理, 更人性化, 这里简要介绍 Engine 9.2 版本新增的一些功能。

ArcGIS Engine 9.2 加入了许多新的功能并且增强了其他功能,增加了许多新的类 Classes, 界面 Interfaces 和方法 Methods, 增加扩展, ArcGIS 追踪分析模块, ArcGIS 逻辑示意图生成扩展模块, ArcGIS 数据互操作扩展模块, 支持开发语言 Microsoft Visual Studio 2005 .NET。

- 3.3 支持文件格式 - GDB (file Geodatabase 和 Access 数据)
- Servers (SDE, IMS, Servers, Arc web)
- 文件格式 (Coverage, VPF, CAD)
- 已知扩展名的栅格数据

此外在编辑工具及功能方面改进比较大, 该文正是基于 ArcGIS Engine 9.2 的相关组件、类库提供的功能进行了接边算法的分析和实现。

4 算法分析

4.1 前提条件

(1) 相邻图幅的待拼接部分应无重叠、残余元素遗留等现象。

(2) 拟合并的所有基础地理信息要素分类分好, 按照基础地理信息要素分类与代码 GBT13923-2006 标准分类存储到 FileGDB 文件中。

(3) 属性数据是搜索同名要素的依据, 必须要正确而完备, 以保证逻辑链接和拼接后空间查询结果的准确性, 图幅之间的间隙在容差范围内。

(4) 几何数据要满足一定的精度要求, 如连续的等深线线划不间断或自相交等。

4.2 算法描述 基础地理信息要素分类与代码中图幅边框本身就是一个要素, 因此使用基于图幅边框的缓冲区对要求更新的面边界进行缓冲, 这样就使得单个边的算法在速度及效率上都得到很大优化。

图幅内保护诸多要素, 哪些要素是要合并的, 必须作一系列分析, 确定要合并后才调用相关函数进行合并操作。

程序中把最简单的限制条件放在整个限制规则范围的最外层, 即“当前要素 OID (主键, 具有唯一性) 必须小于比较要素 OID”放在最外层, 要素间的比较按照取出次序严格按照 OID 值递增的规则进行比较, 这样只比较图1 中的1-2、1-3、1-4、2-3、2-4 要素, 而不会再比较 2-1、2-2 要素, 因此减少了很多不必要的要素合并条件判断, 使大部分不满足合并规则的要素较早排除 (图1)。

次外层的限制规则: 判断当前要素和比较要素是否在接合线两侧, 如果 2 要素在接边线同侧, 说明无合并可能。

限制规则3: 判断当前要素和比较要素是否在合适的距离范围内, 如当前 2 要素的 Extent 是上下相离关系, 说明该 2 要素不具备合并可能性。

限制规则4: 判断属性特征是否相同或是否满足要求。满足合并要求, 将当前要素合并, 并清除原要素。

算法流程图如图2 所示。

4.3 代码实现

4.3.1 合并要素判断模块。

```
Public void findfeaturecanun (arraylist OID, listarray, ipolyline ppolyline, pppolyline, ifeaturelayer pfeaturelayer, in iverticalhorizontal)
```

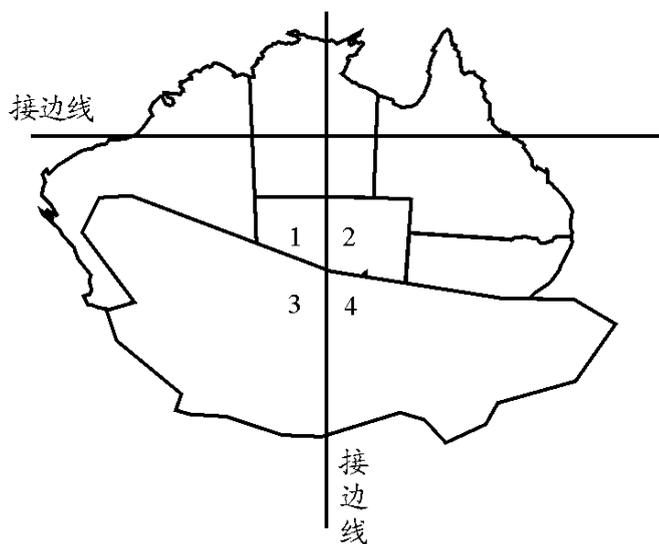


图1 按照 OID 值递增规则比较要素

Fig.1 The comparative factors according to OID progressive increase rule

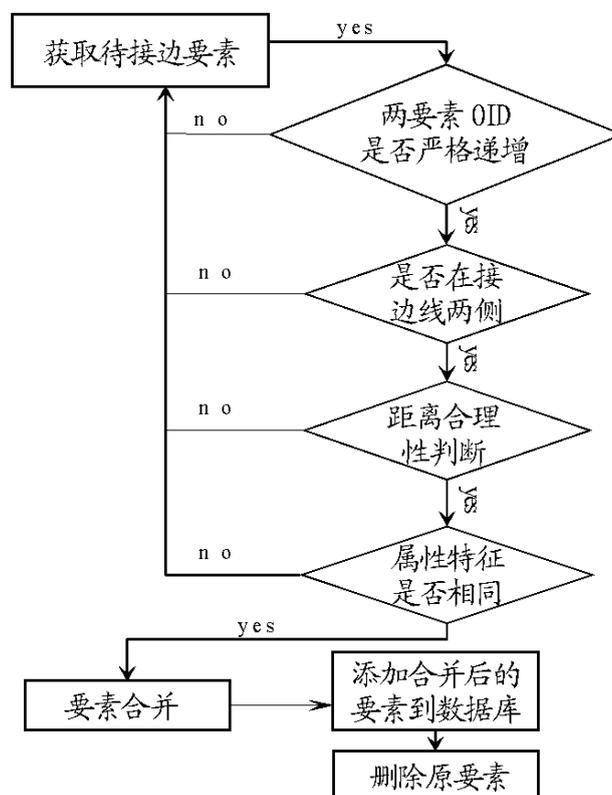


图2 算法流程

Fig.2 The algorithmflow

```
{
for (int inowfea = 0;
    inowfea <= ifeatureselectedcount - 1; inowfea++)
{
    OID1 = OID.listarray[inowfea];
    Featurenow = pfc.getfeature(OID1);
    For (int iconparefea = inowfea;
        iconparefea <= ifeatureselectedcount; iconparefea++)
    {
        if (iconparefea <= inowfea)
        // 判断条件1: 是否 OID 严格按小比大规则
        {
            continue; // Exit For
        }
    }
}
else{
    // 判断条件2: 是否在接合线两侧?
    If (iverticalhorizontal == 0) // 接边线竖向的情况
    {
        OID2 = OID.listarray[iconparefea];
        Featurecompare = pfc.getfeature(OID2);
```

