

文章编号:1672—7940(2005)03—0235—04

MAPGIS 区图元的物理重排序问题

胡 飞¹, 石瑞平², 陈建国¹

(1. 中国地质大学数学地质遥感地质研究所, 武汉 430074;

2. 长安大学, 西安 710000)

摘 要: 主要讨论了 MAPGIS 格式的全国地质工程数据库, 区文件中面积小的区实体常常被覆盖, 在视图中不可见的问题, 给实际应用中带来不便; 并较详细的论述了以 Visual C++ 6.0 为开发语言, 利用 MAPGIS 二次开发功能中提供的类库及 API 函数解决该问题的方法; 编写的转换程序在实际工作收到了较好的应用效果; 最后根据程序开发工作遇到的问题提出几点看法。

关键词: MAPGIS; 二次开发; 工作区; 实体

中图分类号: P628.4; TP311

文献标识码: A

收稿日期: 2005—01—21

PHYSICAL RE-ARRANGEMENT OF MAPGIS POLYGON FILE

HU Fei¹, SHI Rui-ping², CHEN Jian-guo¹(1. *Mathematical Geological Remote Sensing Institute, China University of Geosciences,
Wuhan 430074, China;*2. *Chang'an University, Xi'an 710000, China*)

Abstract: This paper mainly discusses the problem that the small area object of the area file in MAPGIS platform is often covered and can not be visualized. It also introduces the solution to the problem by using Visual C++ 6.0 as the developing language, the library provided by MAPGIS second development and API function. Furthermore, it describes the conversion program has a good effect in the work. Finally, some suggestions are made.

Key words: MAPGIS; second development; workspace; object

1 引言

MAPGIS 是中国地质大学(武汉)开发的一个工具型 GIS, 具有二次开发能力, 其二次开发接口为一组定义在空间模型之上的函数集, 借助于这组接口函数, 用户可在 BORLAND C++, VI-

SUALC++, VISUAL BASIC 等编程环境下, 建造面向特定领域的应用型 GIS。

2 问题的提出

中国地质大学数学地质与遥感地质研究所利用 MAPGIS 二次开发建立的“地质调查工作部署

专题图件空间数据库”, (目的是将中国地质调查局各专业部门原先不同底图、不同数据格式、不同投影、不同坐标等一系列的不同规格的数据图形进行统一化、系统化、规范化的建库和管理, 为中国地质调查局管理工作奠定一个较好的空间数据库基础。) 在实际工作中发现面积小的区实体常常被覆盖, 给工作带来不便。

MAPGIS 的区文件(.wp 格式文件) 中每个区实体的物理存取顺序是按照在作图时各区先后生成顺序^[1]。在读取区文件时, 如在同一坐标点上, 同时存在多个区实体时, 后读取的实体信息将覆盖以前的信息(如图 1), 在当出现面积小的 C 区和 D 区先生成, 并被后生成的 A 区完全覆盖时, 在视图上 C、D 两个区不可见。尤其在专题图件空间数据库处理中, 在同一地区通常有不同类型的多种数据叠加。要解决该问题就必须让 MAPGIS 先读取区文件中面积最大的区实体, 而 MAPGIS6.5 和以前的各版本中并未提供该功能, MAPGIS 提供的排序功能只是逻辑排序, 并不改变区实体在区文件中的物理位置。

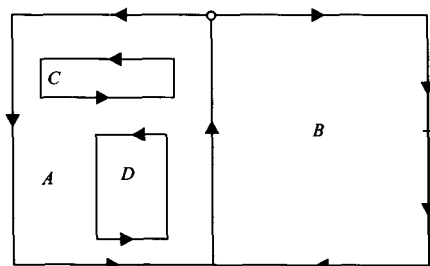


图 1 小实体被覆盖示意图

Fig. 1 The small entity was overlay

为解决上述问题可以利用 MAPGIS 的二次开发功能, 转换区文件中区实体的物理存取顺序, 使之能按面积大小降序排列。下面对定制开发过程中遇到的一些问题, 结合实际应用进行分析。

3 开发中的若干问题

MAPGIS 用于二次开发的组件产品称为 MAPGIS SDK (MAPGIS 软件开发包), 在这个开发包里, 包括 API 函数、MFC 类库、ActiveX 控件。开发包被封装于若干动态链接库(DLL)中。各类开发接口提供了从最基本数据单元的读取、保存、更新和维护到 MAPGIS 地图库的建立, 以

及空间分析, 图象处理等一系列功能。

3.1 MAPGIS 的数据管理

MAPGIS 数据管理的核心就是对工作区的操作, 它可以被看作一个容器(数据池), 里面存放实体的空间数据、拓扑数据、图形数据和属性数据, 每个工作区都对应于一个 MAPGIS 数据文件。工作区分为点、线、区、网、表五种类型。MAPGIS 数据操作的基本单位是空间实体, 也可以称为空间对象(点、线、区、网、属性记录)。在一个工作区中, 存放许多空间实体的个体, 和其他 GIS 数据格式一样, 每个个体都有唯一的编号, 称为实体号(点号、线号、区号、网号、记录号), 对实体数据的存取主要依据实体号。工作区和实体类型的对应如表 1 所示。

表 1 各工作区中可以包含的实体类型

Table 1 In each workspace the entity type that can include

工作区类型	实体类型
点工作区(.WT 文件)	点(PNT)
线工作区(.WL 文件)	线(LIN)结点(NOD)
区工作区(.WP 文件)	线(LIN)结点(NOD)区(REG)
网工作区(.WN 文件)	线(LIN)结点(NOD)网(REG)
表工作区(.WB 文件)	无空间实体, 仅有表格记录

在实际工作中打开的一个工程文件会含有多个工作区(同时打开多个文件), 以下代码^[2]实现对所有已打开的工作区操作:

```
short Ai, an, i;
```

```
CMyApp * pApp = (CMyApp *)  
AfxGetApp();
```

```
AN = _GetAreaNumber(pApp -> m_  
hInst); //取得工作区数
```

```
Ai = _GetFirstAreaNo(pApp -> m_  
hInst); //Ai 为第一个工作区
```

```
for(i=0; i<an; i++, Ai=_GenNextAreaNo(m_hInst)) //遍历各工作区
```

```
{ ...//对工作区 Ai 操作 ...}
```

3.2 区实体的数据组织

由于区对象比点、线对象复杂很多, 因此在利用 MAPGIS SDK 开发前要对其数据组织有深刻的了解。区实体由线实体围成, 下面举例说明。

现有 1, 2, 3 三条弧段, 三条弧段的方向如下图所示, 在弧段 1 上有五点: (200 200), (100

200), (100 150), (100 100), (200 100);在弧段 2 上有三点:(200 100), (200 150), (200 200);在弧段 3 上有五点:(200 200), (300 200), (300 150), (300 100), (200,100)。

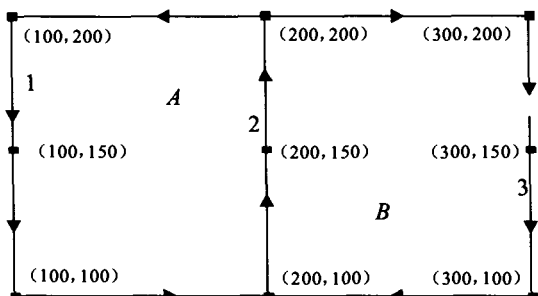


图 2 区实体示意图

Fig. 2 Area entity sketch map

弧段 1 的空间数据如表 2(记为 L1):

表 2 弧段 1 的空间数据存储结构

Table2 Saving structure of space data of Arc 1

(200,200)	(100,200)	(100,150)	(100,100)	(200,100)
-----------	-----------	-----------	-----------	-----------

弧段 2、弧段 3 的空间数据记为 L2、L3,A 区的方向如图箭头所示,则区实体 A 的空间数据如表 3,其中第一个单元存放的是全部线上的坐标点和区域边界的圈数。如圈数大于 2,边界各圈线号用 0 相隔。

表 3 区实体 A 的空间数据存储结构

Table 3 Saving structure of space data of the area entity A

$d1+d2+d3+d4+d5+1$	L1	L2
--------------------	----	----

属性记录:各类实体对应的属性记录结构如表 4:

表 4 属性记录结构

Table 4 Record structure of attribute

Flag	字段 1:ID	字段 2	...	字段 n
------	---------	------	-----	------

Flag 字段 1;ID 字段 2...字段 n 其中 Flag 是删除标记(0:存在,1 已删除),字段 1 一般是 ID 号,但 ID 号与物理存储顺序没有任何关系,该字段可以修改,甚至可以删除该字段,在区实体中一般字段 2 是“面积”,字段 3 是“周长”,“面积”与“周长”这两个字段是不能修改的。其它字段可修改、追加、删除^①。

如果要修改区文件中区实体的物理记录顺序,就不仅要修改区实体空间数据,同时也要修改该区实体的属性数据。因为属性数据记录了该区实体的各种属性(除 ID、面积、周长外可能还有地质年代,矿点数等重要信息)。

3.3 程序实现

编程思想:新开一个工作区,按区实体面积降序拷贝该区实体的空间数据和属性数据,另存新工作区为一个 WP 区文件(排序后的区文件)。关闭新开的工作区。流程如下:

1)求工作区总数 S 和第一个工作区的区号(见 3.1 代码)。

2)循环判断为真,打开第 i 个工作区,先求区实体的总数 n,按物理顺序读出“面积”到数组 A[](堆内存);循环判断为假结束。

3)对面积数组排序。按面积降序排列读取工作区 i 中的区实体的数据,并写入到新工作区 Ai 中,循环至读完所有的区实体。

4)保存新工作区 Ai 为新的区文件。关闭工作区 Ai,释放工作区 Ai(一定要释放)。

5)返回操作 2,执行下一个工作区(区文件)的转换。

核心代码如下:

.....

```

_GetRegNum(Ai, &i, &n); //区实体数
double * Area; //定义动态数组
long * No_Id; //按面积排序后的序号数组
if ((Area = new double[n]) == NULL)
return; //分配堆内存
if ((No_Id = new long[n]) == NULL)
return;
for(i=1; i<=n; i++)
{
    if (_GetAtt(Ai, REG, i, &stru, &att)
    <=0) continue; //读入区实体属性数据
    Area[i-1] = _GetFld(att, stru, "面积", NULL, NULL); //将面积读入数组
    No_Id[i-1] = i; //以该实体的物理序号赋值初值
}
Sort(Area, No_Id, n); // 排序 入口:面
    
```

① 曾文. 地理信息系统应用软件开发. 中国地质大学(武汉)信息工程学院, 2001.

积数组,区实体数;出口:序号 No_Id;

```

.....
AREA_HINST AHInst_new; //开新的工
作区
short Ai_new; //新工作区号
.....
for (i=1; i<=n; i++)//区实体数据拷
贝
{
    ri = No_Id[i-1]; //应该读取的 Ai 区
实体号
    lst[1] = ri;
    _CopyListReg(Ai, lst, Ai_new, 1);
    .....
}
.....
_SaveFileAs(Ai_new); //另存新工作区为
文件
_CloseArea(Ai_new); //关闭该工作区
_FreeWorkArea(AHInst_new); //释放该
工作区
.....

```

3.4 几个注意点

1)排序,在排序中要求输入面积数组,输出排序后的降序序号,即输出的是区实体在新的区文件中的物理位置。在区文件中常常会有面积相同的区,尤其在网格化处理后,很多区面积相同,但输出时序号不能相同。笔者的解决方法是:为面积数组同时定义一个序号数组 No, No[i]的值为该区实体的物理序号,在排序时同时对 No 数

组做交换 swap()。

2)区实体拷贝,在拷贝区实体时同时拷贝区实体空间数据和属性数据^[3],不然生成的新区文件没有意义。

3)一个工程文件内可能有区文件、线文件、点文件、网文件和表文件,但只有区文件里有区实体(REG),程序不会对其它类型文件做转换。在 MAPGIS 中每个文件对应一个工作区,只有在该工作区处于打开或编辑状态下,才能对它执行操作,所以要先检查区文件所在的工作区状态。

4 结 语

笔者已将上述按面积大小进行区图元的物理重排序功能应用于“基于 MAPGIS 的全国地质工作程度数据库应用服务平台”中,取得了较好的效果。

参考文献:

- [1] 吴信才.地理信息系统原理方法及应用[M].武汉:中国地质大学出版社.2000.
- [2] 武汉中地信息工程有限公司. MAPGIS 二次开发帮助文件 [EB/OL]. <http://www.mapgis.com.cn/>. 2001.
- [3] Chen J, Chen Z, Wang Q, et al. Spatial Databoue Mangement System of China Geological Survey Extent [J]. Journal of China University of Geosciences. 2003, 14(3):250-256.