

# 空间数据捕捉与移动方法研究

张成才, 张 昂, 王晓蕾

(郑州大学 水利与环境学院, 河南 郑州 450001)

**摘 要:**空间数据编辑功能是地图制图中的重要组成部分,当进行多个空间数据移动以及进行结点操作时,会影响地图精度。由此,依据面向对象的设计思想,通过 Windows 中 GDI 图形函数设计了空间数据移动与捕捉算法。该算法采用 GDI 函数建立的 MO 空间数据集图像缓存区,既可以实现多个地物的移动操作功能,也可通过建立屏幕光标矩形搜索区,进行结点位置的搜索,实现结点捕捉功能。该算法解决了在移动与结点捕捉过程中产生的拓扑关系错误,应用于实际的工程中,可以极大地提高数据编辑效率和地图制图精度。

**关键词:** MapObjects; 图形编辑; 移动; 捕捉

**中图分类号:** TP391

**文献标识码:** A

**文章编号:** 1673-629X(2011)05-0131-03

## Spatial Data's Snapping and Moving Method Research

ZHANG Cheng-cai, ZHANG Mao, WANG Xiao-lei

(Department of Hydro Informations Technology, Zhengzhou University, Zhengzhou 450001, China)

**Abstract:** Spatial data editing is an important part of cartography, and spatial data moving and node operation may affect the accuracy of the map. According to object-oriented design, using Windows GDI graphics functions to design an algorithm of spatial data capture and moving. This builds a spatial dataset of MO image cache of multiple features to complete mobile operation and establishes the rectangle area of the screen cursor in order to search the nodes which complete the node snapping function. The algorithm is an effective solution to decrease the topological errors of snapping and moving, increasing efficiency of data editing, and improving the accuracy of cartography in application of engineering.

**Key words:** MapObjects; graphic editing; feature moving; snapping

## 0 引 言

空间数据编辑功能是地理信息系统开发中的重要组成部分,在常用的 GIS 开发工具中 MapObjects 的数据处理基本功能齐全,但是扩展功能较弱,如缺少多个地物的移动与结点捕捉功能等。MapObjects 中地图编辑功能原理是将表示空间地物的点坐标等存储在数据集中<sup>[1]</sup>,通过数据集中数据的改变,从而实现地物空间形状的改变。

首先,由于 MapObjects 编辑的 shape 文件采用二进制数据存储格式<sup>[2]</sup>,在编辑过程中操作对象是 MapObjects 组件中的数据集<sup>[3]</sup>(MapObjects2. Recordset),利用 MO 自身提供的函数接口,只能进行单一地物要素的移动,不能同时进行多个地理要素的移动。在实际应用中,仅仅通过一个一个地移动地物,会导致地物之间的位置关系发生改变,影响地图制作精度和开发效率。

其次,一般捕捉是指获取某一模糊公差范围内某一具体结点的坐标值<sup>[4]</sup>,但是空间数据具有地理坐标,捕捉的容差大小设置需要进行转换,制约着结点的一般捕捉功能。由于多种原因存在误操作,如测量,数字化过程中的误差,导致录入的空间地物不正确,需要对地物进行修改操作;如两条弧段的拼接,实现点要素在弧段上的拓扑关系,都需要进行结点的捕捉,同时由于该功能是编辑中的基础功能,使用频繁,从而提高了捕捉功能的实现难度。

## 1 编辑功能实现的技术与方法

为了实现空间数据的编辑功能,在研究过程中使用了面向对象的设计思想<sup>[5]</sup>,设计了空间数据的编辑类对象。通过空间数据的编辑类的设计对象,以及实现的方法,编写常用的空间编辑函数,并将这些函数封装在对象中。利用这些函数实现了地物编辑过程中的诸多功能,如:实现地物的添加、修改、删除、移动,以及空间地物结点的修改、删除、移动等,利用可视化界面的交互操作,实现了对 shape 文件的数据编辑<sup>[6]</sup>。但是,在实现结点操作的过程中,需要设计结点的捕捉操

收稿日期:2010-11-08;修回日期:2011-01-19

基金项目:河南省科技创新人才计划项目(10410050003)

作者简介:张成才(1964-),男,教授,博士,研究方向为水利信息化。

作对象,在该对象中封装了结点捕捉的相应操作功能,如结点的捕捉。

编辑过程中通过可视化界面人机交互操作,首先进行空间数据操作类型的判断,明确正在使用的操作类型,如确定地物的添加、地物的修改等操作,并返回 MO 编辑类中操作枚举类型。其次,进行捕捉操作的判断,根据判断结果进行相应的功能操作。最后将操作结果保存在 shape 文件中,同时利用 MapControl 控件显示编辑后的结果。在编辑过程中,捕捉功能贯穿于整个操作过程中,是实现编辑功能的重要组成部分。实现流程如图 1 所示。

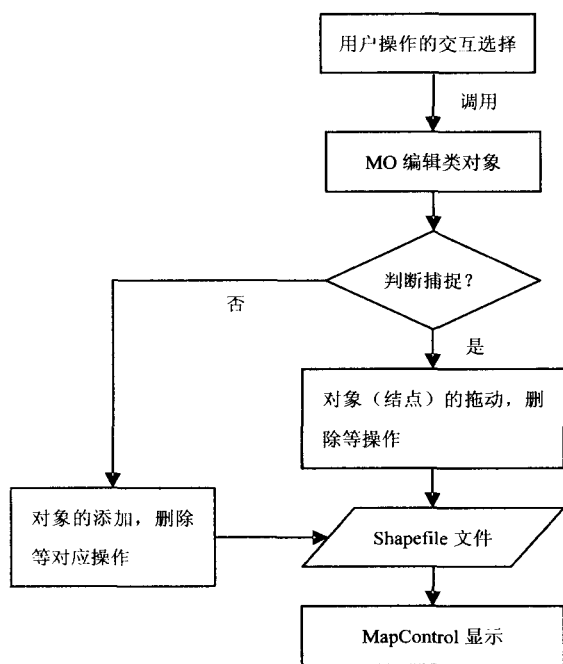


图 1 MO 编辑功能的流程图

## 2 空间数据编辑实现方法与实验结果

### 2.1 MO 捕捉类的设计与实现

结点的编辑是空间数据编辑过程中非常重要的一部分内容,主要分为结点吻合、结点与线的吻合两种方式<sup>[7]</sup>。结点吻合是指数字化过程中应该相交的一点的的数据。由于测量与绘图中的误差没有吻合,地图的数字化过程中,常遇到一个结点与一个线状目标的中间相交。由于测量或数字化误差,它不可能完全交于线目标上,需要进行编辑,称为结点与线的吻合。实现两种空间数据编辑结点的基础是能够实现对结点的精确捕捉。结点捕捉关系着空间数据的数据质量,在编辑过程中捕捉的实现有利于实现正确的拓扑关系,同时也利于拓扑关系的检查<sup>[8]</sup>。

在设计结点捕捉类前,首先要知道结点捕捉的过程。结点的捕捉主要通过进行鼠标光标点与某一地物

要素结点的坐标点的位置关系。设光标点为  $S(x,y)$ , 地物要素的结点的坐标为  $A(x,y)$ , 可设捕捉半径  $D$  (通常为 3 ~ 5 个像素,这主要由屏幕的分辨率和屏幕的尺寸决定)。若  $S$  和  $A$  的距离  $d$  小于  $D$ , 则认为捕捉成功,即认为找到的点是  $A$ , 否则失败,继续搜索其它点捕捉范围。在实际的搜索过程中,将搜索由圆改为矩形,这可大大加快搜索速度<sup>[9]</sup>。

设计捕捉类时 (Cls\_Snap) 利用 Windows GDI (gdi32.dll) 静态库中的 SelectObject, DeleteObject 函数,选择画笔对象<sup>[3]</sup>,通过 Polyline 与 Polygon 的函数建立集合来进行 shape 文件对象的捕捉。

实现过程如下,使用设计的函数搜索捕捉点,用于地物上某一结点的捕捉:

```
Public Function SearchSnapPoint (X As Single, Y As Single) As Boolean
```

’从鼠标的点的位置处捕捉一个地图点,将地图点放在 m\_SnapPoint 变量中,当函数有返回的地图点时,才返回 True 值,用于 m\_map 的 MouseMove 事件

’利用循环进行图层的遍历,使用 SearchSnapPointOnOneLayer 函数进行点的捕捉

```
.....If SearchSnapPointOnOneLayer (pt, m_map.Layers(i)) Then SearchSnapPoint = True
```

```
.....
```

```
End Function
```

在实际的编辑过程应用中,简单地捕捉点的位置的工程意义不大,但是简单的结点捕捉确是进行数据捕捉的基础。实际应用中常常是进行多个图层,或某一图层中多个地理要素点的位置捕捉。例如:需要将两条河流线段进行拼接时,如果不能实现结点一般捕捉,拓扑关系的建立就会出现河流不连续的错误,导致地理空间属性数据出现错误<sup>[10]</sup>。此时结点捕捉的实现就显得尤为重要。下面是 SearchSnapPointOnOneLayer 函数实现的部分代码,修改某一结点时实现与另一个结点捕捉,捕捉的图层对象可以是设置的任意图层。根据搜索得出的结点空间位置信息,利用 GDI 图像处理技术进行捕捉点符号的可视化。下面是利用空间捕捉函数将捕捉到的结点,与另一条线段吻合的效果示意图,如图 2 所示。

```
Private Function SearchSnapPointOnOneLayer (InPoint As MapObjects2.Point, Maplyr As MapObjects2.MapLayer) As Boolean
```

.....’通过遍历选择的空间数据集,进行结点一般捕捉的设置

```
Set NewSnapPoint = Rs.Fields("shape").Value ’获取地物结点
```

```
NewSnapPointType = SptEndpoint
```

```
.....
```

```
Call DrawSnapSymbol (m_SnapPoint, m_SnapPointType)
```

’这个函数利用 GDI+ 技术进行捕捉点的符号可视化

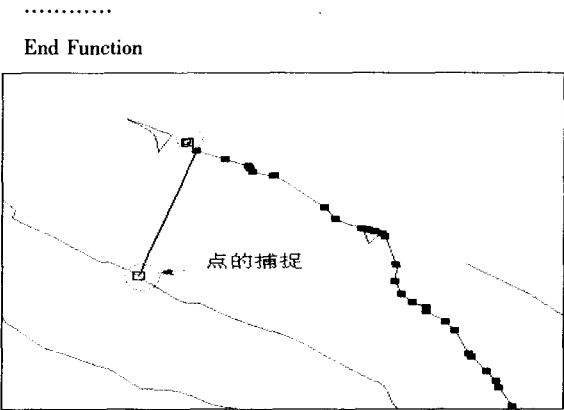


图 2 结点捕捉的效果图

2.2 MO 编辑类中的多要素移动功能的实现

空间数据中,许多人工地物,随着时间的改变,空间的位置属性也进行了相应的改变。例如:某环境监测站或可移动的放射性污染源等。在进行地物的空间数据移动过程中,由于 MO 中空间信息保存的数据集中,仅仅利用 MO 自身提供的函数接口,只能进行单一地物要素的移动,不能进行多个地物的同时移动。因此,根据工程的实际需要,要实现多个要素的同时移动功能,以保证多个地物之间的空间位置相对不变性。多要素移动功能的实现方法是:本地缓存对数据库的一定范围的数据在本地建立镜像,将这些数据按照一定的方式组织在本地的存储器上,本地存储器可以是内存也可以是硬盘<sup>[11]</sup>,利用缓存进行多个地物图像的保存。

根据交互的操作过程,在空间数据编辑 MoEdit 类中相应的过程中设计移动的实现函数,该函数能够进行多个地物的同时选择,之后再使用 GDI 编程技术进行移动操作。其具体实现的过程如下:首先在选择地物的过程中,MouseDown 事件利用 GDI 函数建立一个图像缓存区,存储数据集中的多个地物。其次,在移动地物的过程中,在 MouseMove 事件过程中利用 GDI 函数移动图像缓存区<sup>[12]</sup>。最后,地物在数据集中的保存过程,在 MouseUp 事件中,将本地缓存区中的地物保存到数据集中,并进行坐标的转换。下面是实现多个地物要素同时移动的主要实现代码,效果如图 3 所示。

```
Public Function MapMouseDown(pt As MapObjects2. Point) As Boolean
    If Shift = 0 then '使用 shift 进行地物多选
        '利用循环对多个地物加入要素集合
        m_Features. Item( Fids(i)). IsPointIn( pt)
        m_MoveFeatures = True '设置移动判断为真
    End if
    If m_MoveFeatures then 移动判断,进行移动操作
        '将集合中的地物,添加到地物的缓存区中
        .....
```

```
TempPt. X = m_Features( Fids(i)). X - Dist
TempPt. Y = m_Features( Fids(i)). Y + Dist
TempPts. Add TempPt 将地物所在区域的边界点添加到缓存点集中
.....
TempPoly. Parts. Add TempPts '添加地物的边界矩形到缓存区中
.....
End if
End sub

Public Function MapMouseMove(pt As MapObjects2. Point) As Boolean
    API 显示拖动,显示当前位置,消除原来位置图像
    Call Rectangle( m_Hdc, m_PtApiArray(0). X, m_PtApiArray(0). Y, m_PtApiArray(1). X, m_PtApiArray(1). Y)
    m_PtApiArray(0). X = m_PtApiArray(0). X + (NextX - m_PrevX) 坐标点移动
    .....
    Call Rectangle( m_Hdc, m_PtApiArray(0). X, m_PtApiArray(0). Y, m_PtApiArray(1). X, m_PtApiArray(1). Y)
    End sub

    Public Function MapMouseUp( pt As MapObjects2. Point) As Boolean
        .....if 是否移动 then
            '将 GDI 中的图形信息,存储在要素集中,在将要素集中的数据保存地理数据集中
            Set m_LyrRs. Fields( "shape"). Value = m_Features. Item( Fids(i))
            .....
        end if
    End sub
```

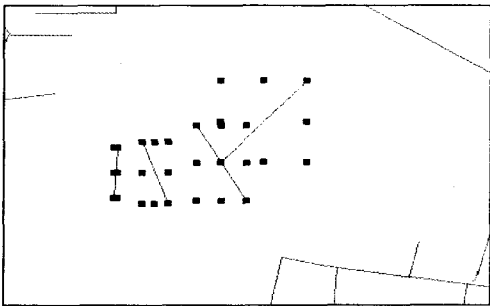


图 3 多地物要素移动的效果图

3 结束语

空间数据捕捉与移动是空间数据编辑过程中重要的操作方法,它可以有效地解决利用 MO 开发的空

间数据编辑功能在实际工程上的应用缺陷,如河流合并,多个地物要素同时移动修改等问题。文中通过使用 Windows 的 GDI 图形函数进行像素内搜索进行捕捉操作,以及使用 GDI 图形函数建立 MO 地理数据集的图

(下转第 142 页)

话工作量大、不方便。通过 LDIF 文件录入必须保证文件符合 LDIF 文件的格式要求,因此也具有不方便性。因此本系统设计并开发了目录信息维护的接口,提供了可以通过从数据库中或者从 Excel 文件中导入数据的用户界面,这给用户的操作带来了方便性和直观性。

## 5 结束语

用户的统一管理是数字化校园建设的基础和关键点,与传统的信息管理者数据库相比,LDAP 具有跨平台性、高校的读性能、支持分布式的目录信息维护、基于结构化的信息表示以及高可用性等逐渐被高校所接受,作为数字校园统一身份认证系统的用户信息存储库。

文中在对 LDAP 和统一身份认证深入研究的基础上,设计了目录信息树以及实现了对目录服务器的访问,并提供了一个基于 Web 的信息录入方式,进而实现了基于 LDAP 的数字校园统一身份认证系统。该系统作为数字校园建设的支柱,为其他各个系统提供了统一的认证和授权。

### 参考文献:

- [1] 郑凯,聂瑞华,梁卓明,等. 数字校园 ESB 技术的分析与实现[J]. 计算机技术与发展,2009,19(11):246-249.
- [2] 李翔,晁爱农,刘孟强. LDAP 的研究及其在统一身份认证系统中的应用[J]. 计算机应用,2008,28(6):98-100.
- [3] 李涛,张波,张晓鹏,等. 基于 LDAP 与 Struts 的数字校园门户统一身份申请系统[J]. 计算机应用与软件,2008,25(3):173-175.

(上接第 133 页)

像缓存区,对图像缓存区进行移动,实现了多地物的移动功能以及地物结点的捕捉功能。该方法已经在蓄滞洪区监测系统中得到了实际的工程应用,有效地提高了地图的制图精度。

### 参考文献:

- [1] 薛伟. MapObjects—地理信息系统程序设计[M]. 北京:国防工业出版社,2004:192-193.
- [2] 李新召. 基于 COM 组件的 Shape 文件生成算法[J]. 矿山测量,2003(4):29-31.
- [3] ESRI. Arcedit Users Guide: Interactive Graphics Editor[M]. California: Environmental Systems Research Institute,1998.
- [4] 刘文宝,夏宗国,崔先国. GIS 结点捕捉的广义算法及误差传播模型[J]. 测绘学报,2001,30(2):140-147.
- [5] 周安宁,沈成武. 对象模型与关系模型结合实现高效的对象系统[J]. 武汉理工大学学报(交通科学与工程版),2001,25(2):124-127.

- [4] 关婷婷,陈性元,张斌. 基于 XML 的数据表现格式在 LDAP 目录中的应用[J]. 计算机工程与设计,2006,27(22):4383-4386.
- [5] 吴杰明,周宁. 基于 LDAP 的信息共享平台研究与实现[J]. 计算机应用,2008,28(4):1042-1044.
- [6] Qadeer M, Salim M, Sana A. Profile management and authentication using LDAP[C]// Proceedings of 2009 International Conference on Computer Engineering and Technology. [s.l.]:[s.n.],2009:247-251.
- [7] Koutsonikola V, Vakali A, Mpalasas A, et al. A structure-based clustering on LDAP directory information[C]// Proceedings of the 17th International Symposium on Foundations of Intelligent Systems. [s.l.]:[s.n.],2008:121-130.
- [8] Salim M, Akhtar M S, Qadeer M A. Data Retrieval and Security Using Lightweight Directory Access Protocol[C]// Proceedings of Second International Workshop on Knowledge Discovery and Data Mining. [s.l.]:[s.n.],2009:685-688.
- [9] 肖婉蓉,杨生举. 基于 LDAP 的统一用户认证系统设计与实现[J]. 计算机科学,2008,35(5):298-300.
- [10] 胡立春,武友新,张烨,等. LDAP 环境下的统一用户管理系统的研究与实现[J]. 计算机工程与设计,2007,28(4):823-826.
- [11] 常潘,沈富可. 基于 LDAP 的校园网统一身份认证的实现[J]. 计算机工程,2007,33(5):281-284.
- [12] Vasilidis D C, Rizos G E. A Trusted Network Model Using the Lightweight Directory Access Protocol[C]// Proceedings of the 7th WSEAS International Conference on Applied Informatics and Communications. Athens, Greece:[s.n.],2007:252-256.

- [6] 蔡德利,郭庆丰,汪春. 扩展 MapObjects 数据源的研究 I: 将 Shapefiles 保存到 ADO.NET 数据集[J]. 计算机工程与设计,2006,27(9):1533-1536.
- [7] Burroughp A, McDonnell R A. Principles of Geographical Information Systems[M]. Oxford: Oxford University Press,1998.
- [8] Jones C B. Geographical Information Systems and Computer Cartography[M]. Harlow: Long-man,1997:87-92.
- [9] 吴磊,黄先锋,舒宁. GIS 大数据量编辑处理中快速捕捉的优化策略[J]. 武汉理工大学学报(交通科学与工程版),2005,29(2):316-318.
- [10] 周迪民,段国云. 地理信息系统属性数据不确定性的研究[J]. 计算机技术与发展,2009,19(12):174-178.
- [11] 杨建昌. GDI+高级编程[M]. 北京:清华大学出版社,2010:201-220.
- [12] 徐正光,田清,张利欣. 图像拼接方法探讨[J]. 微计算机信息,2006(10x):255-257.