

文章编号: 0494-0911(2012)01-0088-03

中图分类号: P271

文献标识码: B

PtInRegion 函数在地籍测量图形检查中的应用

史秋晶, 李羽荟, 张 魁

(江苏省工程物理勘察院, 江苏 南京 210008)

Application of Function PtInRegion in Cadastral Graphic Examination

SHI Qiuqing, LI Yuhui, ZHANG Kui

摘要: API 函数 PtInRegion 的作用是判断点是否在多边形内, 而地籍测量中经常要进行图形检查, 因此利用 PtInRegion 函数编写程序可以检查宗地内的线状地物与宗地面域是否矛盾。在 AutoCAD 环境下利用 VBA 开发图形检查工具时把线状地物微分为若干个, 依次判断各个点是否在宗地面内, 这种方法可避免繁琐的线面空间拓扑关系判断, 也可免除逐个放大线状地物进行检查的不便, 具有很强的实用性。

关键词: PtInRegion; 地籍; 线状地物; 宗地; 检查

一、前言

在第二次全国土地调查中, 测绘成果的图形检查是一项重要的工作, 直接关系到地籍数据库的质量好坏。在一线的实际工作中, 由于权属调查时界址认定和测绘工作不一致, 本宗内的线状地物与宗地面域会出现矛盾的情况, 也就是围墙、栅栏等线状地物有一部分在面域之外, 这给权属界定、面积统计等工作带来了一定的困难, 是应及时清除的。利用 API 函数 PtInRegion 可以编写检查工具, 找出图幅中错误的线状地物与宗地面域, 分析矛盾原因, 进行修改。这个检查工具在地形复杂、宗地内地物较多的情况下很实用。

二、PtInRegion 函数介绍

API (application programming interface) 是 Windows 的 32 位应用程序编程接口, 是一系列很复杂的函数、消息和结构, 它使编程人员可以用不同类型的编程语言编制出运行在 Windows 95 和 Windows NT 操作系统上的应用程序^[1]。PtInRegion 就是这样 API 函数, 它的功能是确定点是否在指定区域内。它的参数有 3 个: 指定区域的句柄、以逻辑坐标表示的点的 X 坐标、以逻辑坐标表示的点的 Y 坐标。如果点在指定区域内, 它的返回值为非零值; 否则为 0。

数字化测图的平台是南方公司的 CASS, 也就是 AutoCAD 平台。在这个平台下进行二次开发, 使用

PtInRegion 函数时, 有以下几个注意事项。

1. 指定区域的表示方法

CAD 平台下的区域是用首尾相接的多段线表示的, 记录多段线各顶点的平面坐标 (X, Y), 一个宗地就是一条多段线。要使用 PtInRegion 函数就需要一个区域, API 函数 CreatePolygonRgn 可以创建一个由一系列点围成的区域, 执行成功时返回值为创建的区域句柄, 失败则为 0。创建的区域不用时一定要用 API 函数 DeleteObject 删除。

2. 坐标值数据类型的处理

CAD 平台下的坐标值 X、Y 为双精度浮点型, 而 PtInRegion 函数要求点的坐标数据类型是长整型。为了提高计算精度, 在调用该函数前根据需要扩大坐标各分量值, 本文在开发图形检查工具时将坐标各分量值扩大了 100 倍。

3. 返回值的不准确性

由于 PtInRegion 函数本身的缺陷, 当点在指定区域的边界的时候, 并不能总是可靠地返回正确的结果^[2]。也就是说, 当点在区域的某一条边界上时, PtInRegion 函数有时返回非零值, 有时返回 0。为了解决这个问题, 需要先判断点是否在区域的边界上, 如果不在, 再调用 PtInRegion 函数判断点是否在区域面域内。

三、图形检查工具开发

在 AutoCAD 环境下利用 VBA 开发图形检查工具, 总体思路为: 将要检查的线状地物(也就是一条

收稿日期: 2010-12-29

作者简介: 史秋晶(1982—), 女, 江苏徐州人, 硕士, 助理工程师, 主要从事测绘与 GIS 工程应用工作。

多段线)看作由无数个(如100个、1000个)点组成的,取得各点的坐标,从线状地物的一端开始,调用PtInRegion函数依次判断这些点是否在宗地的区域内。如果所有点都在宗地内,则该线状地物在宗地内;否则线状地物有落在宗地外的部分。在线状地物与宗地边界相交的地方画圆提示出来,然后查明原因确定是否修改。下面列出部分函数声明和伪代码。

1. 使用 API 函数前的声明

新建面域的 API 函数

```
Private Declare Function CreatePolygonRgn Lib "gdi32"
(lpPoint As Any, ByVal nCount As Long, ByVal nPolyFillMode
As Long) As Long
```

删除面域的 API 函数

```
Private Declare Function DeleteObject Lib "gdi32" (ByVal
hObject As Long) As Long
```

判断点是否在指定区域内的 API 函数

```
Private Declare Function PtInRegion Lib "gdi32" (ByVal
hRgn As Long, ByVal x As Long, ByVal y As Long) As Long
```

2. 定义点坐标用的结构

Private Type COORD 该结构 COORD 有两个分量: x 坐标和 y 坐标

x As Long

y As Long

End Type

3. 判断点是否在宗地的边界上的函数

```
Public Function PointOnPolygon( ByVal p As Variant, ByVal pl
As AcadLWPolyline) As Boolean
```

判断点 p 是否在多段线 pl 代表的宗地的边界上,是返回 true,否则返回 false

4. 判断点是否在宗地内的函数

```
Public Function PointInPolygon( ByVal pt As Variant, ByVal pl
As AcadLWPolyline) As Boolean
```

判断点 pt 是否在多段线 pl 代表的宗地内,是则返回 true,否则返回 false

.....

If PointOnPolygon(pt, pl) = True Then 如果点在边界上,直接返回 true

PointInPolygon = True

Exit Function

End If

```
num = ( UBound( pl. Coordinates) - LBound( pl. Coordinates) + 1) / 2 取得宗地顶点数目
```

以下为取得宗地顶点坐标

```
For i = 0 To UBound( pl. Coordinates) - 1 Step 2
```

```
poly( i / 2 + 1). x = pl. Coordinates( i) * 100
```

扩大坐标

```
poly( i / 2 + 1). y = pl. Coordinates( i + 1) * 100
```

扩大坐标

Next i

```
hRgn = CreatePolygonRgn( poly( 1), num, ALTER-NATE) 创建区域
```

If hRgn Then 如果成功创建

```
If PtInRegion( hRgn, pt( 0) * 100, pt( 1) * 100)
```

< > 0 Then 如果点在区域内部

```
PointInPolygon = True
```

End If

End If

DeleteObject hRgn 删除区域

.....

End Function

5. 将线状地物微分为若干个段

先把代表线状地物的多段线划分为若干条线段,设线段的两个端点为 $p1(x, y)$ 、 $p2(x, y)$,再把这条线段平均分为 1000 份,上面的每个点 $p(x, y)$ 的坐标为

$$p.x = p1.x + (p2.x - p1.x) * i * 0.001$$

$$p.y = p1.y + (p2.y - p1.y) * i * 0.001$$

其中 i 为 0 ~ 1000。这样线状地物就被微分为若干个段,当 i 取不同值时,可以依次对 $p(x, y)$ 进行处理。

6. 主程序: 依次处理线状地物上的所有点

.....

```
For i = 1 To 1000
```

```
p.x = p1.x + (p2.x - p1.x) * i * 0.001
```

```
p.y = p1.y + (p2.y - p1.y) * i * 0.001
```

temp = PointInPolygon(p, pl) 对线状地物上的每个点判断是否在宗地区域内

如果有点不在宗地内,以该点为圆心, RADIUS 为半径画圆,提示有超出宗地范围的情况

```
If temp = false Then
```

```
Call ThisDrawing. ModelSpace. AddCircle( p, RADIUS)
```

```
End If
```

Next i

.....

四、实例

如图 1 所示,运行图形检查工具后,画圆圈的地方显示栅栏超出了宗地区域。由于宗地线(图 1 中的白色粗线)比较宽,掩盖了栅栏,不利用图形检查工具,肉眼很难看出栅栏是否超界。放大图形如

图2所示,可以看到栅栏的一个节点在宗地图形外。



图1 运行图形检查工具后检查出栅栏超界

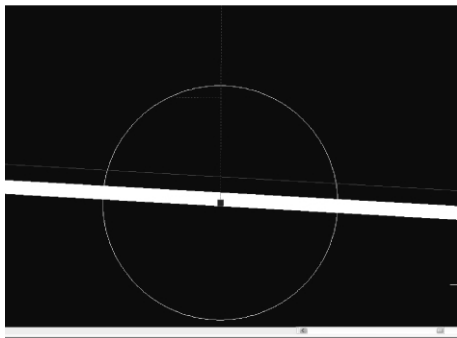


图2 放大能看到栅栏超界

五、结论与问题

利用 AutoCAD 平台下的 VBA 可以开发出检查

工具,用于判断线状地物是否超出宗地范围。总体思路是把线状地物平均微分成若干个节点,对每个节点判断它是否在宗地内,如果所有点都在宗地内,则该线状地物在宗地内;否则线状地物有在宗地范围外的部分。这样就可以避免判断线是否在面内的繁琐的拓扑计算。其中,应用 API 函数 PtInRegion 来判断点是否在宗地内是一个关键。

使用 PtInRegion 函数时将测量的双精度浮点型坐标用扩大的方法转换成了整数型坐标,其中必然产生截断误差,这个误差对计算结果有无影响;微分线状地物的时候,到底分 100 个、1 000 个还是更多,对判断结果和计算速度有无影响。这两个问题都有待于根据进一步的实例计算来研究。

参考文献:

- [1] 张桂勇,陈芳琼. API for Windows 2000/XP 详解[M]. 北京:清华大学出版社,2003.
- [2] 李海姣. Visual C++ 库函数 PtInRegion 存在的问题及修正[J]. 华东交通大学学报,2005,22(5):88-91.
- [3] 邓敏,马杭英. 线与面目标拓扑关系的层次表达方法[J]. 测绘学报,2008,37(4):507-513.
- [4] 廖楚江,杜清远. GIS 空间关系描述模型研究综述[J]. 测绘科学,29(4):79-82.
- [5] 胡克新,周晓光. 地籍空间数据质量检查与修正[J]. 地理信息世界,2006,4(4):70-73.

(上接第 55 页)

框架坐标到二维工程坐标的转换,实现了利用 GPS 进行工程定位的目的。

3) 采用勘测坐标系的坐标起算点和起算方位角,可直接利用控制点的工程坐标和设计图上的工程设计坐标进行施工放样,无需进行二维坐标转换。

4) 利用 3 个 HZMB-CORS 参考站的 ITRF2005 框架坐标和工程坐标作为工程坐标系的起算基准,可保持港珠澳大桥主体工程坐标基准的高精度和稳定性。

参考文献:

- [1] 中华人民共和国交通部. JTG C10—2007 公路勘测规

范[S]. 北京:人民交通出版社,2006.

- [2] 中华人民共和国铁道部. TB10601—2009 高速铁路工程测量规范[S]. 北京:中国铁道出版社,2009.
- [3] 郭际明,孔祥元. 控制测量学(下册)[M]. 武汉:武汉大学出版社,2007.
- [4] 丁士俊,畅开蚺,高锁义. 独立网椭球变换与坐标转换的研究[J]. 测绘通报,2008(8):4-6,35.
- [5] 畅开蚺. 建立城市独立坐标系相关问题的探讨[J]. 城市勘测,2008(1):86-90.