

文章编号: 1001-2486 (2000) 03-0109-05

AVL 系统中车位显示修正算法*

聂洪山, 徐捷

(国防科技大学信息工程研究所, 湖南 长沙 410073)

摘要: AVL 系统就是 GIS 与 GPS 技术集成的成功应用。AVL 系统中 GPS 信号在 GIS 地图上显示时, 由于各种因素的影响, 存在很大的偏差, 对这种偏差的校正是十分必要的。提出了一种偏差校正方法——“线性抽象算法”, 并通过 MapInfo 提供的 OLE Automation 技术在 Visual Basic 中实现了该算法, 实验结果表明这种方法具有较高的精度, 同时也指出了这种方法的不足。

关键词: AVL 系统; GIS; 地图; 偏差修正

中图分类号: TN91 **文献标识码:** B

Study on the Correcting Algorithm of Displaying Points of Vehicles in AVL Systems

NIE Hong-shan, XU Jie

(Institute of information Engineering, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: AVL system is a successful application of the integration of GIS and GPS. When GPS data is displayed on a GIS map, logically the vehicle point should be displayed on the roads. But because of the effect of errors, there are warps between the true vehicle point and the displayed point on the map. It is badly necessary to correct these warps. An algorithm named Linear Abstractin Algorithm (LAA) was put forward. It was realized through Visual Basic and MapInfo's OLE Automation technology. Experimental result shows that LAA has correspondingly perfect precision, and the shortcoming of the algorithm is also pointed out.

Key words: AVL system; GIS; map; wraps correcting

目前, 国际上 3S(GIS, RS, GPS) 的研究和应用正向着集成化方向发展。在这种集成应用中 GPS 主要用于实时快速地提供目标的空间位置, RS 用于实时地或准实时地提供目标及其环境的语义或非语义信息, GIS 则是对多种来源的时间、空间数据进行综合处理、集成管理、动态存取^[1]。GPS/AVL(全球定位系统/自动车辆定位)系统就是 GIS 与 GPS 技术集成的成功应用。

GPS/AVL 系统中, 当 GPS 移动目标在地图上显示时, 逻辑上车辆应该在道路上行驶, 但由于地图本身不够精确、地图数字化误差、GPS 系统误差、GPS 接收机误差等的影响, 导致移动目标在地图上显示时并不能保证其在道路上, 有时甚至相差很远, 而移动目标的精确定位正是系统的关键, 这就要求人为对其进行校正(校正过程也称为道路拟合)。作者在这个问题的研究中提出了一种“线性抽象算法”。

1 算法研究

简单地说, “线性抽象算法”就是对公路进行线性抽象然后再进行校正的算法, 为了达到校正偏差的目的, 需要根据算法设计的要求对地图进行加工, 包括创建公路线性图层、创建交叉路口图层以及对公路图层进行处理, 这要求要能够体现地图本身的特点并且要符合实际情况。

1.1 公路的线性抽象

公路的线性抽象是“线性抽象算法”的基础, 在前期的准备工作中, 在地图公路图层的基础上创建一个公路的线性抽象图层(称为公路线性图层)。因为在 MapInfo 地图中公路对象是一个区域对象, 所以创建公路线性图层时, 很容易用直线段来模拟公路对象的中心线, 对于有弯路的公路段则用几条直线段来

* 收稿日期: 1999-08-25

作者简介: 聂洪山(1975-), 男, 硕士。

模拟,画直线段时不一定要非常精确,但是一定要做到能够代表公路对象。“线性抽象算法”中公路线性图层中的线段对象是用来计算校正后的车辆点的位置。图1中的直线就是公路线性图层中的对象。

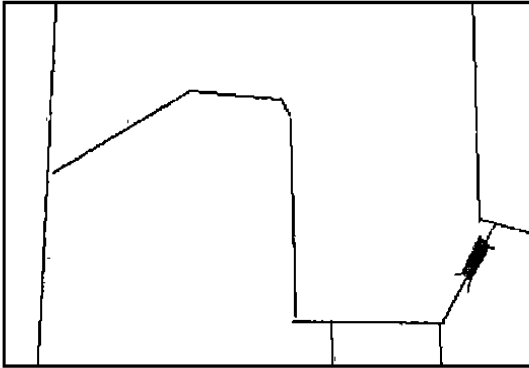


图1 用直线模拟公路

Fig. 1 To simulate roads with lines

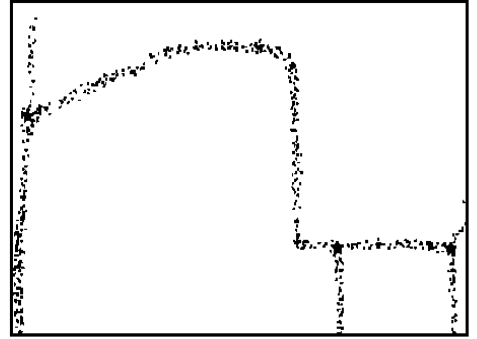


图2 用星号模拟交叉路口

Fig. 2 To simulate crossroads with star shape symbols

1.2 交叉路口图层的设计

对于任何校正算法来说,交叉路口都具有很强的特殊性:车辆可能拐弯、车辆可能停下、道路名称可能发生改变、遇到立交桥GPS信号会产生盲点,所以要在车辆接近交叉路口时进行特殊处理。可以说交叉路口处的处理是整个算法的关键。在“线性抽象算法”的前期准备中,类似公路图层的线性抽象,我们也对交叉路口处进行标记,创建一个隐含图层,这个图层在“线性抽象算法”中是用来判断车辆与交叉路口的空间位置关系,即交叉路口到车辆点的距离是否满足一定条件。图2中的星就是交叉路口图层中的对象。

1.3 校正算法

在“线性抽象算法”实现过程中,校正算法是核心部分,有了前边的准备工作,我们就能够很容易实现偏差校正。校正算法基本思想是找到离车辆点最近的公路线性图层中的一个线段对象,然后用点到直线的垂直点算法计算得到位于该线段上的点,也就是我们要求的校正点。其算法流程如下:

(A) 以车辆点为圆心选取适当的值为半径画圆(这个圆不必显示出来),判断该圆与交叉路口图层中的对象是否相交,如果在查找过程中发现车辆点到离他最近的交叉路口点的距离满足给定的条件(在圆范围内),则停止查找过程,直接将车辆校正点画在该交叉路口上,取下一个车辆点,返回A。

```

DO WHILE 没找到唯一对象
  IF {圆} ∩ {交叉路口图层中的对象} ≠ ∅ THEN '空间位置关系
    IF 唯一对象 THEN
      在离车辆点最近的交叉路口画校正点
    ELSE
      减小圆半径*
    END IF
  ELSE
    转到 STEP B
  END IF
LOOP
  
```

(B) 以车辆点为圆心、选取适当的值为半径画隐含的圆,判断该圆与公路图层中的对象是否相交,如果相交,则计算公路图层中有几个对象与该圆相交,如果找到多于一个对象,则适当减小圆半径,如没有满足条件的公路略层对象则适当加大圆半径,直到找到唯一一个满足条件的对象,记忆该对象的以路名称字段(记忆功能)。

```

DO WHILE 没找到唯一对象
  
```

```

IF{圆} ∩ {公路图层中的对象} ≠ ∅ THEN '空间位置关系
    IF 唯一对象 THEN
        纪录该对象的名称
    ELSE
        减小圆半径*
    END IF
ELSE
    增加圆半径*
END IF
LOOP
    
```

(C) 判断本次循环中取得的公路对象的名称和上次循环中取得的公路对象的名称是否相同, 相同则表示两个点在同一条公路上(公路的名称一致), 转到 D; 如果不同, 那么车辆很有可能是转变了, 但是我们不好判断车辆是向左转还是向右转(见图 3 和图 4), 所以我们采取对这个点不进行处理, 跳过这个点, 但仍记录和车辆点最近的公路对象的名称以备下一次循环是比较之用。取下一个点, 返回 A。

```

IF NAME1= NAME2 THEN
    转到 D
ELSE
    记忆 NME2' 关键步骤
    取下一个点
    转到 A
END IF
    
```

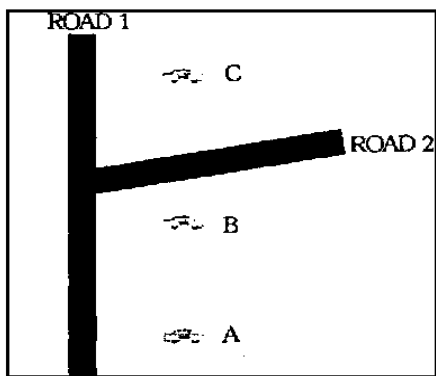


图 3 C 的校正点应放在公路 1 上
Fig. 3 The corrected point of point C should be put on road

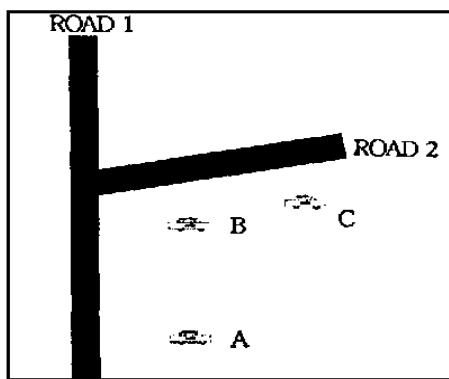


图 4 C 的校正点应放在公路 2 上
Fig. 4 The corrected point of point C should be put on road 2

(D) 以车辆点为圆心、选取适当的值为半径画隐含的圆, 按照与 B 相似的办法找到离车辆点最近的公路线性对象(即公路线性图层中的对象)。

(E) 在 MapInfo 中可以取得公路线性图层中直线段对象的两个端点, 已知车辆点和直线端两个端点坐标可以求出车辆点到该直线段的垂直点, 这个垂直点就是我们想要的校正点。如图 5 所示, 设 A 点的坐标为 $(X1, Y1)$, B、C 的坐标分别为 $(X2, Y2)$ 、 $(X3, Y3)$, 设直线 AB 的斜率为 K, 则 D 点的坐标 $(X4, Y4)$ 可以有下式求得(当 AB 为垂直线时, 需要特殊处理)

$$\begin{aligned}
 X4 &= (X3 + k * (Y3 - Y1 + k * X1)) / (1 + k * k) \\
 Y4 &= (Y1 + k * (X3 - X1 + k * Y3)) / (1 + k * k)
 \end{aligned}$$

(F) 在求得的垂直点上画一个标志点作为车辆的校正点。取下一个车辆点, 返回 A。

在“线性抽象算法”中, 有两个地方是需要特别提及的, 一是公路图层的线性抽象, 可以说这是整个算法得以实现的基础, 没有公路图层的线性抽象, 算法的实现将变得非常复杂, 并且很有可能得不到满意的效果; 二是记忆功能, 这部分对整个系统的实用性有着很大的贡献, 在 (C) 中我们不难自到, 如果没有一次记忆功能, 将很难精确描述车辆轨迹。

2 模拟结果^[2, 3]

MapInfo 是当今世界上最流行的桌面地理信息系统, MapBasic 是理想的在 MapInfo 平台上开发用户定制的应用程序的编程语言。MapInfo 提供了一种 OLE Automation 技术, 它是把 MapInfo 作为一个 OLE 对象放入一种非 MapBasic 语言中, 在用该种语言编制的应用程序界面里实现 MapInfo 的地图功能, 并在其中通过一定方式实现对 MapInfo 的控制。这就是得我们能够在 VB 程序中很方便地实现对 MapInfo 的控制。作者通过将 MapBasic 集成在 Visual Basic 中, 对“线性抽象算法”进行了实现。算法采用的地图是长沙市北区地图, 采用的 GPS 数据是作者模拟的 GN-77 接收板的数据, 图 6 中公路上的点是校正的结果点, 其中, A 处校正点是放在了交叉路口上, 作者在程序中设置如果在距车辆点 100m 范围内有交叉路口(如果交叉路口的个数超过一个, 则缩小搜索范围, 找其中离车辆点最近的交叉路口点), 则把车辆点的校正点画在该交叉路口上, 然后转到下一个点进行校正运算; B、C 两点的情况和 A 点外的情况相同, 所以校正点都画在了交叉路口上; 但 D 点情况有些不同, D 点并没有校正点, 这是因为 BC 公路段和 DE 公路段的名称不同(名称的改变发生在 C 附近的交叉路口处), 而我们现在并不知道 D 点的校正点究竟是放在 BC 公路段上还是放在 DE 公路段上, 所以跳过 D 点; 而在 E 点处, 因为 DE 两点在同一条公路上, 所以只用政常的校正算法即可, 先在公路图层中取得 DE 公路段的名称以用于判断再下一个点是否和 E 点在同一条公路上, 然后在公路线性图层中找到距离车辆点最近的公路线性对象, 求得校正点的位置, 将校正点画出即可; 其余的点像 F 点等都和 E 点的情况类似。

图 7 显示了算法的执行过程, 图中所画的圆是查询公路对象、公路线性对象和交叉路口图层时用到的。我们在地图预处理过程中对公路图层中公路对象的划分是按照实际地理特征划分的, 即公路的名称信息以及宽度、几车道、是否单行道(当两条路相距很近时作为辅助信息)等, 有时还可能根据地图的应用目的把地图上的某些小路删除掉。

作者在模拟实验时曾粗略的计算一下时间, 在赛杨 333 的机器上(64M 内存)做 10 个点的校正, LAA 算法大约需要 8s 多一点的时间, 而 BLAA 算法稍慢, 约需不到 10s, 也就是说校正速率能够满足 GPS 接收机 1s 的数率, 而且如果机器更高档则速度更快。

3 结束语

“线性抽象算法”能够较精确地对车辆点进行定位, 在最大限度上模拟车辆点的实际位置, 算法中加

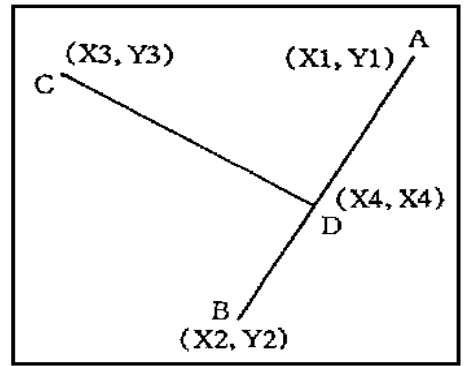


图 5 求点到直线的垂直点

Fig. 5 To calculate the vertical point of A point to A line

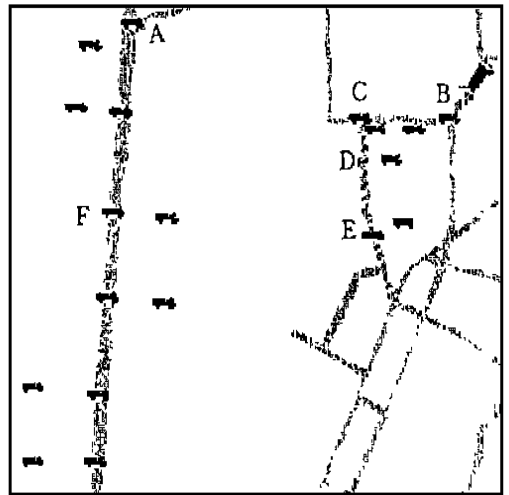


图 6 模拟结果示意

Fig. 6 Simulating results

入了一次记忆功能, 在对交叉路口处进行处理时较好地解决了问题。

但是,“线性抽象算法”一个最大的问题在于当公路图层很大时(公路线性图层也很大)的效率问题, 因为算法对每个点进行校正时都对整个公路图层和公路线性图和交叉口图层等进行几次搜索。

GPS 信号在 GIS 地图中显示时, 偏差的校正有很多种特殊情况, 这是一个非常复杂和困难的问题。本文中所讨论的情况只是其中的一部分, 实际上还有许许多多情况需要加以特别处理, 如路边停车场、居民区小路或倒 V 字形路口等。这种技术要达到真正用到 GPS/ AVL 系统中的程度还有很长的路要走。

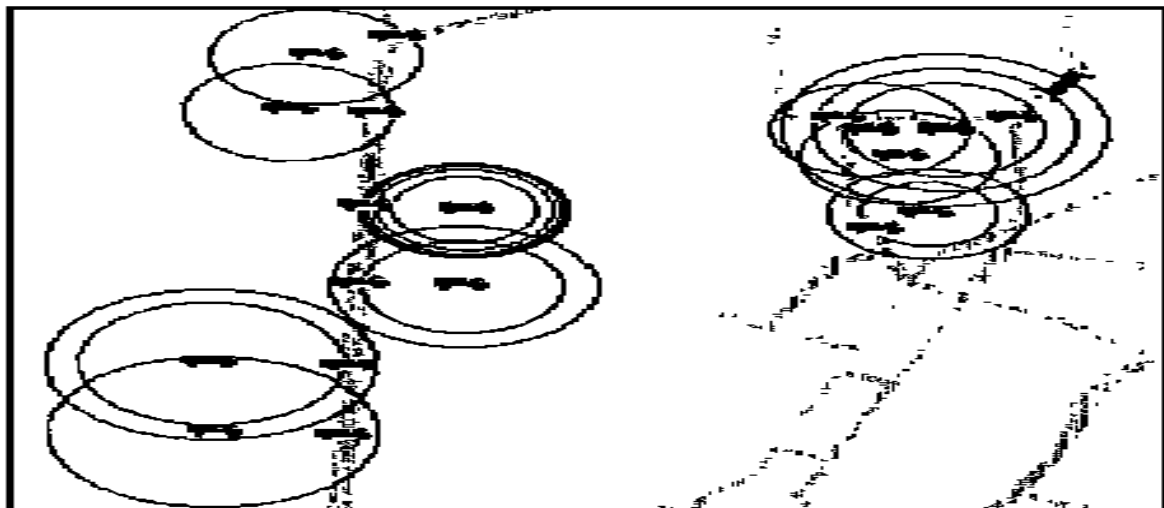


图 7 算法执行过程示意

Fig. 7 The calculating process of the algorithm

参考文献:

- [1] 李德仁. 论 RS, GPS 与 GIS 集成的定义、理论与关键技术[J]. 遥感学报, 1997, (2): 65~ 68.
- [2] MapBasic User' s Guide[M]. 3rd Edition October, 1996.
- [3] MapBasic Professional Reference. 3rd Edition October[M], 1996.
- [4] MapInfo Professional(4.5)用户指南[M].
- [5] MapInfo Professional(4.5)参考手册[M].