

实验研究

文章编号:1671-251X(2014)01-0042-04 DOI:10.13272/j.issn.1671-251x.2014.01.012

张轩,李崇贵,杨帆.煤矿井下人员行走轨迹的算法研究[J].工矿自动化,2014,40(1):42-45.

煤矿井下人员行走轨迹的算法研究

张轩, 李崇贵, 杨帆

(西安科技大学 测绘科学与技术学院, 陕西 西安 710054)

摘要:针对目前煤矿人员定位系统主要采用区域定位方式,存在无法直接解算出井下人员的精确位置 and 无法确定井下人员运动方向的问题,提出了一种基于树形结构、通过构建拓扑关系计算井下人员可能行走轨迹的算法。该算法首先定义了点与弧段、弧段与弧段以及弧段与点的拓扑模型,并以此为基础构建树形结构,对其进行层次遍历,得到井下人员行走轨迹;在计算过程中,该算法利用井下接收器的实时状态信息,自动排除停止工作的接收器和分站对轨迹的影响,有效确保了轨迹的准确性。

关键词:井下人员定位;行走轨迹;拓扑关系;树形结构;层次遍历;接收器

中图分类号:TD655 文献标志码:A 网络出版时间:2013-12-31 09:25

网络出版地址:<http://www.cnki.net/kcms/doi/10.13272/j.issn.1671-251x.2014.01.012.html>

Research of algorithm of walking track for underground personnel

ZHANG Xuan, LI Chonggui, YANG Fan

(College of Geomatics, Xi'an University of Science and Technology, Xi'an 710054, China)

Abstract: For problems that current coal mine personnel positioning system adopts region positioning method, and cannot directly determine exact location and movement direction of underground personnel, an algorithm based on tree structure was proposed, which uses topological relations to calculate track that underground personnel may walk. The algorithm defines a topology model with point and segment, segment and segment as well as segment and point. Based on the model, tree structure is built and personnel walking track is got by its hierarchy traversal. In the calculation process, the algorithm uses real-time status information of receiver to preclude impact of non-working receiver and substations automatically, and ensures tracking accuracy.

Key words: underground personnel positioning; walking tracks; topology relation; tree structure; hierarchy traversal; receiver

0 引言

井下人员定位系统是保证煤矿安全生产及人员调度的重要系统,再现人员在一定时间内的行走轨迹是人员定位系统不可或缺的功能。AQ6210-2007《煤矿井下作业人员管理系统通用技术条件》也要求系统应具有工作人员井下活动路线显示、打

印、查询、异常报警等功能^[1-2]。因煤矿井下环境复杂,不同类型巷道对发射卡信号的吸收不同,大部分煤矿中接收器不能覆盖井下所有区域,致使井下定位大多是以监测分站为基础的区域定位,无法确定人员的移动方向和准确位置。若要比准确地确定人员的运动轨迹,就需要布置足够多的监测分站和接收器,对井下范围较大的矿井将需要较高的费用。

收稿日期:2013-06-04; 修改日期:2013-09-25。

作者简介:张轩(1988-),男,陕西汉中,人,硕士研究生,主要研究方向为GIS在数字矿山中的应用与开发,E-mail:273659371@qq.com。通讯作者:李崇贵。

随着系统设备的增多,系统的使用和维护将变得复杂和困难。若能通过计算机程序分析人员行走轨迹,将弥补因硬件和环境因素造成不能确定人员行走轨迹的不足。这对减少设备投入、提高人员定位系统的易用性和可靠性,具有实际研究价值。本文提出了一种基于树形结构、利用构建拓扑关系计算人员可能行走轨迹的算法。在计算过程中,利用井下接收器的实时状态信息,自动排除停止工作的接收器和分站对轨迹的影响,有效确保了轨迹的准确性。

1 轨迹计算的基本原理

1.1 井下人员定位基本原理和区域特性

在人员定位系统中,需要在井下布置大量的分站和接收器,以便能够接收井下各类作业人员所携带的发射卡所发出的信号。发射卡通过 UDP 协议不断发送发射卡信息,接收器一旦接收到发射卡发送来的信息,便将信息传输到相应的分站,由分站解算发射卡位置并上传到服务器。通过这种方式,可获取井下人员的位置信息和接收器或分站设备的信息。

目前,在我国煤矿普遍使用的人员定位系统主要采用区域定位方式,无法直接解算出人员的精确位置,只能获得人员处于某个接收器或分站的信号接收范围内^[3],无法确定人员的运动方向,运动方向和轨迹只能依靠应用系统计算获得。

1.2 轨迹计算的 3 种情况

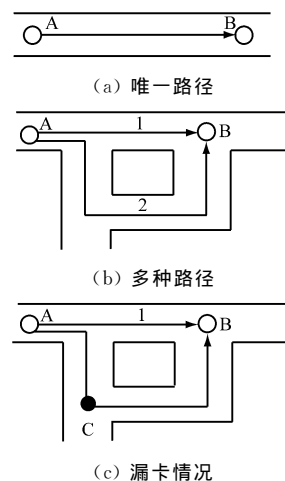
基于区域定位的特性,只需获得人员在一定时间段内所经过的设备(分站)信息,以一定方式连接相邻设备,就能获得人员的行走轨迹^[4]。轨迹计算的 3 种情况如图 1 所示。

因设备的位置分布、设备的工作状态都将影响人员行走的可能路径。当从服务器中获取人员在某相邻时间的位置信息依次为 A 和 B 时,从 A 到 B 的轨迹将可能出现下列情况。

(1) 存在唯一路径,如图 1(a)所示,从 A 走向 B 仅有一种轨迹。

(2) 存在多种路径,如图 1(b)所示,井下设备的位置分布造成从 A 到 B 的轨迹存在多种可能性。

(3) 设备损坏或未正常工作时,如图 1(c)所示,当设备 C 工作正常时,从 A 到 B 仅有编号为 1 的轨迹。而设备 C 损坏或未正常工作时,从 A 到 B 将有 2 条可能轨迹。



○—表示正常运行设备; ●—未能正常工作设备; →—行走轨迹

图 1 轨迹计算的几种情况

2 预设轨迹和设备点的拓扑构建

2.1 预设轨迹

通过绘制预设轨迹,将相邻设备以一定方式进行连接,然后按照预设轨迹和当前设备状态进行人员轨迹的计算。预设轨迹是轨迹计算的基础,并不是简单地将相邻设备按照可能的轨迹进行连接。必须满足相邻设备间的预设轨迹由 1 条或多条弧段构成,每条弧段的端点只能是设备点或其他弧段的端点。弧段绘制过程中,遇见巷道的分叉口或设备点时,弧段必须结束绘制。图 2 为弧段绘制正确的示意图,图中虚线圈的位置必须是弧段的终点或起点。

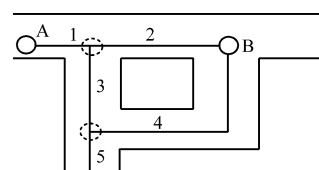


图 2 预设轨迹的绘制

2.2 拓扑构建

传统的拓扑模型用存储点和弧段的编号描述图形之间的关系,本文对其进行了扩充,更加详细地描述图形之间的关系,且这些详细信息在树节点的计算中可有效避免点与弧段、弧段与弧段的复杂坐标比对,极大提高运算速度。

图 3 为设备点 A 到 B 的预设轨迹,其中每条轨迹都标明其绘制方向,用以说明弧段起点和终点,对图 3 建立的部分拓扑关系见表 1—表 3。

表 1 表示设备点 A 与 L1, L2, L3 为关联关系,设备点 B 与 L3, L4 为关联关系。设备点 A 与 L1, L3 的起点坐标相同,设备点 A 与 L2 的终点坐标相同,设备点 B 与 L3, L4 终点坐标相同。

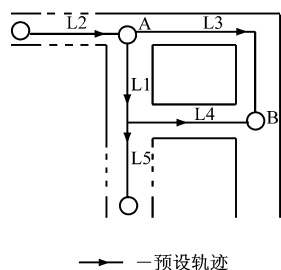


图 3 从设备点 A 到 B 的预设轨迹

表 1 设备点与弧段的拓扑关系

| 设备点编号 | 类型 | 弧段编号 |
|-------|----|--------|
| A | 起点 | L1, L3 |
| | 终点 | L2 |
| B | 起点 | L4 |
| | 终点 | L3, L5 |

表 2 弧段与弧段的拓扑关系

| 弧段编号 | 类型 | 弧段编号 |
|------|-------|--------|
| L1 | 起点—起点 | L3 |
| | 起点—终点 | L2 |
| | 终点—起点 | L4, L5 |
| | 终点—终点 | |

表 3 弧段与设备点关系

| 弧段编号 | 类型 | 设备点编号 |
|------|----|-------|
| L1 | 起点 | A |
| | 终点 | |
| L3 | 起点 | A |
| | 终点 | B |

表 2 表示弧段 L1 与 L2, L3, L4, L5 为邻接关系。L1 的起点与 L3 的起点坐标相同, L1 的起点与 L2 的终点相同, L1 的终点与 L4, L5 的起点相同。

表 3 表示弧段 L1 的起点与设备点 A 坐标相同, L3 的起点与设备点 A 坐标相同, 终点与设备点 B 坐标相同。

利用预设轨迹图层和自身构建弧段与弧段拓扑关系, 设备点图层和预设轨迹图层构建点与弧段拓扑关系、弧段与点的拓扑关系, 并将这些拓扑信息序列化, 保存在特定格式的文件中, 可供每次计算路径时使用, 而不必重新构建拓扑关系。

3 树形结构与轨迹计算

3.1 树的存储结构

树形结构是数据元素之间有分支和层次的关系结构, 它类似于自然界中的树。在计算机中, 树的存储有多种方式, 并且每种方式不仅能存储各节点本

身的信息, 还要能反映树中各节点之间的逻辑关系^[5]。

本文采用双亲孩子表示法。双亲孩子表示法是将双亲表示法和孩子表示法相结合的结果。将各节点的孩子节点分别组成链表, 将链表的指针存放在节点的域中, 再增设一个域, 存储该节点的双亲节点的指针, 节点结构的 C# 和 ArcEngine 代码如下:

```
public class PathCurve
{
    public int Id; //该弧段编号
    public IFeature feaLine; //该弧段线要素
    public PathCurve LastLine; //双亲节点
    //与双亲节点中弧段相关联的端点类型(起点或终点)
    public EndPointType pointType = EndPointType.
FromPoint;
    public List< PathCurve > NextLines = new List<
PathCurve>(); //子节点列表
    //表示该节点是否计算与之邻接的子节点
    public bool isContinue = true;
    //表示该节点是否已经到达目标点
    public bool isReach = false;
    public PathCurve (int lineId, PathCurve lastline,
EndPointType pointtype); //构造函数
    //获得子节点列表的函数
    public void GetNextLines();
    //判断该节点是否已经到达目标或是否停止计算子节点
    public void IsStop (int objPointId, List< int >
ExceptPointIdSet);
    //得到从根节点到该节点的轨迹
    public List< PathCurve > GetPath();
}
```

3.2 树的建立与层次遍历

层次遍历是指从树的根节点开始, 从下至上逐层遍历, 在同一层中, 则按照从左到右的顺序访问每个节点^[5-6]。在本文中采用对树的层次遍历, 并且在遍历节点时动态创建该节点的子节点。

当给定一个轨迹起始点时, 利用拓扑关系可获得一个逻辑深度为 n , 度的最大值为 m (m 为弧段总数) 的树。树中根节点表示起始设备点, 其他节点均表示轨迹中的弧段, 子节点弧段与双亲节点弧段邻接。该树是一个节点繁多、深度不可预计的树。在实际计算中, 当遍历第 i 层时, 动态获取第 $i+1$ 层的部分子节点, 并存储在内存中供下次遍历时使用, 以达到节省时间和空间的效果, 所遍历的这些节点是该树节点集合的子集。任意节点 a_k, a_{k-1}, \dots, a_1 ,

a_0 , 从 a_0 到 a_k 的所有弧段就构成了从起始设备点到 a_k 的端点中与 a_{k-1} 不相关联的端点 a_{k2} 的轨迹。其中 a_i 是 a_{i+1} 的双亲, a_0 为根节点。若 a_{k2} 为目标设备点, 则获得的这条轨迹是正确的。若 a_{k2} 为其他设备点, 在不考虑漏卡的情况下, 这条轨迹则是错误的。且对于任何包含该段轨迹的轨迹都是错误的, 所以, 就不必遍历 a_k 的子节点了。

轨迹计算的步骤如下:

- (1) 通过起始设备点, 产生第 2 层节点;
- (2) 获取设备的运行状态, 确定轨迹不能到达的设备点;
- (3) 从第 2 层开始进行层次遍历, 遍历每层的所有节点;
- (4) 调用节点的 isStop 函数, 判断节点是否关联目标设备点或其他不能到达的设备点;
- (5) 通过 isReach 判断是否已经获得正确轨迹, 若属性值为 True, 则调用 GetPath 获得轨迹, 并存储获得的轨迹;
- (6) 通过 isContinue 属性判断是否调用 GetNextLines 函数获得子节点;
- (7) 遍历完第 i 层, 判断第 $i+1$ 层的节点数, 若等于零, 则遍历结束; 否则, 返回第 3 步继续遍历。

在建立的树中, 所有叶节点均关联设备点, 关联目标设备点的叶节点个数为轨迹个数, 图 4 为图 3 的树形结构。

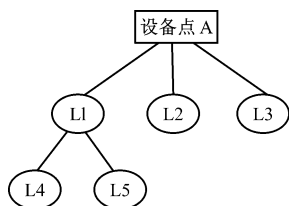


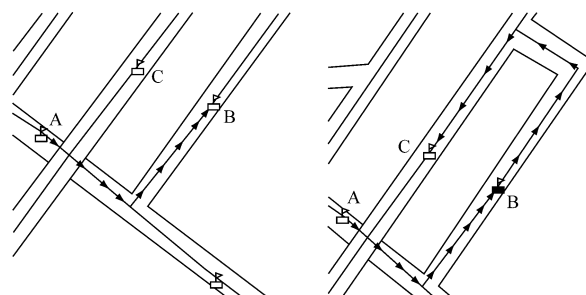
图 4 从设备点 A 到设备点 B 的树形结构

4 应用实例

根据前面提出的结构与算法, 采用 C# 和 ArcEngine 开发了某煤矿的人员定位系统中的轨迹计算功能, 如图 5 所示, 分别显示了设备状态正常和部分设备故障情况下的轨迹。

按照轨迹顺序, 3 个设备依次编号为 A、B、C, 图 5(a) 表示人员从 A 走向 B, 设备 C 所在巷道对 A 所在巷道有压盖情况, 但压盖不会对路径计算造成干扰; 图 5(b) 为人员从走 A 向 C, 在设备 B 状态正常情况下, 不能直接得到 A 到 C 的轨迹, 而设备 B 故障时, 该算法可以忽略掉故障设备, 直接得

到 A 到 C 的路径。



(a) 巷道压盖情况下轨迹 (b) 设备故障情况下轨迹

图 5 轨迹计算结果显示

5 结语

煤矿井下作业人员轨迹回放是人员定位系统中的重要功能之一。人员行走轨迹计算的准确性, 灵活性是轨迹回放的重要研究内容。本文提出了一种基于树形结构、通过构建拓扑关系计算人员可能行走轨迹的算法。在井下区域定位的情况下, 该算法通过计算机编程对人员位置的分析, 推算出人员行走的轨迹, 解决了设备分布和设备故障给轨迹计算造成的不确定性影响; 利用井下接收器的实时状态信息, 自动排除停止工作的接收器和分站对轨迹的影响, 有效确保了轨迹的准确性。

参考文献:

- [1] 何峰. 煤矿井下作业人员管理系统软件的设计[J]. 山西煤炭管理干部学院学报, 2011, 24(3): 12-14.
- [2] MT/T AQ 6210-2007, 煤矿井下作业人员管理系统通用技术条件[S].
- [3] 樊哲. 井下人员定位系统在现代矿山中的应用[J]. 甘肃冶金, 2012, 34(4): 104-106.
- [4] 陈学斌. 煤矿人员管理系统中轨迹回放动画的实现[J]. 采矿技术, 2010, 9(增刊 1): 154-156.
- [5] 张小艳, 龚尚福. 数据结构与算法[M]. 徐州: 中国矿业大学出版社, 2007: 125-168.
- [6] 彭军, 向毅. 数据结构与算法[M]. 北京: 人民邮电出版社, 2013: 116-120.
- [7] 孟磊, 丁恩杰, 冯启言, 等. 基于 WiFi 与 WebGIS 的井下移动目标定位与历史轨迹提取[J]. 地理与地理信息科学, 2012, 28(3): 109-110.
- [8] 尹积婷, 郭永泉. 煤矿井下人员考勤定位管理系统[J]. 测控技术, 2007, 26(3): 80-82.
- [9] 郭秀才, 滕以金, 奚美玲. 井下人员定位跟踪检测系统[J]. 兵工自动化, 2009, 28(4): 58-63.
- [10] 华钢, 周磊, 任凯. 地下煤矿人员跟踪定位系统发展综述[J]. 矿山机械, 2008, 36(6): 36-40.