

基于 SuperMap Object 的煤矿巷道三维拓扑模型生成的研究

刘 巍 胡锦鑫
(中国矿业大学)

摘 要:根据煤矿巷道的特点,利用基于面向对象的开发方法,结合 SuperMap 强大的拓扑处理功能,对巷道线数据集进行二维拓扑处理,生成网络数据集,对网络数据集进行拓扑反馈,提取过滤点数据集,重新对巷道线数据集进行拓扑处理,实现三维拓扑模型的建立。

关键词:SuperMap; 网络数据集; 拓扑反馈; 三维拓扑

中图分类号:TD821 **文献标识码:**A **文章编号:**1009-5683(2008)02-0009-03

Research on 3D Topology Model of Coal Mine Laneway Based on SuperMap Object

Liu wei Hu jinxin

(China University of Mining Technology)

Abstract: According to the characteristics of coal mine laneway, combining mighty function of SuperMap in topology treatment, a development method based on object oriented is used to carry out 2D topology treatment of the laneway line data set to get network data set. Then topology feedback of the network data set is conducted to collect filter points data set. It carries out the topology treatment of the laneway line data set over again so as to establish the 3D topology model.

Keywords: SuperMap; Network data set; Topology feedback; 3D topology

1 引言

煤矿巷道是一个复杂的立体系统,巷道数目繁多,纵横交错,上下重叠,相互关系复杂。建立巷道三维空间拓扑关系模型,是煤矿生产、通风、运输、灾害模拟不同制图和分析的需要。在数字矿山研究领域,目前关于巷道的三维显示主要是基于软件来生成模型,比如 3Dmax 和 AutoCAD 等。在这些软件之上建立的三维模型逼真度虽好,但缺少灵活性,可编辑能力不强,可移植性差,更难以进行拓扑分析。现在存在的三维拓扑的建模方法都较为复杂,很多是趋于理想状态,不能很好的实现。

该系统利用北京超图公司的 SuperMap Object 控件,以组件式开发方式,采用 SuperMap 强大的拓扑分析功能,利用拓扑反馈,较为简单的生成了三维拓扑后的 SuperMap 网络数据集,也就是巷道的三维拓扑模型。

2 巷道三维拓扑建模原理

2.1 从三维到二维

根据巷道的实际情况,将巷道看成线型,针对某

一采区的巷道,其巷道多在 Z 轴方向上密集。图 1 是巷道三维显示在 ZX 面(或近似在 ZX 平面)上的投影。

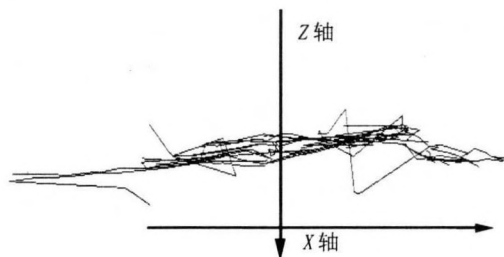


图1 巷道三维显示在 ZX 面

从图 1 可以清楚地看到,巷道的分布在 Z 轴方向的投影是很密集的,这就说明同一采区的巷道高程相差不大,大多数巷道都在同一高程区间内,只有少部分例外。

在 XY 面上的投影见图 2。从方位可以看到巷道分布比在 ZX 面上清楚得多。根据以上分析认为:将巷道数据进行三维成图后,再将其投影到 XY 面上,既有利于观察,也有利于拓扑分析。

目前的 GIS 软件仅能处理二维拓扑数据,无法计算三维线状要素的拓扑关系。利用 SuperMap 对图 2 线数据集进行拓扑处理后的网络数据集如图

3。

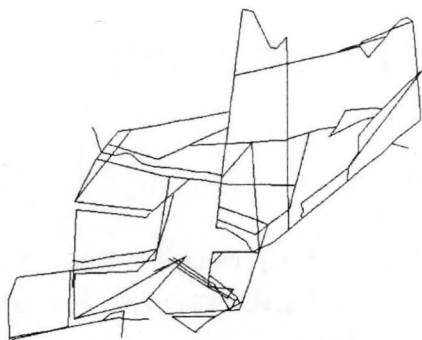


图 2 巷道三维显示在 XY 面

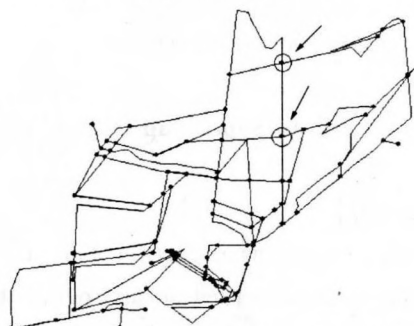


图 3 SuperMap 拓扑分析后的网络数据集

图 3 显示的仅仅是巷道投影在 XY 平面的拓扑关系,其中的许多结点并不符合实际。如箭头所指位置,在三维模型中可以看到形成此结点的两条线是不相交的,而且它们之间的空间距离很远。下面尝试用拓扑反馈建立三维巷道的拓扑模型。

2.2 拓扑反馈

二维拓扑处理后,在所生成的网络数据集中,取其中结点数据集,然后从这个结点数据集中筛选弧段求交点点集,从交点点集中筛选出空间上不可能相交的交点,把这些交点保存在一个点数据集中。拓扑反馈就是反馈出这个点数据集。

这个点数据集就可以作为非打断线的参数,在拓扑分析时予以过滤。

3 实现方法

3.1 程序设计工具

采用面向对象的开发方法,数据库采用 SQL 数据库,选择 C++ 作为开发语言,SuperMap Object 作为开发组件。

3.2 原始数据转换

巷道的原始数据是由矿山地测人员所测的巷道导线点坐标,将其整理成 SQL 数据库见图 4。

要将数据表转换成 SuperMap 的 .sdb 数据,在保存线数据集(这里以线段的方式加入到线数据集中,而不是以整条导线的方式存储,以使异面直线求距离时可以直接提取到线段),在读取 SQL 数据到

sdb 数据时,要在所生成的线数据集(取名为 line)的属性数据表中加入 3 个字段,分别是:ID;Point1Z;Point2Z。分别表示:线段的编号(自动生成,从 1 开始递增记录);线段起点的高程坐标;线段末点高程坐标。属性数据表见图 5。

图 4 原始的 SQL 数据

编号	SeID	SeUserID	SeLength	SeTopoError	ID	Point1Z	Point2Z
1	1	0	67.6554947309906	1	1	-298.34	-300.651
2	2	0	5.08629849925216	0	2	-300.651	-300
3	3	0	48.957840096233	0	3	-300	-295.263
4	4	0	26.34944545285058	2	4	-295.263	-295
5	5	0	52.4024157336348	1	5	-305.946	-303.744
6	6	0	25.43161610095657	0	6	-303.744	-300.235
7	7	0	46.925321129981	0	7	-300.235	-299.123
8	8	0	6.96396446074118	0	8	-299.123	-297.682
9	9	0	37.5482144629136	0	9	-311.065	-315.492
10	10	0	30.7584346217107	0	10	-315.492	-318.496
11	11	0	20.3101675443551	0	11	-318.496	-315.822
12	12	0	24.3566336193143	0	12	-315.822	-304.6
13	13	0	60.9592300914308	0	13	-304.6	-304
14	14	0	9.9854126573069	1	14	-235.288	-239.937
15	15	0	7.12653784033062	0	15	-239.937	-239.927

图 5 line 线集属性数据表

对生成的 line 数据集进行初步的编辑,然后将修改后的线数据集保存,这里仍用 line 名称表示修改后的线数据集。

3.3 交点号数组生成

(1)定义锯齿形二维数组 int NodeLine[][],此数组就是交点信息数组。NodeLine 数组第一维存储所有线段相交产生的交点顺序号,第二维第一位存储交点编号 NodeID,其他位存储该交点相交的所有线段的线段号,也就是在 line 属性数据中的 ID。

(2)对 line 数据集进行拓扑处理,生成网络数据集,取名为 topo。

(3)定义记录集 soRecordset objSubRdt,记录生成的 topo 网络数据集子点数据集中结点的编号 SmNodeID。

(4)依次读取记录集 objSubRdt,定义 int NodeID 记录当前读取的结点号 SmNodeID。

(5)定义记录集 soRecordset objArcIDRdt = objNetDtv.Query("SmFNode = " + NodeID + " or SmTNode = " + NodeID, false, objFields, null)其中 objFields.Add("DISTINCT ID")。该记录集记录的是 SmFNode = NodeID 或 SmTNode = NodeID 的线段 ID

(除去了重复 ID)。

(6)统计 objArcIDRdt 的记录数,如果该记录数大于等于 2,就说明该结点号 NodeID 是线段形成的交点,那么就将该结点号 NodeID 存储到 NodeLine 每行的第一个位置,每行其他位置存储形成该交点的线段号 ID。

(7)NodeLine[][]交点信息数组生成成功。

3.4 异面直线间距离算法

测量工作者所测的巷道数据,因为误差的存在和井下复杂的地理环境,两条实际相交的巷道往往就很难相交,而是在空间上形成异面直线。见图 6。

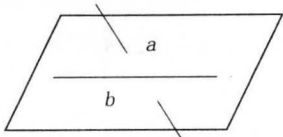


图 6 测量的巷道线段

a 、 b 巷道实际相交,但是由于测量误差导致 a 、 b 不相交。那么求出 a 、 b 两异面直线的距离 L ,如果 L 大于一定阈值,就认为 a 、 b 不相交。小于这一阈值就认为 a 、 b 相交。

异面直线距离的算法采用空间几何向量来实现,这里就不再详述。

设计空间直线类 Line:

```
public class Line
```

```
{
```

```
    public double[ ] bP = new double[3]; //直线的起点
```

```
    public double[ ] eP = new double[3]; //直线的终点
```

```
    public double[ ] direction = new double[3]; //直线的方向向量
```

```
    public void FuZhiDirection() //对直线的方向向量赋值并归一化
```

```
{
```

```
        double a = Math.Sqrt((eP[0] - bP[0]) * (eP[0] - bP[0]) + (eP[1] - bP[1]) * (eP[1] - bP[1]) + (eP[2] - bP[2]) * (eP[2] - bP[2]));
```

```
        direction[0] = (eP[0] - bP[0]) / a;
```

```
        direction[1] = (eP[1] - bP[1]) / a;
```

```
        direction[2] = (eP[2] - bP[2]) / a;
```

```
}
```

```
}
```

异面直线的距离类 lIDistance。在 lIDistance 中

定义 Distance(Line line1, Line line2) 函数进行异面直线 line1 和 line2 的距离计算,返回 double 类型。

3.5 拓扑反馈的实现

有了 NodeLine[][] 二维数组和异面直线距离量算的类,拓扑反馈的实现如下。

(1)依次读取 NodeLine[i][0],此项记录了交点编号 NodeID。

(2)读取 NodeLine[i][1], NodeLine[i][2] 的值,即为第 i 号交点的两条线段号 ID1, ID2。

(3)由 ID1, ID2 就可以查找线数据集 line 中的 ID 号,提取这两条线段首点末点的三维坐标值。

(4)分别定义 Line line1, Line line2, 将 ID1, ID2 号线段的首点末点的三维坐标值赋给 line1, line2 的起点终点属性。

(5)设定阈值 X , 定义点数据集 point, 如果 Distance(Line line1, Line line2) > X , 则将这个结点记录到 point 点集中。

(6)完成拓扑反馈,生成点集 point。

3.6 三维拓扑图生成

利用生成的点数据集 point 作为非打断线的参数,再对 line 数据集进行拓扑处理,就能生成空间拓扑关系比较准确的网络数据集,见图 7。

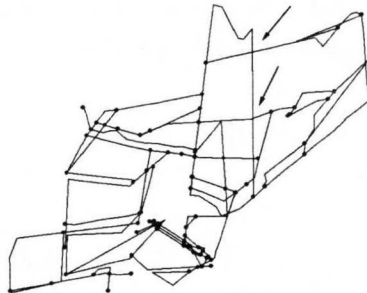


图 7 三维拓扑模型图

与图 3 对比,可以看出,经过拓扑反馈建立的三维巷道拓扑模型更符合实际,如箭头所指位置。

4 结论

巷道三维拓扑模型的正确建立对矿山信息处理具有重要意义,文中方法可以方便地建立巷道的三维拓扑模型,所构模型与实际情况吻合。

参考文献:

- [1] 孔令标,侯运炳. 基于 Arc/Info 的矿井风网解算方法研究[J]. 矿业工程, 北京, 2003.
- [2] Philip J. Schneider, David H. Eberly. 计算机图形学几何工具算法详解[M]. 北京: 电子工业出版社, 2005.
- [3] 黄力波, 刘彦伟, 李志强等. 矿井通风网络图[J]. 焦作工学院学报, 2002, 21(1): 9~12.

(收稿日期 2007-09-09)