

车辆导航系统中最短路径计算的数据模型

罗跃军¹, 李霖¹, 朱敦尧², 郑莉³

(1. 武汉大学 资源与环境科学学院, 湖北 武汉 430079; 2. 光庭导航数据有限公司, 湖北 武汉 430070; 3. 武汉大学 遥感信息工程学院, 湖北 武汉 430079)

摘要: 在自主车辆导航系统中如何实现符合现实状况的大范围的最短路径查询, 对道路等级、道路通行条件、转弯规制等限制进行抽象, 建立了模拟现实道路网络的交通仿真模型; 对于大数据量交通网(全国范围)的路径计算, 提出按道路等级分层建立拓扑网和分区域建立区域小型拓扑网的解决方案。

关键词: 自主车辆导航; 智能交通系统; 计算机化仿真; Mesh

中图分类号: U491.13 **文献标识码:** A **文章编号:** 1007-855X(2004)03-0106-04

Data Model in the Shortest-Path Algorithm of Vehicle Navigation System

LUO Yue-jun¹, LI Lin¹, ZHU Dun-yao², ZHENG Li³

(1. Institute of Resource and Environment, Wuhan 430079, China; 2. Kotei Navi& Data(Wuhan Corporation), Wuhan 430070, China; 3. Institute of Remote Sensing Information Engineering, Wuhan University, Wuhan 430079, China)

Abstract: Problems of the shortest-path algorithm in a transportation network are studied and a transportation simulation model is put forward in which the road grades, the traffic condition and the rules of swerving are abstracted. And the practical method to calculate the paths of the nation-wide transportation network is based on the fact that topological network is established according to the road grades and the regional small-sized network is built in different areas.

Key words: dominated vehicle navigation; intelligent transportation system; computerized simulation; Mesh

0 引言

自主车辆路径导航系统是智能交通系统的重要研究课题, 它要求能够按照存储在其内部的数字地图的拓扑信息, 实时的规划出一条全局最优路径用于车辆导航. 如果应用于实际中, 这里的拓扑网就必须需要真实反应现实情况. 以往的道路最短路径算法基本上是基于简单拓扑网络, 这样的拓扑网络是将现实中复杂的交通道路网络简单化, 数据量较小. 要实现自主车辆路径导航系统, 首先要建立仿真的交通道路网络拓扑模型, 并且能对海量数据进行道路最短路径计算.

1 影响交通网络中最短路径算法的因素

现代交通网发展越来越完善, 复杂度也越来越大. 在交通网中做最短路径查询, 不同于一般简单有向拓扑网中的最短路径查询, 它有其特殊性:

1) 道路分级别, 高等级道路(高速公路、国道等)的通行条件好, 速度快; 低等级道路(街道、乡村路等)的通行条件差, 速度慢, 还有其他等级的道路通行状况好坏各不相同. 在距离大致相同的情况下, 人们大多愿意选择通行条件好通行速度快的道路, 因此道路分级别对最短路径算法在时间方面有很大的影响.

2) 道路有通行条件, 很多道路是单向通行的, 且很多高等级道路以隔离带划分为两条道路, 均是单向

收稿日期: 2004-02-25.

第一作者简介: 罗跃军(1978.6~), 男, 硕士研究生. 主要研究方向: 自主车辆导航数据模型. E-mail: yuejunl@kotei-navi.com

通行, 在拓扑网中表现为有向性. 不仅如此还有其他限制, 如大型货车禁止在细小道路中通行; 很多地方对时间上也有限制, 如某条道路早上 8:00 到晚上 9:00 货车禁止通行, 某条道路平时可以通车, 但是周末和节假日是步行街, 汽车禁止通行. 这些对最短路径计算均有很大的影响.

3) 交通网在道路交叉点处有各种各样的规制条件, 最为常见的就是禁止向左转弯, 如图 1 所示转弯 (1-0-2) 是禁止的, 所以在计算最短路径时不能从道路(1-0)转到道路(0-2). 有时并没有交通规制说向哪里转弯不行, 但是有潜在的禁止转弯规制. 比如图 2, 图中箭头表示单向通行方向, 因为 94-96 是单向通行, 只能向 96 方向通行, 所以不允许(95-96-94)这样的右转弯, 所有通过 96 点向 94 方向转弯的情况都是不允许的, 包括不允许(94-96-94)这样的 U-Turn, 同理, 禁止转弯(94-93-95).

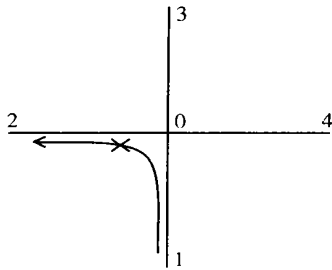


图 1 禁止转弯

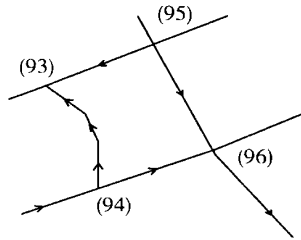


图 2 单向通行图

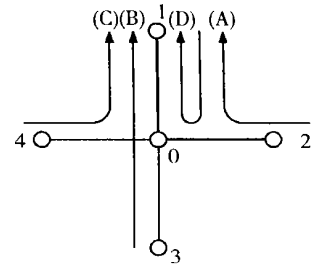


图 3 转弯规则关系图

2 建立交通仿真模型

交通仿真模型是在普通有向拓扑网的基础上建立起来的, 是以节点之间的关系做为拓扑结构. 道路的信息存储在节点与邻节点-节点-邻节点的关系中. 转弯规制存储在邻节点-节点-邻节点的关系中. 对于图 3 我们可以定义拓扑关系于表 1.

表 1 道路信息存储的拓扑关系表

自节点	邻节点	道路等级	距离	通行时间	通行条件	来自节点	转弯规制
0	1	等级 1(高速公路)	500 m	40 s	双向通行	1	禁止转弯
	2	等级 2(国道)	1 000 m	100 s	双向通行	2	特殊转弯规则编码
	3	
	4	

在表 1 中, 特殊转弯规则编码指的是对通行时间、通行车辆的限制, 通过该编码在另外一张数据表(本文略)中就可以查到该通行限制.

3 大范围交通网计算的问题

计算道路最短路径的经典方法是 Dijkstra 算法, 人们在长期的使用过程中感到其算法结构过于复杂且效率较低, 所以对 Dijkstra 算法进行改进产生了大量的 Dijkstra 变种算法, 从而使最短路径计算的效率有了很大的提高. 但无论是怎样的变种算法都需要建立全范围的道路拓拓扑网络, 并将这个范围内的拓扑关系映射到计算机内存中, 然后在内存中进行最短路径道路计算.

在现实情况下国家各经济部门经常需要进行大范围的交通网查询, 物流部门需要跨市跨省进行交通运输, 旅游部门需要到远离市区的旅游景点, 包括私人用户也经常需要驱车到另外的城市, 此时的道路计算所需要的拓扑网络就不能只限于一个城市或一块区域, 它涉及到的道路网络是大面积的. 以全国道路交通网为例, 如果要查询从武汉大学校门口到北京天安门的最短路径, 首先要建立武汉到北京区域范围内的道路拓拓扑网络, 此网络包括武汉和北京之间的所有城市道路网以及城市间的公路网, 此道路网络的节点数是百万级, 拓拓扑网络还包括节点和连接边的关系, 以及上述的连接边属性和转弯规制, 当然在实际应用中若涉及到其他城市数据量会线性增加, 若建立全国范围的道路拓拓扑网络, 节点数是千万级, 无论采用哪种

Dijkstra 变种算法,其拓扑网络的数据量都是海量级的,而普通 PC 机的内存非常有限,并且若在内存量更小的 PDA 这样的嵌入式操作系统上进行道路计算,把海量道路拓扑网映射到内存中,并进行计算是不大现实的.对于这样的大范围交通网的计算问题,文中给出以下解决方案.

4 解决方案

通常的最短路径算法是要建立全范围的拓扑网络,计算机的内存是存放不了全国范围这样级别的数据量.为此,分两部分解决此问题.第一是将大范围的拓扑网数据分 Mesh 存储,第二是建立道路网络的分层系统.

1) 将大范围数据分 Mesh 存储,把全国范围的数据按照一定的经纬度范围做 Mesh 划分,例如将经差 $(1/32)^\circ$ 纬差 $(1/48)^\circ$ 作为 Mesh 的划分单元.在此所划分的是源数据,无论源数据是什么形式,数据库也好,文件系统也好,首先按照 Mesh 划分开后进行存储.图 4 是显示一个城市的交通网数据按 Mesh 划分,实心点表示道路被 Mesh 分割后产生的分割点.这样计算机在计算道路时不必将全部的道路网数据映射到内存中,而是 Mesh 一一映射.道路计算时,首先将起点所在的 Mesh 映射到内存中进行局部道路运算,计算到 Mesh 边界点时,将该邻接的 Mesh 映射到内存中继续运算,把长时间没有参加运算的 Mesh 从内存中清除掉,通过相邻两 Mesh 的边界点之间的拓扑关系,可以讲各个 Mesh 连接起来组成一个大的拓扑网络.进行局部道路计算时可以利用优化后的 Dijkstra 变种算法.因此可以在小内存中进行大数据量处理,但若计算距离太远,就需要加载很多的 Mesh,在内存中加载数据量仍会很大,因此要和 b 部分配合使用.



图 4 城市道路按 Mesh 分割图

2) 将道路按级别分层,在交通仿真模型中提过不同等级的道路通行效率是不同的,道路分等级的功能不完全用于最短时间运算中,而且用于道路网络的分层.试验中将全国道路网络分为 3 层,1 层是最底层,即将最详细的道路加入网络中,只要是汽车能通行的道路均加入其中,见图 5.2 层是将一些细小道路删除,保留城市主干道、高速公路、国道等等,见图 6.3 层是最上层,只保留城市间的道路,如高速公路、国道等,见图 7.



图 5 第 1 层道路网



图 6 第 2 层道路网

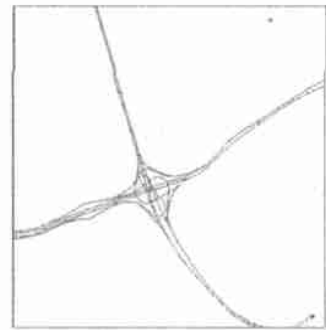


图 7 第 3 层道路网

这三层数据均按 Mesh 分割,但各层 Mesh 的大小并不一样,试验中 1 层 Mesh 大小为经差 $(1/32)^\circ$ 纬差 $(1/48)^\circ$,2 层 Mesh 大小为经差 $(1/8)^\circ$ 纬差 $(1/12)^\circ$,3 层 Mesh 大小为经差 $(1/2)^\circ$ 纬差 $(1/3)^\circ$.在城市内部小区域的道路计算中,只需要第 1 层 Mesh 数据,对于城市内大范围的道路计算需要第 2 层数据,对于城市间的道路计算需要第 3 层数据.同样以从武汉大学校门口到北京天安门的最短路径计算为例,需要通过第 1,2 层 Mesh 的数据计算出武汉大学校门口到武汉城市公路网北出口的最短路径 A,再通过第 3 层 Mesh

的数据计算出武汉到北京之间的最短路径 B, 最后再通过第 1, 2 层 Mesh 的数据计算北京城市公路网南入口到北京天安门的最短路径 C, 将路径 A, B, C 连接起来即可获得所需的最短路径.

3) 建立道路分层模型还必须有各层之间的联系. 如 b 部分的例子, 在计算过程中何时从第 1 层 Mesh 数据换到第 2 层 Mesh 数据, 为什么要在武汉城市公路网北出口处转换到第 3 层 Mesh 数据计算, 这都需要各层之间有所联系. 由于第 2 层道路数据均来自于第 1 层, 因此第 2 层的某节点 a 在第一层中一定有相同的节点 b, 那么就可以记录节点 a 和节点 b 的对应关系, 见图 8. 同理第 3 层和第 2 层也有相同的对应关系.

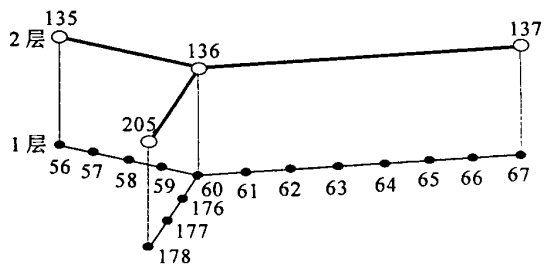


图 8 道路分层模型关系图

在进行道路计算时, 首先在第 1 层的数据中进行计算, 若加载了一定数目的 Mesh 的数据仍没找到终点, 则需要在加载的数据中找到一个与第 2 层有对应关系的节点, 例如图 8 中的 60 节点, 通过它转到第 2 层数据中的 136 节点, 以 136 节点作为起点在第 2 层中进行计算, 若还找不到终点, 就可以按照同样的方法转到第 3 层数据中计算. 若需要从低层转到高层中计算, 由于采用双向算法, 从起点和终点一起探索路径, 所以起点或终点都必须转入高层后再计算.

该解决方案只需要较少的系统消耗, 不仅可以在 PC 机上快速运行, 并且可以在低频 CPU、嵌入式操作系统上、少量内存的情况下较快的运算出跨城市的大范围的道路计算, 在商业应用上可取得很好的经济效益.

5 结束语

该算法作为大范围交通网的最短路径计算是比较成功的, 但还有需要改进的地方, 如 Mesh 的划分是按固定经纬度范围, 因此对于不同地区实际的范围不一样, 尤其在高纬度地区范围比较小, 道路数据量相对较小; 本算法中道路等级的划分是全范围同一的, 但实际上不同地区道路的等级和道路的重要性并不成正比, 在中部和东部地区高速公路和国道是高等级的, 即在第 3 层中, 但在西部地区可能一条普通公路就是贯穿某一地区的一条重要道路, 若把它放在第 1 层或者第 2 层就很可能导致计算的不正确, 这些都是今后要继续研究的问题.

参考文献:

- [1] 赵亦林. 车辆定位与导航系统[M]. 谭国真, 译. 北京: 电子工业出版社, 1999. 123~ 130.
- [2] 周捷. 基于神经网络的自主车辆导航路径计算[J]. 机器人, 1999, 21(5): 363~ 369.
- [3] 王伟, 徐吉谦, 杨涛, 等. 城市交通规划理论及其应用[M]. 南京: 东南大学出版社, 1998. 96~ 100.
- [4] 刁在筠, 郑汉鼎, 等. 运筹学[M]. 北京: 高等教育出版社, 2001. 198~ 206.