

车辆导航中基于约束条件的地图引擎和路径规划

张东^{1,2}, 钱德沛¹, 刘爱龙², 杨学伟²

(1. 西安交通大学电信学院, 西安 710049; 2. 西安测绘研究所, 西安 710054)

摘要: 构造了嵌入式车载实时语音导航系统约束条件集(RCS), 包含6个约束实体和6种实体约束关系, 在此约束条件集下, 提出了路径规划改进算法 Modify_Dij, 构建了支持动态索引机制的并行显示引擎(PDE)。比较了漫游状态下PDE和一般算法跨图幅响应时间以及 Modify_Dij 和经典路径规划算法的时间代价, 结果显示地图数据引擎PDE与路径规划改进算法 Modify_Dij 在时间代价方面具有明显优势。完成了一款支持全国路网、满足实用要求的车辆导航设备, 改进的路径规划算法和并行显示引擎满足设备实用性需要。

关键词: 约束条件; 路径规划; 显示引擎; 嵌入式系统; 车辆导航

Parallel Display Engine Building and Route Planning Under Restriction Conditions for Embedded Vehicle Navigation System

ZHANG Dong^{1,2}, QIAN Depei¹, LIU Ailong², YANG Xuewei²

(1. College of Telecommunication, Xi'an Jiaotong University, Xi'an 710049; 2. Xi'an Research Institute of Surveying and Mapping, Xi'an 710054)

【Abstract】 A restriction conditions set is constructed, which includes six restriction entities and six relationships among them, for embedded vehicle navigation system. Under this restriction conditions set, this paper presents a novel navigation route planning algorithm named Modify_Dij and builds a parallel display engine which can support dynamic indexing mechanism. The experiment results show that Modify_Dij and the parallel display engine have a good response time performance than others. They have been used in a real vehicle navigation equipment.

【Key words】 Restriction conditions; Route planning; Display engine; Embedded system; Vehicle navigation

随着导航地图数据的不断丰富, 车载导航市场的建立和不断完善, 导航系统相关的关键技术基本得到了解决, 支持智能语音输出的车载导航系统得到了越来越广泛的应用。车载导航设备在卫星通信、卫星定位和地理信息系统等相关技术的支持下, 可以根据实时交通状况进行大区域路径规划与实时语音引导, 满足了用户导航的基本要求^[1], 但在实际应用中显现了诸多不足^[2], 如数据调用慢、长距离路径规划慢、有时所选路径欠合理、多变的交通限制影响计算结果等。同时由于系统是在嵌入式系统下运行, 数据资源、计算资源、存储资源的有限性影响了系统的实施, 约束了系统的性能^[3], 如何在有限的资源条件下更高效地实现复杂的算法是导航技术的新课题。本文分析了车载导航设备研究过程中的诸多影响因子, 定义了一个约束导航仪性能的条件集 RCS(Restriction Conditions Set), 给出了约束实体和相互之间的约束关系, 据此对导航的两个最主要的核心技术——地图数据引擎和路径规划进行了优化, 提出了更有效的数据组织方式、数据驱动引擎、路线规划算法等。

本文基于引擎和路径规划研究完成了一款基于全国路网、超过150个城市的满足实用要求的导航仪产品 NAV-01, 支持触摸屏输入和语音导航, 芯片选用主频 206MHz 的 Interstrong ARM SA1110, 64MB SDRAM, 512MB CF, 直流 12V 供电, 屏幕分辨率为 320×240, 外形为宽 198mm、高 122mm、厚 37.6mm, 净重 600g。在实际应用中综合指标得到了明显的改进, 实用效果较好。

1 嵌入式车载导航中约束条件集 RCS 定义

按照实体-关系的描述方法, 定义约束条件集 $RCS=(E,$

$R)$, 其中的实体和关系如下:

定义 1 RCS 实体为一个二元组 $E=(E1, E2)$, $E1=(MR, CR, TR, DR)$, $E2=(DE, RM)$, 其中:

MR: 存储资源, 嵌入式环境下有限的存储资源, 一般为 64MB 或 128MB。

CR: 计算资源, 嵌入式环境主频低, 一般为 200MHz 左右, 存储空间小, CF 卡等外围数据读取慢。

TR: 交通限制, 全国路网中有复杂的交通限制规则, 特别是在城市内, 路网的拓扑关系更复杂且多变, 是导航性能限制的重要参数。

DR: 数据限制, 大数据量和不断更新的地图数据, 使路径选择的实时性和合理性受到了限制。

DE: 地图引擎, 对约束因子 MR、CR、DR 有依赖关系。地图数据引擎主要实现地图数据的组织和调度, 放大、缩小、漫游操作, 图幅自动切换与刷新等功能, 操作的数据对象包括显示、拓扑、索引等数据。动态性与并行调度是引擎中核心实现策略。

RM: 路径规划, 对 MR、CR、TR、DR 均有依赖关系。路径规划模块就是根据车辆当前位置、目的地和当前 TR 计算最值路径, 如最短、最优、收费最少等。数据对象为导航道路拓扑数据。

作者简介: 张东(1967-), 男, 博士生, 主研方向: 车载导航, 嵌入式 GIS 和网络数据分发; 钱德沛, 教授、博导; 刘爱龙、杨学伟, 工程师

收稿日期: 2006-05-19 **E-mail:** navgrid@163.com

定义 2 RCS 关系 $R=\{R_i\}$, R_i 组成实体中的所有关系, 其中:

$R_1=\{MR, DE\}$, 包括全国路网的导航数据量压缩后一般大于 500MB, 有限的存储资源要求数据组织的动态性和实时性。

$R_2=\{CR, DE\}$, 引擎对数据的调度和显示刷新有高的要求, 低的计算能力只能用更高效率的机制来保证调度和刷新的效率, 显示刷新每秒一次。

$R_3=\{MR, RM\}$, 路径规划特别是在大区域时, 参与计算的道路逾百万条, 不可能将数据直接调入内存计算, 在组织和算法设计上要具有动态性和优化处理。

$R_4=\{CR, RM\}$, 路径规划的实时性要求在偏离航线时显得很突出, 行驶中的车辆不能因为路径计算跟不上, 错过相应的路口, 使导航陷入计算陷阱。

$R_5=\{TR, RM\}$, 复杂的交通限制使常规的计算方法失效, 必须对算法进行优化。

$R_6=\{DR, RM\}$, 不断更新的道路网使路线规划和导航的合理性受到限制。

导航的技术环节很多, 但地图引擎和路径规划是最基本部分, 因此主要讨论 RCS 对它们的影响和优化。

2 基于 R_1 、 R_2 约束的并行地图引擎

动态数据调度与驱动策略: 按照 R_1 、 R_2 的要求, 数据的调度必须是小块、多次、实时和动态的, 两级动态索引是一种有效的方法:

(1)以 1 10 000 的图幅为单位进行索引调度;(2)以某一区域为单位形成文件,文件内记录是图幅;(3)一级索引文件,根据图幅号索引到区域文件和二级索引文件(以区域为单位);(4)通过二级索引文件得到地图数据的位置,二级索引中每个记录 24B,包括图幅号、记录起始地址、记录长度、是否有逻辑分块等。

为了突破 R_1 、 R_2 的约束,并行显示引擎 PDE 在完成了下述 3 种相关算法的基础上进行设计:

算法 1 重叠区域判断算法 `Overlap_Pre()`: 构造矩形 `Square(PointLU, PointRD)`, 使地图目标 `Objects \subset Square`, 求二者的重叠关系,当公共区域非空时,计算并显示该目标,减少了运算量,提高了显示速度。

算法 2 动态矢量数据裁减算法 `Object_Cut()`: 比较显示图形数据和显示缓冲区,超出图形进行裁减,具体:(1)在读取次数和数据量之间寻找一个动态平衡点,使每次装入的数据量不能太大。(2)采用层结构来组织和存储数据,减少数据冗余。

算法 3 显示内容的自适应配置算法 `Content_Con()`: 对空间数据根据不同的显示尺度进行分级和取舍,按照层次模型对数据进行组织和显示,解决缩放中固定要素带来的负载过大问题。比例尺越大显示的空间数据就会越详细。

在此基础上实现了 PDE 算法:

(1)内存中同时有 2 个位图在线,指针 `*Pointer1`、`*Pointer2` 指向它们,每个大小为 4×4 个 1 万图幅;

(2)当前显示的位图由状态变量 `BitMapInUse` 来表示,另一幅位图就是待更新的位图;

(3)位图生成中使用 `Overlap_Pre()`、`Object_Cut()`、`Content_Con()`;

(4)在漫游等显示操作时位图后台调度由触发和调度条

件 W_1 控制, W_1 实际上是两个位图的中心区域控制,当超出一个区域后就驱动另一个线程的图调度,后台生成位图;

(5)图幅在调入时和已经在内存中的图幅 ID 进行相关性比较,调入新图幅;

(6)图幅删除按照距离最远策略进行;

(7)位图切换由触发条件 W_2 控制,当屏幕要超出当前位图的范围时, W_2 激发另一个线程已经并行完成的位图,并切换更新屏幕。算法伪代码如下:

```
[PDE 算法]
Begin:
virtual BOOL ZBToMapID(double LL,double BB, CString
&name); //当前位置所在图幅号;
GetMapName(Cstring &MapName, &AdjustMapName);
//相邻图幅号;
virtual BOOL CompareToMem(CString &name, CBitmap*
Pointer1, Mem *p_Mem, String *MapDisplayNo1[16], String
*MapdiaplayLabel1[16], String *MapDisplayNo2[16]);
//通过索引找到相关图,与已经在内存中的图幅号比较,
//确定哪些图要读入,哪些可以被覆盖;
virtual BOOL ReadChoiceToMem(CString &name,
CBitmap* Pointer1); //选择读入内存;
CBitmap* CCreateBitmap(double CenterLongitude, double
Centerlatitude, long* xOffset, long* yOffset, String
*BitMapInUse);
{void Overlap_Pre(); void Object_Cut(); void
Content_Con();}
//根据当前使用位图区的标识生成位图*p_BitMap1 或
/**p_BitMap2;
void SwitchDisplayBitmap(CBitmap* m_pBitmap, long
x, long y, String *BitMapInUse);
//位图切换函数:将*BitMapInUse 所对应的位图切换显//
示在屏幕上;
End
```

图 1 给出了 PDE 算法和常规显示方法 Normal_D 算法在漫游状态下,超出位图区域生成新位图的时间比较,显然 PDE 在多线程并行机制支持下,通过动作预判和两个并行线程预生成位图,并用实时切换的方法,时间指标明显优于 Normal_D 算法。

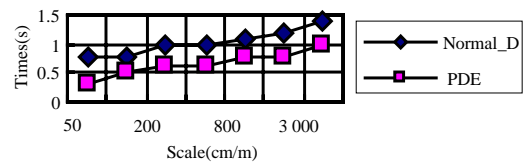


图 1 地图引擎显示时间比较

3 基于 R_3 - R_6 约束关系的路径规划

3.1 交通规制约束分析

交通规制约束在道路网中非常普遍,也直接影响路径规划算法,主要分为如下几类:禁止行驶类,复杂路口类,辅道等预判类。禁止行驶类包括禁止左转、右转、直行、调头、禁止双行等,算法实施中需要将连通性和行驶的方向进行相关计算,图 2 是复合路口中禁止左行的示意,约束条件表示为限制条件: $(R_3, C_2, R_2, C_1, R_5)$,图中 R、C、P 分别指路段、节点和限制路线。复杂路口类主要来自于道路关系表示的难度,复杂路口中的所有节点和路径必须在计算前根据交

通限制进行拓扑重组,对于最短路径计算算法在实际应用中面临的问题较多,也许距离较长的路更具有合理性。辅道、匝道等预判类要求算法具有前瞻性,对于前方的自然右转,可能需要提前 500m 并入辅道行驶,否则到路口难以实现右转。

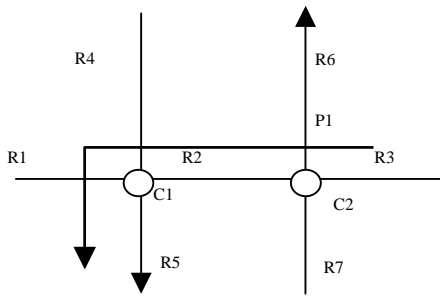


图 2 交通管制约束示意图

3.2 改进的 Dijkstra 算法

最优路径规划是在道路拓扑网络中,在已知起点和终点的情况下,找到一条代价最小的路径^[5]。Dijkstra 算法是导航路径规划中被普遍应用的广度搜索算法,步骤是从初始节点按顺序扩展至下一级子节点,然后每次从子节点最前面的节点扩展,重复上述操作,直到找到目标节点。

研究发现该算法不实用,计算时间和路径合理性没有得到保障,如在中等配置的嵌入式环境下对北京市区 8 幅 1 10 000 导航图进行任意两点之间的路径计算,不做优化的 Dijkstra 算法直接计算时间均在 5min 以上,根本不能满足实用的要求。改进的算法 Modify_Dij 是一种基于通畅指标的算法:在道路拓扑网络中,在已知起点和终点的情况下,找到一条路径,满足距离最短、时间最短、费用最低等条件,算法基本思想如下:

[Modify_Dij 算法]

Input: 参量 {CA, PB, RND, RLB, RNI, RTR}

Output: 规划路径组 RG

Begin:

构造节点临时集合 S1 和永久集合 S2,生成的节点 P1 在 S1 和 S2 中依距离值排序;

每次新展开节点按照交通管制依照高等级道路优先规则,同时兼顾距离阈值,将道路等级和距离综合为一个比较指标 Z;

若发现在 S1 中存在同一节点,比较指标 Z 决定是否进行替换;

每次新展开节点若发现在 S2 中存在同一节点号,跳过该节点;

偏离航线时在偏离点和目标点之间寻找一个中间点,中间点位于已经计算的路径 RR 上;

按照步骤 ~ 计算当前点和中间点之间的路径 RR',从 RR 上截取中间点和目标点之间的路径 R'';

新的 RR=RR'+R'';

在 RR 上进行实时导航,到达终点则结束;

偏离航线转步骤 执行;

End

需要说明的是在实际行驶过程中,路径不能成为驾驶员

的约束,偏离航线是经常性事件,当偏离航线时,需要重新计算新的路径 RR,在 Modify_Dij 算法的基础上,路线的重新规划算法变得简洁,如步骤 ~ 所示。

图 3 给出了算法和常规算法的时间比较,显然 Modify_Dij 算法具有较好的时间成本。如从北京市中心到广州市的路径计算时间略大于 5s,完全可以满足实用要求。需要说明的是测试的起始点相同,但对于不同的测试目的地,时间成本除了和距离有关外,还和测试目的地的位置有关,如目的地是广州市外围的某一点和广州市中心的一点,尽管在距离方面相差很小,但由于市中心道路复杂,数据量陡增,在计算时间上会明显增加,本测试数据没有反映这一点。

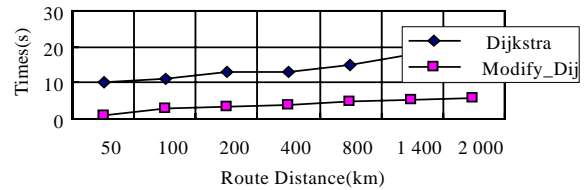


图 3 路径规划时间比较

4 结论

实验表明了以下几点:

(1)建立的约束条件集 RCS 清楚地反映了嵌入式车载导航产品的技术难点,给出了 6 个 RCS 实体和 6 种实体约束关系,给出了导航技术主要技术环节地图引擎和道路规划算法的约束条件和相应的算法,算法具有针对性和实用性。

(2)动态索引机制的建立与应用减少了对内存的过度依赖,也适应导航数据边采集边应用的现实状况,数据的扩充对代码和内存等的影响小,图的查询速度也大大提高,开机时将索引数据和当前位置相关图一次调入内存的时间少于 5s(S=200m),在后续应用中不需要重新载入。

(3)多线程显示引擎 PDE 算法与常规显示算法 Normal_D 相比,具有明显的时间优势,说明 PDE 实现技术可行性强。

(4)改进的算法 Modify_Dij 比常规的 Dijkstra 算法在时间方面有明显的优势,实时性高,特别是在城市复杂路网的情况下,偏离航线后的实时重新计算很多,Modify_Dij 算法显得更加重要。

在后续研究工作中还需要加强规划路径实用性和合理性研究,特别是在城市路网日趋复杂、道路不断更新情况下,合理性会越来越得到重视。

参考文献

- 李德仁,李清泉.论空间信息与移动通信的集成应用[J].武汉大学学报,2002,27(1):1-7.
- 赵亦林.车辆定位与导航系统[M].北京:电子工业出版社,1999.
- Zhang Dong, Liu Ailong, Zhang De. The Present Situation of Vehicle Navigation Techniques in China and the Multi-topological Formation and Route Computing of National Navigation Map[C]// Proc. of the 22th International Cartographic Conference, 2005.
- 郭建忠.系列比例尺条件下海量数据的快速显示[J].测绘学院学报,2005,22(2):136-138.
- 张其善,吴今培,杨东凯.智能车辆定位导航系统及应用[M].北京:科学出版社,2002.