

一种基于 FCD 的城市道路车流速度估计算法

殷伟¹, 郭璘^{2,3}, 方廷健², 王群京¹

(1. 合肥工业大学电气与自动化工程学院, 安徽合肥 230009; 2. 中国科学院合肥智能机械研究所, 安徽合肥 230031;
3. 中国科学技术大学自动化系, 安徽合肥 230027)

摘要:针对在城市路网中大面积采集交通信息所存在的问题,提出了一种基于浮动车辆数据(floating car data, FCD)的城市道路车流速度估计算法,并通过 FCD 系统进行交通速度信息的采集. 速度估计采用了一种改进的弧段 Dijkstra 路径寻优算法,在图的弧段-弧段存储结构中考虑了城市路网限制性道路交通信息的表达,并将路网的一些先验信息引入弧段关系结构中,提高了搜索效率和准确性. 实验结果论证了所提出方法的有效性和可靠性.

关键词:浮动车辆数据; 车流速度估计; Dijkstra; 先验信息

中图分类号: U491 **文献标识码:** A

An FCD based vehicle speed estimation algorithm in urban road networks

YIN Wei¹, GUO Lin^{2,3}, FANG Ting-jian², WANG Qun-jing¹

(1. School of Electrical Engineering and Automation, Hefei University of Technology, Hefei 230009, China;
2. Institute of Intelligent Machines, Chinese Academy of Sciences, Hefei 230031, China;
3. Department of Automation, University of Science and Technology of China, Hefei 230027, China)

Abstract: To solve the problems for large-scale traffic information collection in urban road networks, an FCD (floating car data) based vehicle speed estimation algorithm was proposed and traffic speed information was collected via FCD system. The speed estimation adopted a modified arc-based Dijkstra route searching algorithm. In this algorithm, a prior knowledge and the restricted traffic information of the road networks were included in the arc relation structure, thus improving the searching efficiency and accuracy. The results from experiments conducted in the city of Ningbo show the effectivity and reliability of the proposed methods.

Key words: floating car data; vehicle speed estimation; Dijkstra; prior knowledge

0 引言

交通速度(或称路段平均速度)是交通信息采集要获得的主要数据之一,是评价道路交通运行状况、拥挤水平的重要指标. 目前已有多种交通速度信

息采集方法,如检测线圈、雷达速度计^[1]、视频识别^[2]以及航空影像等,这些方法通常会因为设备成本过高,受工作环境的限制等原因,无法在城市大规模采集交通速度信息,难以满足先进的交通管理系统与出行者服务系统的需求.

收稿日期:2006-09-30;修回日期:2007-02-27

基金项目:国家自然科学基金(60173050)资助.

作者简介:殷伟,男,1962年生,博士生.研究方向:智能交通系统.

通讯作者:王群京,博士/教授. E-mail: wqunjing@sina.com

移动型交通检测器是动态交通流信息采集的主要技术^[3],不但可以获得行程时间、行驶速度等路段交通流参数,而且还具有良好的扩展性和经济性.本文研究了一种基于浮动车辆数据(floating car data, FCD)^[4]的移动型交通检测器,在分析其特点及工作原理的基础上,提出了一种基于 FCD 的车流速度估计算法,并通过实例分析了本文方法在城市大规模交通速度信息采集中的应用.

基于 FCD 的移动检测器动态估算路网中各路段双向的平均车流速度,需要求解所有移动车辆前后两个 GPS 位置的实际路径,并根据路径长度和时间差求出所经过路段的速度,最后进行统计平均求出任意路段的双向平均车流速度.因此,快速有效的路径搜索算法是车流速度估计算法的关键.当起点和终点距离比较近时,通常可以用两点之间的最短路径近似代替其实际路径.最短路径选择算法主要包括图论领域的 Dijkstra 算法^[5]、Floyd 算法^[6]、智能计算领域的遗传算法^[7]、神经网络法^[8]以及人工智能领域中的深度优先、广度优先和其他启发式搜索算法等.这些算法中就通用性、完备性和算法效率而言,Dijkstra 算法相对比较优秀.

王杰臣等^[9]提出的图的节点-弧段联合结构表示法,把一幅图抽象为{节点集合} + {弧段集合},节省了存储空间,提高了速度,为解决最短路径问题提供了切实可行的算法.然而该方法难以表达实际应用中城市路网一些常见的限制性道路交通信息(如禁行、禁左、禁右等信息).我们在传统 Dijkstra 算法的基础上,采用图的弧段-弧段结构表示法存储路网结构,即路网由相互关联的有向弧构成.在实际应用中,为便于在数据库中存储路网信息以及进行地图匹配,非直线路段通常被分为若干子路段(子路段是直线段,可以由它起点和终点的经纬度坐标直接表示).这样,当起点和终点在同一较长路段上不同子路段上时,其路径通常可以看作是确定的,我们可以把这些常用的确定的最短路径信息作为先验信息存储在数据库中,并通过弧段关系保存在路网中,在路径搜索时加以调用.由于采用了图的弧段-弧段存储结构,并充分利用了实际路网的先验信息和一些限制性道路交通信息,本文方法的路径搜索效率和准确性得以提高.

1 基于 FCD 的移动型交通检测器

基于 FCD 的移动型交通检测器是一种将浮动

车辆数据(包括浮动车辆的 GPS 定位数据、时间信息和车辆 ID 号信息等)作为交通参数的移动型交通检测器.通过大量分部在城市道路上的此种移动型交通检测器定时发送变化的时间信息和 GPS 经纬度信息(如图 1 所示),来估算车流速度,从而采集城市道路交通速度信息.

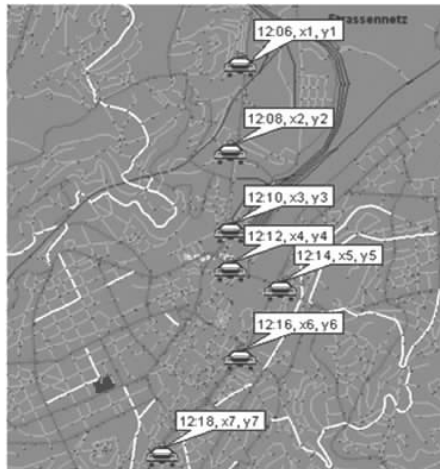


图 1 基于 FCD 的移动型交通检测器

Fig. 1 A mobile traffic detector based on floating car data

相对于其他固定式交通信息检测设备,基于 FCD 的移动型交通检测器有以下优点:

(I) 覆盖范围广.固定式检测器只能检测某些点的交通信息,而移动型交通检测器,不仅能覆盖点、线,而且能覆盖到区域.因为大部分浮动车辆都是出租车,目前在各类大城市中这类交通工具的数量以及每年行驶的公路里程数十分巨大,能够完全覆盖城市路网.

(II) 投入成本低.由于在 FCD 系统中共享了出租车调度管理系统的设施,大大减少了投入费用和维护费用.

(III) 采集的基础数据信息量丰富,数据精度高.根据采集的实时基础数据,能够比较可靠的计算出路段通行速度、出行时间等信息.

(IV) 可以实现对交通信息全天候、实时采集处理.

正因为上述优点,我们可以通过大面积分布在城市路网上的装有 GPS 终端设备的出租车移动型交通检测器定时发送基础数据(包括 GPS 定位信息、时间信息以及车辆 ID 号等)到公路交通运输管理处,并连接到 FCD 服务器,FCD 服务器在后台通过速度估计算法对采集的基础数据进行计算处理,从而实现大规模城市道路交通速度信息的采集.

基于 FCD 的交通信息采集系统主要由移动型

交通检测器系统、数据通信网络、FCD 服务器、中心导航服务器以及终端显示设备组成,系统结构如图 2 所示.该系统通过基于 FCD 的移动检测器采集基础数据,获取实时交通速度信息,并利用速度信息实现交通信息发布、出行时间预测、交通诱导和交通管理等功能.

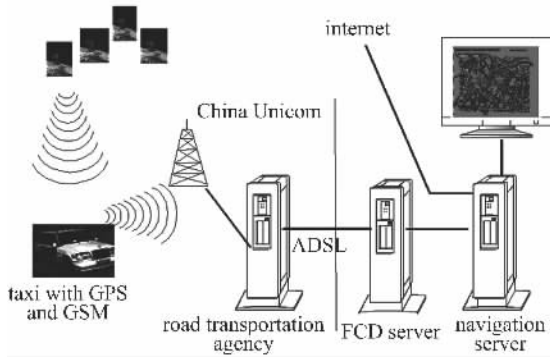


图 2 浮动车辆数据系统

Fig. 2 The system configuration of the FCD system

2 基于 FCD 的车流速度估计算法

FCD 服务器采用速度估计算法,每隔一定时间对移动型交通检测器采集的定位数据进行计算处理,得到路网中所有子路段的交通速度信息.在速度估计算法中,我们通过弧段 Dijkstra 最短路径算法求解移动车辆前后两个位置的准确路径,并通过路径长度和时间差求解所经过的路段的速度,最后估算路网中所有子路段的平均车流速度.

2.1 基于弧段 Dijkstra 的路径寻优算法

在图的弧段-弧段联合结构中,城市路网由相互关联的弧段构成,每一个弧段都是单方向的,即起点到终点是可以通行而终点到起点是无法通行的.两个弧段的关联通过一个弧段关系来表示.图的弧段-弧段结构表示如下:

```
struct Arc (弧结构)
{ int ArcID; // 弧段 ID 号
  int ArcRelationNum // 关联的弧段关系数目
  int* ArcRelationID // 关联的弧段关系 ID 号
  int Start_pt_ID; // 起点 ID 号
  int End_pt_ID; // 终点 ID 号
  int Length; // 弧段的长度
  bool Enable; // 弧段是否可以通行
  string RoadName; // 弧段所在路段名
};

struct ArcRelation(弧段关系)
```

```
{ int ArcRelationID; // 弧段关系 ID 号
  int Start_Arc // 起始弧段 ID 号
  int End_Arc // 终止弧段 ID 号
  int* WayArcs // 途经弧段 ID 号
  int WayLength // 途经弧段长度
  bool Enable; // 是否可以通行
};

struct Network (网络图结构)
{ int ArcNum; // 图的弧段数量
  int ArcRelationNum; // 图的弧段关系数目
  Arc* arcs; // 图包含的弧段集
  ArcRelation* arcrels; // 图包含的弧段关系集
}
```

当弧段关系中起始弧和终止弧直接关联时,途经弧段集为空;否则,它们为虚拟的弧段关系,途经弧段集不为空.通常,我们把路网中一些常用的确定的最短路径信息保存在虚拟弧段关系中.路段是否禁行可以通过弧度的 Enable 属性表示,路段的禁左禁右等信息则通过弧度关系的 Enable 属性表示.如图 3 所示,A 点到 B 点的最短路径信息以及濠河路到大沙泥街的禁止左行信息等都可以通过弧段关系表达.这样,从 A 点到 B 点的路径搜索过程只需要扩展一个节点 A,而不是扩展 6 个节点,搜索效率得以提高;禁行禁左禁右等限制性道路交通信息的利用则避免了可能出错的路径信息,保证了路径搜索的准确性.

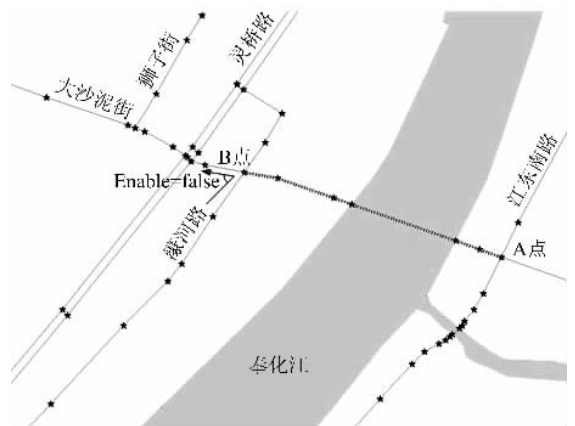


图 3 路网先验信息和限制性道路交通信息的利用
Fig. 3 The use of a prior knowledge and the restricted traffic information of the road network

搜索算法描述如下:

假设每个弧段都有一对标号 (D_i, F_i) , 其中 D_i 是最短路径上从起始弧 S 到弧 i 的路径长度; F_i 则是最短路径中弧 i 的前一个弧段(或弧段集). 我们

求解从起始弧 S 到终止弧 T 的最短路径上依次经过的弧段 $\{P_i, i=1, 2, \dots, n\}$.

(I) 初始化. 设置: ① $D_S = l_S, F_S$ 为空, 其中, l_S 是弧 S 的长度; ② 所有其他弧: $D_i = \infty, F_i = \{\}$; ③ 标记起始弧 S , 其他所有弧设为未标记的, 记 $k = S$.

(II) 扩展弧段 k . 根据弧段 k 关联的弧段关系, 找到与之相连的未标记的弧 i , 并根据下式更新 D_i :

$$D_i = \min[D_i, D_k + l_i + l_{ki}]. \quad (1)$$

式中, l_i 是弧段 i 的长度, l_{ki} 是从弧段 k 到弧段 i 的弧段关系中途弧段的长度.

(III) 选取下一个待扩展的弧段. 从所有未标记的弧段中, 选取 D_i 中值最小的一个, 如果满足

$$D_i = \min(D_i, \text{所有未标记的弧段 } i), \quad (2)$$

则设置 $k = i$, 且 k 为已标记的弧段.

(IV) 根据弧段 k 关联的弧段关系, 从已标记的弧段中找到 k 的关联弧段 s 以及从弧 s 到弧 k 的途经弧段 m , 设置 $F_k = \{s, m\}$.

(V) 如果 $k = T$, 则转到 (VI) 继续; 否则, 转到 (II) 继续.

(VI) 从终止弧 T 开始, 在已标记的弧段中, 依次找出它的前一个弧段 (或弧段集), 直到出现起始弧 S . 从起始弧 S 到终止弧 T 的最短路径就是前面依次经过的弧段 $\{P_i, i=1, 2, \dots, n\}$.

(VII) 算法结束.

2.2 车流速度估计算法

基于 FCD 的车流速度估计算法包括以下步骤:

(I) 数据查询

根据 GPS 定位信息的发送时间从 FCD 数据库中查询当前时刻前一段时间内的有效 GPS 定位数据.

(II) 地图匹配

将查询得到的所有 GPS 定位数据映射到电子地图上对应弧段上的某一位置.

(III) 路径求解

车辆 j 返回的前后两个点通常距离很短, 我们利用两点在地图上的映射位置, 通过 2.1 节介绍的弧段 Dijkstra 路径寻优算法求解出该车在这段时间内的具体路径 $\{P_i, i=1, 2, \dots, n\}$. 这里 P_i 为该车所经过的第 i 个弧段的标号.

(IV) 出行时间计算

在得到车辆 j 所经过的前后相邻两点的路径后, 根据下式求出该车通过弧段 P_i 的出行时间:

$$t_{ij} = \frac{\Delta t_j \times l_i}{\Delta d_j}. \quad (3)$$

式中, t_{ij} 为车辆 j 在弧段 P_i 上的出行时间, Δd_j 为路径长度, Δt_j 为时间差, l_i 为弧段 P_i 的长度.

(V) 速度计算

弧段 P_i 的长度除以其平均出行时间就得到弧段 P_i 上车辆的平均速度, 如下式:

$$\left. \begin{aligned} V_i &= \frac{l_i \times n_i}{\sum_{j=1}^{n_i} t_{ij}}, \text{ if } n_i \neq 0; \\ V_i &= \bar{V}_i, \text{ if } n_i = 0. \end{aligned} \right\} \quad (4)$$

式中, V_i 为弧段 P_i 的平均速度, l_i 为弧段 P_i 的长度, t_{ij} 为第 j 辆车在路径中弧段 P_i 上的出行时间, n_i 为弧段 P_i 上参与计算的车辆数目. 这里, 当 $n_i = 0$, 即该路段上没有数据覆盖时, 我们用历史积累的一周不同时间段的历史平均速度进行补充.

3 实验结果

我们在中国第一个实施 FCD 系统^[10]的城市宁波进行了算法测试. 分布在宁波市区的 3 219 辆出租车移动检测器每隔 1 min 发送一次 GPS 定位信息到 FCD 服务器, FCD 服务器根据本文提出的速度估计算法每隔 5 min 对当前前 10 min 内的基础数据计算处理. 计算结果显示: 在具有 12 944 个节点和 22 964 个弧段的宁波市区路网上, 服务器 (其主要配置: CPU 主频 2.6 GHz/内存 1 GB) 平均计算时间 (不包括查询 GPS 数据库和写速度数据到数据库的时间) 为 26.8 s, 满足实时性要求.

我们在宁波市区的两条主干道上中山路和中兴路上通过实地跑车实验比较了数据库中的标示速度和实测速度, 并比较在两条道路上的出行时间, 评估本文提出的速度估计算法的性能. 实验在一周内让出租车在这两条主干道上出行 20 次, 同时通过 GPS 鼠标和秒表记录下通过每条子路段的速度和时间, 并同 FCD 服务器中的速度值以及根据速度值计算出的出行时间进行了比较, 实验结果如表 1 所示.

从表 1 我们可以看出, 通过基于 FCD 的移动型交通检测器采集基础数据, 并利用本文给出的速度估计算法计算速度, 70% 的出租车速度有 50% 以上的符合度, 75% 的出租车速度有 30% 以下的差异度, 72.5% 的出租车出行时间有 20% 以下的错误率 (符合度、差异度和错误率的定义见表 1). 该结果表明, 通过基于 FCD 的移动检测器采集基础数据并基

于本文给出的速度估计算法估计速度以及根据该速度进行出行时间预测是合理的。

表 1 评估实验结果

Tab. 1 The evaluation results

比较点	分布情况	中山路		中兴路	
		次数	比例	次数	比例
符合度	≥60%	4	20%	9	45%
	50%~60%	10	50%	5	25%
	40%~50%	4	20%	3	15%
	30%~40%	2	10%	2	10%
估算速度和 实测速度比较	<30%	0	0%	1	5%
	≥50%	0	0%	1	5%
差异度	40%~50%	2	10%	2	10%
	30%~40%	3	15%	2	10%
	20%~30%	12	60%	4	20%
	<20%	3	15%	11	55%
出行时间比较 错误率	>50%	0	0%	1	5%
	40%~50%	0	0%	1	5%
	30%~40%	2	10%	3	15%
	20%~30%	1	5%	3	15%
	10%~20%	2	10%	4	20%
	<10%	15	75%	8	40%

【注】①符合度=一次试验估算速度与实测速度基本符合的路段数/总路段数;②差异度=一次试验估算速度与实测速度存在较大差异路段数/路段总数;③错误率=|实际出行时间-估计出行时间|/实际出行时间;④估算速度与实测速度基本符合指估算速度与实测速度差值小于5公里/小时;⑤估算速度与实测速度存在较大差异指估算速度与实测速度差值大于5公里/小时。

考虑到实验中移动检测器发送 GPS 数据的时间间隔较长(为 60 s),实时性不够强,如果能够适当缩短发送 GPS 数据的时间间隔,提高发送信息的实时性,本文方法估计的速度值将会更接近实测值,准确性和可靠性将会进一步提高。不过,如果时间间隔过短(例如小于 10 s),当所有车辆都向管理系统报告其位置时,将会出现通讯堵塞的问题以及通信费用过高的问题。

另外,由于出租车驾驶员比其他驾驶员反应更快、更善于换道,使用出租车作为移动检测器所采集的速度数据一般会比实际速度要快,因此如果能使用更多的普通车辆作为移动检测器,使移动检测器的类型更加多样化,本文方法采集的速度信息将会更接近实际速度值。

4 结论

本文研究了一种基于 FCD 的移动型交通检测器及车流速度估计算法。通过该移动检测器采集基

础交通数据,并通过车流速度估计算法估算速度,能够较为准确地获取城市实时道路交通速度信息。该方法为大规模采集城市道路交通信息提供了一种新的思路,并很好地解决了目前车载导航系统难以实现准确的时间最优路径设计的问题,有很广泛的应用前景。

参考文献(References)

- [1] 杨兆升. 城市交通诱导系统 [M]. 北京:中国铁道出版社, 2004: 42-54.
- [2] Liu Bo, Zhou Xi-han, Zhou He-qin. Vehicle detection and recognition in multi-traffic scenes[J]. Journal of University of Science and Technology of China, 2004, 34(5): 599-606.
刘勃,周西汉,周荷琴. 混合交通场景中的车辆自动检测和识别[J]. 中国科学技术大学学报, 2004, 34(5): 599-606.
- [3] Zhang Jing, Cai Bo-gen, Wu Jian-ping. Study of probe vehicle technology[J]. Journal of Northern Jiaotong University, 2003, 27(3): 80-83.
张静,蔡伯根,吴建平. 移动检测技术的研究[J]. 北方交通大学学报, 2003, 27(3): 80-83.
- [4] Fastenrath D U. Floating car data on a larger scale [R]. ITS-World Congress. DDG Gesellschaft fr Verkehrsdaten mbH, 1997.
- [5] Bondy J A, Murty U S R. Graph Theory with Applications [M]. New York: The Macmillan Press, 1976: 1-22.
- [6] Floyd R W. Algorithm 97: Shortest path [J]. Communications of the ACM, 1962: 5(6):345.
- [7] Chang W A, Ramakrishna R S. A genetic algorithm for shortest path routing problem and the sizing of populations [C]// IEEE Trans on Evolutionary Computation, IEEE, 2002, 6(6): 566-579.
- [8] Araújo F, Ribeiroand B, Rodrigues L. A neural network for shortest path computation [C]// IEEE Trans on Neural Networks. IEEE, 2002, 12 (5): 1 067-1 073.
- [9] Wang Jie-chen, Mao Hai-cheng, Yang De-zhi. United structure of point-arc for network graph and its application in giss shortest path searching[J]. Acta Geodaetica et Cartographica Sinica, 2002, 29(1): 47-51.
王杰臣,毛海城,杨得志. 图的节点-弧段联合结构表示法及其在 GIS 最优路径选取中的应用[J]. 测绘学报, 2002, 29(1): 47-51.
- [10] Wang Jia-jie, Fang Ting-jian, Jchafer R P. China first FCD application at Ningbo city [C]// Proc of 12th World Congress on Intelligent Transport Systems and Services, San Francisco, USA, 2005: 1 250-1 256.