

面向路网连通关系的车辆轨迹数据查询技术研究

李响¹⁾ 刘正军²⁾ 林瑋³⁾ 张喜慧⁴⁾

¹⁾ (华东师范大学地理信息科学教育部重点实验室, 上海 200062)

²⁾ (中国测绘科学研究院对地观测技术国家测绘局重点实验室, 北京 100039)

³⁾ (香港中文大学太空与地球信息科学研究所, 香港) ⁴⁾ (美国北阿拉巴马大学计算机信息系统系, 美国 35632)

摘要 定位及通信技术的发展使获取和传输移动车辆的轨迹信息成为可能。从车辆轨迹数据中可以提取出大量的交通信息, 这些信息是对交通状况的一个直接和全面的反映。但由于轨迹数据固有的网络特性及时空特性, 使针对此类数据的查询成为一个难点, 其中关键技术包括建立用于车辆轨迹数据的索引结构及其对应的轨迹数据查询语言。在比较现有轨迹数据索引结构的基础上, 对轨迹数据查询语言的分类及其与索引结构之间的调用关系进行了初步的探讨, 提出了面向连通关系的查询语言(connectivity-oriented query language, CQL)的定义、分类及实现方法, 并简要讨论了车辆轨迹数据查询原型系统的结构和实现方法。

关键词 轨迹数据 索引结构 查询语言 路网连通关系

中图法分类号: P237 文献标识码: A 文章编号: 1006-8961(2009)07-1245-06

Connectivity-oriented Vehicle Trajectory Data Query

LI Xiang¹⁾, LIU Zheng-jun²⁾, LIN Hui³⁾, ZHANG Xi-hui⁴⁾

¹⁾ (Key Laboratory of Geographical Information Science, Ministry of Education, East China Normal University, Shanghai 200062)

²⁾ (Key Laboratory of Earth Observation Technology of State Bureau of Surveying and Mapping,

Chinese Academy of Surveying and Mapping, Beijing 100039)

³⁾ (Institute of Space and Earth Information Science, The Chinese University of Hongkong, Hongkong)

⁴⁾ (Department of Computer Information Systems, University of North Alabama, Florence, Alabama, USA 35632)

Abstract The advances in positioning and communication technologies make it possible to collect and transfer trajectories of moving vehicles. In addition to some basic applications such as navigation, traffic data can be extracted from vehicle trajectories, such as average moving velocity and travel time on a road segment in a given time period. The network and spatio-temporal characteristics of trajectories, however, make it difficult to evaluate queries applied to them. As a result, many studies have focused on developing index structures and methods for trajectory data. However, few of them have presented the corresponding data query language. In this paper, we discussed some issues related to trajectory data query language, including the definition of the connectivity-oriented data query language (CQL), the classification of CQL, and its implementation based on an index structure proposed by previous study. We further discussed a prototype for vehicle trajectory data query based on CQL in terms of its structure and implementation.

Keywords trajectory data, index structure, data query language, connectivity

基金项目: 国家自然科学基金项目(40701142)

收稿日期: 2009-04-21; 改回日期: 2009-04-25

第一作者简介: 李响(1975~), 男, 副教授。2004年于香港中文大学获地理信息系统专业博士学位。主要研究方向为交通地理信息系统、时空数据模型、空间数据挖掘等。已发表相关学术论文40多篇。E-mail: xli@geo.ecnu.edu.cn

1 引言

定位及通信技术的发展使移动对象轨迹信息的获取和传输成为可能。最常见的方式是,机动车通过 GPS 接收器及无线通信网络,将其当前的位置信息和需求提供给某个位置相关服务(LBS)的提供商,进而获得所需服务,如获知最近的加油站位置及供油情况、前往目的地的最佳路径、前方的交通状况等。除实现上述面向车辆的定位或导航等基本功能外,从车辆轨迹数据中还可以提取出大量的交通信息,如速度、路径选择等。这些信息是对交通状况的一个直接和全面的反映,也是对传统基于固定点的交通数据收集方法的一个重要补充。因此,从交通管理和规划的角度出发,对收集的车辆轨迹数据进行有效管理并从中提取出有用的信息是非常有实际意义的。车辆轨迹数据物理上由大量时间序列点构成,且每个点都具有时空特征,代表当时车辆所处位置。由于通常情况下车辆在道路网络中行驶,因此,又称之为网络限制下的时间序列点数据,点与点之间的时空间隔由位置接收器的采集频率与车辆行驶速度决定。

随着时间推移和提供轨迹信息的机动车数量不断增加,时间序列点数据量必然也将快速增加,尽管目前廉价的存储器和大型商业数据库管理系统已经使海量数据的存储问题不再成为一个挑战,但由于这类数据固有的网络特性及时空特性,使针对此类数据的应用成为一个难点。比如,希望从大量点数据中查询“在 2008 年 6 月 8 日早上 8 点至 10 点间自火车站出发经中山北路、中山西路、机场高速到达机场的平均旅行时间是多少?”,此类查询结果对交通管理和出行计划的制定都具有直接的意义,但目前的地理信息系统产品尚无法有效完成这一查询。因为,首先,网络限制下的时间序列点数据是一个全新的时空数据类型,尚缺乏成熟的支撑技术,如时空索引方法;其次,上述案例属于时空信息查询,而且空间查询条件是面向网络连通关系的^[1],对此类查询条件,目前的地理信息系统产品缺乏有效的描述手段;此外,时间序列点数据仅是车辆行驶过程的离散采样,而且通常以经纬度为空间参考,其蕴涵的连续轨迹信息以及与道路网络之间的位置关系并非直观存在或直接存储于数据库中的,进一步增加了查询的难度。因此,需要设计用于轨迹数据的索

引结构,而为使索引结构得到真正的应用,还需要定义相应的查询语言。相比较与轨迹数据索引结构的快速发展,轨迹数据查询语言方面的研究略显滞后。鉴于此,本文:对现有的众多索引结构加以比较,然后在此基础上对轨迹数据查询语言的分类及其与索引结构之间的调用关系进行初步的探讨。

2 轨迹数据索引结构

基于时间序列点或称为轨迹数据的查询描述既包括空间查询条件也包括时间查询条件,而且空间查询条件往往由网络路径构成(如上节中所列举之范例),因此需要支持网络拓扑关系的时空索引方法。

2.1 用于欧几里得空间轨迹的索引结构

在地理信息系统中,传统空间数据索引方法,如 R-tree^[2], R*-tree^[3], Grid File^[4], 及 K-D-B-tree^[5], 不能处理在轨迹数据中特有的时间信息。尽管存在一些改进方法,但很少是针对受道路网络限制的轨迹数据,他们中大部分假设车辆或其他移动对象是在一个欧几里得空间中自由移动的,而且仅仅索引当前的位置信息,如 RT-tree^[6]、3D R-tree^[7]、HR-tree^[8]、MV3R-tree^[9]、2 + 3 R-tree^[10] 以及 2-3 TR-tree^[11] 等。其他一些方法考虑了连续的轨迹数据,如 STR-tree^[12]、TB-tree^[13] 以及 Chebyshev 多项式^[14] 等。但由于上述这些方法都忽略了移动对象的空间限制(如道路网络),因此他们不能有效地索引道路网络空间中的车辆轨迹数据,也无法解决以网络路径为空间索引条件的信息查询问题。而且,这些方法都是基于一个单一的树状索引数据结构,当数据量持续增加的时候,这一树状结构的更新和维护是相当低效的。

与上述方法不同,SETI 方法^[15] 和 SEB-tree^[16] 把轨迹数据分配到事先划分好的空间静态网格中进行索引。这种思想有效地减少了维护索引结构的负担,但其仍然无法直接应用于道路网络中的轨迹数据。

2.2 用于网络空间轨迹的索引结构

Pfoser 等人提出了一种基于网络的轨迹数据索引方法^[12]。这种方法把 2 维的道路网络转化成 1 维的线段,简化了索引结构,但 2 维道路网络的拓扑信息在转化过程中丢失了,使其无法解决基于网络的信息查询问题。之后出现的 FNR-tree^[17]、MON-

tree^[18]以及 NB-tree^[19]采用了相似的原理,通过建立一组小型索引结构的方法,简化整个索引结构的维护,同时便于索引方法的实现。

Chen 等人采用了相似的思想,提出一个动态数据结构^[20],称作自适应单元,这种结构根据相似运动模式将相邻对象分组,然后,在自适应结构基础上建立路网上的空间索引。然而,上述这些方法也同样没有考虑道路网络的拓扑特性,因此,很难解决基于拓扑连通关系的信息查询问题。

Li 与 Lin 提出了一种基于拓扑的混合式索引结构^[1] (topology-based mixed index structure, TMIS)。其基本思想是建立一个由多个小的索引结构组成的混合索引结构,其中小的索引结构分别用来索引被道路网络中的结点或弧段分割开的轨迹数据,多个索引结构之间通过网络拓扑关系有效地结合起来,因此被称为基于拓扑的混合索引结构。轨迹数据与道路网络间的紧密关系能够方便索引基于网络的查询的空间条件,多个简单索引结构间的不同组合也能有效地支持时间查询条件。一个完整的 TMIS 结构如图 1 所示,弧段 (Arcs)、结点 (Nodes)、转向表 (Turn table) 间通过字段关联包括来自弧段 (FromArc)、进入弧段 (ToArc) 及经过结点 (ViaNod)。4 个指向弧段和结点的索引结构是静止的 (由于弧段和结点信息通常是不变的),其余包括 NodeB-trees、ArcR-trees 和 RealB-trees 则是动态的,可能随新位置点的插入而增加。其中 NodeB-tree 用于索引经过某结点的分段轨迹,具有单一的索引值——进入弧段的时间 *EnterTime*; RealB-tree 用于索引当前进入某弧段但尚未离开的分段轨迹,具有

单一的搜索值——进入弧段的时间 *EnterTime*; ArcR-tree 用于索引进入某弧段并且已经离开的分段轨迹,具有双搜索值——进入弧段的时间 *EnterTime* 以及离开弧段的时间 *LeaveTime*。

从图 1 中可以看到, TMIS 结构是一些基于网络拓扑的简单索引结构的组合。尽管其中小的索引结构数量庞大,但其总数是固定的,而且每个单独的索引结构都是简单结构并相对较小,能够直接应用现有的插入与查询算法,有利于 TIMS 结构的应用与维护。

最近提出的 GSTree^[21]与 TMIS 结构非常相似,也是通过多个小索引结构的组合完成轨迹数据查询,包括用于索引静态网络的 strip trees 以及动态时间的 interval trees,然而遗憾的是, GSTree 目前只能用于回答基于矩形窗口的查询,对基于拓扑连通关系的查询问题没有给出解决方案。

尽管如上文所述,已经出现了一些用于车辆轨迹数据查询的索引结构,但缺乏与之对应的数据查询语言,目前国内外文献中很少发现相关的研究成果,因此,本文后半部分将以 TMIS 结构为基础对用于车辆轨迹数据的查询语言进行初步探讨。

3 车辆轨迹数据查询语言

3.1 查询语言定义

从查询目的考虑,基于车辆轨迹数据的查询可以分为两大类,面向车辆的查询和面向网络的查询。前者通常关注给定的某辆车于某时刻在道路网络上的位置或其他特性,此类查询一般可以通过现有的索引结构和 SQL (structured query language) 解决;而后者关注来自于无明确标识的一个或多个车辆轨迹的聚合信息,如某个时间段,某路段上所有有轨迹记录的车辆的平均旅行速度。面向网络的查询是本文需要讨论的,因为由弧段连通关系构成的网络路径经常作为其空间索引条件 (如本文第一部分所举之范例),所以又称之为面向连通关系的轨迹数据查询。借鉴通用的 SQL 语言,定义面向连通关系的轨迹数据查询语言 (connectivity-oriented data query language, CQL) 如下:

```
SELECT fields or their equations
FROM trajectory tables
WHERE TIME [time_instant1, time_instant2] AND/OR
NET_ROUTE [arc1, node1, arc2, node2, ..., arcn] AND/
```

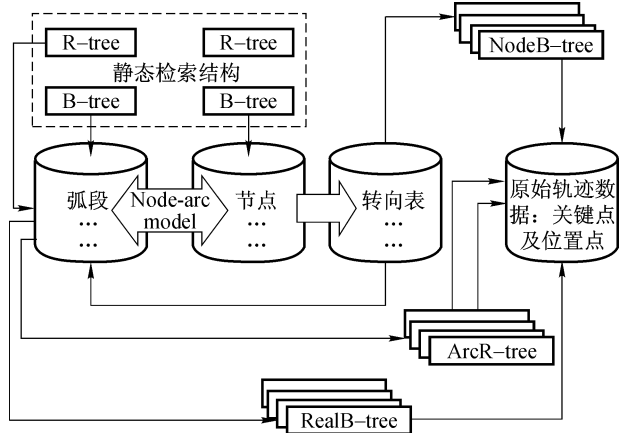


图 1 TMIS 结构的总体构成^[1]

Fig. 1 An overall architecture of the TMIS^[1]

OR other standard SQL query conditions

CQL 定义不同于传统 SQL 语言,在 WHERE 子句中增加了两个新的条件描述方式, TIME [$time_instant_1$, $time_instant_2$] 以及 NET_ROUTE [arc_1 , $node_1$, arc_2 , $node_2$, \dots , arc_n], 其中前者描述了一个时间段(如果 $time_instant_1$ 不同于 $time_instant_2$) 或者一个时间点(如果 $time_instant_1$ 于 $time_instant_2$ 相同); 后者描述了一个由一系列弧段和结点构成的网络路径, 此处凸显了 CQL 面向连通关系的特性。通过执行以下两步查询来解决一个如定义所示的 CQL 查询:

(1) SELECT * into temp_table FROM trajectory tables WHERE other standard SQL query conditions

(2) SELECT fields or their equations FROM temp_table WHERE TIME [$time_instant_1$, $time_instant_2$] AND/OR NET_ROUTE [arc_1 , arc_2 , \dots , arc_n]

其中第 1 步为标准的 SQL 语言, 而第 2 步就需要利用 TMIS 结构分类解决, 见下文描述。

3.2 查询语言分类

因为 TMIS 由众多小的索引结构构成, 而解决不同的查询语句需要根据他们的查询条件组合不同的小索引结构, 为此根据 CQL 定义中的 TIME 条件把查询语句分成以下 3 类, 其中 NOW 代表当前时刻:

(1) 对过去的查询 (Q_{past}), 时间条件可以代表一个时段或一个时刻, 即 $time_instant_1 < time_instant_2 \leq NOW$ 或者 $time_instant_1 = time_instant_2 < NOW$, 如计算过去两小时内从结点 n_1 出发经弧段 a_1 , a_2 到达结点 n_2 的车辆平均旅行时间、昨晚 9 点在弧段 a_1 上的车辆数等。

(2) 对现在的查询 (Q_{now}), 时间条件只可以是一个时刻, 即 $time_instant_1 = time_instant_2 = NOW$, 如计算现在行驶在弧段 a_1 上的车辆数、查找现在距离某加油站 100 m 之内的车量数等

(3) 对未来的查询 (Q_{future}), 时间条件可以代表一个时段或一个时刻, 即 $NOW \leq time_instant_1 < time_instant_2$ 或者 $NOW < time_instant_1 = time_instant_2$, 如预计接下来两个小时内会进入弧段 a_1 的车辆数、预测接下来两个小时内会从结点 n_1 行驶至结点 n_2 的车辆数等。

3.3 实现 CQL 查询

回答一个用 CQL 描述的查询语句的过程如下: 首先, 找到 WHERE 子句; 从该子句中发现关键字

TIME 及 NET_ROUTE; 提取关键字后面的查询条件; 根据 $time_instant_1$ 及 $time_instant_2$ 判断查询类型, 进而调用相应的索引结构和方法获得结果。根据不同的索引类型, 最后一步的实现方法描述如下, 为便于阅读, 其中给出的范例并未用 CQL 描述。

3.3.1 对过去的查询 (Q_{past})

由于 Q_{past} 与历史轨迹数据有关, 因而涉及动态索引结构 ArcR-tree 和 NodeB-tree。如查询 t_1 至 t_2 间通过弧段 a_1 的车辆数量, 可以应用以下两步骤实现:

(1) 找到与弧段 a_1 对应的 ArcR-tree。

(2) 对此 ArcR-tree 应用 R-tree 搜索算法, 查找 $EnterTime \geq t_1$ 且 $LeaveTime \leq t_2$ 的区域内的记录数目, 返回结果。

再如查找 t_1 至 t_2 间通过路径 ($a_1, n_1, a_2, n_2, \dots$) 的所有车辆, 它的空间查询条件为包含一系列弧段与结点的路径, 其实现过程如下:

(1) 取路径中的第一个弧段 a_1 , 查找与该弧段对应的 ArcR-tree。

(2) 对此 ArcR-tree 应用 R-tree 搜索算法, 查找 $EnterTime \geq t_1$ 且 $LeaveTime \leq t_2$ 的区域内的记录, 并将它们保存到一个集合中, 命名为 $Vset$ 。

(3) 取路径中下两个元素, 结点 n_1 与弧段 a_2 。从 Turn table 中查找 ViaNode 为 n_1 、FromArc 为 a_1 、ToArc 为 a_2 的记录, 查找与该记录对应的 NodeB-tree。

(4) 对查找出的 NodeB-tree 使用 B-tree 搜索算法以选择 t_1 至 t_2 间的记录, 并把结果保存到一个集合中, 命名为 $Cset$ 。

(5) 比较 $Vset$ 与 $Cset$ 。保留共同属于两个集合的记录, 将其余记录从 $Vset$ 中删除。

(6) 取路径中的下一个弧段, 及对应的 ArcR-tree, 重复步骤 (2) ~ 步骤 (5), 直到路径中没有剩余弧段。

(7) $Vset$ 中的记录即为最终结果。

上述步骤中, 首先用一个集合存储所有可能的目标, 然后逐步将不可能的目标从集合中删除。

3.3.2 对现在和未来的查询 (Q_{now} 及 Q_{future})

Q_{future} 及 Q_{now} 涉及同样的动态索引结构 RealB-tree。 Q_{now} 关注于一个特殊的时刻——现在。在多数情况下, 它的空间查询条件仅包括一个弧段, 可以简单地找到其相对应的 RealB-tree, 然后, 应用 B-tree

的标准搜索算法获得查询结果。在某些情况下, Q_{now} 的空间查询条件关系到一些结点。如查找现在处于结点 n_1 的车辆数。对此类查询,可以将其转化为数个 Q_{past} 或 Q_{future} 查询,如查找刚刚(或即将)从结点 n_1 通过的各条弧上的车辆数。而其实,在现实世界中这种查询是很少发生的,因为同时满足短时间(“现在”)和小范围空间(“节点”)的查询结果往往会频繁地变化,相应的查询缺少意义。

Q_{future} 的查询结果是对未来的估计,因此需要对某些变量进行假设,如通行时间、行驶速度、路径选择概率等。如查找接下去的1小时内,进入某网络路径的车辆数,其解决思路如下:

(1) 利用递归的方法找出所有可能在1小时内进入该网络路径起点的弧段,此处就需要假设各个弧段的行驶时间。

(2) 找到与这些弧段对应的 RealB-tree, 获知其记录总数。

(3) 根据路径选择概率,计算最终进入该路径并最终沿该路径行驶的比例,与记录数相乘即为所求结果。

4 原型系统设计

如图2所示,该系统应由5部分构成,其中数据库部分包含车辆轨迹数据和道路网络数据以及一些基本的空间或非空间数据索引结构;TMIS扩展模块负责建立和管理用于轨迹数据的小索引结构,并根据3.2节中定义的不同查询类型和参数,按照3.3节提出的索引算法组合小索引结构完成查询需求;查询语句编译器读取如3.1节中定义的CQL语句,并按照3.2节中的分类,把分类结果和参数传递给TMIS扩展模块;输入界面用于输入CQL语句并传

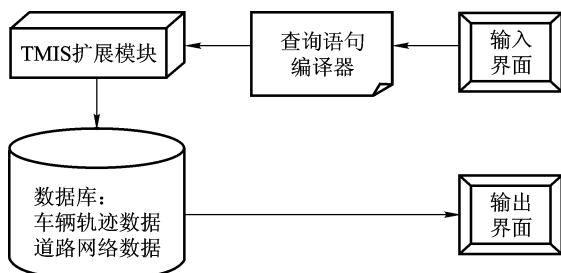


图2 车辆轨迹数据查询原型系统结构

Fig. 2 The structure of a prototype for vehicle trajectory data query

递给语句编译器,可以通过命令行界面或通过交互方式输入语句;结果输出界面显示来自于数据库的查询结果,多数情况下,查询结果为一个具有聚合特征的单一数值,如某个网络路径上的平均行驶时间。

目前,已经收集到了行驶于上海市区的1000辆出租车1周的完整轨迹数据,数据采集频率为10秒每次,数据存储于SQL Server数据库,容量超过12GB。同时也获得了上海路网的中心线数据,一并存储于上述数据库中。TMIS扩展模块通过ArcGIS Engine和高级语言程序编程实现,而该模块与数据库的连接通过ArcSDE实现。查询语句编译器的实现是该系统的难点,目前尚在实验当中,仍需进一步的研究努力。

5 结论

本文比较了目前用于车辆轨迹数据的索引结构,并对用于车辆轨迹数据的查询语言进行了初步的探讨,提出了面向连通关系的查询语言(CQL)的定义、分类及实现方法。CQL的定义是现有SQL的一个延伸,新增了TIME及NET_ROUTE关键词,尤其是NET_ROUTE所指示的网络路径查询条件使CQL区别于其他查询语言。根据TIME条件的不同,CQL查询可分类成 Q_{past} , Q_{now} 及 Q_{future} 。针对不同类型的查询,需要调用或组合不同的动态索引结构, Q_{past} 需要ArcR-tree和NodeB-tree,而 Q_{now} 及 Q_{future} 需要RealB-tree。而在现实世界中,往往涉及更复杂的查询,如查找车辆从a点到b点所采用的不同路径,并比较上周三早晨8点到10点间在每条路径上的平均行驶时间,此类查询是多个基本查询类型的组合,需要多个CQL语句描述。目前的编译器尚仅能工作于简单或标准查询类型的CQL语句,而且CQL的定义结构还相对简单,限制了其描述能力,今后的工作中,将继续完善CQL的语言定义、编译器的扩展及时空查询条件的输入方式,并根据在第4节中的设计思路完成一个用于车辆轨迹数据查询的原型系统。

参考文献 (References)

- 1 Li X, Lin H. Indexing network-constrained trajectories for connectivity-based queries [J]. International Journal of Geographical Information Science, 2006, 20(3): 303-328.
- 2 Guttman A. R-trees: A dynamic index structure for spatial searching [A]. In: Proceedings of the ACM SIGMOD Conference on

- Management of Data [C], Boston, MA, USA, 1984: 47-57.
- 3 Beckmann N, Kriegel H P, Schneider R, *et al.* The R*-tree: An efficient and robust access method for points and rectangles [J]. ACM SIGMOD Record, 1990, **19**(2): 322-331.
 - 4 Nievergelt J, Hinteberger H, Sevcik K D. The grid file: An adaptable, symmetric multi-key file structure [J]. ACM Transactions on Database Systems, 1984, **9**(1): 38-71.
 - 5 Robinson J T. The K-D-B-tree: A search structure for large multidimensional dynamic indexes [A]. In: Proceedings of the SIGMOD International Conference on Management of Data [C], Ann Arbor, MI, USA, 1981: 10-18.
 - 6 Xu X, Han J, Lu W. R-tree: An improved R-tree indexing structure for temporal spatial databases [A]. In: Proceedings of the International Symposium on Spatial Data Handling (SDH) [C], Zurich, Switzerland, 1990: 1040-1049.
 - 7 Theodoridis Y, Vazirgiannis M, Sellis T. Spatio-temporal indexing for large multimedia applications [A]. In: Proceedings of the 3rd IEEE Conference on Multimedia Computing and Systems [C], Hiroshima, Japan, 1996: 441-448.
 - 8 Nanopoulos A, Theodoridis Y, Manolopoulos Y. Indexed-based density biased sampling for clustering applications [J]. Data and Knowledge Engineering, 2006, **57**(1): 37-63.
 - 9 Tao Y, Papadias D. MV3R-tree: A spatiotemporal access method for timestamp and interval queries [A]. In: Proceedings of the 27th International Conference on Very Large Databases [C], Roma, Italy, 2001: 431-440.
 - 10 Nascimento M A, Silva J R O, Theodoridis Y. Evaluation for access structures for discretely moving points [A]. In: Proceedings of the International Workshop on Spatio-Temporal Database Management (STDBM'99)[C], Edinburgh, Scotland, 1999: 171-188.
 - 11 Abdelguerfi M, Givaudan J, Shaw K, *et al.* The 2-3 TR-tree, a trajectory-oriented index structure for fully evolving valid-time spatio-temporal datasets [A]. In: Proceedings of the ACM Workshop on Advances in Geographic Information System, ACM GIS [C], McLean, VA, USA, 2002: 29-34.
 - 12 Pfoser D, Jensen C S. Indexing of network constrained moving objects [A]. In: Proceedings of the 11th ACM International Symposium on Advances in Geographic Information Systems [C], New Orleans, Louisiana, USA, 2003: 25-32.
 - 13 Pfoser D, Jensen C S, Theodoridis Y. Novel approaches to the indexing of moving object trajectories [A]. In: Proceedings of the 26th International Conference on Very Large Databases [C], Cairo, Egypt, 2000: 395-406.
 - 14 CAI Y, Ng R. Indexing spatio-temporal trajectories with chebyshev polynomials [A]. In: Proceedings of the ACM SIGMOD International Conference on Management of Data [C], Paris, France, 2004: 599-610.
 - 15 Chakka V P, Everspaugh A, Patel J M. Indexing large trajectory data sets with SETI [A]. In: Proceedings of the Conf. on Innovative Data Systems Research, CIDR [C], Asilomar, CA, USA, 2003: 164-175.
 - 16 Song Z, Roussopoulos N. SEB-tree: An approach to index continuously moving objects [A]. In: Proceedings of the IEEE International Conference on Mobile Data Management, MDM [C], Melbourne, Australia, 2003: 340-344.
 - 17 Frenzos R. Indexing moving objects on fixed networks [A]. In: Proceedings of the 8th Intl. Symp. on Spatial and Temporal Databases (SSTD)[C], Santorini Island, Greece, 2003: 289-305.
 - 18 De Almeida V, Güting R. Indexing the trajectories of moving objects in networks [J]. GeoInformatica, 2005, **9**(1): 33-60.
 - 19 Kim K, Lopez M, Leutenegger S, *et al.* A network-based indexing method for trajectories of moving objects [J]. Lecture Notes in Computer Science, 2006, **4243**: 344-353.
 - 20 Chen J, Meng X, Guo Y, *et al.* Update-efficient indexing of moving objects in road networks [A]. In: Proceedings of the Third Workshop on Spatio-Temporal Database Management in Conjunction with VLDB 06 (VLDB-STDBM2006)[C], Seoul, Korea, 2006
 - 21 LE T, Nickerson B. Efficient search of path-constrained moving objects [R]. Technical Report TR08-191, Fredericton, Canada: University of New Brunswick 2008.