

Hilbert 曲线的 2 维时空索引

李萍萍, 郝忠孝

(哈尔滨理工大学 计算机科学与技术学院, 黑龙江 哈尔滨 150080)

摘 要:提出了一种针对固定网络中移动对象的时空索引 2DSTH 及其相应的时空查询算法. 2DSTH 采用降低时空数据维度的思想, 使用 Hilbert 曲线将移动对象的 3 维运动空间 (x, y, t) 转换为 2 维的运动空间 (x, t) , 将一个 3 维的时空索引转换为两个 2 维的子索引. 这种 2 维时空索引机制降低了索引的数据维度, 支持当前轨迹和历史轨迹的时空查询操作, 并且有效的优化了窗口查询操作.

关键词:移动对象; 固定网络; 索引结构; 数据降维

中图分类号: TP311.3 **文献标志码:** A **文章编号:** 1007-2683(2009)05-0043-04

Two-dimensional Spatio-temporal Index based on Hilbert Curve

LI Ping-ping, HAO Zhong-xiao

(School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China)

Abstract: A new spatiotemporal index for moving objects on fixed network, named two-dimensional spatiotemporal index for moving objects in network by using Hilbert (2DSTH), and a set of spatiotemporal query algorithms are proposed in this paper. 2DSTH based on the idea of dimensionality reduction, converting the three-dimensional (x, y, t) space to the two-dimensional (x, t) space by Hilbert curve, the single 3D index are replaced by two 2D sub index. The 2DSTH reduces the dimension of spatiotemporal index, supports the efficient query of the past and presents positions of moving objects, moreover it could optimize operations of windows query.

Key words: moving objects; fixed networks; index structures; dimensionality reduction

0 引 言

随着移动定位和无线通讯技术的迅速发展, 车载导航、位置服务、智能交通等应用的大量兴起, 如何有效的索引和查询移动对象的位置信息逐渐成为时空数据库领域的热点问题. 当前, 针对移动对象的索引方法大致有两类: 一类管理移动对象的当前和未来位置信息 (PMR-Quadtree, TPR-Tree, TPR*-Tree, STR IPE 索引结构), 另一类管理移动对象的

历史位置信息 (Quad-Trees, MV3R-Tree, TB-Tree). 上述索引方法都是假设移动对象在任意的 2 维空间中作自由运动, 实际上, 移动对象通常是在一个受约束的 2 维地理空间运动, 例如现实生活中, 大部分移动对象只能在相应的固定网络中运动, 如汽车在道路中行驶, 火车在铁路中运行, 船舶在航线上航行等.

针对大部分移动对象只能运动在固定网络中的特性, 本文提出一种有效管理固定网络移动对象的历史位置以及当前位置信息的 2 维时空索引结构

收稿日期: 2008-05-09

基金项目: 黑龙江省自然科学基金资助项目 (F2006-01)

作者简介: 李萍萍 (1982-), 女, 硕士研究生, E-mail: lppbsh@163.com;

郝忠孝 (1940-), 男, 教授, 博士生导师.

2DSTH (Two-dimensional Spatio-Temporal Index based on Hilbert Curve), 该索引是在 PFOSER 等人^[11]提出的受限运动轨迹索引 TLMC (Trajectory Indexing Using Movement Constraints) 的基础上进行的扩展.

1 2DSTH索引结构

2DSTH索引结构的基本思想是,利用固定网络的空间约束条件降低移动对象的数据维

度,使用空间填充曲线 Hilbert^[2-3]把 3 维的运动空间映射为 2 维空间,使得 3 维的时空索引

转换为两个 2 维的子索引,一个用来索引固定网络,一个用来索引移动对象的位置信息.

1.1 移动对象的轨迹数据

描述移动对象的运动情况时,人们往往忽略移动对象固定不变的大小和形状,只是关心其具体的位置信息,因此,移动对象的轨迹数据是指数据库中存储的移动对象在不同时刻的位置信息.传统的时空数据库中移动对象的运动空间和轨迹如图 1 所示,移动对象的运动空间是由一个 2 维地理空间 (x, y) 和一个 1 维的时间 (t) 组成的 3 维空间 (x, y, t) ,运动轨迹是由 3 维空间 (x, y, t) 中一系列点集合而成的折线.

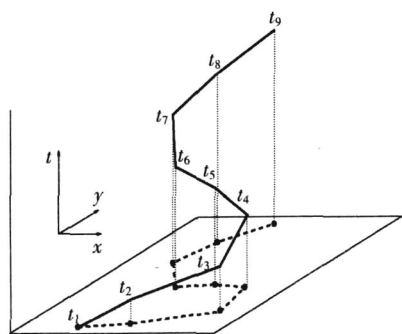


图1 移动对象的运动空间和轨迹

3 维的运动空间 (x, y, t) 直接将时间作为地理信息空间新增加的 1 维,虽然能够直观明了的描述移动对象随时间变化的位置信息,但是忽略了实际应用的特殊性,使得时空查询效率低下.实际应用中,移动对象的运动模式可以分为 3 种:无限制运动、限制运动、固定网络中的运动,其中固定网络中的运动最为普遍,也是本文研究的运动模式.固定网络是由相互连接的道路组成的,可以看作是被嵌入 2 维欧几里德空间的一个 1.5 维空间.

为更好的查询移动对象和固定网络间的位置关

系,同时改进时空索引的查询效率,本文考虑利用固定网络的空间约束条件来降低轨迹数据的数据维度,将 3 维的运动空间 (x, y, t) 转换为 2 维的运动空间 (x, t) ,3 维的轨迹数据也相应的转换为 2 维的轨迹数据.

1.2 数据降维

实现 2 维的时空索引,首先要将固定网络映射到 1 维空间,然后将 3 维的运动轨迹转换为 2 维,最后需要转换相应的时空查询.具体的转换过程如下:

1) 空间映射,本文使用具有良好空间聚集能力的 Hilbert 曲线将 2 维地理空间中的固定网络映射为 1 维空间.其映射过程为:确定固定网络中的路径所在的边,计算并存储每条边中点的 Hilbert 值,计算每条边的长度 $length$,计算并存储每条边在 1 维空间中的坐标 (low, up) ,固定网络相应的被映射为 1 维空间 (x) 中的线段序列 (low_i, up_i) $(i = 1, 2, 3 \dots)$,3 维的运动空间 (x, y, t) 也相应的被映射为 2 维空间 (x, t) .

2) 轨迹转换,完成空间映射后,需要将移动对象的运动轨迹从原始的 3 维的运动空间 (x, y, t) 中转换到相应的映射空间 (x, t) 中,其转换过程为:查找运动轨迹在 2 维网络中经过的路径 P_i ,确定经过的边 H_i 及每条边发生运动的线段 L_i ,由空间映射可得出相应 1 维空间中的线段序列 (low_i, up_i) ,加入时间维后,原始的运动轨迹最终被转换为 2 维的运动空间中的线段序列 (low_i, t, up_i, t) .

3) 查询转换,给定一个时空查询窗口 $w = (x_1, x_2, y_1, y_2, t_1, t_2)$,查找在时间间隔 $t = (t_1, t_2)$ 内,与查询窗口相交的移动对象的运动片段,执行该窗口查询,首先要将时空窗口 w 在固定网络中的空间范围映射到 1 维空间,然后配合时空窗口 w 的时间间隔 (t_1, t_2) 在映射空间中进行查询.

1.3 时空索引 2DSTH 的数据结构

2DSTH 时空索引由静态和动态两部分组成,静态部分使用顶部 2DR-Tree R_{net} 管理固定网络, R_{net} 的每个叶子结点都对应底部的 2 维网格索引 (Grid File); 动态部分由管理移动对象的 Hash 结构 H_{data} 和底部网格索引 G_{object} 组成, H_{data} 用于记录移动对象当前的位置信息, G_{object} 存储经过对应 R_{net} 叶子节点道路的移动对象的具体位置信息以及存在的时间段,同时记录对象的连续运动轨迹信息.

R_{net} 使用普通的空间索引技术索引当前移动对象所在的道路,它的叶子结点根据空间位置包含固定网络中的多条道路,在降低结点数同时减少了

索引结构的死角和重叠区域. 索引树 R_{net} 的叶子结点结构为 $(mbr, polyD, hD, gPT)$, 其中: mbr 是固定网络中路径的最小包围矩形; $polyD$ 是路径标记; hD 是路径所包含边的 Hilbert 值; gPT 是指向与该路径对应的底部 2 维网格索引的指针. 非叶子结点结构为 $(mbr, childPT)$, mbr 是包含所有子结点数据项的最小包围矩形, $childPT$ 是指向子结点的指针.

H_{data} 记录当前移动对象的位置、速度和方向信息, 其记录形式为 (moD, hD, x, v, d) , 其中: moD 是移动对象标记; hD 是当前所在边的 Hilbert 值; x 是移动对象在 1 维映射空间中的坐标值; v 是移动对象的速度; d 是移动对象的运动方向. C_{object} 用于索引移动对象的历史轨迹数据, 记录的结构为 $(moD, gridX, gridY, low, up, t_s, t_e)$, 其中: moD 为移动对象标记; $gridX, gridY$ 为所在网格编号; low, up 为位置间隔; t_s, t_e 为时间间隔. 2DSTH 时空索引具体的数据结构如图 2 所示.

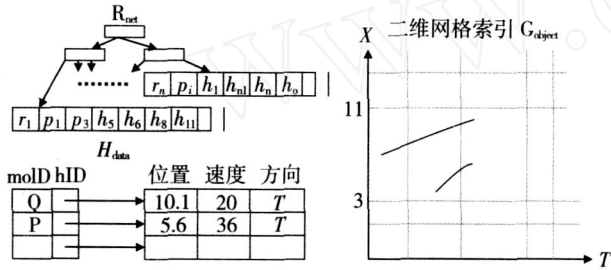


图 2 2DSTH 时空索引的数据结构

2DSTH 时空索引的更新算法如下:

1) 当移动对象进入固定网络时, 通过位置信息在 R_{net} 中找到相应的叶子结点, 然后向 H_{data} 中添加该对象的一条记录并记下相应的位置、速度、方向信息.

2) 当移动对象改变道路时, 重新在 R_{net} 中查找相应的叶子结点, 然后更新 H_{data} 中移动对象的位置、速度、方向信息, 向 2 维网格 C_{object} 中插入移动对象的轨迹折线.

3) 当移动对象改变速度、方向 (调头) 时, 更新该移动对象在 H_{data} 中的位置、速度、方向信息, 向 2 维网格 C_{object} 中插入移动对象的轨迹线段.

4) 当移动对象离开固定网络时, 在 H_{data} 中删除该移动对象的记录.

2 时空查询操作

2.1 查询当前轨迹数据

2DSTH 时空索引支持当前路段、道路的汽车

(假设移动对象为汽车) 查询和当前汽车信息查询. 当前路段、道路汽车查询, 用于查询当前 $polyD = p_i$ 的路段或道路上有哪些汽车在行驶, 其查询算法为: 在 R_{net} 中查找 $polyD = p_i$ 路径所对应的 Hilbert 值 h_i ($i = 1, 2, 3 \dots$), 然后在 hash 表 H_{data} 中查询 $Hid = h_i$ 的记录, 根据移动对象标记获得在此路段或道路上行驶的汽车; 当前汽车查询, 用于查询当前汽车 $moD = m_i$ 的位置、速度和方向信息, 当汽车在道路中匀速行驶时, 由匀速运动开始时的位置近似代替当前位置, 其查询算法为: 在 2DSTH 的 H_{data} 中查询 $moD = m_i$ 的记录, 获得相应汽车的位置、速度和运动方向.

2.2 查询历史轨迹数据

给定一个时空窗口 $w = (x_1, x_2, y_1, y_2, t_1, t_2)$, 查找在时间间隔 (t_1, t_2) 出现在空间范围 $r = (x_1, x_2, y_1, y_2)$ 中的所有汽车, 该查询的查询算法如下: ① 在顶部 R_{net} 中查找与空间范围 r 相交的叶子结点项, 求出这些叶子结点中的路径 p_i ($i = 1, 2, 3 \dots$) 与空间范围 r 相交的路段 p_{ij} ($i, j = 1, 2, 3 \dots$). ② 根据相交路段的 Hilbert 值, 在 C_{object} 中查找相应的 x 坐标值 (low_i, up_i) ($i = 1, 2 \dots$), 求出 x 坐标的最大和最小范围 (low_{min}, up_{max}). ③ 配合时空窗口的时间间隔 (t_1, t_2) , 构造窗口 $w_1 = (low_{min}, up_{max}, t_1, t_2)$, 求出所有和窗口 w_1 有交集的移动对象集合 $moSet$. ④ $moSet$ 中移动对象是否和相交, 若相交则添加到查询结果集合中.

3 2DSTH 性能分析

设计一组实验分析 2DSTH 索引结构的索引性能, 实验采用 Inter Pentium M 1.86G 处理器、512M 内存的硬件环境, 操作系统为 Windows XP Professional, 编程工具为 Visual C++ 6.0. 采用文 [4] 提出的基于固定网络的移动目标生成器生成移动对象的运动轨迹, 固定网络数据采用哈尔滨市 1:1000 比例尺的道路数据.

在实验中, 比较移动对象的同一运动轨迹在 3 维原始空间和 2 维映射空间中的数据量, 由表 1 可以看出, 移动对象的运动轨迹在转换后的 2 维映射空间中占用的存储空间较小.

时空窗口查询执行时间受到空间窗口的大小、时间间隔大小和移动对象数量的影响. 首先, 在三个变量中固定其中的一个使之成为最大值, 采用其它两个变量的不同组合进行时空查询, 执行时间如表 2

所示,由表 2 知时空窗口查询的消耗时间主要受空间窗口大小影响,受时间间隔与移动对象数量的影响不大,然后比较 2DSTH 与 3 维时空索引的时空窗口查询访问 I/O 的次数,由图 3 可以看出,2DSTH 的时空窗口查询访问 I/O 的次数少于 3 维的时空索引,且随着时间间隔的增长,2DSTH 的查询效率明显提高。

表 1 存储空间比较

	3D 空间	2D 空间
移动对象数量	500	500
记录个数	125 000	125 000
存储空间	3.35 MB	2.5 MB

表 2 时空窗口查询的执行时间

移动对象	窗口大小(%) / 时间间隔(%)					
	5	10	25	50	80	100
50	109/3 843	297/3 827	907/3 795	1687/3 756	3391/3 780	3812/3 812
100	110/3 795	250/3 811	719/3 811	1516/3 842	3109/3 858	3828/3 828
150	141/3 905	296/3 827	875/3 811	1813/3 812	3292/3 780	3859/3 859
200	140/3 853	234/3 790	844/3 853	1735/3 884	3563/3 852	3884/3 884

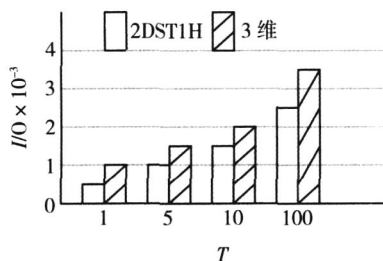


图 3 窗口查询比较

4 结 语

本文提出了一种新的固定网络中移动对象的时空索引 2DSTH。2DSTH 时空索引有一个 2DR-Tree、一个 Hash 表和一个 2 维网格索引组成。2DR-Tree 用于索引固定网络,Hash 表用于记录当前移动对象的位置、速度和方向信息,2 维网格索引用于索引移动对象的历史轨迹数据。2DSTH 时空索引支持大量的当前轨迹和历史轨迹的时空查询操作。通过性能分析得知,2DSTH 时空索引占用的存储空间较小、时空查询操作效率高等优点。下一步工作将在 2DSTH 索引结构中考考虑预测移动对象的未来位置并且对未来位置信息查询进行优化,如何针对实际应用设计高性能的管理固定网络的 2DR-Tree 也将成为后续研究的重点。

参 考 文 献:

- [1] POFSER D, JENSEN C S. Trajectory Indexing Using Movement Constraints[J]. Geo Informatic, 2005, 9(2): 93 - 115.
- [2] 陆 锋,周成虎. 一种基于空间层次分解的 Hilbert 码生成算法. 中国图象图形学报, 2001, 6(5): 465 - 469.
- [3] 周 艳,朱 庆,张叶廷. 给予曲线层次分解的空间数据划分方法[J]. 地理与地理信息科学, 2007, 23(4): 13 - 17.
- [4] BRNKHOFF T. A Framework for Generating Network - base Moving Objects[J]. Geo Informatica, 2002, 6(2): 153 - 180.

(编辑:温泽宇)