



大规模轨迹数据的 Geohash 编码组织 及高效范围查询

向隆刚¹ 王德浩¹ 龚健雅¹

1 武汉大学测绘遥感信息工程国家重点实验室,湖北 武汉,430079

摘要:面向成熟的关系-对象型空间数据库,利用 Geohash 编码的唯一性、一维性和递归性等特征,提出了一种基于 Geohash 编码的大规模轨迹数据组织方法及范围查询技术。该方法结合 Geohash 编码和 B+ 树索引,设计了适应不同尺度范围查询的大规模轨迹数据的关系组织模式,并给出了相应的两阶段查询处理算法,同时提出了一种 Z 合并优化,以进一步提高范围查询的处理效率。实验结果表明,此方法适合于组织管理与查询分析大规模的轨迹数据,其范围查询性能高于内置的 R 树索引。

关键词:轨迹数据;Geohash 编码;范围查询;Z 合并优化

中图法分类号:P208

文献标志码:A

近年来,人们愈加方便地获取位置信息、请求基于位置的服务(location-based service, LBS)^[1],并将人或物体的移动过程以轨迹的形式记录下来。轨迹数据本身的价值催生了基于轨迹的位置服务,如线路分享、车辆调度和安保等^[2]。同时,对轨迹数据进行挖掘分析可以获得有关个人行为模式、人与人之间的相关性以及人在不同地域之间的活动模式等知识,从而为用户提供更加智能、有效、人性化的位置服务^[2-3]。由地理数据、轨迹数据和应用记录等构成的位置大数据已经成为分析地理国情和构建智慧城市的重要战略资源^[4]。

越来越多的应用要求管理移动对象产生的大规模轨迹数据并提供查询分析能力,其中,范围查询的快速处理是需要迫切解决的关键问题之一,常见的解决之道是设计和构造合适的索引结构,以此来大幅减少范围查询的搜索空间。目前,针对轨迹数据索引技术的研究主要集中于二维空间索引结构的变形与扩展方面^[5-14]。文献[5-14]中提出的方法虽能面向各自的查询优化问题,建立相应的有效索引结构来辅助查询处理,但是,这种针对二维空间索引进行的变形或扩展使得其组织与管理方法变得异常复杂,难以集成到空间数据库管理系统(如 Oracle^[16]、PostgreSQL^[17]等)之中。

目前,空间数据库主要以内置的 B⁺ 树或 R 树等简单成熟的索引结构来辅助查询处理。本文

面向关系-对象型空间数据库,提出了一种基于 Geohash^[18]的大规模轨迹数据的高效组织结构及快速范围查询方法。该方法的核心思想是建立不同编码长度的 Geohash 索引表,在多种空间尺度下用覆盖轨迹的网格来近似表达轨迹,从而将针对二维空间中轨迹数据的范围查询降维处理为针对相应 Geohash 索引表的一维查询。此外,本文分析讨论了查询范围初始编码长度的选择问题,并根据 Geohash 编码的递归特征,提出了始于初始编码长度的自底向上 Z 合并优化,以进一步提高范围查询的处理效率。

1 Geohash 技术概述

1.1 Geohash 划分与编码

Geohash 是一种地理编码格式,它沿着经度和纬度的方向交替二分地球表面,并使用一个二进制数(Geohash 编码)表示所形成的互不重叠网格。对全球进行两次划分后形成 4 个编码长度为 2 的 Geohash 编码(00、01、10、11),对应 4 个网格。以编码为 11 的网格为例,经度范围为 $[0, 180^\circ]$,纬度范围为 $[0, 90^\circ]$,若继续对此网格做两次划分,将形成 4 个编码长度为 4 的 Geohash 编码(1100、1101、1110 和 1111)。

Geohash 编码具有几个特点:(1)唯一性^[19]。

每个单元网格都有全球唯一的编码与之对应。(2)一维性。Geohash 用一维数字或字符串表示二维空间区域。(3)递归性。Geohash 下级单元网格由上级单元网格划分而得,其本质上是一种空间 Z 曲线填充,一个“Z”字型内的 4 个 Geohash 网格属于同一个父网格,它们的二进制编码具有相同的前缀。Geohash 现已被应用在兴趣点快速搜索^[20]、面数据区域查询^[21]等方面。

1.2 轨迹的 Geohash

编码确定轨迹的 Geohash 编码需要在给定

编码长度下,计算轨迹穿过了哪些 Geohash 网格。以图 1 为例具体说明轨迹的 Geohash 编码。图 1(a)展示了示例轨迹在编码长度为 17 时穿过的 9 个网格及其编码,而图 1(b)则展示了同一示例轨迹在编码长度增加为 19 时穿过的 22 个网格及其编码。由此可见,一条轨迹可能对应一个或多个 Geohash 编码。同时,可能有多条轨迹穿过同一个 Geohash 网格,因此,轨迹与网格之间是多对多的关系。



图 1 轨迹 Geohash 编码示例

Fig. 1 A Demonstration of Trajectory Geohash Encoding

2 基于 Geohash 的轨迹数据组织与范围查询

针对大规模轨迹数据,本文面向关系-对象型数据库,设计了基于 Geohash 编码的关系组织模式及其范围查询框架(见图 2)。其主要思想是对轨迹进行 Geohash 编码组织,利用一维的 B⁺ 树来处理二维的范围查询。为了兼顾大小迥异的范围请求,本文共设计了 5 种不同码长的 Geohash 索引表,在 5 种空间尺度下表达轨迹与 Geohash 网格之间的多对多关系。范围查询使用经典的两阶段处理策略,索引筛选阶段直接使用标准 SQL 语言查询相应的 Geohash 索引表,以获得候选轨迹集,据此再通过遍历求精来获得最终查询结果。

2.1 关系组织模式

除了轨迹数据表之外,本文还在关系组织模式中引入了 Geohash 索引表,并在其 Geohash 编码字段上建立 B⁺ 树索引。由于 Geohash 是一种固定大小的空间划分编码技术,通常由 R 树支持的范围请求便可转化为针对 Geohash 索引表及 B⁺ 树索引的查询处理。考虑到用户提出的查询范围可大可小,本文建立了码长分别为 19、21、23、25 和 27 的 5 种 Geohash 索引表,以将不同大

小的范围查询导向合适码长的 Geohash 索引表。此外,这种设计有利于范围查询的并行化处理,而且这种并行既可以针对多个范围查询,也可以用于单个查询处理。

2.2 两阶段范围查询处理

范围查询使用经典两阶段处理策略,首先索引筛选,通过 Geohash 索引表筛选出候选集,然后遍历求精,通过检验轨迹的几何信息从候选集中求出最终结果。由于遍历求精阶段并无特殊之处,本节着重讨论索引筛选阶段。

首先计算覆盖了查询范围 q ,且初始码长为 P_0 的 Geohash 网格编码集 C_0 ;然后从码长 P_0 开始,以自底向上方式 Z 合并编码集 C_0 ,得到对应于码长序列(设为 $\langle P_0, \dots, P_l \rangle, P_i = P_0 - 2 \times i, 0 \leq i \leq l \leq 4$)的多个 Geohash 编码集(设为 D_0, \dots, D_l ,分别对应 P_0, \dots, P_l);接着为每一种码长(设为 P_i)生成针对相应 Geohash 索引表的 SQL 语句 S_i ,其 WHERE 条件则由相应 Geohash 编码集(即 D_i)构造而来;最后将不同的 SQL 语句(即 S_0, \dots, S_l)发往不同的 Geohash 索引表,以并行方式执行这些 SQL 语句。需要指出的是, P_0 设置和 Z 合并优化是提高索引筛选性能的关键。

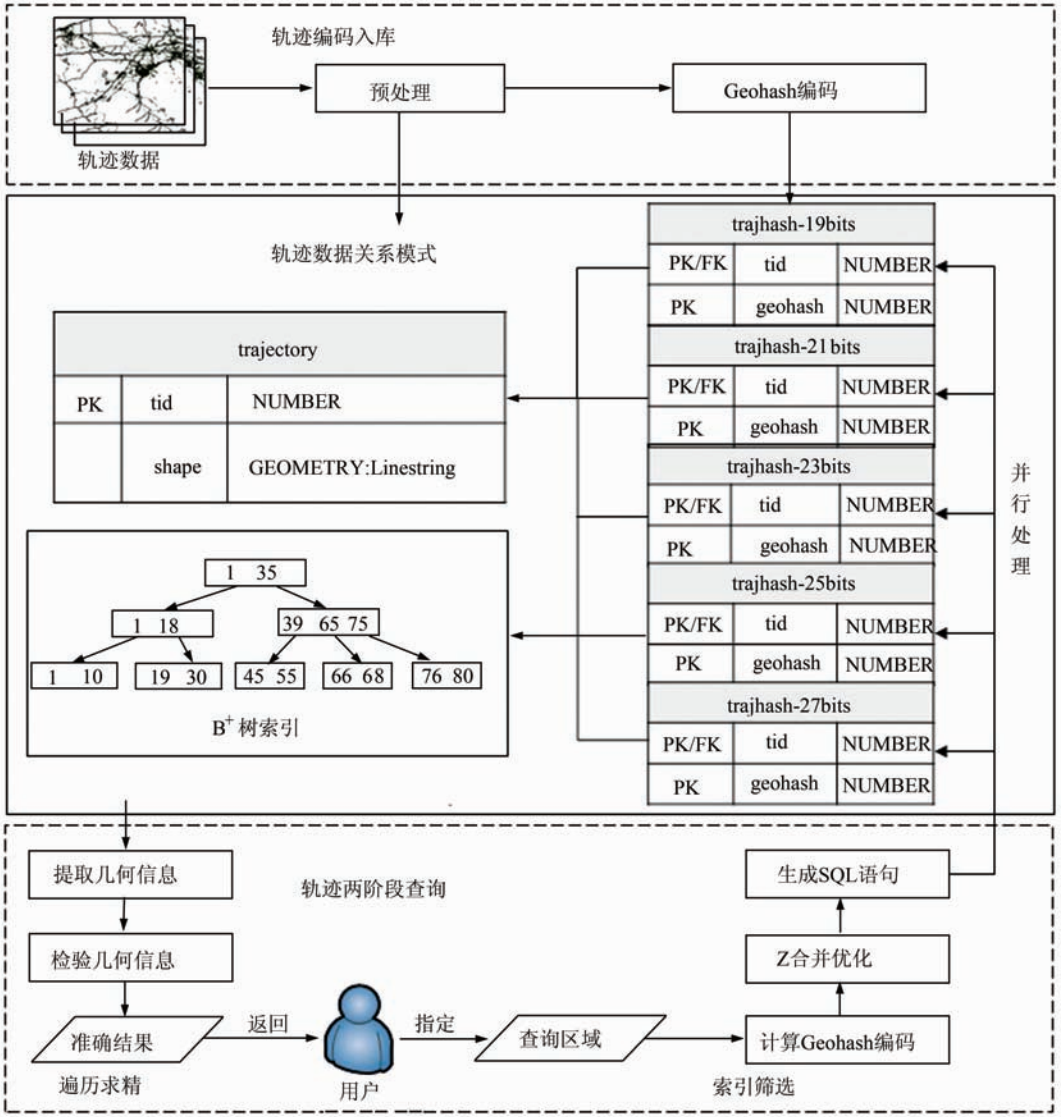


图 2 基于 Geohash 的轨迹数据关系模式及其范围查询框架

Fig. 2 Geohash-Based Relational Scheme of Trajectory Data Andits Processing Framework of Range Query

3 范围查询优化技术

3.1 初始码长选择

当码长为 p 时,范围查询 q 的代价模型可以表示为:

$$C(q, p) = C_f(q, p) + C_r(q, p) \quad (1)$$

其中, $C_f(q, p)$ 表示索引筛选代价; $C_r(q, p)$ 表示遍历求精代价。若保持 q 不变,增加码长, Geohash 编码数目将增多,相应 Geohash 索引表的规模也会增大,从而导致 $C_f(q, p)$ 增大。由此可知, p 越长,索引筛选代价越高,反之亦然,即:

$$p \uparrow C_f(q, p) \uparrow, p \downarrow C_f(q, p) \downarrow \quad (2)$$

同时,随着码长 p 的减小,一般情况下覆盖查询范围的 Geohash 网格面积 S_g 会随之减小。图 3 举例说明了 S_g 与 p 的关系,深色矩形表示查

询范围,其面积 S_q 约为 222.7 km^2 ,红框矩形为覆盖查询范围的 Geohash 网格集。图 3(a) 显示 p 为 25 时, S_g 约为 480 km^2 ,图 3(b) 显示 p 为 27 时, S_g 约为 288.1 km^2 。较小的 S_g 往往意味着较少的轨迹落于其中,因而 $C_r(q, p)$ 也随之减小。所以, p 越长,遍历求精代价越小,反之亦然,即:

$$p \uparrow C_r(q, p) \downarrow, p \downarrow C_r(q, p) \uparrow \quad (3)$$

综合式(1)和式(2)可知,针对某一范围查询窗口,当初始码长 P_0 变长时,索引筛选代价增高,遍历求精代价降低。因此,需要找到合适的 P_0 ,使得式(3)最小。然而在实际应用中,受查询窗口大小、数据分布密度、不同数据库查询效率差异等因素的影响,精确计算出 P_0 是一件非常困难的事情。实验观察发现,当查询窗口的面积小于 $1^\circ \times 1^\circ$ 时, Geohash 编码数目总体较少,索引筛选时间很短,遍历求精占据范围查询的大部分时间,此时

应选择较长的 P_0 (27 位), 以有效降低 $C_r(q, P_0)$ 。随着查询窗口面积逐渐增大, 索引筛选时间对查询效率逐渐形成了较大影响。当面积超过 $1^\circ \times 1^\circ$ 时, 应选择 P_0 为 25, 而当面积超过 $3^\circ \times 3^\circ$ 时, P_0 为 23 较为合适。需要指出的是, 在初步选定 P_0 后, 还应进一步判断以其父网格 (即码长为 $P_0 - 2$) 覆盖 q 时是否满足以下两种情况之一: ① $P_0 - 2$ 生成的网格面积和 P_0 相同。② $P_0 - 2$ 的网格面积所覆盖的轨迹和 P_0 相同, 或者只是稍多一些。若成立, 则应以 $P_0 - 2$ 作为初始码长, 否则以 P_0 作为初始码长。

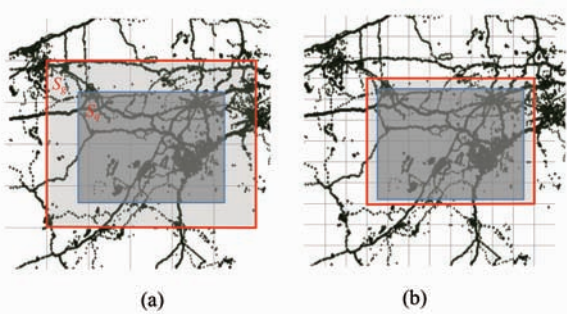


图3 不同码长网格覆盖同一查询范围面积对比

Fig. 3 A Comparison of Covering Areas with Different Geohash Code Precisions

3.2 Z 合并优化

索引筛选阶段, 以初始码长 P_0 计算覆盖查询范围的 Geohash 编码集合可能包含数量巨大的 Geohash 编码, 如果直接将其表达为 SQL 语句的 WHERE 谓词, 将在解析和过滤两方面给查询处理引擎带来较大负担。考虑到 Geohash 编码是一种 Z 曲线填充, 一个“Z”范围内的 4 个编码具有相同的前缀, Z 合并优化利用这一递归特性, 用一个更短的编码替代同一“Z”字内 4 个编码, 以减少编码总数。

Z 合并优化过程首先将编码集 D_i 拆分为两个编码集 $D_{z,i}$ 与 $D_{nc,i}$, 其中, $D_{z,i}$ 包含能够合并的编码, $D_{nc,i}$ 包含不能合并的编码。接着将 $D_{nc,i}$ 作为码长为 P_i 的编码集 (即 D_i) 加入 list。然后 Z 字合并 $D_{z,i}$ 中的编码, 形成码长为 P_{i+1} ($P_{i+1} = P_i - 2$) 的编码集, 并以该编码集作为 D_{i+1} 循环执行上述过程, 直到没有可以合并的编码为止。最终, 算法返回包含多种码长编码集的集合 $\text{list}\{D_0, \dots, D_i\}$ 。

图 4 展示了 Z 合并优化的效果, 图 4(a) 中阴影部分代表的查询范围由 48 个相同长度的 Geohash 编码覆盖。图 4(b) 显示经过 Z 合并优化处理后, 形成了 3 种长度的编码集 $\{D_0, D_1, D_2\}$ 。其

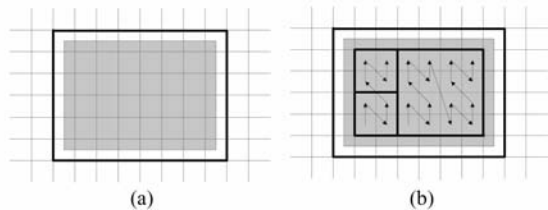


图4 Geohash 编码 Z 合并优化示意

Fig. 4 Diagram of Geohash Code Z-Merge Optimizing Method

中, D_0 包含 24 个编码, D_1 包含 2 个编码, D_2 包含 1 个编码。可见, Z 合并优化既可以减少 Geohash 编码数目以简化查询条件, 又能生成多种码长支持并行查询处理。

4 实验与分析

为验证本文方法的性能, 本文使用 C++ 基于 Oracle 11g 数据库实现了基于 Geohash 大规模轨迹数据组织结构及其范围查询方法。实验数据为开放街道地图 (OpenStreetMap, OSM) 上美国范围用户自愿上传的 31 880 条 GPS 轨迹, 约 11.1 GB。实验环境为 Windows 7 操作系统, 2.93 GHz Intel Core i3 CPU, 4 GB 内存, 931 GB 7200 RPM 硬盘驱动器。

首先建立轨迹数据表, 导入实验数据, 并为轨迹数据添加 R 树索引, 作为对比实验。然后对数据分别建立 19 位、21 位、23 位、25 位、27 位和 29 位 6 种码长的整型 Geohash 索引表。实验中查询范围的选择方法是在美国范围内随机选取点, 以之作为东北角分别生成 $0.01^\circ \times 0.01^\circ$ 、 $0.1^\circ \times 0.1^\circ$ 、 $1^\circ \times 1^\circ$ 、 $2^\circ \times 2^\circ$ 、 $3^\circ \times 3^\circ$ 、 $4^\circ \times 4^\circ$ 、 $5^\circ \times 5^\circ$ 、 $10^\circ \times 10^\circ$ 的矩形。查询时间统计是对每种范围做 500 次查询实验, 计算其平均时间。

1) 实验 1: 初始码长选择实验。针对不同大小范围查询, 分别以 21、23、25、27、29 作为初始码长, 总查询时间对比结果见表 2。在 $1^\circ \times 1^\circ$ 及以下查询范围内, 27 位和 29 位作为初始码长查询效果更好, 但 29 位较 27 位的性能提升不明显, 因此在实际应用中只需建立 27 位的 Geohash 索引表。范围在 $1^\circ \times 1^\circ$ 到 $3^\circ \times 3^\circ$ 之间的查询, 25 位初始码长的查询效果最好。当范围超过 $3^\circ \times 3^\circ$ 时, 23 位初始码长的查询效果最好。以上即是本文推荐的初始码长选择策略, 后文实验均遵照此策略选择初始码长。同时实验 1 也说明本文建立的 5 种 Geohash 索引表可以满足大小迥异的范围查询请求。

表 1 不同查询范围内初始码长选择

Tab. 1 Selection of Initial Code Precision for Range Queries of Various Sizes

查询范围	初始码长				
	21/ms	23/ms	25/ms	27/ms	29/ms
0.01°×0.01°	750.52	569.52	246.7	145.1	132.8
0.1°×0.1	1 098.9	874.1	472.9	315.2	298.7
1°×1°	5 026.3	4 451.9	3 969.1	3 548.5	4 876.9
2°×2°	10 639.7	9 012.7	7 958.9	9 587.7	11 198.3
3°×3°	15 972.1	13 574.4	10 585.7	14 402.4	34 782.7
4°×4°	18 857.8	15 612.3	20 234.1	27 979.3	52 863.1
5°×5°	26 729.4	19 611.6	34 689.4	40 074.1	78 896.3
10°×10°	49 393.5	39 979.3	67 763.6	10 1023.2	20 1486.9

2) 实验 2:Z 合并优化性能测试实验。对比使用和不用 Z 合并优化对索引筛选时间的影响,实验结果见图 5。

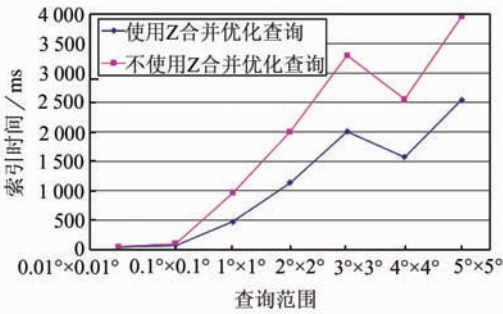


图 5 使用与不使用 Z 合并优化索引筛选时间比较
Fig. 5 Comparison of Indexing Time of Using and not Using Z-Merge Optimization

范围查询的面积较小时,Geohash 编码总量较少,Z 合并优化没有大幅减少索引筛选时间。随着查询范围面积增大,Z 合并优化的优势变得明显,原因在于 Geohash 编码数目随着查询面积的增大快速增加,而 Z 合并优化能有效减少 SQL 语句中查询条件的数目。值得注意的是,在 4°×4°范围处,两条曲线均出现了时间下降的现象,这

是查询范围超过 3°×3°,选择 23 位码长引起的效率提升。实验 2 表明,Z 合并优化能显著地减少索引筛选时间。

3) 实验 3:范围查询效率对比实验。比较本文方法与传统的基于 R 树索引查询方法(简称为 R 树法)的范围查询效率,实验结果见图 6。其中图 6(a)给出了索引筛选阶段对比结果,本文方法的索引筛选时间在 0.01°×0.01°至 5°×5°范围内小于 R 树法,而在 10°×10°范围时超过了 R 树法。这是因为随着查询范围的增大,覆盖查询范围的编码数目增长速度非常快,导致检索 Geohash 索引表的 SQL 语句变得非常庞大复杂,其处理代价随之急剧升高。图 6(b)给出了过滤求精阶段对比结果,可以看到基于 Geohash 的范围查询方法遍历求精时间总是小于 R 树法,说明其索引筛选的准确率更高。图 6(c)给出了查询总时间对比图,显然无论针对何种查询范围,基于 Geohash 的范围查询方法的查询效率总高于 R 树法。但是受到索引筛选时间的影响,基于 Geohash 的范围查询方法在查询范围较小时的效率优势更加明显。

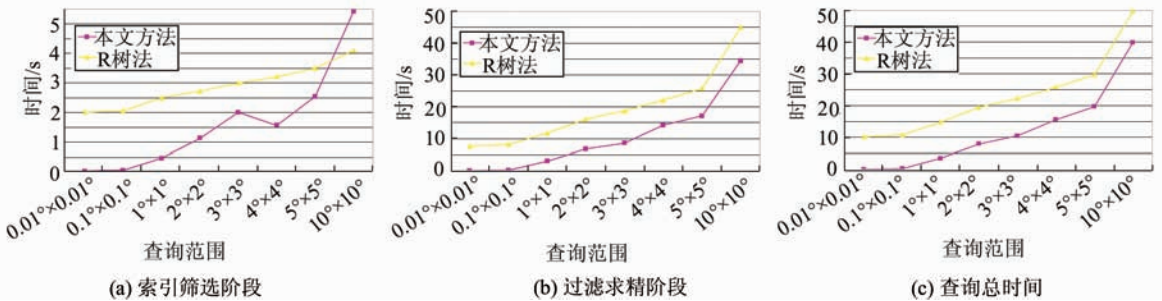


图 6 两种方法查询时间对比

Fig. 6 Comparison of Range Query Performance Using Two Different Indexing Methods

5 结 语

本文提出了大规模轨迹数据的 Geohash 编

码组织方法,采用球面剖分数据模型,可以针对小到城镇、大到国家,甚至全球范围的轨迹数据,进行统一建模。同已有方法相比,本文结合 Geohash 编码和 B+ 树索引,直接在通用数据库系统

及标准 SQL 语句之上实现了针对大规模轨迹数据的高效范围查询,实验结果表明其显著优于传统的 R 树索引方法。显然,此方法同样适合于处理近邻查询。此外,在数据更新时只需在关系表中增加或删除记录即可,更新代价很小。下一步将研究如何在本文提出的框架之上实现更为复杂的轨迹数据查询。

参 考 文 献

- [1] Zhou Aoying, Yang Bin, Jin Cheqing, et al. Location-Based Service: Architecture and Progress[J]. *Chinese Journal of Computer*, 2011, 34(7):1 155-1 171(周傲英,杨彬,金澈清,等. 基于位置的服务:架构与进展[J]. 计算学报, 2011, 34(7):1 155-1 171)
- [2] Zheng Yu, Xie Xin. User Trajectory Mining Based Intelligent Location Service[J]. *China Computer Federation Letter*, 2010, 6(6):23-30(郑宇,谢幸. 基于用户轨迹挖掘的智能位置服务[J]. 中国计算机学会通讯, 2010, 6(6):23-30)
- [3] Li Ting, Pei Tao, Yuan Yecheng. A Review on the Classification, Patterns and Applied Research of Human Mobility Trajectory[J]. *Progress in Geography*, 2014, 33(7):938-948(李婷,裴韬,袁焯城,等. 人类活动轨迹的分类、模式和应用研究综述[J]. 地理科学进展, 2014, 33(7):938-948)
- [4] Liu Jingnan, Fang Yuan, Guo Chi, et al. Research Progress in Location Big Data Analysis and Processing[J]. *Geomatics and Information Science of Wuhan University*, 2014, 39(4):379-385(刘经南,方媛,郭迟,等. 位置大数据的分析处理研究进展[J]. 武汉大学学报·信息科学版, 2014, 39(4):379-385)
- [5] Theodoridis Y, Vazirgiannis M, Sellis T. Spatio-Temporal Indexing for Large Multimedia Application[C]. The IEEE Conference on Multimedia Computing and System, Hiroshima, 1996
- [6] Guttman A. R-trees: A Dynamic Index Structure for Spatial Searching[C]. The ACM Intl. Conf. on Management of Data, Boston, USA, 1984
- [7] Pfoser D, Jensen C S, Theodoridis Y. Novel Approaches to the Indexing of Moving Object Trajectories[C]. The 26th Intl. Conf. on Very Large Data Bases, Cairo, Egypt, 2000
- [8] Yin Zhangcai, Li Lin, Wang Zheng. Spatio-Temporal Index Based on Extended HR-tree[J]. *Geomatics and Information Science of Wuhan University*, 2007, 32(12):1 132-1 134(尹章才,李霖,王峥. 基于 HR 树扩展的时空索引机制研究[J]. 武汉大学学报·信息科学版, 2007, 32(12):1 132-1 134)
- [9] Xia Y, Prabhaker S. Q+Rtree: Efficient Indexing for Moving Object Databases[C]. The Intl. Conf. on Database System for Advanced Application, Kyoto, Japan, 2003
- [10] Finkel R A, Bentley J L. Quadrees: A Data Structure for Retrieval on Composite key[J]. *Acta Information*, 1974, 4(1):1-9
- [11] Chakka V P, Everspaugh A, Patel J M. Indexing Large Trajectory Data Sets With SETI[C]. Innovative Data Systems Research, Asilomar, CA, USA, 2003
- [12] Almeida V T, Guting R H. Indexing the Trajectories of Moving Objects in Networks[C]. The 16th Intl. Conf. on Scientific and Statistical Database Management, Tyoko, Japan, 2004
- [13] Jin Peiquan, Zhang Xu, Yue Lihua. NBR-tree: A Novel Spatio-Temporal Index for Urban Traffic Networks[J]. *Geomatics and Information Science of Wuhan University*, 2010, 35(2):147-151(金培全,张旭,岳丽华,等. NBR-tree:面向城市交通网络的一种新型时空索引[J]. 武汉大学学报·信息科学版, 2010, 35(2):147-151)
- [14] Cudre-Mauroux P, Wu E, Madden S. TrajStore: An Adaptive Storage System for Very Large Trajectory Data Sets[C]. The 26th International Conference on Data Engineering, Long Beach, CA, USA, 2010
- [15] Jinfeng N, China V R. Indexing Spatial-Temporal Trajectories with Efficient Polynomial Approximation[J]. *IEEE Tran on Knowledge and Data Engineering*, 2007, 19(5): 663-678
- [16] Oracle. Oracle and Its Affiliates [EB/OL]. http://docs.oracle.com/cd/E11882_01,2014
- [17] The PostgreSQL Global Development Group. PostgreSQL 9. 4. 1 Documentation [EB/OL]. <http://www.postgresql.org/docs/9.4/interactive,2014>
- [18] Gustavo N. Geohash [EB/OL]. <http://en.wikipedia.org/wiki/Geohash,2014>
- [19] Balkic Z, Šoštarić D, Horvat G. GeoHash and UUID Identifier for Multi-agent Systems[C]. The 6th KES International Conference on Agent and Multi-agent Systems: Technologies and Applications, Dubrovnik, Croatia, 2012
- [20] Ye Xiaorong, Shao Qing. Mobile Advertising System Based on Augmented Reality and Location-Based Services [J]. *Science and Technology Review*, 2013, 31(4):67-73(叶小榕,邵晴. 基于增强现实和位置服务的手机广告系统[J]. 科技导报, 2013, 31(4):67-73)
- [21] Jin An, Cheng Chengqi, Song Shuhua, et al. Regional Query of Area Data Based on Geohash[J].

Geography and Geo-Information Science, 2013, 29
(5):31-35(金安,程承旗,宋树华,等. 基于 Geohash

的面数据区域查询[J]. 地理与地理信息科学,
2013, 29(5):31-35)

Organization and Efficient Range Query of Large Trajectory Data Based on Geohash

XIANG Longgang¹ WANG Dehao¹ GONG Jianya¹

¹ State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing,
Wuhan University, Wuhan 430079, China

Abstract: Efficient organization and querying of trajectory data is one of the research hotspots in the field of spatial databases. By taking advantage of the properties of globally unique, one-dimensional, and hierarchically recursive coding of Geohash codes oriented to relational spatial databases, we propose a Geohash-based method of organizing large trajectory data and the associated range query processing technologies. First, a trajectory relational schema, combining Geohash coding and a B+ tree index, was designed for range queries at multiple scales. A corresponding two-stage range query processing algorithm is introduced. We developed a Z-merge optimization for further improving the efficiency of range query processing. Experimental results based on Oracle11g verify that our approach was fit for organizing large trajectory data while range query performance was much better than traditional R-tree.

Key words: trajectory; Geohash code; range query; Z-merge optimization

First author: XIANG Longgang, PhD, associate professor, specializes in trajectory processing and analyzing. E-mail: geoxlg@whu.edu.cn

Foundation support: The National Natural Science Foundation of China, Nos. 41471374, 41001296.