DOI:10.13203/j.whugis20160523 文章编号:1671-8860(2019)03-0436-07

Geohash-Trees:一种用于组织大规模轨迹的自适应索引

向隆刚[1] 高 萌[1] 王德浩[1] 龚健雅[2]

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
2 武汉大学遥感信息工程学院,湖北 武汉,430079

摘 要:蕴含着挖掘价值的轨迹数据分布在世界各地,且规模庞大。如何在全球范围内组织轨迹数据并支持高效范围查询成为难题。一种自适应索引组织框架被提出来管理查询全球范围大规模轨迹数据集,其基本思想为:针对不同轨迹数据集,根据 Geohash 编码,生成层数最深的 Geohash 格网覆盖住整个轨迹数据集范围;以格网作为根节点,生成 Geohash-Trees;为了加快查询定位到对应索引,根据编码前缀相同的特点设计了字典查询树。Geohash-Trees 是一种基于格网划分的空间索引,它能够根据轨迹密度自适应使用多种剖分策略划分空间,提高范围查询效率。为了支持索引动态更新,设计了增量插入和更新算法。同时,该索引被移植到商用数据库 Oracle 中,利用数据库性能高效管理查询轨迹数据。实验结果表明,该方法在范围查询以及占用空间等方面明显优于 Oracle 内置的 R 树索引。

关键词:轨迹数据;Geohash 编码;自适应性;空间索引;空间分异性

中图分类号:P208 **文献标志码**:A

随着 GPS 技术的精进和完善以及移动通信技术的迅猛发展,越来越多的移动设备中都携带 GPS 芯片组。技术的进步和硬件的普及为基于位置的服务(location-based service, LBS)[1]打下了良好的基础。移动定位应用记录的大量轨迹数据有着非比寻常的挖掘价值。轨迹数据记录的是人或物体移动过程,通过轨迹数据挖掘分析能够得出一系列有价值的结论。例如,对人轨迹进行分析能得出有关个人的行为模式[2]等。这些结论能够有效应用于复杂的社会应用,如智能交通决策[3]和智慧城市建设[4]。

一方面,轨迹数据集包含的轨迹可能分布在不同的地理范围。世界上有很多网站可以让用户上传任意轨迹,例如 Wikiloc (https://www.wikiloc.com/)。轨迹的分布可以来自全世界。另一方面,轨迹数据集在空间分布上是稀疏且不均匀的,轨迹密度和人口分布与活动等息息相关,即存在天然的空间分异性[5]。如何对大规模、多尺度、不均匀分布的轨迹数据进行组织管理以及高效范围查询成为亟待解决的难题,对现有的数据库索引技术提出了严峻的挑战。

近年来,轨迹数据索引的研究可以分为数据划分和空间划分两类。数据划分一般采用二维空间索引 R 树的变种和扩展。例如,文献[6]提出的 3D R-Tree 针对移动对象有时序性的特点,将 R 树扩展成高维索引;文献[7]对 3D R-Tree 的索引对象以及查询处理算法进行了改进,提高了查询效率;文献[8]设计了 STR-Tree 和 TB-Tree,前者改写了 R 树的插入和分裂算法,使其更适用于轨迹数据存储,后者在插入新轨迹时要求叶节点存储的轨迹片段属于同一轨迹;文献[9]提出了 HBSTR 索引,改变时空 R 树插入策略,以节点插入。这类方法大多时空查询能力良好,但是索引结构复杂,更新维护效率较差。空间划分方式一般利用 B^+ 树索引、格网或者四叉树[10]去划分空间,将轨迹分割成轨迹片段,每个空间管理划分到此空间的轨迹片段。例如,文献[11]提出的 B^+-Tree 利用空间填充曲线将移动对象线性化,再利用 B^+ 树对线性元组进行索引;文献[12]提出的 SETI 将空间格网化,每个格网利用 R 树管理轨

收稿日期:2018-01-15
项目资助:国家自然科学基金(41471374,41001296)。
第一作者:向隆刚,博士,教授,主要研究方向为轨迹数据处理与分析。geoxlg@whu.edu.cn
通讯作者:高萌,硕士。gmshepard@whu.edu.cn

迹数据;文献[13]提出利用不同码长的 Geohash[14]编码格网近似表达轨迹,将二维轨迹查询降维成一维的 Geohash 编码查询,提高效率;文献[15]提出了一种混合索引,一方面根据数据空间分布建立索引,另一方面根据历史查询数据聚类建立查询模型,二者通过 Geohash 编码格网进行关联;文献[16]提出的 CSE-Tree 利用四叉树去划分空间,每个分区根据结束和开始时间戳分别建立 B^+ 树索引;文献[17]采用一种动态的四叉树索引去划分空间,能根据轨迹的密集程度自适应地分割空间。这类方法优点在于构建索引、查询效率高,缺点在于需要将轨迹分为片段来管理,集成到现有的空间数据库中很困难。另外,如果用户想得到感兴趣完整的轨迹信息,还需将轨迹片段合成一段完整的轨迹,加大了查询难度。

基于现有研究状况,本文面向对象-关系数据库,提出了一种利用 Geohash 编码技术的框架来组织管理多尺度、大规模、大范围的轨迹数据集。该框架利用 Geohash 编码生成的格网覆盖不同的轨迹数据集,以这些格网作为根节点,生成 Geohash-Trees。该索引考虑到轨迹数据的空间分异性,设计了多种空间划分方式,根据轨迹的密度自适应地划分空间。与四叉树索引不同的地方有两点:(1)自适应空间划分,增加了查询的命中率,提高了查询的效率;(2)轨迹数据不会拆分成轨迹片段,Geohash-Trees 索引的是完整轨迹在轨迹数据库中的 ID,用户可以提取到完整的轨迹信息而不是轨迹片段。

1　Geohash-Trees 自适应索引

1.1　概念和框架

Geohash-Trees 自适应索引采用的是 Geohash 地理编码格式。这种编码方式是将二维经纬度转化为一维字符串,具体原理是沿着经纬度方向交替二分全球表面,利用二进制来表示划分结果,将二进制串用 32 进制编码方式进行编码。

Geohash 具有 3 个特点:(1)全球唯一性。通过 Geohash 划分出来的区域有且只有全球唯一的空间区域与之对应。(2)递归性。同一区域不同层级的编码前缀相同,编码越长,表示的范围越小、越精确。(3)编码的一维性。二维经纬度用一维字符串来表示。降维处理有利于进行检索,提升查询速度。基于以上特点,Geohash 广泛应用于邻近点查询[18]、面数据查询等[19]。

图 1 展示的是索引的框架。框架主要分为存储管理模块、增量更新模块以及查询引擎模块 3 个部分。用户通过查询引擎模块进行轨迹的范围查询。增量更新模块是用来支持索引增量更新功能。存储管理模块主要有字典查找树[20]、Geohash-Trees 以及轨迹数据库。

对导入的每个轨迹数据集都通过 Geohash 编码生成格网来覆盖轨迹数据集所在的兴趣区域,将格网作为根节点,建立 Geohash-Trees。为了快速定位到对应索引进行查询,根据 Geohash 编码前缀相同等特点,建立了字典查找树来管理对应索引根节点入口地址。字典查找树是一种以空间换取时间的树形结构,主要优点是利用字符串公共前缀来降低查询时间开销。字典树作为一种经典的数据结构不在此赘述。

Geohash-Trees 索引的项是完整轨迹在数据库中的 ID。为了解决轨迹稀疏性问题,该索引会随着轨迹的分布自适应划分。单元不仅可能会四分为子单元,也可能横向或纵向二分为子单元。因此,非叶节点的孩子可能为 2 或 4 个。下层空间范围可能是上层空间范围的横(纵)向 1/2 以及 1/4。根节点将覆盖兴趣区域的格网作为键,指向子节点。其他节点根据轨迹分布自适应划分这些格网。将叶子节点层数设为 0,非叶子节点层数为自身到达叶子节点的最长路径。

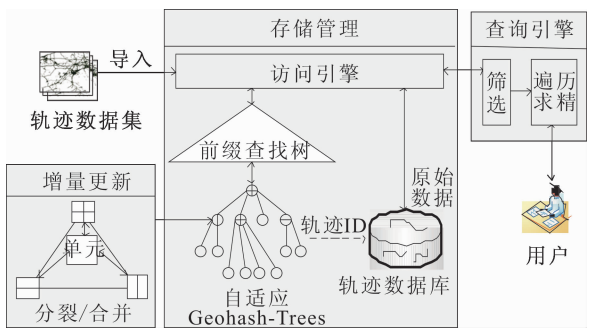


图 1　Geohash-Trees 自适应索引框架

Fig. 1　Framework of Geohash-Trees Index

1.2　单元自适应分裂条件与规则

查询存在筛选以及遍历求精两个过程。二者时间长短和单元分裂有关。如果一个单元分裂,叶子节点数目和树高度都会增加,导致范围查询时间增加。另一方面,搜索空间会减少,筛选出来的候选轨迹集也随之减少。单元分裂越多,筛选代价越高,遍历求精代价越低。因此,设计的单元分裂条件与规则要在筛选时间和遍历求精代价之间找一个平衡点,使总代价最小。

随着轨迹不断被导入,单元中存储的轨迹数目越来越多,当一个单元中轨迹数目达到一个阈

值(P-Threshold)时,单元准备分裂。P-Threshold的大小与文件系统的page大小有关(NTFS系统page大小默认为4 096字节)。同时,单元不能无限分裂下去,无限分裂会增加树的高度,使树的结构变得复杂,搜索树节点变得困难。因此,当单元分裂到设置的最小值时,不会再进行分裂操作。

大型轨迹数据集在空间分布上一般是不均匀的,呈一种偏态分布[21]。大部分轨迹集中于人口密集区域(例如市中心、热门商业区域),而偏远区域(例如郊区、森林)轨迹分布较为稀疏。轨迹分布与人口分布和活动相关,呈现空间分异性。因此,考虑根据轨迹分布的密度来自适性划分空间,能够有效减少搜索空间,提高查询效率。假设某个单元被均匀四分,子单元填充顺序遵循Z字形。一条轨迹可能会与这4个子单元中的一个或者几个相交,也可能完全不相交。根据轨迹和单元相交情况可以得出16种可能,具体情况见图2。这16种相交情况称为轨迹的空间类型,可用4位整数表示:第k位代表第k个子单元是否与轨迹相交,相交为1,不相交为0。例如,如果一条轨迹只与子单元2和子单元4相交,那么该轨迹的空间类型被表示为$(0101)_2=(5)_{10}$,它被简称为type-5。相应地,type-0的轨迹意味着它不与任何子单元相交。

轨迹与单元之间的常见情况如下:如果大部分轨迹属于type-15,则单元并不会分裂。此种情况下分裂,查询空间并不会缩小。然而,如果大部分轨迹属于type-1,分裂有助于缩小查询空间,变为原来的1/4,此种情况就适合分裂。基于以上事实,本文设计了3种分裂规则:

1) 若 $(\alpha_1+\alpha_2+\alpha_4+\alpha_6+\alpha_8+\alpha_9)/\alpha_v \geqslant \Delta_{\sup}$,单元四分。

2) 若 $(\alpha_5+\alpha_{10})/\alpha_v > \Delta_{\sup}$,单元纵向二分。

3) 若 $(\alpha_3+\alpha_{12})/\alpha_v > \Delta_{\sup}$,单元横向二分。

其中,α_k代表对应空间类型的轨迹数目;α_v是除了type-0以外的所有轨迹数目;$\Delta_{\sup}$是单元分裂策略阈值(例如0.5)。

除了传统四分策略,本文还设计了另外两种更好的分裂策略:横向或者纵向二分。考虑到type-5或者type-10轨迹,虽然四分策略能够缩小查询空间,但是纵向二分策略远比四分好。四分策略会比纵向二分策略多引进节点,导致更复杂的结构。

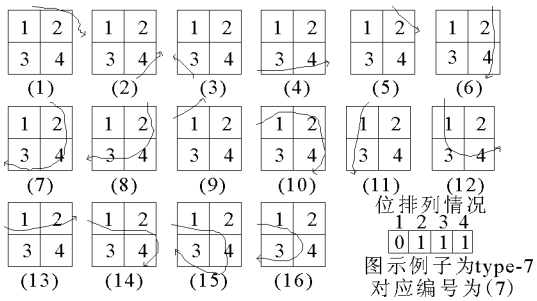


图2 轨迹-单元之间的相交情况以及编码

Fig. 2 Trajectory-Cell Spatial-Type and Their Codes

1.3 索引生成算法

为了保证Geohash-Trees能够进行增量更新,索引生成算法由两部分组成:(1)初始化兴趣区域的Geohash-Trees;(2)轨迹逐个导入,进行索引更新。

树初始化的核心思想是计算合适的Geohash格网作为索引的根节点。具体过程为:(1)根据编码规则去递归搜索至多4块格网(一般来说格网数目为1块、2块或者4块)来覆盖兴趣区域,同时保证这些格网所在层数最深;(2)每个搜索出来的格网还要进行遍历求精,递归检查该格网的子格网是否覆盖兴趣区域,直到不满足条件,返回当前子格网,取代原先搜索出来的格网。因此,根节点中的格网可能不是来自同一Geohash层。根节点将每个返回的格网作为键,指向空的叶子节点。

索引自顶向下生成算法的主要思想是采用自顶向下策略来插入增量。具体步骤如下:

1)如果轨迹信息被接收,将它加入原始轨迹数据库。

2)开始建立Geohash-Trees。如果节点为根节点或者叶子节点,将轨迹ID加入该节点。计算该轨迹和该节点的4个子节点的相交关系,更新节点数据统计,也就是空间类型。如果节点的轨迹数量在阈值以下,则不用分裂。如果超过则分裂。满足分裂条件1,节点单元四分;满足分裂条件2,节点单元纵向二分;满足分裂条件3,节点单元横向二分。分裂的工作主要是创建空节点,例如四分就创建4个。然后将轨迹ID加入对应节点,并计算空间类型。最后检测该节点是否还满足分裂条件,不满足则合并节点。

3)如果节点为中间节点,重复执行第2步,直到遇到叶子节点。

1.4 索引更新算法

索引更新算法主要包括轨迹数据删除和修改。修改可看作先删除旧版本轨迹,再插入新版

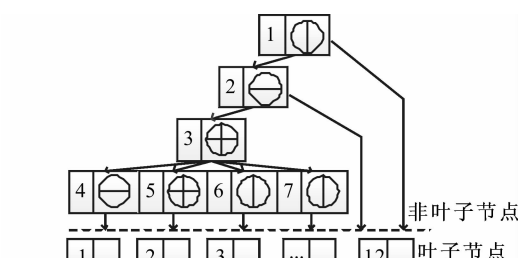
本轨迹,这个过程与 R 树索引修改类似。而对于删除算法而言,类似于插入算法,只用在插入算法的基础上修改以下两点:

1)算法第 2 步中节点加入轨迹 ID,改为删除,且更新节点统计数据,以减少对应空间类型的数量。

2)如果叶子节点为空,或者非叶子节点的子节点被删除,则从树中剔除这些节点。显然,在算法第 2 步可以加上对应步骤。

2　自适应 Geohash-Trees 索引框架的数据库实现

在成熟的空间数据库中,管理轨迹数据并未采用学术界的索引,而是采用 R 树索引。Geohash 编码的一维性易于管理。因此,本文将索引移植到 Oracle 数据库。自适应索引可以用表来表示,每个节点也就是表中的一条记录,另外原始轨迹数据存入另一张表。为了形象说明 Geohash-Trees 在数据库中的情况,将 GeoLife[22-24] 轨迹数据集导入数据库,建立了索引。图 3 是北京市海淀区清华大学附近区域的索引结构,图 4 是该区域自适应索引可视化的情况。图 3 中有 7 个非叶子节点,12 个叶子节点。非叶子节点表中字段 Types 表示该节点如何分裂,字段 Statistics 记录 3 种分裂条件计算出来的分数,字段 Pointers 指向子节点。叶子节点表中字段 TrajIDList 存储的是轨迹 ID 集合。结合图 3、图 4,可以看出非叶子节点是如何进行空间划分的。将此区域的轨迹数据进行了最邻近指数分析。当显著水平 $\alpha = 0.05$ 时,Z 远小于 -1.645,则该区域轨迹数据空间聚集性模式是显著的,存在着空间分异性。



行号	种类	分数	子节点
1			
2			
⋮			
7			

行号	轨迹ID集
1	
2	
⋮	
12	

图 3　Geohash-Trees 扁平化实现

Fig. 3　Flatted Geohash-Trees

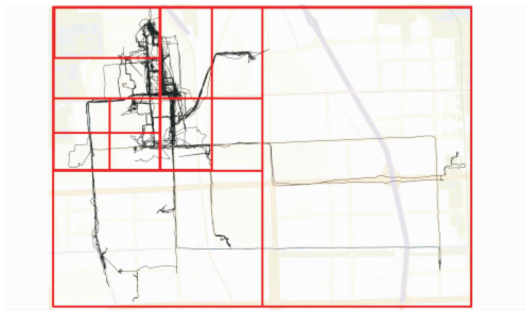


图 4　Geohash-Trees 可视化实例

Fig. 4　Example of Geohash-Trees Visualization

3　Geohash-Trees 自适应索引实验与分析

本文对 Geohash-Trees 自适应索引进行了测试。索引结构从索引插入的时间、索引占用空间大小以及索引查询时间 3 个方面来进行评估和比较。本文选取的比较对象是 Oracle 中内置的 R 树索引。两种索引都是基于 Oracle 实现的,本文在 Oracle 上部署了 Geohash-Trees 自适应索引。

实验软件环境为 Windows 7 64 位操作系统以及 Oracle 12c 数据库,硬件环境为 3.10 GHz Intel Core i3-2100 CPU,8 GB 内存,931 GB 7200RPM 硬盘驱动器。

实验数据来自于 OpenStreetMap 网站,位于德国的 32 766 条 GPS 轨迹,数据大小约 11.0 GB。空间范围 $[\mathrm{lon}_{\min}, \mathrm{lat}_{\min}, \mathrm{lon}_{\max}, \mathrm{lat}_{\max}]$ 为 $[5.72°, 46.55°, 14.35°, 54.58°]$。

3.1　索引插入时间与占用空间对比

目前 Geohash-Trees 自适应索引只有增量建立一种构建方式。整个索引建立时间将近 4 h,而 Oracle 数据库自带 R 树索引建立时间远小于该索引建立时间。在实际情况中,索引只需建立一次即可。如果有新的轨迹数据,则通过增量建立方式来更新维护索引。该索引插入一条轨迹平均时间为 486.25 ms,和 R 树插入时间相差无几。

索引占用空间方面,Geohash-Trees 自适应索引所占空间为 2.1 MB,R 树索引所占空间为 3 MB。这说明该索引更节省空间。

3.2　索引查询时间比较

实验主要围绕空间范围查询设计,选取范围方法是在德国范围随意取一点作为查询矩形的左上角,分别生成 0.01°×0.01°、0.1°×0.1°、1°×1°、2°×2°、3°×3°、4°×4°、5°×5°范围的查询矩形,记为查询范围 1、2、3、4、5、6、7。实验次数是 500 次。查询时间统计分为两部分:(1)索引筛选时

间,(2)索引遍历求精时间。二者记录的是500次查询的平均时间。图5展示的是实验结果。

图5(a)展示的是索引筛选时间对比,Geohash-Trees自适应索引筛选时间远小于R树索引,并且随着查询范围扩大,筛选时间也只是缓慢增加。图5(b)展示的是遍历求精时间对比。由于遍历求精的算法一致,因此遍历求精时间跟候选轨迹数目正相关。由图5(b)可以看出,Geohash-Trees自适应索引候选轨迹数目少于R树候选轨迹数目,说明其筛选准确率高于R树。图5(c)展示的是查询时间总和。无论是小范围查询还是大范围查询,Geohash-Trees自适应索引查询效率高于R树索引。

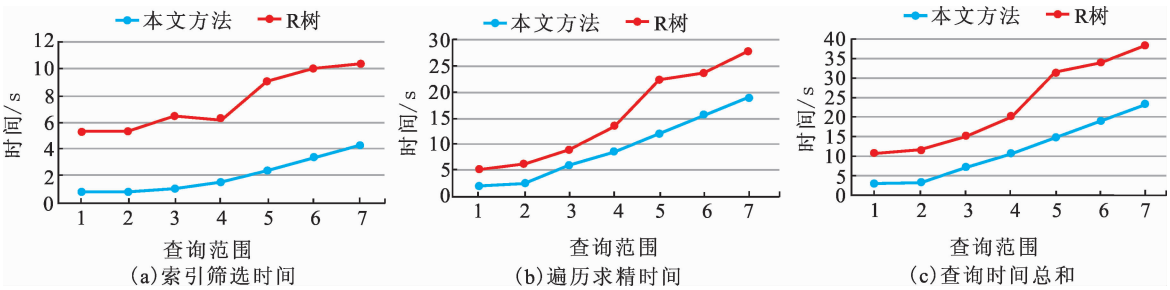


图5 范围查询时间对比

Fig. 5 Comparison of Range Query Performance

4 结 语

本文提出的Geohash-Trees自适应索引框架能够组织管理检索全球范围大规模轨迹数据集,支持空间范围查询以及增量更新。本文方法在Geohash编码格网之上建立了自适应Geohash-Trees,通过多种分裂策略解决轨迹数据在空间分布上的分异性,从而有效减少查询空间,提高空间范围查询效率。为了快速定位到对应索引,本文还充分利用Geohash编码的特点设计了字典查找树。索引实验结果表明,Geohash-Trees自适应索引在范围查询以及占用空间方面表现明显优于R树索引。同时,本文还将该索引移植到商用数据库Oracle中,有利于索引结构的持久化,便于提取轨迹数据信息。未来,要进一步改善Geohash-Trees自适应索引建立方式,支持批量建立,缩短建立时间,使索引更好地适用于实际应用。

参 考 文 献

[1] Tang Keping, Xu Fangheng, Shen Cailiang. Survey on Location-Based Services[J]. *Application Research of Computers*, 2012, 29(12): 4 432-4 436(唐科萍,许方恒,沈才樑. 基于位置服务的研究综述[J]. 计算机应用研究, 2012, 29(12): 4 432-4 436)

[2] Huo Zheng, Meng Xiaofeng. A Survey of Trajectory Privacy-Preserving Techniques[J]. *Chinese Journal of Computers*, 2011, 34(10): 1 820-1 830(霍峥,孟小峰. 轨迹隐私保护技术研究[J]. 计算机学报, 2011, 34(10): 1 820-1 830)

[3] Xie Jiameng, Peng Hong, Zhou Bing, et al. Analysis of Intelligent Traffic Information and Research on Decision Support Based on Data Mining Techniques[J]. *Highway*, 2004(4): 154-158(谢嘉孟,彭宏,周兵,等. 基于数据挖掘技术的智能交通信息分析与决策研究[J]. 公路, 2004(4): 154-158)

[4] Li Deren, Yao Yuan, Shao Zhenfeng, et al. Big Data in Smart City[J]. *Geomatics and Information Science of Wuhan University*, 2014, 39(6): 631-640(李德仁,姚远,邵振峰,等. 智慧城市中的大数据[J]. 武汉大学学报·信息科学版, 2014, 39(6): 631-640)

[5] Wang J F, Zhang T L, Fu B J. A Measure of Spatial Stratified Heterogeneity[J]. *Ecological Indicators*, 2016, 67: 250-256

[6] Theodoridis Y, Vazirgiannis M, Sellis T. Spatio-Temporal Indexing for Large Multimedia Applications[C]. International Conference on Multimedia Computing and Systems, Hiroshima, Japan, 1996

[7] Guo Jing, Liu Guangjun, Guo Lei, et al. A Whole-time Index Design Based on 3D$^+$-TPR-tree for Moving Point Targets[J]. *Acta Geodaetica et Cartographica Sinica*, 2006, 35(3): 267-272(郭晶,刘广军,郭磊,等. 基于3D$^+$-TPR-tree的点目标全时段移动索引设计[J]. 测绘学报, 2006, 35(3): 267-272)

[8] Pfoser D, Jensen C S, Theodoridis Y. Novel Approaches in Query Processing for Moving Object Trajectories[C]. The 26th Intl Conf on Very Large Data Bases, Cario, Egypt, 2000

[9] Ke S, Gong J, Li S, et al. A Hybrid Spatio-Temporal Data Indexing Method for Trajectory Databases[J]. *Sensors*, 2014, 14(7):12 990-13 005

[10] Finkel R A, Bentley J L. Quadtrees: A Data Structure for Retrieval on Composite Key[J]. *Acta Information*, 1974, 4(1):1-9

[11] Jensen C S, Lin D, Ooi B C. Query and Update Efficient B^+-Tree Based Indexing of Moving Objects [C]. The 30th Intl Conf on Very Large Data Bases, Toronto, Canada, 2004

[12] Chakka V P, Everspaugh A, Patel J M. Indexing Large Trajectory Data Sets with SETI[C]. Innovative Data Systems Research, Asilomar, CA, USA, 2003

[13] Xiang Longgang, Wang Dehao, Gong Jianya. Organization and Efficient Range Query of Large Trajectory Data Based on Geohash[J]. *Geomatics and Information Science of Wuhan University*, 2017, 42(1):21-27(向隆刚, 王德浩, 龚健雅. 大规模轨迹数据的 Geohash 编码组织及高效范围查询[J]. 武汉大学学报·信息科学版, 2017, 42(1):21-27)

[14] Fox A D, Eichelberger C N, Hughes J N, et al. Spatio-Temporal Indexing in Non-relational Distributed Databases[C]. International Conference on Big Data, Santa Clara, CA, USA, 2013

[15] Meng Xuechao, Ye Shaozhen. New Spatio-Temporal Index Method Based on Real-Time Data and Query Log Distribution[J]. *Journal of Computer Applications*, 2017, 37(3): 860-865(孟学潮, 叶少珍. 基于实时数据和历史查询分布的时空索引新方法[J]. 计算机应用, 2017, 37(3): 860-865)

[16] Wang Longhao, Zheng Yu, Xie Xing, et al. A Flexible Spatio-Temporal Indexing Scheme for Large-Scale GPS Track Retrieval[C]. Mobile Data Management, Beijing, China, 2008

[17] 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

[18] Liu Qi, Tan Xicheng, Huang Fang, et al. GB-Tree: An Efficient LBS Location Data Indexing Method[C]. The 3rd International Conference on Agro-Geoinformatics, Beijing, China, 2014

[19] 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)

[20] Jankowski R. Advanced Data Structures by Peter Brass Cambridge University Press 2008[J]. *ACM Sigact News*, 2010, 41(1):19-20

[21] Nagin D S, Tremblay R E. Analyzing Developmental Trajectories of Distinct but Related Behaviors: A Group-Based Method[J]. *Psychological Methods*, 2001, 6(1):18-34

[22] Zheng Yu, Zhang Lizhu, Xie Xing. Mining Interesting Locations and Travel Sequences from GPS Trajectories[C]. International World Wide Web Conferences, Madrid, Spain, 2009

[23] Zheng Yu, Li Quannan, Chen Yukun, et al. Understanding Mobility Based on GPS Data[C]. Ubiquitous Computing, Seoul, Korea, 2008

[24] Zheng Yu, Xie Xing, Ma Weiying. GeoLife: A Collaborative Social Networking Service Among User, Location and Trajectory[J]. *Bulletin of the Technical Committee on Data Engineering*, 2010, 33(2):32-39

Geohash-Trees: An Adaptive Index Which can Organize Large-Scale Trajectories

XIANG Longgang[1]　*GAO Meng*[1]　*WANG Dehao*[1]　*GONG Jianya*[2]

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

2　School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China

Abstract: Trajectory data which contains mining valve is widely distributed and large-scale. How to organize trajectory data and retrieve trajectory data efficiently becomes very difficult to solve. We present a framework of adaptive index based on Geohash to organize the worldwide and large-scale trajectory data set. Different trajectory data sets will be covered by Geohash grid which is deepest, and then

we take the grid as root node to generate adaptive Geohash-Trees. In order to quickly locate the corresponding index, we design trie on the basis of the feature of Geohash. Adaptive Geohash-Trees is a spatial index based on grid. It can divide the space according to the track density by adopting a variety of strategies which improves the efficiency of range query. Meanwhile, we design the algorithm of incremental insertion and update for the supporting of real-time update of trajectory data. Furthermore, this framwork has been migrated in Oracle. The experiment results verify that our approach in several aspects such as range query and occupied disk size performs much better than R-Trees.

Key words: trajectory data; Geohash code; adaptability; spatial index; spatial stratified heterogeneity

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

Corresponding author: GAO Meng, master. E-mail: gmshepard@whu.edu.cn

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

(上接第421页)

significance. In order to restore the aircraft's real flight path and to verify the topographical mapping ability of array SAR technology in high-rise buildings, the traditional simulation research based on the assumption of uniform linear motion is abandoned. And the non-ideal trajectory motion error model of MIMO(multiple input multiple output) downward-looking array SAR is analyzed and constructed with high-abrupt urban buildings. The 3D range-Doppler (RD) imaging algorithm of urban buildings MIMO downward-looking array SAR under non-ideal trajectory is proposed. Then the flight path and attitude modeling simulation technique of the aeronautical platform, and a fast and efficient echo simulation technique are used to carry out simulation experiments. The correctness and effectiveness of the imaging algorithm are verified by the imaging results.

Key words: MIMO; downward-looking array SAR; urban buildings; non-ideal trajectory

First author: LIU Hui, PhD, lecturer, specializes in InSAR and array SAR. E-mail: lh860801@163.com

Foundation support: The National Natural Science Foundation of China, Nos. 41071296, 41474010, 61401509, 41371439; Key Scientific Research Project of Henan Higher Education Institutions College and University, No. 19A420008.