基于 HBase 的面向语义单元的室内移动对象索引

张得群1,2，谢传节1,\*，裴韬1

1.中国科学院地理科学与资源研究所 资源与环境信息系统国家重点实验室,北京100101

2.中国科学院大学,北京,100049

Semantic Cell Oriented Indoor Moving Objects Index based on HBase

ZHANG Dequn1,2, XIE Chuanjie1,\*, PEI Tao1

1. *State Key Laboratory of Resources and Environmental Information System, Institute of Geographic Sciences and Natural Resources Research, CAS, Beijing* 100101, *China*

2. University of Chinese Academy of Sciences, Beijing 100049, China

**Abstract**:

With the development of indoor positioning technique, more and more position data of indoor moving objects are recorded by sensors. As the basic work of moving objects database, index technique has become a hot research spot. Majority of moving objects index existed now are for outdoor moving objects which are not suitable for indoor environment, not only that, they only build index on geography coordinates of moving objects, lack supporting of semantic information which can offer effective support for management and analysis of indoor moving objects, there also will be a performance bottleneck when massive data are ingested and frequent querying are asked if implemented on traditional relational database. In this paper, we build a grid of indoor floor environment and create a map relation from grid to semantic cell, then utilize this map to semanticize indoor moving objects’ location if it was contained in a semantic cell, after this work we build an index called SCoII(Semantic Cell Oriented Indoor moving objects Index), SCoII can answer not only semantic spatio-temporal range query but also indoor moving object’s semantic trajectory query, which can support for semantic-based analysis of indoor moving objects. SCoII is implemented on HBase, so it also avoids the performance degradation of traditional relational database when encounting massive data and have indifferent performance of updating and querying without bottleneck, experiments’ results also showed that it can be adapt to big data. Supporting for semantic information of indoor moving object is the most important feature of SCoII, more data mining job can be done on indoor moving object’s semantic location and semantic trajectory such as the simple example given out at the end, management and analysis based on semantic of indoor moving objects will be convenient on SCoII, it lays a foundation of analysis work in the future.

**Key words:** indoor; moving objects; index; semantic; HBase

**\*Corresponding author**: XIE Chuanjie, E-mail: xiecj@lreis.ac.cn

**摘要：**随着室内定位技术的广泛应用，传感器记录了越来越多的室内移动对象的位置数据，索引技术作为移动对象数据分析的基础工作也得到越来越多的研究。已有索引技术多是针对室外空间的移动对象，不能支持室内移动对象数据的三维立体性、轨迹的复杂性、随机性等特点，这些索引技术也仅仅关注了移动对象的位置信息，忽略了语义信息，不能有效地支持室内移动对象的管理和分析，并且当面对海量的移动对象数据时，这些架构在传统关系型数据库上的索引都存在着性能瓶颈问题。本文提出了面向语义单元的移动对象表达模型，利用语义单元将室内移动对象的位置语义化，设计了SCoII (Semantic Cell Oriented Indoor moving objects Index)索引结构对室内移动对象的历史数据进行索引，能够有效支持语义时空范围查询、移动对象语义轨迹查询。索引基于HBase实现，能够适应大规模的并发更新与查询，具有良好的规模扩展性，规避了大数据给传统数据库带来的性能瓶颈问题，实现证明其具有较高的更新和查询性能。该索引的实现方便了基于语义的室内移动对象分析和数据挖掘工作，为以后的分析工作奠定了基础。

**关键词：**室内；移动对象；索引；语义；HBase

**收稿日期：**2016-07-26

**基金项目：**国家自然科学基金资助项目(41590845)

山西省-中科院科技合作项目(20141011001)

**作者简介：**张得群（1991-），男，河南新乡人，硕士生，主要从事室内移动对象管理与分析方面研究。E-mail：dequn92@foxmails.com

**\*通讯作者：**谢传节(1971-)，男，安徽潜山人，博士，副研究员，主要从事地理信息大数据研究。E-mail: [xiecj@lreis.ac.cn](mailto:xiecj@lreis.ac.cn)。

# 1引言

随着室内定位技术的发展和广泛引用，传感器记录了越来越多的室内用户的位置数据。这些海量的移动对象数据背后往往蕴含着丰富的与用户行为相关的重要信息，对这些信息进行数据挖据已经成为一个研究热点。相比于移动对象位置的坐标信息，位置的语义信息具有更加重要的利用价值，可以用来挖掘用户的移动模式、兴趣偏好等信息[1]。许多学者对移动对象的语义轨迹进行了相关性分析、聚类分析、位置匹配，完成了移动对象相似度[2]、路网匹配[3]、交通优化[4]、数据挖掘[5]等工作，可以看出，语义分析是移动对象分析的重要方面并且给这些数据挖掘工作带来重要的研究意义。在室内位置服务领域，除了定位服务提供移动对象的位置坐标外，室内各个房间所代表的空间实体在其性质、功能、用途等属性中包含了大量的语义信息，成为不同语义单元(Semantic Cell)，根据这些语义单元对移动对象进行管理和分析，建立移动对象数据库，是将移动对象位置语义化和数据挖掘的重点。

移动对象索引技术是提高移动对象数据库查询处理性能的关键，相比于其他地理信息数据库，移动对象数据库更新频率高，需要响应诸如轨迹查询之类具有时空两方面约束的复杂查询，因此寻求合适的移动对象索引方法减少搜索空间、加快查询响应速度是移动对象数据库的重要研究部分[6]。但现有的移动对象索引技术多是针对室外环境，并且仅仅关注于对象的位置数据，忽略了环境以及移动对象的语义信息，当处理基于语义约束的查询时，现有索引技术效率低下[7]，不能快速响应甚至不能响应查询。本研究在总结了已有的移动对象索引工作的基础上，设计了基于HBase的面向语义单元的室内移动对象索引，该索引能够有效支持语义时空范围查询、室内移动对象的语义轨迹查询，在此基础上，可以更方便地开展基于语义的室内移动对象数据挖据。

# 2 研究背景

## 2.1室内移动对象特点

相比于室外移动对象，室内移动对象数据还具有以下特点[8] [9]：

1. 室内移动数据的多维性，室内环境通常是多楼层构成的立体空间，不同于室外受限网络中的移动数据如出租车数据，或是可以投影到地球表面的飞行器位置数据等，室内移动对象的位置数据是三维的，且不能简单的投影处理，已有的位置模型不能有效表达室内移动数据；
2. 移动轨迹的复杂性和位置的随机性，由于室内研究的尺度比室外要小，整个环境和室内空间的几何信息必须是多边形表示的，移动对象不能脱离这些多边形，但在多边形内部又有很大的自由，因此室内移动轨迹虽然不如欧式空间的中的自由，但又要比受限网络中的复杂。位置的随机性是指移动对象的位置冲突而造成的运动方向和模式的改变，比如在楼道里面迎面走来一个人，必有一方偏离其原有模式，造成了位置上的随机性；
3. 移动对象的目的性更强。室内空间是由功能独立的单元构成，移动对象每到达一个单元，都与该单元的功能性直接相关，带有很强的目的性。室外移动对象的移动也具有目的性，但多数只能体现在运动的终点上，而在室内运动时间短，目的性更强，从一个室内单元到另一个室内单元背后都隐藏着一定的目的，传统的按照坐标表达的移动对象方法不能体现这些信息；
4. 位置的语义性。室内移动对象时时刻刻都处在某一个室内的语义单元中，因此更容易寻求该位置的语义信息，而目前移动对象数据模型主要着眼于描述移动对象的位置信息[10]，较少考虑语义信息。

考虑到以上特点，现有的室外移动对象表达模型都不能有效适用于室内移动对象位置的表达，因此基于这些室外位置表达模型建立的移动对象索引也同样不适用于室内移动对象，需要根据室内移动对象的特点，构建新的位置表达模型，设计新的索引方式。

另一方面，随着采集数据能力的提高，传感器获得的数据呈爆炸式地增长，传统依托在单机节点的空间数据管理方法已经很难满足大规模的数据存储、更新、检索需求，规模庞大且动态增长的海量信息造成的系统效率下降已经成为很多移动对象数据库的瓶颈问题。以HBase为代表的应大数据时代而生的分布式数据库给移动对象数据库带来了新的解决方案，HBase是根据Google公司Change等人发表的论文“BigTable: A Distributed Storage System for Structured Data”的一个开源实现[11]，其设计旨在能够提供从大规模数据集中随机和实时的高性能读写访问。空间数据既可以被传统的关系型数据库存储，又可以被 NoSQL(Not Only SQL)数据库存储[12]，为了充分利用分布式数据库的特性来提高移动对象数据库的更新和查询效率，可根据需求构建基于HBase的针对室内移动对象的时空索引结构。

## 2.1移动对象索引研究现状

Jensen等人[13]提出了室内移动轨迹索引结构—RTR-tree( Reader-Time R-Tree)和TP2R-tree( Time Parameter Point R-Tree)，从名字可以看出，这两种结构都是基于R-tree实现的。RTR-tree水平线段的组合来表示移动轨迹，可以支持空间范围查询，TP2R-tree通过把轨迹表示为带有时间属性的点的序列，从而支持移动对象的轨迹查询。甘早斌等人[14]在Jensen工作的基础上增加一棵R-tree进行移动对象索引，形成了DR-tree (Dual R-tree)，有助于提高室内移动对象轨迹查询的性能，但空间开销较大[15]。

Shin等人[16]提出的ACII (Adaptive Cell-based Index for Indoor moving objects)索引共有两层结构——MC(Memory Cell)和 MEMO (Memory)。MC结构将室内单元划分成固定大小的Cell建立空间索引树，并以这些Cell为标识对当前时刻所含有的移动对象建立Hash索引，用来索引实时移动对象；MEMO结构以移动对象为主键，存储移动对象的历史轨迹数据。该索引支持全时态的移动对象查询，可以实现较高的数据更新频率，适用于实时应用的场景，实验表明ACII最多使用R-tree及其变种树30%的空间，轨迹查询效率却要高于R-tree[16]。

贲婷婷等人[7, 17]提出了MQII ( Multiple queries indoor index)索引结构，该索引分别建立了Hash表、对象链表、桶链表来存储移动对象历史轨迹信息和当前位置信息。对象链表提供针对移动对象的索引，存放移动对象信息和移动对象的桶链表指针；Hash表存储室内单元信息和该室内单元对应的桶链表地址，每一个单元对应一个通链表；桶链表用来存储该室内单元中对象的信息，包括进出时间等。该方法有效支持对象查询、位置查询、范围查询、时间片查询，实验效率高于ACII索引，但其消耗空间较大，针对大数据量的存储与检索有待进一步改进。

以上的索引结构是建立在传统的关系型数据库上的，虽然能够实现是移动对象的管理与分析，但通常要受限于数据量、更新频率、查询频率等，存在不同的瓶颈。另一方面，分布式数据库如BigTable, HBase等虽然能够应对大规模的并发场景，但并不直接支持时空索引和属性索引，不满足移动对象存储与分析的要求，如果没有合适的二级索引，即使基于MapReduce 的并行计算框架，也要扫描全部数据才能够实现数据检索，造成了计算资源、时间的极大浪费，降低了效率[18]。如何基于分布式数据库的特点针对移动对象数据设计合理数据结构与索引结构，许多学者也做了很多的研究工作。

Nishimura等人[19, 20]设计的MD-HBase首先通过KD-tree和Quad-tree建立空间索引，之后再通过Z-ordering空间填充曲线将其降至一维，将Z-ordering编码作为HBase主键， MD-HBase能够实现较高效率的范围查询，但不支持对象查询，虽然也可以在空间索引树的结点上添加语义信息，但进行语义查询时需要先进行空间索引树搜索，再转换为Z-ordering编码，最后进行实际数据检索，造成了效率极大的下降，并且不能有效支持大范围的空间查询[18]。

LiShen等人[18]在HDFS上构建出的PyroDB，利用Moore空间填充曲线对时空数据进行降维，稳重提出了 Adaptive Aggregation Algorithm(A3) 算法，在进行数据检索前过滤误判的范围，加快了数据检索效率。PyroDB还通过改进HBase集群Split的过程，使得数据能够及时分布到新节点上，解决了数据热点问题，但其只支持时空范围查询。

GeoMesa[21]是一个开源的、分布式的时空数据库引擎，可基于 Accumulo, HBase, Cassandra, Kafka等数据库构建，GeoMesa提供了NoSQL数据库的快速时空数据检索，其角色如同PostGIS对PostgreSQL一样。GeoMesa索引移动对象数据的方法是将二维空间和时间的构成的三维空间均匀分割成时空三维立方体单元(3D Cube)，通过空间填充曲线Z-ordering的值作为HBase行健。GeoMesa通过建立多个索引数据表来支持非时空属性查询，每一个索引表实际上都包含了一个完整的数据备份，查询效率取决于三维立方体单元的大小，粒度越小，查询效率越高，但所耗费空间越大[22]，另外多个索引表中有个数据备份，极大地浪费了存储空间。

Chen等人[23]利用了HBase的内部机制建立了两层索引STEHIX(Spatio-Temporal HBase Index)，首先通过Hilbert空间填充曲线将空间数据的划分成等粒度的单元，将这些单元分布在不同的HRegion后，在HRegion内部对单元空间进行四叉分割细化并建立Quad-tree索引，并对时间进行分段索引，进行时空约束查询时只需要求出时间索引和空间索引的交集部分即可。STEHIX通过这两级索引实现时空范围查询和kNN(k-nearest neighbors)查询，与只设计行健的方式相比，该方法提高了HBase的读取速度，实验证明其效率要高于MD-HBase。

表1给出了几种移动对象的对比，不难看出，由于传统关系型数据库可以建立多属性索引，因此基于传统数据库建立的移动对象索引在对象查询、对象轨迹查询方面具有一定优势，但当面对海量的移动对象数据时，其更新效率和查询效率都不能满足需要。诸如HBase的分布式数据库只支持主键索引，通常是采用空间填充曲线或者利用GeoHash等方法将多维空间降至一维作为HBase的主键，因此多数索引只能实现时空范围查询，对象位置、轨迹查询则较难处理。另外这两种索引方法都忽略了移动对象的语义信息，只是简单的存储了数据，不能有效地支持移动对象数据分析、数据挖掘工作。因此，需要设计新的索引结构，既要能够存储和表达移动对象的语义信息，又要能够满足对象查询、对象轨迹查询的需要，同时还要解决大数据时代下的数据更新、查询效率下降等性能瓶颈问题。

表1 移动对象索引总结

Tab.1 List of Moving Objects Index

|  |  |  |  |
| --- | --- | --- | --- |
| 索引分类 | 索引名称 | 索引结构 | 范围查询 |
| 基于传统数据库的移动对象索引 | RTR-TP2R-tree | R-tree | 室内时空范围查询、轨迹查询 |
| DR-tree | R-tree | 室内对象轨迹查询 |
| ACII | 先 R-tree后 Hash | 全时态移动对象查询 |
| MQII | Hash后使用链表指针 | 对象查询、范围查询 |
| 基于分布式数据库的移动对象索引 | MD-HBase | KD-tree/Quad-tree + Z-ordering | 时空范围查询 |
| Pyro | Moore | 时空范围查询 |
| GeoMesa | Z-ordering | 对象查询、时空范围查询 |
| STEHIX | Hilbert + Quad-tree + 分段索引 | 时空范围查询，kNN |

# 3索引设计

## 3.1面向语义单元的位置表示模型

已有的室外位置模型不能有效支持室内移动对象的表达和索引，本文提出面向语义单元的位置表示模型，包括语义单元、语义位置、语义停留点、语义轨迹等概念。

与室外空间不同，室内空间多数是由多个相互独立的功能区组成，如商场内的一个商铺、走廊、扶梯等，为这些独立的功能区域定义一个唯一的ID，作为最小的语义单元( Semantic Cell)，当用户进入这些语义单元且被记录位置，就可以将该单元的语义信息关联到位置记录上，获得移动对象的语义位置，由语义位置的便可以计算该移动对象的语义轨迹。每一个语义单元可以用定义表示为：

(1)

其中，cid表示该单元的Id值，name表示该单元的语义名称，geom表示该单元的几何信息，是室内三维空间的一个表示，用来判断移动对象与其的位置关系—即移动对象是否落在语义单元内，category表示该单元的分类，如在商场里可以为“女装”、“女鞋”等，tags是该店铺拥有的语义标签，如“时尚”、“休闲”等，也可以再附加其他需要的语义信息。

一个室内移动对象的位置记录可以表示为：

(2)

而一个移动对象的语义位置则是由位置记录关联语义信息，即本文中的语义单元Id构成，可以表示为：

(3)

语义停留点与语义位置不同，语义停留点为移动对象长时间处于同一语义单元，是抽象概念上的一个点，而语义位置是一次定位记录附加语义信息得到的，所以一个语义停留点可能是由多个语义位置记录组成，我们只需要记录该停留点的语义信息、开始时间和结束时间即可。简单用一个四元组表示，cid为其所停留的语义单元id，表示停留的开始时间，表示结束时间：

(4)

然后，一个室内移动对象的语义轨迹就可以表示为语义停留点的一个序列，表示为：

(5)

使用面向语义单元的移动对象模型，我们并不直接处理对象的实际坐标数据，其三维立体性隐含在语义单元的几何信息中，在同一个语义单元中，移动对象的位置包括其轨迹被抽象为一个语义停留点，整个实际的移动轨迹也被简化为语义轨迹，不仅极大地降低了原始轨迹的复杂程度，弱化了位置随机性带来的影响，还能有效支持语义信息的表达，更加方便地进行语义分析和用户特征提取等工作。

## 3.2面向语义单元的移动对象索引结构设计

本文基于面向语义单元的位置表示模型提出的面向室内语义单元的移动对象索引，索引结构包括Hash Table、语义空间表、对象语义位置表三部分。Hash Table用来存储和索引室内语义单元的信息，语义空间表用来存储所有经过该语义单元的移动对象及原始数据记录，对象语义位置表用来存储移动对象的语义位置，整体结构如下图所示。

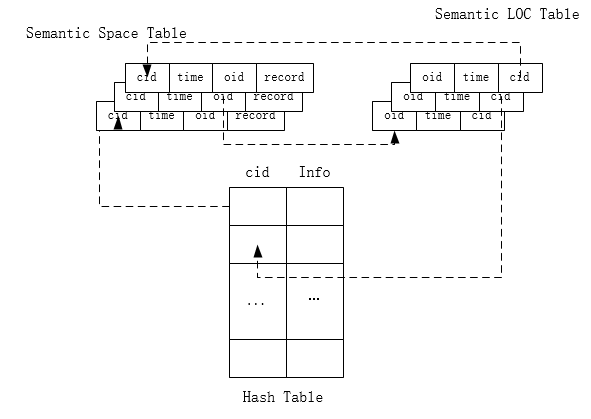


图1 面向语义单元的室内移动对象索引结构

Fig.1 Structure of SCoII

1. Hash Table

Hash Table 用来存储室内单元的语义信息，完成从cid到Cell的映射，可以根据cid快速查询该室内单元语义信息。

1. 语义空间表

语义空间表用来存储移动对象的原始数据记录，存储所有经过该语义单元的移动对象及原始数据记录，实现语义时空范围查询。

1. 对象语义位置表

对象语义位置表存储移动对象在每个定位时间戳是对应的语义位置信息，通过对语义位置构建时间正序的序列，便可以获得该对象在特定时间范围内的语义轨迹。

由于HBase不支持行健索引外的属性索引，必须针对时空信息进行降维，通过合理的规则将两者结合起来，共同组成行健，才能利用HBase的行健索引来达到建立移动对象索引的目的。根据语义位置的定义可以看出，移动对象的语义位置可以使用其所处的室内语义单元的cid进行关联，cid也代表空间信息，在此之上结合时间属性，就可以做到语义上的空间与时间共同索引，这是本文索引的思想基础。由于同一时间同一Cell内会有多个移动对象，为了有效区别开来，再结合每个对象的ID就可以唯一确定一个不重复的RowKey, 表示为：

(6)

其中，Cell用Cell-Id唯一表示，需要保证全局唯一性时间统一采用UNIX时间戳来表示，精确到秒，为十位整数；Object这里用移动对象设备的MAC地址来唯一标示。

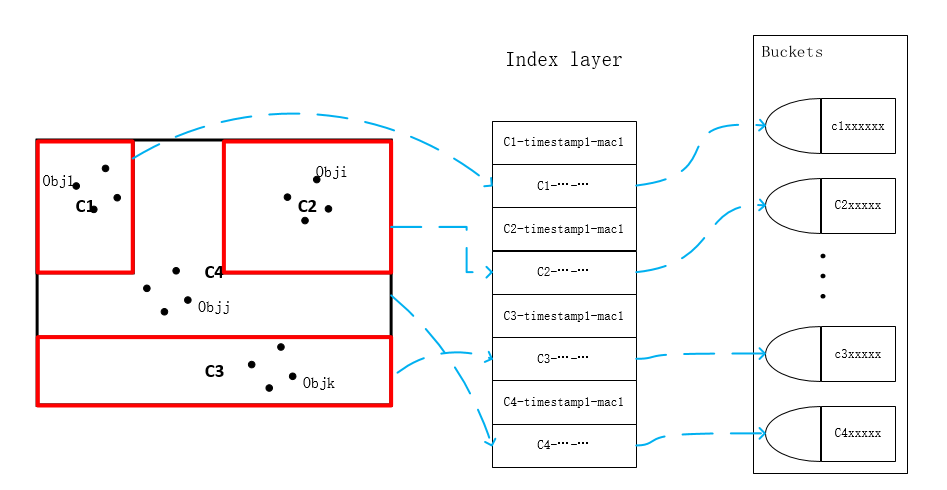


图2 语义空间表结构

Fig.2 Structure of Semantic Space Table

由于HBase不支持除RowKey外的属性索引，对于移动对象的轨迹查询，我们需要另外建立一个索引表，其RowKey组成与原始数据的组成部分一致，但将Object部分与Cell部分对换，这样给定移动对象和时间范围就可以查询出该对象的语义轨迹，如果要查询具体的位置信息，则可以针对查询出来的语义位置表RowKey进行变换成语义空间表的RowKey，继而获得详细数据。需要指出的是该索引是稠密索引，并且只存在行键，不需要存储实际数据。语义轨迹索引的RowKey设计如下：

(7)

综上所述，面向语义单元的室内移动对象索引的存储结构如图3所示。

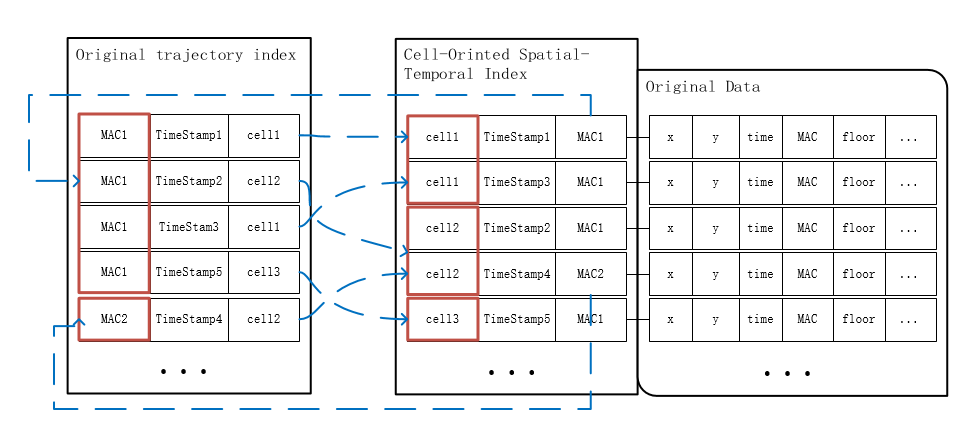


图3 SCoII存储结构

Fig3. Store Structure of SCoII

## 3.3移动对象位置语义化

通过判断对象所落入的语义单元是讲移动对象位置语义化的必要过程，由于室内空间单元形状的不规则性，传统的R-tree及其变种树建立空间索引会引起较多的MBR( Minimum Bounding Rectangle)重合，直接利用这些索引结构判断移动对象与室内单元的位置关系会出现多条搜索路径，效率低下。考虑到室内定位精度的影响，本文提出GridCell多对一的映射关系，利用定位点落入Grid判断的高效性来提高数据的更新速度，判断规则如下：

假设与均相交，相交的面积分比为，给定规则如果满足

（8）

则可以近似认为落入到中的定位点就包含在内，如果不满足条件的话，对 进行四等分，直至条件成立。假设，如图4所示，与均相交，但任何一部分面积都不满足假定规则，所以对进行四叉划分，分别为，，，，计算得到,，可以认为落入这三个区域的移动对象都在内，由于，所以需要对G(12)再次进行四等分，直至递归结束。

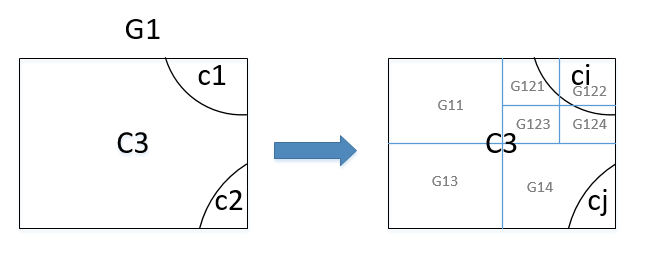


图4 Grid四分细化过程

Fig4. Procedure of Grid SubDivision

图5中，室内空间建立Grid索引后，编号为的网格完全落入单元的内，所以当移动对象落入时可以完全确定其肯定位于内部，虽然只有部分位于内，但其重合部分面积的比例，我们可以近似认为落入到中的定位点位于中，对于单元格内的移动对象，其判断结果是位于外部而不是内部。映射结果如图所示，可以根据定位点的坐标快速判断移动对象所处的Cell。虽然此判断方法具有一定的误判，但可以极大地提高效率，考虑到目前室内定位精度的影响，可以接受这种误差。

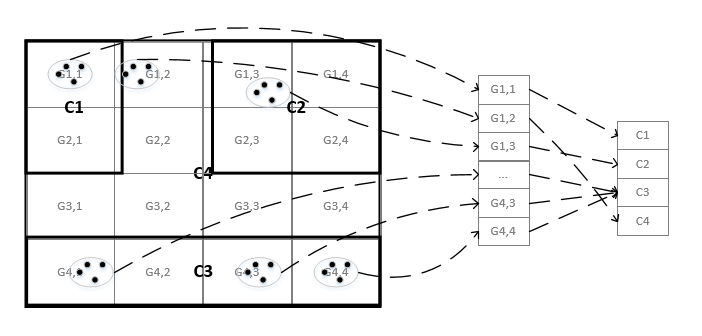


图5 利用Grid索引快速判断移动对象所处的语义单元

Fig5. Utilize Grid Index determine cell which an IMO dropped in.

## 3.4算法描述

### 3.4.1Grid索引生成算法

算法1描述了Grid建立及完成GridCell映射的过程，对于给定的室内空间，第一步计算整个楼层的外界矩形，作为最大Grid，将此Grid进入队列，当队列不为空循环以下过程：队首出队，计算所有与相交的室内单元，如果满足条件(8)，则将此Grid与和其重合面积最大的室内单元添加到Map中，所有落入的移动点都视为落入中，否则将四等分得到SubGrids，并将四个子Grid进入队列，队列为空时整个过程结束。

### 3.4.2数据更新算法

给定数据记录,首先根据得到该定位记录落入的Grid,根的GridCell的映射关系可以得出该记录所处的室内单元，然后可根据公式(6)、公式(7)即可计算出该对象RowKey和对应的轨迹索引MacIdxRowkey作为HBase行健更新数据，过程如算法2所述。

### 3.4.3语义时空范围查询算法

语义时空范围查询可以用来检索该单元的总访问量、平均访问量、移动对象密度等信息，可以用来预计客流量、人员控制等。该查询需要输入三个参数，分别为语义单元，开始时间和结束时间，再根据过滤规则进行数据检索，得到检索结果后进行集合操作，最后返回该集合，整个过程如算法3所述。

### 3.4.4语义轨迹查询算法

因为在语义轨迹索引表中已经包含了语义单元的Id，所以只需要查询该表和Hash表就可以得出移动对象的语义轨迹，如果需要实际的位置轨迹，根据语义轨迹扫描原始记录表就可获得。算法4描述了具体过程：根据移动对象Id和开始结束时间确定扫描范围，仍需要同时满足两个前缀过滤，再根据语义轨迹中的语义单元Id从室内Hash表中检索出实际的语义信息，根据时间顺序排列进行集合操作便可以得到语义轨迹。

# 4系统实现

本文在3台物理计算机上部署了Hadoop 2.6.3、HBase 1.1.2集群环境，其中一个节点作为Master节点,另外两个作为Slave节点。硬件参数为：CPU(intel(R) Xeon(R) E5-2609 @2.40GHz)、RAM(16GB)、Disk(1 TB),操作系统为CentOS 6.4。

实验数据为北京某商场的定位数据，其中每条的记录如表2所示。

表2 定位记录字段组成

Tab.2 Fields of a positioning record

|  |  |
| --- | --- |
| 字段 | 含义 |
| x | X坐标值 |
| y | Y坐标值 |
| floor | 定位点所在的楼层，20040表示F4，10020表示B2 |
| time | 数据采集的时间，精确到秒 |
| mac | 定位对象设备的物理地址 |

## 4.1 GridCell映射关系效率测试

首先研究对比了利用映射关系与直接利用空间索引树判断移动对象落入语义单元的效率。空间索引树采用JTS (Java Topology Suite)开源项目提供的STR树(Sort Tile Recursive tree)。实验统计了不同数据量下两者判断相同定位点数花费的时间，从图中6可以看出，利用映射的时间远小于直接利用空间索引树的时间，效率比约为200:1。利用映射关系可以根据移动对象的地理位置在O(1)时间内计算所处的Grid编号，从HashMap中求出对应的语义单元的时间效率同样为O(1)，而在空间索引树中进行位置关系判断所需的平均时间为O(logn)，其中n为语义单元的个数，所以两种方法的效率差别可以达到两个数量级。使用GridCell虽然极大地提高了判断语义位置的效率，但这是一种非严格的关系判断，存在一定的误判。

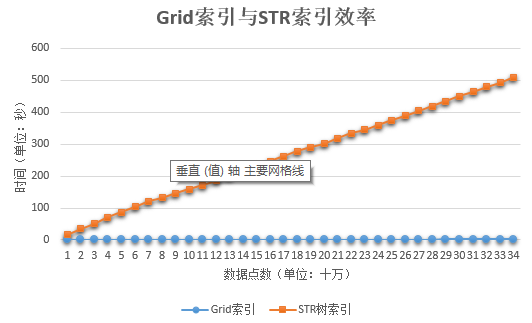


图6 GridCell与STR判断语义单元效率对比

Fig.6 Efficiency comparison between GridCell and STR-tree

## 4.2数据更新效率测试

HBase作为分布式数据库，具有较高的数据更新效率是其特点之一。实验对移动对象数据的更新速度进行了统计，如图7所示，针对实验数据，每万条记录入库的时间为0.35秒左右。实验还选择了PostgreSQL作为关系型数据库的代表，与HBase数据库进行了数据更新速度对比。PostgreSQL是由加州大学伯克利分校计算机系研发的开源数据库，其单表容量在理论上能够达到TB级，在大数据情况下具有完全可用的伸缩性能，有实验证明其更新效率已经高于分布式数据库MongoDB。但从图8中可以看出当数据量增加时，HBase的更新速度明显要快于PostgreSQL,而且数据量越大，效率差越大，一是因为HBase是多节点存储，数据会按照主键分布到不同的节点上，各个节点独立操作，这种并行的处理加快了更新的效率；二是因为HRegion只用在Memory File达到一定数据量后才执行一次磁盘操作，这种批量处理的方式也加快了更新的效率。总的来说，在海量的室内移动对象管理场景下，选择分布式数据库要优于传统的关系型数据库。

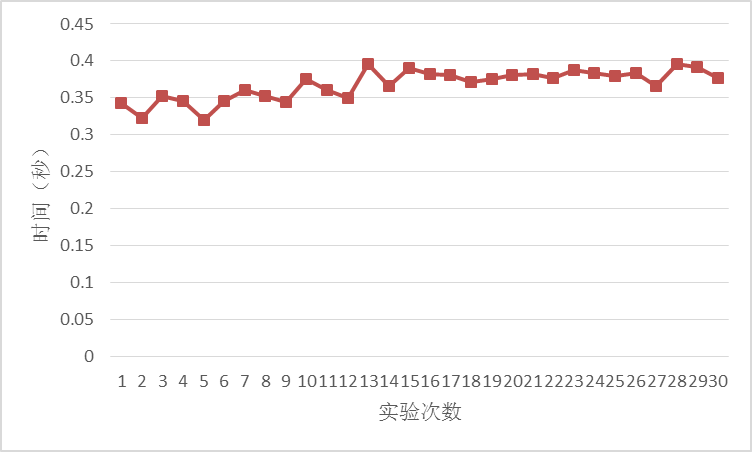


图7 HBase数据更新效率(秒/万)

Fig.7 Time needed of updating every ten thousand points on HBase

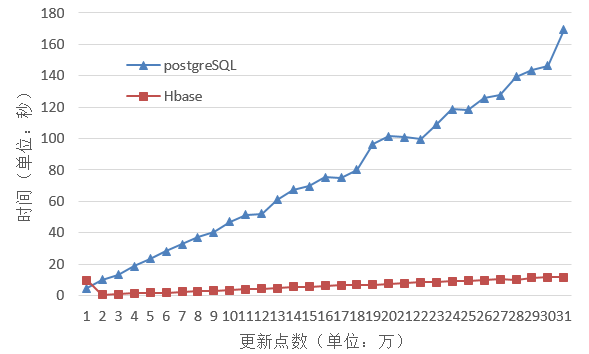


图8 HBase与PostgreSQL更新效率对比

Fig.8 Efficiency comparison between HBase and PostgreSQL

## 4.3基于语义轨迹的用户特征提取

以其中一个用户为例，从数据库中查询其从2014-04-01 08:00 至2014-04-01 18:00语义轨迹，耗时50毫秒，共得到60个语义节点，经过对语义轨迹进行整理，具有可用信息的结果如表3所示。

表3 语义轨迹查询结果示例

Tab.3 A sample result of Semantic Trajectory query

|  |  |  |  |
| --- | --- | --- | --- |
| Cell名称 | 进入时间 | 离开时间 | 停留时长（单位：秒） |
| Rouge Diamant | 12:21:24 | 12:21:26 | 2 |
| 过道 | 12:21:26 | 12:21:28 | 2 |
| Rouge Diamant | 12:21:28 | 12:21:31 | 3 |
| Bcuthentique | 12:21:31 | 12:22:29 | 58 |
| FIVE PLUS+ | 12:23:14 | 12:24:01 | 47 |
| TRENDIANO | 12:27:51 | 12:50:04 | 1333 |
| Levi's ladies | 12:50:04 | 13:04:53 | 889 |
| Jack jones | 13:27:16 | 13:28:16 | 60 |
| TEENIE&WEENIE | 13:28:58 | 13:30:56 | 118 |
| TEENIE&WEENIE | 13:41:22 | 17:43:14 | 14512 |
| H&M | 17:43:14 | 17:57:07 | 833 |

查询结果中的前三行，用户存在从Rouge Diamant出去再回到Rouge Diamant且时间都比较短的不正常情况，根据实际情况分析，用户在过道中停留2秒有两种可能，一是定位误差导致，二是在从的映射引来的误差，也就是说该用户有可能一直停留在Rouge Diamant中，粗略地分析，在过道中的2秒应该归属于Rouge Diamant中。虽然室内定位精度和位置语义化过程引入的误差会影响数据挖掘工作的进行，但是结合了语义信息后，可以根据一定规则对上述两种误差进行一定程度的修正。

其他节点的均为整理后的数据，用户停留时间较长（能达到一分钟或超过一分钟）的室内单元有Bcuthentique, FIVE PLUS+, TRENDIANO, TEENIE WEENIE, Levi's ladies, H&M等，其中在Levi's ladies停留约有15分钟，在TRENDIANO停留时间超过半个小时，在TEENIE WEENIE停留超过了4个小时。这些单元的品牌所共有的标签为“女性 时尚 年轻 欧式”，由此可以初步判断，此用户应该是一位年轻时尚的女性顾客，欧式可能是其偏好的风格，并且根据常理猜测，该用户极有可能在停留时间最长的TEENIE WEENIE购买了商品，进一步也可以推断出其消费水平等更多信息。

# 5总结

本文在总结了已有移动对象索引的基础上，建立了基于HBase的面向语义单元的室内移动对象索引，该索引能够支持语义时空范围查询、移动对象语义轨迹查询，实验证明其具有较好的更新、查询性能。支持语义轨迹查询是该索引的特点，将语义信息集成到移动对象轨迹的表达中，能够更容易地从众多轨迹信息中提取运动规律和模式，更方便地进行室内用户行为识别等工作，为更好的室内位置服务提供了基础。文章最后通过一条语义轨迹的简单分析，完成了粗略的用户属性信息推断，在此基础上，可以利用更多的分析手段描述更精细的用户画像，对其行为进行更精准的识别与预测。

虽然该索引是针对历史数据建立的，但其特性也适合索引移动对象的实时数据，HBase数据库良好的更新性能、查询解决了传统数据库面对海量数据时的瓶颈问题。文中直接存储了移动对象的语义位置，针对语义轨迹查询频率较高的情况，如何建立历史数据库存储查询结果，将本索引上升一层作为实时数据索引，进一步系统效率，成为该索引的改进方向之一。

另外本文只研究了数据的存储和索引问题，考虑到传统计算环境难以满足海量数据的分析要求，如何利用Hadoop、Spark等并行计算环境加快数据的分析速度，也是日后移动对象分析的研究重点。

参考文献（References）:

[1]王倩. 室内移动对象轨迹相似性度量与应用 [D], 中国科学技术大学, 2015.

[2]YING J C, LU H C, LEE W C, et al. Mining user similarity from semantic trajectories. Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks [C]. ACM, 2010, 19-26.

[3]HWANG J R, KANG H Y, LI K J. Spatio-temporal Similarity Analysis Between Trajectories on Road Networks [M]. Springer Berlin Heidelberg, 2005.280-289.

[4]廖律超, 蒋新华, 邹复民, 等. 一种支持轨迹大数据潜在语义相关性挖掘的谱聚类方法 [J]. 电子学报, 2015, 43(05): 956-964.

[5]齐凌艳, 陈荣国, 温馨. 基于语义轨迹停留点的位置服务匹配与应用研究 [J]. 地球信息科学学报, 2014, 16(5): 720-726.

[6]方颖. 移动对象数据库中移动对象索引方法研究 [D], 武汉大学, 2010.

[7]贲婷婷, 秦小麟, 王丽. 基于语义和访问权限的室内移动对象索引 [J]. 计算机科学, 2015, 42(03): 178-184.

[8]杨彬. 室内移动对象的数据管理 [D], 复旦大学, 2010.

[9]LU H, YANG B, JENSEN C S. Spatio-temporal joins on symbolic indoor tracking data. Data Engineering (ICDE), 2011 IEEE 27th International Conference on [C]. IEEE, 2011, 816-827.

[10]汪娜. 面向室内空间的时空数据管理关键技术研究 [D], 中国科学技术大学, 2014.

[11]冯晓普. HBase存储的研究与应用 [D], 北京邮电大学, 2014.

[12]ZHANG N, ZHENG G, CHEN H, et al. Hbasespatial: A scalable spatial data storage based on hbase. 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications [C]. IEEE, 2014, 644-651.

[13]JENSEN C S, LU H, YANG B. Indexing the trajectories of moving objects in symbolic indoor space [M]. Advances in Spatial and Temporal Databases. Springer. 2009: 208-227

[14]甘早斌, 袁永光, 赵贻竹, 等. 基于 DR-tree 的室内移动对象索引研究 [J]. 计算机科学, 2012, 39(10): 177-181.

[15]金培权, 汪娜, 张晓翔, 等. 面向室内空间的移动对象数据管理 [J]. 计算机学报, 2015, 38(09): 1777-1795.

[16]SHIN S, KIM G, BAE H. Adaptive cell-based index for moving objects in indoor [J]. KSII Transactions on Internet and Information Systems (TIIS), 2012, 6(7): 1815-1830.

[17]贲婷婷, 秦小麟, 许建秋. 支持多种查询的室内移动对象索引 [J]. 计算机研究与发展, 2015, 52(09): 2002-2013.

[18]LI S, HU S, GANTI R, et al. Pyro: a spatial-temporal big-data storage system. 2015 USENIX Annual Technical Conference (USENIX ATC 15) [C]. 2015, 97-109.

[19]NISHIMURA S, DAS S, AGRAWAL D, et al. MD-HBase: design and implementation of an elastic data infrastructure for cloud-scale location services [J]. Distributed and Parallel Databases, 2013, 31(2): 289-319.

[20]NISHIMURA S, DAS S, AGRAWAL D, et al. MD-HBase: a scalable multi-dimensional data infrastructure for location aware services. Mobile Data Management (MDM), 2011 12th IEEE International Conference [C]. IEEE, 2011, 7-16.

[21]HUGHES J N, ANNEX A, EICHELBERGER C N, et al. GeoMesa: a distributed architecture for spatio-temporal fusion. SPIE Defense+ Security [C]. International Society for Optics and Photonics, 2015, 94730F-94730F-94713.

[22]WHITMAN R T, PARK M B, AMBROSE S M, et al. Spatial indexing and analytics on Hadoop. ACM Sigspatial International Conference on Advances in Geographic Information Systems [C]. ACM, 2014, 73-82.

[23]CHEN X, ZHANG C, GE B, et al. Spatio-temporal queries in HBase [M]. 2015 IEEE International Conference on Big Data. 2015: 1929-1937.