

# 基于地理空间大数据的高效索引与检索算法

赵慧慧<sup>1,2</sup> 赵凡<sup>2,3</sup> 陈仁海<sup>1,2</sup> 冯志勇<sup>1,2</sup>

<sup>1</sup>(天津大学智能与计算学部 天津 300350)

<sup>2</sup>(天津大学深圳研究院 广东深圳 518000)

<sup>3</sup>(天津大学国际工程师学院 天津 300350)

(1442700849@qq.com)

## Efficient Index and Query Algorithm Based on Geospatial Big Data

Zhao Huihui<sup>1,2</sup>, Zhao Fan<sup>2,3</sup>, Chen Renhai<sup>1,2</sup>, and Feng Zhiyong<sup>1,2</sup>

<sup>1</sup>(College of Intelligence and Computing, Tianjin University, Tianjin 300350)

<sup>2</sup>(Shenzhen Research Institute of Tianjin University, Shenzhen, Guangdong 518000)

<sup>3</sup>(Tianjin International Engineering Institute, Tianjin University, Tianjin 300350)

**Abstract** In recent years, with the rapid development of advanced technologies such as intelligent target recognition, electronic sensors, collaborative control and computer networks, intelligent transportation systems have achieved qualitative leapfrogging. Modern intelligent transportation systems can realize intelligent transportation of vehicles, roads and clouds management platform. However, the intelligent transportation system relies on a large amount of two-dimensional geospatial information data generated every day. Therefore, how to efficiently store and query large-scale geospatial data is of great significance for the future popularization and development of the intelligent transportation system. However, due to the complexity of urban traffic information, large amount of data, and fast update speed, the current spatial indexing technology is difficult to efficiently search for two-dimensional geospatial information data. In order to optimize the storage organization structure of two-dimensional geospatial information data under spatial big data and improve retrieval efficiency, this paper proposes a spatial index tree construction algorithm for multi-layer slice recursion of two-dimensional geospatial information data (multi-layer slice recursive, MSR). The proposed algorithm first sorts and divides the first dimension of the map data to generate FD slices. Then, the second dimension of the map data in the FD slice is sorted to generate SD slices, and in the SD slice, the current slice and the adjacent slices are divided into spatial objects. Finally, data clustering operation is performed on the comparison between the length of the spatial object and the node capacity, and the MSR Tree is recursively generated from the bottom up by judging whether all the slices complete the clustering operation. Experimental results show that the query performance of the 2-dimensional space storage structure constructed by the MSR algorithm is better than the most representative spatial indexing technology based on the R-tree batch-loading algorithm (sort tile recursive, STR), STR-grid

收稿日期:2019-08-16;修回日期:2019-11-18

基金项目:国家自然科学基金项目(61702357,61672377);深圳市科技创新委员会学科布局项目(JCYJ20170816093943197);天津市自然科学基金项目(18JCQNJC00300);天津大学北洋学者青年骨干教师项目(2019XRG-0004)

This work was supported by the National Natural Science Foundation of China (61702357, 61672377), the Shenzhen Science and Technology Foundation (JCYJ20170816093943197), the Natural Science Foundation of Tianjin (18JCQNJC00300), and the Beiyang Scholar Foundation of Tianjin University (2019XRG-0004).

通信作者:陈仁海(renhai.chen@tju.edu.cn)

hybrid algorithm (str-grid), and efficient geometric range query (EGRQ).

**Key words** two-dimensional geospatial information; spatial indexing technology; big geo-data; MSR algorithm; clustering

**摘要** 近年来,随着智能目标识别、电子传感器、协同控制以及计算机网络等先进技术的快速发展,智能交通系统实现了质的飞越,现代智能交通系统可以实现车、路、云端于一体的智能交通运输管理平台。但智能交通系统依赖于每天产生的大量的2维地理空间信息数据,因此,如何对大规模的地理空间数据进行高效的存储和查询对智能交通系统未来的普及和发展具有重要意义。然而,由于城市交通信息的复杂性、数据量大、更新速度快等特征,当前的空间索引技术很难针对2维地理空间信息数据进行高效的检索。为了优化空间大数据下2维地理空间信息数据的存储组织结构、提高检索效率,提出了一种对2维地理空间信息数据进行多层切片递归的空间索引树构造算法(multi-layer slice recursive, MSR)。提出的算法首先对地图数据第1维度进行排序划分切片,生成FD(first division)切片;然后对FD切片中的地图数据进行第2维度排序,生成SD(second division)切片,在SD切片中对当前切片和相邻切片划分空间对象;最后对空间对象长度与节点容量比较进行数据聚类操作,通过判断所有切片是否完成聚类操作,自下而上递归生成MSR树。实验表明,MSR算法构建的2维空间存储结构的查询性能优于现在最具代表性的空间索引技术基于R树的批量加载算法(sort tile recursive, STR)、STR-网格混合算法(str-grid)及高效几何范围查询算法(efficient geometric range query, EGRQ)。

**关键词** 2维地理空间信息;空间索引技术;空间大数据;MSR算法;聚类

中图法分类号 TP311

近年来,国家城市化进程不断加快,2维地理空间信息数据更新迅速,其中道路网络数据不仅是路径规划、城市建设、位置服务及智能交通的基石,更为数字城市、智慧城市的发展奠定了基础<sup>[1]</sup>。在城市信息化浪潮与数据科学崛起的共同推动下,智慧城市的发展不仅需要庞大的地理空间数据作为载体,更加需要高效的空間检索技术推动智慧城市的发展<sup>[2]</sup>。地理空间大数据作为大数据的一个分支,正在全球范围内迅速崛起<sup>[3]</sup>,空间大数据的时代已经来临。

空间大数据无疑将重构很多行业的商业思维和商业模式,其价值不言而喻<sup>[4]</sup>。与此同时,随着定位技术的发展,空间大数据的位置标签越发精确。例如,有了WiFi、卫星定位、移动通信蜂窝定位技术及其变速器、微陀螺、速度传感器等多种多样的微小传感器的支持,才使得虚拟社交网络系统(social network system, SNS)发展到真实世界与虚拟世界相融合的基于位置的社交网络(location based social network, LBSN),并成为互联网企业的必争之地。在人工智能领域,无人驾驶作为其行业最受关注的应用之一,在其应用中道路网络数据是其必不可少的实现基础。例如,高精地图需要个性化驾驶支持,包括各种驾驶行为建议,如最佳加速点及刹车

点、最佳过弯速度等,以提高无人驾驶的舒适度<sup>[5]</sup>。并且无人驾驶汽车可以通过本地计算来实时地对当前路况做出反应,而不用连接到高延迟的远程控制中心<sup>[6]</sup>。在智能交通系统领域,信息采集设备的普及,每天都会采集海量的2维地理空间信息数据,这些数据极为宝贵<sup>[7]</sup>。例如,通过对某个时间段某个路口采集到的车辆情况对该路口的车流量进行分析,可以获悉该路况,向司机及时提醒道路的拥堵状况等。在遥感领域,地理信息系统和定位导航等技术逐渐商业化公众化之后,遥感数据观测与服务也正在从政府主导的公益应用向公众与商业机构主导的商业与公众应用扩展<sup>[8]</sup>。在日常生活中,人类所产生的数据有80%和空间位置有关<sup>[9]</sup>,具有互动、非专业、实时、泛在、按需服务、SOA特征的新地理信息时代<sup>[10]</sup>必将是空间大数据(big geo-data)备受瞩目的时代。

大数据时代科学研究是一个大科学、大需求、大数据、大计算、大发现的过程<sup>[11]</sup>。而空间大数据是在城市基础地理信息基础上,将各个曾经孤立的、服务内容单一的,但具有城市管理功能的空间信息、动态信息和公共信息等城市及信息聚合而成面向大众的、跨领域的、跨系统的社会化服务信息<sup>[12]</sup>。空间大数据作为大数据与地理信息数据的融合体,在信息

时代扮演着重要的角色,已经逐渐发展成为当今社会最基本的信息服务之一。然而,面对海量的数据检索时,单单依靠传统的空间索引技术,频繁的插入和删除操作会增大 2 维地理空间信息数据地重叠率,数据存储中存在许多彼此相似的数据块<sup>[13]</sup>,从而导致查找路径不唯一的情况增多,查询效率变低。

如何在分布式存储的模式下设计空间索引、提高检索效率,是空间索引技术未来重要的发展方向。本文以空间索引技术为主线,总结空间大数据下空间索引技术的背景、研究现状,分析不同索引类型的优劣;重点讨论每个时期空间索引技术的变化与技术创新;然后提出多层切片递归空间索引算法(multi-layer slice recursive, MSR),并与当前主流的空间索引技术基于 R 树的批量加载算法(sort tile recursive, STR)、STR-网格混合算法(str-grid)及高效几何范围查询算法(efficient geometric range query, EGRQ)进行实验对比。实验结果表明,MSR 算法不仅减少了道路网络空间数据的重叠度,提高了检索效率,而且可以很好地利用内存空间,减少存储空间的浪费。最后结合其面临的机遇与挑战,对 2 维地理空间信息数据做出展望。本文提供了实验数据以及实验代码的 GitHub 链接<sup>[14]</sup>,以供其他研究人员研究。

## 1 相关工作

2 维地理空间信息数据是地理空间信息的重要组成部分。2 维地理空间信息数据除了具有一般大数据体量(volume)大、类型多样(variety)、处理速度(velocity)快、价值密度(value)低、真实性(veracity)高等特点外<sup>[15]</sup>,还具有其自身鲜明的特点,主要体现在:1)全样本持续更新,数据呈现的是全样本的、客观的、过程的以及连续性的数据类型特点,数据资源更加全面、丰富、时效性更强;2)多维度,道路网络数据不仅局限于点、线、面,向着更高维空间发展;3)多粒度多层次,道路网络数据中碎片化、零散化、低效率的数据查询和分析普遍存在。面对如此规模庞大并且复杂的 2 维地理空间信息数据,传统的空间索引技术无法满足大规模数据快速检索和查询等需求,此时,需要查询效率高效的空间索引技术。

空间索引结构是一种辅助性数据结构,它基于空间数据的位置、形状以及对象之间的空间关系来组织空间数据结构,目的是提高数据检索及其他空间操作的效率。传统的空间索引技术主要分为三大

类,分别为基于网格的空间索引、基于树结构的空间索引、混合空间索引。

1) 基于网格的空间索引。基于网格的空间索引技术通过横竖线条把空间区域划分为若干个大小相同的网格,每个网格中包含空间对象,当用户查询时,首先检索出用户查询的空间对象所在网格,再从网格中定位到目标空间对象的位置。网格空间索引的方式很简单,当数据分布均匀时,其空间查询效率很高。但是网格的大小会影响到索引表的大小,如果网格太小,索引就会膨胀,不但查询效率低,而且对索引表的维护费用也会增加。目前有很多学者对此做出的分析和研究,提出了不同的解决方案,分别为基于规则网格划分的空间索引、基于多级空间网格划分的空间索引、自适应层级网格化分空间索引。

2) 基于树结构的空间索引。在基于树结构的空间索引技术中,具有代表性的树结构分别是 R 树与二叉树空间索引。国内典型的地理信息软件如 MapGIS 与 SuperMap,其空间索引技术采用的是二叉树索引;国外比较著名的地理信息软件如 MapInfo 公司的 MapInfo,其空间索引技术采用的是 R 树索引;以及国外 Oracle 公司的 Oracle\_Spatial 组件采用二叉树与 R 树混合的空间索引技术。

R 树是一种高度平衡树,它是在 B 树基础上扩展了多维空间。R 树索引技术的本质是面向对象的分割索引技术,分割后的区域与每个结点相对应,每个结点又与 1 个磁盘页对应,尽管磁盘存储系统有效地提升了带宽,但访问延迟难以得到有效解决<sup>[16]</sup>。R 树不但能控制树的深度,而且还能采用最小外包矩形<sup>[17]</sup>表示空间实体,是最早支持多维空间存取的方法之一。该算法提高了空间数据检索和空间操作的效率。

二叉树是另一种常见的树状索引结构,可分为点二叉树和区域二叉树,适用于点数据、区域数据以及高维空间数据的索引结构建立。在二叉树结构中,空间要素标识都记录在外包络矩形覆盖的每个叶结点中,所以在内存中的层次树状结构的查询效率较高。

3) 混合结构空间索引。混合索引结构汲取了各个索引结构的长处,多种空间索引的融合也是空间索引技术发展的方向,其大致可以分为 3 类:基于网格与树结构的混合索引、基于多个树结构的混合索引以及图与树结构的混合索引。

网格与树混合索引的典型是基于固定网格与二叉树结合的空间索引机制。这种混合索引机制通过

使用索引结点表和 Hash 技术管理类对象结点表,达到节省存储空间的目的,因此有效地减少了磁盘读取次数,显著提高了空间查询速度.空间对象与叶子结点出现了多对多的情况,但是相比之下,基于固定网格与四叉树结合的空间索引机制,能够使大面积空间对象重复减少;其不足之处在于算法的空间复杂度比较高,维护比较困难.

多个树结构的混合索引有很多,具代表性的是 QR 树,首先利用四叉树将数据库的研究区域分割为多级子索引空间,然后将 R 树索引应用到各级中的每个子空间.这种混合索引机制的优点是利用缩小的空间范围提高了查询效率,数据量越大索引效果越好.但是该方法没有考虑到大空间对象对数据的影响,在检索大范围空间对象时,其检索效率往往不高;索引占用的空间也比 R 树大.

图与树的混合索引机制,一般将图论的思想应用在空间索引范畴上.基于 MR-Tree 空间索引的 Voronoi 图改进及其并行空间查询方法,在 MR-Tree 索引基础上,引进了邻近关系判断,以替代范围查询中的冗余计算,有效地提高了分布式环境下的空间操作的效率.

国外研究空间索引的成果最早可追溯到 20 世纪 70 年代文献[18]提出的四叉树索引,这是一种树状数据结构,在每个结点上会有 4 个子区块,将数据空间均匀的划分为 4 个象限,数据范围可以是矩形或其他任意形状.文献[19]提出了 KD 树(多维二叉搜索树),KD 树是每个点都为  $K$  维点的二叉树,其在点的查找上有着与二叉树一样良好的性能.文献[20]提出了 KD 树的改进——非同构 KD 树,降低了删除的代价.文献[21]介绍了 R 树这种处理高维空间存储问题的数据结构.文献[22]提出了网格的思想,通过网格划分数据空间,再利用网格地址检索相应磁盘页.这一方法主要用于多维点,对于其他目标并不适用.文献[17]提出了最佳动态 R 树的变种—— $R^*$  树, $R^*$  树和 R 树一样允许矩形的重叠,但在构造算法上  $R^*$  树不仅考虑了索引空间的“面积”,而且还考虑了索引空间的重叠.文献[23]利用 Hilbert 分形曲线,对  $N$  维空间数据进行 1 维排序,进而对树结点进行排序,为获得较高的存储利用率对排序后得到的结点集实施滞后分裂算法,这也形成了 Hilbert R 树算法的核心.文献[24]提出了 R 树的变形树 STR 树.文献[25]提出了一种连续  $K$  最近邻查询算法,将当前参考点进行迭代操作,使当前参考点移动到最近的分割点.

国内研究的空间索引技术起步于 20 世纪 90 年代,文献[26]提出一种 G 树的空间模型和动态网格的操作算法.借鉴 Windows NT 等操作系统采用页面来管理虚拟内存的方式,设计实现了基于页面的新的空间索引机制,有效地解决了  $N$  维空间数据的索引问题.但其结点的空间利用率相比其他索引树而言偏低.文献[27]提出了基于 Hilbert 空间排列的点特征二叉平衡排序树动态索引结构和基于角点回溯的线特征索引结构,为动态内存索引与磁盘索引提供了新的思路,并满足 GIS 数据模型的要求.文献[28]针对现有 R 树索引不支持空间数据多尺度表达的问题,提出了一种用于空间数据多尺度表达的树变形索引结构,该算法在检索多分辨率形式组织的空间数据时效果显著.文献[29]提出了基于线性四叉树结构的 Voronoi 图反向膨胀生成算法.文献[30]提出了基于 MR-Tree 空间索引的 Voronoi 图改进及其并行空间查询方法.

以上描述的空间索引技术及其变体,是针对不同应用、不同类型及不同需求的空间数据对象,在某些领域有其自身的优势,但换另一种数据结构也许在查询效率上就稍逊一筹了.

本文提出的 MSR 算法首先对地图数据划分切片,然后在单一切片内聚类对比,在相邻切片内进行聚类以此保证数据间拥有最大程度的空间邻接性;其次,按照一定范围构建叶子结点类并处理离散值;最后自下而上递归生成 MSR 树.MSR 算法能够很好地处理数据密度高畸变的空间区域并且不会出现磁盘浪费的情况.这是因为 MSR 算法减少了矩形框的重叠率,查找路径不唯一的情况很大程度减少,所以其查询效率较高.通过与当前主流的空间索引技术 STR 算法、STR-网格混合算法(STR\_GRID)及 EGRQ 算法进行实验对比,验证了本文提出的算法的良好性能.

## 2 MSR 算法相关知识

在本节中,首先介绍基于 2 维地理空间信息数据的空间索引机制在构建过程中涉及的概念,然后介绍一种基于 R 树的批量加载算法——STR 算法、一种混合空间索引机制 STR-网格算法及 EGRQ 算法.

### 2.1 基本定义

基于地图数据的空间索引机制主要是针对优化道路网络数据查询而建立的空间索引结构,其中

2 维空间道路网络数据查询主要包括 2 个方式:点查询和区域查询.点查询就是在索引结构中找到距离目标点最近的空间对象,而区域查询是找到给定区域中的所有空间对象,所以区域查询更加复杂.

**定义 1.** 空间数据.空间数据是用来表示空间对象的位置、大小、形状及其分布特征诸多方面信息的数据.它可以是点、线、面以及高维的多面体等,比如地图上常见的建筑物、道路、城市.

**定义 2.** 最小外包矩形<sup>[31]</sup>.最小外包矩形(minimum bounding rectangle, MBR)是指以 2 维坐标表示的若干 2 维形状(例如点、直线、多边形)的最大范围,即以给定的 2 维形状各顶点中的最大横坐标、最小横坐标、最大纵坐标、最小纵坐标定下边界的矩形.最小外包矩形是最小外接框(minimum bounding box, MBB)的 2 维形式.

**定义 3.** R 树<sup>[32]</sup>.R 树是用来做空间数据存储的树状结构,例如给地理位置、矩形和多边形这类多维数据创建索引.

其核心思想是聚合距离相近的结点并在树结构的上一层将其表示为这些结点的最小外包矩形,这个最小外包矩形就成为上一层的一个结点.叶子结点上的每个矩形都代表一个对象<sup>[33]</sup>,结点都是对象的聚合,并且越往上层聚合的对象就越多.也可以把每一层看作是对数据集的近似,叶子结点层是最细粒度的近似,与数据集相似度 100%,越往上层越粗糙.

**定义 4.** 聚类<sup>[34]</sup>.聚类是将数据集中的数据根据各自的特征分成不同的簇,簇间尽可能相异,簇内尽可能相似.

聚类是数据挖掘中的一种无先验条件的无监督分析方法.早期传统的聚类查询算法主要关注于获得精确的聚集统计值<sup>[35]</sup>.

## 2.2 STR 算法的构建过程

STR 算法主要是应用于区域查询而构建的树型索引结构算法,它基于排序思想构建的 R 树可以达到近乎 100% 的空间利用率.STR 算法中假设 2 维空间有  $N$  个对象的最小外包矩形(MBR),每个 R 树结点的最大容量为  $M$ ,STR 算法的基本思想为:按 MBR 中心点的  $X$  坐标对其排序,等分为  $S$  组,并使每一组中包含  $M \times S$  个空间对象,在每一组中用 MBR 中心点的  $y$  坐标进行排序,每  $M$  个 MBR 形成 1 个新的结点,并将新结点的 MBR 作为下一次递归的输入,自下而上生成整棵 R 树.

## 2.3 STR-网格混合算法

STR-网格混合算法首先通过对空间区域进行划分,缩小研究范围,达到提高检索效率的目的.基本思想是通过横竖线条把空间区域划分为若干个相同大小的网格,每个网格代表 1 个存储桶,将落在该网格中的空间对象放入对应的存储桶中.之后根据桶中的空间对象的多少,继续划分网格,直至每个子网格中的空间对象个数满足阈值,最后对每个子网格中的空间对象按照 STR 算法再进行组织索引.当用户进行查询操作时,首先检索出用户查询的空间对象所在网格,然后再通过 STR 算法从网格中定位到所需空间对象的位置.

## 2.4 EGRQ 算法

高效几何范围查询算法 EGRQ,采用安全 KNN 计算、多项式拟合技术以及 R 树索引,来提高空间对象的查询效率.

STR 算法、STR-网格混合算法和 EGRQ 算法都有其固定的缺陷.STR 算法不能很好地处理数据高畸变的空间区域,其查询性能低效.STR-网格混合算法首先通过网格对空间区域进行划分,然后再在每个网格中构造 STR 树.当网格对空间区域进行划分时,将空间区域划分为多个子空间,这种划分方式全凭经验.如果对空间区域划分时出现网格划分过细的情况,则会导致耗费磁盘存储空间和搜索时间过长,磁盘存储冗余大,造成磁盘浪费,内存利用率低;如果网格划分过大,则该方法不能很好地处理数据密度高畸变的空间区域,导致其查询效率降低.本文提出的 MSR 算法能够很好地处理以上问题.

## 3 MSR 算法的设计与实现

在对 2 维地理空间信息数据研究过程中发现,随着空间对象的增多,空间对象的 MBR 之间的重叠现象会不断加大.目前针对 2 维空间对象(点、线、面)的查找方式多以相交、距离等判定方法,这种方法导致空间对象之间的 MBR 重叠数量加大,使查询过程中无用路径的数量变多,从而影响查询效率.本文提出的 MSR 算法从优化空间对象之间的 MBR 重叠数量这一角度出发,根据空间邻接性法则交叉搜索相邻切片中的空间对象,同时使用空间聚类方法修正单维度排序所造成位置分布过于稀疏的问题,以自下而上的方式生成空间索引树型结构.

本文采用的空间对象是全国道路网络数据,

图 1 是北京市道路网络数据的 MBR 的效果展示图。可以看出,空间区域中 MBR 重叠现象严重,如果在数据量庞大的、复杂的、形状多种多样的全国道

路网络数据中查询目标数据,合理、高效的空存储索引结构是简化检索路径、缩短检索时间最有效的方法。

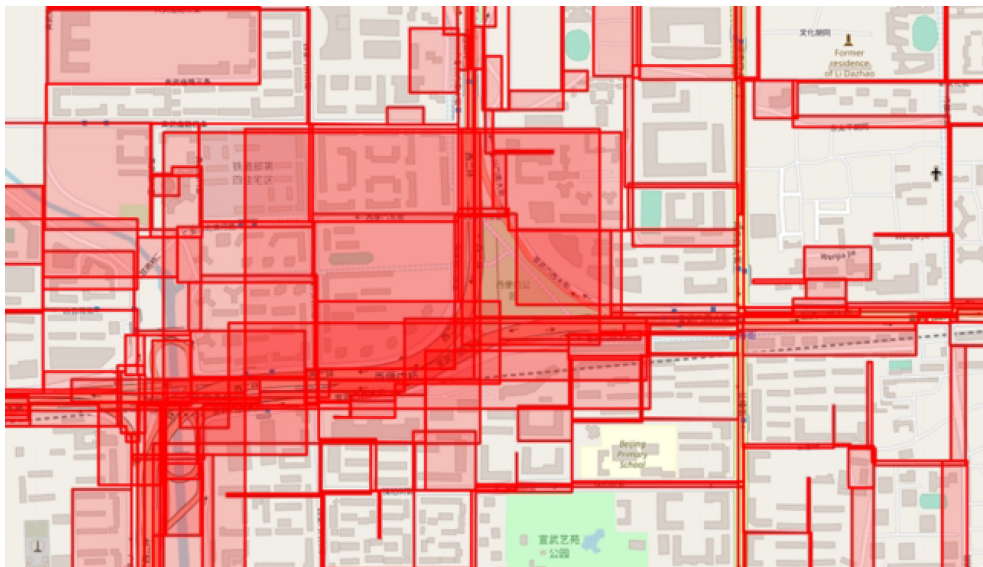


Fig. 1 Road network MBR renderings

图 1 道路网络 MBR 效果图

### 3.1 MSR 算法概述

MSR 结构一共包含 3 种结点:根结点、子结点和叶子结点。根结点和子结点主要存储孩子结点的位置,叶子结点主要存储空间经纬度信息。MSR 算法构造的空间索引树框架图如图 2 所示,其中 2 维地理空间数据在第 1 维度排序且切分后形成的数据集表示为 FD(first division)切片,SD(second division)切片表示 2 维地理空间数据在第 2 维度排序后形成的数据集。经过 FD 切片和 SD 切片操作形成数据集。至此,MSR 构建完成。

下面介绍 MSR 算法的整体流程:

输入:2 维地理空间信息数据中的  $r$  个空间对象,每个结点的最大容量为  $N$ ;

输出:包含  $r$  个空间对象的 MSR 空间索引树。

1) 计算所有 2 维地理空间信息数据的 MBR,对树结点的各项信息做初始化操作;

2) 按照 2 维地理空间信息数据的第一维度进行排序并切分。假设数据个数为  $r$ ,每个结点的最大容量为  $N$ ,对排序好的 2 维地理空间信息数据切分为  $\sqrt{r/N}$  份,保证每个切片中至少包含  $\sqrt{r/N} \times N$  个空间对象;

3) 对每个切片中空间对象按照第 2 维度进行排序;

4) 按照排序后的结果寻找一个未聚类的对象作为初始化类  $C$ ;

5) 考虑到空间对象的邻接性和空间对象的分布情况,根据初始类的 MBR 中心点位置向四周一定的范围内设定搜索框  $S$ ;

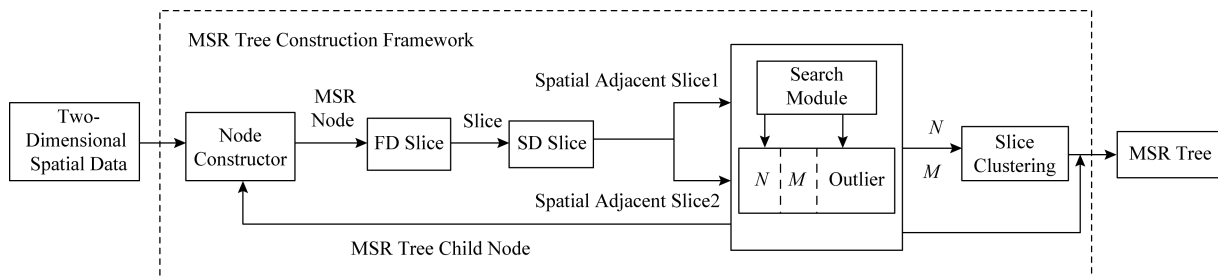


Fig. 2 MSR algorithm framework

图 2 MSR 算法框架

6) 计算搜索框  $S$  包含的当前切片和相邻切片中所有的空间对象, 得出命中对象列表  $H$ , 其长度为  $L$ ;

7) 如果命中个数  $L \geq N - 1$ , 在命中列表中寻找距离类中心最近的空间对象, 将结果归到类中, 更新类大小和类中心, 知道满足结点容量;

8) 如果命中个数  $L < N/2 - 1$ , 则定义为离散值, 在其切片聚类完成后逐一插入到距离最近的类中, 如果命中个数  $N/2 < L < N - 1$ , 则自聚为一类;

9) 按照步骤 4)~8), 每 2 个切片为 1 组, 遍历全部切片, 最后 1 个切片也按照上述原则构造;

10) 按照步骤 1)~9), 自下至上, 递归构建, 最终生成根结点. 至此, 整棵 MSR 树构造完成.

图 3 展示了 MSR 树的生成过程实例, 对空间区域中编号为 1~10 的空间对象求出其 MBR. 然后进行第 1 维数据排序和切分操作, 其结果为: 1, 4, 2, 6, 7, 5 和 8, 9, 3, 10, 再对每个切片进行第 2 维排序操作, 其结果为: 1, 2, 5, 4, 7, 6 和 3, 9, 8, 10. 最后按照 MSR 算法取 2 个切片中的数据聚类为矩形框 11, 12, 13, 14 和 15, 16, 这样第 1 层叶子结点就生成了, 接下来就是不断地执行递归操作, 直到生成根结点.

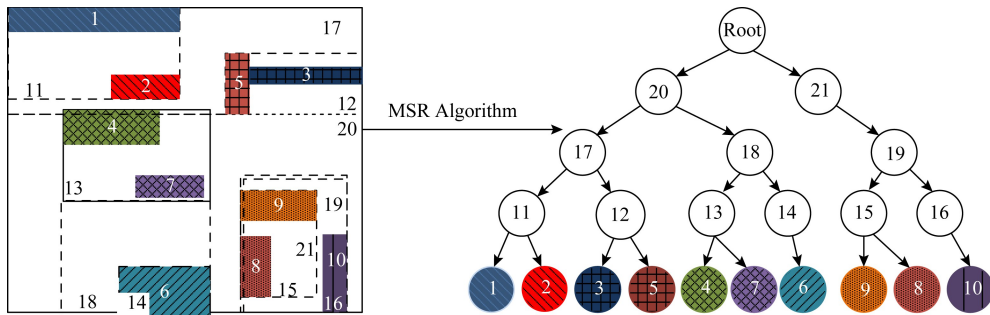


Fig. 3 MSR tree generation process

图 3 MSR 树生成过程

### 3.2 MSR 算法描述

在本节中, 我们给出 MSR 构造算法生成的伪代码, MSR 构造算法的算法描述如下所示:

**算法 1.** 多层切片递归算法.

输入: 当前切片  $Tile_c$  和相邻切片  $Tile_a$ , 搜索框  $B$ ;

输出: MSR 空间索引树;

过程:  $MSR(Tile_c, Tile_a, B) \rightarrow MSR$  空间索引树.

①  $GET\_search\_List(Tile_c, Tile_a, B) \rightarrow H_{list}$ ;  
/\* 根据搜索框  $B$  获取 2 个切片中的命中数据 \*/

② Let  $cluster \leftarrow C$ ; /\* 将当前切片的首个未聚类的数据初始化为新的类 \*/

③ if  $H_{list}.length \geq N - 1$  /\* 根据命中数据的大小分别处理 \*/

$GET\_clustering(H_{list}, C)$ ;  
/\* 聚类操作 \*/

else if  $N/2 - 1 \leq H_{list}.length \leq N - 1$   
 $Cluster \leftarrow H_{list}$ ;

else  $Outlier_{list}$  append  $H_{list}$ ;

④ End if

⑤  $MSRinsert(Nodelist, Outlier_{list}) \rightarrow Nodelist$ ;

/\* 在当前切片完成聚类后, 将离散值插入到距离最近的类中 \*/

⑥  $TreeNodeConstructor(Nodelist) \rightarrow Nodelist$ ;

/\* 将聚类结果构造上层树结点 \*/

⑦ if  $Nodelist.length > 1$  /\* 执行递归操作, 直到生成根结点 \*/

Let  $Tile_c \leftarrow Tile_a$ ;

Let  $Tile_a \leftarrow Tile_a.next$ ;

$MSR(Tile_c, Tile_a, B) \rightarrow MSRTree$ ;

⑧ End if

⑨  $Output(Nodelist)$ . /\* 输出整棵树 \*/

在 MSR 构造算法中, 搜索框是以类的矩形框向四周以一定距离为半径进行搜索的空间范围, 同时每次聚类操作结束后都要把已完成聚类的空间对象进行标记, 以免重复利用. 在聚类过程内部, 要注意每次添加数据后需要实时更新类的大小和中心点, 对于命中空间对象个数的使用规则是参考了  $R$  树结点容量的性质, 即每个结点包含  $[m, N]$  个数据, 其中一般  $m = N/2$ .

### 3.3 MSR 树的更新和检索

MSR 树的查找和 STR 类似, 自下而上逐渐将下层的数据矩形抽象到高层的正方形, 即对下层数据

矩形有正方形化的趋势.检索算法从树的根结点开始,它将所有与查询区域存在交叠的结点的子结点查出,然后将这些子结点作为当前结点检查它们与查询区域是否存在交叠,直到找到叶子结点的空间对象.MSR算法通过判断MBR与查询窗口是否相交,如果相交则遍历所有孩子结点继续判断,直至获取到叶子结点的空间对象信息.

MSR树的更新和R树类似,对于搜索路径上的每个结点,遍历其中的最小外包矩形(MBR),如果搜索到1个叶子结点,则比较搜索框与它的最小外包矩形和数据,如果该叶子结点在搜索框中,则把叶子结点加入搜索结果.添加操作如果遇到结点溢出则分裂,分裂的方法和R树相同,都是更新相关的一条路径上的信息,并且为了尽可能少出现结点溢出的现象,在进行实验时,会为每个结点设置多余空间作为缓冲区.

## 4 实验与结果

在本节中,通过使用本文构建的一种针对2维地理空间信息数据进行多层切片递归的空间索引树构造算法MSR,然后再利用不同量级以及不同城市的地图数据测试MSR算法性能,并且与传统的STR算法<sup>[36]</sup>、STR-网格混合索引(STR\_GRID)算法<sup>[12]</sup>以及EGRQ算法<sup>[37]</sup>进行实验对比.在硬件方面,实验设备为联想Y700-15ISK电脑,搭载的是Windows10操作系统,基本配置为i7-6700HQ 4核处理器,处理速度为主频2.6 GHz超频至3.2 GHz,256 GB固态硬盘+1 TB机械硬盘,RAM的容量为16 GB.

### 4.1 数据集和算法性能度量

本文分别使用北京市道路网数据、天津市道路网数据和全国道路网数据构造MSR算法、STR算法、STR-网格混合索引算法(STR\_GRID)以及EGRQ算法这4种算法的空间索引结构,以区域查询耗时作为衡量算法优劣的指标,详细的道路网络数据信息如表1所示:

Table 1 The Data of Experiment

表1 实验数据

Name of Data	Type of Data	Amount
Beijing Road Network Data	Line	84 543
Tianjin Road Network Data	Line	61 616
National Road Network Data	Line	2 668 904

通过表1可以得到,北京市的道路数量为84 543条,全国路网数据量为2 668 904条,其数据量非常巨大,天津市的道路数量最小.图4展示了北京市的道路网络图.城市道路具有鲜明的特点,其形状多种多样,数据由2维的经纬度坐标组成.图4以北京市的城市交通路网数据为例,绘制北京市道路网络图,其中颜色较深(蓝色)的道路为双向路,颜色较浅(红色)的为单向路,效果如图4所示:

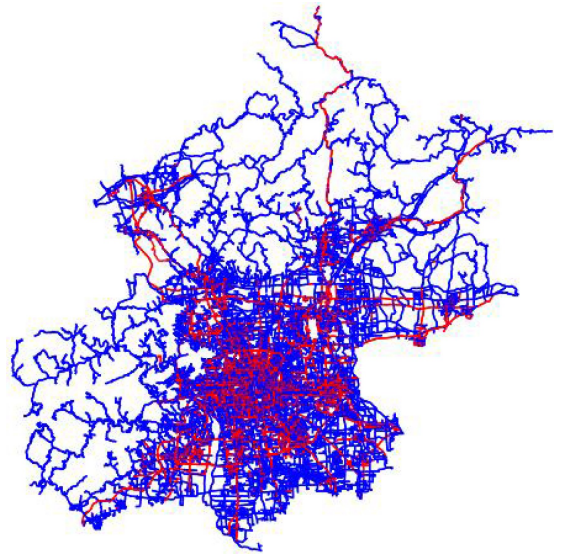


Fig. 4 Beijing road network renderings

图4 北京市道路网络效果图

### 4.2 实验和结果

在本节中,首先通过使用MSR空间索引结构,对生成树的过程中叶结点的矩形框进行效果展示,如图5所示,道路网络数据为北京市道路网络数据,图5中展示了2种算法的差异,MSR算法构造的矩形框多为正方形,而不是STR算法构造的长方形.

其次,通过实现传统的STR算法、STR-网格混合算法(STR\_GRID)以及EGRQ算法作为对比实验用来验证MSR算法的优越性能,其中EGRQ算法的结果是以2为底取对数后的实验结果.本文分别从查询窗口大小、树的层次以及查询个数来测试3种算法的性能,其中树的层次即为树的层数,指的是每层结点中MBR的重叠数量.本实验中,在市级规模的道路网数据中以0.1的差值作为经纬度网格单元的划分标准,在全国道路网数据中以1的差值作为划分标准.

图6所示实验数据为北京市道路网络数据,当查询数目为5 000时,EGRQ算法查询性能最差,查询时间取对数后的结果为11 s;STR\_GRID的查询



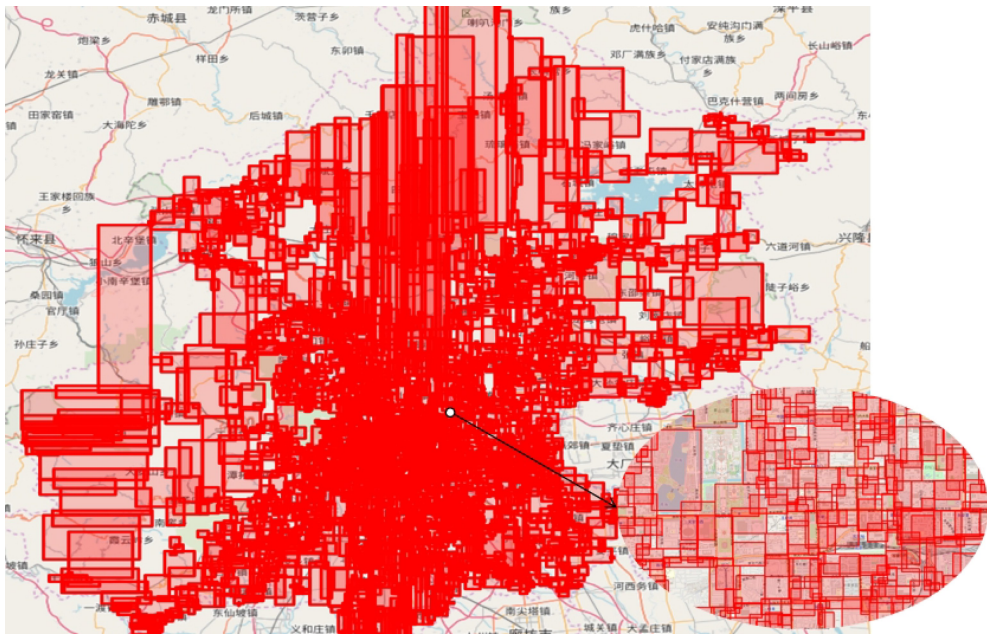


Fig. 5 Effect of MSR on the lower node of Beijing road network data

图5 MSR算法在北京市道路网数据下叶结点效果展示

时间为 10.3 s;其次是传统的 STR 算法,其查询时间为 10 s;MSR 算法性能最优,其查询时间为 9.4 s.

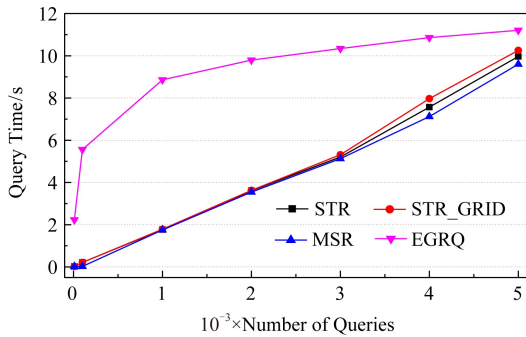


Fig. 6 Comparison of query results under Beijing road network data

图6 北京市道路网数据下的查询结果比较

图 7 所示实验数据是天津市道路网络数据,虽然 STR\_GRID 的查询性能略优于 MSR 算法,但是 2 种算法的查询性能差异并不是很明显.出现这种现象的原因如下:STR\_GRID 是将所有的空间对象划分到各个网格中,每次构建 STR 树时,只是构建每个网格中包含的空间对象,所以其查询时间相对较低.MSR 算法并没有进行划分网格的操作,在构建 STR 树时,面向的是所有的空间对象,所以其查询时间相对较高,但这种现象是很少出现的,尤其是面对大量的空间对象时,MSR 算法性能优于 STR\_GRID.并且 STR-网格算法有着固有的缺陷,该算法

首先通过网格对空间区域进行划分,然后再在每个网格中构造 STR 树.当网格对空间区域进行划分时,将空间区域划分为多个子空间,这种划分方式全凭经验.如果对空间区域划分时,出现网格划分过细的情况,则会导致耗费磁盘存储空间和搜索时间过长,磁盘存储冗余大,造成磁盘浪费,内存利用率低;如果网格划分过细,则该方法不能很好地处理数据密度高畸变的区域.而 MSR 算法能够很好地处理数据密度高畸变的区域并且不会出现磁盘浪费的情况.并且 MSR 算法的查询性能优于传统的 STR 算法和 EGRQ 算法.所以,综上所述,MSR 算法具有最优的查询及存储性能.

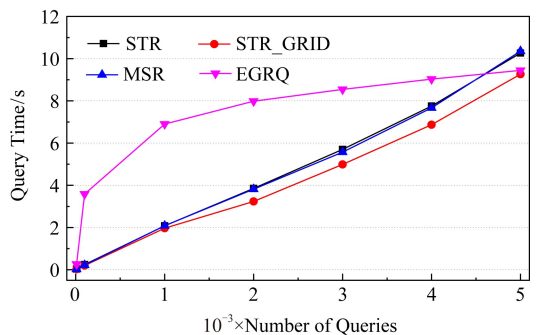


Fig. 7 Comparison of query results under Tianjin road network data

图7 天津市道路网数据下的查询结果比较

通过图 6 和图 7 可知,除了 EGRQ 算法的查询

性能差异明显以外,其他 3 种算法在地市级的道路网络数据中的查询效率相差不大,3 种算法的折线图走势差异很平缓,尤其是当查询数目为 1 000 时,3 种算法的查询时间基本相当,所以猜想是道路数据量导致查询算法性能差异不明显的现象。

将 4 种算法运用到全国道路网络数据中,通过增加构造空间索引结构的数据量,得到的索引结构的查询结果图如图 8 所示:

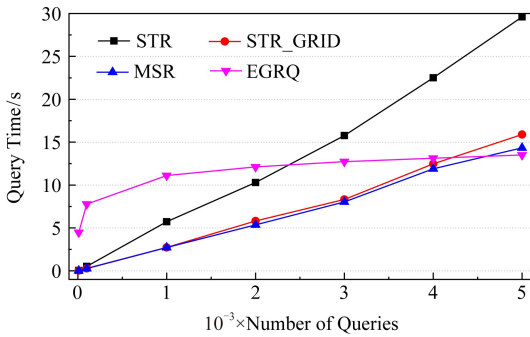


Fig. 8 Comparison of query results under national road network data

图 8 全国道路网数据下的查询结果比较

图 8 所示的查询结果验证了猜想的正确性,空间对象数据量的增大对空间索引的性能要求更加苛刻.从图 8 中可以看到同为纯树型结构的 STR 算法和 MSR 算法查询效果差距较大.当查询数目为 5 000 时,EGRQ 算法通过以 2 为底取对数后的查询时间为 13 s;STR 算法的查询时间为 29.9 s,STR\_GRID 算法的查询时间为 16.2 s,MSR 算法的查询时间为 14 s.在这 4 种算法中,MSR 算法的查询时间最小,EGRQ 算法查询时间最大.图 8 展示的实验结果再次验证了本文所提出的 MSR 算法在查询性能及存储性能的优越性。

图 8 的实验结果中,STR\_GRID 查询效果比 STR 算法效果好很多.为了找到这种现象出现的原因,本文分别统计了北京市、天津市及全国的道路网络数据在 STR 算法中每层树结点中的最小外包矩形框重叠个数,如图 9 所示。

从图 9 中可以很明显地发现,地市级的道路网络数据和全国道路网络数据在 STR 树型空间索引结构的构造过程中矩形框的重叠量存在巨大差异,尤其是在树的第 1 层,STR 算法的 MBR 重叠量为 108137E6,但是本文提出的 MSR 算法的 MBR 重叠量为 700 123,这种差异几乎是指数级别的差距.这种现象出现的原因归结于数据结构的构建原理,这

是不可避免的现象,这也是学术界在数据结构的构建问题上,体现出的一种技术瓶颈.这是 2 种数据的查询时间相差很大的根本原因。

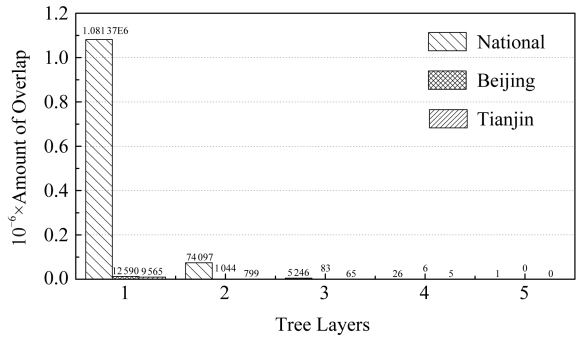


Fig. 9 STR algorithm overlay map

图 9 STR 算法的重叠量展示图

最后在实验的基础上,又分别统计 STR 算法和 MSR 算法在全国道路网数据下,树型空间索引结构在树构造过程中每层树结点中的最小外包矩形框重叠个数,结果如图 10 所示:

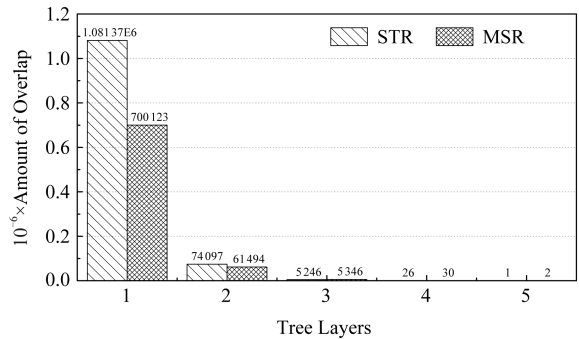


Fig. 10 STR tree and MSR tree layer overlay display

图 10 STR 树和 MSR 树各层重叠量展示图

如图 10 所示,全国道路网络数据量巨大,通过对全国道路网络数据构建空间索引结构时,树结构每一层中,STR 算法和 MSR 算法首先都根据空间对象的最小外包矩形构建树型结构.很明显地看出,树结构第 1 层的最小外包矩形重叠数据差距巨大,如图 10 所示 MSR 算法的最小外接矩形框重叠数量比 STR 算法少了近 30 多万个.当进行检索操作时,由于检索过程中存在检索路径不唯一的情况,从而导致检索效率很低.本文提出的 MSR 算法能够很大程度上减少最小外包矩形的重叠数量,而最小外包矩形的重叠数量,严重影响查询时间,MSR 算法能够很好地解决这种现象,这是算法执行性能优越性的体现。

然而由于 STR\_GRID 和 EGRQ 算法并不是纯树型的索引结构,所以图 10 展示了 STR 算法和 MSR 算法的 MBR 的重叠数量.STR\_GRID 在划分网格这一步骤中,由于其算法在空间区域内对道路网络数据进行切分,通过网格拆分庞大的道路网络数据,这种方法减少了每个网格中道路的数据量,因此 STR\_GRID 也比 STR 算法的查询效率高.但是由于网格索引固有的弊端,比如网格的大小会影响到索引表的大小,如果网格太小,索引就会膨胀.网格过小不能很好地处理数据高畸变的区域,不但查询效率低,而且浪费内存空间.本文提出的 MSR 算法能够很好地处理数据密度高畸变的区域并且不会出现磁盘浪费的情况.

本节在这 3 种不同的影响因素即数据规模、构建树的层数以及查询数目前提下,通过对这 4 种算法即 STR 算法、STR\_GRID、EGRQ 算法以及 MSR 算法进行实验对比,结果均显示 MSR 算法不仅减少了最小外包矩形框的重叠数量,提高了查询效率,而且又可以很好地利用内存空间,减少存储空间的浪费.

## 5 总 结

本文提出了一种对 2 维地理空间信息数据进行多层切片递归的空间索引树构造算法 MSR.首先对地图数据划分切片,然后在单一切片内聚类对比,在相邻切片内进行聚类以此保证数据间拥有最大程度的空间邻接性;其次,按照一定范围构建叶子结点类并处理离散值;最后自下而上递归生成 MSR 树.实验结果表明,MSR 算法不仅减少了最小外包矩形框的重叠数量,提高了查询效率,而且又可以很好地利用内存空间,减少存储空间的浪费,优于目前主流的空间索引算法 STR 算法、STR\_GRID 及 EGRQ 算法.

## 参 考 文 献

[1] Huang Yaohui, Fan Wentao, Liu Liuyang. Road network update and track fusion using trajectory data [J]. Bulletin of Surveying and Mapping, 2018(8): 119-123 (in Chinese)  
(黄窈蕙, 范文涛, 刘柳杨. 利用轨迹数据进行道路网更新及轨迹融合[J]. 测绘通报, 2018(8): 119-123)

[2] Wang Jingyuan, Li Chao, Xiong Zhang, et al. Review of data-centric smart city research [J]. Journal of Computer Research and Development, 2014, 51(2): 239-259 (in Chinese)

(王静远, 李超, 熊璋, 等. 以数据为中心的智慧城市研究综述[J]. 计算机研究与发展, 2014, 51(2): 239-259)

[3] Motieyan H, Mesgari M S. A novel spatial index using spatial analyses and hierarchical fuzzy expert system for obtaining green TOD: A case study in Tehran city [J]. Geocarto International, 2019, 34(1): 1-22

[4] Lu Feng, Zhang Hengcai. Big data and generalized GIS [J]. Geomatics and Information Science of Wuhan University, 2014, 39(6): 645-654 (in Chinese)  
(陆锋, 张恒才. 大数据与广义 GIS [J]. 武汉大学学报: 信息科学版, 2014, 39(6): 645-654)

[5] Tang Jie, Liu Shaoshan. High-precision map service for driverless edges [J]. ZTE Technology Journal, 2019, 25(3): 58-67, 81 (in Chinese)  
(唐洁, 刘少山. 面向无人驾驶的边缘高精地图服务[J]. 中兴通讯技术, 2019, 25(3): 58-67, 81)

[6] Zuo Pengfei, Hua Yu, Xie Xinfeng, et al. Secure encryption method for deep learning accelerator [J]. Journal of Computer Research and Development, 2019, 56(6): 1161-1169 (in Chinese)  
(左鹏飞, 华宇, 谢新锋, 等. 面向深度学习加速器的安全加密方法[J]. 计算机研究与发展, 2019, 56(6): 1161-1169)

[7] Zhang Chao. Research on 100 billion-class intelligent transportation big data storage and retrieval system [D]. Hangzhou: Hangzhou University of Electronic Science and Technology, 2017 (in Chinese)  
(章超. 千亿级智能交通大数据存储与检索系统的研究[D]. 杭州: 杭州电子科技大学, 2017)

[8] Li Guoqing, Huang Zhenchun. Remote sensing big data infrastructure: Integration, management and on-demand services [J]. Journal of Computer Research and Development, 2017, 54(2): 267-283 (in Chinese)  
(李国庆, 黄震春. 遥感大数据的基础设施: 集成、管理与按需服务[J]. 计算机研究与发展, 2017, 54(2): 267-283)

[9] Xu Guanhua. The whole society should pay close attention to the "digital earth" [J]. Chinese Surveying and Mapping, 1999 (3): 6-7 (in Chinese)  
(徐冠华. 全社会要高度关注“数字地球”[J]. 中国测绘, 1999 (3): 6-7)

[10] Li Deren, Shao Zhenfeng. On the era of new geographic information [J]. Scientia Sinica Informationis, 2009, 39(6): 579-587 (in Chinese)  
(李德仁, 邵振峰. 论新地理信息时代[J]. 中国科学: 信息科学, 2009, 39(6): 579-587)

[11] Li Jianhui, Shen Zhihong, Meng Xiaofeng. Scientific big data management: Concepts, technologies and systems [J]. Journal of Computer Research and Development, 2017, 54 (2): 235-247 (in Chinese)  
(黎建辉, 沈志宏, 孟小峰. 科学大数据管理: 概念、技术与系统[J]. 计算机研究与发展, 2017, 54(2): 235-247)

[12] Huang Zhi. Research on hybrid index based on multi-level grid and STR tree [D]. Hang Zhou: Zhejiang University, 2013 (in Chinese)

- (黄志. 基于多级格网与 STR 树的混合索引研究[D]. 杭州: 浙江大学, 2013)
- [13] Li Xiangnan, Zhang Guangyan, Li Qiang, et al. A review of research on solid state disk array construction methods [J]. *Journal of Computer Research and Development*, 2016, 53(9): 1893-1905 (in Chinese)  
(李祥楠, 张广艳, 李强, 等. 固态硬盘阵列构建方法研究综述[J]. *计算机研究与发展*, 2016, 53(9): 1893-1905)
- [14] Zhao Huihui, Zhao Fan, Chen Renhai, et al. MDR tree [OL]. [2019-08-16]. <https://github.com/wangzi6578/MDR>
- [15] Liu Jiping, Zhang Fuhao, Xu Shenghua. A summary of the research progress of big data in government geospatial [J]. *Acta Geodaetica et Cartographica Sinica*, 2017, 46(10): 1678-1687 (in Chinese)  
(刘纪平, 张福浩, 徐胜华. 政务地理空间大数据研究进展综述[J]. *测绘学报*, 2017, 46(10): 1678-1687)
- [16] Lu Youyou, Shu Jiwu. Flash memory system overview [J]. *Journal of Computer Research and Development*, 2013, 50(1): 49-59 (in Chinese)  
(陆游游, 舒继武. 闪存存储系统综述[J]. *计算机研究与发展*, 2013, 50(1): 49-59)
- [17] Beckmann N, Kriegel H P, Schneider R, et al. The R\*-Tree: An efficient and robust access method for points and rectangles [C] //Proc of the ACM SIGMOD Record. New York: ACM, 1990: 322-331
- [18] Finkel R A, Bentley J L. Quad trees a data structure for retrieval on composite keys [J]. *Acta Informatica*, 1974, 4(1): 1-9
- [19] Bentley J L. Multidimensional binary search trees used for associative searching [J]. *Communications of the ACM*, 1975, 18(9): 509-517
- [20] Bentley J L. Multidimensional binary search trees in database applications [J]. *IEEE Transactions on Software Engineering*, 1979, 5(4): 333-340
- [21] Guttman A. R-Trees: A Dynamic Index Structure for Spatial Searching [M]. New York: ACM, 1984: 47-57
- [22] Nievergelt J. Computing with geometric objects [J]. *ACM SIGSAM Bulletin*, 1984, 18(2): 19-19
- [23] Kamel I, Faloutsos C. Hilbert R-Tree: An improved R-Tree using fractals [R/OL]. 1993 [2019-11-13]. <http://drum.lib.umd.edu>
- [24] Leutenegger S T, Lopez M A, Edgington J. STR: A simple and efficient algorithm for R-Tree packing [C] //Proc of the 13th Int Conf on Data Engineering. Piscataway, NJ: IEEE, 1997: 497-506
- [25] Zhang M, Rishe N, Weitong L I U, et al. Efficient progressive continuous  $k$ -nearest neighbor query algorithm for moving objects with a tree-like index: US, 10/331,753 [P]. 2019-06-25
- [26] Cao Jiaheng, Zhang Jian, Tan Hui, et al. A new mechanism for spatial indexing—G Tree [J]. *Geomatics and Information Science of Wuhan University*, 1998, 44(1): 50-53 (in Chinese)  
(曹加恒, 张剑, 谭辉, 等. 空间索引的新机制——G 树[J]. *武汉大学学报: 自然科学版*, 1998, 44(1): 50-53)
- [27] Lu Feng, Zhou Chenghu. A GIS spatial indexing method based on Hilbert arrangement code [J]. *Journal of Computer-Aided Design & Computer Graphics*, 2001, 13(5): 424-429 (in Chinese)  
(陆锋, 周成虎. 一种基于 Hilbert 排列码的 GIS 空间索引方法[J]. *计算机辅助设计与图形学学报*, 2001, 13(5): 424-429)
- [28] Deng Hongyan, Wu Fang, Zhai Renjian, et al. An R-Tree index structure for multi-scale representation of spatial data [J]. *Chinese Journal of Computers*, 2009, 32(1): 177-184 (in Chinese)  
(邓红艳, 武芳, 翟仁健, 等. 一种用于空间数据多尺度表达的 R 树索引结构[J]. *计算机学报*, 2009, 32(1): 177-184)
- [29] Li Jiatian, Chen Jun, Zhao Renliang, et al. Voronoi diagram inverse expansion generation method based on linear quadTree structure [J]. *Acta Geodaetica et Cartographica Sinica*, 2008, 37(2): 236-242 (in Chinese)  
(李佳田, 陈军, 赵仁亮, 等. 基于线性四叉树结构的 Voronoi 图反向膨胀生成方法[J]. *测绘学报*, 2008, 37(2): 236-242)
- [30] Fu Zhongliang, Liu Siyuan. Improvement of Voronoi diagram of MR-Tree spatial index and its parallel space query method [J]. *Geomatics and Information Science of Wuhan University*, 2012, 37(12): 1490-1494 (in Chinese)  
(付仲良, 刘思源. MR-Tree 空间索引的 Voronoi 图改进及其并行空间查询方法[J]. *武汉大学学报: 信息科学版*, 2012, 37(12): 1490-1494)
- [31] Zhu Qijun, Hu Haibo, Xu Cheng, et al. Geo-social group queries with minimum acquaintance constraints [J]. *The International Journal on Very Large Data Bases*, 2017, 26(5): 709-727
- [32] Hadjieleftheriou M, Manolopoulos Y, Theodoridis Y, et al. R-Trees: A dynamic index structure for spatial searching [J/OL]. *Encyclopedia of GIS*, 2017 [2019-11-13]. <http://link.springer.xilesou.top>
- [33] Yu Liang, Su Dan. Discussion on spatial database indexing technology under R tree [J]. *Practical Electronics*, 2016(22): 36-36 (in Chinese)  
(于亮, 苏丹. R 树下的空间数据库索引技术探讨[J]. *电子制作*, 2016(22): 36-36)
- [34] Wan Jing, Zheng Longjun, He Yunbin, et al. Subspace clustering algorithm for high dimensional uncertain data [J/OL]. *Journal of Computer Applications*, 2019 [2019-11-13]. <http://kns.cnki.net/KCMS/detail> (in Chinese)  
(万静, 郑龙君, 何云斌, 等. 高维不确定数据的子空间聚类算法[J/OL]. *计算机应用*, 2019 [2019-11-13]. <http://kns.cnki.net/KCMS/detail>)
- [35] Wang Yijie, Li Xiaoyong, Qi Yapei, et al. Research on uncertain data query technology [J]. *Journal of Computer Research and Development*, 2012, 49(7): 1460-1466 (in Chinese)

(王意洁, 李小勇, 祁亚斐, 等. 不确定数据查询技术研究[J]. 计算机研究与发展, 2012, 49(7): 1460-1466)

- [36] Tian Huizhen. A research based on STR R-tree spatial index [D]. Chengdu: University of Electronic Science and Technology, 2012 (in Chinese)

(田慧贞. 一种基于 STR R-tree 空间索引的研究[D]. 成都: 电子科技大学, 2012)

- [37] Xu Guowen, Li Hongwei, Dai Yuanshun, et al. Enabling efficient and geometric range query with access control over encrypted spatial data [J]. IEEE Transactions on Information Forensics and Security, 2018, 14(4): 870-885



**Zhao Huihui**, born in 1993. Master candidate. Her main research interests include map matching and geographic information system (GIS).



**Zhao Fan**, born in 1992. Master candidate. His main research interests include map matching and geographic information system (GIS).



**Chen Renhai**, born in 1987. PhD. Assistant professor. His main research interests include embedded systems, mobile virtualization, and hardware/software codesign.



**Feng Zhiyong**, born in 1965. PhD. Full professor. Senior member of the IEEE Computer Society and ACM. His main research interests include knowledge engineering, service computing, and security software engineering.

## 《计算机研究与发展》征订启事

《计算机研究与发展》(Journal of Computer Research and Development)是中国科学院计算技术研究所和中国计算机学会联合主办、科学出版社出版的学术性刊物,中国计算机学会会刊.主要刊登计算机科学技术领域高水平的学术论文、最新科研成果和重大应用成果.读者对象为从事计算机研究与开发的研究人员、工程技术人员、各大专院校计算机相关专业的师生以及高新企业研发人员等.

《计算机研究与发展》于1958年创刊,是我国第一个计算机刊物,现已成为我国计算机领域权威性的学术期刊之一.并历次被评为我国计算机类核心期刊,多次被评为“中国百种杰出学术期刊”.此外,还被《中国学术期刊文摘》、《中国科学引文索引》、“中国科学引文数据库”、“中国科技论文统计源数据库”、美国工程索引(Ei)检索系统、日本《科学技术文献速报》、俄罗斯《文摘杂志》、英国《科学文摘》(SA)等国内外重要检索机构收录.

国内邮发代号:2-654;国外发行代号:M603

国内统一连续出版物号:CN11-1777/TP

国际标准连续出版物号:ISSN1000-1239

### 联系方式:

100190 北京中关村科学院南路6号《计算机研究与发展》编辑部

电话: +86(10)62620696(兼传真); +86(10)62600350

Email: crad@ict.ac.cn

http://crad.ict.ac.cn