

路网环境下的最近邻查询技术*

鲍金玲^{1,2}, 王斌¹, 杨晓春¹, 朱怀杰¹

¹(东北大学 计算机科学与工程学院, 辽宁 沈阳 110169)

²(白城师范学院 计算机科学学院, 吉林 白城 137000)

通讯作者: 杨晓春, E-mail: yangxc@mail.neu.edu.cn



摘要: 最近邻查询作为基于位置服务的重要支持性技术之一,引起了众多学者的广泛关注和深入研究.相对于欧式空间而言,路网环境下的最近邻查询更贴近人们的生活,有着更重要的研究意义.路网环境下庞大的数据量和复杂的数据结构,使得最近邻查询的操作代价变得非常昂贵,如何有效地提高查询效率,是研究者面临的主要挑战.对路网环境下的最近邻查询技术进行综述,分别从最近邻查询采用的索引结构和查询处理过程对现有路网环境下的最近邻查询方法进行了分析和比较,也介绍了路网环境下最近邻的变体查询技术的研究情况,最后探讨路网上最近邻查询技术未来的研究重点.

关键词: 基于位置的服务;路网;最近邻查询;欧式距离;路网距离

中图法分类号: TP311

中文引用格式: 鲍金玲,王斌,杨晓春,朱怀杰.路网环境下的最近邻查询技术.软件学报,2018,29(3):642-662. <http://www.jos.org.cn/1000-9825/5442.htm>

英文引用格式: Bao JL, Wang B, Yang XC, Zhu HJ. Nearest neighbor query in road networks. Ruan Jian Xue Bao/ Journal of Software, 2018, 29(3): 642-662 (in Chinese). <http://www.jos.org.cn/1000-9825/5442.htm>

Nearest Neighbor Query in Road Networks

BAO Jin-Ling^{1,2}, WANG Bin¹, YANG Xiao-Chun¹, ZHU Huai-Jie¹

¹(School of Computer Science and Engineering, Northeastern University, Shenyang 110169, China)

²(School of Computer Science, Baicheng Normal College, Baicheng 137000, China)

Abstract: Nearest neighbor query, as one of the building blocks of location-based service, has become a hot research topic in recent years. Compared with Euclidean space, road network is a more practical model in real applications; hence, nearest neighbor query in road network has received broader research efforts. In road network, tremendous data are generated along with sophisticated data structure, making nearest neighbor query computationally expensive. This poses a major challenge to spatial database community on its effort to effectively improve the query processing efficiency for nearest neighbor query. This work summarizes existing nearest neighbor query techniques in road network, and conducts analysis and comparison among them, from various perspectives including indexing structure and algorithm implementation. Additionally, several variants of nearest neighbor query are also summarized in this work. Finally, future research focus and trend for nearest neighbor query in road network are discussed.

Key words: location-based service; road network; nearest neighbor query; Euclidean distance; road network distance

* 基金项目: 国家自然科学基金(61532021, 61572122); 中央高校基本科研业务费专项资金(N161606002); 辽宁省“百千万人才工程”经费

Foundation item: National Natural Science Foundation of China (61532021, 61572122); Fundamental Research Funds for the Central Universities (N161606002); Liaoning BaiQianWan Talents Program

本文由基于图结构的大数据分析与管理技术专刊特约编辑林学民教授、杜小勇教授、李翠平教授推荐.

收稿时间: 2017-07-31; 修改时间: 2017-09-05; 采用时间: 2017-11-07; jos 在线出版时间: 2017-12-05

CNKI 网络优先出版: 2017-12-06 15:23:23, <http://kns.cnki.net/kcms/detail/11.2560.TP.20171206.1522.007.html>

随着无线技术的高速发展和移动通信设备的不断普及,基于位置的服务(location-based service,简称 LBS)得到了广泛的应用.其中的最近邻(nearest neighbor,简称 NN)查询作为基于位置服务中最重要的支持性技术之一,引起了学者们的广泛关注.

NN 查询问题起源于 Knuth 在 1973 年提出的邮局问题^[1].该问题可以简单地描述为:已知 N 维空间中集合 P 包含 n 个数据点,在集合 P 中找出一个数据点 p ,使得 p 到查询点 q 的距离最近.而 $k(k \geq 1)$ NN 查询是 NN 查询的一般化形式.如果数据对象子集 $NN(q)$ 中仅含有一个对象,则该查询是 NN 查询;如果数据对象子集 $NN(q)$ 中含有 $k(k > 1)$ 个对象,则是 k NN 查询.

目前,对于 NN 查询的研究主要基于欧式空间和路网环境.欧式空间中的 NN 查询问题的研究已趋成熟^[2-13],现有方法大多采用 R-tree 索引结构,基于分支界定策略的剪枝技术缩小查询空间.其中的 DF(depth-first)算法^[2]和 BF 算法(best-first)算法^[4]是欧式空间 NN 查询技术中的经典算法,为其他 NN 查询技术奠定了一定的理论基础.然而,欧式空间的 NN 查询是理想情况下的研究,在欧式空间,两点之间的距离指直线距离,这一距离仅仅取决于两点的坐标值,而且对象可以处在空间的任何位置.相比较而言,路网环境下 NN 查询的应用更贴近人们的生活,这方面的研究也有着更重要的意义.但在路网环境中,对象的位置和运动都被约束在网络上,对象之间的距离受限于它所处的网络,由路网的连通性决定(如公路交通网等).这就导致路网环境下的 NN 查询空间检索范围更广,路径计算量更大.因此,欧式空间中的解决方法是不能通过简单的修改就应用于路网环境的.

路网环境下庞大的数据量和复杂的数据结构给 NN 查询带来了极大的挑战,主要体现在以下两个方面.

- (1) 有效的剪枝策略.由于搜索空间较大,NN 查询处理的关键在于剪枝效果.如何构造有效的索引结构和高效的剪枝策略是关键,也是研究的难点.空间索引结构的不断改进在一定程度上提高了对空间数据的处理能力,但是由于空间数据的特殊性,空间索引结构仍无法彻底改变高维数据所带来的束缚.每种索引结构都有自己的优缺点,要想找到一种能够适应各种空间数据类型的索引结构是非常困难的;
- (2) 可伸缩性.不同区域的路网规模不同,对象分布密度不同,用户查询的 k 值也会不停地变化,路网本身也会时时更新,一个好的查询方法要具有很强的伸缩性,更好地适应这些变化.

目前,许多大学和研究机构已经对路网环境下的 NN 查询展开了深入的研究,如南加州大学^[14]、马里兰大学^[15,16]和宾西法尼亚州立大学^[17]、奥尔堡大学^[18-20]、香港科技大学^[21]、香港大学^[22,23]和清华大学^[24]等.路网环境下的 NN 查询技术的研究取得了许多有价值的研究成果,然而目前缺少对这些研究成果的归纳和总结.文献[25]中对路网环境下的最近邻查询技术进行了分析和比较,不过涉及到的只有 5 种比较典型的最近邻查询方法:INE^[21],IER(incremental euclidean restriction)^[21],DisBrw^[16],ROAD^[23]和 G-tree^[24]方法.因此,本文主要对路网环境下的 NN 查询技术进行综述.

本文着重讨论路网环境中的 NN 查询技术,分别从路网数据模型、索引结构和查询过程等方面阐述路网环境下的 NN 查询技术的研究进展.第 1 节介绍路网数据的存储模式.第 2 节介绍查询的处理过程,研究路网环境下主要的 NN 查询技术,并对现有常用的 NN 查询技术进行分析和比较.第 3 节探讨 NN 查询的变体查询技术的研究情况.第 4 节总结全文并展望未来的研究重点.

1 路网数据模型

路网数据模型的建立不仅能够真实地描述道路网络的特性,而且还支持各种基于路网环境的查询研究,而掌握路网数据的一些特有性质,是构建路网数据模型和完成查询处理的关键.

1.1 路网数据的特征

普通意义上的数据具有选择性、可靠性、时间性、完备性、详细性和综合性等特点,而路网数据作为一种特殊的空间数据,除了具有一般数据的全部特征以外,还有一些区别于一般数据的特征^[26],主要表现在以下几个方面.

- (1) 数据结构的复杂性和多样性.对于路网数据来说,对象的信息不但包含其位置信息,还包含对象之间的拓扑结构信息,因此在存储的过程中,需要根据具体情况选择合适的数据结构.这就导致存储路网

信息的数据结构更为复杂,形式更为多样;

- (2) 数据的动态性.路网数据经常会发生更新,例如新建了一条道路,或者某处新开了一家酒店等.道路的属性也是动态变化的,例如在交通高峰时段,某条道路行驶的车辆很多,导致道路拥塞,此时,通过该条道路的时间会变长;高峰过后,通过该条道路的时间又会变短.这就要求存储路网信息的数据结构能够适应由插入、删除或者更新等操作所引起的数据变化,进而保证 NN 查询过程中有效的处理这些更新操作;
- (3) 数据的海量性.一个地区的道路可能会有成千上万条,这些道路之间又会有很多的相交点,道路上分布着密密麻麻的兴趣点,再加上各种道路的规则信息等.因此,路网数据的数据量是非常庞大的.一个城市的地理信息系统中的数据可以达到几十 GB,若将视频数据也加在其中,可以达到 TB 的数量级.因此,对于路网数据的存储,不能仅仅依赖于主存;
- (4) 数据的无序性.路网数据除了空间对象的位置信息,还包含了对象的拓扑信息.路网环境下,对象的拓扑关系受限于网络环境的约束,这些信息有助于空间数据的查询和空间分析,但是因为这些信息的限制,无法对空间数据进行线性排序,这就增加了对路网数据一致性和完整性维护的复杂度.

路网数据的复杂性与多样性,再加上数据的海量性和无序性,给查询优化带来了困难.另外,路网对象的拓扑信息,增加了数据编辑和存储的难度以及空间数据一致性和完整性维护的复杂度.

1.2 路网数据模型

通过观察可以发现:道路网络结构是由路段组成,路段和节点是构成路网的最基本元素.节点是道路的起点、终点和道路间的交叉点.实际中,道路网络除了节点和路段集合外,还包含众多的用户兴趣点(points of interest),当然,也有许多如车辆行驶方向等各种限制策略.因为图论的思想比较简单,容易对道路网络进行建模,路网上的空间问题也便于转化为图论问题.所以,目前国内外有关道路网络的建模方法一般采用图论思想建模,然后基于该路网模型提出解决空间位置查询问题的处理方法.

下面本文以图论的方式介绍路网数据模型.

定义 1(路网(road network)). 将路网结构表示为一个有向带权图 $N=(V,E,W)$,其中, V 表示路网中路段间的交叉点集合, E 表示路网中路段集合, W 表示路段对应的权值.路段的权值可以表示该路段的长度,也可以表示通过该路段花费的时间等属性信息.其中,交叉点 $v \in V$ 是指 3 条及以上路段汇集而成的节点.

定义 2(查询点(query point)). 指查询请求在路网中的位置.在传统的欧式空间中,查询点往往给出查询请求的坐标位置;而在路网环境下,坐标位置对于查询来说没有作用,反而会给查询带来额外的代价.

定义 3(查询对象(query object)). 指查询请求的路网中对象,例如加油站、学校等.路网中每一个对象(包括查询点和查询对象)都位于路段边上,查询对象在路网中可用三元组 $o_i=(n_i,n_j,dist)$ 来表示,其中, n_i 和 n_j 表示路段的起点和终点, $dist$ 表示该对象到路段起点的距离.

定义 4(路网距离(road network distance)). 指从查询点到查询对象的最短路径度量.如果查询路径考虑权重(比如道路拥挤程度、时间速度等限制),路网距离为带权重的综合最短路径的度量.

定义 5(最近邻查询(nearest neighbor query)). 给定数据对象集合 P 和一个查询点 q ,最近邻查询就是在集合 P 中找到一个数据对象子集,满足下面条件:

$$NN(q)=\{p \in P | \forall o \in P, dist(q,p) \leq dist(q,o)\} \quad (1)$$

定义 6(k 最近邻查询(k nearest neighbor query)). 给定数据对象集合 P 和一个查询点 q , k 最近邻查询就是在集合 P 中找到 k 个满足公式(1)要求的数据对象子集.

$k(k \geq 1)$ NN 查询是 NN 查询的一般化形式,如果数据对象子集 $NN(q)$ 中仅含有一个对象,则该查询是 NN 查询;如果数据对象子集 $NN(q)$ 中含有多个对象,则该查询是 k NN 查询.

如图 1 所示,图 1(a)为截取的北京市区域路段的路网,图 1(b)是对北京区域路段网络建模后的路网结构.图 1(b)中, $\{v_1,v_2,\dots,v_{14}\}$ 表示路网中的交叉点, $\{p_1,p_2,p_3\}$ 表示查询对象,在此表示为大学.图 1(c)中, q 为查询点, q 发出查询最近的一所大学的请求,此类查询是 NN 查询.此时, q 的查询结果 $R=\{p_3\}$,其中, q 到 p_3 的网络距离= $dist(q,$

$p_3)=dist(q,v_6)+dist(v_6,p_3)$.如果要查询最近的两所大学,则是 kNN 查询,此时 $k=2$,查询结果 $R=\{p_3,p_2\}$.

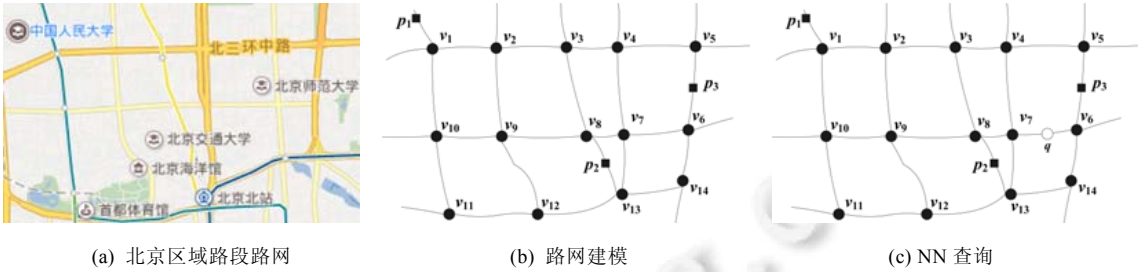


Fig.1 Beijing regional road network modeling

图 1 北京市区域路段网络建模

2 路网环境下的最近邻查询技术的研究

相比较而言,路网环境下的 NN 查询比欧式空间的 NN 查询更贴近人们的生活,这方面的研究也有着更重要的意义.由于路网数据与一般的空间数据的主要区别就是网络的连通性,因此,欧式空间中的 NN 查询方法不能通过简单的修改就直接应用于路网环境.

2.1 最近邻查询过程

路网数据一般复杂且庞大,为了提高查询处理的效率, NN 查询过程通常分为过滤和精炼两步进行,如图 2 所示.

- (1) 过滤:这是 NN 查询技术中与查询效率密切相关的阶段.在这一步骤中,时空对象一般按照某种空间索引结构被近似地以简单的形式进行表示,例如 R 树中将对象表示为最小外包矩形. NN 查询过程在近似的基础上执行过滤步骤,返回一个候选集合,作为完全满足某种要求的所有对象的集合;
- (2) 精炼:这个阶段需要处理的是上一步骤(即过滤)产生的候选集,算法需要依据每个对象的精确几何信息及所提供的精确谓词对候选集进行进一步的检查,确定候选对象是否匹配.

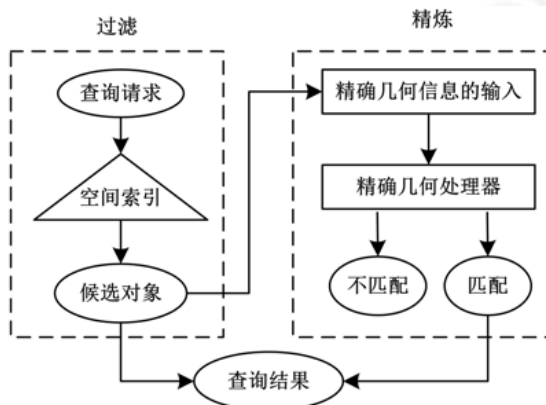


Fig.2 Precess of query

图 2 查询处理过程

2.2 最近邻查询方法

通过对空间数据查询过程的分析可以看出,索引结构和过滤策略直接影响着路网环境下的 NN 查询方法的性能.在路网环境中,对象的位置和运动都被约束在网络上,对象之间的距离受限于它所处的网络,由道路网络

的连通性决定,所以影响路网环境下的 NN 查询方法性能的另外一个因素是 NN 查询过程中所采用的扩展方式.按照查询处理过程中所采用的扩展方式,可以将路网环境下的 NN 查询方法分为 3 类:Dijkstra 式查询方法、启发式查询方法和相邻区域迭代式查询方法.在介绍查询方法之前,先给出一个例子,如图 3 所示, v_1, \dots, v_{12} 是路网中的交叉节点, p_1, p_2, p_3 表示查询对象点, q 是查询点,目标是查找 q 的 k 最近邻点,这个例子将作为下面所有方法的实例.

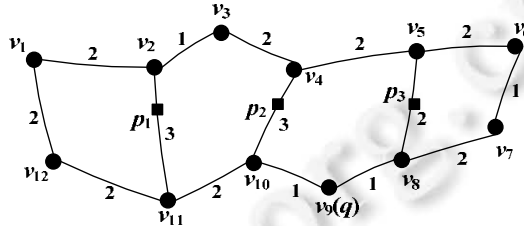


Fig.3 Example of query

图 3 查询实例图

2.2.1 Dijkstra 式查询方法

Dijkstra 算法^[27]是由 Dijkstra 在 1959 年提出的求解图中单源最短路径的经典算法.基本原理是从源点出发,每次新扩展一个到源点距离最近的节点,然后更新与其相邻的节点距离.Dijkstra 式扩展模式解决 kNN 查询问题的基本思想就是从查询点出发,沿着查询点的邻接边向距离查询点最近的节点扩展搜索,然后再更新该节点到其相邻节点的距离,一直找到查询点的 k 个最近邻对象为止.为了提高 kNN 查询的效率,很多 kNN 查询方法需要依靠有效的空间索引结构和剪枝策略.

下面按照不同的空间索引结构,分别探讨基于 Dijkstra 式扩展模式的 kNN 查询方法.

(1) 采用 R-tree 索引结构的查询方法

在路网环境中,R-tree 索引结构主要通过 R-tree 对折线(两个节点的连线)的 MBR 进行索引,以此支持对网络空间属性的查询.以图 3 为例的路网存储结构如图 4 所示,主要包括 3 部分:邻接表组件、边的折线表组件和路网 R-tree 组件.邻接表表示路网的连通性,每个节点记录了它的邻接点的存储位置、与邻接点形成路段的网络距离以及对应的 MBR,还存储了对应路段在折线表中的存储位置.边的折线表存储了网络中每个路段的详细信息,还包括路段对应端点在邻接表中的存储位置.路网 R-tree 组件主要存储每个 MBR 中的边和查询对象信息.

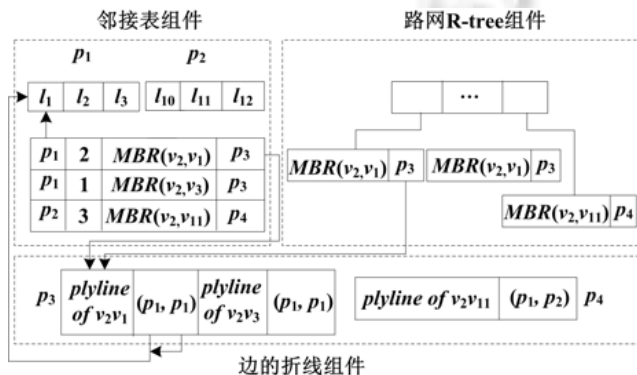


Fig.4 Example of storage schema in Ref.[21]

图 4 文献[21]中的存储模式实例

Papadias 等人^[21]在 2003 年讨论了空间网络数据库环境下的 kNN 查询处理方法,提出了 INE(incremental network expansion)方法,此方法基于 R-tree 索引结构,在路网上多次使用 Dijkstra 算法实现 kNN 查询.首先以查

询点 q 为中心,沿着邻接边逐步进行扩展,在扩展过程中比较所有遇到的对象到查询点的距离,当扩展半径达到查询点 q 的第 k 个对象的距离时,查询结束.

同年,Huang 等人^[19]基于 R-tree 索引结构提出了处理多步骤 kNN 查询算法(MkNN).该算法是对 INE 方法的改进.它采用主存选择性地高速缓存查询结果,然后在下一步的查询中重用缓存的查询结果.基于这一思想提出了 6 种缓存算法,并且通过实验验证了这 6 种缓存算法在不同的路网环境下拥有各自的优势,可以完成不同情况下的多步骤的 kNN 查询.另外,孙亚^[28]在 INE 方法的基础上利用 R-tree 索引结构,提出了一种基于道路网络扩展的 kNN 查询算法 NE(network expansion),可以有效地提高 NN 查询的效率.

INE 方法以及改进的 INE 方法在对一个节点进行扩展的时候,采用的是在该节点的 R 树上进行搜索,并最终返回每个邻边的所有节点.即使此邻边并不包含兴趣点,也要执行搜索,所以它的性能和 Dijkstra 算法一样,需要遍历整个路网.INE 算法的效率取决于待查询对象的分布密度,在查询对象密集的网络中,查询性能较好;而对于较为稀疏的网络,因为搜索空间增大,会执行许多冗余的操作,由此导致算法的性能下降.

(2) 采用 R-tree 与 B-tree 结合的索引结构的查询方法

Imeida 等人^[29]在 2006 年提出了基于 Dijkstra 算法来解决路网环境下的 kNN 查询问题.鉴于道路网络的相对稳定性,采用 R-tree 索引路网节点;而对于查询的对象点,则采用 B-tree 进行索引.基于上述两种索引结构的有效结合,采用改进的 Dijkstra 算法实现增量的 kNN 查询.但是该方法存在一定的局限性,它假设了路网的相对稳定,不适用路网更新的情况.另外,侯士江^[30]也基于 R-tree 与 B-tree 结合的两层存储模式,提出了增量 kNN 查询算法(IkNNQA).

(3) 采用 Voronoi 图与 R-tree 结合的索引结构的查询方法

Voronoi 图与 R-tree 结合的索引结构的主要思想是将整个路网划分为若干个 Voronoi 单元,每个 Voronoi 单元以数据对象 p 为中心,并预先计算每个 Voronoi 单元内部的边界点之间的最短路径距离,将预计算的距离值加以保存,然后通过 R-tree 索引所有的 Voronoi 单元.以图 3 的路网结构为例,Voronoi 的划分情况如图 5(a)所示,存储结构如图 5(b)所示,其中, b_1, b_2, b_3 和 b_4 是划分过程中产生的边界点.

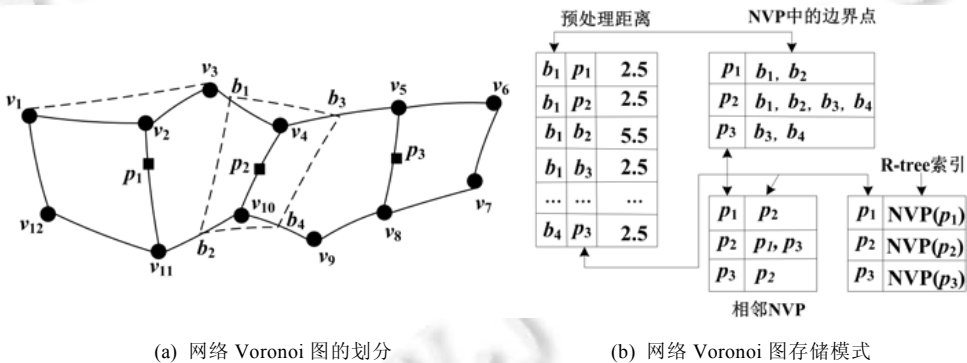


Fig.5 Example of network Voronoi diagram storage schema

图 5 网络 Voronoi 图存储模式实例

Safar^[31]基于 Voronoi 图与 R-tree 结合的索引结构,提出了 PINE(progressive incremental network expansion)方法.PINE 方法的执行过程是:首先定位查询点所在的 Voronoi 单元,确定它的 INN 对象点;然后,利用改进的 Dijkstra 算法扩展到其他的 Voronoi 单元,查询其余的最近邻对象点.PINE 方法是对 VN³方法^[14]的改进,这样可以避免过多的预计算代价和存储代价.但是对于大规模的路网,预计算代价仍然较高;而在 k 值增大时,PINE 方法的查询效率也会明显下降.

(4) 基于中心节点构建索引的查询方法

基于中心节点构建的索引一般以数据点为中心,或者以路网的交通节点(也就是交叉节点)为中心,预先计

算出中心节点周围一定数量的最近邻对象以及节点到最近邻对象的距离,然后存储在相应节点的最近邻列表中.使用这种索引结构处理 k NN 查询的目标就是通过权衡预计算与网络扩展的代价,来平衡 NN 查询与更新的性能.

在 2005 年,Huang 等人^[18]提出了一种 Island 机制处理 k NN 查询.Island 索引结构的特点是:给定的一个距离值 r ,以数据点 dp 为中心、 r 为半径的路网距离区域为 dp 的 Island,所有与 dp 的距离小于 r 的节点都与 dp 在同一个 Island 内.假设 $r=3$,那么以 p_1 为中心的 Island 的划分如图 6(a)所示,存储结构如图 6(b)所示.

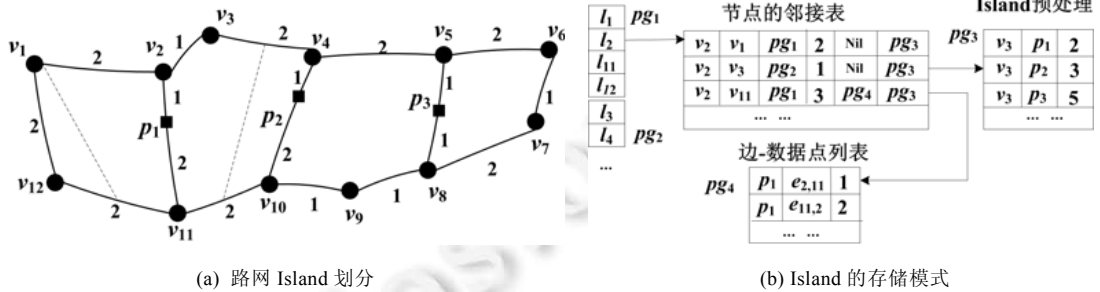


Fig.6 Example of road network Island storage schema

图 6 路网 Island 存储模式实例

首先,预计算到数据点的距离在半径 r 以内的对象点,并存储起来,以此降低查询过程中的计算代价.Island 方法查询过程分两步:首先检查查询点 q 所在的 Island,然后比较 k 与 Island 中的对象个数:如果 q 的 Island 中的对象个数大于 k ,那么距离查询点 q 最近的 k 个对象就是查询结果;否则,沿着 q 所在的 Island,采用 Dijkstra 式扩展模式向其他 Island 搜索.

Island 方法的关键是确定合适的 r 值.当 r 较大时,查询对象的候选集过大,使得预计算的代价较高;而当 r 较小时,会因候选集不足而不得不进行网络扩展,降低了查询的效率.因此,Island 方法在某种程度上依赖于 r 的取值,但是如何确定最佳的 r 值没有一个严格的标准,实施起来比较困难.

CHO 和 CHUNG^[32]在 2005 年提出的 UNICONS(unique continuous search algorithm)算法与 Island 方法非常类似,主要处理路网中 NN 查询和连续 NN 查询问题.它摒弃了 Island 算法中针对查询点进行预计算的策略,而是以路网中的交叉节点为中心,预计算交叉节点的最近邻对象点,得到查询路径上各个交叉节点的最近邻列表,然后通过在线计算得到划分点.

UNICONS 方法的查询过程:首先定位查询点所在的边,然后比较这条边的两个端点(也就是路网上的交叉节点)的最近邻列表中的对象点的数目 m 与 k 值的大小:如果 m 大于 k ,那么选择距离查询点最近的 k 个对象点作为结果;否则,需要通过 Dijkstra 算法向其邻接边扩展查找其他的最近邻对象.

算法中利用预计算的 NN 列表信息来减少当前节点 NN 的计算,但是从某种程度上来说,进一步扩展的最近邻计算会增加查询的时间.该方法的优点是不依赖路网中查询对象的分布密度,因此在密集或者稀疏的网络中都表现出较好的查询性能.

(5) 采用最短路径树索引结构的查询方法

Hu 等人^[23]为了解决路网上 k NN 查询问题,在 2006 年提出了一种利用最短路径树索引网络拓扑的处理方法.该方法将路网生成为内部互联的最短路径树 SPT(shortest path trees).假设数据点位于路网的节点上,并且边与边之间的权重满足三角形不等式,然后将 SPT 转换成 SPIE(shortest path tree with horizontal edges and triangular inequality edges)结构.在这一结构中,同一层的节点之间用快捷边连接,两个节点之间快捷边的权重就是这两个节点之间的最短路径距离.对于 SPIE 结构中的每一个节点,需要存储该节点的后继节点中的最近邻数据点以及该节点到这一最近邻数据点的距离.

SPIE 方法的查询过程:首先,通过 SPIE 索引,在查询点 q 的后继节点中查找最近邻对象,然后再向查询点的

父节点扩展,在其父节点的后继节点中查找 q 的最近邻对象点.如此迭代下去,一直扩展到根节点为止.这一方法可以有效地避免网络扩展产生的代价,但是此方法的性能在很大程度上依赖于已知的对象集合.该方法限制查询点位于数据结点上,但实际上,查询点往往位于网络中的边上;另外,在构建 SPIE 的过程中需要增加节点,这样也就增大了网络的规模,导致预计算代价较高,需要的存储空间较大.

(6) 采用网格索引结构的查询方法

网格索引结构的主要思想是基于哈希网格索引技术,将路网划分成相等的网格,然后通过网格对路网上的节点进行索引.网格间的公共节点称为边界节点.如果一条边的两个节点位于不同的网格中,那么这条边的中点也是边界节点,划分过程如图 7(a)所示.对于划分之后的路网,按照图 7(b)所示的结构存储边的起点和终点,还有同一网格中节点到边界节点的距离信息和同一网格中边界节点之间的距离信息.

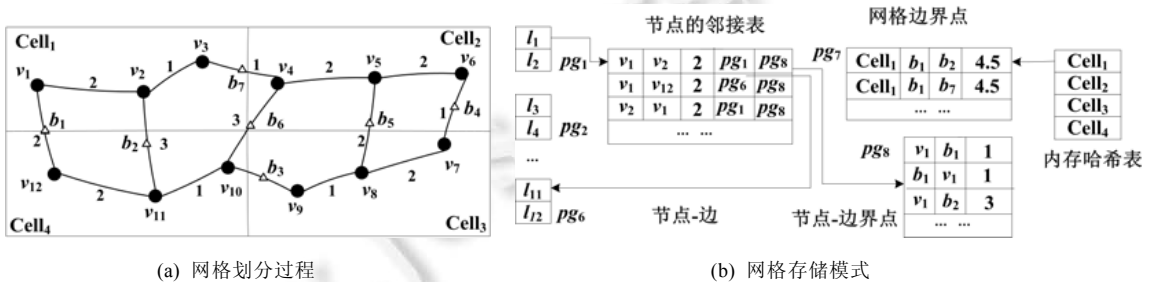


Fig.7 Example of grid storage schema

图 7 网格存储模式实例

2007 年,Huang 等人^[20]提出了 S-GRID 方法处理路网的 k NN 查询问题.该方法也可扩展用于 CkNN 和范围查询,并可以适应道路情况变化的情况.S-GRID 方法的查询过程:先定位查询点所在的网格,接着在网格内查找最近邻的对象点,然后再沿着边界点基于 Dijkstra 式搜索模式向外扩展到其他网格单元查找剩余的最近邻对象点.它是一种为减少磁盘访问、提高查询效率而进行预计算的 k NN 查询方法.

这种网格索引的方式实现起来还是比较简单的,加上高效的编码技术,可以很快地查找到符合条件的对象,但是在网格划分的过程中需要添加额外的边界节点,加大了数据的存储空间.另外,在大规模稀疏的路网中,或者当 k 增大时,该方法的查询效率会明显下降.

(7) 采用 B⁺-tree 索引结构的查询方法

Lee 等人^[17,33]在 2009 年提出了 ROAD 框架,解决路网上的位置查询问题,主要思想就是在分层次的路网结构上进行 Dijkstra 算法的扩展.ROAD 采取路线覆盖和关联目录分别索引路网数据和空间对象.路线覆盖部分是将路网分层进行子图划分(如图 8(a)所示),再利用 B⁺树对路网节点进行索引(如图 8(b)所示),其键值为节点 ID,每一个叶子项指向一个路网节点以及对应的捷径树,非边界节点(比如 v_1)的捷径树只有一项,存储该节点到其邻近节点的边以及长度;而边界节点(比如 v_4)的捷径树按照该点所在区域子网的层次进行构造,该树非叶子节点存同一层包含该节点的区域子网以及每个子网的快捷边,叶子节点存储到它邻近节点的边.关联目录作为对象查找结构,采用 B⁺树对空间对象进行索引,如图 8(c)所示.键值为节点的 ID,叶子节点包含路网节点和区域子网两类:若为路网节点,存储与该节点相邻对象的距离信息;而区域子网节点则存储其对象目录.

利用 ROAD 框架结构执行 NN 查询的过程:首先,从查询点开始扩展,在查询点的邻接子图中如果不包含要查询的对象点,那么沿着快捷边越过该子图;然后,采用 Dijkstra 算法继续向其他的子图扩展.利用 ROAD 框架进行 NN 查询的方法不能有效地过滤掉那些在路网上广泛而均匀分布的对象,ROAD 的性能和 Dijkstra 算法一样,不得不遍历整个路网.

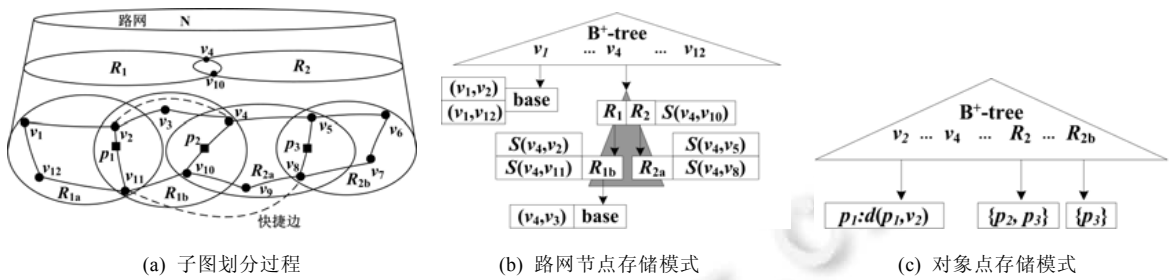


Fig.8 Example of ROAD storage schema

图 8 ROAD 存储模式实例

2.2.2 启发式查询方法

基于启发式扩展模式的 kNN 查询方法主要采用 BF(best-first)算法的思想,按照启发式策略查找最有可能成为 kNN 对象点的候选集,尽量提前结束查找过程.常见的方法是在有效的空间索引结构的基础上,计算路网上的每一个对象点(或者子区域)到查询点的距离下限(可能不是路网距离):如果某个对象点(或者子区域)到查询点的距离下限小于已求得的 kNN 对象点距离,那么这一对象点(或者这一子区域)就是所求候选集,之后再按照距离下限从小到大的顺序计算候选集中的对象点(或者子区域)到查询点的实际路网距离;如果这个距离下限大于已求得的最近邻距离,那么舍弃这一对象点或者这个子区域.在启发式扩展模式的方法中,确定距离下限是非常关键的工作,有效的距离下限可以有效降低搜索代价,提高 kNN 查询效率.

下面针对现有方法中采用的不同空间索引结构,介绍基于启发式扩展模式的 kNN 查询方法.

(1) 采用 R-tree 索引结构的查询方法

Papadias 等人^[21]在 2003 年讨论空间网络数据库环境下的 kNN 查询处理问题时,除了 INE 方法,还提出了 IER 方法.这种方法也是基于 R-tree 完成 kNN 查询的,所采用的存储结构如图 4 所示.IER 方法是一种基于启发式的搜索模式扩展查询最近邻的方法,利用查询点 q 到对象点的欧氏距离作为距离下限完成查询.

IER 方法的查询过程如图 9 所示:首先计算查询点 q 到其欧氏距离最近的对象点 p_2 的路网距离 $d_N(q, p_2)$,然后在距离查询点 q 的半径分别为 $d_E(q, p_2)$ 和 $d_N(q, p_2)$ 的环形区域内查找到查询点 q 的欧氏距离小于 $d_N(q, p_2)$ 的对象点. p_3 是满足条件的对象点,那么计算查询点 q 到点 p_3 的路网距离 $d_N(q, p_3)$.因为 $d_N(q, p_3) < d_N(q, p_2)$,所以接着在距离查询点 q 的半径分别为 $d_E(q, p_3)$ 和 $d_N(q, p_3)$ 的环形区域内查找到查询点 q 的欧氏距离小于 $d_N(q, p_3)$ 的对象点.如此迭代下去,最后得到的路网距离 $d_N(q, p_n)$ 最小的对象点就是距离查询点 q 最近的对象点.

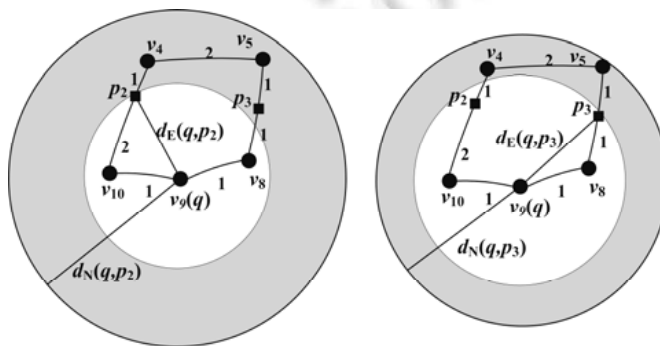


Fig.9 Example of IER

图 9 IER 查询实例

如果查找查询点 q 的 kNN 对象点,方法与查找 NN 对象点的类似,首先选择到查询点 q 的欧氏距离最近的

k 个对象点,然后分别计算这些对象点的路网距离,并对这些对象点按照所求的路网距离升序排序,将其中值最大的 $d_{E\max}=d_N(q,p_k)$ 作为界限,接着计算查询点 q 到第 $k+1$ 个欧式距离近邻点 $p_{(k+1)}$ 的路网距离,如果 $d_N(q,p_k)>d_N(q,p_{(k+1)})$,将 $p_{(k+1)}$ 加入 k 近邻集合,并从 k 近邻集合删除 p_k ,如此迭代下去。

IER 方法中,如果对象点到查询点的欧式距离排序顺序和它们之间的路网距离排序顺序相差不多,那么 IER 方法的查询效率很高;否则,在查找 k 近邻的时候,需要计算几乎所有的对象点到查询点的欧式距离,这样,IER 方法的效率就会大大降低。

(2) 采用 Quadtree 索引结构的查询方法

2005 年,Sankaranarayanan 等人^[15]提出了 SILC(spatially induced linkage cognizance)技术,可以有效解决路网上增量的 kNN 查询问题.2008 年,Samet 等人^[16]基于 SILC 技术提出了 Distance Browsing(简称 DisBrw)方法完成路网上的 kNN 查询问题.该方法对文献[15]中的增量 kNN 查询方法进行了改进,利用查询点到第 k 个最近邻对象的路网距离的估计值作为上限进行剪枝,查询 kNN 对象,而不再是文献[15]中增量的获得 kNN 对象.在具体实现过程中,主要体现在有效减少了优先队列中不必要的插入操作。

SILC 技术是基于路网上的最短路径一致性思想实现的.在路网环境中,较近的两个起始节点到达另外两个较近的目标节点之间的最短路径会共享很多公共的路段,这就是路网中最短路径一致性.对于一个节点 $s \in V$,SILC 技术首先预先计算节点 s 到其他所有节点的最短路径,并对节点 s 的每一个邻接点设置唯一的颜色.每一个节点 $u \in V$ 与节点 s 到节点 u 的最短路径上经过的节点设置成相同的颜色,然后利用区域四叉树对路网进行迭代划分,这样可以将空间复杂度由当时技术的 $O(|V|^3)$ 有效地降低到 $O(|V|^{1.5})$.图 10 说明了节点 v_4 着色编码的过程.SILC 技术可以迅速找到从节点 s 到节点 t 的最短路径.首先,通过节点 s 所在的四叉树可以识别出 t 的颜色,再凭借节点 t 的颜色确定从 s 到 t 的最短路径上的第 1 个节点 v ,然后再通过节点 v 的四叉树,识别从 s 到 t 的最短路径上的下一个节点,如此迭代下去,直到到达节点 t 为止。

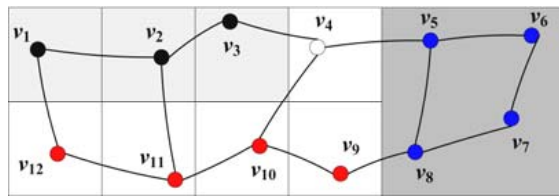


Fig.10 Shortest-Path quadtree of vertex v_4

图 10 节点 v_4 最短路径四叉树

DisBrw 方法使用了一个对象分层的机制去避免计算所有对象的路网距离区间,也就是计算包含对象区域的距离区间.在接受 kNN 查询请求后,DisBrw 方法首先在查询点 $q(v_4)$ 的最短路径四叉树上访问距离查询点最近的包含对象点的子区域,然后再计算查询点到该区域最近的对象点的路网距离,接着,按照同样的策略继续查找其他的子区域.为了实现 kNN 查询,需要在区域四叉树上存储一些额外的信息.对于包含在区域四叉树单元内部的每一个节点 v ,计算四叉树所属的节点 s 到节点 v 的路网距离与他们之间的欧式距离的比值,然后在每一个四叉树单元中存储本单元内的最小比值 λ^- 和最大比值 λ^+ .执行 kNN 查询时,对于遇到的任意节点 t ,通过节点 s 到节点 t 的欧式距离与 t 所在的四叉树单元存储的最小比值 λ^- 和最大比值 λ^+ 相乘,可以得到一个距离区间 $[\delta^-, \delta^+]$.这个距离区间可以作为节点 s 到节点 t 路网距离的下限和上限,帮助剪掉不满足 kNN 查询的对象.在查询的过程中,当得到最短路径中的下一个节点 u 时,修正这一距离区间,此时的距离区间就变为节点 s 到节点 u 的实际路网距离与节点 u 到节点 t 的距离区间的和.如此迭代下去,这一距离区间会很快收敛到一个确定的路网距离值。

文献[16]方法与文献[15]中的方法相比,在存储空间和查询效率方面都有所改进,但他们依然存在着共同的问题:如果在很小的区域聚集了大量的对象,这两种方法就变得效率很低.虽然降低了存储空间,但是预处理代价还是比较大.对于大型公路网,采用这种方法搜索 k 最近邻是不切实际的,尤其是路网数据更新频繁的情况。

(3) 基于距离签名索引结构的查询方法

Hu 等人^[22]于 2006 年提出了一种距离索引技术,叫做 distance signature(简称 Dist Sign).该方法预先计算了所有节点到每个对象的路网距离,然后将这些距离按照长度范围分类,并编码为 distance signatures.如图 11 所示:节点到对象的距离分为 4 类:0~4 算作第 1 类别,4~10 算作第 2 类别,10~16 算作第 3 类别,大于 16 为第 4 类别.用这种距离范围的大小可以表示出节点与对象之间距离的远或近,代替了精确的距离表示,缩减节点与对象距离的存储空间.该方法需要存储每个节点到对象点的距离类别和节点到对象的最短路径中的后继节点,查询点与对象点之间的距离类别就是它们之间的距离界限.

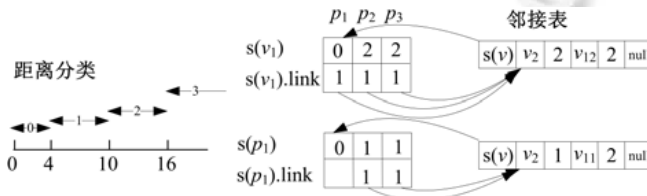


Fig.11 Example of Distance Signature storage schema

图 11 Distance Signature 存储模式实例

Dist Sign 方法查询的过程:计算距离查询点的距离类别序号为 0 的对象点数目,如果这个数目小于 k ,再统计距离类别序号为 1 的对象点数目,依此类推,直到对象点的数目大于或者等于 k 为止;然后再计算查询点到这些对象点的精确距离,最终找到 kNN 对象点.计算查询点到对象点的精确对象距离时,需要访问索引中节点的下一个指针指向的邻接对象节点.虽然此方法采用距离分类压缩了预处理的存储空间,但是需要预处理和存储的代价仍然较大.

(4) 基于 G-Tree 索引结构的查询方法

为了进一步提高 kNN 查询效率,Zhong 等人^[24,34]在 2013 年提出了一种平衡的搜索树 G-Tree 索引结构.首先将整个路网按照层次迭代进行子图划分,如图 12(a)所示;然后构建成 G-Tree 索引结构,如图 12(b)所示.

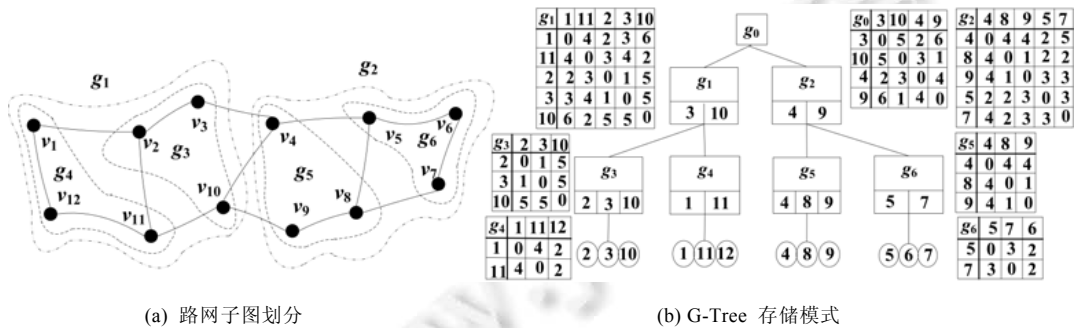


Fig.12 Example of G-Tree storage schema

图 12 G-Tree 存储模式实例

在 G-Tree 中,每一个节点存储着对应的子图信息,包括该子图的边界节点集合和一个预先计算好的距离矩阵.对于非叶子节点,矩阵中存储的是任意两个边界节点之间的最短路径距离;而对于叶子节点,矩阵中的信息就是对应子图中的边界节点与子图中所有节点之间的最短路径距离,并将所有的节点都利用哈希表与叶子节点一一对应.

G-Tree 方法的查询过程:已知查询点 v_q 、对象点集合 C 和 k ,在执行 kNN 查询时,首先利用哈希表确定查询点和对象点所在的叶子节点;接着计算查询点到对象点所在子图的最短路径距离,并将其存入最小优先队列中;然后按照出队顺序计算查询点到对象点的实际距离,如果查询点与子图的最短路径距离大于当前得到的查询

点到对象点的最短路径距离,那么放弃计算查询点到该子图的对象点的最短路径距离。

该方法将空间复杂度降低到 $O(|V|\log|V|)$ 。但是由于该方法的 G-tree 中的非叶子节点的距离矩阵中需要存储每个子图内的边界节点之间的距离信息,因为每个子图的边界节点一定是其孩子的边界节点,所以 G-tree 中存储了大量冗余的距离信息。另外,为了降低存储空间,G-tree 中只存储了子图中的边界节点的距离信息,没有存储任意节点对之间的距离信息,所以当路网规模增大,G-tree 的深度随着增大的时候,搜索代价明显提高,这就导致在大规模路网中执行 kNN 查询时效率降低。

2.2.3 相邻区域迭代式查询方法

采用相邻区域迭代扩展模式的 kNN 查询方法的主要依据就是查询点的 kNN 对象点必定在查询点所在区域的某些邻近区域。主要思想是:按照一定的规则将路网划分为不同的区域,执行 NN 查询时,首先在查询点所在的区域查找对象点,如果查询点所在区域内的对象点数目不满足查询要求,那么访问查询点所在区域的相邻区域,接着查找 NN 对象点。如此迭代下去,直到找到满足条件的 kNN 对象点。

(1) 采用 Voronoi 图与 R-tree 结合的索引结构的查询方法

Kolahdouzan 和 Shahabi^[14]在 2004 年提出了一种路网环境下基于 Voronoi 图的 kNN 查询方法——VN³方法。这种方法的主要思想是将整个路网划分为若干个 Voronoi 单元(如图 5(a)所示)。每个 Voronoi 单元以数据对象 p 为中心,并预先计算出每个 p 所在的单元中所有边界点到 p 的距离以及每个单元内的边界点之间的距离,并将预计算的距离值加以保存,然后通过 R-tree 来索引所有的 Voronoi 单元(存储结构如图 5(b)所示)。

VN³方法将查找第 1 个最近邻问题简化为点定位问题;如果想要查找 kNN($k>1$)对象点,文献[14]证明: q 的 kNN 必定是 q 的 i NN($i<k$)在 NVD(network voronoi diagram)中的某些邻近的 Voronoi 单元,所以接着就逐一访问查询点所在 Voronoi 单元的相邻单元。如此迭代搜索下去,找到第 k 个最近邻对象为止,将查询空间限制在一定范围之内。

由以上分析可知:VN³处理方法在查询 1NN 时的效率最高,它也可以改善文献[21]中 INE 算法在稀疏的网络中进行 kNN 查询时效率降低的情况;但是当查询数目 k 增加时,由于许多路径在多个 Voronoi 区域之间穿过,计算代价很高,严重影响了查询的性能。针对那些数据对象分布密集的网络,划分得到的 Voronoi 单元的数量就很大,预计算代价和存储代价都很高,并且增加了算法精炼步骤的计算代价,影响了算法的效率。因此,VN³方法只适合规模适中对象分布密度适中的网络。

(2) 基于中心节点构建索引的查询方法

王恒^[35]在 2011 年提出了基于预计算的跳跃式 kNN 查询(简称 Leaping-kNN)算法。该方法基于非交叉点子路径上的预计算定理,首先获得路网中的交叉点,并计算出交叉点的 m 个最近邻对象,存储到对应的最近邻列表中。因为查询点的最近邻对象可能被限定在它所在的子路径的开始结点或者结束结点的最近邻列表中,所以当获得查询请求后,确定查询点 q 所在的子路径,如果查询点所在子路径的最近邻列表中的查询对象个数小于 k ,不计算下一个邻接点的 m 个最近邻,而是通过跳跃式方式得到相邻子路径中的开始结点和结束结点的最近邻列表,并加入到候选子集中。最终,通过计算 q 到候选子集中查询对象的距离,取 k 个到达查询点距离最小的查询对象即为查询结果。

该算法在交叉点分布稀疏的路径中具有明显的优势,但是该方法的缺点是在交叉点密集分布的情况下,性能将会退化为普通的处理方法。而且此方法与 m 的取值有关:当 m 较大时,查询对象的候选集过大,使得预计算的代价较高;而当 m 较小时,会因候选集不足而不得不进行相邻子路径的迭代扩展,这样就会降低查询的效率。

2.3 最近邻查询方法比较

本节通过实验分析,从路网规模、对象点的分布密度和 k 值大小的影响方面分析和比较最近邻查询方法的性能。其中,改进的 Dijkstra 方法、MkNN 方法、NE 方法和 IkNNQA 方法都是在 INE 方法的基础上稍作改进实现的,从索引结构到扩展模式都没有变化,而 DisBro 方法是 SILC 方法的升级版,因此本节实验中没有考虑改进的 Dijkstra 方法、MkNN 方法、NE 方法、IkNNQA 方法和 SILC 方法。另外,IER 方法按照文献[25]中的 PHL 改进策略实现。

2.3.1 实验设置

(1) 实验环境

本节所有的算法都是在 Linux 操作系统上使用 C++ 语言实现,运行在以下配置的计算机上: Intel Core i5-4570 CPU@3.2GHz, 32GB RAM, 并且所有算法均基于单线程, 每种算法中的索引结构均基于内存实现.

(2) 数据集

实验中使用了 5 个真实的数据集^[25], 均从美国人口普查局公开发表的美国路网数据中采集, 数据集的规模参见表 1. 实验中所用的对象集包括真实对象集^[25]和人工合成两部分, 其中: 真实对象集见表 2; 人工合成部分主要从真实的路网数据集中随机选取一定数目的节点作为用户的兴趣点(POIs), 按照实验中对象分布密度参数 d 从 0.0001~1 的变化进行设置.

Table 1 Road network datasets

表 1 路网数据集

Name	Region	#Vertices	#Edges
DE	Delaware	48 812	119 004
ME	Maine	187 315	412 352
NW	North-West US	1 089 933	2 545 844
E	Eastern US	3 598 623	8 708 058
US	United States	23 947 347	57 708 624

Table 2 Real-world object sets

表 2 真实对象集

Object Set	United States		North-West US	
	Size	Density	Size	Density
Schools	160 525	0.007	4 441	0.004
Parks	69 338	0.003	5 098	0.005
FastFood	25 069	0.001	1 328	0.001
PostOffices	21 319	0.000 9	1 403	0.001
Hospitals	11 417	0.000 5	258	0.000 2
Hotels	8 742	0.000 4	460	0.000 4
Universities	3 954	0.000 2	95	0.000 09
Courthouses	2 161	0.000 09	49	0.000 05

2.3.2 实验分析

路网的规模、对象点的分布密度和 k 值大小在一定程度上影响着查询方法的性能, 下面从这 3 方面着手分析 k NN 查询方法的性能.

(1) 路网规模的影响(对象点分布密度 $d=0.001, k=10$, 查询时间单位为 μm).

通常情况下, k NN 查询方法的性能受路网规模影响, 随着路网中节点数目 $|V|$ 的增加, k NN 查询过程中的计算代价和存储代价增加, k NN 查询方法的查询效率会降低. 表 3 中的数据显示了 k NN 查询方法受路网规模变化影响的情况. 基于 Dijkstra 式扩展模式搜索的方法变化幅度不大, 而基于启发式搜索模式的方法受影响比较大, 特别是 DisBro 方法, 在数据集 E 和 US 上, 无法正常创建索引. 另外, 基于网络 Voronoi 图的查询方法受影响也比较大, 在数据集 E 和 US 上, 也无法正常创建索引.

(2) 对象点分布密度的影响(NW 数据集, $k=10$, 查询时间单位为 μm).

大多数查询处理方法的性能受网络空间中查询对象的分布密度 d 影响, 一般随着 d 增大, 对象之间的平均距离降低, 查询的执行时间会相应变小(见表 4). 基于 Dijkstra 式扩展模式搜索的方法受对象密度影响比较大, 变化幅度很明显, 特别是 INE 方法, 当对象密度达到 1 时, 成为所有方法中表现最优的. 因为 UNICONS 方法和 Leaping- k NN 方法预处理的最近邻列表中就存储了每个交叉节点的 10 个最近邻对象, 所以在查询过程中受密度影响不是很大, 查询时间变化幅度也比较小.

Table 3 Effect of road network size $|V|$

表 3 路网规模的影响

方法名称	DE	ME	NW	E	US
INE	900	890	1 000	1 150	1 200
PINE	480	480	580	-	-
Island ($r=0,1$)	200	400	600	850	1 500
UNICONS ($m=10$)	600	600	650	670	785
SPIE	680	700	750	780	850
S-Grid	1 000	950	1 200	1 280	1 400
ROAD	200	160	300	400	500
IER	7	18	26	35	-
DisBro	150	250	280	-	-
Distance Signature	700	800	800	800	850
G-tree	42	55	140	400	700
VN^3	1 200	1 000	1 100	-	-
Leaping- kNN ($m=10$)	555	570	550	585	705

Table 4 Effect of density d

表 4 对象分布密度的影响

方法名称	$d=0.0001$	$d=0.001$	$d=0.01$	$d=0.1$	$d=1$
INE	10 000	1 000	100	15	6
PINE	5 000	580	80	25	10
Island ($r=0,1$)	6 000	600	100	20	8
UNICONS ($m=10$)	900	650	100	80	75
SPIE	5 500	750	150	80	25
S-Grid	4 000	1 200	250	120	15
ROAD	900	300	100	30	8
IER	16	26	26	30	30
DisBro	850	280	100	40	30
Distance Signature	1 500	800	120	50	35
G-tree	165	140	70	38	10
VN^3	2 000	1 100	200	80	45
Leaping- kNN ($m=10$)	950	550	100	55	35

(3) k 值变化的影响(NW 数据集,对象点分布密度 $d=0.001$,查询时间单位为 μm)。

表 5 中的数据表明:随着 k 值增大,几乎所有的 kNN 查询方法的查询效率呈明显降低趋势.有些处理方法随着 k 值的增大,查询性能变得不可接受,严重影响了查询的意义.比如 VN^3 方法,在搜索 1NN 时效率最高;当 k 值增大时,效率明显下降,成为所有方法中性能最差的方法.

Table 5 Effect of k

表 5 k 值变化的影响

方法名称	$k=1$	$k=5$	$k=10$	$k=25$	$k=50$
INE	95	450	1 000	2 500	5 000
PINE	5	200	580	800	3 500
Island ($r=0,1$)	80	250	600	1 300	2 800
UNICONS ($m=10$)	80	300	650	600	1 500
SPIE	60	300	750	1 300	2 000
S-Grid	80	650	1 200	2 300	4 500
ROAD	38	150	300	800	1 750
IER	10	15	26	40	70
DisBro	20	100	280	900	2 500
Distance Signature	50	400	800	1 000	2 100
G-tree	35	75	140	200	3 200
VN^3	5	550	1 100	3 500	9 500
Leaping- kNN ($m=10$)	70	270	550	1 800	4 500

除了路网规模、对象点分布密度和 k 值大小对 kNN 查询方法的性能影响之外,创建索引的代价也会影响查询方法的扩展性.比如 DisBro 方法,虽然已经将空间复杂度有效地降低到 $O(|V|^{1.5})$ ($|V|$ 为路网节点的数目),但是创建索引的空间代价和时间代价都很高,当路网上节点个数超过 1 000 万时,DisBro 索引的存储空间已经达

到 17G,影响了 DisBro 查询方法的应用.另外,路网数据是动态变化的,首先是路网中的结点、边和兴趣点是动态变化的;其次,路网中道路的权值是动态变化的.有的索引方法不支持路网更新,严重束缚了查询方法的应用,比如 DisBro 方法,在更新频繁的网络中无法应用.

2.3.3 查询方法分析

下面就从以下 3 个方面比较和评估 kNN 查询方法(见表 6).

(1) Dijkstra 式查询方法

基于 Dijkstra 式搜索模式的查询方法在规模适中的高密度网络中查询性能处于优势,尤其是 INE 方法,在对象分布密集的路网中表现突出,而均采用改进的 Dijkstra 式搜索模式的 ROAD 方法与 INE 方法相比较,有时 ROAD 方法访问快捷边的时间已经超过了通过快捷边绕过不包含对象的子图所节省的时间,所以会出现前者略慢于后者的现象.基于启发式搜索模式中表现最好的 G-tree 方法的查询效率明显不如基于 Dijkstra 式搜索模式的 ROAD 方法.

(2) 启发式扩展模式

随着路网中节点 $|V|$ 数目逐渐增大,尤其是大规模的路网环境下,对象分布密度比较低的情况,基于启发式搜索模式的 kNN 查询方法的适应性会明显好于其他两种搜索模式.文献[25]中的实验结果也表明:随着路网中节点 $|V|$ 数目逐渐增大,改进后的基于启发式搜索策略的 IER 方法处于优势地位.随着查询请求中 k 值的增大,启发式搜索模式的优势也比较明显,受 k 值变化影响幅度没有其他方法那么明显.

(3) 相邻区域迭代式扩展模式

当查询请求中 k 值较小时,尤其是在查询 1NN 时,基于相邻区域迭代式搜索模式的方法效率最高;当 k 值较大时,这一方法要迭代访问当前查询区域的所有相邻区域,冗余计算比较大,查询效率降低.基于 Dijkstra 式搜索模式的查询方法随着 k 值的增大,性能也会明显降低;而基于启发式搜索模式的方法比较占优势,随着 k 值增大,性能降低较慢.

Table 6 Comparison of kNN query techniques

表 6 kNN 查询方法比较

扩展模式	方法名称	网络规模	k 值影响	对象分布密度
Dijkstra 式	INE	中、小规模	较小 k 值	高密度
	PINE	中、小规模	较小 k 值; $k=1$ 时效率最高;	高密度
	Island	中、小规模	较小 k 值	密度影响较小
	UNICONS	中、小规模	较小 k 值	密度影响较小
	SPIE	中、小规模	较小 k 值	高密度
	S-Grid	中、小规模	较小 k 值	高密度
	ROAD	适用大规模	较大 k 值	高密度
启发式	IER	适用大规模	较大 k 值	低密度
	DisBro	中、小规模	较小 k 值	低密度
	Distance Signature	中、小规模	较小 k 值	低密度
	G-tree	适用大规模	较大 k 值	低密度
相邻区域 迭代式	VN^3	中、小规模	较小 k 值; $k=1$ 时效率最高;	低密度
	Leaping- kNN	中、小规模	较小 k 值	低密度

综上所述,基于 Dijkstra 式搜索模式的查询方法一般适用路网规模适中而对象分布密度较高的路网环境;而基于启发式搜索模式的方法一般适用于密度较低的较大规模的网络;基于相邻区域迭代式搜索模式的查询方法则适用于网规模不大的网络中 k 值较小的 NN 查询处理,尤其是 1NN 查询效果最佳.

当然,各种算法的适应性也受其采用的索引结构的限制.有的索引结构(例如 SILC 和基于网络 Voronoi 图的索引结构),在路网规模或者对象点数目较大的网络中,创建索引所用时间以及索引所占用的空间是内存算法无法适应的.

3 路网环境下的最近邻查询技术的扩展

随着基于位置服务的广泛应用,用户不断提出各种复杂多样的位置查询请求.为了满足用户个性化的查询需求,最近邻的变体查询研究也取得了许多成果,其中包括连续最近邻(continuous nearest neighbor,简称 CNN)查询、聚集最近邻(aggregate nearest neighbor,简称 ANN)查询等技术的研究.这些技术主要是结合最近邻变体查询的特征,在已有的最近邻查询方法的基础上进行改进的.

3.1 连续最近邻查询技术

随着时间的变化,移动数据对象的空间位置也发生了改变,比一般的空间数据对象更复杂.连续最近邻(continuous nearest neighbor,简称 CNN)查询面临的挑战就是:如何减少随着查询对象的移动而带来的重复计算.目前,解决这一问题的方法大体可以分为两种情况:静态对象的 CNN 查询技术和移动对象的 CNN 查询技术.

(1) 静态对象的连续最近邻查询技术

给定起点是 s 、终点是 e 的路径 $path$,若查询点 q 在该路径上由 s 向 e 移动,求解查询点 q 在确定路径上的所有最近邻对象.比如查找起点 s 到终点 e 最短路径中的最近加油站.这一问题最终转化为求解 q 位于每一个子路段 $[s_i, s_j]$ 内的最近邻查询问题.典型的处理方法就是根据查询条件计算出安全区域(或者是安全路段),只有当移动对象离开或进入安全区域时才更新查询结果.从本质上讲,就是求移动对象在指定路径中快照的最近邻,相当于查找静态对象的连续最近邻.

文献[36]采用分治思想,将待查询路径分为不含目标点的子路径,利用子路径端点的 kNN 集与分割点的关系,计算出该子路径上的目标分割点和内部分割点的位置,最后合并各子路径的分割点集得到待查询路径的连续 k 最近邻居.文献[37,38]中为了解决网络中的 CNN 查询问题,提出了启发式的区域裁剪算法,通过这一算法可以获得预计算的查询区域,然后再基于查询区域计算查询路径上每个结点的 NN. Shekhar 等人^[39,40]也提出了 4 种解决路径上最近邻查询的方法,其中一种方法是基于区域预计算的.即:预先计算出每个对象点支配的区域,类似于 Voronoi 单元,确定了查询点所在的区域,就可以确定查询点当前的最近邻对象点.最后通过实验证明,基于区域预计算的方法远远优于其他解决方法.但是预计算方法要依赖于路网中数据对象和查询点的分布情况,并且还需要较大的额外存储空间.上述方法只能解决 $k=1$ 的 CNN 查询问题,无法解决 $CkNN$ 查询问题.

Kolahdouzan 等人^[41,42]提出了两种基于 VN^3 的解决网络空间 $CkNN$ 查询的处理方法:IE(intersection examination)和 UBA(upper bound algorithm).这两种方法首先将路网划分为 Voronoi 单元,在每个单元内,最近邻对象点都是唯一确定的,只要确定查询点当前所处的 Voronoi 单元,就可以计算出它的最近邻对象.其中的 UBA 算法对 IE 算法进行了改进,它仅在最近邻查询被需要时才计算,减少了 kNN 的计算次数.Cho 等人^[32]提出的 UNICONS 处理方法也可以处理连续的最近邻查询,实验验证:在解决 $CkNN$ 查询时,UNICONS 方法优于 UBA 查询处理方法.文献[43]提出了一种 IIE 算法解决 $CkNN$ 查询问题,该方法能够有效地确定安全区域的分隔点的位置及其相应的 $kNNs$,可以利用较少的静态查询产生连续查询的结果.

Li 等人^[44]改进了道路网络模型,并在该模型基础上结合 Dijkstra 单源最短路径算法实现了 $CkNN$ 的查询计算.Feng 等人^[45]提出了一种 $CkNN$ 查询方法,该方法只需要计算整条查询路线始末位置的 kNN ,并且减小了算法使用的存储空间.Li 等人^[46]讨论了无线广播环境下路网中 $CkNN$ 查询处理问题,提出了一种基于 Voronoi 图的分布式索引结构 NVD-DI,基于这种索引结构,提出了一种有效的 $CkNN$ 查询处理算法.Shahabi 等人^[47,48]提出了一种基于嵌入技术的方法来解决 $CkNN$ 查询问题:利用欧式距离快速计算出任意两点间的最短路径距离范围,以此作为上下界实现查询过程中的剪枝.文献[49-51]对文献[47,48]的工作进行了一定的改进,使上下界更加接近实际最短路径距离,并且降低了存储开销.

(2) 移动对象的连续最近邻查询技术

基于移动对象的 CNN 查询技术的关键是监视移动对象的变化情况,根据移动对象的当前位置和移动速度,计算某一时间段的连续最近邻.

Jensen 等人^[52]探讨了一种通用的结构框架来解决移动对象在路网中的 NN 查询问题,该结构框架包含了数

据模型和抽象定义,使用类似 Dijkstra 算法的思想来计算从查询点到目标对象间的最短距离,进而完成 CkNN 的查询。

Mouratidis 等人^[53]探讨了移动对象的 CkNN 监测问题,给出了一种增量监测算法 IMA(incremental monitoring algorithm).该算法在移动对象发生更新时,从前面的时间戳结果中增量获得查询结果用来重复评估当前的查询,使得重复查询产生的开销降低.但是位置更新的自然离散性使两个连续更新之间查询对象的 kNN 是未知的,造成 IMA 算法在两个连续更新的时间戳之间无法返回有效结果。

Huang 等人^[54]提出了一种路网中移动对象 CkNN 的监测算法,该算法大体上可分为剪枝阶段和提炼阶段两个部分.方法中,由于剪枝策略中设置的剪枝距离过大,不能有效地去除一些不合格的对象,导致了提炼阶段的性能不佳.Demiryurek 等人^[55]提出 ER-CkNN(Euclidian restriction based CkNN)查询处理方法,用于解决移动查询对象在动态变化的移动空间网络数据中的 CNN 查询.Liao 等人^[56]给出了一种路网有向图模型,并且引入单向网络距离度和双向网络距离度量方式,提出了单向网络扩展(UNE)算法和双向网络扩展(BNE)算法的 CkNN 查询处理方法。

Chen 等人^[57]提出了 k -PNN(k -path nearest neighbor)问题,就是已知用户的当前位置和用户的目标位置,在当前位置到目标位置的最短路径上为用户提供满足其要求的最近邻对象列表,因为用户的当前位置是随时变化的,所以到目标位置的最短路径也是动态的.为了解决这一问题,文献[57]中提出了一种包括查找、验证和监视等 3 个阶段的 BNE(best-first network expansion)算法。

Li 等人^[58-60]研究了解决路网中移动对象的 CkNN 查询的方法,处理速度不确定时移动对象的 kNN 查询问题.Shen 等人^[61]提出了 V-Tree 索引结构,并基于这一索引结构处理路网上的移动对象的 kNN 查询.文献[61,62]中,利用 V-Tree 索引结构有效地降低了移动对象位置变化时索引更新的代价,在查询时可以有效过滤掉无关的节点,提高了移动对象的 kNN 查询性能,是目前最为高效的移动对象 kNN 查询方法。

3.2 聚集最近邻查询技术

聚集最近邻的定义是:已知一组数据点集合 P 和查询点集合 Q ,从数据点集合 P 中查找到达查询点集合 Q 的聚集距离最小的点 p .比如几个朋友位于不同的地方,计划找一个到他们所在地方距离和最小的餐厅聚会. ANN 查询的关键就是如何减少不必要的距离计算和聚集距离计算.目前,主要的方法有利用欧式距离剪枝的改进的 IER 方法、利用平均距离作为剪枝下限的 TA(the threshold algorithm)算法和基于 Voronoi 的迭代扩展方法。

Yiu 等人^[63]为了有效地解决 ANN 问题分别提出了 IER(incremental Euclidean restriction)算法、改进的 IER 算法、TA 算法和 CE(concurrent expansion)算法.其中的 IER 算法是利用欧式距离进行剪枝,利用改进的 A*算法查找 ANN 对象点,而 TA 算法和 CE 算法就是基于路网上的平均距离阈值作为剪枝条件,直接扩展查询实现的,CE 算法是对 TA 算法的改进。

文献[64]也在改进的 IER 框架基础上,采用欧式距离作为剪枝策略,提出了一种单源的多目标 A*算法处理 ANN 查询问题,并在验证阶段进行了改进。

Zhu 等人^[65,66]提出利用 Voronoi 图的相邻性和预处理技术来解决 ANN 查询问题,文中将查询分为查找阶段和剪枝阶段.在查找阶段,分别从每个查询点开始寻找下一个最近邻目标对象,直至某个目标对象被所有查询点都扩展到,从而获得一个 ANN 候选集合;剪枝阶段就是去除 ANN 候选集合中不可能为 ANN 结果的目标对象,直到该候选集合中只有一个目标对象.文献[67]中又提出了一种基于 Voronoi 图的解决 k ANN 问题的方法,该方法首先通过 R 树索引查找出每一个查询点的 1NN,接着,按照某种策略不断计算某个查询 q 的下一个最近邻,然后检测这个最近邻对象点是否被所有的查询点扩展到:如果是,那么它就是所求的 1ANN;如果不是,那么更新它的当前聚集距离,再继续查找其他查询点的下一个最近邻.依此类推,最终可以找到所求的 k ANN 对象点。

还有的学者研究了连续的 ANN 查询问题.Qin 等人^[68]提出了增量的 BUA(bidirectional updating algorithm)算法解决连续 ANN 查询问题,也就是从移动的数据点集中查找到多个查询点的聚合距离最小的 top- k 个数据点.Moumtidis 等人^[69]研究了动态路网中的 ANN 查询.文献[69]中提出了 IMA 算法和 GMA 算法,IMA 算法存储最短路径扩展树并提出了 affecting edges 的概念,通过对目标节点更新、查询点更新和边权值更新这 3 种情况

分别讨论,并分别提出了相应的处理方法.在 GMA 算法中提出了 sequence 的概念,对处于同一 sequence 中的查询点利用 IMA 算法并行处理.

4 总结与展望

本文对路网环境下的最近邻查询技术进行了分类,从索引技术和查询方法两方面介绍了路网环境下的最近邻查询技术,并从网络规模、 k 值的大小和对象分布密度 3 个因素探讨了最近邻查询技术之间的区别.目前的研究已经取得了阶段性的成果,但是还存在值得进一步研究的问题.

- (1) 路网动态更新.路网数据是动态变化的:一是路网中的节点、路段和兴趣点是动态变化的;其次,路网中道路的权值是动态变化的;另外,查询点和查询对象都可能是动态变化的.在上述情况下,如何适应路网数据的动态更新、高效完成最近邻查询处理,是路网环境下最近邻查询技术需要面临的重大挑战.
- (2) 考虑用户的个性化查询.随着基于位置服务的广泛应用,用户往往会有很多个性化的位置查询需求,这些查询大多都是以最近邻查询为基础的,比如带有时间约束或者包含关键字的最近邻查询问题.
- (3) 可伸缩性.不同区域的路网规模不同、对象分布密度不同,而且同一区域路网的规模和对象分布密度也在不断增长,查询方法的可伸缩性就显得非常重要.目前,最近邻技术可伸缩性还有待进一步提高.
- (4) 实验评测.文献[25]对路网环境下的主流技术进行了实验比较,但是评测还可以考虑更多的技术,同时也需要考虑各种技术在路网更新的情况下的适应性.

References:

- [1] Donald K. The Art of Computer Programming (3). Indianapolis: Addison-Wesley Professional, 1973.
- [2] Roussopoulos N, Kelley S, Vincent F. Nearest neighbor queries. *ACM Sigmod Record*, 1995,24(2):71–79. [doi: 10.1145/223784.223794]
- [3] Papadopoulos AN, Yannis M. Performance of nearest neighbor queries in R-trees. In: *Proc. of the Int'l Conf. on Database Theory*. 1997. 394–408. [doi: 10.1007/3-540-62222-5_59]
- [4] Hjaltason GR, Samet H. Distance browsing in spatial databases. *ACM Trans. on Database Systems (TODS)*, 1999,24(2):265–318. [doi: 10.1145/320248.320255]
- [5] Korn F, Sidiropoulos ND, Faloutsos C, Siegel EL, Protopapas Z. Fast nearest neighbor search in medical image databases. In: *Proc. of the VLDB*. 1996. 215–226.
- [6] Seidl T, Kriegel H. Optimal multi-step k -nearest neighbor search. *ACM SIGMOD Record*, 1998,27(2):154–165. [doi: 10.1145/276304.276319]
- [7] Berchtold S, Ertl B, Keim DA, Kriegel H, Seidl T. Fast nearest neighbor search in high-dimensional space. In: *Proc. of the ICDE*. 1998. 209–218.
- [8] Lin K, Yang C. The ANN-tree: An index for efficient approximate nearest neighbor search. In: *Proc. of the Database Systems for Advanced Applications*. 2001. 174–181. [doi: 10.1109/DASFAA.2001.916376]
- [9] Belussi A, Bertino E, Catania B. Using spatial data access structures for filtering nearest neighbor queries. *Data & Knowledge Engineering*, 2002,40(1):1–31. [doi: 10.1016/S0169-023X(01)00033-7]
- [10] Hjaltason GR, Samet H. Ranking in Spatial Databases. 1995. 83–95. [doi: 10.1007/3-540-60159-7_6]
- [11] Henrich A. A distance scan algorithm for spatial access structures. *Int'l Journal of Geographical Information Science*, 1994. 136–143.
- [12] Sharifzadeh M, Shahabi C. Vor-Tree: R-trees with voronoi diagrams for efficient processing of spatial nearest neighbor queries. In: *Proc. of the VLDB*. 2010. 1231–1242. [doi: 10.14778/1920841.1920994]
- [13] Zhu H, Yang X, Wang B, *et al.* Range-Based obstructed nearest neighbor queries. In: *Proc. of the SIGMOD*. 2016. 2053–2068. [doi: 10.1145/2882903.2915234]
- [14] Kolahdouzan MR, Shahabi C. Voronoi-Based k nearest neighbor search for spatial network databases. In: *Proc. of the VLDB*. 2004. 840–851. [doi: 10.1016/B978-012088469-8.50074-7]

- [15] Sankaranarayanan J, Alborzi H, Samet H. Efficient query processing on spatial networks. In: Proc. of the GIS. 2005. 200–209. [doi: 10.1145/1097064.1097093]
- [16] Samet H, Sankaranarayanan J, Alborzi H. Scalable network distance browsing in spatial databases. In: Proc. of the SIGMOD. 2008. 43–54. [doi: 10.1145/1376616.1376623]
- [17] Lee KCK, Lee WC, Zheng BH. Fast object search on road networks. In: Proc. of the EDBT. 2009. 1018–1029. [doi: 10.1145/1516360.1516476]
- [18] Huang XG, Jensen CS, Šaltenis S. The Islands approach to nearest neighbor querying in spatial networks. In: Proc. of the Int'l Symp. on Spatial and Temporal Databases (SSTD). 2005. 73–90. [doi: 10.1007/11535331_5]
- [19] Huang XG, Jensen CS, Šaltenis S. Multiple k nearest neighbor query processing in spatial network databases. In: Proc. of the Advances in Databases and Information Systems. Berlin, Heidelberg: Springer-Verlag, 2006. 266–281. [doi: 10.1007/11827252_21]
- [20] Huang XG, Jensen CS, Liu H, Šaltenis S. S-GRID: A versatile approach to efficient query processing in spatial networks. In: Proc. of the SSTD. 2007. 93–111. [doi: 10.1007/978-3-540-73540-3_6]
- [21] Papadias D, Zhang J, Mamoulis N, Tao YF. Query processing in spatial network databases. In: Proc. of the VLDB. 2003. 802–813. [doi: 10.1016/B978-012722442-8/50076-8]
- [22] Hu HB, Lee DL, Lee VCS. Distance indexing on road networks. In: Proc. of the VLDB. 2006. 894–905.
- [23] Hu HB, Lee DL, Xu JL. Fast nearest neighbor search on road networks. In: Proc. of the EDBT. 2006. 186–203. [doi: 10.1007/11687238_14]
- [24] Zhong RC, Li GL, Tan KL, Zhou LZ. G-Tree: An efficient index for k nn search on road networks. In: Proc. of the CIKM. 2013. 39–48. [doi: 10.1145/2505515.2505749]
- [25] Abeywickrama T, Cheema MA, Taniar D. K -Nearest neighbors on road networks: A journey in experimentation and in-memory implementation. Proc. of the VLDB Endowment, 2016,9(6):492–503. [doi: 10.14778/2904121.2904125]
- [26] Gaede V, Günther O. Multidimensional access method. ACM Computing Surveys, 1998,30(2):170–231. [doi: 10.1145/280277.280279]
- [27] Dijkstra EW. A note on two problems in connexion with graphs. In: Proc. of the Numerische Mathematik. 1959. 269–271. [doi: 10.1007/BF01386390]
- [28] Sun Y. Design and implementation of nearest neighbor query in spatial network database. Computer Science, 2008,35(3):73–75 (in Chinese with English abstract). [doi: 10.3969/j.issn.1002-137X.2008.03.021]
- [29] De AVT, Güting RH. Using Dijkstra's algorithm to incrementally find the k -nearest neighbors in spatial network databases. In: Proc. of the Symp. on Applied Computing. 2006. 58–62. [doi: 10.1145/1141277.1141291]
- [30] Hou SJ, Liu GH, Yu J, Chu BY. An algorithm for k nearest neighbors queries in spatial network databases. Computer Science, 2006,s32(8):360–362 (in Chinese with English abstract).
- [31] Safar M. K nearest neighbor search in navigation systems. Mobile Information Systems, 2005,1(3):207–224. [doi: 10.1155/2005/692568]
- [32] Cho HJ, Chung CW. An efficient and scalable approach to CNN queries in a road network. In: Proc. of the VLDB. 2005. 865–876.
- [33] Lee KCK, Lee WC, Zheng BH, Tian Y. Road: A new spatial object search framework for road networks. TKDE, 2012,24(3): 547–560. [doi: 10.1109/TKDE.2010.243]
- [34] Zhong RC, Li GL, Tan KL, Zhou LZ, Gong ZG. G-Tree: An efficient and scalable index for spatial search on road networks. TKDE, 2015,27(8):2175–2189. [doi: 10.1109/TKDE.2015.2399306]
- [35] Wang H. Pre-Computing-Based algorithm for nearest neighbors leaping query over road network. Journal of Tianjin University of Technology, 2011,27(2):38–42 (in Chinese with English abstract). [doi: 10.3969/j.issn.1673-095X.2011.02.009]
- [36] Zheng Z, Zhang SZ, Guo L, Shi BL. An approach to continuous k nearest neighbor query in road network. Journal of Computer Research and Development, 2007,44(s3):398–401 (in Chinese with English abstract).
- [37] Wang B, Yang XC, Wang GR, Yu G, Zang WY, Yu M. Energy efficient approximate self-adaptive data collection in wireless sensor networks. Frontiers of Computer Science in China, 2016,10(5):936–950. [doi: 10.1007/s11704-016-4525-7]

- [38] Feng J, Watanabe T. Search of continuous nearest target objects along route on large hierarchical road network. In: Proc. of the Data Engineering Workshop (DEWS). 2003. 45–50.
- [39] Shekhar S, Yoo JS. Processing in-route nearest neighbor queries: A comparison of alternative approaches. In: Proc. of the Workshop Geographic Information Systems. 2003. 9–16. [doi: 10.1145/956676.956678]
- [40] Yoo JS, Shekhar S. In-Route nearest neighbor queries. *Geoinformatica*, 2005,9(2):117–137. [doi: 10.1007/s10707-005-6671-1]
- [41] Kolahdouzan MR, Shahabi C. Continuous k -nearest neighbor queries in spatial network databases. In: Proc. of the STDBM. 2004. 33–40.
- [42] Kolahdouzan MR, Shahabi C. Alternative solutions for continuous k nearest neighbor queries in spatial network databases. *GeoInformatica*, 2005,9(4):321–341. [doi: 10.1007/s10707-005-4575-8]
- [43] Guan YY, Xiao YY, Li YK. Continuous k nearest neighbor queries in road networks. *Journal of Tianjin University of Technology*, 2012,28(6):31–33, 43 (in Chinese with English abstract). [doi: 10.3969/j.issn.1673-095X.2012.06.008]
- [44] Li XL, He YB. Continuous k nearest neighbor queries in road network based on network Voronoi diagrams. *Information Technology*, 2007,31(12):103–104, 108 (in Chinese with English abstract). [doi: 10.3969/j.issn.1009-2552.2007.12.031]
- [45] Feng HY, Guo JF. Continuous nearest neighbor queries in road network. *Computer Engineering*, 2010,36(8):79–82 (in Chinese with English abstract). [doi: 10.3969/j.issn.1000-3428.2010.08.028]
- [46] Li YH, Li JJ, Shu LC, Li Q, Li GH, Yang FM. Searching continuous nearest neighbors in road networks on the air. In: Proc. of the Information Systems. 2014. 177–194. [doi: 10.1016/j.is.2014.01.003]
- [47] Shahabi C, Kolahdouzan MR, Sharifzadeh M. A road network embedding technique for k -nearest neighbor search in moving object databases. *Int'l Journal of Geographical Information Science*, 2002. 94–100. [doi: 10.1145/585147.585167]
- [48] Shahabi C, Kolahdouzan MR, Sharifzadeh M. A road network embedding technique for k -nearest neighbor search in moving object databases. *Geoinformatica*, 2003,7(3):255–273. [doi: 10.1023/A:1025153016110]
- [49] Kriegel H, Kroger P, Kunath P, Renz M. Proximity queries in large traffic networks. *Int'l Journal of Geographical Information Science*, 2007. 21–28. [doi: 10.1145/1341012.1341040]
- [50] Wang B, Zhu R, Yang XC, Wang GR. Top- k representative documents query over geo-textual data stream. *World Wide Web-Internet & Web Information Systems*, 2017,20(8). [doi: 10.1007/s11280-017-0470-0]
- [51] Kriegel H, Kroger P, Renz M, Schmidt T. Hierarchical graph embedding for efficient query processing in very large traffic networks. In: Proc. of the SSDBM. 2008. 150–167. [doi: 10.1007/978-3-540-69497-7_12]
- [52] Jensen CS, Kolařvr J, Pedersen TB, Timko I. Nearest neighbor queries in road networks. In: Proc. of the Int'l Symp. on Advances in Geographic Information Systems. 2003. 1–8. [doi: 10.1145/956676.956677]
- [53] Mouratidis K, Yiu ML, Papadias D, Mamoulis N. Continuous nearest neighbor monitoring in road networks. In: Proc. of the VLDB. 2006. 43–54.
- [54] Huang YK, Chen ZW, Lee C. Continuous k -nearest neighbor query over moving objects in road networks. In: Proc. of the Advances in Data and Web Management. 2009. 27–38. [doi: 10.1007/978-3-642-00672-2_5]
- [55] Demiryurek U, Banaeikashani F, Shahabi C. Efficient continuous nearest neighbor query in spatial networks using euclidean restriction. In: Proc. of the SSTD. 2009. 25–43. [doi: 10.1007/978-3-642-02982-0_5]
- [56] Liao W, Zhang Q, Wu XP, Zhong ZN. Research on continuous k nearest neighbor queries in road networks. *Journal of Chinese Computer Systems*, 2010,31(4):666–671 (in Chinese with English abstract).
- [57] Chen ZB, Shen HT, Zhou XF, Yu JX. Monitoring path nearest neighbor in road networks. In: Proc. of the SIGMOD. 2009. 591–602. [doi: 10.1145/1559845.1559907]
- [58] Li GH, Fan P, Li YH, Du JQ. An efficient technique for continuous k -nearest neighbor query processing on moving objects in a road network. In: Proc. of the Computer and Information Technology (CIT). 2010. 627–634. [doi: DOI10.1109/CIT.2010.127]
- [59] Li GH, Li YH, Shu LY, Fan P. Ck NN query processing over moving objects with uncertain speed in road networks. In: Proc. of the APWeb. LNCS 6612. 2011. 65–76. [doi: 10.1007/978-3-642-20291-9_9]
- [60] Fan P, Li GH, Yuan L, Li YH. Vague continuous k -nearest neighbor queries over moving objects with uncertain velocity in road networks. *Information Systems*, 2012,37(1):13–32. [doi: 10.1016/j.is.2011.08.002]

- [61] Shen BL, Zhao Y, Li GL, Zheng W, Qin Y, Yuan B, Rao Y. V-Tree: Efficient k NN search on moving objects with road-network constraints. In: Proc. of the ICDE. 2017. 609–620. [doi: 10.1109/ICDE.2017.115]
- [62] Yang X, Wang B, Yang K, Liu C, Zheng B. A novel representation and compression for queries on trajectories in road networks. IEEE Trans. of Data Engineering (TKDE), 2018. [doi: 10.1109/TKDE.2017.2776927]
- [63] Yiu ML, Mamoulis N, Papadias D. Aggregate nearest neighbor queries in road networks. TKDE, 2005,17(6):1–14. [doi: 10.1109/TKDE.2005.87]
- [64] Htoo H, Ohsawa Y, Sonehara N, Sakauchi M. Aggregate nearest neighbor search methods using SSMTA* algorithm on road-network. In: Proc. of the Advances in Databases and Information Systems. 2012. 181–194. [doi: 10.1007/978-3-642-33074-2_14]
- [65] Zhu L, Jing Y, Sun W, Mao DD, Liu P. Voronoi-Based aggregate nearest neighbor query processing in road networks. In: Proc. of the SIGSPATIAL Int'l Conf. on Advances in Geographic Information Systems. 2010. 518–521. [doi: 10.1145/1869790.1869876]
- [66] Sun WW, Chen CN, Zhu L, Gao YJ, Jing YN, Li Q. On efficient aggregate nearest neighbor query processing in road networks. Journal of Computer Science and Technology, 2015,30(4):781–798. [doi: 10.1007/s11390-015-1560-z]
- [67] Zhu L, Sun WW, Jing YN, Du JF. Voronoi-Based k -aggregate nearest neighbor query processing in road networks. Journal of Computer Research and Development, 2011,s48(10):155–162. [doi: 10.1145/1869790.1869876]
- [68] Qin L, Ding B. Monitoring aggregate k -NN objects in road networks. In: Proc. of the Statistical and Scientific Database Management. 2008. 168–186. [doi: 10.1007/978-3-540-69497-7_13]
- [69] Mouratidis K, Yiu ML, Papadias D, Mamoulis N. Continuous nearest neighbor monitoring in road networks. In: Proc. of the VLDB. 2006. 43–54.

附中文参考文献:

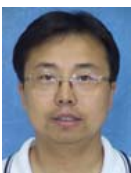
- [28] 孙亚. 空间网络数据库中最近邻查询的设计与实现. 计算机科学, 2008, 35(3): 73–75. [doi: 10.3969/j.issn.1002-137X.2008.03.021]
- [30] 侯士江, 刘国华, 余靖, 褚兵义. 空间网络数据库中的 k 个最近邻查询算法. 计算机科学, 2006, s32(8): 360–362.
- [35] 王恒. 路网中基于预计算的跳跃式查询最近邻的算法. 天津理工大学学报, 2011, 27(2): 38–42. [doi: 10.3969/j.issn.1673-095X.2011.02.009]
- [36] 郑铮, 张守志, 郭立, 施伯乐. 一种解决道路空间中连续 k 最近邻居查询的方法. 计算机研究与发展, 2007, 44(s3): 398–401.
- [43] 管莹莹, 肖迎元, 李玉坤. 基于路网的连续 k 最近邻查询. 天津理工大学学报, 2012, 28(6): 31–33, 43. [doi: 10.3969/j.issn.1673-095X.2012.06.008]
- [44] 李晓丽, 何云斌. 基于网络 Voronoi 图的道路网络连续 k 近邻查询. 信息技术, 2007, 31(12): 103–104, 108. [doi: 10.3969/j.issn.1009-2552.2007.12.031]
- [45] 冯惠妍, 郭俊风. 道路网络中的连续最近邻查询. 计算机工程, 2010, 36(8): 79–82. [doi: 10.3969/j.issn.1000-3428.2010.08.028]
- [56] 廖巍, 张琪, 吴晓平, 钟志农. 道路网络环境下的连续 k 近邻查询处理研究. 小型微型计算机系统, 2010, 31(4): 666–671.



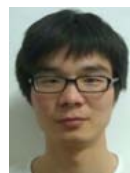
鲍金玲(1977—),女,吉林白城人,博士生,副教授,主要研究领域为空间数据库.



杨晓春(1973—),女,博士,教授,博士生导师,CCF 高级会员,主要研究领域为数据库理论与技术,大数据管理与分析,数据隐私保护.



王斌(1972—),男,博士,副教授,主要研究领域为数据科学,数据流管理,数据质量管理.



朱怀杰(1988—),男,博士生,主要研究领域为空间数据库,隐私保护.