空间网络数据库中最近邻查询的设计与实现

孙 亚

(浙江丽水广播电视大学理工教研室 浙江丽水 323000)

摘 要 随着无线通讯技术、位置定位技术以及数据库技术的发展,使得能为移动用户提供相关的位置服务。K近邻查询是位置服务的一个重要功能。本文主要研究了空间网络数据库中的K近邻查询。以提出的集成道路网络距离与欧式距离的道路网络框架为基础,提出了一种新的KNN查询算法,通过网络扩展方法计算最近邻(NN),避免了不必要的磁盘I/Os,减少了昂贵的最短路径计算,从而有效地提高了算法效率。实验结果说明,在目标点分布比较密集的情况下,算法显著优于其它的算法。

关键词 空间网络数据库,KNN查询,道路网络

Design and Implementation of Nearst Neighbor Query in Spatial Network Database

SUN Ya

(Polytechnic Teaching and Resarch Section, Lishui Broadcast TV University of Zhejiang, Zhejiang Lishui 323000)

Abstract With rapid advances in wireless communications, databases, and positioning technologies, it becomes possible to offer new e-services that provide mobile users with information about points of interesting. K nearest neighbor queries is essential for location-based position. In this paper we investigate k nearest neighbor searches in spatial network databases. We propose a road network architecture that integrates network and Euclidean information. Based on the architecture, we develop an efficient algorithm that calculates nearest neighbor. Experiments show that the algorithm outperforms other algorithms in high object density.

Keywords Spatial network databases, K nearest neighbor queries, Road network

1 引言

近年来,在空间数据库研究领域,K近邻(KNN)查询是一个引起广泛关注的问题。在地理信息系统中,我们经常需要使用KNN查询。KNN查询定义如下:假设有一个空间目标点集以及一个查询点,那么找到离这个查询点最近的K个空间目标。例如,找到离某出租车最近的5个餐馆。在空间网络数据库(SNDB)中,移动对象总是在道路网络上移动。这意味着目标之间(例如出租车与餐馆)的距离量度标准必须用网络距离(例如最短路径、最短时间)来衡量,而不是欧氏距离。目标之间的最短网络距离取决于网络之间的连通性而不是网络的空间位置。本文研究了满足下列两种情况的KNN查询:1)查询点(例如汽车、行人)和查询对象(例如加油站、餐馆)都位于道路网络上;2)距离的量度标准定义为网络上的最短路径距离。

基于欧式距离的最近邻查询算法最早由Roussopoulos等于1995年提出[1]。作者使用分枝界定的R树遍历算法,提出了三个启发式规则来过滤不包含最近邻居的结点,从而减少结点访问个数,减少磁盘I/O,进而有效地提高查询性能。Cheung等进一步改进该算法,移除既不能增强剪枝效果又具有高计算复杂度的剪枝规则,减少了CPU计算代价[2]。G. R. Hjaltson等在文[3]中提出R-tree索引结构结合优先队列的方法。这个方法使得查询时不需要遍历所有的对象,通过利用剪枝策略提高查询效率,而且根据优先队列的特性,对于查找K近邻效率更高。

最近有学者开始研究基于网络距离的最近邻查询算法。Shahabi等在文[4]中提出了一种嵌入式的方法,它的优点是把道路网络转换到高维空间,使得计算比较简单,主要缺点是不能计算精确的距离值。Jensen等在文[5]中提出了针对空间网络数据库中KNN查询的数据模型以及抽象功能的定义,使用类似Dijkstra算法的算法执行查询点与查询目标的在线最短路径计算。Shekhar等在文[6]中提出了在给定路径上对移动查询点查找最近邻的四种可选方法。

Kolahdouzan等在文[8]中提出了一种解决空间网络数据库中KNN查询的VN^3方法。这种方法预计算出目标点的网络Voronoi多边形(NVPs)以及一些网络距离。这种算法的优点在于:第一,根据网络Voronoi图的属性,NVPs能直接找到查询点的第一个最近邻。第二,NVPs的邻接信息能对目标点q的$k(k\geq 1)$近邻查询提供候选集。然而,在需要查找的NNs数目增加及目标点数目增加的情况下,VN^3中NVPs的预计算需要巨大的计算花费。因此,在目标点是高密度的情况下,VN^3的效率将极大退化。

本文提出了一种基于道路网络扩展的有效的KNN查询算法,即NE(Network Expansion)算法。这种方法极大地减少了KNN查询的花费。还提出了一种结合道路网络距离与欧式距离的道路网络框架,这个框架可表示道路的转向限制,使得算法可以获得更有效的结果。作者的贡献总结如下:

- 提出了一个集成道路网络距离与欧式距离的道路网络框架;
- 提出了一种有效的KNN查询算法,可进行道路网络

孙 亚 硕士,国家认证系统分析师,讲师,研究方向:并行分布处理、网络设计与信息系统安全、人工智能。

中的KNN查询；

• 利用网络扩展的方法计算KNN，避免了不必要的磁盘I/O，极大地减少了KNN的计算花费。在真实数据的实验环境中，NE优于VN³算法。

本文结构如下：第2部分介绍所使用的道路网络框架；第3部分提出基于网络距离的KNN查询算法；第4部分利用实验评测了提出的算法；最后是论文总结。

2 道路网络模型

为了支持在道路网络上进行空间信息的查询，笔者提出了一种结合道路网络距离与欧氏距离的道路网络框架。

该道路网络模型支持对交通转向限制的表示，通过添加额外的结点，利用结点之间的连接关系，从而表示出道路的转向限制关系，如图1所示。

道路网络框架如图2所示，共由三部分组成。

1)邻接表。邻接表表示了道路网络的连通性，每个结点的记录存储了邻接点的存储位置、与邻接点形成弧段的网络距离以及对应的MBR，还存储了对应弧段在poly-line表中的存储位置。

2)poly-line表。poly-line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。

3)网络R树。利用R树[10]对poly-line的MBR进行索引，并支持对网络空间属性的查询。

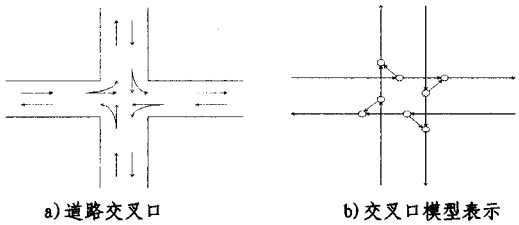

图1 道路交叉口建模示例

道路网络框架支持如下的基本操作。

(1)*check_entity*(*seg*, *p*)：判断点*p*是否位于路段*seg*。假如点*p*位于路段*seg*上，则称为*seg*覆盖*p*；

(2)*find_segment*(*p*)：通过网络R树，获得覆盖点*p*的弧段；

(3)*find_entities*(*seg*)：获得弧段*seg*上的所有目标点；

(4)*compute_ND*(p_1, p_2)：计算点p_1与p_2之间的网络距离。

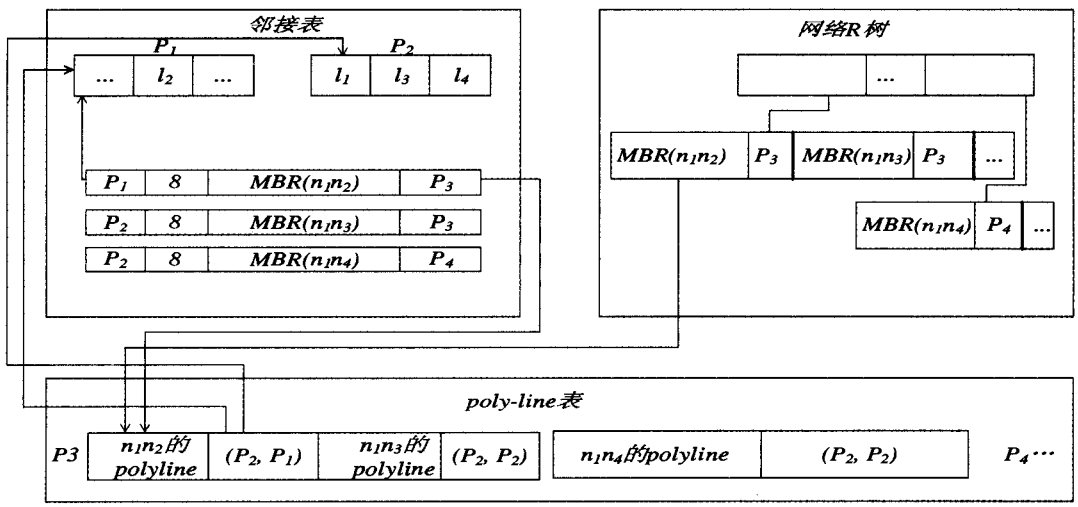

图2 道路网络框架

3 NN查询算法

NE算法的主要思想是从查询点*q*开始执行网络扩展，按照相遇的顺序检测目标，从而避免不必要的磁盘I/Os，减少昂贵的最短路径计算花费。以图3为例，图中实心黑圆表示道路结点，空心圆表示查询点位置，实心矩形块表示查询目标。NE首先定位覆盖*q*的路段n_1n_2，然后获取路段n_1n_2上的所有实体(使用初始操作*find_entities*)。由于n_1n_2上没有别的点，扩展离查询点最近的结点(n_1)(然后，n_1n_2的第二个结点n_2插入队列*Q*)。n_1n_7上没有数据点，然后把n_7插入队列，Q=⟨(n_2,5), (n_7, 12)⟩。n_2的扩展是n_4和n_3，并依次插入队列，Q=⟨(n_4,7), (n_3, 9) (n_7, 12)⟩，n_2n_4上找到点p_5(n_2n_3上没找到点)。距离$d_N(q, p_5)$=6提供了界限d_{Nmax}限制搜索空间。由于*Q*中的下个入口n_4有比d_{Nmax}更大的距离，算法终止。图4是NE的伪代码。

4 实验分析

这一节比较了NE算法与VN³算法KNN查询时I/O花费的差别。VN³算法是目前认为性能比较良好的算法。实验环境为Windows操作系统，CPU频率为2.4GHz，内存为512M。使用了湖北省武汉市的真实地图数据。设定页面大小为4kB，并使用了8MB的LRU缓冲区。道路数据的结构如下：|*N*|=52,868，|*E*|=58,720，*N*是结点集，*E*是弧段集，共有8,542个交叉点。目标点集包括156个学校、358个公汽站、2354个湖泊、3812个地物。为了研究KNN查询的效率，执行了300次查询，查询位置是在道路网络中随机选取的。

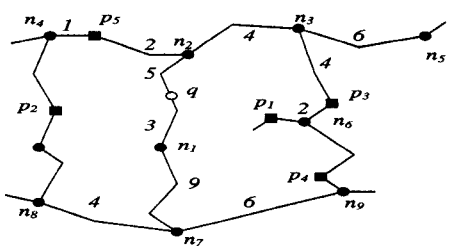

图3 道路网络图

```
算法 NE(q, k)
/* q 是查询点，k是将获得的最近邻的数目 */
1 n_i n_j = find_segment(q)
2 S_cover = find_entities(n_i n_j)
3 {p_1, ···, p_k} 是S_cover据依据网络距离升序排列的 k个最近目标
4 d_Nmax = d_N(q,p_k)
5 Q = <(n_i, d_N(q,n_i)), (n_j, d_N(q,n_j))> //根据d_N排序
6 根据最小的d_N(q,n)值取出Q中的节点n
7 while(d_N(q,n)<d_Nmax
8     for n中每一个未访问过的邻近节点 n_x
9         S_cover = find_entities(n,n)
10        更新 {p_1, ···, p_k} 从 {p_1, ···, p_k}∩S_cover
11        d_Nmax = d_N(q,p_k)
12        en-queue(n_x, d_N(q,n_x))
13    取出Q中的下一个节点n
14 return
```

图 4　NE 算法

作者比较了 NE 算法与 KNN 查询时的效率。图 5 显示了两种方法中 k 的取值从 1 到 32 时磁盘 I/O 的次数。随着目标点数量的增加，页存取的次数迅速降低。当 k 值为 1 时，VN³ 算法的花费为常量，与目标点集的密度无关。如图 5(a)及图 5(b)，VN³ 算法比 NE 算法更有效率，这是由于目标点稀疏分布，NE 需要访问大量的道路网络。

另一方面，如图 5(c)、图 5(d)所示，随着目标集分布密度的提高，NE 优于 VN³ 算法。NE 与 VN³ 算法的效率差别与目标点的分布密度密切相关。在目标点稀疏分布的情况下(例如酒店)，VN³ 算法优于 NE，因为 NVPs 的数目与目标点的数目相当。相反，在目标点分别密集的情况下，VN³ 算法需要大量的网络距离计算，使得效率降低。NE 算法利用网络扩展方法减少了查找空间，NE 的效率优于 VN³ 算法。此外，NE 算法中，处于同一弧段的目标点查找的情况是相同的，因此只需要较少的磁盘 I/Os。综上所述，实验结果表明，除了目标点分布稀疏的特殊情况外，NE 效率均优于 VN³ 算法，并且随着目标点的分布密度以及 k 值的增大，NE 算法越来越优于 VN³ 算法。

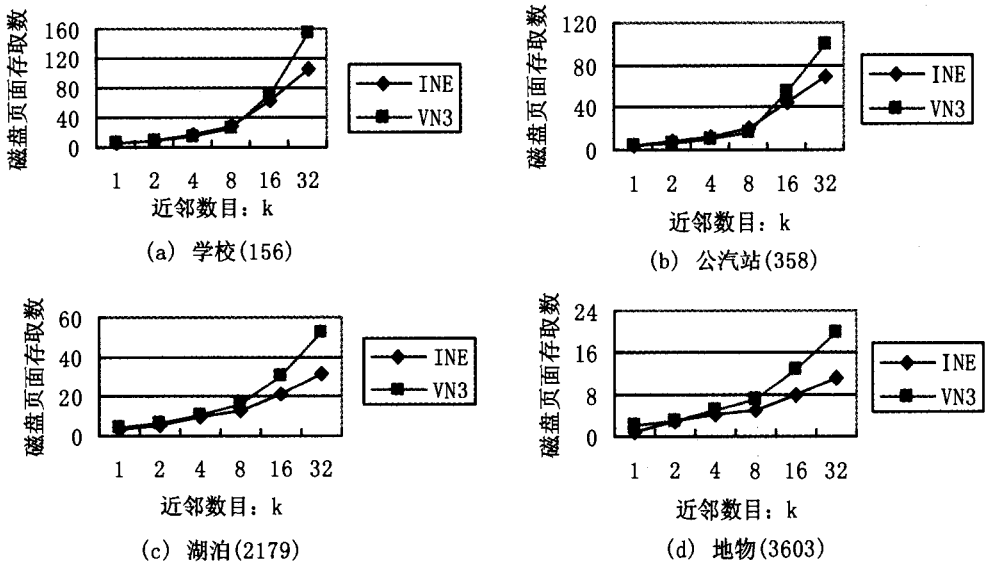

图 5　KNN 查询效率比较

结论　本文主要研究了空间网络数据库中的 K 近邻查询。基于空间网络数据库，作者首先提出了一个集成道路网络距离与欧式距离的道路网络框架。在此基础上，作者提出了一种新的 KNN 查询算法，通过网络扩展计算 NN，避免了不必要的磁盘 I/Os，减少了昂贵的最短路径计算，从而有效提高了算法效率，并对本算法与其它算法的查询效率进行了实验比较。实验结果说明，在目标点集分布比较密集的情况下，算法显著优于其它的算法。

参 考 文 献

1　Roussopoulos N, Vincent F. Nearest neighbor queries. In: Proceedings of ACM SIGMOD, San Jose, CA, 1995

2　Cheung K L, Fu A W C. Enhanced Nearest Neighbour Search on the R-tree. SIGMOD Record, 1998, 27(3): 16～21

3　Hjaltason G R, Samet H. Distance browsing in spatial databases. ACM Transactions on Database Systems (TODS), 1999, 24(2): 265～318

4　Shahabi C, Kolahdouzan M R, Sharifzadeh M. A Road Network Embedding Technique for K-Nearest Neighbor Search in Moving Object Databases. GeoInformatica, 2003, 7(3): 255～273

5　Jensen C S, et al. Nearest neighbor queries in road networks. In: Proceedings of the 11th ACM International Symposium on Advances in Geographic Information Systems, 2003. 1～8

6　Shekhar S, Yoo J S. Processing in-route nearest neighbor queries: a comparison of alternative approaches. In: Proceedings of the 11th ACM International Symposium on Advances in Geographic Information Systems, 2003. 9～16

7　Papadias D, et al. Query Processing in Spatial Network Databases. In: Proc. VLDB, 2003. 802～813

8　Kolahdouzan M, Shahabi C. Voronoi-Based K Nearest Neighbor Search for Spatial Network Databases. In: Proc. of VLDB, 2004

9　Dijkstra E W. A note on two problems in connexion with graphs. Numerische Mathematik, 1959, 1(1): 269～271

10　Guttman A. R-Trees: A Dynamic Index StructureFor Spatial Searching. In: Proc. ACM-SIGMOD International Conference on Management of Data, 47～57