|  |
| --- |
| 学号：201310311011 |
| **上海海事大学** |
| **本科生毕业设计（文献综述）** |
|  |
|  |
|  |
| **基于路网的最近邻查询技术研究** |
| 学 院：信息工程学院 |
| 专 业：计算机科学与技术 |
| 班 级：计算机 131 班 |
| 姓 名：刘嘉慧 |
| 指导老师：徐明 |
| 完成日期：2017年2月22日 |
|  |

**目录**

[**1 引言 1**](#_Toc31683)

[1.1 研究背景 1](#_Toc16805)

[1.2 研究目的及意义 1](#_Toc28777)

[**2 研究现状 2**](#_Toc30864)

[2.1 问题的定义 2](#_Toc8627)

[2.2 相关工作 2](#_Toc9396)

[2.3 索引结构 3](#_Toc5858)

[**3 经典算法与查询策略 4**](#_Toc11128)

[3.1 Dijkstra算法 4](#_Toc22081)

[3.2 R-Tree算法 6](#_Toc24183)

[3.3 四叉树 8](#_Toc9620)

[**4 实验评估 10**](#_Toc1194)

[**5 进度安排 11**](#_Toc3859)

[**参考文献 11**](#_Toc27145)

# 

# 1 引言

## 1.1 研究背景

目前，随着无线网络和全球定位系统的发展，基于位置的服务（Location Based Services，LBS）[1]在各种地图服务和在线导航系统中有着广泛的应用。一方面，存在大量的具有空间特征的查询：包括驾驶导航、行程规划、路线查询、特定对象（如酒店、银行、超市等）查询、本地新闻查询等。这些查询已经成为人们日常工作和生活的重要组成部分。根据Google的统计，2010年提交给Google搜索引擎的查询中大约三分之一具有空间特征。另一方面，Web中出现大量带有位置属性的信息：包括新闻（如与上海东方明珠广播电视塔相关的新闻）、社交网站中分享的内容（由于用户经常通过移动设备上传图片、视频和个人日志等，这些信息的空间位置特征非常明显）、广告（如位于某地电影院的广告）等。目前，带位置属性的数据逐年增多。随着我国自主构建的北斗卫星定位系统逐步投入实用，这种趋势将愈加明显。这一变化又将进一步促使带有空间特征的查询的增加。

## 1.2 研究目的及意义

近年来，随着全球定位系统（Global Positioning System，GPS）设备的普遍应用，越来越多的人正在享用基于位置的服务（Location Based Service，LBS），它通过获取移动用户的位置信息来为用户提供包括交通引导、位置查询、车辆跟踪、儿童看护、紧急呼叫等众多服务。道路网环境下的最近邻（Nearest Neighbor，NN）查询是其中最为重要的服务类型之一，如“查找距离我当前位置最近的一间超市”，由于切合人们的生活实际中的应用需求，得到了越来越广泛的关注。

人们的日常活动受到现有道路网络（简称路网，通常指公路网络。当然，也可以涵盖铁路网、航空运输网、水路交通网以及其它各种通行途径）的约束。时间与空间是人类活动中两个非常重要的特征。其中，时间线性变化，相对于空间特征而言比较简单。空间特征复杂多变，因人而异，因物而异。

将路网与各类带空间位置属性的信息结合起来，可以形成一个广义的路网。该路网不仅是一个连通的交通网络，同时包含了大量的其它信息：1）路网自身属性信息，如：道路类型、长度、通行时间、限速、单/双向等；2）路边实体空间对象信息，即现实中存在于路边的各类实物及其属性信息，如：一个超市及其描述信息；3）虚拟空间对象的信息，即各种带空间特征的网络信息，如：微博中一张带位置描述的照片或者一则针对某地的广告。虚拟空间对象通常与实体空间对象相关联。我们将实体空间对象和虚拟空间对象统称为空间对象。总括起来，一个空间对象主要包含三方面内容：空间位置，文本描述和其它属性信息。充分利用路网和各类空间对象信息，为广大的用户提供各类有关交通、设施和资源等的咨询服务，不仅具有广泛的应用前景，而且能够创造巨大的社会、经济效益。

# 2 研究现状

## 2.1 问题的定义

1. 路网

路网是一个规模很大的复杂网状结构，它本质上可以看作一个包含大量结点和边的连通图。图中的结点可以是交叉路口、道路端点以及其它根据需要引入的结点；图中的边是连接两个结点的路段（Road segment），为有权边，其权值可以是路段的实际长度、通过该路段的时间或者是速度限制等。此外，路网中包含有大量的实体空间对象，如银行、商店、餐馆、ATM机等。这些实体对象主要包含两方面的信息，一是地理位置信息，二是文本描述信息，并且沿着路网分布。现有的研究一般将路网映射为一个无向图或者有向图（考虑道路中的单/双向通行等因素）。实体对象的地理位置表示也主要分为两种，一种是用对象的实际地理位置（如经纬度）表示，另外一种是采用它与所在路段的端点的相对距离来表示。在现有的路网建模中，实体对象的信息和路网的连通信息是相互独立的（位置关系除外），对二者进行的处理需要分别进行，降低了处理的效率，并且很难支持空间关键词查询。

1. 最近邻查询(Nearest Neighbor, NN)

最近邻查询指对于一个给定的对象s，在时间段 T 内找出距离s最近的目标对象 t,例如查询距离某行驶的小汽车最近的警车。最近邻查询是当 K=1，K 最近邻查询的特殊情况。当查询的对象是 K 个而不是最近的那个时，该查询叫做 K 近邻查询[2-4]。

## 2.2 相关工作

路网信息是一个规模很大的数据集，为了实现高效的查询处理需要为其构建合适的索引结构。用R-树索引（B-树在多维空间中的扩展，将距离较近的对象划分到相同的结点中）所有的路段，将每个路段表示成一个最小边界矩形（Minimum Bounding Rectangle, MBR）。同时，为每个路段上的实体对象根据其与路段端点的距离构建一颗B-树。此外，还有其它一系列的索引构建思想，如：对路网进行层次划分[5]；将结点映射到高维空间中，并采用棋盘距离进行近似计算；用Voronoi图将网络分割成Voronoi区域[6]。

空间数据库是用来储存 ,管理各种空间或非空间的信息 的计算机信息应用系统。 其管理带有空间属性信息的能力是 一般传统的关系型数据库所不具备的。 一个信息系统不论是 一般的关系型数据库还是空间数据库 ,其一项根本的任务就 是信息的检索查询。 能否快速的检索信息是数据库性能高低 的一个主要的标志。 传统关系型数据库用 B+ 树为数据建立 索引 ,来提高检索的效率。 但 B+ 树是一维索引 ,无法处理空 间数据库中的二维和多维的空间数据 ,必须另外为空间数据 建立专门的索引机制。

由于空间索引在整个空间数据库中的重要地位 ,一直是 研究的重点。 迄今为止人们提出了 K- D 树、 K - D - B树、BSP树、 R树系列、四叉树、网格等众多空间索引。其中主要常 用的为 R树系列和四叉树系列。 目前国内外主要的空间数据 库大都采用这两类空间索引方法。

## 2.3 索引结构

按照构建索引方式的不同，现有针对空间关键词查询的索引结构主要有两类，一类是独立索引[7-8]，即分别为空间信息和文本信息构建索引；另外一类是混合索引[9-10]，将空间索引和文本索引结合起来。

独立索引是将传统的倒排文件、签名文件等文本索引技术和具有空间过滤功能的Quad树、R树及其变种、网格索引（Grid Index）等空间索引技术分别用于对文本信息和空间信息构建索引，然后分阶段地用于查询。总的来说，基于独立索引的空间关键词查询的效率低，不适合数据规模大、查询响应要求快的情况。

目前，空间关键词查询处理采用的多为混合索引，即将空间索引和文本索引有机地结合起来，构建统一的索引结构。混合索引根据结合的紧密程度又可以分为松散混合索引和紧密混合索引。

早期的混合索引只是将空间索引和文本索引进行松散结合，如[9-10]中的索引方法为每个关键词建立了一棵R\*-树。在查询过程中，首先根据查询关键词找出对应的R\*-树，然后遍历R\*-树查找最终结果。这种方法对单关键词查询非常有效，所需访问的数据量较少（除非包含该关键词的对象很多）。针对这种索引结构进行了改进，考虑了每个关键词包含的对象的个数，只有当某个关键词包含的对象数目超过一定的阈值时才为该关键词建立树形索引，否则直接存储为文件块。这种策略在一定程度上降低了存储开销。[7]中给出了另外一种松散的混合索引，其思想是将数据空间进行划分，然后为每个划分创建倒排索引。这种方法的局限在于难以合理地确定空间划分的粒度。

# 3 经典算法与查询策略

## 3.1 Dijkstra算法

计算公路网络中两点之间的最短路径是一个非常重要的问题，目前在各种地图服务和商用在线导航系统中有广泛的应用。经典的Dijkstra算法[11-12]能够高效的计算出路网中任意两点的最短路径。给定路网G中的起点s和终点t，Dijxstra算法按到s点距离的远近顺序依次访问其它点，直至到达终点t时结束。该算法简单精确，但是却并不适用于大型的公路网络。因为该算法要访问终点t到起始点s的距离范围内所有的点，当两点之间的距离非常远时，Dijkstra算法要访问的点数会非常的庞大。

Dijkstra算法思想为：

设G=(V,E)是一个带权有向图，把图中顶点集合V分成两组，第一组为已求出最短路径的顶点集合（用S表示，初始时S中只有一个源点，以后每求得一条最短路径 , 就将 加入到集合S中，直到全部顶点都加入到S中，算法就结束了），第二组为其余未确定最短路径的顶点集合（用U表示），按最短路径长度的递增次序依次把第二组的顶点加入S中。在加入的过程中，总保持从源点v到S中各顶点的最短路径长度不大于从源点v到U中任何顶点的最短路径长度。此外，每个顶点对应一个距离，S中的顶点的距离就是从v到此顶点的最短路径长度，U中的顶点的距离，是从v到此顶点只包括S中的顶点为中间顶点的当前最短路径长度。

Dijkstra算法举例说明 如下图，设A为源点，求A到其他各顶点（B、C、D、E、F）的最短路径。线上所标注为相邻线段之间的距离，即权值。（注：此图为随意所画，其相邻顶点间的距离与图中的目视长度不能一一对等）

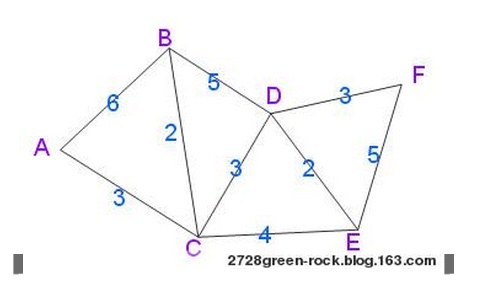


图1：Dijkstra算法举例说明

表1：Dijkstra算法举例说明

|  |  |  |
| --- | --- | --- |
| 步骤 | S集合 | U集合 |
| 1 | 选入A，此时S=<A>  此时最短路径A->A=0  以A为中间点开始找 | U=<B、C、D、E、F>  A→B=6，A→C=3，A→其它U中的顶点=∞  发现A→C=3权值为最短 |
| 2 | 选入C，此时S=<A、C>  此时最短路径A→A=0 A→C=3  以C为中间点，从A→C=3这条最短路径开始找 | U=<B、D、E、F>  A→C→B=5(比上面第一步的A→B=6要短)  此时到D权值更改为A→C→B=5， A→C→D=6， A→C→E=7，  A→C→其它U中的顶点=∞，  发现A→C→B=5权值为最短 |
| 3 | 选入B，此时S=<A、C、B>  此时最短路径A→A=0 A→C=3，A→C→B=5 以B为中间点  从A→C→B=5这条最短路径开始找 | U=<D、E、F> A→C→B→D=10(比上面第二步的A→C→D=6要长)  此时到D权值更改为A→C→D=6， A→C→B→其它U中的顶点=∞，  发现A→C→D=6权值为最短 |
| 4 | 选入D，此时S=<A、C、B、D>  此时最短路径A→A=0 A→C=3，A→C→B=5，A→C→D=6 以D为中间点， 从A→C→D=6这条最短路径开始找 | U=<E、F> A→C→D→E=8(比上面第二步的 A→C→E=7要长) 此时到E权值更改为A→C→E=7，A→C→D→F=9  发现A→C→E=7权值为最短 |
| 5 | 选入E，此时S=<A、C、B、D、E>  此时最短路径A→A=0 A→C=3，A→C→B=5，A→C→D=6，A→C→E=7，以E为中间点， 从A→C→E=7这条最短路径开始找 | U=<F>  A→C→E→F=12(比上面第四步A→C→D→F=9要长)此时到F权值更改为 A→C→D→F=9 发现A→C→D→F=9权值为最短 |
| 6 | 选入F，此时S=<A、C、B、D、E、F> 此时最短路径A→A=0，A→C=3， 选入F，此时S=<A、C、B、D、E、F> 此时最短路径A→A=0，A→C=3， A→C→B=5 A→C→D=6， A→C→E=7，A→C→D→F=9 | U集合已空，查找完毕 |

## 3.2 R-Tree算法

R树在数据库等领域做出的功绩是非常显著的[13-14]。它很好的解决了在高维空间搜索等问题。例如查找20英里以内所有的餐厅。如果没有R树一般怎么解决？一般情况下我们会把餐厅的坐标(x,y)分为两个字段存放在数据库中，一个字段记录经度，另一个字段记录纬度。这样的话就需要遍历所有的餐厅获取其位置信息，然后计算是否满足要求。如果一个地区有100家餐厅的话，我们就要进行100次位置计算操作了，如果应用到谷歌地图这种超大数据库中，这种方法便没有可行性了。

R树就很好的解决了这种高维空间搜索问题。它把B树的思想很好的扩展到了多维空间，采用了B树分割空间的思想，并在添加、删除操作时采用合并、分解结点的方法，保证树的平衡性。

RTree作为一种空间索引主要具有以下一些特性：

（1）R-Tree是n 叉树，n称为R-Tree的扇（fan）。

（2）每个结点对应一个矩形。

（3）叶子结点上包含了小于等于n 的对象，其对应的矩为所有对象的外包矩形。

（4）非叶结点的矩形为所有子结点矩形的外包矩形。

R-Tree的定义很宽泛，同一套数据构造R-Tree，不同方法可以得到差别很大的结构。因此具有两标准：

（1）位置上相邻的结点尽量在树中聚集为一个父结点。

（2）同一层中各兄弟结点相交部分比例尽量小。

R树是一种用于处理多维数据的数据结构，用来访问二维或者更高维区域对象组成的空间数据.R树是一棵平衡树。树上有两类结点：叶子结点和非叶子结点。每一个结点由若干个索引项构成。

算法描述如下：

对象数为n，扇区大小定为fan。

（1）估计叶结点数k=n/fan。

（2）将所有几何对象按照其矩形外框中心点的x值排序。

（3）将排序后的对象分组，每组大小为 \*fan，最后一组可能不满员。

（4）上述每一分组内按照几何对象矩形外框中心点的y值排序。

（5）排序后每一分组内再分组，每组大小为fan。

（6）每一小组成为叶结点，叶子结点数为nn。

（7）N=nn，返回1。

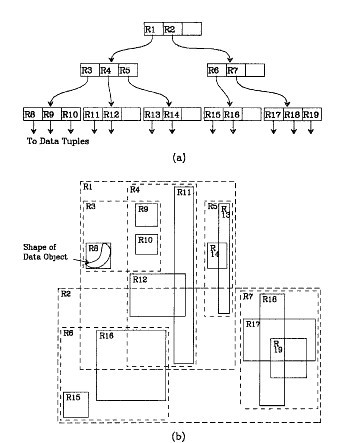


图2：R-Tree算法举例说明

## 3.3 四叉树

四叉树是另一类常见的空间索引[15-16]。与 R树系列不同的是 , 它是属于基于空间划分组织索引结构的一类索引机制。 将已 知范围的空间划成四个相等的子空间。 如果需要可以将每个 或其中几个子空间继续划分下去。 这样就形成了一个基于四 叉树的空间划分。 四叉树索引有满四叉树索引和一般四叉树 索引。 下面的两个四叉树空间索引都是满四叉树索引。

1. 基于固定网格划分的 CELLQTREE

CELLQTREE 是 满四 叉树 空间索 引。 N 层 CEL-LQTREE的叶子结点所对应的子空间将形成一个 2n\* 2n 的网格。 图 2是 N= 2时的示意图。 空间对象的 ID 记录在经所覆 盖的每一个叶子结点中。 但当同一父亲的四个兄弟结点都要 记录该空间对象 ID时 ,则只将该空间对象 ID只记录在该父 亲结点上。并按这一规则向上层推进。如图中 R1 同时经过 5、 6、 7、 8四个兄弟子空间 ,根据上面的规则只需在其父亲结点 1 号结点中记录 R1 的 ID。 CELL Q T REE的构成方式与网格索 引有些类似 ,都是多对多的形式。一个网格可以对应多个空间 对象 ,同时每个空间对象也可以对应多个网格。但与网格索引 不同的是它有效的减少了大空间对象在结点中的重复记录。

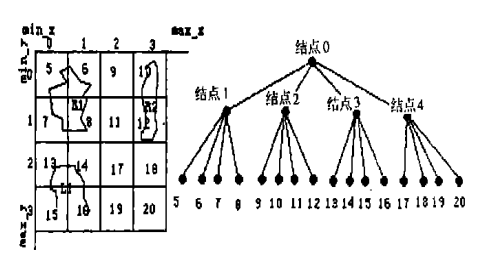


图3：四叉树算法举例说明

CEL L Q T REE的空间对象的插入和删除都较简单。只需在其覆盖的叶子结点和按照上面的规则得到父亲和祖先结点 中记录或删除其 ID 即可 ,没有象 R树一样的复杂耗时的分裂 和重新插入操作。 这是四叉树优于 R树的一个地方。 查询方 式也比较简单。 例如要检索某一多边形内和与其边相交的空 间对象 ,只需先检索出查询多边形所覆盖的叶子结点和其父亲及祖先结点中所有的空间对象 ,然后在进行必要的空间运 算从中检索出满足要求的空间对象。

由于是满四叉树 , CEL LQ T REE采用顺序的数组存储方 式。内存耗费是链表结构的四分之一。其索引结构可以放在内 存中 ,因此无需 I /O方面的耗费。

2. SuperMap的线性可排序四叉树空间索引

Supe rM ap 的可排序四叉树空 间索引与 CEL-L Q T REE一样也是满四叉树空间索引。 但有两点不同的是: ( 1)结点和空间对象的对应关系不同; ( 2)四叉树结点编码方式不同。

线性可排序四叉树结点和空间对象的对应关系为一对 多。一个结点可以对应多个空间对象 ,而一个空间对象只能对 应于一个结点。 它将空间对象记录在包含它的最小子空间所 对应的结点中。 这样可免除由于多对多机制所带来的查询时 需要排重等麻烦。但也带来了精度不够高的缺点。如图 2中空 间对象 R2 在 CEL LQ T REE中记录在 10和 12结点中 ,在线 性可排序四叉树中则记录在其父结点 2号结点中。 当查询区 域与 2结点所对应的子空间相交但不包含 10和 12号结点所 对应的字空间时 ,在线性可排序四叉树中 ,不在查询区域内的 R2 将会被检索出来需要进行进一步的空间判断。 而在 CEL-L Q T REE中则不会出现这样的问题。

线性可排序线性四叉树在编码上放弃了 C EL L Q T R EE 的传统的线性四叉树编码方式 ,采用了其独特编码方式。 图 3 是既划分两层的线性可排序四叉树的编码的示意图。 首先将 四叉树变为二叉树;即在父结点层与子结点层之间插入一层 虚结点 (其不用来不记录空间对象 )。 然后按中序遍历全树的 顺序对结点进行编码 ,包括加入的虚结点。这样编码方式使的 孩子结点和父亲结点号码相连 ,在储存的时候某一结点为根的子树的所有结点为连接的一段 ,不象分散的许多小段。 我们可以用关系型数据库中的二维表来实 现四叉树的空间索引 ,只要在存放空间对象二维表中增加一 个字段用来存放记录空间对象所在结点的编号。 进行空间查 询的时候 ,先根据查询区域生成所要搜索的结点编号的集合。 由于新编码方式的孩子与父亲结合编号的连续性将结点编号 集合变成若干个连续的结点编号的范围 ,这样就可以用 SQ L 语句从表中检索出满足要求的空间对象。 而采用传统的编码 方式四叉树划分较深 ,查询时涉及的结点量巨大的时候 ,很容 易使检索的 SQ L 语句超过允许的长度。但可排序线性四叉树 的编码方式也有其不足的地方。当四叉树结构发生变化时候 , 如向下再划分一层 ,则必须给所有的结点重新编码。这使的可 排序线性四叉树缺少一定的灵活性。 而采用传统编码的四叉 树就没有这方面的问题。

从上面两个四叉树空间索引可以看出 ,这类四叉树空间 索引与 R树相比有如下两个优点: ( 1)可以用顺序存储的线性 表来表示索引 ,内存需求量小 ,可以在内存中实现。 这样无须 I /O 方面的花费 ,使得查询速度得到提升。 ( 2)插入和删除操作简单方便 ,平均耗时远小于 R树的插入和删除的花费。但由 于其是以空间划分来组织索引结构的索引机制 ,有着这类索 引机制的共同的问题: 在建立索引之前必须预先知道空间对 象所分布的范围 ,可调节性比较差。

# 4 实验评估

实验计划在Windows7平台上使用Visual C++ 6.0实现，所有的PC配置为：Intel Core2 CPU,2.50GHz,2GB内存，500GB硬盘。实验数据采用Oldenburg[17]城市的真实路网的数据，该道路网的节点的个数为6105，道路边共有7035条，数据点是随机挑选的道路网络中的节点，用来描述不同类型的对象节点。

# 5 进度安排

|  |  |  |
| --- | --- | --- |
|  | 毕业设计（论文）各阶段任务： | 起 止 时 间 |
| 1 | 查找相关书籍和论文资料，了解其基本原理。 | 2月20日-2月26日 |
| 2 | 了解并熟悉路网建模及其最近邻查询技术，建立路网模型。 | 2月27日-3月5日 |
| 3 | 探索路网最近邻查询技术算法并分析算法性能。 | 3月6日-3月12日 |
| 4 | 准备中期答辩，完成文献综述以及答辩PPT。 | 3月20日-3月27日 |
| 5 | 完成论文初稿的撰写。 | 3月13日-4月14日 |
| 6 | 完成论文的撰写，在老师的指导下修改，准备终期答辩。 | 4月15日-5月1日 |

# 参考文献

[1]**Saltenis S, Jensen C S. Indexing of Moving Objects for Locationbased Services[C]//Proc. of the 18th International Conference on Data Engineering. Washington D. C., USA: IEEE Computer Society,2002: 463-473.**

**[2]李淑芳，胡克寒，张凤丽，等． 基于 SSH2 框架的高校科研网络管理系统的实现**[J].**计算机应用与软件，2010，27( 7) : 195-197．**

**[3]朱亚兴，张延政． 基于 SSH 架构的移动医疗应用服务模型及实现**

[J].**计算机应用与软件，2012，29( 6) : 172-175．**

**[4]吕焕群，翁将锋． 基于 WEB 的大型 Oracle 应用系统性能优化方法**

**研究**[J].**计算机应用与软件，2012，29( 5) : 184-186．**

[5]WANG Dandan, HAO Zhongxiao. Multi-type K nearest neighbor queries in road networks.[J].Computer Engineering and Applications,2012, 48（3）：140-142.

[6]Sharifzadeh M，Shahabi C.Processing optimal sequenceed route queries using voronoi diagrams[J].Geoinformatica，2008，12：411-433.

[7]K.S. McCurley. Geospatial Mapping and Navigation of the Web.WWW.2001:221~229

[8]Y.-Y,Chen,T.Suel,and A.Marowetz. Efficient Query Processing in Geographic Web Search Engines.SIGMOD,2006:277~288

[9]I.D.Felipe,V.Hristidis,andN.Rishe.Keyword Search on Spatial Databases.ICDE,2008:656~665

[10]G.Cong,C.S.Jensen,and D,Wu,Efficient Retrieval of the Top-k Most Relevent Spatial Web Objects.PVLDB,2009:337~348

[11]Edsger Wybe Dijkstra. A note on two problem in connection with graphs. Nu-merical Mathematics,1:269-271,1959.

[12]Ira Pohl. Bi-directional search. Machine Inteligence,6:127-140,1971

[13] A. Guttman, R-Trees a dynamic index structure for spatial search. Proc ACM SIGMOD, 47- 57, 1984

[14]N.Beckman,H.P.Kriegel,R.schneider,B.seeger,The R-tree an efficent and robust acess method for points and rectangels ACM SIGMOD,322-331,1990

[15]Samet Hanan. Foudations of Multidimensional and Metric Data Structures.Morgan-Kaufmann, San Francisco.

[16] H. Samet, J. Sankaranarayanan, and H. Alborzi, “Scalable Network Distance Browsing in Spatial Databases,” Proc. 2008 ACMSIGMOD Conf., pp. 43-54, 2008.

[17]BRINKOFF T．A framework for generating network based moving objects [J].Geoinformatica，2002，6( 2) : 153 － 180．