

基于 Patricia 树的空间索引结构

易显天, 徐 展, 郭承军, 刘 丹, 张 可

(电子科技大学电子科学技术研究院, 成都 611731)

摘 要: 针对空间索引响应近邻查询效率低的问题, 基于二进制 Morton 码和 Patricia 树, 提出一种一维空间索引结构。通过改良 Patricia 树结构及其相关算法提高索引结构的操作效率。基于 Morton 码特点, 融合索引结构和 Morton 码, 使得索引结构拥有高效响应近邻查询的能力, 并同时提出基于 MPT 的近邻算法。将二维空间进行预定规则下的不同粒度的划分, 把分块后的二维空间区域转换为一维编码, 使 MPT 索引具备高效响应区域查询能力。分析区域查询误差出现的原因, 并给出相应解决方案。实验结果表明, 与 B+ 树、Hash 表、Trie 树相比, 该方法在查询速度上更具优势, 基于 MPT 的近邻搜索比基于 R-Tree 近邻搜索效率更高。

关键词: Patricia 树; Morton 码; 近邻搜索; 空间索引; 区域查询

中文引用格式: 易显天, 徐 展, 郭承军, 等. 基于 Patricia 树的空间索引结构[J]. 计算机工程, 2015, 41(12): 69-74.

英文引用格式: Yi Xiantian, Xu Zhan, Guo Chengjun, et al. Spatial Index Structure Based on Patricia Tree[J]. Computer Engineering, 2015, 41(12): 69-74.

Spatial Index Structure Based on Patricia Tree

YI Xiantian, XU Zhan, GUO Chengjun, LIU Dan, ZHANG Ke

(Research Institute of Electronic Science and Technology,

University of Electronic Science and Technology of China, Chengdu 611731, China)

【Abstract】 Aiming at low efficiency problem of spatial index that answers nearest neighbor query and other kind of query, this paper proposes an index structure called Morton Patricia Tree (MPT) which is based on Patricia tree and binary Morton code. It improves the efficiency of MPT index structure by reforming structure of patricia tree and relative algorithm. Based on the characteristics of Morton code, it integrates index structure and binary Morton code to meet the need of neighbor query, and proposes the nearest neighbor search algorithm based on MPT at the same time. MPT gains the capability of region query by dividing the two-dimensional space into different size cubes according to predefined rules and converting each cube into Morton code. The paper analyzes the reason why range query error occurs and puts forward the solution. Experimental result shows that the search speed of MPT index structure is faster than that of B+ tree, Hash table and trie tree. The nearest neighbor query based on MPT is more efficient than that based on R-Tree.

【Key words】 Patricia tree; Morton code; neighbor search; spatial index; region query

DOI: 10.3969/j.issn.1000-3428.2015.12.014

1 概述

移动对象数据库是近年发展起来的数据库研究分支,也是基于位置相关服务一项重要的技术,目前国内外的空间索引技术涌现,但总体来说该领域的研究与发展仍然处于未成熟阶段^[1-2]。

目前移动对象索引研究可以大概分为 2 类:

(1) 基于空间划分的索引结构^[3-4], 如 B+ 树和网格。

(2) 基于数据的划分的索引结构^[5-7], 如 R-Tree 和其相关的演化结构。基于空间划分的索引结构中有一类是一维索引结构, 相较于二维索引, 一维索引发展得更为成熟, 适用范围更广, 对空间数据进行降维处理, 可以重复利用一维索引技术及其相关技术。常见的降维方法是利用空间填充曲线, 将二维空间的信息映射到一维空间上^[8-10]。

目前的索引结构大都不能提供高效的近邻查

基金项目: 宁波市重点科技攻关计划基金资助项目(2011C51007)。

作者简介: 易显天(1989-), 男, 硕士, 主研方向: 空间数据库技术; 徐 展, 高级工程师、硕士; 郭承军, 博士; 刘 丹, 副教授、博士; 张 可, 副研究员、博士。

收稿日期: 2014-12-26

修回日期: 2015-01-22

E-mail: yixiantian7@foxmail.com

询,或者需要复杂结构和算法,以至于结构实现困难。本文提出一种基于 Patricia 树的索引结构 MPT (Morton Patricia Tree),融合 Morton 码、Patricia 树结构和算法,使得 MPT 拥有高效响应的近邻查询能力,且 Morton 码比 Geohash^[11] 编码对二维空间划分有更好的精度控制。同时提出基于 MPT 的近邻算法。通过对二维空间的进行预订规则下不同粒度划分,把分块后的二维空间区域转换为一维编码,使得 MPT 具备快速响应范围查询能力。

2 Patricia 树和二维空间编码方案

2.1 Patricia 树

Patricia 树是一种一维树形索引结构,可以用来存储字符串。结构中只有叶子结点保存字符串信息,中间结点代表其子孙结点中的字符串的某公共前缀,但并不存储字符串而是保存的不同字符串之间的区分信息。索引结构中父亲结点的代表的字符串是其子结点代表字符串的前缀,兄弟结点所代表的字符串具有公共前缀,Patricia 可以使用基于二进制位的方式构建,适用于以 Morton 码为观察对象的查询操作。

2.2 编码原理

二进制 Morton 编码的基本思想是不断交替在经纬度 2 个方向对地球表面进行二分。一直循环划分到任意精度。具体为:当沿着经度方向进行二分,左侧区域编码为 0,右侧区域的编码为 1;当沿纬度方向进行二分,下侧区域编码为 0,上侧区域编码为 1。

经纬度在数次二分之后,把指定位置对应的二进制数从范围由大到小按序排列,可得到对应位置经纬度的编码,例如纬度产生编码:1011100011,经度产生编码,1101001011。将经度代表的编码放在偶数位,将纬度代表的编码放在奇数位,重新组合经纬度的编码可得到新的编码:111001110100100011。这也就是最终使用到的 Morton 编码。

而 Geohash 编码则需要继续对上述编码过程中得到的二进制码进行 Base32 编码转换。5 位二进制码产生一位 Geohash 编码。当 GeoHash 字符串长度为奇数时,此码值代表一个长宽不相等的矩形。另外,由于 Geohash 一个字符代表 5 bit,这意味着位置精度的控制是以每 5 bit 增减,存在着较大精度的浪费。使用二进制 Morton 码,可获得更细粒度的精度。Geohash 和 Morton 编码精度对比如图 1 所示,从图中可以看出,Morton 编码可以达到的面积划分粒度要比 Geohash 细,其中, $pow(2, x)$ 表示 2 的 x 次方,精度可以达到 $A/Pow(2, x)$ (A 为全部面积)。

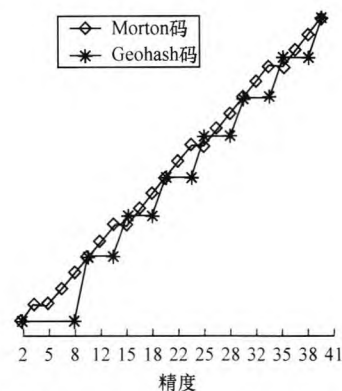


图 1 Morton 码与 Geohash 码精度对比

2.3 编码特点

通过 Morton 编码方案可知其编码具有以下特点:

- (1) 唯一性,根据编码过程中区域划分规则,地球表面在划分过程中的每个单元格都有唯一的编码与其对应。
- (2) 每一个编码并不是表示一个二维点坐标,而是表示一个二维矩形区域。区域的大小由编码长度控制。故编码的前缀可以表示更大范围。
- (3) 每个编码同时表示经度和纬度 2 个坐标。

3 MPT 索引结构

3.1 MPT 索引

使用 Morton 码将二维位置信息转化为一维编码,以 2 个编码相同前缀的长短来表示 2 个二维位置的远近,Morton 码不仅保留了移动对象二维位置的邻近关系,另一方面也能很好地反映不同大小的二维空间区域之间的从属关系。Patricia 树父子结点分别代表的字符串具有前缀对应关系,这也恰好适于 Morton 码前缀匹配操作。基于上述考虑,将移动对象二维位置信息转换为二进制 Morton 码并通过 MPT 插入算法构建索引结构。并改良索引结构和相关算法使 MPT 具有高效响应空间查询的能力。

3.2 Bitwise 操作

MPT 的整个结构是以二进制的位操作为原理构建的,这样同时也能节省空间和加快操作效率。因为在计算机中对位的操作更快,所以树将字符串进行“二进制”化再插入到索引结构中。

MPT 子结点保存的是插入 MPT 的二进制 Morton 码串,非叶子结点保存的是 Morton 码串前缀对应的“状态”,这些状态是 Morton 码串在插入过程中根据不同 Morton 码串基于位的比较而得到的,只有 2 个码串出现二进制位的不同时,才会出现一个新的结点,来保存这个“分歧”状态,而那 2 个码串也因为这个“分歧”的二进制位的 0 或 1

而由这个结点的2个不同的子结点方向各自向下继续比较。也就是说对整个树的操作也就类似于对二叉树的操作。在索引结构中的搜索路径的选择是根据对应位的0或1来决定的。分析同样具有前缀搜索能力的trie树的结构,可以发现trie树结点保留字符串中相同的字符,每条检索路径都是字符串的前缀。如果改变结点本身强调的内容,将字符串之间的不同点的信息保留在分支结构中,那么就可以把所有相同的字符串匹配省略,减少了所需存储信息的数量。

3.3 MPT 结点结构

在实验过程中发现,结点的容量有些许的增加,就会带来整个索引结构体积的大量增加,故在保证索引结构操作效率的情况下,使结点中保存的信息尽量少,以减少空间占用。MPT索引结构如图2所示。

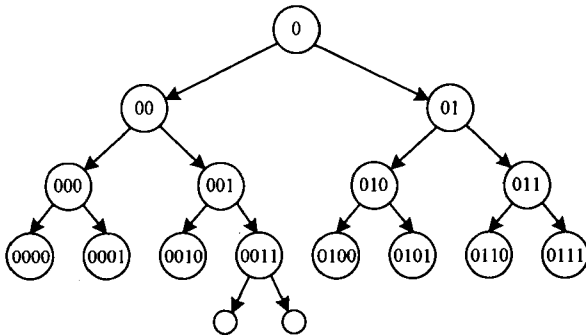


图2 MPT索引结构

非叶子结点结构为 $(Ptr[2], pos, mask)$,其中, Ptr 是指向子结点的指针,每个结点固定包含2个指针; Pos 是结点对应字符在整个字符串中的位置; $mask$ 是一个用于位操作的8位二进制码,用来获取比较的字符的最高不同位。叶子结点的结构为 $(Ptr, MojInfo)$,其中, Ptr 是指向Morton码串的指针; $MojInfo$ 是一个结构体数组。每个数据项结构为 (ID, x, y) ,其中, ID 为移动对象的标识号; x, y 为移动对象的二维位置。为了节省空间,本文使用指针变量的最低有效位来表示这个指针是指向内部结点还是指向叶子结点。大致结构如图2所示。为了示意,画出了一个“满”MPT,但是实际如果没有出现对应的字符串对比。则不会出现因“分歧”而产生的对应的结点。

3.4 查询算法

MPT查询操作先将需要查找的字符串进行“二进制”化,然后根据搜索过程中遇到结点的顺序对相关二进制位进行比较,即在MPT树中进行二分查找,直到查找到对应字符串并进行移动对象信息对比,相同则返回True,不同或查找失败则返回False。伪代码如下:

输入 MPT结构指针 T ,指向字符串指针 u ,移动对象 ID ,移动对象二维位置信息 x, y

输出 True/False

Begin

$len = strlen(u)$; //获取输出Morton码长度

$node * p = T.root$; //获取MPT结构根结点

While(p 是内部结点)

$node * q = (node *) (p - 1)$; //减1获得指向结点指针
// $direction = tellDirection()$ // 以结点 pos 和 $mask$ 参数对Morton码的差异位进行判断,并获取继续比较的子结点方向
// 向 $direction$

$update(p)$; //更新 p ;

$return compare()$; //判断子结点中存储的移动对象信息是否和输入的移动对象信息相同

End.

3.5 算法插入

整个MPT构建过程主要依赖插入算法,由于结构特点,对于同样的Morton码集的输入,对应产生MPT也是完全相同的,因此MPT在操作过程中不需要实现复杂的平衡算法。并且整个MPT结构的高度由输入的字符串的最大长度来决定,而不是由输入字符串的数目来决定。插入操作先处理时空树的情况,然后使用类似于上面描述的搜索方法来查到合适的插入位置,计算相关的 pos 和 $mask$ 参数,然后生成对应新的结点,最终将结点插入。具体算法的伪代码如下:

输入 MPT结构指针 T ,指向字符串指针 u ,移动对象 ID ,移动对象二维位置信息 x, y

输出 True/False

Begin

$len = strlen(u)$; //获取输出Morton码长度

$node * p = T.root$; //获取MPT结构根结点

If($! p$) //如果是空树

以 u 指向的字符串直接构建新结点,并插入

End If

While(p 是内部结点)

对MPT结构进行搜索,找到合适插入 u 的结点位置 loc ;

$config(pos, mask)$; //判断 loc 中存储的Morton码与插入 u 指向的Morton码之间有无区别,如果有,通过位操作计算出新插入字符串对应结点中的 pos 和 $mask$ 参数,并进行设置,若无则直接返回False。

$init(newnode)$; //初始化一个结点的空间,并使其拥有2个子结点指针,结点中保存 $mask$ 和 pos 信息,结点的2个子结点指针其中一个指向移动对象信息构建的子结点,另一个子结点指针 p 保留在将 $newnode$ 插入MPT的过程中使用

$insert(newnode)$; //搜索至插入位置 $newloc$, $newnode$ 代替 $newloc$ 中结点位置

$set(newnode)$; //将 p 另一指针指向原 $newloc$ 处的结点
End.

4 高级查询

4.1 近邻查询

根据 Morton 编码特点可以知道如果 2 个点的距离越近,那么 2 个点位置对应的编码的共同前缀则越长,这样可以通过前缀的比较来进行二维空间的近邻搜索,为了消除在搜索过程中的遗漏,除了使用定位的编码匹配外,还使用周围 8 个区域的编码进行搜索匹配。在近邻搜索算法中每一次循环都在前一次搜索范围外增加一层以目前 Morton 码长所表示的范围的搜索区域。这样可以有效利用上一步的搜索结果。伪代码如下:

```

    输入  纬度 lat, 经度 lon, MPT 结构指针 MPptr,
    近邻搜索目标数 num, 编码匹配范围 extent
    输出  num 个近邻对象的 ID: result
    Begin
        center = MortonEncode( lat, lon, extent) // 计算搜索中心的
        // Morton 码
        searchrange = getNeighbour( center) // 获得中心周围的
        // 8 个搜索框
        Init( resultTemp ), init( resultVec) // 初始化结果集
        While(1)
            res - Num = MPptr. search( searchrange, extent, resultTemp)
            // 搜索, 并获得结果数
            If( resNum < num) // 结果数小目标数
                resultVec = append( resultTemp) // 获取结果
                num = num - resNum // 更新目标数
            Else // 结果数大于目标数
                resul - tTemp = distanceCompare( num, resNum,
                resultTemp) // 进行距离比较
                resultVec = append( resultTemp);
                break // 跳出循环
            End If
            update( searchrange) // 更新搜索现有搜索范围 return
        // resultVec;
    End.

```

4.2 区域查询

对查询区域进行如图 3 所示的划分,可将查询区域分解成多个查询方块,为了能够更好地覆盖整个查询区域,需要对查询区域划分为大小不同的方块。实验过程中发现,将查询区域划分的方块越小,则对查询区域有更精确的覆盖,查询精度更高,但是在划分粒度超过一定范围以后,得到精度上的提升较小,反而由于划分粒度过小,得到的查询方块数量急剧增多,使得查询性能大大下降,于是选择恰当的划分粒度对查询区域进行划分。Morton 码长最长不超过 32 bit, 同一个查询区域在 Morton 码匹配过程中, Morton 码长度种类不超过 5 种,且在面积全匹配的情况下,尽量使用更短的 Morton 码来表示一个区域。二维空间划分方式如图 3 所示。

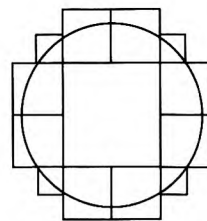


图 3 二维空间划分

在整个编码过程中,是将一个二维平面进行划分,但地球表面是一个球体曲面,需将地球表面通过墨卡托(Mercator)投影来得到这样一个二维平面,在投影的过程中,不同纬度上的地球表面在投影过程中会发生不同程度的形变。那么得到的不同纬度上的划分后的方块也会发生不同程度的形变。这样在用编码所表示方框匹配以中心点+距离来表示区域查询的时候,会带来匹配面积上的误差,且纬度越高,则误差越大。通过在查询面积匹配过程中在匹配面积前乘以一个调节因子 $\cos(\text{latitude} * \pi/180)$ 来消除这种查询误差,实验结果证明效果良好。

5 性能评估

索引结构采用 C++ 实现,实验平台为: Windows XP 系统、英特尔酷睿双核 3.2 GHz 处理器、2.96 GB 内存。使用 Brinkhoff^[12] 的移动对象生成器生成动态移动对象数据,如图 4 所示。数据包含了一定数量的移动对象在一段时间内位置更新的相关信息,数据内容包括移动对象 ID、此次位置更新时间、位置 X 坐标(经度)、位置 Y 坐标(纬度)等。

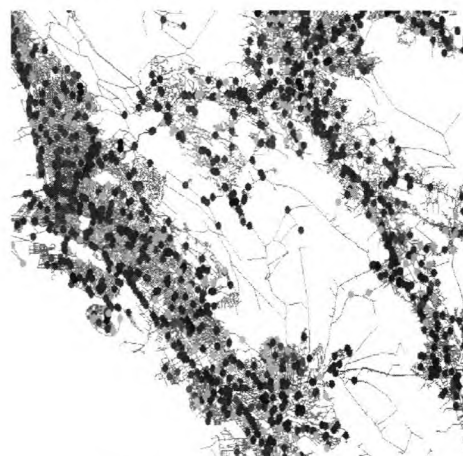


图 4 地理数据生成的地图

5.1 时间效率

将 MPT 与 B+ 树、Hash 表、Trie 树进行查询效率的比较。将生成的移动对象的位置的经纬度数据转换成 32 位二进制 Morton 码,构建索引结构,进行查询效率的比较。和 B+ 树不同, MPT 的普通操作,例如查询、插入、删除需要 $O(K)$ 的时间(K 表示字符串长度),而 B+ 树中这些操作需要 $O(\log n)$ 的时间(n 为字符串数),一般情况下 $O(k) > O(\log n)$,但是在 B+ 树中对于字符串的比较最差情况需要

$O(K)$ 的时间复杂度,当字符串的长度较长,则会明显增加 B+ 树中普通操作所花费的时间,这也是为什么 B+ 树的效率相对不高的原因。而在 MPT 中,搜索过程中所有的比较操作所消耗的时间都是常数时间。哈希表的插入、查找、删除的时间为 $O(1)$,但这是以计算哈希值的过程为常数时间当作前提的,当考虑了哈希值的计算时间,则需要 $O(k)$ 的操作时间,加上如果在操作过程中发生哈希值的碰撞,则需要消耗更多的时间。而 MPT 则最坏时间为 $O(K)$ 。而实验结果也证实了这一点。MPT 比哈希表的效率更高。MPT 相对于 Trie 的优势则体现在 2 种结构的特点,具体原因如 3.2 节所述。实验结果如图 5 所示。由于插入操作和删除操作的大部分时间消耗都是用于查询,因此得到的结果与查询效率对比实验类似,故此不再赘述。

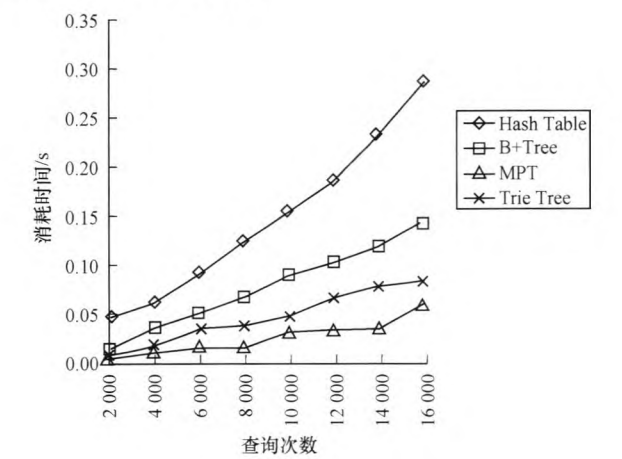


图 5 查询效率对比

图 6 显示了基于 MPT 的近邻搜索算法与基于 R-Tree 的近邻搜索算法效率的对比结果,因为 MPT 的搜索是根据二进制位的比较,而基于 R-Tree 的搜索算法是需要 MBR (Minimum Bounding Rectangle) 的比较,相比之下后者需要花费更多时间,并且 R-Tree 存在 MBR 互相重叠的问题,尤其处理大数据量时,基于 R-Tree 的近邻查询算法性能急剧恶化,而 MPT 因其结构的特性并没有上述问题。实验结果表明,基于 MPT 的近邻查询算法效率更高。

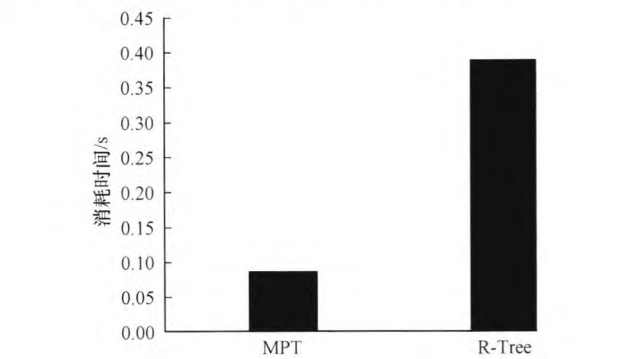


图 6 近邻搜索效率对比

5.2 空间占用

Morton 码的长度越长,则表示的位置越精确,但是使用 MPT 在近邻搜索的过程中,使用过长的 Morton 码对 MPT 近邻搜索的过程并没有帮助,反而因为过长的码长而导致 MPT 中的结点数过多,所以会产生更多空间的占用。

表 1 表示经纬度某方向经过一定次数的划分后对应的 Morton 码在这一方向上表示的范围。可以看到当划分次数为 15 时,Morton 码在这一方向上的精度为 1 223 m,就是约为 1 km 左右,这时候对应的码长为 30,这样的码长是比较适合作为在交通道路网络中近邻搜索的起始码长的。

表 1 Morton 码精度与划分次数对应表

划分次数	对应精度/m	划分次数	对应精度/m
1	20 037 726.370 00	11	19 568.09
2	10 018 863.190 00	12	9 784.04
3	5 009 431.593 00	13	4 892.02
4	2 504 715.796 00	14	2 446.01
5	1 252 357.898 00	15	1 223.00
6	626 178.949 10	16	611.50
7	313 089.474 50	17	305.75
8	156 544.737 30	18	152.87
9	78 272.368 63	19	76.43
10	39 136.184 32	20	38.21

图 7 显示了使用不同码长的 Morton 码所构建的 MPT 索引所占的内存大小,可以发现在码长为 30 左右的时候,当插入 MPT 移动对象数据量到达一定的时候,索引所占的空间增长明显减缓,这是因为索引结构中大量减少了重复信息的存储的缘故。基于这样的存储优势,MPT 索引结构可以在应对大量移动对象信息的情形下使用。

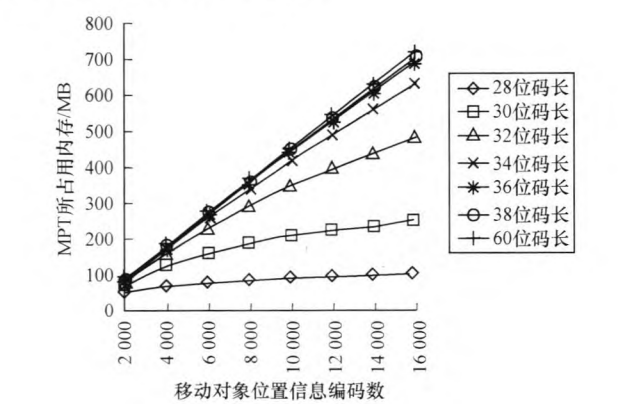


图 7 MPT 空间占用情况

6 结束语

本文在 Morton 和 Patricia 树的基础上提出一种一维索引结构 MPT,通过改良 Patricia 树结构,使得在索引结构操作上有着更好的性能,并且合理利用

Morton 码编码方式,使索引结构拥有比 Geohash 更加细粒度地区域划分精度的控制,同时提出基于 MPT 的近邻算法,其性能优于基于 R-Tree 的近邻搜索算法。通过将二维空间合理划分,并将划分的区域编码,使索引结构具有高效响应区域查询的能力。因为考虑到近邻查询和区域查询都是实时性较强,所以暂时并没有考虑时间因素,下一步将把时间、位置信息编码成一维信息,使得索引结构能够得到更广泛的应用。

参考文献

- [1] 廖巍,熊伟,景宁.移动对象索引技术研究进展[J].计算机科学,2006,33(8):166-169.
- [2] 孟小峰,周龙骧,王珊.数据库技术发展趋势[J].软件学报,2004,15(12):1822-1836.
- [3] Jensen C S, Lin Dan, Ooi B C. Query and Update Efficient B⁺-tree Based Indexing of Moving Objects[C]//Proceedings of the 30th International Conference on Very Large Data Bases. [S. l.]: VLDB Endowment, 2004: 768-779.
- [4] Su Chen, Ooi B C, Tan K L, et al. ST2B-tree: A Self-tunable Spatio-temporal B⁺-tree Index for Moving Objects[C]//Proceedings of ACM SIGMOD International Conference on Management of Data. New York, USA: ACM Press, 2008: 29-42.
- [5] 刘玥,郝忠孝.基于 Buddy * -Hash 的移动对象时空查询方法[J].计算机工程,2010,36(4):47-49.
- [6] Tao Yufei, Papadias D, Sun Jimeng. The TPR * -tree: An Optimized Spatio-temporal Access Method for Predictive Queries [C]//Proceedings of the 29th International Conference on Very Large Data Bases. [S. l.]: VLDB Endowment, 2003: 790-801.
- [7] Simonas S C S J, Leutenegger S T, Lopez M A. Indexing the Positions of Continuously Moving Objects[C]//Proceedings of SIGMOD'00. New York, USA: ACM Press, 1999: 331-342.
- [8] 徐红波,郝忠孝.基于 Hilbert 曲线的近似 k-最近邻查询算法[J].计算机工程,2008,34(12):47-49.
- [9] 刘润涛,陈琳琳,田广悦,等. Z 曲线网格划分的最近邻查询[J].计算机工程与应用,2013,49(22):123-126.
- [10] 徐红波,郝忠孝.基于空间填充曲线网格划分的最近邻查询算法[J].计算机科学,2010,37(1):184-188.
- [11] Liu Jiajun, Li Haoran, Yong Gao et al. A Geohash-based Index for Spatial Data Management in Distributed Memory [C]//Proceedings of the 22nd International Conference on Geoinformatics. Washington D. C., USA: IEEE Press, 2014: 1-4.
- [12] Brinkhoff T. Generating Network-based Moving Objects[C]//Proceedings of the 22nd International Conference on Scientific and Statistical Database Management. Washington D. C., USA: IEEE Press, 2000: 253-255.

编辑 索书志

(上接第 68 页)

仿真结果表明,ONCS 算法通过选择性复制消息,在路由开销指标适中的情况下,具有较高的传输成功率和较低的传输延迟,适用于由人持有移动设备构成的机会社会网络场景。下一步将继续优化社区动态划分策略,使 ONCS 算法适用于实际场景。

参考文献

- [1] 熊永平,孙利民,牛建伟,等.机会网络[J].软件学报,2009,20(1):124-137.
- [2] 刘亚翔,高媛,乔晋龙,等.机会社会网络中基于社区的消息传输算法[J].计算机应用,2013,33(5):1212-1216.
- [3] Pietiläinen A K, Diot C. Dissemination in Opportunistic Social Networks: The Role of Temporal Communities[C]//Proceedings of the 13th ACM International Symposium on Mobile Ad Hoc Networking and Computing. New York, USA: ACM Press, 2012: 165-174.
- [4] 蔡青松,牛建伟,刘畅.一种机会网络中的消息发布/订阅算法[J].计算机工程,2011,37(12):19-22,25.
- [5] Nguyen H A, Giordano S. Context Information Prediction for Social-based Routing in Opportunistic Networks[J]. Ad Hoc Networks, 2012, 10(8): 1557-1569.
- [6] Spyropoulos T, Psounis K, Raghavendra C S. Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks[C]//Proceedings of ACM SIGCOMM Workshop on Delay-tolerant Networking. New York, USA: ACM Press, 2005: 252-259.
- [7] Lindgren A, Doria A, Schelén O. Probabilistic Routing in Intermittently Connected Networks[J]. ACM SIGMOBILE: Mobile Computing and Communications Review, 2003, 7(3): 19-20.
- [8] 牛建伟,周兴,刘燕,等.一种基于社区机会网络的消息传输算法[J].计算机研究与发展,2009,46(12):2068-2075.
- [9] Hui P, Crowcroft J, Yoneki E. BUBBLE Rap: Social-based Forwarding in Delay-tolerant Networks[J]. IEEE Transactions on Mobile Computing, 2011, 10(11): 1576-1589.
- [10] Daly E M, Haahr M. Social Network Analysis for Routing in Disconnected Delay-tolerant MANETs[C]//Proceedings of the 8th ACM International Symposium on Mobile Ad Hoc Networking and Computing. New York, USA: ACM Press, 2007: 32-40.
- [11] 王朕,王新华,隋敬麒.机会网络模拟器 ONE 及其扩展研究[J].计算机应用研究,2012,29(1):272-277.
- [12] 孙践知,刘乃瑞,张迎新,等.机会网络典型路由算法性能分析[J].计算机工程,2011,37(16):86-89.

编辑 金胡考