

动态受限区域内的单纯型连续近邻链查询方法

李 松¹ 张丽平¹ 朱德龙¹ 郝晓红²

(哈尔滨理工大学计算机科学与技术学院 哈尔滨 150080)¹ (哈尔滨理工大学计算中心 哈尔滨 150080)²

摘 要 受限区域内的单纯型连续近邻链查询在空间数据挖掘、数据的相似分析和推理、空间数据库等方面具有重要的作用。为了弥补已有方法的不足,详细研究了动态受限区域内的单纯型连续近邻链查询方法。基于计算几何中的 Voronoi 图给出了 VOR_IN_CRSCNNC 算法、VOR_EX_CRSCNNC 算法和 VOR_DE_CRSCNNC 算法。进一步进行了实验比较和分析。理论研究和实验分析表明,所提出的算法在查询过程中减少了数据逐一筛选和判断的冗余计算,在处理空间数据量较大、初始受限区域数据量较多、受限区域形状较为复杂的单纯型连续近邻链查询方面具有较大的优势。

关键词 空间数据库, Voronoi 图, 最近邻查询, 单纯型连续近邻链, 受限区域
中图分类号 TP311 文献标识码 A

Simple Continues Near Neighbor Chain Query in Dynamic Constrained Regions

LI Song¹ ZHANG Li-ping¹ ZHU De-long¹ HAO Xiao-hong²

(School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China)¹

(Computation Center, Harbin University of Science and Technology, Harbin 150080, China)²

Abstract The simple continues near neighbor chain query in the constrained regions(CRSCNNC-Query) has important significance in the spatial data mining, similarity analysis and reasoning of data, spatial database etc. To remedy the deficiency of the existing work, the simple continues near neighbor chain query in the dynamic constrained regions was studied respectively. The VOR_IN_CRSCNNC algorithm, VOR_EX_CRSCNNC and the VOR_DE_CRSCNNC algorithm were presented based on the Voronoi diagram. Furthermore, the performance of the methods were analyzed and compared by experiment. The theoretical study and the experimental results show that the redundant calculation is reduced and the algorithms hold large advantage at the big data sets and the regions with complex shapes.

Keywords Spatial database, Voronoi diagram, Near neighbor query, Simple continues near neighbor chain, Constrained region

1 引言

数据信息的近邻关系查询在空间数据库、地理信息系统、空间定位与查询、数据挖掘、图像处理与分析、网络智能查询、智能交通控制等领域具有重要的作用^[1,2]。国内外学者对数据集中的近邻关系查询进行了较为广泛的研究,在简单近邻查询、最近对查询、邻近对查询、组最近邻查询、反向最近邻查询等方面取得了一些重要的研究成果^[1]。

近年来,由于数据信息的复杂性和查询需求的不断提高,查询条件变得日益复杂,有关近邻关系的查询进一步扩展到路网中的近邻查询^[3-5]、可视最近邻查询^[5]、可视反向 k 最近邻查询^[6]、聚集最近邻查询^[7,8]、概率频繁最近邻查询^[9]、高维近似最近邻查询^[10]、平面线段最近邻查询^[11]、基于 Voronoi 图的反向最近邻查询^[12]等方面。

已有的研究成果^[1-12]虽然从多方面对近邻关系查询问题

进行了研究,提出了一些较为典型的方法,但没有涉及单纯型连续近邻链(Simple Continues Near Neighbor Chain, SCNNC)查询问题,所提出的方法不适合处理单纯型连续近邻链的查询与分析。文献[13]研究了处理理想二维空间中的单纯型连续近邻链的查询方法,给出了 SCNNC_V_SEARCH 算法。为了处理预定数据链规模的单纯型连续近邻链问题,文献[14]基于空间索引结构 R 树给出了动态数据集中的单纯型连续近邻链查询的算法:SCNNC_R_ST 算法、SCNNC_R_SD 算法和 SCNNC_R_XZ 算法。文献[15]基于空间 Hilbert 曲线给出了预定数据链规模的单纯型连续近邻链的查询算法和更新算法。但文献[13-15]的研究成果主要是针对整个理想空间中的数据集中的单纯型连续近邻链查询问题,没有考虑特定查询区域的限制。实际应用中,大量的查询需求往往不是针对整个理想空间的,经常需要查询特定受限区域中的单纯型连续近邻链。由于受限区域形状的复杂性、多样性和动

到稿日期:2013-08-15 返修日期:2013-10-26 本文受黑龙江省教育厅科学技术研究项目(12531120)资助。

李 松(1977—),男,博士,副教授,主要研究方向为数据库理论与技术、算法设计,E-mail:lisongbeifen@163.com;张丽平(1976—),女,硕士,副教授,主要研究方向为数据结构和算法设计、数据库;朱德龙(1989—),男,硕士生,主要研究方向为数据结构和算法设计;郝晓红(1969—),女,硕士,高级实验师,主要研究方向为数据库理论及应用、算法设计与实现。

态变化性,使得在复杂的受限区域内对数据集的过滤和空间索引结构的构建变得较为困难和低效,且受限区域往往不是静止不变的,随着查询需求的变化,查询区域经常动态改变(例如区域扩张或区域收缩等),从而使得过滤和查询技术变得更为复杂。文献[13-15]的方法不再适合动态复杂受限区域内的单纯型连续近邻链查询问题,实际应用中具有较大的局限性。为了弥补已有方法的不足,本文基于计算几何中的Voronoi图^[16]详细研究了动态受限区域内的单纯型连续近邻链查询方法,给出了相应的算法。

2 基本定义

为了研究动态受限区域内的单纯型连续近邻链查询方法,本节给出了文中所需的基本定义。

定义 1(最近邻查询)^[1] 假设有一 d 维空间的点集 U 和一个查询点 o ,最近邻查询就是找出的子集 $NN(o,U)$: $NN(o,U) = \{u \in U \mid \forall s \in U: D(o,u) \leq D(o,s)\}$ 。若要找出 k 个最近邻,该定义则可扩展成 k 个最近邻的查询,即 $kNN(o,U) = \{u_1, u_2, \dots, u_k\}$ 。其中, $\forall h \in U - kNN(o,U), u \in kNN(o,U)$ 且 $D(o,u) \leq D(o,h); D(u_i, o) \leq D(u_j, o), 1 \leq i < j \leq k$ 。

定义 1 中, $D(o,u)$ 和 $D(o,s)$ 等表示两个数据点之间的距离。在不强调数据集的情况下, $NN(o,U)$ 也可简写为 $NN(o)$, 表示 o 的最近邻。

定义 2(单纯型连续近邻链查询)^[13] 设有一 d 维空间中的数据点集 U , U 中一组有序数据点的集合记为 CL , $CL = \{u_m, u_{m+1}, u_{m+2}, \dots, u_{n-1}, u_n\}$ 。其中, $u_{i+1} \in NN(u_i)$ (或 $u_{i+1} \notin NN(u_i)$ 且 $u_{i+1} \in kNN(u_i)$), $i = m, \dots, n-1$, 称 CL 为 G 集中的一条连续近邻链。 u_m 称为链首点, u_n 称为链尾点。若 CL 满足以下条件,则称 CL 为单纯型连续近邻链:

- (1) $\forall u_i, u_j \in CL$, 若 $i \neq j$, 则有 $u_i \neq u_j$;
- (2) $\forall u_i, u_j \in CL$, 若 $u_{i+1} \neq u_{j+1}$, 则有 $u_i \neq u_j$;
- (3) $\forall u_i \in CL, u_{i+1} \notin \{u_m, \dots, u_i\}$;
- (4) $\forall u_i \in CL$, 若 $u_{i+1} \notin NN(u_i)$ 且 $u_{i+1} \in kNN(u_i)$, 则有 $(k-1)NN(u_i) \subseteq \{u_m, \dots, u_i\}$ 。

在数据集 U 中查找一条单纯型连续近邻链的查询称为单纯型连续近邻链查询,简记为 SCNNC-query。单纯型连续近邻链查询可分为预定数据链规模的单纯型连续近邻链查询和预定链尾点的单纯型连续近邻链查询两大类。本文着重研究预定链尾点的受限区域中的单纯型连续近邻链查询方法。

由单纯型连续近邻链查询的定义可知,单纯型连续近邻链 CL 中的数据点的序号和数据集中数据点的序号具有不同的含义。在 CL 中,后一个数据点的序号总比前一个的序号多 1,即数据点在 CL 中具有严格的先后顺序,而对应的空间数据对象集中的序号则无此先后规律。为便于研究和分析问题,无特殊说明,本文涉及的数据对象点的序号主要是指近邻链 CL 中的序号。

如图 1 所示, $\{u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8, u_9, u_{10}, u_{11}, u_{12}, u_{13}, u_{14}, u_{15}, u_{16}, u_{17}, u_{18}, u_{19}, u_{20}\}$ 是 1 条以 u_1 为链首点、 u_{20} 为链尾点的包含 20 个数据点的单纯型连续近邻链; $\{u_{27}, u_{28}, u_{29}, u_{30}, u_{31}, u_{32}, u_{33}, u_{34}, u_{35}, u_{36}, u_{37}\}$ 则是 1 条以 u_{27} 为链首点、 u_{37} 为链尾点的包含 11 个数据点的单纯型连续近邻链。

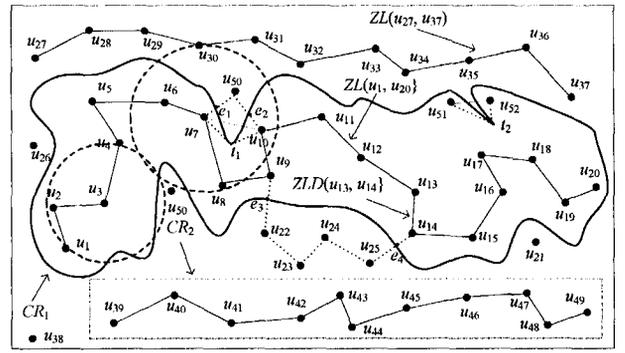


图 1 单纯型连续近邻链和受限区域示例

定义 3(受限区域查询) 设二维空间中的平面区域为 CR , 数据对象集为 U 。 CR 的边界线记为 CR_i 。针对 U , 在 CR 中查询满足某些条件的数据对象称为受限区域查询。 CR 称为受限区域。处于受限区域 CR 之内的数据点称为域内点; 处于 CR 之外的数据点称为域外点; 如图 1 所示, CR_1 和 CR_2 所包围的区域即为两个受限区域。其中, CR_2 为规则的矩形区域, CR_1 为形状较为复杂的不规则区域。

在查询过程中,受限区域 CR 是预先指定的,其形状大小经常会根据查询的需求发生变化,显然, CR 的动态改变将会较大程度地影响受限区域的查询结果。

定义 4(前驱点/后继点) 在单纯型连续近邻链 CL 中, 设前后有序的相邻两个数据点为 u_i 和 u_j , 则称 u_i 是 u_j 的链内直接前驱点, u_j 是 u_i 的链内直接后继点。本文用 $PRE(u_j)$ 表示 u_j 的链内直接前驱点, $ACE(u_j)$ 表示 u_j 的链内直接后继点。对于 CL 中的一个数据点 u_i , 本文称在 CL 中排序在 u_i 之前的数据点集为 u_i 的链内前驱点集, 记为 $QU(u_i)$, 在 CL 中排序在 u_i 之后的数据点集为 u_i 的链内后继点集, 记为 $HJ(u_i)$ 。图 1 中, u_5 即为 u_4 的链内直接后继点, u_4 为 u_5 的链内直接前驱点。

定义 5(连续近邻链折线) 在一条单纯型连续近邻链 CL 中, 由链首点 u_m 开始将连续近邻链中的数据点依次用直线段连接起来直到链尾点 u_n , 所生成的折线称为连续近邻链折线, 记为 $ZL(u_m, u_n)$, 简记为 ZL_i 。如图 1 中的 $ZL(u_1, u_{20})$ 和 $ZL(u_{27}, u_{37})$ 。 u_i 和其链内直接后继点 $ACE(u_i)$ 之间的直线段称为连续近邻链折线段, 记为 $ZLD(u_i, ACE(u_i))$ 。如图 1 中的 $ZLD(u_{13}, u_{14})$ 。

定义 6(最近邻判定圆域) 设 u 和 h 之间的最短距离为 $MIN(u, h)$, 则以 u 为圆心、 $MIN(u, h)$ 为半径生成的圆域称为 u 的判定圆域, 记为 $Circle(u, MIN(u, h))$ 。若 h 是 u 的最近邻, 则称该圆域为 u 的最近邻判定圆域, 记为 $Circle(u, NN_MIN(u, h))$ (如图 1 所示, $Circle(u_3, NN_MIN(u_3, u_4))$ 即为 u_3 的最近邻判定圆域)。

3 动态受限区域内的单纯型连续近邻链查询

实际查询中,受限区域 CR 往往不是静止不动的,随时间的变化和查询条件的改变, CR 的形状、大小和位置经常会发生动态改变。 CR 的动态变化使得受限区域内的单纯型连续近邻链查询变得更为复杂。本节着重研究动态受限区域内的单纯型连续近邻链查询问题。3.1 节给出了初始受限区域内

的单纯型连续近邻链查询方法;3.2节给出了受限区域扩张情况下的单纯型连续近邻链查询方法;3.3节给出了受限区域缩小情况下的单纯型连续近邻链查询方法。

3.1 初始受限区域内的单纯型连续近邻链查询

在动态受限区域内的单纯型连续近邻链查询过程中,相关的计算与查询总是从受限区域 CR 在某一时刻的一个初始状态开始,本小节首先讨论初始受限区域内的单纯型连续近邻链查询方法。

为了处理初始受限区域内的单纯型连续近邻链查询问题,根据数据对象集 U 中的数据点和受限区域 CR 的空间位置关系,基于“先过滤后计算”的思想,我们首先给出一种较直接的方法(本文简称为 $FL_IN_CRSCNNC$ 方法)。其核心思想是首先确定初始受限区域 CR 的空间位置和形状,逐一计算和确定数据对象集 U 中处于 CR 内的数据对象点集 U' ,去除冗余对象点;再在 CR 内利用空间填充曲线-格雷曲线(如图2所示)和典型索引结构 B 树的结合对 U' 进行空间数据的组织和查询,最后利用索引结构的组合进行查询,从而得出满足条件的查询结果。

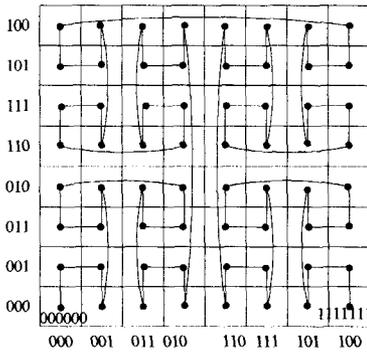


图2 三阶格雷曲线示例

利用 $FL_IN_CRSCNNC$ 方法处理受限区域内的 SCNNC 查询问题时,当受限区域 CR 的形状为矩形、圆形等较为规则的图形时,利用矩形区域的左下角和右上角坐标(或圆形区域的圆心坐标及半径)等信息可较方便地过滤出 CR 内的数据对象点,同时利用格雷曲线和 B 树组织矩形或圆形等规则区域内的数据对象点也具有较大的优势。因而, $FL_IN_CRSCNNC$ 方法较适合处理受限区域 CR 较为规则且数据量较少的 SCNNC 查询问题。现实中,受限区域 CR 的形状往往非常复杂,当数据对象集的规模较大时, CR 内有效数据的筛选将变得困难低效,且 CR 复杂的形状较大程度地影响数据点的近邻关系,构建空间索引结构的代价较大。为了弥补 $FL_IN_CRSCNNC$ 方法的不足,本节利用 Voronoi 图^[16](如图3所示)和判定圆域及相关性质给出了基于 Voronoi 图的受限区域内的单纯型连续近邻链的查询方法。

利用 Voronoi 图计算生成点的最近邻或 k 最近邻,仅需考虑其周围一级邻接生成点或多级邻接生成点。大量的冗余数据被有效地筛选掉。

受限区域 CR 形状的复杂性,使得空间数据点之间的原始距离关系受到较大程度的影响,定义7和定义8分别给出了域外/内阻断和链内截点的概念。

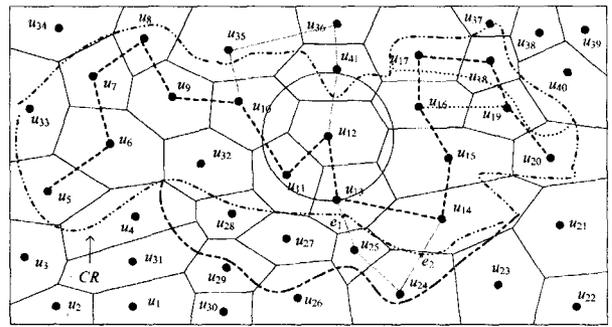


图3 Voronoi 图和动态受限区域示例

定义7(域外阻断/域内阻断) 对于空间数据集中的两个数据点 u_j 和 u_i ,若 u_j 和 u_i 分别为某受限区域 CR 的域内点和域外点,则称 u_j 和 u_i 被域外阻隔,表示为 $OZD(u_j, u_i)$;若 u_j 和 u_i 均在受限区域 CR 内,但 u_j 和 u_i 之间的直线段被 CR 的边界线 CR_i 截断,即 u_j 和 u_i 之间不能直线可达,则称 u_j 和 u_i 被域内阻断,表示为 $IZD(u_j, u_i)$ 。若 u_j 和 u_i 被域内阻断,则 u_j 和 u_i 通过折线可达,其最短折线距离表示为 $D_Z(u_j, u_i)$ 。如图1所示, u_7 和 u_{50} 被域外阻隔; u_7 和 u_{10} 被域内阻隔。

定义8(域内截点) 对于一条初始单纯型连续近邻链 CL ,在受限区域环境下,若 CL 的某些连续近邻链折线段 $ZLD(u_i, ACE(u_i))$ 被某受限区域 CR 的边界线 CR_i 截断,且 u_i 为 CR 的域内点,则称 u_i 为相对于 CR 的域内截点,表示为 $JD(u_i, CR_i)$ 。域内截点可有多个,称在 CL 中排序最靠前的域内截点为域内首截点,记为 $SJD(CL, CR_i)$ 。图3中, u_{10} , u_{12} , u_{13} 即为 CL 的域内截点。

基于以上定义,具体查询算法如算法1所示。

算法1 $VOR_IN_CRSCNNC(u_s, u_e, U, CR_i)$

输入:空间数据对象集 U ,链首点 u_s ,链尾点 u_e ,受限区域边界线 CR_i
输出:由链首点 u_s 开始至链尾点 u_e 的一条单纯型连续近邻链 CL

begin

$CL \leftarrow \emptyset; G \leftarrow \emptyset; u \leftarrow u_s; //$ 初始化

if 受限区域 CR 为规则区域 then

$CL \leftarrow FL_IN_CRSCNNC(u_s, u_e, U, CR_i);$

else

if U 的 Voronoi 图没生成 then

 Create_Voronoi(U);

$U \leftarrow U - u_s;$

$CL \leftarrow u_s;$

(1) $q \leftarrow NN_V(u, U); //$ 利用 Voronoi 图计算 u 点的最近邻

$CL \leftarrow q; u \leftarrow q;$

$U \leftarrow U - q;$

if $q = u_e$ then

$r \leftarrow |CL|; \text{转}(2);$

else

 清空 q ; 转(1);

(2) for $i=1$ to $r-1$ do

$G \leftarrow ZLD(u_i, ACE(u_i)) \cap CR_i;$

//提取 CL 的折线段和受限区域边界线的交集

if $ZLD(u_i, ACE(u_i)) \cap CR_i \neq \emptyset;$

$Ljd[] \leftarrow u_i; //$ 提取域内截点并置入数组

```

if IZD( $u_i$ , ACE( $u_i$ ))=true then //若  $u_i$  和其链内直接
后继点 ACE( $u_i$ )被域内阻断
  IYZ[]← $u_i$ ;
  在 CR 中计算  $u_i$  的最近邻 NN( $u_i$ );
  HYZ[]←NN( $u_i$ );
if G=∅ then
  return CL;
else
  for j=1 to |Ljd| do
    以 Ljd[j]为链首点计算 CR 内的局部连续最近链
    CLj;
    确定 CLj 之间的共同连接点;
    update CL; //更新 CL
  return CL;
end

```

3.2 受限区域扩张环境下的单纯型连续近邻链查询

3.1 节基于 Voronoi 图研究了初始受限区域内的 SCNNC 查询方法,给出了 VOR_IN_CRSCNNC 算法。随时间变化,受限区域 CR 往往会发生动态改变,本节着重研究 CR 扩张情况下的单纯型连续近邻链查询方法。本文所指的 CR 的扩张实质上是其边界线的 CR_l 的整体或局部扩大。显然,CR 的扩张将会对单纯型连续近邻链查询结果直接产生影响。如图 3 所示,CR 扩张后, $u_{24}, u_{25}, u_{27}, u_{28}$ 等数据对象点由域外点转变成了域内点,图 3 中以 u_5 为链首点、 u_{20} 为链尾点的单纯型连续近邻链由 $\{u_5, u_6, u_7, u_8, u_9, u_{10}, u_{11}, u_{12}, u_{13}, u_{14}, u_{15}, u_{16}, u_{17}, u_{18}, u_{19}, u_{20}\}$ 改变为 $\{u_5, u_6, u_7, u_8, u_9, u_{10}, u_{11}, u_{12}, u_{13}, u_{25}, u_{24}, u_{14}, u_{15}, u_{16}, u_{17}, u_{18}, u_{19}, u_{20}\}$ 。

受限区域的扩张变动使得单纯型连续近邻链的计算、查询与更新变得较为复杂。采用“基于受限区域更新的重计算”的策略,我们可得出处理受限区域扩张环境下单纯型连续近邻链的一种方法(本文简称 GR_EX_CRSCNNC 方法)。该方法的主要思路是首先计算扩张后的受限区域 CR' 的边界线 CR'_l ,再针对新的受限区域的边界线 CR'_l 调用 VOR_IN_CRSCNNC 算法完成受限区域扩张后的单纯型连续近邻链的查询。该方法在受限区域形状较为规则(如矩形)且区域的动态扩张频率较低的情况下较为可行。由于该方法在受限区域每次进行扩张时都需要重新计算由链首点到链尾点的整条单纯型连续近邻链,没有充分利用该次扩张前的近邻链信息,在受限区域形状不规则且区域边界局部改变较为频繁的情况下则具有较低的查询效率,查询过程中具有较多的冗余计算。为了弥补该方法的不足,基于“启发式局部更新”的策略,我们进一步提出了 VOR_EX_CRSCNNC 方法。该方法充分利用每次扩张之前的近邻链信息,利用 Voronoi 图和近邻判定圆等对近邻链进行局部计算和更新。

为了进一步探讨 VOR_EX_CRSCNNC 方法,首先给出选换点的概念,如定义 9 所示。

定义 9(选换点) 对于无受限区域的二维空间内的一条单纯型连续近邻链 CL ,在某初始受限区域 CR 环境下,若 CL 中的某些数据点被替换掉, CL 更新为 CL' ,则这些替换掉的数据点称为 CL' 相对于 CL 的初始选换点。若 CR 经过动态变化(扩张或收缩等), CL' 中的某些数据点被替换掉, CL' 更

新为 CL'' ,则定义这些更换掉的数据点为 CL'' 相对于 CL' 的一级选换点,同理可定义二级选换点和多级选换点。

对于无受限区域的二维空间内的一条单纯型连续近邻链 CL ,在某初始受限区域 CR 环境下,若初始选换点集为空,则可说明在 CR 环境下, CL 没有变化,即 CR 对 CL 没有产生影响。显然,当 CR 扩张后, CL 依然保持不变。故可首先通过判断初始受限区域对初始选换点集的影响情况进行分情况处理。当初始选换点集不为空时,则进一步研究扩张后的受限区域 CR' 的边界线 CR'_l 和扩张前的单纯型连续数据链及初始选换点集的关系,对选换点集进行计算和更新。

VOR_EX_CRSCNNC 方法的核心步骤如下:

第 1 步 完成单纯型连续近邻链的初始化。利用文献 [13] 提出的 SCNNC_V_SEARCH 算法生成无受限区域中的单纯型连续近邻链 CL_1 ;利用 3.1 节提出的 VOR_IN_CRSCNNC 算法生成初始受限区域 CR 环境下的单纯型连续近邻链 CL 。

第 2 步 确定 CL_1 的连续近邻链折线是否和初始受限区域 CR 的边界线 CR_l 相交,若不相交(即基于 CR, CL' 相对于 CL 的初始选换点集为空),则直接返回 CL_1 为查询结果;若相交(即基于 CR, CL' 相对于 CL 的初始选换点集不为空),则顺序进入第 3 步。

第 3 步 确定经过一次扩张后的新的受限区域 CR' 的边界线 CR'_l ;计算一级选换点集、域内截点和阻断点信息、 CR' 环境下的链内截点集 JD ;基于 CL ,在 CR' 环境下利用所计算的数据信息进行连续近邻链的局部判断和更新。

基于以上讨论,本节进一步给出受限区域扩张环境下的单纯型连续近邻链的具体查询算法,如算法 2 所示。

算法 2 VOR_EX_CRSCNNC(u_s, u_e, U, CR'_l)

输入:空间数据点集 U ,链首点 u_s ,链尾点 u_e ,扩张后的受限区域的边界线 CR'_l

输出:由链首点 u_s 开始至链尾点 u_e 的一条单纯型连续近邻链 CL'

```

begin
  CL'←∅;CL1←∅; //初始化
  确定初始受限区域边界线 CRl;
  if 没有生成无受限区域环境下的 SCNNC then
    CL1←SCNNC_V_SEARCH( $u_s, u_e, U$ ) //调用文献[13]中的
    算法生成无受限区域环境下的 SCNNC
    k←|CL1|; //获取 CL1 的 SCNNC 的规模
    for i=1 to k-1 do
      依次遍历 CL1 中的数据点  $u_i$ ;
      E←ZLD( $u_i, ACE(u_i)$ )∩CRl;
    if E=∅ then
      CL'←CL; return CL';
    if 没有生成初始受限区域 CR 内的 SCNNC then
      CL←VOR_IN_CRSCNNC( $u_s, u_e, U, CR_l$ );
      //调用算法 1 生成 CR 内的 SCNNC
    else
      确定 CR'l; 计算初始选换点集 XH;
      计算一级选换点集 XH';
      计算 CR' 环境下的链内截点集 JD;
      Rjd[]←Ljd[]; FYZ[]←IYZ[]; DYZ[]←HYZ[];
      //数组 Rjd[], FYZ[] 和 DYZ[] 存储算法 1 中所计算出的相关

```

```

CL 的域内截点和域内阻断点信息
for i=1 to |FYZ| do
    HYZ'[i]←ACE(FYZ[i]);
if XH'=XH and HYZ'[i]=DYZ[i] then
    CL'←CL;return CL';
else
    计算发生改变的选择点、域内截点和域内阻断点的近邻信息;
    局部更新 CL;
    CL'←CL;return CL';
end

```

受限区域的局部扩张往往仅影响已有单纯型连续近邻链的局部数据,故算法 2 调用文献[13]中的算法首先生成无受限区域环境下的单纯型连续近邻链,调用算法 1 生成受限区域 CR 内的单纯型连续近邻链。算法 2 中的数组 $Rjd[]$ 、 $FYZ[]$ 和 $DYZ[]$ 存储的是算法 1 中所计算出的相关 CL 的域内截点和域内阻断点信息,算法 2 在已有单纯型连续近邻链的基础上充分利用这些信息进行近邻链的局部更新和调整。

3.3 受限区域收缩环境下的单纯型连续近邻链查询

3.1 节研究了受限区域扩张环境下的单纯型连续近邻链查询的方法。在许多情况下,查询受限区域收缩后的单纯型连续近邻链也是较为常见的。对于链首点和链尾点预先确定的一条单纯型连续近邻链,受限区域收缩前后经常会发生显著改变。如图 3 所示,受限区域 CR 收缩后, u_{17} 、 u_{18} 等数据对象点由域内点转变成了域外点,相应地,图 2 中以 u_5 为链首点、 u_{20} 为链尾点的单纯型连续近邻链由 $\{u_5, u_6, u_7, u_8, u_9, u_{10}, u_{11}, u_{12}, u_{13}, u_{14}, u_{15}, u_{16}, u_{17}, u_{18}, u_{19}, u_{20}\}$ 改变为 $\{u_5, u_6, u_7, u_8, u_9, u_{10}, u_{11}, u_{12}, u_{13}, u_{25}, u_{24}, u_{14}, u_{15}, u_{16}, u_{19}, u_{20}\}$ 。因本文研究的是链首点和链尾点已知,待查数据链包含的数据点的个数未知的单纯型连续近邻链查询问题,故无特殊说明,预定的链首点和链尾点总是处于收缩后的受限区域中。

与 3.1 节的探讨类似,处理受限区域收缩环境下的单纯型连续近邻链查询问题,根据“基于受限区域更新的重计算”的策略,我们可相应得出一种简单方法(本文简称 $GR_DE_CRSCNNC$ 方法)。 $GR_DE_CRSCNNC$ 方法的主要思路是首先计算收缩后的受限区域 CR' 的边界线 CR'_i ,再针对新的受限区域的边界线 CR'_i 调用 $VOR_IN_CRSCNNC$ 算法完成受限区域收缩后的单纯型连续近邻链的查询。显然,该方法在数据集规模较大,受限区域的形状不规则,受限区域收缩频率较高的情况下查询效率较低,每次收缩后的重新计算增大了计算的冗余性。为了弥补该方法的不足,且充分利用受限区域每次收缩之后的近邻链信息,与 3.1 节的方法类似,我们基于“启发式局部更新”的策略进一步提出了 $VOR_DE_CRSCNNC$ 方法。该方法的核心思想是:首先调用 $VOR_IN_CRSCNNC$ 算法生成初始受限区域 CR 内的单纯型连续近邻链 CL ;确定收缩后的受限区域 CR' 的边界线 CR'_i ;分析 CR'_i 对 CL 的影响情况,根据得出的链内截点和选换点信息进行计算,继而对 CL 进行局部更新。

基于以上讨论,本节进一步给出受限区域扩张环境下的单纯型连续近邻链的具体查询算法,如算法 3 所示。

算法 3 $VOR_DE_CRSCNNC(u_s, u_e, U, CR'_i)$

```

输入:空间数据集  $U$ ,链首点  $u_s$ ,链尾点  $u_e$ ,受限区域收缩后的边界线  $CR'_i$ 
输出:由链首点  $u_s$  开始至链尾点  $u_e$  的一条单纯型连续近邻链  $CL'$ 
begin
    OT←∅;CL'←∅; //初始化
if 没有生成初始受限区域  $CR$  内的 SCNNC then
    CL←VOR_IN_CRSCNNC( $u_s, u_e, U, CR_i$ );
    //调用算法 1 生成  $CR$  内的 SCNNC
k←|CL|; //获取  $CR$  内的 SCNNC 的规模
确定  $CR'_i$ ;
for i=1 to k-1 do
    OT←ZLD( $u_i, ACE(u_i)$ )∩ $CR'_i$ ;
if OT=∅ then
    CL'←CL;return CL';
else
    for j=1 to |OT| do
        确定域内截点和一级选换点;
        计算域内截点的最近邻;
        局部更新 CL;
        CL'←CL; return CL';
end

```

4 实验比较与分析

为了处理动态受限区域内的单纯型连续近邻链查询(DCRSCNNC 查询)问题,第 3 节详细研究了动态受限区域环境下的单纯型连续近邻链查询方法,提出了 $VOR_IN_CRSCNNC$ 算法、 $VOR_EX_CRSCNNC$ 算法和 $VOR_DE_CRSCNNC$ 算法。本节在 AMD FX,4G CPU,8G 内存,Windows XP 环境下,利用 C++builder6.0 对所提算法性能进行了实验比较。所用空间数据集是由空间数据生成器(GEDATA 5.0)生成的空间模拟数据。

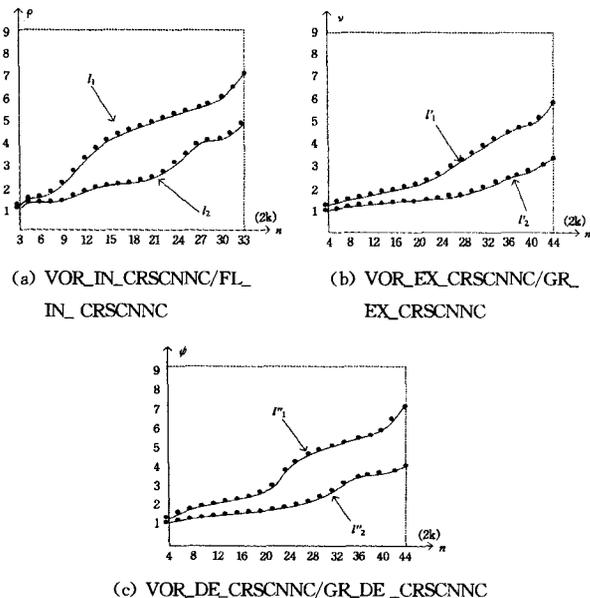


图 4 实验比较结果

设数据集 U 包含的数据量为 n ,形状不规则的初始受限区域为 CR 所含的数据点的数量为 m 。 ρ 表示在不同条件下 $VOR_IN_CRSCNNC$ 算法相对于直接计算方法 $FL_IN_CRSCNNC$ 方法的计算效率的比率; ν 表示在不同条件下

VOR_EX_CRSCNNC 算法相对于“基于受限区域更新的重计算”策略的 GR_EX_CRSCNNC 方法的计算效率的比率; ψ 表示在不同条件下 VOR_DE_CRSCNNC 算法相对于 GR_DE_CRSCNNC 方法的计算效率的比率。基于不同的数据集和受限区域包含数据对象点的不同,图 4 主要展示了以下 3 种主要实验类型的实验结果。

类型 1 VOR_IN_CRSCNNC 算法相对 FL_IN_CRSCNNC 方法计算效率的比较(图 4(a)中,横坐标表示空间数据集包含的数据量,单位为 $2k, k=1024$;纵坐标 ρ 表示 VOR_IN_CRSCNNC 算法相对于 FL_IN_CRSCNNC 方法的计算效率的比率;曲线 l_1 对应 $m=4850$ 时的实验结果;曲线 l_2 对应 $m=1420$ 时的实验结果)。

类型 2 VOR_EX_CRSCNNC 算法相对 GR_EX_CRSCNNC 方法计算效率的比较(图 4(b)中,横坐标表示空间数据集包含的数据量,单位为 $2k, k=1024$;纵坐标 ν 表示 VOR_EX_CRSCNNC 算法相对于 GR_EX_CRSCNNC 方法的计算效率的比率;曲线 l_1' 对应 $m=3370$ 时的实验结果;曲线 l_2' 对应 $m=750$ 时的实验结果)。

类型 3 VOR_DE_CRSCNNC 算法相对 GR_DE_CRSCNNC 方法计算效率的比较(图 4(c)中,横坐标表示空间数据集包含的数据量,单位为 $2k, k=1024$;纵坐标 ϕ 表示 VOR_DE_CRSCNNC 算法相对于 GR_DE_CRSCNNC 方法的计算效率的比率;曲线 l_1'' 对应 $m=4180$ 时的实验结果;曲线 l_2'' 对应 $m=820$ 时的实验结果)。

由图 4(a)可知,当受限区域中的数据点个数一定时,随着空间数据集的增大,VOR_IN_CRSCNNC 算法相对于 FL_IN_CRSCNNC 方法具有更高的计算效率。对于同一数据集,VOR_IN_CRSCNNC 算法处理当受限区域中的数据点的个数较多时的查询问题具有更明显的优势。由图 4(b)可知,当受限区域中的数据点个数一定时,随着空间数据集的增大,VOR_EX_CRSCNNC 算法相对 GR_EX_CRSCNNC 方法的计算效率的优势更为明显。对于同一数据集,处于受限区域中的数据点的个数越多,VOR_EX_CRSCNNC 算法的计算效率的优势越明显。由图 4(c)可知,当受限区域中的数据点个数一定时,VOR_DE_CRSCNNC 相对 GR_DE_CRSCNNC 方法计算效率的比率更大。对于同一数据集,VOR_DE_CRSCNNC 算法更适合处理当初受限区域的数据量较大时的情况。

结束语 受限区域内的单纯型连续近邻链查询是数据对象近邻关系查询领域的一个重要内容。为了有效处理动态受限区域内的单纯型连续近邻链查询问题,本文首先基于计算几何中的 Voronoi 图研究了初始受限区域中的单纯型连续近邻链查询方法,给出了 VOR_IN_CRSCNNC 算法;进一步基于“启发式局部更新”的策略详细研究了受限区域动态扩张和收缩环境下的单纯型连续近邻链查询方法,分别给出了 VOR_EX_CRSCNNC 算法和 VOR_DE_CRSCNNC 算法。理论与实验表明,本文所提的方法在处理空间数据量较大、初

始受限区域包含数据量较多、受限区域形状较为复杂的单纯型连续近邻链查询方面具有较大的优势。未来的研究重点主要集中在以下两点:

1. 不确定受限区域(例如 Vague 区域^[17])中的单纯型连续近邻链查询。
2. 复杂含洞受限区域中的单纯型连续近邻链查询。

参考文献

- [1] 郝忠孝. 时空数据库查询与推理[M]. 北京:科学出版社,2010
- [2] 李松,张丽平,孙冬璞. 空间关系查询与分析[M]. 哈尔滨:哈尔滨工业大学出版社,2011
- [3] Mouratidis K, Yiu Man-lung, Dimitris papadias. Continuous nearest neighbor monitoring in road networks[C]// VLDB. Seoul, Korea, 2006
- [4] 李艳红,李国徽,杜小坤. 路网中双色数据集上连续反向 k 近邻查询处理的研究[J]. 计算机科学,2012,39(11):131-136
- [5] Nutanong S, Tanin E, Zhang Rui. Incremental evaluation of visible nearest neighbor queries[J]. IEEE Transactions on Knowledge Engineering, 2010, 22: 665-681
- [6] Gao Yun-jun, Zheng Bai-hua, Chen Gen-cai, et al. Visible Reverse k -Nearest Neighbor Query Processing in Spatial Databases [J]. IEEE Trans. Knowl. Data Eng, 2009, 21: 1314-1327
- [7] Hu Ling, Jing Yi-nan, Ku Wei-Shinn, et al. Enforcing k Nearest Neighbor Query Integrity on Road Networks[C]// Proceedings of the 20th International Conference on Advances in Geographic Information Systems. 2012: 422-425
- [8] Elmongui H G, Mokbel M F, Aref W G. Continuous aggregate nearest neighbor queries[J]. GeoInformatica, 2013, 17(1): 63-95
- [9] 苗东菁,石胜飞,李建中. 一种局部相关不确定数据库快照集合上的概率频繁最近邻算法[J]. 计算机研究与发展, 2011, 48(10): 1812-1822
- [10] 袁培森,沙朝锋,王晓玲,等. 一种基于学习的高维数据 c -近似最近邻查询算法[J]. 软件学报, 2012, 23(8): 2018-2031
- [11] 刘润涛,郝忠孝. 空间数据库平面线段快速最近邻查询算法[J]. 计算机研究与发展, 2011, 48(12): 2379-2384
- [12] 李松,郝忠孝. 基于 Voronoi 图的反向最近邻查询方法研究[J]. 哈尔滨工程大学学报, 2008, 29(3): 261-265
- [13] 李松,张丽平,蔡志涛. 数据集中单纯型连续近邻链查询方法[J]. 计算机工程, 2012, 38(4): 82-85
- [14] Zhang Li-ping, Li Song, Li Lin, et al. Simple Continues Near Neighbor Chain Query of the Datasets Based on the R Tree[J]. Journal of Computational Information Systems, 2012, 8(22): 9159-916
- [15] 张丽平,李林,李松,等. 预定数据链规模的单纯型连续近邻链查询[J]. 计算机工程, 2012, 38(10): 51-53
- [16] Sacl J R, Urrutia J. Voronoi diagrams[M]// Handbook on Computational Geometry. Ottawa: Elsevier Science, 2000: 201-290
- [17] 李松,郝忠孝. 基于 Vague 集的含洞不规则 Vague 区域关系[J]. 计算机研究与发展, 2009, 46(5): 823-831