

基于 Voronoi 图的线段最近对查询

杨泽雪^{1,3} 郝忠孝^{1,2}

(哈尔滨理工大学计算机科学与技术学院 哈尔滨 150080)¹

(哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001)²

(黑龙江工程学院计算机科学与技术系 哈尔滨 150050)³

摘要 最近对查询是空间数据库中的重要查询之一。已有的关于最近对查询的研究基本集中在点对象上,对空间对象无法抽象为点的对象则研究较少。提出基于平面线段的最近对查询,即找出两个平面线段集中距离最近的线段对。提出基于 Voronoi 图的线段最近对查询算法,该方法构造两个线段集的 Voronoi 图,利用 Voronoi 图的最近邻近特性和局域动态特性找到互为最近邻的线段对,从中找到结果,以缩减大量的计算代价。对线段集中增加线段和删除线段的情况做了相应的处理。实验证明,该算法具有较高的查询效率。

关键词 线段 Voronoi 图,空间数据库,线段最近对,线段最小距离

中图法分类号 TP311.131 **文献标识码** A

Line Segment Closest Pair Queries Based on Voronoi Diagram

YANG Ze-xue^{1,3} HAO Zhong-xiao²

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

(College of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)²

(Department of Computer Science and Technology, Heilongjiang Institute of Technology, Harbin 150050, China)³

Abstract Closest pair(CP) query is one of the important spatial queries in spatial database. But the existed researches on CP query are mainly focused on point objects, which can rarely solve some instance that spatial object can not be abstracted as a point. The question of closest pair query between line segment and line segment was first put forward. That is finding the line segment pair which has the shortest distance among all the line segment pairs of two line segment sets. The line segment closest pair query algorithm based on Voronoi diagram was proposed, and the relevant theorem and proof were given. The Voronoi diagram of two line segment sets was constructed respectively in the method. By making use of nearest adjacent property and local dynamic property, the line segment pairs that are the nearest neighbor mutually were found from which the result was got, in order to reduce a lot of computational cost. The two circumstances of insert line segment and delete line segment were dealt with. Experiments demonstrate the proposed algorithms have high query efficiency.

Keywords Line segment Voronoi diagram, Spatial database, Line segment closest pair, Line segment nearest distance

1 引言

最近对查询是空间数据库中重要的查询之一,在利用空间数据进行决策以及数据处理操作领域有广泛的应用。文献[1,2]给出了存储在 R-树索引中的两个数据集的最近对处理算法 K-CPQ,该算法将两个点集分别存储在两棵 R*-树中,从根结点开始向下深度优先遍历两棵 R*-树,通过启发式规则和更新策略避免遍历大量结点;文献[3]提出了一种新的索引结构 bRdnn-树来处理最近对问题;文献[4]提出了一种高维空间中基于 Z 曲线的近似 k-最近对查询算法;文献[5]研究了受限最近对查询,给出了相应的解决方法。以上的最近对查询算法主要集中在点对象的研究上,即将空间对象抽象为

点对象进行研究。然而有些空间对象无法抽象为点,如河流、道路等,为此将这些空间对象抽象为平面线段,提出基于线段的最近对查询。空间数据库中基于线段的查询研究较少,主要有点和线段之间的最近邻查询^[6],即在点集中找到某条线段的最近邻,没有解决线段之间的最近邻。文献[7]中首次解决了平面线段之间的最近邻问题,提出基于线段最近邻的概念和相应的查询算法,但其中没有提到平面线段之间的最近对查询。文献[8,9]研究了基于平面线段的反向最近邻查询。本文提出平面线段的最近对查询。给定平面中两个线段集,构造两个线段集的 Voronoi 图,利用 Voronoi 图的最近邻特性,找到两个集合中互为最近邻的线段对的集合,在此集合中找到线段最近对。

到稿日期:2011-07-16 返修日期:2011-11-23 本文受黑龙江省教育厅科学技术研究项目(12511102)资助。

杨泽雪(1978-),女,博士生,讲师,主要研究方向为空间数据库查询,E-mail: yzx1978@126.com;郝忠孝(1940-),男,教授,博士生导师,主要研究方向为空间数据库理论、无环数据库、主动数据库、时态数据库和空间数据库。

2 相关定义和性质

定义 1^[10] (线段 Voronoi 图) 给定一组互不相交的生成线段 $L=\{l_1, \dots, l_n\}$, 每一线段的两端点已知, 其中 $2 < n < \infty$, 且当 $i \neq j$ 时, $l_i \neq l_j$ 。其中 $i, j \in I_n = \{1, \dots, n\}$ 。线段 Voronoi 图将平面分成若干个相连区域, 称为 Voronoi 区域, 每个 Voronoi 区域与每一个生成线段对应。每个生成线段的 Voronoi 区域由下式给出: $VL(l_i) = \{p | d(p, l_i) \leq d(p, l_j)\}$, 其中 $j \neq i, j \in I_n, d(p, l_i)$ 为 p 与线段 l_i 之间的最小距离 (欧氏空间中点 p 与 l_i 之间的直线距离)。由 l_i 所决定的区域称为 Voronoi 多边形。由 $VL(L) = \{VL(l_1), \dots, VL(l_n)\}$ 所定义的图形被称为线段 Voronoi 图。共享相同的棱的 Voronoi 多边形被称为邻接多边形, 它们的生成线段被称为邻接生成线段。图 1 给出了 $q_1 - q_{10}$ 10 条线段的 Voronoi 图。

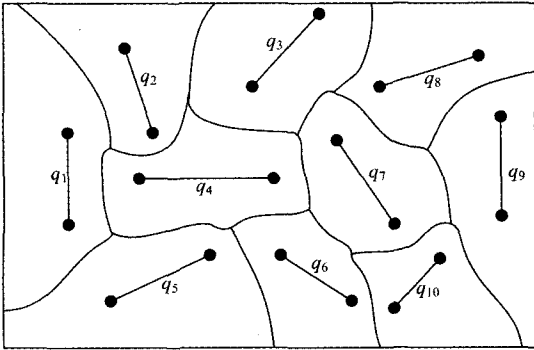


图 1 线段 Voronoi 图

性质 1^[10] 一个线段 Voronoi 图的每条棱是由直线段和抛物线段组成的, 每条棱最多由 7 个部分组成。

性质 2^[11] (最近邻特性) 生成线段 l_i 的最近邻在 l_i 的邻接生成线段之中。

性质 3^[11] (线性特性) 令 n_v 和 n_e 分别为生成线段和 Voronoi 棱的数量, 则 $n_e \leq 3n_v - 6$ 。

性质 4^[11] (局域动态特性) 由性质 3 可得, 每个 Voronoi 棱由两个 Voronoi 多边形所共享, 而每个 Voronoi 多边形的 Voronoi 棱的平均数目最多是 6, 因此每个生成线段最多有 6 个邻接的生成线段。

定义 2 (k 级邻接生成线段) 给定一组生成线段 $L=\{l_1, \dots, l_n\}$ 生成的 Voronoi 图。其中 $2 < n < \infty$, 且当 $i \neq j$ 时, $l_i \neq l_j, i, j \in I_n = \{1, \dots, n\}$ 。 l_i 的 k 级邻接生成线段定义如下:

a) 二级邻接生成线段。 $AG_1(l_i) = \{l_j | VL(l_i) \text{ 和 } VL(l_j) \text{ 有公共边}\}$ 。

b) $k(k > 2)$ 级邻接生成线段。 $AG_k(l_i) = \{l_j | VL(l_i) \text{ 和 } VL(l_j) \text{ 有公共边}, l_j \in AG_{k-1}(l_i)\}$ 。

例如如图 1 线段 A, B, C, D 的二级和三级邻接生成线段如表 1 所列。

表 1 Table 表

生成线段	二级邻接生成线段	三级邻接生成线段
q_1	q_2, q_4, q_5	q_3, q_6, q_7
q_2	q_1, q_3, q_4	q_5, q_6, q_7, q_8
q_3	q_2, q_4, q_7, q_8	$q_1, q_5, q_6, q_9, q_{10}$
q_4	$q_1, q_2, q_3, q_5, q_6, q_7$	q_8, q_9, q_{10}

定义 3^[8] (线段之间的最近距离) 已知线段 L 和线段

K , 点 $l, l_i, l_j \in L$, 点 $k, k_i, k_j \in K$, $dist(l, k)$ 表示点 l 到点 k 的距离, 设线段 L 与线段 K 的最近距离为 $dist(L, K)$, 则 $dist(L, K) = \{dist(l_i, k_i), dist(l_i, k_i) \leq dist(l_j, k_j)\}$ 。由定义可知, $dist(L, K) = dist(K, L)$ 。

由定义可知, $dist(L, K) = dist(K, L)$ 。若两条线段相交, 则它们的最近距离为 0。

定义 4^[8] (线段最近邻(LNN)查询) 已知线段 L 和线段集 H , 线段 $H_i, H_j \in H$, 则 $LNN(L) = \{H_i | dist(L, H_i) \leq dist(L, H_j)\}$ 。

定义 5 (线段最近对(LCP)查询) 给定两个线段集 $P = \{p_1, p_2, \dots, p_n\}$ 和 $Q = \{q_1, q_2, \dots, q_m\}$, 则线段最近对定义为 $LCP(P, Q) = \{ \langle p_x, q_y \rangle | p_x \in P \wedge q_y \in Q, dist(p_i, q_j) \geq dist(p_x, q_y), \forall p_i \in P \wedge \forall q_j \in Q \}$ 。

定义 6 (互为最近邻线段对) 给定两个线段集 $P = \{p_1, p_2, \dots, p_n\}$ 和 $Q = \{q_1, q_2, \dots, q_m\}$, 若其中某线段对 $\langle p_x, q_y \rangle$ ($p_x \in P \wedge q_y \in Q$) 满足 $LNN(p_x) = q_y$ 且 $LNN(q_y) = p_x$, 则 $\langle p_x, q_y \rangle$ 称为互为最近邻线段对。

3 基于 Voronoi 图的线段最近对查询

3.1 相关定理

成为线段最近对的两条线段一定互为最近邻线段对。找到线段最近对的方法可以先找到两个集合中互为最近邻线段对的集合, 则线段最近对一定在互为最近邻线段对中。

定理 1 已知线段集 $P = \{p_1, p_2, \dots, p_n\}, Q = \{q_1, q_2, \dots, q_m\}$, 若 $\langle p_x, q_y \rangle$ ($p_x \in P \wedge q_y \in Q$) 是两个线段集的线段最近对, 则 $\langle p_x, q_y \rangle$ 一定是互为最近邻线段对。

证明 (反证法): 假设 $\langle p_x, q_y \rangle$ 不是互为最近邻线段对, 则一定存在某 p_u 或 q_v 满足 p_x 的最近邻为 q_v 或 q_y 的最近邻为 p_u , 此时 p_x 与 q_v 之间的距离或者 q_y 与 p_u 之间的距离一定小于 p_x 与 q_y 之间的距离。而这与定义 4 中定义的线段最近对的距离小于两个线段集中任意一对线段对的距离相矛盾, 证毕。

定理 2 已知线段集 $P = \{p_1, p_2, \dots, p_n\}, Q = \{q_1, q_2, \dots, q_m\}$, 若 $\langle p_x, q_y \rangle$ ($p_x \in P \wedge q_y \in Q$) 是互为最近邻线段对, 则 p_x 一定与 $VL(q_y)$ 相交且 q_y 一定与 $VL(p_x)$ 相交。

证明 (反证法): 假设 p_x 与 $VL(q_y)$ 不相交, 由定义 1, 线段 Voronoi 图将平面分成若干个相连区域, 可知 p_x 一定与其它一个 Voronoi 区域或几个 Voronoi 区域相交。不失一般性, 假设 p_x 与 $VL(q_i)$ 相交, 则此时 p_x 中一定有点落入 $VL(q_i)$ 。依据定义 1, 落入 Voronoi 区域内的点与生成线段的距离小于 Voronoi 区域外的点到该线段的距离, 可知 p_x 与 q_i 之间的距离小于 p_x 与 q_y 之间的距离。而由已知 p_x, q_y 是互为最近邻线段对可知, q_y 是 p_x 的最近邻, 矛盾, 为此假设不成立, 得到 p_x 一定与 $VL(q_y)$ 相交; 同理可得, q_y 一定与 $VL(p_x)$ 相交。证毕。

找到互为最近邻线段对的方法:

给定线段集 $P = \{p_1, p_2, \dots, p_n\}, Q = \{q_1, q_2, \dots, q_m\}$, 构造两个线段集的 Voronoi 图, 首先找到线段集 P 中每条线段的互为最近邻线段对, 然后找到线段集 Q 中每条线段的互为最近邻线段对。对于 P 中每一条线段 p_i , 找到 $VL(Q)$ 中与

p_i 相交的 Voronoi 区域。在这些 Voronoi 区域中,计算每条生成线段与 p_i 之间的距离,距离最小的生成线段 q_j 是 p_i 的最近邻;对于 q_j ,找到 $VL(P)$ 中与 q_j 相交的 Voronoi 区域,在这些 Voronoi 区域中,计算每条生成线段与 q_j 之间的距离,距离最小的生成线段 p_m 是 q_j 的最近邻,再判断 p_m 是否就是 p_i 。若是,则 p_i 与 q_j 是互为最近邻线段对,否则, p_i 与 q_j 不是互为最近邻线段对。同理可处理 Q 集。如果存在两条线段相交,则其对应最近距离为 0,为此互相相交线段对一定是线段最近对。本文规定,线段最近对的结果集中包含相交的线段对,除此之外还包含不相交但距离最小的线段对。

图 2 为两个线段集 $P=\{p_1, p_2, \dots, p_{10}\}, Q=\{q_1, q_2, \dots, q_{10}\}$ 的 Voronoi 图。对于线段 p_1 ,因为只与 $VL(q_1)$ 相交,所以 q_1 是 p_1 的最近邻;而对于 q_1 ,因为 q_1 与 $VL(p_1)$ 、 $VL(p_2)$ 和 $VL(p_3)$ 相交,所以需要计算 p_1 、 p_2 和 p_3 与 q_1 之间的距离,而其中 p_1 与 q_1 之间的距离是最小的,因此 p_1 是 q_1 的最近邻。最后得到 p_1 与 q_1 是互为最近邻的线段对。对于线段相交的情况,如线段 p_4 ,与 $VL(q_2)$ 和 $VL(q_4)$ 相交,而 p_4 与 q_4 相交,为此 p_4 与 q_4 一定是结果;除此之外还要找到不相交线段的最近对,为此还要判断 p_4 与 q_2 是否互为最近邻, q_4 与 p_3 是否互为最近邻,最后判断为 p_4 与 q_2 是候选, q_4 与 p_3 不是候选。

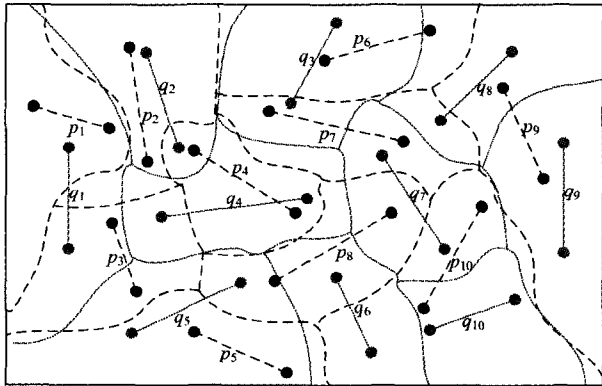


图 2 两个线段集的 Voronoi 图

计算线段之间距离的方法如下,线段与 Voronoi 区域相交分 3 种情况:

1) 一个端点落入 Voronoi 区域。如图 2 中线段 q_1 ,上端点落入 $VL(p_1)$,此时计算 p_1 与 q_1 之间的距离,只需计算 q_1 的上端点与 p_1 之间的距离。

2) 两个端点落入 Voronoi 区域。如图 2 中线段 q_9 ,两个端点都落入了 $VL(p_9)$,此时计算 p_9 与 q_9 之间的距离,可按文献[8]中线段与线段最近距离的算法来计算。

3) 没有端点落入 Voronoi 区域。如图 2 中线段 q_1 ,两个端点都没落入 $VL(p_2)$,但线段 q_1 与 $VL(p_2)$ 相交,此时计算 p_2 与 q_1 之间的距离,可按文献[8]中线段与线段最近距离的算法来计算。

3.2 基于 Voronoi 图的线段最近对查询算法

算法的基本思想:给定两个线段集 P 和 Q ,首先构造 P 线段 Voronoi 图 $VL(P)$ 和 Q 线段 Voronoi 图 $VL(Q)$ 。对于 P 中的每条线段 p ,找到 Q 中与 p 互为最近邻的线段 q ,则线段对 $\langle p, q \rangle$ 成为线段最近对的候选。同理找到 Q 中的每条线

段 q' 的互为最近邻的线段 p' ,则线段对 $\langle p', q' \rangle$ 成为线段最近对的候选。最后从候选集中选择距离最小的线段对。由此可得基于 Voronoi 图的线段最近对查询算法如下。

算法 1 VLCP(P, Q) (基于 Voronoi 图的线段最近对查询算法)

输入:线段集 $P=\{p_1, p_2, \dots, p_n\}, Q=\{q_1, q_2, \dots, q_m\}$;

输出:线段集 P 与线段集 Q 中的最近对 R 。

begin

$R=\Phi, C=\Phi$; // R 为结果集, C 是候选集 //

生成 P 的线段 Voronoi 图 $VL(P)$,生成 Q 的线段 Voronoi 图 $VL(Q)$;

(1) for each $p \in P$ do

for $VL(Q)$ 中与 p 相交的每个 $VL(q)$ 所对应的生成线段 q do

if $p \cap q \neq \Phi$ then

$R=R \cup \{(p, q), 0\}$;

else 计算 $d(p, q)$;

对于具有最小值的线段 q do

for $VL(P)$ 中与 q 相交的每个 $VL(p')$ 所对应的生成线段 p' do

if $p' \cap q \neq \Phi$ then

$R=R \cup \{(p', q), 0\}$;

else 计算 $d(p', q)$;

对于具有最小值的线段 p' do

if $p'=p$ then

$C=C \cup \{(p', q), d\}$;

(2) for each $q \in Q$ 且 $q \notin C$ do

for $VL(P)$ 中与 q 相交的每个 $VL(p'')$ 所对应的生成线段 p'' do

if $p'' \cap q \neq \Phi$ then

$R=R \cup \{(p'', q), 0\}$;

else 计算 $d(p'', q)$;

对于具有最小值的线段 p'' do

for $VL(Q)$ 中与 p'' 相交的每个 $VL(q')$ 所对应的生成线段 q' do

if $p'' \cap q' \neq \Phi$ then

$R=R \cup \{(p'', q'), 0\}$;

else 计算 $d(p'', q')$;

对于具有最小值的线段 q' do

if $q'=q$ then

$C=C \cup \{(p'', q), d\}$;

将 C 中对象按照 d 值排序;

具有最小 d 值的 (p, q) 即为结果;

$R=R \cup \{(p, q), d\}$;

end

3.3 对数据集进行更新的处理

3.3.1 插入线段的情况

一个数据集中插入了新的线段,若则找到插入线段在另一个数据集中的最近邻线段,将其距离与最近对结果中的距离进行比较。如果该距离小于结果中的距离,则改变结果,同时更新插入线段的局部 Voronoi 图,以便再插入新的线段时计算最近对。如果两个数据集同时插入新的线段,可按此方法做相应处理。

算法 2 INSERT-VLCP(P, Q) (插入线段的更新算法)

输入:线段集 $P=\{p_1, p_2, \dots, p_n\}, Q=\{q_1, q_2, \dots, q_m\}$, 线段集 P 中插入的线段 p ;

输出:更新的最近对 R ;

```

begin
  for VL(Q)中与 p 相交的每个 VL(q)所对应的生成线段 q do
    if  $p \cap q \neq \emptyset$  then
       $R = R \cup \{(p, q), 0\}$ ;
    else 计算  $d(p, q)$ ;
  对于  $r \in R$  且  $r.d \neq 0$  do
    if  $d_{\min} < r.d$  then //  $d_{\min}$  为最小距离//
       $R = R - r$ ;
       $R = R \cup \{(p, q), d_{\min}\}$ ;
  重构线段 p 局部 Voronoi 图;
end

```

3.3.2 删除线段的情况

一个线段集中删除了一条线段,若删除线段是结果集中的线段,且该线段与另一个线段集中的其他线段相交,则删除相交线段对即可;否则需要找到新的最近对。

算法3 DELETE-VLCP(P,Q)(删除线段的更新算法)

输入:线段集 $P = \{p_1, p_2, \dots, p_n\}$, $Q = \{q_1, q_2, \dots, q_m\}$, 线段集 P 中删除的线段 p;

输出:更新的最近对 R。

```

for  $r \in R$  且  $r.d = 0$  do
  if  $p \in r$  then
     $R = R - r$ ;
  for  $r \in R$  且  $r.d \neq 0$  do
    if  $p \in r$  then
       $R = R - r$ ;
      找到算法 1 中 C 的 d 值次小的对象  $(u, v)$ ;
       $R = R \cup \{(u, v), d_{\min}\}$ ; //  $d_{\min}$  为当前的最小距离//
end

```

4 实验结果

实验是在 Pentium IV 2.8 GHz CPU, 512MB 内存, Windows XP 平台上用 Visual C++ 6.0 实现的。

实验将本文算法与线性扫描算法(记为 DLCP)和 K-CPQ ($K=1$)进行比较。DLCP 依次计算一个线段集中每条线段与另一个线段集中每条线段之间的距离,并从中选取距离最小的线段对。

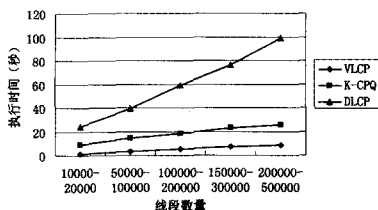


图3 算法 VLCP 和算法 DLCP 及 K-CPQ 的比较

随机生成两个线段集,其线段数量分别为 10000, 20000, 50000, 100000, 100000, 200000, 150000, 300000, 200000, 500000, 实验比较 3 个算法的执行时间。图 3 给出了算法 VLCP 和算法 DLCP 及 K-CPQ 的执行时间的比较。由图 3 可知,算法 VLCP 的执行时间最短,算法 DLCP 的执行时间最长。这是因为随着线段数量的增加,算法 VLCP 通过 Voronoi 图划分区域,去掉了大量的候选,减少了比较次数,节省了大量时间。

结束语 本文给出了基于平面线段的最近对查询的定义,提出了解决该查询的方法和相关的定理及其证明。利用 Voronoi 图的邻接特性和局域动态特性确定候选,去掉了大量不可能成为候选的数据,减少了大量的距离计算。对数据集中增加线段和删除线段的情况分别进行了处理。通过实验验证了算法具有较好的性能。

参考文献

- [1] Corral A, Manolopoulos Y, Theodoridis Y, et al. Closest pair queries in spatial databases [C]//Proc. of 2000 ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 2000: 189-200
- [2] Corral A, Manolopoulos Y, Theodoridis Y, et al. Algorithms for processing k-closest-pair queries in spatial databases [J]. Data and Knowledge Engineering, 2004, 49(1): 67-104
- [3] Yang C, Lin K-I. An index structure for improving closest pairs and related join queries in spatial databases [C]//Proc. of the International Database Engineering and Applications Symposium. Piscataway: IEEE, 2002: 140-149
- [4] 徐红波, 郝忠孝. 一种基于 Z 曲线近似 k-最近对查询算法 [J]. 研究计算机研究与发展, 2008, 45(2): 310-317
- [5] Liu X, Liu Y, Xiao Y. Processing constrained K closest pairs query in spatial databases [J]. Wuhan University Journal of Natural Sciences, 2006, 11(3): 543-546
- [6] Bepamyatnikh S, Snoeyink J. Queries with segments in voronoi diagrams [C]//Proc. of the 10th Annual ACM-SIAM Symp on Discrete Algorithms. Baltimore: ACM SIAM, 1999: 122-129
- [7] 郝忠孝, 王玉东, 何云斌. 空间数据库平面线段近邻查询问题 [J]. 计算机研究与发展, 2008, 45(9): 1539-1545
- [8] 杨泽雪, 郝忠孝. 基于 Voronoi 图的线段反向最近邻查询 [J]. 计算机工程, 2011, 37(16): 30-32
- [9] 郝忠孝. 时空数据库查询与推理 [M]. 北京: 科学出版社, 2010: 119-120
- [10] 周培德. 计算几何 [M]. 北京: 清华大学出版社, 2005: 235-237
- [11] Kolahdouzanm, Shahab I C. Voronoi-based k nearest neighbor search for spatial network databases [C]//Proc. of VLDB. Toronto, Canada: [s. n.], 2004

(上接第 110 页)

- [6] Zhang Mao-yuan, Lu Zheng-ding, Zou Chun-yan. A Chinese word segmentation based on language situation in processing ambiguous words [J]. Information Sciences, 2004, 162(3/4): 275-285
- [7] Apache Lucene [OL]. <http://lucene.apache.org/>, 2011-02-18
- [8] 刘永丹, 曾海泉, 李荣陆, 等. 基于语义分析的倾向性文本过滤 [J]. 通信学报, 2004, 25(7)
- [9] 赵佳鹤, 王秀冲, 刘亚欣. 基于语义分析的主题信息采集系统的设计与实现 [J]. 计算机应用, 2007, 27(2)

- [10] Gildea D, Jurafsky D. Automatic labeling of semantic roles [J]. Computer, Linguist, 2002, 28(3): 245-288
- [11] LTP [OL]. <http://ir.hit.edu.cn/ltp/>, 2011
- [12] 郑榕增, 林世平. 基于 Lucene 的中文倒排索引技术的研究 [J]. 计算机技术与发展, 2010, 20(3)
- [13] 吴众欣, 沈家力. Lucene 分析与应用 [M]. 北京: 机械工业出版社, 2008