第 45 卷 第 6 期 2018 年 6 月
计 算 机 科 学 COMPUTER SCIENCE
Vol. 45 No. 6 June 2018

障碍环境中线段组最近邻查询方法研究

郭莹莹 张丽平 李 松

（哈尔滨理工大学计算机科学与技术学院 哈尔滨 150080）

摘 要 为了解决现有成果无法有效处理障碍环境下的线段组最近邻查询问题，提出了障碍环境中线段组最近邻查询方法。查询过程分为过滤阶段和精炼阶段两个部分。在过滤过程中，首先根据线段 Voronoi 图的性质以及线段障碍组最近邻查询的定义，提出了针对数据线段的剪枝定理，并提出了 OLGNN_Line_Filter 算法；根据线段障碍距离的定义，进一步提出针对障碍物的剪枝定理，并给出了 OLGNN_Obstacle_Filter 算法。在精炼过程中，为了得到更精确的查询结果，提出了相应的精炼定理和精炼算法 STA_OLGNN。理论研究和实验表明，所提算法能够有效地处理障碍环境下的线段组最近邻查询问题。

关键词 空间数据库，线段障碍距离，线段障碍组最近邻，线段 Voronoi 图

中图法分类号 TPP31.13 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.06.030

Group Nearest Neighbor Query Method of Line Segment in Obstacle Space

GUO Ying-ying ZHANG Li-ping LI Song

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

Abstract In order to make up the deficiencies of the existing research results which can not effectively deal with the group nearest neighbor query based on line segment in obstacle space, the group nearest neighbor query method of line segment in obstacle space was proposed. This query process is divided into two stages, including the filtering process and refining process. In the filtering process, according to the properties of the line segment Voronoi graph and the definition of OLGNN, corresponding pruning theorem of data line was proposed, and the OLGNN_Line_Filter algorithm was put forward. According to the definition of line obstruction distance, corresponding pruning theorem of obstruction was proposed, and the OLGNN_Obstacle_Filter algorithm was put forward. In the refining process, in order to get more accurate query results, the corresponding refining theorem and STA_OLGNN algorithm were proposed. Theoretical research and experimental results show that the proposed algorithm can effectively deal with the problem of group nearest neighbor query of line segment in obstacle environment.

Keywords Spatial database, Line segment obstruction distance, OLGNN, Voronoi diagram of line segment

1 引言

组最近邻查询[1-2]是传统最近邻查询[3]的扩展形式之一。目前，国内外学者就组最近邻查询问题已提出了多种方法。例如，文献[4]提出了一种处理模糊地理空间对象的组最近邻查询方法。文献[5]提出了高效概率阈值组最近邻查询算法。文献[6]提出了一种基于位置隐私保护的组最近邻查询问题的方法。现实生活中，查询往往受地理环境的限制，需要考虑空间对象间存在障碍物的情况。为此，文献[7]实现了障碍空间中查询点的影响范围的查找。文献[8]提出了一种障碍空间中保持位置隐私的最近邻查询方法。文献[9]解决了障碍空间中的最近邻查询问题。

在对实际问题进行查询时，如果将某些查询对象抽象为点进行研究将会对查询结果的精确度造成较大影响，例如山脉、河流、大型建筑等。因此，文献[10]首次提出了空间数据库中基于线段的最近邻查询算法。文献[11]根据 SI-树提出了一种空间数据库平面线段快速最近邻查询算法。文献[12]利用 Voronoi 图的最近邻近特性和局域动态特性找到了互为最近邻的线段对。文献[13]解决了移动线段对象的最近邻查询问题。

本文算法的查询过程分为过滤过程和精炼过程两部分。针对数据线段的过滤，提出了相关剪枝定理，并给出了剪枝算法 OLGNN_Line_Filter。为了进一步减少无关数据对查询的影响，提出了针对障碍物的剪枝算法 OLGNN_Obstacle_Filter。在精炼过程中，给出了比较障碍距离的定理，并在过滤算法的基础上提出了 STA_OLGNN 算法，从而得到障碍环境

到稿日期：2017-04-27 返修日期：2017-06-05 本文受黑龙江省教育厅科学技术研究项目(12531z004)资助。

郭莹莹(1993－)，女，硕士生，CCF 会员，主要研究方向为数据挖掘、数据查询；**张丽平**(1976－)，女，硕士，副教授，CCF 会员，主要研究方向为数据结构和算法设计、数据库，E-mail:zhangliping@0703@163.com(通信作者)；**李 松**(1977－)，男，博士，副教授，CCF 会员，主要研究方向为空间数据查询与处理、算法设计。

中的线段组最近邻查询结果。

2　基础定义

本文在障碍最近距离[9]、线段最近距离[10]、k 级邻接生成线段[14]定义的基础上，进一步给出了如下定义。

定义 1(线段可视化)　障碍环境中任意两条线段 l_i 和 l_j 分别连接两线段的左、右端点，形成多边形 P。如果 P 与所有障碍物均不相交，则称线段 l_i 与线段 l_j 完全可视；如果存在障碍物与 P 相交的边数大于一条，则称线段 l_i 与线段 l_j 完全不可视。

定义 2(线段障碍最近距离)　障碍环境中存在查询线段 l_q，设其左右两端点分别为 m 和 n。如果两线段完全可视，两线段的欧氏距离为最短可视化距离 $MVdist(l_i,l_q)$，则障碍距离等于两线段的最短可视化距离，即 $Odist(l_i,l_j)=MVdist(l_i,l_j)$。如果两线段之间完全不可视，则两线段的障碍距离为这两线段绕过障碍物的最短距离，即 $Odist=\min\{dist(m,O_k)+dist(O_k,l_i),dist(n,O_k)+dist(O_k,l_i)\}$。

定义 3(线段障碍组最近邻查询)　障碍环境中，给定障碍物集合 $O=\{O_1,O_2,\cdots,O_k\}$，查询线段集 $Q=\{l_{q1},\cdots,l_{qm}\}$，生成线段集 $L=\{l_1,\cdots,l_n\}$，生成线段 l 到查询线段集 Q 的障碍距离为 $Odist(l,Q)=\sum_{i=1}^{m}|ll_{qi}|$，则线段障碍组最近邻查询定义为 $OLGNN(Q)=\{l_i|Odist(l_i,Q)<Odist(l_j,Q),i\neq j\}$。

线段障碍组最近邻查询示例如图 1 所示。

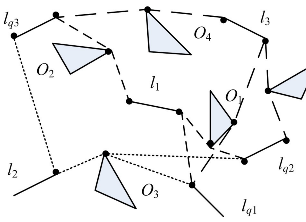

图 1　线段障碍组最近邻查询示例

Fig. 1　Query example of OLGNN(Q)

3　障碍环境中线段最近邻查询算法

3.1　过滤过程

在数据线段的过滤过程中，为便于研究，将每个障碍物抽象为三角形进行表示，构成障碍物集合 $O=\{O_1,O_2,\cdots,O_k\}$，O 随机地分布在 $VL(L)$ 中。抽取除数据线段外的任意线段组成查询线段数据集 $Q=\{l_{q1},l_{q2},\cdots,l_{qm}\}$，$Q$ 随机地分布在 $VL(L)$ 中。

定理 1　障碍环境中，给定数据线段 l_j 为 l_i 的邻接生成线段，则距离 l_i 最近的数据线段一定存在于 l_j 所在的 Voronoi 多边形中。

证明：利用反证法进行证明。如图 2 所示，数据线段 l_2 为 l_1 的邻接生成线段，数据线段 l_3 为 l_1 的 2 级邻接生成线段。根据线段距离定义 $dist(l_1,l_2)=dist(a,m)+dist(m,b)$，同理定义 $dist(l_1,l_3)=dist(a,n)+dist(n,c)$。假设 $dist(l_1,l_2)>dist(l_1,l_3)$，由线段 Voronoi 图[14]特性可知 $dist(c,n)=dist(n,b)$，且在欧氏空间中满足三角不等式定理，因此 $dist(n,b)+dist(a,b)>dist(a,n)$，$dist(l_1,l_2)<dist(l_2,l_3)$，与假设矛盾。故原定理成立。证毕。

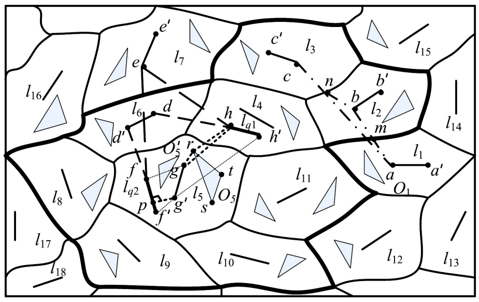

图 2　剪枝定理证明情况

Fig. 2　Proof of pruning theorem

由定理 1 可进一步得到推论 1。

推论 1　距离查询线段 l_q 最近的数据线段一定存在于其所在的 Voronoi 多边形或与其邻接的 Voronoi 多边形中。

定理 2　查询线段集 Q 的线段障碍组最近邻，一定存在于查询线段所在的 Voronoi 多边形和与其邻接的 Voronoi 多边形外侧边界所组成的区域 E 内。

证明：利用反证法进行证明。如图 2 所示，给定查询线段集 $Q=\{l_{q1},l_{q2}\}$，图 2 中粗黑色边界所包围的区域为 E。根据线段障碍组最近邻定义，$Odist(l_6,Q)=dist(f,d')+dist(h,d)$。同理，$Odist(l_7,Q)=dist(f,e)+dist(h,e)$。假设 $Odist(l_6,Q)>Odist(l_7,Q)$。根据推论 1，进一步可推导得到 $Odist(l_6,Q)<Odist(l_7,Q)$，与假设矛盾。故原定理成立。证毕。

结合以上针对数据线段剪枝定理的论述，给出 OLGNN_Line_Filter 算法。该算法的主要思想为：首先根据所给定的数据集构造障碍环境中的线段 Voronoi 图；然后根据定理 1 和定理 2 划分区域 E，对区域 E 外的数据线段进行剪枝，将区域 E 内的数据线段加入到数据线段的候选集合 S_C 中。

基于以上讨论，进一步给出数据线段的过滤算法，如算法 1 所示。

算法 1　OLGNN_Line_Filter(L,O,Q)

输入：数据线段集合 L，障碍物集合 O，查询线段集合 Q

输出：过滤后的数据线段候选集合 S_C

```
Begin
1. Create_Voronoi(L,O,Q);
2. SC←∅;//初始化集合 SC
3. Judge_Position(Q);//判断查询线段的位置
4.   if lq∈VL(li) then
5.     SC←SC+li;//推论 1
6.   if VL(lj)是 VL(li)的邻接多边形 then
7.     E=VL(li)∪VL(lj);//定理 2
8.     SC←SC+lj;
9.     if VL(lk)∉E then
10.      SC←SC-lk;
11. return SC;
End
```

为了进一步排除无关数据对算法效率的影响，提出了针对障碍数据的剪枝定理 3 和定理 4，并给出了相应的证明过程。

定理 3　存在于区域 E 外的障碍物对线段障碍组最近邻查询没有影响。

证明：给定查询线段集合 $Q=\{l_{q1},l_{q2}\}$，根据定理 2 划分

区域 E，如图 2 中的粗黑色边界所示。O_1 属于 $VL(l_1)$ 且为区域 E 外的障碍物。由定理 2 可知，Q 中查询线段的组最近不可能存在于区域 E 外的数据线段中，即 l_1 不属于 S_C，从而可以得出查询过程中障碍物 O_1 对查询没有影响。证毕。

定理 4 区域 E 内存在数据线段 $l_i \in S_C$，障碍物 $O_i \in VL(l_i)$，假设 l_i 与 Q 之间存在最短可视化距离 $MVdist(l_i,Q)=\{MVdist(l_i,l_{q1}),MVdist(l_i,l_{q2}),\cdots,MVdist(l_i,l_{qm})\}$，如果 l_i 与 $MVdist(l_i,Q)$ 集合中的任意最短可视化距离都不相交，则 O_i 对查询没有影响。

证明：给定查询线段集合 $Q=\{l_{q1},l_{q2}\}$，存在障碍物 O_5' 和 O_5，如图 2 所示。根据线段最短可视化距离定义可得：$MVdist(l_5,l_{q1})=dist(g,h)$，$MVdist(l_5,l_{q2})=dist(g',p)$。$l_5$ 与 l_{q1} 之间的障碍距离表示为 $Odist(l_5,l_{q1})=dist(g,r)+dist(g,h)$，影响 l_5 到 Q 的距离，因此障碍 O_5 对查询存在影响。同理，障碍物 O_5' 与 $dist(g,h)$ 和 $dist(g',p)$ 都不相交，此时 l_5 到 Q 的障碍距离没有发生改变，因此障碍 O_5' 对查询没有影响。证毕。

根据定理 3 和定理 4，进一步给出针对障碍物数据的过滤算法。该算法的主要思想为：首先根据定理 3 找到区域 E 的范围，并将属于区域 E 内的障碍物加入到障碍物候选集合 S_O 中。通过判断障碍物与最短可视化距离的位置关系，对 S_O 中的障碍物数据进行进一步筛选。

基于以上讨论，进一步给出障碍物数据的过滤算法，如算法 2 所示。

算法 2 OLGNN_Obstacle_Filter(L,O,Q)

输入：数据线段集合 L，障碍物集合 O，查询线段集合 Q

输出：过滤后的障碍物候选集合 S_O

```
Begin
1. S_O←∅；//初始化集合 S_O
2. Find_Position(E)；//划分区域 E
3. Judge_Position(O_x)；//判断障碍物的位置
4.   if O_x∈E then //定理 3
5.     S_O←S_O+O_x；
6.     if l_i∈S_C and O_i∈VL(l_i) then //定理 4
7.       Calculate_MVdist(l_i,Q)；
8.       if O_i∩MVdist(l_i,Q)=NULL then
9.         S_O←S_O−O_i；
10. return S_O；
End
```

3.2 精炼过程

定理 5 存在数据线段 l_i 和查询线段 l_q，连接两线段端点形成四边形 T，$O_i \in VL(l_i)$ 且与 T 存在交集，如果 l_i 与 l_q 之间存在最短可视距离 $MVdist(l_i,l_q)$，则 $MVdist(l_i,l_q)<Odist(l_i,l_q)$。

证明：根据最短可视化距离定义可将 l_8 与 l_{q1} 之间的最短可视化距离表示为 $MVdist(l_8,l_{q1})=dist(a',h)$，如图 3 所示。根据障碍距离定义可得，$Odist(l_8,l_{q1})=dist(f,p)+dist(p,a)$。连接两线段端点形成四边形 $ff'aa'$，得到不等式 $dist(f,a)>dist(a',h)$。由于欧氏空间满足三角不等式定理，因此 $MVdist(l_8,l_{q1})<Odist(l_8,l_{q1})$。证毕。

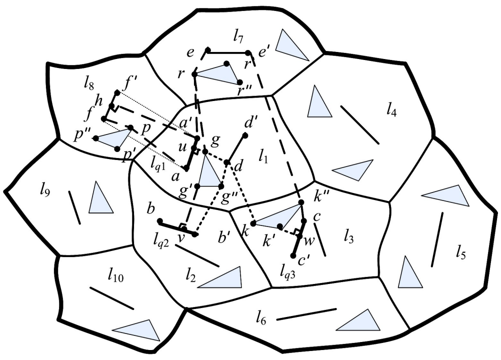

图 3 精炼定理证明情况

Fig. 3 Proof of refining theorem

定理 6 区域 E 内假设数据线段 l_i，l_j 与查询线段集 Q 之间都不存在最短可视化距离，将 l_i 到 Q 所经过的 Voronoi 多边形总个数记为 M，l_j 到 Q 所经过的 Voronoi 多边形总个数记为 N，如果 $M<N$，则 $Odist(l_i,Q)<Odist(l_j,Q)$。

证明：利用反证法进行证明。给定查询线段集 $Q=\{l_{q1},l_{q2},l_{q3}\}$，如图 3 所示。假设 $Odist(l_1,Q)>Odist(l_7,Q)$，根据障碍环境中线段组最近邻查询定义可得：$Odist(l_1,Q)=dist(d,g)+dist(g,u)+dist(d,g'')+dist(g'',b')+dist(d,k)+dist(k',w)$。同理，$Odist(l_7,Q)=dist(e,r)+dist(r,a')+dist(e,r)+dist(g',v)+dist(e',k'')+dist(k'',c)$。此时，$l_1$ 到 Q 所经过的 Voronoi 多边形个数 $M=2$，l_7 到 Q 所经过的 Voronoi 多边形个数 $N=4$，根据推论 1 可得 $Odist(l_1,l_{q3})<Odist(l_7,l_{q3})$，进一步推导可得 $Odist(l_1,Q)<Odist(l_7,Q)$，与假设矛盾。故原定理成立。证毕。

根据以上理论基础，提出精炼算法。该算法的主要思想为：首先，结合过滤算法中得到的 S_C 集合中的候选数据线段，以及 S_O 集合中的候选障碍物，确定查找范围；然后，判断 S_C 集合中的数据线段与 Q 中的查询线段受障碍物影响的情况，进一步根据定理 5 和定理 6 精炼 S_C 集合，最终得到准确的查询结果集 S_R。

基于以上讨论，进一步给出精炼算法，如算法 3 所示。

算法 3 STA_OLGNN(S_C,S_O,Q)

输入：数据线段候选集 S_C，障碍物候选集 S_O，查询线段集 Q

输出：障碍环境中的线段组最近邻查询结果集 S_R

```
Begin
1. S_R←∅；//初始化集合 S_R
2. count=0；//初始化计数变量
3. sumcount_M=0；
4. sumcount_N=0；
5. 根据 S_C 和 S_O 确定查找范围；
6. if l_k 与 Q 之间存在 MVdist(l_k,Q) then //定理 5
7.   Odist(l_k,Q)=MVdist(l_k,Q)；
8.   S_R←S_R+l_k；
9. else
10.  遍历 l_i 到 Q 中查询线段所经过的 VL(l)；//定理 6
11.    sumcount_M=count+1；
12.  遍历 l_j 到 Q 中查询线段所经过的 VL(l)；
13.    sumcount_N=count+1；
14.    if sumcount_M< sumcount_N then
```

```
        Odist(l_i,Q)<Odist(l_j,Q);
        S_R←S_R+l_i;
return S_R;
End
```

4 实验

本节通过实验对所提算法的性能进行分析。实验环境为：2.0 GHz CPU，4GB 内存，500 GB 硬盘，Windows7 系统。实验的数据集采用美国旧金山和加州的路网信息[15]。在实验过程中，对数据的分布和相关信息进行了适当的调整。

由于现有障碍环境中的组最近邻查询算法都是以点为查询对象进行的，因此无法将现有方法与本文方法进行直接比较。为了得到比较算法，分别以文献[11]提出的基于 SI-树的 LNN 算法和文献[12]提出的基于 Voronoi 图的 LkNN 算法为基础，对两种算法进行适当的调整，从而实现障碍环境中的线段组最近邻查询，形成两种对比算法。本文将这两种算法分别称为 OLGNN_Basic1 算法和 OLGNN_Basic2 算法。

首先，测试查询线段集 Q 中线段数量对算法运行时间的影响。将数据线段数量设定为 20000 条，障碍物数量设定为 1000 个，测试结果如图 4 所示。从图 4 中可以看出，本文算法的 CPU 增长幅度较小，其原因在于两种对比算法需要计算 Q 中每条查询线段的线段障碍最近邻，会造成搜索区重合，花费了较多的查询时间。

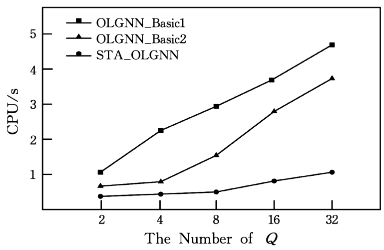

图 4　Q 对查询运行时间的影响

Fig. 4　Effect of Q on query running time

然后，测试障碍物数量 O 对算法运行时间的影响。将数据线段数量设定为 20000 条，查询线段数量设定为 8，实验结果如图 5 所示。STA_OLGNN 算法的运行时间受障碍物数量改变的影响较小，其原因在于本文算法中有针对障碍物的过滤过程，在查询过程中只需要考虑有效障碍物对查询的影响即可，减少了算法运行的 CPU 时间。

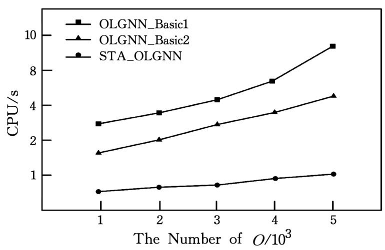

图 5　O 对查询运行时间的影响

Fig. 5　Effect of O on query running time

最后，测试数据规模 L 的大小对 3 种算法运行时间的影响。将查询线段数量设定为 8，障碍物数量设定为 1000，比较结果如图 6 所示。相较于两种对比算法，本文所提 STA_OLGNN 算法的 CPU 运行时间随着数据量的增加变化幅度较小，其原因在于所提算法只需要判断查询范围内的数据线段即可，因此具有较高的查询效率。

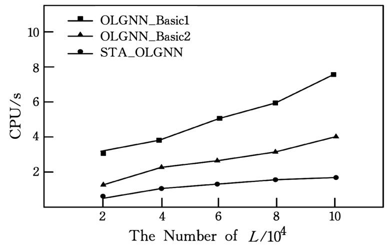

图 6　L 对查询运行时间的影响

Fig. 6　Effect of L on query running time

结束语　本文提出了障碍环境中的线段组最近邻查询算法，解决了在障碍环境下需要精确测量时的组最近邻查询问题。实验证明，本文所提算法具有较高的效率。未来研究重点将集中在基于网格数据的障碍线段组最近邻查询研究中。

参 考 文 献

[1] ROUMELIS G, VASSILAKOPOULOS M, CORRAL A, et al. Plane-sweep algorithms for the k group nearest-neighbor query [C] // International Conference on Geographical Information Systems Theory, Applications and Management. 2015: 83-93.

[2] SON W, BAE S W, AHN H K. Group nearest-neighbor queries in the L 1 plane[J]. Theoretical Computer Science, 2015, 592 (C): 39-48.

[3] ZHANG L P, ZHAO J Q, LI S, et al. Research on Methods of Construction of Voronoi Diagram and Nearest Neighbor Query in Constrained Regions[J]. Computer Science, 2014, 41(9): 220-224. (in Chinese)
张丽平，赵纪桥，李松，等. Voronoi 图的构建与受限区域内的最近邻查询方法研究[J]. 计算机科学，2014，41(9)：220-224.

[4] NURAIN N, ALI M E, HASHEM T, et al. Group nearest neighbor queries for fuzzy geo-spatial objects [C] // International ACM Workshop on Managing and Mining Enriched Geo-Spatial Data. New York: ACM, 2015: 25-30.

[5] CHEN M, JIA Z X, GU Y, et al. Group nearest neighbor queries over existentially uncertain data[J]. Journal of Chinese Computer Systems, 2012, 33(4): 684-687. (in Chinese)
陈默，贾子熙，谷峪，等. 面向存在不确定对象的组最近邻查询方法[J]. 小型微型计算机系统，2012，33(4)：684-687.

[6] HASHEM T, ALI M E, KULIK L, et al. Protecting privacy for group nearest neighbor queries with crowd sourced data and computing[C] // 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Zurich, Switzerland, 2013. New York, NY, USA: ACM, 2013: 559-562.

[7] YU X N, GU Y, ZHANG T C, et al. A method for reverse k-nearest neighbor queries in obstructed spaces[J]. Chinese Journal of Computers, 2011, 34(10): 1917-1925. (in Chinese)
于晓楠，谷峪，张天成，等. 一种障碍空间中的反 k 最近邻查询方法[J]. 计算机学报，2011，34(10)：1917-1925.

（下转第 192 页）