ISSN1000-9825 Journal of Software 软 件 学 报 2000,11(9):1237~1240

求一个包含点集所有点的最小圆的算法*

汪卫[1] 王文平[2] 汪嘉业[3]

[1](复旦大学计算机系 上海 200433)
[2](香港大学计算机系 香港)
[3](山东大学计算机系 济南 250100)
E-mail:weiwang1@fudan.edu.cn/jywangz@yahoo.com

摘要 提出一种算法,以解决求一个最小圆包含给定点集所有点的问题.证明了这种算法的时间复杂性为$O(|\lg(d/R)|*n)$,其中R是所求的最小圆的半径,d为点集中不在圆周上但距圆周最近的点到圆周的距离.

关键词 最小圆,计算几何.
中图法分类号 TP301

求一个最小圆包含给定点集所有点的问题是人们在实践和理论上都十分感兴趣的一个问题.由于这个圆的圆心是到点集中最远点最近的一个点,因而在规划某些设施时很有实用价值.这个圆心也可看成是点集的中心.此外,在图形学中,圆也常可取作边界盒,使用它可减少很多不必要的计算.在空间数据库中可将该问题用于建立空间数据的索引以提高查询速度[1,2].这个问题看起来十分简单,但用直观的算法去解此问题,其复杂性可达$O(n^4)$,其中n为点集中点的数目.有关此问题的讨论在计算几何的专著及论文中未见报道[3~5].本文提出了一种新的算法并证明了这种算法的时间复杂性为$O(|\lg(d/R)|*n)$,其中R是所求的最小圆的半径,d为点集中不在圆周上但距圆周最近的点到圆周的距离.

1 算 法

第1步.在点集中任取3点A,B,C.

第2步.作一个包含A,B,C三点的最小圆.圆周可能通过这3点(如图1所示),也可能只通过其中两点,但包含第3点.后一种情况圆周上的两点一定是位于圆的一条直径的两端.

第3步.在点集中找出距离第2步所建圆圆心最远的点D.若D点已在圆内或圆周上,则该圆即为所求的圆,算法结束.否则,执行第4步.

第4步.在A,B,C,D中选3个点,使由它们生成的一个包含这4点的圆为最小.这3点成为新的A,B和C,返回执行第2步.

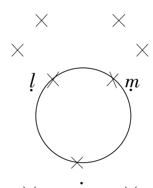

Fig 1
图1

若在第4步生成的圆的圆周只通过A,B,C,D中的两点,则圆周上的两点取成新的A和B,从另两点中任取一点作为新的C.

2 算法正确性

本节要证明上述算法一定能终止,且最后一次求得的圆即是所要求的包含点集所有点的最小圆.

引理1.算法第4步所生成的圆的半径随着迭代过程递增.

* 本文研究得到国家自然科学基金(No.69973028)资助.作者汪卫,1970年生,博士,讲师,主要研究领域为计算几何,数据库.王文平,1963年生,博士,副教授,博士生导师,主要研究领域为计算机图形学,计算几何,CAGD.汪嘉业,1937年生,教授,博士生导师,主要研究领域为计算机图形学,计算几何,CAGD.
本文通讯联系人:汪嘉业,济南 250100,山东大学计算机系
本文2000-02-28收到原稿,2000-05-12收到修改稿

证明. 因为第 4 步每一次生成的圆是包含原来的 A,B,C 三点，又要加上圆外的一点，而上一次生成的圆是包含 A,B,C 的最小圆，因而新圆的半径一定比原来的圆半径要大. □

定理 1. 上述算法是收敛的，且最后得到包含点集所有点的最小圆.

证明. 因为在点集中任取 3 点或两点生成的包含这 3 点或两点的最小圆的个数是有限的. 由引理 1 可知，算法进行过程中所生成的圆的半径是递增的，因而经过有限次迭代后，可求得这些最小圆中半径最大的一个. 从算法第 3 步可知，只有当点集中所有点都在由 3 个点或两个点生成的最小圆内时，算法才结束，因而最后得到的圆一定是包含点集中所有点的最小圆. □

3 时间复杂性

我们用 R 表示包含点集所有点的最小圆的半径，用 r_i 表示本文的算法迭代第 i 次所得到的最小圆的半径，O_i 表示其圆心. 设 A,B,C 是算法第 i 次迭代后所求得的 3 个点. D 是点集中距它们所生成的最小圆的圆心最远的点(如图 2 所示). 不失一般性，以后假设 C 点和 D 点在 AB 连线的同侧.

引理 2. 在 $|O_iD|$ 和 r_i 不变的情况下，当 A 和 B 位于圆的直径两端，且 $O_iD \perp AB$ 时，包含 A,B,C,D 四点的最小圆的半径为最小.

证明: A,B,C 三点不能同时位于一段小于 180°的弧上，否则，此时最小圆应是其中两点连线为直径的圆. 第 3 点在圆内.

设 AB 连线不是 O_i 圆的直径(如图 3 所示). 假设 AB' 是 O_i 圆的直径. 因为 B 只能和 D 处于 AB' 的不同侧，因而包含 ABD 的最小圆一定包含 B'，但包含 $AB'D$ 的最小圆可以不包含 B，因而，包含 ABD 的最小圆的半径只能大于或等于包含 $AB'D$ 的最小圆的半径. 故而为了找到最小圆半径最短的情况，我们只要讨论 AB 是 O_i 圆直径的情况. 设图 4 中 A_1B_1 是 O_i 圆的另一直径，包含 A_1,B_1,D 的最小圆的圆心为 F，包含 A,B,D 最小圆的圆心为 E. 它们的半径分别为

$$R_2 = DF = A_1F, \quad R_1 = DE = AE.$$

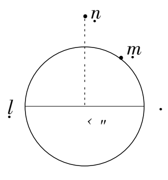

Fig. 2
图 2

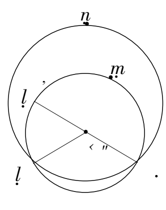

Fig. 3
图 3

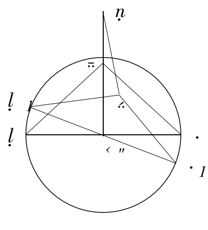

Fig. 4
图 4

当 $\angle DB_1A_1 \leqslant 90°$ 时，B_1 点在包含 B_1,A_1,D 的最小圆的圆周上，此时 $\angle FO_iA_i = 90°$. 当 $\angle DB_1A_1 > 90°$ 时，B_1 点在包含 B_1,A_1,D 的最小圆的圆内，此时，因为 DFA_1 在直线 O_iF 并行 B_1D，$\angle FO_iA_1 = \angle DB_1A_1$，故无论哪种情况都有 $\angle FO_iA_1 \geqslant 90°$. 在 $\triangle FO_iA_1$ 中有

$$|O_iF|^2 \leqslant |A_1F|^2 - |O_iA_1|^2,$$

即

$$|O_iF| \leqslant \sqrt{R_2^2 - r_i^2}. \tag{1}$$

在 $\triangle O_iEA$ 中有

$$|O_iE| \leqslant \sqrt{R_1^2 - r_i^2},$$

在 $\triangle O_iFD$ 中有

$$|O_iF| + |DF| \geqslant |O_iE| + |ED|,$$

即

$$R_2 + \sqrt{R_2^2 - r_i^2} \geqslant R_1 + \sqrt{R_1^2 - r_i^2}. \tag{2}$$

由式(2)易知 $R_2 \geqslant R_1$. 这就证明了引理 2 的结论. □

在引理 2 中,若包含 A, B, C 的最小圆是以 AB 为直径的圆,而 C 在圆内,这种情况已包括在上述证明之内.

引理 3. 对于相邻二次迭代最小圆半径,有以下估计:

$$r_{i+1} - r_i \geqslant (1-\rho)(R - r_i), \quad 0 < \rho < 1. \tag{3}$$

证明:由引理 2 可知,只要证明 AB 是最小圆的直径以及 $O_iD \perp AB$ 的情况(如图 5 所示). 设包含 A, B, D 的圆的圆心是 E,即

$$r_{i+1} = |ED| = |EA| = |EB|.$$

由直角△O_iEA 得

$$|O_iE|^2 = |AE|^2 - |O_iA|^2,$$

$$(|OD| - r_{i+1})^2 = r_{i+1}^2 - r_i^2.$$

即

$$r_{i+1} = \frac{|OD|^2 + r_i^2}{2|OD|},$$

$$r_{i+1} - r_i = \frac{(|OD| - r_i)^2}{2|OD|}.$$

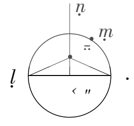

Fig 5
图 5

由算法可知 O_i 点在所求的包含点集所有点的最小圆内,故

$$2R > |O_iD| > R > r_i,$$

所以

$$r_{i+1} - r_i \geqslant \frac{(R - r_i)^2}{4R} \geqslant \frac{R + r_i}{4R}(R - r_i) \geqslant (1-\rho)(R - r_i), \quad 0 < \rho < 1. \qquad \square$$

在一般情况下,可以证明存在常数 ρ_0,使 $0 < \rho_0 < 1$ 且 $\rho < \rho_0$.

引理 4. 对于第 i 次迭代最小圆半径 r_i,有以下估计:

$$|R - r_i| \leqslant (\rho_0)^i R. \tag{4}$$

证明:由式(3)可知

$$|R - r_{i+1}| \leqslant \rho_0 |R - r_i|,$$

所以

$$|R - r_i| \leqslant \rho_0 |R - r_{i-1}| \leqslant \ldots \leqslant (\rho_0)^i |R - r_0| \leqslant (\rho_0)^i R. \qquad \square$$

设 d 是点集中不在所求的最小圆的圆周上的点到圆周的最短距离.

引理 5. 设算法共迭代 j 次后在求得最后结果,即 $r_j = R$, $r_{j-1} < r_j$,则有下式成立:

$$R - r_{j-1} > \frac{d^2}{8R}. \tag{5}$$

证明:由引理 2 知如图 6 所示的情况,求得的 r_j 最小,也即此情况 $R - r_{j-1}$ 为最小,而且这使得 d 最大. 因而只要证明图 6 情况,即式(5)成立,则式(5)对一般情况也成立.

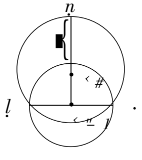

Fig 6
图 6

令

$$\varepsilon = |O_{j-1}O_j|, \quad r_{j-1} = \sqrt{|O_jA|^2 - |O_jO_{j-1}|^2} = \sqrt{R^2 - \varepsilon^2},$$

故

$$R - r_{j-1} = R - \sqrt{R^2 - \varepsilon^2}. \tag{6}$$

$$d = R + \varepsilon - r_{j-1} = R + \varepsilon - \sqrt{R^2 - \varepsilon^2},$$

$$\varepsilon^2 + (R - d)\varepsilon + \frac{1}{2}(d^2 - 2Rd) = 0,$$

$$\varepsilon = \frac{-(R-d) + \sqrt{(R-d)^2 - 2d^2 + 4Rd}}{2} = \frac{-(R-d) + \sqrt{R^2 - d^2 + 2Rd}}{2} > \frac{d - R + \sqrt{R^2}}{2} = \frac{d}{2}.$$

把式(6)作 Taylor 展开,可知

$$R - r_{j-1} = R - R\left(1 - \frac{1}{2}\left(\frac{\varepsilon}{R}\right)^2 - \frac{1}{4}\frac{\left(\frac{\varepsilon}{R}\right)^4}{\sqrt{1-\left(\theta\frac{\varepsilon}{R}\right)^2}}\right).$$

其中 $0 < \theta < 1$,故

$$R - r_{j-1} > \frac{R}{2}\left(\frac{\varepsilon}{R}\right)^2 = \frac{\varepsilon^2}{2R} > \frac{d^2}{8R}. \qquad \square$$

定理 2. 该算法的时间复杂性为 $O\left(\left|\lg\left(\frac{d}{8R}\right)\right| n\right)$,其中 n 为点集点的个数.

证明 设算法共迭代 j 次后达到最后结果,即 $r_j = R$,$r_{j-1} < r_j$.由式(5)知 $R - r_{j-1} > \frac{d^2}{8R}$.根据算法,只要再迭代一次便由 r_{j-1}求到了 $r_j = R$.式(4)给出了迭代的收敛速度,当 i 足够大时,使

$$(\rho_0)^i R < \frac{d^2}{8R}, \tag{7}$$

便有

$$|R - r_i| < \frac{d^2}{8R}.$$

因而此时只要再迭代一次,算法便结束了.由式(7)可知,算法总的迭代次数不会超过

$$\left[\left|\lg\frac{d}{8R}\right| / (\lg\rho_0)\right] + 2 = O\left(\left|\lg\frac{d}{8R}\right|\right),$$

其中$[x]$是不超过 x 的最大整数.因为每迭代一次的时间复杂性为 $O(n)$,总的时间复杂性为 $O\left(\left|\lg\frac{d}{8R}\right| n\right)$.

参考文献

1 Samet H. The Design and Analysis of Spatial Data Structure. New York:Addison-Wesley Publishing Company,1989

2 Samet H. Application of Spatial Data Structure. New York:Addison-Wesley Publishing Company,1989

3 O'Rourke J. Computational Geometry in C. New York:Cambridge University Press,1994

4 Preparata F P,Shamos M I. Computational Geometry an Introduction. New York:Springer-Verlag,1985

5 Toussaint G T. Computational Geometry. Amsterdam,New York,Oxford:North-Holland,1985

An Algorithm for Finding the Smallest Circle Containing all Points in a Given Point Set

WANG Wei[1] WANG Wen-ping[2] WANG Jia-ye[3]

[1](*Department of Computer Science Fudan University Shanghai* 200433)

[2](*Department of Computer Science The University of Hong Kong Hong Kong*)

[3](*Department of Computer Science Shandong University Ji'nan* 250100)

Abstract To seek a smallest circle containing all the point of a given point set is an interesting problem in both practice and theory. In this paper, an algorithm of finding a smallest circle containing all the points given is presented. The time complexity of the algorithm is $O(|\lg(d/R)| * n)$, where R is the radius between the smallest circle, d is the smallest distance between the points of the set that are not on the circle and the circle.

Key words Smallest circle, computational geometry.