文章编号 : 1002-0446(2004) 03-0193-05

一种新的基于切线的路径规划方法*

吴峰光 , 奚宏生
(中国科学技术大学自动化系 , 安徽 合肥 230027)

摘 要 : 提出了一种崭新的基于切线的路径规划方法 .它在二维离散姿态空间中 ,用障碍物的边界线建立环境模型 ,用凸壳和切线构造局部最短路径 ,复杂地形能被主干线结构有效地分解 .跳跃式扫描技术和按需扩展搜索图的策略使它优于切线图法 .它能非常有效地利用稀疏环境和处理较大的规划空间 ,并且能适应未知和动态的环境 , 这使它成为远距离漫游的理想导航方法 .仿真表明 ,本文算法通常都能得到全局最优路径 ,并且规划速度快 、内存需求小 ,非常适合于实时应用 .

关键词 : 路径规划 ; 切线图 ; 动态环境 ; 远距离漫游
中图分类号 : TP24 文献标识码 : B

A New Tangent Based Path Planner

WU Feng-guang , XI Hong-sheng
(*Department of Automation , University of Science and Technology of China , Hefei 230027 , China*)

Abstract : This paper presents a novel tangent based path planning method . The planner makes use of the obstacle contours to model the environment in a discrete 2D configuration space . Locally shortest paths are constructed by convex hulls and tangents . Complex terrain is effectively decomposed by the Trunk Structure . It makes a major improvement over the tangent graph approach by expanding only necessary parts of the tangent graph and using the Bouncing Scan technique . It is capable of large space , can take full advantage of sparse environments and deal with unknown and dynamic environments , which make it an ideal planner for long range navigation . Simulation results show that it generates global optimal path in most situations , and is very time and space efficient thus suitable for real time applications .

Keywords : path planning ; tangent graph ; dynamic environment ; long range navigation

1 引言 (Introduction)

本文在切线图法[1 ,2]的基础上 ,提出了一个崭新的基于切线的路径规划算法 ,它有效地解决了切线图法中存在的一些重要问题 .切线图法以多边形障碍物模型为基础 ,任意形状的障碍物只能用近似的多边形来代替 .由于需要预先构造完整的切线图 ,所以必须拥有所在环境的完全知识 ,并且在障碍物或凸边界线段较多的环境下存在组合爆炸的问题 .

本文在离散的二维姿态空间中用障碍物的边界线建立起了精确 、紧凑并且高效的环境模型 .借助于主干线结构对地形进行有效分解 ,使算法能处理任意复杂的地形 .切线图是动态生成的 :在 A* 搜索中进行节点扩展时 ,根据实际需要作相应的切线[3] ,这能有效减小图的规模 .算法能根据传感器的输入 ,动态地调整内部环境模型 ,并进行最好信息规划 ;切线 、凸壳 、节点 、边等很多知识都可以被重复利用 ,因而能有效地应用于未知和动态环境 .

2 路径构造 (Path construction)

2 .1 凸壳和切线

文献[3] 引入了局部最短路径的概念 ,并且证明了对一个平面上的质点机器人 ,任何局部最短路径都是由凸的障碍物边界线段和它们之间的公切线所组成 .因为一条全局最短路径的每一个部分也必然是局部最短路径 ,所以可以用凸边界线段和公切线来构造全局最短路径 .

直接寻找和判定障碍物的凸边界线段代价很大 ,本文引入边界线段的凸壳的概念来有效解决这一问题 .对一条边界线段 C ,如果存在一条折线 H , 满足 : H 是凸的 ; H 的顶点顺序取自 C ; H 不与 C 交叉 ,则称 H 是 C 的凸壳 .凸壳由一系列凸边界线段和公切线组成 ,如果不与任何障碍物碰撞的话 ,它就是一条能绕过对应边界线段的局部最短路径 .

* 收稿日期 :2003 - 07 - 20

设一条线段 AB 与障碍物 O 的边界相交于 A(B)点,从交点出发沿 O 的边界线向两边走,找到第一个不在直线 AB 上的点 C 和 D.如果 C、D 位于直线 AB 的同一边,则称线段 AB 与障碍物 O 相切,点 A(B)称作切点.如果一条线段同时与两个障碍物相切,则称它是两个障碍物的公切线.

在规划过程中,规划器首先作出公切线,然后作出公切线之间的边界线段的凸壳,让凸壳和公切线连接成路径,这样就能有效地构造局部最优路径.我们使用平均时间复杂度为线性对数阶的算法来寻找凸壳和切线,它们能有效地处理普通的路径,同时如果碰到复杂的边界线,则返回,由主干线算法作处理.

2.2 跳跃式扫描

在求凸壳和切线以及许多其它场合中,都可以使用一种简单有效的算法来提高效率,这就是跳跃式扫描技术.

组成边界线的点序列的坐标值是连续有界变化的,即对任意两个相邻边界点 p、q,都有$|x_p - x_q| \leqslant 1$,$|y_p - y_q| \leqslant 1$,$d(p,q) \leqslant \sqrt{2}$.利用这一特点,在边界线上进行的很多查找任务就可以使用较大的步长跳跃式地进行.

运用跳跃式扫描的关键,是对每种特定的任务定义一个步长函数 $step(S)$,它表示从当前边界点 S 开始往目标顺序扫描的时候,至少可以直接往前跳多少个边界点,而不至于错过目标.

如图 1(a)所示,为了查找一条边界线与水平直线的 L 交点,定义 $step(S) = d(S,L)$,并以此步长进行扫描.可以看到从 S_0 开始,经过短短几步搜索就在 S_4 处找到了交点.查找过程的时间复杂度,在最好情况下是 $O(1)$,在典型情况下是 $O(\log n)$,在最坏情况下是 $O(n)$,三种情况分别对应图 1(b)中的 C_1、C_2、C_3.在边界线上查找一给定坐标点 T 时,步长函数可定义为 S、T 两点之间的曼哈顿距离:$step(S) = max(|x_s - x_t|, |y_s - y_t|)$.

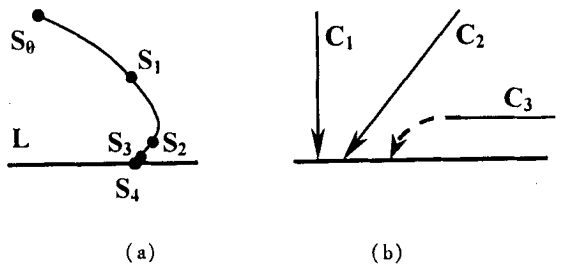

图 1 跳跃式扫描

Fig.1 Bouncing scan

3 路径选择(Path selection)

3.1 主干线结构

为了有效地、有选择地处理复杂的地形,本文提出了基于主干线结构的分解方法.在图 2(a)所示的路径规划问题中,分别位于两个障碍物边界线上的起始点 S 和目标点 T 都被一个双螺旋形障碍物包围.有 3 个障碍物与线段 ST 相交,它们直接挡住了从 S 到 T 的直线路径.为了作出绕过它们的所有可能路径,用直线 ST 对它们进行分割,如图 2(b)所示.称这里的向量 $\overrightarrow{S_T}$ 为主干线,称主干线和被它分割得到的障碍物边界线段为主干线结构.

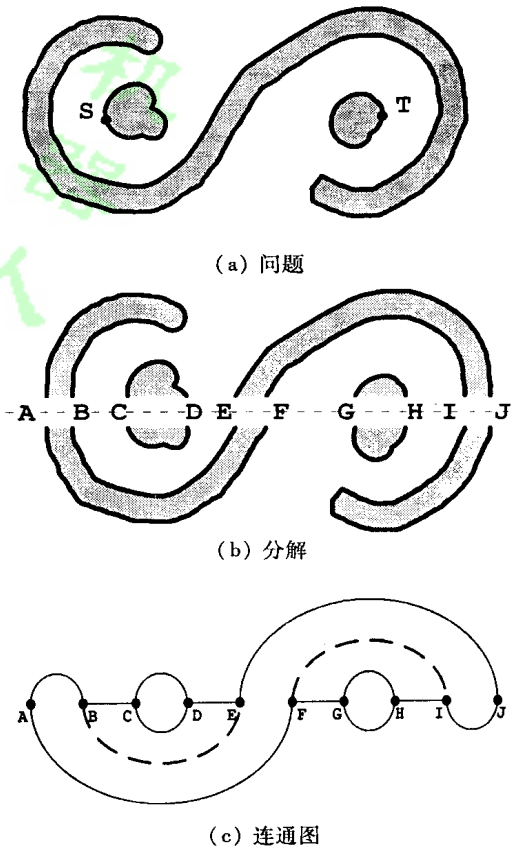

图 2 基于主干线结构的地形分解

Fig.2 Terrain decomposition based on the trunk structure

主干线结构最重要的表示是其连通图.如图 2(c)所示,连通图是一种链式结构.其节点代表主干线与障碍物的交点,边代表主干线结构中的障碍物边界线段,和主干线被障碍物分割而成的部分线段.图中几乎每个节点都有三个邻居节点:所在主干线段另一端的节点,左方障碍物边界线段另一端的节点,以及右方边界线段另一端的节点.

建立主干线结构的主要工作是对障碍物进行切割分段.如果一条边界线被切成 m 段,

每段长 n，则应用了跳跃式扫描后的算法平均时间复杂度是 $O(m\log n)$.

主干线结构实际上对地形作了两级简化:从环境中抽取出与主干线相交的障碍物;将这些障碍物分割成边界线段.这样,借助于主干线结构,就把复杂障碍物之间的公切线求作问题,转化成了主干线结构上的边界线段之间的公切线求作问题.

3.2　图搜索

在切线图法中,算法是分两步进行的:首先在姿态空间的基础上建立一个显式图,求出所有的节点和连接,然后在这个图上进行最优路径的搜索.这种做法的问题是,第一步得到的图需要在理论上是足够大的,以便下一步中任何可能的规划任务都能在图中找到路径.这就导致了一个无法克服的矛盾:第一步中需要生成很大规模的图,但是在下一步的规划中却往往都只用到其中一小部分,剩余的大量无用节点不但浪费存储空间,而且对搜索效率也产生很大的负面影响.

本文算法也是分两步进行,但是第一步仅仅通过地图扫描找到相关的边界线,建立起简单的环境模型.这一步简单而且快速.然后我们进行第二步的路径规划,使用 A^* 算法在一个隐式图上搜索,一边搜索一边新建所需的节点和边.这样,图的建立和使用是同时进行的,只有对当前规划任务有用的节点和边才会被加入图中.

A^* 算法是一种非常有效的搜索方法,本文应用它来寻找全局较优路径.在 A^* 算法中,需要把路径抽象为一组节点序列:路径上的每段障碍物边界线段对应的凸壳及其后继的公切线视为边,将边界线段的起点作为一个节点.规划任务的起始点和目标点分别用一个虚拟的点状障碍物来代表,以便于算法作统一处理.在计算节点 n 的估价函数 $\hat{f}(n)=\hat{g}(n)+\hat{h}(n)$ 的时候,简单的取 $\hat{h}(n)$ 为节点到目标点的直线距离.

在扩展节点的时候,需借助于主干线结构生成必要的切线、凸壳,并以它们为基础构造当前节点的子节点.其基本思路如下:

(1) 在当前扩展节点和目标节点之间建立一主干线结构.

(2) 在干线结构上,作出所有属于当前节点所在障碍物的边界线段到其它边界线段的切线.

(3) 如果得到的切线与某些障碍物相交,则以它为主干线建立新的主干线结构,递归进入(2).

(4) 对所有新得到的无碰撞切线,作凸壳以连接当前节点与新作切线的起点.对于这些凸壳:

a) 如果无碰撞,则以它及对应的切线为基础,创建当前节点的子节点及连接它们的边;

b) 否则以它的第一个碰撞的部分为主干线,递归进入(2).

节点扩展是由中央向两翼波及的过程,只会根据需要向直接或间接挡住去路的障碍物作切线,这在障碍物比较稀疏的地形条件下特别有利.

4　动态环境(Dynamic environments)

在现实世界中,地形信息通常不能事先获得,需要依靠移动机器人自身配备的传感器来对未知环境进行在线探索和路径规划.本文算法具有非常好的动态性,能很好地适应动态环境的要求.

4.1　最好信息规划

为了能处理未知和动态的环境,规划器的工作流程被设计为这样一个循环的过程:底层送入新的位置和环境信息—更新内部环境模型—调整搜索图—重规划—送底层执行.其中第一步和最后一步由仿真软件模块完成.

依靠规划器内部维护的一个环境模型,在动态环境中仍然可以使用全局路径规划算法.在静态环境中,这个环境模型是根据已知的地形图通过一次扫描获得的;在未知环境中,它一开始为空,以后根据传感器不断送入的局部环境信息,不断地增补和修正,以反映当前已知的环境;在动态的环境中,它以已知的地形图为起点,用最新的传感器信息不断地修正之.称这种根据目前已探明的所有环境信息进行的规划为最好信息规划,它已经被实践证明具有良好的表现.

4.2　环境模型的更新

环境模型的局部更新机制是这样的:传感器探测出机器人附近的最新环境信息,以二维概率数组的形式送给环境建模模块.环境建模算法以某一概率值为界把环境分为自由区域和障碍物区域,并进行一遍扫描,获得当前机器人所在自由区域的所有边界线.然后将这些新得到的边界线与被更新区域上的原有边界线进行分析比较.完全位于更新区域内的旧边界线,如果有相同的新边界线与之对应,则表示障碍物没变化,删除对应的新边界线,否则删除旧边界线.部分位于更新区域内的旧边界线,如果其位于更新区域内的部分与某一新边界线一致,则删除此新边界线,原旧边界线保持不变.否则,删除旧

边界线及任何与之交叉/重叠的新边界线,并把旧边界线位于更新区域外的部分与新边界线合成一条新的边界线.最后,所有新边界线被加入环境模型.

4.3 计算结果缓存及更新

由于每次重规划的时候,环境一般只发生较小的变化.所以如果在每次规划的时候,缓存一些有用的计算结果,在以后的规划中就有很大的可能重复利用这些结果.切线、凸壳、节点和边,它们的计算代价大,存储代价小,因而是非常好的缓存对象.在算法实现中维护了一些障碍物对象,对象中保存边界线、凸壳、节点、边等相关属性.

在更新了环境模型之后,必须同步更新建立在它基础上的这些缓存的知识.环境模型的变化可以归结为边界线的新建和删除.如果一条边界线被删除,则删除所有与之相关的节点和边、切线和凸壳,最后删除与之相应的障碍物对象.从被删除的边的父节点出发的有碰撞边需要重新检查自己是否仍是有碰撞的.而当一条新的边界线被加入模型时,需要找到所有与之相交的被缓存的无碰撞切线,这些切线被去掉无碰撞标记,并且其相应的边被标记为有碰撞的,需要扩展.这样就维护了数据的一致性和完整性.

5 性能分析(Performance analysis)

5.1 与环境复杂程度的关系

本文算法大致上可分为环境建模和路径规划两个平行的模块,它们的时空复杂性需要分别讨论.

环境建模的基本任务就是通过扫描得到所有相关的边界线,其时间复杂度为 $O(n)$,其中 n 为机器人所在的自由连通区域的点数.环境模型的空间复杂度是 $O(p)$,其中 p 是边界点的总数.在 $k\times k$ 的空白环境中,$n=k^2$,$p=0$;在一个被大片连续的障碍物区域占满的环境中,机器人只能在一个很小的自由区域内活动,此时 n 和 p 趋于 0;在一个平均分布着大量相互分离的障碍物的环境中,n 不大,但是 p 很大,可达 $O(k^2)$.

规划模块又可分为底层的切线、凸壳、主干线算法和高层的图搜索与扩展算法.底层算法的时间复杂度都是线性对数阶的,并且在运用跳跃式扫描等优化之后能做到更好.而高层算法的有效性基本上取决于规划过程中所生成的图的规模,其复杂度是 $O(n^2)$,其中 n 是一个取决于被扩展到的障碍物数量、边界长度及形状复杂性的一个量.基于切线的算法对环境中障碍物的数目非常敏感.如果环境是障碍物密集型的,那么将会生成大量的切线、节点和边,从而增加算法的时空代价.不过由于我们的算法实现了按需扩展策略,情况要比切线图法好得多.如果最后规划得到的路径代价是 c,则根据 A^* 算法的特性,扩展区域被限定在以起始点 S 和目标点 T 为两个焦点、长半径和短半径之和为 c 的一个椭圆内部(实际限定域还要小).基本上位于此区域内部的切线才会被生成,严格位于其中的才可能被扩展.

5.2 与规划空间大小的关系

在工作空间固定、建模分辨率可变的情况下,环境建模时间和空间复杂度分别可表示为 $O(k^2)$ 和 $O(k)$.对于一个给定的工作空间,如果把环境建模的分辨率提高一倍,则环境建模需耗费 4 倍的时间,环境模型需耗费 2 倍的存储空间.因为模型简单,建模算法简单,所以这种增长的起点很低,对全局影响小.建模时间虽然是平方增长的,但其威胁并不大,因为实际规划中不断重复进行的环境建模任务,一般都局限在传感器有限的探测范围里.

在规划模块中,重要的底层算法都因为有了跳跃式扫描这一有效的优化,其增长率会越来越小.而最为关键的是,切线、凸壳、节点和边的数目基本保持不变,搜索图的规模几乎不受建模精度影响.

本文算法对规划空间的大小不是很敏感,这使算法能够对工作空间作较高精度的离散化和建模,从而得到更高精度的路径.

5.3 与切线图法的比较

由于实现了按需扩展,生成的切线图的规模要比切线图法小得多.图 3 所示,在一个相同的环境下画出了可视图、切线图与本文算法在一次规划中得到的路径和生成的全部切线.本文算法的图规模比前两者小.在复杂的环境中,这一优势将变得更为突出.

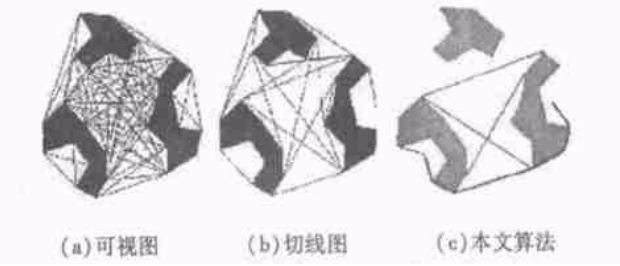

图 3 搜索图的规模对比

Fig.3 Size comparison of search graphs

6 仿真结果(Simulation results)

我们在 Linux 环境下用 C++语言开发了路径规划仿真程序.进行了大量的实验,下面选取其中两个仿真实例来演示算法的有效性.仿真实验是在一

台 CPU 为 P Ⅲ1 G 的服务器上进行的,所用环境地图的大小都是 512 × 512,机器人大小设为 2,规划出的路径以较粗的黑线显示.

如图 4(a)所示的稀疏环境,在相应的姿态空间中有 130 个障碍物,它们的边界点数目是 9138,这样只需约 36 KB 的内存来保存其环境模型.建立姿态空间及环境模型的总时间约为 10 ms,可见环境建模的速度是非常快的.我们进行了五次规划任务,平均规划时间是 12 ms.各次规划中,起始点和目标点的平均直线距离是 539.51,得到的路径平均长 540.28,所得路径的质量非常好.

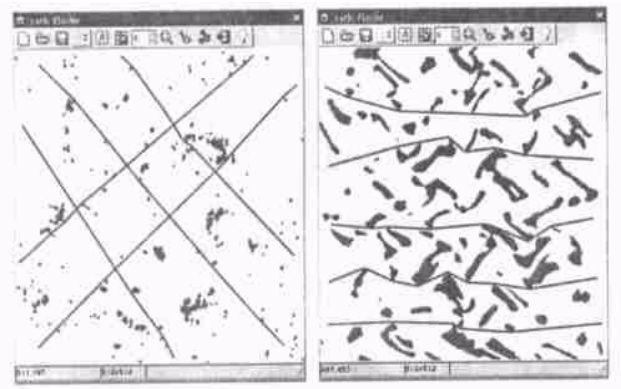

图 4　两种地形中的五次规划

Fig.4　Two terrains (each with five paths planned)

图 4(b)是一个更复杂的地形图.其中有障碍物 79 个,边界点总数为 18436,需要的环境模型存储量约为 73 KB.建立环境模型也只花了 10 ms.在图中以从上到下的顺序进行了五次水平方向的规划,花费的规划时间分别是 60、40、180、50、30 ms,平均 72 ms.环境发生微小变化后,重规划的时间分别为 30、10、80、20、20 ms,平均 32 ms,是首次规划时间的 $\frac{1}{3}$.

各次规划中,起始点和目标点的平均直线距离是 479.05,得到的五条路径平均长度是 491.74.从图上容易判断,它们都是最优路径.这说明按需扩展的策略虽然不能在理论上保证最终路径的最优性,但在实际应用中它一般都能找出最优路径.

规划过程中生成的搜索图是很小的.平均每次规划中生成的节点数是 52.2,边的数目是 172.8,其中无碰撞的边数是 41.2.这么小的图只需不到 100 KB 的存储空间,同时也使搜索时间非常短.

再来看看底层算法的效率.直线逼近法作切线的平均迭代次数是 1.86,也就是说,一般 Q_2 就已经是切点了,可见其收敛速度极快.采用跳跃式扫描后,求切线和凸壳中的平均步长为 5.5,而所有类型的查找任务的加权平均步长更是达到了 18.6.

图 5 显示了程序在未知环境中进行的路径规划.灰色的区域是环境中的所有障碍物,其中有黑色边界线的是在规划过程中被传感器模块完整感知的一些障碍物.黑色的曲线是机器人走过的轨迹,因为地形信息是被逐步得到的,所以比已知地图的情况要差一些.

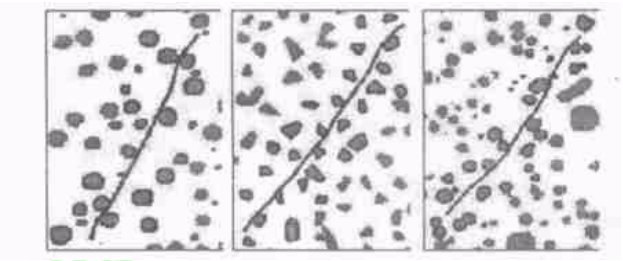

图 5　未知环境中的规划结果

Fig.5　Planning results in unknown environments

7　结论(Conclusion)

本文提出了一种新的基于切线的路径规划方法.它有以下优点:能够处理任意形状的障碍物和复杂的地形条件;执行速度快,内存需求小;能非常有效地利用稀疏的地形;能胜任比较大的工作空间和进行长距离漫游导航;自然的面向对象模型和方法使得一些策略容易应用.在今后的研究中,我们将进一步提升算法在未知和动态环境中重规划的能力,以及处理障碍物密集型环境的效率.

参考文献　(References)

[1] Liu Y H, Arimoto S. Proposal of tangent graph and extended tangent graph for path planning of mobile robots[A]. Proceedings of the 1991 IEEE International Conference on Robotics and Automation[C]. Sacramento, California: IEEE Computer Society Press, 1991. 312 - 317.

[2] Liu Y H, Arimoto S. Computation of the tangent graph of polygonal obstacles by moving-line processing[J]. IEEE Transactions on Robotics and Automation, 1994, 10(6): 823 - 830.

[3] Doyle A B, Jones D I. A tangent based method for robot path planning[A]. Proceedings of the 1994 IEEE International Conference on Robotics and Automation[C]. San Diego: IEEE Computer Society Press, 1994. 1561 - 1566.

作者简介:

吴峰光 (1977-),男,硕士生.研究领域:机器人路径规划.

奚宏生 (1950-),男,教授,博士生导师.研究领域:鲁棒控制,离散事件动态系统及其应用等.