

# 一种基于可视图法的机器人全局路径规划算法

杨淮清, 肖兴贵, 姚 栋

(沈阳工业大学 信息科学与工程学院, 沈阳 110178)

**摘 要:** 针对全局路径规划方法中基于自由空间的路径规划方法在环境发生变化时适应性不强、实时性较差和需要重新建立连通模型等问题, 提出了一种基于可视图法的移动机器人路径规划算法, 该算法较好地弥补了自由空间法存在的缺陷. 在实际建模期间, 对于那些轮廓复杂的障碍物, 可把它近似地看作矩形或多个矩形的组合体, 以此来建立所描述障碍物的边界地图, 并根据所得地图实现机器人的路径规划. 仿真结果表明了该算法的有效性.

**关 键 词:** 机器人学; 环境建模; 可视图; 自由空间法; 全局路径规划

中图分类号: TP 24

文献标志码: A

## A V-graph based global path planning algorithm for mobile robot

YANG Huai-qing, XIAO Xing-gui, YAO Dong

(School of Information Science and Engineering, Shenyang University of Technology, Shenyang 110178, China)

**Abstract:** A V-graph based global path planning algorithm for mobile robot was proposed for such problems that the adaptability and real-time ability of the free space based path planning method are not good enough to change with environment and the connectivity model needs to be reestablished. The proposed algorithm can well remedy the shortages of the free space based path planning method. The obstacles with complex contour can be regarded as a combination of rectangles to establish the obstacle boundary map. The path planning algorithm for mobile robot can be realized according to the obtained map. The simulation results prove the effectiveness of the proposed algorithm.

**Key words:** robotics; environment modelling; V-graph; free Space method; global path planning

路径规划是移动机器人系统中最重要的一部分之一, 其方法可划分为两大类: 一种是基于环境完全信息的全局路径规划<sup>[1-2]</sup>, 另一种则是基于传感器检测的局部路径规划<sup>[3-4]</sup>. 全局路径规划根据环境模型找出从起始点到目标点符合一定性能要求的最优或可行路径, 涉及的基本问题便是环境模型描述和搜索策略. 按照环境模型表示方式差异, 可以将全局路径规划区分成多种, 富有代表性的方法为障碍空间法<sup>[5-6]</sup>和自由空间法<sup>[7]</sup>. 自由空间法的基本思想是采用预先定义的基本形状构造自由空间, 并将自由空间表示为连通图, 再通过对图的搜索来实现行走路径规划. 该方法比较灵活, 起始点与目标点改变不会引起连

通图重构, 算法复杂度与障碍物的数量成正比. 该方法的主要缺点是对环境模型要求比较苛刻, 且在环境发生变化时连通模型需要重新构建, 因而算法对环境的适应性不甚理想. 此外, 利用该方法建立起来的系统, 在实时性方面也表现得不尽人意. 鉴于此, 本文提出了一种基于可视图法<sup>[8]</sup>的移动机器人路径规划算法. 可视图法可归结为障碍空间法中的一种变异. 该方法将所有障碍物顶点和机器人起始点及目标点用线段相连, 如果这些线段不与障碍物相交则认为线段是“可视的”, 再据此搜索从起始点到目标点的最优路径. 文中算法实现的主要技术路线为: 先把那些轮廓复杂的障碍物近似地看作矩形或多个矩形的组合, 由

收稿日期: 2007-09-17.

基金项目: 辽宁省教育厅基金资助项目(2008500).

作者简介: 杨淮清(1956-), 男, 陕西延安人, 副教授, 主要从事人工智能、机器人环境建模、路径规划和特种机器人等方面的研究.

此建立起描述障碍物的边界地图,再根据所得地图实现机器人路径规划。

## 1 环境地图构造

最短路径是一条连接起始点和目标点的折线<sup>[9]</sup>,除了起始点和目标点之外,各段线段的端点都是障碍物的顶点,因此所求最短路径仅和障碍物的边界点有关。考虑到移动机器人的安全性,要对各障碍物矩形化后的区域进行膨胀处理。在膨胀处理后的环境模型上,很容易获得路径规划区域近旁各障碍物的轮廓点。根据每个轮廓点便可构造出对应的节点空间布局,膨胀处理确保了这些节点均不会位居任何障碍区域之内。所有节点被保存在一个 Node 类型的结构中,用于后续的环境模型构建。Node 类型的简化定义如下:

```
struct Node {
    point p;
    struct Line * lines;
};
```

其中 Line 类型定义为:

```
struct Line {
    point From;
    point Throu;
    point To;
    float Dist;
};
```

Node 的属性 lines 用来存放某点到另一点所经由的路径信息。其中,点 lines[k].From 和点 lines[k].Throu 总是可视的,可以依据位置关系将其区分为“外可视点”和“内可视点”。通过环境空间里的任意位置向障碍物引两条切线,所得切点便是外可视点。一个位置同其内部的障碍物最多只有两个外可视点,若针对该障碍还存在其他可视轮廓点则属于内可视点。图 1 中,  $N_6$ 、 $N_7$  是  $N_3$  的外可视点,  $N_5$  是  $N_3$  的内可视点。内可视点可视为无效可视点,因为它们对于机器人路径搜索不具价值,反而增加计算负担。

可视点间的连线可称为“可视线”。删除那些不必要的可视线,如连接无遮挡的外可视点的可视线,连接内可视点的可视线。当障碍物边线全部由直线构成时,可视点一定经过其顶点。本文只考虑了一种简化形式下的无效可视点判定,即障碍物由单一矩形代替的情况。此时,空间中任何位置相对于一个障碍物而言,最多只能有 3 条可视线,其中 2 条是外可视线,1 条是内可视线。

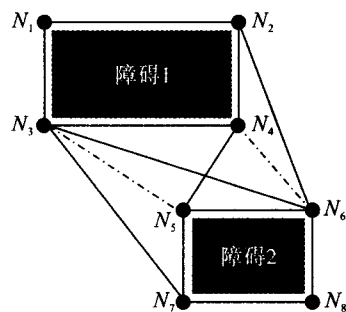


图1 各节点间的可视线

Fig. 1 Visible lines between nodes

无效可视线可用如下方法判断:当一节点与某障碍物有 3 条可视线时,若 1 条可视线与其他 2 条可视线所构成的转向角乘积小于等于零,则该可视线即视为无效连线,应从可视图中删去(最多删除一次)。图 1 中的实线为有效可视线,虚线为无效可视线,有效可视线即是外可视线。

对于复杂障碍物,可将其近似看成多个矩形的组合,如图 2 所示。算法实现时仍可将其视为两个独立的矩形障碍物,只是删除了两个不存在节点。多个矩形的组合可依此类推。

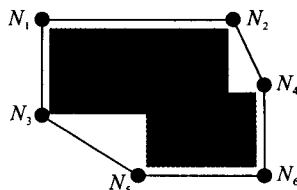


图2 复杂障碍物及其可视线

Fig. 2 Complex obstacle and its visible lines

## 2 算法描述与实现

为了克服基于自由空间的路径规划方法对环境适应性不强、实时性较差等缺点,在判定地图上两点是否可行时,本文引入了最大最小化原则<sup>[10]</sup>,使得路径生成规则简单,生成速度快且适应性更强。

### 2.1 期望路线段与障碍物的相对位置关系

在路径规划过程中,需要判断期望线段是否和障碍物相交,图 3 所示为由端点 S 和 G 确定的线段与障碍物  $O_1$  及  $O_2$  的相对位置关系。

检验期望线段与障碍物的相交性就是判定期望行走路线与障碍物是否存在冲突。如果障碍物位于期望行走路线之上,或者离得过近,机器人就得绕行。可以采用如下两种途径进行判别:

1) 求取路线段与障碍物边线交点并判定交点是否位于两者端点之内

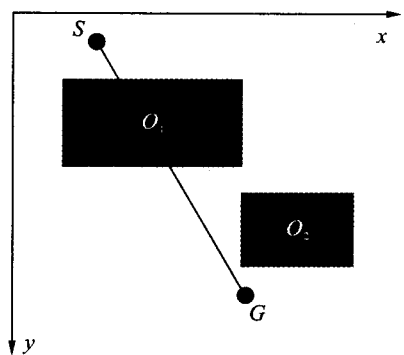


图3 线段SG与O<sub>1</sub>相交  
Fig.3 Cross of line SG and obstacle O<sub>1</sub>

求出由S、G所确定直线与某一障碍物(如O<sub>1</sub>)的4条边的交点(当该直线与障碍物的某些边线平行时不存在交点),如果这些交点既位于线段SG之内,又处于障碍物(如O<sub>1</sub>)边缘之上,则在S、G点之间一定存在障碍物(如O<sub>1</sub>).判定空间中一点(x,y)是否在线段和障碍物上的表达式均为

$$x \geq \min(x_i) \wedge x \leq \max(x_i) \wedge y \geq \min(y_i) \wedge y \leq \max(y_i) \quad (1)$$

式中,(x<sub>i</sub>,y<sub>i</sub>)为某一线段或障碍物的任一顶点.若该表达式全都为真,则线段SG与障碍物O<sub>1</sub>相交.

2) 确定路线段与障碍物边线是否彼此位于所构成线段异侧

若一条期望路线穿越障碍物,则至少从构成该障碍物的边线穿越两次.由于期望路线或障碍物边线都可以用直线方程刻画

$$Ax + By + C = 0 \quad (2)$$

而点(x,y)到该直线的距离(带符号)可由式(3)给出

$$d = (Ax + By + C) / \sqrt{A^2 + B^2} \quad (3)$$

若点位于有向线段左侧,d大于零;若点位于有向线段右侧,则d小于零.利用上述关系,若某障碍物边线端点A、B到期望路线S、G所构线段的距离分别为d<sub>a</sub>和d<sub>b</sub>,S、G到A、B所构线段的距离分别为d<sub>s</sub>和d<sub>g</sub>,当它们同时满足关系式d<sub>a</sub>·d<sub>b</sub><0和d<sub>s</sub>·d<sub>g</sub><0时,障碍物的该边线将与期望行走路线发生冲突.

第一种判别方法简单明了、易于理解,但需要解二元一次方程组并进行数值大小的逻辑判定.第二种方法需要计算4个距离值,至少需要2次开平方运算,但逻辑清晰且数据适应性容易得到保证,实验系统采用了此方法.

## 2.2 算法实现中主要类型描述

算法实现中需要对数据类型进行描述,主要

包括:

### 1) 节点映射类 Map

Map类型的对象存储所有障碍物及其所有有效轮廓点对应的节点,主要属性如下:

```
struct Obstacle {
    int x,y,Height,Width;
};
class Map {
    public: struct Node * pNodes;
           struct Obstacle * pObstacles;
};
```

### 2) 路径类 Path

Path类型的对象存储一个地图对象、两个Node类型的指针和初始点S及目标点G的坐标,主要属性如下:

```
class Path {
    public:
        point S;
        point G;
        struct Line * pVisible S;
        struct Line * pVisible G;
        Map mMyMap;
};
```

其中,pVisible S存储起始点S可视的所有有效节点lines中的所有元素;pVisible G存储目标点G可视的所有有效节点lines中的所有元素.如图4所示,S的有效可视节点为N<sub>2</sub>和N<sub>3</sub>.

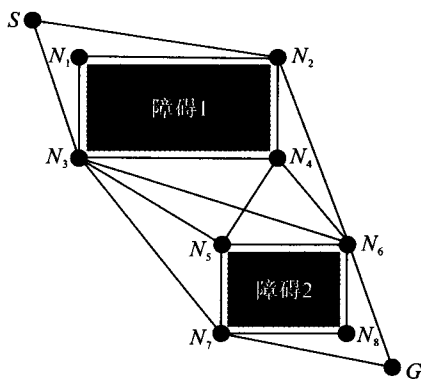


图4 与S及G外可视的节点  
Fig.4 Effective visible nodes of S and G

## 2.3 算法描述与实现

根据环境地图的构造和障碍物与线段的位置关系分析,可给出如下算法:

输入:用户定义的障碍物、初始点S及目标点G.

输出:存放所求路径上的点列向量Route.

注释:向量中的插入采用push\_back方法.

### 1) 数据准备

利用障碍物创建所有有效节点及各节点的 lines 信息,进而实例化 Map 对象 mMyMap;

用 mMyMap、初始点  $S$  及目标点  $G$  实例化一个 Path 对象 mMyPath;

定义一个  $\text{vector} < \text{point} >$  类型的向量容器 Route 存储最终所得到的路径点,将  $S$  插入到 Route 容器中;

定义一临时变量 PassPoint,令其等于  $S$ .

2) 判断  $G$  是否是 PassPoint 的可视点,若是,则转到 6),否则转到 3).

3) 对所有的 pVisible  $S$ 、pVisible  $G$  中的元素做循环

若 pVisible  $S[i].\text{To} = \text{pVisible } G[j].\text{From}$ , 且  $S$  到点 pVisible  $S[i].\text{From}$  的距离、 $G$  到点 pVisible  $G[j].\text{From}$  的距离与 pVisible  $S[i].\text{Dist}$  之和最小时,判定 pVisible  $G[j].\text{From}$  是否为  $S$  的可视点.若是,将 pVisible  $G[j].\text{From}$  插入 Route,并令 PassPoint = pVisible  $G[j].\text{From}$ ,转到 6),否则,将 pVisible  $S[i].\text{From}$  插入 Route,令 PassPoint = pVisible  $S[i].\text{From}$ ,转到 3).

4) 判定 pVisible  $G[j].\text{From}$  是否为 PassPoint 的可视点,若是,将 pVisible  $G[j].\text{From}$  插入 Route,令 PassPoint = pVisible  $G[j].\text{From}$ ,并转到 6),否则,转到 5).

5) 对所有的节点循环

若 pNodes  $[i].\text{p} = \text{PassPoint}$ ,对 pNodes  $[i].\text{lines}$  的元素循环,当 pNodes  $[i].\text{lines}[j].\text{To} = \text{pVisible } G[j].\text{From}$  时,将 pNodes  $[i].\text{lines}[j].\text{Throu}$  插入 Route,令 PassPoint = pNodes  $[i].\text{lines}[j].\text{Throu}$ ,转到 4).

6) 判断  $G$  是否为 PassPoint 的可视点,若是,将  $G$  插入 Route.

7) 算法结束.

## 3 仿真结果

利用上述算法构造了仿真实验.仿真语言环境为 C++ Builder,运行于 PC 机.图 5、6 给出的仿真结果显示,环境地图是按要求临时构造的,算法很好地适应了环境.在机器人行走过程中,可以随时改变目标点  $G$ , $G$  的坐标一旦改变,系统将立即重新规划出一条由机器人当前所在的位置到新目标位置的最优路径或可接受路径,并继续行走,直至到达目标点,一定程度上满足了实时性的要求.

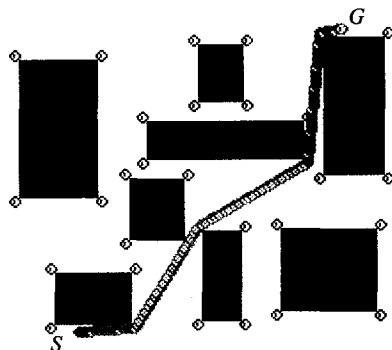


图5 仿真结果一

Fig.5 First simulation result

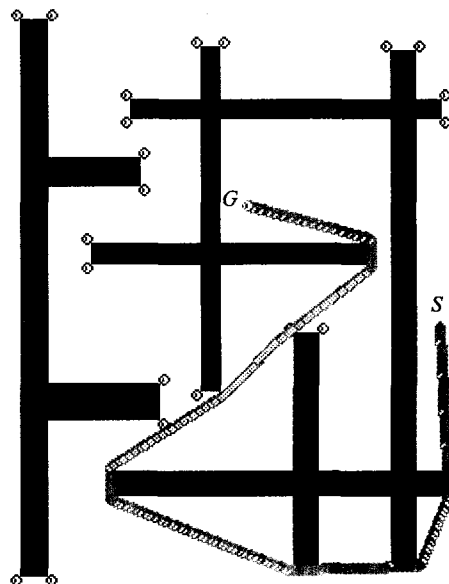


图6 仿真结果二

Fig.6 Second simulation result

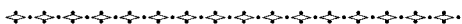
## 4 结论

本文提出的机器人全局路径规划算法直接在有关环境地图上进行,避免了基于自由空间的路径规划方法要求建立网络连通模型和由于搜索而造成的适应性不强、实时性较差等不足.应该指出的是,由于算法仅是对一个矩形描述的障碍物建立的,因此其规划结果并不能保证最优性.不过,实验表明,该算法获得的路径非常接近最优值.

### 参考文献 (References):

- [1] Feder H J S, Slotin J J E. Real-time path planning using harmonic potentials in dynamic environments [C]// Proceedings of the IEEE International Conference on Robotics and Automation. Albuquerque, NM, 1997:1408-1422.
- [2] Latombe J C. Robot motion planning [M]. Kluwer Academic Publishers, 1991.
- [3] Yang X D, Yamamoto M, Mohri A. Path planning for mobile robot in uncertain workspace using obstacle to-

- [4] Nagatani K, Yuta S. Path and sensing point planning for mobile robot navigation to minimize the risk of collision [J]. Journal of the Robotics Society of Japan, 1995, 13(8): 1130-1137.
  - [5] 蔡自兴, 彭志红. 一种新的路径编码机制在移动机器人路径规划中的应用 [J]. 机器人, 2001, 23(3): 230-233.  
(CAI Zi-xing, PENG Zhi-hong. The application of a novel path encoding mechanism in path planning for a mobile robot [J]. Robot, 2001, 23(3): 230-233.)
  - [6] Jayalakshmi A G, Prabhu H, Rajaram R. An adaptive mobile robot path planner for dynamic environments with arbitrary-shaped obstacles [J]. International Journal of Computational Engineering Science, 2003, 4(1): 67-84.
  - [7] Robert A, Moshe K. Robot motion planning on N-dimensional star worlds among moving obstacles [J]. IEEE Transactions on Robotics and Automation, 1998, 14(2): 320-325.
  - [8] Lozano-Perez T, Wesley M. An algorithm for planning collision-free paths among polyhedral obstacles [J]. Communications of the ACM, 1979, 22(5): 436-450.
  - [9] 宋国宁, 刘述忠, 刘金德, 等. 寻找最佳路径: 搜索、剪枝与学习 [C]// 机器人学论文集. 沈阳: 自动化研究所, 1991: 235-239.  
(SONG Guo-ning, LIU Shu-zhong, LIU Jin-de, et al. Find best path [C]// Robotics Collection. Shenyang: Institute of Automation, 1991: 235-239.)
  - [10] 赵晖, 王天然. 可分解无回溯移动机器人路径规划算法 [C]// 中国科学院机器人学开放研究实验室年报: 下册. 沈阳: 自动化研究所, 1993: 528-534.  
(ZHAO Hui, WANG Tian-ran. Decomposable no retrospective path planning algorithm for mobile robot [C]// Journal of Chinese Academy of Sciences Laboratory of Robotics: Next List. Shenyang: Institute of Automation, 1993: 528-534.)
- (责任编辑: 邓美艳 英文审校: 王丽梅)



(上接第 224 页)

- [2] 魏明,曹正清,龚家伟.汽车驾驶训练模拟器视景系统的设计与实现[J].中国农业大学学报,2003,8(4):55-57.  
(WEI Ming, CAO Zheng-qing, GONG Jia-wei. Designing and implementing of the visual system of automobile driving training simulator [J]. Journal of China Agricultural University, 2003, 8(4): 55-57.)
  - [3] 魏明.汽车驾驶训练模拟器视景系统软件的研究与开发[D].北京:中国农业大学,2001.  
(WEI Ming. Study and development on the software of visual system of vehicle driving training simulator [D]. Beijing: China Agricultural University, 2001.)
  - [4] 马骏,朱衡君.利用P-Buffer模拟虚拟场景的后视镜和鸟瞰图[J].计算机辅助设计与图形学学报,2006,18(5):748-752.  
(MA Jun, ZHU Heng-jun. Simulation of rearview mirror and aircscape in virtual environment with P-Buffer [J]. Journal of Computer-Aided Design & Computer Graphics, 2006, 18(5): 748-752.)
  - [5] Chris W. Using P-Buffers for off-screen rendering in OpenGL [EB/OL]. [2005-04-19]. <http://www.oss.>
  - [6] 霍金 K,阿斯特 D. OpenGL 游戏程序设计 [M]. 北京:科学出版社,2006.  
(Hawkins K, Astle D. OpenGL game programming [M]. Beijing: Science Press, 2006.)
  - [7] 段小娟,张茂军,李国辉,等.虚拟实景空间中的纹理映射技术研究[J].计算机工程,2001,27(5):17-18.  
(DUAN Xiao-juan, ZHANG Mao-jun, LI Guo-hui, et al. Research of texture mapping in image-based virtual space [J]. Computer Engineering, 2001, 27(5): 17-18.)
  - [8] Wright R S, Lipchak Jr B. OpenGL 超级宝典 [M]. 3版.北京:人民邮电出版社,2005.  
(Wright R S, Lipchak Jr B. OpenGL super dictionary [M]. 3th ed. Beijing: Post & Telecom Press, 2005.)
  - [9] 关沫,黄有群. OpenGL 实用编程技巧两则 [J]. 沈阳工业大学学报,2000,22(4):321-324.  
(GUAN Mo, HUANG You-qun. Two practical skills for OpenGL programming [J]. Journal of Shenyang University of Technology, 2000, 22(4): 321-324.)
- (责任编辑:邓美艳 英文审校:王丽梅)

(责任编辑:邓美艳 英文审校:王丽梅)

[sgi.com/projects/Ogl-sample/registry](http://sgi.com/projects/Ogl-sample/registry).

- [6] 霍金 K, 阿斯特 D. OpenGL 游戏程序设计 [M]. 北京: 科学出版社, 2006.  
(Hawkins K, Astle D. OpenGL game programming [M]. Beijing: Science Press, 2006.)
- [7] 段小娟, 张茂军, 李国辉, 等. 虚拟实景空间中的纹理映射技术研究 [J]. 计算机工程, 2001, 27(5): 17-18.  
(DUAN Xiao-juan, ZHANG Mao-jun, LI Guo-hui, et al. Research of texture mapping in image-based virtual space [J]. Computer Engineering, 2001, 27(5): 17-18.)
- [8] Wright R S, Lipchak Jr B. OpenGL 超级宝典 [M]. 3 版. 北京: 人民邮电出版社, 2005.  
(Wright R S, Lipchak Jr B. OpenGL super dictionary [M]. 3th ed. Beijing: Post & Telecom Press, 2005.)
- [9] 关沫, 黄有群. OpenGL 实用编程技巧两则 [J]. 沈阳工业大学学报, 2000, 22(4): 321-324.  
(GUAN Mo, HUANG You-qun. Two practical skills for OpenGL programming [J]. Journal of Shenyang University of Technology, 2000, 22(4): 321-324.)

(责任编辑:邓美艳 英文审校:王丽梅)