

文章编号: 1001-0920(2011)11-1695-04

基于几何相交测试的机器人路径规划算法

周之平, 黎明, 华路

(南昌航空大学 信息工程学院, 南昌 330063)

摘要: 针对机器人路径规划问题, 提出一种基于几何相交测试的路径规划方法. 该方法首先搜索位于当前路径点到目标点连线上的首障碍栅格; 然后结合贪婪法、回溯法和邻域搜索策略从障碍栅格邻域搜索下一个路径栅格; 接着从新的路径点出发迭代搜索后续的路径点, 从而确定从起点到目标点的路径; 最后对得到的最好路径进行路径点合并以提高路线的连贯性. 实验结果表明, 新方法规划的路径性能优于其他同类算法, 路径呈现出更好的连续性, 规划时间能满足实际应用的要求.

关键词: 栅格; 相交测试; 连贯性; 路径规划

中图分类号: TP391

文献标识码: A

Geometry intersection testing based robot path planning algorithm

ZHOU Zhi-ping, LI Ming, HUA Lu

(School of Information Engineering, Nanchang Hangkong University, Nanchang 330063, China. Correspondent: ZHOU Zhi-ping, E-mail: zhouzhp.2008@yahoo.com.cn)

Abstract: For the problem of robot path planning, a geometry intersection testing based path planning algorithm is presented. Firstly, the first obstacle grid is searched in the configuration space which lies on the oriented segment from start to target. By combining the greedy method and backward strategy, the grid where next way point lies is determined from the neighbors of the first obstacle grid by local searching. Then subsequent way points are obtained iteratively scratching from the last obtained way point in order to get some routes from start to target as possible. Finally, the best route is refined by incorporating sub-routes to maintain the consistency of trajectory. The experimental results show that, the presented method can plan a shorter and more reasonable path than other algorithms, which represents higher consistency of trajectory, and the computing time can meet the requirement of practical application.

Key words: grid; intersection testing; consistency; path planning

1 引言

路径规划问题是机器人、飞行航机规划、大规模集成电路等应用领域的一个研究热点, 采用的主要方法有空间规划方法和智能搜索算法. 空间规划方法包括可视图法、Delauny 三角化方法和 Voronoi 图法; 智能搜索算法主要有 A* 算法^[1]、蚁群算法^[2-3]、人工势场法^[4]和遗传算法^[5-6]. 空间规划方法多用于由多边形集合构成的障碍环境, 对于由曲线边界构成的封闭区域实现难度较高. A* 算法通过比较当前路径栅格的 8 个邻居的启发式函数值 F 来逐步确定下一个路径栅格, 当存在多个最小值时, A* 算法不能保证搜索的路径最优. 蚁群算法模拟蚂蚁觅食过程, 利用信息素更新方法逐段强化较好的子路径, 从而获得一条可

行路径. 蚁群算法受起止点位置和障碍分布的影响, 环境复杂时蚂蚁容易陷入不可行点, 甚至出现路径迂回和死锁. 文献 [7] 将遗传算法的快速收敛性与蚁群算法的信息正反馈机制有效结合, 有效地避免了蚂蚁陷入死胡同. 人工势场法将起点、障碍物和终点视为带电粒子, 起点在障碍物的斥力与目标点的引力所形成的合力的作用下运动, 从而产生路径, 当运动目标移动到障碍物附近或窄带中时, 该方法易出现抖动现象. [8] 提出了一种人工势能场与自适应遗传算法相结合的路径规划方法, 通过引入引力场动态调整规则, 避免陷入局部极小区域. [9] 通过引入虚拟障碍、安全距离约束和威胁角约束, 抑制了抖动. 遗传算法求解路径规划问题时采用变长编码, 并通过引入插入算

收稿日期: 2010-06-01; 修回日期: 2010-11-23.

基金项目: 国家自然科学基金项目(60963002); 江西省自然科学基金项目(2009GZS0090).

作者简介: 周之平(1976—), 男, 副教授, 博士, 从事路径规划与进化计算等研究; 黎明(1965—), 男, 教授, 博士生导师, 从事进化计算理论与图像处理等研究.

子和删除算子消除环路, 保证路径点的空间连续性. [10] 限定机器人沿水平垂直方向移动, 简化可行路径的生成过程, 但所规划的路径未必是最优解. [11] 提出了一种改进模拟退火算法 (MSAA) 与共轭方向法相结合的路径规划算法, 以避免算法早熟收敛, 但算法的性能受初始温度和冷却表设定的影响.

针对上述算法存在的缺陷, 本文针对栅格表示的规划空间, 基于贪婪法和回溯法的思想, 结合几何求交方法和局部优化搜索, 提出一种基于几何相交测试的机器人路径规划算法, 以提高路径搜索的效率.

2 问题描述

2.1 路径规划问题

路径规划问题是指在给定有障碍物的规划空间中, 搜索从起点到目标点的无障碍且最短路径. 为了便于计算, 通常采用栅格化模型简化表示规划空间, 即将环境空间分割为大小相等的栅格, 以自由栅格的中心作为路径点规划出一条最优路径.

2.2 栅格的数据结构

栅格数据结构定义如下:

```
typedef struct Grid {
    int rowIndex; // 行索引
    int colIndex; // 列索引
    enum{ OBSTACLE, START, END,
        USED, UNUSED } property; // 栅格状态
};
```

3 基于几何相交测试的路径规划算法

与其他方法不同的是, 本文通过计算与有向线段首相交的不可用栅格, 从该栅格邻域内选择空闲栅格作为下一个路径栅格来逐步规划出合理路径. 通过路径点邻域变异和回退操作避免出现搜索停滞和陷入无限循环, 并采用贪婪法策略对子路径合并以增强路径的连贯性, 从而进一步优化路径.

3.1 确定有向边 L 与环境空间首相交的障碍栅格

设 $g_{start}(i_1, i_2), g_{end}(j_1, j_2)$ 分别为 L 的起点 p_{start} 和目标点 p_{end} 所在的栅格, 根据 L 的走向分为 8 种情形, 如图 1 所示. ①~④ 表示 L 与坐标轴平行, ⑤~⑧ 表示 L 走向位于第 1 象限~第 4 象限. 对于 ①~④, 计算过程比较简单, 只需逐个测试 L 所跨越行或列上的栅格状态障碍是否为障碍物或者在当前路径上即可. 对于 ⑤~⑧, 此时路径可能跨越环境空间中的多行或多列, 计算首相交栅格过程更加复杂. 以 ⑤ 为例, 依次从 $i_2 + 1$ 到 j_2 列栅格作中心垂线 L_{vert} . 先求解 L_{vert} 与 L 的交点 P , 根据 P 所在位置补充 L 所跨越的中间栅格; 然后按照各栅格出现的先后次序逐个测试其是否为障碍物或者在当前路径上. 若是, 则返

回首相交栅格的索引; 否则, 在下一列栅格中心作垂线, 直至找到相交栅格或者 j_2 列测试完为止. 运行结果如图 2 所示, 图 2 中: S 为起点, T 为目标点, 实心圆圈为首相交栅格所在的位置 (下同). 对于 ⑥~⑧, 方法类同, 这里不再赘述.

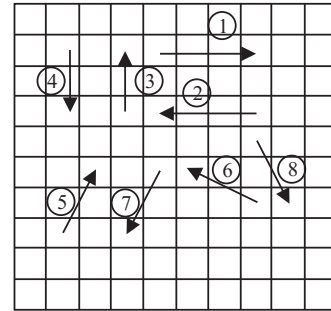


图 1 8 种可能的路径走向

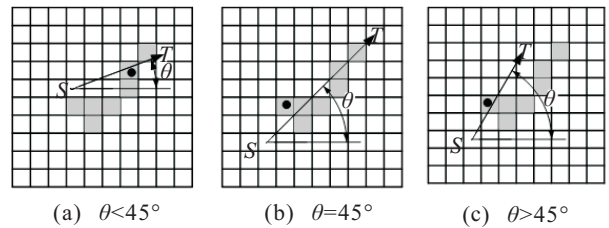


图 2 情形 ⑤ 时与路径相交栅格示意图

3.2 路径点插入操作

找到起止点连线与规划空间 E 首相交的障碍栅格 $g_{FirstInter}$ 后, 若 $g_{FirstInter}$ 邻域中存在可直达当前路径栅格 $g_{Current}$ 的自由栅格 (“UNUSED”), 则可确定下一个路径栅格. 根据 $g_{Current}$ 与 $g_{FirstInter}$ 的不同位置关系, 分 3 种情形进行讨论, 如图 3 所示.

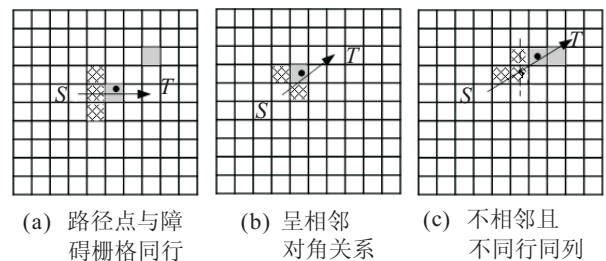


图 3 根据首相交栅格确定候选下一个路径点

1) $g_{Current}$ 与 $g_{FirstInter}$ 位于同一行或列. 选择障碍栅格邻域中靠近当前路径点一侧且距离 $g_{Current}$ 最近的自由栅格中心作为下一个路径点, 如图 3(a) 所示.

2) $g_{Current}$ 与 $g_{FirstInter}$ 呈相邻对角关系. 此时至多存在 2 个可用邻居栅格, 若为自由栅格, 则选取其中心作为下一个路径点, 如图 3(b) 所示.

3) $g_{Current}$ 与 $g_{FirstInter}$ 不相邻且不在同一行或列. 以 L 走向位于第 1 象限为例, 首先在 $g_{FirstInter}$ 前一列栅格的中心作垂线 L_1 , 求解 L_1 与直线 L 的交点 P_1 , 并根据 P_1 的位置确定下一个路径点. 若 P_1 不在栅格的边界上, 则直接取交点 P_1 作为下一个路径点; 否则,

将 P_1 定位到边界正下方的栅格, 选取该栅格可用邻居栅格中到 $g_{Current}$ 最近的栅格中心作为下一个路径点, 如图 3(c) 所示.

对于其他位置关系情形, 当障碍栅格邻域内存在到当前路径点的可用栅格时, 下一个路径点的确定方法类似. 一旦确定了一个路径点, 则可用其作为新的起点, 进而规划出下一个路径点. 当存在多个最近候选邻居时, 则将其分别加入当前测试的局部路径中, 进而产生多条路径.

3.3 路径点回退与邻域变异操作

若 $g_{FirstInter}$ 邻近区域内不存在可达栅格 $g_{Current}$ 的自由栅格, 则第 3.2 节的方法无效, 必须采用一定的策略找到下一个路径点. 根据 $g_{FirstInter}$ 在环境空间中所处位置的不同可分为 2 种情形: 1) $g_{Current}$ 邻域中存在可用栅格; 2) $g_{Current}$ 邻域中不存在可用栅格. 图 4 为路径点删除和邻域变异操作. 图 4 中: 灰色圆点表示已经得到的路径点, 黑色圆点表示障碍栅格所在位置, “◇”表示通过邻域变异操作得到的下一个路径栅格, “×”表示已回退的路径栅格, “▲”表示回退当前路径点后通过进一步搜索得到的下一个路径栅格.

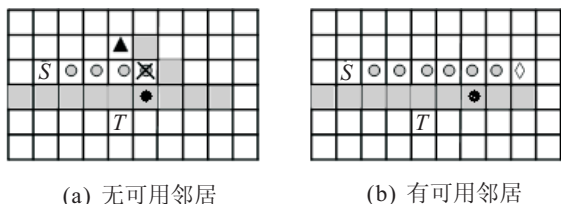


图 4 路径点删除与邻域变异操作

从图 4 可以看出, 此时存在从起点到目标点的路径. 但是, 因为障碍栅格靠近 $g_{Current}$ 的一侧无可用邻居, 所以从当前路径点不能到达目标点. 为使得算法能有效地找到一个出口, 当 $g_{Current}$ 的邻域内无可用栅格时, 执行路径点回退操作, 同时将已回退路径点所在栅格的属性标志为“OBSTACLE”, 以避免无限循环, 如图 4(a) 所示. 当 $g_{Current}$ 的邻域内存在可用栅格时, 执行邻域变异操作, 即在水平和垂直方向邻居中选择可用栅格的中心作为下一个路径点, 如图 4(b) 所示. 若存在多个可用栅格, 则在当前测试的局部路径基础上产生多条路径, 以避免遗漏最好解. 路径点回退和邻域变异操作可有效地避免搜索过程陷入死循环.

3.4 子路径的合并过程

按照第 3.2 节和第 3.3 节的方法进行路径规划时, 当起点与目标点距离较远且障碍物密度较高或者起点与目标点被障碍物阻断时 (I 型和 H 型障碍环境), 规划的路径曲折程度较大, 且所需要存放的路径点很多, 此时可对路径进一步简化. 本文采用贪婪法对已

获得的子路径进行合并, 去除多余的中间路径点, 如图 5 所示. 图 5 中: 粗实线表示随机环境和 I 型障碍环境时算法路径规划结果, 细实线表示对已规划的路径进行子路径合并后得到的新路径. 可以看出, 新路径比原来的路径短很多, 且路径点的数目也大大减少.

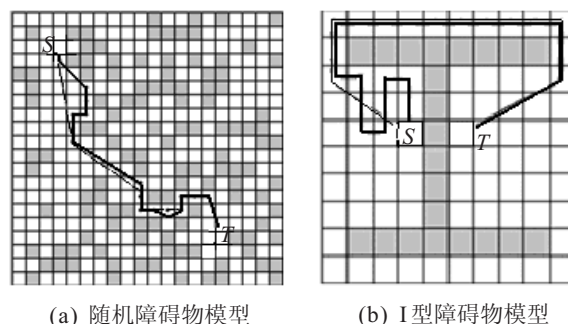


图 5 子路径合并结果示意图

3.5 算法描述

Step 1: 根据起点 p_{start} , 目标点 p_{end} 和障碍环境 E 确定首相交栅格 g_{Inter} , 若 g_{Inter} 不存在, 则转至 step 4; 否则, 转至 Step 2.

Step 2: 按照第 3.2 节的规则在 g_{Inter} 邻域内搜索距离当前路径点最近的自由栅格作为下一个路径栅格. 若不存在, 则按照第 3.3 节所述规则进行路径点回退或邻域变异操作; 否则, 将新得到的路径点加入路径中, 转至 Step 3.

Step 3: 对于当前已规划的所有局部路径, 以路径上最后 1 个元素 (路径点) 为新起点 p_{start} , 重复 step 1 ~ step 2, 逐步获得后续的路径点, 直至 p_{start} 和 p_{end} 可直连为止, 转至 step 4.

Step 4: 在已经获得的可行路线集合中选择最好的路径进行子路径合并, 输出结果.

4 仿真实例与算法性能分析

4.1 仿真实例

选定如下两种实例环境测试:

实例 1 障碍物密度为 $p=0.1$ 的随机障碍环境;

实例 2 由连续封闭几何图形构成的障碍物模型.

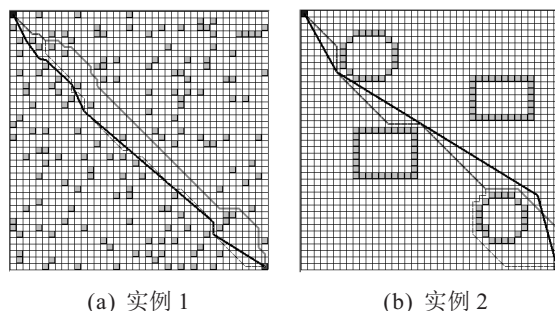


图 6 路径规划结果比较

栅格规模为 40×40 , 大小为 10, 起点、目标点和环境空间的配置如图 6 所示。

4.2 蚁群算法参数设置

蚁群算法的主要参数有蚂蚁数目 antNumber , 最大运动步数 stepMax , 信息素启发系数 α , 距离启发系数 β , 信息素挥发系数 ρ , 最大进化代数 maxGen . 选择不同的参数对算法的影响很大, 为了保证蚂蚁尽可能找到可行路径, 设定 $\text{antNumber} = 20$, $\text{stepMax} = 4 \times (M + N)$, 其中 M 和 N 为栅格的行数和列数. 对于规模为 40×40 , 栅格大小为 10 的规划空间, 将起点、目标点和障碍物分别放置在左上角、右下角和对角线中心位置, 利用蚁群进行测试的结果如表 1 所示 (A^* 算法规划距离为 563.259). 由表 1 可知, 随着进化代数的增加, 算法规划的路径变好. 在 ρ 值固定的情况下, $\alpha = 5$, $\beta = 1$ 时与其他两组参数相比呈现出更好的性能, 突出了信息素的主导作用.

表 1 $\rho = 0.5$ 时不同参数的测试结果

maxGen	α	β	distance	time / s
20	5	1	1 727.523	13.151
20	2.5	2	/	15.683
20	1	5	980	9.910
50	5	1	563.259	15.262
50	2.5	2	1 465.685	28.766
50	1	5	800.00	12.937
100	5	1	563.259	17.666
100	2.5	2	563.259	25.099
100	1	5	563.259	29.703
200	5	1	563.259	29.358
200	2.5	2	563.259	54.968
200	1	5	563.259	52.526

表 2 为 $\alpha = 5, \beta = 1, \text{maxGen} = 100$ 的情况下, 选取不同 ρ 值的测试结果. 由表 2 可以看出, 经过 100 次进化均能找到最好解, $\rho = 0.5$ 时算法的时间性能最佳. 为此, 蚁群算法参数选用 $\alpha = 5, \beta = 1, \text{maxGen} = 100, \rho = 0.5$.

表 2 ρ 取值不同时算法规划的结果比较

ρ	distance	time / s
0.1	563.259	27.986
0.3	563.259	19.070
0.5	563.259	17.003
0.7	563.259	33.341
0.9	563.259	23.115

4.3 仿真结果与性能分析

为了验证算法的性能, 针对上述两种实例进行测试, 在 CPU 2.0 G, 内存 1 G 配置下, 利用 VC++ 6.0 编程分别对本文提出的新算法 GIB (geometry intersection based algorithm), 蚁群算法 (ACA) 和 A^* 算法进行测试, 结果如图 6 所示. 图 6 中: 黑色粗实线表示 GIB 算法, 虚线表示蚁群算法, 灰色粗实线表示 A^* 算法. 由图 6

可知, 采用 GIB 算法得到的路径弯折次数较少, 算法规划的路径能紧贴着障碍物边界迅速到达目标点. 主要原因有: 1) 采用当前路径点到目标点连线确定首相交栅格避免了对中间路径点进行不必要的处理; 2) 采用贪婪法对子路径进行合并, 消除了中间路径点, 使得规划的路径更为合理.

表 3 为 3 种算法规划路径时性能比较, 由表 3 可知, GIB 算法得到的路径最好, 与 A^* 算法相比缩短了约 6%, 与蚁群算法相比缩短了约 7%. 由于确定首相交栅格时求解交点增加了计算时间, GIB 与 A^* 算法相比耗时较长, 但算法经过少量的计算 (不足 200 ms) 即可找到最好解. 这是因为 GIB 算法所需的交点求解次数与环境的行 (或列) 数呈线性关系, 且确定下一个路径点时至多只需测试邻近的 3 个邻居. 从得到的路径长度可以看出, GIB 在计算时间上所付出的代价是值得的.

表 3 3 种算法规划路径时性能比较

测试实例	distance			time / ms		
	ACA	A^*	GIB	ACA	A^*	GIB
实例 1	612.548	604.264	565.445	12 766.3	7.3	193.6
实例 2	606.690	592.548	562.604	15 504.7	6.0	69.66

5 结 论

本文针对机器人路径规划问题, 提出一种基于几何相交测试的路径规划方法. 该方法通过求解首相交障碍栅格快速地消除了中间路径点的计算; 通过回退操作和邻域变异操作有效地避免了搜索陷入死循环; 通过邻域搜索生成多条可行路径, 从而避免了最优路径的遗漏; 通过路径点合并, 使得规划路径更为合理、连续. 实验结果表明, 新方法规划的路径性能优于其他同类算法, 规划时间能满足实际应用的要求. 下一步的研究工作是结合 A^* 的启发式思想、空间多粒度分块表示方法和层次树技术, 对动态环境下的路径进行规划.

参考文献(References)

- [1] David S, Premysl V. Accelerated A^* trajectory planning: Grid-based path planning comparison[C]. Proc of the 2009 IEEE Int Conf on Automated Planning and Scheduling. Thessaloniki, 2009: 74-81.
- [2] Tan G Z, He H, Aaron S. Ant colony system algorithm for real-time globally optimal path planning of mobile robots[J]. Acta Automatica Sinica, 2007, 3(33): 270-281.
- [3] Wang H J, Xiong W. Research on global path planning based on ant colony optimization for AUV[J]. J of Marine Science and Application, 2009, 8(1): 58-64.