

一种改进的全区域覆盖算法

曾维彪, 蔡自兴

(中南大学信息科学与工程学院, 长沙 410083)

摘 要: 针对未知静态区域, 在 Acar 算法的基础上提出一种改进的全区域覆盖算法。该算法根据机器人沿障碍物边界行走时, 机器人行走方向是否发生改变来检测关键点, 按一定规则依次覆盖所有障碍物的相邻单元, 当机器人找到凹关键点且左、右凸关键点集都为空时, 全区域覆盖成功, 解决了 Acar 算法的一些缺陷。实验验证了算法的可行性和正确性。

关键词: 全区域覆盖; 关键点; 移动机器人

Improved Complete Coverage Algorithm

ZENG Wei-biao, CAI Zi-xing

(College of Information Science and Engineering, Central South University, Changsha 410083)

【Abstract】 An improved complete coverage algorithm based on Acar algorithm is presented for static environment with unknown obstacles. If the walking direction of robot is changed, the critical points can be detected. the neighboring cells of all obstacles are covered by robot according to some rules. When the robot finds concave critical points and both the left and the right convex critical point set are empty, the complete coverage is accomplished. The algorithm can overcome some defects of Acar algorithm. Experiments validate the feasibility and correctness of the algorithm.

【Key words】 complete coverage; critical point; mobile robot

1 概述

在现实生活中, 全区域覆盖有着广泛的应用, 如军事侦察、室内清扫、修剪草坪等。基于传感器的全区域覆盖是指在未知环境中利用传感器的信息寻找一条路径, 使机器人遍历所有能走到的空闲区域。Huang最早提出基于传感器的全区域覆盖^[1], 但没有讨论机器人的具体实现。Butler要求机器人装载接触传感器且障碍物和环境为方型^[2]。

Acar算法在理论和实际上都取得了很大成功^[3], 但也存在一定限制:

(1) 要求机器人装载全方位传感器。

(2) 要求障碍物边界曲线光滑, 切线函数不产生任何退化的关键点, 切线不能与障碍物边界平行。

(3) 同一切线上不能存在多个关键点。

(4) 采用 Reeb 图表示单元和关键点的连接关系以保证完全覆盖区域, 但限定一个单元仅连接 2 个关键点, 否则 Reeb 图无法或很难表示单元和关键点的连接关系。

本文在 Acar 算法的基础上, 提出一种改进的全区域覆盖算法, 可解决上述部分问题。

2 一种改进的全区域覆盖算法

2.1 基本概念

切线: 切线函数 h 是一实值函数: $CS \rightarrow R$, 一条切线 CS_i 是 h 的原象, 即 $CS_i = \{x \in CS \mid h(x) = \lambda\}, \lambda \in R$, CS 指机器人的工作区域^[3]。

覆盖方向: 令覆盖方向为从左至右。

关键点: 切线与障碍物边界的交点数为切线连通数, 使切线连通数发生改变的障碍物边界上的点为关键点^[3]。

根据覆盖方向和障碍物类型, 关键点有左凹、右凹, 左凸、右凸 4 种, 如图 1 所示。

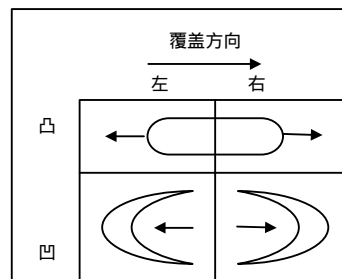


图 1 关键点的类型^[3]

机器人行走方向: 设机器人在 t_0 时候的坐标为 (x_0, y_0) , 在 t_1 时候的坐标为 (x_1, y_1) , 如果: $x_0 < x_1$, 机器人行走方向为从左至右; $x_0 = x_1$, 机器人行走方向不变; $x_0 > x_1$, 机器人行走方向为从右到左。

当前关键点: 当前时候的左凸关键点集中, X 坐标最小、 Y 坐标最大的关键点为当前关键点。包含当前关键点的障碍物为当前障碍物。

右端点: 左凸关键点集为空时, 右凸关键点集中 X 坐标最大、 Y 坐标最小的关键点。

新起点: 机器人从右端点处直线往下行走走到区域边界上某点, 该点为新起点。

最近关键点: 在环形运动的一个循环中, 如果: (1) 机器人在前进阶段检测到关键点 CP, 则 CP 是最近关键点。

(2) 机器人在前进阶段检测到 0 个或 1 个关键点, 在后退阶段

基金项目: 国家自然科学基金资助重点项目(60234030); 国家基础研究基金资助项目(A1420060159)

作者简介: 曾维彪(1981 -), 男, 硕士, 主研方向: 智能机器人; 蔡自兴, 教授、博士生导师、纽约科学院院士

收稿日期: 2008-01-20 **E-mail:** zengweibiao168@21cn.com

检测到 1 个或多个关键点, 但回归阶段没有检测到关键点, 则后退阶段中最后一个检测到的关键点是最近关键点。(3) 机器人在前进阶段和后退阶段都没有检测到关键点, 但回归阶段检测到 1 个或多个关键点, 则最后一个检测到的关键点是最近关键点(注: 回溯阶段不检测关键点; 前进阶段、后退阶段、回归阶段、回溯阶段的概念详见 2.3.1 小节)。

2.2 关键点的检测方式

本文设定区域边界不和任何障碍物相连, 从而区域边界不存在关键点, 机器人仅沿障碍物边界行走时才检测关键点。

(1) 右凸关键点: 机器人沿障碍物边界行走时, 处于障碍物边界右边且机器人行走方向由从左至右变为从右至左。

(2) 右凹关键点: 机器人沿障碍物边界行走时, 处于障碍物边界左边且机器人行走方向由从左至右改变为从右至左。

(3) 左凸关键点: 机器人沿障碍物边界行走时, 处于障碍物边界左边且机器人行走方向由从右至左改变为从左至右。

(4) 左凹关键点: 机器人沿障碍物边界行走时, 处于障碍物边界右边且机器人行走方向由从右至左改变为从左至右。

2.3 环形运动

为避免遗漏关键点, 机器人以环形运动的方式覆盖区域, 环形运动的运动轨迹类似Acar算法的循环运动^[3], 由多个循环组成, 如图 2 所示。

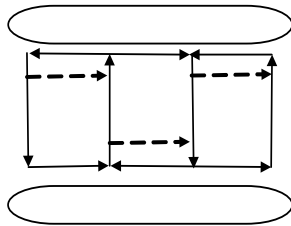


图 2 环形运动

2.3.1 环形运动的一个循环

(1) 前进阶段

机器人从循环起点沿切线行走走到对面边界, 再按覆盖方向沿边界前进, 当下面任意一种情况发生时, 前进阶段终止: 1) 到达预期切线。2) 机器人没有到达预期切线, 在沿障碍物边界行走时改变机器人行走方向, 此时有: 如果找到凸关键点, 机器人沿边界后退(后退过程中, 机器人不检测关键点), 直到与关键点的 Y 坐标相同, 前进阶段终止; 如果找到凹关键点, 前进阶段终止。

(2) 后退阶段

如果前进阶段到达预期切线或找到右凸关键点, 则机器人从当前位置沿切线行走走到对面边界, 再按相反的覆盖方向沿边界前进; 如果前进阶段中找到凹关键点, 则机器人从当前位置继续沿边界前进。

当下面任意一种情况发生时, 后退阶段终止: 1) 到达包含循环起点的切线。2) 没有到达包含循环起点的切线, 但在沿边界行走时找到凸关键点。此时, 新的后退阶段开始, 机器人在关键点处直线往(上/下)行走走到某边界, 再按相反的覆盖方向沿边界行走。如后退阶段中找到凹关键点, 则机器人继续沿边界前进。

(3) 回归阶段

后退阶段终止后, 机器人到达包含循环起点的切线。此时, 如果机器人处于循环起点, 则本次循环结束, 否则开始回归阶段。

机器人沿切线往循环起点行走, 如果机器人遇见障碍物,

则按覆盖方向沿障碍物边界行走, 绕过障碍物回到包含循环起点的切线(注: 机器人绕过障碍物时会找到关键点, 与后退阶段不同, 回归阶段中机器人必须回到包含起点的切线)。

(4) 回溯阶段

在回溯阶段中, 机器人不检测任何关键点, 机器人改变行走方向与检测关键点无关。

如果前 3 个阶段没有找到关键点, 则机器人从循环起点沿边界走到包含下一个循环起点的切线。

如果前 3 个阶段找到关键点, 但不是当前关键点则有:

1) 若循环起点处于当前障碍物的边界, 则机器人沿边界行走走到包含最近关键点的切线。2) 若循环起点不处于当前障碍物的边界, 则机器人从循环起点沿切线走到当前障碍物的边界, 再沿边界行走走到包含最近关键点的切线。

如果前 3 个阶段找到了当前关键点, 意味着已完成当前障碍物上下相邻单元的覆盖, 回溯阶段不予执行。

回溯阶段可保证机器人环绕当前障碍物边界一周。同时, 若一个循环中找到关键点, 回溯阶段将使机器人处于最近关键点。前 3 个阶段示意图如图 3~图 5 所示。

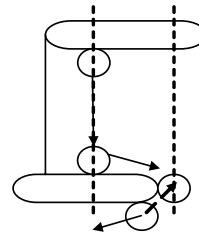


图 3 前进阶段

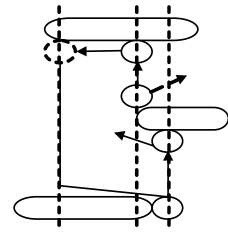


图 4 后退阶段

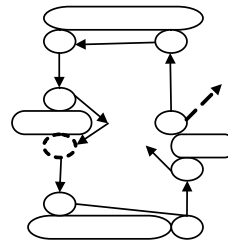


图 5 回归阶段

2.3.2 一个循环中检测到关键点的处理

(1) 找到凹关键点, 置存在凹关键点的标志为真, 但不存储其坐标。

(2) 找到右凸关键点, 存储其坐标到右凸关键点集。

(3) 找到左凸关键点且当前时候不存在当前关键点, 存储左凸关键点的坐标到左凸关键点集中。

(4) 找到左凸关键点且当前时候存在当前关键点, 则有:

1) 如果左凸关键点处于当前障碍物的边界且正好是当前关键点, 则不存储其坐标, 并从左凸关键点集中删除当前关键点。2) 如果左凸关键点处于当前障碍物的边界但不是当前关键点, 则不存储其坐标。3) 如果左凸关键点不处于当前障碍物的边界, 则存储其坐标到左凸关键点集中。

2.3.3 一个循环结束后当前单元的选择

(1) 存在凹关键点的标志为真且左、右凸关键点集为空, 算法结束。

(2) 没有找到任何关键点, 继续覆盖当前单元。

(3) 找到的关键点中不存在当前关键点, 从循环结束处开始覆盖新的当前单元(前方必是已覆盖区域)。

(4) 找到的关键点中存在当前关键点, 则有: 1) 如果左凸

关键点集非空, 机器人定位到新的当前关键点, 开始覆盖新的当前单元; 2) 如果左凸关键点集为空但右凸关键点非空, 则机器人定位到右端点, 置右凸关键点集为空, 机器人从右端点行走到新起点, 开始覆盖新的当前单元。

(5) 机器人的当前位置已覆盖, 则沿边界穿越已覆盖区域, 开始覆盖新的当前单元。

2.4 算法思想

针对存在未知障碍物的静态区域, 已知区域边界且区域边界不和障碍物相连。

机器人从区域最左下端开始, 根据覆盖方向以环形运动的方式覆盖空闲区域, 覆盖过程中对已覆盖区域做某种标记以避免重复覆盖。机器人沿障碍物边界行走时, 如果机器人行走方向发生改变, 则可找到关键点。

根据关键点位置, 采用Boustrophedon分解法实现区域的单元分解^[4]。根据关键点的类型, 存储左、右凸关键点的坐标到左、右关键点集或置存在凹关键点的标志为真等。

如果左凸关键点集非空, 则根据左凸关键点集中关键点的坐标定出当前关键点, 机器人依次环绕覆盖当前关键点所在障碍物的上下相邻单元。

如果左凸关键点集为空而右凸关键点集非空, 则机器人行走到新起点, 并清空右凸关键点集, 表明完成了新起点以左所有空闲单元的覆盖。机器人继续按覆盖方向以环形运动的方式覆盖空闲区域。

如果左、右凸关键点集均为空且存在凹关键点的标志为真, 则全区域覆盖成功。

2.5 算法步骤

(1) 机器人在区域最左下端; 置存凹关键点的标记为假。

(2) 机器人的当前位置是否已覆盖? 是: 机器人沿边界穿越已覆盖区域, 转(3)。否: 转(3)。

(3) 机器人以环行运动的方式覆盖空闲区域并标记已覆盖; 在一个循环中, 如果机器人找到凹关键点, 则置存在凹关键点的标记为真, 但不存储凹关键的坐标。

(4) 环行运动的一个循环结束后, 机器人是否找到关键点? 是: 转(5)。否: 转(2)。

(5) 循环中找到的关键点中是否有凹关键点? 是: 转(6)。否: 转(7)。

(6) 左、右凸关键点集是否为空? 是: 算法结束。否: 置存在凹关键点的标记为假, 转(2)。

(7) 循环中找到的关键点中是否有凸关键点? 是: 转(8)。否: 转(2)。

(8) 当前时候是否存在当前关键点? 是: 转(9)。否: 转(2)。

(9) 循环中找到的关键点中是否有当前关键点? 是: 转(10)。否: 转(2)。

(10) 当前障碍物环绕覆盖完毕, 从左凸关键点集中删除当前关键点。

(11) 左凸关键点集是否为空? 是: 转(12)。否: 转(13)。

(12) 从右凸关键点集中选择右端点, 机器人从右端点走到新起点, 清空右凸关键点集, 转(2)。

(13) 选择新的当前关键点, 机器人定位到新的当前关键点, 转(2)。

3 实验结果及分析

3.1 实验场景

实验在中南大学智能所自行研制的移动机器人 MORCS-1 上进行^[5], 实验环境为隔离板、墙壁等围成一个封

闭区域, 一些机箱盒组成一个方型障碍置于区域中间。机器人初始位置位于区域的最左下端。

为与 Acar 算法更好地进行比较, 实验特意将机箱盒组成方型障碍物, 机器人行走的切线与方型障碍物边界平行。

3.2 实验结果

实验结果如图 6 所示, 浅色表示机器人的覆盖区域, 深色表示区域边界或障碍物。

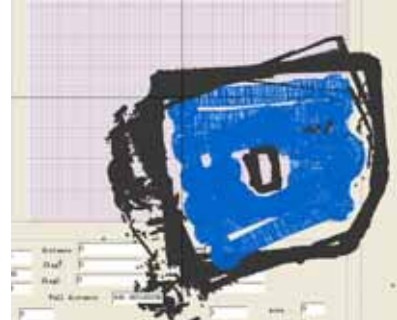


图 6 全区域覆盖成功

由于要保证机器人的安全性, 机器人与墙壁或障碍物要保持一定距离, 见图 6 中区域边界和障碍物周边的空白部分, 部分浅色区域与深色区域重叠以及右上部分的小黑色块是机器人环境建模的原因, 实际上机器人没有撞到边界, 右上部分也不存在小障碍物。

由于机器人的物理特性, 机器人做环形运动时, 每次转弯不一定正好是 90° , 有时会遗漏部分空闲区域, 有时会重复覆盖部分区域。

3.3 实验分析

3.3.1 机器人的运动轨迹

MORCS-1 从区域最左下端开始覆盖区域, 机器人的运动轨迹如图 7 所示。

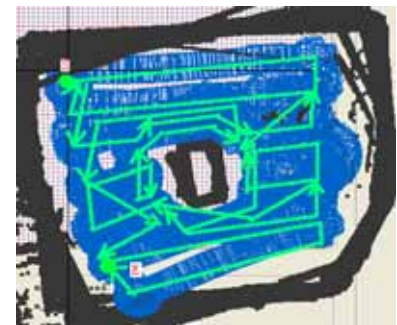


图 7 机器人的运动轨迹

(1) 在某个循环中, MORCS-1 找到右凸关键点, 当次循环结束后, 运用机器人自身的反应式行为控制系统定位到右凸关键点^[5]。

(2) 在某个循环中, MORCS-1 找到当前关键点, 当次循环结束后, 当前障碍物上下相邻区域的覆盖完成。

(3) 在某个循环中找到右凹关键点, 且左、右关键点集均为空, 本次循环结束后, 全区域覆盖成功。

3.3.2 结果分析

MORCS-1 的激光雷达传感器能获取前方 180° 的环境信息, 在实验中运用机器人沿障碍物边界行走时, 机器人行走方向是否发生改变的方法可很好地检测关键点, 验证算法不需要全方位传感器。

MORCS-1 能处理由机箱盒组成的方型障碍物以及切线

(下转第 198 页)