文章编号:1674-2869(2013)04-0066-06

记忆运动方向的机器人避障算法

鲁统伟[1],林 芹[1],李 熹[2],邹 旭[2]
(1.武汉工程大学计算机科学与工程学院,湖北 武汉 430074;
2.智能机器人湖北省重点实验室,湖北 武汉 430074)

摘 要:在一般的避障环境中,Tangentbug算法表现的非常鲁棒,但当避障环境中有对称障碍物的时候,Tangentbug算法容易产生路径的死循环,从而导致终点不可到达.然而在机器人避障过程中,对称障碍物是非常常见的.针对这个问题,提出了基于记忆机器人运动方向的Tangentbug算法.该算法中,机器人每经过一个位置点,就把当前位置点和选择的运动方向记录下来,为后面的更新运动方向做好准备.首先,机器人扫描到障碍物时计算出机器人与障碍物的相遇方向;其次,根据障碍物的边缘,统计局部地图信息,得到局部切线图,找到离终点和当前点距离和最近的点作为机器人的下一个目标点,得到机器人的运动方向;然后,在机器人绕行障碍物时结合记忆的运动方向和局部切线图产生的最小距离和更新下一步运动方向.在整个避障过程中,不停的更新相遇方向和运动方向,最终实现机器人的直行和绕行,从而到达终点.通过大量实验验证,实验结果表明该算法不仅可以实现机器人在对称障碍物环境中顺利到达终点,也可以在非对称障碍物环境中达到终点,验证了该算法的有效性和鲁棒性.

关键词:Tangentbug;避障;运动方向;仿真
中图分类号:TP391.9 **文献标识码**:A **doi**:10.3969/j.issn.1674-2869.2013.04.015

0 引 言

机器人同时定位与地图创建即SLAM(Simultaneous Location and Mapping)指的是机器人在自身位置不确定的条件下,在完全未知的环境中创建地图,同时利用地图进行自主定位和导航.他需要将移动机器人的位置和环境特征坐标表示在一个状态向量中,在机器人的步行过程中通过对环境特征的观测做最有准则的估计和更新[1-2].SLAM中的问题可以归结为以下四点:地图表示、不确定信息处理、数据关联、搜索规划[3-4].其中搜索规划一直是移动机器人导航技术的重要环节.

所谓搜索规划是指移动机器人在有障碍物的工作环境中寻找一条从给定起点到终点的适当的运动路径,使机器人在运动过程中能安全、无碰撞的绕过所有障碍物[5-6].

SLAM导航算法发展至今,已经有很多代表性算法[7-8].如Lumelsky和Stepanov1980年首次提出的bug1、bug2算法,随后提出的Alg1、Alg2算法,都是事先规定了某一个运动方向,简单易懂,但是计算时间、路径长度都很长[9].Kamon和Rivlin在1997年提出的Distbug算法,由于不需要记住以前的点,因此内存要求比较少,路径和Alg2相似,路径长度和时间复杂度都比较长[9].以及1994年Stentz提出的动态A*算法,能够产生很短的路径,但是由于时间复杂度和空间复杂度都非常大,需要创建地图.Tangentbug(切线局部规划算法)算法是bug类算法中比较出色的一种[9].相对于以上算法来说,它最突出的特色是可以根据扫描范围生成LTG(Local Tangent Graph,局部切线图),分析LTG智能选择运动方向,计算简单,实时性强,内存要求不高,能够在较短的时间内根据局部信息获得较短的路径并具有全局收敛的特点[10].

1 Tangentbug算法

Tangengbug算法是Kamon和Rivlin在1995年提出的,该算法已被证明能够仅通过局部信息生成较短较优的路径[8].该算法假设机器人有一个距离传感器,能够扫描到半径为R范围为360°的区域,机器人利用传感器收集局部地图信息,生成局部切线图,并利用局部切线图选择路径.

Tangentbug算法流程图如图1所示,其中LTG(Local Tangent Graph)为局部切线图,d_{min}为沿障碍物运动过程中到目标点T的最短距离,

收稿日期:2013-01-13
作者简介:鲁统伟(1979-),男,山东莒南人,讲师,博士.研究方向:数字图像处理与智能系统.

d_{leave}为扫描范围内到目标点 T 最近的距离，H_i 即 Hit 点，为机器人与障碍物相遇的点.

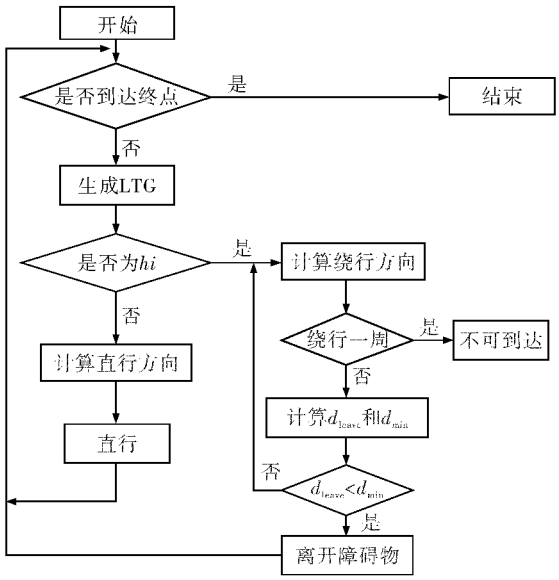

图 1　Tangentbug 流程图

Fig. 1　Flow chart of Tangentbug

Tangentbug 算法中最主要的两个部分为生成 LTG(局部切线图)和运动方向选择. 它们贯穿算法的始终.

LTG(局部切线图)是机器人在行走过程中通过传感器获得的局部地图信息. 机器人在运动过程中通过传感器感知障碍物，并记录以机器人 local_point(当前位置)为中心，传感器扫描范围 R 为半径的圆与障碍物的交点 O_i，连接 O_i 与 local_point 和终点 T，生成局部切线图[11-12]. 机器人通过切线图中各点的信息，如该点到机器人当前位置和终点的距离和，计算出下一步的移动路径.

运动方向确定：机器人在行走过程中始终通过传感器扫描当前环境，并通过计算 LTG 来获得运动方向 next_point(下一目标点). 机器人传感器扫描范围内无障碍物时，机器人计算扫描范围上所有点到终点的距离即 $d(x,T)$，并找到最近点即机器人到终点的直线与扫描范围的交点作为运动方向[13]. 当扫描范围内出现障碍物时，对 LTG 中每个交点 O_j，计算距离和 $\mathrm{Dist}(O_j)=d(x,O_j)+d(O_j,T)$，当该等式最小时点 O_j 记做 O_{min}，选择 O_{min}作为下一个运动方向点 next_point.

2　基于记忆运动方向的 Tangentbug 算法

按照 Tangentbug 算法中根据计算 O_{min}(距离和最小点)来选择下一个运动方向 next_point 时，在某些环境特别是对称环境中会出现路径循环. 如图 2 所示，在点 P_1 处，扫描计算得到的 O_{min} 为 P_2 和 P_0，选择 P_2 作为 next_point 时，运动到 P_2 后扫描计算所得的下一个运动方向 next_point 又为 P_1，这样循环往复从而导致终点不可达.

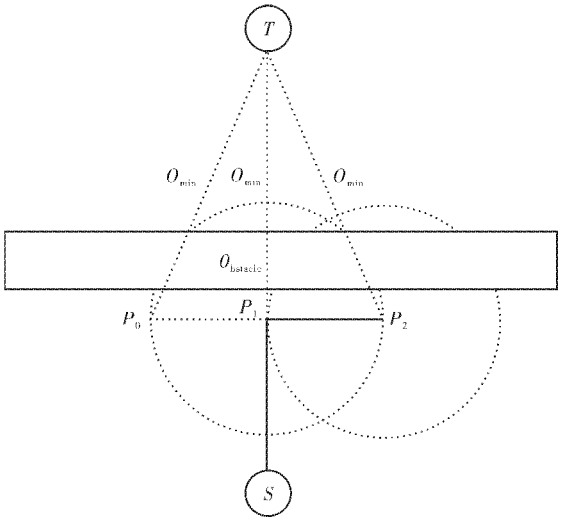

图 2　路径循环示例

Fig. 2　The example of cycle

为避免此类情况，提出基于记忆机器人运动方向的 Tangentbug 算法. 该算法主要包括两个部分：机器人直线运动和机器人绕障碍物移动[12-13]. 该算法不仅通过计算 O_{min}来确定 next_point，而且计算机器人与障碍物相遇方向 hit_direct 和机器人运动方向 move_direct，并记住运动方向. 在绕行中根据运动方向并结合计算 Dist(距离和)来选择 next_point，在直行和绕行转换时候根据 hit_direct 计算并更新机器人运动方向 move_direct.

Tangentbug 算法和其他 SLAM(同时定位与地图创建)算法一样将机器人比拟为一个质点，但是在实际比赛中是不可行的，因此将机器人设定为一个形状的物体，并在算法中考虑到其形状对是否遇见障碍物和 next_point 的影响做出修改.

Tangentbug 算法问题可以细分为以下几个方面：1)机器人与障碍物相遇方向的计算；2)机器人运动方向的计算；3)确定机器人直线运动的方向；4)确定机器人从直线运动到沿障碍物边界运动的切换点即 Hit 点；5)确定机器人绕障碍物边界运动的方向；6)确定机器人离开障碍物开始直线运动的切换点即 leave 点；7)是否能够到达终点. 具体思路如下：

在扫描过程中，对扫描圆上每个点 scan_point[i](scan_point 数组，用于存放扫描圆上的点信息)处，计算以下信息：该点像素值 clr；该点到机器人位置 local_point 和终点的距离和 Dist(scan_point[i])；机器人运动方向 move_direct；机器人与障碍物的相遇方向 hit_direct；是否为 Hit 点(相

遇点)；是否到达终点；是否为 leave 点(离开点).并将扫描的点的信息存入 scan_point 数组中.根据数组中点的颜色信息，分割数组：即将数组中的第一个颜色信息为障碍物的点 i 和最后一个颜色信息为障碍物的点 j 及其之间的点放入数组 obstcl_point(用于存放障碍物信息的障碍物数组)，表示障碍物，其余点放入 free_point(用于存放自由点的数组)中，表示可行区域.

1)障碍物相遇方向的计算：当扫描范围内出现障碍物时，对 obstcl_point(障碍物数组)中点的 hit_direct(相遇方向)进行统计，分别比较方向为上下左右的数目，选择其中数目最大者作为障碍物相遇方向.其中每个点的上下左右方向则通过比较该点与起点的 XY 坐标轴的距离差 sub$x$(x 方向距离差)和 sub$y$(y 方向上距离差)来确定.当 sub$x$ 的绝对值大于 sub$y$ 时运动方向为上下方向，反之为左右方向.

2)运动方向的计算：move_direct(运动方向)在没有扫描到障碍物时仅通过比较与起点的 XY 坐标轴的距离差 sub$x$(x 方向距离差)和 sub$y$(y 方向上距离差)来确定.在扫描到障碍物时候，运动方向要根据障碍物相遇方向来确定，当障碍物相遇方向为上下时，即障碍物在机器人上下方向这时候运动方向仅比较 sub$x$ 来确定为左或右，反之当障碍物在机器人左右方向上，机器人的运动方向仅判断是上或者下.

3)机器人直线运动：机器人直线运动可以分为两种情形：一是在该点处机器人扫描范围内无障碍物及该点为 leave 点(离开点).此时将数组 free_point(空闲数组)中的数值按照插入排序法排序.从最小点开始循环判断，从起点到出发到该点过程中是否会撞到障碍物，及该点的运动方向与上次直线运动的方向是否一致.找到符合的点即不会撞见障碍物且与上次直线运动方向一致作为 next_point(下一运动目标)，并更新 move_direct(运动方向).二是在该点处机器人扫描范围内出现障碍物，并能够到达且未到达终点，分别对数组 free_point 和 obstcl_point(障碍物数组)进行排序，并求出最小值，并将两个数组的最小值进行比较.选择途中不会撞见障碍物的且运动方向与上次直线运动方向一致的点作为 next_point 并更新 move_direct，此时 move_direct 应根据障碍物相遇方向重新计算.

4)Hit 点的确定：Tangentbug 算法和其他算法一样，将机器人看作一个质点，但是在实际中，特别是避障比赛中，是不可以看成质点的，因此在仿真中假设机器人为一个长方体，在避障中主要考虑长和宽.在扫描过程中通过判断长方形上离中心距离为 L 的八个方向上的点的像素值来判断是否为障碍物.如图3所示，当 A 点像素为 RGB(169,169,169)则表示机器人遇见障碍物，机器人当前位置为 Hit 点(相遇点).

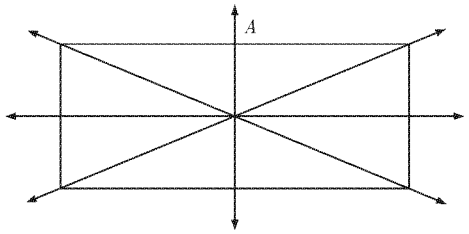

图3 机器人判断 Hit 的八个方向

Fig. 3 The eight direction of robot to check Hit

5)绕行方向：绕行方向主要通过计算 LTG(局部切线图)中交点信息得到.在绕行时，通过传感器扫描障碍物，生成 LTG 图，并通过遍历 LTG 图中所有 O 点(切点)信息(包括该点的像素，该点的 Dist 值，该点是否为终点等)，并将 LTG 图中 O 点按照 Dist(即该点到当前机器人位置与到终点的距离和)信息进行排序，按照 Dist 增大的方向遍历数组，选择满足以下条件的点作为 next_point(下一运动目标)，并更新 move_direct(运动方向)：**a.** 从起点运动到该点的过程中不会遇见障碍物.**b.** 该点的运动方向与上次机器人的运动方向一致.**c.** 该点在不是机器人第三次到达.

6)Leave 点的确定：首先更新 leave_point(离开点)和 min_point(最小点，即绕行中遇见的点中距离和最小的点)：在绕行中计算 Free_point(空闲数组)和和 LTG 中的 obstcl_point(障碍物数组)的 Dist(距离和)并排序，选择满足绕行方向的三个条件且 Dist 最小的点分别和 min_point 和 leave_point比较，当 free_point 小于 min_point，则用最小值更新 min_point，同理 obstcl_point 最小点小于 leave_point，那么更新 leave_point.比较 min_point 和 leave_point，当满足条件 d_{leave}(离开点的距离和)$<d_{\min}$(最小点的距离和)时跳转到直线运动.

7)能否到达终点：在运动过程中用 vector 数组记住每个 next_point(下一运动目标)，在存入前与数组中的点进行比较.如果出现一次重复，那么表示运动方向选择可能错误，更改该点的运动方向为相反方向继续扫描；如果重复两次则表示机器人又一次回到该点，终点不可到达.

3　Tangentbug 算法的仿真

3.1　环境仿真

智能机器人避障涉及到机器人结构技术、控制技术、传感器技术、障碍物检测与定位、路径规划、导航、和信息融合等等方面，是 FIRA ROBO WORLD CUP(国际机器人足球联盟世界杯)的常规比赛项目. 在该比赛中，裁判在 6×6 m 的场地中摆放用大于机器人的纸板代替的障碍物、终点和用白色胶带代表的边界，其中障碍物至少一个，形状随机且颜色和终点不同. 机器人需要在裁判设计的避障环境中，通过传感器扫描当前环境，根据里程计获知自身坐标和方向，通过视觉处理获知当前环境，并在最短时间内，避开障碍物到达终点. 而对称环境即存在一个或者多个障碍物相对于终点和起点的直线对称，是比赛中常见的避障环境.

在仿真过程中，将机器人设定为 3×4 像素的长方形，场地范围为 400×400 像素的正方形，避障边界不超过场地范围，扫描范围假设为 35 像素. 并用黑色代表边界，灰色代表障碍物，底部圆点代表起点，顶部圆点代表终点. 图 4 为避障比赛中最常见的环境的仿真，且为对称环境.

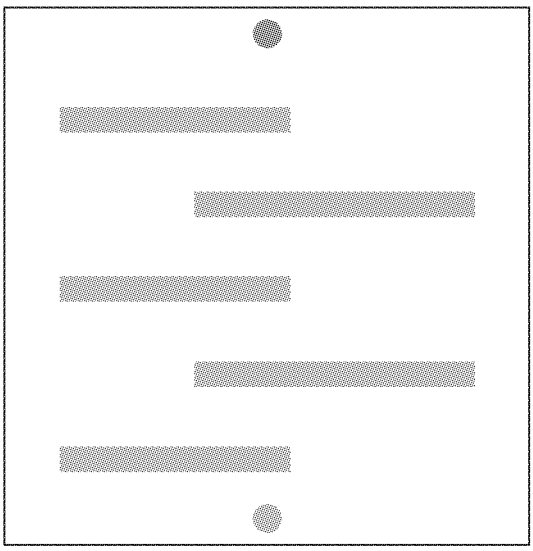

图 4　对称避障环境示例

Fig. 4　The example of symmetric obstacle avoidance environment

3.2　处理过程

扫描仿真：机器人扫描器扫描以机器人为中心，扫描范围 R 为扫描半径的圆内的区域. 仿真通过小角度逼近画圆弧实现扫描，圆的中心为机器人当前位置，半径为扫描范围 R. 画圆函数具体实现思路如图 5 所示，从 0°开始到 360°结束，每隔 θ 度画一条直线(图中实线)，长度为 $R * \sin(i * \theta)$.

识别仿真：在仿真中通过获取颜色 GetPixel()函数获得当前点颜色，并判断颜色值来识别当前环境，如 GetPixel()函数值为 255 时表示为终点. 下一段路径仿真：通过画线实现. 用蓝色直线连接各个 next_point(下一运动目标)仿真机器人运动路径.

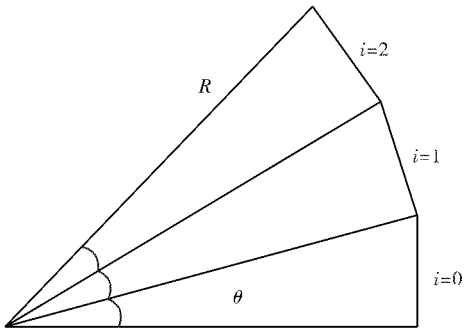

图 5　扫描圆仿真

Fig. 5　Simulation of scan circle

3.3　仿真结果

仿真中分别对 6 种存在对称障碍物环境进行实验，实验结果如图 6 所示.

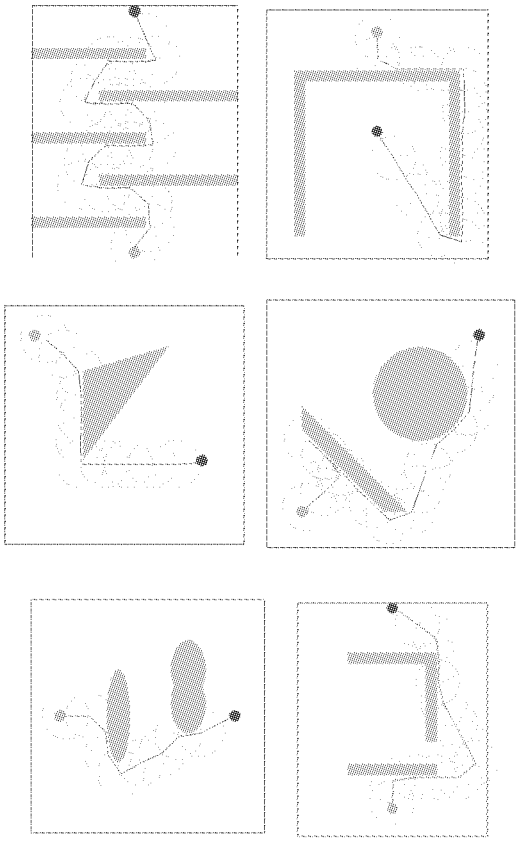

图 6　仿真结果示例

Fig. 6　The example of simulation result

4 结 语

以上针对Tangentbug算法在存在对称障碍物环境中易产生循环路径从而导致不可到达的情况进行改进，提出了基于记忆运动方向的Tangentbug算法.通过计算障碍物相遇方向来更新机器人运动方向，并在绕行中分别保持其运动方向，在直行和绕行转换时更新运动方向的约束下选择下一个运动点.并打破机器人看作质点的约束，将机器人简化为一个长方形，从八个方向进行判断是否遇见障碍物，从而有效避免了机器人与障碍物的碰撞.

从仿真结果可以看出机器人可以按照比赛要求从起点，无碰撞的到达终点.然而机器人避障导航是一项复杂的任务，尽管改进在多种存在对称环境中进行了试验，但是本算法在路径长度、时间复杂度等方面仍需进一步深入研究.

致谢

在算法设计过程中，智能机器人湖北省重点实验室的闵锋和李迅两位老师提供的设备和技术帮助；在实验数据分析过程中，李哲靖和王朝亮同学的积极讨论和解决此过程遇到的各种问题等，在此一并表示诚挚的感谢！

参考文献：

[1] 迟建男，徐心和.移动机器人即时定位与地图创建问题研究[J].机器人，2004，26(1)：92-96.

[2] 张捍东，郑睿，岑豫皖.移动机器人路径规划技术的现状和展望[J].系统仿真学报，2005，17(2)：439-443.

[3] 刘祥，陈建新.一种基于有限视场的移动机器人避障路径规划算法[J].空间控制技术与应用，2008，34(4)：11-16.

[4] 罗荣华，洪炳荣.移动机器人同时定位与地图创建研究进展[J].机器人，2004，26(2)：182-186.

[5] 王坤.液体状态机在机器人足球路径规划中的应用[D].武汉：武汉工程大学，2009.

[6] 孔伟，张彦铎.基于遗传算法的自主机器人避障方法研究[J].武汉工程大学学报，2008，30(3)：110-113.

[7] 何俊学，李战明.基于视觉的同时定位与地图创建方法综述[J].计算机应用研究，2010，27(8)：2839-2843.

[8] 石杏喜，赵春霞，郭剑辉.基于PF/CUPF/EKF的移动机器人SLAM框架算法[J].电子学报，2009，37(8)：1865-1868.

[9] James Ng. A Practical Comparison of Robot Planning Algorithm Given Only Local Information [D]. Australia: The University of Western Australia, 2005.

[10] Yufka A, Parlaktuna O. Performance Com-parison of Bug Algorithms for Mobile robots [C]//5th International Advanced Technologies Sympos-ium (IATS'09). Mustafa ACARER, Halil Ibrahim DEMIRCI, Cevdet GOLOGLU. Karabuk, Turkey: Karabuk University, 2009: 61-65.

[11] Choset H, Lynch K, Hutchinson S, et al. Principles of Robot Motion: Theory, Algorithms and Im-plementation[M]. Cambridge: The MIT Press, 2005: 11-18.

[12] 赵祚喜，汪宁，张智刚，等.一种适用于非360°探测机器人的避障导航算法[J].机械工程学报，2010，46(19)：44-52.

[13] Kamon I, Rivlin E, Rimon E. A New Range-sensor Based on Globally Convergent Navigation Algorithm for Mobile Robots[C]//In Proceedings of the 1996 IEEE International Conference on Robotics and Automation Minneapolis. M T J Tarn. Minneapolis, Minnesota: IEEE Conference Publications, 1996(1): 429-435.

Obstacle avoidance algorithm of robot based on recording move direction

LU Tong-wei[1] ,*LIN Qin*[1] , *LI Xi*[2] ,*ZOU Xu*[2]

(1. School of Computer Science and Engineering, Wuhan Institute of Technology, Wuhan 430074, China;
2. Hubei Provience key Laburatory of Intelligenty Robot, Wuhan 430074, China)

Abstract: Tangentbug algorithm performance is very robust in the general Obstacle Avoidance environment. When the Obstacle Avoidance environment has symmetrical obstacles, Tangentbug algorithm is prone to producing the infinite loop of path leading to the end point unreachable. However, the symmetrical obstacle is very common in the robot Obstacle Avoidance competition. The Tangentbug algorithm based on the memorizing moving direction was put forward. The robot recorded each passing by location point and corresponding moving direction for further updating direction of movement. Firstly, the direction between robot and obstacle was calculated when obstacle was scanned. Secondly, according to the edge of the obstacle, the statistics of local map information was got to find the nearest point to the end point and the local point as the next moving direction. Moreover, the moving direction was updated based on the minimum distance of the Local Tangent Graph and moving direction in memory when robot moved around the obstacle. The meet direction and the moving direction were updated in the entire process of obstacle avoidance. So that the robot would reach the end point by moving straight to the goal and moving around the obstacle. Experimental results show that the robot not only reaches the end point in a symmetrical obstacle environment, but also reaches the end point in the asymmetric obstacle environment, which verifies the effectiveness and robustness of the algorithm.

Key words: Tangentbug; obstacle avoidance; move-direction; simulation

本文编辑：陈小平