

动态环境下基于多人工鱼群算法和避碰规则库的机器人路径规划

徐晓晴, 朱庆保

(南京师范大学计算机科学与技术学院, 江苏南京 210097; 江苏省信息安全保密技术工程研究中心, 江苏南京 210097)

摘 要: 为了提高机器人路径规划的速度、环境适应能力和高效动态避碰问题, 提出了一种基于多人工鱼群的机器人路径规划算法和基于避碰规则库的动态避障算法. 该算法中, 人工鱼以其与目标点的距离为食物浓度, 两个邻近栅格的距离为步长, 其觅食行为作为默认行为, 在一定条件下执行聚群或追尾动作, 并采用两鱼群双向搜索机制在静态环境下规划出较优路径. 在此基础上, 机器人查询动态避障规则库获得避碰方法, 从而实现与动态障碍的避碰. 大量仿真实验结果表明, 该方法具有较高的收敛速度和较强的搜索能力, 能在非常复杂的动静态障碍环境中, 迅速规划出一条安全避碰的优化路径.

关键词: 动态环境; 机器人路径规划; 多人工鱼群; 避障规则库

中图分类号: TP242 **文献标识码:** A **文章编号:** 0372-2112 (2012)08-1694-07

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2012.08.032

Multi-Artificial Fish-Swarm Algorithm and a Rule Library Based Dynamic Collision Avoidance Algorithm for Robot Path Planning in a Dynamic Environment

XU Xiao-qing, ZHU Qing-bao

(School of Computer Science and Technology, Nanjing Normal University, Nanjing, Jiangsu 210097, China;
Jiangsu Research Center of Information Security & Privacy Technology, Nanjing, Jiangsu 210097, China)

Abstract: In order to improve the convergence speed and the environmental adaptability of the path planning algorithm, a robot path planning algorithm based on multi-artificial fish-swarm is proposed. We present also a dynamic obstacle avoidance algorithm based on the rule-base of collision avoidance in dynamic environment to avoid collisions with the moving obstacles. In our approach, the distance between a fish and a goal is defined as food concentration and the distance between two neighbor grids is defined as step length. The preying behavior of fishes is regard as default behavior and perform clusters act or rear-end act is activated in some certain condition. Then the optimal path in static environment is planned by the search mechanism of bi-directional fish-swarms. After that, the effective collision avoidance behavior is obtained, from the obtained dynamic obstacle information through sensors. Many simulation experiments have shown that the algorithm has a fast convergence speed and strong search capability. Even in complex environments which have static and dynamic obstacles, it can avoid collision safely and plan an optimal path rapidly at the same time.

Key words: dynamic environments; robot path planning; multi-artificial fish-swarm; the rule-base of collision avoidance

1 引言

移动机器人路径规划是智能机器人的重要研究内容. 在静态环境下, 国内外学者已经做了大量的工作并取得了很多成果, 诸如传统的自由空间法^[1]和人工势场法^[1]等等. 自由空间法的复杂度与障碍的多少成正比, 因此不适于复杂障碍环境. 人工势场法存在容易陷入局部最优解和终点不可达的问题. 近些年来, 不少学者提出了基于遗传算法^[2]和蚁群算法^[3~6]的智能路径规划方法. 遗传算法容易产生很多含有障碍的无效路径, 影

响了算法的效率, 且实时性较差. 基本的蚁群算法存在搜索时间长, 易于停滞等缺点已成共识. 因此, 研究速度更快、环境适应性更强的算法是不断追求的目标.

在动态环境中, 由于存在运动的障碍物, 机器人易与这些动态障碍发生碰撞, 因而比静态环境的规划要困难的多, 从而引起了学者们的广泛关注并成为近年来的研究热点^[7~18]. 许多在静态环境下的经典算法也被拓展、改进后应用到动态路径规划中. 例如文献^[8~11]提出了改进的人工势场算法; 文献^[12]提出了动态窗口算法. 这些算法在安全躲避障碍方面取得了不错的成果,

然而时间复杂度却随着环境的复杂程度而增加.为了克服这些不足,一些学者提出了改进的蚁群算法,例如,文献[13]首先用蚂蚁算法规划静态环境下的全局路径,在此基础上,进行动态碰撞预测,据此进行局部避障规划.但该算法需要进行二次规划,对规划算法的实时性要求较高.另一类算法是基于预测控制的速度控制算法^[14~17],该类算法通过调整机器人的运动速度来躲避动态障碍,使机器人在一定程度上适应了动态环境.但机器人不能改变行走的路径,无法避开迎面而来的动态障碍.还有一些其它类型的算法,诸如:受滚动约束的轮式机器人运动规划算法^[18]则将优化方法融合到快速随机搜索树中,克服了随机扩展树搜索过于均一、无启发的缺点,但在复杂环境下限制了搜索树的规模,使得有可能无法搜索有效节点,不能及时避开障碍.

在动态环境下,机器人需要实时避碰,从上述文献看,避碰算法受环境约束且时间复杂度高,为此,本文提出了基于避碰规则库的避碰算法,该算法将避碰策略抽象成简单的规则,并建立动态障碍避碰规则库,机器人根据测得的动态障碍信息,通过该规则库直接获得避碰方法,不需要对路径进行二次规划,大大地节省了时间.同时,该规则库可以作为一种模型,应用到简单或复杂的环境,其规则与环境无关,因此算法复杂度不会受环境影响.

另一方面,在动态环境下,要求路径规划用时短,以使机器人能快速安全地躲避动态障碍.为了加快路径规划的收敛速度和环境适应性,提出了基于多人工鱼群的路径规划算法.人工鱼群算法^[19,20]具备并行处理和全局寻优的能力,寻优速度较快,能够快速跳出局部极值点,用于机器人路径规划具有独特的优势.然而该算法是一种连续优化算法,多应用于求解线性方程、优化参数取值等问题,仅有很少文献用人工鱼群算法对机器人用 Dijkstra 算法得到全局路径进行局部优化^[20],虽然优化了路径,但没有对 Dijkstra 算法进行改进,时间复杂度仍然随着环境的复杂程度而增加.为了用人工鱼群算法进行机器人路径规划,本文根据路径规划特点,对人工鱼群算法进行了改进,给出了适于路径规划的拥挤度、视野域等相关概念和人工鱼的行为策略及双鱼群双向搜索算法.

综上,本文算法首用鱼群算法得到静态环境下的全局路径,再根据避碰规则库对动态物进行碰撞预测,并采取相应的避障策略.大量仿真实验结果表明了效果十分令人满意.

2 基于多人工鱼群的机器人路径规划算法

2.1 基于多人工鱼群的机器人路径规划算法的基本思想和算法策略

本文采用栅格法建模,坐标建立、栅格与序号的关系与文献[2]相同.本算法的基本思想是用人工鱼在栅格环境中进行节点寻优选择,通过鱼的觅食、聚群或追尾实现最优路径的搜索.人工鱼群算法是一种连续优化算法,而基于栅格的机器人路径规划是一种离散优化问题.因此,当用人工鱼群算法解决基于栅格的机器人路径规划问题时要解决 3 个问题:(1)食物浓度、视野域及拥挤度因子如何表示;(2)人工鱼如何选择下一步节点,其追尾、聚群、觅食如何进行;(3)如何提高人工鱼的搜索能力.

(1) 基本定义

定义 1 设人工鱼当前位置为 $P(x_f, y_f)$, 食物浓度则表示为 $h = f(P)$, 其中 h 为 P 与目标位置的欧式距离,如式(1)所示:

$$h = \sqrt{(x_f - x_g)^2 + (y_f - y_g)^2} \quad (1)$$

式中, (x_f, y_f) , (x_g, y_g) 分别为 P 和目标位置的坐标;

定义 2 任意栅格间的距离指两栅格间的连线长度,记作 $d(g_i, g_h)$ 或 $d(P(x_i, y_i), P(x_h, y_h))$. $d(g_i, g_h)$ 由式(2)确定:

$$d(g_i, g_h) = \sqrt{(x_i - x_h)^2 + (y_i - y_h)^2} \quad (2)$$

定义 3 $Visual(P(x_f, y_f)) = \{g | g \in A, d(g, P(x_f, y_f)) \leq r\}$ 称作人工鱼的视野域.其中, r 为视野域半径,为设定的常数.由于本文采用栅格法建模,因此,视野域范围表示为以人工鱼所在位置为中心的 $r * r$ 的矩形范围.

定义 4 记 δ 为拥挤度因子,由式(3)计算:

$$\delta = a * N_{\max} \quad (0 < a < 1) \quad (3)$$

式中, a 称为极值接近水平, N_{\max} 为鱼群中人工鱼的总数.

(2) 节点选择及人工鱼行为策略

在人工鱼算法中,人工鱼以随机的步长移动,而在栅格环境中存在障碍,因此人工鱼只能以两相邻的栅格距离作为步长,且只能移动到所在栅格的可行邻域栅格.同理,在人工鱼进行聚群或追尾时,当目标点(最优位置)不在可行邻域栅格时,则只能移动到可行邻域栅格中与目标点最近的栅格.

定义 5 假设任意相邻栅格的垂直或水平距离为 1,记 $allowed_{g_i} = \{g | g \in A, d(g, g_i) \leq \sqrt{2}, g \notin S\}$ 为栅格 g_i 的可行邻域栅格.

定义 6 人工鱼从 $allowed_{g_i}$ 中随机选择一个栅格记作 $random(allowed_{g_i})$.

人工鱼觅食行为是自由搜索的行为,聚群或追尾行为是追逐最优点的行为.聚群或追尾行为有加快收敛速度的功能,但也有可能导致算法陷于局部最优.为

此,我们将人工鱼觅食行为作为默认行为,只有当所走步数满足一定条件时才判断是否采取聚群或追尾行为.我们制定的人工鱼行为策略描述如下(以一条人工鱼为例):

(1)觅食行为:设人工鱼当前栅格为 g_i ,其食物浓度为 h_{g_i} ,下一步要选择的栅格为 g_n ,对应的食物浓度为 h_{g_n} .令 t 表示从 $allowed_{g_i}$ 中选择栅格的次数, tn 表示允许人工鱼从 $allowed_{g_i}$ 中选择栅格的最大

次数.则觅食过程如图 1 所示.

(2)聚群行为:设某人工鱼当前栅格为 g_i ,其伙伴中心位置 g_c 由式(4)确定:

$$g_c = \left(\sum_{j=1}^{N_f} g_j \right) / N_f \quad (4)$$

其中 g_j 代表其它人工鱼所在栅格,且 $g_j \in Visual$, N_f 为视野域内除当前人工鱼外的人工鱼数目.

设下一步要选择的栅格为 g_n , g_n 可由公式(5)确定:

$$g_n = \begin{cases} g_c, & \text{if } h_{g_c} * N_f \leq \text{delta} * h_{g_i} \text{ and } g_c \in allowed_{g_i} \\ \arg \min_{g \in allowed_{g_i}} \{d(g, g_c)\}, & \text{if } h_{g_c} * N_f \leq \text{delta} * h_{g_i} \\ & \text{and } g_c \notin allowed_{g_i} \\ -\infty, & \text{if } h_{g_c} * N_f > \text{delta} * h_{g_i} \end{cases} \quad (5)$$

(3)追尾行为:设某条人工鱼 f_i 的当前栅格为 g_i ,下一步要选择的栅格为 g_n . f_i 在视野域 $Visual$ 中寻找另一条人工鱼 f_j ,假设 f_j 所在栅格为 g_j ,则必须满足: $\forall g_k \in Visual, g_j \in Visual, g_k \neq g_j$,都使得 $h_{g_i} > h_{g_j}$ 成立,其中 g_k 代表其它人工鱼所在栅格.如果存在这样的人工鱼 f_j ,则 g_n 可由公式(6)确定:

$$g_n = \begin{cases} g_j, & \text{if } h_{g_j} * N_f \leq \text{delta} * h_{g_i} \text{ and } g_j \in allowed_{g_i} \\ \arg \min_{g \in allowed_{g_i}} \{d(g, g_j)\}, & \text{if } h_{g_j} * N_f \leq \text{delta} * h_{g_i} \\ & \text{and } g_j \notin allowed_{g_i} \\ -\infty, & \text{if } h_{g_j} * N_f > \text{delta} * h_{g_i} \end{cases} \quad (6)$$

为了提高搜索能力和收敛速度,本文提出了多人工鱼群算法(Multi-artificial Fish-swarm Algorithm, MAF-SA).该算法在起始点和目标点各设置一组鱼群,两鱼群相互协作完成最优路径的搜索.记 $Fish_i = \{f_1, f_2, \dots, f_k, \dots, f_{N_{\max}}\}$ 为一个鱼群中人工鱼的集合($i = 1, 2$), $f_k \in Fish_i$ 表示某条鱼, N_{\max}^i 表示第 i 个鱼群中鱼的总数. $Fish_1$ 中的鱼以 g_{begin} 为起点,以 g_{end} 为食物源; $Fish_2$ 的

鱼反之.两组鱼群相向搜索,以使搜索不易陷于停滞,提高搜索收敛速度.

2.2 算法步骤

定义 7 设人工鱼 $f_k \in Fish_i$ 已走过的节点依次为 $g_a, g_f, \dots, g_h \in A$,这一有序集合记为 f_k 的路径表 $Path_k^i$.

定义 8 令人工鱼 $f_i \in Fish_1, f_j \in Fish_2$, $Path_k^1, f_i$ 和 f_j 分别从 g_{begin} 和 g_{end} 出发,经过 t 个时刻, f_i, f_j 的路径表分别为 $Path_k^1, Path_j^2$,若 $Path_k^1 \cap Path_j^2 \neq \emptyset$,则称 f_i, f_j 相遇.

定义 9 对于人工鱼找到的路径 $Path_k^i = \{g_s, g_i, \dots, g_j, g_g\}$,其长度 len_k^i 由公式(7)得到:

$$len = d(g_s, g_i) + \dots + d(g_j, g_g) \quad (7)$$

由于两个鱼群除了出发点不同外,算法完全相同,因此以下以 $Fish_1$ 的搜索算法为例,并将表示 $Fish_1$ 的下标省略.外层主循环(Step 2-8)寻找从 g_{begin} 到 g_{end} 的路径,重复次数由 MAX 决定.内层主循环(Step 3-7)描述每条人工鱼下一步栅格的选择策略.当两个鱼群中有人工鱼相遇或者人工鱼所走步数超过参数 S_{\max} 时,内层循环终止.其中, MAX 根据实验需要设定, $S_{\max} = row * col * 2$ (row 是栅格的行数, col 是栅格的列数),人工鱼所走步数为 10 的倍数时判断是否采取聚群或追尾行为.算法步骤如下:

Step1 设定视野域半径 r ,极值接近水平 α ,最大选择次数 tn ,根据公式(3)计算出拥挤度因子 delta .设置迭代次数计数器 $n = 0$,步数计数器 $s = 0$.设置最优路径长度 $len = \infty$,并为每一条人工鱼建立路径表 $Path_k^1$ ($k = 1, 2, \dots, N_{\max}$),建立最优路径表 $Gpath$.

Step2 将 N_{\max} 条人工鱼放置在起始点 g_{begin} ,并将 g_{begin} 加入路径表 $Path_k^1$ ($k = 1, 2, \dots, N_{\max}$).设置 $k = 1$, k 用来遍历鱼群.

Step3 如果 $s \% 10 = 0$,转 Step6.

Step4 人工鱼 f_k 当前所在栅格为 g_k ,其可行域栅格域为 $allowed_{g_k}$.设下一步要选择的栅格为 g_n ,设置栅格选择计数器 $t = 0$.

Step5 栅格 $g_n = \text{random}(allowed_{g_k})$,比较 h_{g_n} 和 h_{g_k} .如果 $h_{g_n} < h_{g_k}$ 或 $t > tn$ 时,转 Step7;否则,令 $t = t + 1$,转 Step5.

Step6 产生一个随机数 $fw = \text{rand}(2)$, $\text{rand}(2)$ 表示产生属于 $[0, 2)$ 的整数.

如果 $fw = 0$,由公式(4)计算出中心位置 g_c ,由公式(5)确定 g_n .如果 $g_n \neq -\infty$,转 Step7;否则,转 Step4.

如果 $fw = 1$,在视野域 $Visual$ 中寻找另一条人工鱼,假设此人工鱼所在栅格为 g_j ,此人工鱼要满足: $\forall g_a \in Visual, g_a \neq g_j$,都使得 $h_{g_i} \leq h_{g_j}$ 成立,其中 g_a 代表其它人工鱼所在栅格.如果存在这样的人工鱼,则由

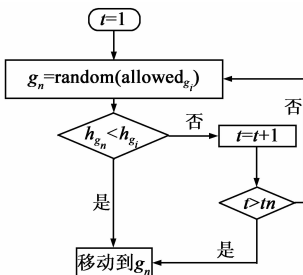


图1 觅食过程

公式(6)确定 g_n . 当 $g_n \neq -\infty$ 时, 转 Step7; 否则, 转 Step4.

Step7 将 g_n 加入到路径表 $Path_k$, 如果 $k < N_{\max}$, 令 $k = k + 1$, 转 Step3.

Step8 当两个鱼群中的人工鱼都走过一步之后, 令 $s = s + 1$, 根据定义 8 检查两个鱼群中是否有鱼已经相遇. 如果相遇, 得到路径 $Path$, 由公式(5)计算出路径长度 len_n . 将 len_n 与历史最优解 len 比较, 如果 $len_n < len$, 令 $len = len_n$, 并将 $Gpath$ 替换为 $Path$, 转 Step9. 否则, 如果 $s < S_{\max}$, 令 $k = 1$, 转 Step3.

Step9 $n = n + 1$, 如果 $n \geq MAX$, 算法结束, $Gpath$ 中保留的为最优路径. 如果 $n < MAX$, $s = 0$, 清空所有人工鱼的路径表, 转 Step2.

3 基于规则推理的动态避碰策略及算法

3.1 避碰分析与避碰策略

机器人和障碍的运动方向如图 2 所示. 用 DB 表示动态障碍, 机器人和 DB 的速度分别用 v_r 和 v_d 表示, 方向分别用 $Rdir$ 与 $Bdir$ 表示. 设机器人走一个栅格所用时间为 t_R . t 时刻机器人在栅格 g_{rnow} 处发现在栅格 g_{bnow} 处有一个 DB . 此后, 机器人沿导航路径继续行进, 其下一步要走的栅格序号为 g_{rnext} , 期间机器人不断探测 DB , 发现 DB 经过栅格 $g_{bnext1}, g_{bnext2}, \dots, g_{bnextn}$, 记 $B_{next} = \{g_{bnext1}, g_{bnext2}, \dots, g_{bnextn}\}$ 且算出 DB 走一个栅格所用时间为 t_D . 在一个不长的时间段内, 机器人假定 DB 做匀速直线运动, 当它探测到 DB 走到下一个栅格后, 可得知其方向和直线轨迹. 根据 v_r 和 v_d 及它们的相对方向, 进行碰撞分析并预测碰撞点, 分析如下:

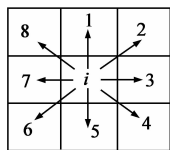


图2 运动方向

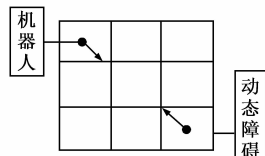


图3 正面碰撞情况

正面碰撞: 当 $Rdir$ 与 $Bdir$ 相反, 且 DB 在机器人前面, $g_{rnext} \in B_{next}$ 则机器人与 DB 必将发生正面碰撞, g_{rnext} 为碰撞点. 当 $Rdir = Bdir$, DB 在机器人前面, 若 $v_d < v_r$, $g_{rnext} \in B_{next}$, 则可能会导致正面碰撞, g_{rnext} 为碰撞点. 在图 3 中将 DB 反向, 可以看出, 由于机器人速度快, 导致机器人与 DB 碰撞.

追尾碰撞: 当 $Rdir = Bdir$, 且 DB 在机器人后面, 如果 $g_{rnext}, g_{bnext} \in B_{next}$, 则可能发生追尾碰撞, g_{rnext} 为碰撞点, 将图 3 中的机器人反向, 即可以看出这是显然的.

侧面碰撞: 当 $Rdir$ 与 $Bdir$ 既不相同也不相反, 则 DB 在机器人侧面, $g_{rnext} = g_{bnexti}$, $g_{bnexti} \in B_{next}$, 若 $t_D * i = t_R$, 则机器人与障碍将发生侧面碰撞, g_{rnext} 为碰撞点.

根据上述碰撞情况, 当机器人预测与障碍发生碰撞时, 采取的避碰策略分为以下三类:

等待策略: 如果为侧面碰撞, 机器人只要在原地等待, 等障碍通过碰撞点即可避碰.

回退策略: 如果为正面碰撞或追尾碰撞, 且机器人在前一步时的运行方向与当前运行方向不同, 这时机器人只需直接回退到上一步栅格即可避碰, 如图 4 所示.

选点策略: 如果为正面碰撞或追尾碰撞且机器人前一步的方向与障碍当前方向同向, 只要在预测的碰撞点邻域重找一个空白点作为下一步点, 即可避碰. 如果预测的碰撞点邻域中不存在空白点且 $v_d \leq v_r$, 则采取回退策略, 回退到上一步栅格. 如图 5 所示.

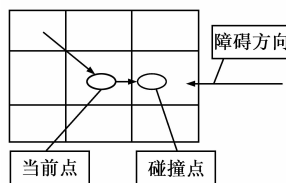


图4 直接回退情况图

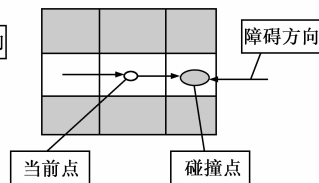


图5 间接回退情况图

3.2 动态避碰算法

机器人首先规划静态环境下最优路径 $Gpath$, 与所有动态障碍将发生碰撞的类型集记为 $L = \{0, 2, 1, \dots, 1, \dots\}$, 其中 0、1 和 2 分别代表侧面碰撞、正面碰撞和追尾碰撞. 设机器人沿最优路径行走的下一栅格为 g_{rnext} , 障碍数记为 bn . 机器人每走一步都判断它走到 g_{rnext} 是否与任一动态物碰撞, 称为一次碰撞判断, 如果有碰撞, 则称 g_{rnext} 为碰撞点, 其判断过程如图 6 所示.

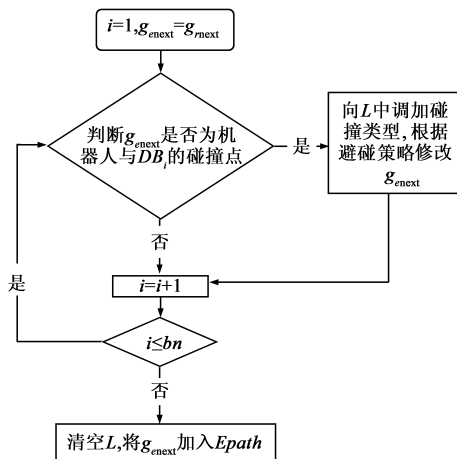


图6 一次碰撞判断过程

3.2.1 动态避碰规则

避碰规则设计为五输入单输出型, 输入为 $\{DIR, NCT, PCTL, N, V\}$, 输出为 $\{P\}$. 其中, DIR 表示机器人上一步运动方向和动态障碍运动方向的关系, NCT 为预测的碰撞类型, $PCTL$ 为预测的碰撞类型集, N 是机器人当

前无碰撞可行域, V 表示机器人与动态障碍的速度关系, P 是机器人将在全局路径上进行调整的类型. 以上参数的定义如表 1 所示, 避碰规则见表 2.

表 1 碰撞规则参数定义

输入 \ 输出	子集	符号含义
DIR	SD, OD, OT	同向, 反向, 其他情况
NCT	NFI, NPI, NSI	正面碰撞, 追尾碰撞, 侧面碰撞
PCTL	LE, LN	LE 表非空, LN 表为空
N	EXIST, NULL	存在可选点, 空
V	DLTR, DEQR, DGTR	$v_d < v_r, v_d = v_r, v_d > v_r$
P	W, DR, SP	等待策略, 回退策略, 选点策略

表 2 避碰规则

IF	THEN	IF	THEN
LN & NSI	W	LN & NFI & SD & EXIST & DEQR	SP
LN & NFI & OT	DR	LN & NFI & SD & EXIST & DLTR	SP
LN & NPI & OT	DR	LN & NFI & OD & EXIST & DEQR	SP
LN & NPI & OD & EXIST & DEQR	SP	LN & NFI & OD & EXIST & DLTR	SP
LN & NPI & OD & EXIST & DLTR	SP	LN & NPI & SD & EXIST & DEQR	SP
LE & EXIST & DEQR	SP	LN & NPI & SD & EXIST & DLTR	SP
LE & EXIST & DLTR	SP		

采用正向推理的方法, 将输入与规则的条件进行匹配, 根据匹配成功的规则结论避碰.

以上规则是根据机器人可能发生的侧面、正面和追尾碰撞分析总结出来的, 包括了所有的碰撞类型, 因此这些规则是完备的.

3.2.2 算法步骤

Step1 用 MAFSA 算法求出静态环境下的最优路径 $Gpath$, $i = 1$ 中栅格的个数为 sl . 初始化动态障碍 DB_j ($j = 1, 2, \dots, bn$) 和机器人动态路径表 $Epath$. 将起始点 g_{begin} 加入 $Epath$, 令 $L = \emptyset$. 设置 $i = 1$, i 用来遍历最优路径表 $Gpath$, 并指向下一步要选取的栅格; 设置 $j = 1$, j 用来遍历所有动态障碍.

Step2 若 $i > sl$, 转 Step6. 获取机器人静态环境下的下一步行走栅格 $g_{next} = g_i, g_i \in Gpath$, 设在动态环境下机器人的下一步栅格为 g_{enext} , 初始化 $g_{enext} = g_{rnext}$.

Step3 如果 $j > bn$, 转 Step5. 机器人确定下一步运动方向为 $Rdir$, 走一个栅格所用时间为 t_R , 由传感器探测出动态障碍 DB_j 的方向 $Bdir_j$, 速度为 v_d , 走一个栅格所用时间为 t_{D_j} . 机器人判断它和 DB 的位置关系并按上述分析判断 g_{rnext} 是否为碰撞点, 同时确定 DIR 和 NCT 的取值. 若 g_{rnext} 不是碰撞点, 令 $j = j + 1$, 转 Step3.

Step4 机器人所在栅格为 g_{rnow} , 上一步栅格为

g_{pre} , 将 g_{rnext} 暂记为障碍栅格, 求出可行域 $allowed_{g_{rnow}}$, 确定 N 的取值. 如果 $allowed_{g_{rnow}} = \emptyset$, 则 N 取 NULL; 否则 N 取 EXIST. 如果 $L = \emptyset$, 则 PCTL 取值为 LN; 否则 PCTL 取值为 LE. 当 $v_d > v_r$ 时, V 取值为 DGTR; 当 $v_d < v_r$ 时, V 取值为 DLTR; 否则, V 取值为 DEQR. 根据规则库的输入, 查询障碍规则库得到输出 P . 如果 P 取值为 W, 令 $g_{enext} = g_{rnow}$; 如果 P 取值为 DR, 令 $g_{enext} = g_{rpre}, i = i - 1$; 如果 P 取值为 SP, $g_{enext} = random(allowed_{g_{rnow}}), i = i - 1$. 若 NCT 取值为 NFI, 那么将 1 加入到 L 中; 若 NCT 取值为 NPI, 那么将 2 加入到 L 中; 否则将 0 加入到 L 中. $j = j + 1$, 转 Step3.

Step5 令 $i = i + 1$, 将 g_{enext} 加入 $Epath$, 将 g_{rnext} 记为空白栅格, 令 $L = \emptyset$, 转 Step2.

Step6 算法结束, 行走的路径保存在 $Epath$ 中.

4 仿真实验

为了验证算法的效果, 作者进行了大量的计算机仿真实验. 实验环境为: WIN7, Intel Core2, 2G 内存. 编译工具: VS2008. 设任意水平或垂直的相邻栅格的距离为 1. 图 7 和图 8 中的黑色栅格表示障碍, 曲线为规划的路径.

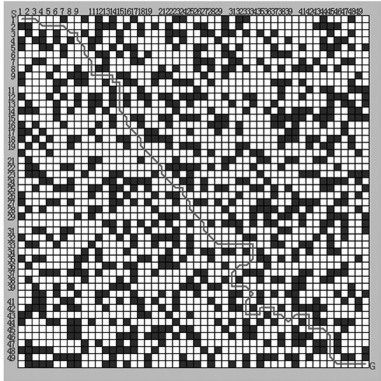


图 7 随机生成障碍物

4.1 算法效果仿真实验

(1) 路径规划算法的环境适应性和收敛速度仿真实验

为了检验基于 MAFSA 算法的环境适应性和收敛速度, 我们做了大量的仿真实验, 效果均令人非常满意. 在 50×50 的栅格环境中, 随机产生 1000 个障碍, 用本文算法规划最优路径, 共独立实验 20 次, 获得最优路径的平均规划用时为 0.034s, 图 7 显示了其中一例.

(2) 参数对收敛速度影响

大量实验表明, 算法参数 α 取 0.8 或 0.9, r 取 4 时, 收敛速度较快.

(3) 动态障碍避碰仿真实验

图 8 简要仿真了动态避碰过程. 图中的圆点分别表

示机器人探测到的两个动态障碍 DB_1 和 DB_2 . 假定所有动态障碍的速度都与机器人的速度相同. 机器人起始点和目标点分别为 1 号和 400 号栅格, 在无动态障碍时机器人用 MAFSA 算法规划出的路径如图 8(a) 中线

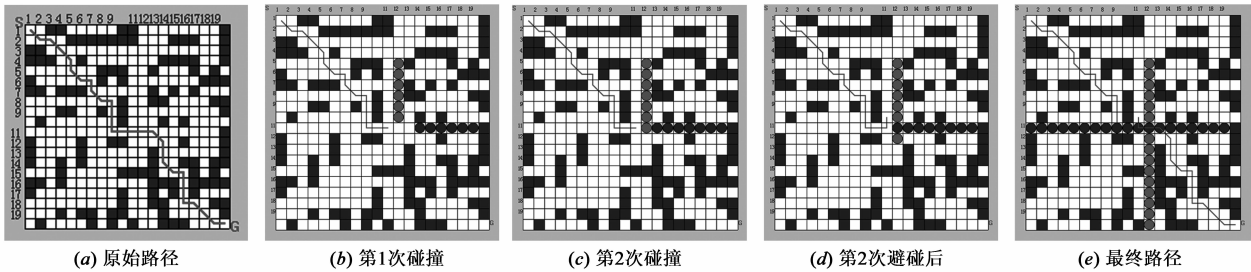


图8 多障碍碰撞过程

4.2 与其它算法实验比较

由于无法获得其它文献动态避障的比较数据, 所以在静态环境下将本文路径规划算法与某些代表性算法进行了比较. 几种方法的实验结果总结于表 3. 表中数据是对同一地形做 10 次实验的结果, 耗时则获得解的平均时间. 其中, 第三列 20×20 环境使用的是文献 [3] 中的环境, 蚁群算法的结果都引自该文献.

表 3 与几种先进算法实验结果比较

最优解	地形环境	文献[3]中 20×20 环境	参数配置
最优解 比例	本文算法	100%	$r = 4, \alpha = 0.8, m = 3, N_{\max} = 10, MAX = 6$
	随机树算法	80%	
	蚁群算法	100%	$m = 10, \alpha = 1, \beta = 4, \rho = 0.2$
获得最优解 耗时(秒)	本文算法	< 0.01	$r = 4, \alpha = 0.8, m = 3, N_{\max} = 10, MAX = 6$
	随机树算法	4.0	
	蚁群算法	< 0.01	$m = 10, \alpha = 1, \beta = 4, \rho = 0.2$

5 结语

本文提出了基于多人工鱼群的机器人路径规划算法, 采用该算法得到静态环境下的全局最优路径, 在此基础上, 分析建立了动态障碍的避碰规则. 机器人在行走时实时地预测是否会与动态物碰撞, 根据预测的碰撞类型, 从避碰规则库获得避碰策略, 从而实现动态避碰. 该方法不需要对路径进行二次规划, 大大地节省了时间. 同时, 这些规则与环境无关, 算法复杂度不会受环境障碍多少的影响. 仿真实验表明, 本文算法的环境适应能力和搜索速度都非常令人满意, 即使在障碍非常复杂的情况下也能快速地规划出全局路径, 且能安全动态避障. 由于算法与栅格大小没有太大关系, 所以在实际应用中, 可以用机器人的行走直径为单位建立栅格, 只要用机器人的传感器获得栅格中动、静态障碍信息, 即可以建立环境栅格图, 用本文算法即可实现动态避障规划, 并进行实际应用. 本文进一步的工作是将

条所示. 机器人从出发点沿该路径前进并在前进过程中不断探测动态障碍, 其动态避碰过程如图 8(b) ~ (e) 所示.

算法与实际机器人的传感器信息相结合, 将算法推向实用, 同时研究收敛性等相关理论.

参考文献

[1] 王保红. 移动机器人路径规划方法的研究与展望[J]. 沈阳工程学院学报(自然科学版), 2011, 7(4): 348 – 351.
Wang Bao-hong. Study and future development of mobile robot path planning technology[J]. Journal of Shenyang Institute of Engineering(Natural Science), 2011, 7(4): 348 – 351. (in Chinese)

[2] 毕慧敏, 董海鹰. 改进遗传算法在机器人路径规划中的应用[J]. 测控技术, 2006, 25(4): 53 – 55.
Bi Hui-min, Dong Hai-ying. Application of improved genetic algorithm in path design of robot[J]. Measurement and Control Technique, 2006, 25(4): 53 – 55. (in Chinese)

[3] 朱庆保, 张玉兰. 基于栅格法的机器人路径规划蚁群算法[J]. 机器人, 2005, 27(2): 132 – 136.
Zhu Qing-bao, Zhang Yu-lan. An ant colony algorithm based on grid method for mobile robot path planning[J]. Robot, 2005, 27(2): 132 – 136. (in Chinese)

[4] 张美玉, 黄翰, 郝志峰. 基于蚁群算法的机器人路径规划[J]. 计算机工程与应用, 2005, 25: 34 – 37.
Zhang Mei-yu, Huang Han, Hao Zhi-feng, Yang Xiao-wei. Motion planning of autonomous mobile robot based on ant colony algorithm[J]. Computer Engineering and Applications, 2005, 25: 34 – 37. (in Chinese)

[5] 柳长安, 鄢小虎, 刘春阳, 吴华. 基于改进蚁群算法的移动机器人动态路径规划方法[J]. 电子学报, 2011, 39(5): 1220 – 1224.
Liu Chang-an, Yan Xiao-hu, Liu Chun-yang, Wu Hua. Dynamic path planning for mobile robot based on improved ant colony optimization algorithm[J]. Acta Electronica Sinica, 2011, 39(5): 1220 – 1224. (in Chinese)

[6] 陈雄, 赵一路, 韩建达. 一种改进的机器人路径规划的蚁群算法[J]. 控制理论与应用, 2010, 27(6): 821 – 825.

- Chen Xiong, Zhao Yi-lu, Han Jian-da. An improved ant colony optimization algorithm for robotic path planning [J]. Control Theory & Applications, 2010, 27(6): 821 – 825. (in Chinese)
- [7] Cai Wenbin, Zhu Qingbao, Hu Jun. Path planning based on biphasic ant colony algorithm and fuzzy control in dynamic environment [A]. 2010 2nd International Conference on Intelligent Human-Machine Systems and Cybernetics [C]. Hangzhou: IEEE Press, 2010. 333 – 336.
- [8] S Ge, Y Cui. Dynamic motion planning for mobile robots using potential field method [J]. Auton Robots, 2002, 13(3): 207 – 233.
- [9] K Valavanis, T Hebert, R Kolluru N. Tsourveloudis, mobile robot navigation in 2-d dynamic environments using an electrostatic potential field [J]. Systems, Man, And Cybernetics—Part a: Systems and Humans, 2000, 30(2): 187 – 196.
- [10] 况菲, 王耀南, 张辉. 动态环境下基于改进人工势场的机器人实时路径规划仿真研究 [J]. 计算机应用, 2005, 25(10): 2415 – 2417.
Kuang Fei, Wang Yao-nan, Zhan Hui. Real time path planning of mobile robot in dynamic world based on improved artificial potential field [J]. Computer Applications, 2005, 25(10): 2415 – 2417. (in Chinese)
- [11] Lu Yin, Yixin Yin, Cheng-Jian Lin. A new potential field method for mobile robot path planning in the dynamic environments [J]. Asian Journal of Control, 2009, 11(2): 214 – 225.
- [12] M. Seder, I Petrovic. Dynamic window based approach to mobile robot motion control in the presence of moving obstacles [A]. IEEE International Conference on Robotics and Automation [C]. Roma, Italy: IEEE Press, 2007: 1986 – 1992.
- [13] 朱庆保. 动态复杂环境下的机器人路径规划蚂蚁预测算法 [J]. 计算机学报, 2005, 28(11): 1898 – 1906.
Zhu Qing-Bao. Ants predictive algorithm for path planning of robot in a complex dynamic environment [J]. Chinese Journal of Computers, 2005, 28(11): 1898 – 1906. (in Chinese)
- [14] K Kant, S W Zucker. Toward efficient trajectory planning: the path-velocity decomposition [J]. The International Journal of Robotics Research, 1986, 5(3): 72 – 89.
- [15] K. Fujimura, H. Samet. A hierarchical strategy for path planning among moving obstacles [J]. Transactions on Robotics and Automation, 1989, 5(1): 61 – 69.
- [16] A F Foka, P E Trahanias. Predictive control of robot velocity to avoid obstacles in dynamic environments [A]. Intelligent Robots and Systems [C]. Las Vegas, Nevada: IEEE Press, 2003. 370 – 375.
- [17] T. Fraichard, C Laugier. Dynamic trajectory planning, path velocity decomposition and adjacent paths [J]. Transactions on Robotics and Automation, 1993: 1592 – 1597.
- [18] 樊晓平, 李双艳. 带滚动约束轮移式机器人动态规划的研究 [J]. 控制与决策, 2005, 20(7): 786 – 793.
Fan Xiao-ping, Li Shuang-yan. Algorithm of dynamic motion planning for wheeled mobile robot with kinematic nonholonomic constraints [J]. Control and Decision, 2005, 20(7): 786 – 793. (in Chinese)
- [19] 李晓磊, 邵之江, 钱积新. 一种基于动物自治体的寻优模式: 鱼群算法 [J]. 系统工程理论与实践, 2002, 22(11): 32 – 38.
Li Xiao-lei, Shao Zhi-jiang, Qian Ji-xin. An optimizing method based on autonomous animats: fish-swarm algorithm [J]. Systems Engineering-theory & Practice, 2002, 22(11): 32 – 38. (in Chinese)
- [20] 聂黎明, 周永权. 基于人工鱼群算法的机器人路径规划 [J]. 计算机工程与应用, 2008, 44(32): 48 – 50.
Nie Li-ming, Zhou Yong-quan. Path planning of robot based on artificial fish-swarm algorithm [J]. Computer Engineering and Applications, 2008, 44(32): 48 – 50. (in Chinese)

作者简介



徐晓晴 女, 1988 年出生于江苏省宜兴市。南京师范大学计算机科学与技术学院硕士在读, 从事智能计算、机器人路径规划研究。



朱庆保 男, 1955 年出生于山东省淄博市。南京师范大学计算机科学学院教授, 博士生导师, 主要从事人工智能和机器人技术的研究。

E-mail: zhuqingbao@njnu.edu.cn