

◎工程与应用◎

改进BIT*与DWA算法的动态路径规划

汪 馨, 姚念民, 谭国真

大连理工大学 计算机科学与技术学院, 辽宁 大连 116024

摘 要:传统批通知树(batch informed trees, BIT*)算法结合了RRT*算法和A*算法的优势,但是该算法在复杂环境下无法躲避未知的动态障碍物,无法完成动态路径规划。针对该问题,提出了一种将改进的BIT*算法和改进的DWA算法相融合的算法。在传统BIT*算法的基础上对路径进行拉伸优化,提取关键转折点,减少路径长度;对传统DWA算法的距离评价函数进行改进、引入轨迹点评价函数,避免局部规划过分偏离,也减少了已知障碍物对路径的影响;将改进的BIT*算法与改进的DWA算法相融合,将提取的关键转折点作为DWA的中间目标点,弥补全局规划算法无法躲避动态障碍物的缺点以及局部规划算法全局能力低下的缺点。在动静态地图中对RRT*算法、BIT*算法、DWA算法、改进BIT*算法以及融合算法进行仿真实验,仿真结果表明:在复杂环境中,改进的BIT*算法具有更短的路径和更少的拐点;与传统的DWA算法相比,融合算法规划的路线更平滑,机器人既能实时动态避障抵达终点,又能更加贴近全局路径,保证路线全局最优。

关键词:路径规划;改进BIT*算法;DWA算法;RRT*算法

文献标志码:A **中图分类号:**TP242.6 **doi:**10.3778/j.issn.1002-8331.2112-0095

Dynamic Path Planning Based on Improved BIT* and DWA Algorithm

WANG Xin, YAO Nianmin, TAN Guozhen

College of Computer Science and Technology, Dalian University of Technology, Dalian, Liaoning 116024, China

Abstract: Traditional batch informed trees (BIT*) algorithm combines the advantages of RRT* algorithm and A* algorithm, but this algorithm cannot avoid unknown dynamic obstacles and cannot complete dynamic path planning in complex environment. To solve this problem, an algorithm combining the improved BIT* with DWA algorithm is proposed. Firstly, the stretch optimization of the path is carried out based on the BIT* algorithm to extract the key turning points and reduce the length of the path. Then, the distance evaluation function of the traditional DWA algorithm is improved and the track point evaluation function is introduced to avoid excessive deviation of local planning and reduce the influence of known obstacles on the path. Finally, the improved BIT* algorithm is fused with the improved DWA algorithm, and the extracted key turning points are taken as the intermediate target points of DWA to compensate for the shortcomings of the global planning algorithm that cannot avoid dynamic obstacles and the local planning algorithm that has low global capability. The RRT* algorithm, BIT* algorithm, DWA algorithm, improved BIT* algorithm and fusion algorithm are simulated in dynamic and static maps. The simulation results show that the improved BIT* algorithm has shorter paths and fewer inflection points in complex environments. Compared with the traditional DWA algorithm, the route planned by the fusion algorithm is smoother. The robot can not only reach the destination in real time and dynamically avoid obstacles, but also get closer to the global path to ensure the global optimal route.

Key words: path planning; improved BIT* algorithm; DWA algorithm; RRT* algorithm

移动机器人要想在特定环境下安全地完成任
务,既要做到实时感知周围环境信息、无碰撞地到达目标

点,也要使路径质量尽可能得高、使移动机器人尽可能
快地沿着路径平顺地到达目标点。常用的全局路径规

基金项目:国家自然科学基金(07020001)。

作者简介:汪馨(1998—),女,硕士研究生,研究方向为移动机器人与路径规划, E-mail: 2562786316@qq.com;姚念民(1974—),男,博士,教授,研究方向为自然语言处理、人工智能;谭国真(1960—),男,博士,教授,研究方向为群体智能、可信任AI。

收稿日期:2021-12-06 **修回日期:**2022-01-24 **文章编号:**1002-8331(2022)20-0247-08

划算法包括:RRT(rapidly exploring random tree, RRT)算法^[1-2]、A*算法^[3-4]、遗传算法^[5]、蚁群算法^[6]等。常用的局部路径规划算法有动态窗口法(dynamic window approach, DWA)^[7-8]、人工势场法^[9]等。

全局路径规划无论是基于采样的RRT算法,还是基于启发式搜索的A*算法,都可以在相对短的时间内搜索出一条从起点到终点的全局路径,但是往往无法躲避局部障碍物,对局部动态环境的适应性不强,无法满足在未知环境下避障的要求。局部路径规划DWA算法可以在线实时规划,具有良好的避障能力,但是在大规模复杂环境中如果缺少中间点引导,则可能无法得到最优路径,甚至无法达到目标点。因此,结合两者的优势,实现两种算法融合是当前研究的热点^[10-11]。Li等人^[12]扩展A*算法的搜索邻域,将改进的A*算法与DWA算法融合,虽然保证了更流畅的路径,但是前期规划时间过长。劳彩莲等人^[13]对DWA的评价函数进行改进,将A*算法与改进的DWA算法进行融合,实现实时最优路径规划。Wang等人^[14]提出RRT与DWA相结合的算法,在动态约束条件下为机器人实现了有效和平滑的路径,但是该算法仅考虑了静态的障碍物。张杰^[15]用3次B样条曲线的路径平滑方法对RRT*算法进行优化处理,同时将DWA算法与改进RRT*相结合,对全局最优目标函数进行设计,该混合算法提高了行驶速度。

利用启发式搜索的全局规划,要么牺牲了实时的效率,要么只将启发式方法应用于搜索的某些方面而降低了效率。利用抽样搜索的全局规划,要么牺牲了实时的效率,要么通过在解决方案成本以外的指标上排序搜索而降低了效率^[16]。

针对这些问题,本文引入BIT*这种将A*算法与RRT*算法相融合的快速规划路径算法^[17],该算法在超椭圆子集中执行排序搜索,可以找到更好状态的集合,即更短的全局路径^[18],但是其冗余节点会使得搜索成本增加,所以对路径进行拉伸优化,缩短路径的同时减少节点数量,避免将无价值的节点和边加入到生成树中。将全局路径点作为改进的DWA算法的中继点,增加轨迹点评价函数,将动态因素考虑在内、对距离评价函数进行细分,实现实时避障的任务。最后,通过对比仿真实验验证了融合算法的优越性。

1 改进BIT*算法的全局路径规划

1.1 问题描述

可行路径是一条能够避开障碍物且连接起点与终点的路径。最优路径是要使成本函数(例如,路径长度)最小化的可行路径。由此可知,解决一个最优规划问题需要解决潜在的可行规划问题^[19]。

定义 $X \subseteq R^n$ 表示路径规划问题的状态空间,定义 $X_{\text{obs}} \subset X$ 表示与障碍物碰撞的状态,即有障碍物的区

域。定义函数 $\sigma: [0, 1] \rightarrow X_{\text{free}}$ 表示一条无碰撞可行路径,其中 $\sigma(0) = X_{\text{start}}, \sigma(1) = X_{\text{goal}}$ 。定义 S 表示所有路径集合, S_{free} 表示所有无碰撞可行路径的集合,则最优路径规划问题可以被描述为,寻找一条可行路径 σ^* ,使得成本函数 $c: S_{\text{free}} \rightarrow R_{\geq 0}$ 最小,即:

$$c(\sigma^*) = \min_{\sigma \in S_{\text{free}}} \{c(\sigma) | \sigma(0) = X_{\text{start}}, \sigma(1) = X_{\text{goal}}\} \quad (1)$$

其中, $R_{\geq 0}$ 表示非负实数的集合。

1.2 函数定义

定义 c_{best} 表示当前最优路径长度。定义搜索树模型 $T = (V, E)$, 其中 $V \subset X_{\text{free}}$ 表示树模型中的节点, $E = \{(v_1, v_2)\}$ 表示树模型中的连线。

定义函数 $\hat{g}(x)$ 表示从起点到点 $x \in X$ 的欧拉距离,定义函数 $g_T(x)$ 表示在当前树模型 T 中,从起点到点 $x \in X$ 的路径长度。定义函数 $g(x)$ 表示当前树模型中点 $x \in X$ 的实际最优路径长度,则由上可知:

$$\forall x \in X, \hat{g}(x) \leq g(x) \leq g_T(x) \quad (2)$$

定义函数 $\hat{c}(x, y)$ 表示点 $x \in X$ 和点 $y \in X$ 构成连线的欧拉距离,定义函数 $c(x, y)$ 表示点 $x \in X$ 和点 $y \in X$ 在树模型中的连线长度,如果连线于障碍物相交,则认为其成本无穷大,即 $\hat{c}(x, y) = \infty$ 或者 $c(x, y) = \infty$ 。

$$\forall x, y \in X, \hat{c}(x, y) \leq c(x, y) \leq \infty \quad (3)$$

1.3 传统BIT*算法主体模块

由文献[16]可知,可以将BIT*算法分为:批采样点生成、连线、选边、节点扩展以及修整模块。

(1) 采样点生成。在椭圆区域中进行采样,可以先在标准圆方程中采样,再将采样点旋转平移到实际采样区域。由图1可知,长轴长度为 c_{best} ,表示当前状态下最优路径长度,短轴长度为 $\sqrt{c_{\text{best}}^2 - c_{\text{min}}^2}$,其中 c_{min} 表示起点与终点之间的欧拉距离。

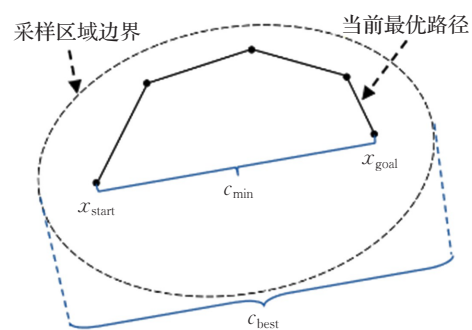


图1 生成采样点区域

Fig.1 Region to generate sample points

(2) 连线选择。队列 Q_E 存储以最优移动成本递增排列的连线 (v, x) , 队列 Q_V 存储以最优移动成本递增排列的节点 $v \in V$, 函数 $\text{BestInQueue}(Q_E)$ 可以计算当前树模型中的最优连线, 函数 $\text{BestInQueue}(Q_V)$ 可以计算当前树模型中最优节点。函数 $\text{BestValue}(Q_E)$ 可以

计算当前树模型中的连线的最优移动成本,函数 $BestValue(Q_V)$ 可以计算当前树模型中节点的最优移动成本。只有当节点 $v \in V$ 的最优移动成本小于树模型中连线的最优移动成本时,才将该节点进行连线扩展,避免在各节点上的遍历预算,减少计算规模。

(3)连线操作。通过 $BestInQueue(Q_E)$ 计算获得 Q_E 中的最优连线 (v_m, x_m) ,并将该连线从队列 Q_E 中移除,以进行连线的下一步相关操作。

(4)节点扩展。首先,从 Q_v 中弹出最优节点 v ,然后获取与节点 v 欧拉距离小于 r 的节点构成采样点 x_{rand} ,当 x_{rand} 中的节点 x ,满足公式(3)时,将连线 (v, x) 加入 Q_E 。

$$\hat{g}(v) + \hat{c}(v, x) + \hat{h}(x) < g_T(x_{goal}) \tag{4}$$

(5)修剪操作。从采样点中移除估计成本大于给定成本的节点,从已有节点中移除估计成本大于给定成本的节点,从已有连线中移除估计成本大于给定成本的连线。

1.4 改进 BIT*算法主体模块

已知 BIT* 的收敛速度取决于椭圆区域的大小,而椭圆区域的大小取决于当前最优路径长度 c_{best} ,所以重新审视当前路径,减少冗余点带来的路径成本。

改进路径点过程如图 2 所示,黑色实线是原始路径。首先连接 x_{start} 和 x_3 ,当连线与障碍物不发生碰撞时,连接 x_{start} 与 x_4, x_5, \dots, x_n ,当 x_{start} 与 x_n 的连线与障碍物发生碰撞时,以发生碰撞的障碍物转角点为中心,机器人的直径长度为半径做圆,得到点 x_{start} 与圆的切点 x'_2 ,则 x'_2 成为下一路径点,与 $x_n, x_{n+1}, \dots, x_{goal}$ 相连,如图 3(a)(b),重复这个过程直到到达目标。

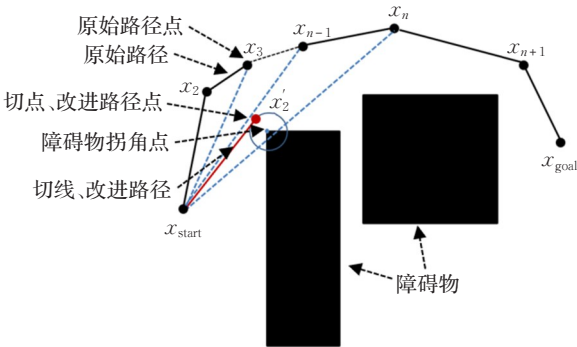


图2 更新路径点示意图

Fig.2 Updating waypoint

算法 1 为调整路径函数的伪代码,该算法对路径点进行循环处理,直到到达目标点。第 6 行的 NoCollision 函数检查两个路径点之间是否存在障碍物,第 10 行 Collision 函数找到距离 X_n 最近的发生碰撞的障碍物转角点,第 11 行 ContactPoint 函数求出切点,即找到下一路径点。

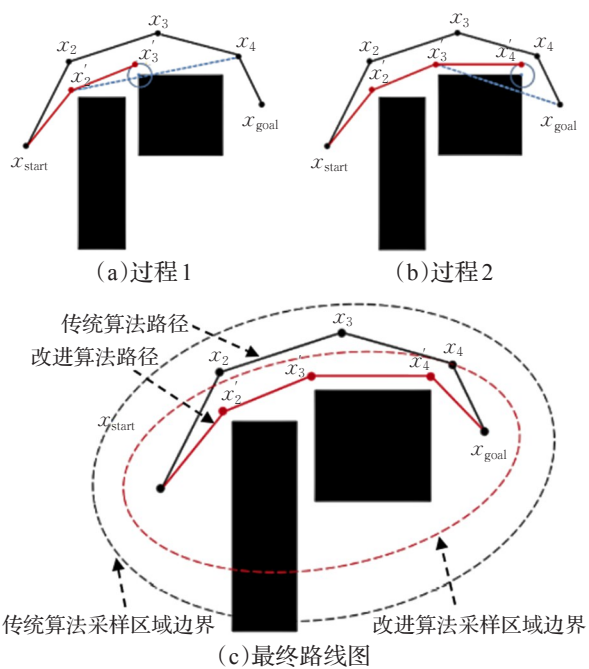


图3 改进 BIT*过程示意图

Fig.3 Process of improved BIT*

算法 1 调整路径

输入:当前最优路径

输出:调整后最优路径

```
1. function Modified(Path)
2.   i ← 0; Num ← NumOfPoints(Path)
3.   Xcurrent ← Xstart
4.   NewP ← +Xstart
5.   while i < Num - 1 do
6.     if NoCollision(Xcurrent, Path[i + 2]) then
7.       i = i + 1
8.       Continue
9.     else
10.      CollisionPoint ← Collision(Xcurrent, Path[i + 2])
11.      Xn ← ContactPoint(Xcurrent, CollisionPoint, Dia)
12.    end if
13.    ModifiedPath ← +Xn
14.    Xcurrent ← Xn
15.    i = i + 1
16.  end while
17.  ModifiedPath ← +Xgoal
18.  return ModifiedPath
19. end function
```

显然,利用三角形任意两边之和大于第三边的原理,很容易证明改进后的路径具有更少的节点和更短的路径。

在图 3 中,黑色实线是原始路径,红色实线表示改进后的新路径。可以看出,新路径的长度比原来的路径短。

在图 3(c)中,黑色虚线椭圆是由传统 BIT* 算法得

到的下一个采样区域的边界,红色虚线椭圆是改进后产生的边界。可以看出红色椭圆区域的采样点更有可能缩短当前路径,加快收敛速度。

2 改进DWA算法的局部路径规划

2.1 运动学模型

设在 t 时刻下,移动机器人的线速度为 $v(t)$,角速度为 $\omega(t)$,朝向角度为 $\theta(t)$,坐标为 (x_t, y_t) ,变化时间间隔为 Δt ,则机器人运动学模型可表达为:

$$\begin{cases} x(t) = x(t-1) + v(t)\Delta t \cos(\theta(t-1)) \\ y(t) = y(t-1) + v(t)\Delta t \sin(\theta(t-1)) \\ \theta(t) = \theta(t-1) + \omega(t)\Delta t \end{cases} \quad (5)$$

2.2 速度采样

根据环境和机器人物理特性的限制,DWA算法可以将采样速度限制在一定的空间内,其在速度空间受到的主要约束如下:

(1)线速度和角速度的约束。线速度与角速度都存在最值,其速度对 (v, ω) 满足的约束公式为:

$$V(x) = \{(v, \omega) | v \in [v_{\min}, v_{\max}], \omega \in [\omega_{\min}, \omega_{\max}]\} \quad (6)$$

其中, v_{\min} 、 v_{\max} 分别为最小、最大线速度, ω_{\min} 、 ω_{\max} 分别为最小、最大角速度。

(2)加速度和减速度的约束。对于移动机器人而言,加速度受制于电机的输出扭矩,会存在最大加速度,以及最大减速度,具体约束为:

$$V(d) = \{(v, \omega) | v = v_t + a_v \Delta t, \omega = \omega_t + a_\omega \Delta t, a_v \in [-a_{vd \max}, a_{vi \max}], a_\omega \in [-a_{\omega d \max}, a_{\omega i \max}]\} \quad (7)$$

其中, v_t 表示 t 时刻的线速度, $a_{vd \max}$ 表示线速度中最大减速度, $a_{vi \max}$ 表示线速度中最大加速度, ω_t 表示 t 时刻的角速度, $a_{\omega d \max}$ 表示角速度中最大减速度, $a_{\omega i \max}$ 表示角速度中最大加速度。

(3)制动距离约束。考虑到机器人的移动安全性,移动机器人能够在不与障碍物发生碰撞的情况下停下来。具体制动约束如下:

$$V_e = \{(v, \omega) | v \leq \sqrt{2dist(v_t, \omega_t)a_{vd \max}}, \omega \leq \sqrt{2dist(v_t, \omega_t)a_{\omega d \max}}\} \quad (8)$$

其中, $dist(v_t, \omega_t)$ 表示在速度 (v_t, ω_t) 时,移动机器人与障碍物的最近距离。

最终移动机器人的速度需要对以上三个约束空间取交集: $V = V_s \cap V_d \cap V_e$ 。

针对一系列满足要求的速度采样点,将其带入评价函数,选取最优轨迹点作为下一时刻的运动轨迹。

2.3 DWA算法评价函数

传统DWA算法的评价函数包含方位评价、速度评价和距离评价三个方面,具体公式如下:

$$G(v, \omega) = \alpha Head(v, \omega) + \beta Vel(v, \omega) + \gamma Dist(v, \omega) \quad (9)$$

其中, $Head(v, \omega)$ 为方位角评价函数,用来计算移动机器人在有限速度下到达模拟轨迹终点方向与目标方向之间的角度偏差; $Vel(v, \omega)$ 为速度评价函数,用于计算当前预估的运行速度; $Dist(v, \omega)$ 为距离评价函数,用于计算机器人预测点位置与最近障碍物之间的距离。 α 、 β 、 γ 为每一项的加权系数,评价函数 $G(v, \omega)$ 表示机器人在保持与障碍物的距离同时以较大的速度朝着目的地前进。

2.4 DWA算法评价函数优化

传统DWA算法容易陷入局部最优,针对该问题,用改进的BIT*算法得到的全局路径来引导DWA算法进行局部规划。为更好接近全局规划路径,在评价函数中添加轨迹点评价函数 $Point(v, \omega)$,用于计算DWA算法在 t 时刻预测的轨迹点 (x_t, y_t) 与BIT*求得的全局路径点 (x_g, y_g) 之间的最小距离。

$$Point(v, \omega) = \min[(x_t - x_g)^2 + (y_t - y_g)^2] \quad (10)$$

针对移动障碍物问题,对传统DWA算法中的距离评价函数进行改进,对距离评价函数进行细分,划分为 $Dist_S(v, \omega)$ 和 $Dist_D(v, \omega)$ 这两个距离评价项,分别用于评价移动机器人与静态障碍物和动态障碍物的距离,对动、静态障碍物分别设置不同的安全距离。

设静态障碍物坐标为 (x_s, y_s) , D_s 为 t 时刻的轨迹点距离静态障碍物最近的距离。 R 为移动机器人的底盘半径,不规则的移动机器人以最长底盘半径为 R ,在此基础上加入 $0.2R$ 来保证与静态障碍物之间的安全距离。 $Dist_S(v, \omega)$ 约束公式如下:

$$D_s = \min[(x_t - x_s)^2 + (y_t - y_s)^2] \quad (11)$$

$$Dist_S(v, \omega) = \begin{cases} \frac{1}{D_s}, D_s \geq 1.2R \\ 0, D_s < 1.2R \end{cases} \quad (12)$$

设动态障碍物坐标为 (x_i, y_i) , D_d 为 t 时刻的轨迹点距离动态障碍物最近的距离。 V_t 表示移动机器人在 t 时刻的速度, S 为机器人在时间间隔 Δt 内运动的路径,由于可能会存在变加速问题,所以将其扩展为1.2倍,来保证与动态障碍物之间的安全距离。 $Dist_D(v, \omega)$ 约束公式如下:

$$D_d = \min[(x_t - x_i)^2 + (y_t - y_i)^2] \quad (13)$$

$$Dist_D(v, \omega) = \begin{cases} \frac{1}{D_s}, D_s \geq 1.2S \\ 0, D_s < 1.2S \end{cases} \quad (14)$$

$$S = V_t \times \Delta t \quad (15)$$

则,最终的DWA评价函数为:

$$G(v, \omega) = \alpha Head(v, \omega) + \beta Vel(v, \omega) + \gamma Dist_S(v, \omega) + \sigma Dist_D(v, \omega) + \varphi Point(v, \omega) \quad (16)$$

表示机器人在全局规划的路径基础上保持与障碍物的距离,同时按照以较大的速度朝着目的地前进。

3 融合算法

改进的 BIT*算法在静态环境中能很好地完成全局路径规划,但是当出现未知障碍物时,就无法及时避障,其局部路径规划能力较弱。DWA 算法所生成的路径为局部最优并非全局最优,如果缺少全局指引,只有目的地一个方向指引,会容易陷入局部最优,导致全局路径变大,甚至全局路径规划失败。

针对上述问题,首先用改进的 BIT*算法在已知障碍物的地图中规划全局路径,即在移动前,找到全局最优路径,然后将提取的关键转折点作为移动机器人的中继点。移动过程中,如果传感器感知到未知障碍物阻挡了算法所规划的路径,则用改进的 DWA 算法根据更新的障碍物信息实时规划新场景下的避障路径。融合算法添加了轨迹点评价函数,避免局部规划过分偏离,同时对未知障碍物分配更高的比重,减少已知障碍物对路径的影响,融合算法流程图如图 4 所示。

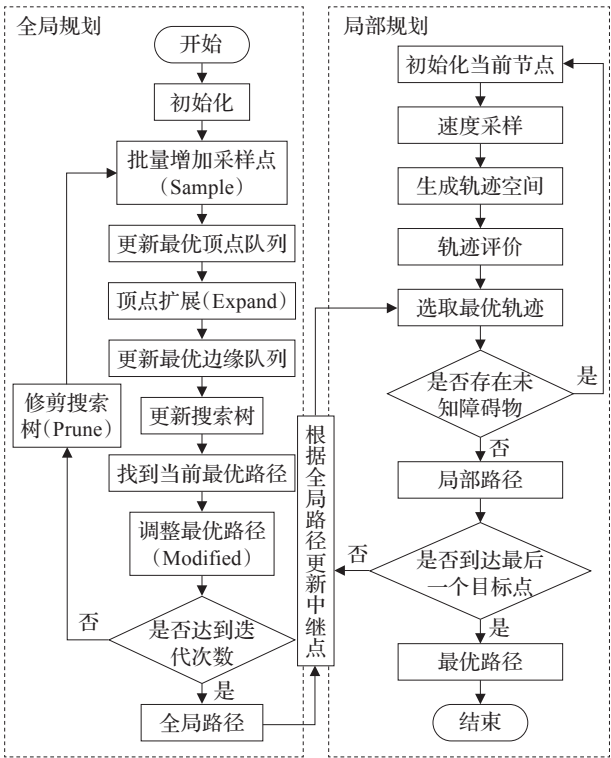


图 4 融合算法流程图
Fig.4 Flow chart of fusion algorithm

4 仿真实验

4.1 仿真环境描述

为了验证改进算法的有效性,采用 python 开发的开源工具 PathEnv。仿真环境为二维静态地图,设置为 30×30 的正方形,其中黑色格子代表障碍物,白色格子

代表可移动区域,蓝色格子代表起点,红色格子代表终点。动态环境中,黄色格子代表未知静态障碍物,紫色格子代表未知动态障碍物,紫色虚线代表动态障碍物的移动路径。

地图 1 为简单地图,起点(2,29),目标点(20,4),地图 2 包含有窄通道,起点(6,20),终点为(23,10),地图 3 比较复杂,包含有较多的窄通道,此环境下的实验能体现算法优劣,设置起点(2,29),目标点(20,4)。在三组地图中分别对 RRT*算法、BIT*算法、改进的 BIT*算法以及融合算法进行仿真实验,对四种算法求得的路径进行统计与比较。在动态环境下对改进的 BIT*算法、传统 DWA 算法以及融合算法进行仿真实验,统计数据进行分析比较。

相关运动学参数设置:初始方位角 $-\pi/2$,最大线速度 1.5 m/s,最小线速度 0,即设置为不可倒退,最大角速度 40 rad/s,最大线加速度为 0.2 m/s,最大角加速度为 50 rad/s,线速度分辨率为 0.01 m/s,角速度分辨率为 0.1 rad/s,采样周期 0.1 s,机器人半径设置为 0.3 m。

4.2 静态环境仿真结果

对 RRT*算法、BIT*算法、改进的 BIT*算法以及融合算法在三组静态地图中对进行仿真实验,其仿真结果如图 5、图 6、图 7。对上述三组二维静态地图仿真环境进行数据统计如表 1。

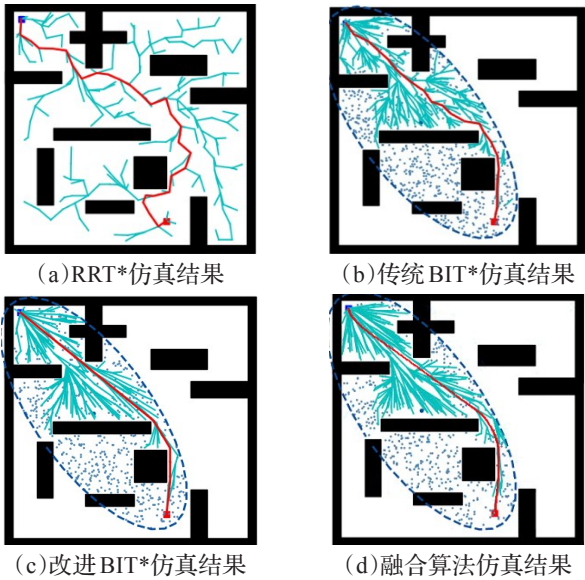


图 5 地图 1 仿真结果
Fig.5 Simulation results of Map1

图中蓝绿色线段表示路径扩展过程,即搜索树的扩展过程,蓝色颗粒表示随机生成的采样点,蓝色虚线椭圆表示更新的采样区域边界,红色实线表示算法最终规划的路径。

为进一步验证融合算法在大规模环境下的有效性,格外增加地图 4,大小为 60×60,起点(0,60),目标点

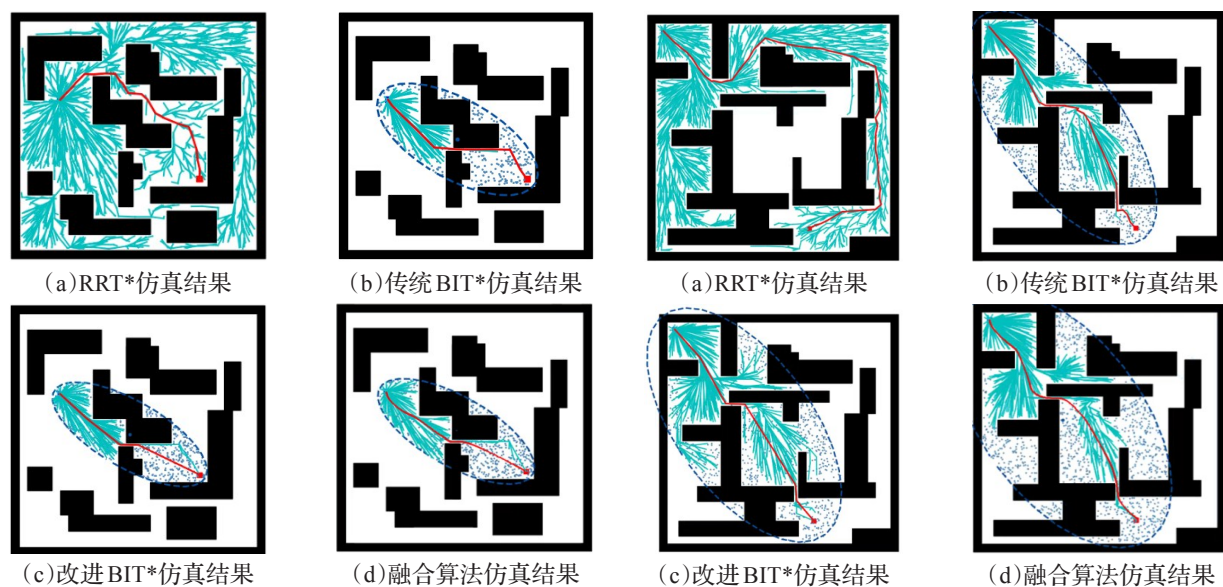


图6 地图2仿真结果

Fig.6 Simulation results of Map2

图7 地图3仿真结果

Fig.7 Simulation results of Map3

表1 三组静态地图仿真结果

Table 1 Three groups of map simulation results

地图编号	路径算法	迭代次数	转折点点个数	转折点优化比例/%	路径长度/m	路径长度优化比例/%
地图1	RRT*算法	500	29	—	41.13	—
	BIT*算法	500	23	20.69	34.11	17.07
	改进BIT*算法	500	5	82.76	33.35	18.92
	融合算法	500	—	—	34.10	17.09
地图2	RRT*算法	2 000	24	—	26.89	—
	BIT*算法	1 000	12	50.00	21.81	18.89
	改进BIT*算法	1 000	4	83.33	20.19	24.92
	融合算法	1 000	—	—	21.22	20.97
地图3	RRT*算法	8 000	57	—	60.29	—
	BIT*算法	3 000	20	64.91	36.99	38.65
	改进BIT*算法	3 000	7	87.72	32.16	46.66
	融合算法	3 000	—	—	34.73	42.40
地图4	A*算法	—	8	—	90.81	—
	蚁群算法	800	13	—	93.39	—
	改进BIT*算法	1 000	4	50.00	86.95	4.25
	融合算法	1 000	—	—	87.54	3.60

(60,0),对A*算法、蚁群算法、改进的BIT*算法以及融合算法进行仿真实验,其仿真结果如图8,数据统计如表1。

其中,图8(a)中灰色点区域表示已搜索区域,红色实线表示算法最终规划的路径。

由表1可知,改进的BIT*算法提供了质量更高的全局规划路径,其优化主要体现在转折点点个数以及路径长度上。在前三个地图中,改进BIT*算法的路径转折点较RRT*算法平均优化84.60%,比传统BIT*算法平均高39.40%,改进BIT*算法的路径长度较RRT*算法平均优化30.17%,比传统BIT*算法平均高5.30%。在地图4中,改进BIT*算法提供了更少的路径转折点和更短的全局路径,在大规模地图中也有可行性。

在简单地图1和大规模地图4中,路径长度优化幅度较小,但在拥有狭窄通道的地图2中,路径长度优化24.92%,尤其是包含更多狭窄通道的地图3,其轨迹优化比例为46.66%,可知改进的BIT*算法可以在更少的迭代次数下找到更优的路径,尤其是在拥有狭窄通道的复杂环境下。融合算法的路径长度平均优化率为21.01%,可以看出在使用改进BIT*算法和改进DWA算法的融合,在保证局部连贯性的同时也贴近了全局规划,满足移动机器人静态环境下的路径规划需求。

4.3 动态环境仿真结果

在添加了未知动静态障碍物的地图3中分别对改进BIT*算法、改进动态窗口法以及融合算法进行仿真实验,其仿真结果如图9所示。

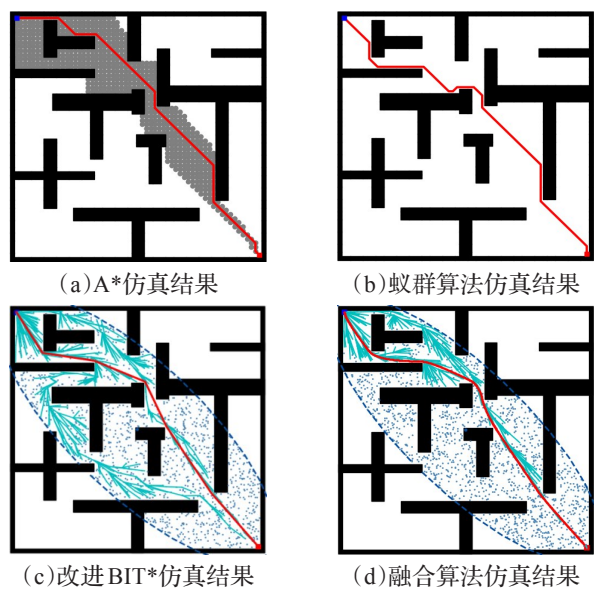


图8 地图4仿真结果

Fig.8 Simulation results of Map4

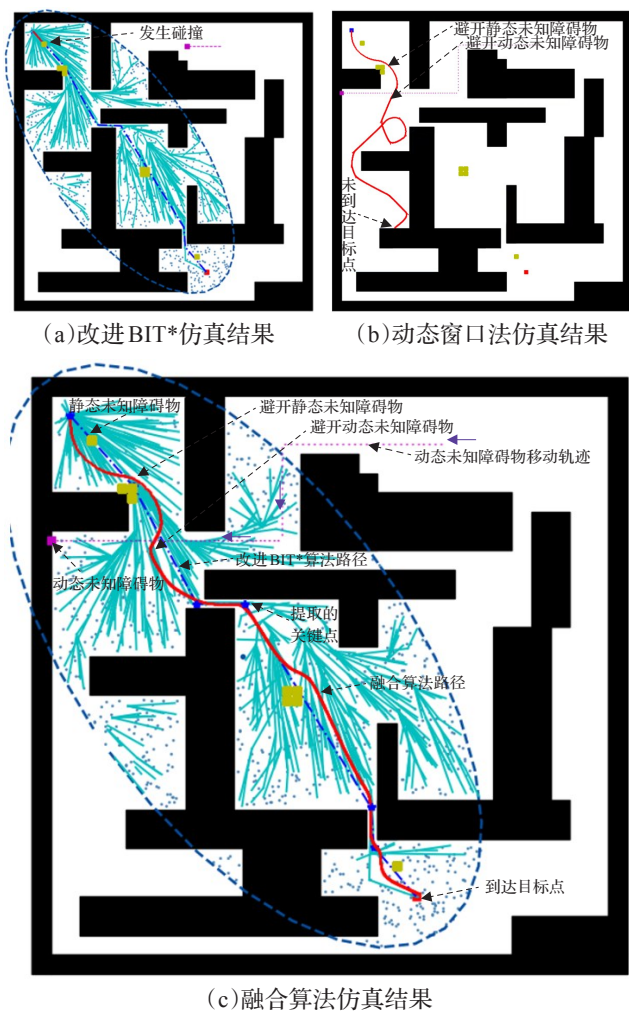


图9 动态地图仿真结果

Fig.9 Dynamic simulation results of fusion algorithm

地图中,黄色格子代表未知静态障碍物,紫色格子代表未知动态障碍物,紫色虚线代表动态障碍物的移动路径,紫色箭头代表动态障碍物的移动方向。蓝色虚线

表示改进 BIT*算法规划的全局路径,蓝色星点为提取的关键路径点,红色实线表示实际运动路径。

由仿真结果图 9(a)可知,改进的 BIT*算法虽然提供了转折点更少、路径更短的全局规划路线,但是在机器人移动后,如果在已规划路径上出现未知障碍物,则会与障碍物距离为负,即发生碰撞,那碰撞之后的路径则无效,所以,改进的 BIT*算法适合在机器人移动前对其进行全局规划,但在移动过程中则无法解决未知障碍物问题。由仿真结果图 9(b)可知,改进的 DWA 算法,可以很好地避开未知动静态障碍物,但是在没有全局规划的指引下,会陷入局部最优,无法到达目标点。

由仿真结果图 9(c)可知,当碰到未知静态障碍物时,融合算法可以在贴近全局规划的同时进行有效避障;当未知障碍物阻挡过程性目标点时,机器人距离障碍物一定距离时会舍弃该点,更新下一目标点;当碰到未知动态障碍物时,融合算法可以进行有效避障。

总之,在改进的 BIT*算法提供的全局规划路径下,融合算法满足机器人既要避开障碍物又要全局路径最优的需求。

5 结束语

本文针对移动机器人在复杂环境下的动态路径规划问题做了以下研究:

(1)全局路径规划采用改进的 BIT*算法,在传统 BIT*算法的基础上对路径进行拉伸优化,对比传统 RRT*算法和 BIT*算法,改进的 BIT*算法可以在更短的迭代时间内找到更优的无碰撞路径,且在大规模环境中也有可行性。

(2)局部路径规划采用改进的 DWA 算法,通过对距离函数进行改进的同时引入轨迹点评价函数,避免了局部规划过分偏离,也减少了已知障碍物对路径的影响,有效改善传统 DWA 算法局部最优的问题。

(3)融合算法将全局规划的关键转折点当作局部规划的中间过程点,将全局规划能力与实时避障能力相结合,保证路径质量的同时避开未知动静态障碍物,满足机器人路径规划需求,实现了更为有效的实时避障的全局路径规划。

本文主要在传统算法基础上进行改进,并与传统算法进行对比,但目前已提出将路径规划与强化学习等知识相结合的研究^[20],所以之后会将本文提出的算法与更前沿的知识相结合,在更大、更动态的仿真环境中进行实验,得到更灵活的路径规划算法。

参考文献:

[1] 李金良,舒翰儒,刘德建,等.基于改进 RRT 路径规划算法[J].组合机床与自动化加工技术,2021(2):22-24.
LI J L,SHU H R,LIU D J,et al.Path planning algorithm

- based on improved RRT[J].Modular Machine Tool & Automatic Manufacturing Technique, 2021(2):22-24.
- [2] 龙建全, 梁艳阳. 多路口环境下 RRT 的最优路径规划[J]. 计算机工程与应用, 2020, 56(19):273-278.
LONG J Q, LIANG Y Y. Optimal path planning of RRT in multi-intersection environment[J]. Computer Engineering and Applications, 2020, 56(19):273-278.
- [3] 吴鹏, 桑成军, 陆忠华, 等. 基于改进 A* 算法的移动机器人路径规划研究[J]. 计算机工程与应用, 2019, 55(21):227-233.
WU P, SANG C J, LU Z H, et al. Research on mobile robot path planning based on improved A* algorithm[J]. Computer Engineering and Applications, 2019, 55(21):227-233.
- [4] 王晓燕, 吕金豆. 基于改进 A* 势场法的机器人动态路径规划研究[J]. 制造业自动化, 2021, 43(1):83-87.
WANG X Y, LV J D. Research on robot dynamic path planning based on improved A* and artificial potential field method[J]. Manufacturing Automation, 2021, 43(1):83-87.
- [5] 徐力, 刘云华, 王启富. 自适应遗传算法在机器人路径规划的应用[J]. 计算机工程与应用, 2020, 56(18):36-41.
XU L, LIU Y H, WANG Q F. Application of adaptive genetic algorithm in robot path planning[J]. Computer Engineering and Applications, 2020, 56(18):36-41.
- [6] 孟冠军, 陈信华, 陶细佩, 等. 基于混合蚁群算法的 AGV 路径规划[J]. 组合机床与自动化加工技术, 2021(1):70-73.
MENG G J, CHEN X H, TAO X P, et al. AGV path planning based on hybrid ant colony algorithm[J]. Modular Machine Tool and Automatic Processing Technology, 2021(1):70-73.
- [7] 王永雄, 田永永, 李璇, 等. 穿越稠密障碍物的自适应动态窗口法[J]. 控制与决策, 2019, 34(5):927-936.
WANG Y X, TIAN Y Y, LI X, et al. Self-adaptive dynamic window approach in dense obstacles[J]. Control and Decision, 2019, 34(5):927-936.
- [8] 卞永明, 季鹏成, 周怡和, 等. 基于改进型 DWA 的移动机器人避障路径规划[J]. 中国工程机械学报, 2021, 19(1):44-49.
BIAN Y M, JI P C, ZHOU Y H, et al. Obstacle avoidance path planning of mobile robot based on improved DWA[J]. Chinese Journal of Construction Machinery, 2021, 19(1):44-49.
- [9] 李志锟, 赵倩楠. 融合人工势场蚁群算法的移动机器人路径规划[J/OL]. 电光与控制:1-8[2021-12-09]. <http://kns.cnki.net/kcms/detail/41.1227.tn.20211103.1002.002.html>.
LI Z K, ZHAO Q N. Mobile robot path planning combined with artificial potential field ant colony algorithm [J/OL]. Electronics Optics & Control:1-8[2021-12-09]. <http://kns.cnki.net/kcms/detail/41.1227.tn.20211103.1002.002.html>.
- [10] 高佳佳. 基于全局地图的移动机器人路径规划研究[D]. 西安:西安工业大学, 2019.
GAO J J. Research on path planning of mobile robot based on global map[D]. Xi'an:Xi'an University of Technology, 2019.
- [11] 刘建娟, 薛礼啟, 张会娟, 等. 融合改进 A* 与 DWA 算法的机器人动态路径规划[J]. 计算机工程与应用, 2021, 57(15):73-81.
LIU J J, XUE L Q, ZHANG H J, et al. Robot dynamic path planning based on improved A* and DWA algorithm[J]. Computer Engineering and Applications, 2021, 57(15):73-81.
- [12] LI X, HU X, WANG Z, et al. Path planning based on combination of improved A-STAR algorithm and DWA algorithm[C]//2020 2nd International Conference on Artificial Intelligence and Advanced Manufacture(AIAM), 2020:99-103.
- [13] 劳彩莲, 李鹏, 冯宇. 基于改进 A* 与 DWA 算法融合的温室机器人路径规划[J]. 农业机械学报, 2021, 52(1):14-22.
LAO C L, LI P, FENG Y. Path planning of greenhouse robot based on fusion of improved A* and DWA algorithm[J]. Transactions of the Chinese Society for Agricultural Machinery, 2021, 52(1):14-22.
- [14] WANG J, WU S, LI H, et al. Path planning combining improved rapidly-exploring random trees with dynamic window approach in ROS[C]//2018 13th IEEE Conference on Industrial Electronics and Applications(ICIEA), 2018:1296-1301.
- [15] 张杰. 基于 ROS 的移动机器人路径规划研究[D]. 贵阳:贵州大学, 2020.
ZHANG J. Research on path planning of mobile robot based on ROS[D]. Guiyang:Guizhou University, 2020.
- [16] GAMMELL J D, SRINIVASA S, BARFOOT T D. Batch informed trees (BIT*): sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs[C]//2015 IEEE International Conference on Robotics and Automation(ICRA), 2015:3067-3074.
- [17] GAMMELL J D, BARFOOT T D, SRINIVASA S. Batch informed trees (BIT*): informed asymptotically optimal anytime search[J]. The International Journal of Robotics Research, 2020, 39(5):543-567.
- [18] XU P, WANG N, DAI S L. Motion planning for mobile robot with modified BIT* and MPC[J]. Applied Sciences, 2021, 11(1):426.
- [19] KARAMAN S, FRAZZOLI E. Sampling-based algorithms for optimal motion planning[J]. The International Journal of Robotics Research, 2011, 30(7):846-894.
- [20] FRANCIS A. Long-range indoor navigation with PRM-RL[J]. IEEE Transactions on Robotics, 2020, 36(4):1115-1134.