

DOI:10.19651/j.cnki.emt.2518889

基于 A* 与 DWA 算法的果园导航机器人研究*

王晶¹ 高亚鹏¹ 李海芳^{1,2}

(1. 太原理工大学计算机科学与技术学院 太原 030024; 2. 山西工商学院计算机信息工程学院 太原 030006)

摘要: 路径规划算法是移动机器人实现导航技术的关键。针对果园环境中传统路径规划算法在节点遍历、搜索效率、路径平滑性及避障能力等方面的不足,本文提出了一种改进 A* 算法与 DWA 算法结合的路径规划方法,有效提升规划路径的全局最优性和实时避障能力。首先采用三维点云数据构建二维栅格地图,为导航机器人提供精确的环境模型。通过矩形扩展搜索策略优化传统 A* 算法的邻域搜索方式,结合关键路径节点选取方法和基于动态相切圆的路径平滑技术,生成符合果园作业需求的全局路径。优化传统 DWA 算法的评价函数,引入角度偏差、路径偏离及障碍物信息等因素,提高避障决策的全局导向性和局部响应能力。最后,构建改进 A* 算法与改进 DWA 算法的融合架构,实现全局导航与局部避障的协同工作。仿真结果表明,本文改进算法在路径规划效率、路径质量及避障能力等方面具有显著优势,满足了果园环境下移动机器人路径规划的实际需求,有效支撑了果园智能化管理的需求。

关键词: 果园导航; A* 算法; DWA 算法; 融合方法; 路径规划

中图分类号: TN965.8 **文献标识码:** A **国家标准学科分类代码:** 520.60

Research on orchard navigation robot based on A* and DWA algorithms

Wang Jing¹ Gao Yapeng¹ Li Haifang^{1,2}

(1. College of Computer Science and Technology, Taiyuan University of Technology, Taiyuan 030024, China;

2. College of Computer and Information Engineering, Shanxi Technology and Business College, Taiyuan 030006, China)

Abstract: Path planning algorithms are key to enabling mobile robot navigation. In view of the deficiencies of traditional path planning algorithms in orchard environments in terms of node traversal, search efficiency, path smoothness and obstacle avoidance ability, this paper proposes an improved path planning method combining the A* algorithm and the DWA algorithm, which effectively improves the global optimality and real-time obstacle avoidance of the planned path. Firstly, a two-dimensional raster map is constructed using three-dimensional point cloud data to provide an accurate environmental model for the navigation robot. The neighborhood search method of the traditional A* algorithm is optimized through the rectangular expansion search strategy. Combined with the selection method of critical path nodes and the path smoothing technology based on dynamic tangent circles, a global path that meets the operation requirements of orchards is generated. Optimize the evaluation function of the traditional DWA algorithm, introduce factors such as heading angle, path deviation and obstacle information, and improve the global orientation and local response ability of obstacle avoidance decision-making. Finally, A fusion architecture of the improved A* algorithm and the improved DWA algorithm is constructed to enable coordinated global navigation and local avoidance. The simulation results show that the improved algorithm in this paper has significant advantages in terms of path planning efficiency, path quality and obstacle avoidance, meeting the actual needs of mobile robot path planning in the orchard environment and supporting intelligent orchard management.

Keywords: orchard navigation; A* algorithm; DWA algorithm; fusion method; path planning

0 引言

随着农业现代化和智能化进程的深入推进,果园管理

逐步向高效化和精细化方向发展。果园环境因其地形复杂、动态性强和非结构化等特点,成为农业机器人应用的重点领域,其中果园移动机器人的核心任务是实现高效的自

收稿日期:2025-05-22

* 基金项目:山西省重点研发计划项目(2022ZDYF128)、山西省基础研究计划项目(202303021212066)资助

主导航。导航系统在地图环境的实现依赖于路径规划技术^[1],即查找连接起点与终点的最优无障碍路径。

常用到的全局路径规划算法有遗传算法^[2]、蚁群算法^[3]、Dijkstra算法^[4]和A*算法^[5]等。A*算法通过引入启发式代价函数显著提高了搜索效率,尤其在栅格地图环境中表现优异,因此广泛应用于移动机器人全局路径规划任务中。针对传统A*算法存在的搜索范围大、路径冗余和平滑性不足等问题,研究者从不同的角度对算法进行了改进。Lin等^[6]提出一种改进的A*算法用于无人车路径规划,该算法设置冗余安全空间以过滤狭窄道路和避免碰撞,结合预判规划策略和冗余拐点剔除策略以优化行驶路径,在轨迹优化中采用安全走廊方法节省计算机时间,但仍是面向开阔环境的单次全局规划。Yin等^[7]对传统A*算法进行优化,引入新评价函数和双向搜索策略,有效减少遍历节点数量,消除了无效路径节点,应用B样条插值法平滑路径,去除不必要的拐点,有效优化了室内结构化环境下机器人的运动轨迹,但难以适应动态性强的果园环境。Li等^[8]提出基于双向交替搜索策略的A*算法,使用指数衰减对启发式函数进行加权,引入关键路径节点过滤策略减少路径中的冗余节点,最后通过贝塞尔曲线得到具有连续曲率的平滑路径,但其使用的路径平滑策略依赖于固定控制点,在果园密集环境下可能灵活性不足。

随着环境复杂性和动态性的增加,单纯依赖全局路径规划的方式往往难以应对实际应用中的挑战。因此,研究者逐渐倾向于将全局路径规划与局部路径规划相结合,通过整合全局路径提供的宏观导航信息和局部路径规划的实时避障能力,提升系统的整体性能和适应性。动态窗口法(dynamic window approach, DWA)是当前被广泛采用的局部避障技术,具有响应时间短和计算效率高的特点^[9]。因此,A*算法与DWA算法的融合逐步成为了研究热点,并在一般环境中展现出良好效果,然而在应对果园这一特定复杂环境时,仍存在明显不足。曾宪阳等^[10]对A*算法进行改进,通过动态加权处理启发函数,并利用Floyd算法去除路径中的冗余点,同时引入安全距离机制以防碰撞,进一步结合DWA算法进行局部避障,但关键节点选取时未能充分结合机器人的运动特性和环境特征,从而在一定程度上影响了规划的适应性。袁新亚等^[11]在改进的A*算法中引入基于障碍率的权重因子和双向优化策略,通过融入两种新的障碍物评价函数对DWA算法进行改进,并通过调整权重系数有效避免局部最优解问题,最后将改进的DWA算法与A*算法结合,使机器人搜索到最优路径并生成平滑轨迹,但对于果园中连续弓形通道的复杂地形,基于固定权重或优化策略生成的路径仍不够自然。Liao等^[12]设计融合自适应A*算法与改进DWA算法的路径规划策略,在启发式搜索过程中引入权重调整机制,并引入Douglas-Pucker算法简化路径节点,加入轨迹点预估函数到DWA算法的评价函数中,并使用B样条插值法平滑优

化整条路径,但在果园场景下的避障决策未充分考虑与全局规划路径的贴合程度。

基于已有的相关研究内容,针对现有方法在果园场景下存在的搜索冗余问题、节点选取缺乏动态适应性、固定控制点平滑方法在果园狭窄通道中缺乏灵活性以及避障时全局导向性不足等问题,本文进一步对A*算法与DWA算法进行了优化,以实现果园环境下移动机器人全局导航与局部避障的高效协同工作,主要包括:

1)引入矩形扩展方式,将传统A*算法的八邻域搜索改进为矩形扩展搜索方式,只访问矩形的边界节点,解决了现有方法在果园复杂地形中存在的搜索冗余问题,加快了算法的搜索速度。

2)设计关键路径节点选取方案,基于搜索节点的相对位置关系匹配机器人特性做出动态调整,克服了现有节点选取策略缺乏动态适应性的不足,增强了路径规划的安全性和适应性。

3)采用基于动态相切圆的路径平滑策略,通过动态调整相切圆来对局部路径进行精细化地平滑处理,改善了移动机器人的运动控制性能。

4)在DWA算法评价函数中引入全局路径偏离距离评价项,改进方位角和障碍物评价函数,增强了全局导向性,融合改进后的A*和DWA算法,提升了果园场景下规划的安全性和鲁棒性。

1 环境建模与算法基础

1.1 环境建模

在果园导航机器人的路径规划任务中,首先需要建立环境模型,以描述果园的结构与障碍分布。传统A*算法是一种经典的图搜索算法,其路径规划时使用的地图通常以栅格地图的形式来表达,这种形式简洁直观,易于维护,且当前运用较为广泛。在现代机器人技术中,三维激光雷达能够捕获环境的详细深度信息,生成高分辨率的三维点云数据,提高地图的准确性和丰富性。使用三维激光雷达可以构建果园环境的三维点云图,虽然可以反映果园的三维空间结构信息,但是不能直接用于路径规划。因此,本文提出通过果园三维点云数据构建二维栅格地图的方案。

三维激光雷达采集的点云数据通常以三维坐标 (x, y, z) 的形式存储,机器人在地面上进行路径规划,因此基于点云数据得到的环境地图也应该在地面这个平面上。三维点云投影到二维平面的整体过程如图1所示,图中黑色圆点代表采集的点云数据,白色方格代表无障碍物,黑色方格代表有障碍物。具体步骤如下:

1)提取所需的点云数据。为确保生成的地图只包含机器人能够接触到的环境特征,通过地平面拟合法去除地面点云数据,滤除高于机器人作业高度的点云数据,保留导航相关的有效数据。

2)三维点云投影到地面。忽略三维点云数据的高度信

息,即只保留数据的 x 坐标和 y 坐标,进而将三维坐标数据映射为二维坐标数据。

3)平面划分为栅格单元。确定投影范围,在平面上生成固定大小的栅格单元,并将投影点映射到相应的栅格单元中,统计各单元内的点数。

4)去除孤立噪声投影点。遍历所有栅格单元,对投影

点数量少于 3 且周围八邻域内无点的孤立单元,视为噪声并清除单元内的投影点。

5)确定栅格地图的状态。将点数不为 0 的单元状态记为存在障碍,用黑色表示,否则将栅格单元状态记为不存在障碍,用白色表示。

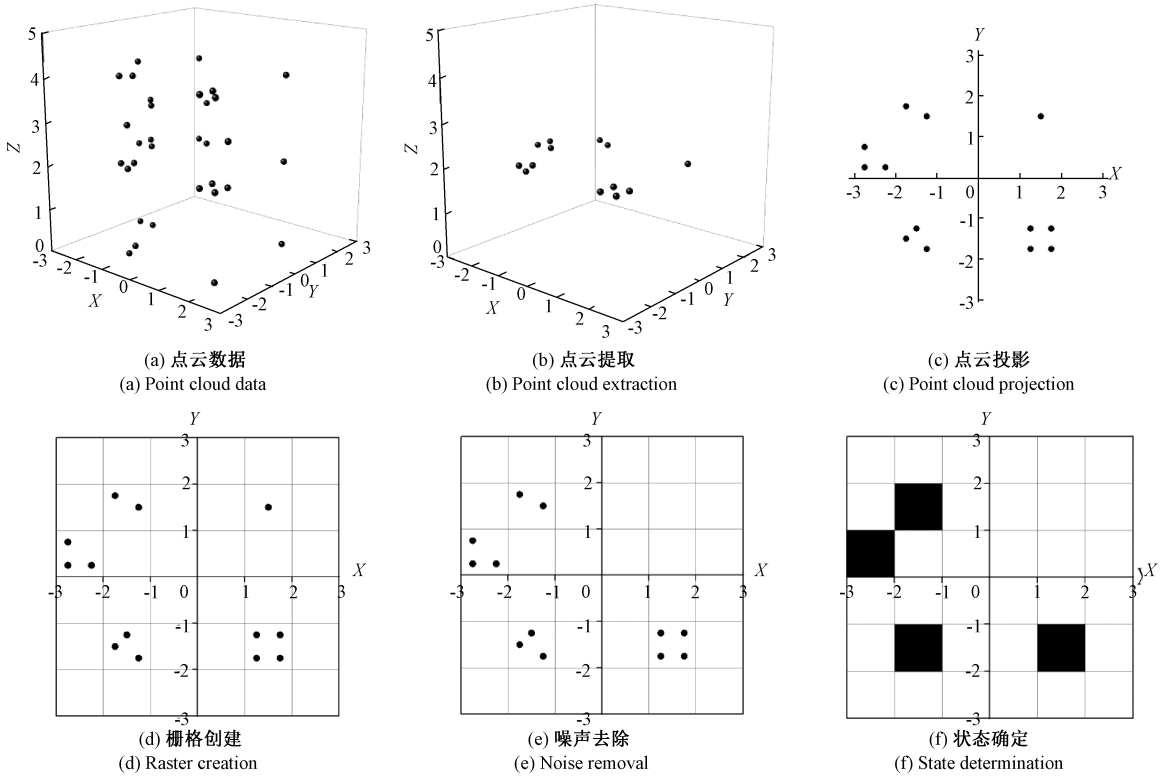


图 1 创建栅格地图的过程示意图

Fig. 1 Schematic diagram of the grid map creation process

随着农业技术的发展,果园中更倾向于规范化的果树种植,以提高果树的生长效率、管理便利性和果实产量。得到规范化果园的栅格地图后,为解决地图中果树离散性问题,还需要基于果树尺寸与所在行的长度进一步做拟合的处理。考虑到折线覆盖策略下的路径呈“弓”字形,对树行进行膨胀处理,并延续树行末端到墙壁,构建“弓”字形的作业走廊。为保障机器人的安全,采用在障碍物周围生成一层膨胀区域的优化方法,确保生成的行进路径与周围障碍物保留合理的缓冲距离。最终的果园地图如图 2 所示,白色表示可通过区域,黑色表示果树或围墙障碍物,深灰色表示膨胀区,黑色和深灰色均不允许通行,左下绿格表示起始节点,右上红格表示目标节点。

1.2 传统 A* 算法

A* 算法是一种带有启发机制的图搜索算法,融合了广度优先搜索(breadth first search, BFS)算法与 Dijkstra 算法的优势,依赖代价函数指导节点扩展方向,是常用的完成机器人从起始节点到目标节点的静态全局路径规划算法。传统 A* 算法的扩展方式为八邻域策略,如图 3 所

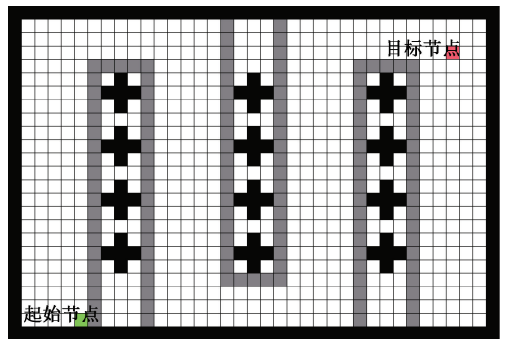


图 2 模拟果园环境的栅格地图

Fig. 2 Grid map of the orchard environment

示,中心星形表示当前节点,矩形表示当前节点向周围八个方向的搜索节点,圆形表示未搜索节点。算法从起始节点向周围的八个相邻节点扩展,将代价函数值最低的相邻节点作为新节点,继续在新节点的八个相邻方向上扩展,重复进行此过程,直到抵达目标节点并生成最优路径。

A* 算法的关键在于代价函数,函数表达式为:

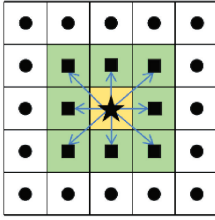


图 3 八邻域搜索

Fig. 3 Eight-neighborhood search

$$f(n) = g(n) + h(n) \quad (1)$$

其中, n 为当前节点, $f(n)$ 为当前节点 n 的总代价, $g(n)$ 为起始节点到当前节点的实际代价, $h(n)$ 为当前节点到目标节点的预估代价, 也被称为启发函数。计算代价值的常用方法有曼哈顿距离、切比雪夫距离和欧氏距离 3 种, 使用欧氏距离估计的路径比其他距离更接近于实际路径, 因此本文采用欧氏距离。设置起始节点的坐标为 (x_s, y_s) , 当前节点的坐标为 (x_c, y_c) , 目标节点的坐标为 (x_g, y_g) , 则表达式分别为:

$$g(n) = \sqrt{(x_s - x_c)^2 + (y_s - y_c)^2} \quad (2)$$

$$h(n) = \sqrt{(x_g - x_c)^2 + (y_g - y_c)^2} \quad (3)$$

当 $g(n)$ 的值远大于 $h(n)$ 时, 算法优先考虑广度优先搜索, 从起始节点逐层进行扩展, 增大了搜索空间, 忽略了启发函数引导搜索方向的作用, 导致搜索过程需要更多的时间和计算资源, 但可以确保找到最短路径。相反, 当 $h(n)$ 的值远大于 $g(n)$ 时, 算法优先考虑深度优先搜索, 从起始节点沿着图的深度方向扩展节点, 直至到达最深节点, 然后回溯到上一层继续深度搜索, 导致算法过早地收敛到一个不是最优解的路径上, 但可以有效提高搜索速度。因此, 在使用 A* 算法时, 应根据具体的工作环境设计合适的启发函数, 以确保其启发效果, 实现既快速又准确地规划路径。

传统 A* 算法虽然能够较快地找到从起始节点到目标节点的全局导航路径, 但规划的路径仍然存在大量冗余节点和拐点, 导致计算量过大, 搜索效率较低, 路径不够平滑, 不利于机器人运动底盘的路径跟踪控制。为此, 本文在传统 A* 算法的基础上进行改进, 从搜索策略、代价函数和路径平滑 3 个方面进行算法性能的优化。

2 改进 A* 算法

2.1 矩形扩展策略

在路径规划算法中, 邻域扩展产生子节点, 以便算法找到下一个路径节点。传统 A* 算法的扩展规则是将当前节点周围 8 个方向的相邻节点作为子节点, 在每个扩展节点上重复此过程, 会导致花费大量的时间。为此本文引入矩形扩展策略, 将八邻域搜索改进为矩形扩展搜索, 只访问矩形的边界节点, 加快了算法的搜索速度。

矩形搜索策略不再是以传统的单个节点搜索地图, 而是以矩形边界作为扩展对象, 有效提升搜索过程的整体运行效率。从左下角起始节点开始, 沿水平方向扩展, 直至遇到障碍物时停止, 得到初始线性区域, 如图 4(a) 的线段所示。将此区域作为待扩展矩形的边界, 沿垂直方向扩展至障碍物停止, 得到首个不存在障碍节点的矩形区域, 如图 4(b) 的矩形所示。依据矩形各个边界的方位判断边界节点在相应方位上是否与障碍节点相邻, 将矩形区域边界上所有不与障碍物相邻的节点作为后继搜索节点。选择最佳的后继搜索节点作为关键路径节点, 这些后继搜索节点共同组成下一线性区域。如图 4(c) 所示, 矩形右边界节点在水平向右方向上有 3 个节点不与障碍节点相邻, 组成线性区域。以此作为待扩展矩形的边界再次扩展, 扩展方向取决于该线性区域在当前矩形边界处的位置, 同样遇到障碍物时停止, 得到下一矩形区域。图 4(c) 中的线性区域

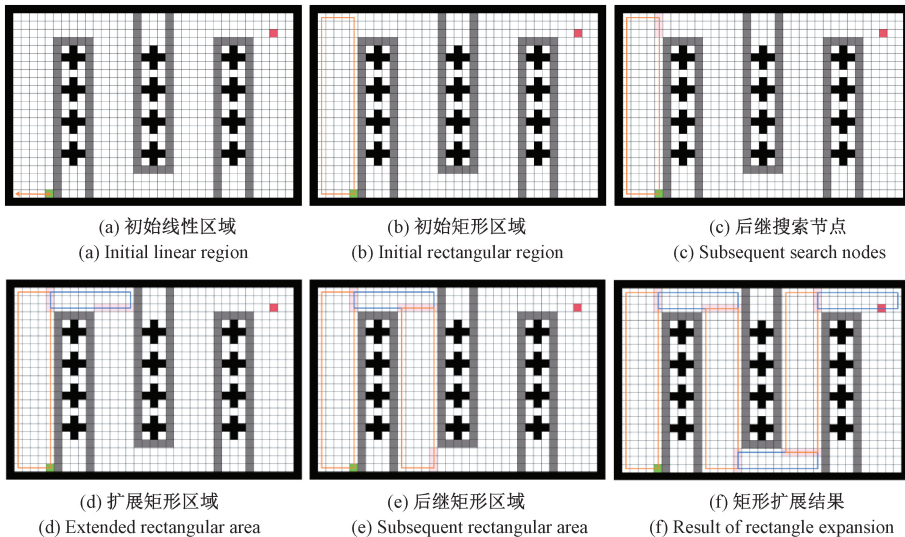


图 4 矩形扩展策略示意图

Fig. 4 Schematic diagram of the rectangular expansion strategy

传统的平滑方法一般采用贝塞尔曲线^[15]或B样条曲线^[16]来优化路径平滑度,这些方法对控制点的位置和数量有较高的要求,且在调整过程中容易出现曲率突变,通常需要多次迭代才能达到理想的平滑效果。考虑到导航路径与障碍物的位置关系以及机器人的运动学特性,本文采用基于动态相切圆的路径调整方法,利用2.3节所得到的关键路径节点作为控制点,通过动态调整相切圆来对局部路径进行平滑处理,调整的原理如图7所示。

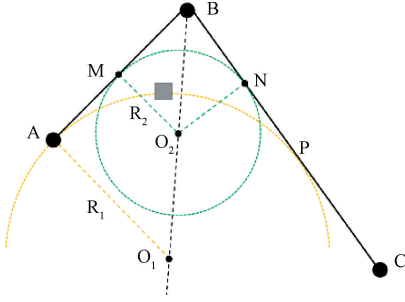


图7 局部路径调整示意图

Fig.7 Schematic diagram of local path adjustment

具体的步骤如下:

1) 确定初始切点。选择关键节点间相邻折线段AB与BC中较短的线段(以图7中AB为例)。将点A(x_a, y_a)作为初始切点,在此点上作线段AB的垂线,并使其与 $\angle ABC$ 的角平分线交于点 $O_1(x_1, y_1)$ 。以 O_1 为圆心、 AO_1 为半径,则圆心坐标和半径分别可通过以下公式计算:

$$x_1 = \frac{x_a + k_1 \cdot y_a}{1 + k_1 \cdot k_2} \quad (6)$$

$$y_1 = \frac{k_2 \cdot (x_a + k_1 \cdot y_a)}{1 + k_1 \cdot k_2} \quad (7)$$

$$R_1 = \frac{\sqrt{(k_1^2 + 1) \cdot (y_a - k_2 \cdot x_a)^2}}{|1 + k_1 \cdot k_2|} \quad (8)$$

其中, k_1 为线段AB的斜率, k_2 为 $\angle ABC$ 角平分线的斜率。相切圆的方程为:

$$(x - x_1)^2 + (y - y_1)^2 = R_1^2 \quad (9)$$

2) 绘制相切圆。以点 O_1 为圆心、 R_1 为半径,绘制一个与AB相切于点A处的圆,与BC相切于点P,从而得到由A至P形成的圆弧 \widehat{AP} 。

3) 检测障碍物冲突。评估圆弧 \widehat{AP} 是否与障碍物发生碰撞。如果圆弧轨迹与障碍物不发生碰撞,则使用圆弧 \widehat{AP} 替换原折线段AB与BP,实现路径转折点的平滑处理。如果圆弧轨迹与障碍物发生碰撞,则需要对切点进行动态调整。将初始切点A沿线段AB方向移动固定距离至新位置M,将点M设为新的切点,再以 O_2 为圆心和 R_2 为半径重新绘制相切圆,并检测是否与障碍物发生碰撞。如果仍发生碰撞,则继续调整切点,直到找到不与障碍物发生碰撞的点为止,最终得到由M至N形成的圆弧 \widehat{MN} 。使

用圆弧 \widehat{MN} 替换原折线段MB与BN,完成路径转折点的平滑处理。

4) 结束条件。当导航路径上的所有关键路径节点均完成平滑处理后,结束平滑过程。

在图6所示含有关键节点的导航路径上应用本节基于动态相切圆的调整方法,得到如图8所示平滑优化后的导航路径。相较于原始路径,优化后的路径曲率连续且更加平滑,更有利于移动机器人在实际应用中的路径跟踪和运动控制,进而提高机器人的工作效率和安全性。

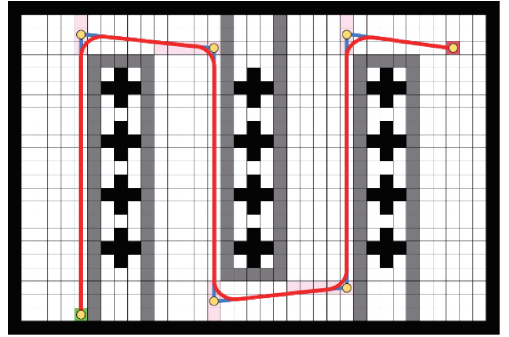


图8 平滑优化后的导航路径

Fig.8 Smoothly optimized navigation path

3 改进A*算法与DWA算法融合

3.1 DWA算法

在已知果园环境地图的情况下,机器人可以做好全局路径规划,但在行驶过程中可能会遇到未知障碍物,从而引发碰撞的风险。因此,为实现移动机器人的实时避障能力,本文采用DWA算法,该算法基于预测控制的理论框架,虽非最优解策略,但在未知环境地图下仍能稳定地躲避障碍,且具备运算开销低、响应时间短和易于实现等优势,适用于局部导航任务。算法首先根据当前机器人的位置和速度信息,在速度参数空间中生成多组线速度和角速度,将这些速度组合代入到运动模型中并模拟出移动轨迹,基于评价函数对预测轨迹进行评估,最后从所有预测轨迹中选择最佳轨迹所对应的速度组作为机器人的移动速度,如此循环直至机器人到达目标节点。

DWA算法基于机器人的初始位姿和速度信息模拟出运动轨迹信息,移动机器人在极小时间间隔 Δt 内位移极短,因此相邻两点间的行驶路线可以被近似为一条直线段,从而得到机器人运动学模型的数学表达式为:

$$\begin{cases} x_n = x_{n-1} + v \cdot \Delta t \cdot \cos\theta_{n-1} \\ y_n = y_{n-1} + v \cdot \Delta t \cdot \sin\theta_{n-1} \\ \theta_n = \theta_{n-1} + \omega \cdot \Delta t \end{cases} \quad (10)$$

其中, x_n, y_n, θ_n 为机器人 n 时刻的位姿信息; $x_{n-1}, y_{n-1}, \theta_{n-1}$ 为机器人 $n-1$ 时刻的位姿信息; v 和 ω 分别为机器人的线速度和角速度。

完成对机器人运动学特性的建模后,可基于速度参数

来预测行驶路径。在速度空间 (v, ω) 中, 存在多组线速度和角速度。在实际采样过程中, 考虑到硬件性能和外部条件的限制, 需要对可供选择速度的范围进行合理的约束。

移动机器人自身速度约束为:

$$v_m = \left\{ \begin{array}{l} (v, \omega) \mid v \in [v_{\min}, v_{\max}] \\ \omega \in [\omega_{\min}, \omega_{\max}] \end{array} \right\} \quad (11)$$

其中, v_{\min} 、 v_{\max} 分别为机器人的最小线速度和最大线速度; ω_{\min} 、 ω_{\max} 分别为机器人的最小角速度和最大角速度。

移动机器人电机性能加减速约束为:

$$v_d = \left\{ \begin{array}{l} (v, \omega) \mid v \in [v_c - \dot{v}_b \Delta t, v_c + \dot{v}_a \Delta t] \\ \omega \in [\omega_c - \dot{\omega}_b \Delta t, \omega_c + \dot{\omega}_a \Delta t] \end{array} \right\} \quad (12)$$

其中, v_c 、 ω_c 为机器人当前的线速度和角速度; \dot{v}_b 、 \dot{v}_a 为最大线减速度和最大线加速度; $\dot{\omega}_b$ 、 $\dot{\omega}_a$ 为最大角减速度和最大角加速度。

移动机器人在最大减速度下的安全制动约束为:

$$v_a = \left\{ \begin{array}{l} (v, \omega) \mid v \leq \sqrt{2dist(v, \omega)\dot{v}_b} \\ \omega \leq \sqrt{2dist(v, \omega)\dot{\omega}_b} \end{array} \right\} \quad (13)$$

其中, $dist(v, \omega)$ 为机器人与最近障碍物的距离。

最终, 移动机器人的速度采样空间 v_r 为满足上述 3 种约束的集合:

$$v_r = v_m \cap v_d \cap v_a \quad (14)$$

3.2 改进评价函数

对速度信息采样, 生成多条可能的预测路径, 基于评价函数对这些路径进行计算, 选取出评价指标值最高的预测路径作为行驶轨迹。传统 DWA 算法的评价函数 $G(v, \omega)$ 表达式为:

$$G(v, \omega) = \sigma(\alpha \cdot heading(v, \omega) + \beta \cdot dist(v, \omega) + \gamma \cdot velocity(v, \omega)) \quad (15)$$

其中, $heading(v, \omega)$ 为方位角评价函数, 表示当前选择的速度组预测的轨迹末端朝向与目标节点位置航向角之间的角度偏差, 角度偏差越小则取值越大; $dist(v, \omega)$ 为障碍物评价函数, 表示当前选择的速度组模拟的轨迹距离障碍物的最近距离, 距离越大取值越大; $velocity(v, \omega)$ 为速度评价函数, 表示当前选择的速度组模拟的轨迹的速度大小, 速度越大则取值越大; σ 为平滑系数, 表示归一化处理; α 、 β 、 γ 为相应评价函数的权重系数。

传统 DWA 算法容易出现陷入局部最优和偏离全局路线的问题, 因此在设计评价函数时需要平衡不同因素之间的权重, 以便在考虑全局路径信息的同时, 兼顾局部避障和路径优化效果。为更好的满足果园环境应用场景, 并确保路径的平滑性和安全性, 本文在传统评价函数的基础上, 做出三方面的改进处理: 优化方位角评价函数, 预测轨迹末端方向与改进 A* 算法路径节点之间的角度偏差值替代传统仅与目标节点之间的角度偏差值, 以此优化后的评价子函数可显著增强路径搜索过程的针对性; 新增全局路

径偏离距离评价函数, 计算预测轨迹路线与全局规划路线之间的距离, 确保在运动过程中机器人能更紧密地沿着全局最优路径前进; 调整障碍物评价函数, 区别处理静态已知障碍物和未知障碍物, 降低它们之间的相互干扰, 以更好地综合考虑环境信息。最终修改后的评价函数为:

$$G(v, \omega) = \sigma(\alpha \cdot heading(v, \omega) + \beta \cdot dist_{off}(v, \omega) + \lambda \cdot dist_k(v, \omega) + \mu \cdot dist_{uk}(v, \omega) + \gamma \cdot velocity(v, \omega)) \quad (16)$$

其中, $heading(v, \omega)$ 为方位角评价函数, 表示预测轨迹末端朝向与关键路径节点位置的角度偏差, 角度偏差越小取值越大; $dist_{off}(v, \omega)$ 为路径偏离距离评价函数, 表示预测轨迹路线与全局规划路线之间的距离, 距离越小取值越大; $dist_k(v, \omega)$ 为预测轨迹终点与已知障碍物之间的最近距离; $dist_{uk}(v, \omega)$ 为预测轨迹终点与未知障碍物之间的最近距离; σ 为平滑系数; α 、 β 、 λ 、 μ 、 γ 为相应评价函数的权重系数。

3.3 融合算法

改进后的 A* 算法可以在已知全局地图信息的状态下, 规划出一条包含起始节点、目标节点和关键路径节点的最优路径, 该算法在全局规划能力方面表现出色, 但遇到未知的障碍物时, 算法很难适应未知环境并及时更新路径。具有良好实时避障能力的 DWA 算法在仅依赖单一目标节点引导时, 往往会陷入局部最优, 导致生成的最终路径难以达到全局最优的目标。因此, 为克服各自的局限, 本文结合改进后的 A* 算法和 DWA 算法, 使用改进 A* 算法进行全局规划得到最优路径, 将规划出的关键节点作为改进 DWA 算法的全局参考, 分段进行局部动态避障, 确保整体路径的最优性, 最终安全到达目标节点。融合算法可以充分发挥两种算法的优势, 当路径上出现静态未识别的障碍物, 既可以使移动机器人及时绕行障碍物, 又可以最大程度地保证全局路径的最优化, 提高移动机器人在果园场景下路径规划的安全性和鲁棒性。融合算法的具体流程如图 9 所示, 具体的内容如下:

首先, 利用三维点云数据创建二维栅格地图, 确定路径起始节点并设定路径目标节点, 为后续规划奠定基础。随后, 在该地图上应用本文改进 A* 算法完成全局路径规划。采取矩形扩展策略筛选后继搜索节点, 判断矩形区域是否包含目标节点。若未包含则继续扩展, 若包含则选取关键节点策略确定关键路径节点, 形成初始全局导航路径。再通过调整相切圆策略对拐点处平滑, 得到优化后的全局路径。进入融合规划流程, 将本文改进 A* 算法规划所得路径的起始节点设置为当前节点, 下一路径关键节点设为局部目标节点。根据机器人当前速度约束, 在线速度与角速度空间内进行随机采样, 生成多个候选速度组合。基于本文改进 DWA 算法的评价函数对各候选组合对应的模拟轨迹进行打分, 选取得分最高的速度组并将其对应轨迹作为当前控制指令, 驱动机器人沿该轨迹前行。在机器

人沿轨迹移动过程中,若未能及时到达局部目标节点或检测到未知障碍物,算法将实时重新采样并打分更新控制指令,以保证避障与全局收敛的平衡。若到达局部目标节点,判断该局部目标节点是否为全局目标节点。若不是则更新下一路径关键节点为新的局部目标节点,重复上述局部规划流程,若是则融合算法输出最终路径,流程结束,得到一条连续且曲率平滑的最终导航路径。

通过上述融合流程,本文改进 A* 算法提供全局最优路径框架与关键路径节点指引,本文改进 DWA 算法基于关键路径节点分段动态避障,二者协同规划,既保持了全局规划的最优性,又通过局部动态调整应对未知障碍物,完成了果园环境下移动机器人高效且安全的导航任务。

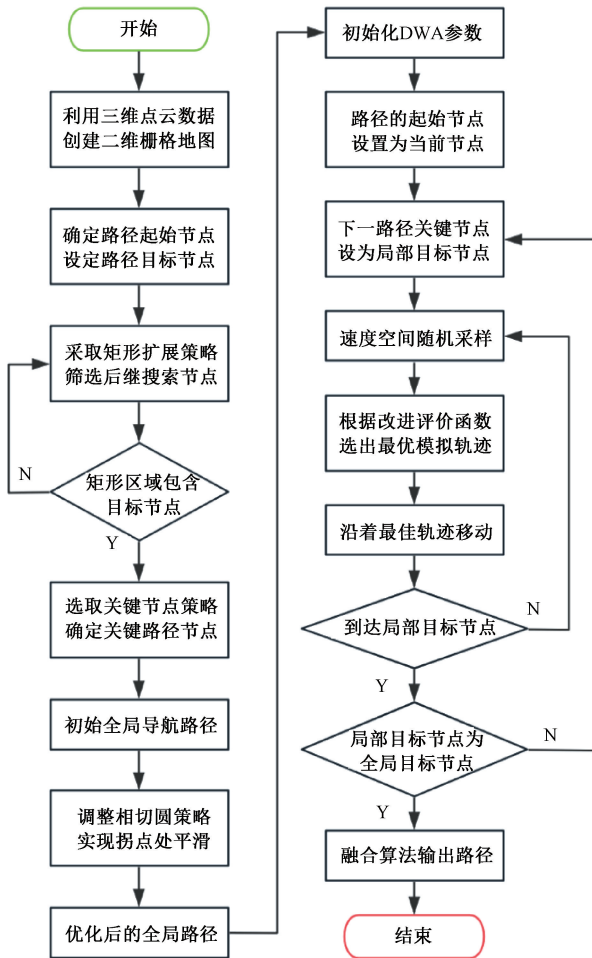


图 9 融合算法流程图

Fig. 9 Flowchart of fusion algorithm

4 实验与结果分析

4.1 改进 A* 算法实验

为了评估所提算法的性能提升效果,使用 Windows 11 操作系统、Intel Core i5 处理器、3.10 GHz 主频、16 GB 运行内存和 Python 3.10 软件环境的配置,选取传统 A* 算

法、文献[17]改进算法和本文改进算法应用于基于果园环境所绘制的栅格地图上。文献[17]聚焦于非结构化、不平整地形下的机器人路径规划问题,与果园环境的复杂、非结构化特性具有相似性,为解决果园场景下的路径规划提供了思路。文献[17]在 A* 算法的基础上引入双向搜索策略提升计算速度,并通过路径节点过滤减少关键节点数量,增加坡度约束保障路径安全性,以及利用贝塞尔曲线拟合实现路径平滑,为本文在优化算法搜索效率以及改进路径平滑性方面提供了参考。文献[17]虽在通用非结构化地形取得了成果,但果园场景具有自身的独特性。不同于文献[17]的双向搜索,本文提出矩形扩展搜索,针对果园狭窄通道与复杂布局,优化搜索方式以提高效率。在关键路径节点选取上,区别于文献[17]的固定策略,本文基于机器人特性与果园环境特征动态调整,更贴合果园实际作业需求。本文路径平滑采用动态相切圆策略,在果园复杂地形下实时适应调整,比文献[17]的贝塞尔曲线拟合更具灵活性。

为了大致模拟机器人的工作环境,按照栅格法构造了 37×25 的环境地图,设定路径规划的起始节点坐标为(6, 2),目标节点的坐标为(34, 22),单位栅格的边长为 1 m。设置圆形边界表示障碍区域,空白表示可通行区域,菱形表示果树,左下角设为路径起点,右上角设为路径终点,叉形表示路径搜索节点。各 A* 算法规划得到初始导航路径,在路径平滑后得到最终的路径,各算法的最终规划结果如图 10 所示。

由图 10(a)可知,传统 A* 算法成功规划出一条从左下角起始节点通往右上角目标节点的一条可通行路线,叉形表示算法运行过程中的路径搜索节点,该算法路径搜索节点数量多,规划搜索范围大,且路径轨迹转折数量也较多,难以满足果园环境下移动机器人对平稳行驶的要求。文献[17]在传统 A* 算法的启发函数中引入指数权重以提高寻路效率,图 10(b)中路径搜索节点数量较传统算法有所减少,但也存在转折点较多的问题,行驶效率未显著提升。随后,文献[17]通过贝塞尔曲线拟合路径使其变得平滑,图 10(c)中的导航路径相比于图 10(b)中的原始路径更为平滑,有助于机器人的运动控制。本文基于传统 A* 算法提出矩形扩展搜索策略,只访问矩形的边界节点,并结合关键节点选取方案优化路径。如图 10(d)所示,路径搜索节点数量相较于前述算法显著减少,且路径轨迹上仅含 5 个转折点,使机器人在行驶过程中尽量保持直行。在路径平滑优化方面,本文不采用传统的平滑方法,而是基于动态相切圆的路径平滑方法,仅在转折点处用圆弧代替折线,生成满足最大曲率约束的平稳曲线。如图 10(e)所示,所得导航路径能使机器人在转弯处实现更加平稳的转向,有利于其更换树行继续进行采摘作业。综合评价,由本文改进算法生成的路径更贴近树列,更符合果园环境下机器人进行果实采摘作业的行驶轨迹,机器人在直行和转弯路

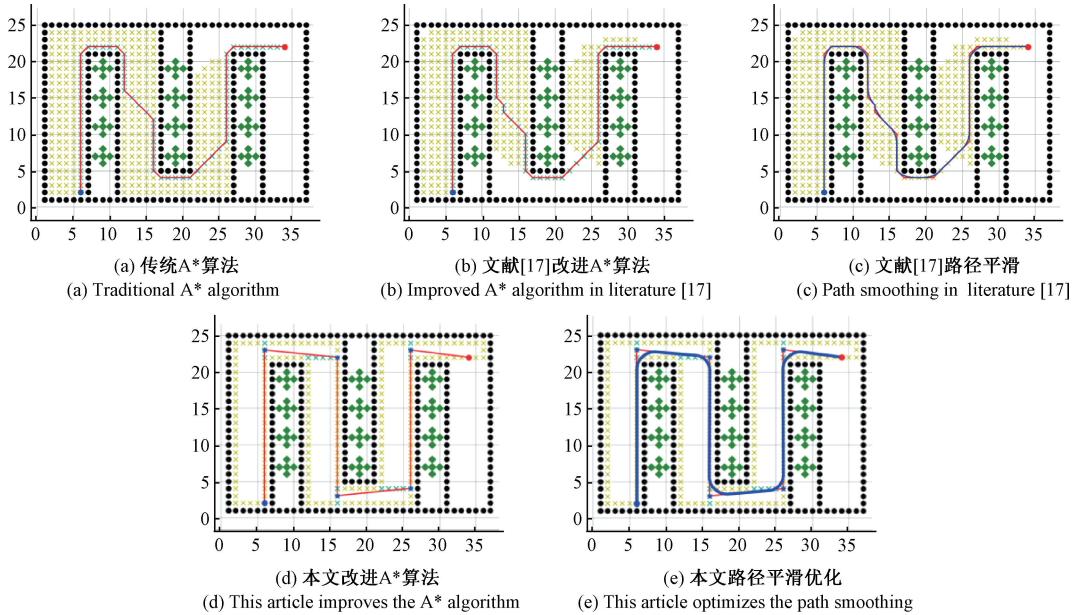


图 10 37×25 地图路径规划结果

Fig. 10 Path planning results in 37×25 map

段都能与障碍物保持合理的避让距离,保障行进的安全性。

再扩大地图规模,建立 57×41 的环境地图模型,将传统 A* 算法、文献[17]改进算法和本文改进算法对比仿真测试,验证算法的环境适用性及灵活性。各算法的结果如图 11 所示。对比发现,传统 A* 算法的搜索节点和转折点数量较多,文献[17]改进算法虽减少了搜索节点数量,但

转折点数量未明显下降。本文算法显著减少了搜索节点的数量以及转折点的数量,且平滑后的导航路径轨迹非常契合在果园环境下移动机器人采摘作业的行驶轨迹,机器人可以尽量保持直线行驶,且在转弯处实现平稳地转向,并能安全行驶到下一列果树的作业起始位置,直至安全抵达目标位置,实现了机器人在规范化种植的果园中合理且安全地执行果实采摘作业。

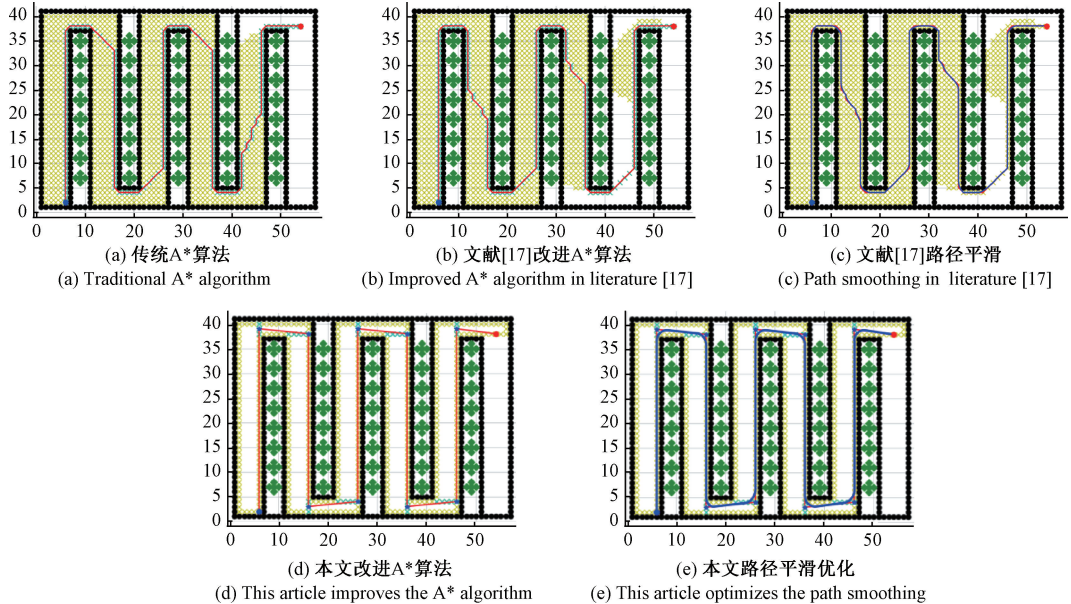


图 11 57×41 地图路径规划结果

Fig. 11 Path planning results in 57×41 map

本实验设定相同起点和终点,记录各个算法的路径长度、平均规划时间、搜索节点数量、路径节点数量、转角数

量、转弯角度以及是否存在曲率突变的性能评价指标。A* 算法具备最优搜索特性,规划后得到唯一路径,但在不

同硬件配置和运行环境下,单次运算所花费的时间会有所不同。为减少运算时间对实验结果的干扰,在统一的环境下对每种算法运行 10 次,并计算每种算法的平均规划时

间,评估改进后算法的执行效率。不同算法路径的性能比较如表 1 所示。

表 1 不同 A* 算法路径的性能比较

Table 1 Comparison of path performance for different A* algorithms

地图规格	算法名称	路径长度/m	规划时间/s	搜索节点个数	路径节点个数	转角数量个数	转弯角度/(°)	是否曲率突变
37×25	传统 A* 算法	76.38	1.79	360	72	12	540	是
	文献[17]算法	76.38	1.56	314	72	14	630	是
	文献[17]平滑	75.27	1.67	—	—	—	—	否
	本文改进算法	87.16	1.31	207	7	5	457	是
	本文平滑优化	84.95	1.43	—	—	—	—	否
	本文比传统算法性能改变比例/%	+11.2	-20.1	-42.5	-90.3	-58.3	-15.4	—
57×41	传统 A* 算法	205.36	7.98	1020	196	26	1170	是
	文献[17]算法	205.36	6.59	908	196	28	1260	是
	文献[17]平滑	202.69	6.72	—	—	—	—	否
	本文改进算法	225.26	5.27	501	11	9	817	是
	本文平滑优化	221.28	5.51	—	—	—	—	否
	本文比传统算法性能改变比例/%	+7.8	-31.0	-50.9	-94.4	-65.4	-30.2	—

表 1 中的数据表明,在 37×25 的栅格地图中,文献[17]改进算法与传统 A* 算法相比,规划的路径长度和路径节点数量一致,搜索节点数量和规划时间有所减少,但转折点数量和转弯角度有所增加,且路径都存在曲率突变。文献[17]算法的路径在经过平滑处理后,路径长度有所下降,但规划时间略微有所增加,平滑后的路径不存在曲率突变。本文改进 A* 算法规划的路径长度大于传统 A* 算法和文献[17]算法,这主要归因于保证路径的合理性,路径长度的增加是为了使机器人更顺利地执行对果树的采摘作业。除这一影响路径表现的因素外,本文改进算法的路径规划时间、搜索节点数量、路径节点数量、转折点数量以及转弯角度均小于传统 A* 算法和文献[17]算法。与传统 A* 算法相比较,本文算法规划路径的搜索节点数减少了 42.5%,路径节点数减少了 90.3%,转折点数减少了 58.3%,累计转弯角度减少了 15.4%,但存在着路径曲率突变的问题。使用本文提出的路径平滑方法对路径进行优化后,路径长度有所减少,规划时间有所增加,相比于传统 A* 算法路径长度增加了 11.2%,规划时间减少了 20.1%,消除了折线路径的曲率突变问题。

在 57×41 的栅格地图中,与传统 A* 算法相比较,本文改进算法路径长度增加了 7.8%,规划时间减少了 31.0%,路径的搜索节点数减少了 50.9%,路径节点数减少了 94.4%,转折点数减少了 65.4%,累计转弯角度减少了 30.2%,同时全路径曲率保持连续。综合评价,本文对于 A* 算法所提出的改进策略有着明显优势,能有效地完成机器人的路径规划任务,虽然在路径长度上有所牺牲,

但是路径轨迹更贴合果园环境下机器人的采摘作业路线,且其他性能指标均有不同程度的优化,验证了本文算法的合理性及优越性。

4.2 融合算法仿真实验

为验证融合 DWA 算法后移动机器人对障碍物的避障能力,在改进 A* 规划算法得到的路径上添加未知障碍物,选取传统 DWA 算法、文献[18]改进算法和本文改进算法进一步展开实验。文献[18]与本文均对 DWA 算法的评价函数进行改进,以提升路径规划效果。文献[18]通过将前后时刻航向角之差引入评价函数,优化了复杂环境下的转向角度,减少了障碍物处的转向幅度,与果园中分布不规则的障碍物特性相符,为本文在保持平顺避障与路径贴合性方面提供了启示。本文在此基础上,优化方位角评价函数、新增全局路径偏离距离评价函数并调整障碍物评价函数。前者聚焦室内场景,未考虑全局路径偏离及障碍物类型区分,本文更贴合果园环境下移动机器人的作业需求,增强全局导向性与环境适应性。

37×25 和 57×41 仿真环境下的路径分别如图 12 和图 13 所示。深灰色矩形表示未知障碍物,实曲线与虚曲线分别表示本文 A* 算法的路径以及不同融合算法的路径。如图 12(a)所示,在已知地图环境下,由本文 A* 算法规划得到了一条从起点通往终点的安全可通行路径,而当地图上出现未知障碍物时,规划路径并不具备避开障碍物的能力,需要进行局部路径规划修正。从图 12(b)~(d)可以看出,融合 DWA 算法后,机器人依据全局路径为指引,对未知障碍物作合理的响应,并与障碍物保持合适的间隔

距离。本文改进 A* 算法融合传统 DWA 算法得到的路径如图 12(b) 所示,传统算法未能对已知和未知障碍物加以区分,尽管实现了避障,但较为偏离全局路径,且与障碍物间距较大。本文改进 A* 算法融合文献[18]改进 DWA 算法得到的路径如图 12(c) 所示,尽管驱使机器人在转向过程中尽可能保持小角度避障,但也较为偏离全局路径。本文改进 A* 算法融合本文改进 DWA 算法得到的路径如图 12(d) 所示,本文采用以局部目标节点为指引,考虑与全局路径的偏离程度以及不同类型障碍物信息,最终得到的路径在实现避障的同时更贴合全局路径,更适合果园环境下的采摘任务。

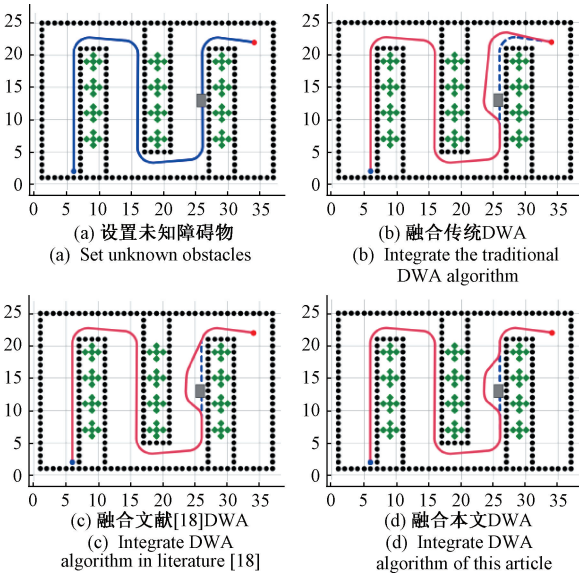


图 12 37×25 地图融合算法规划结果
Fig. 12 Planning results in 37×25 map

扩大地图规模后进一步验证融合算法,如图 13(d) 所示,本文改进 A* 算法融合本文改进 DWA 算法得到一条平滑且安全的路径,既不与已知和未知障碍物存在碰撞问题,也不过于远离,在保证全局路径最优的基础上,实现了更贴合全局路径的局部避障功能,为移动机器人在果园环境中的路径规划与导航控制提供了可靠的支持。

在 37×25 和 57×41 环境下分别进行 10 次仿真实验,计算不同融合算法的平均路径长度和规划时间,得到算法的性能比较如表 2 所示。在 37×25 的环境下,融合本文改进 DWA 算法相比较于传统 DWA 算法与文献[18]改进 DWA 算法,路径长度的改善程度分别为 1.16% 和 0.59%,规划时间的改善程度分别为 13.59% 和 6.81%。在 57×41 的环境下,融合本文改进 DWA 算法相比较于传统 DWA 算法与文献[18]改进 DWA 算法,路径长度的改善程度分别为 0.5% 和 0.31%,规划时间的改善程度分别为 7.37% 和 4.07%。采用本文改进 A* 算法融合本文改进的 DWA 算法后有效缩短了规划路径的长度和时间,实现了移动机器人在果园环境下面对未知障碍物时的避障

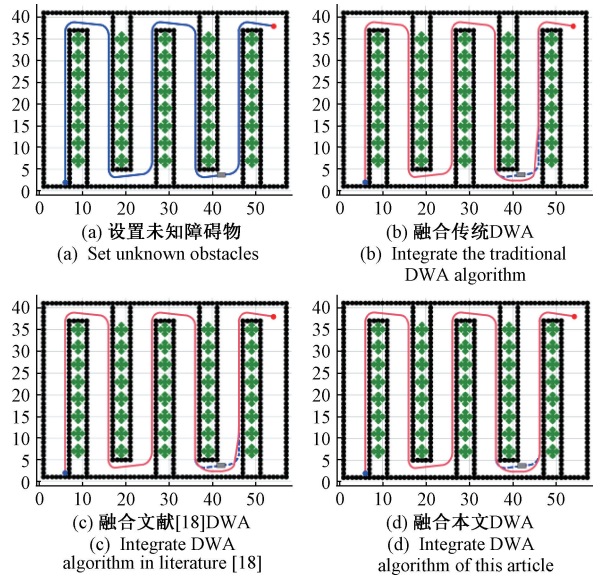


图 13 57×41 地图融合算法规划结果
Fig. 13 Planning results in 57×41 map

能力,更加符合果园中采摘作业行走路线的需求。

表 2 不同融合算法路径的性能比较
Table 2 Comparison of path performance for different fusion algorithms

地图	融合算法名称	路径长度/m	规划时间/s
37×25	传统 DWA	89.31	25.82
	文献[18]DWA	88.79	23.94
	本文 DWA	88.27	22.31
57×41	传统 DWA	226.09	61.85
	文献[18]DWA	225.64	59.72
	本文 DWA	224.95	57.29

在 Gazebo 仿真平台中搭建果园仿真环境,以 Komodo-02 履带式机器人底盘搭载雷达传感器获取环境信息,构建地图验证融合算法的有效性。设计的果园环境如图 14 所示,图中包含了果园环境中的墙壁与树木。图 15(a)所示为 Komodo-02 机器人的实物图,设计机器人仿真模型并显示在果园仿真环境中,如图 15(b)所示。

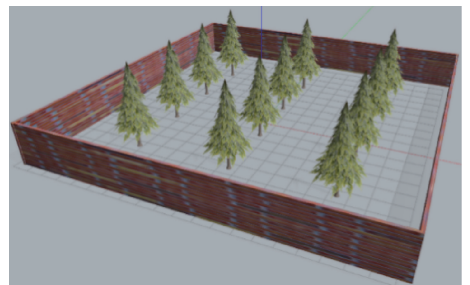
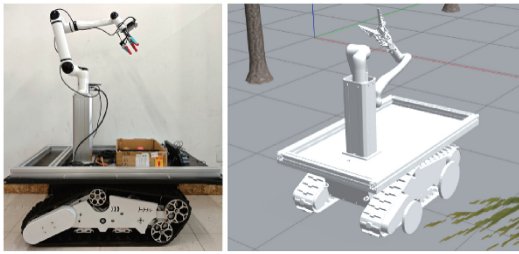


图 14 果园仿真环境
Fig. 14 Orchard simulation environment



(a) 机器人实物图
(a) Physical robot
(b) 机器人模型图
(b) Robot model diagram

图 15 Komodo-02 履带式机器人底盘

Fig. 15 Komodo-02 tracked robot chassis

在搭建果园场景与机器人模型后,基于前述的环境建模策略,得到果园环境的二维栅格地图,在 Rviz 工具中配置相关组件,呈现出仿真环境与机器人模型,如图 16 所示,灰色区域表示可通行区域,黑色区域表示墙壁及树干障碍,白色区域表示局部地图,机器人模型位于出发位置。

在原有的果园环境中添加静态未知障碍物,测试融合算法的导航与避障效果,图 17 所示为最终结果。移动机器人首先基于本文的 A* 算法进行全局路径规划,得到如图 17(a)中部所示的路线。在沿着规划路线行驶的过程中,机器人基于雷达信息探测出前方的未知障碍物,使用本文优化的 DWA 算法动态调整局部路线,如图 17(b)所

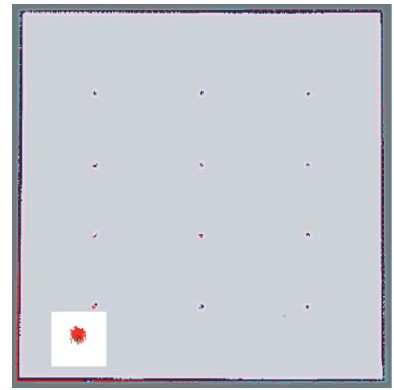
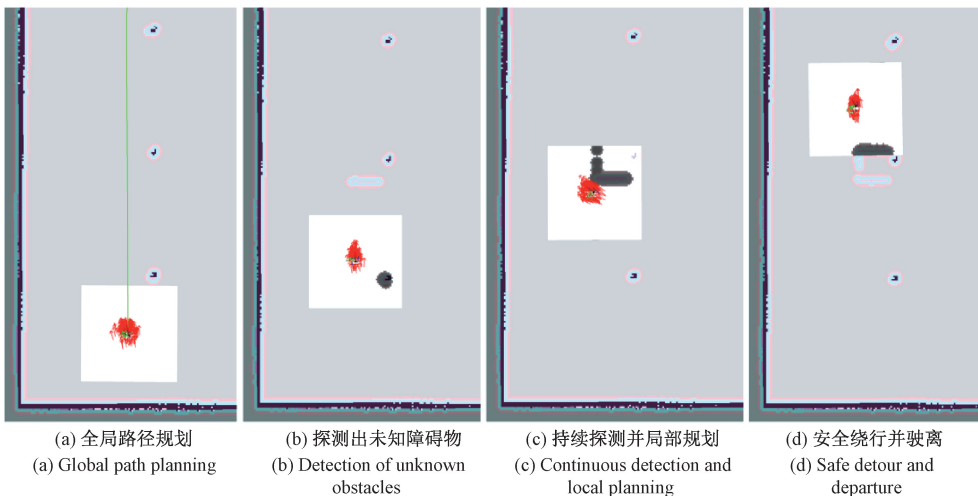


图 16 果园环境地图与机器人模型

Fig. 16 Orchard environment map and robot model

示。在持续前进的过程中,不断探测障碍物的位置信息,实时修正行驶速度和行进角度,通过规划的局部路线实现避障功能,如图 17(c)所示,机器人从障碍物的左侧安全绕行,且与障碍物保持合理的间距。最终在绕过添加的未知障碍物之后,机器人继续移动至全局路径,如图 17(d)所示。实验结果表明,本文提出的改进 A* 和 DWA 融合路径规划算法,使得机器人在沿着全局最优路径移动的过程中,具备对静态未知障碍物的规避能力,验证了本文融合算法的可行性。



(a) 全局路径规划
(a) Global path planning
(b) 探测出未知障碍物
(b) Detection of unknown obstacles
(c) 持续探测并局部规划
(c) Continuous detection and local planning
(d) 安全绕行并驶离
(d) Safe detour and departure

图 17 融合算法避障实验结果

Fig. 17 Experimental results of obstacle avoidance using the fusion algorithm

5 结 论

针对果园场景下 A* 算法节点遍历较多、规划时间较长、路径不够平滑及靠近障碍物等不足,本文设计一种结合改进 A* 与 DWA 算法的路径规划策略。在 A* 算法的基础上引入矩形扩展搜索、关键节点选取及动态相切圆平滑方法,生成符合果园环境需求的全局导航路径。在 DWA 算法中优化方位角与障碍物评价函数并新增全局路径偏离指标,辅助机器人选择更优的局部运动轨迹。在两

种地图规模下对比不同 A* 算法的性能,结果表明本文改进算法显著缩短了规划时间,减少了搜索与路径节点数量,降低了转折点及转弯角度,虽然路径长度略有增加,但整体路径更为合理平滑。进一步比较融合不同 DWA 算法的性能,结果显示本文改进算法可有效缩短路径长度和加快规划速度,并在未知障碍物场景中展现出更强的避障能力。最终基于果园场景与机器人模型的仿真测试,有效验证了本文融合算法可实现全局导航与局部避障的协同规划,提升了移动机器人的导航效率和安全性。后续工作

将与其他最新改进算法进一步综合对比,并计划在实际机器人平台上开展果园实地测试,以评估算法在真实环境中的性能,并持续优化算法的细节。

参考文献

- [1] LIU L X, WANG X, YANG X, et al. Path planning techniques for mobile robots: Review and prospect[J]. Expert Systems with Applications, 2023, 227: 120254.
- [2] CAI Y R, LIU H B, LI M F, et al. A method of dual-AGV-ganged path planning based on the genetic algorithm[J]. Applied Sciences, 2024, 14(17): 7482.
- [3] DONG ZH, ZHANG X H, YANG W J, et al. Ant colony optimization-based method for energy-efficient cutting trajectory planning in axial robotic roadheader[J]. Applied Soft Computing, 2024, 163: 111965.
- [4] MIYOMBO M E, LIU K Y, MULENGA C M, et al. Optimal path planning in a real-world radioactive environment: A comparative study of A-star and Dijkstra algorithms [J]. Nuclear Engineering and Design, 2024, 420: 113039.
- [5] KABIR R, WATANOBE Y, ISLAM M R, et al. Enhanced robot motion block of A-star algorithm for robotic path planning[J]. Sensors, 2024, 24(5): 1422.
- [6] LIN ZH, WU K, SHEN R L, et al. An efficient and accurate A-star algorithm for autonomous vehicle path planning [J]. IEEE Transactions on Vehicular Technology, 2023, 73(6): 9003-9008.
- [7] YIN CH P, TAN CH Y, WANG CH Q, et al. An improved A-star path planning algorithm based on mobile robots in medical testing laboratories [J]. Sensors, 2024, 24(6): 1784.
- [8] LI CH G, HUANG X, DING J, et al. Global path planning based on a bidirectional alternating search A* algorithm for mobile robots [J]. Computers & Industrial Engineering, 2022, 168: 108123.
- [9] FOX D, BURGARD W, THRUN S. The dynamic window approach to collision avoidance [J]. IEEE Robotics & Automation Magazine, 1997, 4(1): 23-33.
- [10] 曾宪阳,张加旺.改进 A* 算法融合 DWA 机器人路径规划研究[J].电子测量技术,2025,48(6):20-27.
ZENG X Y, ZHANG J W. Path planning for robots with improved A* algorithm and fused DWA [J]. Electronic Measurement Technology, 2025, 48(6): 20-27.
- [11] 袁新亚,戴娟,孙胜强,等.融合 A* 和 DWA 算法的移动机器人路径规划方法[J].电子测量技术,2024, 47(4):95-103.
YUAN X Y, DAI J, SUN SH Q, et al. Integration of A* and DWA algorithms for mobile robot path planning [J]. Electronic Measurement Technology, 2024, 47(4):95-103.
- [12] LIAO T J, CHEN F, WU Y T, et al. Research on path planning with the Integration of adaptive A-star algorithm and improved dynamic window approach [J]. Electronics, 2024, 13(2): 455.
- [13] 赖荣荣,窦磊,巫志勇,等.融合改进 A* 算法和动态窗口法的移动机器人路径规划[J].系统仿真学报,2024, 36(8):1884-1894.
LAI R SH, DOU L, WU ZH Y, et al. Fusion of improved A* and dynamic window approach for mobile robot path planning [J]. Journal of System Simulation, 2024, 36(8): 1884-1894.
- [14] 龚鹏,李文博,胡为,等.融合改进安全 A* 算法和 DWA 算法的无人车路径规划研究[J].火力与指挥控制,2024,49(4):44-50.
GONG P, LI W B, HU W, et al. Research on unmanned vehicle path planning with fusion of improved safety A* algorithm and DWA algorithm [J]. Fire Control & Command Control, 2024, 49(4):44-50.
- [15] LAI R SH, WU ZH Y, LIU X G, et al. Fusion algorithm of the improved A* algorithm and segmented Bézier curves for the path planning of mobile robots[J]. Sustainability, 2023, 15(3): 2483.
- [16] 张可琨,鲍久圣,艾俊伟,等.基于改进 A* 与 DWA 算法的井下搬运机器人自主行走路径规划[J].煤炭科学技术,2024,52(11):197-213.
ZHANG K K, BAO J SH, AI J W, et al. Autonomous walking path planning of underground handling robot based on improved A* and DWA algorithm [J]. Coal Science and Technology, 2024, 52(11):197-213.
- [17] 郭聚刚,于军琪,冯春勇,等.基于改进 A* 算法的机器人不平坦地形全局路径规划[J].计算机工程与应用, 2025, 61(5):309-322.
GUO J G, YU J Q, FENG CH Y, et al. Global path planning for robots on uneven terrain based on an improved A* algorithm [J]. Computer Engineering and Applications, 2025, 61(5):309-322.
- [18] 韩毅,刘毅超,关甜,等.改进 A* 和动态窗口法的无人车路径规划[J].山东大学学报(工学版),2025,55(3): 16-24.
HAN Y, LIU Y CH, GUAN T, et al. Improved A* and dynamic window approach for unmanned vehicle path planning [J]. Journal of Shandong University (Engineering Science), 2025, 55(3):16-24.

作者简介

王晶,硕士研究生,主要研究方向为机器人系统与控制、机器人自主导航。

E-mail: 318969420@qq.com

高亚鹏(通信作者),博士研究生,讲师,主要研究方向为智能机器人技术、深度强化学习、机器人自主学习。

E-mail: gaoyapeng@tyut.edu.cn

李海芳,博士研究生,教授,博士生导师,主要研究方向为人工智能与图像识别、计算机视觉、脑科学与智能计算。

E-mail: lihaifang@tyut.edu.cn