

基于改进 RRT 算法的猕猴桃采摘机器人全局路径规划

崔永杰^{1,2} 王寅初¹ 何智^{1,3} 曹丹丹¹ 马利¹ 李凯^{1,3}

(1. 西北农林科技大学机械与工程学院, 陕西杨凌 712100; 2. 农业农村部农业物联网重点实验室, 陕西杨凌 712100;
3. 陕西省农业信息感知与智能服务重点实验室, 陕西杨凌 712100)

摘要: 为提高猕猴桃采摘机器人导航效率, 提出一种基于采样状态实时引导随机树扩展的改进方法 (Straight - RRT)。首先, 针对传统 RRT 算法盲目搜索的问题, 引入评价指数与阈值划分采样状态, 根据采样状态决定采样节点的选取方式, 实时引导随机树的扩展。其次, 为增强算法对不同环境的自适应性及快速避开不规则障碍物, 引入动态阈值并优化最近节点选择机制。最后对路径进行优化处理, 去除路径冗余点并采用贝塞尔曲线平滑路径减小路径复杂度。基于棚架式猕猴桃果园环境进行路径规划实验, 实验结果表明改进后算法在猕猴桃果园环境中具有更好的适应性及规划效率, 为提高猕猴桃采摘机器人导航效率提供了解决方法。

关键词: 猕猴桃; 采摘机器人; RRT 算法; 路径规划; 评价指数; 采样状态

中图分类号: S126 文献标识码: A 文章编号: 1000-1298(2022)06-0151-08 OSID: 

Global Path Planning of Kiwifruit Harvesting Robot Based on Improved RRT Algorithm

CUI Yongjie^{1,2} WANG Yinchu¹ HE Zhi^{1,3} CAO Dandan¹ MA Li¹ LI Kai^{1,3}

(1. College of Mechanical and Electronic Engineering, Northwest A&F University, Yangling, Shaanxi 712100, China
2. Key Laboratory of Agricultural Internet of Things, Ministry of Agriculture and Rural Affairs, Yangling, Shaanxi 712100, China
3. Shaanxi Key Laboratory of Agricultural Information Perception and Intelligent Services, Yangling, Shaanxi 712100, China)

Abstract: In order to improve the navigation efficiency of the kiwifruit harvesting robot, the rapidly expanding random tree (Rapidly-exploring random trees, RRT) algorithm was researched, and an improved method (Straight - RRT) for real-time guided random tree expansion based on sampling state was proposed. Firstly, aiming at the problem of blind search in the traditional RRT algorithm, an evaluation index and a threshold were introduced to divide the sampling state, and the selection method of sampling nodes was determined according to the sampling state, so as to guide the expansion of the random tree in real time. Secondly, in order to enhance the adaptability of the algorithm to different environments, a dynamic threshold was introduced, which was adaptively adjusted according to the complexity of the environment, and at the same time, the nearest node selection mechanism was optimized to make the random tree avoid irregular obstacles faster. Finally, the path was optimized, the redundant points of the path were removed, and the Bezier curve was used to smooth the path to reduce the complexity of the path. Combined with the environment of the kiwifruit orchard, the effective harvesting area of the kiwifruit harvesting robot was analyzed, and a navigation method to achieve full coverage of the kiwifruit orchard was proposed. Based on the kiwifruit orchard environment, the path planning experiment was carried out, and the algorithm was compared with other improved algorithms. The path planning experiment results showed that the improved algorithm had better adaptability and planning efficiency in the kiwifruit orchard environment, which can provide a basis for improving the navigation efficiency of the kiwifruit harvesting robot.

Key words: kiwifruit; harvesting robot; RRT algorithm; path planning; evaluation index; sampling status

收稿日期: 2022 - 02 - 14 修回日期: 2022 - 04 - 13

基金项目: 国家自然科学基金项目 (31971805)

作者简介: 崔永杰 (1971—), 男, 教授, 博士生导师, 主要从事果蔬生产自动化研究, E-mail: agriculture@nwfau.edu.cn

0 引言

路径规划是猕猴桃采摘机器人的关键技术之一,主要分为基于地图环境已知的全局路径规划和基于传感器信息的局部路径规划^[1-3]。全局路径规划主要包括寻找一条将起始点和目标点连接同时避免与障碍物发生碰撞的路径,其路径长短、算法时间和空间复杂度是评判全局路径规划算法性能的主要指标^[4-6]。常见的全局路径规划算法有 A* 算法、RRT 算法、遗传算法等^[7-9]。其中由 LAVALLE^[10]提出的 RRT 算法概率完备、结构简单且具有灵活的搜索能力,适用于各种复杂环境下的路径规划,但因其随机采样的机制而存在节点利用率低、路径复杂度高且收敛速度慢等问题^[11-13]。

针对 RRT 算法存在的不足,学者们提出多种改进方法,文献[14]提出目标引力-RRT 算法,将人工势场法中的引力思想融入 RRT 算法,通过采样节点与目标点的共同引力作用影响随机树的扩展,增强了节点扩展过程中的目标偏向性,有效提高了路径的搜索效率和收敛速度。文献[15]提出 RRT-Connect 算法,该算法基于构造两个快速搜索随机树分别从起始点和目标点生长,双向生长特性有效缩短了算法收敛时间。文献[16]提出 RRT* 算法,在随机树节点扩展过程中引入了随机几何图与剪枝优化理论,保证了该算法的渐进最优性,极大降低了算法路径复杂度。文献[17]提出的 Quick-RRT* 算法,利用三角不等式原理改进 RRT* 父节点选择和重新布线的方式,提高了初始路径的质量和收敛到最优解的速度^[18-19]。

本文针对传统 RRT 算法搜索效率低、收敛速度慢且路径复杂度高问题,提出一种基于采样状态实时引导随机树扩展的改进方法。首先引入评价指数与阈值实时划分随机树的采样状态,根据采样状态决定采样节点的选取方式,实时引导随机树的扩展,避免随机树的盲目探索。其次,为增强算法对不同环境的自适应性及快速避开不规则障碍物,引入动态阈值及改进最近节点选择机制。最后对路径进行优化处理,得到平滑且复杂度低的路径。

1 算法原理

1.1 问题描述

定义 $X = R^d$ 为配置空间, $X_{obs} \in X$ 为障碍空间、 $X_{free} \in X$ 为无障碍空间, $x_{init} \in X_{free}$ 为起始点, $x_{goal} \in X_{free}$ 为目标点。 $\|x_1, x_2\|$ 为配置空间 X 中节点 x_1 、 x_2 的标准欧氏距离, 设路径 $\beta: [0, 1] \in X$ 为一有界变化函数。 τ 为路径 β 上的点, 如果 $\forall \tau \in [0, 1]$,

$\beta(\tau) \in X_{free}$ 则路径 β 为可行路径。

问题 1: 在给定的条件 ($X_{free}, x_{init}, x_{goal}$) 下寻找一条可行路径, 其中 $\beta: [0, 1] \rightarrow X_{free}$, $\beta(0) = x_{init}$ 且 $\beta(1) = x_{goal}$ 。

问题 2: 在给定的条件 ($X_{free}, x_{init}, x_{goal}$) 下在最短的时间 t 内找到可行路径 $\beta: [0, 1]$ 。

1.2 RRT 算法

RRT 算法基于构造一棵随机采样并逐步扩展的无冲突路径树, 从起始点寻找目标点^[20]。随机树以起始点 x_{init} 为根节点, 在每次迭代中从配置空间 X 中随机选取一个采样节点 x_{rand} , 搜索随机树中的所有节点并找到距离 x_{rand} 最近的节点记为 $x_{nearest}$ 。以 $x_{nearest}$ 为起点沿着 $x_{nearest}$ 和 x_{rand} 的方向以一个特定的步长创建一个新节点 x_{new} , 如果 $x_{new} \in X_{free}$ 且 x_{new} 和 $x_{nearest}$ 之间的连线 $\sigma \in X_{free}$, 则将 x_{new} 添加到随机树中, 并将 $x_{nearest}$ 作为它的父节点。计算每一个新节点与目标点的距离, 如果小于设定的值且两点的连线与障碍物无碰撞则认为找到路径, 将新节点与目标点相连, 通过从目标点回溯到起始点来绘制路径, 否则重复上述搜索过程。

2 改进 RRT 算法

2.1 评价指数

Straight-RRT 将生成的新节点分为两类, 分别是向着目标收敛的新节点和趋于探索的新节点。设评价指数 U 初始值为 1, 实数 i, e 为评价指数增、减量, 每当新节点加入随机树中计算新节点 x_{new} 与目标点 x_{goal} 的欧氏距离 $d = \|x_{new}, x_{goal}\|$, 如果新节点不是当前随机树中最接近目标点的节点 (如图 1a 所示), 则认为该新节点是趋于探索的新节点, 评价指数加 i , 否则认为是向着目标收敛的新节点, 如图 1b 所示, 评价指数减 e 。评价指数可以实时反映随机树当前的探索情况, 评价指数越大表示在当前阶段内随机树向着未知区域探索的点占比越大, 向着目标收敛的点占比越小。

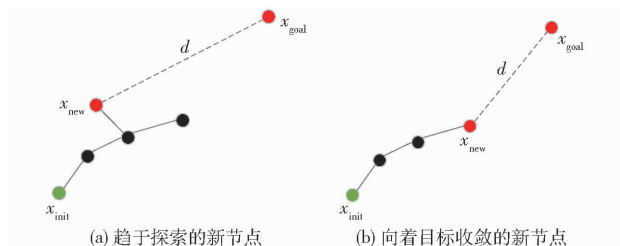


图 1 评价指数

Fig. 1 Evaluation index

2.2 扩展阶段划分

设定初始阈值为一常数 a , 根据评价指数是否达到阈值将随机树的扩展过程分为探索阶段和收敛

阶段。当评价指数小于阈值时,随机树为探索阶段,采样节点依旧为自由空间内随机点,如图 2a 所示。随着迭代次数的增加,当评价指数的值达到阈值时,说明该阶段趋于探索的点占比过大,随机树进入收敛阶段,在收敛阶段随机树以目标点作为采样节点进行扩展,如图 2b 所示。直至与障碍物发生碰撞,随机树结束收敛阶段,如图 2c 所示。随后评价指数回到初始值 1,随机树重新回到探索阶段,如图 2d 所示,当评价指数下一次达到阈值时随机树会再次进入收敛阶段。通过探索和收敛阶段的不断转换,使随机树在扩展过程中能够根据当前的探索情况实时修正采样节点的采样位置,减少盲目探索,提高算法的收敛速度。

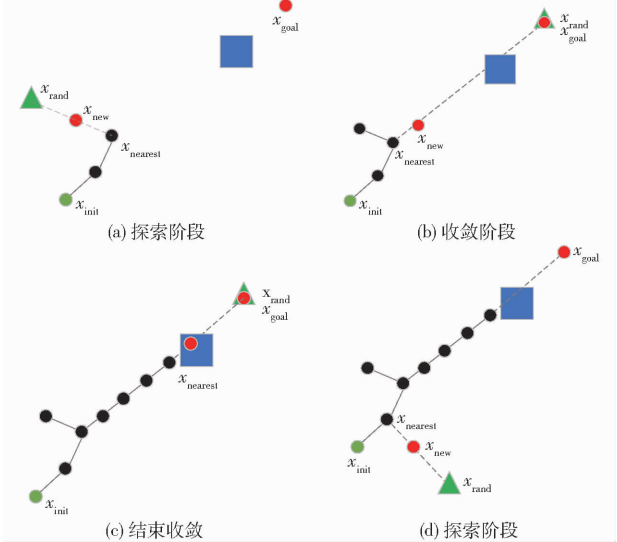


图 2 探索及收敛阶段

Fig. 2 Exploration and convergence phase

2.3 改进收敛阶段最近节点选择机制

每当随机树结束一次收敛阶段时,将开始进入收敛阶段时的最近节点 $x_{nearest}$ 以及在此次收敛阶段扩展的新节点 x_{new} 记录到点集 R 中,下一次随机树进入收敛阶段将不会选择点集 R 中的点作为最近节点 $x_{nearest}$ 。

如图 3a 所示,评价指数达到阈值随机树进入收敛阶段,随机树以节点 x 为最近节点向着目标点扩展,直到与障碍物发生碰撞,随机树结束收敛阶段,将节点 x 以及在此次收敛阶段扩展的新节点记录在点集 R 中,如图 3b 所示(点集 R 中的点被标记为黄色)。当评价指数达到阈值随机树再一次进入收敛阶段时如图 3c 所示,虽然节点 y 距离采样点最近,但在收敛阶段时点集 R 中的点不会被选为最近节点 $x_{nearest}$,所以点 z 作为最近节点 $x_{nearest}$ 向着目标方向扩展,直至与障碍物发生碰撞结束收敛阶段,将点 z 以及此次收敛阶段扩展的新节点记录到点集 R

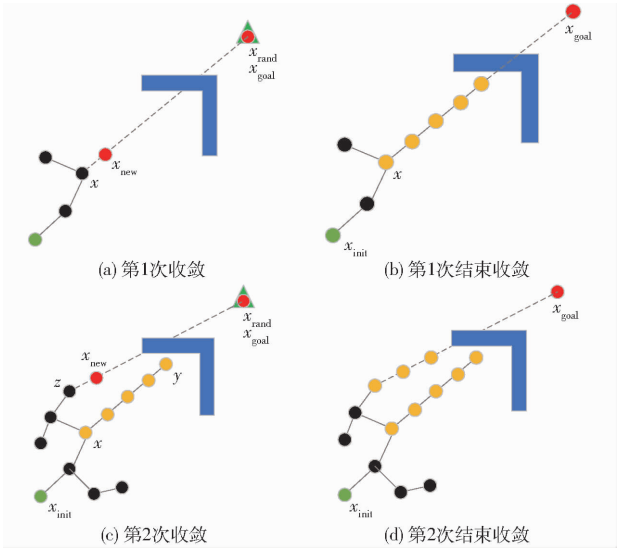


图 3 收敛阶段最近节点选择

Fig. 3 Selection of the nearest node in convergence stage

中,如图 3d 所示。通过改进收敛阶段的最近节点选择机制可避免在收敛阶段以重复的点作为最近节点,从而更快地避开不规则障碍,加快算法收敛速度。

2.4 动态阈值设定

Straight-RRT 通过评价指数是否达到阈值来判断当前随机树的扩展情况从而调整采样方式,对于简单的环境,不需要随机树进行过多的探索所以较小的阈值可以更快地找到路径,对于障碍物较多的复杂环境需要更大的阈值以便随机树对环境进行足够的探索。为了使算法适应各种环境,本文引入动态阈值,将随机树进入收敛阶段的次数作为实时评判环境复杂度的标准,阈值 T 随着随机树的探索过程进行自适应调整,阈值 T 表达式为

$$T = a + n \tag{1}$$

式中 n ——当前随机树进入收敛阶段的次数

如图 4a 所示,当前随机树进入收敛阶段的次数为 1,则当前阈值为 $T = a + 1$,如图 4b 所示,随机树进入收敛阶段的次数为 2,则当前阈值为 $T = a + 2$ 。动态阈值在随机树扩展过程中实时变化使随机树在复杂的环境下能够进行足够的探索,从而提高算法对不同环境的自适应性。

2.5 路径优化

RRT 算法因其随机采样的特性导致其规划的路径产生较多冗余点,需要对其进行优化处理降低路径复杂度^[21]。首先将路径起始点作为修剪起始点,依次检测路径后续节点与修剪起始点相连是否与障碍物发生碰撞,如果无碰撞则删除两节点之间的路径节点并直接将两节点相连,随后将新得到的路径中的第 2 个点作为修剪起始点,再次执行上述操作,直至目标点成为修剪起始点,结束修剪过程得

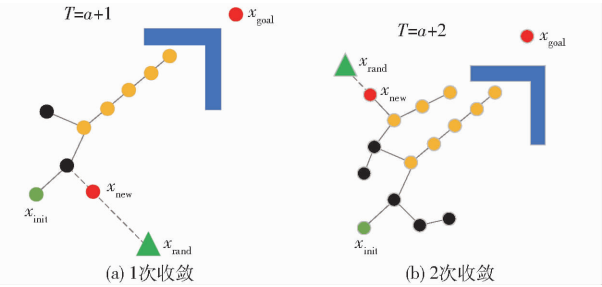


图 4 动态阈值
Fig. 4 Dynamic threshold

到最终路径。如图 5a 所示,初始路径为 1-2-3-4-5-6-7-8-9,修剪后路径为 1-5-7-9。但修剪后路径依旧存在转折点,为使路径更加平滑,对路径上每个转折点处进行二次贝塞尔曲线拟合,定义拟合空间 $n+1$ 个点的位置 $P_i (i=0,1,2,\cdots,n)$,则 n 次贝塞尔曲线表达式为

$$P(\alpha) = \sum_{i=0}^n P_i B_{i,n}(\alpha) \quad (\alpha \in [0,1]) \quad (2)$$

其中

$$B_{i,n}(\alpha) = C_n^i \alpha^i (1-\alpha)^{n-i} = \frac{n!}{(n-i)! i!} \alpha^i (1-\alpha)^{n-i} \quad (3)$$

式中 P_i ——贝塞尔曲线的控制点
 α ——贝塞尔曲线参数
 $B_{i,n}(\alpha)$ —— n 次 Bernstein 基函数

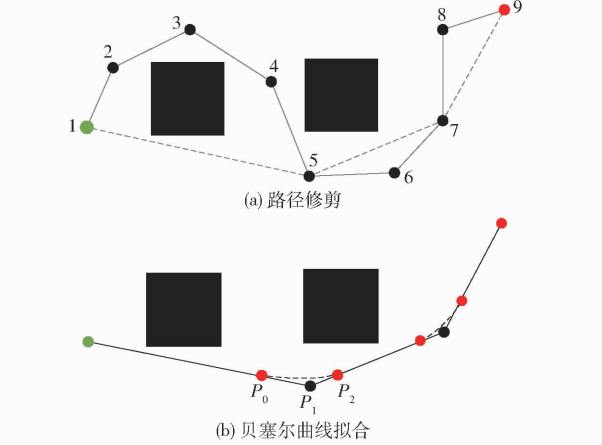


图 5 路径优化示意图
Fig. 5 Schematic of path optimization

因二次贝塞尔曲线需 3 个点进行拟合,为了保

持拟合曲线和原轨迹曲线一致性,如图 5b 所示,设每个转折点是 P_1 ,在转折点两侧线段上分别选取两点 P_0 及 P_2 作为贝塞尔曲线的控制点,且两点与转折点的距离为其所在线段长度的 $1/10$,对每处转折点进行二次贝塞尔曲线拟合,最后将多段贝塞尔曲线进行拼接,得到一条复杂度低且平滑的路径。

3 算法仿真与实验

3.1 不同环境下仿真实验

为验证改进算法的有效性,将 Straight-RRT 算法与 RRT 算法、文献[14]中的目标引力-RRT 算法(引力系数取 0.5)、RRT-Connect 算法在如图 6 所示环境下进行对比仿真实验,仿真实验中对 4 种算法均进行路径优化处理,4 种算法搜索步长取 2, Straight-RRT 初始阈值 a 取 5,评价指数增、减量 i 、 e 分别取 1、0.5,地图 1、2、3 规模均为 $[70,70]$,在 3 种环境下每种算法分别进行 100 次独立运行。算法运行环境为 64 bit Windows 10,处理器 Intel(R) Core(TM) i5-6300HQ,内存 4 GB。

如图 6 所示,地图 1 为障碍物相对较少的环境,用于检验算法在简单环境下的搜索能力,起始点位置为 $[5,35]$,目标点位置为 $[65,35]$,地图 2 为多障碍物排列杂乱的环境,用于检验算法在多障碍物环境下的搜索能力,起始点位置为 $[5,5]$,目标点位置为 $[65,65]$,地图 3 为迷宫环境,地图起始点位置为 $[15,7]$,目标点位置为 $[65,65]$ 。算法运行效果如图 7~9 所示。从图 7~9 可以看出,4 种算法均能完成起始点到目标点的全局路径规划,且经过路径优化处理后均能获得较为平滑的路径。其中本文改进的 Straight-RRT 算法的采样节点数比其他算法明显减少,4 种算法 100 次运行的平均搜索时间、平均迭代次数如表 1 所示。

从表 1 可以看出,在 3 种地图环境下 Straight-RRT 的搜索时间及迭代次数均小于其他规划算法,在地图 1 环境下 Straight-RRT 算法相比于 RRT 算法、目标引力-RRT 算法及 RRT-Connect 算法的平均搜索速度分别提升了 88.23%、74.35%、16.66%,迭代次数缩减了 65.77%、49.53%、

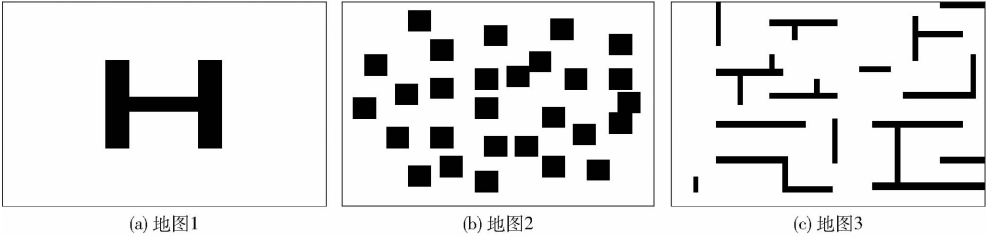


图 6 仿真环境

Fig. 6 Simulation environment

38.76%。地图 2 环境下平均搜索速度分别提升 90.35%、74.11%、12.00%，迭代次数缩减 77.15%、60.82%、27.10%。地图 3 环境下平均搜索速度分别提升 79.38%、49.15%、25.30%，迭代次数缩减 52.51%、40.76%、40.35%。4 种算法的初始路径及路径优化后的路径长度如表 2 所示。

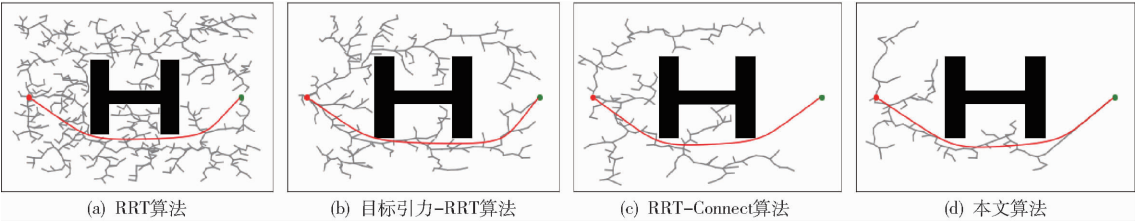


图 7 地图 1 算法运行效果
Fig. 7 Algorithm operation effect of map 1

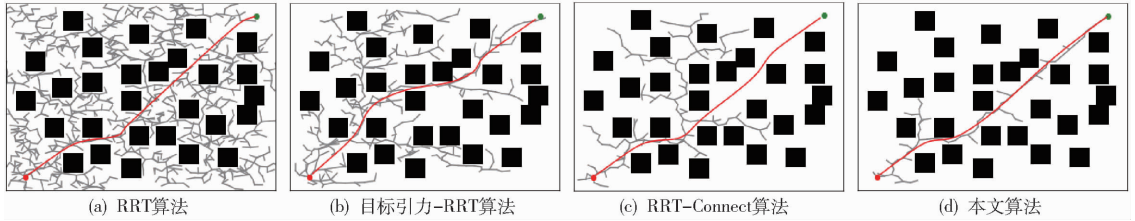


图 8 地图 2 算法运行效果
Fig. 8 Algorithm operation effect of map 2

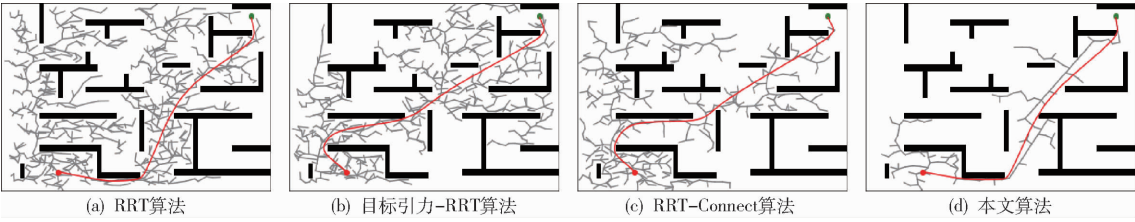


图 9 地图 3 算法运行效果
Fig. 9 Algorithm operation effect of map 3

表 1 不同环境下算法仿真结果

地图	搜索时间/s				迭代次数			
	RRT 算法	目标引力-RRT 算法	RRT-Connect 算法	本文算法	RRT 算法	目标引力-RRT 算法	RRT-Connect 算法	本文算法
1	0.17	0.078	0.024	0.020	637	432	356	218
2	0.228	0.085	0.025	0.022	871	508	273	199
3	0.587	0.238	0.162	0.121	1 628	1 305	1 296	773

表 2 路径优化前后对比结果

地图	初始路径长度				优化后路径长度			
	RRT 算法	目标引力-RRT 算法	RRT-Connect 算法	本文算法	RRT 算法	目标引力-RRT 算法	RRT-Connect 算法	本文算法
1	99.75	88.89	107.05	95.17	77.51	77.31	76.22	76.98
2	115.85	106.06	108.79	106.47	95.86	93.86	91.20	93.52
3	127.08	116.21	126.67	120.08	100.31	98.96	102.72	99.22

4 种算法 100 次仿真实验中迭代次数如图 10 所示。从图 10 可以看出,本文算法在保持更少迭代次数的同时,分布结果也较为集中。综合上述仿真实验可知,本文改进的 Straight-RRT 算法有效地提高了 RRT 算法的收敛速度、路径质量及稳定性。

4 种算法在不同地图环境下,总体初始路径的平均长度分别为 97.71、109.29、122.50,优化后路径的平均长度为 77.00、93.61、100.30,平均路径缩减了 21.19%、14.34%、18.12%。进一步说明了路径优化有效地减小了 4 种算法初始路径的复杂度。

3.2 猕猴桃果园环境场地实验
基于西北农林科技大学眉县猕猴桃试验站环境进行路径规划实验,果园为棚架式种植模式,内部环境如图 11 所示,横向间隔为 4 m,列向间隔 2 m,高 1.8 m,为避免猕猴桃采摘机器人行至地膜区域,将

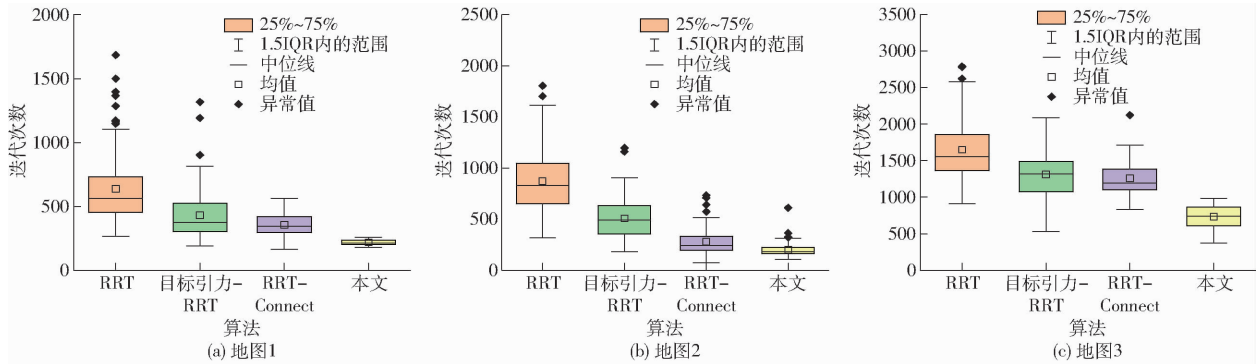


图 10 迭代次数箱线图
Fig. 10 Number of iterations boxplot

地膜区域设为障碍物,此外为保证机器人安全移动,对障碍物边界进行膨胀处理,二维环境地图如图 12 所示,图中黑色区域为障碍物区域,蓝色区域为膨胀区域。

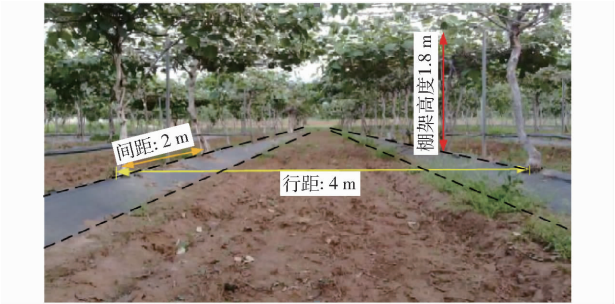


图 11 眉县猕猴桃试验站环境
Fig. 11 Environment of Meixian kiwifruit test station

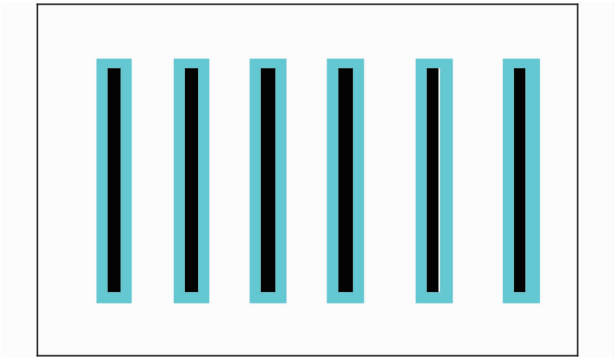


图 12 猕猴桃果园二维地图
Fig. 12 Two-dimensional map of kiwifruit orchard

猕猴桃采摘机器人如图 13 所示,由履带式移动底盘搭载 6 自由度 UR5 机械臂组成,机械臂工作空间为半径 0.85 m 的半球区域,机械臂底座高度为 1.1 m。结合棚架高度,计算出猕猴桃采摘机器人实际有效采摘区域为图 14 中左侧所示的黄色区域,将该区域投影至二维平面,计算出有效采摘区域在二维平面的投影为一个半径为 0.48 m 的圆形区域,其最大内接正方形边长为 0.68 m,见图 14 右侧。

为实现猕猴桃采摘机器人有效采摘区域对猕猴桃果园的全覆盖,对猕猴桃果园进行分区处理。将果园每个行间区域设置为单独的工作区,在每个工



图 13 猕猴桃采摘机器人
Fig. 13 Kiwifruit harvesting robot

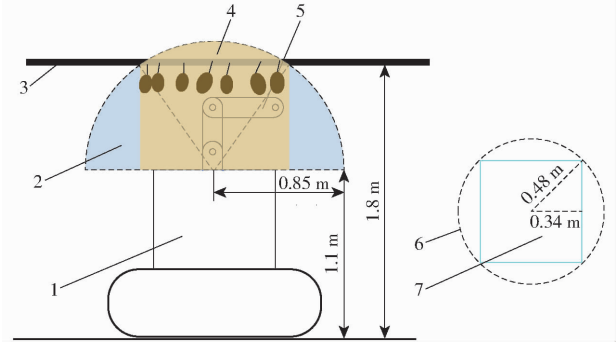


图 14 猕猴桃采摘机器人有效采摘区域
Fig. 14 Effective harvesting area of kiwifruit harvesting robot

1. 履带式移动底盘 2. 机械臂工作空间 3. 棚架 4. 有效采摘区域 5. UR5 机械臂 6. 有效采摘区域二维平面投影 7. 有效采摘区域二维平面投影的最大内接正方形

作区内,取猕猴桃采摘机器人有效采摘区域二维平面投影的最大内接正方形,依次排布直至铺满整个工作区,并依次进行编号,如图 15 所示。其中每个正方形区域的中心点即为采摘机器人的一个导航目标点,在一个工作区内,猕猴桃采摘机器人以 1 号点为起始点,按编号顺序依次经过该工作区内所有导航目标点后,随后移动到下一个工作区的 1 号点,继续按编号依次遍历每个导航点,直至遍历所有工作区的导航目标点,则认为猕猴桃采摘机器人的有效采摘区域完成了对猕猴桃果园的全覆盖。

选取 2 个工作区进行路径规划实验,为验证算

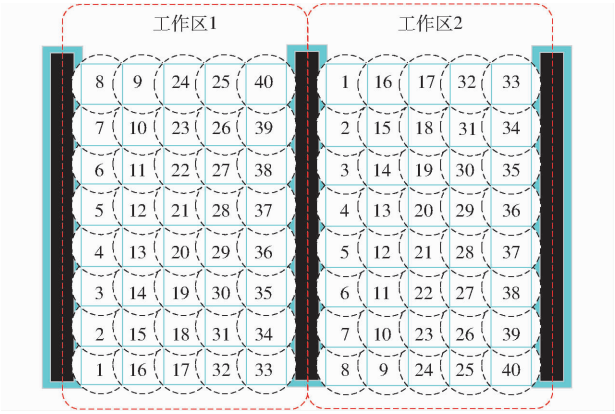


图 15 采摘工作区域划分

Fig. 15 Division of harvesting work area

法对复杂环境的适应性,对工作区 1 随机增加一些障碍物。地图环境如图 16a 所示,红色点为一个工作区中的起始导航点,绿色点为最终导航点,黄色点为猕猴桃采摘机器人需要依次经过的中间导航点。采用本文算法在此地图环境下进行路径规划实验,路径规划效果如图 16b 所示,其中红色线段、蓝色线段、紫色线段分别为猕猴桃采摘机器人在工作区 1、工作区 2、工作区间的规划路径。从图 16 可以看出,规划路径从工作区 1 的起始导航点开始依次遍历了每一个导航点,同时具有较低的路径复杂度,可以实现猕猴桃采摘机器人有效采摘区域对猕猴桃果

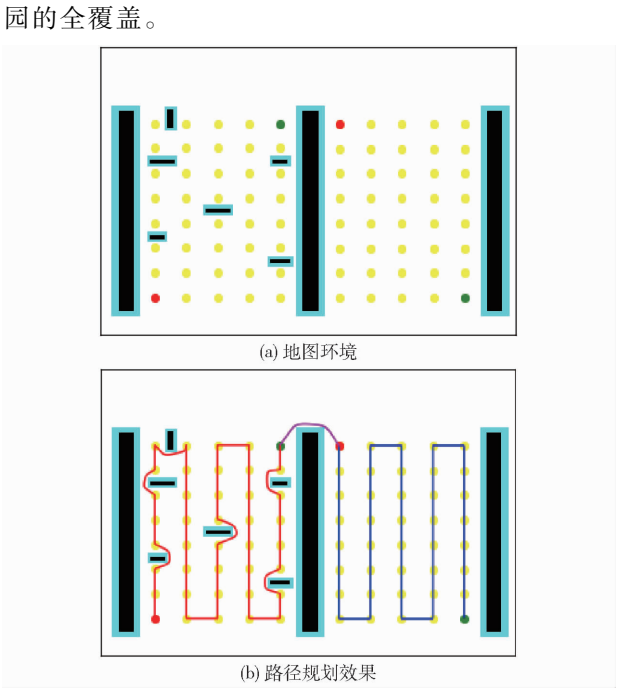


图 16 地图环境及路径规划效果

Fig. 16 Map environment and path planning effect

使用 RRT 算法、目标引力-RRT 算法、RRT-Connect 算法分别在此地图环境下进行路径规划实验,总路径及工作区 1、工作区 2、工作区间路径规划结果如表 3 所示。

表 3 猕猴桃果园环境路径规划结果

Tab. 3 Planning results of environmental paths in kiwifruit orchards

路径	搜索时间/s				迭代次数			
	RRT 算法	目标引力-RRT 算法	RRT-Connect 算法	本文算法	RRT 算法	目标引力-RRT 算法	RRT-Connect 算法	本文算法
总路径	4.703	2.249	1.646	1.208	16 741	7 244	3 188	2 256
工作区 1	2.769	1.609	1.005	0.729	10 538	4 986	1 699	1 522
工作区 2	1.535	0.440	0.371	0.300	5 065	1 627	502	516
工作区间	0.399	0.200	0.270	0.179	1 138	631	987	218

在工作区 1 中,本文算法搜索时间相比于其他 3 种算法分别缩减了 73.67%、54.69%、27.46%,说明本文算法在工作区中存在障碍物的情况下具有更强的适应性及搜索能力。由于工作区 2 中不存在障碍物,4 种算法在工作区 2 中的搜索效率均大幅提升,除 RRT 算法外其他 3 种算法搜索时间均保持在 0.5 s 以内。工作区间路径为从工作区 1 的最终导航点到工作区 2 的起始导航点的路径,在工作区间路径中 4 种算法的平均搜索时间为 0.262 s,本文算法相比于平均搜索时间缩短了 31.67%,说明本文算法在工作区间的路径规划上具有更高的搜索效率。从总体路径来看,本文算法的搜索时间相比其他 3 种算法分别缩减了 74.31%、46.28%、26.60%,迭代次数缩减了 86.52%、68.85%、29.23%。实验结果表明本文改进算法在猕猴桃果园环境中具有更

好的适应性及规划效率。

4 结论

- (1)引入评价指标,通过对新节点的分类实时更新评价指标。基于评价指标与阈值划分采样阶段,通过改变不同阶段的采样策略实时引导随机树的扩展。
- (2)优化收敛阶段的最近节点选择机制,使随机树更快避开不规则障碍物。
- (3)引入动态阈值,阈值随着环境复杂度实时调整,提高算法对不同环境的自适应性。
- (4)对路径进行优化处理,去除路径冗余点并对路径进行二次贝塞尔曲线拟合。
- (5)基于棚架式猕猴桃果园环境进行路径规划实验,本文改进算法在猕猴桃果园环境下搜索时间

为 1.208 s,相比 RRT 算法、目标引力-RRT 算法及 RRT-Connect 算法分别缩减了 74.31%、46.28%、26.60%,实验结果表明本文改进算法在猕猴桃果园环境中具有良好的适应性及规划效率。

参 考 文 献

[1] 周俊,何永强.农业机械导航路径规划研究进展[J].农业机械学报,2021,52(9):1-14.
ZHOU Jun, HE Yongqiang. Research progress on navigation path planning of agricultural machinery[J]. Transactions of the Chinese Society for Agricultural Machinery, 2021, 52(9): 1-14. (in Chinese)

[2] 曹如月,张振乾,李世超,等.基于改进 A* 算法和 Bezier 曲线的多机协同全局路径规划[J].农业机械学报,2021,52(增刊):548-554.
CAO Ruyue, ZHANG Zhenqian, LI Shichao, et al. Multi-machine cooperation global path planning based on A-star algorithm and Bezier curve[J]. Transactions of the Chinese Society for Agricultural Machinery, 2021, 52(Supp.): 548-554. (in Chinese)

[3] 谭建豪,潘豹,王耀南,等.基于改进 RRT* FN 算法的机器人路径规划[J].控制与决策,2021,36(8):1834-1840.
TAN Jianhao, PAN Bao, WANG Yaonan, et al. Robot path planning based on improved RRT* FN algorithm[J]. Control and Decision, 2021, 36(8): 1834-1840. (in Chinese)

[4] 褚凯轩,常天庆,王全东,等.基于有效拐点和最短最小路径的蚁群路径规划方法[J].农业机械学报,2021,52(12):400-407.
CHU Kaixuan, CHANG Tianqing, WANG Quandong, et al. Path planning method of ant colony algorithm based on effective turning point and shortest-minimum path[J]. Transactions of the Chinese Society for Agricultural Machinery, 2021, 52(12): 400-407. (in Chinese)

[5] 李洋,徐达,周诚.基于自适应步长 RRT 的双机器人协同路径规划[J].农业机械学报,2019,50(3):358-367.
LI Yang, XU Da, ZHOU Cheng. Cooperation path planning of dual-robot based on self-adaptive stepsize RRT[J]. Transactions of the Chinese Society for Agricultural Machinery, 2019, 50(3): 358-367. (in Chinese)

[6] 朱大奇,颜明重.移动机器人路径规划技术综述[J].控制与决策,2010,25(7):961-967.
ZHU Daqi, YAN Mingchong. Survey on technology of mobile robot path planning[J]. Control and Decision, 2010, 25(7): 961-967. (in Chinese)

[7] 阮晓刚,周静,张晶晶,等.基于子目标搜索的机器人目标导向 RRT 路径规划算法[J].控制与决策,2020,35(10):2543-2548.
RUAN Xiaogang, ZHOU Jing, ZHANG Jingjing, et al. Robot goal guide RRT path planning based on sub-target search[J]. Control and Decision, 2020, 35(10): 2543-2548. (in Chinese)

[8] 王宇,王文浩,徐凡,等.基于改进蚁群算法的植保无人机路径规划方法[J].农业机械学报,2020,51(11):103-112.
WANG Yu, WANG Wenhao, XU Fan, et al. Path planning approach based on improved ant colony optimization for sprayer UAV[J]. Transactions of the Chinese Society for Agricultural Machinery, 2020, 51(11): 103-112. (in Chinese)

[9] 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 Cailian, LI Peng, FENG Yu. Path planning of greenhouse robot based on fusion of improved A* algorithm and dynamic window approach[J]. Transactions of the Chinese Society for Agricultural Machinery, 2021, 52(1): 14-22. (in Chinese)

[10] LAVALLE S M. Rapidly-exploring random trees: a new tool for path planning [R]. AMES: Computer Science Department, Iowa State University, 1998.

[11] 宋晓茹,任怡悦,高高,等.移动机器人路径规划综述[J].计算机测量与控制,2019,27(4):1-6.
SONG Xiaoru, REN Yiyue, GAO Song, et al. Survey on technology of mobile robot path planning[J]. Computer Measurement & Control, 2019, 27(4): 1-6. (in Chinese)

[12] 林依凡,陈彦杰,何炳蔚,等.无碰撞检测 RRT* 的移动机器人运动规划方法[J].仪器仪表学报,2020,41(10):257-267.
LIN Yifan, CHEN Yanjie, HE Bingwei, et al. Non-collision checking RRT* algorithm for mobile robot motion planning[J]. Chinese Journal of Scientific Instrument, 2020, 41(10): 257-267. (in Chinese)

[13] 刘成菊,韩俊强,安康.基于改进 RRT 算法的 RoboCup 机器人动态路径规划[J].机器人,2017,39(1):8-15.
LIU Chengju, HAN Junqiang, AN Kang. RoboCup robot dynamic path planning based on improved RRT algorithms[J]. Robot, 2017, 39(1): 8-15. (in Chinese)

[14] CAO X, ZOU X, JIA C, et al. RRT-based path planning for an intelligent litchi-picking manipulator[J]. Computers and Electronics in Agriculture, 2019, 156(2): 105-118.

[15] KUFFNER J J, LAVALLE S M. RRT-Connect: an efficient approach to single-query path planning[C]//Proceedings of the 2000 IEEE International Conference on Robotics and Automation, ICRA IEEE, 2000.

[16] KARAMAN S, FRAZZOLI E. Sampling-based algorithms for optimal motion planning[J]. The International Journal of Robotics Research, 2011, 30(7): 846-894.

[17] JEONG I B, LEE S J, KIM J H. Quick-RRT*: triangular inequality-based implementation of RRT* with improved initial solution and convergence rate[J]. Expert Systems with Applications, 2019, 123: 82-90.

[18] LI Y J, WEI W, GAO Y, et al. PQ-RRT*: an improved path planning algorithm for mobile robots[J]. Expert Systems with Applications, 2020, 152: 113425.

[19] TAHIR Z, QURESHI A H, AYAZ Y, et al. Potentially guided bidirectionalized RRT* for fast optimal path planning in cluttered environments [J]. Robotics and Autonomous Systems, 2018, 108: 13-27.

[20] 陈秋莲,蒋环宇,郑以君.机器人路径规划的快速扩展随机树算法综述[J].计算机工程与应用,2019,55(16):10-17.
CHEN Qiulian, JIANG Huanyu, ZHENG Yijun. Summary of rapidly-exploring random tree algorithm in robot path planning [J]. Computer Engineering and Applications, 2019, 55(16): 10-17. (in Chinese)

[21] 余伶俐,龙子威,周开军.基于贝塞尔曲线的机器人非时间轨迹跟踪方法[J].仪器仪表学报,2016,37(7):1546-1572.
YU Lingli, LONG Ziwei, ZHOU Kaijun. Non-time trajectory tracking method based on Bezier curve for robot[J]. Chinese Journal of Scientific Instrument, 2016, 37(7): 1546-1572. (in Chinese)