doi:10.6041/j.issn.1000-1298.2024.10.008

基于 PIB－RRTstar 的荔枝采摘机械臂运动规划方法

熊俊涛　陈浩然　姚兆燊　宁起鹏
（华南农业大学数学与信息学院，广州 510642）

摘要：为解决非结构环境下，采摘机械臂路径规划时存在的效率低、采摘成功率不高的问题，提出一种结合人工势场法的四向搜索 RRTstar 算法。首先通过人工势场法对空间进行分割，获取空间分割点 x_{split}，进行四向搜索；其次结合人工势场法引导随机采样，提高采样节点质量；然后基于节点历史扩展信息，添加信息因子，实现自适应动态步长扩展。最后通过贪婪回溯对生成路径进行优化。通过二维模拟实验、6 自由度机械臂下的仿真实验与采摘实验验证提出算法的有效性。二维仿真对比实验表明：相比于 RRTstar 算法，改进算法路径成本缩短 2.01%，时间代价降低 98.81%，迭代次数减少 92.49%。在机器人操作系统（Robot operating system，ROS）中进行 6 自由度机械臂下的仿真实验，相比于 RRTstar 算法，规划时间减少 93.17%，路径成本降低 36.62%，扩展节点数减少 97.00%。最后使用 6 自由度机械臂进行采摘实验，实验结果表明本文算法有效提升采摘成功率，采摘成功率达 85%，并在结合姿态约束方法后，采摘成功率达 95%。所提出的路径规划方法在路径规划时间上存在一定优势，采摘实验证明本文算法可提高荔枝采摘成功率。

关键词：荔枝；采摘机械臂；路径规划；RRTstar 算法；人工势场法
中图分类号：S282　　**文献标识码**：A　　**文章编号**：1000-1298(2024)10-0082-11　　**OSID**：

Motion Planning for Lychee Picking Manipulator Based on PIB－RRTstar Algorithm

XIONG Juntao　CHEN Haoran　YAO Zhaoshen　NING Qipeng
(*College of Mathematics and Information*, *South China Agricultural University*, *Guangzhou* 510642, *China*)

Abstract: In order to solve the problems of low efficiency and poor picking success rate in the path planning of picking robotic arm in unstructured environment, a four-way search RRTstar algorithm combined with artificial potential field method was proposed. Firstly, the space was segmented by the artificial potential field method to obtain the spatial segmentation point x_{split} for four-way search; secondly, the random sampling was guided by the artificial potential field method to improve the quality of the sampled nodes; then, based on the node history expansion information, the information factor was added to achieve the adaptive dynamic step size expansion. Finally, the generation path was optimized by greedy backtracking. The effectiveness of the proposed algorithm was verified by two-dimensional simulation experiments, simulation experiments under 6-degree-of-freedom robotic arm and picking experiments. The 2D simulation comparison experiment showed that compared with the RRTstar algorithm, the path cost of the improved algorithm was shortened by 2.01%, the time cost was reduced by 98.81%, and the sampling nodes were reduced by 92.49%. Simulation experiments under 6-degree-of-freedom robotic arm in robot operating system (ROS) showed that compared with RRTstar algorithm, the planning time was reduced by 93.17%, the path cost was reduced by 36.62%, and the number of expansion nodes was reduced by 97.00%. Finally, the picking experiment was carried out with a 6-degree-of-freedom robotic arm, and the experimental results showed that the algorithm effectively improved the picking success rate, which reached 85%, and after combining the attitude constraint method, the picking success rate reached 95%. The proposed path planning method had certain advantages in path planning time, and the picking test proved that the algorithm improved the success rate of lychee picking, and contributed to the development of lychee picking robot.

Key words: lychee; picking manipulator; path planning; RRTstar algorithm; artificial potential field method

收稿日期：2024－02－14　修回日期：2024－04－23
基金项目：国家自然科学基金项目(32071912)、广东省农业科技创新十大主攻方向“揭榜挂帅”项目(2022SDZG03)和广东省大学生科技创新培育专项资金项目(pdjh2023a0075)
作者简介：熊俊涛(1981—)，男，教授，博士，主要从事农业机器人智能认知研究，E-mail：xiongjt2340@163.com

0　引言

随着智慧农业的快速发展，柑橘、荔枝、番石榴等季节性水果的采摘过程正向自动化方向转型[1]。将机械臂应用于果蔬采摘成为研究热点。在自然环境下，果树生长形态呈无序性[2]，在这种非结构化的果树环境下，机械臂路径规划存在很大的挑战。机械臂路径规划常在高维度的关节空间中进行，其中基于快速搜索随机树（Rapid-exploration random tree，RRT）的采样算法，由于具有不需要对障碍物建模以及对高维空间探索的优良特性得到广泛应用，但 RRT 算法存在采样盲目、路径曲折、平滑度低等问题，针对这些问题，研究人员开展大量研究[3－10]。

RRTstar[11]算法是 RRT 算法发展中的里程碑式改进，被视为 RRT 最优秀的变体。在 RRTstar 的基础上，通过优化采样[12－13]、搜索树剪枝[14－15]、节点拒绝[16－17]以及双向采样[18－20]等策略提高路径获取效率，但这些方法没有有效利用探索空间中的障碍物信息。为解决这个问题，文献［21］提出 P－RRTstar，通过引入人工势场（Artificial potential field method，APF）[22]替代随机采样，有效利用环境信息，但存在受限于局部最优的问题。除以上针对采样过程的不足对采样方式进行修改外，近年还出现采用神经网络指导采样的优化方法[23]。

采摘机械臂在采摘作业时，不仅要考虑机械臂路径规划问题，也需要结合实际采摘环境和机械臂运动学特性进行应用优化。考虑到采摘机械臂运动学特性问题，文献［24］提出一种基于粒子群算法（PSO）的 APSO 算法完成机械臂逆运动学求解，并且提出 At－BiRRT 解决轨迹求解问题。文献［25］针对柑橘采摘机器人采摘困难问题，引入预采摘点分割采摘轨迹规划，加快轨迹获取。文献［26］将番茄采摘机器人工作空间分解为多个子模块，分别在每个子模块进行规划，获取路径。文献［27］对番茄机器人采摘空间进行切割，丢弃无法抵达的空间，保证采样过程的有效性。

针对 RRTstar 算法路径规划效率缓慢、轨迹冗长等问题，本文结合人工势场法对环境的敏感性，提出 PIB－RRTstar 算法。通过人工势场法获取空间分割点实现多向搜索；结合人工势场法引导采样降低采样盲目性；利用搜索树全局和局部的探索信息实现自适应步长；路径再优化进一步提高路径质量。并针对机械臂采摘作业时出现的摆臂问题，结合机械臂运动学特性，优化逆运动学求解，以期有效获取采摘姿态，提高采摘成功率。

1　PIB－RRTstar 算法

PIB－RRTstar 算法中使用一些变量与特殊函数，如表 1 所示。

1.1　空间分割与四树探索策略

传统的 RRTstar 算法采用单向搜索策略，这种方式需要探索更多的未知空间才能获取路径，增加搜索时间和计算复杂度。如图 1a 所示，灰色区域表示 RRTstar 算法预计探索空间。为改进这一缺陷，B－RRTstar 算法采用双向搜索策略，即同时使用两个随机扩展树，分别从起点和终点开始扩展，这种双向搜索方式显著减少所需探索的未知区域，获得更高的路径探索效率，如图 1b 所示。本文基于 B－RRTstar 的双树搜索优势，引入中间点 x_{split} 对地图空间进行分割。如图 1c 所示，引入分割点 x_{split} 后所需探索的未知空间进一步减少，算法能够在更少的采样次数下获取到初始路径。而对于分割点选取，常用策略往往简单地采用起点 x_{init} 与终点 x_{goal} 连线的中点作为空间分割点[28]，即

$$x_{\text{mid}} = \frac{x_{\text{init}} + x_{\text{goal}}}{2} \tag{1}$$

式中　x_{init}——起点位置　　x_{goal}——终点位置

x_{mid}——分割点位置

这种方法虽简单易行，但并未考虑到环境中障碍物的分布情况。障碍物分布复杂时，这种方法可能导致路径规划算法无法有效避开障碍物或找到最优路径。鉴于人工势场法对障碍物分布的敏感性，本文提出一种基于人工势场法的空间分割点求取策略，即

$$x_{\text{split}} = x_{\text{goal}} + \sum_{i=1}^{n} \sqrt{S(obs_i)} \frac{\frac{x_{\text{init}} + x_{\text{goal}}}{2} - \text{center}(obs_i)}{\text{distance}\left(obs_i, \frac{x_{\text{init}} + x_{\text{goal}}}{2}\right)} \tag{2}$$

式中　$S(obs_i)$——障碍物 obs_i 面积

$\text{distance}\left(obs_i, \frac{x_{\text{init}} + x_{\text{goal}}}{2}\right)$——$obs_i$ 与 x_{mid} 距离

$\text{center}(obs_i)$——障碍物 obs_i 中心点位置

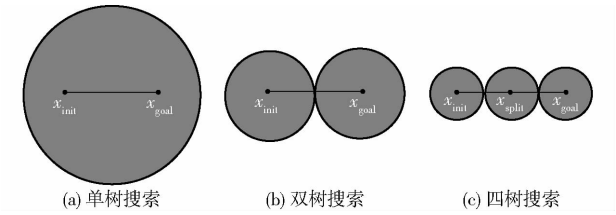
该策略不仅考虑起点和终点之间的位置关系，还充分考虑障碍物对分割点位置的影响。通过计算障碍物大小和障碍物与中点距离，可得每个障碍物对分割点的斥力势场。这些斥力势场共同作用于分割点，引导其移动到更合适的位置。具体来说，当空间中无障碍物时，分割点则为终点；此时将直接生成起点 x_{init} 通向目标点 x_{goal} 的路径。而当存在障碍物时，障碍物将产生斥力势场，影响分割点位置。斥力

表 1 变量与函数列表

Tab. 1 List of variables and functions

变量/函数	含意
x_{init}	起点
x_{split}	基于人工势场法获取的空间分割点
x_{goal}	终点
rad	初始扩展步长
MaxSample	最大采样次数,超过此值则算法停止
Flag	标记位,为 0 则表示探索 x_{init} 与 x_{split} 间的路径,为 1 则表示探索 x_{split} 与 x_{goal} 间的路径
V	点集,存储树节点
E	边集,存储树节点间的连接关系
T	$T=\{V,E\}$,T 代表随机搜索树,B-RRTstar 存在两棵搜索树,则分别用 T_1、T_2 表示
NodeRecord	树节点信息集合,存储所有树节点,以及树节点最新扩展步长、树节点已连接节点数、扩展失败次数
Path	存储路径,初始为空
i	累积迭代次数,每轮循环加一
k	存储树节点个数,扩展成功时加一
x_{rand}	存储获取的采样点
$x_{nearest}$	随机树中距离 x_{rand} 最近的树节点,通过 Nearest 函数获取
Newrad	自适应步长变量,通过自适应步长计算函数 GetStep 获取
RGD	势场采样函数,通过 Flag 进行调节,当探索 x_{init} 与 x_{split} 间的路径将以更贪婪的方式进行;探索 x_{split} 与 x_{goal} 间的路径则额外添加障碍物信息
Nearest	函数,寻找距离 x_{rand} 最近的树节点
Steer	函数,获取 $x_{nearest}$ 向 x_{rand} 延伸 Newrad 所形成的节点
x_{new}	存储 Steer 返回的节点,该节点为本文扩展的新节点
CollsionFree	碰撞检测函数,检测 $x_{nearest}$ 和 x_{new} 的连线间是否有障碍物,若无则返回 True,否则返回 Flase
Rnear	常量,控制 Chooseparent 和 Rewire 的搜索范围
Near	函数,寻找搜索树中和 x_{new} 距离小于 Rnear 的节点集合
X_{near}	点集,存储搜索树中和 x_{new} 距离小于 Rnear 的节点
Chooseparent	在 X_{near} 中寻找能使 x_{new} 至 x_{init} 路径代价更低的父节点
Rewire	树结构重构函数,测试 x_{new} 作为 X_{near} 中节点的父节点时,是否能够降低该节点至 x_{init} 的路径代价
$x_{connect}$	存储另一棵搜索树中与 x_{new} 距离小于 R_{near} 的节点,可为空
Update	使用本轮循环的变量信息更新 NodeRecord
Connect_tree	双树连接函数,当两棵搜索树存在连接点 $x_{connect}$ 时触发
Swap	双树交换函数,使得两棵搜索树节点个数保持平衡

势场强度与障碍物尺寸成正比,和障碍物与分割点



(a) 单树搜索 (b) 双树搜索 (c) 四树搜索

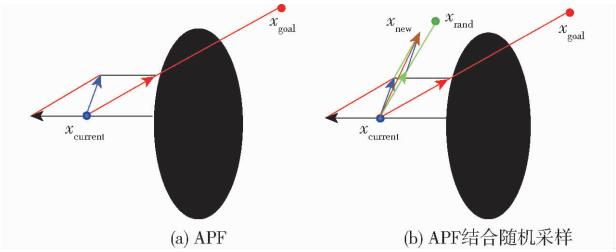
图 1 多向搜索范围示意图

Fig. 1 Schematics of multi-directional search range

间距成反比。同时,障碍物中心点用于计算斥力势场向量作用方向。选取分割点后,使用 $Flag_1$ 和 $Flag_2$ 进行标记,分别表示起点与分割点间以及分割点与终点间的路径探索区域。最后,基于分割点进行四树探索。这 4 个随机扩展树分别为起点到分割点、分割点到起点、分割点到终点、终点到分割点进行搜索。这种四树搜索的方式能够进一步提高算法探索效率,减少搜索时间,从而更快地找到 1 条从起点到终点的可行路径。

1.2 基于人工势场法的采样策略优化

人工势场法作为一种有效的路径规划策略,其核心在于通过构建虚拟的引力势场和斥力势场来引导质点从初始位置安全地到达目标位置。如图 2a 所示,质点 $x_{current}$ 在终点 x_{goal} 产生的引力势场和黑色障碍物产生的斥力势场共同作用下,能够避开障碍物并朝着终点移动。然而人工势场法却存在两个严重缺陷。首先,当障碍物产生的斥力势场与目标点产生的引力势场达到平衡时,质点将陷入局部最小值,导致其在该位置停滞不前,无法继续向目标点移动。其次,当质点进入封闭障碍物区域时,使用人工势场法可能在该区域内进行大量重复且无意义探索。为克服这些缺陷,本文将人工势场法与 RRTstar 算法相结合。如图 2b 所示,通过 RRTstar 在每轮迭代中获取的随机采样点,形成随机方向的引力势场,当质点陷入局部最小值时,RRTstar 算法可以通过随机采样点产生的引力势场来打破平衡状态,使质点能够继续向目标点移动,而在封闭障碍物区域中,随机采样点产生的随机引力势场将有概率引导质点向未知方向移动,减少在封闭区域内进行无意义探索。而人工势场法的引入也使得 RRTstar 具有更强的目标导向性,有助于减少盲目的随机采样带来的



(a) APF (b) APF结合随机采样

图 2 引入 APF 的算法改进策略

Fig. 2 Algorithm improvement strategy of introduced APF

无效扩展，加快算法收敛效率并且获取更高质量的路径。

1.3　自适应步长策略

基本 RRTstar 算法中以固定步长对空间进行探索，这个固定步长会影响在探索通畅空间时随机树扩展速度。为加快算法探索速度，提出了动态步长调整函数，即算法不再以固定步长 rad 进行探索，探索步长 steer 计算公式为

$$rad_N=\begin{cases}\dfrac{t}{s}\left[\alpha n_1-\beta n_2+\dfrac{(n_1-n_2)t}{s}\right]+\dfrac{(s-t)rad_0}{s}+rad_{N-1} & \left(\dfrac{n_1}{n_2}>rad_0\dfrac{t}{s}\right)\\ rad_{N-1}-\dfrac{n_2}{n_1} & \left(\dfrac{t}{s}rad_0\geqslant\dfrac{n_1}{n_2}\geqslant 1\right)\\ rad_0 & \left(\dfrac{n_2}{n_1}<1\right)\end{cases}\tag{3}$$

式中　rad_N——第 N 轮的扩展步长

t——搜索树节点数量

s——总探索数

n_1——x_{nearest} 节点连接数

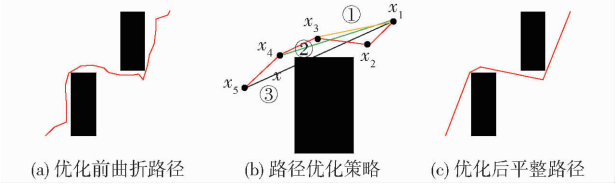
n_2——x_{nearest} 累积扩展失败数

α、β——常系数

通过式(3)将探索步长选取划分为 3 个子区间，第 1 个子区间表明本轮迭代的 x_{nearest} 具有良好的扩展性，当前探索区域障碍物分布稀疏，步长增加，加快扩展。第 2 个子区间表明对于 x_{nearest} 曾扩展失败，但 x_{nearest} 扩展失败数远小于搜索树 T 历史平均扩展失败数，意味着整个环境存在一定数量障碍物，因此将探索步长缓慢减少，防止继续碰撞。第 3 个子区间表明 x_{nearest} 累积扩展失败数过大，此时 x_{nearest} 周围有大量障碍物，因此将当前扩展步长跳变为初始步长，在复杂障碍物环境下缓慢探索。基于上述策略，随机搜索树在无障碍区域将快速扩展，而在障碍物复杂区域将减缓探索速度，注重逃离当前区域。

1.4　路径优化策略

RRTstar 系列算法以随机搜索树为核心，通过迭代的方式向采样点方向扩展树节点，能够有效地探索空间并寻找从起点 x_{init} 到终点 x_{goal} 的无碰撞路径。此时生成的路径中往往存在冗余节点，影响路径质量，如图 3a 所示，为消除这些冗余节点，可通过回溯寻找无碰撞的最远祖先节点，实现路径优化。具体过程如图 3b 所示，其中红色为优化前路径。x_1 通过循环不断向更远的祖先节点搜索。在第 1 轮循环中，x_1 尝试与 x_3 连接，并成功找到 1 条无碰撞的路径。在第 2 轮循环中，x_1 尝试与更远的节点 x_4 连接，同样未发生碰撞。然而，在第 3 轮循环中，当 x_1 尝试与 x_5 连接时，发生碰撞，此时算法会回退到上一轮，并记录节点 x_4 为可连接的最远祖先节点。之后树结构进行重组，节点 x_1 直接与 x_4 相连。下一轮循环将由 x_4 尝试向祖先节点探索，直到回溯至起点 x_{init}，图 3c 为路径优化后平整路径。通过这种方式，路径中的冗余节点得以删除，从而减小路径总代价。



(a) 优化前曲折路径　(b) 路径优化策略　(c) 优化后平整路径

图 3　路径优化对比

Fig. 3　Comparison of path optimization

1.5　PIB－RRTstar 算法建立

通过节点自适应步长加快探索速度；引入人工势场法进行采样优化，引导采样方向；基于空间分割和四树搜索缩减探索空间范围；使用向祖先节点回溯的路径优化策略减少路径代价。基于以上策略，提出 PIB－RRTstar 算法。算法总流程如图 4 所示，首先输入已有环境信息，采用式(3)获取分割点 x_{split}，开启起点 x_{init} 与分割点 x_{split} 间的双向搜索，分割点 x_{split} 与 x_{goal} 间的双向搜索，两种搜索通过线程池实现并行搜索。当获取路径后，合成由 x_{init} 经 x_{split} 至 x_{goal} 的路径 path $=\{x_{\text{init}},x_1,x_2,\cdots,x_{\text{split}},\cdots,x_{\text{goal}}\}$，开启路径优化，由 x_{goal} 向 x_{init} 回溯，剔除 path 中存在的冗余树节点。PIB－RRTstar 算法具体实现如伪代码所示：首先由起点和终点构建随机探索树 T_1、T_2。初始化必要参数信息。当循环条件满足时，执行以下操作：使用势场函数 RGD 引导采样，获取采样点 x_{rand}(6～9 行)，通过函数 Nearest 找到搜索树中距离 x_{rand} 最近的点 x_{nearest}，使用步长计算函数 GetStep 获取扩展步长 Newrad，最后使用扩展函数 Steer，由 x_{nearest} 向 x_{rand} 扩展，获得 x_{new}(10～12 行)。获取 x_{new} 后，进行碰撞检测 CollisionFree，若 x_{nearest} 扩展 x_{new} 时与障碍物产生碰撞，则本轮探索失败，记录树节点 x_{nearest} 扩展失败(22～23 行)，反之，则本轮探索成功，之后使用 Chooseparent 函数为 x_{new} 寻找代价更低的父节点，x_{new} 加入到搜索树中，树节点个数 t 加 1(14～18 行)。而后尝试基于 x_{new} 对搜索树中其他节点进行优化(19 行)。算法本轮搜索和优化任务完成，尝试获取连接点 x_{connect} 连接搜索树 T_1 和 T_2(20 行)。若连接点 x_{connect} 成功获取，则使用 Connect_tree

连接两棵搜索树,生成路径 Path 并返回(25 ~ 29 行),否则交换搜索树(31 行),继续下一轮迭代,直到循环结束。

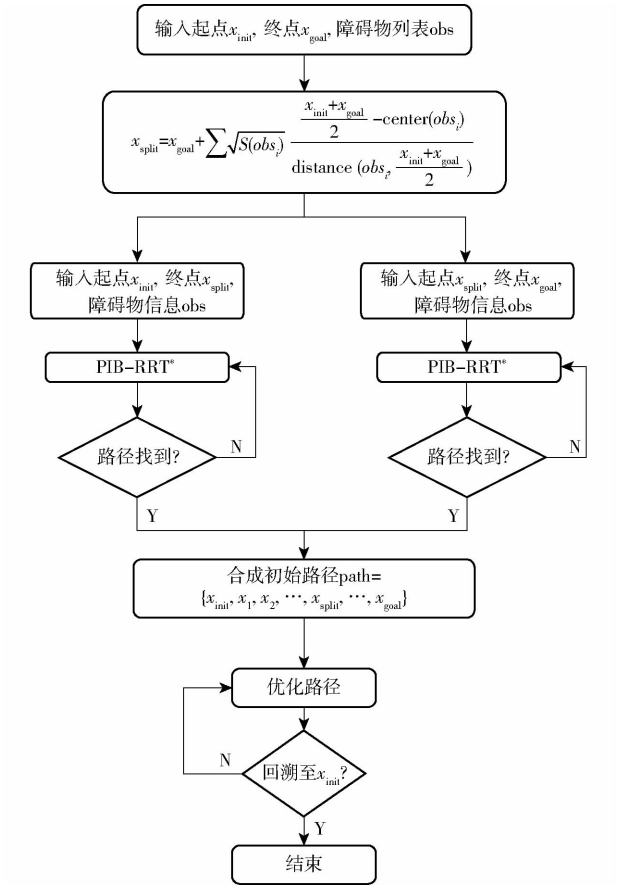


图 4 PIB - RRTstar 算法流程图

Fig. 4 Flowchart of PIB - RRTstar algorithm

PIB - RRTstar 算法伪码为

Input: x_{init}, x_{goal}, Obstacles. MaxSample, rad, Flag

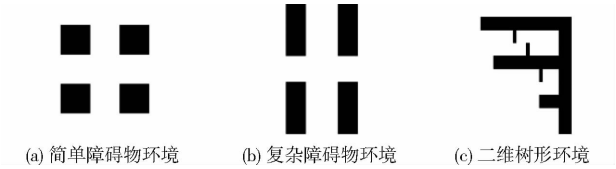
Output: Path

1. $V_1 \leftarrow \{x_{init}\}$; $E_1 \leftarrow \varnothing$; $T_1 = \{V_1, E_1\}$
2. $V_2 \leftarrow \{x_{goal}\}$; $E_2 \leftarrow \varnothing$; $T_2 = \{V_2, E_2\}$
3. NodeRecord $\leftarrow$ {(x_{init}, rad, 0, 0), (x_{goal}, rad, 0, 0)}; Path $\leftarrow \varnothing$; i $\leftarrow$ 0; k $\leftarrow$ 0;
4. while i $\leqslant$ MaxSample do
5. i = i + 1;
6. if Flag = = 0:
7. $x_{rand} \leftarrow$ RGD(x_{goal}, Sample());
8. else
9. $x_{rand} \leftarrow$ RGD(x_{goal}, Obstacles, Sample());
10. $x_{nearest} \leftarrow$ Nearest(V_1, x_{rand})
11. NewRad $\leftarrow$ GetStep(NodeRecord, $x_{nearest}$, i, k)
12. $x_{new} \leftarrow$ Steer($x_{nearest}$, x_{rand}, NewRad)
13. if CollisionFree($x_{nearest}$, x_{new}, Obstacles) then
14. $X_{near} \leftarrow$ Near(V_1, x_{new}, R_{near});
15. $x_{parent} \leftarrow$ Chooseparent(X_{near}, x_{new}, $x_{nearest}$)
16. $V_1 \leftarrow V_1 \cup \{x_{new}\}$
17. $E_1 \leftarrow E_1 \cup \{(x_{parent}, x_{new}\}$
18. $T_1 \leftarrow$ Rewire(T_1, x_{new}, X_{near})
19. $x_{connect} \leftarrow$ Near(T_2, x_{new}, R_{near})
20. Update(NodeRecord, $x_{nearest}$, NewRad, 1, 0)
21. else
22. Update(NodeRecord, $x_{nearest}$, NewRad, 0, 1)
23. end If
24. If $x_{connect} \neq \varnothing$ then
25. Path $\leftarrow$ Connect_tree(T_1, T_2, x_{new}, $x_{connect}$)
26. end If
27. If Path $\neq \varnothing$ then
28. Return Path
29. else
30. Swap(T_1, T_2)
31. end If
32. end while
33. return Path

2 二维空间下仿真验证

为验证所提出算法的有效性,使用 PyCharm 软件平台,采用 Python 编程语言进行二维环境下算法仿真模拟。本实验设定初始步长 rad 为 10,搜索半径 R_{near}为 20,并设定最大迭代次数为 1 000 次。实验所使用的计算机硬件信息为:i7 - 11800H 型 CPU,主频为 2.3 GHz;显卡为 NVIDIA GeForce RTX 3060 Laptop GPU;内存容量为 16 GB;硬盘容量为 500 GB;操作系统为 Windows 11 专业版。实验设置 3 种不同的模拟环境:简单障碍物环境(图 5a)、复杂障碍物环境(图 5b)以及二维树形环境(图 5c)。在简单障碍环境与复杂障碍环境中,障碍物均匀分布,均设置起点(0,0)和目标点(200,200),以测试算法稳定性。二维树形环境则模拟树枝分布情况,其中起始点设为(0,0),目标点设为(132,150),目标点被树形障碍物包裹,以模拟果实生长于树枝干的采摘环境。在这 3 种不同环境下进行 40 次模拟实验,对比 RRT、RRTstar、RRT - connect、B - RRTstar、P - RRTstar 以及本文提出的 PIB - RRTstar 等 6 种算法路径规划效果。

图 6 为 3 种障碍物环境下 6 种算法的仿真结果,表 2 为 3 种障碍物环境下 6 种算法结果。由表 2



(a) 简单障碍物环境 (b) 复杂障碍物环境 (c) 二维树形环境

图 5 二维仿真地图

Fig. 5 Two-dimensional simulation maps

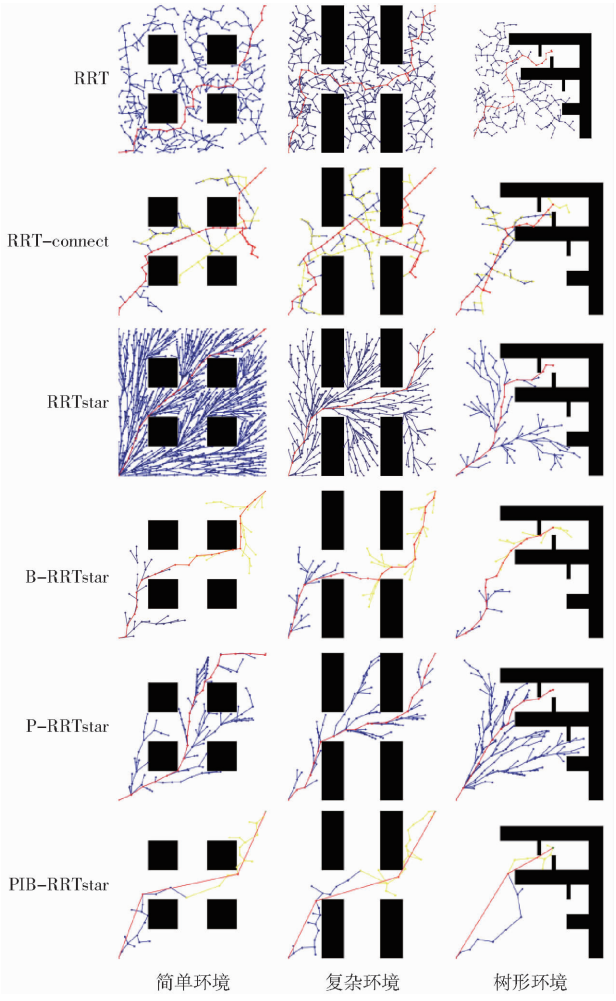


图 6　二维环境仿真对比

Fig. 6　Comparison of 2D environmental simulations

表 2　二维仿真数据统计

Tab. 2　Statistics of 2D simulation data

算法	迭代次数	规划时间/s	路径长度/mm
RRT	435.430 1	0.983 3	350.110 2
RRT－connect	108.201 8	0.271 2	429.189 3
RRTstar	501.015 4	3.590 7	278.933 1
B－RRTstar	55.151 8	0.126 1	306.462 9
P－RRTstar	191.136 4	1.924 2	285.458 4
PIB－RRTstar	37.642 1	0.042 4	273.323 1

可知，RRT－connect 在迭代次数上相较于 RRT 算法具有显著优势。这一优势源自 RRT－connect 的双向搜索策略以及贪婪扩展，通过同时从起点和目标点构建搜索树，并贪婪地向对立树进行扩展，提高路径探索效率。然而，由于采样次数相对较少以及对生成树结构并未优化，RRT－connect 生成路径质量极差。对比 RRT 与 RRTstar：RRTstar 通过引入 Chooseparent 和 Rewire 模块，能够在搜索过程中重构随机搜索树，从而生成质量更高的路径。然而，这两种机制的引入也带来额外的碰撞检测开销，对比 RRT 与 RRTstar 在的迭代次数和规划时间可以发现，当两者迭代次数接近时，RRTstar 将花费更多的时间开销。对同样使用双向搜索策略的 RRT－connect 与 B－RRTstar 进行对比，分析可知 B－RRTstar 通过交替探索两棵随机树来平衡叶子节点的数量，从而实现一种较为平衡的探索方式，而 RRT－connect 则表现出一种贪婪的探索特性，它不断朝着采样点扩展直到遇到障碍物，这种策略使得搜索树中呈现出大量的共线边和相邻节点的共线形态，在障碍物稀疏的情况下，RRT－connect 可以更快速地生成一条质量相对较高的路径，但在障碍物较密集时，RRT－connect 的路径质量将进一步降低。B－RRTstar 继承 RRTstar 的树结构优化模块，而 RRT－connect 与 RRT 相同，在随机搜索树扩展的过程中，并未对树结构进行调整。造成路径质量的进一步降低。. B－RRTstar 迭代过程中会尝试使用 x_{new} 与对立树中距离 x_{new} 最近的节点进行连接，从而生成路径，而 RRT－connect 仅尝试 x_{new} 与本轮对立树贪婪搜索扩展的节点进行连接尝试，无视距离 x_{new} 最近的对立树中的节点，因此造成路径迂回以及大量的无意义搜索。由于以上差异性，造成 RRT－connect 在路径长度、迭代次数、时间代价上都表现出极差的性能。P－RRTstar 算法通过引入人工势场法优化采样模块，采样节点质量有所提高，能够以更少的采样次数获取初始路径，实验数据表明相较于 RRTstar，P－RRTstar 在迭代次数和规划时间上有一定提升，但生成路径质量相对较差，这是由于 RRTstar 通过更多的随机采样优化路径，但这也造成更多的时间开销。

PIB－RRTstar 算法相较于 RRTstar 进行一系列改进。通过引入自适应步长，算法能够在更少的采样次数下探索更多的空间，提高搜索效率。人工势场的引入不仅保证搜索树向目标点的探索，还增强算法对环境的适应能力。四树搜索的策略减小空间探索范围，进一步加快探索效率。此外，通过向祖先节点回溯的方式，算法剔除生成路径上的冗余节点，从而提高路径质量。与 RRTstar 算法相比，本文提出的 PIB－RRTstar 算法在路径质量上提高 2.01%，迭代次数减少 92.49%，时间代价降低 98.81%，表明了优越性能。

3　机械臂仿真实验

3.1　机械臂采摘姿态约束

与二维仿真不同，机械臂路径规划在关节空间中进行，即采样点为机械臂各轴旋转角（θ_1，θ_2，θ_3，θ_4，θ_5，θ_6）。而终点姿态关节角度并不能直接获取，这是由于从深度相机获取的采摘点位置信息为笛卡

尔空间坐标,因此在规划前,须将坐标转换为机械臂关节角度,即进行逆运动学求解。逆运动学求解是机械臂运动规划中的关键步骤,它能够将目标位置与姿态信息映射为机械臂各关节的旋转角度。采用四元数来描述笛卡尔空间下的机械臂末端姿态,即

$$Pose = \{p_x, p_y, p_z, o_x, o_y, o_z, o_w\} \tag{4}$$

式中 $Pose$——机械臂末端姿态

p_x、p_y、p_z——笛卡尔空间坐标信息

o_x、o_y、o_z、o_w——四元数信息

四元数的使用有助于更精确地表达机械臂末端执行器空间位置和朝向。特别地,针对夹剪型末端执行机构,需要设定特定四元数,以确保末端以水平姿态进行夹剪,从而提高采摘成功率。然而,机械臂采摘作业还面临一些特殊的挑战。例如,荔枝果串作为采摘点,其柔性末枝和树叶不应被视为障碍物,此外,由于机械臂硬件的限制,如末端夹剪控制器和图像传感器安装位置对机械臂运动的限制,部分逆运动学求解的采摘姿态无法完成采摘任务。这种情况如图 7 所示。机械臂末端呈相同姿态,然而机械臂其他轴的转动角不同,这展现逆运动学求解的不确定性,当求解姿态呈图 7a 和图 7b 的姿态时,机械臂二轴和三轴将出现较大幅度的摆动,将这种情况定义为“大摆臂”,此时无法完成采摘工作。因此,基于实验观察,对关节姿态求解进行进一步限定。限定方法为

$$\theta_{goal} = \left\{ \theta_{cand}, \text{if}\left(\sum_{i=1}^{6} k_i(\theta_{cand_i} - \theta_{init_i}) < R \right) \right.$$
$$\left. 且(\forall i \in \{1,2,3,4,5,6\}, \theta_{cand_i} < r_i) \right\} \tag{5}$$

式中 θ_{cand}——采摘姿态候选集

θ_{init}——初始姿态关节角度

θ_{goal}——最佳采摘姿态

k_i——关节约束系数

R、r_i——固定常数

通过使用式(5)限定终点姿态相对起始姿态的每个关节最大转动值和起始姿态至终点姿态的总转动角度获取更适用于机械臂采摘的姿态,将这个限制方法与本文提出的 PIB - RRTstar 结合,获取结合姿态限制的新算法 PosePIB - RRTstar,并加入仿真对比实验。

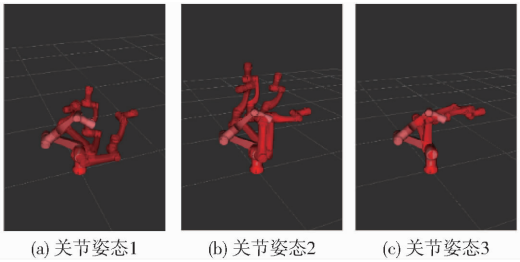


图 7 逆运动学求解姿态

Fig. 7 Inverse kinematics solved pose

3.2 机械臂关节空间仿真实验

机械臂关节空间仿真实验基于机器人操作系统(Robot operating system,ROS)进行。在 ROS 的众多核心插件中,Open Motion Planning Library(OMPL)作为一个开源的运动规划库,集成包括 RRT、RRT - connect、RRTstar 等在内的多种主流基于采样的运动规划算法。ROS 通过引入 OMPL 作为插件,为机器人提供规划策略。为验证本文所提出算法在机械臂运动约束下的性能,基于 ROS 和 OMPL 框架部署 PIB - RRTstar 算法与 PosePIB - RRTstar,并与 OMPL 中已有的 RRT、RRT - connect、RRTstar 等算法对比。仿真对比结果如图 8 所示,图中灰色为机械臂终点姿态,黄色为机械臂起始姿态,绿色为树型障碍物,蓝色曲线为机械臂运动轨迹。以弧度制表示的起始姿态关节角度为(1.08, - 0.96, - 2.30, 0.09, - 0.99, 0.54) rad。以三维空间坐标 + 四元数信息表示的终点位置为(0.15, 0.20, 0.70, 0.33, - 0.59, 0.63, 0.37) mm。

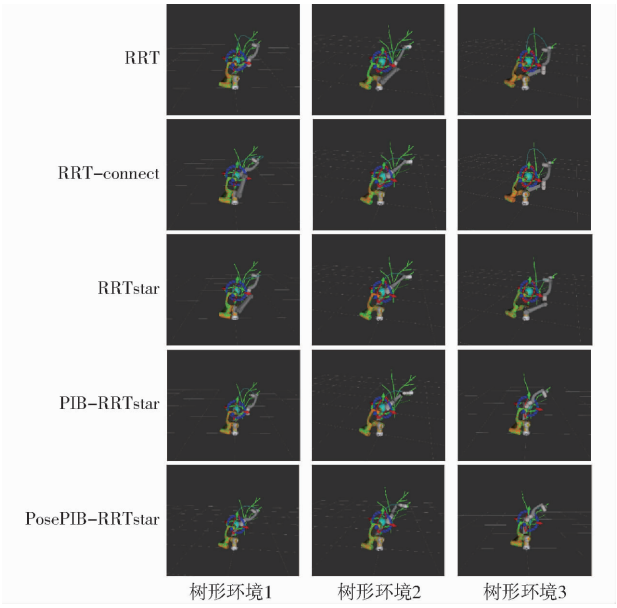


图 8 三维环境仿真对比

Fig. 8 Comparison of 3D environmental simulations

通过对 3 种不同树形场景下的机械臂运动学约束进行仿真,观察到 RRT - connect 算法在性能上优于 RRTstar 算法,这主要源于 RRT - connect 的贪婪搜索特性。在二维仿真环境中,当障碍物分布密集时,RRT - connect 因其贪婪的节点扩展策略而更容易遭遇碰撞,导致节点扩展失败。而在本文所建立的树形枝干环境中,障碍物分布相对稀疏,RRT - connect 的贪婪搜索策略所受阻碍较小,因此能够以较少的采样次数快速找到一条可行路径。与此同

时，本文提出的算法结合人工势场法和四向搜索策略，在障碍物稀疏的情况下同样能够以较少的采样次数达到目标区域。通过冗余节点删除，得到的初始路径长度更短，从而提高路径规划的效率。此外，在 PIB－RRTstar 算法基础上，针对机械臂运动特性，引入姿态限制函数，减小关节空间中起始姿态转变为采摘姿态所需转动的角度，从而进一步缩短路径长度。表 3 数据表明，与 RRTstar 相比，本文提出的 PIB－RRTstar 算法采样次数减少 96.93%，规划时间减少 92.01%，路径长度缩短 32.73%。经过姿态约束后的 PIB－RRTstar 性能进一步提升，相较于 RRTstar 算法，采样次数减少 97.00%，规划时间减少 93.17%，路径长度缩短 36.62%。这充分证明在机械臂运动学约束下，本文算法在树形环境中依然能够展现出优越的性能。

表 3　三维仿真数据统计

Tab. 3　Statistics of 3D simulation data

算法	迭代次数	规划时间/s	路径长度/rad
RRT	148.531 2	0.024 4	28.654 8
RRT－connect	4.920 1	0.009 5	9.606 8
RRTstar	144.548 3	0.060 1	13.634 1
PIB－RRTstar	4.350 0	0.004 8	9.171 2
PosePIB－RRTstar	4.331 8	0.004 1	8.641 3

4　机械臂采摘实验

4.1　实验材料

本实验基于真实荔枝树模型进行。完整的采摘装置由 Hans robot Elfin－5 型机械臂、夹剪型末端执行器、Realsense D435i 型深度相机、SDA25×20 薄型气缸等组成。如图 9 所示。机械臂共 6 轴，具有 6 自由度。其中用于荔枝识别定位的 Realsense D435i 型深度相机采用“Eye-in-hand”的方式固定于机械臂 6 轴末端；夹剪型末端执行器与机械臂第 6 轴连接，作为机械臂第 6 轴的延伸，末端执行器在采摘过程中的运动由机械臂第 6 轴控制运动；SDA25×20 薄型气缸为末端执行器剪切荔枝果梗提供动力。



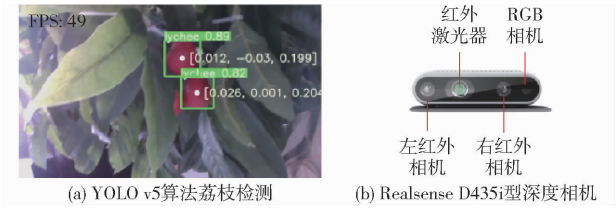
图 9　机械臂采摘示意图

Fig. 9　Schematic of picking manipulator

4.2　采摘系统构建

采摘系统分为荔枝识别定位模块、机械臂控制规划模块以及末端夹剪模块。

视觉识别定位模块：使用 YOLO v5 算法对荔枝果实进行识别，并采用 Realsense D435i 型深度相机实现荔枝果实定位，获取果实三维坐标。荔枝果实识别定位效果如图 10 所示，图中 FPS 表示模型对果实的识别速度。使用绿色方框锚定荔枝果实，图中三维数组为目标的的三维坐标信息。



(a) YOLO v5算法荔枝检测　　(b) Realsense D435i型深度相机

图 10　视觉识别定位模块

Fig. 10　Visual recognition positioning system

机械臂控制规划模块：机械臂控制规划系统基于 ROS 实现，ROS 控制机械臂流程如图 11 所示。图中各模块作用如下：move_group 订阅获取采摘点信息。geometry_msgs 使用采摘点信息获取机械臂末端执行器空间姿态，并传递给运动学求解模块 kinematic_plugins。kinematic_plugins 基于机械臂 URDF 模型，求解末端执行器对应机械臂关节角度，并传递给运动规划组件 OMPL。OMPL 使用内嵌的路径规划算法，在关节空间中规划出一系列关节点，即关节转动路径。并传递至 robot_trajectory。robot_trajectory 通过线性插值等方法平滑关节转动路径，并生成关节点对应速度和加速度信息，即生成运动轨迹。ros_control 利用运动轨迹信息生成控制信号，控制真实机械臂进行运动。

夹剪模块：在对机械臂规划控制，使之精确到达荔枝采摘点后，针对荔枝果串的切割需求，设计一款末端执行器，其结构如图 12 所示。图 12a 为末端执行器梳理姿态。在果串切割之前，执行器将保持此姿态，以便于在接近果串时，通过其独特的结构设计对果串进行有序的梳理和引导，确保果串能够顺利进入预设的 3 个夹持区域中。在果串进入夹持区域后，夹持机构将紧密夹持果串，确保其在受到剪切机构施加的剪切力时不会发生位移，保障果串收割准确和效率。

4.3　机械臂采摘实验

在室内环境中搭建仿真荔枝树，挂上荔枝果串模型，用于模仿真实荔枝采摘环境。为评价本文算法的采摘性能，在荔枝果树上挂 5 串荔枝，如图 13 所示。每个果实进行 20 次重复采摘实验。图 14 为不同算法在采摘同一果串时，机械臂姿态变化情况。算法采摘效率如表 4 所示。

由表 4 可知，实际采摘中机械臂关节转动角远

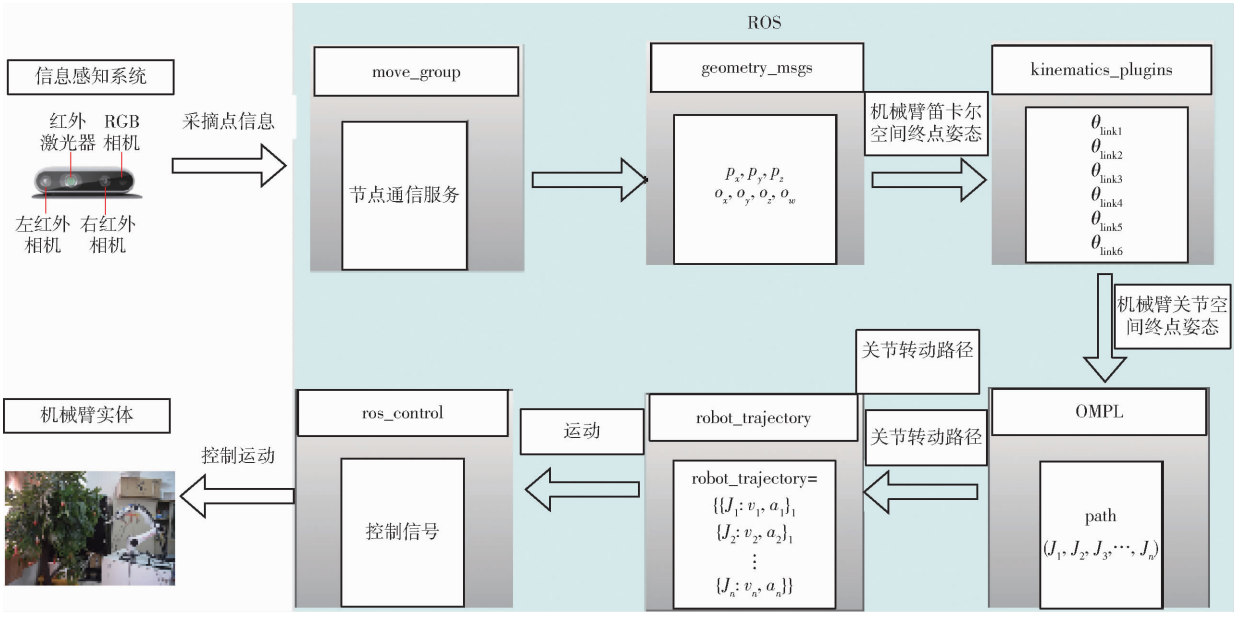


图 11 机械臂控制规划模块

Fig. 11 Robotic arm control planning system

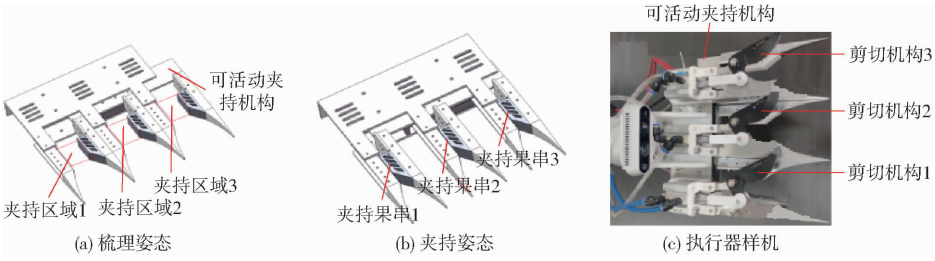


图 12 末端执行器结构

Fig. 12 Structure of end effector



图 13 荔枝果树模型

Fig. 13 Lychee fruit tree model

小于仿真环境中规划路径转动角。原因为在控制规划系统中,OMPL 调用路径规划算法生成关节空间下的路径后,robot_trajectory 会对这些路径进行线性插值,使得轨迹更加平滑,从而减小轨迹代价。而这种轨迹优化基于原始路径进行,因此,拥有较佳质量的初始路径能够生成更高质量轨迹。在这一方面,RRTstar 系列算法展现出其优越性,进一步证明路径规划算法重要性。本文提出的 PIB - RRTstar 和 PosePIB - RRTstar 算法在采摘成功率方面取得大幅度提升。主要得益于 PIB - RRTstar 算法中使用的空间分割点对路径生成的引导作用。机械臂运动轨迹将遵循起点区域至分割点区域,再至终点区域的规律,从而有效提高采摘成功率。此外,在姿态约束后,机械臂摆臂现象得到有效控制,进一步提升采摘成功率。尽管本文算法在采摘成功率方面表现出色,但在采摘时间代价方面却有所增加。这是由于 robot_trajectory 在对生成路径进行插值时,分别对起点区域至分割点区域、分割点区域至终点区域两条路径进行插值处理。这一过程中,当机械臂运动至分割点区域时,其角速度会缓慢下降,从而导致整个采摘过程时间成本上升。实验数据表明,本文提出的算法在采摘成功率方面具有显著优势。采摘实验结果表明,相较于 RRTstar 算法,本文提出的 PIB - RRTstar 算法采样次数减少 93.85%,路径代价减少 2.53%,采摘执行时间增加 3.25%,PIB - RRTstar 算法的采摘成功率提升 30 个百分点。并且在使用姿态选取策略后,采摘成功率进一步提升至 95%。



图 14　采摘实验对比

Fig. 14　Comparison chart of harvesting trials

表 4　采摘实验结果

Tab. 4　Results of harvesting experiments

算法	迭代次数	执行时间/s	路径长度/rad	采摘成功率/%
RRT	68.32	8.067 7	2.512 4	51
RRT－connect	4.94	7.685 8	2.745 7	55
RRTstar	70.33	7.597 1	2.366 2	55
PIB－RRTstar	4.32	7.844 1	2.306 3	85
PosePIB－RRTstar	4.33	7.764 4	2.306 1	95

5　结论

（1）提出 PIB－RRTstar 算法。二维仿真实验结果表明，与 RRTstar 算法相比，本文提出的 PIB－RRTstar 算法在路径质量上提高 2.01%，迭代次数减少 92.49%，时间代价降低 98.81%。

（2）机械臂采摘荔枝实验证明，PIB－RRTstar 采摘成功率达 85%，运行时间为 7.844 1 s。相比于 RRTstar 算法，采摘成功率提升 30 个百分点，结合本文提出的姿态约束方法，荔枝采摘成功率达到 95%。

参　考　文　献

[1]　常有宏，吕晓兰，蔺经，等. 我国果园机械化现状与发展思路[J]. 中国农机化学报，2013，34(6)：21－26.
CHANG Youhong, LÜ Xiaolan, LIN Jing, et al. Present state and thinking about development of orchard mechanization in China[J]. Journal of Chinese Agricultural Mechanization, 2013, 34(6): 21－26. (in Chinese)

[2]　徐丹琦. 基于 kinect 相机的自然生长状态下果树枝干的三维构建[D]. 桂林：广西师范大学，2021.
XU Danqi. 3D construction of fruit tree branches in natural growth based on kinect camera[D]. Guilin: Guangxi Normal University, 2021. (in Chinese)

[3]　STENTZ A. Optimal and efficient path planning for partially-known environments[C] // Proceedings of the 1994 IEEE International Conference on Robotics and Automation. IEEE, 1994: 3310－3317.

[4]　LAVALLE S. Rapidly-exploring random trees: a new tool for path planning[R]. Research Report 9811, 1998.

[5]　KAVRAKI L E, SVESTKA P, LATOMBE J C, et al. Probabilistic roadmaps for path planning in high-dimensional configuration spaces[J]. IEEE Transactions on Robotics and Automation, 1996, 12(4): 566－580.

[6]　阳涵疆，李立君，高自成. 基于关节构形空间的混联采摘机械臂避障路径规划[J]. 农业工程学报，2017，33(4)：55－62.
YANG Hanjiang, LI Lijun, GAO Zicheng. Obstacle avoidance path planning for hybrid picking robotic arm based on joint conformation space[J]. Transactions of the CSAE, 2017, 33(4): 55－62. (in Chinese)

[7] 马冀桐,王毅,何宇,等. 基于构型空间先验知识引导点的柑橘采摘机械臂运动规划[J]. 农业工程学报,2019,35(8):100 - 108.
MA Jitong, WANG Yi, HE Yu, et al. Motion planning of citrus harvesting manipulator based on informed guidance point of configuration space[J]. Transactions of the CSAE, 2019, 35(8): 100 - 108. (in Chinese)

[8] 崔永杰,王寅初,何智,等. 基于改进 RRT 算法的猕猴桃采摘机器人全局路径规划[J]. 农业机械学报,2022,53(6):151 - 158.
CUI Yongjie, WANG Yinchu, HE Zhi, et al. Global path planning of kiwifruit harvesting robot based on improved RRT algorithm [J]. Transactions of the Chinese Society for Agricultural Machinery, 2022, 53(6): 151 - 158. (in Chinese)

[9] 荀一,李道政,王勇,等. 基于 VS - IRRT 算法的采摘机械臂路径规划[J]. 农业机械学报,2023,54(2):129 - 138.
XUN Yi, LI Daozheng, WANG Yong, et al. Motion planning of harvesting manipulator based on VS - IRRT algorithm[J]. Transactions of the Chinese Society for Agricultural Machinery, 2023, 54(2): 129 - 138. (in Chinese)

[10] 邹宇星,李立君,高自成. 基于改进 PRM 的采摘机器人机械臂避障路径规划[J]. 传感器与微系统,2019,38(1):52 - 56.
ZOU Yuxing, LI Lijun, GAO Zicheng. Obstacle avoidance path planning for harvesting robot arm based on improved PRM[J]. Transducer and Microsystem Technologies, 2019, 38(1): 52 - 56. (in Chinese)

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

[12] NASIR J, ISLAM F, MALIK U, et al. RRT* - SMART: a rapid convergence implementation of RRT[J]. International Journal of Advanced Robotic Systems, 2013, 10(7): 299.

[13] GAMMELL J D, SRINIVASA S S, BARDOOT T D. Informed RRT*: optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic[C] // 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2014: 2997 - 3004.

[14] AKGUN B, STILMAN M. Sampling heuristics for optimal motion planning in high dimensions[C] // 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2011: 2640 - 2645.

[15] GUO J, XIA W, HU X, et al. Feedback RRT* algorithm for UAV path planning in a hostile environment[J]. Computers & Industrial Engineering, 2022, 174: 108771.

[16] ZHANG H, WANG Y, ZHENG J, et al. Path planning of industrial robot based on improved RRT algorithm in complex environments[J]. IEEE Access, 2018, 6: 53296 - 53306.

[17] QI J, YANG H, SUN H. MOD - RRT*: a sampling-based algorithm for robot path planning in dynamic environment[J]. IEEE Transactions on Industrial Electronics, 2020, 68(8): 7244 - 7251.

[18] QURESHI A H, AYAZ Y. Intelligent bidirectional rapidly-exploring random trees for optimal motion planning in complex cluttered environments[J]. Robotics and Autonomous Systems, 2015, 68: 1 - 11.

[19] WANG X, WEI J, ZHOU X, et al. AEB-RRT*: an adaptive extension bidirectional RRT* algorithm[J]. Autonomous Robots, 2022, 46(6): 685 - 704.

[20] GHOSH D, NANDAKUMAR G, NARAYANAN K, et al. Kinematic constraints based bi-directional RRT (KB - RRT) with parameterized trajectories for robot path planning in cluttered environment[C] // International Conference on Robotics and Automation (ICRA). IEEE, 2019: 8627 - 8633.

[21] QURESHI A H, AYAZ Y. Potential functions based sampling heuristic for optimal path planning[J]. Autonomous Robots, 2016, 40: 1079 - 1093.

[22] KHATIB O. Real-time obstacle avoidance for manipulators and mobile robots[J]. The International Journal of Robotics Rresearch, 1986, 5(1): 90 - 98.

[23] WANG J, CHI W, LI C, et al. Neural RRT*: learning-based optimal path planning[J]. IEEE Transactions on Automation Science and Engineering, 2020, 17(4): 1748 - 1758.

[24] LIN G, ZHU L, LI J, et al. Collision-free path planning for a guava-harvesting robot based on recurrent deep reinforcement learning[J]. Computers and Electronics in Agriculture, 2021, 188: 106350.

[25] WANG Y, LIU D, ZHAO H, et al. Rapid citrus harvesting motion planning with pre-harvesting point and quad-tree[J]. Computers and Electronics in Agriculture, 2022, 202: 107348.

[26] CHEN B, GONG L, YU C, et al. Workspace decomposition based path planning for fruit-picking robot in complex greenhouse environment[J]. Computers and Electronics in Agriculture, 2023, 215: 108353.

[27] 张勤,刘丰溥,蒋先平,等. 番茄串收机械臂运动规划方法与试验[J]. 农业工程学报,2021,37(9):149 - 156.
ZHANG Qin, LIU Fengpu, JIANG Xianping, et al. Motion planning method and experiments of tomato bunch harvesting manipulator[J]. Transactions of the CSAE, 2021, 37(9): 149 - 156. (in Chinese)

[28] 王坤,黄勃,曾国辉,等. 基于改进 RRT - Connect 的快速路径规划算法[J]. 武汉大学学报(理学版),2019,65(3):283 - 289.
WANG Kun, HUANG Bo, ZENG Guohui, et al. Faster path planning based on improved RRT - Connect algorithm[J]. Journal of Wuhan University (Natural Science Edition), 2019, 65(3): 283 - 289. (in Chinese)