

DOI:10.19651/j.cnki.emt.2518546

基于 KDB-RRT* 算法的智能体路径规划*

魏文卿^{1,2} 魏坤^{1,2} 张建辉^{1,2}

(1. 天津职业技术师范大学汽车与交通学院 天津 300222; 2. 智能车路协同与安全技术国家地方联合工程研究中心 天津 300222)

摘要: 针对复杂环境中智能体路径规划的挑战,尤其是算法收敛速度慢、路径冗余度高及平滑性不足等问题,提出一种基于 KD 树的目标偏置双 RRT* 算法(KDB-RRT*)。该算法基于 RRT* 算法引入双向搜索策略,加入 KD 树结构提高节点查找速度,构建目标偏置的动态圆采样策略平衡搜索效率,设计基于引力场的双向生长引导模型,引入 Sigmoid 函数并结合障碍物密度实现步长自适应调整,并采用 DP 算法对原始路径进行剪枝处理,运用三次 B 样条曲线平滑路径。在“Z 型”和“回型”仿真环境中验证了 KDB-RRT* 的可行性,在多种复杂地图环境中与 RRT* 算法、Bi-RRT 算法、Improved RRT* 算法进行对比实验;最后,在 ROS 机器人上进行路径规划实验。在“Z 型”和“回型”仿真环境中,KDB-RRT* 算法与 RRT* 算法相比,平均规划时间分别缩短 70.2% 和 28.0%,平均路径长度分别减少 4.8% 和 10.4%,节点利用率分别增加 16.27% 和 13.58%。结果表明 KDB-RRT* 算法为解决非结构化环境下的高效路径规划提供了新方法,其动态采样模型与路径优化框架对移动机器人导航系统具有重要参考价值。

关键词: 路径规划;RRT* 算法;KD 树;Sigmoid 函数;DP 算法

中图分类号: TP242;TN05 **文献标识码:** A **国家标准学科分类代码:** 510.2040

Path planning for intelligent agent based on the KDB-RRT* algorithm

Wei Wenqing^{1,2} Wei Kun^{1,2} Zhang Jianhui^{1,2}

(1. School of Automobile and Transportation, Tianjin University of Technology and Education, Tianjin 300222, China; 2. National and Local Joint Engineering Research Center for Intelligent Vehicle Road Collaboration and Safety Technology, Tianjin 300222, China)

Abstract: To address the challenges of path planning for intelligent agents in complex environments, particularly issues such as slow algorithm convergence, high path redundancy, and insufficient smoothness, this paper proposes a target-biased bidirectional RRT* algorithm based on KD-tree (KDB-RRT*). The algorithm introduces a bidirectional search strategy based on RRT*, incorporates a KD-tree structure to accelerate node lookup, constructs a target-biased dynamic circular sampling strategy to balance search efficiency, designs a bidirectional growth guidance model based on gravitational fields, implements adaptive step-size adjustment using the Sigmoid function combined with obstacle density, and employs the DP algorithm for original path pruning and cubic B-spline curves for path smoothing. The feasibility of KDB-RRT* is verified in “Z-shaped” and “loop-shaped” simulation environments, and comparative experiments are conducted with RRT*, Bi-RRT, and Improved RRT* algorithms in various complex map environments. Finally, path planning experiments are performed on a ROS robot. In the “Z-shaped” and “loop-shaped” simulation environments, compared with the RRT* algorithm, KDB-RRT* reduces the average planning time by 70.2% and 28.0%, decreases the average path length by 4.8% and 10.4%, and increases the node utilization rate by 16.27% and 13.58%, respectively. The results show that the KDB-RRT* algorithm provides a new method for efficient path planning in unstructured environments, and its dynamic sampling model and path optimization framework have important reference value for mobile robot navigation systems.

Keywords: path planning; RRT* algorithm; KD-tree; Sigmoid function; DP algorithm

0 引言

随着人工智能发展迅速,智能体作为标志性产品被广

泛应用于多个领域,路径规划技术^[1]是智能体研究的核心技术之一,它直接决定了智能体移动路径的质量,进而影响其执行任务的效率与成功率。因此对于智能体自主路径规

收稿日期:2025-04-08

* 基金项目:天津高等教育科技发展基金(2021KJ021)、天津市揭榜挂帅科研项目(KRKC012216)资助

划的研究对于人工智能行业的发展有着重要意义。

目前路径规划的方法主要包括 A* 算法^[2]、人工势场法^[3]、随机采样算法^[4]、蚁群算法^[5]、遗传算法^[6]等,近几十年来,基于随机搜索策略的快速扩展随机树算法^[7](rapidly-exploring random tree, RRT)因其概率完备性、可扩展性等优点,在智能体的路径规划中得到了广泛的应用,但也存在规划时间成本过大、规划路径成本过大、规划路径质量过低等缺点,不能保证规划速度和规划效率。为了改善这些问题,国内外各类学者在 RRT 算法的基础上提出各种变体来解决。Meng 等^[8]提出渐进最优的快速探索随机树星算法(rapidly-exploring random tree star, RRT*),该算法通过重新选择父节点和重新布线来提高路径的质量,逐步收敛到渐进最优解,但计算成本较高,算法效率相对较低,搜索时间长。Wang 等^[9]提出双向快速探索随机树算法(bidirectional rapidly-exploring random tree, Bi-RRT),双向扩展 RRT,从起始点和目标点同时生长两棵树,直至相遇,通过平衡两棵树的扩展来提高搜索效率和稳定性,但计算成本较高,且在某些复杂环境下可能受到环境约束的限制。Chen 等^[10]提出 RRT-Connect 算法,结合 RRT 和贪心算法,从起点和终点构建两棵树交替生长,路径搜索效率高,但路径质量可能不如 RRT*,且在某些情况下可能陷入局部最优。谢高杨等^[11]提出一种改进 RRT 算法,将车速于扩展步长匹配,为不同车速下无人靶车的路径规划问题提供了一种新的解决方案,但复杂情况下路径的质量可能下降。Lim 等^[12]提出 Ex-RRT* 算法,该算法通过为每个节点到最近障碍物的距离增加代价函数,以保证路径规划过程中生成的节点对障碍物有适当的裕度,这虽然提高了生成路径的安全性。顾子侣等^[13]提出基于目标启发的待扩展节点选择策略,虽然降低了算法的随机性,但面对复杂环境收敛效率仍缓慢。马晓群等^[14]提出一种依赖环境复杂度的 IRRT*-Connect 算法,增强了算法适应复杂环境的能力,但仍需解决计算效率低的问题。李文君等^[15]针对 RRT* 算法在多障碍物环境中生成的路径存在冗余点过多、呈锯齿状、贴近障碍物等问题,提出一种 Safe-Smooth RRT* 算法,提升了路径的平滑度,但随着障碍物的增加,算法性能可能下降。刘文光等^[16]提出约束范围内重选父节点的 RRT 算法,但多障碍物环境下可能导致搜索过程时间较长。

为了解决 RRT* 算法计算成本过高、搜索效率较低、路径质量不高的问题,本文提出基于 KD 树(K-Dimensional Tree)的目标偏置双 RRT* 算法(KDB-RRT*)。该算法在 RRT* 的渐进最优特性的基础上引入双向搜索策略,通过三阶优化框架实现性能突破:结合 KD 树空间索引技术加速近邻节点检索,降低计算复杂度;构建动态半径采样策略,设计双向引力场引导模型,通过目标偏置机制增强路径扩展的方向性,并基于 Sigmoid 函数实现自适应调节步长,提升复杂区域的避障能力;采用 DP 算法(Douglas-

Peucker)进行路径剪枝,运用三次 B 样条曲线对初始路径进行曲率连续化处理,实现路径长度与平滑度的多目标优化。最后,本文对 KDB-RRT* 算法与 RRT* 算法、Bi-RRT 算法、Improved RRT* 算法进行对比实验,结果表明,该算法在保持概率完备性的同时,显著提升了路径规划效率和运动轨迹质量。

1 RRT* 算法基本原理

传统 RRT 算法的路径增长方向由随机引导点的生成决定,其核心在于路径树在空间的随机扩展,以探索潜在路径。如图 1 所示,首先,在自由空间内确定起始节点 P_{start} ,并随机采样得到 P_{rand} 。接着,在已有树上找到距 P_{rand} 最近的节点 P_{near} ,尝试连接 P_{rand} 与 P_{near} 。若连接线段未与障碍物相交,则沿此方向扩展距离 ϵ 得到新节点 P_{new} ,并将其添加到树上。此过程循环进行,直至目标节点(或接近目标节点的节点)被添加。最后,从目标节点回溯至起始节点,即可得到路径。

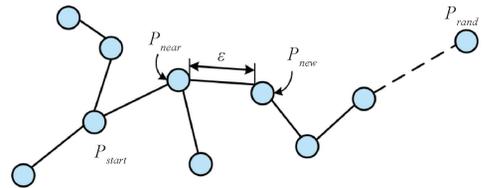


图 1 RRT 算法基本原理图

Fig. 1 Basic principle diagram of RRT

RRT* 算法是在 RRT 算法的基础上,借助了计算路径代价的大小来选择拥有最小路径代价的节点作为父节点,从而改变了原有随机选择父节点的方式,在这一过程中,大致分为如图 2 所示的两个阶段: P_{new} 重新选择父节点、重新布线随机树。第一个阶段 P_{new} 重新选择父节点的过程如图 2(a)所示,生成扩展节点 P_{new} 后,以 P_{new} 为中心画出绿圆,圆内节点构成邻近节点集合。通过比较各邻近节点作为父节点时的路径代价,选择代价最小的节点作为 P_{new} 的新父节点,以优化路径。第二阶段旨在进一步减小路径代价,检查 P_{new} 的邻近父节点,若更改父节点能降低路径代价,则进行替换。如图 2(b)所示,对于 P_{new} 的邻近节点,算法计算其分别通过原父节点和 P_{new} 作为父节点时的路径代价,并选择代价较小的路径。此过程迭代进行,直至无法进一步减小路径代价,从而得到最优路径。

2 B-RRT* 算法基本原理

B-RRT* 相较于 RRT* 算法,在节点扩展策略上进行了创新,引入了双向搜索策略的概念,如图 3 所示,在起点和终点分布构建起点树和终点树,其中起点树以目标点 P_{goal} 为终点,终点树以起始点 P_{start} 为终点,两棵随机树交替扩展,具体生长过程如 1.1 节中 RRT 算法一致,直到检测到两棵树的新节点欧氏距离小于设定的阈值或相遇。此

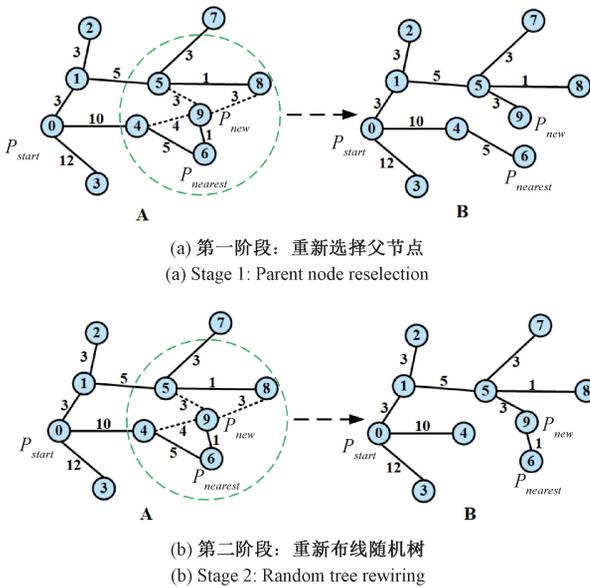


图 2 RRT* 算法基本原理图

Fig. 2 Basic principle diagram of RRT*

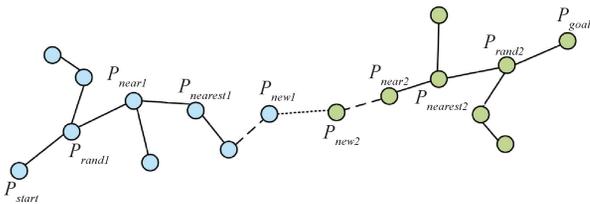


图 3 B-RRT* 算法基本原理图

Fig. 3 Basic principle diagram of B-RRT*

时,连接这两个新节点,即可生成一条从起点到终点的完整路径。

3 KDB-RRT* 算法设计

3.1 基于 KD 树的快速邻域查询

尽管 B-RRT* 算法提高了搜索效率,但在每次迭代过程中,都需要遍历搜索节点,再利用两点间的欧式距离选择合适的节点,会产生大量的计算工作,严重影响算法在迭代过程中的规划效率。因此,本文提出 KDB-RRT* 算法,采用 KD 树对最近节点搜索以及重新选择父节点的部分进行改进,将树节点的位置信息存储在 KD 树中,通过比较的方式替代复杂的计算,提高搜索效率。

KD 树^[17]在 1975 年提出,是一种用于多维空间数据组织和存储的树形数据结构,它将二叉搜索树的概念推广到多维数据中,通过超平面将多维空间划分为多个不相交的子空间,从而实现数据的快速索引和搜索^[18-20]。

假设现存在随机树 T 上节点集合 $P_{new} \{A(1,8), B(8,7), C(6,3), D(3,4), E(4,6), F(9,1)\}$,对于上述集合的构造 KD 树流程为(如图 4 所示):

1)选择分割轴。分别计算 x 维度、 y 维度的方差,比较

两者的大小,选择方差较大的维度作为分割轴。通过式(1)计算可得出本例中 x 维度、 y 维度的方差分别为 $\sigma_x^2 = 7.81, \sigma_y^2 = 5.56$,由于 $\sigma_x^2 > \sigma_y^2$,因此选择 x 维度作为第一个分割轴。

$$\sigma^2 = \frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + \dots + (x_n - \bar{x})^2}{n} \quad (1)$$

2)确定分割轴上中值点。按照分割轴所在的维度将数据点进行排序,选择中位数作为根节点。本例中将所有数据点按 x 维度排序, $A(1,8), D(3,4), E(4,6), C(6,3), B(8,7), F(9,1)$,可得中值点为 $C(6,3)$,即 $C(6,3)$ 为根节点。

3)左右子树的确定。以选定的分割轴当成划分维度、中值点为分割平面,分别构建左子树、右子树。本例中左子树集合为 $\{A(1,8), D(3,4), E(4,6)\}$,右子树集合为 $\{B(8,7), F(9,1)\}$ 。

4)左右子树集合,分别继续进行步骤 1)~3),不断进行分割,直至每个子树集合中只存在一个数据点。但需注意,选择分割轴时,通常按照维度顺序轮询。本例中由于第 1 层分割时选择 x 维度作为分割轴,因此,第 2 层分割时选择 y 维度作为分割轴,第 3 层分割时选择 x 维度作为分割轴,以此类推。

在本例中步骤 4)如下:

(1)对于左子树,选择 y 维度作为分割轴,分割轴上的中值点为 $E(4,6)$,进一步分割为左子节点 $A(1,8)$,右子节点 $D(3,4)$ 。

(2)对于右子树,选择 y 维度作为分割轴,选择 $B(8,7)$ 作为右子树的根节点,因为它是两个点中 y 坐标较大的一个,进一步分割右子节点 $F(9,1)$ 。

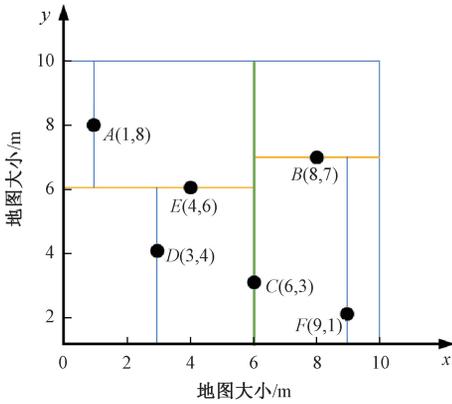
通过上述流程构建的 KD 树二维空间结构如图 4(a)所示,组织构建如图 4(b)所示。利用该 KD 树,以寻找最近邻节点 P_{rand} 为例,假设此时的 P_{rand} 为 $O(2,6.5)$,具体搜索过程如下(如图 5 所示):

1)从根节点 $C(6,3)$ 开始,假设 C 是最近邻点,以点 O 为圆心, CO 为半径确定一个红圆,如图 5(a)所示,此时红圆与 $y=6$ 和 $y=7$ 都交割,由于 $6 < 7$,因此先进入 C 的右子空间进行搜索。

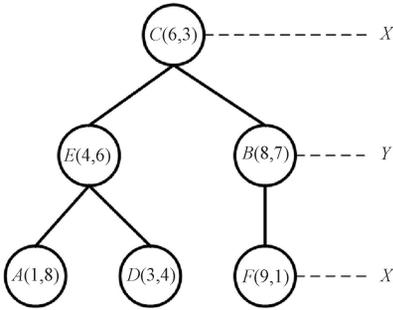
2)查找到 $E(4,6)$,在进行查找时是由 $y=6$ 为分割超平面的,由于查找点 y 值为 6.5 ,因此进入右子空间查找到 $D(3,4)$ 形成搜索路径: $C(6,3) \rightarrow E(4,6) \rightarrow D(3,4)$ 。取 $D(3,4)$ 为当前最近邻点。以 O 为圆心, OD 为半径确定一个蓝圆,如图 5(b)所示。

3)回溯到 $E(4,6)$,计算其与目标点 O 之间的距离, $OE < OD$,更换 $E(4,6)$ 为当前最近邻点。用同样的方法再次确定一个黄圆,如图 5(c)所示。

4)由于黄圆和 $y=4$ 超平面相交,所以需要进入 $E(4,6)$ 结点的另一个子空间进行查找。 $A(1,8)$ 与目标点 O 距离为 $OA, OA < OE$,所以最近邻点更新为 $A(1,8)$,最近距离更新为 OA 。同样可以确定一个绿圆,如图 5(d)所示。



(a) 二维空间构建KD树
(a) KD-tree construction in 2D space



(b) KD树的组织构建
(b) Structural organization of KD-tree

图 4 KD 树组织与二维空间划分

Fig. 4 KD-tree organization and 2D space partitioning

5)接着根据规则回退到根结点 $C(6,3)$, 绿圆与 $x=6$ 的超平面不相交, 因此不用进入 $C(6,3)$ 的右子空间进行查找。至此, 搜索路径回溯完, 返回最近邻点 $A(1,8)$, 最近距离 OA , 即 A 为所要寻找的最近邻节点 P_{near} 。

3.2 基于目标偏置的动态圆采样策略

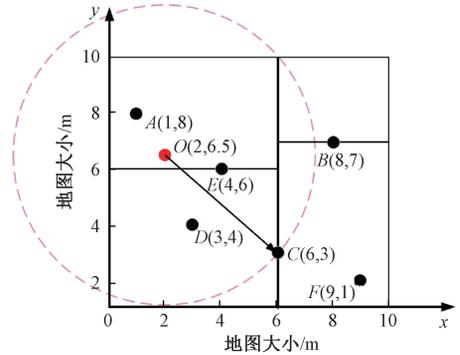
尽管引入 KD 树的 KDB-RRT* 算法大大增加了路径规划的效率, 但仍无法避免在复杂环境下采样随机性强而导致的收敛速度低的问题, 因此, 本文提出基于目标偏置的动态圆采样策略。定义动态采样半径如式(2)所示。

$$R(d) = R_0 \times [1 - (\frac{d}{D})^k] \quad (k \geq 1) \quad (2)$$

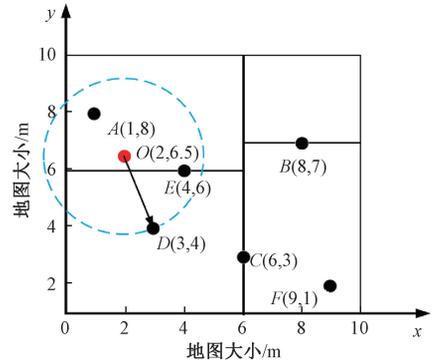
其中, R_0 为初始采样半径; $d = \|P_{rand} - P_{goal}\|$ 为当前节点到目标点的欧式距离; $D = \|P_{start} - P_{goal}\|$ 为起始点到目标点的欧式距离, 用于归一化 d ; k 为调整因子, 用于控制半径随距离变化的速率。当 $d \rightarrow D$ (即当前节点距离目标点较远) 时, 扩大搜索范围加速趋近目标区域; 当 $d \rightarrow 0$ (即当前节点距离目标点较近) 时, 缩小范围实现局部精细化搜索。

在迭代过程中, 以当前节点 P_{near} 为中心建立动态采样区域 Ω , 如式(3)所示, 将随机点 P_{rand} 的采样空间约束在 Ω 内。整个路径规划过程中动态采样过程如图 6 所示。

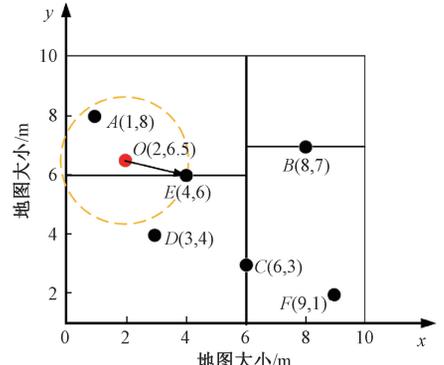
$$\Omega = \{p \in R^2 \mid \|p - P_{near}\| \leq R(d)\} \quad (3)$$



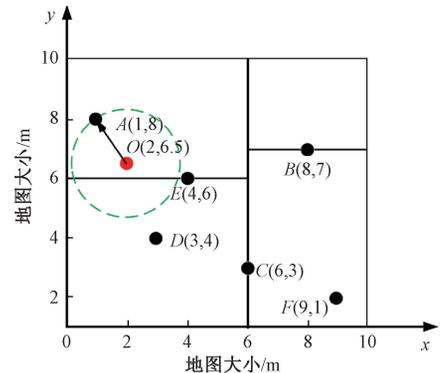
(a) 以C为最近邻点
(a) Designate C as the nearest neighbor



(b) 更新D为最近邻点
(b) Update D as the nearest neighbor



(c) 更新E为最近邻点
(c) Update E as the nearest neighbor



(d) 更新A为最近邻点
(d) Update A as the nearest neighbor

图 5 KD 树最近邻搜索

Fig. 5 KD-tree nearest neighbor search

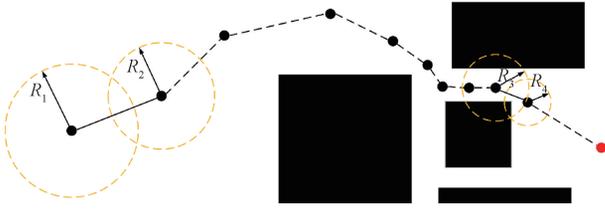


图6 动态圆采样区域的生成

Fig. 6 Generation of dynamic circular sampling regions

3.3 基于引力场的生长引导模型

在B-RRT*算法的两棵随机树生长的过程中,两棵树分别向着彼此的起点作为目标点进行生长,新节点的扩展方式如式(4)和图7所示。

$$P_{new} = P_{near} + s \cdot \vec{F} \quad (4)$$

图7 B-RRT*算法新节点扩展示意图

Fig. 7 Principle diagram of new node expansion in B-RRT*

其中, $\vec{F} = \frac{P_{rand} - P_{near}}{\|P_{rand} - P_{near}\|}$ 为随机扩展方向向量, s 为扩展固定步长。

但这样扩展过于随机且趋向目标性不强,为进一步强化算法在双向搜索中的方向性,本文建立基于引力场的生长引导模型。对于起点树 T_1 和终点树 T_2 , 分别设置目标引力向量:

$$\vec{G}_1 = \lambda \times \frac{P_{goal} - P_{near}}{\|P_{goal} - P_{near}\|}, \vec{G}_2 = \lambda \times \frac{P_{start} - P_{near}}{\|P_{start} - P_{near}\|} \quad (5)$$

其中, λ 为引力增益次数。则新节点生成策略改进为如式(6)和图8所示。

$$P_{new} = P_{near} + s' \cdot \vec{F} + \vec{G}_i \quad (6)$$

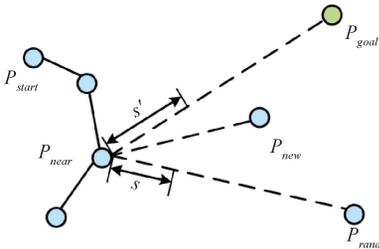


图8 基于引力场引导的新节点扩展示意图

Fig. 8 Schematic diagram of new node expansion guided by gravitational field

其中, s' 为动态调整的扩展步长。该机制使双树在保持随机探索能力的同时获得目标导向性,显著提高相遇

概率。

3.4 基于障碍物密度的自适应步长

针对障碍物分布不均匀特性,提出基于Sigmoid函数的步长动态调节方法:

$$s' = \begin{cases} s, d_{obs} \geq h \\ \frac{s}{1 + e^{-\alpha \cdot (d_{obs} - h)}}, d_{obs} < h \end{cases} \quad (7)$$

其中, d_{obs} 为 P_{near} 到最近障碍物的距离; h 为安全阈值, α 为调节因子。当所处区域障碍物稀疏,即 $d_{obs} \geq h$ 时,采用固定步长 s 保持扩展效率;当进入障碍物密集区域,即 $d_{obs} < h$ 时,步长随障碍物接近度呈非线性衰减,在提升安全性的同时避免过度保守的路径迂回。

3.5 路径优化

经过上述一系列的优化策略后,所规划的路径冗余点虽大幅度减少但仍存在较多,因此本文采用道格拉斯-普克算法^[22](Douglas-Peucker algorithm, DP)来进行剪枝处理。DP算法的核心思想是通过递归的方式,将一条折线简化为一条更少点的折线,同时尽可能保留原始折线的主要特征。给定路径点列 $\{Q_i\}_{i=0}^n$, 给定阈值 D , 起点 Q_0 , 终点 Q_{n-1} , 计算所有中间点 Q_i ($1 \leq i \leq n-2$) 到线段 Q_0Q_{n-1} 的垂直距离 d_i , 找到最大距离 d_{max} 及其对应的点 Q_{max} 。若 $d_{max} \leq D$, 则简化后的路径为 $\{Q_0, Q_{n-1}\}$; 若 $d_{max} > D$, 则将折线分为两部分: $\{Q_0, Q_1, \dots, Q_{max}\}$ 和 $\{Q_{max}, \dots, Q_{n-1}\}$ 。分别对以上过程递归,将递归处理后的结果合并,得到最终剪枝路径。剪枝过程如图9所示。

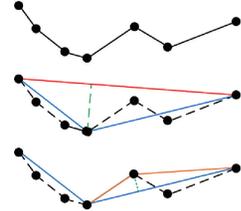


图9 DP算法剪枝

Fig. 9 Pruning with the Douglas-Peucker

为减小路径的弯曲度并使其更顺畅,本文选用了三次B样条曲线^[23]来平滑剪枝后的路径。在每个路径段落间插入三次B样条曲线,把单独的路径点连成一条平滑的曲线。而且,通过调整控制点的位置及权重,三次B样条曲线还可以进一步改善路径的形态。给定路径点列 $\{H_i\}_{i=0}^n$, 构造参数曲线:

$$C(u) = [P_0, P_1, \dots, P_n] \begin{bmatrix} N_{0,3}(u) \\ N_{1,3}(u) \\ N_{2,3}(u) \\ N_{3,3}(u) \end{bmatrix} = \sum_{j=0}^n P_j \cdot N_{j,3}(u) \quad (8)$$

其中, $N_{j,3}$ 为三次B样条基函数,控制点 $\{P_j\}$ ($j = 0, 1, \dots, m$) 通过最小二乘法拟合剪切后的路径。引入曲率约

束优化模型:

$$\min_P \sum_{i=0}^n \| P(u_i) - H_i \|^2 + \gamma \int_0^1 \kappa^2(u) du \quad (9)$$

其中, $\kappa(u) = \frac{\| C'(u) \times C''(u) \|}{\| C''(u) \|^3}$ 为曲率, γ 为平滑

权重系数。以图 10 所示曲线拟合为例,图中实线表示原始路径点连接后的曲线,虚线表示三次 B 样条拟合后的曲线。根据路径中原有的 4 个点,按照三次 B 样条曲线函数求出满足控制的点,并按照顺序进行连线,生成曲线,修改局部路径弯曲度的同时保持路径原有走向基本不变。

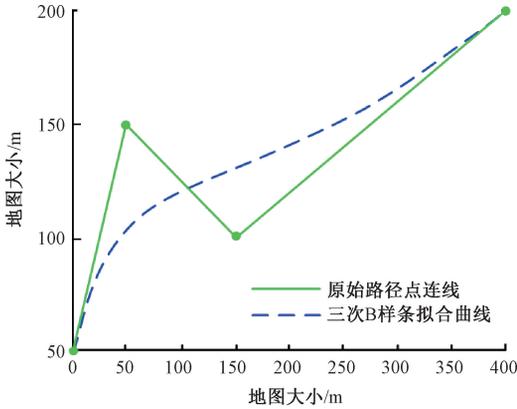


图 10 B 样条曲线

Fig. 10 B-spline curve

3.6 算法实现

基于上述改进思想,为保证智能体快速规划出一条无碰撞的最优路径,提出一种基于 KD 树的目标偏置双 RRT* 算法(KDB-RRT*)。根据图 11 和 12 所示的 KDB-RRT* 算法流程图和 KDB-RRT* 算法示意图,以下是 KDB-RRT* 算法的具体步骤:

- 1) 初始化地图信息、基本参数、起点、终点等,构建随机树,添加起点、终点分别作为起点树 T1、终点树 T2 的起点。
- 2) (以起点树为例)采用动态圆采样策略获得采样点 P_{rand} 。
- 3) 构建 KD 树,利用 KD 树的最近邻搜索找到邻近节点 P_{near} 。
- 4) 在生长引导模型引导下,采用自适应步长生成新节点 P_{new} 。
- 5) 判断生成的新节点 P_{new} 是否满足碰撞检测的要求,即 P_{new} 是否在障碍物上、 P_{near} 于 P_{new} 的连线是否经过障碍物。
- 6) 若满足碰撞检测,利用 KD 树的最近邻搜索找到距离 P_{new} 更近的 $P_{nearest}$ 替换 P_{near} 作为父节点;若不满足碰撞检测,则返回步骤 2) 继续采样。
- 7) 将 $P_{nearest}$ 添加到随机树上。
- 8) 选择到达 P_{new} 路径代价更小的节点,作为下一个可连接的邻近节点,重新布线随机树。
- 9) 判断起点树和终点树是否连接,连接条件为起点树

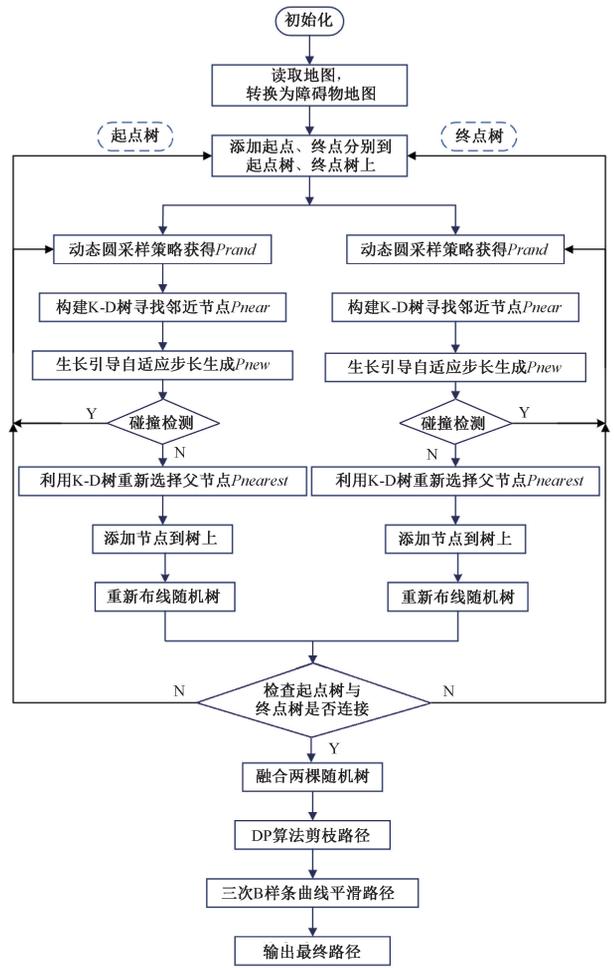


图 11 KDB-RRT* 算法流程图

Fig. 11 KDB-RRT* algorithm flowchart

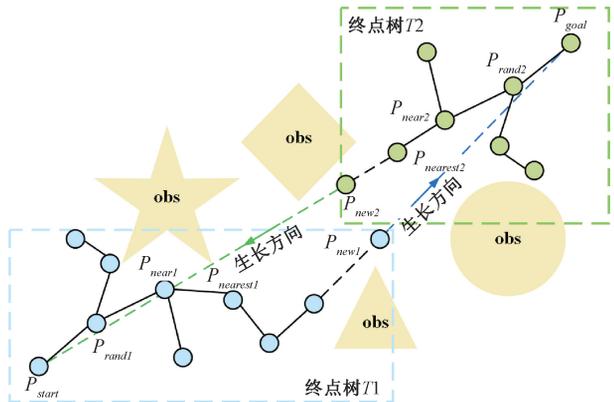


图 12 KDB-RRT* 算法基本原理图

Fig. 12 Basic principle diagram of KDB-RRT*

的 $P_{1_{new}}$ 与终点树的 $P_{2_{new}}$ 距离小于给定的阈值或一棵树的 P_{new} 在另一棵树上。

10) 若已经连接,那么融合两棵树;若没有连接,则返回步骤 2) 继续采样。

11) 使用 DP 算法对已生成的初始路径进行剪枝。

12)采用三次 B 样条曲线的方法平滑规划好的路径。

4 实验分析

4.1 实验设计

为了验证本文提出的 KDB-RRT* 算法的可行性与优越性,利用 MATLAB R2022a 进行仿真实验验证,使用笔记本电脑的系统为 Windows11,处理器为 13th Gen Intel® Core(TM)i7-13700 H 2.40 GHz,运行内存为 16.0 GB。

为验证 KDB-RRT* 算法的可行性,建立如图 13 所示的两种仿真实验环境,设置实验环境地图规模为 500×500 。其中图 13(a)“Z 型”地图选取 (1,500) 为路径起点, (500,1) 为路径终点;图 13(b)“回型”地图选取 (1,500) 为路径起点, (500,1) 为路径终点。地图中黑色为障碍物区域,白色为同行的自由区域。

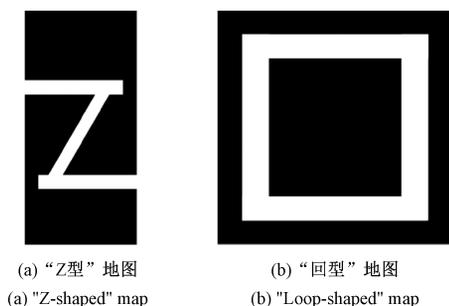


图 13 仿真实验地图 1
Fig. 13 Simulation environment 1

为验证 KDB-RRT* 算法的优越性,建立如图 14 所示的 4 种仿真实验环境,设置实验环境地图规模为 500×500 ,选取 (1,500) 为路径起点, (500,1) 为路径终点。

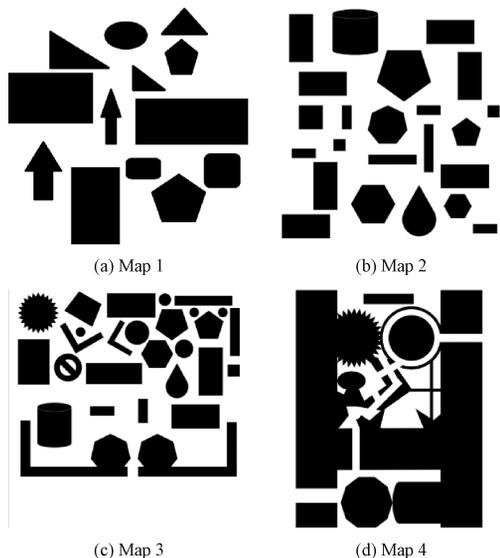


图 14 仿真实验地图 2
Fig. 14 Simulation environment 2

4.2 验证 KDB-RRT* 算法的可行性

基于 4.1 节图 13 搭建的“Z 型”地图和“回型”地图,分

别使用引入 KD 树的 KDB-RRT* 算法和 RRT* 算法执行路径规划任务。目标阈值为 20,初始扩展步长为 30,最大迭代次数为 5 000,各实验均运行 50 次。不同算法的路径规划效果如图 15、16 所示。

从图 15、16 的仿真结果可以看出,KDB-RRT* 算法在两种典型复杂环境(“Z 型”地图、“回型”地图)中均表现出较强的路径规划能力。

“Z 型”地图中,KDB-RRT* 算法规划的路径,如图 15(a)所示,在狭窄的“Z”形通道中完成多次转向,仅在关键转折点处微调方向,表明算法能快速识别狭窄通道的几何特征,减少冗余探索。且始终与障碍物保持精确距离,无局部震荡,这表明算法在非结构化狭窄通道中能有效平衡路径平滑性与安全性。而 RRT* 算法规划的路径,如图 15(b)所示,在通道内频繁转向,形成锯齿状轨迹,源于随机采样导致的冗余节点扩展。且部分路径段与障碍物距离较极端(尤其在通道口处),说明 RRT* 因缺乏动态偏置策略,倾向于生成保守但次优路径。

“回型”地图中,KDB-RRT* 算法规划的路径,如图 16(a)所示,在包含嵌套障碍物的回型环境中,路径成功穿越多层障碍间隙,从起点(左侧绿点)到终点(右侧红点)形成全局优化轨迹。且路径转折处弧度均匀,未出现尖角,适合智能体运动控制。而 RRT* 算法规划的路径,如图 16(b)所示,在中间障碍层附近出现多次折返,反映算法陷入局部极小值,需多次回溯以跳出无效区域。且路径总长度显著长于 KDB-RRT*,尤其在回型结构的外层绕行时消耗额外代价。

在“Z 型”地图中,KDB-RRT* 的平均规划时间(4.10 s)比 RRT* (13.77 s)缩短 70.2%,平均路径长度(1258.25 cm)比 RRT* (1321.92 cm)缩短 4.8%,平均路径节点个数减少 67.5%,节点利用率高 16.2%。“回型”地图中,KDB-RRT* 的平均规划时间(0.85 s)仅为 RRT* (3.04 s)的 28.0%,平均路径长度(646.47 cm)较 RRT* (721.68 cm)减少 10.4%,平均路径节点个数减少 69.6%,节点利用率高 13.5%。核心指标对比结果如表 1 和 2 所示。

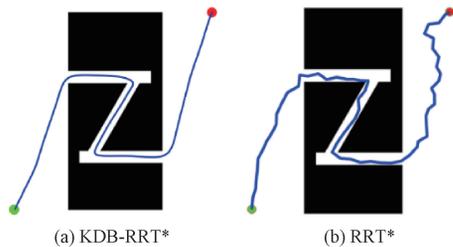


图 15 “Z 型”地图实验结果图
Fig. 15 Experimental results for the “Z-shaped” environment

4.3 验证引入 KD 树的优越性

基于 4.1 节图 14(a)搭建的地图 map 1,分别使用 RRT* 算法和引入 KD 树的 KDB-RRT* 算法执行路径规划

任务。目标阈值为 20,初始扩展步长为 30,最大迭代次数为 5 000,各实验均运行 50 次。不同算法的路径规划效果如图 17 所示。

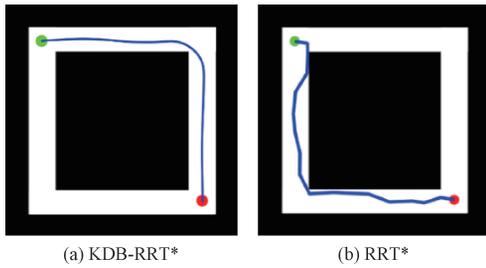


图 16 “回型”地图实验结果

Fig. 16 Experimental results for the “Loop-shaped” environment

表 1 “Z 型”地图对比实验数据

Table 1 Comparative experimental data for the “Z-shaped” environment

算法	平均规划时间/s	平均路径长度/cm	平均路径节点个数	节点利用率/%
KDB-RRT*	4.10	1 258.25	269	54.7
RRT*	13.77	1 321.92	827	38.5

表 2 “回型”地图对比实验数据

Table 2 Comparative experimental data for the “Loop-shaped” environment

算法	平均规划时间/s	平均路径长度/cm	平均路径节点个数	节点利用率/%
KDB-RRT*	0.85	646.47	70	81.4
RRT*	3.04	721.68	230	67.9

图 17(a)和(b)分别表示了未使用 KD 树优化的 RRT* 算法和使用 KD 树优化的 KDB-RRT* 算法的路径结果。从实验结果可看出,引入 KD 树优化的 KDB-RRT* 算法(图 17(b))路径规划中,对于最优节点的寻找次数(除蓝色加粗线条路径外地图内的其他线条数)明显低于 RRT* 算法(图 17(a)),并且路径呈现更高平滑度与方向连续性。对比实验数据表明:KDB-RRT* 的平均规划时间由 RRT* 的 12.95 s 降至 1.13 s,效率提升 91.3%;平均路径总长度由 790.79 cm 缩短至 747.06 cm,降幅 5.5%;平均迭代次数减少 91.0%;平均节点利用率增加 39.39%。直观对应图 17(b)中红绿色细线条数较少与图 17(a)红色细线条数较多,以及图 17(b)路径拐点的稀疏化与图 17(a)路径的密集锯齿状波动。这一优化得益于 KD 树对高维空间的动态分层索引机制,通过动态调整邻近节点连接策略,优先选择父节点到目标点的方向性最优连接,在降低路径复杂度的同时提升搜索效率,使路径更适配智能体工作时对实时性、低能耗及运动稳定性的严苛需求。优化前后核心指标对比如表 3 所示。

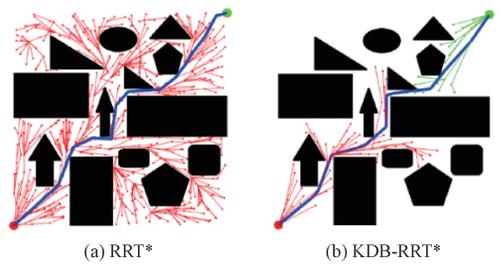


图 17 引入 KD 树的路径优越性对比实验结果

Fig. 17 Comparative experimental results of path superiority with KD-tree integration

表 3 引入 KD 树的路径优越性对比实验数据

Table 3 Comparative experimental data for path superiority with KD-tree integration

算法	平均规划时间/s	平均路径长度/cm	平均迭代次数	节点利用率/%
KDB-RRT*	1.13	747.06	97	80.41
RRT*	12.95	790.79	1 075	41.02

4.4 验证引入 DP 算法剪枝的优越性

基于 4.1 节图 14(a)搭建的 map 1,使用引入 KD 树的 KDB-RRT* 算法执行路径规划任务得到初始路径,同时引入 DP 算法对初始路径进行剪枝处理。引入 DP 算法剪枝处理前后的路径对比效果如图 18 所示。

图 18(a)和(b)分别表示了使用 KDB-RRT* 算法规划后未使用 DP 算法剪枝处理和使用 DP 算法剪枝处理的路径结果。从实验结果可看出,在背景密集黑色几何障碍物的复杂环境下,引入 DP 算法剪枝处理后的路径去除了原始成功规划路径中的冗余拐点,显著提高了路径的整体平滑度与连续性。剪枝处理后的路径在路径长度、拐点个数等方面都得到了提升。路径总长度由平均 769.20 cm 缩短至平均 758.37 cm,减少 10.83 cm;路径拐点个数从平均 18 个减少至平均 5 个,降幅达 72.2%。这一结果表明,通过引入 DP 算法剪枝冗余节点,在降低路径复杂度的同时,还实现了路径长度的优化,有利于减少智能体工作过程中的频繁转向能耗,增强实际导航的稳定性。优化前后路径核心指标的对比如表 4 所示。

4.5 验证 KDB-RRT* 算法的优越性

基于 4.1 节图 14(b)~(d)搭建的不同类型的仿真实验环境,将本文提出的 KDB-RRT* 算法与 RRT* 算法、Bi-RRT 算法、文献[13]中 Improved RRT* 算法等路径规划算法进行算法优越性的对比实验,并从平均规划时间、平均路径长度、平均拐点个数以及规划失败次数 4 个方面的路径规划数据进行分析,以验证 KDB-RRT* 算法的优越性。目标阈值为 20,初始扩展步长为 30,最大迭代次数为 5 000,各实验均运行 50 次。不同算法结果图如图 19~21 所示。

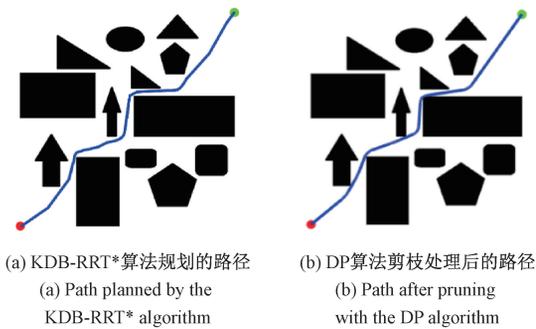


图 18 引入 DP 算法剪枝的路径优越性对比实验结果

Fig. 18 Comparative experimental results of path superiority after DP-based pruning

表 4 引入 DP 算法剪枝的路径优越性对比实验数据

Table 4 Comparative experimental data for path superiority after DP-based pruning

DP 剪枝前后	平均路径长度/cm	平均拐点个数
DP 剪枝前	769.20	18
DP 剪枝后	758.37	5

图 19(a)~(d)表示在同样的地图中 KDB-RRT* 算法与 RRT* 算法、Bi-RRT 算法、文献[13]中 Improved RRT* 算法路径规划的结果。从路径规划结果可以看出,虽然 4 种算法的规划成功率均为 100%,但 KDB-RRT* 算法规划的路径在密集障碍间隙中形成连续平滑曲线,仅在关键瓶颈处出现必要转向;而 RRT* 算法规划的路径在狭窄通道内频繁折返;Bi-RRT 算法规划的路径存在大量的转弯;Improved RRT* 算法规划的路径虽然曲率变化减小了但仍存在一些大角度转弯。KDB-RRT* 算法的平均规划时间仅为 1.11 s,较 RRT* 算法、Bi-RRT 算法、Improved RRT* 算法分别缩短了 95.8%、90.9% 和 97.6%;平均路径长度为 774.89 cm,较 RRT* 算法、Bi-RRT 算法、Improved RRT* 算法分别缩短了 1.1%、2.2%、6.3%;平均拐点数 9 个,较 RRT* 算法、Bi-RRT 算法、Improved RRT* 算法分别缩短了 65.4%、81.2%、11.1%。不同算法基于图 14(b)map 2 对比实验的核心指标如表 5 所示。

表 5 对比实验数据 (map 2)

Table 5 Comparative experimental data (map 2)

算法	平均规划时间/s	平均路径长度/cm	平均拐点数	规划成功率/%
KDB-RRT*	1.11	774.89	9	100
RRT*	26.68	783.60	26	100
Bi-RRT	12.23	791.87	48	100
Improved RRT*	4.86	780.14	10	100

图 14(c)的地图 map 3 较图 14(b)的地图 map 2 更为复杂,图 20(a)~(d)表示在同样的地图中 KDB-RRT* 算法与 RRT* 算法、Bi-RRT 算法、文献[13]中 Improved RRT*

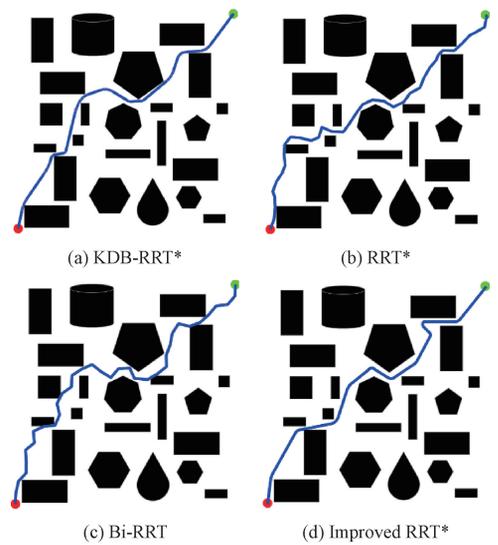


图 19 不同算法 map 2 对比实验结果

算法路径规划的结果。从路径规划结果可以看出 KDB-RRT* 算法规划的路径表现得更优越。KDB-RRT* 算法规划的路径呈连续平滑曲线,仅在关键狭窄通道处进行必要转向。而 RRT* 算法规划的路径局部放大可见锯齿状波动;Bi-RRT 算法规划的路径存在大量的曲率较大的转弯;Improved RRT* 算法规划的路径虽减少了转弯次数但转弯角度仍较大。KDB-RRT* 算法的平均规划时间较 RRT* 算法、Bi-RRT 算法和 Improved RRT* 算法分别缩短了 97.5%、95.1% 和 85.8%;平均路径长度较 RRT* 算法、Bi-RRT 算法、Improved RRT* 算法分别缩短 8.5%、4.8% 和 2.5%;平均拐点数较 RRT* 算法、Bi-RRT 算法、Improved RRT* 算法分别减少 76.0%、93.0% 和 60.0%;且规划成功率为 100%,较 RRT* 算法、Bi-RRT 算法、Improved RRT* 算法分别提升了 32%、26%、0%。不同算法基于图 14(c)map 3 对比实验的核心指标如表 6 所示。

表 6 对比实验数据 (map 3)

Table 6 Comparative experimental data (map 3)

算法	平均规划时间/s	平均路径长度/cm	平均拐点数	规划成功率/%
KDB-RRT*	1.17	792.48	4	100
RRT*	46.76	866.32	25	68
Bi-RRT	24.12	832.49	57	74
Improved RRT*	8.26	812.37	10	100

图 14(d)的地图 map 4 为 4 种地图中最为复杂的一种,图 21(a)~(d)表示在同样的地图中 KDB-RRT* 算法与 RRT* 算法、Bi-RRT 算法、文献[13]中 Improved RRT* 算法路径规划的结果。从 4 种路径规划结果可以看出 KDB-RRT* 算法规划的路径表现得更优越。KDB-RRT* 算法通过 3.1 节~3.5 节改进,在密集障碍物间生成曲率连续的平滑路径(平均拐点数 9 个),仅在几何瓶颈处调整

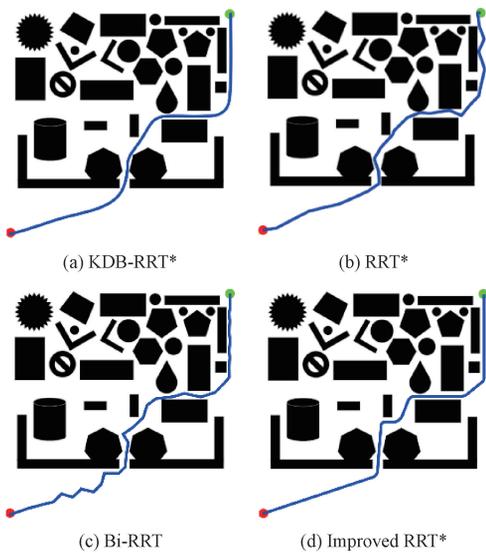


图 20 不同算法 map 3 对比实验结果

Fig. 20 Comparative experimental results of different algorithms in map 3

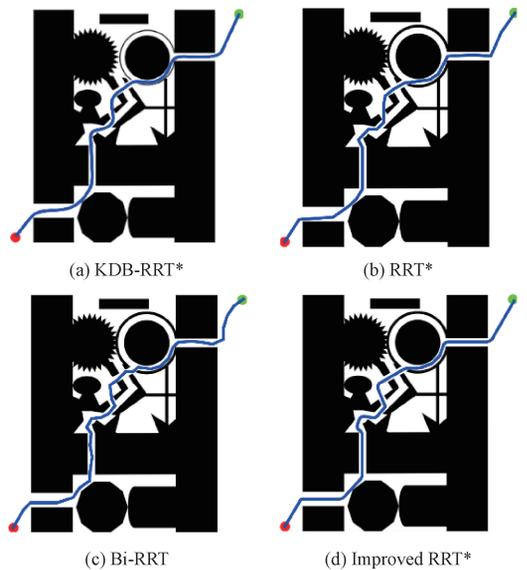


图 21 不同算法 map 4 对比实验结果图

Fig. 21 Comparative experimental results of different algorithms in map 4

方向,且规划成功率 100%;而 RRT* 算法因均匀随机采样特性,在狭窄通道内出现路径震荡(24 个拐点),规划成功率仅为 28%;Bi-RRT 算法的尽管增加了算法的搜索效率但仍产生大量弯折(73 个拐点),规划成功率为 52%;Improved RRT* 算法虽然改进扩展节点的选择策略降低了算法随机性但规划出的路径仍存在拐点较多的情况(15 个拐点),规划成功率 98%。KDB-RRT* 算法平均规划时间仅为 1.96 s,较 RRT* 算法、Bi-RRT 算法、Improved RRT* 算法分别缩短 97.9%、95.9%和 77.2%;平均路径长度为 791.88 cm,较 RRT* 算法、Bi-RRT 算法、Improved RRT* 算法,分别缩短 8.6%、6.4%、2.2%;平均拐点数 9 个,较 RRT* 算法、Bi-RRT 算法、Improved RRT* 算法,分别减少 62.5%、87.7%和 66.7%。不同算法基于图 14(d) map 4 对比实验的核心指标如表 7 所示。

表 7 对比实验数据 (map 4)

Table 7 Comparative experimental data (map 4)

算法	平均规划时间/s	平均路径长度/cm	平均拐点数	规划成功率/%
KDB-RRT*	1.96	791.88	9	100
RRT*	94.77	866.13	24	28
Bi-RRT	48.57	842.31	73	52
Improved RRT*	8.58	809.31	15	98

综上,本文所提出的 KDB-RRT* 算法在 3 种不同类型的障碍物环境中均展现出更平滑的路径,且与障碍物保持安全距离。从对比实验数据中可以看出,KDB-RRT* 算法在确保成功规划路径的基础上,在平均规划时间、平均路径长度、平均拐点数上都有显著的改进。这意味着本文提出的 KDB-RRT* 算法规划时更加高效,需要更少的节点来找

到解决方案,能够快速响应路径规划请求。

4.6 基于 KDB-RRT* 算法的智能体路径规划实验

为了进一步验证本文所提出 KDB-RRT* 算法的可行性,使用如图 22 所示的塔克创新 TARKBOT 系列 ROS 机器人作为实验中的智能体,采用 ROS 提供的即时定位与地图构建工具进行环境地图的构建,并通过自适应蒙特卡洛算法实现智能体的自主定位,并搭建了如图 23 所示的两种实验场景进行实验验证。以图 23 中最下方 UGV 的位置为原点,UGV 的右面为 x 轴正方向,UGV 的前面为 y 轴正方向,其中绿色标注物为 UGV(起点),蓝色标注物为目标位置 Goal(终点),黄色标注物为障碍物。在这两个场景中保持起点和终点不变,只改变障碍物的数量和分布,以观察不同场景下算法的规划能力。

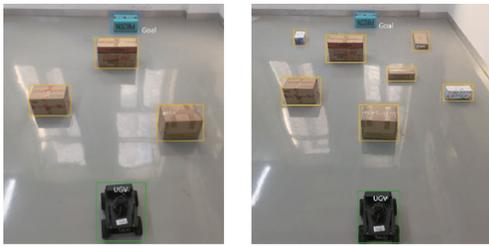


图 22 塔克创新 TARKBOT 系列 ROS 机器人

Fig. 22 XTARK TARKBOT series ROS robots

图 24 展现了图 23 所示的两种场景下进行路径规划的结果,结果表明不管是哪一种场景下,智能体总能在不发生碰撞的前提下规划出一条接近最优的路径。

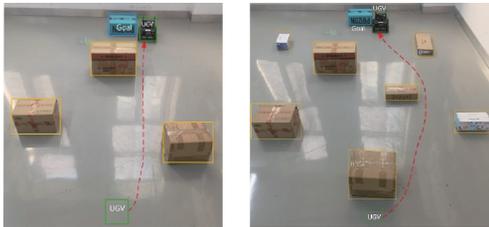
表 8 记录了 20 组在以上两种场景下实验的平均规划路径长度、平均规划时间以及规划过程中发生碰撞的次数。



(a) 场景一 (a) Scenario 1
(b) 场景二 (b) Scenario 2

图 23 智能体路径规划实验环境设置

Fig. 23 Experimental environment setup for intelligent agent path planning



(a) 场景一 (a) Scenario 1
(b) 场景二 (b) Scenario 2

图 24 智能体路径规划结果

Fig. 24 Results of intelligent agent path planning

从表中的数据可以得出,对于场景一,在没有发生碰撞的前提下,平均规划路径长度约为 12.78 m、平均规划时间约为 3.28 s;对于场景二,在没有发生碰撞的前提下,平均规划路径长度约为 13.43 m、平均规划时间约为 5.46 s。

表 8 智能体路径规划实验数据

Table 8 Experimental data for intelligent agent path planning

场景	平均路径长度/m	平均规划时间/s	是否发生碰撞
场景一	12.78	3.28	否
场景二	13.43	5.46	否

该实验结果表明本文提出的 KDB-RRT* 算法能够在不同环境下为智能体规划出相对较优的路径,验证了 KDB-RRT* 算法在不同场景中对智能体路径规划的可行性。

5 结 论

针对复杂环境下智能体路径规划存在的收敛速度慢、路径冗余度高及平滑性不足等问题,本文提出基于 KD 树的目标偏置双 RRT* 算法(KDB-RRT*)。该算法在引入双向搜索策略的 RRT* 基础上,构建了多维度改进框架:采用 KD 树结构优化近邻查询,降低了节点搜索复杂度;引入动态圆采样策略与引力场引导模型,有效平衡全局探索与局部精细化搜索;创新性地结合 Sigmoid 函数实现自适应

步长动态调整,提升密集障碍区域的避碰能力;通过 DP 算法剪枝与三次 B 样条曲线优化,消除路径冗余节点并增强运动连续性。

为验证算法的可行性以及优越性,本文搭建了“Z 型”、“回型”以及 4 种不同复杂程度的地图环境,进行多次仿真对比实验,并在 ROS 机器人进行路径规划实验,进一步验证该算法的实用性和有效性。本文只分析了相对理想的静态环境进行算法的验证,忽略了实际工作环境中的动态障碍、地面粗糙度等影响。未来的研究将针对真实的工作环境展开仿真和验证,以提高算法在更复杂环境中的适应性和判断性。

参考文献

- [1] AGGARWAL S, KUMAR N. Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges [J]. Computer Communications, 2020, 149: 270-299.
- [2] 郭志军,尹亚昆,李亦轩,等.融合改进 A* 和 TEB 算法的移动机器人路径规划[J].河南科技大学学报(自然科学版),2023,44(4):57-65,7.
GUO ZH J, YIN Y K, LI Y X, et al. Mobile robot path planning integrating improved A* and TEB algorithms[J]. Journal of Henan University of Science and Technology (Natural Science Edition), 2023, 44(4):57-65,7.
- [3] 孙金玉,付主木,陶发展,等.改进人工势场的智能车避障算法研究[J].河南科技大学学报(自然科学版),2022,43(5):28-34,41,5-6.
SUN L Y, FU ZH M, TAO F ZH, et al. Research on intelligent vehicle obstacle avoidance algorithm based on improved artificial potential field[J]. Journal of Henan University of Science and Technology (Natural Science Edition), 2022, 43(5):28-34,41,5-6.
- [4] 左国玉,关海山,郑榜贵.基于多次采样启发式策略的改进 RRT 路径规划算法[J].计算机测量与控制,2024,32(7):280-287.
ZUO G Y, GUAN H SH, ZHENG B G. Improved RRT path planning algorithm based on multi-sampling heuristic strategy [J]. Computer Measurement & Control, 2024, 32(7):280-287.
- [5] 王敏,石明航,韩晔昕,等.基于改进蚁群算法的无人艇全局航路规划模型[J].河南科技大学学报(自然科学版),2024,45(5):50-56,118.
WANG M, SHI M H, HAN Y X, et al. Global route planning model for unmanned surface vehicle based on improved ant colony algorithm[J]. Journal of Henan University of Science and Technology (Natural Science Edition), 2024, 45(5):50-56,118.
- [6] 王雅清,倪晓昌,李静,等.移动机器人路径规划算法研究进展[J].智能计算机与应用,2024,14(11):211-217.
WANG Y Q, NI X CH, LI J, et al. Research progress of mobile robot path planning algorithms [J].

- Intelligent Computer and Applications, 2024, 14(11): 211-217.
- [7] 邓益昭,涂海燕,宋明俊.基于改进 RRT 的机器人路径规划算法[J].组合机床与自动化加工技术,2024(6): 6-11.
DENG Y ZH, TU H Y, SONG M J. Robot path planning algorithm based on improved RRT [J]. Modular Machine Tool & Automatic Manufacturing Technique, 2024(6):6-11.
- [8] MENG B H, GODAGE I S, KANJ I. RRT*-based path planning for continuum arms[J]. IEEE Robotics and Automation Letters, 2022, 7(3): 6830-6837.
- [9] WANG W, GAO H, YI Q, et al. An improved RRT* path planning algorithm for service robot[C]. 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC). IEEE, 2020, 1: 1824-1828.
- [10] CHEN J, ZHAO Y, XU X. Improved RRT-connect based path planning algorithm for mobile robots[J]. IEEE Access, 2021, 9: 145988-145999.
- [11] 谢高杨,房立清,李亚男,等.基于改进 RRT 算法的无人靶车路径规划研究[J].火炮发射与控制学报,2024, 45(3):80-86.
XIE G Y, FANG L Q, LI Y N, et al. Research on unmanned target vehicle path planning based on improved RRT algorithm[J]. Journal of Gun Launch & Control, 2024, 45(3):80-86.
- [12] LIM S, SANGROK S. Safe trajectory path planning algorithm based on RRT* while maintaining moderate margin from obstacles [J]. International Journal of Control, Automation and Systems, 2023, 21(11): 3540-3550.
- [13] 顾子倡,刘宇,岳广,等.基于改进 RRT 算法的快速路径规划[J].兵器装备工程学报,2022,43(10):294-299.
GU Z L, LIU Y, YUE G, et al. Fast path planning based on improved RRT algorithm [J]. Journal of Ordnance Equipment Engineering, 2022, 43(10): 294-299.
- [14] 马晓群,王昊,刘磊,等.基于 IRRT-Connect 的自适应路径规划算法[J].电子测量技术,2024,47(15):82-88.
MA X Q, WANG H, LIU L, et al. Adaptive path planning algorithm based on IRRT-Connect[J]. Electronic Measurement Technology, 2024, 47(15):82-88.
- [15] 李文君,李忠伟,罗偲.基于 RRT* 算法的移动机器人安全光滑路径生成[J].电子测量技术,2024,47(2): 51-60.
LI W J, LI ZH W, LUO C. Safe and smooth path generation for mobile robot based on RRT algorithm[J]. Electronic Measurement Technology, 2024, 47(2):51-60.
- [16] 刘文光,刘浩伟,罗通,等. RRT 算法路径优化及仿真验证[J].重庆理工大学学报(自然科学),2022, 36(11):1-7.
LIU W G, LIU H W, LUO T, et al. RRT algorithm path optimization and simulation verification [J]. Journal of Chongqing University of Technology (Natural Science), 2022, 36(11):1-7.
- [17] 陈文龙,时宏伟.基于 KD 树改进的 DBSCAN 聚类算法[J].计算机系统应用,2022,31(2):305-310.
CHEN W L, SHI H W. Improved DBSCAN clustering algorithm based on KD-tree[J]. Computer Systems & Applications, 2022, 31(2):305-310.
- [18] 薛丁文,李建中.基于 KD 树的 k-means 聚类算法优化[J].智能计算机与应用,2021,11(11):194-197.
XUE D W, LI J ZH. Optimization of k-means clustering algorithm based on KD-tree[J]. Intelligent Computer and Applications, 2021, 11(11):194-197.
- [19] SUN Y, ZHAO T, YOON S, et al. A hybrid approach combining R*-tree and k-d trees to improve linked open data query performance [J]. Applied Sciences, 2021, 11(5): 2405.
- [20] XIU X J, ZHANG J. Grid k-d tree approach for point location in polyhedral data sets - application to explicit MPC[J]. International Journal of Control, 2020, 93(4): 872-880.
- [21] 陈耀威.复杂环境下自动泊车路径规划与跟踪控制研究[D].吉林:吉林化工学院,2024.
CHEN Y W. Research on path planning and tracking control for automatic parking in complex environments[D]. Jilin: Jilin Institute of Chemical Technology, 2024.
- [22] 肖洒,吕勇明,吴海彬.一种基于 DP-KMP 的机器人避障交互学习方法[J].仪器仪表学报,2024,45(11): 65-78.
XIAO S, LYU Y M, WU H B. An interactive learning method for robot obstacle avoidance based on DP-KMP[J]. Chinese Journal of Scientific Instrument, 2024, 45(11):65-78.
- [23] 陈都,侯明,张学东.改进 RRT 结合 B 样条的移动机器人路径规划研究[J].电子测量技术,2022,45(23): 38-44.
CHEN D, HOU M, ZHANG X D. Path planning for mobile robots based on improved RRT combined with B-spline [J]. Electronic Measurement Technology, 2022, 45(23):38-44.

作者简介

魏文卿,硕士研究生,主要研究方向为移动机器人路径规划。

E-mail: Eeqing@163.com

魏坤(通信作者),博士,讲师,主要研究方向为智能网联汽车环境感知及运动规划。

E-mail: yangquanweikun@163.com

张建辉,硕士研究生,主要研究方向为智能车目标检测。

E-mail: Xiaohui009819@163.com