

DOI:10.19651/j.cnki.emt.2313652

## 融合三步平滑与改进 RRT 算法的无人机路径规划\*

施成龙<sup>1</sup> 行鸿彦<sup>1,2</sup> 王水璋<sup>2</sup> 姜华生<sup>1</sup>

(1.南京信息工程大学电子与信息工程学院 南京 210044; 2.南京多基观测技术研究院 南京 211500)

**摘要:**为解决路径规划领域中原始 RRT 所具有规划时间长、路径不平滑、路径代价大的缺点,本文先在原始 RRT 算法的基础上加入了动态步长策略以及嵌入了 Dijkstra 算法对效率进行改进,然后在得到的路径上分别加入下样本平滑、上样本平滑和关键点平滑对路径平滑度和路径代价进行改进。MATLAB 实验表明,本文算法在规划时间上较传统 RRT 提升 45% 左右,较 Astar、RRTstar、GA 算法有着分别 30%~70% 不等的领先;在路径长度方面,本文算法较传统 RRT 有着近 40% 的提升,相较其他算法也有着不同程度的领先。由此可以得出,本文所提方法,可以较好地应用于路径规划。

**关键词:** RRT 算法;路径平滑;路径规划;Dijkstra

**中图分类号:** V249;V279;TP18 **文献标识码:** A **国家标准学科分类代码:** 510.80

## UAV path planning combing three-step smoothing and improved RRT algorithm

Shi Chenglong<sup>1</sup> Xing Hongyan<sup>1,2</sup> Wang Shuizhang<sup>2</sup> Lou Huasheng<sup>1</sup>

(1. School of Electronic and Information Engineering, Nanjing University of Information Science and Technology, Nanjing 210044, China; 2. Nanjing Institute of Multi-based Observation Technology, Nanjing 211500, China)

**Abstract:** In order to solve the shortcomings of the original RRT in the field of path planning, such as long planning time, uneven path and high path cost, this paper first added dynamic step strategy and embedded Dijkstra algorithm on the basis of the original RRT algorithm to improve efficiency. Then, lower sample smoothing, upper sample smoothing and key point smoothing are added to the obtained path to improve the path smoothness and path cost. MATLAB experiments show that the proposed algorithm improves the planning time by about 45% compared with the traditional RRT algorithm, and is 30% to 70% ahead of Astar, RRTstar and GA algorithms respectively. In terms of path length, the proposed algorithm has a nearly 40% improvement over the traditional RRT, and it also has different degrees of lead compared with other algorithms. It can be concluded that the method proposed in this paper can be well applied to path planning.

**Keywords:** RRT algorithm; path smoothing; path planning; Dijkstra

## 0 引言

无人机的路径规划即在一定区域范围内、一定的约束条件下根据先验知识和实时传感信息,给定无人机的初始点和目标点,让其自动搜索出一条安全可靠的飞行路线,其中要考虑避障和最短路线问题。

LaValle 提出的快速扩展随机树<sup>[1]</sup>(rapidly-exploring random tree, RRT)算法通过对状态空间中的采样点进行碰撞检测,避免了对空间的建模,能够有效地解决高位空间和复杂约束下的路径规划问题,在无人机路径规划问题里

面有着良好的应用前景。但基本 RRT 算法的节点随机性较强,所有的节点需要遍历,导致路径搜索效率低,且路径通常不是很平滑,整体性较差。

由于 RRT 算法存在着遍历时间长、冗余节点多、路径不平滑等缺点,国内外许多学者都致力于对 RRT 算法进行改进。文献[2]提出了 RRT-connect 算法,在起始点和终点生成两棵随机树来大大减短遍历时间,但是生成的路径拐点太多,不够平滑;文献[3]通过目标偏向策略及气味扩散的方法向目标节点进行扩散,并且使用了三次 B 样条曲线对路径进行了优化,取得了较好的效果;张海波等<sup>[4]</sup>将动态

收稿日期:2023-05-17

\* 基金项目:国家自然科学基金(62171228)、国家重点研发计划(2021YFE0105500)项目资助

窗口算法(dynamic window approach, DWA)与 RRT 算法相结合,实现了动态环境中的路径规划,但是其路径平滑性不够;文献[5]在 RRTstar 的基础上加入启发式动态约束思想,实现了低成本形成较优的运动路径;文献[6]将可变采样区域与目标偏置策略相结合,在时间和效率上达到了较好的效果;Wang 等<sup>[7]</sup>提出了一种基于可变采样域的改进 RRT 算法和一种贪心生长策略的映射压缩算法(variable sampling and greedy growing strategy and map compression RRT, VSDGC-RRT),提高了采样节点的利用率,减少了迭代次数,缩短了路径规划的时间。

针对 RRT 算法搜索效率低、路径代价大的缺点,分析传统的 RRT 算法,进行改进研究,加入无人机的约束条件,将 RRT 与 Dijkstra 相结合,得到路径的最优解,采用三步平滑策略对路径进行平滑,得到最终规划路径,的是。通过仿真验证本文提出的改进 RRT 算法在二维、三维环境下的应用效果。

## 1 无人机路径规划的基本理论

无人机的路径规划主要包括无人机性能约束、路径规划算法、路径优化 3 个部分,本文所采用的算法为改进的 RRT 算法,路径优化提出了一种新的三步平滑策略。

### 1.1 无人机性能约束

无人机与机器人、无人车在实际中的区别在于,无人机在飞行过程中需要受到自身的飞行性能约束,如最大航程、最大转角、最大爬升角等,在进行路径规划时,需要充分考虑各类的约束条件。下面是本文所考虑的性能约束:

#### 1) 最大航程

如果把整条航线看成是  $n$  条长度为  $l_i$  的线段,那么无人机最大航程约束为<sup>[8]</sup>:

$$\sum_{i=1}^n l_i \leq L_{\max} \quad (1)$$

其中,  $L_{\max}$  是最大飞行航程。

#### 2) 最大转角

最大转角是指无人机在路径中切换方向时,原方向与现方向的夹角,公式如下:

$$\arccos \frac{a_i^T a_{i+1}}{|a_i| |a_{i+1}|} \leq \varphi_{\max} \quad (2)$$

其中,  $a_i$  为无人机第  $i$  短航线的水平投影向量,  $\varphi_{\max}$  是最大转角。

#### 3) 最大爬升/俯冲角

最大爬升/俯冲角时无人机在向上/向下时与竖直方向形成的夹角<sup>[9]</sup>。

$$\arctan \frac{|z_{i+1} - z_i|}{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}} \leq \vartheta_{\max} \quad (3)$$

其中,  $x_i$ 、 $y_i$ 、 $z_i$  是无人机在  $i$  时刻的三维坐标点,  $\vartheta_{\max}$  是最大爬升角/俯冲角<sup>[10]</sup>。

此外,无人机还受到其他的性能约束,比如最大的飞行

高度、续航时间等,因其受无人机质量、品牌的影响,本文暂不考虑这些性能约束。

### 1.2 基本 RRT 算法

RRT 算法是一种基于概率采样的增量式搜索算,搜索所需边界点少,搜索范围趋向于未知区域,可达性好,具有概率完整性,无需对空间进行建模,能高速搜索复杂高维空间,适用于非完整与高自由度状态下的路径规划,在无人机领域有着良好的应用。其基本实现步骤如下<sup>[11]</sup>:

步骤 1) 初始化:确定起点和终点,将起点作为根节点,创建一棵空的 RRT 树;

步骤 2) 随机采样:在空间中随机生成新的点  $r$  (50% 为随机点, 50% 为目标点,

目的是增强 RRT 向 goal 点生成的导向性);

步骤 3) 最近邻搜索:在 RRT 树中搜索最近的节点,将其作为当前节点,最近的依据是点  $r$  与 RRT 中的哪一个节点的欧氏距离最小;

步骤 4) 扩展树:从当前节点向目标节点按照步长采取一步扩展,生成一个新节点。

步骤 5) 碰撞检测:检查新节点与上一节点的连线是经过障碍物,如果经过障碍物,则返回步骤 3),若没有,则将新节点添加到 RRT 中;

步骤 6) 判断是否到达目标:如果新节点与目标节点距离在一个预定义的范围,则说明到达目标节点,算法结束。

步骤 7) 重复步骤 2)~6),直到到达目标或达到最大迭代次数。

### 1.3 Dijkstra 算法

Dijkstra 算法是一种用于解决单源最短路径问题的算法,由其提出者迪杰斯特拉得名<sup>[12]</sup>。因其贪心思想,有着较强的目标导向型,是一种启发式算法。Dijkstra 算法的核心思想是根据距离起始点最短路程长度递增的次序对路径进行迭代,最终得到从起始点到目标点之间的最短路径。其具体实现步骤如下:

步骤 1) 初始化:将起点  $s$  到其他所有节点的距离都设为无穷大,  $s$  到自己的距离为 0。

步骤 2) 同时,设一个集合  $S$ ,用于存放已经找到最短路径的节点。

步骤 3) 选择节点:从未处理的节点中选择一个距离起点  $s$  最近的节点  $u$ ,将其加入集合  $S$ 。

步骤 4) 更新距离:对于  $u$  的所有邻居节点  $v$ ,如果从  $s$  出发,经过  $u$  到  $v$  的距离比已经记录的最短距离要短,那么更新  $v$  的最短距离。

步骤 5) 重复:重复执行步骤 2) 和 3),直到所有节点都被加入集合  $S$ 。此时,集合  $S$  中保存了起点  $s$  到所有节点的最短路径。

具体来说,第 3) 步中的更新操作包括以下两个步骤:

1) 计算新的距离。从  $s$  到  $u$  的距离加上从  $u$  到  $v$  的距

离,即为从  $s$  到  $v$  的距离。

2)更新最短距离。如果从  $s$  出发到  $v$  的距离比已有的最短距离更短,那么更新  $v$  的最短距离,并将  $v$  的前驱节点设置为  $u$ 。

#### 1.4 动态步长策略

步长是影响 RRT 算法搜索时间的一个重要因素,在与障碍物无碰撞的情况下,步长越大,搜索时间就越短。但在实际有障碍物的时候,步长太大就会与障碍物碰撞;步长太小就会产生过多的冗余节点,同时大大增加搜索的时间。基于此问题,最好的解决办法就是根据周围的环境和障碍物的分布情况来设置动态步长。

本文选取的动态步长策略为:

1)设置初始步长  $step$

2)在第一次扩展时,判断新节点与起点的连线之间是否存在障碍物,若否,则去新节点为当前的起始点,向下一个点扩展;若是,则执行步骤 3)

3)根据下式选取新步长进行扩展,其中, $S$  为以距离随机点最近的树节点和目标点为对角线的矩形的面积; $m$  为矩形中的障碍物的面积。

$$step = step + (1 - m/S) \times step \quad (4)$$

根据上式可知,障碍物占矩形面积越大,动态步长就越小,这样就能够快速搜索到目标点。

#### 1.5 优化路径的三步平滑方法

1)下样本平滑:从起点开始,沿着连接树搜索,检查当前节点与下一个节点的连线与障碍物是否碰撞,若不碰撞则继续检查再下一个节点;若碰撞,则将上一个节点标记为当前的断点,并将前一个点与当前最后一个断点连接起来,本质上下样本优化是贪婪算法的一种变体,优化示意图如图 1 所示。其中,有点标记的为原始路径,虚线是为了显示两个点之间的连线区域与障碍物有碰撞因此把断点舍弃,得到了最终如图所示的短路径。可以看出,经过下样本优化路径比原始路径更短,但仍然需要更进一步的优化。

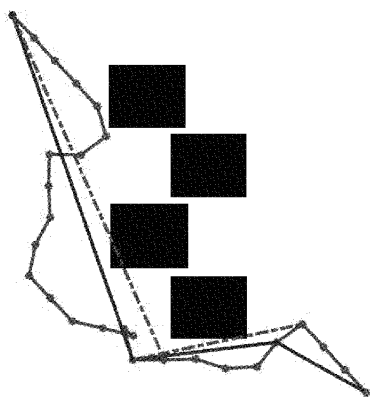


图 1 下样本优化

2)上样本平滑:经过下样本优化过的路径变短,但节点间仍然是之字形连接,离障碍物不够近,特别是在拐角处,

这在无人机的飞行中会多损失能量。与下样本相比,上样本优化是生成更多更接近障碍物且整体路径更短的节点。

定义如下:将下样本轨迹点定义为  $DST$  (down sample trajectory), 上样本轨迹点定义为  $UST$  (up sample trajectory), 令  $p$  代表一个路径的节点, 则  $DST_n = \{p_1^{DS}, p_2^{DS}, \dots, p_n^{DS}\}$ ,  $UST_m = \{p_1^{US}, p_2^{US}, \dots, p_m^{US}\}$ 。用迭代的方式来获取点, 令  $DST=UST$  作为第一次迭代, 定义向下采样轨迹点的累计距离之和为  $C_s$ ,  $C_{s_i}$  的定义则有:

$$C_{s_i} = \sum_{i=1}^i p_i^{DS}, \forall i = 1, 2, \dots, n \quad (5)$$

根据起点和终点之间的距离对随机两个距离进行采样, 有:

$$rand_{1,2} = x \cdot C_{s_{i=n}}, x \sim U(0,1) \quad (6)$$

其中,  $rand_{1,2}$  满足  $rand_1 \leq rand_2$ 。然后通过迭代将  $\gamma_1$  和  $\gamma_2$  插入到  $p_i^{US}$  和  $p_{j+1}^{US}$  之间的区域, 其中  $i$  满足式(7),  $j$  满足式(8),  $\gamma_1$  和  $\gamma_2$  则有式(10)和(11)给出, 同时  $\gamma_1$  和  $\gamma_2$  连接的路径也要检测是否与障碍物碰撞。

$$C_{s_i} \leq rand_1 < C_{s_{i-1}} \quad (7)$$

$$C_{s_j} \leq rand_2 < C_{s_{j+1}} \quad (8)$$

$$a_1 = \frac{rand_1 - C_{s_i}}{C_{s_{i+1}} - C_{s_i}} \quad (9)$$

$$a_2 = \frac{rand_2 - C_{s_j}}{C_{s_{j+1}} - C_{s_j}} \quad (10)$$

$$\gamma_1 = (1 - a_1) \times p_i^{US} + a_1 \times p_{i+1}^{US} \quad (11)$$

$$\gamma_2 = (1 - a_2) \times p_j^{US} + a_2 \times p_{j+1}^{US} \quad (12)$$

则可以得到新的上样本点  $UST'_m$ , 有:

$$UST'_m = \{p_1^{US}, \dots, p_i^{US}, \gamma_1, \gamma_2, p_{j+1}^{US}, \dots, p_m^{US}\} \quad (13)$$

然后使得  $UST'_n = UST'_m$ , 取尽  $m, n$  的值, 由于这是一种采样迭代的方法, 因此迭代次数越多, 效果越好, 图 2、3 分别是迭代次数 10 和 999, 迭代次数与路径的长度关系结果由图 4 给出, 各图中的粗线为原路径, 细线是对应迭代次数之后的上样本平滑效果。

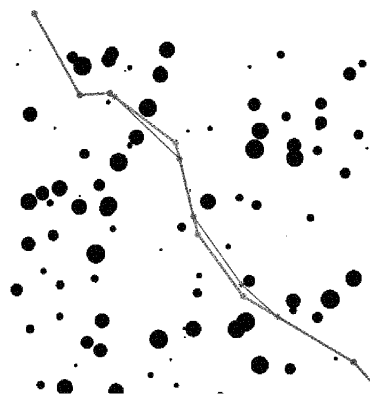


图 2 迭代 10 次

3)关键点平滑:在前两步优化之后,路径整体比较平滑、路径长度也比较小,但是对于某些点还不是整体的最优解,因此需要对关键点进行优化,B 样条曲线是一种平滑的

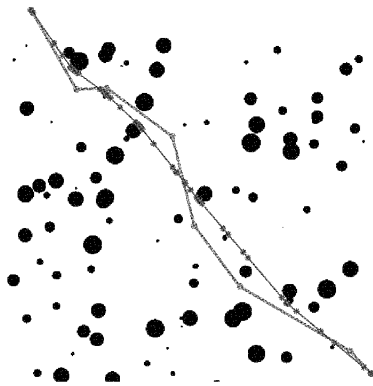


图 3 迭代 999 次

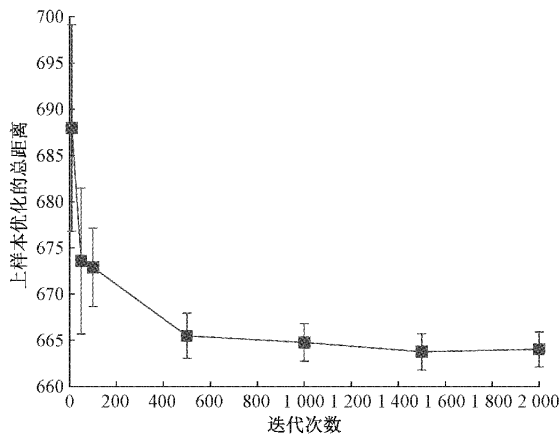


图 4 迭代次数与路径长度的关系

曲线,是一种针对点的平滑,其本质是贝塞尔曲线的一种特例,是在贝塞尔曲线的优点的基础上克服了缺乏局部性质、连续性差的缺点,常用于路径平滑<sup>[13]</sup>。通常用法是,通过控制点对路径进行  $n$  次 B 样条曲线处理,三次 B 样条曲线则是  $n=3$  的情况。B 样条曲线的定义如下:

把一条轨迹看成是  $n$  个点的组成,各个点的坐标是  $(x_i, y_i)$ ,其中  $i$  为  $[0, n]$  的整数,那么函数  $S(x)$  就是  $K$  阶连续样条函数,  $S(x)$  满足以下两个条件:

(1) 在  $[x_{i-1}, x_i] (i=1, 2, \dots, n)$  内,  $S(x)$  是具有至少  $K$  阶的连续函数

(2)  $i$  取  $[1, n-1]$  处成立

(3)  $y_i = S(x_i)$  下面对 B 样条曲线进行定义,给定  $m+n+1$  个平面或控制点  $P_i, i$  为  $[0, m+n]$  的整数,则  $n$  次的参数曲线段定义为:

$$S(x) = \sum_{i=0}^n P_{i+k} G_{i,n}(t), t \in [0, 1] \quad (14)$$

其中,基函数  $G_{i,n}(t)$  定义为

$$G_{i,n}(x) = \frac{1}{n!} \sum_{j=0}^n (-1)^j C_{n+1}^j (t+n-i-j)^n \quad (15)$$

其中,  $t \in [0, 1], i=0, 1, \dots, n$

三次 B 样条是取  $n=3$ , 则有三次 B 样条曲线的基函数如下:

$$\begin{cases} G_{0,3}(t) = \frac{1}{6}(-t^3 + 3t^2 - 3t + 1) \\ G_{1,3}(t) = \frac{1}{6}(3t^3 - 6t^2 + 4) \\ G_{2,3}(t) = \frac{1}{6}(-3t^3 + 3t^2 + 3t + 1) \\ G_{3,3}(t) = \frac{1}{6}t^3, \end{cases} \quad t \in [0, 1] \quad (16)$$

则可得  $P_{0,3}(t)$  的表达式:

$$P_{0,3}(t) = \frac{1}{6} [\mathbf{1} \ t \ t^2 \ t^3] \begin{pmatrix} 1 & 4 & 1 & 0 \\ -3 & 0 & 3 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{pmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} \quad (17)$$

其中,  $P_i$  的定义为:

$$\begin{cases} P_{0,3}(0) = \frac{1}{2}(P_2 - P_0) \\ P_{0,3}(1) = \frac{1}{2}(P_3 - P_1) \\ P''_{0,3}(0) = P_2 - 2P_1 + P_0 \\ P''_{0,3}(1) = P_3 - 2P_2 + P_1 \end{cases} \quad (18)$$

由上述公式可得三次 B 样条的控制点及优化曲线。再结合前文所描述的双向插点法,可对三次 B 样条曲线进行进一步的优化,取得更好的平滑效果。图 5 是使用三次 B 样条曲线进行路径平滑处理的示意图,折线线段为原始路径,相对平滑的线段为经过三次 B 样条平滑之后的路径:

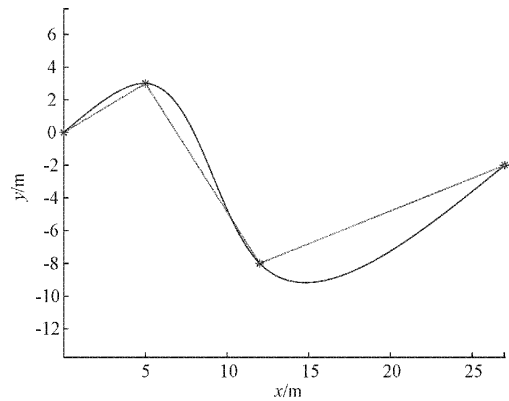


图 5 三次 b 样条优化

## 2 RRT 算法改进

为解决 RRT 算法的随机性过大、搜索时间长的问题,现嵌入 Dijkstra 算法,利用其目标导向性来弥补这两个缺点。RRT 算法和 Dijkstra 算法的融合是通过在 RRT 算法的基础上加入 Dijkstra 算法的思想来实现的<sup>[14]</sup>。同时,为解决 Dijkstra 算法得出的路径的不平滑问题,加入改进的三次 B 样条曲线进行平滑。具体的算法流程如下:

步骤 1) 初始化: 生成起点作为树的根节点, 并将其加入树结构中。

步骤 2) 随机生成新点: 根据动态步长策略, 生成一个新点, 作为当前要连接的目标点。

步骤 3) 找到最近的点: 在树结构中找到一个与目标点最近的节点, 作为新点的父节点。

步骤 4) 使用 Dijkstra 算法更新节点: 从新点开始, 使用 Dijkstra 算法向树结构中更新节点的距离。

步骤 5) 连接点: 连接新点和其父节点, 将新点加入树结构中。

步骤 6) 判断是否到达目标点: 如果新点已经到达目标点, 则结束算法; 否则, 返回步骤 2)。

步骤 7) 构建路径: 使用树结构上的连接关系构建路径。

步骤 8) 得到的路径作为输入进行三步平滑优化。

### 3 实验与仿真

为验证改进的 RRT 算法的有效性, 本文采取在 MATLAB 2020b 中搭建仿真平台, 构建地图环境。本文实验将无人机视为一个质点, 并每种场景进行 50 次路径规划实验<sup>[15]</sup>。实验应用的主机配置为酷睿 i5, 主频 2.9 GHz, 16 GB 内存。实验构建的栅格地图尺寸为  $500 \times 500$ , 几种算法的起始点均为  $[10, 10]$ , 终点均为  $[490, 490]$ 。地图中黑色部分为障碍物, 无人机的最大转角设置为  $\pi/2$ , 初始步长设为 20。

同时为了体现本文算法的优越性, 将本文算法与原始的 RRT 算法、遗传算法 (genetic algorithm, GA)、RRT-star 算法、Astar 算法以及引言中提到的 VSDGC-RRT 算法从一般地图到相对复杂的地图中进行对比。实验 1 为在一般地图中的几种算法的对比, 实验 2 为复杂迷宫地图中的几种算法的对比, 实验 3 为本文算法在三维地图中的应用。

#### 3.1 实验 1 一般地图

图 6 是一般地图中分别为 VSDGC-RRT、本文算法、Astar 在一般地图中的一个对比效果。表 1 从路径长度和时间的角度分析了几种算法的差别。根据表 1 的结果, VSDGC-RRT 算法用时最少, 但是如图 6(a) 所示, VSDGC-RRT 得到的路径不平滑, 且路劲代价也大。再看本文算法, 如图 6(b) 所示, 虚线表示原始 RRT 算法得到的路径, 带圆圈的细实线表示嵌入 Dijkstra 算法得到的路径, 无标记的细实线是加入动态步长后得到的路径, 粗实线是经过三步平滑优化得到的最终路径。可以看到, 相较于虚线代表的原始 RRT 路径, 圆圈细实线的路径目标导向型更强, 且嵌入了 Dijkstra 的路径不必要的节点少了很多, 这归功于 Dijkstra 的贪心思想以及启发性思想, 但由于 Dijkstra 的不可避免的缺陷, 导致路径不够平滑, 因此加入了三步平滑优化, 得到了较为完美的粗实线。从图 6(b) 的对比上不

难看出, 在进行三步平滑过后的路径, 突然的转折点更少, 且路径长度明显减少, 在实际中可以很大程度上减少无人机电能的消耗, 更符合无人机路径规划。图 6(c) 是 Astar 算法, 可以看出, Astar 算法有着一定的目标导向型, 但是得到的路径缺少平滑性。结合时间、路径长度和平滑性来看, 本文算法是 3 种算法中的最优算法。

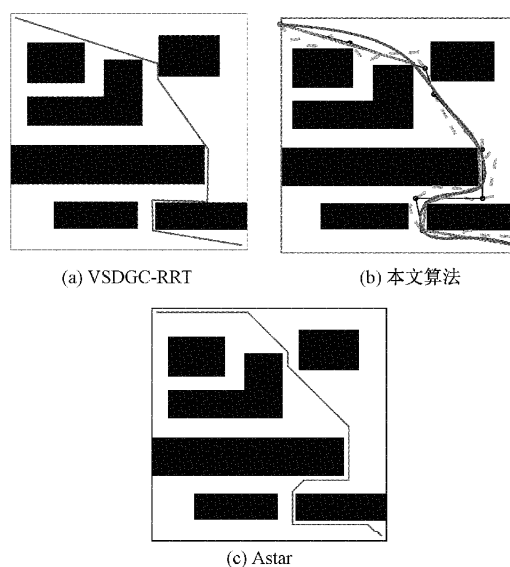


图 6 一般地图中的算法对比

表 1 一般地图中算法对比

算法	路径长度/m	时间/s
RRT	1 139.6	11.4
VSDGC-RRT	1 236.1	0.3
Astar	1 044.7	9.1
本文算法	963.2	6.3

#### 3.2 实验 2 迷宫地图

图 7 分别是遗传算法、本文算法和 RRTstar 算法在迷宫地图中的对比效果图, 表 2 从路径长度和时间的角度分析了几种算法的差别。如图 7(a) 所示, GA 算法的平滑性尚可, 但这是在迭代了 500 次得到的, 导致规划时间大大增加, 这在表 2 中可以得到直观的结果, 且得到的路径从平滑度和长度来说不如本文算法得到的路径。图 7(b) 是本文算法在迷宫地图中的展示, 跟在一般地图中的表现对比, 可以得出结论: 本文算法不论是在一般地图中还是在迷宫地图中, 表现都是最好的算法。RRTstar 算法在迷宫地图中的表现如图 7(c) 所示, RRTstar 由于加入了 rewrite 重新选取父节点, 导致了路径连接都是直线, 也不够平滑。

结合两个实验, 可以得出结论, 本文改进的算法, 在路径长度和平滑性在几种算法中表现优异, 这在无人机的实际应用中减少无人机自身能量消耗以及高效完成任务是至关重要的, 无疑是几种算法中的最优选。两个实验充分说明本算法在任何地图中具有普适性, 为了进一步验证本文



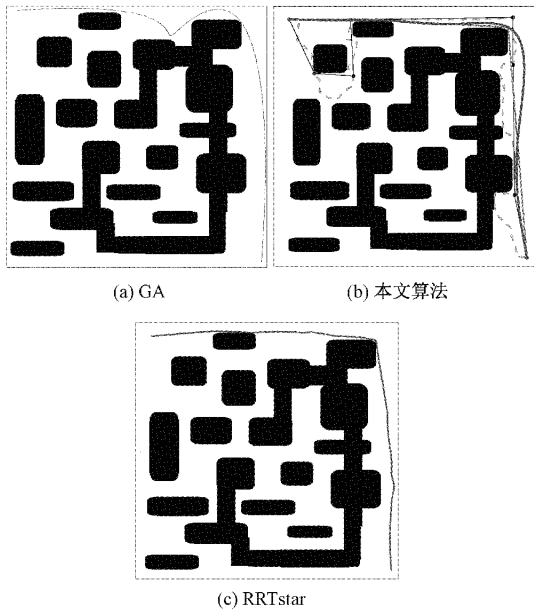


图 7 迷宫地图中的算法对比

表 2 迷宫地图中算法对比

算法	路径长度/m	时间/s
RRT	1 549.2	15.3
GA	1 051.0	43.8
RRTstar	914.7	25.6
本文算法	900.7	9.6

算法在无人机实际应用中的表现,在实验三中进行三维地图的仿真。

### 3.3 实验 3 三维地图

如图 8 所示是本文算法在三维地图中的应用,其中细实线是本文算法生成的最终的路径,可以看出得到的路径平滑且路径长度比起虚线部分的原始 RRT 以及有端点的嵌入 Dijkstra 算法的路径短,且相对比较平滑,充分证明本文算法不仅在二维地图中,同时在无人机实际三维应用中也有较好的效果。

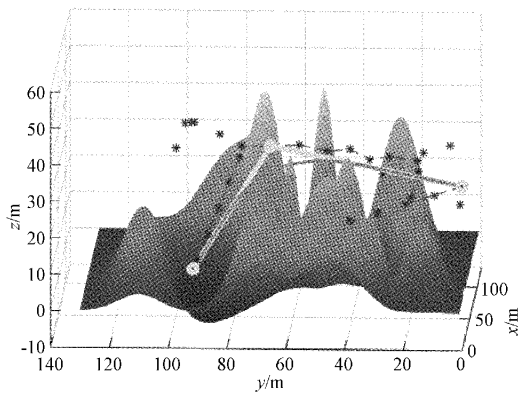


图 8 本文算法在三维地图中的应用

## 4 结 论

针对无人机路径规划中存在的路径不平滑、路径代价大的问题,本文在改进 RRT 算法的同时又提出了一种三步平滑的优化算法。首先,在 RRT 算法生成路径的基础上,嵌入 Dijkstra 算法,可以看出,在启发式算法的思想作用下,冗余的节点得到了有效的减少;针对路径规划时间问题,加入了动态步长策略进行了改进;然后针对路径不平滑的问题,对得到的路径进行路径优化,由于传统的三次 B 样条优化的效果不够好,因此本文提出了三步平滑的策略,三步平滑包括:下样本平滑、上样本平滑和关键点平滑,形成了面和点的优化格局。通过在一般地图和复杂迷宫地图中与其他经典的改进 RRT 算法的仿真可以得出结论,该算法不论是从得到的路径的平滑性还是路径长度上来说,都要明显优于其他同类型的算法,特别是在迷宫地图中,本文算法在路径长度上相较于原始 RRT 算法改进了 42%,生成的路径考虑了无人机的自身限制以及能量的消耗,符合无人机一般作业的路径规划的需要。实验三还表明,本文算法在三维地图中也取得了良好的效果,相信可以在未来无人机的实际应用中有着良好的前景。

### 参考文献

- [1] LAVALLE S M. Rapidly-Exploring random trees: A new tool for path planning [R]. Computer Science Department, Iowa State Unversity, 1998.
- [2] KUFFNER J J, LAVALLE S M. RRT-connect: an efficient approach to single-query path planning [C]. 2000 IEEE International Conference on Robotics and Automation. ICRA2000, April 24-28, 2000.
- [3] 刘紫燕,张杰.改进 RRT 算法的室内移动机器人路径规划[J].计算机数字与工程,2019,47(8):1941-1946.
- [4] 张海波,严小珊,毕齐林,等.融合动态窗与改进 RRT 的全位置机器人路径规划算法研究[J].机床与液压,2022,50(23):35-43.
- [5] 刘春,罗继祥.用于三维空间避障的 HDCS-RRT\* 算法机械臂运动规划研究[J/OL].机械科学与技术:1-8 [2023-07-22].
- [6] 倪建云,李浩,谷海青,等.基于改进 VSRB-RRT 算法的机器人路径规划仿真实验[J/OL].实验技术与管理:1-12[2023-07-02].
- [7] WANG, Z, SHAN L, CHANG, et al. Variable sampling domain and map compression based on greedy RRT algorithm for robot path planning [C]. 2020 39th Chinese Control Conference (CCC), Shenyang, China:3915-3919.
- [8] 李伟东,李乐.基于改进 RRT 算法的无人车路径规划[J].计算机测量与控制,2023,31(1):160-166.

- [9] 孙淑光,孙涛.基于融合 A\* 算法的无人机路径规划研究[J].电子测量技术,2022,45(9):82-91.
- [10] 王康,郭剑东,桑标.基于综合改进 RRT 算法的无人机三维航路规划[J].计算机仿真,2021,38(9):40-44,69.
- [11] 宋晓琳,周南,黄正瑜,等.改进 RRT 在汽车避障局部路径规划中的应用[J].湖南大学学报(自然科学版),2017,44(4):30-37.
- [12] 梁中一,程方晓,魏巍.改进 RRT\* 路径规划算法[J].长春工业大学学报,2020,41(6):602-607.
- [13] 贺伊琳,高奇,赵丹,等.基于改进 RRT 算法的无人驾驶汽车轨迹规划[J].西北大学学报(自然科学版),2018,48(5):651-658.
- [14] 崔挺,李俨,张明庄.基于改进 RRT 算法的无人机航迹规划[J].电子设计工程,2013,21(12):50-53.
- [15] 徐小强,宋子奇,冒燕,等.基于 APF-RRT\* 融合算法的无人机全局路径规划研究[J].武汉理工大学学报,2021,43(9):72-78.

### 作者简介

**施成龙**, 硕士研究生, 主要研究方向为无人机路径规划技术、仪器仪表技术、信号检测与处理。

E-mail:1548880353@qq.com

**行鸿彦**(通信作者), 博士, 教授, 博士生导师, 主要研究方向为气象仪器设计与计量、信号检测与处理等。

E-mail:xinghy@nuist.edu.cn

**王水璋**, 硕士, 高级工程师, 主要研究方向为低空安全与无人机应用系统、信号与信息处理、物联网信息系统等。

E-mail:shuimuw@163.com

**娄华生**, 硕士研究生, 主要研究方向为雷电监测技术、仪器仪表技术、信号检测与处理。

E-mail:2653701223@qq.com