文章编号:1671-4598(2023)12-0203-07 DOI:10.16526/j.cnki.11-4762/tp.2023.12.030 中图分类号:TP242.4 文献标识码:A

基于改进 A* 算法的无人机三维空间避障路径规划

高九州，张 焯
（吉林建筑大学 电气与计算机学院，长春 130118）

摘要：无人机在有障碍物的三维空间环境中飞行，采用常规 A* 算法进行避障航线的规划存在搜索节点多、搜索区域大、搜索时间长、搜索效率低、生成的航线拐角多且含有大量非必要冗余航点、没有考虑无人机自身体积与尺寸而引发的飞行中与障碍物边界碰撞的航线不安全等问题；因此，设计一种改进 A* 算法，首先，考虑无人机本身体积与尺寸，提出一种消除边界碰撞事故的子节点扩展方法；其次，改进评价函数，减少往复搜索次数，缩小搜索区域面积，提高搜索效率；然后，根据 Floyd 思想，对生成的航线进行简化处理，消除航线中的冗余航路点，减少航线转角数量，达到简化航线并改善航线平滑度的效果；最后，非线性仿真及飞行试验表明了改进的 A* 算法生成的航线更加安全、高效，并使无人机的飞行连续和顺畅。

关键词：无人机；改进 A* 算法；航线安全；评价函数；简化航线

Obstacle Avoidance Path Planning of UAV in 3D Space Based on Improved A* Algorithm

GAO Jiuzhou, ZHANG Zhuo
(School of Electrical and Computer Science, Jilin Jianzhu University, Changchun 130118, China)

Abstract: When unmanned aerial vehicles (UAVs) fly in the space with obstacles, traditional A* algorithms have many problems in obstacle avoidance route planning, such as many search nodes, large search area, long search time, low search efficiency, uneven path, redundant path points, many turning points, and unsafe path caused by collision without considering the volume and size of UAV itself. Therefore, an improved A* algorithm is designed. Firstly, considering the volume and size of UAV, a sub node extension method to eliminate the boundary collision accidents is proposed. Secondly, an evaluation function is optimized to reduce the number of sub interval search, reduce the search area and improve the search efficiency. Then, Floyd algorithm is introduced to simplify the generated path and eliminate many redundant points and turning points, the path is smoothed by cubic spline interpolation method, which eliminates the inflection point of the path and improves the smoothness of the path. Finally, the simulation results show that the path generated by the improved A* algorithm is more safe and efficient, which makes the flight action of UAV more continuous and smooth.

Keywords: UAV; improved A* algorithm; path safety; evaluation function; simplification path

0 引言

无人机避障航线规划是指无人机飞行在有障碍物的三维空间环境里，以航线、航时、能耗等某一指标为基准，通过对空间环境数据采集、分析、计算与比较，得出一条指标性能最佳的安全无碰撞航线[1]。避障航线的规划按照对空间环境建模的实时性又可以分为静态航线规划和动态航线规划。静态航线规划[2]是指无人机的空间环境是固定不变的，通过对飞行环境进行建模，再以某一指标为基准，分析、计算与比较得到一条从起点到目标点的最优航线，即飞行前飞控系统已装载好航路点和航线[3-5]。动态航线规划[2]是指飞行过程中的空间环境信息是事先未知的，且环境信息可能是动态变化的情况下，通过无人机自身实时感知周边环境信息，建立周边环境的实时动态模型，再通过分析、计算与比较得到一条绕过当前障碍物的最佳航线，进而引导飞行至目标点，即一边飞行一边设计当前避障航线[6-7]。

A* 算法是在迪克斯特拉（Dijkstra）算法与广度优先算法（BFS，breadth first search）的基础上，兼顾搜索效率与最优路径的一种启发式搜索算法[8-9]。研究人员也从多个角度对 A* 算法进行了改善与优化，李世国等人[10]采用从起点和终点分别同步采用 A* 进行搜索，直至搜索路径会合，该方法减少了搜索时间，提高了搜索效率，但若会合处存在障碍物，有可能导致会合失败。槐创锋等人[11]将常规 A* 算法 3×3 的搜索邻域扩展为 7×7，搜索路径得到

收稿日期:2023-02-16； **修回日期**:2023-03-15。
基金项目:吉林省自然科学基金(YDZJ202201ZYTS561)。
作者简介:高九州(1987-),男,博士,讲师。
引用格式:高九州,张 焯. 基于改进 A* 算法的无人机三维空间避障路径规划[J]. 计算机测量与控制,2023,31(12):203-209,223.

简化，且提高了路径的平滑性，但增加了搜索时间，降低了搜索效率。田华亭、张红梅、赵晓等人[12-14]从搜索方向选择、评价函数构建、速度控制等几个方面对 A * 算法进行优化与改进，提高了搜索速度，缩短了路径长度，提高了路径的平滑度。张敬寒等人[15]提出一种扩大搜索邻域的改进 A * 算法，提高了算法寻路效率，张帅等人[16]提出圆形节点扩展，实现了变方向和变步长扩展子节点，Mario K 等人[17-18]都通过改进优化函数等到了一种最短路径，提高了搜索效率。但上述研究人员都没有考虑实际运动过程中，无人机/无人车/机器人由于本身体积与尺寸的原因可能发生与障碍物发生碰撞的问题。杨奇峰[19]等人设计了动态障碍物运动趋势预测算法，提升了动态避障算法的效率与安全性。胡中华等人[20]采用 B 样条曲线插补法优化避障航迹，使整体航迹更加平滑，更有利于无人机/无人车/机器人运动速度的平稳与协调。

本文提出一种改进 A * 算法，首先，该算法的设计充分考虑了无人机本身的体积与尺寸问题，在常规 A * 算法的基础上进行子节点扩展方法的改进，保证了飞行过程中始终与障碍物保持一定的安全距离，防止无人机与障碍物发生碰撞。其次，对常规 A * 算法的评价函数进行了改进，增加将当前节点的父节点到目标点的距离作为评价指标，达到了减少无用的往复搜索次数的目的，使搜索范围和搜索子节点数量减少。最后，引入 Floyd 思想，对航线进行简化处理，消除大量航线冗余点，避障总航线变短，航线转折角变平滑，使无人机更平稳地跟踪航线以飞抵目标点。

1 常规 A * 算法

1.1 建模

航线规划是基于空间环境模型的，因此，需要对含有障碍物的空间环境进行建模，再使用 A * 算法进行避障航线的规划。栅格就是将空间分割成有规律的网格，每个网格即一个单元，并在各单元上赋予相应的属性值（障碍或自由空间）。单元的位置由它的行列号定义，所表示的实体位置隐含在栅格行列位置中。栅格的单位长度由实际情况决定，比如无人机的体积与尺寸。

在实际的路径搜索问题，将搜寻的三维空间环境分割为有规律的栅格，每个栅格称为一个单元，每个栅格的位置由其行和列决定，这样就简化搜索区域为 3 维数组，数组的每一项代表一个栅格，它的状态就是可走（walkalbe）和不可走（unwalkable），可走即该栅格没有障碍物，算法可拓展到该栅格，不可走即该栅格有障碍物，算法不能拓展到该栅格。

栅格地图形象简单且能有效地表达三维环境中的障碍物信息，所以本文在进行航线规划时采用栅格法进行三维地图建模。

1.2 A * 算法

A * 算法是以计算节点的评价函数为核心，在扩展航线节点的每一步过程中，始终选取评价函数的函数值最小的节点作为子节点，进而再以该子节点作为父节点进行下一步航线子节点的扩展与寻优，即依次寻找评价函数值最小的子节点，形成寻优航线。首先，将起点作为第 1 个节点放入 CLOSE 列表中，计算周围最多 26 个节点的评价函数（由于飞行边界和障碍的原因，周围可扩展节点数可能小于 26），并将这些节点及节点信息（评价函数值）放入 OPEN 列表中，若 OPEN 列表中已存在当前节点，则比较该节点的评价函数值，将评价函数值较小的节点的节点信息更新至 OPEN 列表中。在 OPEN 列表中，选取评价函数值最小的节点作为当前节点的子节点。将选中的子节点作为父节点，同时将其节点信息放入 CLOSE 列表中，使用新的父节点再进行子节点扩展并计算各子节点的评价函数值，通过比较评价函数值得到新的子节点，并不断更新 OPEN 和 CLOSE 列表，直至搜索到目标节点。

评价函数 $f(n)$ 的表达式为：

$$f(n) = g(n) + h(n) \tag{1}$$

式中，$g(n)$ 为起始节点到当前节点的实际航线代价；$h(n)$ 为当前节点到目标点的估计航线代价。

计算 $g(n)$、$h(n)$ 有 3 种计算方法，分别为曼哈顿距离、欧几里得距离和切比雪夫距离。曼哈顿距离是两个点在标准坐标系上的绝对轴距总和。欧几里得距离是指两点之间的距离，即两点之间直线最短的那段距离。切比雪夫距离是向量空间中的一种度量，是空间两点各坐标数值差的最大值。曼哈顿距离、欧几里得距离和切比雪夫距离的具体计算如下

$$\begin{aligned} g(n) &= |x_i - x_{start}| + |y_i - y_{start}| + |z_i - z_{start}| \\ g(n) &= \sqrt{(x_i - x_{start})^2 + (y_i - y_{start})^2 + (z_i - z_{start})^2} \\ g(n) &= max(|x_i - x_{start}|, |y_i - y_{start}|, |z_i - z_{start}|) \end{aligned} \tag{2}$$

本文采用欧几里得距离作为航线代价的计算方法，其表达式为：

$$\begin{cases} g(n) = \sqrt{(x_i - x_s)^2 + (y_i - y_s)^2 + (z_i - z_s)^2} \\ h(n) = \sqrt{(x_t - x_i)^2 + (y_t - y_i)^2 + (z_t - z_i)^2} \end{cases} \tag{3}$$

式中，(x_i, y_i, z_i) 为当前点；(x_s, y_s, z_s) 为起始点；(x_t, y_t, z_t) 为目标点。

A * 算法在搜索过程中需要不断更新 OPEN 和 CLOSE 列表中的节点及其相关信息，计算每个节点的评价函数值并通过比较进而选取评价函数值最小的节点作为下一步搜索的父节点，逐步进行，依次搜索至目标点，形成搜索航线。因为每一步的搜索代价都是最小的，因此整体搜索航线也是最优的。但是，在搜索过程中常规 A * 算法也存在一些不足之处：

1）三维环境信息采用周边 26 节点搜索方法，在航线搜索过程中搜索节点数量过多，每个节点都需要计算其评价函数值，导致计算量过大，计算时间过长，搜索效率较低。

2）规划得到的航线冗余点过多，航线转折角过多，导致飞行航线并非最短且航路平滑度较差，不利于无人机的

平稳飞行。

3）在规划航线时，只考虑障碍物节点信息，没有考虑无人机本身具有一定的体积与尺寸，当无人机航线以栅格对角线的形式斜穿障碍物顶点时，其机身或桨叶边缘可能碰撞到障碍物导致无人机飞行不稳，甚至存在翻坠的风险。

2　优化 A＊算法

2.1　航线安全规划

常规 A＊算法在进行航线规划时将无人机看作质点，但实际情况下无人机的体积和尺寸不能忽略。实际飞行过程中无人机在沿着对角线航路方向飞行的过程中可能会发生与障碍物碰撞的事故。以二维 XY 空间为例进行说明，如图 1 所示，起点坐标 A，终点坐标 G，障碍物位于图中黑色区域，边界节点坐标包含（8，3）和（8，7）。栅格的单位距离至少要大于无人机的半轴距才可能保证沿着障碍物边界飞行过程中无碰撞，按照常规 A＊算法得到的航线，在航线点 B 到航线点 C 的航线中，障碍点（8，3）距离航线的最小距离小于 1 个栅格单位，会引发无人机与障碍物发生碰撞。同样，航线点 D 到航线点 E 的航线中，障碍点（8，7）距离航线的最小距离小于 1 个栅格单位，也会引发无人机与障碍物发生碰撞。

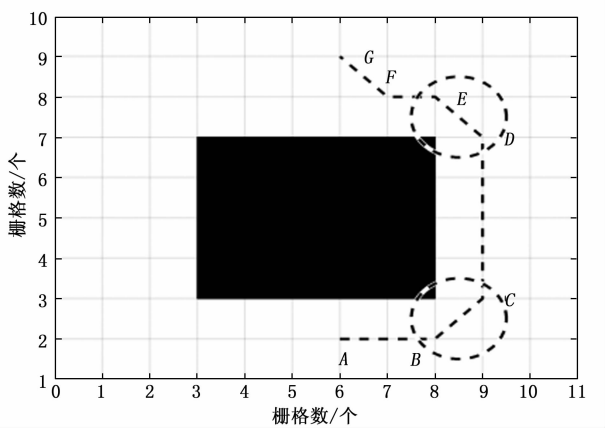


图 1　未设置安全距离的 A＊航线

因此本文对常规 A＊算法进行了改进设计，在扩展子节点时，如果当前节点的 X 正、X 负、Y 正、Y 负、Z 正、Z 负的某些方向上存在障碍物，则不采用其对角航线上的子节点，即不进行该方向的对角线子节点扩展，不生成距离障碍物小于 1 个栅格单位的对角线航线。

如图 2 所示，假设当前航点 $A\rightarrow B$，航点 B 的上侧存在障碍物节点，按照常规 A＊算法，扩展的最佳子节点为 $B\rightarrow C\rightarrow D$，但该航线距离障碍物节点（4，2）小于 1 个栅格的距离，故在此步子节点扩展中不允许生成子节点 C，最佳子节点依次应为 $B\rightarrow E\rightarrow C\rightarrow D$，此处节点 C 由其父节点 D 扩展生成并选为最优节点。即当前节点的 X 正、X 负、Y 正、Y 负、Z 正、Z 负方向上存在障碍物节点时，依次不允许生成的子节点如下详细说明。

1）X 正方向存在障碍物，不允许生成与 X 正相关的子节点，如 X 正 Y 正 Z 正、X 正 Y 正 Z 零、X 正 Y 正 Z 负、X 正 Y 零 Z 正、X 正 Y 零 Z 负、X 正 Y 负 Z 正、X 正 Y 负 Z 零和 X 正 Y 负 Z 负等 8 个子节点；

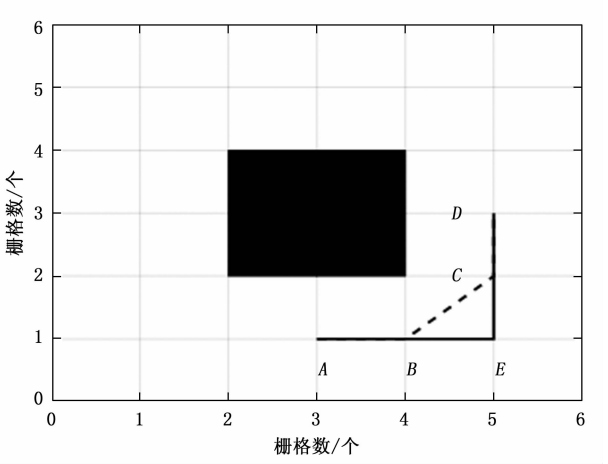


图 2　改进扩展子节点方式

2）X 负方向存在障碍物，不允许生成与 X 负相关的子节点，如 X 负 Y 正 Z 正、X 负 Y 正 Z 零、X 负 Y 正 Z 负、X 负 Y 零 Z 正、X 负 Y 零 Z 负、X 负 Y 负 Z 正、X 负 Y 负 Z 零和 X 负 Y 负 Z 负等 8 个子节点；

3）Y 正方向存在障碍物，不允许生成与 Y 正相关的 8 个子节点，如 X 正 Y 正 Z 正、X 零 Y 正 Z 正、X 负 Y 正 Z 正、X 正 Y 正 Z 零、X 负 Y 正 Z 零、X 正 Y 正 Z 负、X 零 Y 正 Z 负和 X 负 Y 正 Z 负等 8 个子节点；

4）Y 负方向存在障碍物，不允许生成与 Y 负相关的 8 个子节点，如 X 正 Y 负 Z 正、X 零 Y 负 Z 正、X 负 Y 负 Z 正、X 正 Y 负 Z 零、X 负 Y 负 Z 零、X 正 Y 负 Z 负、X 零 Y 负 Z 负和 X 负 Y 负 Z 负等 8 个子节点；

5）Z 正方向存在障碍物，不允许生成与 Z 正相关的 8 个子节点，如 X 正 Y 正 Z 正、X 零 Y 正 Z 正、X 负 Y 正 Z 正、X 正 Y 零 Z 正、X 负 Y 零 Z 正、X 正 Y 负 Z 正、X 零 Y 负 Z 正和 X 负 Y 负 Z 正等 8 个子节点；

6）Z 负方向存在障碍物，不允许生成与 Z 负相关的 8 个子节点，如 X 正 Y 正 Z 负、X 零 Y 正 Z 负、X 负 Y 正 Z 负、X 正 Y 零 Z 负、X 负 Y 零 Z 负、X 正 Y 负 Z 负、X 零 Y 负 Z 负和 X 负 Y 负 Z 负等 8 个子节点。

每个方向有 2 种约束可能，6 个方向共计 64 种可能。这样避免了航线规划中出现对角线方向上的航线与障碍物发生碰撞的可能，从而保证无人机与障碍物之间的安全距离。

安全设计后的航线如图 3 所示。此设计略微增加了避障航线的总长度，但对航线的安全性有了保证，不会发生飞行器与障碍物边界发生碰撞的可能。

2.2　评价函数优化

常规 A＊算法会在开列表 OPEN 中不断比较和选取评价函数值最小的节点，这会导致来回往返的搜索，为解决

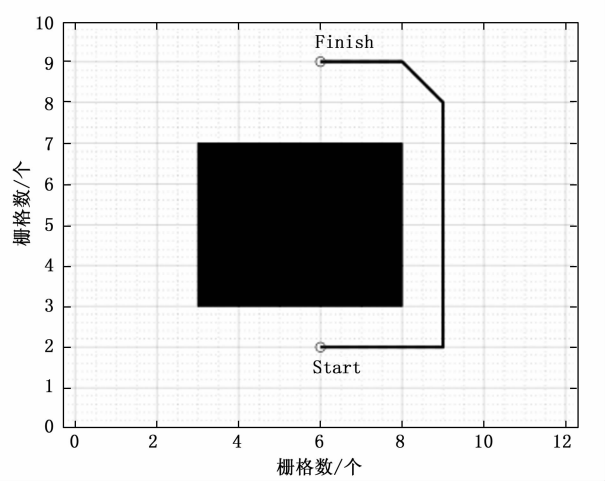


图 3 改进扩展子节点生成的航线

低效率往返搜索的问题，本文首先对启发函数的 $h(n)$ 加设了权重，即

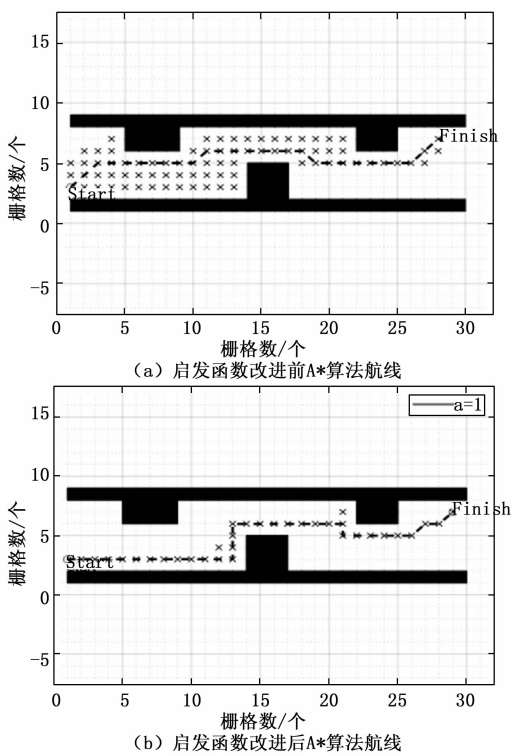
$$f(n) = g(n) + \alpha h(n) \tag{4}$$

改进后的算法在航线搜索快速性方面有所增强，但仍存在无用的往返搜索。在此基础上，加入当前子节点的父节点到目标点的距离作为对评价指标，即

$$f(n) = g(n) + \alpha[h(n) + h(p)] \tag{5}$$

其中：$h(p)$ 为当前节点的父节点到目标点的距离；α 为权重值。

改进评价函数前后的 A * 算法二维空间实例如图 4 所示，相比常规 A * 算法，其航线长度、搜索节点数比较见表 1。



(a) 启发函数改进前A*算法航线

(b) 启发函数改进后A*算法航线

图 4 二维空间实例图

表 1 不同方法节点数与航线长度对比

Method	Weight	OPEN number	CLOSE number	Path Length
A *	——	102	80	29
UA *	0.9	78	36	33
UA *	1	77	35	33
UA *	1.1	76	35	33
UA *	1.2	75	34	33
UA *	1.3	75	34	33

从表 1 可以看出，启发函数改进后，搜索的节点少了近 25%，闭环节点少了近 55%，即明显减少了往返无用搜索的次数，算法的搜索效率得到了提高，同时权值可变化范围并不大，超过一定范围后，权值的变化对航线搜索结果影响不大。

2.3 航线简化处理

当使用常规 A * 算法生成一条从起点到目标点的避障航线时，由于只能沿着栅格点依次扩展形成避障航线，这样的航线会存在冗余航路点和过多的转角点。本文采用弗洛伊德（Floyd）思想对常规 A * 算法生成的航线进行简化处理，将不必要的冗余点和转角点做删除处理，缩短整个航线的长度并使航线变得相对平滑，更有利于无人机飞行的平稳与顺畅。

Floyd 算法的本质是运用动态规划的思想寻找给定的加权图中多源点之间最短路线的方法。本文结合 A * 算法与 Floyd 算法对生成的避障航线进行简化处理，通过使用 Floyd 算法从起始点到目标点进行了多次迭代优化，大大缩短了规划出的避障航线长度。

Floyd 算法原理如图 5 所示，常规 A * 算法规划出的航线为 $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F \rightarrow G \rightarrow H$，即每一步只能沿着栅格的横向、纵向或对角线方向移动，但显然节点 C 和 D 是冗余的，即 $B \rightarrow C \rightarrow D \rightarrow E$ 的航线并不合理，既存在转折，航线又长。可将其简化为 $A \rightarrow E \rightarrow F \rightarrow G \rightarrow H$ 或 $A \rightarrow F \rightarrow G \rightarrow H$，航线 $A \rightarrow G \rightarrow H$ 与障碍物相交，故不合理。比较航线 $A \rightarrow E \rightarrow F \rightarrow G \rightarrow H$ 或 $A \rightarrow F \rightarrow G \rightarrow H$，在 Floyd 算法下，航线 $A \rightarrow F \rightarrow G \rightarrow H$ 比航线 $A \rightarrow E \rightarrow F \rightarrow G \rightarrow H$ 更加简短，更合理，但是该航线距离障碍物节点最短距离小于 1 个栅格单位，故根据航线安全规则，舍弃航线 $A \rightarrow F \rightarrow G \rightarrow H$，选取航线 $A \rightarrow E \rightarrow F \rightarrow G \rightarrow H$。

通过 Floyd 算法优化，删除了一些多余的节点，航线总长度变短，航线转角变少且相对平缓，航线得到改善，无人机更加平稳省时飞行。

如图 6 所示，改进后 $A*$ 算法流程说明如下：

1）创建 OPEN 和 CLOSE 列表，把起点加入 OPEN 列表中。

2）重复如下过程：

（1）遍历 OPEN 列表，查找代价函数值最小的节点，把它作为当前要处理的节点。

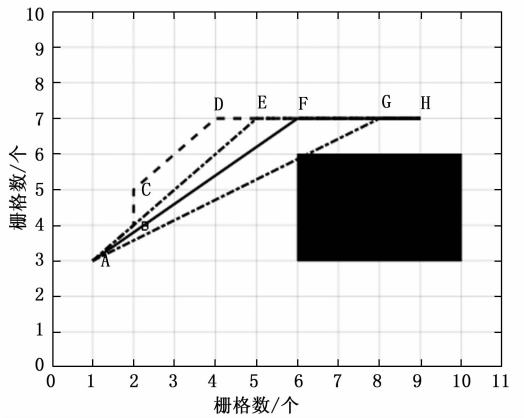


图 5　Floyd 最短航线原理

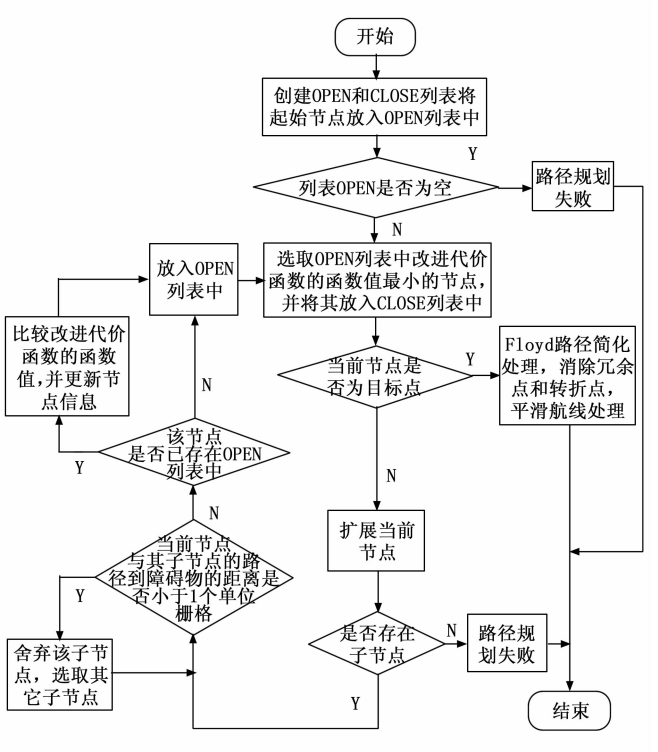


图 6　改进后 $A*$ 算法流程图

(2) 把这个节点移到 CLOSE 列表中，同时判断当前列节点是否为目标点，如果是，跳出循环，进行步骤 3)，如果不是继续执行。

(3) 对当前方格的最多 26 个相邻方格进行子节点扩展，如果没有可扩展的子节点则跳出循环，路径规划失败。如果存在可扩展的子节点，判断扩展方向是否存在障碍，如果可扩展的方向全是障碍，则跳出循环，路径规划失败。如果存在某方向无障碍，则正常进行该方向子节点扩展。

(4) 判断新扩展的子节点是否已在 OPEN 列表中。如果不在，将该节点添加进 OPEN 列表中，如果已在，比较节点的代价函数值，将代价函数值较小的节点信息更新至 OPEN 列表中。具体方法如下，用 G 值作参考。更小的 G 值表示更好的路径。将更小 G 值的父节点设置为当前方格，并重新计算它的 G 和 F 值。按 F 值重新排序。

3) 进行 Floyd 路径简化处理，消除冗余点和转折点，平滑航线。

3　仿真与测试

为验证所设计的改进 $A*$ 算法的合理性与有效性，本章分别先后进行数字仿真分析和实际飞行测试，即在常规 $A*$ 算法航线规划的基础上，在对其进行安全航线规划、评价函数优化、航线简化和航线平滑处理后进行仿真分析与飞行测试。

3.1　仿真分析

本文采用 Matlab-M 函数搭建数字仿真算法模型，主程序 Main 函数主要进行飞行区域设定，飞行起点和终点坐标设定，障碍物范围设定，A＊算法流程程序执行。调用函数包括 distance 欧几里得距离函数，expand _ array 子节点扩展函数，insert _ open 子节点函数，min _ fn 目标点判断函数，node _ index 规划航线节点函数，upgradedistance 优化评价函数等。设定飞行起点坐标 (1，1，1)，终点坐标 (40，12，15)，航线上设置 3 组障碍物，如图 7～12 所示，第 1 组障碍 XYZ 范围分别为 5～8、1～12 和 1～10，第 2 组障碍 XYZ 范围分别为 20～25、1～15 和 1～25，第 3 组障碍 XYZ 范围分别是 30～38、5～20 和 1～20。图 7 是常规 A＊算法下的包含闭列表数据的航线图，图 8 是优化 A＊启发函数算法下的包含闭列表数据的航线图，黑点代表经过比较分析后放入闭列表中的数据步点，绿点划线是在安全

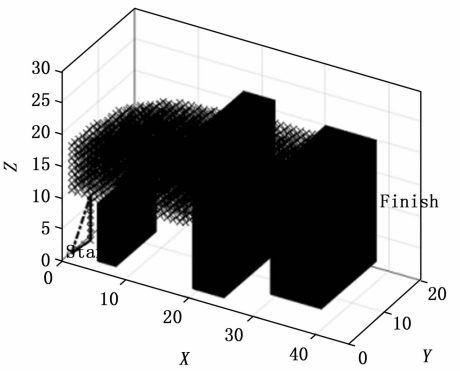


图 7　含闭列表数据的安全简化 A＊算法航线

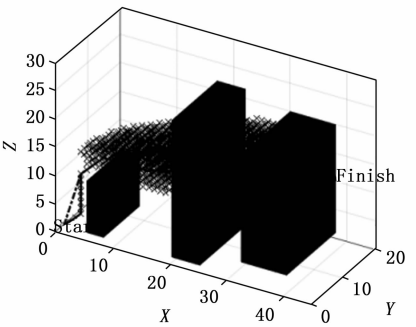


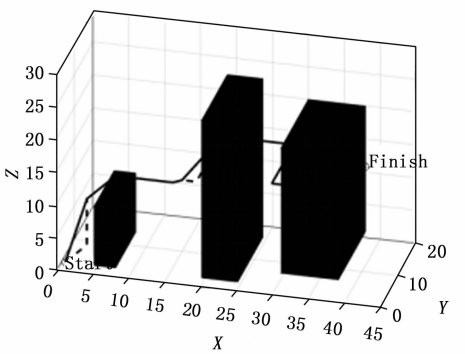
图 8　安全简化优化启发函数 A＊算法航线

航线规则下的 A * 算法航线，红虚线是在安全航线规则和 Floyd 简化航线的设计下 A * 算法航线。表 2 为与图 7～8 对应的是否有启发函数优化的 A * 方法生成的航线所包含的开列表节点数、闭列表节点数和总航线长度对比，由表可知，而知总航线长度一致，但启发函数优化后的开列表节点数只有常规方法下的 52%，闭列表节点数只有常规方法下的 40.8%，可知优化后航线搜索数量明显减少，搜索时间变短，搜索效率提高。

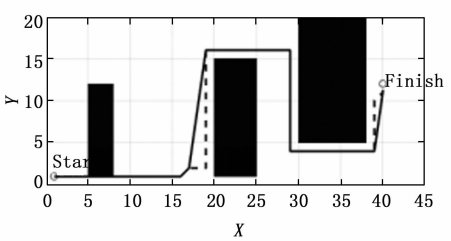
表 2 三维仿真节点数与航线长度对比

Method	Weight	OPEN number of nodes	CLOSE number of nodes	Path Length
A *	——	5 132	3 811	42.871 9
UA *	1	2 680	1 555	42.871 9

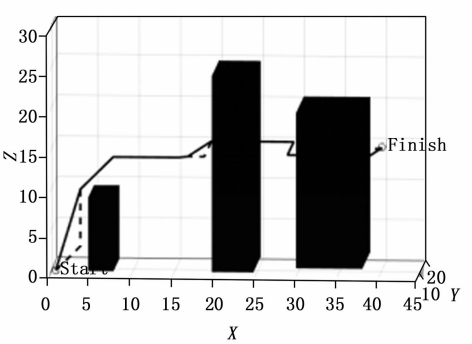
图 9 是不包含闭列表数据的多视角航线图，虚线是在安全航线规则下的 A * 算法航线，实线是在安全航线规则和 Floyd 简化航线的设计下 A * 算法航线。



(a)安全简化航线



(b)俯视角下安全简化航线



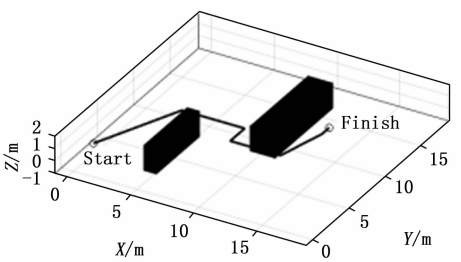
(c) 正视角下安全简化航线

图 9 多视角下安全简化航线

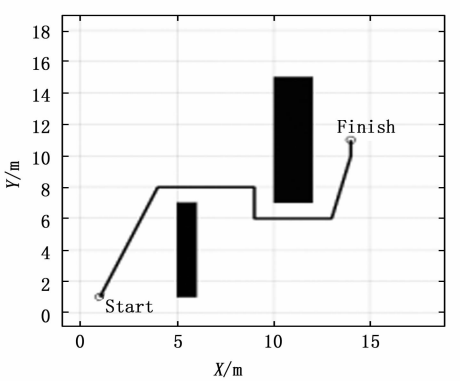
3.2 飞行测试

在 Pixhawk2 开源飞控的基础上，本文所设计的避障算法经过程序的编译，烧录至飞控中，采用本团队自主研发的小型四旋翼飞行器在开阔平坦的操场上设置障碍并进行飞行测试。Pixhawk 是一个独立的高可用性，为学术界和工业界提供高端行业标准自动驾驶仪硬件，同时支持 APM 和 PX4 两种固件。Pixhawk2 是 Pixhawk 的 2 代升级产品，其特点包含 IMU、FMU、底板分离，利于自定义硬件开发；三套 IMU，两套气压计，IMU 恒温，IMU 减震；爱迪生板卡，英伟达 GPU 接口预留，为视觉避障留有接口；外设接口丰富，牢固，板载电源冗余度设计。本团队自主研发的小型四旋翼飞行器采用大疆 E800KV350 动力套装，轴距 600 mm，采用格式 12 000 mAH 电池进行系统供电，起飞重量约 3.5 kg，此配置下实测航时约 25 分钟，最大实飞高度 300 m，可用于搜救与查勘。

实测地为学校操场，海拔 200 m，横侧风风速小于 3 级，设置 2 组障碍，第 1 组障碍 *XYZ* 的范围依次是 4～5、0～7 和 0～1，第 2 组障碍 *XYZ* 的范围依次是 9～11、6～14 和 0～2，生成的避障航迹如图 10 所示。



(a)测试飞行设计航线



(b)测试飞行设计航线（俯视）

图 10 避障航迹图

测试平台及环境如图 11 所示，本地坐标系下的飞行航线如图 12 所示，飞行起点坐标（1，1，1），终点坐标（14，11，1），飞行航线总长度 24.856 6 m，飞行航线与障碍物最小距离大于设定的栅栏单位距离 1 m，总体上完成了航线安全平滑的目的。

全局坐标系下的航线如图 13 所示，飞行起点坐标（49.790 33，125.404 49），终点坐标（49.790 45，125.404 512），

图 11　测试平台及环境

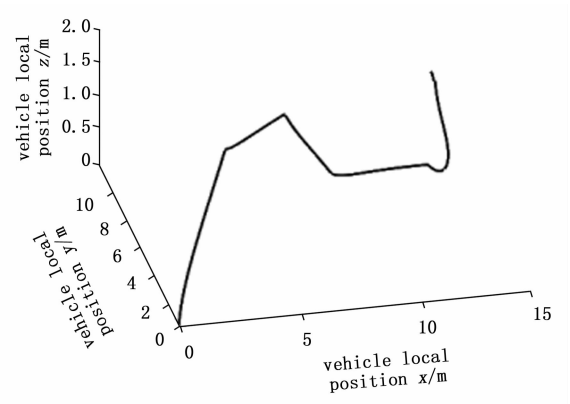


图 12　本地坐标系下的飞行航线

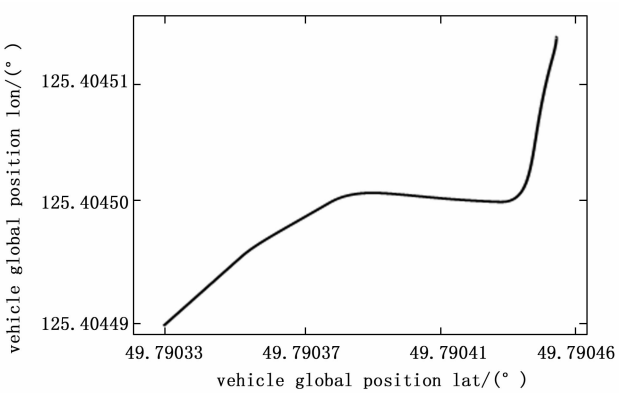


图 13　全局坐标系下的飞行航线

两点直线距离约 13.5 m，飞行过程中，地速设为 1 m/s，航线总体上经过简化和平滑处理，整个飞行过程平稳而快速。其地速及姿态指令如图 14 所示，最大姿态指令不超过 5°，转弯时，机身抖动幅值小，且恢复迅速，飞行平稳，整个飞行过程中，由于避障转弯会加减速，但基本维持在 1 m/s，且飞行无停顿。通过飞行测试，进一步证明本文所设计的基于改进 A * 算法的无人机避障航线规划的合理性和实用性。

4　结束语

本文基于常规 A * 算法，针对无人机避障航线规划的问题，提出一种改进型 A * 算法，该算法具有以下特点：

1）充分考虑无人机本身体积与尺寸，提出无碰撞扩展子节点的方法，保证生成的航线与障碍物距离大于设定的安全界限。

2）对评价函数进行优化，优化后的 A * 算法的往返搜索次数和搜索子节点数量明显减少，搜索区域减小，搜索时间变短，搜索效率提高。

3）基于 Floyd 最短路径方法，对 A * 算法生成的避障航线进行优化，在保证航线安全界限的前提下，删除冗余和转折航点，航线转角相对平缓，航线总长度减小。对生成的最短航线进行平滑性处理，使实际飞行航线平滑，飞行状态平稳。

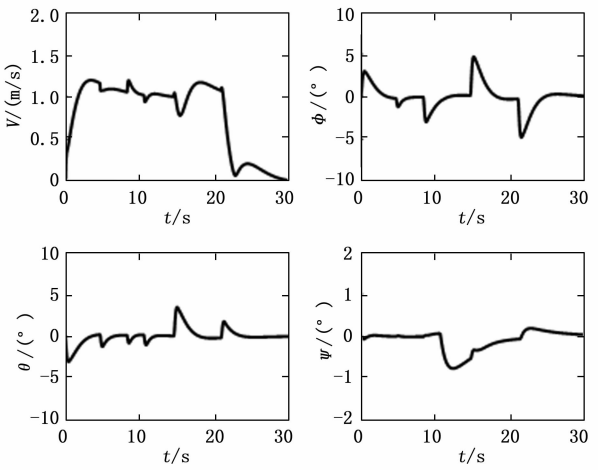


图 14　地速与飞行姿态

最后，通过仿真与实飞测试分别验证了本文设计的基于改进 A * 算法的无人机避障航线的合理性和有效性，为进一步开展无人机自主避障飞行的应用推广打下了坚实的基础。

参考文献：

[1] 董　箭，初宏晟，卢柘樟，等．基于 A 星算法的无人机路径规划优化模型研究［J］．海洋测绘，2021，41（3）：28－31.

[2] 霍凤财，迟　金，黄梓健，等．移动机器人路径规划算法综述［J］．吉林大学学报（信息科学版），2018，36（6）：639－647.

[3] 王志明．无人机路径规划算法研究［D］．长春：长春工业大学，2019.

[4] 张志文，张　鹏，毛虎平，等．改进 A * 算法的机器人路径规划研究［J］．电光与控制，2021，28（4）：21－25.

[5] ANDERSE M，GONZALEZ N，SOLERM L. Informed scenario based RRT for aircraft trajectory planning under ensemble forecasting of thunder storms［J］. Transportation Research Part C：Emerging Technologies，2021，129：103232.

[6] 吴　健．基于 A-star 改进路径规划算法研究［D］．马鞍山：安徽工业大学，2019.

（下转第 223 页）