

# 狭窄空间无人机动态三维 A\* 算法研究

李兆强, 张 拓

(西安建筑科技大学 信息与控制工程学院, 西安 710055)

**摘要:** 针对三维飞行器在动态环境下使用三维 A\* 算法进行局部仿真时, 环境信息未知, 存在冗余点和拐点, 导致收敛时间长、路径节点扩展代价大、易陷入局部最优问题, 提出一种基于全局与局部相结合的动态三维 A\* 寻路算法; 此算法首先改进评价函数的权值系数动态分配, 减小路径冗余点和拐点, 全局规划算法耗时降低 2%, 路径长度缩短 7%, 局部规划算法耗时降低 20%, 路径长度缩短 12.4%; 其次改进路径生成策略, 有效提高算法效率, 避免陷入局部最优, 从而实现飞行器在三维动态环境中的路径规划; 将改进后的算法进行仿真对比, 仿真结果表明, 改进后的算法路径更加合理, 算法耗时和路径长度更短。

**关键词:** 三维 A\* 算法; 路径规划; 权值系数分配; 路径生成策略

## Research on Improved Global and Local Dynamic 3D A\* Algorithm

Li Zhaoqiang, Zhang Tuo

(School of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an 710055, China)

**Abstract:** Aiming at the local simulation of a 3D aircraft using a 3D A\* algorithm in a dynamic environment, the environmental information is unknown, there are redundant points and inflection points, which leads to long convergence time, large path node expansion cost, and easy to fall into local optimal problems. Global and local dynamic 3D A\* path finding algorithm is presented. This algorithm first improves the dynamic allocation of weight coefficients of the evaluation function, reduces path redundancy and inflection points. The global planning algorithm reduces time by 2%, the path length is reduced by 7%, the local planning algorithm reduces time by 20%, and the path length is reduced by 12.4%; Secondly, improve the path generation strategy, effectively improve the efficiency of the algorithm, and avoid falling into the local optimum, so as to achieve the path planning of the aircraft in the three-dimensional dynamic environment. The improved algorithm is compared with simulation. The simulation results show that the improved algorithm has a more reasonable path, and the algorithm takes less time and has a shorter path length.

**Keywords:** 3D A\* algorithm; path planning; weight coefficient allocation; path generation strategy

## 0 引言

舰船舱室发生火灾时, 被困人员需要及时逃离危险, 在三维空间中, 智能飞行器的引导成为关键。在设计智能飞行器时, 快速寻找最优路径和及时躲避动态和静态障碍物的能力起着重要作用。而启发式搜索算法 A\* 及其改进算法以其快速高效的特点被广泛用于路径规划问题<sup>[1-7]</sup>。

在传统 A\* 算法基础上, 陈豪等引入了方向矢量和并行搜索, 使机器人路径搜索同时具备方向性和并行性<sup>[8]</sup>, 未能拓展到三维空间; 王殿君提出采用变步长搜索方法, 同时采用实时调整权值的路径评价函数, 提升了优化效率和优化结果<sup>[9]</sup>, 但是未考虑动态障碍物的影响; 何雨枫提出在代价函数计算、开启集更新方式等方面对传统 A\* 算法进行了改进<sup>[10]</sup>, 但存在收敛速度较慢导致飞行器救援时间过长, 易陷入局部最优的问题。

本文提出一种基于改进的三维局部 A\* 路径规划算法。针对飞行器在局部规划路径时环境信息未知, 存在冗余点

和拐点, 导致收敛时间长、路径节点扩展代价大、易陷入局部最优问题, 在代价函数权值分配, 局部与全局路径结合等方面对 A\* 算法进行改进, 并与文献 [10] 进行了 Matlab 仿真与数据对比。

## 1 问题描述及环境设定

### 1.1 问题描述

在已知模型的舰船舱室内发生火灾时, 由于烟雾阻扰, 通道内存在的静态障碍物和火灾带来的动态障碍物 (大火, 掉落的残渣物等), 被困人员无法快速安全地逃离危险区域, 这使得智能飞行器引导被困人员逃离至安全地带成为关键, 这一条件使得无人机需要局部路径规划<sup>[11]</sup>。飞行器先获取静态障碍物信息, 在传感器感知范围内未检测到动态障碍物, 可利用三维 A\* 算法在已知的障碍之间寻找一条先验路径, 由于飞行器对环境动态障碍物没有先验知识, 如果飞行器在有限的感知范围内, 实时获取到有关动态障碍物的信息, 则利用本文提出使用改进三维 A\* 局部搜索算法。

### 1.2 环境设定

无人机路径规划的初始阶段是围绕无人机建立基于图形的结构或数字模型, 并在没有碰撞障碍物的情况下移动至目标。无人机的三维工作空间的长度, 宽度和高度分别

收稿日期: 2020-02-03; 修回日期: 2020-03-10。

基金项目: 陕西省教育厅基金(14JK1404); 西安建筑科技大学青年科技基金(RC1244, QN1233)。

作者简介: 李兆强(1981-), 男, 陕西泾阳人, 博士, 副教授, 主要从事先进控制理论与应用方向的研究。

是  $L, W, H$ 。工作区分为  $R \times S \times T$  个相同大小的网格。每个网格的信息由式 (1) 表示。整个工作空间的信息可以表示为:

$$\varphi = \sum N_{ijk} (i \in [1, R], j \in [1, S], k \in [1, T]) \quad (1)$$

每个栅格的状态信息具体的表示如式 (2):

$$N_{ijk} = \begin{cases} 0, & \text{无障碍} \\ 1, & \text{有障碍} \end{cases} \quad (2)$$

其中: 当  $N_{ijk}$  值为 0 时, 表示当前网格是自由空间, 没有障碍物。当  $N_{ijk}$  值为 1 时, 表示当前网格是障碍物。理论上, 无人机飞行时有很多方向和细微的动作。然而, 考虑到模型的复杂性, 每个网格的中心被用作 A\* 算法的计算单元, 称为“节点”。当障碍物包含在一定的网格尺寸范围内时, 相应的节点被视为障碍物节点, 剩余的节点为自由节点。规定无人机在任何节点时, 可上下、左右、相邻和对角相邻节点移动, 即最多有 26 个运动方向。

## 2 A\* 传统算法与局部算法

### 2.1 A\* 传统算法

A\* 算法是最著名的路径规划算法之一, 它结合了基于最短路径的搜索和启发式搜索<sup>[12]</sup>。A\* 算法定义为最佳优先级算法, 通过从起点到目标点的距离  $f(n)$  来评估配置空间中的每个单元:

$$f(n) = h(n) + g(n) \quad (3)$$

A\* 算法的搜索思想是在搜索过程中, 选择所有  $f(n)$  小的点作为路径候选点, 构造路径候选点集。搜索到了目标点后, 从目标点开始, 在路径候选点集中向前搜索最佳路径。

在式 (3) 中, 节点  $n$  的评估函数的值由  $f(n)$  表示,

从单元到目标状态的启发式距离由  $h(n)$  表示<sup>[11]</sup>, 通过所选单元序列从初始点到目标点的路径长度表示为  $g(n)$ 。如式 (4)、(5) 所示。

$$h(n) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (4)$$

$$g(n) = |x_2 - x_1| + |y_2 - y_1| + |z_2 - z_1| \quad (5)$$

A\* 算法的具体实现取决于两个基本集合<sup>[13]</sup>: 开放集和封闭集, 记录为 OPEN 集和 CLOSED 集。开放集用于存储要检查的节点的信息, 封闭集用于存储已选择且不需要再次检查的节点的信息。A\* 算法的路径搜索过程是通过反复检查和更新开放集来实现的。

在 A\* 算法求解路径规划问题的过程中, 评价函数的选择尤为重要。由式 (3) 可知, 评价函数由现在成本函数  $h(n)$  和过去成本函数  $g(n)$  组成。当  $g(n)$  权重为 1,  $h(n)$  权重为 0 时, A\* 算法可视为 Dijkstra 算法, 此时算法更偏向起点, 是一种经典的最短路径查找算法。当  $g(n)$  权重为 0,  $h(n)$  权重为 1 时, A\* 算法可视为 BFS 算法。此时, 算法的启发性更强, 更偏向靠近目标点的节点。合理配置和改进评价函数评价比, 将使规划的路径更快、更平滑<sup>[13]</sup>。

### 2.2 A\* 局部算法

A\* 算法通过检查、比较和探测节点来实现对最佳路

径的搜索。在已知先验环境中没有动态障碍的情况下, 可以达到较好的规划效果。但是, 其本质为全局路径搜索方法, 在局部路径规划领域的应用受到限制。何雨枫等提出了其在未知动态障碍物环境下的局部路径规划算法。

在局部路径规划时, 传感器探测范围与 A\* 算法的环境信息要求之间相矛盾, 为解决这一问题, 假设传感器检测范围从当前节点扩展到下一个节点, 扩展集在三维空间中为 26 个节点, 即 a\* 算法的单步扩展的步长, 无人机的运动和 A\* 算法的扩展是同步执行的, 直接将 A\* 算法计算出的每个路径点提供给无人机。当无人机位于新节点中时, 机载传感器会扫描该节点的周围环境信息, 并将其提供给 a\* 算法。

图 1 是局部规划和全局规划的流程图。其中, 图 1 (a) 是局部规划的流程图, 图 1 (b) 是全局规划的流程图。

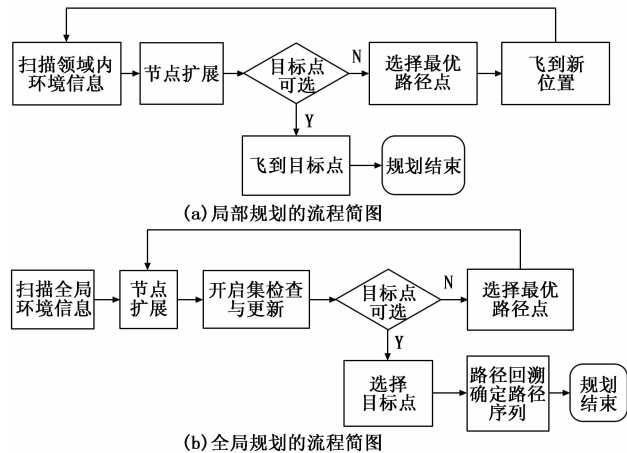


图 1 局部规划和全局规划的流程对比图

在局部路径规划中, 当无人机飞向新节点时, 清除开启集, 在新节点上重新扫描当前节点周围的环境信息, 找到所有可扩展的节点, 并将  $f(n)$ 、 $h(n)$  和  $g(n)$  存储在开启集中, 然后完成节点扩展。这一策略保证了无人机能够应对临时出现的火灾或火灾带来的塌方等动态障碍物, 解决了未知动态障碍物环境下的局部路径规划的问题, 局部规划算法程序流程如图 2 所示。

在局部路径规划过程中, 文献 [10] 对路径规划过程进行了细化, 改进了成本函数的选择和开启集的更新方法, 但存在收敛速度慢, 导致规划算法耗时长、路径扩展长、易陷入局部最优的问题。针对该问题, 在评价函数的权值分配和路径生成策略方面进行改进。

## 3 基于改进 A\* 全局与局部结合的算法

### 3.1 改进评价函数的权值分配

由 2.1 可知, 在路径搜索过程中,  $g(n)$  和  $h(n)$  对路径评估的影响不同, 选择适当的加权评估方法<sup>[14]</sup>, 会让路径规划更加迅速, 更平滑, 并消除路径扩展的冗余点。本文建立式 (6) 的评估方法:

$$f(n) = \alpha g(n) + \beta h(n) \quad (6)$$

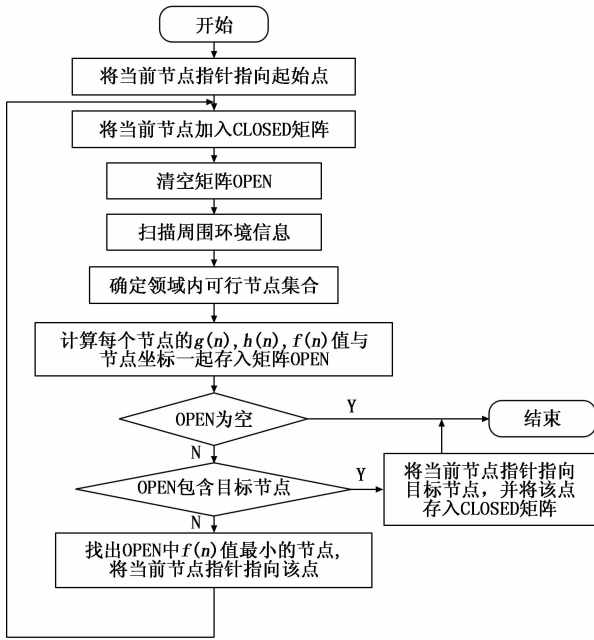


图 2 局部路径规划算法程序流程

在式 (6) 中,  $\alpha$  是到达当前节点的实际成本  $g(n)$  的权重,  $\beta$  为当前节点到目标节点的估计代价  $h(n)$  的权值。两项权值系数之和为 1, 随着实时节点的扩展, 当  $g(n)$  的影响因子越来越小,  $h(n)$  的影响因子越来越大时, 既能达到平衡启发式的目的, 又能优化路径缩短寻路时间的目的, 所以评价函数中  $\alpha$  和  $\beta$  应满足式 (7) ~ (9)。

$$\alpha + \beta = 1 \quad (7)$$

$$\bar{\alpha}(n) < 0 \quad (8)$$

$$\beta(n) > 0 \quad (9)$$

$$e^{-0.1n} + (1 - e^{-0.1n}) = 1 \quad (10)$$

构建  $g(n)$  与  $h(n)$  的权重系数如式 (10) 所示。在式中,  $n$  为实时路径扩展点, 取值范围为  $[1, N]$ ,  $N$  为路径扩展节点的个数。

$$f(n) = e^{-0.1n} * g(n) + (1 - e^{-0.1n}) * h(n) \quad (11)$$

将式 (11) 引入评价函数中可得可调节权重比例的评价函数  $f(n)$ , 但是由于扩展路径点  $n$  为一个有限值, 所以  $\alpha$  不能接近极限值 0,  $\beta$  不能接近极限值 1, 达不到权值分配的目的。本文引入自变量放大和比例缩减。

$$f(n) = e^{-0.1n \times \epsilon / N} * g(n) + (1 - e^{-0.1n \times \epsilon / N}) * h(n) \quad (12)$$

在式 (12) 中,  $\epsilon$  取值为远大于  $N$  的数, 此时, 随着路径节点  $n$  的扩展,  $\alpha$  的变化范围由 1 变化为 0,  $\beta$  的变化范围由 0 变化为 1, 实现了实时动态调节权值系数的目的。

### 3.2 改进路径生成策略

在文献 [10] 中, 当舰舱过于复杂, 障碍物过多时, 无人机在局部搜索过程中不能接近目标节点, 甚至陷入局部最优, 从而导致无人机无法完成任务。本文将改进路径生成策略, 进一步优化路径。

在无人机开始阶段, 无人机传感器模型在 Matlab 中显示为  $1.45r$  为半径的淡灰色球形区域, 即为传感器的扫描范

围, 在无人机开始阶段, 扫描全局环境信息, 默认按照全局规划路径进行节点扩展, 同时扫描邻域环境信息, 若未检测到邻域内的火灾或火灾带来的塌方等动态障碍物, 则继续按照全局路径规划路线飞行, 若检测到火灾或火灾带来的塌方等动态障碍物, 进行局部路径规划, 直到在传感器探测范围内无动态障碍物时, 完成局部规划进而开始全局路径规划……当目标点出现在 OPEN 集中, 则完成路径规划。这一策略保证了无人机不会在每一步扩展节点时重新规划路径, 缩短了规划时间, 同时避免陷入局部最优的情况。

改进三维 A\* 全局与局部结合算法部分的程序流程图如图 3 所示。

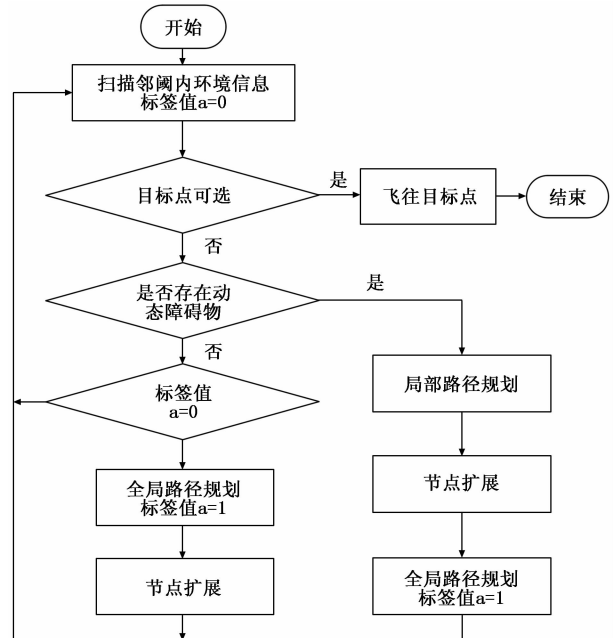


图 3 改进三维 A\* 全局与局部结合算法程序流程

与文献 [10] 中的算法相比, 评价函数权值分配的改进能够更快逃离初始点和接近目标点, 减少了路径规划长度; 路径生成策略的改进, 不会在每一步扩展节点时, 重新规划路径, 同时避免了陷入局部最优问题, 在很大程度上缩短了路径规划时间。改进算法有效缩短了路径规划长度, 提高了路径规划效率。

### 4 仿真分析

为了验证改进算法的有效性, 本文采用 Matlabr2014 (b) 软件平台进行仿真, 硬件平台为 Intel (R) Core (TM) i5-45703.20 GHz CPU, RAM 8 GB。

本文将环境信息按照栅格法将空间分解成  $20 * 20 * 20$  个小方格, 起始点目标点已标出, 黑色实心点为无人机的空间模型, 淡灰色球形区域为无人机传感器的感知范围, 浅色柱状物为静态障碍物示例, 深色柱状物为火灾或火灾带来的塌方等动态障碍物示例, 环境空间模型如图 4 所示。

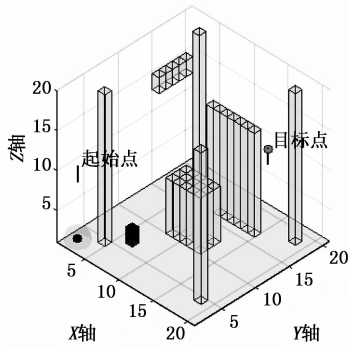


图 4 环境空间模型

本实验分为三部分: 一部分为改进的评价函数权值分配在全局 A\* 算法的仿真对比试验; 第二部分为验证改进的路径扩展策略算法, 与文献 10 的算法进行仿真对比; 第三部分为在相同障碍物环境中, 文献 10 算法陷入局部最优时, 使用本文提出的算法的仿真对比试验。

#### 4.1 实验一改进评价函数权值分配

本文改进了动态调整 A\* 算法评价函数的权重比例, 原算法与改进后的算法在 A\* 全局寻路算法中如图 4 所示。图 5 (a), (b) 分别为三维仿真的 XY 视图, 三维视图。在图 5 中, 虚线为原 A\* 三维全局规划算法, 实线为改进评价函数的权重比的算法, 明显可以看出, 改进算法减少了冗余点, 优化了路径, 改进评价函数的权重比算法优于原算法。

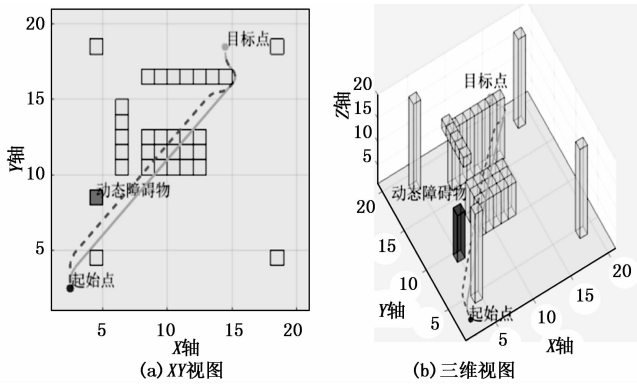


图 5 改进权重比例在全局 A\* 算法的对比

在相同障碍物、起始点、目标点环境下, 将两种算法的子程序分别运行 50 次后, 得到的相关参数平均值如表 1 所示。

表 1 相关参数对比表

算法类别	规划时间/s	语句执行步数	路径代价
原算法	2.138 6	64 251	24.755 9
改进权重比算法	2.096 6	64 202	22.846 1

在表 1 中, 两种算法语句执行步数相差不大, 但算法耗时缩短了 2%, 路径长度缩短了 7.7%。这是因为在全局路径规划中, 每一步路径扩展时, 无人机的传感器扫描耗

费一定时间, 但是路径距离缩短弥补了时间上的不足。

#### 4.2 实验二改进路径生成策略

本文引入了数据标签, 改进了路径生成策略, 避免了步步计算扩展路径, 在改进局部算法仿真实验中, 无人机默认使用全局规划算法, 如果在途中无人机的探测范围内未检测出火灾或火灾带来的塌方等动态障碍物, 则不用进行局部规划, 从而更快速到达目标点; 如果在途中无人机的探测范围内检测出火灾或火灾带来的塌方等动态障碍物, 则进行局部规划, 避过动态障碍物后, 重新规划全局仿真, 到达目标点。

本部分仿真分为两种情况, 第一种情况为在全局规划中未检测出动态障碍物, 算法分析如下所示。

原三维局部 A\* 算法仿真如图 6 所示, 图 6 (a), (b) 分别为原算法三维仿真的 XY 视图, 三维视图。提出的改进路径生成策略的算法仿真如图 7 所示。图 7 (a), (b) 分别为改进算法三维仿真的 XY 视图, 三维视图。

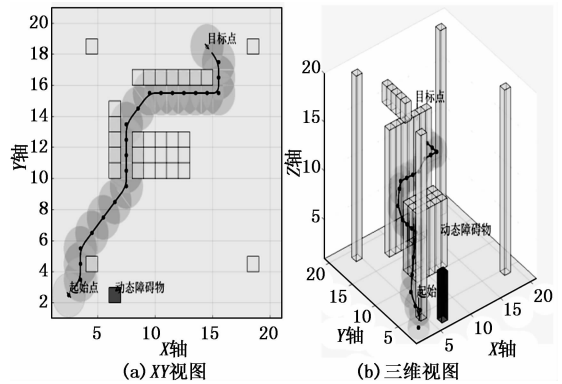


图 6 原算法仿真实验结果

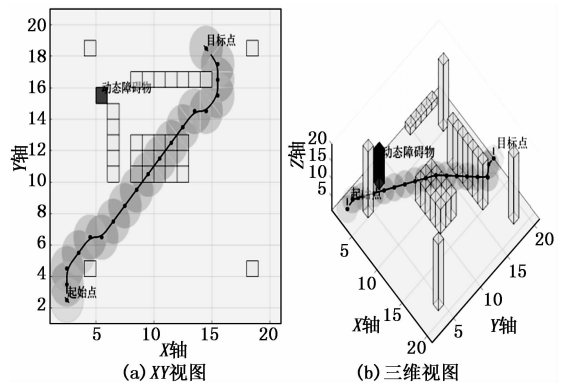


图 7 改进算法仿真实验结果

两种算法所的参数对比如表 2 所示。

表 2 相关参数对比表

算法类别	规划时间/s	语句执行步数	路径代价
原算法	3.354 6	5 146	29.517 8
改进路径生成策略算法	2.686 0	5 202	25.846 1

在表 2 中，相比原 A\* 算法，在改进路径生成策略算法中，语句执行步数有所提高，但是总规划时间约提高了 20%，路径长度缩短了 12.4%。这是因为全局与局部算法相结合，提高了算法的执行次数，但是无人机在默认按照全局路径飞行过程中，未检测到动态障碍物，总路径规划时间大大提高，缩短了路径长度。

第二种情况为在全局规划中检测出动态障碍物，算法仿真分析如下所示。

原三维局部 A\* 算法仿真如图 8 所示，图 8 (a)，(b) 分别为原算法三维仿真的 XY 视图，YZ 视图，XZ 视图，三维视图。本文提出的改进路径生成策略的算法仿真如图 8 所示。图 9 (a)，(b) 分别为原算法三维仿真的 XY 视图，YZ 视图，XZ 视图，三维视图。

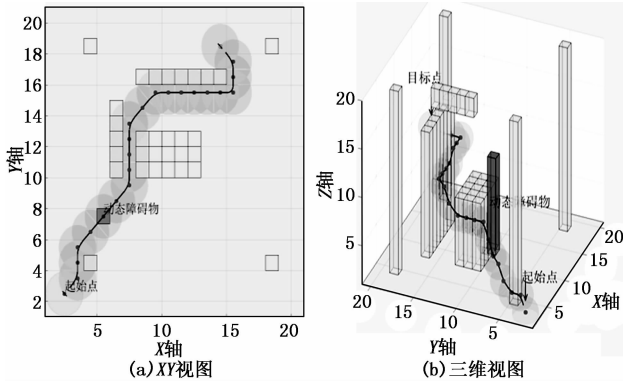


图 8 原算法仿真实验结果

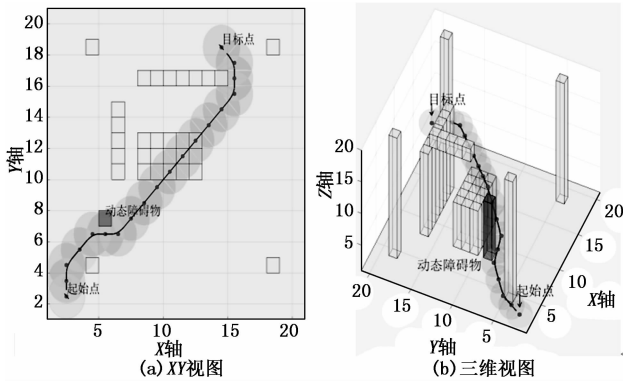


图 9 改进算法仿真实验结果

两种算法所的参数对比如表 3 所示。

表 3 相关参数对比表

算法类别	规划时间/s	语句执行步数	路径代价
原算法	3.408 9	5 152	29.530 8
改进路径生成策略算法	3.421 2	7 850	27.356 4

由表 3 可看出，语句执行步数多了 50%，算法耗时比原文增加了 3.6%，但是路径代价缩短了 7.4%。这是因为在默认的全局路径规划线路中出现了火灾或火灾带来的塌方等动态障碍物，于是采用局部路径扩展算法，避开动态

障碍物后，重新规划全局路径，所以语句执行步数有所增加，在保证算法耗时无明显增加的前提下，路径代价缩短幅度更大。

### 4.3 实验三局部最优对比

在复杂地图中，原文局部 A\* 算法易陷入局部最优情况，本文提出的改进路径生成策略算法能够避免陷入局部最优。图 10 和图 11 为原算法与改进算法仿真实验图。图 10 (a)，(b) 分别为原算法三维仿真的 XY 视图，三维视图。图 11 (a)，(b) 分别为原算法三维仿真的 XY 视图，YZ 视图，XZ 视图，三维视图。在图 10 中，原 A\* 算法由于地图较为复杂，局部路径扩展陷入局部最优，在相同地图中，改进算法有效避免陷入局部最优，如图 11 所示。

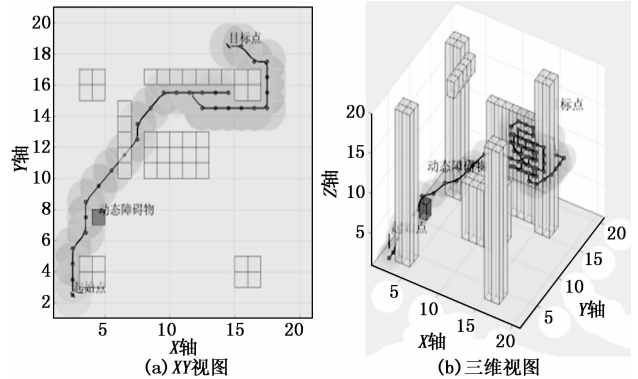


图 10 原算法仿真实验结果

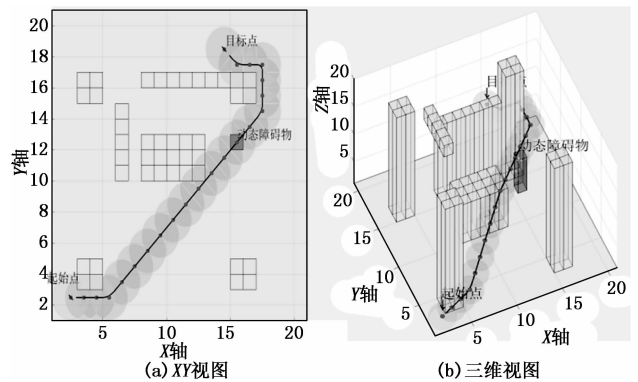


图 11 改进算法仿真实验结果

两种算法所的参数对比如表 4 所示。

表 4 相关参数对比表

算法类别	规划时间/s	语句执行步数	路径代价
原算法	51.354	97 384	125.630
改进路径生成策略算法	2.804 7	5648	32.425 1

在表 4 中，在改进路径生成策略算法中，采用全局与局部相结合的路径扩展策略，改进算法耗时大大降低，路径长度大幅缩短，在避开静态和动态障碍物后，快速到达目标点，取得理想效果。