

低空复杂环境下基于采样空间约减的无人机 在线航迹规划算法

温乃峰¹ 苏小红¹ 马培军¹ 赵玲玲¹

摘要 针对低空复杂环境下障碍物密集且类型多样、带有多通道并存在不确定信息的无人机在线航迹规划问题, 为了减少碰撞检测次数, 提高航迹搜索速度, 降低航迹代价, 提出一种基于采样空间约减的无人机在线航迹规划算法. 算法通过引入代价模型, 提出约减域逐步构造方法, 引导规划树快速有效扩展, 改善了基于动态域的快速拓展随机树 (Dynamic domain rapidly-exploring random tree, DDRRT) 算法中存在的采样空间过度约减问题. 算法通过密度划分索引的方法逐步构建多棵 Kd 树 (K-dimensional tree) 并采用多近邻节点搜索方法, 加快了近邻树节点搜索速度. 仿真实验结果表明, 与 DDRRT 方法相比, 该方法在保证对采样空间约减合理性的同时, 提高了航迹规划效率和通道内的寻路能力.

关键词 在线航迹规划, 多约束条件, 快速拓展随机树算法, 采样空间约减, 碰撞检测

引用格式 温乃峰, 苏小红, 马培军, 赵玲玲. 低空复杂环境下基于采样空间约减的无人机在线航迹规划算法. 自动化学报, 2014, 40(7): 1376–1390

DOI 10.3724/SP.J.1004.2014.01376

Sampling Space Reduction-based UAV Online Path Planning Algorithm in Complex Low Altitude Environments

WEN Nai-Feng¹ SU Xiao-Hong¹ MA Pei-Jun¹ ZHAO Ling-Ling¹

Abstract The unmanned aerial vehicle (UAV) online path planning in low altitude complex environments is complicated due to the planning spaces of densely distributed obstacles with various shapes, narrow passages for the solution path to pass through, and uncertain information. For solving this problem, a sampling space reduction-based algorithm is proposed to reduce the number of collision detection calls, accelerate the path-search process and decrease the path cost. To deal with the over-reduction problem existing in the dynamic domain rapidly-exploring random tree (DDRRT) method, the algorithm makes the space reduction gradually by employing a cost model. Thus the planning tree can extend rapidly and efficiently under the guidance of the reduction. It also promotes the near neighbors searching speed by a new storage structure for tree nodes and a novel near neighbor searching approach. Indexes are built based on the density of tree nodes to construct the storage structure composed by multiple K-dimensional trees (Kd trees). Simulation results certify that our algorithm can ensure the rationality of the sampling space reduction and improve the efficiency of path planning and the ability of path-searching in passages, as compared to the DDRRT.

Key words Online path planning, multi-constraint, rapidly-exploring random tree (RRT), sampling space reduction, collision detection

Citation Wen Nai-Feng, Su Xiao-Hong, Ma Pei-Jun, Zhao Ling-Ling. Sampling space reduction-based UAV online path planning algorithm in complex low altitude environments. *Acta Automatica Sinica*, 2014, 40(7): 1376–1390

当无人机执行威胁区突防或救援任务时, 需要在障碍物密集、不规则、存在大量通道区域、存在未知信息的低空复杂环境下飞行^[1–3], 为了满足航迹

的安全、可飞和优化性, 规划过程必须满足环境的多约束条件^[4–7]. 此时, 在任务分配的有限时间内在线搜索和优化航迹, 提高算法在通道内的寻路能力问题变得更加具有挑战性. 在瞬息万变的战场环境下, 无人机常常需要临时更改航线, 以应对突发状况^[1], 此时无人机系统必须具备在线规划能力.

根据航迹点获取方式, 将航迹规划方法分为确定性方法和随机化方法^[6]. 确定性方法是基于环境模型规划航迹. 主要方法有 A* 算法和人工势场法. 其中, A* 算法利用先验信息启发搜索, 以减小规划范围, 但难以处理高维空间下的规划问题; 人工势场

收稿日期 2013-01-31 录用日期 2014-01-03
Manuscript received January 31, 2013; accepted January 3, 2014
国家自然科学基金 (61175027) 资助
Supported by National Natural Science Foundation of China (61175027)
本文责任编辑 李平
Recommended by Associate Editor LI Ping
1. 哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001
1. School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001

法易陷入局部最优, 在通道中存在摆动现象^[1, 8-9]. 确定性方法难以处理多约束条件, 难以重新规划航迹. 随机化方法是通过对航迹段碰撞检测, 避免对障碍物的建模^[6], 其优点是可扩展性强, 可以分段检测约束满足情况^[10], 方便嵌入启发式信息, 能有效避免局部最优和组合爆炸问题^[11], 能自动消除空间分辨率、时间复杂度、航迹精度之间的矛盾^[12]. 主要方法有随机路线图方法 (Probabilistic road map, PRM) 和快速拓展随机树算法 (Rapidly-exploring random tree, RRT) 方法. 随机路线图方法的预处理过程存在多个两点边值问题^[13-14], 难以处理未知环境信息. RRT 方法通过连续位形空间内随机采样点的引导, 逐步构造以航迹起点为根, 连通目标点的规划树, 离散搜索航迹. 其中, 位形空间包含航迹点所有的位置和姿态^[6]. 该方法复杂度低, 重规划容易, 适用于存在硬性时间限制的在线规划问题. 但是, 在通道内的寻路能力不强^[15-16], 而该能力是无人机利用地形躲避威胁的重要保证; 难以保证在复杂环境下的规划效率及鲁棒性^[15-18]; 且未考虑航迹优化问题.

为了提升 RRT 算法在狭窄通道内寻路能力, 文献 [15-16] 提出基于环境特征采样的 RRT 方法, 但该方法会增加无人机发生碰撞的概率.

为了提高 RRT 方法在复杂环境下的规划效率和鲁棒性, Yershova 等提出了基于动态域的快速拓展随机树算法 (Dynamic domain rapidly-exploring, DDRRT)^[17-18], 但存在采样空间的过度约减问题.

为了启发规划树向低代价区域拓展, 分支定界方法通过增加目标函数上/下界以排除非最优解区域^[19]. 文献 [20] 基于分支定界方法提出了 RRT* 方法, 通过为树节点选择当前最优扩展以优化规划树

拓扑结构, 但待选扩展方式数量难以确定. Urmson 等提出了启发式 RRT (Heuristically biasing RRT) 算法, 根据估计最优路径代价与经过采样点的估计最优路径代价比值, 建立启发式函数以约减高代价采样点^[21], 但最优路径代价难以估计; Lee 等提出了基于代价的 RRT (Cost-based RRT) 方法, 根据采样点的代价确定其被保留的概率^[22]; Jaillet 等提出了基于转移的 RRT (Transition-based RRT) 方法, 基于 Metropolis 法则和 Boltzmann 概率模型构建树节点向采样点转移的测试模型, 以筛选采样点^[23].

为了在线优化航迹, Ferguson 等提出时间动态调配的 RRT 算法, 迭代修正和优化航迹^[24]; Yang 等基于多约束条件约减航迹点^[25]; Karaman 等基于 RRT* 方法和时间动态调配的思想, 在有效时域内逐步优化航迹^[26].

1 DDRRT 算法对采样空间的过度约减问题

如图 1(a) 中多边形区域所示为各树节点的 V 图区域, 等价于树节点的近邻区域及其采样域, 其数学表述如定义 1 所示. 空间 V 图划分过程自动隐式进行, 对其进行标识便于算法原理分析. 圆形区域所示为扩展不成功的树节点的动态域, 空白部分所示为所有树节点的约减域. 树节点动态域、可视域、可拓展区定义如下.

定义 1. 树节点 $Node_i$ 的近邻区域 $V(Node_i) = \{N' \mid Dis(Node_i, N') \leq Dis(Node_j, N'), j \neq i\}$, 其中, $Dis(Node_i, N')$ 为 $Node_i$ 与 N' 间距离, 树节点的近邻区对应其可被选择扩展的采样域.

定义 2. 树节点 $Node_i$ 的动态域 $DD(Node_i) = \{N' \mid Dis(Node_i, N') \leq R\}$, 其中, $Dis(Node_i, N')$

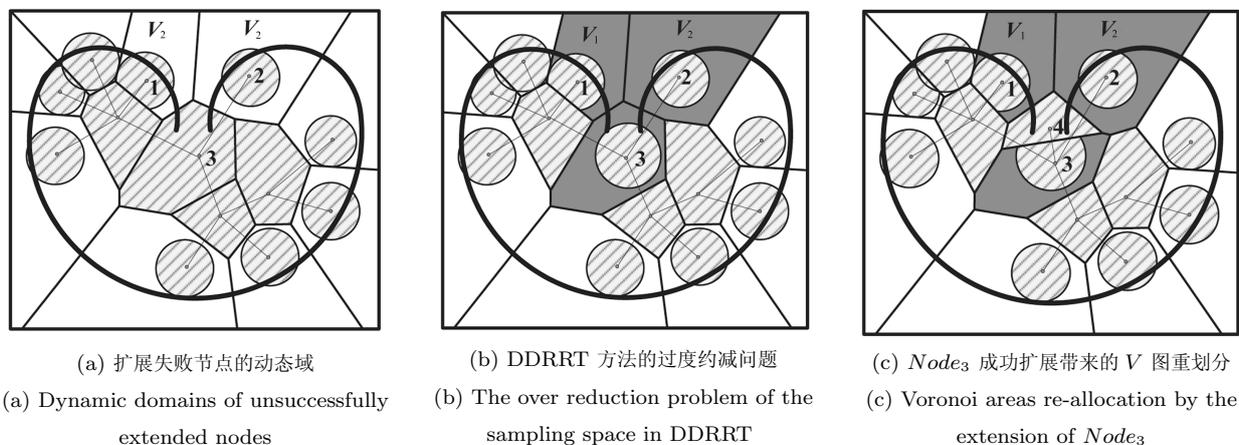


图 1 DDRRT 算法解决 Bug trap 问题的过程

Fig. 1 The process of solving the Bug trap puzzle by the DDRRT algorithm

为 $Node_i$ 与 N' 间距离, R 为动态域半径.

定义 3. 树节点 $Node_i$ 的可视域 $VD(Node_i) = \{N' \mid L(Node_i, N') \subseteq C_{free}, N' \subseteq C_{free}\}$, 其中, C_{free} 为位形空间内的无碰撞采样空间, 即 $C_{free} \cap C_{obs} = \phi$, C_{obs} 为障碍物占据空间, $L(Node_i, N')$ 为 $Node_i$ 与 N' 间连线.

定义 4. $Node_i$ 的可拓展区 $D_{extend i}$ 对应其可拓展成功的采样域, RRT 方法族中, 树节点 $Node_i$ 的可拓展区为 $D_{extend i} = V(Node_i) \cap VD(Node_i)$.

RRT 算法难以解决如图 1(a) 所示的 Bug trap 难题, 图 1 中黑色曲线为凹障碍物, 规划树根节点位于凹障碍物内, 其规划解必须经过狭窄通道逃离凹障碍物^[2, 6, 12, 15-18]. 如图 1(a) 所示, 首先, Bug trap 问题解必须通过狭窄通道, 而 RRT 方法在通道内的寻路能力不强, 以线性扩展函数为例, 树节点在通道内的可拓展区较小^[27-28]; 其次, RRT 方法难以逃离凹障碍物, 这是因为树节点被选择扩展的概率与其采样域大小成正比, 叶子节点的采样域比内节点大, 因此被扩展几率更大^[6, 15-16, 27-28], 如图 1(a) 所示, 算法不能通过叶子节点的扩展来逃离障碍物, 而通过内节点的扩展修正航迹解的速度较慢; 另外, 在 Bug trap 问题中, 传统的带有目标倾向的启发方法 (以树节点与目标间距离定义启发式函数值) 易使规划算法陷入局部最优^[29].

为了提高 RRT 方法在复杂环境下的效率和鲁棒性, 文献 [17] 提出了 DDRRT 方法, 该方法也是现有的唯一能够解决 Bug trap 问题的方法, 它通过对树节点采样空间的约减, 以调整空间采样概率模型, 利用环境信息减少碰撞检测调用次数, 同时采用 Kd 树 (K-dimensional random tree) 存储结构加快最近邻节点搜索速度, 但该算法仍存在采样空间过度约减问题.

DDRRT 方法的基本思想是, 为扩展不成功树节点 $Node_i$ 添加半径为 $R (0 \leq R \leq \infty)$ 的动态域 $DD(Node_i)$ 时, 只有当采样点落在 $Node_i$ 的 $V(Node_i) \cap DD(Node_i)$ 时, 才对 $Node_i$ 进行扩展, 即对落在 $Node_i$ 的 $V(Node_i)$ 内、动态域外的采样点进行直接滤除^[17], 以此对 $V(Node_i)$ 进行约减 (其可拓展成功采样域同时被缩小为 $DD(Node_i) \cap D_{extend i}$, 其中, $D_{extend i}$ 为约减前的 $Node_i$ 的可拓展区), 以调整树节点被选择扩展的概率.

DDRRT 方法存在采样空间过度约减问题, 该问题在求解 Bug trap 问题时更加突出. 如图 1(a) 所示, $Node_i$ 的动态域 $DD(Node_i) (1 \leq i \leq n)$, n 为树节点总数, $Node_i$ 对应约减域 $RD_i = V(Node_i) - DD(Node_i)$, 所有树节点的约减域 $RD = \bigcup_{i=1}^n RD_i$, RD 中存在大量合理采样空间,

对其约减会降低算法效率. 如图 1(b) 所示, 若 $Node_3$ 扩展失败, 则 $V(Node_3)$ 中通道部分合理采样域也被约减, 算法更难以求得规划解. 只有当 $Node_3$ 向通道方向扩展成功后, 空间进行 V 图重划分, $V(Node_1)$ 和 $V(Node_2)$ 缩小, 其部分区域成为 $Node_4$ 的采样域, 过度约减问题才得到改善, 如图 1(c) 所示, 但该事件发生概率较小.

产生过度约减问题有两个原因: 1) 动态域大小难以确定, 以图 1(b) 中的 $Node_3$ 为例, 若 $DD(Node_3)$ 变小, 则 $V(Node_3)$ 被约减空间变大, 过度约减问题更严重; 反之, 则可能会降低算法的效率. 2) 动态域的位置定义不合理, 当树节点在某个方向扩展失败后, 算法对以它为中心的所有方向的采样域都进行了约减, 而实际情况, 障碍物对规划树的阻碍只能是某些方向的, 因此, 动态域难于定义的原因是无法考虑 V 图划分的随机性和障碍物的方向性. 3) DDRRT 算法缺乏对规划树扩展的启发, 算法是单纯基于欧氏距离的, 欠缺对环境特征信息的利用, 如图 1(a) 所示, 与通道区域位置关系更直接的 $Node_3$ 无法在区域 $V(Node_1)$ 和 $V(Node_2)$ 内扩展, 造成其中的部分通道采样区被浪费.

本文针对 DDRRT 方法存在的过度约减问题, 结合代价启发式方法, 提出基于采样空间约减的无人机在线航迹规划算法.

2 基于空间约减的在线航迹规划算法

航迹规划的任务是寻找一条可到达目标点的安全、可飞和可控的航迹. 本文算法为多阶段规划算法. 首先, 基于采样空间约减, 快速搜索多边形航迹; 然后, 结合 RRT* 方法在线优化航迹; 最后, 对航迹进行可飞性处理.

2.1 算法框架

本文提出的基于采样空间约减的在线航迹规划算法模型如图 2 所示. 模块 1 为规划提供决策支持; 模块 2 根据规划树生长情况和环境特征定义动态采样边界并建立代价模型, 将计算结果传递给模块 3; 模块 3 规划局部航迹, 同时约减不合理采样空间, 更新环境信息, 将实时规划结果反馈给模块 2; 算法在系统分配时间内, 迭代执行第 2、3 模块, 最后在规划树上搜索一条安全的多边形航迹; 模块 4 对多边形航迹进行可飞性处理.

2.2 预处理

本文算法在规划之前, 根据环境先验信息对空间进行预处理. 算法根据无人机导航性能要求^[30], 以及飞行器的最小直行航迹段长度为标准进行空间栅格化, 并定义栅格化后的子区域为基本约减单元.

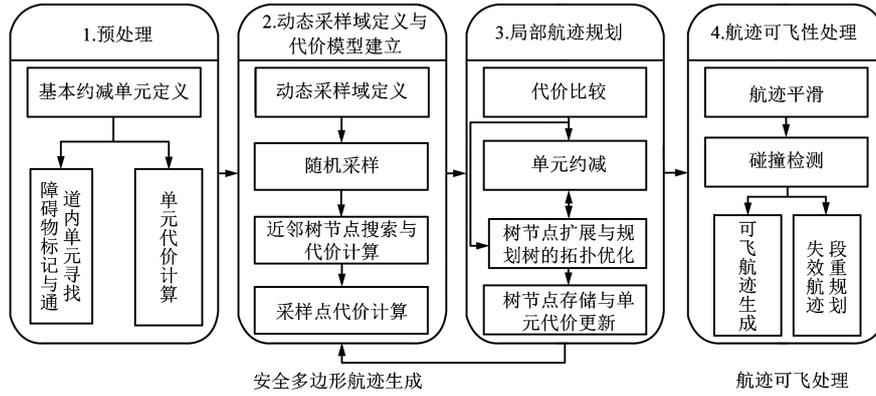


图2 基于采样空间约减的无人机在线航迹规划算法框架

Fig. 2 The UAV online path planning algorithm framework based on sampling space reduction

基本约减单元代价 C_{area} 的计算方法如下式所示:

$$C_{area} = \alpha_1 \times C_{nos} - \alpha_2 \times C_{rd} + \alpha_3 \times C_{ph} \quad (1)$$

其中, α_1 、 α_2 、 α_3 为权值参数, 且 $\alpha_1 + \alpha_2 + \alpha_3 = 1$; C_{nos} 为单元内部障碍物复杂度相关代价. 障碍物的存在会阻碍算法规划过程^[23], 定义 $C_{nos} = \sum_{i=1}^n (\frac{L_i}{L_a} + \frac{W_i}{W_a} + \frac{H_i}{H_a})$, 其中, n 为与单元相交的障碍物数量. L_i 、 W_i 、 H_i 为第 i 个障碍物与单元相交部分长方体包围盒的长、宽、高. L_a 、 W_a 、 H_a 为单元长、宽、高. C_{rd} 为代价奖励值, 通过减小通道区域的代价, 引导规划树向其中扩展, 以提高算法在通道内的寻路能力. C_{ph} 为代价惩罚值, 通过增加不合理扩展区和威胁区代价, 降低树向其中扩展的概率. 初始 C_{area} 值根据任务特点与先验信息确定, 并在任务执行过程中, 根据无人机反馈的环境信息和规划树扩展情况实时更新.

算法对与障碍物相交的基本约减单元进行标记, 并根据单元与障碍物的位置关系, 判断其是否处于通道区域. 算法认为, 如果出现下面两种状况, 则单元处于通道内: 1) 单元与多于一个障碍物相交或与同一障碍物多次相交; 2) 一定范围内的相邻单元与多于一个障碍物相交或同一障碍物多次相交, 范围大小根据任务特点定义.

2.3 动态采样边界定义与代价模型建立

如图 3(a) 所示, 规划环境相同的前提下, 采样边界越小, 内节点扩展概率越大, 规划树的修正趋势越强, 修正节点越多, 越有利于发现狭窄通道区域中的航迹, 但同时规划树向外拓展速度越慢; 反之, 叶子节点扩展概率越大, 如图 3(b) 中叶子节点占据的 V 图区域明显大于图 3(a), 因此, 规划树的扩张趋势越强. 约减域的大小、位置对规划求解过程影响与采样边界大小成反比^[17-18, 27-28]. 为了均衡规划树的扩张和修正趋势, 增强约减操作对规划过程的影响, 本文依据规划树规模建立动态采样边界.

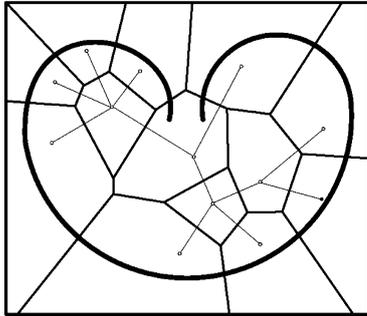
本文规定任意时刻的动态采样边界为当前规划树的最小包围盒向外扩充 l ($l \geq 1$) 倍 RRT 步长. 无人机系统存在最小直行航迹段长度约束和最小转弯半径约束^[31-32]. 故定义 RRT 方法的步长 $\epsilon = s_{\min} + 2r_{\min}$ (s_{\min} 为最小直行航迹段长度, r_{\min} 为最小转弯半径). 此时 $V_L \geq V_{OL} \geq V_I$ 发生的可能性较大. 其中, V_L 为规划树的非最外侧叶子节点的平均 V 图区域, V_{OL} 为树的最外侧叶子节点的 V 图区域, V_I 为规划树的内节点的平均 V 图区域.

$V_L \geq V_{OL} \geq V_I$ 的原因如下: 图 3(c) 虚线框为动态采样边界, 内部实线框为规划树最小包围盒. 假设树最外侧叶子节点的正前方 $2l$ 倍 RRT 步长位置存在一个假想规划树节点, 如图 3(c) 中虚线边和它连接的动态采样边界外的节点所示. 此时, 最外侧叶子节点变成规划树的内节点, 并且距相邻叶子节点 $2l$ 倍步长, 而规划树的其他相邻内节点之间距离为一倍 RRT 步长. 根据 V 图的划分方法^[27-28], $V_{OL} \geq V_I$ 的可能性较大. 由于其他叶子节点距动态采样边界边缘更远, $V_L \geq V_{OL}$ 可能性较大, 因此, 出现 $V_L \geq V_{OL} \geq V_I$ 的可能性较大. 在均匀采样无障碍物情况下, $S(V_{OL}) = (2l)^{d-1} \cdot S(V_I)$, S 为估值函数, d 为空间维度. 当 $l = 1$ 时, 规划树仍能均衡扩张、修正趋势, 算法仍优于 DDRRT 方法, 当 $l = \infty$ 时, 算法效率向 RRT 方法退化, 而该均衡性要求二维空间中 $l < 5$, 三维空间中要求 $l \leq 3$, 本文取 $l = 3$.

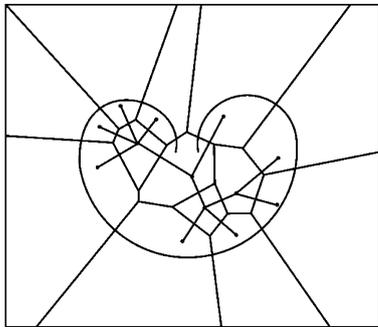
本文的代价模型指约减单元代价、代价距离、采样点及规划树节点代价定义和采样点代价上限定义. 航迹点的方向和位置同样重要^[5, 33], 定义航迹点的方向为在规划树中的入边方向. 根据飞行任务特点定义代价距离, 如下式所示:

$$C(Node_i, Node_j) = \beta_1 \times D(Node_i, Node_j) + \beta_2 \times C_{turn} + \beta_3 \times C_{slope} \quad (2)$$

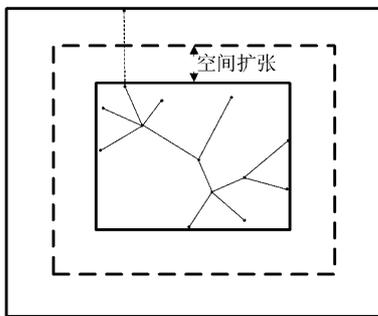
其中, $C(Node_i, Node_j)$ 为 $Node_i$ 与 $Node_j$ 间的代价距离; $\beta_1, \beta_2, \beta_3$ 为权值参数, 且 $\beta_1 + \beta_2 + \beta_3 = 1$, $D(Node_i, Node_j)$ 为 $Node_i$ 与 $Node_j$ 间的欧氏距离, C_{turn} 为两节点的单位方向向量在水平面上投影的夹角, C_{slope} 为两节点的单位方向向量在垂直平面上投影的夹角. 两相邻航迹点间转弯角和爬坡角越小, 代价距离越近^[25].



(a) 采样边界较小时空间 V 图划分
(a) Voronoi areas allocation under small sampling boundary



(b) 采样边界扩大后空间 V 图划分
(b) Voronoi areas allocation under larger sampling boundary



(c) 根据规划树规模定义动态采样边界
(c) The definition of dynamic sampling boundary by the scale of planning tree

图 3 采样边界对规划树扩展过程影响及动态采样边界定义
Fig. 3 The sampling boundary's definition and its influence on planning tree exploration

定义树节点代价 C_{node} 为通过该节点的航迹最优代价估值, 如下式所示:

$$C_{node} = \gamma \times C_{s2n} + (1 - \gamma) \times C_{n2g}^* \quad (3)$$

其中, γ 为权值参数, C_{s2n} 为从规划树根节点到达树节点的航迹点间代价距离和, C_{n2g}^* 为树节点与目标间代价距离.

为了保证算法的概率完备性, 本文采用随机均匀采样策略. 采样点代价上限 $UBound = \min\{\max\{C_{node}, PCost-max\}, \max\{C_{node}\}$ 为欧氏距离测度下采样点多近邻树节点代价的最大值, $PCost-max$ 为预定义的最大航程代价. 定义采样点代价 C_{rand} 为经过采样点近邻节点和采样点的航迹集合中最优航迹代价估值, 如下式所示:

$$C_{rand} = \omega_1 \times \min\{(C_{near} + C_{seg})\} + \omega_2 \times C_{r2g}^* + \omega_3 \times C_{area} \quad (4)$$

其中, $\omega_1, \omega_2, \omega_3$ 为权值参数, 且 $\omega_1 + \omega_2 + \omega_3 = 1$; C_{near} 为规划树根节点到达采样点近邻树节点的航迹代价集合, 即式 (3) 中 C_{s2n} 的集合; C_{seg} 为近邻节点与采样点间的代价距离; 本文定义 $\min\{(C_{near} + C_{seg})\}$ 对应树节点为采样点的代价最近邻树节点; C_{r2g}^* 为采样点与目标间代价距离; C_{area} 为采样点所在基本约减单元代价. 式 (3) 中未考虑 C_{area} , 避免当采样点与规划树节点处于相同约减单元时, C_{area} 影响被消除.

2.4 局部航迹规划

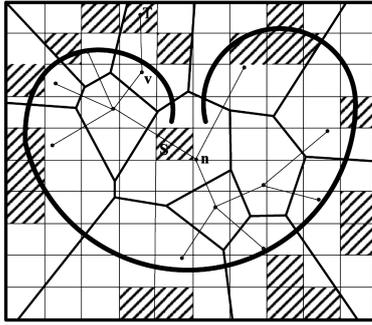
2.4.1 单元约减

本文的采样空间约减过程为, 若采样点代价大于上限值, 则认为该采样点不处于合理采样区, 滤除该采样点; 否则, 由采样点的代价最近邻树节点向采样点进行扩展, 生成新树节点和航迹段. 若新航迹段与障碍物间存在碰撞或采样点不满足代价限制, 则判断采样点所在基本约减单元是否处于通道区域, 若否, 则算法执行约减操作.

本文将约减操作定义为规划树节点与基本约减单元的关联代价惩罚操作, 即针对该树节点增加单元的 C_{area} 值, 以降低该树节点向该单元扩展的概率, 多次惩罚使得落入该单元的规划树节点的近邻采样点代价大于上限而被直接滤除, 相当于算法对此树节点约减了该单元.

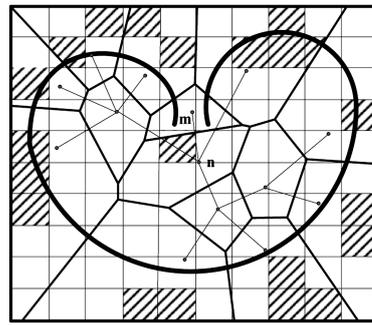
图 4 为本文算法解决 Bug trap 问题过程的演绎, 图中最外侧边框为当前时刻的动态采样边界. 图 4(a) 中阴影部分为所有树节点的约减域, 由于树节点的采样域外空间与树节点扩展无关, 因此, 约减操作相当于算法只对单元与相应规划树节点采样域相交部分进行了约减, 图 4(a) 中算法针对 $Node_v$

约减 T 所在单元时, 算法实际上只约减了单元与 $V(Node_v)$ 的相交部分. 这样做的好处是, 可以避免针对某树节点的约减对其他树节点的扩展造成影响, 避免浪费采样空间.



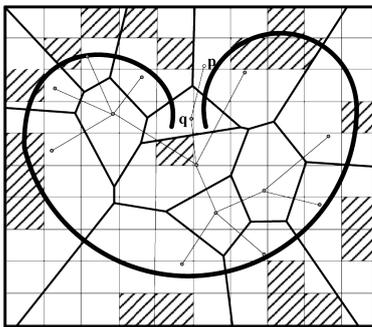
(a) 阴影区域为最近邻树节点的相关约减区域

(a) Shadowed areas are the reduction spaces related to the nodes nearest to the areas.



(b) $Node_n$ 成功扩展对关联约减区带来的变化

(b) The change of the reduction space related to $Node_n$ by the successful extension of $Node_n$



(c) 代价距离测度打破 V 图对树节点扩展的限制

(c) Cost distance relaxes the constraints of Voronoi areas to nodes exploration.

图 4 欧氏距离测度下本文算法解决 Bug trap 问题的过程
Fig. 4 The process of solving the Bug trap puzzle by our algorithm under the measurement of Euclidean distance

为了与 DDRRT 方法作相似性比较, 在欧氏距离测度下分析本文算法, 此时, 采样点代价 $C_{rand} = \omega_1 \times (C_{nearest} + C_{seg}) + \omega_2 \times C_{r2g}^* + \omega_3 \times C_{area}$, 其

中, $C_{nearest}$ 为欧氏距离测度下, 规划树根节点到达采样点最近邻树节点的扩展代价.

如图 4(b) 所示, 当 $Node_n$ 扩展成功后, 空间 V 图重划分, 图 4(a) 中 $Node_n$ 的约减区域 S 与图 4(b) 中 $V(Node_m)$ 相交部分超出 $V(Node_n)$, 约减失效, 阴影消失. 图 4(c) 中, $Node_q$ 与采样点 P 间虚线表示在代价距离测度下, $Node_q$ 可以打破 V 图限制向 P 扩展, $Node_q$ 的近邻区域划分更合理. 图 4 中, 通道区域不存在单元约减.

为了缓解 DDRRT 方法的过度约减问题, 本文定义了空间约减单元和动态采样边界, 构建了代价模型, 并在此基础上, 根据规划树生长情况和环境特征, 逐步构建与通道区域不相交的约减域, 只对障碍物方向区域进行约减, 并且针对某树节点的约减不会对其他规划树节点产生影响.

2.4.2 树节点扩展

无人机的多约束条件包括由障碍物和任务特性引起的全局约束、飞行器自身的局部运动学约束和在航程和油耗等方面的优化约束. 满足全局约束的航迹是安全的, 满足局部约束和优化约束的航迹是可飞的^[5, 31-32]. 本文的树节点 $Node$ 的扩展过程是由 $Node$ 向近邻采样点 $RandPt$ 进行直线扩展并生成新的航迹段 $c: [0, \Delta t] \rightarrow C$. 其中, C 为位形空间, $c(0) = Node$, $c(\Delta t) = NewPt$ ^[6, 34-36]. 若新航迹段是安全的, 则判断其局部约束满足情况. 本文通过 Dubins 控制函数 $(x, u, \Delta t, NewPt) = Dubins(Node, Rand)$ 检测 $Node$ 与 $NewPt$ 间是否存在曲率半径 $r = r_{min}$, 公切线长度 $s \geq s_{min}$ 的 Dubins 航迹 (此时的航迹为两点间最短航迹, 为 Dubins 曲线存在的极端情况). 若是, 则局部约束条件满足. 若新航迹段同时满足全局和局部约束, 则树节点扩展成功.

二维空间中可直接采用图 5(a)~(d) 所示的 CLC (C 为圆弧, L 为直线) 形式的二维 Dubins 曲线进行局部约束检测. 但在三维空间中, 若相邻两点的位姿异面, 则检测过程较复杂. 本文参考文献 [5, 33] 提出的直接由二维 Dubins 曲线扩展到三维的方法, 在进行二维检测前, 对位姿进行共面变换. 根据本文树节点方向的定义可知, 相邻树节点的方向向量共面 (令该平面为 S), 因此, 对 $Node$ 与 $NewPt$ 的共面变换只需绕 $Node$ 与 $NewPt$ 的方向向量, 将其主法向量均旋转至 S 内 (即滚动过程), 此时, $Node$ 与 $NewPt$ 的位姿均处于 S 内, 可采用二维 Dubins 曲线进行局部约束检测. Dubins 航迹只是无人机的初始控制航迹^[37-38], 为了保证航迹解的优化性和完备性, 在航迹平滑过程考虑所有形式的 Dubins 曲线.

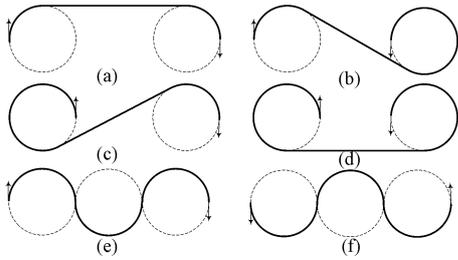


图 5 共面的两个位姿间 6 种形式的二维 Dubins 曲线
Fig. 5 The six styles of the two dimensional Dubins curves between two coplanar poses

Dubins 航迹长 $S_{Dubins} = \phi_s/\kappa_s + S_t + \phi_f/\kappa_f = f(\phi_s, \phi_f)$. 其中, ϕ_s 、 ϕ_f 为起止圆弧的圆心角弧度, κ_s 、 κ_f 为起止圆弧曲率, S_t 为直线航迹段长.

2.4.3 树节点存储

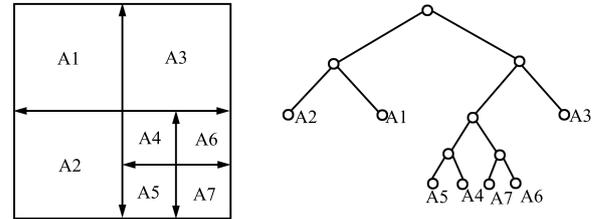
采用 Kd 树方法存储航迹点, 可以加快近邻节点的搜索速度, 但当 Kd 树的规模较为庞大时, 对树节点的频繁操作会产生过大的代价^[6, 39-40], 因此, 本文将采用多 Kd 树方法存储航迹点并为其建立索引树, 但是若子树数量过多, 索引树过大, 算法效率向传统二叉树方法退化. 为了解决这一问题, 本文根据航迹点分布密度在规划空间划分索引.

索引树叶子节点代表某个索引区域, 根节点对应整个规划区域. 当索引节点对应区域内的航迹点数量大于 m 且该区域范围大于某分裂阈值 Sc 时 (当 $m = Sc = \infty$ 时, 该存储方法等同于传统 Kd 树方法; 当 $m = Sc = 0$ 时, 方法等价于顺序存储方法), 对其执行划分操作, 重构索引树. 图 6(a) 为二维空间内索引划分与索引树构造过程与结果示例. 首先, 找到横轴方向的中点, 作垂直分裂面, 将空间一分为二, 按照与分裂面的左右关系生成两个新的索引树节点, 作为分裂前区域索引节点的子节点; 然后, 在纵轴方向执行同样的操作. d 维空间中, 在各维度上依次执行上述操作, 生成 $2^{d+1} - 2$ 个新的索引树节点, 包括 2^d 个叶子节点.

在索引树叶子节点对应的 2^d 个子空间内, 根据传统方法构造子 Kd 树. 图 6(b) 为二维空间内 Kd 树的构造过程与结果示例. 首先, 在横轴方向上找到处于中间位置航迹点 D_8 作为子 Kd 树根节点, 在 D_8 上作垂直分类面, 将航迹点分为两个子集, 作为 D_8 的左右子树; 然后, 在已划分的子集中, 寻找纵轴方向的中间航迹点作为各自子树的根节点, 执行划分和存储操作, 当空间维度为 d 时, 依次在第 i ($i \leq d$) 维度上执行上述操作; 递归执行上述步骤, 直到每个子集中包含的航迹点数小于等于 1^[6, 39-40].

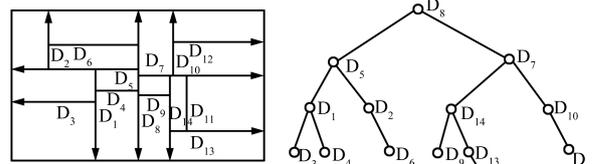
本文根据规划树节点的分布情况, 每次返回 N 个采样点的近邻树节点 (其中, $N \geq K$, K 为单次返回的近邻节点数下限值), 以解决 K 近邻方法中

K 值难以确定的问题. 为了加快近邻树节点的搜索速度, 本文采用基于 Kd 树存储结构搜索多近邻节点^[39-40], 按照树遍历的方式, 返回如定义 5 所示的近邻树节点集合.



(a) 索引区域划分过程及索引树构造结果

(a) The schematic of the process of area indexes subdivision and the result index tree



(b) 子 Kd 树的构造过程及结果

(b) The schematic of the process and the result of sub Kd trees construction

图 6 航迹点存储结构构造过程与结果示例

Fig. 6 The schematic of the construction and result of the storage structure of path tree nodes

定义 5. 采样点的近邻树节点集合 $Near(Rand, R_n) = \{Node | Dist(Rand, Node) \leq R_n\}$, 其中, $Rand$ 为采样点, $Node$ 为规划树节点, $Dist(Rand, Node)$ 为 $Rand$ 与 $Node$ 之间的距离, R_n 为近邻球半径.

2.5 航迹可飞性处理

为了提高航迹可飞性, 本文采用 Dubins 曲线对多边形航迹进行平滑. 在三维空间中, 参考基于微分几何的方法^[5, 33], 将三维 Dubins 平滑降维成二维问题, 构造异面位姿间的 Dubins 航迹过程是: 首先, 定义起止位姿的运动面, 在起始运动面内对起始位姿进行组合转弯; 然后, 对其执行滚动操作以将其转换到终止运动面; 最后, 采用二维 Dubins 曲线完成航迹构造. 为了使平滑后的航迹更加贴合初始航迹以尽量避免碰撞, 加快平滑速度, 本文优先采用航迹点的内测 (航迹点出、入边构成的小于 π 的夹角角分线方向对应的一侧, 如图 7(a) 所示) 曲率圆构造航迹. 为了解决由突然出现的障碍物导致的部分航迹段失效问题, 需要调整或重新规划失效部分, 可以采用添加障碍物安全圆、曲率调整、航迹点重采样方法^[5], 如图 7(b)、7(c)、7(d) 所示. 图 7(b) 方法不适用于三维空间中存在凹障碍物的情况. 图 7(c) 方法计算复杂度较大. 本文采用图 7(d) 方法.

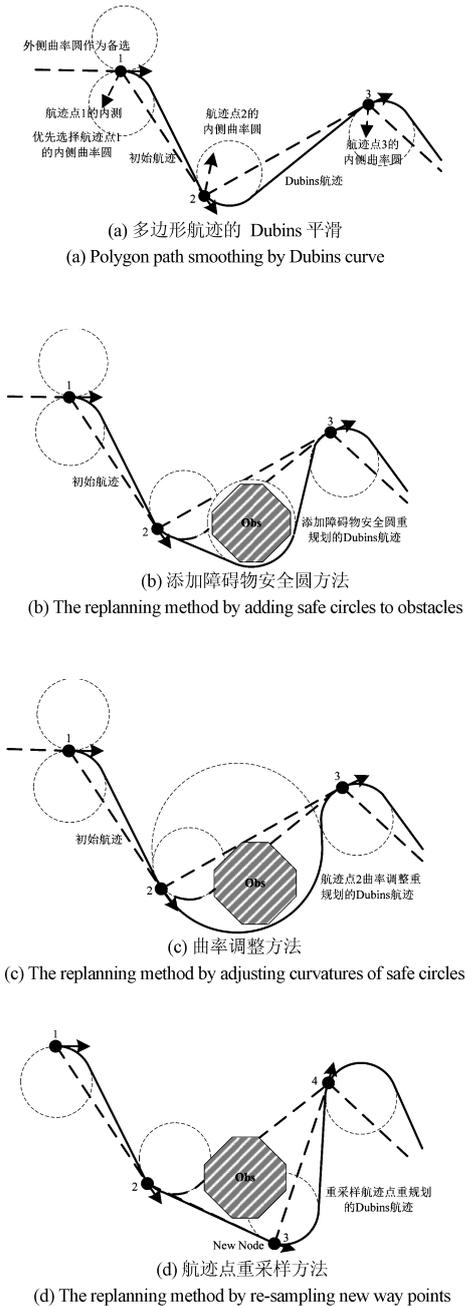


图7 多边形航迹的Dubins平滑及失效航迹段的重规划
Fig.7 Smoothing the polygon path by Dubins curve and the replanning of invalid sub path

2.6 算法实现

算法1. 基于采样空间约减的在线规划算法

步骤如下:

步骤1. 根据算法模型模块1 预处理规划空间.

步骤2. 定义动态采样边界并在其中随机采样, 寻找采样点 $Rand$ 的 K 近邻树节点 S_{near} , 计算 S_{near} 代价、 $Rand$ 代价和采样点代价上限 $UBound$, 为采样点选择代价最小的近邻节点 q_n , 以此计算采样点代价 C_{rand} .

步骤3. 若 $C_{rand} > UBound$, 对采样点所在基本约减单元 $Rarea$ 执行与 q_n 关联的代价惩罚操作, 即该单元代价 C_{area} 增加为 $(1 + \delta) \times C_{area}$, 返回步骤2; 否则, 由 $Node_i$ 向 $Rand$ 扩展.

步骤4. 若扩展成功且规划树中存在连通目标点的分支, 则执行规划树拓扑结构优化循环, 依次对 S_{near} 内 $Node_i$ 外的其他近邻树节点 $Node_k$ 进行判断: 判断将新树节点 $NewPt$ 作为父节点能否降低自身代价, 若能且 $NewPt$ 与 $Node_k$ 间存在无碰撞航迹段, 则对 $Node_k$ 进行父节点重定位. 算法以随机采样点为引导, 逐步修正/优化初始随机航迹. 若扩展失败, 对 $Rarea$ 执行与 $Node_i$ 关联的代价惩罚操作.

步骤5. 更新环境信息. 若规划未超时, 返回步骤1; 否则, 算法结束并输出规划树.

算法2. 采样点的 K 近邻树节点搜索算法

步骤如下:

步骤1. 搜索采样点 $Rand$ 最近邻索引区域对应的 Kd 树根节点, 并二叉搜索 $Rand$ 的近似最近邻节点 $ApNearestNode$, 如下: 在 Kd 树的第 i 层比较 $Rand$ 与树节点的第 $i \% d + 1$ 维坐标的关系, 确定继续搜索树节点的左或右子树, 返回最后到达的 Kd 树叶节点, 其中 d 为空间维度, $\%$ 为取模运算符. 将 $ApNearestNode$ 加入近邻节点集合 S_{near} . 将 $Rand$ 与 $ApNearestNode$ 距离作为初始近邻球半径 R_n . 寻找与近邻球相交的索引区域对应 Kd 树根节点, 并将其加入待检测近邻节点集合 $TestNodes$.

步骤2. 若 $TestNodes$ 非空, 则执行步骤3; 否则, 执行步骤4.

步骤3. 若 $TestNodes$ 中节点 $CurrentFather$ 上的分类面与近邻球相交, 则将其存入 S_{near} 并将其左右子节点均添加至 $TestNodes$; 否则, 将与 $NewPt$ 同侧 (与 $NewPt$ 较近) 的 $CurrentFather$ 子节点添加至 $TestNodes$, 返回步骤2.

步骤4. 若搜索到的近邻节点数 N 小于 K 并且 R_n 小于其上限值 R_{limit} , 则令 $R_n = (1 + \zeta) \times R_n$, 返回步骤1; 否则, 算法结束并输出 N 、 S_{near} .

2.7 算法复杂度分析

碰撞检测是随机化方法中最耗时的部分^[17-18], 如果定义碰撞检测为基本操作, n 为算法采样次数, 则传统 RRT 方法的时间复杂度为 $O(n)$, 本文方法通过约减不合理采样空间和代价启发, 以减少碰撞检测次数, 因此在航迹搜索阶段的时间复杂度低于 RRT 方法; 在航迹优化阶段, 本文结合 RRT* 方法优化规划树拓扑结构, 文献 [19] 证明了 RRT* 方法与 RRT 方法时间复杂度相同, 因此, 本文方法的总

体时间复杂度低于 RRT 方法. DDRRT 方法同样在近邻节点搜索和碰撞调用次数间作一个折衷^[17-18], 其时间复杂度低于 RRT 方法. 文献 [19] 证明了当采样次数趋于无穷时, RRT* 方法收敛于最优解的概率趋于 1, 本文方法得到的航迹解是优化的, 而 DDRRT 方法和 RRT 方法不保证航迹解的优化性.

令规划树节点数为 m , 空间维度为 d , 近邻节点查询的基本操作为距离计算, 假设当前时刻, 本文方法通过 h 棵 Kd 树存储规划树节点, 此时建立 Kd 树的时间复杂度为 $O(hd \log(\frac{m}{h}))$, 空间复杂度为 $O(2h + dm)$, 算法查询近邻节点的时间复杂度为 $O(c^d \log(\frac{m}{h}))$, c 为与距离测度相关的常量, 当 m 远大于 h 时, 本文查询近邻节点的空间复杂度与传统 Kd 树方法近似, 为 $O(dm)$ ^[39-40].

3 仿真实验结果与分析

本文分别在不同环境下对相关算法进行 100 次蒙特卡罗仿真实验, 实验环境配置为 Intel Pentium 4, 2.5 GHz CPU, 1 GB RAM, Windows XP OS.

3.1 无人机模型与数值仿真环境

3.1.1 无人机运动模型

本文基于 Dubins 技术的无人机运动学模型为:

$$\begin{cases} \dot{x}(t) = v \cos \varphi(t) \cos \theta(t) \\ \dot{y}(t) = v \cos \varphi(t) \sin \theta(t) \\ \dot{z}(t) = v \sin \varphi(t) \end{cases}$$

其中, $x(t)$ 、 $y(t)$ 、 $z(t)$ 指无人机在 x 、 y 、 z 方向位移的时间导数或速度, v 为线速度, $\varphi(t)$ 为俯仰角, $\theta(t)$ 为转向角, $\varphi(t)$ 的时间导数 $\dot{\varphi}(t) = u_\varphi$, u_φ 为无人机俯仰控制, u_θ 为转向控制. 令 $\dot{\omega}(t) = \text{Rotate}(\dot{\theta}(t), \dot{\varphi}(t))$ 为三维空间中的无人机转向控制函数, $\dot{\omega}(t) \in [-\omega, +\omega]$, $\omega = v_{\min}/r_{\min}$, $r_{\min} = (v_{\min}^2)/(g \times (n_{y \max}^2 - 1)^{\frac{1}{2}})$, r_{\min} 为无人机最小转弯半径, $n_{y \max}$ 为无人机的最大法向过载. 当 $\varphi(t) = 0$ 时, 三维模型转换为二维模型.

无人机运动学约束包括: 1) 最小转弯半径约束 $r \geq r_{\min}$, 即无人机的转弯半径 r 大于等于 r_{\min} ; 2) 最小直行航迹段长度 (无人机最小反应距离) 约束 $s \geq s_{\min}$, 即无人机的直行航迹段长度 s 大于等于最小直行航迹段长度 s_{\min} .

3.1.2 数值仿真环境

定义算法效率指标: 1) 搜索性能指标: 航迹搜索时间 (Time); 2) 规划复杂度指标: 航迹搜索阶段结束时的规划树节点数 (Number of tree nodes, NTN) 和碰撞检测调用次数 (Number of collision detection calls, NCDC); 约减合理性指标, 反映

空间约减后规划树周围的可拓展区情况, 包括航迹搜索阶段结束时的树节点扩展成功率 (Extending success ratio, ESR), $ESR = NTN/NCDC$; 4) 近邻节点搜索算法和树节点存储结构性能指标, 包括航迹搜索阶段结束时的近邻节点搜索时间和近邻节点搜索时间与航迹搜索时间比 (Neighbors searching time ratio, NSTR), $NSTR = NST/\text{Time}$, 其中, 本文采用改进的存储结构和近邻节点搜索算法, DDRRT 方法和 RRT* 方法采用传统 Kd 树存储结构和近邻节点搜索算法, RRT 方法采用顺序存储结构和基于距离计算的近邻节点搜索方法; 5) 最终多边形航迹代价指标 (Path cost, PC), $PC = \sum_{i=1}^{n-1} C(Node_i, Node_{i+1})$, 其中, $Node_i$ 为多边形航迹内的航迹点, n 为多边形航迹点个数; 6) 算法可靠性和鲁棒性指标: 规划失败率 (Failure ratio, FR), $FR = \text{失败次数}/100$, 本文规定, 如果算法在 600s 有效时间内仍未发现连通目标点的航迹, 算法不满足时效性, 执行失败.

实验 1~4 的仿真图中带阴影的实体表示障碍物, 树状线条集为规划树, 光滑曲线为 Dubins 航迹, 图 8 中网格表示基本约减单元, 网格中的阴影表示基本约减单元与障碍物相交.

本文令 DDRRT 方法的动态域半径 $R = 10\epsilon$ (ϵ 为 RRT 步长), 此时 DDRRT 算法效率较高^[17]. 本文算法中代价模型的权值反映对航迹解不同侧面的启发式引导, 为了弱化主观因素影响, 本文采用 7 级判断矩阵标度的层次分析法, 根据经验知识和仿真任务特点定义权值参数. 定义式 (1) 中 $\alpha_1 = 0.2, \alpha_2 = \alpha_3 = 0.4$; 式 (2) 中 $\beta_1 = 0.6, \beta_2 = \beta_3 = 0.2$; 式 (3) 中 $\gamma = 0.6$; 式 (4) 中 $\omega_1 = 0.5, \omega_2 = 0.3, \omega_3 = 0.2$. 定义无人机最小转弯半径 $\rho_{\min} = 10 \text{ units}$; 最小航迹段长度 $s_{\min} = 6 \text{ units}$. 由于近邻节点搜索过程非常快, 因此, 算法 2 中的参数 ζ 值对算法效率影响有限, 取一个较小的值即可满足要求, 定义 $\zeta = 0.2$. 索引树划分参数 $m = 200$, Sc 等于 4 倍约减单元.

3.2 实验 1: 二维复杂环境下的仿真实验及参数确定

3.2.1 二维复杂环境下的航迹规划/重规划

图 8 为当算法 1 的关联代价惩罚参数 $\delta = 1.5$, 单次返回近邻节点数下限 $K = 5$ 时, 在障碍物密集、带有多个凹障碍物和多条通道、部分信息未知的环境下的航迹规划/重规划的仿真结果, 图 8 (a)、(b)、(c) 分别为环境信息已知条件下 DDRRT 算法、RRT* 算法和本文算法在航迹搜索阶段的规划树和最终 Dubins 航迹. 与 DDRRT 和 RRT* 方法相比, 本文方法能够通过较少的规划树分枝找到

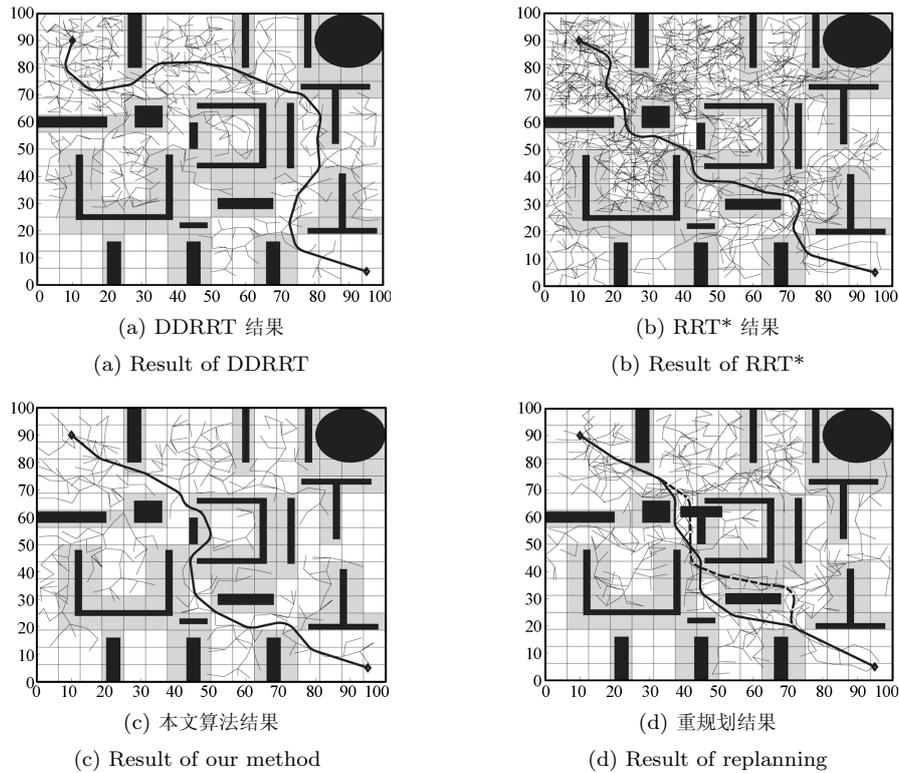


图 8 实验 1 环境下航迹搜索阶段的规划树和最终航迹

Fig. 8 The planning tree after the path searching stage and the final path of the first experiment

凹障碍物的出口,能够发现狭窄通道内的路径,说明本文方法的空间约减效果较合理.图 8(d)为部分航迹由于横坐标 45、纵坐标 62 处突然出现障碍物而失效后局部重规划的结果,图中虚线表示原航迹段,实线所示为重规划后的航迹段,重规划结果不唯一.

表 1 为本文算法、DDRRT 算法、RRT* 算法、传统 RRT 算法的主要效率指标统计结果,本文算法 Time、NTN、NCDC、PC、FR 优于 DDRRT 与 RRT 算法;ESR 指标与 DDRRT 算法相近,优于 RRT 与 RRT* 算法;NSTR 指标值略高于 DDRRT 和 RRT* 算法,低于 RRT 算法,其原因是,一方面,本文方法在规划树每次扩展过程中要返回多个近邻节点,另一方面,当树节点较少时,本文方法查找近邻节点时可能要搜索多棵 Kd 树.RRT* 算法的 PC 指标最优,但是其 Time、NTN、NCDC、ESR、FR 指标与 RRT 方法相近,由于其存储结构与 DDRRT 方法相同,因此其 NST、NSTR 指标与 DDRRT 方法相近.本文方法在航迹搜索结束后,才在剩余规划时间内结合 RRT* 方法优化航迹,因此,在后续实验中不对 RRT* 方法效率进行讨论.

3.2.2 参数选择对算法效率的影响分析

图 9 为 $\delta = 1.2$ 时,在实验 1 环境下按照不同 K 值分别进行 100 次实验得到的算法效率指标值随 K 变化的情况,同时统计出 $K = 5$ 时算法效率指标

值随 δ 变化的情况,如图 10 所示.选择算法性能较优时 K 与 δ 值,对本文算法和传统算法效率进行比较. FR 与 K 、 δ 关系不大,故未列出.

表 1 二维复杂环境下多种算法规划效率指标对比
Table 1 Comparison on efficiency indicators of multi-algorithm path planning in two dimensional complex environments

算法	Time (s)	NTN	NCDC	ESR	NST (s)	NSTR	PC	FR
本文算法	4.88	405	920	0.4402	0.3228	0.0661	135.86	0.01
DDRRT	10.20	783	1552	0.5045	0.5521	0.0541	173.25	0.02
RRT*	25.63	1265	3495	0.3619	0.9825	0.0383	130.21	0.06
RRT	22.84	963	3119	0.3088	2.8633	0.1254	165.36	0.05

本文算法中 K 与规划树拓扑优化性相关, K 越大,则树结构越优化,但搜索算法计算量变大.当 $K = \infty$ 时,本文算法求解过程与 RRT* 方法相近,但复杂度低于 RRT* 方法;当 $K = 1$ 时,其求解过程与 DDRRT 方法相同,但由于其缓解了过度约减问题,所以优于 DDRRT 方法, K 可取较小整数值.由于空间约减算法无法保证被约减区域所有部分都是不合理的,对其直接约减可能会造成过度约减问题,因此本文提出基于代价的约减方法.当 $\delta = 0$ 时,本文算法效率向 RRT 方法退化;当 $\delta = \infty$ 时,本文算法约减确定性与 DDRRT 方法相同,同样由于其缓解了过度约减问题,因此优于 DDRRT 方法.

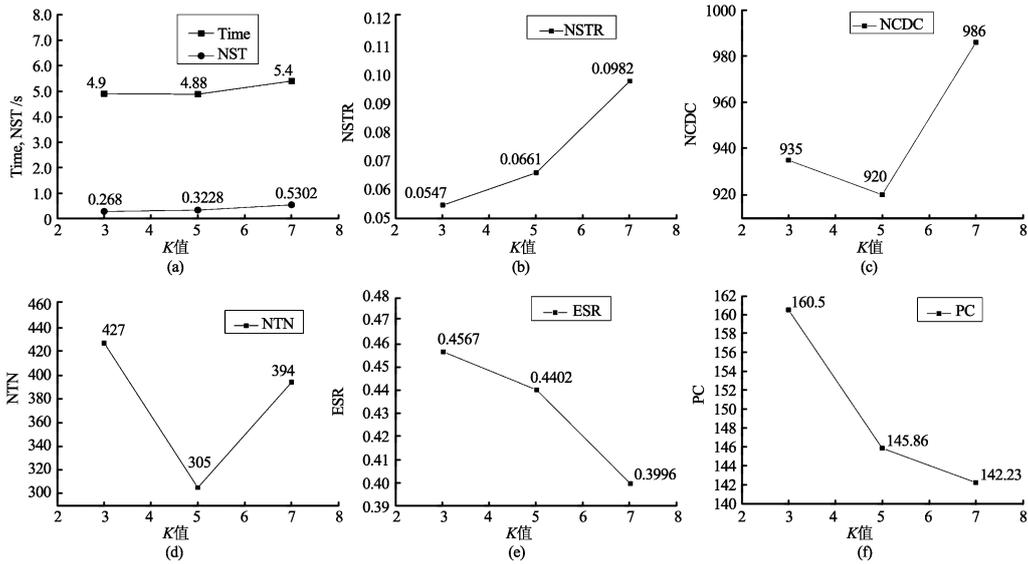


图9 K 值大小对算法效率影响统计图

Fig. 9 The statistical diagram of the influence of the parameter K to the performance of our method

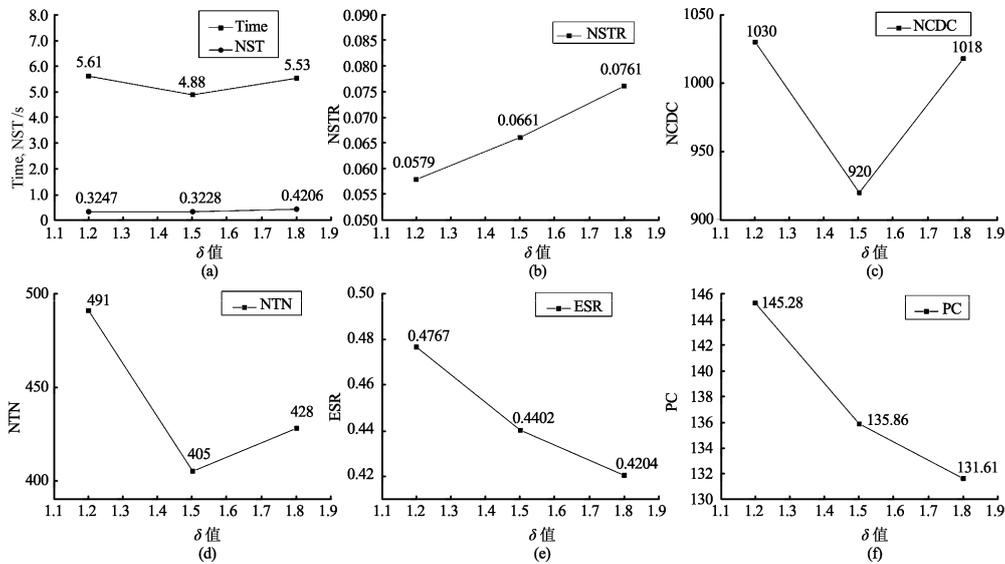


图10 关联代价惩罚参数 δ 对算法效率影响统计图

Fig. 10 The statistical diagram of the influence of the correlation cost punishment parameter δ to the performance of our method

较小的 δ 值可以满足本文算法要求。

$K = 3, 5, 7$ 时, 单次返回的近邻节点数 N 平均值分别为 4.43、6.37、9.05。图 9(a) 为 Time、NST 随 K 值变化情况; 图 9(b)~(f) 为 NSTR、NCDC、NTN、ESR、PC 随 K 值变化情况, 规划树更倾向于在图 8 中的通道区域内搜索代价更小的航迹, 使得 ESR 和 PC 降低。适当的 K 值可以启发规划过程, 有利于降低 Time、NCDC、NTN; 但是, 过大的 K 值使得算法过度倾向于在通道内搜索航迹, 反而增加算法的规则难度, 增加

Time、NCDC、NTN。结果表明, K 对算法效率影响较小, $K = 5$ 为算法效率均衡点。

图 10(a) 为 Time 与 NST 随 δ 变化情况, 图 10(b)~(f) 分别为 NSTR、NCDC、NTN、ESR、PC 随 δ 变化情况, δ 值越大, 算法对空间约减越严格, 算法更倾向于在图 8 中的通道区域内搜索代价更小的航迹, ESR、PC 值越低, 同时规划树修正节点越多, NST、NSTR 越大。适当增加 δ 值, 可以启发规划过程, 有利于降低 Time、NCDC 和 NTN。但是, 过大的 δ 使得空间约减过于严格, 反而增加

Time、NCDC 和 NTN. 结果表明, δ 对算法效率影响较小, $\delta = 1.5$ 为算法效率均衡点. 在其他实验环境下可以统计出类似结果, 因此在后续实验中, 令 $K = 5$, $\delta = 1.5$.

3.3 实验 2: 多约束条件/迷宫环境下的航迹规划

如图 11 为本文算法在 Labyrinth 迷宫环境下的规划结果, Labyrinth 环境由狭窄通道、凹障碍物和陷阱构成, 实验目的是测试本文算法在多约束复杂环境下的规划效率和鲁棒性.

实验 2 的统计结果如表 2 所示, 本文算法在 Time、NTN、NCDC、ESR、PC 指标上均优于 DDRRT 算法, NSTR 指标略高于 DDRRT 算法, 原因与实验 1 类似. 本文算法和 DDRRT 算法与 RRT 算法相比, 效率提升较大. 本文方法的 FR 指标值与 DDRRT 算法相近, 低于 RRT 算法.

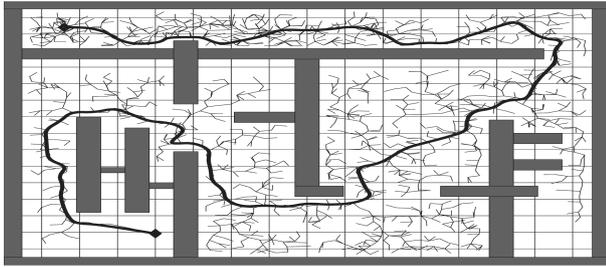


图 11 实验 2 环境下航迹搜索阶段的规划树和最终航迹
Fig. 11 The planning tree after the path searching stage and the final path of the second experiment

表 2 多约束 Labyrinth 环境下多种算法规划效率指标对比
Table 2 Comparison on efficiency indicators of multi-algorithm path planning in multi-constraint of labyrinth environment

算法	Time (s)	NTN	NCDC	ESR	NST (s)	NSTR	PC	FR
本文算法	39.43	998	4 353	0.2293	2.8325	0.0718	306.11	0.01
DDRRT	46.62	1 083	4 864	0.2227	2.6137	0.0561	389.13	0.02
RRT	169.27	1 279	10 028	0.1275	15.6218	0.0923	383.22	0.08

3.4 实验 3: 连续通过两条狭窄通道的三维航迹规划

为了测试算法在通道内的寻路能力, 本文设计了图 12 所示的实验 3 环境, 航迹起点位于左下角, 终点位于右下角, 规划难点在于航迹解必须通过障碍物上的两个通道.

实验 3 结果数据统计如表 3 所示, 从表中可以看出, 本文算法和 DDRRT 算法解决通道内寻路问题的效率远高于传统 RRT 算法, 本文方法效率高于 DDRRT 方法. 本文方法的 Time、NTN、NCDC 均小于 DDRRT 方法, 说明本文方法对采样空间约减更加合理. ESR 略低于 DDRRT 方法和 RRT 方法, 这是因为本文算法倾向于搜索代价更低路径, 在

实验 3 环境下, 碰撞发生的概率更大. 本文算法的 NSTR 指标值与 DDRRT 算法相近, 低于 RRT 算法. 与 DDRRT 方法和 RRT 方法相比, 本文方法生成的航迹代价更低. 本文方法的 FR 指标值与 DDRRT 算法相近, 低于 RRT 算法.

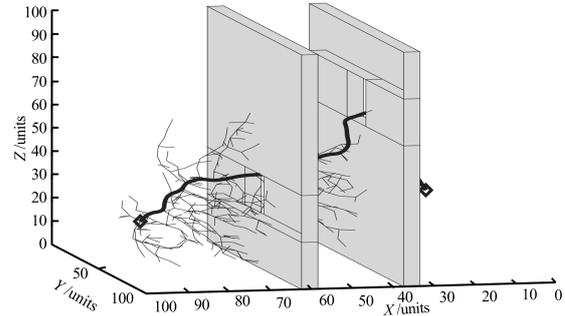


图 12 实验 3 环境下航迹搜索阶段的规划树和最终航迹
Fig. 12 The planning tree after the path searching stage and the final path of the third experiment

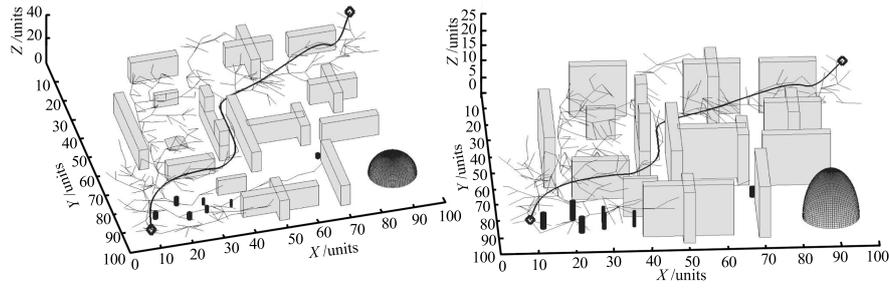
表 3 三维多约束通道环境下多种算法规划效率指标对比
Table 3 Comparison on efficiency indicators of multi-algorithm path planning in three dimensional passage environment

算法	Time (s)	NTN	NCDC	ESR	NST (s)	NSTR	PC	FR
本文算法	46.86	1 098	3 240	0.3129	1.7422	0.0372	101.680	0.02
DDRRT	75.07	1 841	4 063	0.4531	2.3326	0.0311	180.264	0.03
RRT	213.26	9 143	22 480	0.4067	14.7281	0.0691	211.343	0.10

3.5 实验 4: 不同规模的三维低空环境下航迹规划

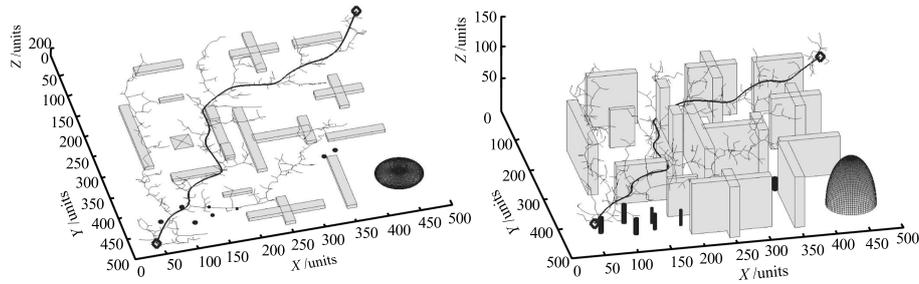
图 13 中的半球形障碍物表示辐射半径无穷的辐射源, 其可视域为辐射暴露区, 为了躲避辐射威胁, 算法尽量在由地形及建筑物产生的辐射遮蔽区内规划航迹, 而这对算法在通道内的寻路能力要求较高. 为了考察算法效率随规划空间规模增长的变化情况, 实验 4 分别在规划范围为 $100 \times 100 \times 50$ (图 13 (a))、 $500 \times 500 \times 150$ (图 13 (b)) 和 $1000 \times 1000 \times 300$ (图 13 (c)) 的环境下进行仿真实验. 其中, 图 13 (b) 中障碍物的长、宽为图 13 (a) 的 5 倍, 图 13 (c) 中障碍物为图 13 (a) 的 10 倍. 图 13 (b) 中障碍物高为图 13 (a) 的 3 倍, 图 13 (c) 中障碍物高为图 13 (a) 的 6 倍, 障碍物的相对位置不变, 规划树步长和其他参数值不变.

实验结果统计如表 4 所示, 本文算法在 Time、NTN、NCDC、PC 指标上均优于 DDRRT 算法, 相比 RRT 算法有大幅度提高. 当规划空间范围为 $1000 \times 1000 \times 300$ 时, 本文方法的 Time 相比 DDRRT 方法减少 24.55% 左右, 相比 RRT 方法减少 80% 以上, 说明本文方法更适合处理大范围空间内的航迹规划问题. 本文方法的 ESR 略高于 DDRRT 方法, 高于 RRT 方法. 随着空间扩大, 本



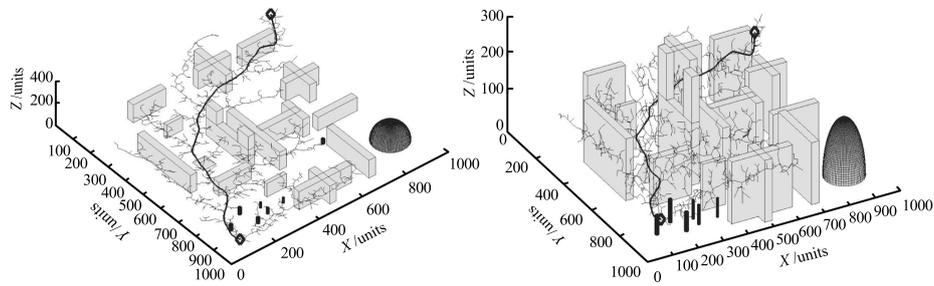
(a) 100×100×50 空间内算法仿真结果

(a) The simulation results in 100×100×50 space



(b) 500×500×150 空间内算法仿真结果

(b) The simulation results in 500×500×150 space



(c) 1 000×1 000×300 空间内算法仿真结果

(c) The simulation results in 1 000×1 000×300 space

图 13 实验 4 环境下航迹搜索阶段的规划树和最终航迹

Fig. 13 The planning tree after the path searching stage and the final path of the fourth experiment

表 4 三种不同规模的多约束复杂环境下多种算法规划效率指标对比

Table 4 Comparison on efficiency indicators of multi-algorithm path planning in three scales of multi-constraint complex environment

指标	图 13 (a) 环境			图 13 (b) 环境			图 13 (c) 环境		
	本文算法	DDRRT	RRT	本文算法	DDRRT	RRT	本文算法	DDRRT	RRT
Time (s)	3.66	3.88	12.58	9.05	10.02	30.84	22.90	30.35	162.00
NTN	347	364	516	699	767	1 128	1 045	1 357	6 243
NCDC	771	892	1 956	1 487	1 790	3 911	2 187	3 145	24 011
ESR	0.450	0.408	0.264	0.470	0.429	0.288	0.478	0.432	0.260
NST (s)	0.311	0.251	2.019	0.821	0.833	6.218	1.595	2.538	34.732
NSTR	0.085	0.065	0.161	0.091	0.083	0.202	0.070	0.084	0.214
PC	124.5	195.1	182.0	705.3	902.2	1 205.0	1 385.0	1 793.8	2 195.0
FR	0.00	0.01	0.03	0.01	0.03	0.04	0.02	0.01	0.12

文算法 ESR 变化较稳定, 说明本文方法对采样空间约减较合理; 同时, 本文方法与 DDRRT 算法的 FR 变化较小, 而 RRT 算法 FR 增加较为明显. 当规划树规模增加时, 本文算法的 NSTR 指标逐渐优于 DDRRT 方法, 说明本文的近邻节点搜索方法和树节点存储结构适合处理较大规模规划树上的近邻节点搜索问题, 本文算法生成航迹 PC 值最小.

综合上述实验可知, 在范围较小、障碍物较简单的环境下, 本文方法效率略高于 DDRRT 方法; 随环境范围增加, 本文方法比 DDRRT 方法效率更高, 效率指标变化更平稳, 在通道内的寻路能力更强. 本文算法可扩展应用到机器人路径规划等多个领域.

4 结论

针对 DDRRT 方法对采样空间过度约减的问题, 本文提出一种基于约减域逐步构造的无人机在线航迹规划算法, 本文的主要创新点如下:

- 1) 建立代价模型并根据环境特征和规划树生长情况逐步构造动态采样边界和约减域;
- 2) 结合 RRT* 方法在线优化航迹;
- 3) 提出航迹点的多 Kd 树存储结构和近邻节点搜索算法.

实验表明, 本文方法能够在保证采样空间约减合理性的同时, 提高算法的规划效率和在通道区域的寻路能力以及在空间规模增加时的鲁棒性. 本文的下一步研究方向是解决存在动态威胁的环境下的航迹规划问题.

References

- 1 Ye Wen, Fan Hong-Da, Zhu Ai-Hong. *Mission Planning for Unmanned Aerial Vehicles*. Beijing: National Defense Industry Press, 2011. 1–201
(叶文, 范洪达, 朱爱红. 无人飞行器任务规划. 北京: 国防工业出版社, 2011. 1–201)
- 2 Frazzoli E, Dahleh M A, Feron E. Real-time motion planning for agile autonomous vehicles. *Journal of Guidance Control and Dynamics*, 2002, **25**(1): 116–129
- 3 Kim Y, Gu D W, Postlethwaite I. Real-time path planning with limited information for autonomous unmanned air vehicles. *Automatica*, 2008, **44**(3): 696–712
- 4 Zhao Ming, Su Xiao-Hong, Ma Pei-Jun, Zhao Ling-Ling. A unified modeling method of UAVs cooperative target assignment by complex multi-constraints conditions. *Acta Automatica Sinica*, 2012, **38**(12): 2038–2048
(赵明, 苏小红, 马培军, 赵玲玲. 复杂多约束 UAVs 协同目标分配的一种统一建模方法. 自动化学报, 2012, **38**(12): 2038–2048)
- 5 Tsourdos A, White B A, Shanmugavel M. *Cooperative Path Planning of Unmanned Aerial Vehicles*. West Sussex: Wiley & Sons, 2011. 1–185
- 6 LaValle S M. *Planning Algorithms*. Cambridge: Cambridge University Press, 2006. 482–580
- 7 Yuan Kui, Li Yuan, Fang Li-Xin. Multiple mobile robot systems: a survey of recent work. *Acta Automatica Sinica*, 2007, **33**(8): 785–794
(原魁, 李园, 房立新. 多移动机器人系统研究发展近况. 自动化学报, 2007, **33**(8): 785–794)
- 8 Zhu Yi, Zhang Tao, Song Jing-Yan. Study on the local minima problem of path planning using potential field method in unknown environments. *Acta Automatica Sinica*, 2010, **36**(8): 1122–1130
(朱毅, 张涛, 宋靖雁. 未知环境下势场法路径规划的局部极小问题研究. 自动化学报, 2010, **36**(8): 1122–1130)
- 9 Zhang Chun-Gang, Xi Yu-Geng. A real-time path planning method for mobile robot avoiding oscillation and dead circulation. *Acta Automatica Sinica*, 2003, **29**(2): 197–205
(张纯刚, 席裕庚. 一种克服振荡与死循环的机器人实时路径规划方法. 自动化学报, 2003, **29**(2): 197–205)
- 10 LaValle S M, Kuffner J J. Randomized kinodynamic planning. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. Detroit, USA: IEEE, 1999, **1**: 473–479
- 11 Cheng Peng, Shen Zuo-Jun, LaValle S M. RRT-based trajectory design for autonomous automobiles and spacecraft. *Archives of Control Sciences*, 2001, **11**(3): 167–194
- 12 Toda Y, Kubota N. Path planning using multi-resolution map for a mobile robot. In: *Proceedings of the Society of Instrument and Control Engineers Annual Conference*. Tokyo, Japan: IEEE, 2011. 1276–1281
- 13 Poppinga J, Birk A, Pathak, K, Vaskevicius N. Fast 6-DOF path planning for Autonomous Underwater Vehicles (AUV) based on 3D plane mapping. In: *Proceedings of IEEE International Symposium on Safety, Security, and Rescue Robotics*. Kyoto, Japan: IEEE, 2011. 345–350
- 14 Hsu D, Kindel R, Latombe J C, Rock S. Randomized kinodynamic motion planning with moving obstacles. *The International Journal of Robotics Research*, 2002, **21**(3): 233–255
- 15 Zucker M, Kuffner J, Bagnell J A. Adaptive workspace biasing for sampling-based planners. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. Kobe, Japan: IEEE, 2008. 3757–3762
- 16 Rodriguez S, Tang X Y, Lien J M, Amato N M. An obstacle-based rapidly-exploring random tree. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. Orlando, USA: IEEE, 2006. 895–900
- 17 Yershova A, Jaillet L, Siméon T, LaValle S M. Dynamic-domain RRTs: efficient exploration by controlling the sampling domain. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. Barcelona, Spain: IEEE, 2005. 3856–3861
- 18 Jaillet L, Yershova A, LaValle S M, Siméon T. Adaptive tuning of the sampling domain for dynamic-domain RRTs. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. Edmonton, Canada: IEEE, 2005. 2851–2856
- 19 Montemanni R, Gambardella L M, Donati A V. A branch and bound algorithm for the robust shortest path problem with interval data. *Operations Research Letters*, 2004, **32**(3): 225–232
- 20 Karaman S, Frazzoli E. Optimal kinodynamic motion planning using incremental sampling-based methods. In: *Proceedings of the IEEE Conference on Decision and Control*. Georgia, USA: IEEE Control Systems Society, 2010. 7681–7687
- 21 Urmson C, Simmons R. Approaches for heuristically biasing RRT growth. In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*. Las Vegas, USA: IEEE, 2003, **2**: 1178–1183

- 22 Lee J, Pippin C, Balch T. Cost based planning with RRT in outdoor environments. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Nice, France: IEEE, 2008. 684–689
- 23 Jaillet L, Cortés J, Siméon T. Transition-based path planning on configuration-space costmaps. *IEEE Transactions on Robotics*, 2010, **26**(4): 635–646
- 24 Ferguson D, Stentz A. Anytime RRTs. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Beijing, China: IEEE, 2006. 5369–5375
- 25 Yang K J, Sukkarieh S. 3D smooth path planning for a UAV in cluttered natural environments. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robot and Systems. Nice, France: IEEE, 2008. 794–800
- 26 Karaman S, Walter M R, Perez A, Frazzoli E, Teller S. Anytime motion planning using the RRT. In: Proceedings of the IEEE International Conference on Robotics and Automation. Shanghai, China: IEEE, 2011. 1478–1483
- 27 Lindemann S R, LaValle S M. Incrementally reducing dispersion by increasing Voronoi bias in RRTs. In: Proceedings of the IEEE International Conference on Robotics and Automation. New Orleans, USA: IEEE, 2004, **4**: 3251–3257
- 28 Lindemann S R, LaValle S M. Steps toward derandomizing RRTs. In: Proceedings of the Fourth International Workshop on Robot Motion and Control. Poznan, Poland: IEEE, 2004. 271–277
- 29 Bruce J, Veloso M. Real-time randomized path planning for robot navigation. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Lausanne, Switzerland: IEEE, 2002, **3**: 2383–2388
- 30 Shi B F, Cheng P, Cheng N. 3D flight path planning based on RRTs for RNP requirements. In: International Conference on Information and Automation. Shenyang, China: IEEE, 2012. 51–56
- 31 Yang G, Kapila V. Optimal path planning for unmanned air vehicles with kinematic and tactical constraints. In: Proceedings of the 41st IEEE Conference on Decision and Control. Las Vegas, USA: IEEE, 2002, **2**: 1301–1306
- 32 Guo Suo-Feng, Shen Gong-Zhang, Wu Cheng-Fu. *Advanced Flight Control System*. Beijing: National Defense Industry Press, 2003. 180–185
(郭锁凤, 申功璋, 吴成富. 先进飞行控制系统. 北京: 国防工业出版社, 2003. 180–185)
- 33 Shanmugavel M, Tsourdos A, Zbikowski R, White B A. 3D dubins sets based on coordinated path planning for swarm of UAVs. In: Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit. Colorado, USA: AIAA, 2006. 21–24
- 34 Ardiyanto I, Miura J. Real-time navigation using randomized kinodynamic planning with arrival time field. *Robotics and Autonomous Systems*, 2012, **60**(12): 1579–1591
- 35 Lopez T, Lamarche F, Li T Y. Space-time planning in changing environments: using dynamic objects for accessibility [Online], available: <http://www.wileyonlinelibrary.com>, March 21, 2012
- 36 Aoude G S, How J P, Garcia I M. Two-stage path planning approach for solving multiple spacecraft reconfiguration maneuvers. *The Journal of the Astronautical Sciences*, 2008, **56**(4): 515–544
- 37 Scheuer A, Fraichard T. Continuous-curvature path planning for car-like vehicles. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Grenoble, France: IEEE. 1997. **2**: 997–1003

- 38 Garcia I, How J P. Trajectory optimization for satellite re-configuration maneuvers with position and attitude constraints. In: Proceedings of the 2005 American Control Conference. Michigan, USA: American Automatic Control Council. 2005. **2**: 889–894
- 39 Yershova A, LaValle S M. Improving motion-planning algorithms by efficient nearest-neighbor searching. *IEEE Transactions on Robotics*, 2007, **23**(1): 151–157
- 40 Atramentov A, LaValle S M. Efficient nearest neighbor searching for motion planning. In: Proceedings of the IEEE International Conference on Robotics and Automation. Washington D C. USA: IEEE, 2002. **1**: 632–637



温乃峰 哈尔滨工业大学计算机科学与技术学院博士研究生. 主要研究方向为信息融合, 无人机导航与在线航迹规划, 多无人机协同. 本文通信作者.

E-mail: wennaifeng@126.com

(**WEN Nai-Feng** Ph. D. candidate at the School of Computer Science and Technology, Harbin Institute of Tech-

nology. His research interest covers information fusion, UAV navigation and online path planning, and UAVs co-operation. Corresponding author of this paper.)



苏小红 博士, 哈尔滨工业大学计算机科学与技术学院教授. 主要研究方向为信息融合, 软件工程和神经网络.

E-mail: sxh@hit.edu.cn

(**SU Xiao-Hong** Ph. D., professor at the School of Computer Science and Technology, Harbin Institute of Tech-

nology. Her research interest covers information fusion, software engineering and neural networks.)



马培军 博士, 哈尔滨工业大学计算机科学与技术学院教授. 主要研究方向为信息融合, 软件工程和模式识别.

E-mail: ma@hit.edu.cn

(**MA Pei-Jun** Ph. D., professor at the School of Computer Science and Technology, Harbin Institute of Tech-

nology. His research interest covers information fusion, software engineering and pattern recognition.)



赵玲玲 博士, 哈尔滨工业大学计算机科学与技术学院讲师. 主要研究方向为信息融合和目标跟踪.

E-mail: zhaolinglinghit@126.com

(**ZHAO Ling-Ling** Ph. D., lecturer at the School of Computer Science and Technology, Harbin Institute of Tech-

nology. Her research interest covers information fusion and target tracking.)