

DOI: 10.13382/j.jemi.B2003353

# 激光雷达无人飞行器自主着陆选址方法

邢 闻<sup>1</sup> 朱利丰<sup>1</sup> 李 聪<sup>2</sup> 徐雅芬<sup>2</sup> 赵国普<sup>3</sup> 宋爱国<sup>1</sup>(1. 东南大学 仪器科学与工程学院 南京 210000; 2. 上海航空航天控制技术研究所 上海 200000;  
3. 江苏四明工程机械有限公司 扬州 225000)

**摘要:**无人飞行器在陌生场景中的自主路径规划与着陆工作一直是相关领域研究的重点,提出了一种基于激光雷达采集到的点云数据来给飞行器推荐最优着陆地址的方法。该方法通过飞行器位姿信息辅助以及选址窗口维护等手段校正并扩充了雷达采集到的原始点云数据,之后再利用改进的RANSAC算法对生成的候选平面进行评估选择,最后反馈最优着陆点的位置坐标信息。仿真环境下的实验结果表明该方法运算结果稳定准确,且运算速度满足飞行器实时工作需求,同时计算所需空间时间开销较小,能够符合在现实场景中正常工作的标准。

**关键词:**激光雷达;无人飞行器;RANSAC;着陆点选址

**中图分类号:** TN953 **文献标识码:** A **国家标准学科分类代码:** 460.40

## Landing site recommendation for unmanned aerial vehicles based on lidar data

Xing Wen<sup>1</sup> Zhu Lifeng<sup>1</sup> Li Cong<sup>2</sup> Xu Yafen<sup>2</sup> Zhao Guopu<sup>3</sup> Song Aiguo<sup>1</sup>

(1. School of Instrument Science and Engineering, Southeast University, Nanjing 210000, China; 2. Shanghai Aerospace Control Technology Institute, Shanghai 200000, China; 3. Jiangsu Siming Engineering Machinery Co., Ltd., Yangzhou 225000, China)

**Abstract:** The autonomous path planning and landing of unmanned aerial vehicle in open scenes has been the focus of research in related fields. A method based on the point cloud data collected by the lidar to recommend the optimal landing address for the UAV is proposed. This method corrects and expands the original point cloud data by using the UAV pose information and maintaining the site selection window, then the RANSAC algorithm is improved to evaluate and select the generated candidate planes, and finally output the coordinate information of the optimal landing site. The experiments in the simulation environment show that the results of this method are stable and accurate, and the computational speed satisfies the requirements of the UAV to work in real time. At the same time, the space and time cost of this method is light, which meets the standard of the operation in real scenes.

**Keywords:** lidar; unmanned aerial vehicles; RANSAC; landing site selection

## 0 引言

定位与环境建模是无人飞行器自动控制的重要技术环节之一,利用单目视觉设备如机载相机、激光雷达等传感器可以有效采集视觉信息,再使用计算机资源对收集到的信息进行分析处理就可以帮助无人飞行器进行轨迹估计、三维环境建模等工作<sup>[1-2]</sup>。激光雷达(lidar)凭借其大范围深度信息的可靠感知等优势,已在无人系统环境感知得到了诸多应用<sup>[3-4]</sup>。然而,各种视觉信息采集设

备高昂的硬件成本以及飞行器实际飞行过程中由于本身位姿变化和激光雷达硬件误差可能导致的输入点云扭曲、缺失以及噪点<sup>[5-6]</sup>等问题一直是无人飞行器自主飞行工作中面临的主要困难。近年来随着激光雷达的硬件成本不断降低,越来越多的研究人员也会选择通过分析点云数据来控制飞行器实时调整飞行方向与姿态以完成各项工作。其中主流方法主要有3类。

1) 基于激光雷达,纯粹分析全局点云数据中所包含的地形几何信息来完成任務。Scherer等<sup>[7]</sup>将地形光滑度函数与着陆点优化函数相结合进而分两步选出环境中

最适合飞行器着陆的区域,在此基础上辅助以机载 RGB 相机的图像信息则能够更好的确定最优着陆点,缩小候选范围;章志诚等<sup>[8]</sup>通过改进传统势场法实现了飞行过程中的实时避障工作。这类方法无需对整个场景进行重建,因此可以在大多数陌生复杂地形条件下正常工作,但十分依赖输入点云的几何特征,并且大多数时候输入的是已知的全局点云数据,当面对单帧时刻激光雷达采集到的条带状点云数据时表现往往不尽如人意。

2) 借助同步定位与构图<sup>[9]</sup>技术实现飞行器自主飞行也是当前的一大主流研究方向。孔天恒等<sup>[10]</sup>提出了一种基于扩展卡尔曼滤波(EKF)<sup>[11]</sup>技术实现飞行器的定位与构图,而石鹏等<sup>[12]</sup>则在经典同步定位与构图框架中引入了感知环境突变的检测方法,加强了惯性辅助系统与算法的组合程度。这类方法由于惯性系统的引入往往稳定性出色,输出结果准确性较高,但是数据非流式,随着飞行时间的增加,累积下来的视觉信息往往会增大计算机内存压力并降低程序运算速度。

3) 随着机器学习研究在人工智能领域的兴起,不少研究人员也将神经网络引入了无人飞行器的飞行工作中。Badino 等<sup>[13]</sup>将体积占用图与 3D 神经网络(CNN)耦合从而训练出高效准确的飞行器着陆点检测系统;为了防止飞行器在室内飞行时与各种物件产生碰撞摩擦,Gandhi 等<sup>[14]</sup>在仿真环境中模拟各类碰撞情况,整理出了庞大的飞行器碰撞数据集,并利用卷积神经网络加以训练从而保证了飞行器在发生碰撞前自主避开障碍物。机器学习的方法凭借其在训练后输出结果极高的准确性与稳定性优势明显,但在前期准备阶段对输入的训练数据要求较高,且应用场景往往受限于训练场景,在陌生环境下该方法的结果可能会出现较大偏差。

本文提出了一种基于机载激光雷达,通过分析流式输入点云数据检测飞行器下方环境中存在的最优着陆区域的方法。该方法可以辅助无人飞行器实时进行自主着陆地点的选择。考虑到飞行器飞行过程中由于自身姿态变换以及雷达硬件误差可能导致的输入点云倾斜扭曲以及缺失等问题,本文的程序在每一时刻进行平面检测前将会依据飞行器姿态信息对输入数据进行预处理,校正原始点云并对条带状的点云数据进行记录补充从而保证输出结果满足飞行器着陆要求。

在对点云数据进行预处理操作后,本文采用随机采样一致性(RANSAC)方法<sup>[15-16]</sup>随机采样并生成候选平面,并在此基础上加以改进,通过打分投票手段从所有候选平面中得到最优着陆区域。与传统 RANSAC 平面检测<sup>[17-18]</sup>方法不同的是,传统方法仅将点云中每个点与随机生成的平面进行匹配,最后将所有点划分到各自相对应的平面中并归为一类,因此该方法大多只应用于平面分割与识别任务,而当飞行器在高空飞行时采集到的地

面点云数据往往存在较多凹凸起伏,并且不同平面之间的面积以及光滑度也会有较大差异,此时单纯对平面进行识别与分割无法选出一块最优的着陆候选地址,同时也会因为地形不平坦导致分割出的结果零碎不利于飞行器的自主着陆,因此本文则在 RANSAC 方法的基础上增加对生成平面的打分与选择操作得到飞行器下方最适合着陆的一块区域,实验结果表明多次随机检测出的平面各项指标一致性较高。

## 1 飞行器自主选址方案

通过机载激光雷达与定位系统可以有效实时获取飞行器下方的地形信息<sup>[19]</sup>,本文选取动态旋转式激光雷达作为点云采集设备,雷达为纵向固定于飞行器正下方,即飞行器飞行时的线速度向量正好位于雷达在采集地形信息时激光束旋转构成的平面上,如图 1 所示。由于该装配方式的影响,雷达的视野范围主要受不同激光束之间的最大夹角限制,因此采集到的原始地形点云会呈条带状,且距离雷达越远的地方返回的点云信息越稀疏。为了避免激光束 360° 旋转时,雷达正上方即飞行器产生的点云以及雷达下方距离飞行器过远区域信息量较少点云造成的影响,本文选择仅输入雷达正下方 160° 范围内的点云数据。

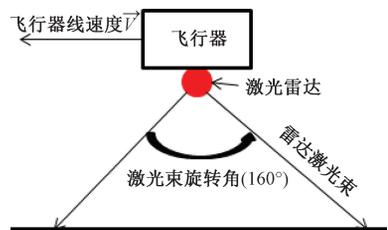


图 1 雷达装配示意图

Fig. 1 Lidar installation diagram

无人飞行器自主选址方案系统框架如图 2 所示,主要由 3 个部分组成,1) 输入点云预处理;2) 地形点云数据维护管理;3) 平面检测与着陆点推荐跟踪算法。

其中点云预处理主要为利用飞行器位姿对原始点云进行矫正的工作,而地形点云数据推测则是通过指定区域内点云窗口维护来实现,平面检测算法本文选择 RANSAC 算法。该方案优点如下:1) 运算速度快,可以保证飞行器在较高飞行速度下实时选出最合适的着陆地点;2) 适用范围广,由于是基于流式输入数据以及几何特征分析,可以在各类陌生场景中正常工作;3) 程序框架轻便,计算过程对硬件设备要求不高,计算方法通用性强便于移植其他各类无人平台辅助工作。

在实验中选取硬件装配方式为雷达激光束扫描方向沿飞行器前进方向摆放,即雷达坐标系中  $z$  轴与飞行器

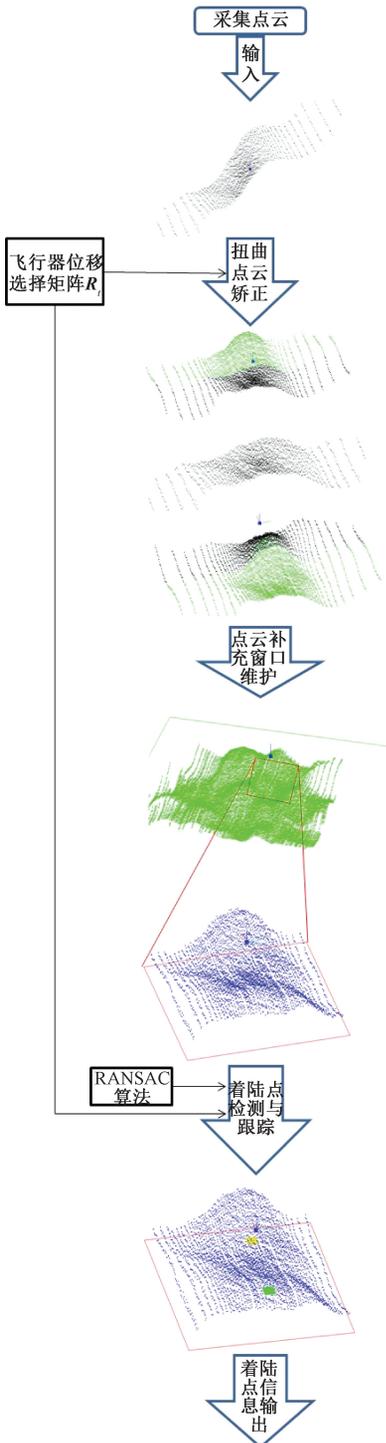


图 2 自主选址框架

Fig. 2 Framework of autonomous site selection

线速度方向垂直,当选取激光束扫描方向与飞行器前进方向垂直,即雷达坐标系中  $z$  轴与飞行器线速度方向共面的装配方式时,随着飞行的飞行移动,由于点云数据仍然可以进行有效维护管理,因此本文提出的方法依旧可行。

### 1.1 点云预处理方法

为了保证输出的选址结果符合实际场景下平面特征需求,需要在进行平面检测前对输入点云进行预处理操作加以校正扩充点云信息提高结果准确度。

本文将输入点云预处理过程分为两步,首先第 1 步根据飞行器的姿态速度信息对当前时刻采集到的新点云进行校正。其中姿态速度信息可由 INS 辅助以 GPS 系统获得<sup>[20]</sup>。设飞行器在第  $t$  帧时飞行的位移线速度与姿态角速度构成位移旋转矩阵为  $R_t$  :

$$R_t = \begin{bmatrix} R_{tx}R_{ty}R_{tz} & V_t \\ 0 & I \end{bmatrix} \quad (1)$$

式中:  $R_{tx}$ 、 $R_{ty}$ 、 $R_{tz}$  分别为飞行器沿  $x$ 、 $y$ 、 $z$  三个方向的姿态角速度分量矩阵。而  $V_t$  为飞行器的位移线速度向量,其表达形式为  $V_t^T = \{v_x, v_y, v_z\}$ , 其中  $v_x$ 、 $v_y$ 、 $v_z$  分别为飞行器沿  $x$ 、 $y$ 、 $z$  三个方向的线速度分量。

由于雷达扫描一圈用时很短,本文将每一时间帧内飞行器的运动视为匀速运动,在采样过程中输入原始点云  $P$  中的每一个点  $p \in P$  得到的均为雷达坐标系下的坐标,飞行器的位姿变化将使得世界坐标系下的标准点云  $P'$  产生一致的变换,记  $p' \in P'$ 。即:

$$p = R_t p' \quad (2)$$

考虑到雷达扫描频率,在同一帧中当雷达的激光束旋转到不同角度时,采集到的点云之间会存在一定的时间间隔,从而导致不同线的点云之间产生扭曲飘移,因此为了克服所带来的影响,本文将式(2)加以改进。

$$p = \frac{n}{f} R_t p' \quad (3)$$

其中,  $n$  为每一个点在该帧中对应的激光束的采样次数,  $f$  为雷达采样频率,因此为了得到标准世界坐标系下环境点云的坐标,只需将飞行器位移旋转变化矩阵求逆再根据式(4)即可求解出来。

$$p' = \frac{f}{n} R_t^{-1} p \quad (4)$$

点云矫正处理前输入的点云与矫正处理后点云的对比如图 3 所示,可以看出输入的原始点云存在明显的倾斜与歪曲,而点云矫正可以较好地将点云数据纠正,使其符合真实环境场景。

在飞行器实际工作过程中,每一帧雷达采集得到的点云数据仅为飞行器下方一块带状区域的环境点云,同时由于雷达不同激光束之间存在的角度差,因此返回的点云之间也会产生一定的空隙,且距离雷达中心越远,点云空隙越大,如图 3 圈出范围。毫无疑问这些因素会致使输入的视觉信产生缺失,不利于后续平面检测以及跟踪算法的实现。

所以本文点云预处理的第 2 步操作就是对当前帧采集到的点云进行补充,完善视觉信息,点云信息补充具体

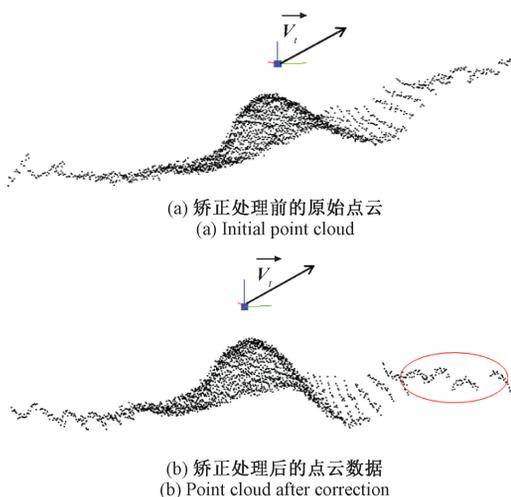


图 3 矫正处理前后点云对比

Fig. 3 Comparison of point cloud before and after correction

实施步骤如下。

1) 规定飞行器正下方  $160\text{ m} \times 160\text{ m}$  的矩形区域为点云维护窗口, 记为  $W_1$ , 同时将  $W_1$  内部区域均匀划分为  $1\ 600 \times 1\ 600$  的网格, 并为每个单独的网格标上序号添加标记位  $flag$ , 记网格集合为  $M$ 。

2) 读入初始  $t_0$  时刻雷达采集到的原始点云集合  $P_0$ , 计算  $P_0$  中哪些点落在  $M$  内部并根据每个点的坐标信息确定其所处网格序号, 有点占据的网格将会被标记成已有点, 其对应的标记位  $flag = 1$ , 同时该网格内部也将保存对应点的坐标, 若有多个点同时占据某个网格, 则该网格将会改为保存这些点的几何中心坐标, 而未有点占据的网格内部则作为空网格不做标记处理, 其对应的标记位  $flag = 0$ , 网格内部也将不会保存任何点的坐标, 经此番处理后的网格数据将更新为  $M_0$  并会作为下一时刻的输入数据补充点云信息; 读入下一时刻  $t_1$  时刻雷达新采集到的点云数据  $P_1$  同时输入上一时刻通过计算得到的网格数据  $M_0$ , 首先将上一时刻保存的网格内所有点进行坐标变换,  $t_1$  时刻  $P_0$  相对于飞行器的坐标将会因为飞行器的位子变换发生对应的移动, 变为  $P_0'$ 。

$$P_0' = -(t_1 - t_0)VP_0 \quad (5)$$

其中,  $V$  依然为飞行器位移线速度, 值得注意的是由于每一帧进行网格化处理的点云均为扭曲校正后的数据集合, 因此在这里不再考虑飞行器旋转对  $P_0$  产生的影响。按照步骤 2), 将  $P_1$  与  $P_0'$  中处于网格内部的点云保存下来并更新维护窗口, 更新后网格数据为  $M_1$ ; 之后的每一时刻均照 3) 循环迭代更新网格数据, 利用过去时刻采集到的点云数据完善补充当前时刻平面检测所用地形环境信息。

维护窗口  $W_1$  内点云数据如图 4 所示, 其中绿框为窗口范围, 内部即为网格数据  $M$ , 窗口内部的绿色点云是

经过网格化处理后保存下来的地形点云信息, 点云集合记为  $\hat{P}_0$ , 根据图中显示, 保留下来的点云  $\hat{P}_0$  的密度明显比雷达采集输入的原始点云更高, 包含的点云范围也更加广阔, 这说明该方法可以有效丰富平面检测时的点云信息, 弥补雷达硬件以及飞行器偏移旋转带来的采样漏缺, 从而提高输出结果的准确性与鲁棒性。

由于正常情况下飞行器选择着陆地点时只需要在其正下方一小块特定区域选择一块合适的地址, 因此在进行平面检测与跟踪前本文将缩小着陆地点搜寻范围, 具体做法为在  $W_1$  内部创建一个相对较小的维护窗口  $W_2$ , 将每一时刻更新得到的  $W_1$  内部数据作为建立窗口  $W_2$  时的输入数据, 再按照前文建立维护窗口的步骤实时更新保存  $W_2$ , 最后仅对  $W_2$  内保留下来的点云执行平面检测与跟踪算法。本文中选取  $W_2$  大小为  $40\text{ m} \times 40\text{ m}$  的矩形范围, 内部网格分辨率为  $100 \times 100$ , 维护窗口  $W_2$  内点云数据图如图 4 所示, 其中红框界定了维护窗口  $W_2$  的大小范围, 窗口内部的蓝色点云则是经过将  $W_2$  网格化处理后每个细胞网格内部保存下来的地形点云信息, 该点云集合记为  $\hat{P}_1$ 。

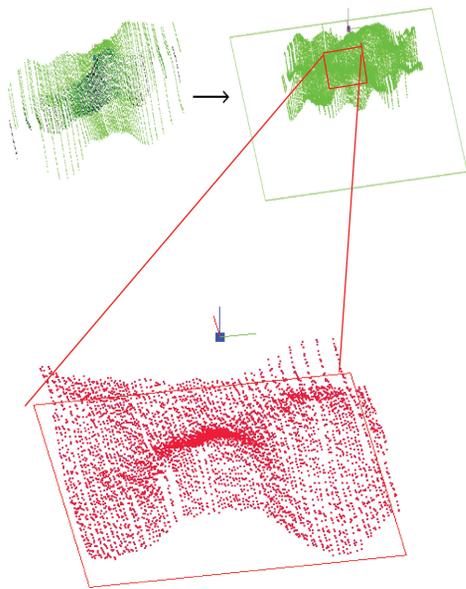


图 4 维护窗口内部点云

Fig. 4 Point cloud in typo windows

## 1.2 着陆区域推荐方法

为了选出适合无人机降落的地址区域, 算法的任务是在输入的密集点云中找出一块特殊的点云区域, 该区域需要满足几何特征: 1) 平面光滑度低; 2) 区域面积足够放置无人机; 3) 区域对应平面法向量应与水平面呈近似垂直关系。

本文改进随机采样一致性 (RANSAC) 算法完成符合要求的平面区域检测任务。该方法核心思想在于随机采

样与一致性投票,计算伪代码如下。

#### 算法 1 RANSAC

输入:采集原始点集  $P$

输出:候选着陆平面  $A$

1. 初始化:平面  $A$  的分数  $Score_A = 0$ ,未处理点云数目  $U$ ,原始点云数目  $I$ ,  $U = I$ ;

2. **for**  $U > \frac{1}{10}I$  **do**

3. 从  $P$  中随机选取不共面三点  $p_1, p_2, p_3$ ;

4. 计算  $p_1, p_2, p_3$  构成的平面  $A'$  的平面方程;

5. 根据  $A'$  上内点 (*inliers*) 数目  $innum$ ,  $A'$  的法向量以及光滑度确定  $A'$  的分数  $Score_{A'}$ ;

6.  $U = U - innum$ ;

7.  $P = P - inliers$ ;

8. **if**  $Score_{A'} > Score_A$  **then**

9.  $Score_A = Score_{A'}$ ;

10.  $A = A'$ ;

11. **end if**

12. **end for**

构成一个平面则至少需要 3 个点,因此第一步随机取点需要从输入的  $W_2$  内点云中随机选取 3 个点,记为  $p_1, p_2, p_3$ 。为了保证 3 点可以构成一个平面,且 3 点在实际场景中的同一平面区域上(3 点中间不存在点云断层的情况),所以随机取点时也需要对选取的点加上约束条件:

1) 三点不共线,所有方向向量之间不可满足平行关系;

2) 3 点不可以距离过远,当足够近的 3 点构成平面时,生成的平面才会尽量减少中间出现异常凸起或者点云断层的情况。在程序中即计算 3 点任意两点间的距离,并人为设定一个距离阈值,只有当 3 个距离都小于该阈值时才对  $p_1, p_2, p_3$  进行保留,如果不满足条件则继续循环选取直至找到。由于仅在维护窗口内的点云中选取随机点,因此为了保证随机点之间联系足够紧密,本文选择维护窗口边长的  $1/10$  即 4 m 为随机点任意两点间的距离阈值。接下来计算 3 点构成的平面方程。3 点中由任意两点可以构成位于平面上的一个方向向量记为  $\vec{n}_1$ 、 $\vec{n}_2$ , 其中:

$$\vec{n}_1 = p_1 - p_2 \quad (6)$$

$$\vec{n}_2 = p_1 - p_3 \quad (7)$$

将得到的方向向量  $\vec{n}_1, \vec{n}_2$  叉乘,令其结果为  $\vec{n}$ , 即:

$$\vec{n} = \vec{n}_1 \times \vec{n}_2 \quad (8)$$

则  $\vec{n}$  为 3 点所构成平面的法向量,其对应的三点坐标  $(A, B, C)$  为平面方程中  $x, y, z$  项前面的系数,最后再将  $p_1, p_2, p_3$  中任意一点坐标代入方程中即可得出最后的平面方程表达式,表达式如下:

$$Ax + By + Cz + D = 0 \quad (9)$$

其中  $D$  的值为:

$$D = -(Ax_1 + By_1 + Cz_1) \quad (10)$$

选取  $p_1$  的坐标  $(x_1, y_1, z_1)$  代入。当确定平面方程后,记平面为  $S$ ,为了算出地形场景中可能存在的平面区域数量,还需要判断有多少点云落在生成的平面  $S$  上,只有当该数目足够大时才可以将其视为工作环境中的一个平面,具体方法是遍历障碍物上所有点云,计算每个点与  $S$  之间的距离  $dis$ :

$$dis = \frac{|Ax_0 + By_0 + Cz_0 + D|}{\sqrt{A^2 + B^2 + C^2}} \quad (11)$$

以  $p_1, p_2, p_3$  三点构成三角形的中心为圆心,遍历其半径为 2 m 圆形范围中所有属于  $P_1$  集合的点,设  $(x_0, y_0, z_0)$  为遍历过程中  $P_1$  内任一点  $p_0$  的三维坐标,  $dis$  为  $p_0$  与  $S$  之间的距离,当  $dis$  值小于设定阈值  $max\_dis$  时则视  $p_0$  为  $S$  上的一点,本文考虑到输入抖动以及噪声所造成的点云坐标误差基本保持在 0.2 m 以内,因此选取  $max\_dis = 0.2$  m,每出现这样一个  $p_0$  便使计数变量  $innum$  与  $p\_num$  加 1,并且将  $p_0$  视为已处理点剔除出输入点云集合,而如果超出距离阈值则说明  $p_0$  距离  $S$  太远,视作不在该平面上,仅使计数变量  $p\_num$  加 1。本文将平面上点数  $innum$  与区域内点数  $p\_num$  两者之间的比值作为衡量平面  $S$  光滑程度的标准,记作光滑度  $\mu$ :

$$\mu = \frac{innum}{p\_num} \times 100\% \quad (12)$$

为验证本文所定义光滑度的合理性,通过对地形点云数据进行采样分析,图 5 所示为本文定义的平面光滑度  $\mu$  的变化曲线,横坐标为区域内所有点与  $S$  距离  $dis$  的平方和,该平方和为反映区域平面光滑度的标准指标。从图 5 可以看出本文中定义的  $\mu$  与该指标基本呈负线性相关,因此  $\mu$  也可以准确反映平面光滑度,即  $\mu$  的值越大说明该区域内点云光滑度越高,越小则说明光滑度越低。

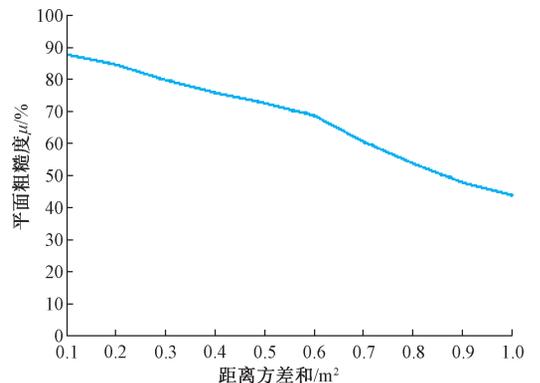


图 5 平面光滑度  $\mu$  变化曲线

Fig. 5 Change graph of plane smoothness  $\mu$

传统 RANSAC 算法在随机生成初始平面后将会遍历输入点云中的所有点并判断该点是否在生成平面上从而识别出区域内所包含的所有平面,而生成平面的具体信息以及是否适合飞行器着陆则需要相应的打分与选择操作来实现。当所生成平面的相关信息都已完备后,本文的方法只保留分数最高的平面。而平面的打分依据则来源于前文中提到的无人机安全着陆的选址要求是平面光滑度高且法向量近似与水平面垂直。首先给光滑度标准  $\mu$  设定一个阈值  $\mu_0$ , 当计算结果小于  $\mu_0$  时光滑度分值为 0, 当结果大于  $\mu_0$  时, 光滑度分值会随着  $\mu_0$  的增大而线性增加, 说明平面越光滑, 本文设  $\mu_0 = 70\%$ 。

在对平面法向量进行打分时, 由于无人机对着陆区域法向量的要求是尽可能垂直于水平面, 因此法向量在水平面(本文为  $xoy$  平面)上的相应坐标(即  $xy$  轴坐标)应为 0 或尽可能接近于 0。

因此本文的程序最后得到平面的法向量的  $xy$  坐标应该在 0 上下一个较小范围内浮动, 将法向量视为打分标准的话, 那么程序应按照如下规则进行打分,  $xy$  坐标绝对值与 0 的差值均值越小, 则法向量分值越大。

其中具体的分数计算公式与均值计算公式如下:

$$\rho_N = \frac{(|x| + |y|)}{2} \quad (13)$$

$$Score = \alpha S_\mu + \beta S_N \quad (14)$$

式中:  $Score$  为平面对应总分;  $\rho_N$  为法向量均值;  $|x|$ 、 $|y|$  为法向量  $xy$  坐标绝对值;  $\alpha$  为光滑度权重;  $\beta$  为法向量权重。当  $\rho_N$  越接近 0, 其对应分数  $S_N$  就越高, 最高为 1。

$S_N$  与  $\rho_N$  关系式为:

$$S_N = 1 - \sqrt{\rho_N} \quad (15)$$

$S_\mu$  为平面光滑度分值:

$$S_\mu = \frac{inum}{p\_num} \quad (16)$$

由于飞行器在最终着陆时着陆在一个坡度较大的斜坡面上的危险程度要大于停留在粗糙度较高的区域, 因此最终分值内代表选取平面坡度情况的法向量权重要略大于光滑度权重。选址结果误差随  $\alpha$  取值变化曲线如图 6 所示, 可以看出  $\alpha$  在取值 0.4~0.5 得到的结果误差最小, 因此本文中选取  $\alpha = 0.4$ ,  $\beta = 0.6$ 。

每当生成一个平面便有以上打分规则对其进行打分, 只有当分数高于人为设定的最低分数时才会保留该平面, 并将该平面上所有点标记为已处理点, 由于最终选择的平面坡度范围应在安全坡度  $30^\circ$  范围以内, 而当坡度值为  $30^\circ$  时对应的法向量分数应为 0.78, 同时设定的光滑度阈值  $\mu_0 = 70\%$ , 此时对应的平面光滑度分值为 0.7, 再根据多组实验确定的分值权重  $\alpha$  与  $\beta$  的取值, 因此本文最终确定的分数阈值为 0.75。确定平面后每次进行迭代循环都是在未处理点中继续搜寻候选平面, 从而简

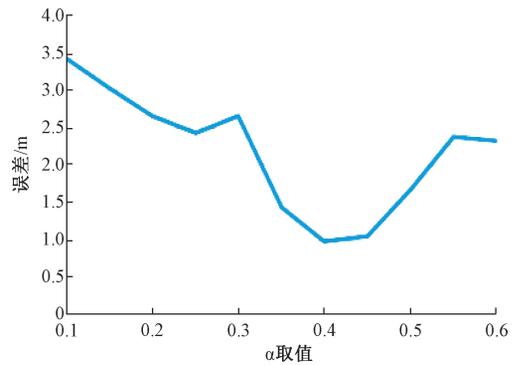


图 6 误差随  $\alpha$  变化曲线

Fig. 6 Error curve with different  $\alpha$  value

化运算步骤, 加快运行速率。当未处理点足够少时则认为飞行器下方点云处理完毕, 从前面得到的所有候选平面中找出得分最高的平面作为飞行器降落区域在结果中标记, 并将该着陆点与飞行器的相对距离和中心坐标实时反馈给飞行器, 记该着陆点为  $D_1$ 。

### 1.3 平面跟踪方法

为了避免检测过程中由于随机算法导致的连续时间帧内  $D_1$  出现较大抖动的情况, 在使用平面检测算法的同时, 本文加入平面跟踪算法, 当  $W_2$  内出现适合飞行器降落的着陆点后, 程序会根据飞行器的实时飞行位姿变换, 每个时间帧调整更新着陆点  $D_1$  的位置坐标完成对视野范围内最佳着陆点的跟踪任务, 从而减少随机算法带来的结果不确定性与选址计算所用时间成本。设更新后的着陆点为  $D_1'$ , 则  $D_1'$  与  $D_1$  的位置坐标之间应满足如下关系式:

$$D_1' = -tVD_1 \quad (17)$$

其中,  $t$  为相邻时间帧之间的时间间隔,  $V$  是飞行器位移线速度。对最佳着陆点的跟踪会一直持续到该着陆点超出  $W_2$  窗口范围, 此时程序将会对窗口内部点云进行平面检测工作, 重新确定新的着陆点并在后续时刻加以跟踪。

考虑到飞行器在探索先前时刻未经过的区域时,  $W_1$  窗口内不包含的雷达新采集到的地形点云, 记为  $P_{new}$  中可能存在新的最佳着陆点, 且其地形信息与此时正在跟踪的着陆位置  $D_1$  相差不大, 因此本文在进行平面跟踪的同时也在点集  $P_{new}$  中进行平面检测操作, 检测是否存在新的着陆点可以作为合适的降落区域, 若存在则与  $D_1$  一同标注出来, 记为  $D_2$ , 同时返回其相关地形信息作为备选着陆点。如果在检测打分过程中  $D_2$  区域的最终分数  $s_2$  大于正在跟踪的  $D_1$  的相应分数  $s_1$ , 则说明  $D_2$  相较于  $D_1$  更适合飞行器进行着陆工作, 此时会将  $D_1$  的位置坐标更新为  $D_2$  的位置坐标, 在下一时刻加以跟踪。检测与跟踪算法同步进行的流程如图 7 所示。

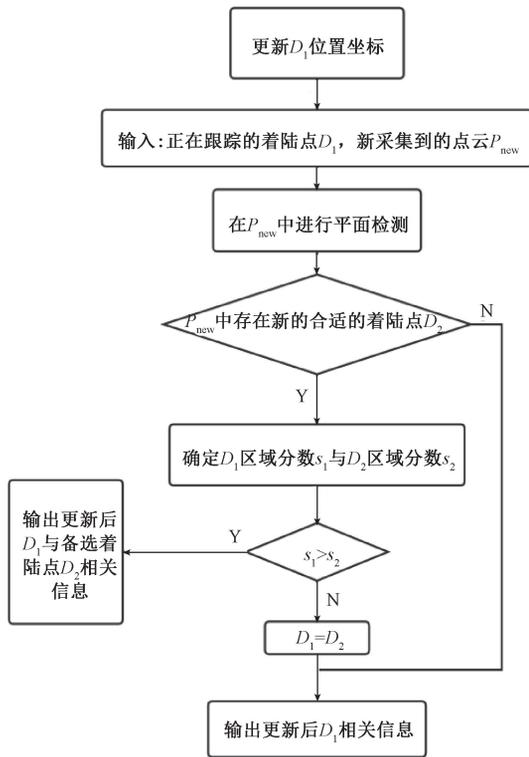


图 7 平面检测与跟踪算法同步流程

Fig. 7 Flow chart of planar detection and tracking

## 2 实验与分析

### 2.1 仿真实验环境搭建

本文选用的激光雷达为 RS-LiDAR-32 型,这是一种 32 线激光雷达,激光束旋转频率为 5 Hz 即每 0.2 s 扫描一圈,相邻激光束之间角度差为  $1^\circ$ ,因此雷达垂直视野范围为  $32^\circ$ ,最远扫描距离为 200 m。采样频率为 900 Hz,即同一时刻点云中相邻两条点云线之间对应的激光束旋转角度差为  $2^\circ$ 。

本文实验中功能程序在 Intel(R) Core(TM) i5-8400 CPU @ 2.80 GHz CPU 以及 8.00 GB 内存硬件条件下运行,软件开发系统为 Windows 10x64 位系统。

为对所提出方法进行数值仿真验证,本文选择使用 Unity3d 模拟自然地形,并在根据所选用的激光雷达的型号指标,对其点云采集进行了相应的仿真,将仿真过程中扫描到的地形点云数据输入进算法中观察着陆点选址的结果情况。仿真平台下模拟环境如图 8 所示。

可以看出虚拟自然地形存在凹凸不平明显不适合无人飞机降落的山地丘陵区域,同时也有平坦广阔的地形区域,而这正是适合无人飞机着陆的地址,本文希望能够利用雷达扫描到的点云数据在地貌复杂的环境中正确选择适合无人飞机着陆的地址,并实时反馈所选区域的位置以及

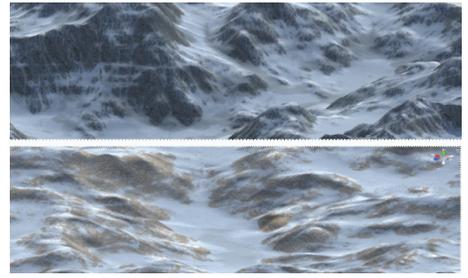


图 8 模拟场景图

Fig. 8 Simulated scene graph

距离等信息。由此可见模拟出的场景符合实际场景情况,能够用于作为程序算法的输入来进行仿真实验。

为了正确采集到模拟环境场景中的点云数据,还需要在软件中设计模拟激光雷达仪器,用于当做信息采集装备。本文中的虚拟雷达可以随时对雷达的线数以及扫描频率进行修改以满足实际需要。

### 2.2 平面检测与跟踪结果分析

由于相邻帧之间采集到的点云数据变化较小,难以看出检测跟踪结果之间的差异,因此本文中选择每隔 10 帧截取一次输出结果。最终得到的着陆点  $D_1$  在图像中由绿色方块标出,方块大小为  $2\text{ m} \times 2\text{ m}$  的正方形。由图 9 与表 1 可以看出,在第 36 帧时检测到  $D_1$  符合飞行器降落需求,在 36~56 帧之间,执行跟踪算法,  $D_1$  根据飞行器的移动而位置坐标产生相应平移,区域信息未变,但 56 帧之后由于  $D_1$  已经超出维护窗口  $W_1$  的范围,因此重新调用检测算法,获取新的着陆点。输出的  $D_1$  的距离由  $D_1$  中心点与坐标原点之间距离表示,坡度记水平面为  $0^\circ$  由  $D_1$  平面法向量算得,平面光滑度则由式 (12) 计算得出,表 1 的结果表明,连续时间帧内着陆点  $D_1$  都处于无人机下方地形中相对平坦的区域内,且该区域面积足够,平面法向基本垂直于水平面,符合无人飞机着陆要求。

表 1  $D_1$  信息表Table 1 Information sheet of  $D_1$ 

帧数	距离/m	坡度/ $^\circ$	光滑度/%
36	13.97	0.46	75.04
46	11.70	0.46	75.04
56	17.01	0.46	75.04
66	12.95	3.03	78.96

当新采集到的点云  $P_{\text{new}}$  中存在不同于正在跟踪的着陆点  $D_1$  的着陆区域时,最后输出结果会加入备选着陆点  $D_2$ ,图像中用黄色方块标出,大小与  $D_1$  相同。由图 10 与表 2 可以看出第 14 帧时,  $P_{\text{new}}$  中不存在合适的着陆点,因此  $D_2$  不进行标注,14~34 帧之间对  $D_1$  进行跟踪操作。在第 24 帧时虽然  $P_{\text{new}}$  中出现了新的备选着陆点,但是由于  $D_2$  相对于  $D_1$  分数更低因此继续跟踪  $D_1$ ,而在第 34 帧

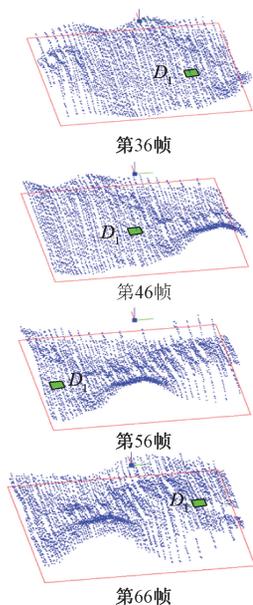


图 9 输出着陆点  $D_1$  示意图

Fig. 9 Diagrams of  $D_1$

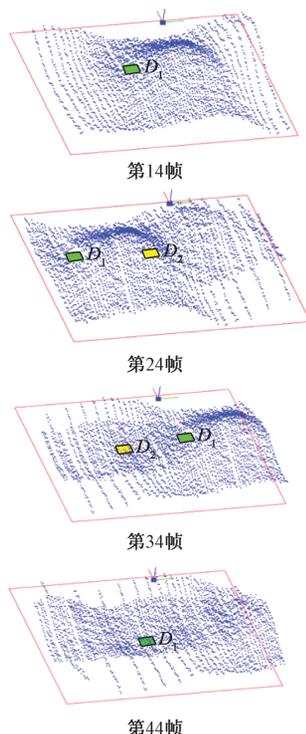


图 10 输出着陆点  $D_1$  与备选着陆点  $D_2$  示意图

Fig. 10 Diagrams of  $D_1$  and  $D_2$

时出现的  $D_2$  在坡度以及平面光滑度等方面数据都比  $D_1$  更适合飞行器着陆,因此会将  $D_1$  的位置更新为  $D_2$  的位置坐标并继续跟踪。

表 2  $D_1$  与  $D_2$  信息表

Table 2 Information sheet of  $D_1$  and  $D_2$

帧数	距离/m		坡度/(°)		光滑度/%	
	$D_1$	$D_2$	$D_1$	$D_2$	$D_1$	$D_2$
14	11.29	—	4.90	—	76.50	—
24	15.25	8.08	4.90	16.33	76.50	73.15
34	12.26	11.64	4.90	1.02	76.50	77.31
44	11.35	—	1.02	—	77.31	—

连续帧中程序运算时间以及维护窗口  $W_2$  内点云  $P_1$  数量的变化曲线如图 11 所示。由图 11 可以看出,随着点数的增多,运算时间有所增加,但计算时间峰值小于 0.1 s,可以保证实时的着陆点推荐响应,保持在无人机降落过程可接受的范围以内。同时维护窗口  $W_2$  内点数会因为过去时刻点云信息的保留而逐渐增加,丰富的飞行器进行着陆选址时的视觉信息,弥补了仅依靠雷达采集到的原始点云数据中存在的缺失问题,而当点数增加到窗口分辨率上限  $100 \times 100$  时,  $W_2$  内也将不会继续累加所有采集到的点云,节省了计算机对内存的消耗。

为验证所提出随机采样算法的有效性,通过遍历当前时刻维护窗口  $W_2$  内所有可能的着陆点位置并比对区域地形信息可以得到最理想的着陆区域作为参考,如图 12(a) 中圆圈所示。而其中绿色标记则为多次调用检测算法得到的着陆位置,可以看出,虽然检测算法存在一定的随机性,但是多次调用过程中输出结果并不会出现异常的抖

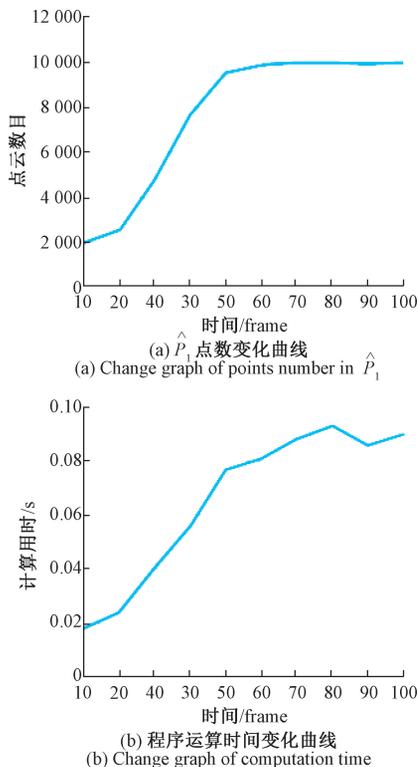
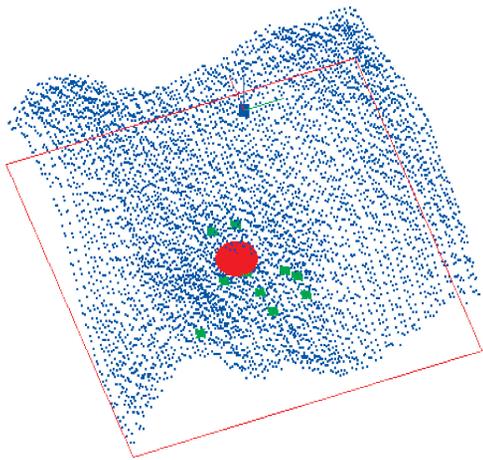


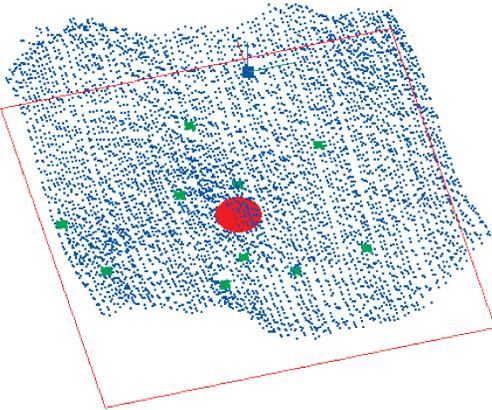
图 11 点数与运算时间变化曲线

Fig. 11 Change graph of points number and computation time

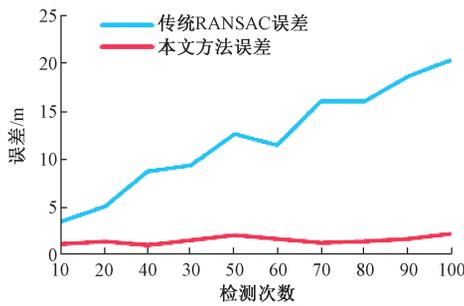
动,且大多靠近理想着陆点,满足飞行器着陆需要。



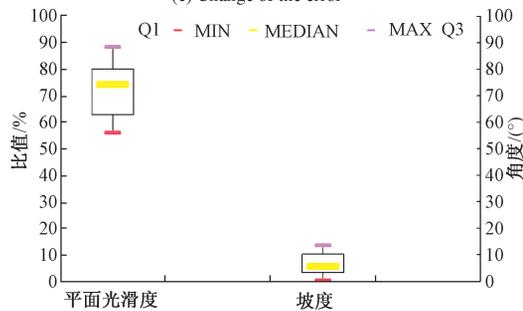
(a) 检测结果分布  
(a) Distribution of detection results



(b) 传统RANSAC检测结果分布与误差  
(b) Distribution and error of traditional RANSAC detection results



(c) 误差变化  
(c) Change of the error



(d)  $\hat{D}_1$  信息箱型  
(d) Box of the information of  $\hat{D}_1$

图 12 着陆点  $D_1$  分布与信息

图 12(b) 所示为传统 RANSAC 算法得到的检测结果分布,可以看出由于传统 RANSAC 方法仅仅进行识别平面工作并未对最合适的着陆点进行进一步选择,因此最终得到的着陆结果随机性较高,从而导致结果分布较为分散并未集中于理想着陆点周围,并且当平面面积越大时传统方法的结果越会倾向于落在该平面中,而理想结果往往不一定是面积最大的平面,因此误差会随着检测次数逐渐上升,结果稳定性方面也就无法得到有效保障。

图 12(c) 所示为本文方法的检测结果与传统 RANSAC 方法的检测结果误差对比,图中误差值为多次检测结果与理想着陆点的平均距离误差。相比较之下传统方法的结果与理想着陆点之间误差也较大,且重复检测次数越多,结果随机性越高误差也会越大,这就不符合飞行器在进行实际着陆工作时的安全需要,而本文的方法误差基本维持在 3 m 以下,可以保证最终选出的着陆地点的安全性与稳定性。

图 12(d) 所示为本文检测算法得到的结果区域的光滑度与坡度信息,其中平面光滑度由式(12)计算得出,坡度为平面法向量与  $z$  轴夹角,水平面为  $0^\circ$ 。从图 12(d)中也可以看出检测输出的着陆点位置从着陆面平坦程度以及倾斜程度等方面来看都满足飞行器降落需要,可以作为候选着陆点供飞行器选择。

### 3 结 论

本文针对使用激光雷达的小型无人飞行器进行着陆点自主选址工作的研究,提出了一种实时的地形点云维护及着陆点推荐方法,解决了飞行器自主着陆选址时受硬件条件以及环境影响误差较大计算实时性低的问题。

根据飞行器位姿校正点云坐标并借助过去时刻采集到的点云信息补充当前时刻的点云数据,完善了视觉信息;设计了面向着陆点推荐的基于雷达点云的平面检测 RANSAC 算法;利用平面检测得到的结果实时返回当前时刻场景中的最优着陆地点并在显示界面加以标注。

在仿真实验场景中取得的数据结果表明本方法基本满足无人机着陆的面积以及平坦程度的需求,具有较高的实时性与准确性。在后续工作中,将会对着陆区域提出更高的要求,改进平面检测算法,进一步提高在不同场景下的适用性与鲁棒性。

### 参考文献

[ 1 ] MILLER B, MILLER A, POPOV A, STEPANYAN K. UAV Landing based on the optical flow video navigation [J]. Sensors (Basel), 2019,19(6):1351.  
 [ 2 ] GALIMOV M, FEDORENKO R, KLIMCHIK A. UAV positioning mechanisms in landing stations: Classification

Fig. 12 Distribution and the related information of  $D_1$

- and engineering design review [J]. *Sensors* 2020, 20(13): 3648.
- [ 3 ] 赵俊宏,吕恩利,陆华忠,等. 基于激光雷达的托盘位姿识别算法及验证 [J]. *仪器仪表学报*, 2017, 38(10):2469-2476.
- ZHAO J H, LYU EN L, LU H ZH, et al. Pallet localization detecting algorithm based on laser scanning[J]. *Chinese Journal of Scientific Instrument*, 2017,38(10): 2469-2476.
- [ 4 ] 沈小军,杜勇,王仁德,等. 基于地面激光雷达的输电线路铁塔倾斜度测量[J]. *电子测量与仪器学报*, 2017,31(4):516-521.
- SHEN X J, DU Y, WANG R D, et al. Inclination measurement of transmission line tower based on terrestrial 3D LIDAR [J]. *Journal of Electronic Measurement and Instrumentation*, 2017, 31 ( 4 ) : 516-521.
- [ 5 ] 余洪山,付强,孙健,等. 面向室内移动机器人的改进 3D-NDT 点云配准系统 [J]. *仪器仪表学报*, 2019, 40(9):151-161.
- YU H SH, FU Q, SUN J, et al. Improved 3D NDT point cloud registration algorithm for indoor mobile robot[J]. *Chinese Journal of Scientific Instrument*, 2019,40(9): 151-161.
- [ 6 ] 朱玲嫵,常建华,李红旭,等. 改进的激光雷达回波信号去噪方法 [J]. *电子测量与仪器学报*, 2017, 31(10):1608-1613.
- ZHU L Y, CHANG J H, LI H X, et al. Improved denoising method of lidar echo signal [J]. *Journal of Electronic Measurement and Instrument*. 2017,31(10): 1608-1613.
- [ 7 ] SCHERER S, CHAMBERLAIN L, SINGH S. Autonomous landing at unprepared sites by a full-scale helicopter [J]. *Robotics and Autonomous Systems*, 2012, 60(12): 1545-1562.
- [ 8 ] 章志诚,杜昌平. 基于激光雷达的多旋翼飞行器实时避障系统 [J]. *计算机测量与控制*, 2016, 24(9): 117-121.
- ZHANG ZH CH, DU CH P. A real-time obstacle avoidance system for multicopter based on laser radar[J]. *Computer Measurement and Control*, 2016, 24(9): 117-121.
- [ 9 ] YANG T, LI P, ZHANG H, et al. Monocular vision SLAM-based UAV autonomous landing in emergencies and unknown environments [J]. *Electronics*, 2018, 7(5):73.
- [ 10 ] 孔天恒,方舟,李平. 基于雷达-扫描器/惯性导航系统的微小型无人机室内组合导航[J]. *控制理论与应用*, 2014, 31(5): 607-613.
- KONG T H, FANG ZH, LI P. Indoor integrated navigation of micro aerial vehicle based on radar-scanner and inertial navigation system [J]. *Control Theory and Applications*, 2014, 31(5): 607-613.
- [ 11 ] GRISETTI G, STACHNISS C, BURGARD W. Improved techniques for grid mapping with Rao-Blackwellized particle filters [J]. *IEEE Transactions on Robotics*, 2007, 23(1): 34-46.
- [ 12 ] 石鹏,赖际舟,吕品,等. 复杂环境下微小飞行器惯性/激光雷达 Robust-SLAM 方法 [J]. *导航定位与授时*, 2019, 6(1):14-21.
- SHI P, LAI J ZH, LYU P, et al. A robust-SLAM method for MAVs using lidar and IMU in complex environment [J]. *Navigation Positioning and Timing*, 2019, 6(1):14-21.
- [ 13 ] BADINO H, HUBER D, KANADE T. Integrating LIDAR into stereo for fast and improved disparity computation [C]. *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, 2011:405-412.
- [ 14 ] GANDHI D, PINTO L, GUPTA A. Learning to fly by crashing [C]. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017: 3948-3955.
- [ 15 ] MA J, ZHAO J, TIAN J, et al. Robust point matching via vector field consensus [J]. *IEEE Transactions on Image Processing*, 2014,23(4): 1706-1721.
- [ 16 ] WANG H, SCHMID C. Action recognition with improved trajectories [C]. *2013 IEEE International Conference on Computer Vision*, 2013:3551-3558.
- [ 17 ] ZHANG Z, HUANG Y, ZHANG W, et al. Comparisons of planar detection for service robot with RANSAC and region growing algorithm [C]. *36th Chinese Control Conference*, 2017.
- [ 18 ] LI L, YANG F, ZHU H. An improved RANSAC for 3D point cloud plane segmentation based on normal distribution transformation cells [J]. *Remote Sensing*, 2017, 9(5):433-449.
- [ 19 ] 方根在,黎福海. 小型飞行器姿态估计系统设计与实现 [J]. *电子测量与仪器学报*, 2017,31(3): 474-480.
- FANG F Z, LI F H. Design and implementation of MAV attitude estimation system [J]. *Journal of Electronic Measurement and Instrumentation*, 2017, 31 ( 3 ) : 474-480.
- [ 20 ] LI W J, FU Z Y. Unmanned aerial vehicle positioning based on multi-sensor information fusion [J]. *Geo-spatial Information Science*, 2018, 21 ( 4 ) : 302-310.

## 作者简介



邢闻, 2018 年于东南大学获得学士学位, 现为东南大学硕士研究生, 主要研究方向为移动机器人, 点云数据处理等。

E-mail: 220183279@seu.edu.cn

**Xing Wen** received his B. Sc. degree from Southeast University in 2018. Now he is a

M. Sc. candidate at Southeast University. His main research interests include mobile robotics, point cloud data processing, etc.



朱利丰, 2007 年于复旦大学获得学士学位, 2012 年于北京大学获得博士学位, 现为东南大学副教授、硕士生导师, 主要研究方向为计算机图形、辅助设计与仿真, 虚拟现实与人机交互等。

E-mail: lifengzhu@seu.edu.cn

**Zhu Lifeng** received his B. Sc. degree from Fudan University in 2007 and Ph. D. from Peking University in 2012.

He is an associate professor and M. Sc. supervisor at Southeast University. His main research interests include computer graphics, aided design and simulation, virtual reality and human-computer interaction.



宋爱国, 分别于 1990 年和 1993 年获得南京航空航天大学的学士学位和硕士学位, 1996 年于东南大学获得博士学位, 现为东南大学教授, 东南大学机器人传感器与控制实验室的负责人, 主要研究方向为遥操作机器人, 机器人传感技术和远程康复机器人等。

E-mail: a.g.song@seu.edu.cn

**Song Aiguo** received the B. Sc. degree in automatic control and the MS degree in measurement and control from the Nanjing University of Aeronautics and Astronautics in 1990 and 1993, and the Ph. D. degree from Southeast University in 1996, respectively. He is currently a professor at Southeast University. His main research interests include teleoperation robot, robot sensing technology, and telerehabilitation robot.