

褚福春, 宫金良, 张彦斐. 基于密度自适应的 RANSAC 非结构化环境下果园机器人导航 [J]. 华南农业大学学报, 2022, 43(5): 99-107.
CHU Fuchun, GONG Jinliang, ZHANG Yanfei. Orchard robot navigation in unstructured environment based on density adaptive RANSAC[J]. Journal of South China Agricultural University, 2022, 43(5): 99-107.

基于密度自适应的 RANSAC 非结构化 环境下果园机器人导航

褚福春¹, 宫金良¹, 张彦斐²

(1 山东理工大学 机械工程学院, 山东 淄博 255000; 2 山东理工大学

农业工程与食品科学学院, 山东 淄博 255000)

摘要:【目的】提出一种基于多传感器融合的果园导航方案, 解决果园机器人在 GPS 导航过程中受果树遮挡导致信号弱、定位效果差的问题。【方法】通过 16 线激光雷达采集高精度的三维点云数据, 利用 Voxel grid filter 滤波算法进行点云预处理, 降低点云密度并去除离散点, 将果树行通过欧几里类算法进行聚类, 采用改进的随机采样一致性 (Random sample consensus, RANSAC) 算法拟合出果树行直线, 根据平行直线的关系, 推算得到导航线, 并融合惯性测量单元 (Inertial measurement unit, IMU) 对果园机器人进行高精度定位。基于差速转向和纯追踪模型进行轨迹跟踪, 实现果园机器人在果树行间自主导航以及自动换行的目标。【结果】在将激光雷达和 IMU 的数据进行融合后, 获取到果园机器人的准确位姿, 当机器人以速度 0.8 m/s 在果园作业时, 对比最小二乘法和传统 RANSAC 法产生的偏差, 基于密度自适应 RANSAC 法产生的横向偏差不超过 0.1 m、航向角偏差不超过 1.5°, 均为 3 种方法中的最小值。但当机器人速度增加到 1.0 m/s 时, 各项偏差均明显增大。【结论】本文提出的基于多传感器融合的果园机器人导航技术适用于大多数规范化果园, 具有重要推广价值。

关键词: 农业机器人; 自主导航; 直线拟合; 差速转向; 改进 RANSAC 算法

中图分类号: S224;S628

文献标志码: A

文章编号: 1001-411X(2022)05-0099-09

Orchard robot navigation in unstructured environment based on density adaptive RANSAC

CHU Fuchun¹, GONG Jinliang¹, ZHANG Yanfei²

(1 School of Mechanical Engineering, Shandong University of Technology, Zibo 255000, China;

2 School of Agricultural Engineering and Food Science, Shandong University
of Technology, Zibo 255000, China)

Abstract: 【Objective】An orchard navigation scheme based on multi-sensor fusion was proposed to solve the problems of weak signal and poor positioning effect caused by tree occlusion in the GPS navigation process of orchard robot. 【Method】High-precision 3D point cloud data were collected by 16-line lidar, point cloud was preprocessed by Voxel grid filter algorithm, point cloud density was reduced and discrete points were removed,

收稿日期: 2021-11-21 网络首发时间: 2022-05-07 16:56:07

网络首发地址: <https://kns.cnki.net/kcms/detail/44.1110.S.20220507.1338.002.html>

作者简介: 褚福春, 硕士研究生, 主要从事农业机器人自主导航技术研究, E-mail: 1371368298@qq.com; 通信作者: 宫金良, 副教授, 博士, 主要从事机器人与智能农机装备研究, E-mail: gjlwing@qq.com

基金项目: 山东省引进顶尖人才“一事一议”专项经费 (鲁政办字 [2018]27 号); 山东省重点研发计划 (重大科技创新工程) (2020CXGC010804); 山东省自然科学基金 (ZR202102210303); 淄博市重点研发计划 (校城融合类) 生态无人农场研究院项目 (2019ZBXC200)

fruit tree rows were clustered by Euclidian algorithm, and the straight lines of fruit tree rows were fitted by improved random sampling consistency (RANSAC) algorithm. According to the relationship of parallel lines, the navigation line was calculated and integrated with inertial measurement unit (IMU) for high-precision positioning of orchard robot. Based on differential steering and pure tracking model, the goal of autonomous navigation and automatic line wrapping of orchard robot was realized. 【Result】 After the data fusion of lidar and IMU, the accurate position and pose of the robot were obtained. Compared with the deviation produced by the least square method and the traditional RANSAC method, the lateral deviation based on density adaptive RANSAC method was less than 0.1 m and the heading angle deviation was less than 1.5° when the robot was operating in the orchard at the speed of 0.8 m/s. The deviations were the minimum in the three methods. However, when the robot speed increased to 1.0 m/s, all the deviations increased obviously. 【Conclusion】 The orchard robot navigation technology based on multi-sensor fusion proposed in this paper is suitable for most standardized orchards and has important promotion value.

Key words: Agricultural robot; Autonomous navigation; Linear fitting; Differential steering; Improved RANSAC algorithm

我国作为水果生产大国，水果种植面积与产量均为世界第一。但是果树种植过程中存在着自动化水平低、管理不规范和采摘效率低等问题。针对这一系列问题，加大对果园智能装备的发展力度，尤其是加大对丘陵山区果园管理装备的改造已迫在眉睫。提高果园生产作业的自动化水平，不仅可以提高水果产量，还能够减少人工劳动力，降低生产成本，对我国许多水果种植地区具有比较重要的现实意义，因此研究出一款应用于果园的自主作业机器人势在必行^[1-5]。

智慧农业是农业信息化发展从数字化到网络化再到智能化的高级阶段,对农业发展具有里程碑意义,已成为世界现代农业发展的趋势。为进一步提高生产效率和水果品质,农业机械装备的智能化是未来发展的必然趋势。随着智慧农业的不断发展,智能化机器人技术被广泛应用于果园除草、剪枝、采摘等作业环节,有效地减轻了果农的劳动强度,提高了作业效率与质量^[6-9]。机器人在果园等非结构化的环境下行间行驶的安全性是研究重点,合理、可靠、安全的导航路径规划方法是保证机器人安全行驶的重要手段之一^[10-11]。传统的 GPS 导航方案,在果园中由于受到果树遮挡,卫星信号弱,无法进行高精度定位^[12]。激光雷达作为一种新兴的导航定位传感器,具有精度高、扫描速度快和抗干扰能力强等特点,在果园环境感知中得到了越来越广泛的应用^[13-15]。

本文提出一种果园机器人导航技术,基于三维激光雷达和惯性测量单元 (Inertial measurement unit, IMU) 等传感器,采集果园三维点云数据,利用

改进的随机采样一致性 (Random sample consensus, RANSAC) 算法,拟合推算得到导航线。根据三维点云信息,提出换行策略,实现换行作业。基于果园机器人的差速转向结构和纯追踪模型进行轨迹跟踪,实现果园机器人在非结构化环境下的自主导航。

1 果园导航方法

首先使用 16 线激光雷达采集果园点云数据,在经过点云滤波处理后,降低了点云密度。然后进行地面点云的分割,去除地面点云后通过改进的 RANSAC 拟合出果树行直线,根据平行直线之间的关系,进一步推算得到导航线,结合横向偏差、航向偏差以及激光雷达 IMU 融合定位数据,通过纯跟踪控制算法调整机器人位姿。在行驶到行末触发换行判定程序,确定机器人是否需要换行,完成换行后继续执行果树行内导航程序。如此,便实现果园机器人的自主导航,主要导航技术框架如图 1 所示。

1.1 果树行内直线行驶

果树行直线提取的方法通常是在获取当前帧的果树点云后,将该帧点云投影到世界坐标系中,然后将该帧点云分割成左右两树行的点云集,再从左右树行点云集中拟合出两侧果树行的直线,常用的获取方法有 RANSAC 算法和最小二乘法 (Least squares method, LSM)。

LSM 算法是一种被普遍使用的数学优化方法,该方法通过最小误差平方和来求解点云数据中的最佳函数参数^[16]。采用 LSM 算法拟合果树行直线,

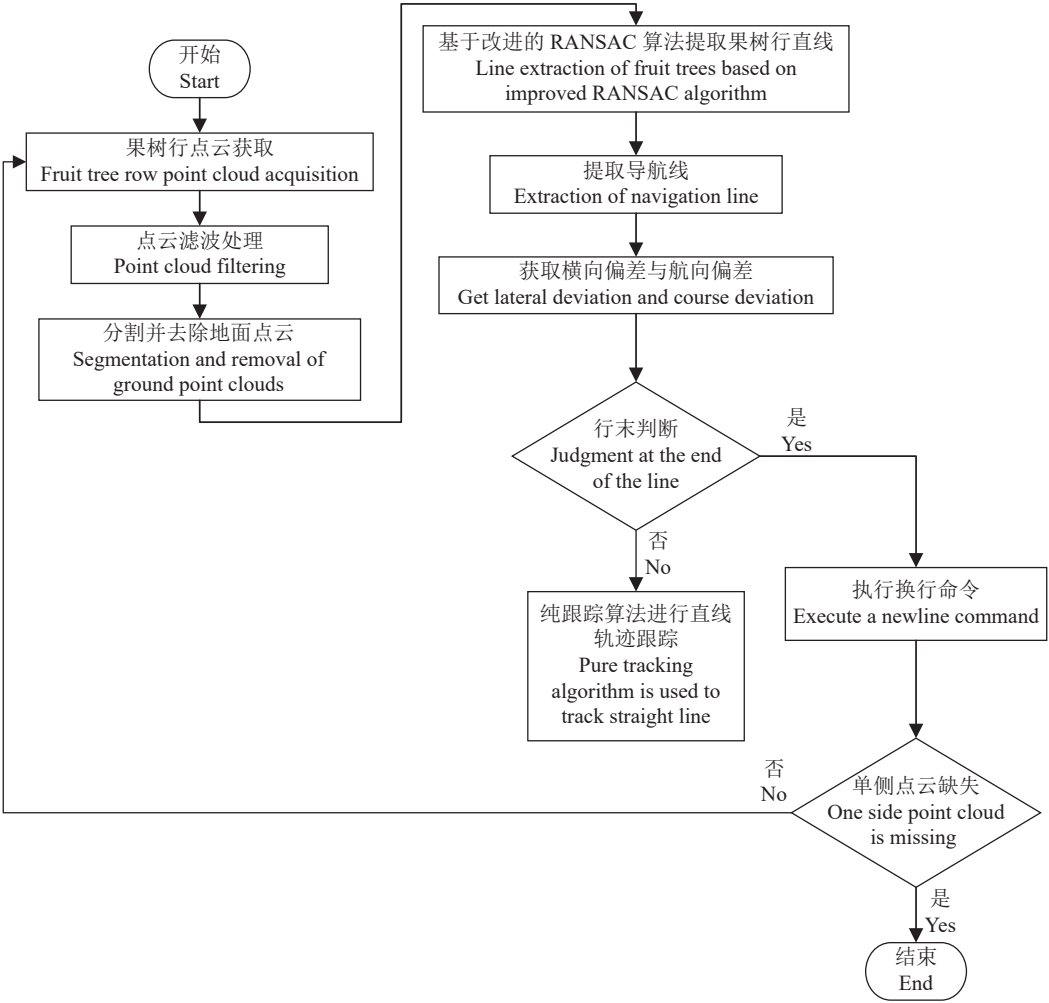


图 1 果园机器人导航技术框架

Fig. 1 Framework of orchard robot navigation technology

当点云坐标满足 $f(Q)$ 取到最小值, 即:

$$f(Q) = \min (X'Q - Y)^T (X'Q - Y), \quad (1)$$

式中, $Q=[k \ b]^T$ 为果树行直线的参数矩阵, k 为果树行直线的斜率, b 为果树行直线的截距; X' 为点云 X 轴坐标值组成的矩阵, Y' 为点云 Y 轴坐标值组成的矩阵。

1981 年有学者提出 RANSAC 算法, 最初用于解决定位的问题, 用迭代法求解含有局部点的观测数据的数学模型参数^[17-18]。RANSAC 算法通过对观测值随机抽样估算参数, 该算法的简要步骤为:

- 1) 随机从数据集中选取 n 个样本数据, 建立参数模型 M ;
- 2) 计算数据集中所有数据与参数模型 M 的误差, 若误差小于阈值, 加入内点集;
- 3) 如果当前内点集元素个数大于最优内点集, 则更新最优内点集, 同时更新迭代次数 h , h 的表达式为:

$$h = \frac{\lg(1 - \eta)}{\lg(1 - \lambda^m)}, \quad (2)$$

式中, η 为置信度, 一般取 0.995, λ 为内点的比例, m 为计算模型所需要的最少样本数;

- 4) 如果迭代次数大于 h , 则退出, 否则迭代次数加 1, 并重复上述步骤。

RANSAC 算法较 LSM 算法具有较大的优势, 尤其是在行距较大的果园中导航精度更高, 鲁棒性更好。本文基于传统 RANSAC 算法, 结合果园机器人的实际应用场景, 提出一种基于点云密度自适应的 RANSAC 算法, 即使存在部分点云缺失的情况, 也能较好地拟合出果树行直线。传统的 RANSAC 算法固定阈值, 在拟合点云较为稀疏的地方可能会出现较大偏差。改进后的 RANSAC 算法具体步骤为:

- 1) 将当前帧的左右两侧果树行点云 ξ_i 进行聚类分割, 并去除地面和杂草点云的干扰, 得到左右两侧点云集合 ξ_{li} 和 ξ_{ri} ;
- 2) 一般果园都是按照平行直线种植, 本文设定

左、右 2 条平行直线分别为 L_l 和 L_r :

$$L_l = k_l x + b_l, \tag{3}$$

$$L_r = k_r x + b_r, \tag{4}$$

式中, k_l 、 k_r 为果树行直线斜率, b_l 、 b_r 为果树行直线截距;

3) 分别从左右两侧点云集合 ξ_{li} 和 ξ_{ri} 中随机选取 2 点, 确定构成果树行直线的参数;

4) 根据点云密度设定不同初始阈值 R_i , 从而分割出内外点, 选取出内点集合, 将最大内点集合的阈值设为最佳阈值 R_{best} , 将其他内点集合的阈值更新为 R_{best} , 进而确定与直线 L_i 的几何距离不大于 R_{best} 的点云集合 $\xi(L_i)$, 即为 L 的最大一致集合。设定内点 (x_i, y_i) , 拟合直线 $Ax + By + C = 0$, 则有:

$$d_i = \frac{|Ax_i + By_i + C|}{\sqrt{A^2 + B^2}}, \tag{5}$$

$$d_i \leq R_i, \tag{6}$$

$$R_{best} = \max \xi_i \{R_1, R_2, \dots, R_i\}; \tag{7}$$

5) 重复 n 次随机选择, 得到直线 $L_1, L_2, \dots, L_n$ 和相应的一致集合 $\xi(L_1), \xi(L_2), \dots, \xi(L_n)$;

6) 确定最大一致集合, 将最大一致集合中的点拟合成一条直线即为最佳直线 L , 通过 ξ_{li} 和 ξ_{ri} 获取的最佳直线分别为 L_l 和 L_r , 即获取到最佳斜率 k_l 和 k_r , 最佳截距 b_l 和 b_r ;

7) 在获取到左右两侧果树行最佳直线后, 根据平行直线关系, 进一步提取出导航线 L_c , 果园机器人通过纯跟踪算法沿导航线行驶, 完成果树行内的自主导航。其中, L_c 的计算公式为:

$$L_c = kx + \frac{b_l + b_r}{2}. \tag{8}$$

由于 L_l 、 L_r 和 L_c 为三条平行直线, 故 $k=k_l=k_r$ 。

1.2 行末换行策略

一般果园的树行两端的特征很明显, 即行首和行末都不存在果树, 因此存在一定的转向空间。当激光雷达检测到果园机器人两侧大片点云数据缺失的时候, 可以判定机器人已到达换行区域, 工控机下发换行指令, 给定机器人底盘特定的速度指令, 保证每次换行转弯动作的准确性, 默认左转, 可根据果园实际情况进行调整。完成换行动作后, 继续向前行驶一段距离 d_{ts} , 确保其完全进入到下一行, 然后继续执行树行跟踪的指令, 具体过程如图 2。

设定转弯半径为 R_t , 换行时的路径为 S_{th} , 则

$$S_{th} = \pi R_t + d_{ts}. \tag{9}$$

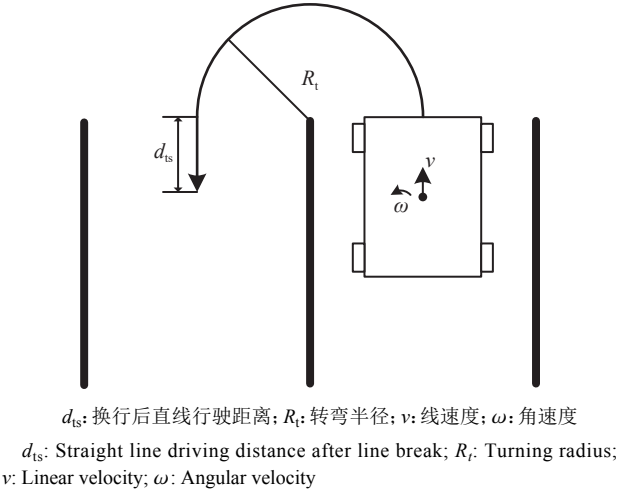


图 2 果园机器人换行策略

Fig. 2 Line breaking strategy of orchard robot

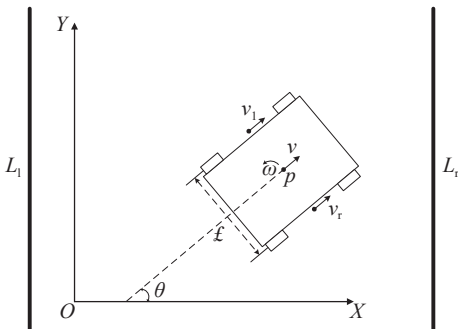
2 运动控制

2.1 基本运动模型

本试验采用的是差速转向机器人底盘, 同侧两车轮通过链传动共用一个电机来驱动, 故同侧前后两车轮转速大小相等。可直接接收来自工控机发送的线速度 v 和角速度 ω (主要是指 x 轴方向上的线速度和 z 轴方向上的角速度), 实现机器人的各种运动。图 3 中 XOY 为建立的大地坐标系, v 为机器人瞬时线速度, 机器人正前方为正; v_r 为右侧轮速度; v_l 为左侧轮速度; ω 为瞬时角速度, 逆时针方向为正; θ 为姿态角; ℓ 为轮距; p 为机器人中心位置; L_l 、 L_r 为左、右侧果树行; 机器人的位姿可用矢量 $P=[x, y, \theta]^T$ 表示。其中:

$$v = \frac{v_l + v_r}{2}. \tag{10}$$

机器人主要有 3 种运动状态:



v : 线速度; v_r : 右侧轮速度; v_l : 左侧轮速度; ω : 角速度; θ : 姿态角; ℓ : 轮距; p : 机器人中心位置; L_l 、 L_r : 左、右两侧果树行
 v : Linear velocity; v_r : Right wheels speed; v_l : Left wheels speed; ω : Angular velocity; θ : Attitude angle; ℓ : Wheelspan; p : Center position of robot; L_l 、 L_r : Fruit tree rows on the left and right sides

图 3 运动模型

Fig. 3 Motion model

1) $v_l \neq v_r$, 方向相同且均不为 0 时, 机器人做圆周运动, 此时

$$\omega = \frac{v_r - v_l}{l}; \quad (11)$$

2) $v_l = -v_r$ 时, 机器人原地转动, 此时 $v = 0$;

3) $v_r = v_l$ 时, 机器人直线行驶, 此时 $\omega = 0$ 。

在驱动轮与地面接触运动为纯滚动的情况下, 机器人的运动学模型可以表示为:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{l} & -\frac{1}{l} \end{bmatrix} \begin{bmatrix} v_r \\ v_l \end{bmatrix}. \quad (12)$$

2.2 融合定位

IMU 以 100 Hz 的频率输出加速度和角速度数据, 激光雷达以 10 Hz 的频率输出点云数据。每收到一帧点云数据, 系统首先扩增一个代表当前激光雷达位姿初始估计的状态向量, 然后使用优化后的点云观测模型对特征进行提取和跟踪; 每收到一帧 IMU 数据, 根据运动模型对 IMU 状态进行估计, 同时更新系统状态的协方差矩阵; 最终通过扩展卡尔曼滤波算法融合 IMU 与激光雷达的观测数据更新位姿状态, 以 100 Hz 的频率输出结果。相邻测量时刻间的状态量定义如下:

$$\begin{cases} S = [S_{u0}, S_{u1}, \dots, S_{ui}], \\ S_{ui} = [p_{wui}, v_{wui}, q_{wui}, u_a, u_g], \end{cases} \quad (13)$$

式中: w 表示世界坐标系; u 表示 IMU 坐标系; i 表示测量时刻; S 为系统状态量; S_{ui} 为 t 时刻 IMU 状态量, 包括相对于世界坐标系的平移、速度和旋转, 分别以 p 、 v 和四元数 q 表示; u_a 和 u_g 分别为加速度计和陀螺仪的零偏。

本文提出的以激光雷达为主, 融合 IMU 信息的定位方案, 可实现果园机器人的厘米级定位, 对非结构化环境下的导航十分重要。

2.3 轨迹跟踪

果园机器人通过纯跟踪算法结合传感器数据确定前视距离, 最终将机器人位姿信息上传至工控机, 通过对比当前帧与前一帧位姿变化, 确定横向偏差和航向偏差是否超过设定阈值, 工控机下发速度指令给单片机, 驱动左右两侧电机, 保证机器人能够按照预定导航线轨迹行驶, 完成树行跟踪的目标。

纯追踪算法是一种模拟人类驾驶习惯的几何追踪模型, 根据车辆的行驶速度和路径信息设定轨迹上的一个路径点, 计算车辆当前位置距离预设目标点所需转过的角度, 控制车辆向路径点行驶^[19]。

具体跟踪方法是: 在目标轨迹 K 上选择一个点 B 作为跟踪点, 则机器人从当前位置到达点 B 的轨迹可以视为一段半径为 R 的圆弧轨迹, 对点 B 跟踪的关键在于对圆弧轨迹半径 R 的求解, 图 4 中 α 为航向偏差, f 为前视距离, d 为横向偏差。综合各参数, 可计算得到轨迹半径 R 。

具体计算步骤为:

$$\frac{f}{\sin(2\alpha)} = \frac{R}{\sin\left(\frac{\pi}{2} - \alpha\right)}, \quad (14)$$

$$\frac{f}{2\sin\alpha\cos\alpha} = \frac{R}{\cos\alpha}, \quad (15)$$

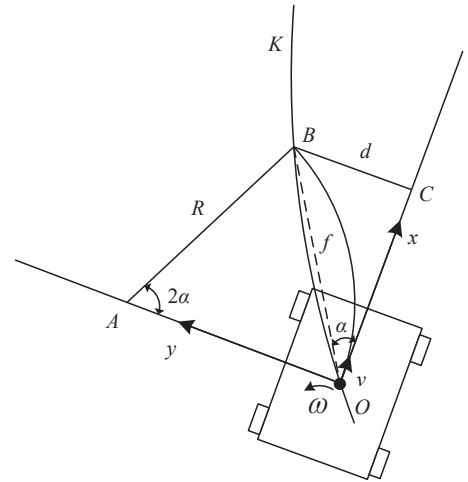
$$R = \frac{f}{2\sin\alpha}, \quad (16)$$

$$\sin\alpha = \frac{d}{f}, \quad (17)$$

$$R = \frac{f^2}{2d}. \quad (18)$$

在求得轨迹半径 R 之后、线速度 v 给定的情况下可以求解出角速度 ω , 计算公式为:

$$\omega = \frac{v}{R}. \quad (19)$$



K : 目标轨迹; B : 目标点; α : 航向偏差; f : 前视距离; d : 横向偏差; A : 圆弧轨迹对应的圆心; R : 轨迹半径; C : OC 边上的垂足; O : 机器人中心位置
 K : Target trajectory; B : Target point; α : Course deviation; f : Forward-looking distance; d : Lateral deviation; A : Center of the arc trajectory; R : Trajectory radius; C : Vertical foot at the edge of OC ; O : Robot center position

图 4 果园机器人轨迹跟踪示意图

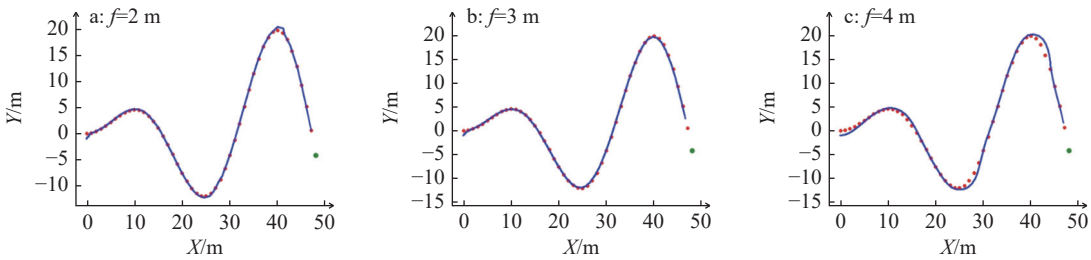
Fig. 4 Orchard robot trajectory tracking diagram

2.4 路径跟踪仿真结果分析

纯追踪算法中前视距离的大小将会直接影响机器人的轨迹跟踪效果, 前视距离较大时, 机器人将会以较小的曲率向路径行驶, 但系统响应较慢, 可能会产生“抄近路”的问题, 影响控制精度; 当

前视距离较小时, 机器人将会以较大曲率向路径行驶, 机器人会因此频繁转向, 产生较大的振荡, 降低机器人的稳定性。因此, 前视距离的大小对导航十分重要。

通过 Python3.5 IDLE 进行路径跟踪仿真, 结合果园机器人差速运动模型和纯跟踪算法, 初始位姿设置为 (0, -1, 0), 前视距离分别设置为 2、3、4 m, 在 0.8 m/s 速度下进行仿真, 结果如图 5 所示。



红色圆点表示目标轨迹点, 绿色圆点表示轨迹终点, 蓝色曲线表示实际轨迹
The red dots represent the target track points, the green dots represent the end of the track, and the blue curve represents the actual track

图 5 不同前视距离 (f) 下路径跟踪仿真结果

Fig. 5 Path tracking results at different forward-viewing distances(f)

从仿真结果中可以看出, 前视距离为 2 m 时, 机器人在曲率较大的路径处的实际运动轨迹较为曲折, 实际运动轨迹和目标轨迹出现明显偏差; 前视距离为 4 m 时, 机器人虽然会以平滑的运动轨迹行驶, 但也出现了实际轨迹与目标轨迹存在较大偏差的问题; 只有当前视距离为 3 m 时, 机器人能够较好地追踪目标轨迹, 故设定机器人前视距离为 3 m。

3 试验验证及结果

3.1 硬件系统组成

图 6a 为果园机器人整体框架, 由机器人底盘、工控机、Velodyne 16 线激光雷达和 9 轴 IMU 组成。机器人长、宽和高分别为 60、45 和 25 cm, 整体采用差速转向的结构设计, 使其在果园中具有良好

的通行能力。机器人左右两侧各安装一个驱动电机, 通过链传动的方式带动四轮转动。图 6b 为果园机器人实物图, 除导航所必须的硬件结构, 还配备了高清摄像头, 可以实时获取果园真实画面。为方便各传感器融合, 定义果园机器人坐标系, 以 IMU 中心为坐标原点, 机器人正前方为 x 轴方向, 平行于车轴且指向左侧的方向为 y 轴正方向, 过坐标原点垂直指向正上方为 z 轴正方向。

3.2 试验方案

为了验证果园机器人自主导航系统的实际效果, 在山东理工大学兰玉彬教授团队与淄博市沂源县山东中以现代智慧农业有限公司共建的智慧化无人果园试验基地进行了试验。该果园果树行距约为 4 m, 果树高约 3.5 m, 行距约 4 m。图 7a 为果园真实环境, 可以看到果树枝叶茂盛, 对 GPS 信号遮

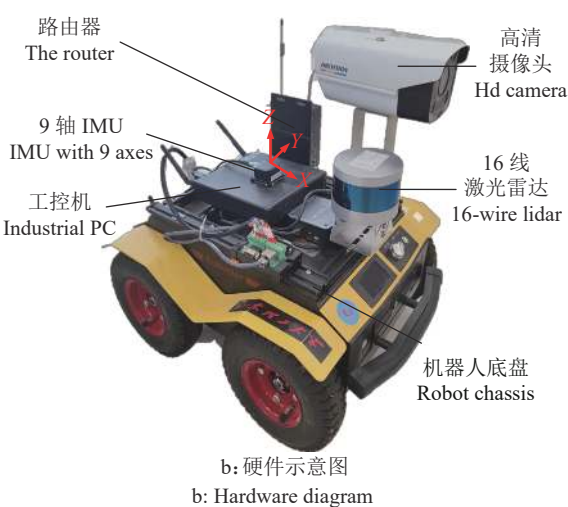
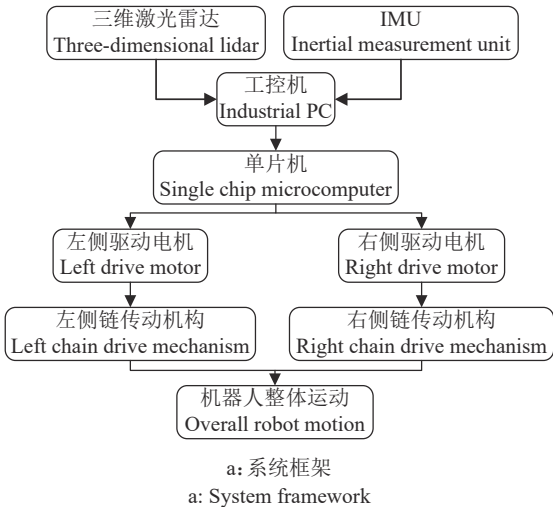
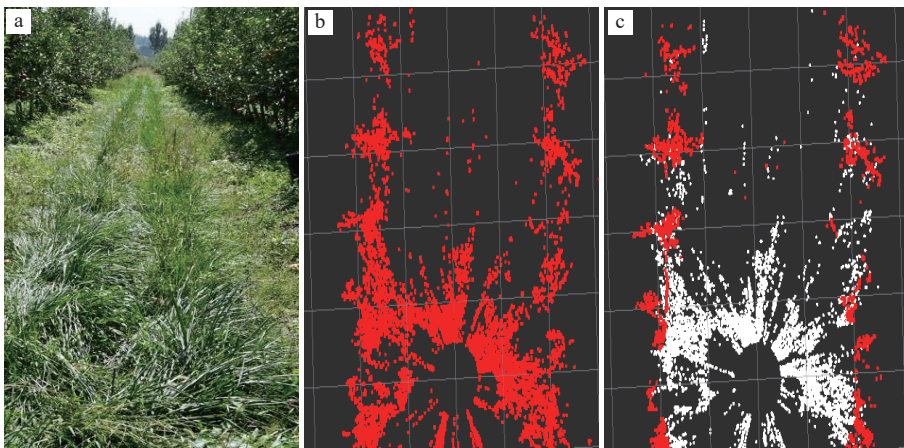


图 6 果园机器人基本组成

Fig. 6 Basic composition of orchard robot



a: 果园真实环境; b: 原始点云; c: 分割地面后的点云, 其中, 白色部分为地面点云, 红色部分为非地面点云, 即果树行点云

a: Real environment of the orchard; b: Original point cloud; c: Point cloud after ground segmentation. In c, the white part is ground point cloud, and the red part is non-ground point cloud, namely, fruit tree row point cloud

图 7 点云预处理
Fig. 7 Point cloud pretreatment

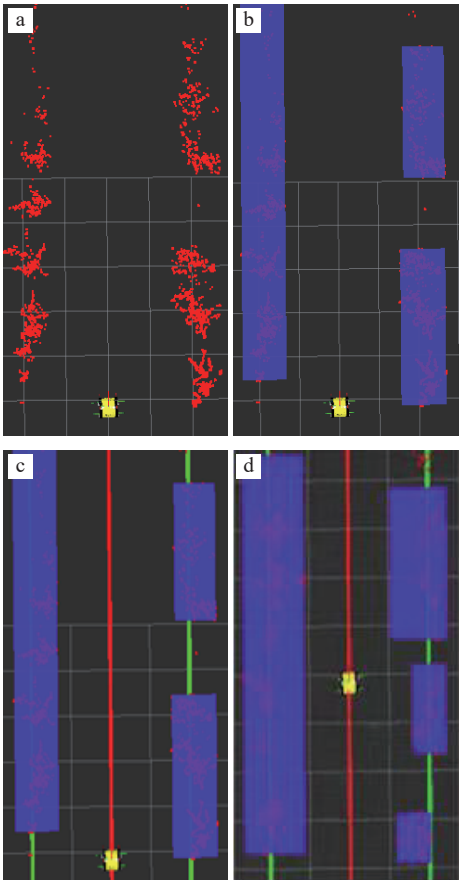
挡十分严重。

通过三维激光雷达获取点云数据 (图 7b) 后, 首先进行预处理降低点云密度, 借助滤波算法去除地面和杂草的干扰, 采用平面模型拟合当前的地面。由于现实的地面并不是一个“完美的”平面, 尤其是在果园这种环境中, 地面非常不平整, 要很好地完成地面分割, 就必须处理存在一定坡度变化的地面情况, 即不能将这种坡度的变化视为非地面, 不能因为坡度的存在而引入噪声。一种简单的处理方法就是沿着 x 方向 (车头的方向) 将空间分割成若干个子平面, 然后对每个子平面使用地面平面拟合算法从而得到能够处理陡坡的地面分割方法。对于给定的点云集合设为 P , 最终将会被分割为地面点云集 P_g 和非地面点云集合 P_{ng} , 如图 7c 所示。

分割出地面点云后, 利用滤波算法将其过滤掉, 处理后的结果如图 8a。通过聚类算法将果树行点云簇进行聚类, 最终会形成一个将点云包裹的三维 Bounding Box, 其主要使用了一种基本的数据结构 KD 树, 它是在一个欧几里德空间中组织点的基本数据结构, 本质上是一个每个节点都为 k 对应维点的二叉树, 聚类后的效果如图 8b。

基于传统的 RANSAC 算法, 提出一种密度自适应的改进方法, 根据不同点云集设置不同初始阈值, 筛选出最大内点集合, 更新并确定最佳阈值, 经过多次迭代, 更高效且更准确地拟合出果树行直线, 根据平行直线关系, 推算导航线 (图 8c)。由图 8d 可以看到, 即使在部分点云缺失和存在离群果树点云的情况下, 该算法还是能较好地推算出导航线。

通过多次试验, 统计果园机器人实际行驶轨迹与导航线之间的横向偏差和航向偏差, 验证本方案



a: 去除地面后的点云; b: 聚类后的点云; c: 生成导航线后的点云; d: 存在离散果树的点云; c、d 中, 绿色直线为拟合得到的果树行直线, 红色直线为推算得到的导航线

a: Point cloud after ground removal; b: Point cloud after clustering; c: Point cloud with the navigation line; d: Point clouds with discrete fruit trees. In c and d, the green line is the fruit tree line obtained by fitting, and the red line is the navigation line obtained by calculation

图 8 拟合导航线
Fig. 8 Fitting navigation lines

的可行性。对比 LSM 算法、传统 RANSAC 算法和改进后的 RANSAC 算法分别在 0.8 和 1.0 m/s 的不

同速度下产生的横向偏差和航向偏差，每隔 10 m 进行一次记录，进行 5 次记录，并取平均值。

3.3 试验结果

综合多次试验结果，对比 LSM 算法、传统 RANSAC 算法和改进 RANSAC 算法，当机器人以 0.8 m/s 的速度行驶，改进后的 RANSAC 算法

产生的横向偏差的最大值和平均值分别为 0.10 m 和 0.084 m，航向偏差的最大值和平均值分别为 1.5°和 1.34°，相较 LSM 算法和传统 RANSAC 算法均为最小，具体结果见表 1。但当机器人以 1.0 m/s 的速度行驶时，各项偏差均明显增大，具体结果见表 2。

表 1 速度 0.8 m/s 下 3 种算法产生的偏差结果
Table 1 The deviation results generated by the three algorithms under the speed of 0.8 m/s

算法 Algorithm	横向偏差/m Lateral deviation	平均横向偏差/m Mean lateral deviation	航向偏差/(°) Course deviation	平均航向偏差/(°) Mean course deviation
LSM算法 LSM algorithm	0.20 0.18 0.23 0.21 0.19	0.202	2.1 1.8 2.2 1.7 2.0	1.96
传统RANSAC算法 Traditional RANSAC algorithm	0.16 0.15 0.13 0.18 0.17	0.158	1.3 1.8 1.5 1.2 1.8	1.52
改进RANSAC算法 Improved RANSAC algorithm	0.08 0.10 0.09 0.07 0.08	0.084	1.5 1.5 1.2 1.3 1.2	1.34

表 2 速度 1.0 m/s 下 3 种算法产生的偏差结果
Table 2 The deviation results generated by the three algorithms under the speed of 1.0 m/s

算法 Algorithm	横向偏差/m Lateral deviation	平均横向偏差/m Mean lateral deviation	航向偏差/(°) Course deviation	平均航向偏差/(°) Mean course deviation
LSM算法 LSM algorithm	0.25 0.28 0.26 0.23 0.24	0.252	2.3 2.5 2.6 2.4 2.7	2.50
传统RANSAC算法 Traditional RANSAC algorithm	0.23 0.22 0.24 0.22 0.21	0.224	2.2 2.4 1.9 2.5 2.6	2.32
改进RANSAC算法 Improved RANSAC algorithm	0.14 0.09 0.08 0.13 0.10	0.108	2.0 1.8 2.1 2.2 2.0	2.02

4 结论

果园机器人通过激光雷达采集点云数据，经过点云降噪、分割地面与非地面点云和果树行点云聚

类等一系列处理后，依靠改进的 RANSAC 算法拟合果树行最佳直线，根据平行直线关系，推算得到导航线。通过纯跟踪算法对果园机器人进行运动控

制, 实现果园机器人的自主导航。基于本文提出的行末换行策略, 实现机器人换行作业。通过咨询果园管理专业人员, 了解到果园机器人在行距为 4 m 的果园进行巡检、除草等作业时, 横向偏差不得大于 0.15 m、航向偏差不得大于 2°。本研究试验结果显示: 当机器人以 0.8 m/s 的速度在果园行驶作业时, 改进后的 RANSAC 算法产生的横向偏差不超过 0.10 m, 航向偏差不超过 1.5°。因此, 本文提出的导航方法基本满足生产要求, 能够极大地提高果园管理的自动化水平, 具有比较重要的现实意义。

未来将在保证定位导航精度的前提下, 稳步提升果园机器人的行驶速度, 提高作业效率。融合视觉算法, 提高果园机器人的环境感知能力, 实现对杂草和障碍物的识别, 进而提高控制精度, 使机器人在非标准化果园也能够正常行驶。

参考文献:

[1] 王祺, 王艳玲, 俄胜哲. 我国果园机械装备现状及发展思路[J]. 农业机械, 2019(1): 109-111.

[2] LAN Y B, CHEN S D. Current status and trends of plant protection UAV and its spraying technology in China[J]. International Journal of Precision Agricultural Aviation, 2018, 1(1): 1-9.

[3] GONG J L, WANG M X, ZHANG Y F, et al. Flow and sound field analysis of agricultural ultrasonic atomizing nozzle[J]. International Journal of Precision Agricultural Aviation, 2019, 2(2): 32-37.

[4] 秦喜田, 刘学峰, 任冬梅, 等. 我国果园生产机械化现状及其发展趋势[J]. 农业装备与车辆工程, 2019, 57(S1): 35-38.

[5] 张漫, 季宇寒, 李世超, 等. 农业机械导航技术研究进展[J]. 农业机械学报, 2020, 51(4): 1-18.

[6] 陈媛媛, 游炯, 幸泽峰, 等. 世界主要国家精准农业发展概况及对中国的发展建议[J]. 农业工程学报, 2021, 37(11): 315-324.

[7] 刘成良, 林洪振, 李彦明, 等. 农业装备智能控制技术研究

现状与发展趋势分析[J]. 农业机械学报, 2020, 51(1): 1-18.

[8] 何勇, 蒋浩, 方慧, 等. 车辆智能障碍物检测方法及其农业应用研究进展[J]. 农业工程学报, 2018, 34(9): 21-32.

[9] 李道亮, 李震. 无人农场系统分析与发展展望[J]. 农业机械学报, 2020, 51(7): 1-12.

[10] 周建军, 周文彬, 刘建东, 等. 果园机器人自动导航技术研究进展[J]. 计算机与数字工程, 2019, 47(3): 571-576.

[11] 李秋洁, 丁旭东, 邓贤. 基于激光雷达的果园行间路径提取与导航[J]. 农业机械学报, 2020, 51(S2): 344-350.

[12] 莫冬炎, 杨尘宇, 黄沛琛, 等. 基于环境感知的果园机器人自主导航技术研究进展[J]. 机电工程技术, 2021, 50(9): 145-150.

[13] BLOK P M, SUH H K, VAN BOHEEMEN K, et al. Autonomous in-row navigation of an orchard robot with a 2D LIDAR scanner and particle filter with a laser-beam model[J]. Journal of Institute of Control, Robotics and Systems, 2018, 24(8): 726-735.

[14] 姬长英, 周俊. 农业机械导航技术发展分析[J]. 农业机械学报, 2014, 45(9): 44-54.

[15] 赛炜, 孙忠涵. 基于激光雷达的机器人智能导航系统研究[J]. 激光杂志, 2019, 40(11): 182-186.

[16] ZHANG H F, HONG Y, QIU J L. An off-policy least square algorithms with eligibility trace based on importance reweighting[J]. Cluster Computing, 2017, 20(4): 3475-3487.

[17] FOTOUHI M, HEKMATIAN H, KASHANI-NEZHAD M A, et al. SC-RANSAC: Spatial consistency on RANSAC[J]. Multimedia Tools and Applications, 2019, 78(7): 9429-9461.

[18] SANGAPPA H K, RAMAKRISHNAN K R. A probabilistic analysis of a common RANSAC heuristic[J]. Machine Vision and Applications, 2019, 30(1): 71-89.

[19] 张华强, 王国栋, 吕云飞, 等. 基于改进纯追踪模型的农机路径跟踪算法研究[J]. 农业机械学报, 2020, 51(9): 18-25.

【责任编辑 李晓卉】