

基于Ros的智产苹果机器人

王冠洲^{*}, 王艳辉, 申雨泽

西京学院计算机学院, 陕西 西安

收稿日期: 2025年2月19日; 录用日期: 2025年3月17日; 发布日期: 2025年3月24日

摘要

随着现代农业的快速发展,对于果园管理的智能化和自动化需求日益增加。特别是在苹果测产领域,传统的人工计数方法不仅耗时耗力,而且容易受到主观因素的影响,导致测产结果不准确。因此,开发一种能够自动、准确地测量苹果数量的智能机器人具有重要的现实意义。本文提出了一种基于ROS(机器人操作系统)和Arduino平台的履带式苹果测产智能机器人。该机器人利用ros_arduino_bridge实现上位机和下位机之间的有效通信,并搭载了激光雷达进行环境感知与导航,同时结合RGB相机和以YOLOv5图像识别算法为核心的目标追踪算法,实现了对单棵苹果树上苹果数量的精确统计。

关键词

ROS, Arduino, YOLO, 识别, 追踪

Intelligent Apple Robot Based on Ros

Guanzhou Wang^{*}, Yanhui Wang, Yuze Shen

School of Computer Science, Xijing University, Xi'an Shaanxi

Received: Feb. 19th, 2025; accepted: Mar. 17th, 2025; published: Mar. 24th, 2025

Abstract

With the rapid development of modern agriculture, the demand for intelligent and automated orchard management is increasingly growing. Especially in the field of apple yield estimation, traditional manual counting methods are not only time-consuming and labor-intensive but also prone to subjective influences, leading to inaccurate yield estimation results. Therefore, developing an intelligent robot capable of automatically and accurately measuring the number of apples holds significant practical importance. This paper proposes a crawler-type intelligent apple yield estimation robot based on ROS (Robot Operating System) and the Arduino platform. The robot utilizes ros_arduino_bridge to achieve effective communication between the upper and lower computers. It is

^{*}第一作者。

equipped with LiDAR for environmental perception and navigation, and integrates an RGB camera with a YOLOv5 image recognition algorithm as the core of its target tracking algorithm, enabling precise counting of apples on individual apple trees.

Keywords

ROS, Arduino, YOLO, Distinguish, Track

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

本文提出了一种基于 ROS (机器人操作系统)和 Arduino 平台的履带式苹果测产智能机器人。该机器人利用 `ros_arduino_bridge` 实现上位机和下位机之间的有效通信，并搭载激光雷达进行环境感知与导航，同时结合 RGB 相机和以 YOLOv5 图像识别算法为核心的目标追踪算法，实现对单棵苹果树上苹果数量的精确统计。

在机器人的设计中，本项目充分利用了 ROS 的模块化特性，将不同功能划分为独立的节点，并通过消息机制实现节点间的通信与协作。Arduino 作为下位机控制器，负责硬件的直接驱动与里程计数据采集，通过 `ros_arduino_bridge` 与上位机进行数据传输与指令交互。履带式的底盘设计使得机器人具有强大的爬坡性，不仅稳定性好，而且能有效保护土壤。激光雷达为机器人提供了实时的环境地图与定位信息，确保机器人在复杂果园环境中能够自主导航并避免碰撞。

RGB 相机作为机器人的视觉传感器，负责捕捉果园的图像信息。本项目采用 YOLOv5 算法对图像进行处理，该算法凭借其高效的实时性与准确性，在苹果识别任务中表现出色。为进一步提高苹果识别的精度和鲁棒性，本研究将针对 YOLOv5 进行优化，结合苹果识别场景特点调整模型架构和参数，以提升其在复杂环境中的表现。同时，基于快速随机树(RRT)算法的改进，我们将详细阐述改进算法的原理、实现细节以及性能评估，确保机器人能够在动态果园环境中实现更加精确的路径规划。

在实验设计方面，本文将设计更为全面的实验方案，收集更详细的实验数据，包括准确率、召回率、运行时间等指标，力求通过充分的数据支撑验证方法的有效性。通过与现有方法的对比，并进行统计分析，我们将进一步证明所提出方法在苹果测产方面的优势。

本文详细阐述了该机器人的设计原理、系统架构、算法实现以及实验验证过程。实验结果表明，该机器人能够显著提高苹果测产的效率与准确性，为现代农业管理提供了一种新的解决方案。本文还探讨了该机器人在未来可能的应用场景与改进方向，为相关领域的研究提供了有价值的参考。

2. 整体设计方案与硬件系统组成

2.1. 平台架构

该平台采用分布式架构，在模式上采用“上位机 + 下位机”的控制模式。Ubuntu 是一种完全基于 Linux 的操作系统，ROS 是机器人编程的系统框架[1]。以树莓派 4B 作为上位机，搭载激光雷达和 RGB 相机，用于苹果检测和环境感知；而下位机则由 Arduino 实现，用于机器人的基本控制。此外，该系统还利用 PC 端作为从机，提供数据查看和分析的接口。平台图如图 1 所示。

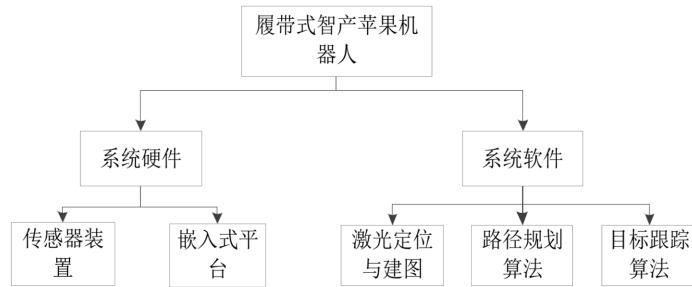


Figure 1. Platform architecture diagram

图 1. 平台架构图

2.2. 系统框架

本设计的核心框架分为可以概括为三部分：应用层、决策层、控制层。应用层以 RGB 摄像头模块为核心，进行苹果识别、追踪，并且可以在上位机进行苹果个数统计实时显示以及最终结果展示。以树莓派 4B 作为机器人的“大脑”，不但可以通过小车运动过程中激光雷达扫描的信息建立成 2D 图像，还可以选取图像中的目标位置使小车自主导航、避障[2]。ArduinoMEGA2560 作为运动控制的核心，其主要作用是控制两个直流减速电机的速度和方向。本系统将树莓派当作主机，PC 端当作从机，既可以在远程桌面上控制并接收数据，也可以在 PC 端控制并接收数据。此外，三个层级之间使用串口进行通信。系统设计框图如图 2 所示。

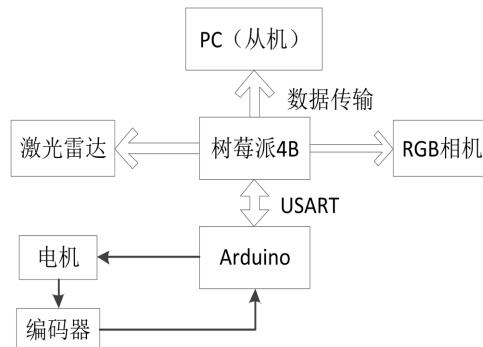


Figure 2. System design block diagram

图 2. 系统设计框图

2.3. 硬件选择

2.3.1. 板载计算机选取

为了满足智能苹果测产机器人对数据处理能力和实时性的高要求，本项目选择树莓派作为核心控制单元。树莓派作为一款功能强大的微型算机，其小巧的体积和强大的性能使其成为嵌入式系统的理想选择。树派具备强大的计算能力可以高效地处理来自激光雷达和 RGB 相机采集的大量数据。作为通信枢纽，连接着上位机和下位机实现了 Arduino 等硬件与远程服务器之间的数据传输和指令交换由于树莓派支持多种操作系统和编程语言，本项目得以在其上运行 YOLOv5 等复杂的图像识别算法。树莓派 4b 模型图如图 3 所示。

2.3.2. 激光雷达选取

思岚 A1 激光雷达采用了激光三角测距技术，结合自主研发的高速视觉采集处理机构，每秒能进行高

达 8000 次以上的测距动作。这一高性能的测距能力确保了机器人能够实时、准确地获取苹果树及其周围环境的详细数据，为后续的苹果识别和数量统计提供了坚实的基础。具备 360°全方位激光扫描功能，其测距核心顺时针旋转，实现了对周围环境的无死角扫描。这一特点使得机器人能够全面、完整地构建苹果树的轮廓图，并准确地识别出每一个苹果的位置和数量。思岚 A1 激光雷达作为核心传感器之一，与 RGB 相机等其他传感器协同工作共同完成了对苹果树及其周围环境的感知和建图。通过与 ROS 操作系统的集成，机器人能够实时接收并处理来自各个传感器的数据，实现自主构建地图、实时路径规划与自动避开障碍物等功能。思岚 A1 激光雷达模型图如图 4 所示。



Figure 3. Raspberry Pi 4b model diagram
图 3. 树莓派 4b 模型图



Figure 4. Silan A1 lidar model diagram
图 4. 思岚 A1 激光雷达模型图

2.3.3. 相机模块选取



Figure 5. RGB camera module model diagram
图 5. RGB 相机模块模型图

为了实现智能苹果测产机器人对苹果树图像的清晰捕捉和精确分析，本项目精心选取了高性能的

RGB 相机模块作为视觉系统的核心。该相机模块具有高清分辨率，能够捕捉到苹果树图像的细节信息，为后续的图像识别算法提供丰富的数据支持。高帧率保证了相机模块能够实时地捕捉到苹果树的动态变化，确保机器人在移动过程中也能获取到稳定的图像数据。考虑到果园环境的光照变化较大本项目选取了具有强光抑制和弱光增强功能的相机模块，确保在不同光照条件下都能获取到清晰的图像数据。RGB 相机模块模型图如图 5 所示。

2.3.4. 单片机选取

在构建智能苹果测产机器人的过程中，本项目选择了 Arduino 作为下位机控制系统的中心。通过 Arduino 的 PWM 输出和 GPIO 引脚实现了对履带式小车电机的精确控制，包括前进、后退、转向等动作。Arduino 通过 ros_arduino_bridge 与上位机进行通信，实现了对机器人状态的实时监控和远程控制。同时，它还能够与其他设备(如 GPS 模块、无线通信模块等)进行通信，扩展机器人的功能。由于 Arduino 具有较快的处理速度和较低的延迟，它能够在短时间内对传感器数据进行处理并作出相应的响应，确保机器人在复杂环境中能够稳定、安全地运行。Arduino 模型图如图 6 所示。

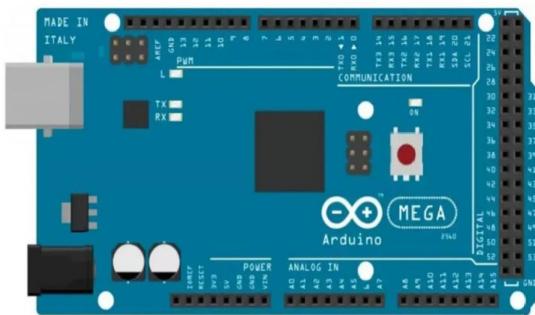


Figure 6. Model diagram of microcontroller
图 6. 单片机模型图

3. 算法设计

3.1. 路径规划算法

路径规划是指根据已知的环境地图等条件，在满足特定条件的情况下，规划出一条从起点到终点的无碰撞路径[3]。路径规划算法的性能优劣对于机器人能否顺利完成任务有较大的影响。而在路径规划算法中又分为了全局路径规划和局部路径规划。它们通常被结合使用以应对复杂的环境和任务需求。全局路径规划为机器人或车辆提供了一个大致的行驶方向和路径，而局部路径规划则根据实时感知到的环境信息来调整和优化路径。常见的路径规划算法包括 Dijkstra 算法、A*算法、RRT 算法等。

本次实验采取了一种新的 RRT*改进算法[4]。在算法开始时设置概率阈值初始值为 p_{ini} 。当随机树扩展到障碍物附近时，则认为其已陷入局部极小值，此时通过式(1)计算 p_{bias} 的值。随着采样点落入障碍物次数的增加， p_{bias} 的值会逐渐减小。此时的采样策略会增加随机树逃脱障碍物附近区域的概率，也就会更快的跳出局部极小值。

$$p_{bias} = \frac{p_{ini}}{2^n} \quad (1)$$

式中： p_{bias} 为概率阈值， p_{ini} 为概率阈值的初始值， n 为上一次采样成功至此次采样期间，采样失败的次数。

在改进的 RRT 算法中，路径生成过程如下所示。

3.1.1. 节点采样与选择

节点采样过程采用不同的策略，像 RRT*采用基于最近邻的采样策略来进行树的扩展。

3.1.2. 障碍检测

所有节点的扩展都需要进行碰撞检测，确保扩展的路径不与障碍物相交。对于复杂的环境，可以使用栅格地图或快速的几何检测算法进行碰撞检测。

3.1.3. 路径平滑化与优化

当路径生成后，可以通过优化算法(如梯度下降或平滑滤波)进一步减少路径的弯曲度，提升路径的平滑性和自然性。

我们先后采用了 RRT 算法和 RRT*算法在相同的环境中进行路径规划实验，得出的结论见表 1。

Table 1. Comparison between RRT and RRT* algorithms

表 1. RRT 与 RRT* 算法对比

指标	RRT*	RRT	提升幅度
路径长度	15.3 m	18.5 m	-17.3%
计算时间	0.58 s	0.75 s	-22.7%
路径平滑度	85%	72%	+13%
碰撞检测率	0%	2%	-2%

由上表可得 RRT*相比传统 RRT 在路径长度上有了显著降低，计算时间也有所优化，同时路径更加平滑，碰撞率几乎为零。在动态障碍物的影响下，RRT*能够更有效地调整路径，避免碰撞，展现了更高的可靠性和适应性。

3.2. 目标识别与追踪算法

对于苹果的识别与追踪，本项目采用了 YOLOv5 与 DeepSORT 算法的组合。YOLOv5 是一种高效且精准的目标检测方法，它通过优化的网络结构、自适应锚框、Mosaic 数据增强等技术，在保证实时处理速度的同时，提升了检测精度和鲁棒性。此外，YOLOv5 的轻量化设计使其能在资源受限的设备上高效运行，而活跃的社区持续为其带来新特性和改进，使其在目标检测领域具有广泛的应用潜力。

为提升 YOLOv5 在苹果识别任务中的性能，本研究从数据增强、网络结构优化、锚框调整及训练调优等方面进行了一系列优化措施。

3.2.1. 数据增强

为了增加模型的鲁棒性，我们对训练数据集进行了多种数据增强。具体包括随机裁剪、旋转、翻转、亮度调整、高斯噪声、模糊以及对比度变化等技术。这些增强方法有助于模型应对不同的苹果颜色、形状、尺寸和环境光照条件。此外，针对苹果识别任务的背景差异，增强了多样化的场景变化，以提升模型对不同环境的适应能力。

3.2.2. 网络结构优化

YOLOv5 的网络结构在本研究中得到了优化。首先，我们引入了 CSPNet (Cross-Stage Partial Network)，以提升模型的表达能力和梯度传播效果，从而提高精度并减少计算量。其次，通过增加模型深度和使用 Focus 模块，进一步改进了特征提取效率。此外，为了增强模型的特征提取能力，骨干网络从标准的 YOLOv5 网络替换为 EfficientNet，以提高苹果的检测精度。

3.2.3. 锚框优化与损失函数改进

为了提升苹果的定位精度，本文采用了 k-means 聚类算法对锚框进行了重新计算。通过优化锚框的大小和比例，进一步确保了锚框与苹果的实际形态更为匹配。此外，在损失函数的设计上，我们引入了 Focal Loss，以解决类别不平衡问题，尤其是在背景区域的干扰下，提高了苹果的检测精度。

我们分别使用 COCO 数据集和 VOC2007 数据集在原始的 YOLOv5 算法和改进后的 YOLOv5 算法上进行比较，得出的结果如表 2、表 3 所示。

Table 2. COCO dataset comparison

表 2. COCO 数据集对比

模型版本	mAP (COCO)	推理时间(秒/图像)	FPS (图像/秒)	模型大小(MB)	F1-score
原始 YOLOv5	37.5	0.2	5	180	0.735
改进 YOLOv5	42.1	0.18	5.5	200	0.755

Table 3. Comparison of VOC2007 dataset

表 3. VOC2007 数据集对比

模型版本	mAP (COCO)	推理时间(秒/图像)	FPS (图像/秒)	模型大小(MB)	F1-score
原始 YOLOv5	75.3	0.1	10	150	0.78
改进 YOLOv5	78.9	0.09	11.1	170	0.81

通过以上对比可以得出改进版 YOLOv5 在精度和推理速度上都有显著提升。

YOLOv5 本身并不提供追踪功能，但可以与 DeepSORT 追踪算法结合使用[5]。DeepSORT 利用卡尔曼滤波器和匈牙利算法进行目标跟踪。卡尔曼滤波器预测目标在下一帧中的位置，而匈牙利算法则负责将检测到的目标与预测的目标进行匹配。为了提高匹配的准确性，DeepSORT 引入了深度学习特征[6]。它使用卷积神经网络提取目标的深度特征，并计算目标之间的相似度。该算法综合考虑目标的位置信息和深度特征，从而实现更精确的目标追踪。目标识别与追踪原理图如图 7 所示。

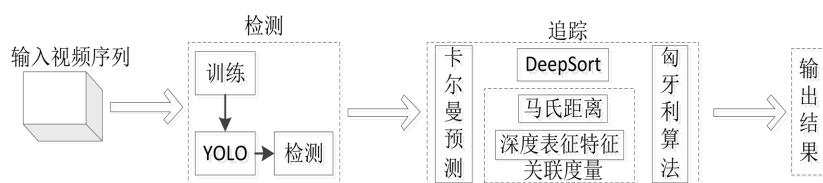


Figure 7. Principle diagram of target recognition and tracking

图 7. 目标识别与追踪原理图

4. 地图构建与定位

4.1. 地图构建

地图构建，也称为 SLAM (定位与地图构建)，是机器人通过自身传感器如激光雷达、摄像头等实时感知周围环境，并据此构建出环境地图的过程

SLAM 问题可以描述为，移动机器人从开机到 t 时刻一系列传感器测量数据 $z_{1:t}$ 以及一系列控制数据 $U_{1:t}$ 的条件下，同时对地图 m 、机器人位姿 $X_{1:t}$ 进行的估计，显然是一个条件联合概率分布。用以下公式来描述 SLAM 的过程：

$$p(X_{1:t}, m | U_{1:t}, Z_{1:t})$$

gmapping 是 ROS 开源社区中较为常用且比较成熟的 SLAM 算法之一[7]。它基于粒子滤波的方法，能够根据移动机器人里程计数据和激光雷达数据来绘制二维的栅格地图。在 Rviz 中，gmapping 算法可以通过订阅相应的传感器话题并发布地图话题来实现地图的构建和显示。我们利用 gmapping 算法实现了果园环境的地图构建，结合机器人所配备的激光雷达传感器和同步定位与建图技术，成功地生成了一个高效的环境地图。gmapping 基于粒子滤波原理，通过处理激光雷达数据和机器人的运动轨迹，不断优化机器人的位置估计并生成二维地图。其核心过程包括数据采集、粒子滤波和地图生成，其中激光雷达扫描数据提供环境信息，粒子滤波估算机器人的位置，而地图则展示了墙体和障碍物等静态元素。为了实时观察地图的构建过程，我们使用了 ROS 中的 Rviz 可视化工具，它能够显示机器人传感器数据、运动轨迹和构建的地图。通过启动 gmapping 节点，我们输入激光雷达扫描数据和机器人的位姿信息，配置 Rviz 显示地图和激光扫描数据，实时查看地图构建的进展。随着机器人在果园中行进，gmapping 算法根据传感器数据不断更新地图，Rviz 也实时展示机器人的运动路径和地图的逐步完善。为了进一步提高地图的精度，我们根据实际情况对 gmapping 的参数进行了优化，如调整粒子数量、更新频率和地图分辨率等。

4.2. 定位

为了实现机器人在果园环境中的精确定位，我们采用了 AMCL (Adaptive Monte Carlo Localization) 算法。AMCL 基于粒子滤波技术，能够在已知地图的基础上估算机器人在复杂环境中的位置和姿态，特别适用于动态和非结构化的场景[8]。

AMCL (自适应蒙特卡洛定位) 算法的工作流程如图 8 所呈现，涵盖了从初始化到定位估计的各个关键步骤。

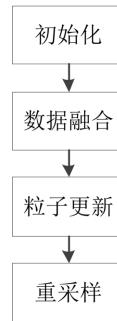


Figure 8. AMCL algorithm flowchart
图 8. AMCL 算法流程图

首先，在初始化阶段，系统生成多个粒子，这些粒子代表可能的机器人位置和朝向。接着，数据融合步骤中，机器人通过传感器(如激光雷达)收集环境数据，并与已知地图进行比对，计算每个粒子的权重。接下来，粒子更新阶段，AMCL 利用机器人运动模型和传感器数据更新粒子的位置，并根据匹配结果调整其权重。为了避免粒子退化，系统会进行重采样，通过根据粒子权重重新生成新的粒子群来保持定位精度。最后，经过不断的粒子更新和重采样，AMCL 能够提供机器人的精确位姿估计[9]。

在果园这种环境中，由于传统的 GPS 定位技术无法提供足够的精度，因此我们选择了 AMCL 作为定位方案。为此，机器人配备了激光雷达传感器，实时采集周围的环境信息，并生成激光点云数据。AMCL 将这些数据与预先构建的果园地图进行匹配，从而实时估算机器人的位置。为了确保 AMCL 的精确性，我们首先利用激光雷达收集的数据来构建果园的高精度地图，地图中包括了树木、道路等重要元素，经

过多轮数据采集和优化，最终得到了高质量的静态地图。

通过激光雷达与 AMCL 算法的结合，机器人能够在果园中实现高效、精准的定位。激光雷达不断提供实时数据，而 AMCL 则利用这些数据与地图进行匹配，持续更新粒子权重，从而实时调整机器人的位姿，避免了传统定位技术中的误差。在实际应用中，定位精度会受到粒子数量、激光雷达数据质量和地图准确性等因素的影响。调整粒子数量后，AMCL 在大部分果园环境中能够提供稳定的定位结果，尤其是在存在大量静态障碍物的区域，算法展现出了极好的稳定性和鲁棒性[10]。

AMCL 算法作为机器人定位的核心模块，与路径规划和目标识别等其他模块紧密协作。系统的整体工作流程包括初步定位、实时跟踪和误差修正，确保机器人在执行任务时保持高精度定位，并能够应对环境的变化和传感器可能出现的问题。

5. 实验测试

为了验证机器人控制系统的可行性，对机器人整机工作流程进行实验测试。机器人在收到给定苹果树的目标位置指令后，机器人会进行起始点到目标点的路径规划并会在出现实时障碍物时自动避开，当到达目标位置时进行苹果的识别与追踪，最终完成计数。

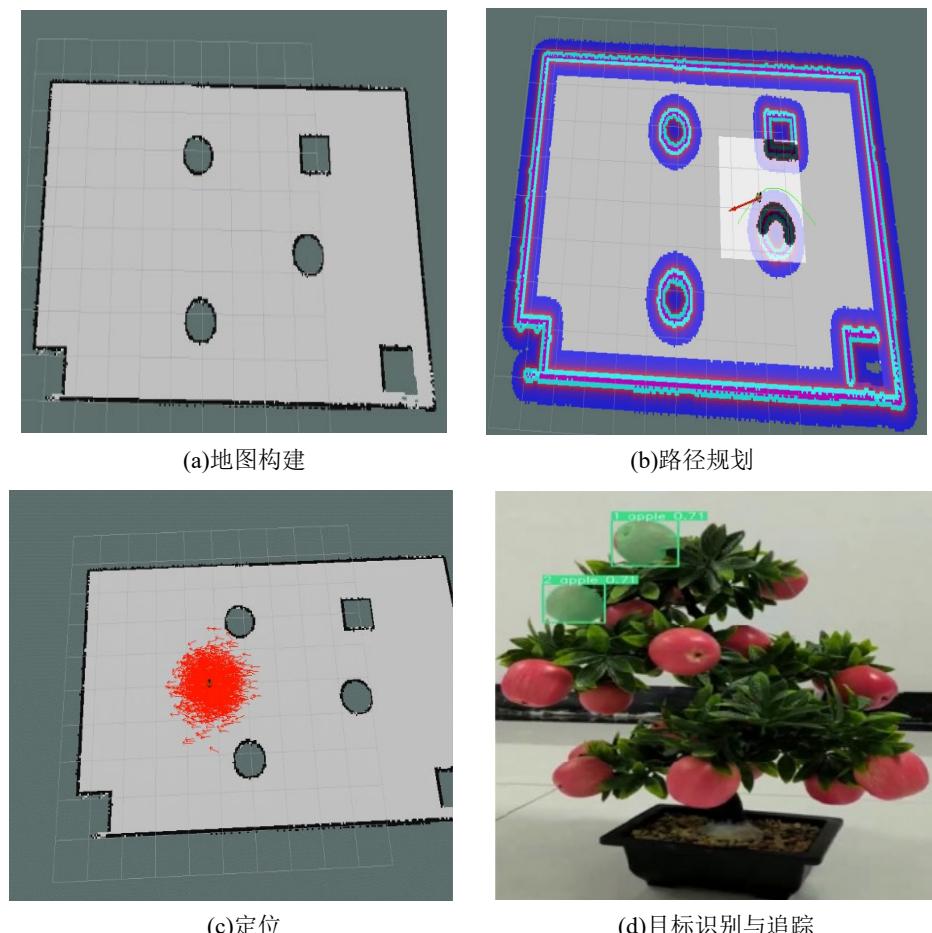


Figure 9. Robot experimental test
图 9. 机器人实验测试

实验结果如下图所示。图 9(a)为机器人建图结果，当建图完成后我们可以得到当前果园的地图信息，

在该地图上我们可以选定果树的位置，使机器人开始运动。图 9(b)为机器人路径规划结果，机器人根据起始位置和目标位置的距离设计出最优的路径，并对运动时出现的实时障碍物进行躲避。图 9(c)为机器人定位过程，红色箭头越密集的地方代表机器人此时在该位置的机率越大，反之则越小。图 9(d)为机器人目标识别与追踪过程，将识别出的苹果用绿色框标定，并在该苹果顶部显示当前计数的数量。经实验该机器人拥有较强的自主控制能力和对图像的识别能力。

6. 未来展望

本文介绍了基于 ROS 的履带式苹果测产智能机器人。该系统是实现智能机器人在果园中自主导航、遍历苹果树以及执行苹果数量统计任务的关键组成部分。在路径规划算法的实现上，本文采用了多种传感器技术，特别是激光雷达和 RGB 相机，来获取果园的实时环境信息。在目标识别与追踪算法实现上，使用 YOLOv5 结合 DeepSORT 的方式来实现苹果的识别与追踪。除此之外，还介绍了 SLAM 在 ROS 系统中的应用以及导航的实现过程。未来将继续致力于履带式苹果测产智能机器人的研发和优化工作，以推动智能农业技术的不断发展和应用。相信，在未来，智能机器人将在农业生产中发挥更加重要的作用，为农业产业的可持续发展贡献更多的力量。

参考文献

- [1] 左培良, 王建平, 郑鑫, 等. 基于 ROS 的移动工业机器人定位算法分析[J]. 现代制造技术与装备, 2024, 60(11): 70-72.
- [2] 于子珺, 刁立强, 曹镇宇, 等. 基于树莓派和 ROS 系统的机器人研究[J]. 物联网技术, 2024, 14(12): 112-114+118.
- [3] Ebert, S., Mey, J., Schöne, R., Götz, S. and Aßmann, U. (2024) Distributed Petri Nets for Model-Driven Verifiable Robotic Applications in Ros. *Innovations in Systems and Software Engineering*, **20**, 531-557. <https://doi.org/10.1007/s11334-024-00570-5>
- [4] 赵桂清, 崔传辉, 高德营, 等. 复杂动态环境下先验知识引导机器人路径规划[J]. 现代制造工程, 2025(1): 50-56.
- [5] 韩泽山. 基于 YOLO 与 DeepSORT 的多目标跟踪算法研究[D]: [硕士学位论文]. 张家口: 河北建筑工程学院, 2024.
- [6] Zhao, S. and Hwang, S. (2024) Complete Coverage Path Planning Scheme for Autonomous Navigation Ros-Based Robots. *ICT Express*, **10**, 83-89. <https://doi.org/10.1016/j.icte.2023.06.009>
- [7] 勒伍杨, 喻俊松, 万生鹏. 基于 ROS 的激光 SLAM 地图优化技术研究[J]. 南昌航空大学学报(自然科学版), 2024, 38(4): 95-102.
- [8] 李宏达. 基于 ROS 的自主导航机器人设计[D]: [硕士学位论文]. 哈尔滨: 哈尔滨师范大学, 2021.
- [9] 李阳. 基于 ROS 系统移动机器人 SLAM 算法研究与实现[D]: [硕士学位论文]. 西安: 西安工程大学, 2018.
- [10] Wang, S. (2023) Research and Design of Intelligent Anti-Epidemic Disinfection Robot Based on ROS. *Journal of Physics: Conference Series*, **2562**, Article ID: 012050. <https://doi.org/10.1088/1742-6596/2562/1/012050>