

SARAL-Bot：用于草莓植物护理的自主机器人

Arif Ahmed Ritvik Agarwal Gaurav Srikar Nathaniel Rose Parikshit Maini
University of Nevada Reno

Abstract—草莓种植需要大量劳动力来监测和保持植物健康。为了解决这个问题，团队 SARAL 开发了一种自主机器人，用于 2024 年 ASABE 学生机器人挑战赛，能够导航、检测不健康的叶子并进行移除。该系统解决了劳动力短缺问题，降低了成本，并通过基于视觉的植物评估支持可持续农业。本研究展示了机器人技术现代化草莓种植和实现可扩展、智能农业解决方案的潜力。

2024 年 ASABE 学生机器人挑战赛旨在模拟草莓种植任务，这为展示机器人在农业中的潜在益处提供了极好的机会。草莓在全球范围内被广泛消费，并且根据联合国粮食及农业组织（FAO）的报道，全球草莓的产量正在增加。草莓的生产是劳动密集型的，任务如种植、监测植物健康、去除不健康的叶子以及采摘都需要大量的人工努力。这些任务对人类工人来说可能是耗时的、昂贵的且需要体力的。能够自主导航草莓田地、识别健康和 unhealthy 植物部分并准确去除不健康叶子和花的机器人的发展，有潜力彻底改变草莓种植行业。通过自动化这些任务，农民可以：

通过参与 2024 年 ASABE 学生机器人挑战赛，我们团队旨在为开发创新的机器人解决方案做出贡献，以满足草莓种植业的需求。我们的机器人展示了自主系统在提高农业效率、作物健康和可持续性方面的潜力，从而最终使农民、消费者和环境受益。

为了满足草莓市场的需求，显然我们需要增加草莓的产量。最近关于农业的机器人研究报告指出，机器人解决方案可以帮助监测植物并增加草莓的总体产量 [?], [?]。为了获得高产的草莓，重要的是要识别植物是否受到营养不良、疾病和/或土壤 pH 值不正确的影响。有趣的是，仅通过检查叶子的颜色就可以识别出几个因素¹。例如，黄色或苍白的叶子表明由于浇水过多而导致植物受到了水分应激。当叶子开始出现这样的特征时，有必要修剪那些不健康的叶子，以防止症状扩散到其他叶子。同样，有一些症状，如真菌感染、空气流通问题和寒潮问题等。这些问题可以很容易地通过机器人解决方案识别出来 [?]。

我们的团队为 2024 年 ASABE 学生机器人挑战的高级组开发了一种创新解决方案，这个方案利用了现成的机器人 HiWonder ArmPi Pro，并在此基础上增加了定制的硬件和软件。

为了在遵循尺寸限制的同时提高机器人的定位能力，我们在 ArmPi Pro 的后部安装了一个面向前方的 Intel D435i² RGBD 摄像头，并配备了线性执行器。线性执行器使得深度摄像头（即 D435i RGBD 摄像头）拥有更好和更广的视野。此外，这种设计配置还帮助我们使用称为实时外观地图（RTABMAP）[?] 的 VIO 算法执行稳健的视觉惯性里程计（VIO）。

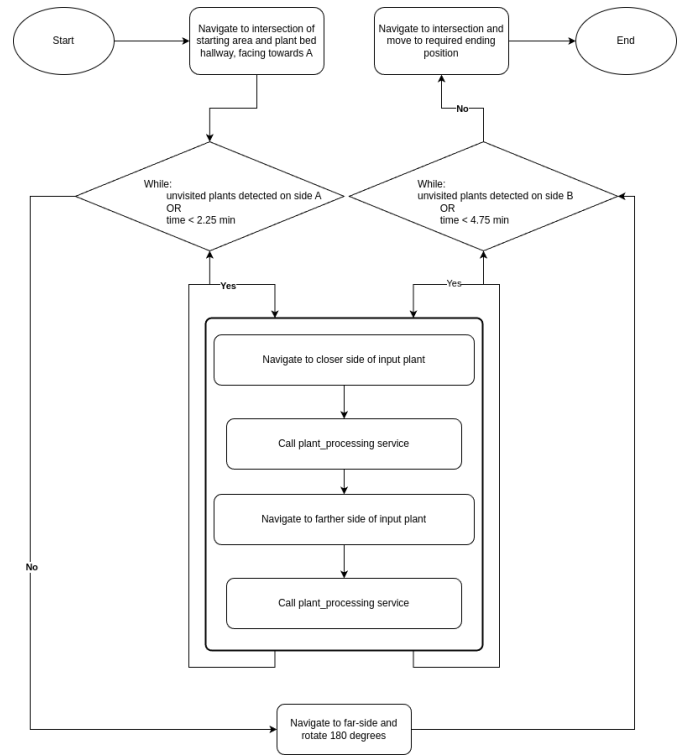


Fig. 1. SARAL-Bot 行为协调逻辑协调器流程图

1) 软件架构：我们的软件栈是基于 ROS2 Humble，一个最新的 LTS 版本，确保了兼容性和稳定性。³ 我们开发了一个模块化架构，它由三个主要系统组成（都在 ROS2 Humble 中）：

- 1) 行为协调器 - 负责任务顺序
- 2) 导航系统 - 负责移动控制
- 3) 植物加工系统——负责植物修剪

这些子系统然后通过我们制作的一个 ROS1 节点与提供的 ArmPi Pro 的控制节点进行通信，该节点通过一个网络服务器暴露出各种 REST API 端点。最后，为了确保可靠且可重复的部署，我们使用 Docker 进行环境管理。⁴

2) 机器人行为和任务策略：机器人的行为主要集中在协调我们的导航和植物处理系统。鉴于竞技场的静态特性，我们通过限制规划为预定义的无障碍直线运动来优化我们的导航策略。

3) 臂控制和逆运动学：为了控制 ArmPi Pro 的底盘和机械臂，我们使用了板载的 Raspberry Pi，运行一个带有

¹<https://www.epicgardening.com/strawberry-problems/>

²<https://www.intelrealsense.com/depth-camera-d435i/>

³<https://docs.ros.org/en/humble/index.html>

⁴<https://www.docker.com/>

ROS Melodic 节点的 Flask 服务器。这使得现有硬件与我们基于 ROS2 的软件栈实现了无缝集成。机械臂的逆运动学是通过 MoveIt 2 处理的，一个强大的运动规划框架。这使得我们的机器人能够准确地到达不同高度和距离的树叶和花朵。

为了检测并区分健康的叶子、不健康的叶子和花朵，我们采用基于 HSV 的颜色阈值法，随后进行轮廓检测。这种计算机视觉技术在不同的光照条件下能够保证稳健的性能。

我们独特的硬件增强组合、模块化软件架构、优化的导航策略以及先进的计算机视觉技术展示了我们团队在应对竞赛挑战中的创新方法。我们的团队确立了明确的设计目标和标准，以指导我们为 2024 年 ASABE 学生机器人挑战赛开发机器人的工作。这些目标是基于对比赛规则、约束和挑战的深入理解，确保我们的机器人适合于有效地执行所需任务。我们的主要设计目标是开发一种能够自主导航场地、检查植物并进行选择性收获的机器人，同时遵循比赛规则中规定的尺寸限制。我们旨在设计一个紧凑高效的机器人，在规定的时间内完成任务。我们为机器人设定了具体的性能目标，包括：这些定量目标帮助我们集中精力并评估设计的成功。我们的设计目标影响了我们对系统架构和硬件组件的选择。我们选择 ArmPi Pro 作为基础机器人，以实现全方位导航能力，并在线性执行器上添加了 Intel D435i RGBD 摄像头，以增强定位和物体检测。我们选择了 ROS2 Humble 作为软件框架，因为它具有稳定性、模块化和社区支持。

4) 模块化、可扩展性和易用性：在我们的设计中，我们优先考虑模块化和可扩展性，以便随着项目的推进轻松集成新功能和算法。我们还旨在创建一个用户友好的系统，使我们的团队成员能够轻松设置、操作和维护该系统。

5) 可靠性与鲁棒性：确保我们机器人的可靠性和稳健性是一个关键的设计目标。我们的目标是开发一个能够稳定执行任务并优雅地处理意外情况的系统。为此，我们对硬件和软件组件进行了广泛的测试和验证。例如，为了在遵循预算的前提下增加我们系统的计算能力，我们决定采用树莓派的分布式网络，而不是单个高端计算机。

6) 成本和资源优化：最后，我们力求在整个设计过程中优化成本和资源利用。我们精心选择了在性能和性价比之间取得平衡的组件，并利用开源软件和库来减少开发时间和成本。通过定义明确和具体的设计目标和标准，我们为机器人开发建立了坚实的基础，确保它能够有效应对 2024 年 ASABE 学生机器人挑战赛的挑战。

I. 零件清单和表格

在材料清单的表格 I 中，我们展示了部件清单和价格。该表对组件进行了分类（机器人、计算资源、传感器、电源、3D 打印部件和其他杂项物品），并提供了每个类别中每个组件的详细信息。我们还强调，机器人使用的计算资源总量符合所规定的控制组件预算，即 \$ 500.00。

II. 硬件描述

A. 底盘和机械臂：HiWonder Arm Pi Pro

ArmpiPro 底盘完全由硬铝合金制造，以确保其耐用且轻便。底盘宽 256mm、长 298mm、高 148mm，符合比赛的尺寸限制。底盘上有多个螺纹孔，以便于定制和简化线路管理。SARAL Bot 的底盘通过定制的 3D 打印后部进行定制，以稳定线性执行器，同时增加额外组件的空间。

Item	Cost Per Unit	
HiWonder Arm Pi Pro	\$ 529.00	
1 × Aluminum alloy chassis (256mm × 298mm)		
4 × Aluminum mecanum wheels (50mm × 100mm)		
1 × LX-225 servo (included in ArmPi Pro)		
3 × LX-15D servo (included in ArmPi Pro)		
1 × ID 1 servo (included in ArmPi Pro)		
1 × Gripper (included in ArmPi Pro)		
4 × Motor encoders		
1 × Arm Pi Pro 7.4V lithium ion battery (6000mAh)		
1 × Raspberry Pi 4 (included in ArmPi Pro)		
Added Computational Resources		
2 × Raspberry Pi 5	\$ 90.00	
1 × TP-Link 5 port Ethernet switch	\$ 18.00	
Sensors		
2 × Intel D435i depth camera	\$ 327.00	
Power Suppliers		
1 × KBT 12V lithium ion battery (2400mAh)	\$ 23.00	
1 × Krisdonia 5V power bank (25000mAh)	\$ 89.50	
Added Hardware		
1 × 12V Linear actuator(180mm length w/125mm stroke)	\$ 28.00	
1 × 12V 2-Channel Relay Switch	\$ 8.99	
Custom 3D Printed Parts		
1 × 3D printed chassis back (225.5mm × 112mm × 140mm)	\$ 18.80	
1 × 3D printed custom gripper (100mm length)	\$ 0.94	
TOTAL COMPUTE RESOURCES		\$ 260.00

TABLE I
物料清单及价格

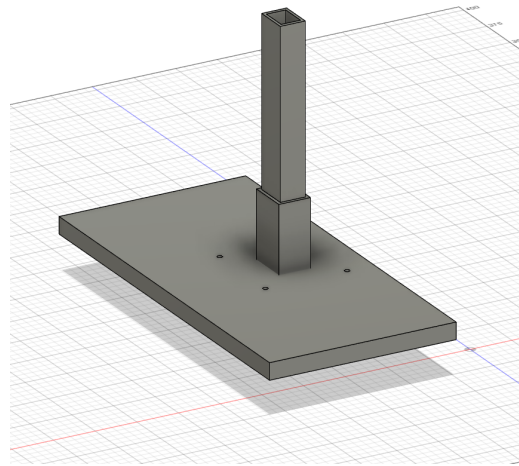


Fig. 2. 定制底盘背板：该定制设计附在我们铝制底盘的背部，以增加我们可用于组件的空间，并稳固地固定线性执行器。定制背板厚 10 毫米，宽 225.50 毫米，长 112.00 毫米。

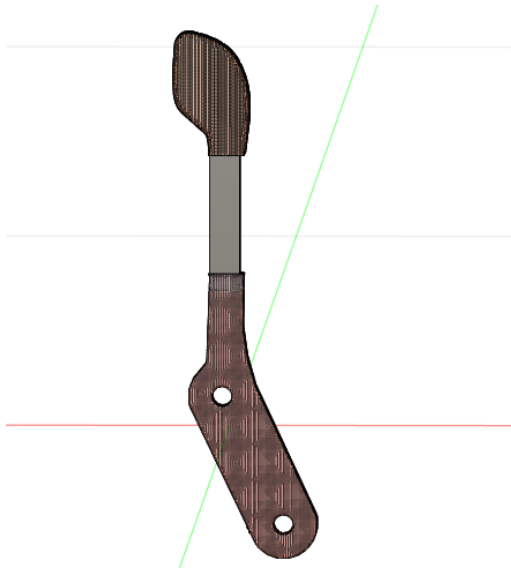


Fig. 3. 定制夹持器：该定制夹持器的设计用于将原始夹持器的长度从 60mm 扩大到 100mm。

为了替换 Arm Pi Pro 配备的初始轮子，选择了新的麦克纳姆轮。这些新轮子由铝合金制造，降低了损坏风险，重量更重，意味着它们为移动平台提供了更好的稳定性，并且配备了全橡胶滚轮，在穿越泥炭苔藓填充的地形时提供更好的牵引力和控制。



Fig. 4. 铝合金麦克纳姆轮：这些新的麦克纳姆轮完全由铝合金制成，比原来的塑料麦克纳姆轮更耐用且更重。它们还使用全橡胶滚轮，提高了抓地力。

安装在我们 saral_bot 上的机械手臂具有 5 个自由度，由伺服电机驱动。最初，手臂中每个关节的方向都被配置为执行俯仰操作，除了第 1 和第 5 关节及末端执行器，它们是偏航方向的。这种配置限制了手臂的伸展能力，特别是在接触位于植物模型后方的不健康叶子时。为了增强其伸展能力，我们改变了第 4 个关节的方向以执行滚动操作。这一调整使手臂能够绕过植物的任一侧延伸，侧向弯曲前臂，并用夹持器准确到达叶子。手臂修改后的方向配置详见表格 II。

ArmPi Pro 机器人的内置夹持器太短，无法在不干扰周

围健康叶子的情况下够到植物后面的叶子。我们开发了一种较长的叉形夹持器，可以在不影响相邻叶片的情况下有效地抓住不健康的叶子。这些夹持器的设计如图 3 所示。机器人的最初臂展为 363 毫米。使用延伸夹持器后，总臂展为 403 毫米。

ArmPi Pro 机器人最初配备了一个内置摄像头，安装在机械臂的第 5 节上。然而，对于此次挑战来说，手眼协作的配置并不理想，还有效地降低了末端执行器的输出功率，因此我们拆除了预装的摄像头。

当不用于操作时，机械臂被置于默认姿态，以避免阻碍线性执行器顶部的感知视角。

我们机器人的计算资源由三个 Raspberry Pi (RPi) 单板计算机组成：两个 RPi 5 型号和一个与 ArmPi Pro 集成摄像头的 RPi 4。这种分布式计算架构使我们能够有效地处理机器人操作所需的各种计算任务。两个 RPi 5 型号专门用于运行我们定制的 ROS2 节点，这些节点负责高级决策、导航和植物处理。这些节点使用 ROS2 的内置通信基础设施进行通信，利用发布-订阅和服务-客户端模式。选择 RPi 5 型号是因为它们的处理能力和 I/O 能力相较于之前的 Raspberry Pi 代更强大，以确保我们的 ROS2 系统在遵循计算预算的同时具有顺畅和敏捷的性能。与 ArmPi Pro 集成的 RPi 4 模型负责运行机器人电机和舵机的低级控制软件。该控制软件基于 ROS1 Melodic，通过使用 Flask 网络服务器实现的自定义 ROS1-ROS2 桥与在 RPi 5 型号上运行的高级 ROS2 节点通信。在第 ?? 节的 ROS1-Flask Server 子节中对此进行了进一步讨论。为了促进三个 RPi 板之间的通信，我们使用了 TP-Link 5 端口以太网交换机。交换机建立了一个局域网 (LAN)，通过我们配置的 IP 连接 RPi，从而使它们能够高效地交换数据和消息。选择有线以太网连接而非无线通信，确保了计算资源之间可靠和低延迟的通信，这对于实时机器人控制和协调至关重要。通过在多个 RPi 板之间分配计算工作负载并建立一个稳固的有线网络用于通信，我们为机器人的机器人创建了强大而灵活的计算基础设施。这种架构使我们能够高效地处理传感器数据，执行复杂计算，并协调机器人的各个子系统，最终使其能够以高精度和可靠性进行导航、检查植物和执行选择性采摘任务。

B. 传感器

我们使用两台英特尔 D435i 深度相机进行视觉感知（见图 ??）。D435i 是一款高性能、低功耗的立体视觉相机，能够提供准确的深度信息。它具有一对深度传感器、一个 RGB 传感器和一个红外投影仪，使其能够在各种光照条件下捕捉高分辨率的 3D 图像。一台相机安装在机器人的后端线性执行器上，启动时会展开。这个位置提供了对植物的更是等距视图，同时允许机械臂功能而不遮挡它们。另一台 RGB 相机安装在前方，使机器人能够进行精确的定位和地图绘制。

此外，车轮上的电机编码器和手臂伺服电机被用于实现所需的输出。使用了三种不同类型的伺服电机：LX-225、LX-15D 和 ID 1 伺服电机。更多关于伺服电机的详细信息也在表格 II 中描述。这些伺服电机均以 0.3° 的精度进行操作。

系统设计使用了三种不同的电源，以满足不同组件的功率需求。第一个电源由 Arm Pi Pro 提供，是一个 7.4V、6000mAh 的锂离子聚合物 (Lipo) 电池，用于为移动平台和机械臂上的 7.4V 电机供电。该 7.4V Lipo 电池还通

Joint ID	Servo Type	Orientation	Torque	Limits (In Radians)
Joint 1	LX- 15 D	Yaw	15kg/cm with 6V 17kg/cm with 7.4V	$-\frac{\pi}{2}$ to $\frac{\pi}{2}$
Joint 2	LX- 225	Pitch	25kg/cm with 7.4V	$-\frac{\pi}{2}$ to $\frac{\pi}{2}$
Joint 3	LX- 15 D	Pitch	15kg/cm with 6V 17kg/cm with 7.4V	$-\frac{\pi}{2}$ to $\frac{\pi}{2}$
Joint 4	LX- 15 D	Roll	15kg/cm with 6V 17kg/cm with 7.4V	$-\frac{\pi}{2}$ to $\frac{\pi}{2}$
Joint 5	ID 1	Yaw	8kg/cm with 7.4V	$-\frac{\pi}{2}$ to $\frac{\pi}{2}$

TABLE II
“零状态”下机器人关节的伺服规格

过一个降压芯片提供 RPi 4 所需的 5V 电压。下一个电源是另一个 12V、2400mAh 的 Lipo 电池，通过一个 5V 到 12V 的双通道继电器开关为线性驱动器供电，该开关根据接收到的 5V 输入提供准确的 12V 输入给线性驱动器，从而实现对线性驱动器的精确控制，这对机载深度摄像机的定位以进行植物检测和操作至关重要。最后一个电源是一个 5V、2500mAh 的移动电源，为两个 RPi 5 模块、两个 D435i 深度摄像机和以太网交换机供电。

我们选择 Robot Operating System 2 (ROS2) Humble 而非 ROS1 来处理我们的所有感知和规划任务。总的来说，ROS2 相较于 ROS1 是一个更可靠和高质量的框架。与 ROS1 使用的 TCP/UDP 相比，ROS2 使用高度安全的 DDS 通信。此外，ROS2 支持每个进程多个节点，而 ROS1 每个进程仅允许一个节点。然而，由 ArmPi Pro 提供的现有轮子和机械臂电机的控制软件是基于 ROS1 Melodic 的。为了避免重新发明轮子，我们在此基础上构建了面向比赛的软件。我们的软件架构由三个主要模块组成：行为协调器、导航系统和植物处理系统。这些模块共同实现了比赛中所需的机器人功能。

为了对接基于该系统构建的 ROS1 系统和 ROS2 系统，我们创建了一个同时运行内部网络服务器的 ROS1 节点。这个服务器公开了 REST API 端点，当我们的 ROS2 系统以所需的电机输入作为参数访问这些端点时，它会发布相应的 ROS1 消息来控制车轮和机械臂电机。

由于支持 Armpo Pro 的 Pi 上运行的 Ubuntu 18.04 的 ROS2 (Dashing) 不足，我们选择了 Flask REST API 而不是 ROS2 桥。因此，Flask REST API 为 ROS1 和 ROS2 之间的通信提供了可靠的通道。

C. 行为协调器

行为协调器（如图 1 所示）负责协调整个机器人行为和管理高层决策过程。它使用自定义的 ROS2 消息与导航系统和植物处理系统通信，以根据当前状态和传感器数据触发适当的动作。行为协调器实现了以下高层逻辑：

- 1) 导航到起始区域和植物床走廊的交叉口，面向植物床 A。
- 2) 当在 A 侧检测到未被访问的植物或经过的时间少于 ~2.25 分钟时，导航到 A 侧下一个检测到的植物的最近一侧并调用植物处理系统。
- 3) 走到竞技场另一侧，并旋转 180 度。

- 4) 当在 B 侧检测到未访问的植物或经过的时间少于 ~4.75 分钟时，导航到在 B 侧检测到的下一个最近的植物的一侧，并调用植物处理系统。
- 5) 导航到交叉口并移动到所需的终点位置。

机器人前装摄像头为机器人相对于全局框架的定位提供了必要的信息。当线性执行器伸展时，后端朝前摄像头沿垂直轴上升，植物处理系统使用该摄像头来视觉分析植物。识别出的目标，即不健康的叶子和不需要的花朵及其 3D 位置，然后用于执行机械臂定位和采摘动作。

D. 导航系统

导航系统（见图 5）负责在场地内定位机器人并规划路径以到达预定的目标位置。它依赖于来自 Intel RealSense D435i 深度摄像头的传感器数据，并使用 RTABMAP 进行视觉 SLAM（同步定位与建图），从而构建环境的精确地图。随着机器人行为的推进，导航目标姿态由行为协调器决定并传达，确保在连接当前姿态和目标姿态的直线路径上没有障碍物。现在，使用简单的 PID 控制器来最小化两种姿态之间的误差。完成后，控制权返回给行为协调器。

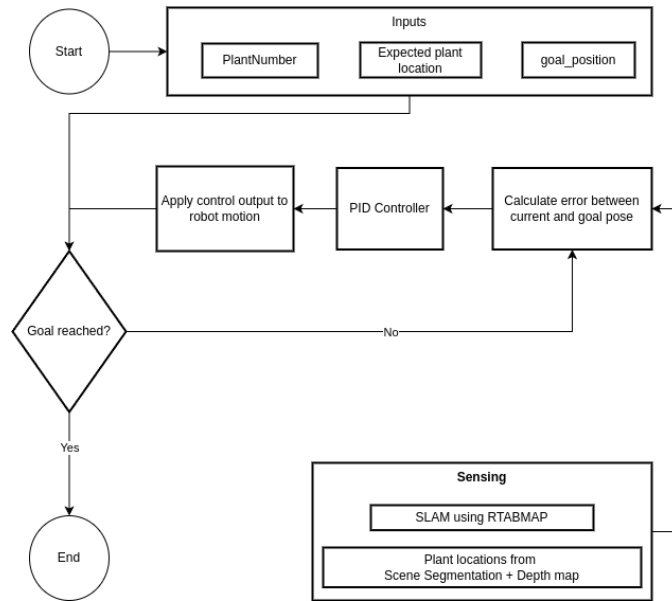


Fig. 5. SARAL-Bot 导航系统流程图

E. 植物加工系统

植物处理系统由行为协调器在机器人接近植物时激活。它利用安装在线性执行器上的后端前置深度摄像头捕捉植物的 RGBD 图像。一个用于检测和定位每株植物的叶子和花的视觉管道被建立。该管道将 RGB 转换为 HSV（色调、饱和度、亮度）颜色空间，并根据比赛规则中提供的信息中的色彩阈值识别健康叶、非健康叶和花。为了减少误判，使用 RANSAC 算法 [?] 进行平面性检查，丢弃任何检测到的非平面物体。

然后利用来自相机的深度信息将检测到的叶子和花相对于相机框架进行定位。在检测、识别和定位植株部分之后，“植物处理系统”执行以下任务：

- 1) 使用自定义 ROS2 消息发布植物部件的数量和姿态。
- 2) Transform the poses of the unhealthy leaves and flowers to the manipulator's base link.

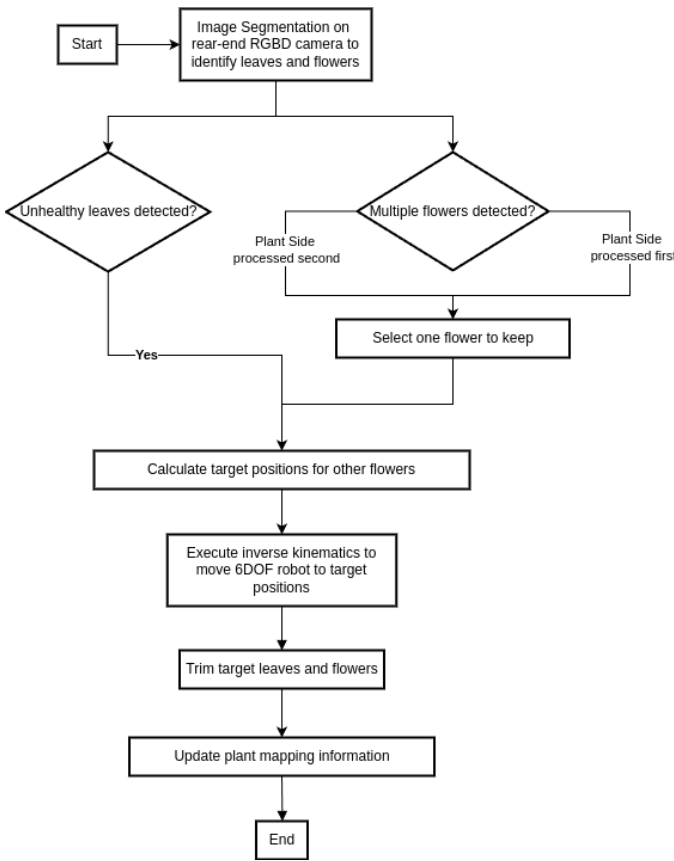


Fig. 6. SARAL-Bot 植物处理系统流程图

- 3) 基于行为节点的输入，如果我们正在处理植物的近侧，我们会跳过修剪一朵花。另一方面，如果我们正在处理植物的远侧，根据近侧剩下的花的数量是 0 或 1 朵，我们分别修剪至除一朵或所有花以外的所有花。
- 4) 使用 REST API 请求向 ArmPi Pro 的关节状态控制节点发送操作命令，从而使改装的夹爪达到修剪目标所需的姿势。

通过将软件架构分解为这三个主要模块，我们确保了关注点的明确分离，并实现了子系统之间的高效通信。行为协调器负责高层决策，导航系统负责定位和运动控制，而植物处理系统则专注于检测、定位和操作植物部件。

如前所述，Armapi Pro 的 Raspberry Pi 运行着 ROS Melodic，它管理与操作相关的主题，以及用于机器人硬件的其他操作主题和服务。操作主题处理一组整数数组，这些整数对应于手臂关节的数量（范围在 0 到 1000 之间），每个整数表示一个关节的旋转角度，以弧度为单位。

在 ROS 2 端，我们的重点是进行逆向运动学和避障。在逆向运动学中，目标位置（3D 位置和方向）通过检测和定位获得。然后我们计算逆向运动学，以模拟机械臂中每个关节的姿态，使末端执行器到达目标位置。估算的姿态，以弧度表示，被缩放到 0 - 1000 之间，通过之前提到的 REST API 调用传递给 ROS 1 主题。

最初，我们使用了机器人工具箱（RTB）进行逆运动学计算，但发现 RTB 有时会估算出超过关节机械限制的姿势，从而导致不现实和不稳定的动作。此外，RTB 缺乏用于避障的预定义包。避障对于防止机械臂碰撞到 saral —

机器人的后部硬件组件，以及避免与植物及其健康结构碰撞至关重要。

Moveit2 是一个易于使用、结构良好、稳定且可靠的机器人操控平台。其内置和即插即用的功能（如逆向运动学和避障）使机械臂的操作更真实和稳定。

为了利用 Moveit2 软件包，我们创建了我们机器人的 URDF（统一机器人描述格式）文件，该文件描述了机械臂的每个部分，包括链接、关节及其属性。使用 URDF 文件，我们生成了一个 SRDF（语义机器人描述格式）文件以及几个用于碰撞设置、运动学和控制器的配置文件。这些文件最初用于模拟目的。在模拟阶段，我们使用 ROS2 控制作为控制器来操作我们的机械臂以实现所需的目标姿势。在实时操作中，我们使用 Armapi Pro 的内置软件包作为控制单元。

逆运动学操作非常简单，避免了超过关节限制的动作。除了管理关节限制外，机械臂还必须避免与后方的计算块和植物的健康部分发生碰撞。为便于实现这一点，我们在规划场景中添加了一些用于可视化的对象；例如，机械臂后面放置了一个长方体对象，在植物中心位置放置了一个圆柱体对象。这个设置有助于规划操作路径，以避免这些对象并成功到达目标姿态。

III. 测试和性能数据的适当性

我们首先测试本文中描述的每个模块。然后，我们在模拟的竞赛环境中测试完整的竞赛流程（见图 ??，8）。

在这一小节中，我们评估了导航系统的性能。在测试中，我们的机器人通过使用基于 PID 的控制方法自动导航模拟场地，沿着一组静态路径点进行行进，同时从 Intel RealSense D435i 深度摄像头收集 RGBD 和 IMU（惯性测量单元）数据。RTABMAP 实时处理输入的数据，构建环境的三维地图（图 7），并估计机器人在地图中的位置。

这些结果展示了我们导航系统的鲁棒性和可靠性。通过在模拟环境中对导航系统进行更全面的测试和验证，我们将更加确信其在实际比赛中表现良好的能力。从这些测试中获得的积极结果强调了我们所选择的方法的适用性以及我们的机器人应对真实挑战的准备程度。

A. 检测

在图 9 中，我们展示了单株植物检测的结果。我们使用准确率作为评估我们用于检测健康或不健康叶簇和花朵的 HSV 颜色阈值化方法的指标。在不同的视角和光照条件下，我们的初步结果显示了良好的准确性，但也存在误报。例如，由于具有相似的颜色特性，非植物对象被检测为花朵或叶簇。

当我们加入一个平面性检查时，它可以消除误报。我们在图 10 中可视化了在我们的模拟测试环境中测试的植物检测到的花朵物体的平面。这里，拟合的平面包含了 251 个内点。

为了验证我们操作系统的性能，我们在模拟和现实世界的环境中进行了一系列测试。测试的目的是评估我们逆运动学（IK）求解器和避障算法的准确性、可靠性和效率。起初，我们利用彼得·科克博士的 Robotics Toolbox (RTB) Python 来进行逆运动学处理，并在模拟环境中测试操作流程。然而，我们遇到了 RTB 估计超出关节机械极限姿态的问题，导致动作不真实且不稳定。此外，RTB 缺乏在有障碍物情况下的集成规划，这对于防止自碰撞和与植物的碰撞至关重要。为了解决这些限制，我们转向使

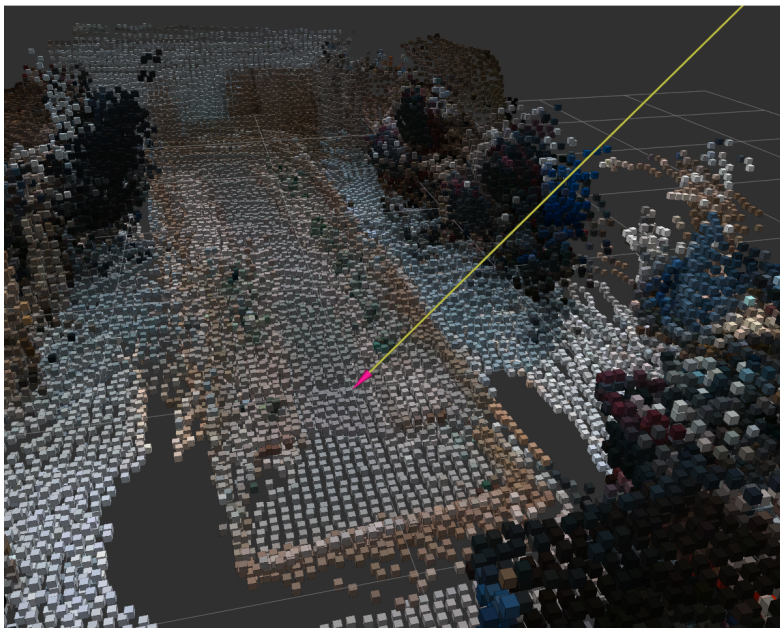


Fig. 7. 从 RTABMAP 数据库检索的地图重建

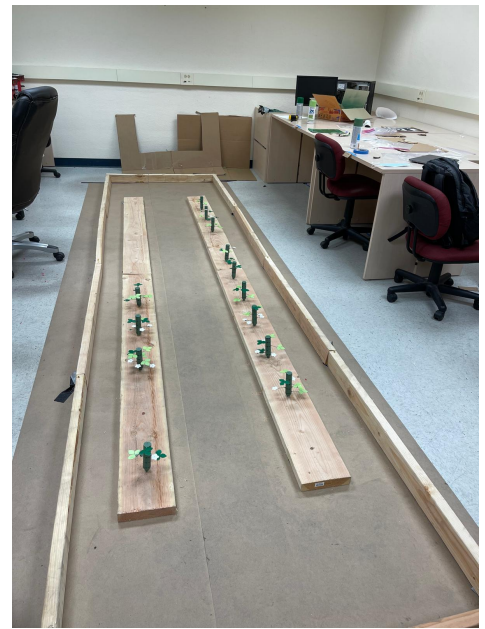


Fig. 8. 竞技场的真实数据

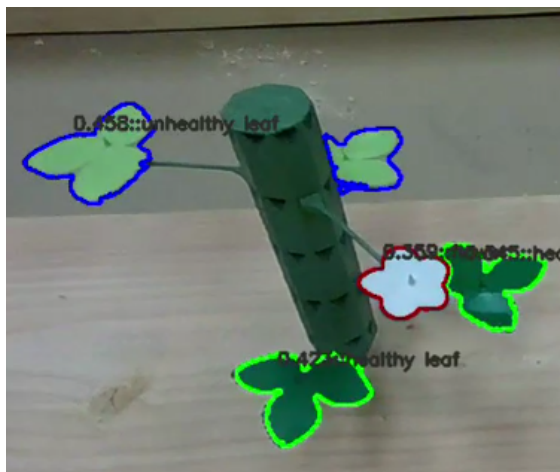


Fig. 9. 检测到健康、病变的叶簇和花朵

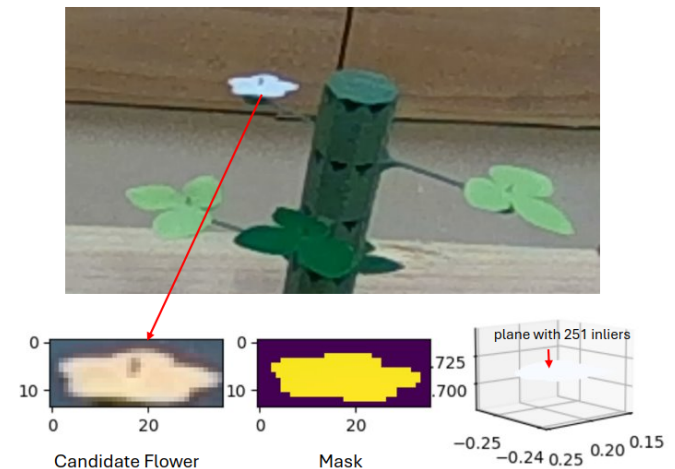


Fig. 10. 可视化检测到的花对象的平面，具有 251 个内点

实验，让我们对操作系统为比赛做好准备感到信心十足。

用 Moveit2，这是一个强大而复杂的平台，用于植物操作任务。Moveit2 内置的逆运动学和障碍物避让功能使我们能够进行真实且稳定的机械臂操控。在仿真阶段，我们使用 ROS2 控制作为控制器来操控机械臂，以实现所需的目标位姿。对于我们预期在机器人运行过程中遇到的所有相关测试案例，目标位姿均在不违反任何关节限制或发生自碰撞的情况下实现。为了测试我们的障碍物避让管道的有效性，我们在规划场景引入了虚拟对象，代表机器人后部的计算模块和植物的健康区域。然后，我们让机械臂在避开这些障碍物的同时达到目标位姿。经过在仿真环境中验证操作流程后，我们将其部署在物理机器人上并进行了现实世界的测试。我们的系统能够成功抓取和修剪用于比赛的植物模型中的不健康叶片和花朵，从而证明了操作系统的有效性。从 RTB 向 Moveit2 的转变被证明是一个显著的改进，使我们能够克服 RTB 的局限性并实现稳定可靠的操作。全面的测试和验证过程，涵盖了仿真和现实世界

IV. 目标的实现

在我们的机器人开发过程中，我们始终专注于实现本报告 Sec ?? 中设定的目标和设计标准。通过利用创新的硬件增强、模块化的软件架构和先进的算法，我们成功地创建了一个能够应对比赛挑战的机器人。现在，我们将讨论从两个主要目标中获得的成果。

我们的首要目标是开发一种能够自主导航比赛场地并遵循指定尺寸限制的机器人。将 ArmPi Pro 底盘与定制硬件添加项（如线性执行器和深度摄像头）整合，使我们能够创造出一个紧凑且灵活的平台。使用 RTABMAP 进行同时定位和地图构建，再结合高效的路径规划和运动控制算法，使我们的机器人能够以精准和可靠的方式在比赛场地中导航。

我们的第二个目标是实现精确和稳健的植物检测以及选择性采摘/修剪。在机器人后部的线性执行器上安装的

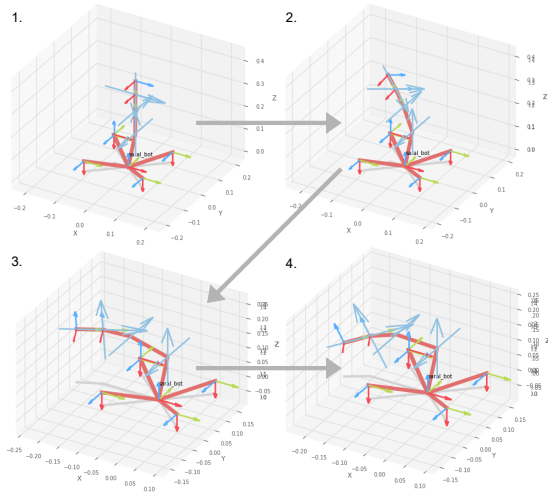


Fig. 11. 模拟显示了植物处理系统如何定位自身以修剪不健康的叶簇。

RGBD 相机可以清晰地查看植物，从而有效地检测和定位叶子和花卉。我们的基于 HSV 的颜色阈值和轮廓检测算法，加上点云处理，使机器人能够高精度地区分健康和不健康的植物部分。Moveit2 的逆运动学和避障功能的集成确保了修剪过程中机器人手臂的精确和无碰撞操作。除了竞赛目标外，我们还旨在创建一个模块化和可扩展的系统，易于适应和维护。使用 ROS2 作为主要的软件框架，可以促进模块化和可重用的代码库，允许无缝集成新功能和算法。行为协调器、导航系统和植物处理系统之间的关注点分离，促进了干净和可维护的软件架构。此外，利用 Docker 进行环境管理，确保了跨不同系统的一致性和可重复的部署。

在整个开发过程中，我们进行了广泛的测试和评估，以验证机器人性能和可靠性。这些目标的实现证明了我们团队的辛勤工作、奉献精神和技术专长。成功地将导航、感知和操作能力整合到一个紧凑且高效的系统中，展示了自主机器人在改变我们对植物检验和选择性收割等任务的处理方式方面的潜力。展望未来，我们相信，通过这个项目所获得的知识经验和将为农业机器人进一步的进步奠定基

础。我们系统的模块化和可扩展性为将机器人适应于比赛之外的各种农业应用打开了可能性。通过继续完善和扩展我们的机器人所展示的能力，我们旨在为开发创新解决方案做出贡献，以解决现代农业面临的挑战，最终促进效率、可持续性和作物健康的改善。

我们谨向所有支持和为我们开发用于 2024 ASABE 学生机器人比赛的机器人做出贡献的人士表达最诚挚的感谢。首先，我们向我们的教师导师 Parikshit Maini 博士致以诚挚的谢意，感谢他在整个项目过程中不懈的指导、专业知识和指导。他的宝贵见解和鼓励对我们解决困难并实现目标起到了关键作用。我们感谢 ASABE（美国农业与生物工程学会）组织这次比赛，并为学生提供展示农业机器人技能和创新的平台。比赛一直是我们的灵感和动力来源，激励我们突破知识、技能和能力的界限。我们也感谢开源社区，特别是 ROS2、RTABMAP、Moveit2 及其他库和工具的开发者 and 贡献者，他们对我们机器人的软件栈至关重要。他们致力于创建可访问且强大的软件极大地促进了我们的开发过程。

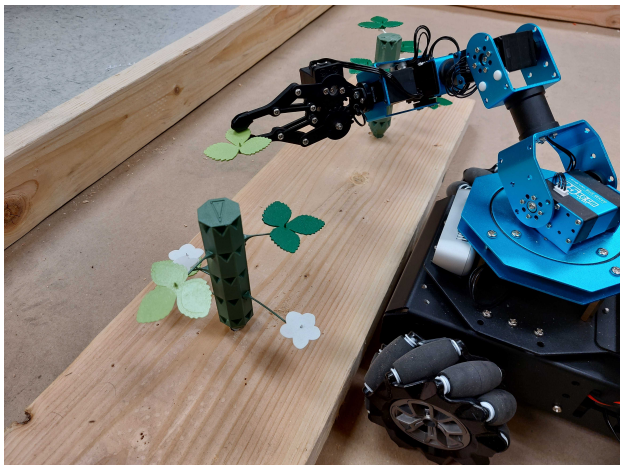


Fig. 12. 植物处理系统用 5 自由度机械臂和 1 自由度横向运动修剪不健康的叶簇