IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Abu Dhabi, UAE, 2024.

HortiBot: An Adaptive Multi-Arm System for Robotic Horticulture of Sweet Peppers

Christian Lenz*[1,3] Rohit Menon*[2,3] Michael Schreiber[1] Melvin Paul Jacob[2]
Sven Behnke[1,3,4] Maren Bennewitz[2,3,4]

***Abstract*— Horticultural tasks such as pruning and selective harvesting are labor intensive and horticultural staff are hard to find. Automating these tasks is challenging due to the semi-structured greenhouse workspaces, changing environmental conditions such as lighting, dense plant growth with many occlusions, and the need for gentle manipulation of non-rigid plant organs. In this work, we present the three-armed system HortiBot, with two arms for manipulation and a third arm as an articulated head for active perception using stereo cameras. Its perception system detects not only peppers, but also peduncles and stems in real time, and performs online data association to build a world model of pepper plants. Collision-aware online trajectory generation allows all three arms to safely track their respective targets for observation, grasping, and cutting. We integrated perception and manipulation to perform selective harvesting of peppers and evaluated the system in lab experiments. Using active perception coupled with end-effector force torque sensing for compliant manipulation, HortiBot achieves high success rates in our indoor pepper plant mock-up.**

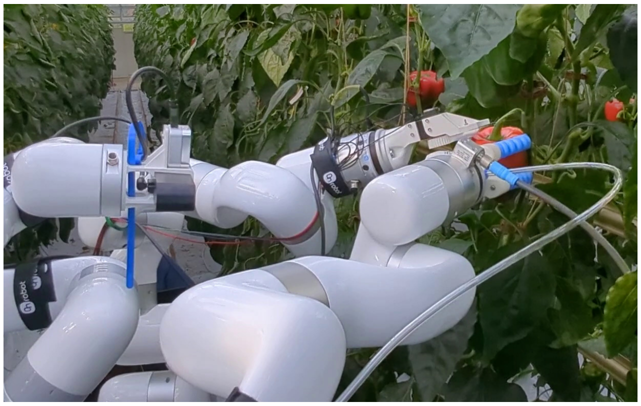

Fig. 1: HortiBot: A three-arm system with active perception and dual-arm manipulation for robotic horticulture. The right arm is used for grasping, the left arm performs cutting, and the central arm moves stereo cameras for mapping and online observation.

I. Introduction

Horticultural tasks such as pruning, thinning, pollination, and selective harvesting are labor-intensive and need to be carried out several times a season [1]. In contrast to the mechanization of large-scale grain and cereal farms, the automation of precision horticulture requires robots. Robotic manipulation in horticulture presents several challenges due to semi-structured greenhouse workspaces, variations in environmental conditions such as lighting, complex and irregular plant structures, varying plant organ sizes and shapes, dense plant growth with many occlusions and obstacles, and the need for gentle manipulation of non-rigid plant organs [2].

While there is an extensive body of work focusing on fruit detection and localization, research on the full robotic harvesting pipeline is limited [3]. Most selective harvesting systems use specialized hardware for manipulators and end-effectors [1]. In a recent review, Rajendran *et al.* [4] suggest equipping selective harvesting robots with cooperative active and interactive perception for improved fruit detection and force sensing-enabled two-arm manipulation capabilities—to match humans in handling complex fruit clusters. With humanoids having potential to become general-purpose autonomous workers adapting to different tasks [5], we aim to close the research gap in horticulture manipulation by proposing a non-specialized solution. HortiBot is a three-arm system for active perception and dual-arm manipulation in horticulture. The highly flexible robot is built from off-the-shelf components for multiple horticultural tasks. Unlike most other works that focus on only vision, control, or motion planning, we present a fully integrated system. Our contributions include:

- work space analysis and design of a three-arm system with stereo cameras and force-torque sensors,
- visual perception of sweet pepper plants combining fruit instance mapping with a novel peduncle detection approach and stem detection,
- online active perception during manipulation for refining of targeted pepper and peduncle localization,
- dual-arm manipulation using parameterized motion primitives and collision-aware online trajectory generation, and
- a thorough evaluation of the selective harvesting capabilities in lab experiments using real sweet peppers.

II. Related Work

With advancements in robotics and deep learning methods, different aspects of horticulture have been automated using robotic systems such as pollination [6] and dormant pruning [7]. Of the many tasks in the horticultural industry, selective harvesting is the one most often addressed by robotic solutions [3]. The typical phases of selective harvesting are fruit detection and localization, end-effector motion planning, fruit attachment to the end-effector, fruit detachment

[1]: Autonomous Intelligent Systems Lab, University of Bonn, Germany
[2]: Humanoid Robots Lab, University of Bonn, Germany
[3]: Center for Robotics, Bonn, Germany
[4]: The Lamarr Institute, Bonn, Germany

This work has partially been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy, EXC-2070 – 390732324 – PhenoRob.

*These authors contributed equally to this work.

from the plant, and transport to a storage container. The surveys compiled over the years [1]–[4] show that while substantial progress has been made in fruit detection and robotic hardware customization, the harvesting systems are still not ready for commercialization due to low success rates and high cycle times.

Whereas most attempts at autonomous harvesting have focused on citrus fruits or apples due to sparse foliage and easier localization, there have been only three reported attempts on the development of a full pipeline for sweet pepper harvesting: CROPS [8], Harvey [9], and SWEEPER [10]. Sweet peppers are among the most difficult crops to autonomously harvest due to variation in shape and size, and severe occlusions by leaves leading to failures in both pepper and peduncle localization [8].

In CROPS [8], the focus of the research was on end-effector design, with color-based pepper detection and time of flight measurement for 3D localization. Bac *et al.* [8] also developed a stem-dependent grasp pose calculation. However, neither sweet pepper pose estimation nor peduncle localization was the focus of this work, which led to low success rates and high cycle times.

In Harvey [9], a sweet pepper pose estimation and grasping algorithm [11] together with MiniInception [12], a mixture of lightweight CNN approach for peduncle segmentation, was deployed to improve the harvesting performance. However, the peduncle localization accuracy is still limited with an F1-score of 0.502 and led to detachment failures. Furthermore, Harvey used a customized end-effector with a suction cup and did not focus on motion planning for crop damage avoidance, or active perception.

Arad *et al.* [10] focused on finding the best fit crop conditions and on testing & validation of SWEEPER in a commercial glasshouse. Semantic segmentation-based fruit and stem detection were deployed on a 6-DoF industrial robot arm with a customized end-effector, which caught the fruit after harvesting. Due to the lack of peduncle localization, cutting failures were reported.

To the best of our knowledge, HortiBot is the first attempt at selective harvesting in general, and sweet peppers in particular, that focuses on all the aspects of harvesting: fruit detection and peduncle localization, active perception, environment-aware motion planning and force sensing-enabled adaptive manipulation. HortiBot is a general-purpose system that can also be used for other horticulture operations such as leaf pruning and pollination.

III. Hardware Setup and System Overview

While we focus on selective harvesting in this work, HortiBot is intended for autonomous operation of different horticulture operations such as leaf pruning, pollination, and crop monitoring. This necessitates the use of dual-arm manipulation. Additionally, an articulated head is necessary to enable the manipulation system to perceive in the presence of occlusions due to leaves and other plant organs during task completion. The effectiveness of using a camera mounted on an arm for visual tele-manipulation has already been demonstrated [13], [14].

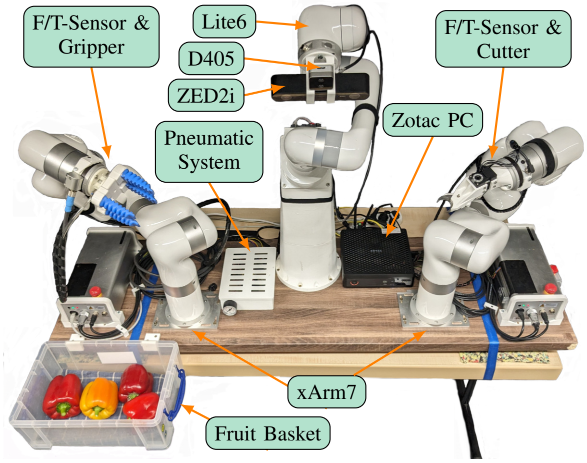

Fig. 2: HortiBot hardware setup.

A. Hardware Setup

HortiBot consists of two 7-DoF UFactory xArm7 and one 6-DoF UFactory Lite6 equipped with sensors and end-effectors to autonomously perform greenhouse applications such as selective harvesting in an adaptive manner (see Fig. 2). The system is mounted on the PATHoBot platform [15], designed to operate in commercial glasshouse environments using the available structure. It navigates on pipe-rails between individual crop rows and uses a scissor-lift to bring HortiBot to the desired height (up to 3 m).

Both UFactory xArm7s are equipped with an OnRobot HEX-E force-torque sensor. The right arm has a pneumatic four-finger soft gripper referred to as *Grasper*. The pneumatic pump and two air valves are controlled using the digital outputs of the xArm controller. The left arm is equipped with a custom designed 1-DoF scissor, referred to as *Cutter*. The Lite6 carries two stereo cameras: a ZED2i stereo camera with deep learning based depth inferencing for medium range sensing, and a RealSense D405 for short range sensing, referred to as *Observer*. The Zed2i with a wide angle field of view of 110 ° has better performance in sunlight with its stereo based depth sensing and polarized lenses.

All three arms are connected to a common emergency-stop button for safe operation. All necessary components including the control PC (Zotac ZBox with core i7-13700H, 32GB RAM and RTX4070 mobile GPU) running ROS-Noetic on Ubuntu 20.04, are mounted on a wooden platform which can be fitted onto the PATHoBot platform easily.

B. System Overview

We use an adaptive autonomous behavior approach to perform selective harvesting. Fig. 3 shows a brief overview of the different phases of the autonomous harvesting workflow. We carry out an initial mapping of the sweet peppers as described in Sec. IV to create a world model of the pepper plants. Using this model, we select the fruits based on their reachability. Thereafter, we activate the online fruit following

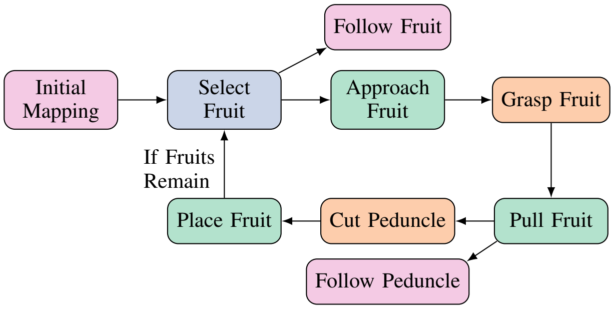

Fig. 3: Workflow for autonomous selective pepper harvesting. Colors depict different actions: Perception (see Sec. IV), Logic (see Sec. V-C), manipulation using Motion Primitives (see Sec. V-A), and Online Trajectory Generation (see Sec. V-B).

while simultaneously approaching the fruit with the *Grasper* and the *Cutter*. Once, we grasp the fruit, we pull it and then refine the *Cutter* pose based on the updated peduncle localization from the *Observer* as described in Sec. V. We adaptively adjust the cutter position based on force feedback and cut the peduncle after which the *Grasper* transports it to the storage container and places it there. Except for the initial mapping, the cycle is repeated until no fruits remain.

C. Workspace Analysis

The limited space in the glasshouse crop rows, and arm specifications (workspace and kinematics) must be taken into account when designing the platform. The goal is to maximize the common workspace between the two manipulation arms while reducing the potential for collisions. We sampled 840 different arm mounting poses for both arms with different x- and y-positions and roll and pitch angles and tested each by counting collision-free IK-solutions reaching 270 sampled fruit poses. For each sample, the grasp and cut end-effector pose is calculated for the corresponding arm. The fruit poses have been sampled to be comparable to real poses in glasshouses and are shown in Fig. 4. Over 90% of the sampled fruits are reachable with both arms in the selected configuration. Repositioning the PATHoBot platform allows to access the remaining fruits.

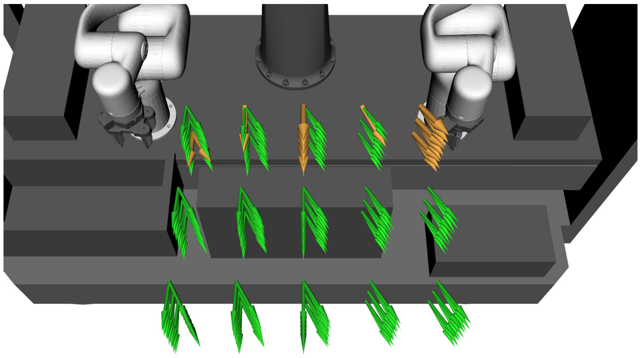

Fig. 4: Workspace analysis: Reachable (green) and non-reachable (yellow) fruit poses for both manipulation arms in the selected arm configuration.

D. Calibration

Some transformations must be calibrated before the system can be used with the required accuracy. We perform a classical hand-eye calibration approach to estimate the transformations between *Grasper's* and *Cutter's* mounting poses, *Grasper's* and *Observer's* mounting poses and the camera mounting pose, similar to [16]. Custom 3D printed magnetic ArUco markers can be attached to the *Grasper's* and *Cutter's* last link with known transformations. We collect about 2,000 samples with different arm configurations, extract pixel location in the images using the ArUco marker detection of the OpenCV library and computed the projected marker location into the image plane using forward-kinematics. Finally we minimize the squared error function over all samples, yielding the optimized transformations with a mean reprojection error of 2.8 pixels over the recorded samples.

In addition to the hand-eye calibration, the force-torque sensors need to be calibrated. We collect 45 force-torque measurements from different sensor poses and use a standard least squares solver to determine the optimal parameters for the end-effector mass, the 3D center of mass with respect to the sensor frame, and the force and torque bias. The calibration procedure is performed once after hardware changes or when the sensor bias drift becomes too large. Currently, sensor drift is not compensated online, instead we use the relative change over a short time horizon.

IV. Perception and World Modeling

A dynamic world model of the pepper plants is necessary for successful autonomous harvesting. To this end, perception of pepper plant organs is carried out in two stages as shown in Fig. 6. In the initial mapping stage, the manipulation arms are in stowed position and the *Observer* records the pepper plant detections at different poses to create a world model of the pepper plants with sweet pepper fruits, associated peduncles and nearby stems. The manipulation system uses this pepper plant model to determine the reachable fruits and selects them serially for harvesting. During the manipulation phase, the *Observer* performs online fruit following (Sec. IV-D) for dynamic fruit and peduncle localization to account for perturbations in the fruit locations owing to the manipulation arms touching parts of the plants.

A. Plant Organ Detection

For creating a world model of pepper plants, we need to detect and localize the pepper fruits, peduncles and stems. We adopted a multi-pronged approach for detecting these plant organs. We combined the synthetic capsicum dataset [17], Kaggle sweet pepper dataset [18], and BUP20 dataset [19], to create an extensive dataset resulting in more than 130,000 instances of sweet pepper. Since the synthetic dataset provides only semantic segmentation annotations, the detection of instances utilized OpenCV's [20] contour finding and refinement to generate instance segmentation masks for sweet peppers and peduncles.

Reliable peduncle detection is necessary for the *Cutter* to find the cutting point. However, detecting peduncles in

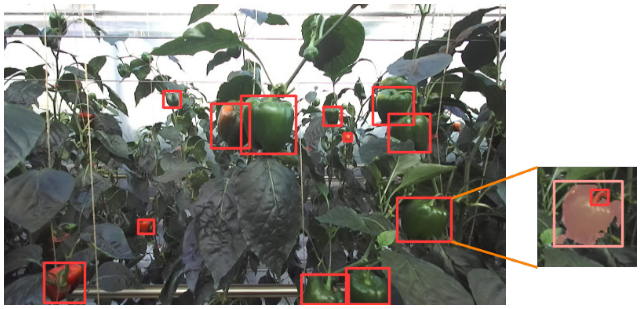

Fig. 5: Cropped peduncle detection. Pepper and cropped peduncle detection applied on the fruits in the Campus Klein Altendorf glasshouse pepper plants. As can been seen, there are multiple peppers with peduncles not easily identifiable in the full image. The image on the right shows the cropped image obtained by inflating the pepper's bounding box and the resultant pepper and peduncle detected.

the full image is a challenging task as the mean Average Precision mAP@50 was only 0.435, for a model trained on the aforementioned dataset using the full images. Hence, we developed a new approach for peduncle detection using cropped images. From the original dataset, we created an additional dataset containing cropped images of size 96x96 pixels centered around the sweet peppers, annotated with pepper fruit and peduncle masks. The cropped peduncle dataset contained more than 50,000 instances of peduncles and more than 100,000 instances of sweet peppers. The results of the cropped peduncle detection method are presented in Sec. VI-A.

During run-time, the sweet pepper instance segmentation model is used to detect peppers in the full image. For each pepper detected, a cropped image is created by inflating the bounding box by 50 % as shown in Fig. 5. The cropped peduncle instance segmentation model is applied on this cropped image, and the peduncle mask and bounding box, if any detected, are transferred to the full image. The peduncle detections are annotated with the associated fruit instance id for subsequent merging in the 3D domain. YOLOv8's tracking mode was utilized to enable tracking of the sweet peppers and their associated peduncles across images.

During our initial trials, the *Grasper* used to accidentally grasp the stems, especially when the peppers were located behind the stems. Hence, it was imperative to detect and localize stems to enable the manipulation system to select a grasp that avoids grasping the stem during the fruit grasping. We trained DeeplabV3Plus-Pytorch's [21] semantic segmentation model on the synthetic capsicum dataset for semantic stem detections. At run-time, using contour detections, the semantic masks of the stems were converted to a YOLOv8 consistent instance detection format.

B. 3D Mapping

Instead of an iterative search, detect, and harvest approach for every pepper, which leads to higher cycle times, the *Observer* performs an initial mapping of the pepper plants using a fixed number of poses that span across the reachable fruit locations (Sec. III-C). This also enables the fruits to be viewed from different observations poses leading to better shape estimation. At each observation pose, the depth segments and pepper masks are combined to form the instance id and semantic id annotated point cloud segments. We adapted Voxblox++ [22] to integrate the pepper cloud segments based on YOLOv8's tracked instance id, as well as geometric overlap, to form an instance aware surface map of the sweet pepper fruits.

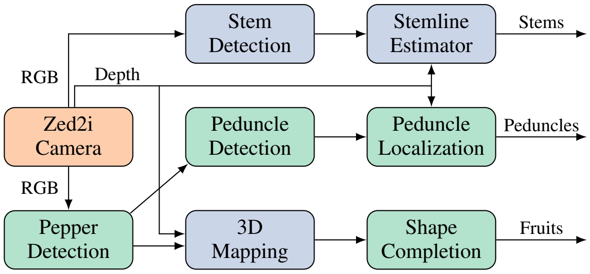

Fig. 6: Perception pipeline Hardware, Initial Mapping, Common. Stem detection and 3D mapping is done only during the initial mapping phase. Pepper and peduncle detection are carried out during initial mapping (Sec. IV-B) and fruit following (Sec. IV-D).

We also implemented an instance layer based extraction of point clouds from the Voxblox++ map to obtain the merged yet partial sweet pepper shapes formed after the initial mapping. The *Observer* then performs superellipsoid fitting [23] to estimate the completed shape, fruit pose and fruit dimensions from the partial sweet pepper shapes. To mitigate the problem of over-segmentation in the 3D domain due to YOLOv8 losing the instance id tracking on account of occlusions, it then performs 3D overlap detection using the completed shapes' poses and dimensions. When the overlap exceeds a certain threshold, the fruit with the smaller proportion of observed surface, computed as in our previous work [24], is discarded.

C. Pepper Plant Modeling

It is not sufficient to detect and localize peppers, peduncles and stems separately. They must also be associated with each other to form a usable world model that the manipulation system can utilize.

Once the 3D mapping motion is completed, merging and association of the detections at each *Observer* pose is performed to build the pepper plant model. As peduncle and stem point cloud segments have a relatively low number of points, their integration using Voxblox++ is not reliable. The *Observer* identifies and tracks peduncles (fruit stalks) and stems concurrently, while it performs the next mapping motion. It maintains a separate set of peduncles and stem segments to which the existing detections are added or merged.

Once the completed shapes are estimated, the *Observer* attaches the peduncles to the completed pepper shapes. It extracts each peduncle's bottom and top points as fruit point and stem point, after removing any outliers. If the peduncle's bottom point is close enough($\leq$1 cm) to the top center of the pepper, it associates the peduncle to the pepper. If multiple peduncle segments belong to the same pepper, it merges them and recalculates the peduncle endpoints accordingly.

For stem localization, the *Observer* rejects stems that are too short. It then estimates the stem's 3D line using PCL's [25] 3D RANSAC model. New stem detections are compared with existing ones, and if they align well, they are merged in a greedy manner, and the 3D line parameters are recalculated. If it finds a stem close to a pepper (within 5 cm in the x-y plane), it considers the pepper to be attached to that stem.

The *Observer* feeds the entire plant model consisting of the peppers with associated peduncles and stems to the manipulation system for selective fruit harvesting.

D. Online Fruit Following and Pose Update

Once the manipulation system selects a fruit for harvesting (see Sec. III-B), it transmits the selected fruit id to the *Observer* for fruit following. The online perception comprises two concurrent threads: one, running at 5 Hz, computes the viewpose and sets goals for the online trajectory generation method detailed in Sec. V-B. The other, running at 10 Hz, updates the fruit's pose estimate using instantaneous sweet pepper and peduncle detections for grasp and cut pose refinement.

During the grasping phase, the *Observer* fixates on the selected fruit by moving to the corresponding viewpose. The viewpose position p_{vp} is calculated in the local trolley frame where the x axis is aligned along the platform length, y axis pointing towards the fruits and z axis vertically aligned. We need the *Observer* arm to be above the *Cutter* i.e. p^z_{vp} to allow it easy access for peduncle cutting. At the same time, the *Observer* arm needs p^x_{vp} to be away from the vertical plane of the bases of the manipulation arms, whereas p^y_{vp} needs to maintain at least 35 cm from the fruit center for improved localization. p_{vp} is computed using the fruit center p_f as follows:

$$p^x_{vp} = 0.8 * (p^x_f - p^x_h) + p^x_h \tag{1}$$

$$p^y_{vp} = p^y_f - 0.35 \tag{2}$$

$$p^z_{vp} = p^z_f + l^z_f + 0.15 \tag{3}$$

where l_f and p_h represents the fruit bounding box dimensions, and the position of the base of the head arm respectively. The normalized direction vector dir_{vp} for the orientation of the viewpose is computed as follows:

$$dir_{vp} = \frac{p_f - p_{vp}}{\|p_f - p_{vp}\|} \tag{4}$$

dir_{vp} and p_{vp} are combined to form the SE3 viewpose. During the cutting phase, the *Observer* moves 5 cm closer to the fruit while moving 2cm higher as well for peduncle fixation.

During the grasping phase, for the online pepper pose update, the *Observer* gets the currently detected pepper cloud segments, smooths them using a moving least squares filter and subsequently performs shape estimation. In this phase, we bias the fitting on the currently detected peppers to be closer to the initial fruit center and fruit dimensions, under the assumption that the initial mapping estimates are better due to the multi-view merging.

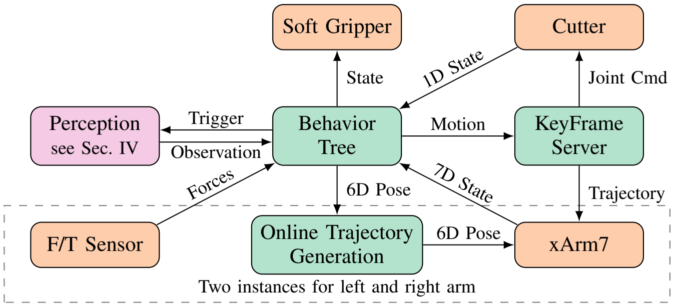

Fig. 7: Manipulation hard- and software pipeline. Behavior tree-based decisions control both xArm7s using motion primitives or online trajectory generation based on perception observations and sensor measurements. Colors correspond to categories Hardware, Manipulation, and Perception.

If the center of a currently detected pepper's completed shape is less than 1 cm away from the initial mapping estimate, we greedily assign this detection and its completed shape to be the current estimate of the fruit. However, if the center of the completed shape is more than 1 cm but less than 3 cm, then the fruit is added to a list of potential candidates for the current estimate and we choose the one nearest to the original estimate. Using complementary filtering, we filter the current estimates for the fruit centers as follows:

$$p^{filt}_f = \alpha \cdot p^{curr}_f + (1 - \alpha) \cdot p^{prev}_f \tag{5}$$

where p^{curr}_f and p^{prev}_f, p^{filt}_f, represent the current, previous and filtered estimates of the fruit center.

Once the fruit is grasped, the *Observer* switches to online peduncle localization only. The cropped peduncle detection method relies on the grasped pepper being detected in the full image. However, due to occlusions by the gripper, the pepper detection fails frequently which leads to downstream failures in the peduncle detection. Hence, the *Observer* uses the gripper tool center point (TCP) pose to construct a 3D bounding box using the fruit center p_f and fruit dimensions l_f as follows:

$$x_{bound} = p^x_f \pm l^x_f \qquad y_{bound} = p^y_f \pm l^y_f \tag{6}$$

$$z^{bottom}_{bound} = p^z_f \qquad z^{top}_{bound} = p^z_f + l^z_f + 0.05 \tag{7}$$

The 3D bounding box points are converted to 2D points on the image using the camera parameters. The region of interest for peduncle detection is computed as the minimum and maximum of the 2D points. The peduncle detected in the RoI is converted to cloud segment and added to a buffer with length 4 and maximum age of 0.5 s. The valid frames of the buffer are merged and smoothed to obtain the updated fruit and stem points.

V. Dual-Arm Manipulation

Autonomous horticultural operations require adaptive manipulation capabilities to robustly handle different plant arrangements and cope with dynamic changes such as moving a fruit while grasping it.

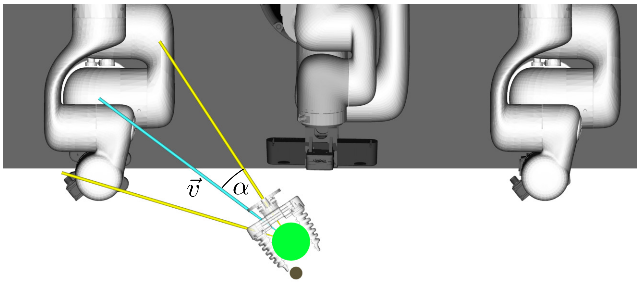

Fig. 8: Grasp direction (shown by the gripper model) for the fruit (green circle) is selected to be opposite of the corresponding stem (brown circle) without exceeding an angular deviation (yellow lines) from the vector v connecting the arm mounting point and the fruit center (light blue line). If no corresponding stems are detected, v is used as the grasp direction.

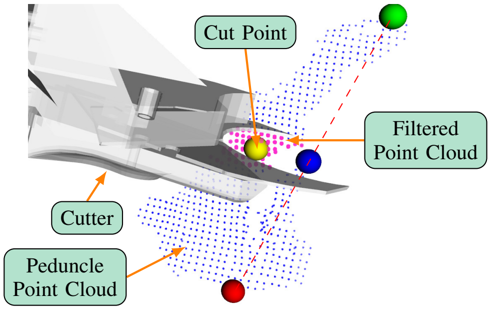

Fig. 9: Cut pose (shown by the cutter model) is computed using the peduncle point cloud. Green and red spheres indicate the highest and lowest peduncle points. Magenta points show the filtered peduncle cloud using a box filter centered at the blue sphere. The cut position (yellow sphere) is the centroid of the filtered cloud.

Our dual-arm manipulation method controls two xArm7s, a soft gripper, and a custom cutter based on observations provided by the perception pipeline (see Sec. IV) and force-torque sensors attached to each arm (see Fig. 7). The required motions vary in execution length and the goal pose update frequency, i.e. long motions (>0.5 sec) with a goal pose known before motion execution (for example approaching the fruit) and motions with a dynamic goal poses such as grasping and cutting the fruit, which have a non-zero start velocity. We use two different motion generation and execution methods, *Parameterized Motion Primitives (PMP)* and *Online Motion Generation (OTG)* to handle these requirements, which are described in the following.

A. Parameterized Motion Primitives (PMP)

All motions with a fixed goal pose (specified offline or online), are generated using *Parameterized Motion Primitives (PMP)*. A PMP consist of one or multiple keyframes each specifying one or multiple kinematic chains to be manipulated. The target configuration can be defined in Cartesian or joint space per chain and the generated motion linearly interpolates between the start and each keyframe goal configuration. We use *nimbro_ik* [26] to generate joint space configurations for Cartesian goal poses which uses a *selectively damped least squares (SDLS)* solver [27] and allows to define cost-functions which are optimized in the null-space. In this setup, we penalize the elbow crossing a vertical plane towards the system's center to reduce potential collisions. Self-collisions are checked along the trajectory and reported before motion execution. PMPs are either predefined offline for static motions such as placing the fruit in the container, or are parametrized online using sensor data for example when approaching the fruit.

B. Online Trajectory Generation (OTG)

Since every trajectory generated using PMP assumes zero start velocity, online replanning is not feasible. Instead, we switch the xArm control mode to OTG, which allows online replanning to follow a 6D end-effector goal pose considering the current robot state including current joint velocities and velocity and acceleration limits. However, this control mode does not provide any kind of (self-) collision checks. Therefore, we added collision checking on top of the OTG control mode. We compute the minimal distance $dist_c$ between any two links at the current robot state and $dist_n$ for an extrapolated state assuming constant velocity using *MoveIt* [28]. Next, we compute $\alpha_{\{c,n\}}$ which is used to reduce the motion velocity when approaching a collision:

$$\alpha_{\{c,n\}} = \frac{dist_{\{c,n\}} - 0.5}{3.0 - 0.5} \tag{8}$$

and ensure that, $\alpha_{\{c,n\}} \in [0, 1]$. If $alpha_n \leq alpha_c$, i.e., the robot is approaching a self-collision, we reduce the current motion velocity exponentially with the scalar $\beta \in [0, 1]$ which is defined as follows:

$$\beta = \frac{2^{5\alpha_n} - 1}{2^5 - 1} \tag{9}$$

This prevents the arm from self-colliding and but does not prevent the arm to move away from collisions.

In addition, the OTG controller reports the current status. The control-loop runs with 30 Hz which is sufficient for our application. We run three instances of this controller, one for each xArm7, and the *Observer* in case of online fruit following (see Sec. IV-D).

C. Adaptive Manipulation

Autonomous selective and adaptive harvesting requires a system which acts based on various sensor measurements. We calculate grasp and cut poses based on the online perception results and adjust manipulation trajectories using force-torque measurements.

The grasp direction is always orthogonal to the fruit's main axis (which is mostly vertical). The grasp direction is computed based on the fruit position and stem detection (see Fig. 8). Let $\vec{v} \in R^2$ be the vector between the *Gripper's* mounting center and the fruit center projected onto the x-y plane. We use $\vec{v}$ as the grasping direction if no stem detection is available for the selected fruit.

If stem detections are available, the fruit is grasped such, that the stem is avoided as much as possible. Let β be the angle between $\vec{v}$ and the selected grasp direction. We select the grasp direction opposite of the stem, with the condition

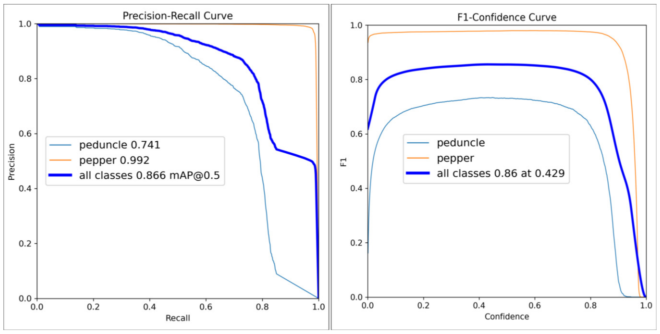

Fig. 10: Mask PR curve and F1 curve for peduncle detection in cropped image.

$\beta \leq 20°$. This avoids the stem (as much as possible) while creating feasible arm configurations.

We use the peduncle point cloud PC to calculate the optimal cut pose (see Fig. 9). Let $\vec{p}$ be the vector connecting the highest and lowest point of PC (in the vertical axis) and let M be the midpoint of $\vec{p}$. We filter PC using a box filter which is centered at M and aligned in the direction of $\vec{p}$. The centroid of the filtered PC is used as the cutting position. The cutting orientation is fixed relative to the *Cutter's* mounting pose, similar to the grasp orientation. We use a fixed cut position above the fruit in case of missing peduncle detections and ensure a minimum distance between grasp and cut pose to avoid self-collisions.

We update the grasp and cut pose with 100 Hz using the latest perception results and generate new arm trajectories using OTG (see Sec. V-B). In addition, we detect reaching the fruit or peduncle while grasping and cutting by monitoring the force measurements relative to the start of the motions. The current motion is stopped if the observed forces exceed a predefined threshold.

VI. Results

We evaluated two major aspects of our approach, namely peduncle detection and the adaptive autonomous selective harvesting performance. In the latter, we evaluate the success rate and execution time for different phases of the approach.

A. Peduncle Detection

While sweet pepper detection is a fairly mature research area, peduncle detection still has a lot of scope for improvement, with the MiniInception [12] approach reporting an F1 score of 0.313 and 0.564 for unfiltered and filtered data, respectively. Our method demonstrates a far superior performance with a mean average precision mAP@50 of 0.741 for peduncle segmentation as can be seen in Fig. 10. Similarly, the F1 score for peduncle detection is significantly better at 0.781.

The mean processing time for both pepper and peduncle detection and localization is 100 ms, which enables the system to perform these tasks in real time.

B. Experimental Harvesting Setup

We created an indoor mock-up of sweet pepper plants using real sweet peppers, artificial leaves and thin pipes as stems (see Fig. 11). 3D printed holders were used to attach the leaves and peduncles to the stems. We conducted six trials with 4 fruits to be harvested in every trial. The fruits were distributed among 3 stems. We used red (21), yellow (1) and orange (2) sweet peppers with sufficiently long peduncle for mounting reasons. The fruits were located roughly 0.5 m away from the HortiBot platform, similar to the glasshouse rows. The initial mapping was carried out using 5 observation poses. We did not use the D405 camera as it is needed only for close range sensing in the glasshouse.

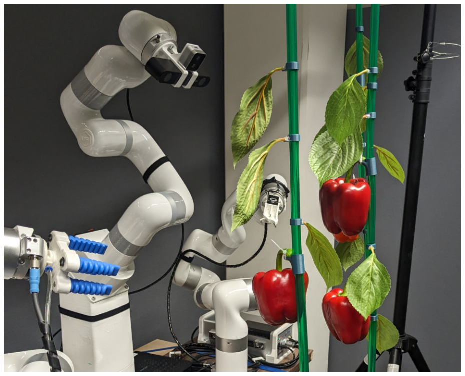

Fig. 11: Experimental harvesting setup.

TABLE I: Experimental Results.

Trial	Success				Time
	Grasp	Cut	Place	Overall	[m:s]
1	4/4	3/4	3/3	3/4	1:45
2	4/4	3/4	3/3	3/4	1:47
3	4/4	3/4	3/3	3/4	1:49
4	4/4	4/4	3/4	3/4	1:49
5	4/4	4/4	4/4	4/4	1:49
6	4/4	4/4	4/4	4/4	1:48
Total	24/24	21/24	20/21	20/24	

C. Harvesting Trials Results

We evaluated the full system pipeline by analyzing the execution time (recorded automatically) and the success rate (recorded manually) for each phase.[1] The overall success rate for the entire harvesting cycle is 83.33% with 20 out of a total of 24 fruits harvested successfully, as can been seen in Tab. I. HortiBot was able to successfully grasp all fruits without using any expensive grasp detection approach. The flexible pneumatic gripper perfectly adapted to the fruit shape. This validates our approach to use shape completion based grasp pose estimation with force sensing enabled compliant grasping. The cutting phase had a lower success rate of 87.5% with 21 out of 24 peduncles cut. The three failures were due to peduncle localization errors with depth registration issues. While peduncle detection in the RGB image was successful in all the cases, the depth rendering was poor due to the thin structures. While transporting the fruit, we had one failure in trial 4 due to an imperfect grasp resulting in the fruit slipping out of the gripper.

[1] https://www.ais.uni-bonn.de/videos/IROS_2024_Lenz/

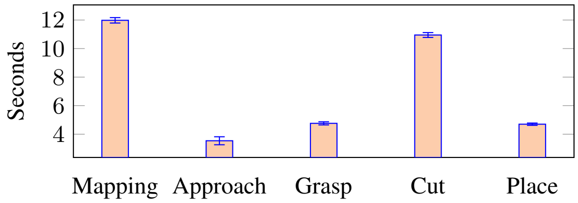

Fig. 12: Mean execution time and standard error for different phases of the harvesting cycle over six trials with four fruits each. Note, mapping is performed once per trial, all other phases once per fruit.

The execution time for harvesting is another key factor for determining the performance of the system. The overall mean time needed for each trial was 1 min 48 s, thus leading to an average time of 26.95 s per fruit including the failure cases. As can been seen in Fig. 12, the initial mapping needed 11.98 s on an average for a total of 5 poses with all the fruits successfully detected in all the trials. Using *PMP*, the approach phase required an average of only 3.54 s per fruit for the *Grasper* and the *Cutter* to reach their respective pre-grasp and pre-cut poses. The grasp phase required 4.76 s per fruit including opening and closing the gripper, and pre-grasp to grasp pose *OTG* motion. However, it was the peduncle cutting that required the longest time with each fruit needing around 10.95 s. This was due to the noisy peduncle localization update which caused the *OTG* cut motion to continue refining the cut pose until it found a stable pose. The transport of fruits was relatively fast with the place phase needing only 4.7 s per fruit.

VII. Summary

In this work, we presented HortiBot, a fully integrated system that focuses on all aspects of robotic harvesting. With an articulated head for active perception, and force-sensing enabled bi-manual manipulation, HortiBot can carry out different horticulture tasks. We developed a novel peduncle detection method that has significantly better detection accuracy, leading to 87.5% success in peduncle cutting. We also developed a novel collision-aware online trajectory generation method that is able to perform pose tracking at 30 Hz frequency. Using force-sensing based compliant grasping and cutting, we achieved an overall success rate of 83.33% and a cycle time of 27 s per fruit on an indoor mock-up of pepper plants, which outperforms state-of-the-art selective harvesting robots. We plan to deploy HortiBot mounted on the PATHoBot [15] for sweet pepper harvesting in glasshouse scenarios in the future.

References

[1] C. W. Bac, E. J. Van Henten, J. Hemming, and Y. Edan, "Harvesting robots for high-value crops: State-of-the-art review and challenges ahead," *Journal of Field Robotics (JFR)*, vol. 31, no. 6, 2014.

[2] G. Kootstra, X. Wang, P. M. Blok, J. Hemming, and E. Van Henten, "Selective harvesting robotics: current research, trends, and future directions," *Current Robotics Reports*, vol. 2, pp. 95–104, 2021.

[3] H. Zhou, X. Wang, W. Au, H. Kang, and C. Chen, "Intelligent robots for fruit harvesting: Recent developments and future challenges," *Precision Agriculture*, vol. 23, no. 5, 2022.

[4] V. Rajendran, B. Debnath, S. Mghames, W. Mandil, S. Parsa, S. Parsons, and A. Ghalamzan-E, "Towards autonomous selective harvesting: A review of robot perception, robot design, motion planning and control," *Journal of Field Robotics (JFR)*, 2023.

[5] Y. Tong, H. Liu, and Z. Zhang, "Advancements in humanoid robots: A comprehensive review and future prospects," *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 2, 2024.

[6] K. Li, Y. Huo, Y. Liu, Y. Shi, Z. He, and Y. Cui, "Design of a lightweight robotic arm for kiwifruit pollination," *Computers and Electronics in Agriculture*, vol. 198, 2022.

[7] A. You, N. Parayil, J. G. Krishna, U. Bhattarai, R. Sapkota, D. Ahmed, M. Whiting, M. Karkee, C. M. Grimm, and J. R. Davidson, "Semi-autonomous precision pruning of upright fruiting offshoot orchard systems: An integrated approach," *IEEE Robotics and Automation Magazine (RAM)*, 2023.

[8] C. W. Bac, J. Hemming, B. Van Tuijl, R. Barth, E. Wais, and E. J. van Henten, "Performance evaluation of a harvesting robot for sweet pepper," *Journal of Field Robotics (JFR)*, vol. 34, no. 6, 2017.

[9] C. Lehnert, C. McCool, I. Sa, and T. Perez, "Performance improvements of a sweet pepper harvesting robot in protected cropping environments," *Journal of Field Robotics (JFR)*, vol. 37, no. 7, 2020.

[10] B. Arad, J. Balendonck, R. Barth, O. Ben-Shahar, Y. Edan, T. Hellström, J. Hemming, P. Kurtser, O. Ringdahl, T. Tielen, *et al.*, "Development of a sweet pepper harvesting robot," *Journal of Field Robotics (JFR)*, vol. 37, no. 6, 2020.

[11] C. Lehnert, I. Sa, C. McCool, B. Upcroft, and T. Perez, "Sweet pepper pose detection and grasping for automated crop harvesting," in *IEEE Intl. Conf. on Robotics & Automation (ICRA)*. IEEE, 2016.

[12] C. McCool, T. Perez, and B. Upcroft, "Mixtures of lightweight deep convolutional neural networks: Applied to agricultural robotics," *IEEE Robotics and Automation Letters (RA-L)*, vol. 2, no. 3, 2017.

[13] D. Rakita, B. Mutlu, and M. Gleicher, "An autonomous dynamic camera method for effective remote teleoperation," in *ACM/IEEE Intl. Conf. on Human-Robot Interaction (HRI)*, 2018.

[14] C. Lenz, M. Schwarz, A. Rochow, B. Pätzold, R. Memmesheimer, M. Schreiber, and S. Behnke, "NimbRo wins ana avatar xprize immersive telepresence competition: Human-centric evaluation and lessons learned," *International Journal of Social Robotics*, 2023.

[15] C. Smitt, M. Halstead, T. Zaenker, M. Bennewitz, and C. McCool, "Pathobot: A robot for glasshouse crop phenotyping and intervention," in *IEEE Intl. Conf. on Robotics & Automation (ICRA)*. IEEE, 2021.

[16] M. Schwarz and S. Behnke, "Low-latency immersive 6d televisualization with spherical rendering," in *IEEE-RAS Intl. Conf. on Humanoid Robots*. IEEE, 2021.

[17] R. Barth, "Synthetic and empirical capsicum annuum image dataset," 2016.

[18] L. E. Montoya Cavero, "Sweet pepper recognition and peduncle pose estimation," 2021.

[19] M. Halstead, S. Denman, F. Clinton, and C. McCool, "Fruit detection in the wild: The impact of varying conditions and cultivar," in *Digital Image Computing: Techniques and Applications (DICTA)*, 2020.

[20] G. Bradski, A. Kaehler, *et al.*, "Opencv," *Dr. Dobb's journal of software tools*, vol. 3, no. 2, 2000.

[21] L.-C. Florian and S. H. Adam, "Rethinking atrous convolution for semantic image segmentation," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 6, 2017.

[22] M. Grinvald, F. Furrer, T. Novkovic, J. J. Chung, C. Cadena, R. Siegwart, and J. Nieto, "Volumetric Instance-Aware Semantic Mapping and 3D Object Discovery," *IEEE Robotics and Automation Letters (RA-L)*, vol. 4, no. 3, July 2019.

[23] S. Marangoz, T. Zaenker, R. Menon, and M. Bennewitz, "Fruit mapping with shape completion for autonomous crop monitoring," in *IEEE Intl. Conf. on Automation Science and Engineering (CASE)*. IEEE, 2022.

[24] R. Menon, T. Zaenker, N. Dengler, and M. Bennewitz, "NBV-SC: Next best view planning based on shape completion for fruit mapping and reconstruction," in *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2023.

[25] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *IEEE Intl. Conf. on Robotics & Automation (ICRA)*. IEEE, 2011.

[26] M. Schwarz, A. Milan, C. Lenz, A. Munoz, A. S. Periyasamy, M. Schreiber, S. Schüller, and S. Behnke, "NimbRo picking: Versatile part handling for warehouse automation," in *IEEE Intl. Conf. on Robotics & Automation (ICRA)*. IEEE, 2017.

[27] S. R. Buss and J.-S. Kim, "Selectively damped least squares for inverse kinematics," *Journal of Graphics tools*, vol. 10, no. 3, 2005.

[28] D. Coleman, I. Sucan, S. Chitta, and N. Correll, "Reducing the barrier to entry of complex robotic software: a moveit! case study," *Journal of Software Engineering for Robotics*, 2014.