2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Madrid, Spain, October 1-5, 2018

A Novel Autonomous Robot for Greenhouse Applications

Lars Grimstad[1], Rémy Zakaria[1,2], Tuan Dung Le[1] and Pål Johan From[1]

Abstract— **This paper presents a novel agricultural robot for greenhouse applications. In many greenhouses, including the greenhouse used in this work, sets of pipes run along the floor between plant rows. These pipes are components of the greenhouse heating system, and doubles as rails for trolleys used by workers. A flat surface separates the start of each rail set at the greenhouse headland. If a robot is to autonomously drive along plant rows, and also be able to move from one set of rails to the next, it must be able to locomote both on rails and on flat surfaces. This puts requirements on mechanical design and navigation, as the robot must cope with two very different operational environments. The robot presented in this paper has been designed to overcome these challenges and allows for autonomous operation both in open environments and on rails by using only low-cost sensors. The robot is assembled using a modular system created by the authors and tested in a greenhouse during ordinary operation. Using the robot, we map the environment and automatically determine the starting point of each rail in the map. We also show how we are able to identify rails and estimate the robots pose relative to theses using only a low-cost 3D camera. When a rail is located, the robot makes the transition from floor to rail and travels along the row of plants before it moves to the next rail set which it has identified in the map. The robot is used for UV treatment of cucumber plants.**

I. Introduction

Vegetable production in greenhouses is one of the most efficient methods to obtain high quality and fast growing food. The most common products found in greenhouses are high value crops that grow fast in a controlled environment.

The greenhouse environment is well suited to be automated as several tasks can easily be performed using a robot. Examples are autonomous robots for plant treatment such as herbicides and pesticides, collection of data, UV treatment of plants, and so on. In many parts of the world, labor is both costly and a scarce resource that introduces a high level of uncertainty to the production as it is governed by both political and social regulations. Robotics has the potential to deliver more efficient and reliable food production without being reliant on human labor to the same extent as with current production methods.

Applying robots in agriculture has attracted a lot of attention. Much research is focused on the outdoor environment, with applications such as weed detection ([1], [2], [3]), and crop scouting ([4], [5], [6]), but also greenhouse robots can be found in literature. Greenhouse robots have been developed for spraying ([7]), de-leafing ([8]) and harvesting of different crops, such as tomato ([9]), strawberry ([10], [11], [12]) and cucumber ([13]). Some automated commercial system also exist, like the S55 Spray Robot from Wanjet, an automatic robot that can work both on rails and on concrete floors. This is solved by using two systems: a robot that can move on the rails and a carrier to move the robot between the rails. The Micothon Amazone Greenhouse robot is a robot for automatic spraying on rails, but it needs to be manually transported between rails.

Greenhouses normally consist of long rows of plants. Along these rows, many greenhouses have long pipes that are used for heating. These pipes also double as rails for trolleys used for plant treatment and in-field transportation. At the end of the rails, there is normally a concrete floor or similar. This layout and design of the greenhouse represents a challenge to robots, as the robots will have to locomote in two very different operation environments, namely on rails and flat surfaces. This puts constraints both on the mechanical design of the robot and on navigation and path planning.

In our case, we need a robot design that is advanced enough to be able to transition between these two environments and simple and robust enough to be a reliable tool for the farmer. Our prototype robot is assembled from Thorvald II robot modules ([14]) with a custom frame and specially designed double wheels capable of moving on both flat surfaces and rails.

The presented system allows for quick setup of a robot to a new greenhouse. We only need the robot to generate a simple 2D map of the greenhouse headland and to input a few key parameters like rail spacing, and the robot is ready to operate. Rails and robot goal poses used for creating global plans are automatically identified from the map. Locally the robot is able to identify rails and estimate its own pose relative to these using a 3D camera.

There are many tasks in a greenhouse that can be solved using the type of robot presented here. Plant treatment, logistics tasks and picking tasks are just some examples. Different applications may, however, require very different tools to be fitted to the robot. As we want our robot to be of use in as many applications as possible, the robot design includes a quick-release fastening system for tools. As a first application, we have selected treating cucumber with UV lights.

UV light prevents powdery mildew from establishing on plants, and robotic treatment of plants with UV is therefore of high value to the farmer. While the effect of UV on powdery mildew is well known ([15]), less is known about the effect on insects in the crop. For this reason the presented robot will be used in an experiment on the effect of UV on insects

[1]Faculty of Science and Technology, Norwegian University of Life Sciences, Drøbakveien 31, 1433 Ås, Norway lars.grimstad@nmbu.no
[2]Saga Robotics, Sagaveien 3, 1433 Ås, Norway,

in cucumber crops.

The paper is organized as follows: Section II presents an overview of a standard greenhouse and the constraints found in this environment. Sections III and IV gives an overview of the new robot that has been developed for greenhouses, and Section V presents the autonomous planning and navigation in a greenhouse environment. The experimental setup and results are presented in Sections VI and VII, and the concluding remarks in Section VIII.

II. Greenhouse Environment

Greenhouse technology is widely used in food production. However, robots are currently not a common sight in modern greenhouses.

A. Current Technology

Several plants that grow in greenhouses, such as cucumber which is discussed in this paper, are placed in long rows above the ground. In the path between two consecutive rows, a pair of metal pipes is placed firmly on the ground. The pipes are components of the greenhouse heating system, but they are also commonly used as rails for trolleys pushed by workers, and for semi-automatic machines. The semi-automatic machines generally use a motor with a simple controller for driving back and forth on the rails. A worker manually turns the motor on and off to move and stop at desired locations along the path. In Figure 1 and 2, the inside structure of a greenhouse is presented.

B. Benefits of Robots

For robotic solutions to be efficient, the robots need to be able to cover the whole greenhouse without human intervention. Greenhouses are typically large with thousands of plants which require very efficient systems, or even swarms of robots ([16], [17], [18]). The ability for robots to autonomously drive from one row to another enables them to get to where they are most needed without human interaction, increasing the overall efficiency of the system.

Much can be gained by robotizing dangerous and time-consuming tasks in the greenhouse environment. UV light treatment is one such task. It is time consuming, it needs to happen at night, and it requires the operator to wear uncomfortable safety equipment. Lamps are costly, so it is not an option to install fixed lamps in each row or to have a simple trolley for each rail.

III. The Thorvald II Robotic system

Thorvald II is a modular robotic system developed by the authors that enables fast and easy creation of environment-specific agricultural robots. The system is based on several modules that can be put together in arbitrary configurations to create robots with various properties. A robot can easily be reconfigured by using only simple hand tools. This is particularly practical when used as a research platform, as the same robot can be used in widely different environments. A narrow robot configured for running in polytunnels can be reconfigured into a wider robot for running in the open field in a matter of minutes.

The robot's software fully supports the modularity in hardware, and all robots run the same software for driving the base, with the only aspect separating one configuration from another being a configuration file specifying type and relative placement of modules.

A handful of Thorvald II robots are depicted in Figure 3. These robots are all quite conventional configurations. The system also allows for more unconventional designs, like the robot presented in this paper.

Modules relevant for this paper are described in brief below. For a more detailed description of the Thorvald system, we refer the reader to previously published work ([14], [19]).

A. Drive Module

The drive module contains a brushless DC motor connected through a transmission assembly to a wheel at the output. This module is used to propel the robot. Drive modules can be connected directly to the robot's frame, or optionally to a steering module, which allows the module to turn about the vertical axis.

B. Battery Enclosure Module

In addition to housing a battery, this module holds electronic components and acts as a connection point for power and communication to other modules. Several battery enclosures can be connected in parallel to increase a robot's range. The main electric components are located in this module. These are essential for any Thorvald II robot, and must be included.

C. Motor Controller Module

A motor controller module is used for running up to two drive modules. In addition to a two channel motor controller, the module contains electric components to make the module adhere to the standardized Thorvald II connections used by all modules driving motors.

IV. Environment-Specific Robot Configuration

The presented robot has been assembled from Thorvald II modules and components in combination with a few custom parts. The differential drive robot uses two standard Thorvald II drive modules for propulsion. As the robot must be able to locomote on concrete as well as on rails, the drive modules have been fitted with specially designed double wheels. The inner wheel on each set has a diameter of 350 mm, and is used for driving on flat surfaces. The outer wheel has a diameter of 240 mm, is made from POM-c, and is used for driving on rails. The drive modules are fitted to a simple custom frame made from sheet metal, which contains components normally found inside the Thorvald II battery enclosure module and motor controller module.

Passive rail wheels and caster wheels have been added to the rear of the frame to support the robot when it is driving on rails and concrete floors, respectively. The robot has a footprint of 0.92 m x 1.05 m and a mass of 147 kg (including one battery pack). The robot is depicted in Figure 4.

Fig. 1: A panorama view of inside a greenhouse

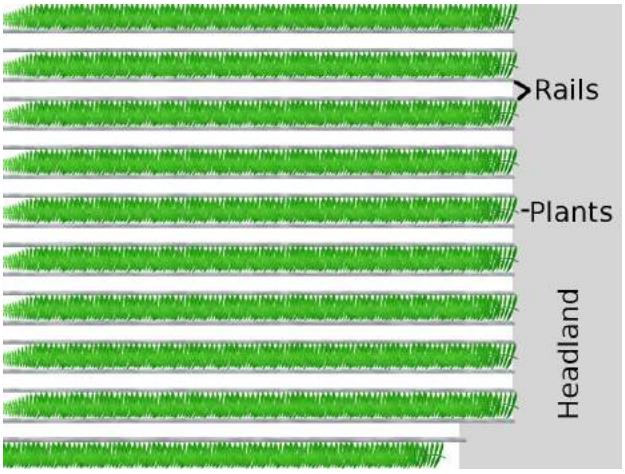


Fig. 2: Schematic view of a greenhouse

Fig. 3: Various robots made from Thorvald II modules

The robot can easily be modified to fit various rail widths commonly found in greenhouses. This is done by shifting the drive modules sideways, or by mounting the wheel inside-out on the drive module, so that the wheels for driving on flat surfaces are outside the rail wheels. Different track widths are shown in Figure 5.

(a) **(b)**

Fig. 4: The presented robot is designed for use in a greenhouse environment. (a) The robot on rails; (b) the robot on the headland floor. One of the two drive modules and the custom sheet metal frame has been highlighted.

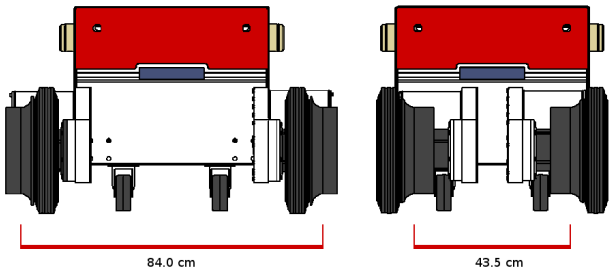


Fig. 5: The robot's drive modules can slide to adjust track width. Mounting the wheels with the other side out increase the range of achievable widths even further. It is also possible for the two drive modules to swap places. Thus the robot can be used with a wide range of commonly found rail systems.

A. Power System

The presented robot shares electric components with other Thorvald II robots. This includes configuration-independent circuit boards and connectors for connecting modules together. The robot can house up to two 48 V, 70 A h batteries, but is currently fitted with only one, as this holds enough energy to complete UV treatment in our current greenhouse. The robot is equipped with a power inverter for powering UV lights. This is assumed to become obsolete in the future, as we intend to migrate to LED lights when these become available.

(a) (b) (c)

Fig. 6: As the robot is intended to be used to solve many tasks, a quick-release tool connection system has been designed. (a) The robot with a UV light tool; (b) closeups of the mechanical interface between the platform and the tool; (c) the robot suspended from the roof. Here the tool connection is supporting the full weight of the robot.

B. Software

The robot also runs the exact same software as other Thorvald II robots for driving the base, calculating odometry, generating the robot description etc. Software components shared with other Thorvald robots are described elsewhere ([20]). The details on the workings of additional task-specific programs for navigating in a greenhouse environment are described in this paper.

C. Tool Connection

As described above, there are several tasks that are suited to be robotized in a greenhouse. Different tasks will require the use of different tools. Therefore, to make the robot more versatile, a tool mounting system has been designed for the robot. Four short cylinders, each with a cone shaped cavity at the top, are fixed to the upper part of the robot's chassis. Four studs with steel balls are fixed to the bottom of each tool. When a tool is mounted to the robot, the steel balls rest in the cone-shaped cavities. A latch on either side presses the steel balls down and locks the tool in place. The system is forgiving when it comes to lack of accuracy when placing the tool on the robot, as gravity helps slide the steel balls into place in the cones. This allows for quick and easy tool change. Once connected, the tool sits firmly on the robot. As depicted in Figure 6, the connection is strong enough to support the entire weight of the robot.

D. Sensors

The presented robot aims to be a low cost autonomous platform for solving a wide range of tasks in the greenhouse. To this end, the robot has been equipped with several sensors. The sensors are depicted in Figure 7 and briefly discussed below.

1) Camera: Due to the nature of UV light treatment task, the robot must be able to operate in dark environment. Hence, a RGB camera can not be used here. An Intel RealSense ZR300 ([21]) is chosen. It is a stereo RGBD camera system, which consists of two infrared (IR) cameras, one color camera and one fisheye optical sensor. The ZR300 also includes a 6 degree of freedom (DOF) inertial measurement unit (IMU). The ZR300 uses an active texture projector to create unstructured light illuminations for obtaining depth information from two IR cameras. Depth information is then converted to pointcloud data for object detection.

Fig. 7: The robot's sensors

2) Ultrasonic Range Finders: The robot has an array of down-facing ultrasonic range finders in front and between the drive wheels. These sensors are used for detecting whether the robot is driving on flat floor or on rails, and to prevent the robot from driving outside an edge. When the robot is transitioning from flat floor to rails, the sensors are used to verify that the robot has aligned correctly to the rails. If the robot for some reason should miss the rails, the sensors will see the edge of the concrete floor in good time to stop the robot.

3) 2D LIDAR: A front-facing Hokuyo UTM-30LX-EW scanning laser rangefinder is mounted on the top of the robot. The robot uses this sensor for mapping and localizing in the map when driving from one set of rails to the next. The sensor is also used for obstacle detection.

4) IMU and Odometry: Even though there is already an IMU rigidly attached to the ZR300 camera, the ZR300 is mounted tilting toward the ground. This makes acceleration and gyroscope measurements from the IMU contain more noise when transforming to the robot body frame, which in turn, might worsen the robot's localization. Therefore, an additional UM7 IMU is employed. The UM7 is mounted flatly on the robot body frame, parallel to the ground floor. Its measurements are used for robot mapping and localization. The two drive modules are equipped with encoders, measurements from which are used for estimating robot velocity.

5) Safety Bumper: The robot has a front-mounted bumper, which only purpose is to stop the robot in case all other systems fail. If the bumper is depressed, it will require a human operator to reactivate the robot.

V. Planning and Navigation

In this section we first give an overview of the planning and navigation for the robot when it is operating in the greenhouse. We then describe in detail how each component of the system works.

A. System Overview

There are several problems that must be solved for the robot to successfully navigate in a greenhouse environment,

i) the robot must be able to navigate between sets of rails on a flat floor, ii) it must be able to transition from the flat floor to rails, iii) it must be able to know when it has reached the end of the rail and should start reversing back towards the headland, and finally iv) the robot must be able to detect when it has reached the flat floor.

Here we give a short overview of how the robot overcomes the challenges listed above, using UV treatment as our application. The most important steps are explained in more detail below.

For the UV application, the robot needs to drive back and forth on every rail in the greenhouse once. For each plant to receive the correct dose of UV light, the robot must maintain a speed of $0.5\,\mathrm{m/s}$ when driving on the rails. At the end of the rail, the UV lights should be turned off before the robot reverses back to the headland.

1) Navigating on the Headland: First, the robot uses 2D LIDAR and odometry together with a map (generated by the robot itself using a standard SLAM approach) and automatically generated target poses to navigate and roughly position itself in front of a given set of rails.

2) Transitioning to Rails: To transition from the floor to the rails, the robot must be aligned to, and directly in front of the rails. The wheels allow for a maximum transverse misalignment of approximately $\pm 3\,\mathrm{cm}$. For this level of accuracy in localization, the map is too coarse. The robot therefore uses its front-mounted 3D camera to identify the rails, and estimates its own pose in the rail coordinate frame. The robot then calculates a set of rotations and translations that need to be executed in order to reach its desired pose in front of the rails. With odometry as feedback, it rotates so that it is aligned to the rails, reverses back, rotates towards the rails, drives forward and rotates to align to the rails. It then uses data from the 3D camera once again to estimate the resulting pose in the rail frame. If the robot's pose is outside the preset limits, it will retry the alignment. If the pose is within the preset limits, the robot will attempt to enter the rails.

When attempting to enter the rails the robot slowly translates towards the rails, using its down-facing ultrasonic range finders to sense the floor in front of it. If the robot should detect the edge of the concrete floor in front of a rail wheel, and not a rail, it will back out and retry the alignment step. The robot has successfully entered the rail when it senses that the rail wheels are on the rails and the front of the robot has passed the edge of the headland floor.

If the robot should fail to identify the rails after navigating to the target pose in the map, it will start searching by rotating in place, altering between ever increasing clockwise and counterclockwise rotations (up to a certain limit) until it detects the rails.

3) Driving on the Rails: Before the robot starts to drive, it stores its current pose in the map. This is given as the robot's pose in the map on return.

For the UV light application, the robot turns on the UV lights and drives forward on the rails at constant speed. It turns off the UV lights and starts reversing back when it reaches the end of the rail. The robot estimates the location of the end of the rail by sensing the distance to the wall on the far side of the greenhouse from 2D LIDAR data.

From odometry, the robot has an estimate of the length of the current rail set. As we do not want the robot to exit the rails at full speed it slows down after reversing 90 % of the rail length. The down-facing range finders senses the transition from rails to floor.

The array of down-facing range finders are used to constantly monitor the type of surface the robot is driving on. Based on this, the wheel radius used when calculating commands and odometry is automatically set to the radius of whichever wheel set is propelling the robot.

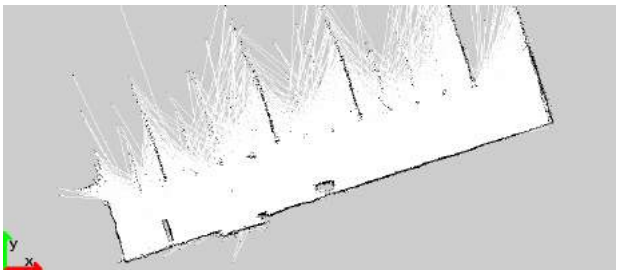
B. Mapping and Identifying Rails in Map

The robot uses a map and 2D LIDAR to localize when moving on the headland floor. A global costmap is generated from the map, and a local costmap is generated from the robot's 2D LIDAR data. If a target pose is specified in the map frame, the robot will plan a trajectory based on its global and local costmaps, and move to the target pose, updating its trajectory if it encounters unforeseen obstacles. When the robot is tasked with moving from one set of rails to the next on the headland floor, it therefore needs goals to be defined in front of the next set of rails in the map frame. To manually define a target pose in front of each rail is cumbersome and would require too much time to set up. The process of defining target poses in front of greenhouse rails has therefore been automated.

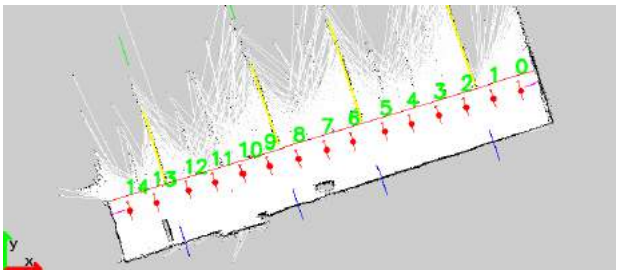
First we create a map. As we only use this map when the robot is moving on the greenhouse headland, only this part of the greenhouse needs to be mapped. The map is created with GMapping ([22]), which takes 2D LIDAR data and odometry as input and outputs a grid map. The map is recorded only once for each greenhouse, and driving back and forth on the headlands yields an adequate map in a matter of minutes. The finished map is stored to be used by the robot to localize on the headlands. The map is depicted in Figure 8a.

We now need to define target robot poses in front of each rail in the map. As this is a 2D map, the rails in-between the rows of plants are not visible. We do however have four clearly visible lines in the map. These lines are components of the greenhouse ventilation and heating system, and are located underneath some of the plant rows. This means that they are centered between two neighboring sets of rails and can thus be used for estimating rail positions.

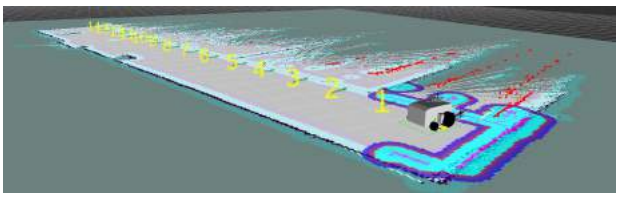
First we run through the map and cluster together neighboring occupied cells, throwing away clusters containing too few cells. We then fit lines to each cluster using least squares regression, and, judging by correlation coefficients, assume the best lines to be ventilation lines or plant rows. We then search for the rear wall along lines to determine which point in each cluster is at the start of the plant row. These points are all at the edge of the headland floor. We can now sort the lines by where they appear along this edge. We search to the sides of the outer lines to determine the location of the side walls. As we know the distance between sets of rails,



(a)



(b)



(c)

Fig. 8: (a) A map of the greenhouse headland; (b) target poses (red arrows) in front of the rails have been generated and the location of the edge of the headland floor has been estimated; (c) the greenhouse, seen by the robot as it arrives at target pose 0 in front of the outer rail set. Here we see all the target poses. A virtual wall in the costmaps prevents the robot from planning outside the edge of the headland.

we can calculate how many rail sets are located between neighboring lines, including end walls. We now estimate the origin and orientation of every rail frame in the map frame. Target robot poses in the map frame are then outputted according to desired poses in the rail frames and read by the main program on startup.

The map analyzer also outputs the estimated location of the end of the headland floor. The main program uses this information to include a virtual wall in the robot's costmaps. This prevents the robot from driving outside the edge of the concrete floor when navigating between sets of rails. Figure 8b shows the analyzed map. Figure 8c shows how the robot is using the extracted information.

The map is analyzed only once for each greenhouse. Individual poses can be adjusted by the user if needed. Thus far, there has not been a need for manual corrections in the greenhouse where the robot is being tested.

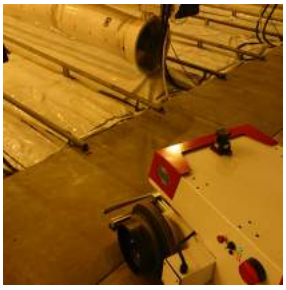
C. On-Line Rail Identification Using 3D Camera

For rail detection, point cloud data $\mathcal{Q}_k$ generated by the ZR300 camera is used to determine the position and orientation of the robot's base link frame $\mathcal{F}_b$ as seen form the rail frame $\mathcal{F}_r$. For a given set of rails, we define the origin of $\mathcal{F}_r$ to be between, and at the very start of the rails with x-axis in the direction of the rails and the z-axis directly up. The origin of $\mathcal{F}_b$ is in the middle of the drive wheels (front wheels) with the x-axis forward and z-axis up. The algorithm for rail detection is illustrated in Algorithm 1. The output from this algorithm is filtered so that only estimates that are consistent with the previous estimate are let through. Figure 9 shows the robot identifying a set of rails.

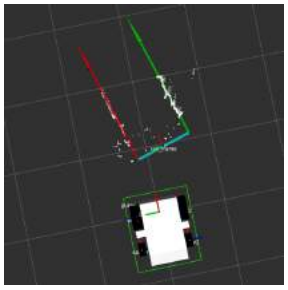
As we cannot guarantee that the camera is perfectly mounted to the robot, we need to be able to correct for error in camera position and orientation. To calculate this error, the robot is placed on a set of rails. Here the robot is perfectly aligned to, and centered on the rails, and should therefore observe a pose of zero y and zero *yaw* in the rail frame. The robot uses this assumption to calculate correction values.

Algorithm 1: Rail fitting

input : $\mathcal{Q}_k$
output: $\mathcal{F}_b$ in $\mathcal{F}_r$
begin
  Step 1: Transform $\mathcal{Q}_k$ to $\mathcal{F}_b$ and crop volume to expected height of rails;
  Step 2: Cluster points from $\mathcal{Q}_k$ and throw away clusters with too few points ;
  Step 3:
  for *each cluster* **do**
    Perform least square line regression on x and y values;
  end
  Step 4: Select the line L with the best r value
  if *rail is found in expected range to either side of L* **then**
    Calculate origin and orientation of $\mathcal{F}_b$ in $\mathcal{F}_r$
    Return $\mathcal{F}_b$ in $\mathcal{F}_r$;
  else
    Rail frame not found ;
  end
end



(a)



(b)

Fig. 9: The robot is identifying rails using a 3D camera. (a) The robot close to a set of rails; (b) the rails successfully identified by the robot.

Fig. 10: Ground truth poses are obtained using a total station

VI. EXPERIMENTAL SETUP

To evaluate the robot's performance, we conduct a series of tests inside a greenhouse.

A. Rail Detection Algorithm

When the robot is to transition from floor to rails, it is critical that it is able to obtain an accurate estimate of its pose relative to the rails. It is therefore necessary to evaluate the performance of our rail detection algorithm. To this end, we place the robot within sight of the rails. We then use a Leica TCA 1100 robotic total station (Figure 10) and prisms to obtain a ground truth measurement of the robot's pose to the nearest millimeter in the rail coordinate frame. This pose is then compared to the pose estimated by the robot. We then move the robot to a new pose and repeat.

B. Edge Detection

If the robot for some reason should try to enter the rails without proper aligning itself to the rails, it is important that it is able to detect the edge of the floor. This redundancy is important to guarantee safe operation. This is tested by forcing the robot to try entering a set of rails which it is not aligned to.

C. Overall Performance

To establish whether our complete system works as expected, we let the robot perform a cycle of the UV light treatment. At the time of testing, the plants in the greenhouse are too young for UV according to our treatment schedule. The light tool is therefore not mounted to the robot, and turning on and off the UV lights is indicated by sound from the robot's on-board speaker.

VII. RESULTS

In this section we present the results that were obtained during the tests in the greenhouse.

A. Rail Detection Algorithm

In Table I, the locations of the robot with respect to the rails, as calculated by the robot, and the ground truth obtained with the total station are presented. This gives us a measurement of how well we can identify the robot's pose in the rail frame. One can see that the estimation error in the y-direction and *yaw* are small, while the error is larger in the x-direction. The largest absolute error measured in y was 0.04 m, and the RMS values was 0.02 m. For *yaw*, the results were 0.032 rad and 0.014 rad, for maximum absolute error and RMS, respectively. This is sufficiently accurate to be used when the robot is to align with the rails. With a maximum absolute error of 0.24 m and a RMS value of 0.11 m, the error in the x-direction is larger. This is as expected, and of minor concern, as the robot only considers the error in y and *yaw* when aligning.

TABLE I: Ground truth and estimated positions of the robot in the rail coordinate frame

Ground Truth			Robot's estimate		
x [m]	y [m]	yaw [rad]	x [m]	y [m]	yaw [rad]
-0.639	-0.005	0.002	-0.765	-0.015	0.021
-0.640	-0.002	-0.002	-0.762	-0.003	0.007
-0.636	0.000	0.004	-0.764	-0.002	0.013
-0.848	0.029	-0.014	-0.773	0.038	-0.021
-0.846	0.003	-0.014	-0.794	0.025	-0.020
-0.844	-0.004	0.000	-0.800	0.010	-0.007
-0.558	0.077	-0.037	-0.609	0.117	-0.055
-0.554	-0.009	-0.002	-0.733	0.011	-0.012
-0.548	-0.017	0.007	-0.786	-0.023	0.023
-0.792	-0.086	0.045	-0.768	-0.077	0.053
-0.791	-0.035	0.014	-0.794	0.004	-0.018
-0.787	-0.037	0.018	-0.797	-0.023	0.021

What is measured here is the robot's ability to estimate its own pose in the rail frame, not its ability to correct the estimated error in pose. From running the UV light program, we observed that the number of alignment corrections varied from row to row. For some rows the robot would successfully align itself to the row on first try. For other rows the robot would repeat the rail alignment several times before observing an error sufficiently low for it to enter the rail.

B. Edge Detection

When provoked into attempting to enter a set of rails when misaligned, the robot detected the edge off the floor in good time to stop the robot. On a few occasions the robot falsely detected a floor edge even though it was properly aligned to the rails and would have made the transition without problems. On no occasions did the robot mistake an edge for a rail, which means that the robot never concluded to be on a rail when it in reality had missed.

C. Overall Performance

The automatically generated target poses in the map were accurate enough to place the robot well within sight of the rails for all the tested rows. No manual corrections were necessary. Once the robot navigated to the target pose, the rail detecting algorithm identified the robot pose with respect to the rails with sufficient accuracy to allow the robot to drive onto the rails. Most of the times the robot needed few attempts to align with the rails, and only occasionally did it require several tries.

VIII. CONCLUSIONS

In this paper we have presented a fully working robot for autonomous operation in greenhouses. The robot is able to automatically estimate locations of rails in a map, and it can also identify rails by using its on-board sensors. We have verified that our system works through field trials

conducted in a greenhouse during normal production. The robot successfully navigated in the greenhouse environment, transitioning between rails and headland floor, and moving from rail set to rail set. We showed that the presented algorithm for rail identification enables the robot to accurately estimate its pose relative to the rails.

The mechanical design includes a novel system for attaching tools to the platform, which greatly increase the usability of the robot. We have also showed that robots for greenhouses can be constructed using the Thorvald II modular system without making platform-specific alterations to code or the electric system.

REFERENCES

[1] M. Di Cicco, C. Potena, G. Grisetti, and A. Pretto, "Automatic model based dataset generation for fast and accurate crop and weeds detection," *International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[2] A. Milioto, P. Lottes, and C. Stachniss, "Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in cnns," *International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[3] P. Lottes and C. Stachniss, "Semi-supervised online visual crop and weed classification in precision farming exploiting plant arrangement," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2017.

[4] D. Albani, J. IJsselmuiden, R. Haken, and V. Trianni, "Monitoring and mapping with robot swarms for agricultural applications," in *Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on*. IEEE, 2017, pp. 1–6.

[5] M. Popovic, T. Vidal-Calleja, G. Hitz, I. Sa, R. Siegwart, and J. Nieto, "Multiresolution mapping and informative path planning for uav-based terrain monitoring," *International Conference on Intelligent Robots and Systems (IROS), 2017*, 2017.

[6] T. Mueller-Sim, M. Jenkins, J. Abel, and G. Kantor, "The robotanist: a ground-based agricultural robot for high-throughput crop phenotyping," in *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE, 2017, pp. 3634–3639.

[7] P. J. Sammons, T. Furukawa, and A. Bulgin, "Autonomous pesticide spraying robot for use in a greenhouse," in *Proceedings of the Australian Conference on Robotics and Automation, Sydney, Australia*, 2005.

[8] E. Van Henten, B. Van Tuijl, G.-J. Hoogakker, M. Van Der Weerd, J. Hemming, J. Kornet, and J. Bontsema, "An autonomous robot for de-leafing cucumber plants grown in a high-wire cultivation system," *Biosystems Engineering*, vol. 94, no. 3, pp. 317–323, 2006.

[9] Y.-C. Chiu, P.-Y. Yang, and S. Chen, "Development of the end-effector of a picking robot for greenhouse-grown tomatoes," *Applied engineering in agriculture*, vol. 29, no. 6, pp. 1001–1009, 2013.

[10] S. Yamamoto, S. Hayashi, H. Yoshida, and K. Kobayashi, "Development of a stationary robotic strawberry harvester with a picking mechanism that approaches the target fruit from below," *Japan Agricultural Research Quarterly: JARQ*, vol. 48, no. 3, pp. 261–269, 2014.

[11] F. Dimeas, D. V. Sako, V. C. Moulianitis, and N. A. Aspragathos, "Design and fuzzy control of a robotic gripper for efficient strawberry harvesting," *Robotica*, vol. 33, no. 5, pp. 1085–1098, 2015.

[12] S. Hayashi, K. Shigematsu, S. Yamamoto, K. Kobayashi, Y. Kohno, J. Kamata, and M. Kurita, "Evaluation of a strawberry-harvesting robot in a field test," *Biosystems engineering*, vol. 105, no. 2, pp. 160–171, 2010.

[13] E. J. Van Henten, J. Hemming, B. Van Tuijl, J. Kornet, J. Meuleman, J. Bontsema, and E. Van Os, "An autonomous robot for harvesting cucumbers in greenhouses," *Autonomous Robots*, vol. 13, no. 3, pp. 241–258, 2002.

[14] L. Grimstad and P. J. From, "The thorvald ii agricultural robotic system," *Robotics*, vol. 6, no. 4, 2017. [Online]. Available: http://www.mdpi.com/2218-6581/6/4/24

[15] A. Suthaparan, A. Stensvand, K. Solhaug, S. Torre, L. Mortensen, D. Gadoury, R. Seem, and H. Gislerød, "Suppression of powdery mildew (podosphaera pannosa) in greenhouse roses by brief exposure to supplemental uv-b radiation," *Plant disease*, vol. 96, no. 11, pp. 1653–1660, 2012.

[16] A. Martinoli, "Swarm intelligence in autonomous collective robotics: From tools to the analysis and synthesis of distributed control strategies," *Unpublished doctoral manuscript, EPFL Ph. D. Thesis*, no. 2069, 1999.

[17] R. H. Byrne, J. J. Harrington, S. E. Eskridge, and J. E. Hurtado, "Cooperating mobile robots," Feb. 3 2004, uS Patent 6,687,571.

[18] P. R. Wurman, R. D'Andrea, and M. Mountz, "Coordinating hundreds of cooperative, autonomous vehicles in warehouses," *AI magazine*, vol. 29, no. 1, p. 9, 2008.

[19] L. Grimstad and P. J. From, "Thorvald ii - a modular and reconfigurable agricultural robot," in *IFAC 2017 World Congress*, 2017.

[20] ——, "A configuration-independent software architecture for modular robots," in *Proceedings of the Australian Conference on Robotics and Automation, Delft, Netherlands*, 2018.

[21] L. Keselman, J. I. Woodfill, A. Grunnet-Jepsen, and A. Bhowmik, "Intel realsense stereoscopic depth cameras," *CoRR*, vol. abs/1705.05548, 2017.

[22] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.