

# On-the-Go Tree Detection and Geometric Traits Estimation with Ground Mobile Robots in Fruit Tree Groves

Dimitrios Chatziparaschis,<sup>†</sup> Hanzhe Teng,<sup>†</sup> Yipeng Wang,<sup>†</sup> Pamodya Peiris,<sup>†</sup>  
Elia Scudiero,<sup>‡</sup> and Konstantinos Karydis<sup>†</sup>

**Abstract**—By-tree information gathering is an essential task in precision agriculture achieved by ground mobile sensors, but it can be time- and labor-intensive. In this paper we present an algorithmic framework to perform real-time and on-the-go detection of trees and key geometric characteristics (namely, width and height) with wheeled mobile robots in the field. Our method is based on the fusion of 2D domain-specific data (normalized difference vegetation index [NDVI] acquired via a red-green-near-infrared [RGN] camera) and 3D LiDAR point clouds, via a customized tree landmark association and parameter estimation algorithm. The proposed system features a multi-modal and entropy-based landmark correspondences approach, integrated into an underlying Kalman filter system to recognize the surrounding trees and jointly estimate their spatial and vegetation-based characteristics. Realistic simulated tests are used to evaluate our proposed algorithm’s behavior in a variety of settings. Physical experiments in agricultural fields help validate our method’s efficacy in acquiring accurate by-tree information on-the-go and in real-time by employing only onboard computational and sensing resources.

## I. INTRODUCTION

Real-time tree crop monitoring is important in agriculture because it allows growers to make informed decisions about management practices, address potential issues promptly, and ultimately optimize crop yields and profitability. Growth rates, yields, and fruit quality can vary significantly on a tree-by-tree basis in commercial orchards due to the spatiotemporal variability of environmental factors such as soil conditions and microclimate [1], as well as management practices (e.g., non-uniform irrigation distribution [2]) and their interaction with individual tree genetic variability [3]. Unfortunately, it is impractical to directly measure the status (e.g., leaf/stem water potential, fruit quantity and quality, etc.) over time and for every single tree. Thus, different types of technology have been employed to gather indirect (i.e. proximal and remote sensing) by-tree data in an efficient and scalable manner.

Unmanned Aerial Vehicles (UAVs), in particular, have been deployed in farms to acquire information from above and map field characteristics [4]–[7]. Yet, the accuracy of

<sup>†</sup> Dept. of Electrical and Computer Engineering, and <sup>‡</sup> Dept. of Environmental Sciences, Univ. of California, Riverside, 900 University Avenue, Riverside, CA 92521, USA; {dchat013, hteng007, ywang1040, ppeir022, scudiero, karydis}@ucr.edu.

We gratefully acknowledge the support of NSF # IIS-1901379 and # CMMI-2046270, USDA-NIFA # 2021-67022-33453, ONR # N00014-18-1-2252 and # N00014-19-1-2264, The University of California # UC-MRPI M21PR3417, a Frank G. and Janice B. Delfino Agricultural Technology Research Initiative Seed award and an OASIS-IFA award. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

these approaches is highly connected to the resolution of the used sensors and flight altitude. Moreover, UAVs are not practical for collecting data from under or on the sides of tree canopies. More recently, mobile robots have been used for in-field near-ground sensing to collect various types of data about tree geometric traits and health markers [8], [9].

Of interest to this work is in-situ proximal sensing [10] with ground robots. In our previous works we have considered proximal sensing of soil apparent electrical conductivity which can help estimate spatial variability of many agronomically-relevant soil properties, such as soil moisture, particle size fraction, and salinity [11]–[13], and developed a leaf visual detection and pose estimation system which can assist toward plant phenotyping [14], [15]. We envision that through data fusion from multiple ground (and remote) sensors, we may apply current automated sampling methods [16] and develop new models to assess orchard status on-the-go.

In this work, we develop a real-time algorithmic framework for tree-crop proximal sensing of plant geometric and vegetation index markers by using a semi-autonomous ground mobile robot in large-scale field environments. Our robot is equipped with a 3D LiDAR sensor and an RGN (Red-Green-NIR) camera module, and performs real-time fusion of the two sensor modalities to obtain and identify the surrounding tree candidates and extract their geometric and vegetation index information. We present a novel landmark association approach based on both 2D and 3D sensing modalities to obtain real-time tree correlations and match them with globally georeferenced trees to update their status in the surveying map. In sum, this work contributes:

- A real-time LiDAR-camera fusion algorithm to obtain accurately the surrounding tree geometric traits and georeference attained information on a global map.
- A multivariate, entropy-based correspondences algorithm, integrated into an underlying Kalman Filter, for robust 3D tree cluster matching for localization.
- Robot-assisted proximal RGN sensing from the ground enabling integration of domain-specific vegetation data for active landmark detection outdoors.

Our framework is deployed and tested experimentally in the field under real-world conditions (e.g., significant wind gusts and frequent ambient light variations). Algorithm efficacy is also tested within realistic digital twins over multiple distinctive case studies. In both settings, we compare our system with a conceptually-related tree detection methodology and present the results. Our whole system is autonomous-

ready, e.g., by employing Global Navigation Satellite System (GNSS) and waypoint navigation, to help stimulate further agricultural robotics research, but it can be integrated directly into existing commercial (semi-) autonomous agricultural vehicles (e.g., smart tractors) as well.

## II. RELATED WORKS

Tree detection and by-tree geometric traits estimation have been active research areas in both remote sensing and robotics. Typically (airborne) data are collected and then post-processed (e.g., [17]–[20]) to obtain counts of observed trees, crop density and growth information, and field properties (e.g., soil moisture). Machine learning approaches have been employed as well [21]–[24] on imagery and point cloud data. Using offboard processing, Zhang *et al.* [25] developed a method for standing tree stem detection in rich LiDAR data based on the point cloud’s surface curvature and a segmentation-based approach. Also, Roy *et al.* [26] demonstrated an offline method to merge reconstructed views from two sides of an orchard row into a unified 3D model.

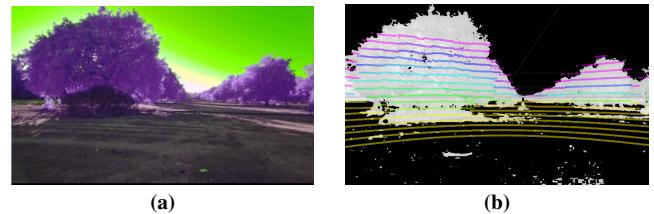
In studies involving robot-assisted proximal sensing, trees are often used as landmarks for localization. In most cases, tree detection relies on trunk recognition, as the trunk serves as a sturdy landmark in dynamic environments. Jelavic *et al.* [27] showcased a robust and lightweight method for tree detection based on Principal Component Analysis (PCA) on the tree clusters, given 3D LiDAR scans in local forest patches. Shalal *et al.* [28] demonstrated a mobile robot with an integrated 2D LiDAR sensor and a color camera, and Yu *et al.* [29] presented a localization system based on fusion of camera and ultrasonic data. In the same context, Durand-Petiteville *et al.* [30] used four stereo cameras to perceive tree trunks through the formed concavities in the point cloud space. Although these studies demonstrated practical applications in tree detection, it is notable that trunks may not always be directly visible in the scene, as the latter can be drastically affected by dense, dynamic, and verdant environmental conditions.

To this end, several works have proposed using trained detectors for trunk (or other tree parts) to improve robustness. Wang *et al.* [31] presented a PointRCC detector that is trained on stereo data (pseudo-lidar representation) to identify trees in a forest environment. Liu *et al.* [32] demonstrated a monocular camera and a LiDAR system for fruit counting and tree trunk detection method, based on Convolutional Neural Networks. Also, Semantic LiDAR Odometry and Mapping (SLOAM) [33] has an integrated tree detection algorithm based on trunk detection, which is trained on a custom dataset created in forest environments. However, the performance of these approaches highly depends on additional computational resources, precise training, and the generalizability of the method to be robust in different environments and physical conditions of the trees.

To the author’s best of knowledge, there is a lack of works on full tree characteristics estimation from the ground utilizing vegetation-enabled features. We present our approach to localizing tree landmarks, associating them with existing



**Fig. 1:** (a) Clearpath Jackal equipped with a Velodyne VLP-16 LiDAR and a MAPIR Survey3N RGN camera during field experiments. (b) The designed simulation world in Gazebo along with the simulated Jackal mobile robot. This world model contains a  $2 \times 3$  tree grid of 3 orange and 3 lemon trees placed at a  $5 \text{ m width} \times 5 \text{ m length}$  relative distance from each other.



**Fig. 2:** Snapshot from robot operation in the orchard. (a) RGN raw imagery data. (b) Corresponding binary thresholded NDVI frame with projected 3D LiDAR data colorized based on  $z$ -level value.

tree candidates, and spatially characterizing them without exclusively relying on any specific part of their structure. We also test against the most relevant system [27], since it can obtain full tree-cluster dimensions during the navigation of a robotic harvester in a forest.

## III. SYSTEM DESCRIPTION

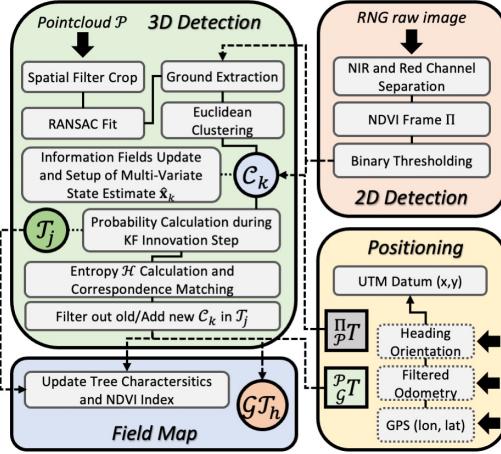
We use the Clearpath Jackal wheeled mobile robot (Figure 1a).<sup>1</sup> The robot has body dimensions of  $50.8 \text{ cm} \times 43.2 \text{ cm} \times 25.4 \text{ cm}$  and weighs  $17 \text{ kg}$ . It features an NVIDIA Xavier AGX internal PC, a GPS receiver, an integrated IMU sensor, and two wheel encoders. It is equipped with a 3D LiDAR sensor (Velodyne VLP-16) and a RGN camera (MAPIR Survey3N); both are affixed to the robot’s chassis.

**3D LiDAR:** The publication rate of the LiDAR sensor was set to  $10 \text{ Hz}$  thus providing up to 300,000 points per second. Each point in the pointcloud  $\mathcal{P} \subseteq \mathbb{R}^3$  holds also information about the 3D LiDAR  $ring\_id$  that was detected.

**RGN Camera:** The main purpose of the RGN camera is the inclusion of Near Infrared (NIR) band in imaging. Given readings in the red and green channels of the sensor, NIR provides critical information used to compute the Normalized Difference Vegetation Index (NDVI), defined as  $NDVI = (NIR - Red) / (NIR + Red)$ . NDVI can help pinpoint and quantify areas with lower chlorophyll status and vegetation health, since it is a ratio of near-infrared radiation (highly reflected by vegetation) and red light (mainly absorbed by vegetation) [34].

The camera provides 30 fps of 720p RGN feed through a USB connection directly to the onboard computer. Figure 2a shows a snapshot of the raw RGN imagery information captured in the field. During field survey, we use the reflectance calibration ground target package (V2) to regulate

<sup>1</sup> We note that the algorithm developed herein is general and it can be employed with other wheeled robots that can operate in the field too.



**Fig. 3:** System diagram of the proposed 2D-3D fusion system with main sub-components, along with the positioning node, the field map node, and the raw sensory input streams. The solid lines indicate the inner connectivity of the components and the dotted arrows the information exchange between the developed nodes.

the minimum and maximum reflectance values and obtain NDVI readings in the low-high normalized health range of  $[-1.0, 1.0]$ , respectively. We also calibrated the camera to acquire intrinsic properties and lens distortion coefficients [35], used for 3D LiDAR points' projection on the RGN image plane. Let  $q_{\mathcal{O}} \in \mathcal{O} \subseteq \mathbb{R}^3$  be the 3D points in the camera's coordinate system and  $q_{x,y} \in \Pi \subseteq \mathbb{R}^2$  their representation on the image plane on pixel position  $(x, y)$ .

**LiDAR – RGN Camera Extrinsic Calibration:** Key in our LiDAR-camera fused detection approach (Section IV) is to define the relative pose of the two sensors (i.e. extrinsic calibration). To do so, we employed the package developed in [36], which is a scene-based calibration approach to estimate the extrinsic parameters and obtain the LiDAR-to-camera frame transformation matrix  $\Pi_T \in SE(3)$ . Figure 2b shows projected LiDAR points on the NDVI frame.

**Positioning in Global and Local Frames:** To correlate detected tree candidates in the environment with ground truth tree landmarks, we employ the robot localization software [37] from the ROS [38] navigation stack. It uses the robot's GNSS data, the earth-referenced heading, and odometry information to generate a Cartesian datum in the Universal Transverse Mercator (UTM) coordinate system. This aligns with our experiments (Section V) as we assume access to tree field geomaps and the robot's initial pose.

#### IV. PROPOSED ALGORITHM

Figure 3 illustrates the overall flow diagram of our framework. The main parts of the framework include multi-modal data fusion, correspondence matching, and estimation of desired traits (herein tree height and width).

##### A. 2D and 3D Data Fusion and Tree Detection

The overall goal of our algorithm is to detect and recognize  $M$  tree candidates,  $\mathcal{T}_j$  with  $j \in [1, M]$ , in a captured point cloud  $\mathcal{P}$ , and match them with  $N$  global georeferenced trees on the field,  $\mathcal{GT}_h$  with  $h \in [1, N]$ , to update their width and height information in real-time and on-the-go.

---

##### Algorithm 1: Tree Detection, Identification, and Characteristics Estimation

---

```

input : key_params,  $v_x, \omega_\theta$ 
output: Global georeferenced trees  $\mathcal{GT}$  with their characteristics
1 define  $\mathcal{T} = \emptyset$  tree candidates/landmarks in memory;
2 while newly received point cloud  $\mathcal{P}$  do
3    $\mathcal{C}_k = \text{compute\_clusters}(\mathcal{P})$ ;
4   for all clusters  $k$  in  $\mathcal{C} \subseteq \mathcal{P}$  do
5      $m_k = \text{compute\_characteristics}(k, \text{key\_params})$ ;
6     if  $\mathcal{T} = \emptyset$  then
7        $\hat{\mathbf{x}}_k = m_k$ ;  $\mathbf{P}_k = I_{\text{length}(\text{key\_params})}$ ;
8       define new_treek :  $\hat{\mathbf{x}}_k, \mathbf{P}_k, \{\mathbf{F}_k, \mathbf{H}_k, \mathbf{Q}_k, \mathbf{R}_k\}$ ;
9       execute  $\mathcal{T}.\text{insert}(\text{new\_tree}_k)$ ; continue;
10    for all trees  $j$  in  $\mathcal{T}$  do
11       $\hat{\mathbf{x}}_j \leftarrow \mathbf{F}_j \hat{\mathbf{x}}_j + \mathbf{B}_j v_x$ ;
12       $\mathbf{i}_j = \text{norm\_residuals}(m_k - \mathbf{H}_j \hat{\mathbf{x}}_j)$ ;
13       $\mathbf{S}_j = \mathbf{H}_j \mathbf{P}_j \mathbf{H}_j^\top + \mathbf{R}_j$ ;
14       ${}^k \mathbf{p}_j = \exp(-0.1 \cdot d_{\mathcal{M}}(\mathbf{i}_j, \mathbf{S}_j^{-1})) \in [0, 1]$ ;
15       $\text{max}_j = \text{argmax}_j({}^k \mathbf{p}_j)$ ;
16       $\mathcal{H} = - \sum_j {}^k \hat{\mathbf{p}}_j \log_M({}^k \hat{\mathbf{p}}_j)$  where  $\sum_j {}^k \hat{\mathbf{p}}_j = 1$ ;
17      if  ${}^k \mathbf{p}_{\text{max}_j} > th_p$  and  $\mathcal{H} < th_{\mathcal{H}}$  then
18        execute  $\text{association\_pairs}.\text{insert}([\mathcal{T}_{\text{max}_j}, \text{new\_tree}_k])$ ;
19      else
20        execute  $\mathcal{T}.\text{insert}(\text{new\_tree}_k)$ ;
21    for all association-pairs  $\sim (\forall [\mathcal{T}_j, \text{new\_tree}_{k'}])$  do
22       $\text{Kalman}.\text{Update}(m_{k'}, \hat{\mathbf{x}}_{j'}, \mathbf{P}_{j'})$ ;
23       $\mathcal{T}_{j'} = \text{new\_tree}_{k'}$ ;
24    for all trees  $j$  in  $\mathcal{T}$  do
25      execute  $\text{min}_h = \text{argmin}_h(\ell^2\text{-norm}(\mathcal{GT}_h, \frac{g}{p} T^{-1} \cdot \mathcal{T}_j))$ ;
26      execute  $\mathcal{GT}_{\text{min}_h} = \mathcal{T}_j$ ;

```

---

First, the three channels in the RGN camera are split to compute the points' NDVI value  $v_{ndvi} \in \mathbb{R}$ . Binary thresholding is applied to the computed single-channel NDVI frame to segment the vegetation areas (selected values were set in the range  $q_{x,y} \in (-0.3, 1.0]$  following calibration testing in the field). At the same time, spatial filtering and Random Sample Consensus (RANSAC) fit is applied to the received point cloud  $\mathcal{P}$  along with an additional check of point projection on  $\Pi$  to remove points that belong on the ground surface. The remaining point cloud is downsampled and Euclidean Clustering [39] is applied to discretize and obtain appearing clusters in the  $\mathcal{P}$  space, all while preserving the closest  $v_{ndvi}$  values. The process takes place in the function *compute\_clusters* in line 3 of Algorithm 1.

For each cluster  $\mathcal{C}_k = \{p \in \mathcal{P}; v_{ndvi}\} \in \mathbb{R}^4$  we compute the centroid point  $p_{ct} \in \mathbb{R}^3$ , the total number of points  $num_{pt} \in \mathbb{N}$ , and the average NDVI value  $\bar{v}_{ndvi} \in \mathbb{R}$  as the main correspondence matching characteristics. These features will be used for the state estimation of each landmark  $\mathcal{T}_j$ .<sup>2</sup> We thus have  $\hat{\mathbf{x}}_k = [p_{ct}, num_{pt}, \bar{v}_{ndvi}] \in \mathbb{R}^{5 \times 1}$  and  $\mathbf{P}_k = I_5$ , where  $\hat{\mathbf{x}}_k$  is the state estimate and  $\mathbf{P}_k$  the estimate covariance. Lines 6 to 9, indicate the initialization of the first tree clusters in memory, as individual Kalman Filter estimates with the internal models  $\{\mathbf{F}_k, \mathbf{H}_k, \mathbf{Q}_k\} = I_5$  and  $\mathbf{R}_k = 0.1 \cdot I_5$ .

<sup>2</sup> It is possible to consider additional attributes, e.g., convex hull area and volume of the points included in each  $\mathcal{C}_k$ . We anticipate that doing so will have limited impact on the pipeline's runtime considering the small  $\mathcal{C}_k$  size while increasing system capabilities. A more formal treatise of this topic falls outside the scope of this paper, and is part of future work.

## B. Tree Landmarks Correspondence Matching

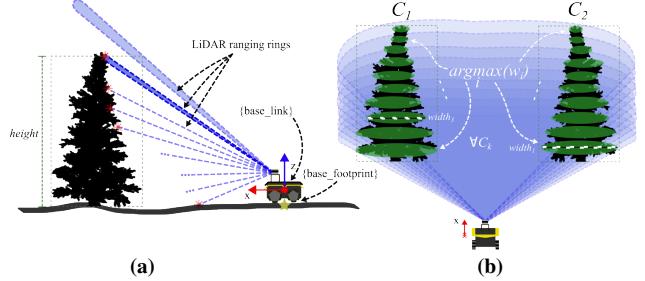
To achieve robust landmark prediction and association while surveying, we develop a probability calculation approach during the Kalman innovation step. The main task is to associate newly retrieved clusters  $\mathcal{C}_k$  with landmarks  $\mathcal{T}_j$  stored in memory. We first employ the control-input matrix  $\mathbf{B}_j = [\cos(\theta)\Delta t \ 0; \ \sin(\theta)\Delta t \ 0]$  in the prediction step (line 11), and then obtain the residual innovation  $\mathbf{i}_j$  for each tree  $\mathcal{T}_j$  in memory and normalize it to reflect the percentage change (line 12).

Then, we compute the Mahalanobis distance  $d_{\mathcal{M}}$  to determine the divergence of the innovation (error) residual from zero. The exponential distribution function is used to transform distance values into a probability  ${}^k p_j \in [0, 1]$  (line 14). Once the tree association probabilities for each cluster have been computed, we obtain the cluster with the closest match and index it as  $\max_j$ . To increase tree association robustness, we also compute the entropy  $\mathcal{H}$  across the normalized probabilities of all landmark pairs (line 16). This metric can quantify the level of uncertainty in landmark-to-cluster associations and help decide if a new landmark instance that matches the characteristics of a specific  $\mathcal{C}_k$  should be created and added in memory. When the cluster complies with both probability and entropy criteria, it is considered a match with  $\mathcal{T}_{\max_j}$  (lines 17-18). The cut-off values for these two criteria may need to be adjusted as the application domain varies; in this work, we determined them empirically via preliminary testing (i.e.  $th_p := 0.5, th_{\mathcal{H}} := 0.8$ ). Given the computed pairs  $(\mathcal{T}_j, \mathcal{C}_k)$ , we discard any matching duplicates (i.e. any potential instances of equal probability) and proceed with updating characteristics of interest (herein, tree width and height; see below), state estimates, and covariance matrices (lines 21-23 in Algorithm 1).

Since each cluster is expressed in the LiDAR frame  $\mathcal{P}$ , we use the inverse of the transformation matrix  ${}_{\mathcal{P}}^{\mathcal{G}} T$  to transform each  $\mathcal{T}_j$  to the global Cartesian frame created by the UTM datum. Thus, we find the closest  $\mathcal{G}\mathcal{T}_h$  candidate by computing the  $\ell^2$ -norm in  $x - y$  plane (line 25), and as  $(\mathcal{T}_j, \mathcal{G}\mathcal{T}_h)$  pairs are obtained the corresponding global georeferenced trees are updated (line 26). To handle the (likely) event of operating in GNSS-denied areas [40], [41], our developed system preserves and updates the local  $\mathcal{T}_j$  map during navigation, associating it with  $\mathcal{G}\mathcal{T}_h$  during periods with minimal GNSS interference.

## C. Tree Characteristics Estimation: Height and Width

As the robot moves in the field, the LiDAR sensor does not always fully cover the tree trunk and crown. As visualized in Figure 4a, some LiDAR rays may cast on the tree's crown and others may cast away from the tree. However, as the used LiDAR has a fixed vertical angular resolution of  $2^\circ$ , the sensor may raycast on the tree crown's topmost part as the robot moves and hence provide information about the tree's height. Thus, to estimate the height of the tree, we keep each tree candidate's  $\mathcal{T}_j$  top point's  $ring\_id$  detection with the prior detections and capture the tree height on the change



**Fig. 4:** (a) In height estimation, the robot senses a change on top  $ring\_id$  detection of the cluster  $\mathcal{C}_k$  and thus registers that height value with respect to  $\{base\_footprint\}$ . (b) In width estimation, the robot computes a tree's width value for each of the 10  $z$ -slices of the clusters and stores the maximum one.

of the top-ranging ring. The estimated height is refined each time we detect a tree candidate; the maximum value is kept.

For width estimation, we divide each cluster in a constant number of increasing  $z$ -level point cloud batches (in our experiments we use 10) in space  $\mathcal{P}$ , and we seek the farthest 3D point pairs. Given these pairs in different  $z$ -levels, we compute the  $\ell^2$ -norm and we find the maximum value, which we pinpoint as the tree's diameter (i.e. width). As the robot navigates, the width of a tree candidate is recomputed for each new point cloud  $\mathcal{P}$  received, and we keep the maximum value as the width of the tree candidate. Figure 4b illustrates a sketch of the width estimation procedure.

## V. EXPERIMENTAL TESTING AND VALIDATION

We evaluated our proposed algorithm extensively in both realistic simulated and real-world field survey experiments. In the simulation study we performed a survey on a  $2 \times 3$  citrus tree grid setup and assessed our algorithm's performance based on 1) using different tree placement densities, and 2) following different paths in the field. For the real-world field experiments, we deployed our robot in an orange tree grove and evaluated our algorithm's real-time performance and accuracy by following different survey paths in the orchard. To further enhance our evaluation results, we tested our approach on a larger citrus grove from the CitrusFarm [42] open dataset. Ground truth information was obtained directly from the 3D tree models in the simulation cases, and from a prior synthetic map created by the authors in the real-world cases. Comparisons against the most closely-relevant work of Jelavic *et al.* [27] (Robotic Precision Harvesting [RPH]) are also conducted.

### A. Evaluation in Simulated Tests

Simulated tests were conducted within the Gazebo simulator. The mobile robot model was modified to integrate the actual sensors' configuration and noise models. To increase the fidelity of the robot's traversability with respect to the real world, we created a custom 3D world, generated by aerial imagery data of a citrus field at UCR's Agricultural Experimental Station (AES;  $33^\circ 58' 3.2592'' N, 117^\circ 20' 7.0296'' W$ ) with the Agisoft Metashape software. Various realistic citrus (lemon and orange) tree instances were created by the Helios library [43] and integrated into the simulated world.

**TABLE I:** Field Experiment Results

Path Type	Tree Row	Simulated Fields						Real Fields							
		GT Width (m)	$\mu \pm \sigma$ (m)	MAPE Ours (%)	MAPE RPH (%)	GT Height (m)	$\mu \pm \sigma$ (m)	MAPE Ours (%)	MAPE RPH (%)	Path Type	Tree Row	MAPE Ours (%)	MAPE RPH (%)	MAPE Ours (%)	MAPE RPH (%)
Straight	front	2.71	$2.65 \pm 0.08$	<b>0.23</b>	2.55	2.62	$2.58 \pm 0.09$	8.17	<b>6.50</b>	Straight	front	<b>5.71</b>	10.19	<b>4.69</b>	12.24
	middle	2.67	$2.50 \pm 0.31$	<b>1.93</b>	2.78	2.74	$2.44 \pm 0.16$	<b>4.00</b>	10.27		middle	<b>0.24</b>	6.99	<b>9.74</b>	11.46
	back	2.71	$2.43 \pm 0.31$	<b>0.17</b>	4.07	2.68	$2.26 \pm 0.29$	<b>2.39</b>	8.74		back	<b>11.94</b>	13.43	15.20	<b>7.35</b>
S-type	front	2.71	$2.98 \pm 0.36$	<b>0.29</b>	1.05	2.62	$3.55 \pm 1.48$	<b>3.03</b>	6.83	S-type	front	<b>2.58</b>	9.78	<b>10.82</b>	12.86
	middle	2.67	$2.59 \pm 0.12$	12.14	<b>10.49</b>	2.74	$2.95 \pm 1.07$	14.71	<b>11.25</b>		middle	<b>2.60</b>	5.89	<b>6.34</b>	14.39
	back	2.71	$2.70 \pm 0.39$	<b>0.37</b>	2.19	2.68	$2.65 \pm 0.47$	<b>7.45</b>	10.32		back	<b>11.79</b>	13.43	12.25	<b>6.62</b>
N-type	front	2.71	$2.99 \pm 0.56$	<b>5.75</b>	12.04	2.62	$3.38 \pm 1.16$	<b>6.50</b>	7.30	N-type	front	<b>5.30</b>	15.63	<b>8.57</b>	10.00
	middle	2.67	$2.91 \pm 0.34$	<b>7.57</b>	23.45	2.74	$3.01 \pm 1.01$	<b>1.12</b>	15.33		middle	<b>3.56</b>	31.37	<b>9.02</b>	11.95
	back	2.71	$2.96 \pm 0.13$	<b>11.46</b>	13.35	2.68	$2.90 \pm 1.00$	<b>1.76</b>	4.09		back	<b>16.57</b>	35.67	<b>9.07</b>	19.12
4 × 4	front	2.71	$2.63 \pm 0.09$	<b>0.30</b>	4.51	2.58	$2.58 \pm 0.08$	<b>10.49</b>	10.97	CitrusFarm	row_1	4.77	<b>3.71</b>	<b>9.53</b>	13.28
	middle	2.67	$2.51 \pm 0.43$	<b>10.02</b>	10.59	2.62	$2.31 \pm 0.30$	11.85	<b>8.80</b>		row_2	<b>2.51</b>	4.93	<b>4.26</b>	11.91
	back	2.77	$2.82 \pm 0.27$	<b>5.21</b>	6.99	2.67	$2.42 \pm 0.36$	<b>5.16</b>	9.50						
6 × 6	front	2.71	$2.67 \pm 0.06$	<b>0.22</b>	2.64	2.58	$2.58 \pm 0.08$	<b>8.72</b>	11.87	CitrusFarm	row_3	<b>5.26</b>	7.34	9.95	<b>6.95</b>
	middle	2.67	$2.52 \pm 0.27$	<b>1.36</b>	2.25	2.62	$2.45 \pm 0.19$	8.59	<b>6.36</b>		row_4	<b>10.97</b>	23.88	<b>7.08</b>	12.15
	back	2.77	$2.44 \pm 0.28$	<b>3.64</b>	4.50	2.67	$2.19 \pm 0.48$	<b>4.30</b>	12.26		row_5	<b>12.64</b>	16.32	<b>3.02</b>	3.93

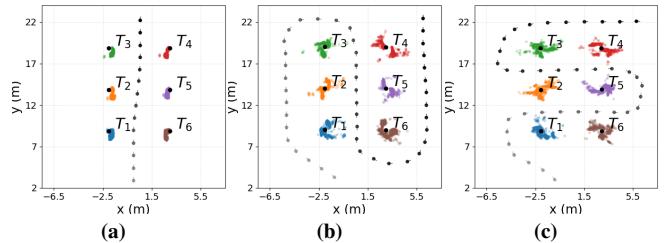
Figure 1b depicts the robot traversing the simulated field. In all simulated tests, the robot had constant linear ( $v_x = 0.5 \text{ m/sec}$ ) and angular ( $\omega_\theta = 1 \text{ rad/sec}$ ) velocities.

**Using different tree placement densities:** We generated three different world models by having the citrus trees 4, 5, and 6 m away from their neighboring trees. These grids are denoted  $\{4, 5, 6\} \text{ m width} \times \{4, 5, 6\} \text{ m length}$ , respectively. The instance of 5 m width  $\times$  5 m length is depicted in Figure 1b. These selections were made so as to be consistent with tree placement in commercial citrus orchards. From an algorithmic standpoint, they enable us to examine our method’s performance in both sparsely- and densely-planted orchards. Table I presents the evaluation results of five independent trials in each of the three tree-grid cases.

To perform the simulated surveys, we provided a goal waypoint in all grid cases for the robot to follow. The goal was set 23 m ahead from the origin along the longer side of the field (see Figure 5a). The robot does not stop at any point unless it has reached the goal.

Runtime comparisons between our method and RPH indicated a twofold time increase in tree detection in the latter case. RPH requires on average 0.53 sec for detecting the clusters in point clouds of 27-30K points, whereas our method has 0.22 sec on average detection runtime in the same point clouds and can provide up to 2x more tree detections in the same time frame. One of the main reasons causing delays in RPH seems to be the PCA step for locating clusters with z-axis divergence. Tree detection accuracy was similar to our *S*-case as both methods use Euclidean clustering, as a basis, to form the clusters  $\mathcal{C}_k$ .

Results from tree characteristics extraction (rows of Path Type: “Straight” in Table I) show that our algorithm enables the robot to perform both tree width and height estimation very accurately in all cases. The maximum Mean Absolute Percentage (MAP) errors observed were 12.1% and 14.7% for width and height estimations, respectively, over all tree configurations. Since the robot starts from point (0,0), it captures more observations (greater view) of the *front* row, and thus achieved 2%-5% of absolute percentage error less compared to those for the *middle* and *back* row. On the



**Fig. 5:** Panoramic views of the simulated tests for (a) straight-line, (b) *N*-type, and (c) *S*-type robot trajectories on the 5 m width  $\times$  5 m length tree grid setup. Along with the sampled and decreasingly transparent (with respect to time) robot poses, there is a colorized illustration of the detected tree candidates’  $p_{centroid}$  points, which are projected on the UTM  $x - y$  plane to demonstrate their distribution around the groundtruth  $\mathcal{GT}$  tree positions.

other hand, height estimation error decreased by 3% in more sparse fields, namely cases of  $5 \times 5$  and  $6 \times 6$  because of the broader field of view of the robot. The RPH method had up to 3% higher average MAP error in both width and height estimation, which can be linked to the slower runtime and thus a smaller number of tree observations. In our approach, comparably higher errors (on average) can be linked to the fact that the algorithm is using ranging information from the topmost part of a tree to estimate its height, but the former may be assessed multiple times during robot traversal thus including measurement variability.

**Following different paths in the field:** The second set of simulated tests examined the behavior of the algorithm as the followed path may vary. We used the  $5 \times 5$  tree placement grid, and tested our algorithm’s performance in more complex path types, namely *N*-type and *S*-type paths, in comparison to the aforementioned straight-line findings (first three rows of Table I). Sample *N*-type and *S*-type paths are shown in Figure 5.

From the reported results, we observe that our algorithm scores 4.3% and 8.3% of MAP error in width estimation and 8.4% and 3.1% of MAP error in height estimation for *S*-type and *N*-type trajectories, respectively. One of the main factors contributing to the difference in accuracy compared with the straight case, is that the robot performs closer maneuvers



**Fig. 6:** Snapshot from the *N*-type field experiment in the orchard. (a) Tree detection status along with detected  $C_k$  tree clusters. The image depicts information stats of each detected and matched cluster during that time (green rings are used for trees' width estimation, and red points are part of each cluster). (b) The corresponding RGN frame along with the projected LiDAR information and depicted tree information. (c) A closer third-person view of the Jackal in RViz closely matching the RGN frame. (Figure best viewed in color.)

inside the field. Similar performance is also notable in the RPH case, with an increase up to 8% and 5% of average MAP error in width and height estimation, respectively. In both systems, *N*-type surveys led to better estimations of the height level of the trees due to the longer straight lines within the field, while *S*-type surveys helped with a better estimation of the tree width due to the multiple traverses closer to the tree rows. In all, despite the different maneuvers in the field, our algorithm had an average of 5.3% in MAP error in both of these cases, demonstrating its robustness in width and height estimation.

#### B. Evaluation and Validation in Field Experiments

We evaluated our algorithm to validate its efficacy in field experiments. We performed a real survey in a citrus tree grove at UCR AES. We focused on a  $2 \times 3$  tree grid part of the selected field (to be consistent with the simulation test setup). This grid featured  $6.5 \text{ m}$  width  $\times 7 \text{ m}$  length tree placement, which is a setup not included in simulated tests. On the experiment day, the reported wind speed was at  $16 \text{ km/h}$ , with gusts up to  $40 \text{ km/h}$ .

Quantitative results for all cases are provided in Table I, whereas Figure 6 visualizes instances of the algorithm's behavior. Despite the windy conditions in the field, the robot scored a maximum of 16.6% and 15.2% MAP error in tree width and height estimation, respectively, over all trajectory types. Consistent with simulation findings, longer in length and more straight paths led to better results as observed by the straight case scoring on average 3% and 4% less MAP errors, compared with *S* and *N* cases, respectively. Similarly, on the RPH case, the height estimation results appear to be similar in all cases, averaging at 2.5% MAP more than ours. However, in width estimation on RPH case, the *N* case was the most noisy, as the windy conditions along with the restricted runtime of the algorithm, are not ideal for width estimation (as seen also in the simulated experiments), by scoring 10% of additional average MAP error. Overall, our algorithm gave less than 17% MAP error of geometric trait estimation, over all tested cases, even at challenging field conditions (high wind and diverse ambient light variations).

Additionally, in these cases, the robot collected the average NDVI measurements from each tree. This information is integral into our landmark association system but at the same time it can be “overloaded” to help monitor the identified trees' health (which is currently the use of NDVI field data).

For the cases of the *N* and *S*-typed trajectories (since they cover all the sides of the inspected trees), we observed resulting scores of 80% of an average relative distance, by noting +0.55, +0.61, and +0.62 of an average NDVI index, for the *front*, *middle*, and *back* row of the inspected trees.

To further support our evaluation step, we tested on a multi-modal dataset from CitrusFarm, that includes RGN and 3D LiDAR modalities collected by a mobile robot. We selected a survey path with *N*-type trajectory, in a  $5 \times 3$  citrus tree grove of  $29 \times 21 \text{ m}^2$  area size. With reference to Table I, our system performs well by having maximum MAP error at 12.7% at width estimation and 9.9% at height estimation, respectively, showcasing its scalability into larger fields. It is worth noting that, there was no reported wind in the used dataset of CitrusFarm as it was captured on a sunny day, which contributes to the observed improvements on average MAP errors. The RPH approach showcased an additional 3.4% of average MAP error in both estimation cases.

## VI. CONCLUSIONS

In this work, we introduced an algorithm for real-time and on-the-go tree detection and geometric traits estimation with wheeled mobile robots using onboard sensors. At its core, the algorithm uses diverse multi-modal sensory data (domain-specific RGN images and 3D point clouds) for tree detection, without relying on any specific tree parts. Key to this process is our proposed multivariate and entropy-based approach, based on an underlying Kalman Filter, which helps obtain local tree associations, calculate their physical and vegetation-based characteristics, and assign/correlate them with global georeferenced trees for inspection updates. Extensive evaluation in both simulation tests and physical field experiments in an orchard demonstrated the efficacy of our method and real-time robustness. Future research directions enabled by this work include extending to multi-robot systems for larger field coverage and employing multi-modal sensing for in-field and real-time analysis of tree geometry, health, and crop yield indices.

## REFERENCES

- [1] E. M. Perry, R. J. Dezzani, C. F. Seavert, and F. J. Pierce, “Spatial variation in tree characteristics and yield in a pear orchard,” *Precision Agriculture*, vol. 11, pp. 42–60, 2010.
- [2] L. McClymont, I. Goodwin, M. O’Connell, D. Whitfield, *et al.*, “Variation in within-block orchard canopy cover-implications for crop water requirement and irrigation management,” *Acta Horticulturae*, no. 889, pp. 233–239, 2011.

[3] A. Coupel-Ledru, B. Pallas, M. Delalande, F. Boudon, E. Carrié, S. Martinez, J.-L. Regnard, and E. Costes, “Multi-scale high-throughput phenotyping of apple architectural and functional traits in orchard reveals genotypic variability under contrasted watering regimes,” *Horticulture Research*, vol. 6, 2019.

[4] X. Liu, S. W. Chen, G. V. Nardari, C. Qu, F. C. Ojeda, C. J. Taylor, and V. Kumar, “Challenges and opportunities for autonomous micro-uavs in precision agriculture,” *IEEE Micro*, vol. 42, no. 1, pp. 61–68, 2022.

[5] J. Kim, S. Kim, C. Ju, and H. I. Son, “Unmanned aerial vehicles in agriculture: A review of perspective of platform, control, and applications,” *IEEE Access*, vol. 7, pp. 105100–105115, 2019.

[6] D. Chatziparaschis and P. Partsinevelos, “Real-time unmanned aerial vehicle surveying using spatial criteria: a simulated study,” *Journal of Applied Remote Sensing*, vol. 17, no. 1, pp. 014519–014519, 2023.

[7] P. Radoglou-Grammatikis, P. Sarigiannidis, T. Lagkas, and I. Moscholios, “A compilation of uav applications for precision agriculture,” *Computer Networks*, vol. 172, p. 107148, 2020.

[8] M. Bergerman, S. M. Maeta, J. Zhang, G. M. Freitas, B. Hamner, S. Singh, and G. Kantor, “Robot farmers: Autonomous orchard vehicles help tree fruit production,” *IEEE Robotics & Automation Magazine*, vol. 22, no. 1, pp. 54–63, 2015.

[9] S. S. H. Hajjaj and K. S. M. Sahari, “Review of agriculture robotics: Practicality and feasibility,” in *IEEE Int. Symposium on Robotics and Intelligent Sensors (IRIS)*, pp. 194–198, 2016.

[10] V. Adamchuk, W. Ji, R. V. Rossel, R. Gebbers, and N. Tremblay, *Proximal Soil and Plant Sensing*, ch. 9, pp. 119–140. John Wiley & Sons, Ltd, 2018.

[11] M. Campbell, K. Ye, E. Scudiero, and K. Karydis, “A portable agricultural robot for continuous apparent soil electrical conductivity measurements to improve irrigation practices,” in *IEEE Int. Conf. on Automation Science and Engineering (CASE)*, pp. 2228–2234, 2021.

[12] D. Chatziparaschis, E. Scudiero, and K. Karydis, “Robot-assisted soil apparent electrical conductivity measurements in orchards,” *arXiv preprint arXiv:2309.05128*, 2023.

[13] E. Scudiero, A. Singh, G. Mahajan, D. Chatziparaschis, K. Karydis, D. Houtz, and T. H. Skaggs, “On-the-go microwave radiometry for high-resolution soil moisture mapping in micro-irrigated orchards.” [Abstract], 2022. ASA, CSSA, SSSA International Annual Meeting.

[14] M. Campbell, A. Dechemi, and K. Karydis, “An integrated actuation-perception framework for robotic leaf retrieval: Detection, localization, and cutting,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 9210–9216, 2022.

[15] A. Dechemi, D. Chatziparaschis, J. Chen, M. Campbell, A. Shamshirgaran, C. Mucchiani, A. Roy-Chowdhury, S. Carpin, and K. Karydis, “Robotic assessment of a crop’s need for watering: Automating a time-consuming task to support sustainable agriculture,” *IEEE Robotics & Automation Magazine*, vol. 30, no. 4, pp. 52–67, 2023.

[16] X. Kan, T. C. Thayer, S. Carpin, and K. Karydis, “Task planning on stochastic aisle graphs for precision agriculture,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3287–3294, 2021.

[17] R. M. Palenichka and M. B. Zaremba, “Multiscale isotropic matched filtering for individual tree detection in lidar images,” *IEEE Trans. on Geoscience and Remote Sensing*, vol. 45, no. 12, pp. 3944–3956, 2007.

[18] J. Li, W. Zhang, W. Diao, Y. Feng, X. Sun, and K. Fu, “Csf-net: Color spectrum fusion network for semantic labeling of airborne laser scanning point cloud,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 15, pp. 339–352, 2022.

[19] J. Vauhkonen, L. T. Ene, S. Gupta, J. C. Heinzel, J. Holmgren, J. Pitkänen, S. Ø. Solberg, Y. Wang, H. Weinacker, K. M. Hauglin, V. T. Lien, P. Packalen, T. Gobakken, B. Koch, E. Næsset, T. Tokola, and M. Maltamo, “Comparative testing of single-tree detection algorithms under different types of forest,” *Forestry*, vol. 85, pp. 27–40, 2011.

[20] J. Secord and A. Zakhori, “Tree detection in urban regions using aerial lidar and image data,” *IEEE Geoscience and Remote Sensing Letters*, vol. 4, no. 2, pp. 196–200, 2007.

[21] Y. Wang, X. Yan, H. Bao, Y. Chen, L. Gong, M. Wei, and J. Li, “Detecting occluded and dense trees in urban terrestrial views with a high-quality tree detection dataset,” *IEEE Trans. on Geoscience and Remote Sensing*, vol. 60, pp. 1–12, 2022.

[22] H. Wan, Z. Fan, X. Yu, M. Kang, P. Wang, and X. Zeng, “A real-time branch detection and reconstruction mechanism for harvesting robot via convolutional neural network and image segmentation,” *Computers and Electronics in Agriculture*, vol. 192, p. 106609, 2022.

[23] Q. Xie, D. Li, Z. Yu, J. Zhou, and J. Wang, “Detecting trees in street images via deep learning with attention module,” *IEEE Trans. on Instrumentation and Measurement*, vol. 69, no. 8, pp. 5395–5406, 2020.

[24] A. Gupta, J. Byrne, D. Moloney, S. Watson, and H. Yin, “Tree annotations in lidar data using point densities and convolutional neural networks,” *IEEE Trans. on Geoscience and Remote Sensing*, vol. 58, no. 2, pp. 971–981, 2020.

[25] W. Zhang, P. Wan, T. Wang, S. Cai, Y. Chen, X. Jin, and G. Yan, “A novel approach for the detection of standing tree stems from plot-level terrestrial laser scanning data,” *Remote Sensing*, vol. 11, no. 2, 2019.

[26] P. Roy, W. Dong, and V. Isler, “Registering reconstructions of the two sides of fruit tree rows,” in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 1–9, 2018.

[27] E. Jelavić, D. Jud, P. Egli, and M. Hutter, “Robotic precision harvesting: Mapping, localization, planning and control for a legged tree harvester,” *Field Robotics*, vol. 2, pp. 1386–1431, 03 2022.

[28] N. Shalal, T. Low, C. McCarthy, and N. Hancock, “Orchard mapping and mobile robot localisation using on-board camera and laser scanner data fusion – part a: Tree detection,” *Computers and Electronics in Agriculture*, vol. 119, pp. 254–266, 2015.

[29] Z. Yu, B. Zhang, et al., “A camera/ultrasonic sensors based trunk localization system of semi-structured orchards,” in *IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics (AIM)*, pp. 168–174, 2021.

[30] A. Durand-Petiteville, E. Le Flecher, V. Cadenat, T. Sentenac, and S. Vougioukas, “Tree detection with low-cost three-dimensional sensors for autonomous navigation in orchards,” *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3876–3883, 2018.

[31] B. H. Wang, C. Diaz-Ruiz, J. Banfi, and M. Campbell, “Detecting and mapping trees in unstructured environments with a stereo camera and pseudo-lidar,” in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 14120–14126, 2021.

[32] X. Liu, S. W. Chen, C. Liu, S. S. Shivakumar, J. Das, C. J. Taylor, J. Underwood, and V. Kumar, “Monocular camera based fruit counting and mapping with semantic data association,” *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2296–2303, 2019.

[33] S. W. Chen, G. V. Nardari, E. S. Lee, C. Qu, X. Liu, R. A. F. Romero, and V. Kumar, “Sloam: Semantic lidar odometry and mapping for forest inventory,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 612–619, 2020.

[34] Y. Huang, Z. Ren, D. Li, and X. Liu, “Phenotypic techniques and applications in fruit trees: A review,” *Plant Methods*, vol. 16, 08 2020.

[35] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Trans. on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[36] D. Tsai, S. Worrall, M. Shan, A. Lohr, and E. Nebot, “Optimising the selection of samples for robust lidar camera calibration,” in *IEEE Int. Intelligent Transportation Systems Conf.*, pp. 2631–2638, 2021.

[37] T. Moore and D. Stouch, “A generalized extended kalman filter implementation for the robot operating system,” in *Int. Conf. on Intelligent Autonomous Systems (IAS-13)*, pp. 335–348, Springer, 2016.

[38] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, A. Y. Ng, et al., “Ros: an open-source robot operating system,” in *ICRA workshop on open source software*, vol. 3, p. 5, Kobe, Japan, 2009.

[39] R. B. Rusu, “Semantic 3d object maps for everyday manipulation in human living environments,” *KI-Künstliche Intelligenz*, vol. 24, pp. 345–348, 2010.

[40] F. Rovira-Más, I. Chatterjee, and V. Sáiz-Rubio, “The role of gnss in the navigation strategies of cost-effective agricultural robots,” *Computers and electronics in Agriculture*, vol. 112, pp. 172–183, 2015.

[41] M. Pini, G. Marucco, G. Falco, M. Nicola, and W. De Wilde, “Experimental testbed and methodology for the assessment of rtk gnss receivers used in precision agriculture,” *IEEE Access*, vol. 8, pp. 14690–14703, 2020.

[42] H. Teng, Y. Wang, X. Song, and K. Karydis, “Multimodal Dataset for Localization, Mapping and Crop Monitoring in Citrus Tree Farms,” in *Advances in Visual Computing: 18th International Symposium (ISVC)*, 2023.

[43] B. Bailey, “Helios: A scalable 3d plant and environmental biophysical modeling framework,” *Frontiers in Plant Science*, vol. 10, 10 2019.