

Article

Development of a Low-Cost System for 3D Orchard Mapping Integrating UGV and LiDAR

Harold F. Murcia ^{1,*}, Sebastian Tilaguy ¹ and Sofiane Ouazaa ²¹ Facultad de Ingeniería, Universidad de Ibagué, Ibagué 730003, Colombia; sebastian.tilaguy@unibague.edu.co² Centro de Investigación Nataima, Corporación Colombiana de Investigación Agropecuaria-AGROSAVIA, Km 9 vía Espinal-Chicoral, Espinal 733529, Colombia; souazaa@agrosavia.co

* Correspondence: harold.murcia@unibague.edu.co

Abstract: Growing evaluation in the early stages of crop development can be critical to eventual yield. Point clouds have been used for this purpose in tasks such as detection, characterization, phenotyping, and prediction on different crops with terrestrial mapping platforms based on laser scanning. 3D model generation requires the use of specialized measurement equipment, which limits access to this technology because of their complex and high cost, both hardware elements and data processing software. An unmanned 3D reconstruction mapping system of orchards or small crops has been developed to support the determination of morphological indices, allowing the individual calculation of the height and radius of the canopy of the trees to monitor plant growth. This paper presents the details on each development stage of a low-cost mapping system which integrates an Unmanned Ground Vehicle UGV and a 2D LiDAR to generate 3D point clouds. The sensing system for the data collection was developed from the design in mechanical, electronic, control, and software layers. The validation test was carried out on a citrus crop section by a comparison of distance and canopy height values obtained from our generated point cloud concerning the reference values obtained with a photogrammetry method. A 3D crop map was generated to provide a graphical view of the density of tree canopies in different sections which led to the determination of individual plant characteristics using a Python-assisted tool. Field evaluation results showed plant individual tree height and crown diameter with a root mean square error of around 30.8 and 45.7 cm between point cloud data and reference values.

Keywords: sensing; unmanned ground vehicle; 3D reconstruction; point clouds; crop monitoring



Citation: Murcia, H.F.; Tilaguy, S.; Ouazaa, S. Development of a Low-Cost System for 3D Orchard Mapping Integrating UGV and LiDAR. *Plants* **2021**, *10*, 2804. <https://doi.org/10.3390/plants10122804>

Academic Editor: Kioumars Ghamkhar

Received: 25 October 2021

Accepted: 14 December 2021

Published: 17 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Information on geometrical and structural characteristics of crops reveals important insight and provides decisive knowledge for management within an orchard. Numerous techniques and sensors have been used to extract useful information efficiently and effectively and thus facilitate early detection and action tasks for farmers. These technologies are expected to revolutionize agriculture, enabling timely decision-making, promising significant cost reduction, and increasing the crop yield [1]. Such decisions allow the effective application of farm inputs, supporting the pillars of precision agriculture: apply the right practice, at the right place, at the right time, and with the right quantity. Within this wide range of sensing possibilities, Light Detection and Ranging (LiDAR) technologies are one of the options of interest given their ability to generate 3D LiDAR point clouds such as elevations models for altitude and slope mapping. The processing of 3D LiDAR point clouds also enhances crop analysis by building semantic models with qualitative information at different dimensions, which can be abstracted with machine learning algorithms. The highest-end LiDARs have an accurate and fast response for three-dimensional measurements. Moreover, they can provide additional information with respect to the conventional, e.g., multiple echoes response, the associated returned energy, and the pulse width time of each return [2]. These top devices are more expensive, bigger, and heavier

for use with mobile vehicles, which makes access and application difficult. Mobile crop mapping consists of digital modeling of the crop with a sufficiently dense point cloud to describe the morphological characteristics of the terrain and the plants. This information can be used to assess the growth of the crop, estimate geometric tree characteristics, detect fruits, characterize a crop, estimate a crop yield, monitor the crop canopy, determine the crop biomass, leaf area index, and other high-throughput phenotyping parameters [3]. In this context, a vehicle navigates in the crop environment and accumulates the point clouds obtained by the LiDAR taking into account its location relative to an initial position. In this way, the successive accumulation of the transformed point clouds to a reference pose results in reconstruction in the form of a point cloud with at least three dimensions of information. Low-cost possibilities are part of the prospects and future development of such technologies according to [3]. The challenge with low-cost systems involves the deployment of simple devices for 3D reconstruction, e.g., the absence of an accurate device with good resolution for the measurement of the position of the mobile on the trajectory and the use of two-dimensional LiDAR sensors, which offer only one plane of information.

Most of the reported work uses a Terrestrial Laser Scanner (TLS) configuration, which can be considered as a fixed scanner with a large sensing range over a tripod with at least one rotational actuator. Recent research has focused on applications such as Stem–Leaf segmentation [4,5], wheat height detection [6], canopy characterization [7,8], corn detection [9], leaf area evaluation [10], leaf index estimation [11], biomass estimation [12], and plant structure phenotyping for an indoor environment with fixed structure implementations but using a transnational movement [13]. A second configuration defined in this paper—Mobile Terrestrial Laser Scanner (MTLS)—consists of a similar TLS configuration but includes a mobile platform with a location sensor such as a Global Position System (GPS) [14] or a computational process to estimate it. Moreover, there are other mobile platforms where a LiDAR device is mounted on-board of an unmanned vehicle (UGV or UAV) with a navigation system and computational algorithms in a continuous operation mode to locate and map the environment at the same time. This concept has been addressed in the field of robotics for many years under the concept of Simultaneous Localization and Mapping (SLAM) [15]. Consulted applications use the generated point clouds for fruit detection, yield prediction [14], canopy characterization [16], automatic panicles detection [17], high-throughput phenotyping [5,18], low cost phenotyping [16], and fast phenotyping [19], among others. Table 1 summarizes the applications, crops, and used sensors on recent works.

Table 1. MTLS.

Application	Crop	Sensing Device	Localization Device	Platform	Ref.
Fruit detection, yield prediction, and canopy characterization	Apple	3D Velodyne VLP-16	GPS1200+ Leica RTK-GNSS	MTLS	[14]
Canopy mapping.	Apple	3D Velodyne VLP-16	GPS 3D Robotics	Tractor	[16]
Panicle measurement.	Sorghum	3D FARO Focus X330	RTK GNSS	Tractor	[17]
Precision fruticulture and horticulture.	Vineyards	3D MS Kinect V2	RTK GNSS	UGV	[20]
Fast phenotyping.	Vineyards	2D SICK LMS-400	GPS/IMU - Advanced Navigation, RTK	UGV	[19]
High-throughput phenotyping	Maize	3D Velodyne HDL64-S3	GNSS receiver with two antennas	UGV	[5]
	Ryegrass	2D SICK LMS-400	Here+ V2 RTK GPS	UGV	[18]

The development of a high-resolution phenotyping platform integrated with post-processing technology offers the opportunity to assess complex traits more effectively. Compared to other platform types, UGVs reduce personnel time to cover large test plots

and provide more accurate measurement because they can operate at constant speed and avoid human error. They can also be less constrained by operating times or load capacity.

The quality of the point cloud obtained depends on all the system elements, however the LiDAR sensor and localization subsystem have a great impact on the 3D mapping. Popular LiDAR sensors used on these applications are 3D Velodyne VLP-16, 3D Velodyne HDL64, 2D Sick LMS-400, and Faro Focus X330, which provides approximately hundreds of thousand points/second and have low sensitivity to optical noise in outdoor conditions. Likewise, the use of positioning devices, including sensor fusion and Real-Time Kinematics (RTK) techniques, which provide real-time location corrections, implies a large increase in the quality of 3D models. Although the development and evaluation of sensor-based technologies for estimating plant indices is an area of active development and current scanning platforms encompass both airborne and ground-based systems, there are still costs associated with the chosen devices, data processing software, and data integration that may limit the widespread adoption of these systems.

The present work focuses on terrestrial applications. In this paper, we describe the development and assessment of a low-cost mobile scanner based on a 2D LiDAR sensor on-board a ground robot to navigate semi-controlled through an orchard. It is shown how low-cost devices can allow the mapping and generation of point clouds for the purpose of incorporating remote sensing techniques in crop management applications.

2. Materials and Methods

2.1. Terrestrial Mobile Robot

The used platform in this work is an own development to make 3D models of orchards and small crops with low-cost devices in comparison with commercial versions. The operating parameters were defined from problem requirements to select the needed components, then the prototype was designed in mechanical, electronic, and software layers. The design and implementation details of the platform called alphaRover are presented in Sections 2.1.1 and 2.1.2. This Unmanned Ground Vehicle (UGV) has a skid-steer configuration with six wheels and rocker-bogie suspension. The robot uses Pololu gearmotors with magnetic encoders of 6533 PPM and a Xsens Mti-30 sensor to estimate the robot position and trajectory, which is a full gyro-enhanced Attitude and Heading Reference System (AHRS) with roll/pitch (static | dynamic) 0.2° | 0.5° and yaw 1.0° . A NVIDIA Jetson TK1 board with Ubuntu 14.04 and Robotic Operating System ROS indigo manipulates the motors energy per each side (left and right) by using a Roboclaw $2 \times 15A$ unit. The robot was operated by using a wifi network which connected a remote pilot computer with a gamepad Logitech F710. The on-board scanner subsystem includes a servomotor Dynamixel AX-12A which supports a LiDAR with a tilt inclination of 45° with respect to the ground. Figure 1 illustrates the functional scheme of the system and its main elements. At the end of the route, the system records the data in a raw file, which is processed in subsequent stages to generate the point cloud and to determine the crop parameters that are derived from the point cloud.

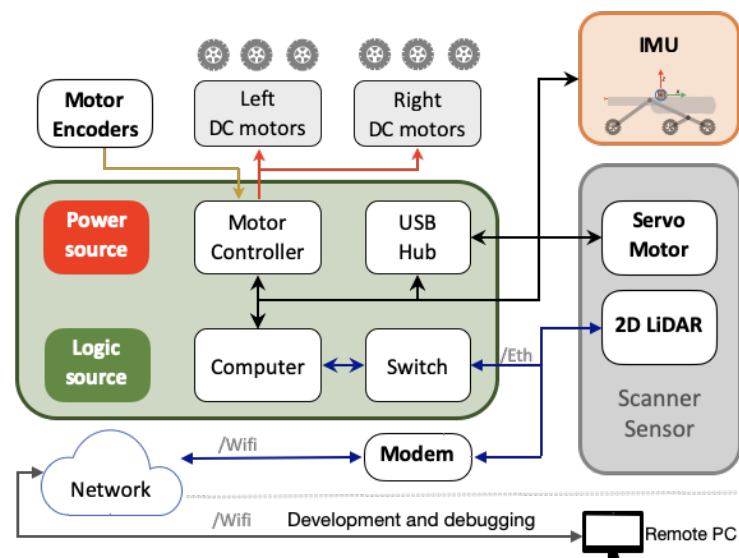


Figure 1. Functional diagram of the 3D mapping system.

2.1.1. Hardware Design

Mechanical design consisted in the selection of shapes, sizes, and materials for the construction of the robot structure, which was digitally designed by using SolidWorks software to evaluate the skid-steering architecture and a rocker-bogie suspension [21]. These configurations were defined aiming to minimize the effect of the irregular terrain profile in experimental conditions, reducing strong disturbances on the perception devices. The design of the six-wheel bogie suspension system has a shock absorber located in the rocker arm to protect the motors at the rear of the robot by reducing the force of impacts when overcoming obstacles. This improvement increases the reliability of the structure on rough or uneven terrain by allowing each of the wheels is in contact with the surface at all times. The mechanical design of rocker-bogie begins with considerations of the work ground. Therefore, we consider obstacles less than 5 cm high and 7 cm long. Using Equation (1), the minimum necessary radius of the wheel is 7.4 cm. Therefore, we select wheels with a radius of 7.5 cm.

$$r_{min} = \frac{h_{obs}^2 + w_{obs}^2}{2h_{obs}} \quad (1)$$

Taking into account the rocker-bogie scheme shown in Figure 2a, the suspension design begins by defining the vehicle size and the top angle of the fixed part α . Furthermore, to simplify and make the design systematic, we assume the angle β as $\frac{\pi-\alpha}{2}$ rad, so therefore \overleftrightarrow{AB} is determined as a hypotenuse. The \overleftrightarrow{BD} segment and h_1 distance are in function of the vehicle and wheels sizes. Thus, the segment \overleftrightarrow{AD} can be obtained from Equation (2). By assuming h_2 distance as the wheel diameter \overleftrightarrow{DE} and \overleftrightarrow{CE} can be determined as as hypotenuses in function of the ε and γ angles, respectively. Thus, \overleftrightarrow{AE} is calculated as the difference between \overleftrightarrow{AD} and \overleftrightarrow{ED} .

$$\overleftrightarrow{AD}^2 = \overleftrightarrow{BD}^2 + \overleftrightarrow{AB}^2 - 2(\overleftrightarrow{AB})(\overleftrightarrow{BD}) \cos \beta \quad (2)$$

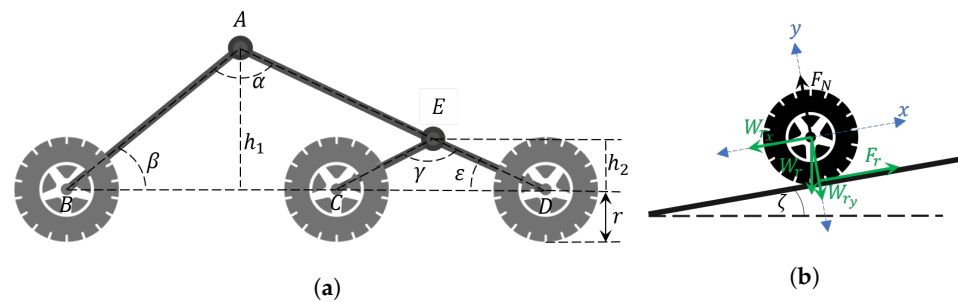


Figure 2. Design of the rocker-bogie suspension system: (a) complete rocker-bogie suspension; (b) wheel force diagram.

In motor selection, it is important to consider requirements when working in the worst case. We assume a surface with an inclination angle of ζ_s , an estimated mass of the vehicle m_r , an estimated load capacity m_L , a minimum linear speed v_{min} , a minimum linear acceleration a_{min} , the coefficient of friction μ , and a safety factor SF to oversize the design. The process began by analyzing the Figure 2b, where all forces are concentrated on one wheel to simplify the design. As each wheel has its own motor and the motors are the same (with the same reference), the obtained torque was defined as the sum of the torque contributed by each motor. Finally, the design parameters obtained are presented in Table 2.

Table 2. Related parameters to design criteria.

Variables	Values	Unities
ζ_s	$\frac{\pi}{6}$	rad
m_r	20	Kg
m_L	5	Kg
v_{min}	0.7	$\frac{m}{s}$
a_{min}	0.5	$\frac{m}{s^2}$
μ	0.7	-
SF	70	%
g	9.8	$\frac{m}{s^2}$

The coefficient of friction was obtained from a table from the materials literature. The total mass that the motors have to move was defined as $m_T = m_r + m_L$. Therefore, the components of the robot weight vector were calculated as

$$\begin{aligned} W_{ry} &= -m_T \cos(\zeta_s)g \\ W_{rx} &= -m_T \sin(\zeta_s)g \end{aligned} \tag{3}$$

The normal force has the same magnitude but the opposite direction that y component of weight, $F_N = -W_{ry}$. The friction force was defined as $F_r = -\mu F_N$, where the negative sign represents the opposition of movement. The total torque in the wheel was defined as

$$\sum \tau = I\dot{\omega} \tag{4}$$

where $\dot{\omega}$ is the angular acceleration and I is the moment of inertia, which, considering the wheel as a solid cylinder, then $I = \frac{1}{2}mr^2$. Assuming there is no external force on the wheel, and using the parameters from Table 2, the total torque for the alphaRover is

$$\tau = rF_r + \frac{1}{2}mra = 6.9 \text{ N.m} \tag{5}$$

Then, given the number of wheels, the torque for each motor τ_m must be

$$\tau_m = \frac{\tau}{6} SF = 1.955 \text{ N.m} \tag{6}$$

After a review of possibilities, the selected motor was the Pololu 37Dx73L gearmotor, an 8 watt motor with a 2.2 N.m torque at 12 VDC. Finally, the validation design was made by comparing the linear speed v_m of the alphaRover using the selected motor in relation with the desired speed from Table 2. In this way, the velocity in RPM must be converted to $\frac{m}{s}$:

$$v_m = RPM \frac{2\pi}{60} r = 0.7854 \frac{m}{s} \quad (7)$$

As $v_m > v_{min}$, the selected engine fulfills the design requirements.

On the one hand, the power consumption of the control elements and sensors was considered in order to define the necessary power supply for these electronic components and the appropriate control device. In addition, all design considerations include a desired minimum operating time, an autonomy time considered as $t_o \approx 1$ h. The design starts with an analysis of required energy, the consumption of control elements and sensors, which is presented in the Table 3.

Table 3. Power consumption of control elements and sensors.

Elements	Power [W]
Computer (Jetson TK1)	3
USB hub	
TP-Link AC750 wireless router	12
LiDAR	
UTM-30LX-EW	8.5
Switch	
TP-Link	4
Servo motor	
Dynamixel AX-12A	13.5

The elements are connected with DC/DC converters which have an average efficiency of 80%. Therefore, the total power consumption is approximately 51.25 W. Then, considering a LiPo battery of 6 cells, the average current consumption is calculated as

$$I_b = \frac{P_T}{N_C * V_C} \quad (8)$$

where P_T , N_C , and V_C are the total energy consumption, the number of cells, and the voltage per cell, respectively. Then, taking into account I_b current, the operating time t_o , and a safety factor to oversize the design, the minimum capacity of the battery to guarantee the desired operating time is

$$C_{Logic_{min}} = I_b t_o SF \approx 4 \text{ Ah} \quad (9)$$

For the power analysis in motors, the mass of the real vehicle was measured $m \approx 14$ Kg. This mass was used to determinate the average working torque with a constant vehicle movement speed:

$$\bar{\tau}_M = \frac{rmg}{6} \sin\left(\frac{\pi}{2}\right) \approx 1.1 \text{ N.m} \quad (10)$$

For the scanning application, it is important to move at a constant and slow speed to ensure good resolution. These design considerations are explained in Section 2.1.2. The angular speed is determined from the reference linear speed of the robot:

$$\omega_{sacan} = \frac{v_{sacan}}{r} \quad (11)$$

Thus, with the new torque $\bar{\tau}_M$ and the motor speed ω_{sacan} , the power consumption of each motor is

$$P_M = \tau \omega_{sacan} \quad (12)$$

For the nominal motor voltage a 3 cell battery was selected. To guarantee the same current, each motor on the same side was connected on the same channel. Therefore, the current per channel is

$$I_{ch} = 3 \frac{P_M}{V} \quad (13)$$

The driver current was selected considering the worst case and an efficiency of $\eta_{driver} = 0.8$:

$$I_{stall} = \frac{8 W}{V} \quad (14)$$

$$I_{stall_{max}} = 3I_{stall}SF \quad (15)$$

The power controller for motors must support at least 1.6 A per channel. The battery selection must take into account the consumption of the motors, the efficiency of the selected controller, and the desired operating time:

$$C_{Power_{min}} = 2 \frac{I_{ch}}{\eta_{driver}} t_o SF \approx 6.3 \text{ Ah} \quad (16)$$

Finally, the battery capacity selected was of 6 Ah.

2.1.2. Software Design

The software is composed by two layers: preprocessing and processing. The preprocessing layer is directly related to the fusion of the acquired data from the different perception elements involved in the generation of the three-dimensional point cloud, which represents the environment mapped by the robot. The processing layer determines the crop parameters from the point-cloud generated and user support. The robot uses an Ubuntu 14.04 operating system and Robotic Operating System ROS in its Indigo version. The system allows a manual or semi-controlled operation mode defined by the pilot from an external computer connected to the robot's network via WiFi. In the manual mode, the energy applied to the motors depends exclusively on the joystick's game-pad signal; conversely, in the semi-controlled mode, it has an intermediate wheel speed control layer that assists the action on the motors, facilitating the navigation even on slopes or irregular terrains. This last mode is explained in detail in the Section 2.1.3.

Figure 3 illustrates the computational software diagram of the system represented in ROS nodes (circles) and ROS topics (connection lines) which were developed or reconditioned from ROS programming community in C++ and Python languages. The first group of nodes corresponds with direct measurements and actions such as remote game-pad signals (Joy), Inertial Measurement Unit IMU (Xsens), Extended Kalman Filter (EKF) and motor control unit (Roboclaw). The second group of nodes contents the administration nodes which perform tasks such as remote connection with the pilot (Pilot), perception devices management, action elements management, data registration, and the synchronized data recording to generate the offline point cloud at the end of the robot's paths (Control). The last group contents the nodes from the scan unit, composed by a servomotor (Dynamixel), the 2D LiDAR sensor (URG), and the central subsystem (Scan) to control both elements.

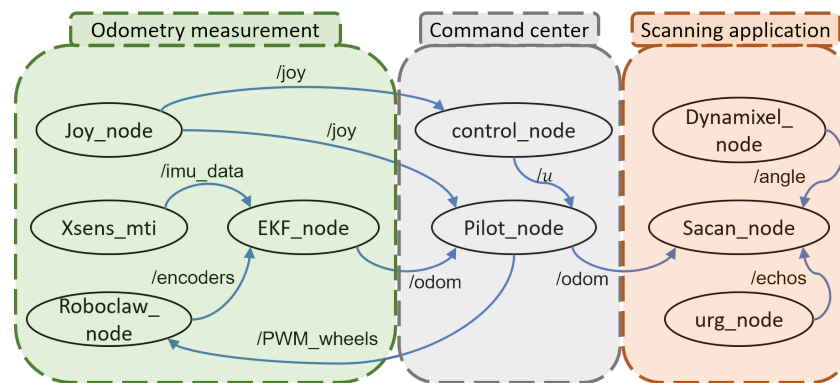


Figure 3. Computational software diagram in ROS nodes representation, from left to right the first, second, and third group of nodes. The nodes communicate with each other with messages, which are organized into specific categories called topics.

2.1.3. Embedded Cruise Control

The second manipulation mode presents some operation advantages for the pilot such as making movements at constant speed in front of perturbations in the wheels due to ground irregularities, upward or downward slopes, and minor obstacles; in this way, it is possible to make soft displacements avoiding accelerations that can affect the instruments and the robot. Additionally, it is possible to maintain a working rhythm independently of small discharges in the battery, as the control regulates the energy applied on the motors according to perturbations or voltage changes. To ensure a constant speed of the mobile during the scanning process, it was necessary to implement a speed control on each set of wheels, with a great robustness to cope with different terrains and obstacles, in addition to the effects of vibration and mechanical degradation. This design is based on the consideration of a second-order dynamic system:

$$G(s) = k \frac{\omega_0^2}{s^2 + 2\zeta\omega_0s + \omega_0^2} \quad (17)$$

A PID controller with robustness greater than 80% was designed using FRtool [22] with a mathematical model presented in the Equation equation and its estimated parameters. The settling time (t_s) is another important parameter for the controller design, which has a trade-off with robustness. As our application does not require large accelerations, the speed does not have to change rapidly, so its settling time could be as slow as $t_s \approx 2$ seconds. Finally, the maximum overshoot parameter (M_p), which is always desired to be as small as possible, was defined with a tolerance of less than 5%. Table 4 presents the estimated model parameters for the Equation (17), the design parameters for the control, and the obtained constants for the PID controller in its discrete implementation described in the following equation:

$$C(z) = K_p + K_i \frac{T_s}{z-1} + \frac{K_d}{T_s} (z-1) \quad (18)$$

Figure 4 illustrates the control scheme, which was implemented as difference equations into the “control node” rosnod as shown in Figure 3. The gestures of Channels 1 and 2 are generated in the gamepad to generate the acceleration and direction signal respectively. A gain K transforms the operated signals from the gamepad in the $\left[\frac{\text{rad}}{\text{seg}} \right]$ control set-points. These reference signals are saturated to guarantee the values of the angular speed limits. $C(s)$ and $G(s)$ are transfer function blocks that represent the PID controllers and the motors, respectively. The importance of speed control lies in giving the robot the ability to follow a constant speed reference and give it a certain degree of settlement in the presence of disturbances either from the ground or from its own construction characteristics.

Table 4. Design parameters for wheel speed controllers.

Parameter	Value	Units
k	0.1667	rad/s
ω_o	5.003	rad/s
ζ	0.9033	-
T_s	0.1	s
Robustness	>0.8	-
Settling time	<2	s
Overshoot	<0.5	-
K_p	6.677	rad/s
K_i	28.656	rad/s
K_d	0.0734	rad/s

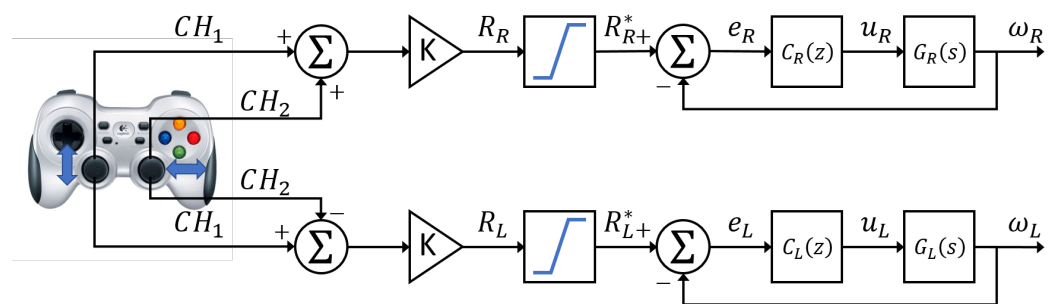


Figure 4. Control scheme for the speed control system. CH_1 and CH_2 represent the gamepad gestures; ω_R and ω_L represent the controlled wheel speed for the right side and the left side, respectively.

2.2. Pose Estimation

The skid-steer topology is based on Instantaneous Centers of Rotation ICR parameters. The ICR_v is the point where the vehicle’s rotation takes place without translation movement. The ICR_r and ICR_l are the traction points representation on the local frame for the right and left sets of wheel, respectively [23]. Figure 5 shows our local reference framework and the position of each ICR point, which are necessary to describe the movement of the vehicle.

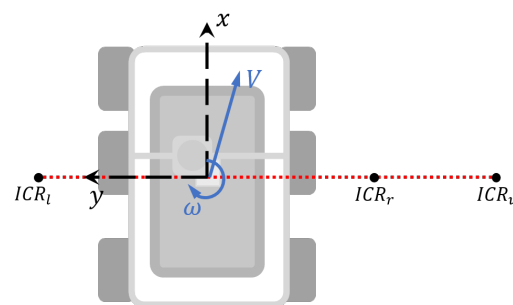


Figure 5. ICR on the vehicle plane.

The behavior of the position of a six-wheel skid-steer topology robot was modeled using equations presented in [24]. The kinematics relation between wheel velocities ($v_{r_i} = r\omega_i$) and the vehicle’s translational velocity v are expressed as

$$\begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = \frac{r}{y_{ICR_r} - y_{ICR_l}} \begin{bmatrix} -y_{ICR_l} \alpha_r & y_{ICR_r} \alpha_l \\ x_{ICR_v} \alpha_r & -x_{ICR_v} \alpha_l \\ \alpha_r & -\alpha_l \end{bmatrix} \begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} \tag{19}$$

where $\alpha_{r,l}$ are correction factors, ω is the angular speed around z axis, and magnitude of the vehicle’s translational speed can be described as $|v| = \sqrt{v_x^2 + v_y^2}$. To simplify the analysis,

in this document $\alpha_c = \alpha_l = \alpha_r = 1$ is considered; also is assumed $x_{ICR_v} = 0$, given the center of mass is in the center of vehicle. Finally, given each set of wheels is symmetric, is assumed that $y_{ICR_r} = -y_{ICR_l}$. In this way, the model showed in (19) is simplified as

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \frac{r\alpha_c}{2y_{ICR}} \begin{bmatrix} -y_{ICR} & y_{ICR} \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} \quad (20)$$

ICR coordinates are variable parameters which can be estimated from experimental measurements [25]. This estimation is possible thanks to the cruise control implementation, where (v_r, v_l) are regulated. Equation (21) shows the relationship $y_{ICR}(v_r, v_l, \phi_{mag})$.

$$y_{ICR} \approx \frac{\int v_r dt - \int v_l dt}{2\Delta\theta_{mag}} \quad (21)$$

where $\Delta\theta_{mag}$ is the change of robot orientation angle, which for this work is the measurement from a magnetometer in order to calculate the y_{ICR} . Some related works simplify the analysis in the form of a differential model; however, it is important to assess the need of extending the classical differential model. In this way, the related work in [25] introduces an expression to quantify the efficiency of the differential model with respect to the presented skid-steer model. This expression compares the relationship between L and the difference of ICR points of each side as follows:

$$\chi = \frac{L}{y_{ICR_r} - y_{ICR_l}}, \quad (0 \leq \chi \leq 1) \quad (22)$$

When the χ value is 1 it implies that the differential model is enough to describe the movement of vehicle. However, based on some experiments, our χ value reached a maximum of 0.03, which indicates that even the best case is far from being explained by a differential model. Once our kinematic model is defined, we proceed with the discrete implementation. Therefore, it was necessary to rewrite the model in term of differential equations (Δ), where $V = \frac{\Delta S_p}{T_s}$ and $\omega = \frac{\Delta\theta}{T_s}$,

$$\begin{bmatrix} \Delta S_p \\ \Delta\theta \end{bmatrix} = \frac{r\alpha_c T_s}{2y_{ICR}} \begin{bmatrix} -y_{ICR} & y_{ICR} \\ 1 & -1 \end{bmatrix} \begin{bmatrix} \omega_r \\ \omega_l \end{bmatrix} \quad (23)$$

where ΔS_p and $\Delta\theta$ is the magnitude of the change in position and orientation, respectively. T_s is the sampling period defined as 100 ms. Considering the new representation and defining that $\Delta S_{p_k} = S_{p_k} - S_{p_{k-1}}$ and $\Delta\theta_k = \theta_k - \theta_{k-1}$. Equation (19) can be rewritten as

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} \Delta S_{p_{k-1}} \cos\left(\theta_{k-1} + \frac{\Delta\theta_k}{2}\right) \\ \Delta S_{p_{k-1}} \sin\left(\theta_{k-1} + \frac{\Delta\theta_k}{2}\right) \\ \Delta\theta_k \end{bmatrix} \quad (24)$$

Equation (24) corresponds to a nonlinear model of estimated position, where x_k and y_k correspond to the position of the vehicle at x axis and y axis, respectively, and θ_k is the orientation angle in the inertial robot frame, which is obtained from an IMU. To correctly estimate the pose and orientation it was necessary to implement an Extended Kalman Filter (EKF); this algorithm is necessary to mix the information of sensors and minimize the measurement noise. Then, we initialize R_k , Q_k , and P , where Q_k is the covariance matrix of the process noise, R_k is the covariance matrix of the measurement noise, and P the covariance error matrix, which must initialize with very large values.

The algorithm was implemented in a rosnode programmed in Python language mentioned into the Section 2.1.2. A pose estimation algorithm for the vehicle was implemented in a real-time node and was tested in a sports field using a trajectory defined by traffic cones an monitored from an aerial video, as shown in Figure 6a. This trajectory was intended to verify the behavior of the EKF implemented in trajectories with turns in all directions.

The implemented EKF only estimated the *yaw* angle and the *x* and *y* coordinates. The *z* coordinate and tilt angles (*roll* and *pitch*) are assumed to be 0, in consideration of flat terrain thanks to the absorption of obstacles by the rocker-boogie topology and because of the simplified implementation of the mapping algorithm.

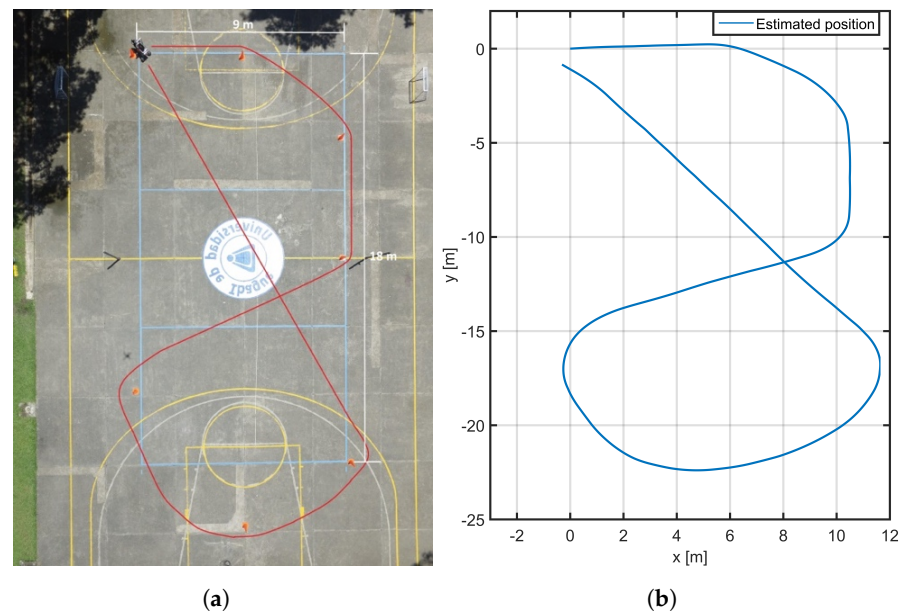


Figure 6. Result of trajectory estimation: (a) test setup and robot trajectory; (b) XY trajectory estimation.

Figure 6b shows the trajectory reconstruction obtained in a rosbag file where EKF data were captured from the test carried out on the sports field. This result not only presents the correct implementation of the kinematic model, but also minimizes the measurement noise from the encoders and the IMU.

2.3. Lidar Mapping

The preprocessing involved the reconstruction of the navigation environment and each of the trees from the LiDAR data with the scanning laser rangefinder 2D Hokuyo UTM-30LX-EW, which is a small device for outdoor robotic applications with a millimeter resolution with a 30 m range and 270° field of view. The angular resolution is 0.25° and the accuracy of the distance measurement is greater than 3 cm, depending on distance, the angle of incidence of the light beam and illumination of the environment. The simplest way to create the 3D model of the crop with a 2D laser sensor is by scanning the scene while the sensor remains static with a fixed inclination. The purpose of our mapping system is to acquire the LiDAR responses in a local coordinate frame and convert it to an absolute reference coordinate frame centered in P_0 point, which corresponds with the initial point of the robot's mission. According to a previous work [26], the final 3D point cloud in P_0 consists in a XYZ space in function of LiDAR horizontal angles H and radial distances R , as well as the servomotor vertical angle v , the XY position derived from the rover trajectory and some system parameters which represent the distances and possible angular deviations. With these measurements, the 3D position of an illuminated point P is given by a sequence of transformations from the 2D point $P_{i(R,H)}$ to P_1 , P_1 to P_2 , P_2 to P_3 , and P_3 to P_0 , respectively, as shown in Figure 7. In a general representation, these transformations are formulated from a basic transformation matrix T_p with $[4 \times 4]$ size:

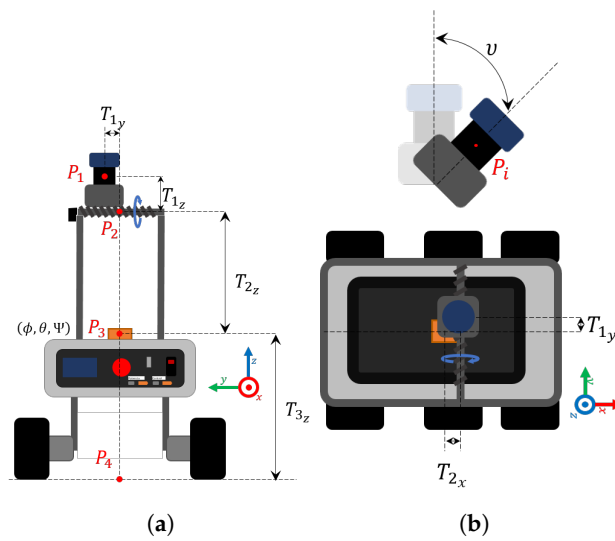


Figure 7. Schematic diagram of the mapping system and notations for different coordinated frame points to transform initial points from LiDAR to the reference frame P_0 : (a) rear view of vehicle, (b) top view of the vehicle.

$$T_p = R_z \cdot R_y \cdot R_x \cdot T \quad (25)$$

where T is a three-dimensional translation transformation with three parameters: t_x , t_y , and t_z . Rx_p represents a rotation transformation around X axis (pitch), Ry_p represents a rotation transformation around Y axis (roll), and Rz_p represents a rotation transformation around Z axis (yaw), for a total of six parameters per frame transformation: three angles (φ , ϑ , Ψ) and three distances (t_x , t_y , t_z). Each one of the above-mentioned matrices are described as follows:

$$T = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (26)$$

$$R_x = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c(\varphi) & -s(\varphi) & 0 \\ 0 & s(\varphi) & c(\varphi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (27)$$

$$R_y = \begin{bmatrix} c(\vartheta) & 0 & s(\vartheta) & 0 \\ 0 & 1 & 0 & 0 \\ -s(\vartheta) & 0 & c(\vartheta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (28)$$

$$R_z = \begin{bmatrix} c(\Psi) & -s(\Psi) & 0 & 0 \\ s(\Psi) & c(\Psi) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (29)$$

According to the above, the final XYZ reconstruction is composed of four transformations from P_i to P_0 , which are based on the Equation (25). Each coordinate of P_i is determined from the measured distance from the LiDAR to the illuminated object and the horizontal angle at which the beam was generated. Let us assume R as a diagonal matrix containing the range values in meters obtained from the LiDAR sensor for each horizontal angle beam.

$$R = \begin{bmatrix} R_1 & & \\ & \ddots & \\ & & R_N \end{bmatrix} \quad (30)$$

H comprises the horizontal angles of the 2D LiDAR for each range R measurement and is considered as a matrix of size $[N \times 1]$, where N is the number of beams or samples per each 2D LiDAR scan. In our case, the sensor is symmetrical and has a measuring range of 240° , therefore H_0 has a value of -120° and $\Delta_H = (240/N)^\circ$.

$$H = H_0 + \begin{bmatrix} 0 \\ \Delta_H \\ \vdots \\ N\Delta_H \end{bmatrix} \quad (31)$$

The corresponding equation for the matrix P_1 with size $[4 \times N]$, which is obtained from initial data is given as

$$P_1 = \begin{bmatrix} X_1 \\ Y_1 \\ \vec{0} \\ \vec{1} \end{bmatrix} = \begin{bmatrix} R \cdot \sin(H) \\ R \cdot \cos(H) \\ 0 \dots 0 \\ 1 \dots 1 \end{bmatrix} \quad (32)$$

The transformation matrix from P_1 to P_2 is given as

$$Tp_1 = Ry_{(v)} \cdot T_1 \quad (33)$$

where $Ry_{(v)}$ is the rotation matrix according to the motor angle v . The transformation matrix from the servo motor axis P_2 to the centered IMU frame P_3 is given as

$$Tp_2 = Rz_{(\Psi)} \cdot Ry_{(\theta)} \cdot Rx_{(\phi)} \cdot T_2 \quad (34)$$

In which the movement pitch is represented by θ , the movement roll is represented by ϕ , and the movement yaw is represented by Ψ . A third transformation consists of a vertical translation to the reference Z level at P_4 , by including the distance from IMU sensor to the floor.

$$Tp_3 = T_3 \quad (35)$$

The last transformation matrix contents the rover movement information. The matrices $T_{(Xk)}$ and $T_{(Yk)}$ include the estimated motion information of the robot on its X and Y axes in the absolute frame with respect to the P_0 point by using an EKF based on the work in [27].

$$Tp_4 = T_{(Yk)} \cdot T_{(Xk)} \cdot T_4 \quad (36)$$

With the following assumptions: (1) a null error on angle around Z-axis: $e_\Psi = \Psi_{real} - \Psi_{estimated} = 0$; (2) a minimal effect given angular deviation errors on physical implementation: $Rx_1 = Rz_1 = Rx_3 = Ry_3 = Rz_3 = Rx_4 = Ry_4 = Rz_4 \approx 0$; (3) a correct XY UGV pose estimation $e_{xy} = P_{real} - P_{estimated} = 0$; (4) P_0 is the position $P_{(x,y,z,\theta,\phi,\Psi)}$ of the UGV at the zero moment; and (5) a flat navigation environment without elevations on the ground, finally, Equation (37) summarizes the complete transformation:

$$P_0 = Tp_4 \cdot Tp_3 \cdot Tp_2 \cdot Tp_1 \cdot P_1 \quad (37)$$

In which P_1 corresponds with the first frame obtained from the initial LiDAR data, Tp_1 , Tp_2 , Tp_3 , and Tp_4 are the transformation matrices to transform the data to the reference frame centered in P_0 .

2.4. Field Data Collection

The system was evaluated on 40 citrus trees distributed along a groove of the crop, and for the reconstruction of the 3D model, from the total group, a region consisting of 24 plants was selected to validate the determination of morphological parameters in each of the plants. The orchard site is located at Agrosavia Nataima Research Center, Chicoral, Tolima, Colombia (Latitude: N 4° 21' 21.54", Longitude: W 54° 55' 40.59", Altitude: 409 m above sea level). The tree to tree spacing in the row was approximately 5.0 m with 8.0 m between furrows and approximately average height of 2.5 m. Figure 8 illustrates the experimental field and the used platform among two planting lines into the crop.



Figure 8. The mobile 3D LiDAR mapping system into the citrus crop.

Ground truth sampling was performed from a 3D reference point cloud using a DJI Mavic 2 pro aerial vehicle and Pix4D professional mapping software with its mapping and modeling tool mapping and the data capture application capture [28]. Once the point cloud was obtained, a point cloud processing tool with a user interface called CloudCompare was used to obtain measurements of average canopy height and radius for each tree in two of the crop rows [29].

During the field data collection, the UGV moved through a crop furrow in an on-the-fly mode, with continuous rosbag data recording process. The rosbag collects system variables in a synchronized manner: IMU data have a sampling rate of 100 Hz, the signals coming from the vehicle's encoders have a sampling rate of 55 Hz, and the LiDAR responses, as well as other robot variables such as current, voltage, temperature, and communication information. The data collection experiment of 115 m furrow took approximately three minutes. The sampling rate of the LiDAR is 40 Hz, each scan contains approximately 1440 light beams per return, i.e., approximately 7200 scans and 10 million points in the case of the first echoes. However, given the geometry of the system, not all of the 360° range is exploited, so the number of measurements per scan is reduced to less than half, and although the sensor reports three echoes in range and intensity, not all points contain information. Once the vehicle has finished its route, the rosbag recording process was stopped and the raw data were extracted for offline processing on a different computer than the robot.

2.5. Crop Parameters Estimation

The 3D reconstructed trees were then used to extract the tree to tree distances and furrow to furrow distance for a section of two furrows, as well as canopy heights and crown diameter through a developed algorithm in Python. The height of the tree canopy was defined as the height from the ground to the height where the highest leaves are found,

in other words the distance in z axis from a z_{ref} point to the highest point of the tree. The algorithm starts with the determination of the reference plane to determine canopy height z_{ref} by using least squares method and some picked points from the ground and the center point $Pt_{(i,j)(x,y)}$ of each tree (i) per furrow (j) from the point cloud top view, in order to calculate the tree to tree distances, the furrow to furrow average distance and the canopy heights for each tree. The z_{max} parameter is obtained by finding the maximum value in a cylindrical region $R_{(i,j)(t)}$ (see Equation (38)) with a user-defined radius $r_{(i,j)(t)}$ centered on point $P_{(i,j)(x,y)}$, which is the tree center defined from the graphical representation.

$$R_{(i,j)(t)}: \forall \{x, y, z\} | \left(\sqrt{\Delta x^2 + \Delta y^2} \leq r_{(i,j)(t)} \right) \wedge \left(z \geq z_{ref} \right) \quad (38)$$

where $\Delta x = x - x_{o(i,j)}$ and $\Delta y = y - y_{o(i,j)}$, and x_o and y_o are the center coordinates of the tree obtained from $Pt_{i,j}$. Then, once the z_{ref} and center points are defined for each tree, the method obtains the parameter as follows: let be $Pt_{(i,j)(x,y)}$ a $[N_t \times 2N_F]$ matrix with the xy coordinates of the centers of each tree, $Ht_{(j)}$ is a $[N_t \times N_F]$ matrix with the high of each tree, $T_{dst(j)}$ is a $[N_t - 1 \times N_F]$ matrix with the distances tree to tree per furrow and F_{dst} a $[N_F - 1 \times 1]$ matrix with the distances furrow to furrow:

$$Ht[i, j] = \max \left(R_{(i,j)(z)} \right) - z_{ref} \quad (39)$$

with $i = 0, 1, \dots, N_t$ and $j = 0, 1, \dots, N_F$.

$$T_{dst}[i, j] = \sqrt{(x_{o(i,j)} - x_{o(i-1,j)})^2 + (y_{o(i,j)} - y_{o(i-1,j)})^2} \quad (40)$$

with $i = 1, 2, \dots, N_t$ and $j = 0, 1, \dots, N_F$.

To find the furrow to furrow distance, it is not possible to guarantee the same number of trees per furrow or the perfect perpendicularity to each other; therefore, S_F is a first degree polynomial generated from minimizing the squared error of all the Pt per furrow defined as $S_F(x_a) = ax_a + b$, where x_a is the vector of x -axis values in the plot, and the proposed method consist to measure the average distance between lines.

$$\begin{aligned} F_{dst0}[i] &= \left[\sqrt{(S_F(x_{a(0,j)}) - S_F(x_{a(0,j-1)}))^2} \right]^T \\ F_{dstf}[i] &= \left[\sqrt{(S_F(x_{a(f,j)}) - S_F(x_{a(f,j-1)}))^2} \right]^T \\ F_{dst} &= \frac{F_{dst0} - F_{dstf}}{2} \end{aligned} \quad (41)$$

with $i = 0, 1, \dots, N_t$ and $j = 1, 2, \dots, N_F$.

The method details are shown in Algorithm 1. The algorithm takes as inputs the generated point cloud from the crop, and the user-defined parameters in the assisted interface.

Algorithm 1: Crop parameters determination

Result: $Ht_{(j)}, T_{dst(j)}, F_{dst}, \overline{Ht}_{(j)}, \overline{T_{dst(j)}}, \overline{F_{dst}}$
input matrix: $Pt_{(i,j)(x,y)}$
input constants: coefficients of z_{ref} plane: c_x, c_y, c_z
Manual selection of trees $R_{(i,j)(t)}$;
for $j := 0, N_F$ **do**
 for $i := 0, N_t$ iterations **do**
 Calculate canopy height $Ht_{(j)}$ based on Equation (39);
 Calculate distance between trees $T_{dst(j)}$ based on Equation (40);
 end
 Calculate $\overline{Ht}_{(j)}$ as the average of $Ht_{(j)}$;
 Calculate $\overline{T_{dst(j)}}$ as the average of $T_{dst(j)}$;
 Calculate F_{dst} based on Equation (41);
end
Calculate $\overline{F_{dst}}$ as the average of F_{dst} ;

3. Results**3.1. Generated Point Clouds**

To validate the design presented in the Section 2.1.3, the vehicle was lifted off the ground to avoid external disturbance and to see the closed-loop response of angular velocity and its ability to follow a reference imposed by gamepad in the same conditions of the experiment for the model parameter estimation. The results of this experiment are presented in Figure 9.

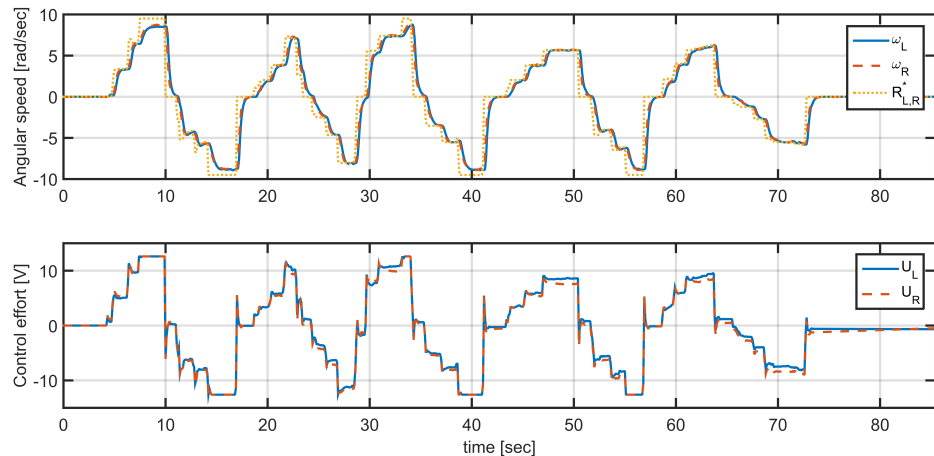


Figure 9. Speed setting of one of the left motors (blue) and one of the right motors (red) with respect to the angular speed reference (yellow). The horizontal axis is in samples, with a sample time $T_s = 0.1$ s.

To evaluate the scope of the controller a new experiment was carried out to compare vehicle responses with and without speed control. The vehicle was placed on a flat platform with an approximate inclination of 30 deg, where in open-loop condition the vehicle slides down as its weight exceeds the inertia of motors, as shown in Figure 10a. Figure 10c shows speed reference (yellow line) located at $0 \frac{rad}{s}$, the orange and blue curves correspond to the measured angular velocity of the motors right and left, respectively, which show an angular velocity different from zero in both cases. A new test was performed under the same circumstances, but this time with the speed control. Figure 10d shows the angular speed of the right and left wheels when implementing the wheel speed control for the same reference value. In this case, although there are some disturbances caused by noise and

measurement errors in the encoders, the operation of the control is evident, as the vehicle does not slide down because the angular speeds of the wheels are following their references.

After postprocessing, a point cloud was obtained with the 3D description of the route environment; an example of the generated point cloud of the citrus crop with a 20 m wide cut-out is presented in Figure 11. In addition to the geometric information, some additional information layers were acquired such as intensity, a digital representation for every point of the return strength of the laser pulse that generated the point which it is based, in part, on the reflectivity of the object struck by the laser pulse; other obtained information layers were the LiDAR horizontal angle, the LiDAR range, and the number of returns per pulse. Moreover, to better explain the scene, a fragment of an aerial photograph of the same section of the UGV's route is presented. The generated point cloud consists of 3,405,554 points, of which 3,284,177 points correspond to first echoes, 4478 points correspond to middle echoes and 116,899 points correspond to last echoes. The horizontal angle shows that major objects are illuminated with higher angles because of tilt inclination of the laser. In contrast, the central light beams have a shorter path and mainly illuminate low sections such as the floor. In the same way, with respect to the returns, the beams that illuminated low vegetation and tree areas had the highest number of returns in comparison with solid areas such as the ground.

3.2. Crop Parameters Estimation

The results of the algorithm implemented on Python described in Section 2.5 are presented in Table 5. The estimation of Crop parameters begins with the uploading of the point cloud generated by the robot to the software. Next, the user selects the z_{ref} parameter in the graphical interface to remove the ground of point cloud and plot an understandable top view (view blue data in Figure 12). Once the scene is displayed, the user uses the mouse cursor to click in the center of each tree where a red "x" symbol appears, The user defines a circle which describes the canopy radius $r_{(i,j)t}$ and forms a cylinder from the crop topmost to the z_{ref} point. When the click is released, a yellow "x" symbol appears and a red circle delimits the cylindrical region $R_{(i,j)t}$. When all trees in current furrow have been selected this step is repeated for a new furrow, by pressing the "space" key on the keyboard to start with a new furrow and continue marking each tree. Once all the trees in all the furrows have been marked the software is able to calculate the crop parameters. The results of this process are presented in the Figure 12.

Finally, the following crop parameters are registered and displayed:

- Canopy height. Measure the height of the canopy and color each point within the region $R_{(i,j)t}$ in green color.
- Crown diameter. From the maximum radius defined graphically by the user with the red rings, the system registers the diameter of the canopy in the corresponding tree.
- Distance between trees. Measure the tree to tree distance into the same furrow and draw a dashed red line between centers of trees $Pt_{(i,j)(x,y)}$
- Distance between furrows. Measure the furrow to furrow based on the estimation of a first-degree polynomial function in each furrow S_F , which is drawn with dotted black line. The achieved average distance between both straight lines in the evaluated range is illustrated by dashed green line.

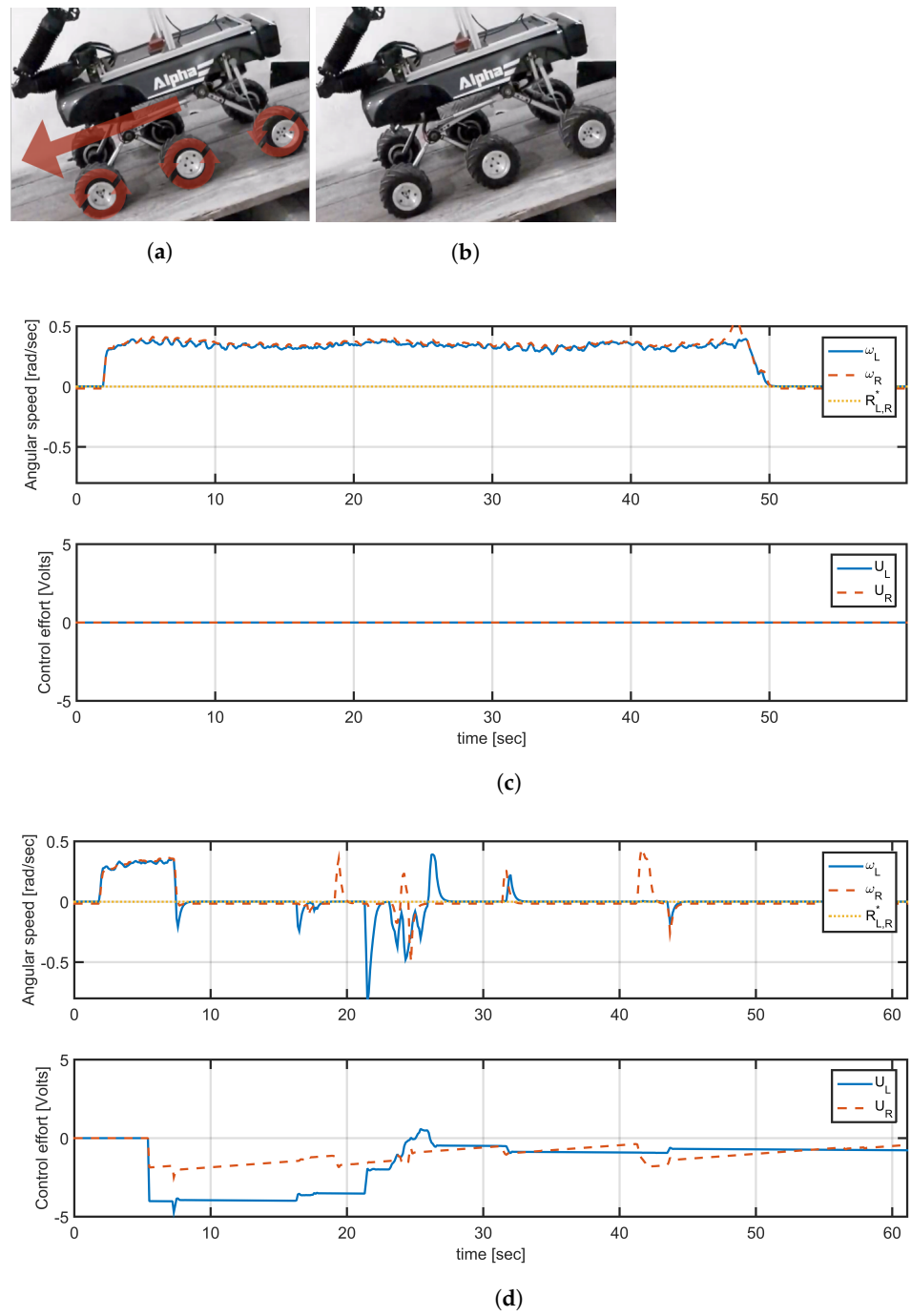


Figure 10. Speed control response in rad/s to a 30° incline: (a) the robot rolls on the ramp, (b) angular speed responses without control, (c) the robot holds its position without rolling effect, and (d) angular speed responses with. Left motors speed (blue), speed of one of the right motors speed (red), angular speed reference (yellow). The horizontal axis is in samples, with a sample time $T_s = 0.1$ s.

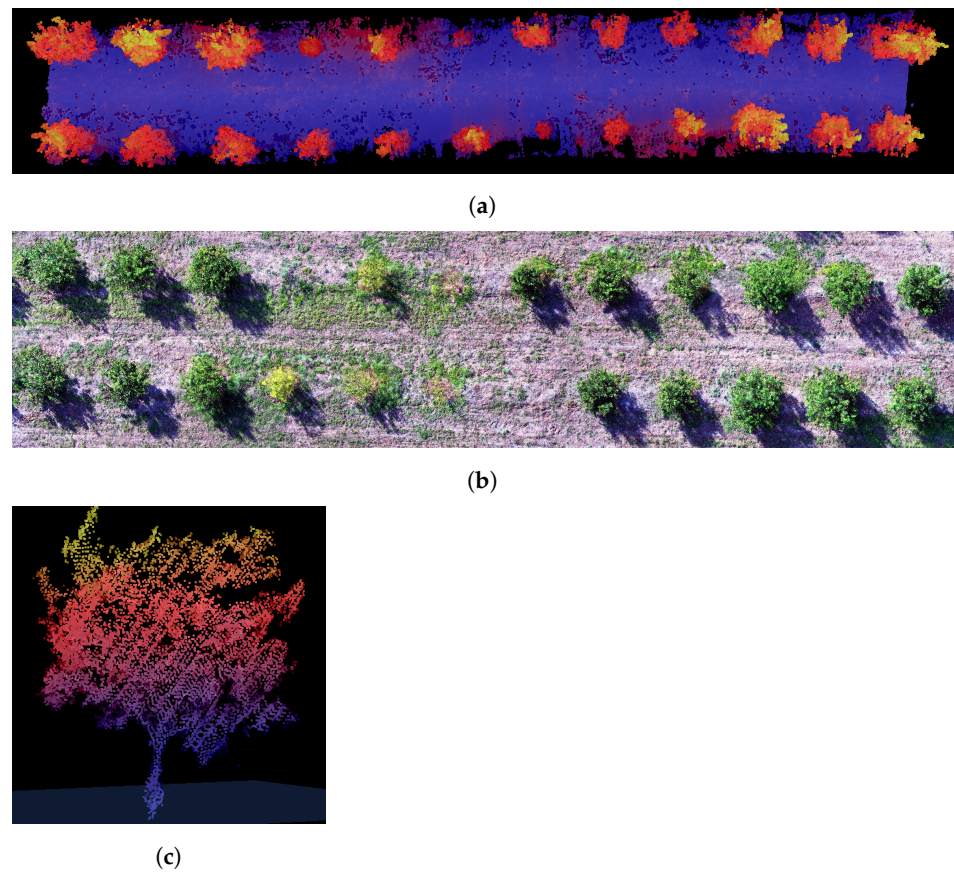


Figure 11. Obtained point cloud from the citrus crop segment: (a) Top view of the point cloud, (b) RGB image acquired with an aerial vehicle to compare, and (c) frontal view of a segmented tree.

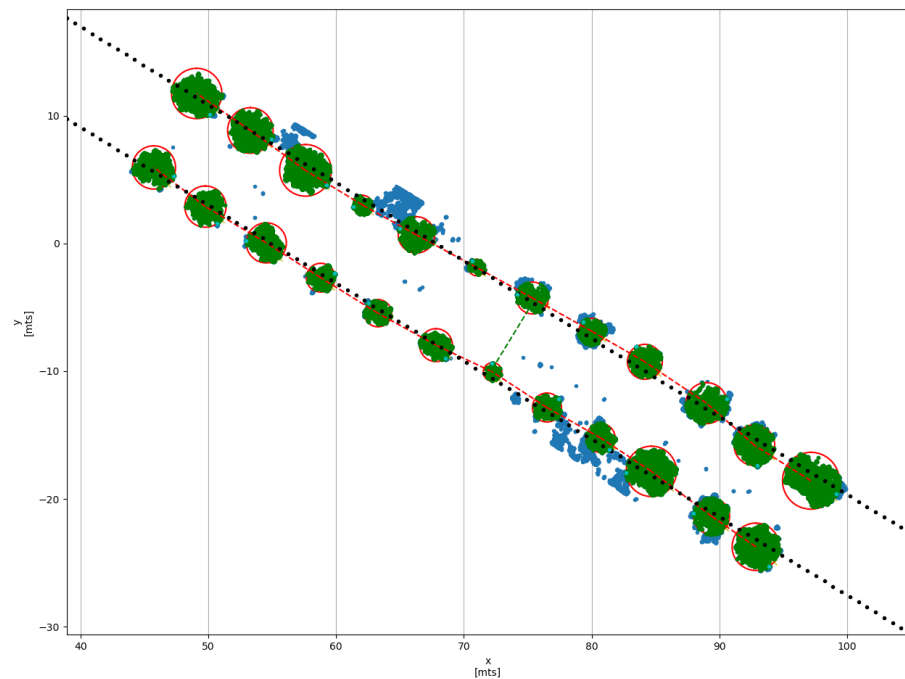


Figure 12. Assisted measurement of crop parameters from the top view of the point cloud with a ground removing in the interactive Python script.

The validation for the obtained results was carried out by a comparison of estimated data from our software with respect to the field measurements with a reference photogrammetry method. That information is presented in Table 5. The error of singular parameters as high or diameter of canopy was calculated using the *RMSE* (Root Mean Squared Error) and *R squared* R^2 as shown in the Equations (42) and (43). For common parameters such as distance between trees or distance between furrows, the comparison was carried out by the experimental error (*Ee*) calculation as the average of measurements with respect to the average of reference measurements, as shown in Equation (44). Additionally, for singular parameters the average value of error between real and estimated information by tree was calculated. The sample data used in this experiment consisted of a group of 24 trees of the citrus family, which are distributed in two rows.

$$RMSE(x, \hat{x}) = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2} \quad (42)$$

$$R^2(x, \hat{x}) = \frac{cov(x, \hat{x})}{var(x)var(\hat{x})} \quad (43)$$

where x_i and \hat{x}_i , are the field measurement take as reference and the estimated data by software, respectively.

$$Ee(\bar{x}_e, \bar{x}_r) = \frac{|\bar{x}_e - \bar{x}_r|}{\bar{x}_r} \quad (44)$$

where \bar{x}_e and \bar{x}_r are the average of estimated and real values, respectively.

Table 5. Experimental error of obtained parameters from point cloud with respect to reference measurements.

Parameter	Reference [m]	Obtained [m]	RMSE [m]	R^2	Ee[%]
h_c	—	—	0.308	0.732	9.282
d_c	—	—	0.457	0.637	17.294
\bar{D}_{t2t}	5.087	5.116	-	-	0.575
\bar{D}_{f2f}	8.294	7.759	-	-	6.443

4. Discussion

Geometric characterization of crops provides information about orchard variability and vigor, enabling the farmer to make faster and better decisions in tasks such as irrigation, fertilization, pruning, among others. Appropriate field management requires methods of measuring plant height with a good precision, accuracy, and resolution. With low-cost systems for 3D orchard and small crops mapping based on the combined use of LiDAR devices and terrestrial mobile platforms, we were able to generate 3D models and compute morphological parameters at individual plant level from a cultivated citrus crop. While most of the mapping systems consulted make use of freight vehicles such as tractors, the use of small mobile platforms facilitates mobility between small crop areas and reduces costs. The development of the mobile platform of this work was realized with about 1000 USD. The electronics related to laser mapping, navigation, processing, and powering have varied costs depending on characteristics such as precision, accuracy, or robustness. The LiDAR sensor for outdoor applications has about half of the value in the incidence on the total cost and the navigation unit ~30%, the most influential elements according to the authors of [30], who report the development of a low-cost system for 3d mapping in aerial platforms with a hypothetical selling price about 22,500 USD. In most of the cases of the related works, LiDAR involves investments in excess of 7000 USD and the navigation device in excess of 3000 USD, without taking into account the costs of processing software, computer unit, and battery. However, the proliferation of new embedded systems and low cost devices has facilitated access to these technologies, in our system the total related to navigation,

processing and laser registration was around 6500 USD. In addition, the software elements used in the development of this work for both reconstruction and processing are free to use.

This work is an alternative low-cost way of mapping orchards and small crops by generating a 3D model from simple 2D laser devices compared to multi-laser scanners. This reconstruction technology is easily employable in any type of crop where a terrestrial mobile platform can transit, with plants of any size and without the need to include additional infrastructure or intervene in the scanning environments as [31,32] given that it is a non-invasive system. The combined use of the laser measurement system with robotic mobile platforms avoids the need to use robust platforms such as tractors or scooters [14,16,17,33] which cannot always be moved between crops due to the size of the furrows and the type of crop. In addition, the use of the ROS robotics environment makes it compatible with different data computation tools and, as presented in the Section 3.2, it is possible to process the bag of data to generate a 3D model and process it. In our platform, we propose a user-friendly processing tool that reduces human intervention and at the same time determines the individual calculation of morphological parameters in each plant and furrow of the map in which the user is interested. Generally, the data acquisition with the laser scanner in the field worked very well. Nevertheless, problems occur from noise in the point clouds, due to wind, rain, insects, or small particles in the air, reflections on water, and other effects. These issues for terrestrial laser scanning applications in agriculture are also reported in [34]. A resulting limitation of our system due to the use of low-cost mapping and location systems is that the error in the position estimation increases with longer distances, and therefore the quality of the maps is affected in its accuracy, which is why in its current configuration it is only reliable in orchards or small crops. The presented system was able to estimate canopy and crop geometrical parameters at the same time with acceptable correlation coefficients of $RMSE = 30.8$ cm, $R^2 = 0.73$ and $Ee = 9.28\%$, and $RMSE = 45.7$ cm, $R^2 = 0.64$ and $Ee = 17.29\%$ for the canopy height and canopy diameter, respectively. Knowing those parameters on the tree structure, especially the canopy height and diameter, could be valuable for the planning and optimization of harvesting strategies [35]. For example, depending on the amount of fruits in the top parts of the trees and considering the extra costs involved to pick them (use of ladders or elevation platforms), the farmer could decide not harvest the highest areas. Geometric parameters estimated from raw data were mainly affected by the uncertainties in the laser scanner, the presence of weed plants between trees, the calibration of the setup, and the used pose estimation system. Different works have analyzed the error propagation in scanner laser systems. Similar results were found by the authors of [36] who used an UAV-LiDAR to estimate fresh biomass and crop height for three different crops (potato, sugar beet, and winter wheat). The height estimates worked well for sugar beet ($R^2 = 0.70$, $RMSE = 7.4$ cm) and wheat ($R^2 = 0.78$, $RMSE = 3.4$ cm). However, for potato plant height ($R^2 = 0.50$, $RMSE = 12$ cm), it proved to be less reliable due to the complex canopy structure and the ridges on which potatoes are grown. Small differences on results were found in [8] with an R^2 of 0.87%, 0.88%, and 0.77% for height, stem diameter, and canopy volume, respectively, the authors of which used a 2D light detection and range (LiDAR) laser scanner system mounted on a tractor and tested on a box with known dimensions for an apple orchard to obtain the 3D structural parameters of the trees, using a real-time kinematic global navigation satellite system (RTK-GNSS). The authors of [14] developed a system which was able to detect and localize more than 80% of the visible fruits, predict the yield with a root mean square error lower than 6% and characterize canopy height, width, cross-sectional area, and leaf area. The system used in their study was based on the Mobile Terrestrial Laser Scanner MTLs described in [37] and replacing the 2D LiDAR sensor with the 3D Velodyne VLP-16. Obtaining an accurate characterization of trees by non-destructive methods at different growth stages provides valuable information that can be used for enhancing precision in orchard management [38,39]. Among the geometrical parameters of the plants, crown diameter and crown height are of particular interest, as both combine the width, height, geometrical shape, and structure of the trees [40]. Such

parameters are commonly used for farmers, e.g., when establishing a biomass model for plants [41], in herbicide management, and in pruning directives [42]. However, in this study, the prototype for the data collection was developed using low-cost components, making it accessible. Consequently, laser scanning methods are a promising tool for precision agriculture. The vehicle uses a rocker-bogie suspension and a speed control to facilitate its manipulation and operation by the user avoiding the effects of sudden movements. This system was able to detect geometric variables in citrus trees that can be used in agricultural applications to measure tree growth for individual tree orchard management, while considering mechanical pruning, irrigation, and spraying.

5. Conclusions

A system for orchard mapping integrating an UGV and a LiDAR 2D was developed, allowing generating 3D reconstructions of a crop segment and the subsequent determination of spatial characteristics under field conditions such as distance between furrows, distances between trees, tree heights, canopy heights, and canopy diameter, by using an assisted algorithm. The prototype for the data collection was developed using low cost components, making it accessible. The vehicle uses a rocker-bogie suspension and a speed control to facilitate its manipulation and operation by the user avoiding the effects of sudden movements. The reliability of the system was achieved by measuring the variables simultaneously from a generated point cloud with PIX4D and a quadrotor comparing the results with error indicators. For the particular measurements the *RMSE* indicator was determined, obtaining error values of 30.8 cm and 45.7 cm for the height of the trees and the diameter of the crowns, respectively. Additionally the experimental *Ee* error was determined for the other variables obtaining 0.57% and 6.44% for the tree spacing and furrow to furrow distance, respectively. Future work includes the use of visual and LiDAR SLAM methods to improve pose estimation and consequently the quality of generated point clouds. Furthermore, the development of machine learning techniques to simplify and speed up the calculation of morphological indices in large-scale crops, that can offer a timely use in the agricultural processes of fruit orchards.

Author Contributions: H.F.M. conceived the idea of the paper. S.T. performed the experimental work, analyzed the data, and wrote the first draft of the manuscript. S.O. selected the crops and advised on plant indices. All authors contributed to the writing of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was conducted with financial support from Universidad de Ibagué (19-489-INT).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Our code implementation is available online at https://github.com/HaroldMurcia/miniRover_LiDAR_citrush_crop.git, accessed on 25 November 2021.

Acknowledgments: We would like to thank the Universidad de Ibagué for the technical support in this work and the Colombian agricultural research corporation Agrosavia for allowing us to carry out the data collection experiments at the Nataima Research Centre.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Khanal, S.; KC, K.; Fulton, J.P.; Shearer, S.; Ozkan, E. Remote Sensing in Agriculture—Accomplishments, Limitations, and Opportunities. *Remote Sens.* **2020**, *12*, 3783. [CrossRef]
2. Mallet, C.; Bretar, F. Full-waveform topographic lidar: State-of-the-art. *ISPRS J. Photogramm. Remote Sens.* **2009**, *64*, 1–16. [CrossRef]
3. Jin, S.; Sun, X.; Wu, F.; Su, Y.; Li, Y.; Song, S.; Xu, K.; Ma, Q.; Baret, F.; Jiang, D.; et al. Lidar sheds new light on plant phenomics for plant breeding and management: Recent advances and future prospects. *ISPRS J. Photogramm. Remote Sens.* **2021**, *171*, 202–223. [CrossRef]

4. Jin, S.; Su, Y.; Wu, F.; Pang, S.; Gao, S.; Hu, T.; Liu, J.; Guo, Q. Stem-Leaf Segmentation and Phenotypic Trait Extraction of Individual Maize Using Terrestrial LiDAR Data. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 1336–1346. [[CrossRef](#)]
5. Qiu, Q.; Sun, N.; Bai, H.; Wang, N.; Fan, Z.; Wang, Y.; Meng, Z.; Li, B.; Cong, Y. Field-based high-throughput phenotyping for maize plant using 3d LIDAR point cloud generated with a “phenomobile”. *Front. Plant Sci.* **2019**, *10*, 554. [[CrossRef](#)]
6. Guo, T.; Fang, Y.; Cheng, T.; Tian, Y.; Zhu, Y.; Chen, Q.; Qiu, X.; Yao, X. Detection of wheat height using optimized multi-scan mode of LiDAR during the entire growth stages. *Comput. Electron. Agric.* **2019**, *165*, 104959. [[CrossRef](#)]
7. Yuan, H.; Bennett, R.S.; Wang, N.; Chamberlin, K.D. Development of a peanut canopy measurement system using a ground-based lidar sensor. *Front. Plant Sci.* **2019**, *10*, 203. [[CrossRef](#)]
8. Tsoulias, N.; Paraforos, D.S.; Fountas, S.; Zude-Sasse, M. Estimating canopy parameters based on the stem position in apple trees using a 2D lidar. *Agronomy* **2019**, *9*, 740. [[CrossRef](#)]
9. Xu, L.; Xu, T.; Li, X. Corn Seedling Monitoring Using 3-D Point Cloud Data from Terrestrial Laser Scanning and Registered Camera Data. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 137–141. [[CrossRef](#)]
10. Berk, P.; Stajanko, D.; Belsak, A.; Hocevar, M. Digital evaluation of leaf area of an individual tree canopy in the apple orchard using the LIDAR measurement system. *Comput. Electron. Agric.* **2020**, *169*, 105158. [[CrossRef](#)]
11. Wu, D.; Phinn, S.; Johansen, K.; Robson, A.; Muir, J.; Searle, C. Estimating changes in leaf area, leaf area density, and vertical leaf area profile for Mango, Avocado, and Macadamia tree crowns using Terrestrial Laser Scanning. *Remote Sens.* **2018**, *10*, 1750. [[CrossRef](#)]
12. Schulze-Brüninghoff, D.; Hensgen, F.; Wachendorf, M.; Astor, T. Methods for LiDAR-based estimation of extensive grassland biomass. *Comput. Electron. Agric.* **2019**, *156*, 693–699. [[CrossRef](#)]
13. Wang, H.; Lin, Y.; Wang, Z.; Yao, Y.; Zhang, Y.; Wu, L. Validation of a low-cost 2D laser scanner in development of a more-affordable mobile terrestrial proximal sensing system for 3D plant structure phenotyping in indoor environment. *Comput. Electron. Agric.* **2017**, *140*, 180–189. [[CrossRef](#)]
14. Gené-Mola, J.; Gregorio, E.; Auat Cheein, F.; Guevara, J.; Llorens, J.; Sanz-Cortiella, R.; Escolà, A.; Rosell-Polo, J.R. Fruit detection, yield prediction and canopy geometric characterization using LiDAR with forced air flow. *Comput. Electron. Agric.* **2020**, *168*, 105121. [[CrossRef](#)]
15. Lemaire, T.; Lacroix, S.; Solà, J. A practical 3D bearing-only SLAM algorithm. In Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, AB, Canada, 2–6 August 2005. [[CrossRef](#)]
16. Chakraborty, M.; Khot, L.R.; Sankaran, S.; Jacoby, P.W. Evaluation of mobile 3D light detection and ranging based canopy mapping system for tree fruit crops. *Comput. Electron. Agric.* **2019**, *158*, 284–293. [[CrossRef](#)]
17. Malambo, L.; Popescu, S.C.; Horne, D.W.; Pugh, N.A.; Rooney, W.L. Automated detection and measurement of individual sorghum panicles using density-based clustering of terrestrial lidar data. *ISPRS J. Photogramm. Remote Sens.* **2019**, *149*, 1–13. [[CrossRef](#)]
18. Nguyen, P.; Badenhorst, P.E.; Shi, F.; Spangenberg, G.C.; Smith, K.F.; Daetwyler, H.D. Design of an Unmanned Ground Vehicle and LiDAR Pipeline for the High-Throughput Phenotyping of Biomass in Perennial Ryegrass. *Remote Sens.* **2020**, *13*, 20. [[CrossRef](#)]
19. Siebers, M.H.; Edwards, E.J.; Jimenez-Berni, J.A.; Thomas, M.R.; Salim, M.; Walker, R.R. Fast phenomics in vineyards: Development of grover, the grapevine rover, and LiDAR for assessing grapevine traits in the field. *Sensors* **2018**, *18*, 2924. [[CrossRef](#)] [[PubMed](#)]
20. Rosell-Polo, J.R.; Gregorio, E.; Gene, J.; Llorens, J.; Torrent, X.; Arno, J.; Escola, A. Kinect v2 sensor-based mobile terrestrial laser scanner for agricultural outdoor applications. *IEEE/ASME Trans. Mechatronics* **2017**, *22*, 2420–2427. [[CrossRef](#)]
21. ©2002–2021 Dassault Systèmes. SOLIDWORKS. [Computer Software]. Available online: <https://www.solidworks.com/> (accessed on 1 December 2021).
22. De Keyser, R.; Ionescu, C. FRtool: A frequency response tool for CACSD in Matlab. In Proceedings of the 2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, Munich, Germany, 4–6 October 2006; pp. 2275–2280.
23. Huskić, G.; Buck, S.; Herrb, M.; Lacroix, S.; Zell, A. High-speed path following control of skid-steered vehicles. *Int. J. Robot. Res.* **2019**, *38*, 1124–1148. [[CrossRef](#)]
24. Martínez, J.L.; Mandow, A.; Morales, J.; Pedraza, S.; García-Cerezo, A. Approximating kinematics for tracked mobile robots. *Int. J. Robot. Res.* **2005**, *24*, 867–878. [[CrossRef](#)]
25. Mandow, A.; Martínez, J.L.; Morales, J.; Blanco, J.L.; García-Cerezo, A.; González, J. Experimental kinematics for wheeled skid-steer mobile robots. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, San Diego, CA, USA, 29 October–2 November 2007; pp. 1222–1227. [[CrossRef](#)]
26. Murcia, H.F.; Monroy, M.F.; Mora, L.F. 3D scene reconstruction based on a 2D moving LiDAR. In *Communications in Computer and Information Science*; Springer: Cham, Switzerland, 2018; Volume 942. [[CrossRef](#)]
27. Barrero, O.; Tilaguy, S.; Nova, Y.M. Outdoors Trajectory Tracking Control for a Four Wheel Skid-Steering Vehicle*. In Proceedings of the 2018 IEEE 2nd Colombian Conference on Robotics and Automation, Barranquilla, Colombia, 1–3 November 2018. [[CrossRef](#)]
28. Professional Photogrammetry and Drone Mapping Software | Pix4D. [Computer Software]. Available online: <https://www.pix4d.com/> (accessed on 1 December 2021).
29. CloudCompare (Version 2.11.1) [GPL Software]. 2021. Available online: <http://www.cloudcompare.org/> (accessed on 1 December 2021).

30. Torresan, C.; Berton, A.; Carotenuto, F.; Chiavetta, U.; Miglietta, F.; Zaldei, A.; Gioli, B. Development and Performance Assessment of a Low-Cost UAV Laser Scanner System (LasUAV). *Remote Sens.* **2018**, *10*, 1094. [[CrossRef](#)]
31. Bai, G.; Ge, Y.; Scoby, D.; Leavitt, B.; Stoerger, V.; Kirchgessner, N.; Irmak, S.; Graef, G.; Schnable, J.; Awada, T. NU-Spidercam: A large-scale, cable-driven, integrated sensing and robotic system for advanced phenotyping, remote sensing, and agronomic research. *Comput. Electron. Agric.* **2019**, *160*, 71–81.10.1016/j.compag.2019.03.009 [[CrossRef](#)]
32. N, V.; K, S.; P, S.T.; MJ, H. Field Scanalyzer: An automated robotic field phenotyping platform for detailed crop monitoring. *Funct. Plant Biol. FPB* **2016**, *44*, 143–153. [[CrossRef](#)]
33. Walter, J.D.C.; Edwards, J.; McDonald, G.; Kuchel, H. Estimating Biomass and Canopy Height With LiDAR for Field Crop Breeding. *Front. Plant Sci.* **2019**, *10*, 1145. [[CrossRef](#)]
34. Lumme, J.; Karjalainen, M.; Kaartinen, H.; Kukko, A.; Hyypä, J.; Hyypä, H.; Jaakkola, A.; Kleemola, J. Terrestrial laser scanning of agricultural crops. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2008**, *37*, 563–566.
35. Bargoti, S.; Underwood, J.P. Image Segmentation for Fruit Detection and Yield Estimation in Apple Orchards. *J. Field Robot.* **2017**, *34*, 1039–1060. [[CrossRef](#)]
36. ten Harkel, J.; Bartholomeus, H.; Kooistra, L. Biomass and Crop Height Estimation of Different Crops Using UAV-Based Lidar. *Remote Sens.* **2019**, *12*, 17. [[CrossRef](#)]
37. Escolà, A.; Martínez-Casasnovas, J.A.; Rufat, J.; Arnó, J.; Arbonés, A.; Sebé, F.; Pascual, M.; Gregorio, E.; Rosell-Polo, J.R. Mobile terrestrial laser scanner applications in precision fruticulture/horticulture and tools to extract information from canopy point clouds. *Precis. Agric.* **2017**, *18*, 111–132. [[CrossRef](#)]
38. Andújar, D.; Dorado, J.; Bengochea-Guevara, J.M.; Conesa-Muñoz, J.; Fernández-Quintanilla, C.; Ribeiro, Á. Influence of wind speed on RGB-D images in tree plantations. *Sensors (Switzerland)* **2017**, *17*, 914. [[CrossRef](#)]
39. Rosell, J.R.; Sanz, R. A review of methods and applications of the geometric characterization of tree crops in agricultural activities. *Comput. Electron. Agric.* **2012**, *81*, 124–141. [[CrossRef](#)]
40. Auat Cheein, F.A.; Guivant, J.; Sanz, R.; Escolà, A.; Yandún, F.; Torres-Torriti, M.; Rosell-Polo, J.R. Real-time approaches for characterization of fully and partially scanned canopies in groves. *Comput. Electron. Agric.* **2015**, *118*, 361–371. [[CrossRef](#)]
41. Lin, K.; Lyu, M.; Jiang, M.; Chen, Y.; Li, Y.; Chen, G.; Xie, J.; Yang, Y. Improved allometric equations for estimating biomass of the three *Castanopsis carlesii* H. forest types in subtropical China. *New For.* **2017**, *48*, 115–135. [[CrossRef](#)]
42. Rosell, J.R.; Llorens, J.; Sanz, R.; Arnó, J.; Ribes-Dasi, M.; Masip, J.; Escolà, A.; Camp, F.; Solanelles, F.; Gràcia, F.; et al. Obtaining the three-dimensional structure of tree orchards from remote 2D terrestrial LIDAR scanning. *Agric. For. Meteorol.* **2009**, *149*, 1505–1515. [[CrossRef](#)]