Development of a Simulation Tool for 3D Plant Modeling based on 2D LiDAR Sensor

Harold F. Murcia, David A. Sanabria, Dehyro-Méndez and Manuel G. Forero
Facultad de Ingeniería, Universidad de Ibagué, Carrera 22 calle 67, Ibagué
730002, Colombia; e-mail: harold.murcia@unibague.edu.co

***Abstract*—The three-dimensional modeling of plants allows not only the use of color information, as in conventional digital image processing, but also the use of geometric information for the morphological extraction of their features and the subsequent analysis of their phenotype. The generation of point clouds is one of the initial stages of this process, which is carried out in different ways. One of the techniques used for this purpose uses a rotating platform and laser sensors, which employ multiple beams of light to illuminate the measurement area and determine its depth with the principle of time of flight (ToF). However, the algorithms used to perform the three-dimensional reconstruction must be calibrated in a process that may include a large number of experiments. For this reason, artificial three-dimensional point clouds generated by simulators may be suitable, both for the validation of reconstruction algorithms on those platforms and for the analysis of plant phenotype characteristics under almost realistic conditions. Thus, with this aim, this paper describes the development of an open-source tool for the generation of artificial 3D plant point clouds, based on the simulation tool Gazebo and the Robot Operating System (ROS). This work in progress allows validating different reconstruction algorithms, as well as the characteristics of LiDAR sensors and turntables to generate 3D models in an open file format. Our source implementation is freely available online and can be obtained from https://github.com/HaroldMurcia/3D-plantModeling-with-2DLiDAR.**

***Index Terms*—point cloud reconstruction, simulation of 3D modeling, LiDAR, plant phenotyping, machine vision, robot operating system**

I. Introduction

Plant phenotyping allows the analysis of complex plant traits like growth, architecture, physiology, yield, and some other parameters which determine more complex features. Reliable, automatic, and multifunctional phenotyping technologies are considered relevant tools for the rapid advancement of genetic gain in breeding programs [1]. Therefore, in the study of plant phenomics, the measurement of the 3D morphology of a plant plays an important role [2]. Improving the efficiency of phenotyping processes has become an important task for plant breeding programs [3], [4]. However, plant breeding can be a time-consuming and resource-intensive process; and in turn, the efficient use of those resources may be critical for the final results [5]. Nowadays many researchers use computer vision techniques combined with non-invasive sensors to study the phenotype of plants. These applications use different types of sensors to acquire multidimensional phenotypical data, such as red, green and blue cameras (RGB), RGB depth cameras (RGB-D), stereoscopic vision, structure light sensors, light detection and ranging devices (LiDAR), among others. Among these options LiDAR technology has increased its popularity among experts in different areas, so they are widely used for three-dimensional data acquisition [1], [6]. LiDAR is a remote sensing technology which measures the distance to an object by illuminating the target with laser and then analyzing the reflected light. LiDAR sensor employ a direct ranging measurement, determining the time-of-flight (ToF) of a light pulse, by measuring the elapsed time between the emitted and received beam to calculate the correct distances between objects. A 2D LiDAR device includes a rotating mirror that directs the emitted light beam depending on its angular position to obtain different range measurements of a scanned scene plane. For a 3D LiDAR, the idea is the same, but several laser beams spread out on the vertical axe are shot to get data on X, Y and Z axes. Each laser beam will have an angle delta with the other beams. However 3D LiDAR devices are expensive in comparison with general sensors and offers only a view of the scanned object, so as well as 2D sensors, a synchronization of different measurements is needed to have a full reconstruction.

Given the growing interest in generating 3D models of plants for study, both in field and laboratory applications, different hardware and software developments have emerged for their generation and further processing [7], [8]. However, the harmony of the elements involved in the reconstruction algorithms, together with the need to generate artificial models that speed up information processing studies, make it necessary to propose simulation tools that allow testing the different reconstruction algorithms in a 360° scan. At the same time, it makes possible to generate complete point clouds that bring together real models for the corresponding morphological analyses. Computer simulation is one of these possibilities, its essence is to sample as many conditions as possible that can be found in practice for any field, is fast and uses few physical resources, which makes it easily compliant with plant breeding. A virtual plant is a resource based on real structural and morphological data of plants. This computational approach brings a more intuitive data simulation of plants because allows generating more realistic 3D models in 360°. A 3D plant model is especially useful for research in phenotyping for plant breeding. Computer simulation can integrate physiological crop models, environmental information, and genetic composition of different crops to fill the gap between genotype and phenotype. Thus, plant breeding simulation platforms are becoming powerful tools to simulate the plant breeding process [5]. Understanding the biological processes involved in the development and functioning of plants requires efficiently using and combining computer models or methods from different scientific fields. Since these models are developed using different programming languages, different degrees of modularity, and operability, little attention is paid to code reuse and dissemination, such as packaging, installation procedures, portability to other operating systems, etc. This makes it difficult to exchange, combine or reuse models and simulation tools. The choice of programming language and the simulation environment also has important implications. Many modeling frameworks choose to utilize more efficient and flexible languages such as C++. To improve ease of use, many other developments adopts programming languages such as Python or Java [9], [10]. In this way, it is important to consider multilingual platforms that support developments from different programming possibilities. The Robot Operating System (ROS) is a middleware that is being highly adopted by many robotics platforms, and the proposed 3D plant modeling framework is suitable for mobile robots within agriculture tasks. In addition, ROS is compatible with 3D simulation environments as Gazebo, a simulation backend very widely used for robotic applications. It has a collection of tools and libraries to simplify the task of creating complex and robust robotic behavior across a wide variety of robotic platforms.

In this paper, an open-source tool for the simulation of the acquisition and generation process of three-dimensional plant point clouds is presented. A Gabezo environment (world) in which a low-cost 2D LiDAR sensor can be simulated was developed. Besides, a Gazebo plugin with ROS was created to interact with a simulated rotating platform, control the properties of the sensor and the reconstruction algorithm. A 3D point cloud database was created from a group of predefined virtual plants and the error between several measurements on these and the 3D models were calculated.

II. RELATED WORKS

The analysis of the digital representation of individual plants in three dimensions, in combination with detection technologies such as visible images and laser sensors in simulated environments, is a way to determine the topology of the plants, quantify the geometry and simultaneously evaluate their impact on plant phenotyping. In [9] was presented an open-source platform, called OpenAlea, that provided a user-friendly environment for modelers, and advanced deployment methods. OpenAlea allowed researchers to build models using a visual programming interface and provided a set of tools dedicated to the modeling of plants. In [10] was developed a three-dimensional plant and environmental modeling framework called Helios, which is a model coupling framework designed to provide maximum flexibility in integrating and running arbitrary 3D environmental system models. Helios comes with model plug-ins for radiation transport, photosynthesis, solar position, procedural tree generation, among others. Additional plug-ins are also available for visualizing model geometry and data and for processing and integrating LiDAR scanning data. In [11] some experiments in the 3D simulation environment Gazebo were carried out, with artificial maize plants in laboratory and on a small maize field, using the FX6 LIDAR by Nippon Signal. By using an algorithm based on an approach detecting the ground to segment the point cloud into the soil and other objects, an agricultural robot was able to detect reliably single plants in crop rows in real time. In [12] integrated hardware and software tools were developed in a project called PlantScan to provide automated, non-invasive analysis of plant structure and morphology. It provided an automatic digitalization of plants in three dimensions, enabling plant scientists to better understand the complex interactions involved in plant growth. In [2] a LiDAR-based device and a rotational stage to provide continuous rotation at 360° was developed, generating 3D point clouds for phenotyping of corn and sorghum plants. An LMS 511 SICK AG LiDAR module was selected for instrument development. The rotational speed was maintained at 3°/s. The LiDAR module and the rotational stage were controlled and synchronized in a program developed in LabVIEW. They established an algorithm that processed point clouds in four stages: background removal and voxelization, clustering and segmentation, triangulation and surface adjustment, and extraction of morphological features. Four plant morphological traits were extracted: individual leaf area, total leaf area, leaf inclination angle, and leaf angular distribution. The technique was validated with a leaf area measuring instrument on a group of 20 plants (10 maize's and 10 sorghum's) in various vegetative stages starting at six weeks of age. Point clouds processing was carried out in Matlab v2016 and the precision of the method was evaluated by calculating the coefficient R^2 and the mean absolute error (MAE) showing highly correlated data.

III. MATERIALS AND METHODS

Our system was inspired in a configuration which combines a 2D LiDAR, a DC motor, a motor driver, a rotating platform, an angular measurement or estimation subsystem, and a computer with a ROS environment as shown in Figure 1. For the simulation environment, the electronic components and mechanical references of the platform were irrelevant. Software process started reading the sensors signals to conditioning the information and to apply the kinematic transformation to a frame reference. This process was repeated for each angular position of the motor depending on the resolution and speed of rotation.

As a first step in this development, a 3D CAD model of the reference system was made using Solidworks 2017 software. Then, it was required to migrate each of the pieces modeled in Solidwork to Gazebo simulator [13]. This modeling was condensed into URDF and SDF files, which were written in XML code. In these files, variables such as lighting, obstacles, terrain, friction between the robot and the environment, the

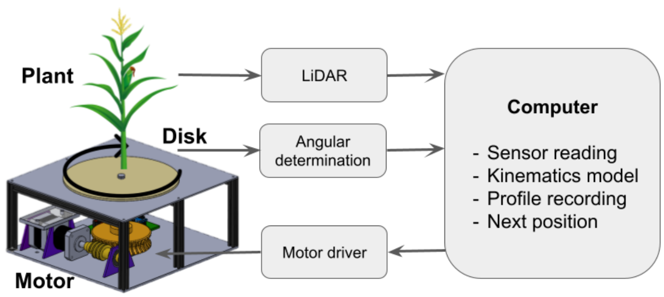


Fig. 1: Reference configuration for the simulation system

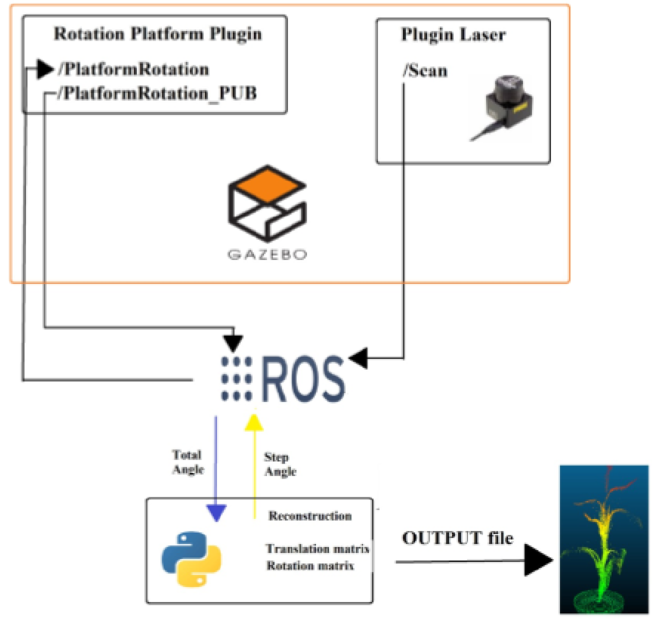


Fig. 2: Process flow diagram

force needed to move the robot, etc. can be detailed. It also must be specified the joins and plugins that should interact. This model specifies the parts of the robot or links, the unions that exist between each link, and the plugin or complement that is necessary. Each link had a name that identifies it, as well as the definition of the area, figure or point with which it will be in contact with the other parts of the system and the world in which it is simulated. Additionally, a visual part was defined which is what the developer will be able to see in the simulator. Likewise, the links and plugins that must interact were specified. In this model, the parts of the robot or the links, the connections between each link and the plugin or complement that are necessary were specified. A C++ plug-in was then created to allow the system to perform different actions such as movement, rotation, data capture through sensors, among others. This plug-in sent a position expressed in radians to the platform. The movement was controlled by a PID, while the angle of the platform was published. A second plugin performed the sensor function, which was connected to a laser sensor link emitting the laser distances and thus simulating the LiDAR sensor. Sensor parameters such as minimum and maximum distance, resolution, and beam angle must be configured within the SDF model before launching the world. Finally, the system was linked to ROS to be able to use the created plugins and assign tasks to the system. The control of the platform was carried out through a code in Python. For this, a node and some topics were created, which were in charge of receiving and publishing the angle that the platform should move and the current angle respectively. Figure 2 summarizes the simulation process in a flowchart involving the Gazebo simulator, the plugins and the main program. A Gazebo world launcher was created to build this system and communicate its simulation environment, allowing to carry out a command to start ROS and Gazebo. The world in the simulator will have the system model, which is the rotating platform, and physical characteristics of both the robot and its environment.

Once the 3D model of the system and the simulation parameters of the world were configured in Gazebo, a node was developed in Python to manage the movements of the engine, the reading of the sensors, the conditioning of the data and its respective storage. The movements are set in steps, whose values in degrees depends on the resolution with which the simulation is configured. The reading refers to the angular position estimated from the initial position, and the steps are measured by an encoder. On the other hand, the reading of the LiDAR sensor is done through the "scan" topic, which has a message format "LaserScan". The data conditioning, which is the previous step to the storage, extracts the estimated value of the rotation from the initial angle and the measured angle. The range data, shown in meters, and the angle of measurement of the lidar, were transformed into a cloud of XYZ points. As shown in Figure 3, in this model the platform (P2) had as its origin a Cartesian coordinate system different from the LiDAR sensor, so it was necessary to apply a translation matrix to work in a coordinate system where the LiDAR would be in the point P2. The conditioning of the data into a single frame of reference can be done employing a matrix equation, in which transformation matrices between translations and rotations are included, as specified in a previous work [14].

The proposed Kinematic model aims to transform all the data obtained into a reference framework centered on point P2. In a general representation, the transformation matrices were based on a basic point transformation for a 3D space. The measurements obtained from the LiDAR were converted into XYZ Cartesian coordinates as $X = ranges * cos(\alpha)$, $Y = ranges * sin(\alpha)$ and Z a nule vector with size $[Nx1]$, where α is the LiDAR angle and N is the number of voxels per laser scan. So, the parameters t_x, t_y and t_z shown in (1) represents the respective translations from point P1 to point P2 along X, Y, Z axes. T is then, the translation matrix.

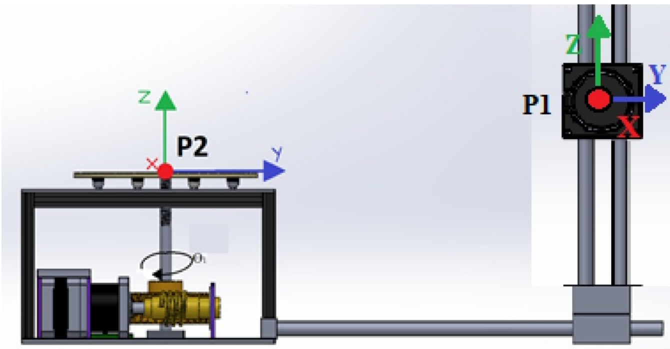


Fig. 3: Schematic diagram and notations for the reconstruction system based on rigid transformations from LiDAR raw data to a reference point

$$T = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

R_z is the rotation matrix around the Z axis, which is a function of the radial distance γ, as shown in (2).

$$R_z = \begin{bmatrix} cos(\gamma) & -sin(\gamma) & 0 & 0 \\ sin(\gamma) & cos(\gamma) & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

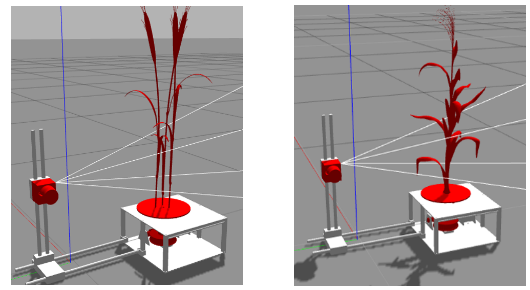
The resulting 3D reconstruction is a three-frame transformation based on previous matrices, as shown in (3).

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} cos(\gamma) & -sin(\gamma) & 0 & 0 \\ sin(\gamma) & cos(\gamma) & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

where γ represents the rotated angle for the moment $t = 0$ with respect to the initial position. The transformation presented in (3) was applied for each movement from the initial state to the last position. Finally, a flat file in TXT format is created with the transformed data in a X,Y,Z structure; which can be easily transformed to other formats such as LAS, PCD or PLY by using additional software such as: CloudCompare, Laspy for Python, LasTools, etc. The tool was developed in ROS Melodic, Gazebo 9 and Python 2.7. The simulations were executed on a computer with Intel CORE i5 7th Gen 2.5 GHz with 8 GB of RAM, under Ubuntu 18.04 operating system. Simulation time expend a couple of minutes according to the maximum angle and step resolution. The simulated sensor has the characteristics of a low-cost device, Hokuyo URG-04LX-UG01 2D LiDAR with a scanning angle of 240° and in 0.36° resolution steps. Laser parameters such as maximum distance, minimum distance, opening angle and resolution can be changed by configuring the platform.world file, in the section of the platform model, subsection <link> camera, in the parameter <laser sensor>.

IV. RESULTS

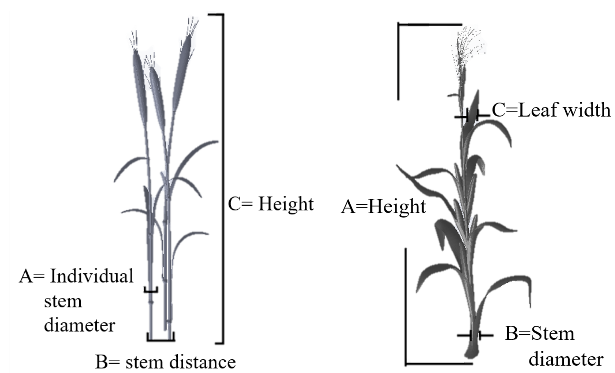
Figure 4 shows two open access plant models with a STL file integrated in a SDF format [15]: a wheat plant and a maize plant; which were used into the simulated platform. Both plants were scanned with a resolution of 1 degree along the 360 degrees.



(a) (b)

Fig. 4: Simulated platform in Gazebo with plants: (a) wheat virtual plant, (b) maize virtual plant

The 3D point clouds obtained in the Gazebo simulation environment were saved in LAS format and displayed together with the predefined virtual plant models in the CloudCompare software. The whole acquisition process took about 40 minutes. Figure 5 details the measurements done in the virtual plants to estimate the error of three parameters in each model, called A, B, and C, which correspond to longitudinal measurements of each model. The error was calculated according to (4), where S is the measurement made in CloudCompare on the point cloud generated by simulation and R is the measurement on the virtual plant, taken as reference.

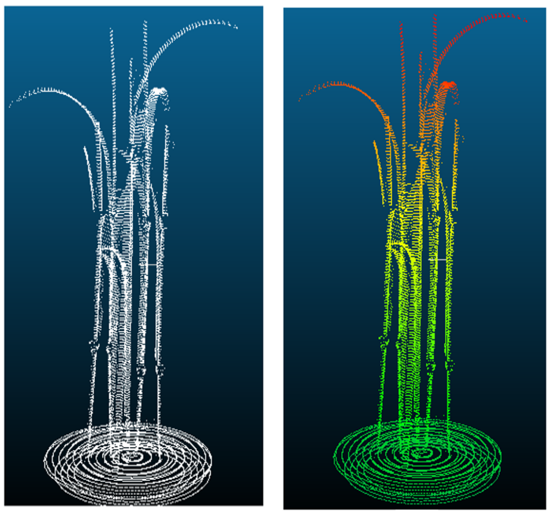


(a) (b)

Fig. 5: Plant models: (a) wheat virtual plant, (b) maize virtual plant

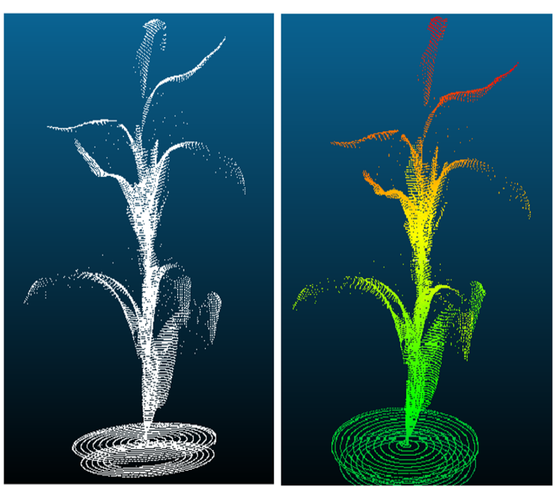
$$error[\%] = \left| \frac{S - R}{R} \right| \tag{4}$$

Figure 6 shows a frontal view of the resulting 3D point cloud after simulation of acquisition with the wheat virtual plant. From this 3D reconstruction were extracted the height of the plant, an individual stem diameter, and the distance between stems. In Figure 7 can be seen the 3D point cloud obtained using the maize virtual plant as the target for the simulation process. From this 3D reconstruction were extracted the height of the plant, the stem diameter, and leaf width. Tables I and II show all measurements for each point cloud and the calculated error for 1 degree and 0.5 degree steps respectively.



(a) (b)

Fig. 6: Reconstructed 3D wheat plant point cloud with 1 degree steps: (a) frontal view, (b) frontal view using a color scale based on altitude



(a) (b)

Fig. 7: Reconstructed 3D maize plant point cloud with 1 degree steps: (a) frontal view, (b) frontal view using a color scale based on altitude

TABLE I: Error determination from the simulated and virtual point cloud measurements for the wheat plant with 1 degree steps

Parameter	Simulated [mm]	Ref. value [mm]	Error [%]
Resolution: 1 degree steps			
A	5.00	4.00	25.00
B	47.00	42.00	11.90
C	442.00	555.00	20.36
Resolution: 0.5 degree steps			
A	4.20	4.00	5.00
B	42.0	42.00	0.00
C	490.0	555.00	11.71

TABLE II: Error determination from the simulated and virtual point cloud measurements for the maize plant with 1 degree steps

Parameter	Simulated [mm]	Ref. value [mm]	Error [%]
Resolution: 1 degree steps			
A	411.00	417.00	1.43
B	12.00	12.50	4.00
C	15.78	18.60	15.15
Resolution: 0.5 degree steps			
A	412.69	417.00	1.03
B	12.00	12.50	4.00
C	16.00	18.60	13.97

According to the results from Tables I and II, both reconstruction processes had similar accuracy. In both Tables it is evident that the values obtained with the point cloud from the simulation were mostly lower than those obtained with the virtual plant. In the same way, it is evident that the general error decreases as the resolution of the steps on the platform improves. A factor that might influenced the mayor error in some results was the simulation process, which required the use of a large number of computational resources, so it would be affected by the performance of the equipment used.

V. CONCLUSIONS

This paper describes the development of a software tool to simulate the 3D plant modeling process in a ROS-Gazebo Framework. This simulation can be used as a base work for simulating other objects and trying new reconstruction configurations and algorithms. It can also be used to generate artificial point clouds at different resolutions for plant phenotyping applications. All code from this project is free to use, distribute, and modify.

The approach carried out, under a computer simulation environment, brings an approximation to real scenarios in reconstruction systems, but since there is a gap between our knowledge and actual plant phenotyping practices, a huge amount of information would be needed to emulate an accurate 3D plant point cloud reconstruction process. In a real situation, plants can move due to airflows, occlusions are more common, and lighting conditions must be managed too. So, we need more innovative tools to help us integrate the knowledge we have gained on plant morphology analysis during years into plant-phenotyping practices that make us address those problems and accelerate plant improvement programs.

Future work would involve to build plugins for simulating laser intensity and pulse width information, the option of validation algorithms for kinematic model tuning and automatic parameter estimations, as well as migrate the simulation to a real scanning platform.

Acknowledgments

This work was funded by the OMICAS program: Optimizacion Multiescala In-silico de Cultivos Agricolas Sostenibles (Infraestructura y validacion en Arroz y Caña de Azucar), anchored at the Pontificia Universidad Javeriana in Cali and funded within the Colombian Scientific Ecosystem by The World Bank, the Colombian Ministry of Science, Technology and Innovation, the Colombian Ministry of Education and the Colombian Ministry of Industry and Turism, and ICETEX under GRANT ID: FP44842-217-2018.

References

[1] Z. Li, R. Guo, M. Li, Y. Chen, and G. Li, "A review of computer vision technologies for plant phenotyping," vol. 176, p. 105672. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0168169920307511

[2] S. Thapa, F. Zhu, H. Walia, H. Yu, and Y. Ge, "A novel LiDAR-based instrument for high-throughput, 3d measurement of morphological traits in maize and sorghum," vol. 18, no. 4, p. 1187. [Online]. Available: http://www.mdpi.com/1424-8220/18/4/1187

[3] L. Li, Q. Zhang, and D. Huang, "A review of imaging techniques for plant phenotyping," vol. 14, no. 11, pp. 20 078–20 111. [Online]. Available: http://www.mdpi.com/1424-8220/14/11/20078

[4] S. Paulus, J. Behmann, A.-K. Mahlein, L. Plümer, and H. Kuhlmann, "Low-cost 3d systems: Suitable tools for plant phenotyping," vol. 14, no. 2, pp. 3001–3018. [Online]. Available: http://www.mdpi.com/1424-8220/14/2/3001

[5] X. Li, C. Zhu, J. Wang, and J. Yu, "Computer simulation in plant breeding," in *Advances in Agronomy*. Elsevier, vol. 116, pp. 219–264. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/B9780123942777000063

[6] Z. Fang, S. Zhao, S. Wen, and Y. Zhang, "A real-time 3d perception and reconstruction system based on a 2d laser scanner," vol. 2018, pp. 1–14. [Online]. Available: https://www.hindawi.com/journals/js/2018/2937694/

[7] M. Vázquez-Arellano, D. S. Paraforos, D. Reiser, M. Garrido-Izard, and H. W. Griepentrog, "Determination of stem position and height of reconstructed maize plants using a time-of-flight camera," vol. 154, pp. 276–288. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S016816991830379X

[8] D. Reiser, M. Vázquez-Arellano, D. S. Paraforos, M. Garrido-Izard, and H. W. Griepentrog, "Iterative individual plant clustering in maize with assembled 2d LiDAR data," vol. 99, pp. 42–52. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0166361517304748

[9] C. Pradal, S. Dufour-Kowalski, F. Boudon, C. Fournier, and C. Godin, "OpenAlea: a visual programming and component-based software platform for plant modelling," vol. 35, no. 10, p. 751. [Online]. Available: http://www.publish.csiro.au/?paper=FP08084

[10] B. N. Bailey, "Helios: A scalable 3d plant and environmental biophysical modeling framework," vol. 10, p. 1185. [Online]. Available: https://www.frontiersin.org/article/10.3389/fpls.2019.01185/full

[11] U. Weiss and P. Biber, "Plant detection and mapping for agricultural robots using a 3d LIDAR sensor," vol. 59, no. 5, pp. 265–273. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0921889011000315

[12] X. Sirault, J. Fripp, A. Paproki, P. Kuffner, C. Nguyen, R. Li, H. Daily, J. Guo, and R. Furbank, "PlantScan: A three-dimensional phenotyping platform for capturing the structural dynamic of plant development and growth."

[13] S. E. Blog. (2020) Modeling for gazebo. [Online]. Available: https://blogs.solidworks.com/teacher/wp-content/uploads/sites/3/WPI-Robotics-SolidWorks-to-Gazebo.pdf

[14] H. F. Murcia, M. F. Monroy, and L. F. Mora, "3d scene reconstruction based on a 2d moving LiDAR," in *Applied Informatics*. Springer International Publishing, vol. 942, pp. 295–308. [Online]. Available: http://link.springer.com/10.1007/978-3-030-01535-0_22

[15] Gazebo tutorial: Import meshes. [Online]. Available: http://gazebosim.org/tutorials?tut=import_mesh&cat=build_robot

This is to certify that

Harold Murcia

Has presented a paper titled

Development of a Simulation Tool for 3D Plant Modeling based on 2D LiDAR Sensor
at the **Virtual Symposium in Plant Omics Science**

Held between November 23rd - 27th, 2020 at Santiago de Cali, Colombia

Andrés Jaramillo Botero
Scientific Director

Jaime Aguilar
Institutional Strengthening Sub-director

Support: