Automatic Generation of Labeled 3D Point Clouds of Natural Environments with Gazebo*

Manuel Sánchez, Jorge L. Martínez, Jesús Morales, Alfredo Robles and Mariano Morán

***Abstract*— Progress in applying supervised learning for natural scene classification is impeded by the lack of appropriate datasets for training. This paper describes the automatic generation of synthetic three-dimensional (3D) scans of natural environments with each point labelled individually with its element class. The developed software employs the robotic simulator Gazebo to obtain range and intensity measurements from a 3D laser rangefinder aboard a ground mobile robot. Precisely, the returned intensity values are used to annotate every 3D point within its corresponding class 100% error free. Several examples are provided to show the utility of the proposed approach.**

***Index Terms*— 3D laser rangefinder, autonomous ground vehicles, sensor simulation, supervised learning.**

I. Introduction

Scene classification is a relevant issue for autonomous navigation of ground mobile robots in natural environments [1]. To apply supervised learning techniques for semantic segmentation it is necessary to employ training data already tagged.

In scientific-technique literature, it is possible to find several repositories with three-dimensional (3D) point clouds from urban environments labeled with different elements such as facade, vehicle, ground, wire, vegetation, etc. [2] [3]. Usually, these datasets are obtained from real 3D scans that have been tagged manually [4] or annotated interactively with the help of software tools [5].

These repositories can be employed by supervised learning algorithms to train predictive models that can be applied later to classify new unlabeled scenes [6] [7]. However, as far as the authors know, there are no labeled repositories for the navigation of ground mobile robots in natural environments. In this case, recorded 3D point clouds are hard to interpret because of varying point density, occlusions and lack of structure. Thus, tagging point by point real 3D laser scans from natural scenes is a difficult, tedious and error-prone task.

An alternative is to use synthetic depth data. This kind of data has been useful to distinguish objects in indoor scenes [8], to estimate human poses [9], to detect cyclists in urban traffic scenes [10], to train autonomous driving [11] [12], or to assess traversability on planetary terrain [13]. In these cases, the software tools used for generating the 3D virtual data are very diverse and include ray tracing from a video game [11], the generic Matlab testbed [13] and a specific toolbox for simulating range scanners [14].

This work is intended to synthesize automatically labeled 3D point clouds of natural environments with the robotic simulator Gazebo [15]. In particular, it emulates the 3D laser rangefinder of the ground mobile robot Andabata [16]. This 3D sensor is based on rotating a standard two-dimensional (2D) scanner around its optical center [17].

The paper is organized as follows. Next section overviews the 3D laser rangefinder that will be simulated. Then, section III presents the natural environment generated with Gazebo. Section IV describes the developed application. Some illustrative examples obtained with the proposed approach are shown in section V. Finally, section VI is devoted to conclusions and future work.

II. The 3D laser scanner of Andabata

Andabata is a skid-steer mobile robot powered by batteries for outdoor navigation (see Fig. 1). On top of the robot and centered, a 3D laser rangefinder with a horizontal field of view of 360° provides information of the surroundings [16].

The 3D scanner is based on the unrestrained rotation of a 2D laser rangefinder around its optical center to produce vertical 2D scans in a continuous way [17]. The horizontal

Fig. 1. Andabata mobile robot with its 3D laser scanner on top.

*This work was supported partially by the Spanish project DPI 2015-65186-R.

All authors are within Universidad de Málaga, Andalucía Tech, Dpto. Ingeniería de Sistemas y Automática, 29071-Málaga, Spain. Emails: mansanmon@outlook.es, jlmartinez@uma.es, jesus.morales@uma.es, aroblesaragon@outlook.com, marianomoran@uma.es

resolution r_h is adjustable by varying the rotation speed of the 2D sensor.

The main features of the employed 2D laser scanner, a Hokuyo UTM-30LX-EW, of this 3D sensor are: minimum range of $0.1\,\mathrm{m}$, maximum range of $15\,\mathrm{m}$ (under direct sunlight), vertical resolution $r_v = 0.25°$, field of view of $270°$ and measurements of reflectivity of surfaces (i.e., intensity).

The blind zone of 3D laser rangefinder is a cone whose vertex is in its optical center which is located at a height $h = 0.723\,\mathrm{m}$ above the ground. This distance coincides with the base radius of the cone on the terrain. In this way, the whole robot falls within the cone and does not interfere with any measurements.

III. Modeling of Natural Environments with Gazebo

Gazebo allows the development of realistic models of large-scale environments. For example, it is employed currently in the DARPA subterranean challenge[1]. Nevertheless, it has been found that the simulation time increases linearly with the number of elements of the environment [18].

The first step is to build a digital elevation map (DEM) for the terrain from a 2D grayscale image. However, there is a relevant bug related with laser sensor measurements on abrupt zones of Gazebo height maps[2]. To avoid this problem, a DEM has been generated with Blender[3] and treated as another element of the environment.

Apart from uneven terrain, several 3D models of natural and artificial elements have been incorporated to the virtual environment. In this way, elements like grass, bushes, rocks, tables, benches and fences have been placed on the ground with different sizes.

Some 3D models have been imported in parts. Such is the case with trees, where trunks and branches are inserted apart from their crowns. Moreover, power lines are separated into the pole and the wire elements.

Special 3D models are employed to fill some gaps on the DEM. This is the case of flooring and water elements to create a trail and a lake, respectively. A cave has also been included in the environment by covering a hole on the terrain partially with special flooring.

A general view of the generated environment with Gazebo can be observed in Fig. 2. All in all, its maximum dimensions are $150\,\mathrm{m}$ length, $150\,\mathrm{m}$ wide and $20\,\mathrm{m}$ height. It contains different zones, namely, hills, forests, a park, a lake, a cave and a trail.

IV. Developed Application

Figure 3 shows the general outline of the application, which involves the robotic simulator Gazebo [15], the robot operating system (ROS) [19] and the computing environment Matlab.

[1]https://www.subtchallenge.com/resources.html

[2]http://answers.gazebosim.org/question/20267/problem-with-laser-sensor-and-heigthmap/

[3]https://www.blender.org

Gazebo is employed, apart from building the model of a natural environment, to simulate the measurements of a standard 2D laser rangefinder. Matlab is used to provide an additional rotation to the 2D sensor and to compose the labeled 3D point cloud. ROS serves as an interface between Gazebo and Matlab.

First, the user indicates the position and orientation from which the 3D scan will be acquired on the ground. This is accomplished by means of a Matlab interface that includes an aerial view of the synthetic environment (see Fig. 4).

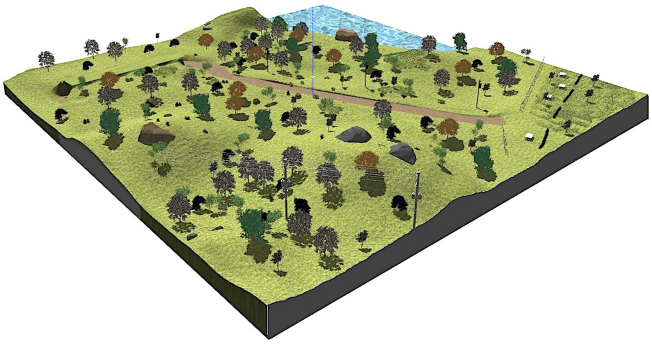

Fig. 2. General view of the generated natural environment.

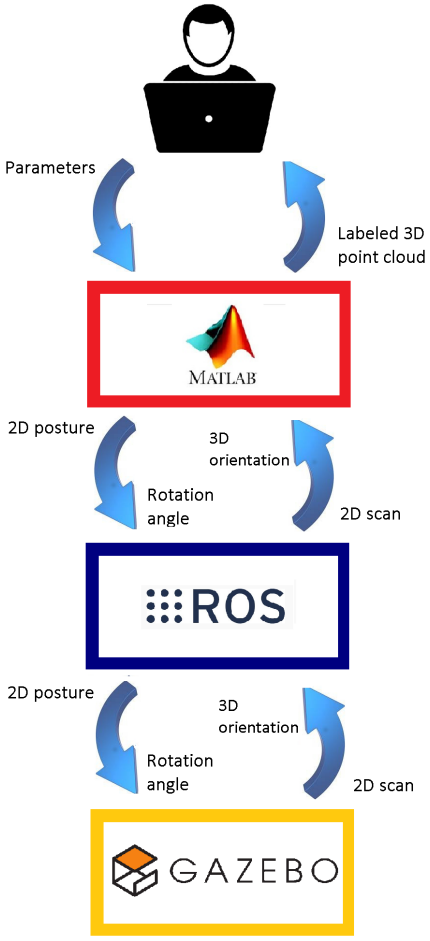

Fig. 3. General schema of the proposed approach.

Then, Matlab obtains successive 2D scans from the Gazebo simulation of the 2D laser scanner with different rotation angles, and combines them together in a complete 3D point cloud.

Next subsections describe the processing associated with each software component.

A. *Gazebo sensor simulation*

The 2D Hokuyo sensor is already included in the Gazebo model database[4]. Simulated range measurements are affected by Gaussian noise with null bias and a standard deviation of $1.5\,\mathrm{cm}$.

Apart from noisy ranges, the 2D laser scanner obtains intensity estimates that depend on the values assigned to each element of the virtual environment. Precisely, this data will allow to tag correctly each 3D point within its corresponding class.

Table I indicates the arbitrary color tags and reflectivity values assigned to each element of the synthetic environment. To emulate a laser beam that is deflected by the water surface, ranges with water reflectivity will be removed from the 3D point cloud. This is why, the water element has no color tag assigned.

Instead of building a complete model for Andabata in Gazebo, a simpler structure has been developed to carry the 3D laser scanner. The structure contain two arms for the local X and Y axes and a column for the Z axis that points upwards (see Fig. 5-left).

The 2D laser scanner is attached at the upper extreme of the column with its optical center aligned with the Z axis. Successive rotation angles are applied to the column in order to obtain a full 3D scan by turning it $180°$ with a selectable horizontal resolution r_h.

[4]https://bitbucket.org/osrf/gazebo_models/src

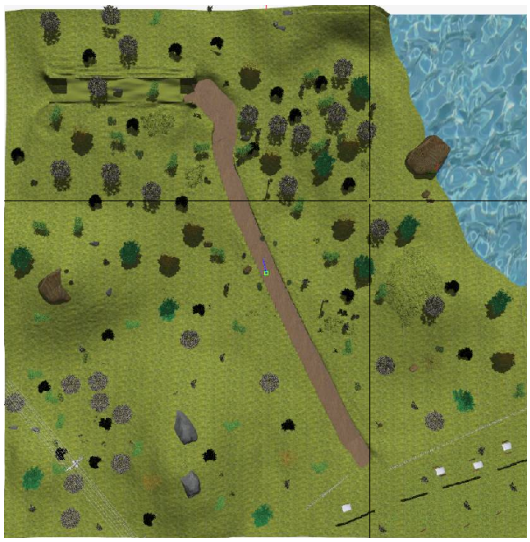

Fig. 4. Aerial view of the natural environment. The intersection between the two perpendicular black lines indicates the position from which the 3D scan will be acquired.

At both ends of the arms there are one-dimensional (1D) laser scanners pointing downwards. These sensors employ the same Hokuyo model, but with the number of laser beams restricted to one. The distances provided by the four 1D scanners are employed to place the structure with pitch and roll angles similar to those that Andabata would have on the ground and to obtain the height h for the 3D sensor (see Fig. 5-right). On the other hand, the four intensity measurements are used to check that the structure is located entirely on the ground.

A representation of a simulated 2D scan, once the structure has been placed properly on the ground of the virtual environment, can be observed in Fig. 6.

TABLE I

COLOR TAGS AND REFLECTIVITY VALUES ASSIGNED TO THE ELEMENTS.

Element	Color Tag	Reflectivity
Water		120
Terrain & flooring	Brown	100
Trunk & branch	Magenta	90
Treetop	Olive	80
Table	Yellow	75
Bush	Green	70
Fence	Black	60
Bench	Pink	55
Low grass	Orange	50
Rock	Gray	40
High grass	Red	30
Wire	Blue	20
Pole	Cyan	15

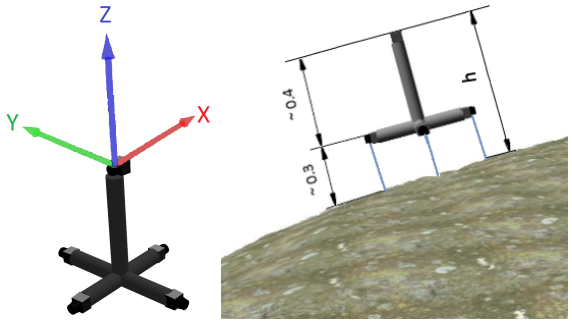

Fig. 5. The structure to carry the 3D laser rangefinder (left) and its placement on the terrain (right).

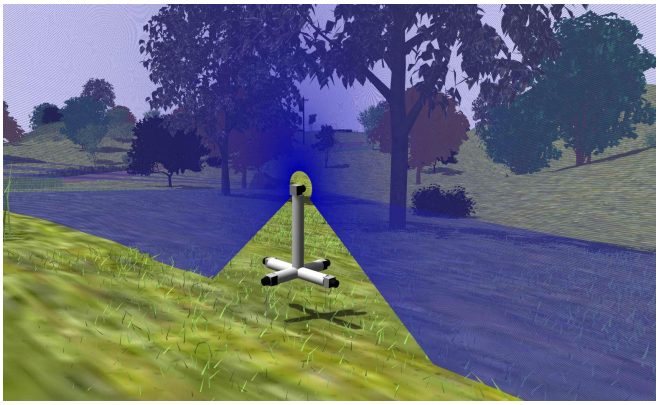

Fig. 6. Simulation of a 2D scan in the synthetic environment.

B. Implementation in ROS

Fig. 7 shows the ROS computation graph developed for the application, which includes two nodes and seven topics. The Gazebo node was generated using the gazebo_ros[5] packages, whereas the Matlab node was created with the Robotics System Toolbox[6].

The `/laser1D1-/laser1D4` topics contain the range and intensity of every 1D laser scanner. Similarly, the `/laser2D` topic holds a complete 2D scan composed of 1080 range and reflectivity measurements.

Gazebo send messages with the current 3D global orientation of the 2D laser scanner through the `/gazebo/model_states` topic. On the other way, the `/gazebo/set_model_state` topic is employed by Matlab to place the 3D sensor structure on the ground, and to provide the successive rotation angles for the 2D sensor.

C. Matlab processing

The Matlab program of the application provides the interface with the user, places the 3D sensor structure on the ground and indicates rotation angles for the 2D sensor. In addition, it composes the whole 3D point cloud from consecutive 2D scans.

Firstly, the 3D Cartesian coordinates of every range are calculated depending on their angle inside the 2D scan and on the additional rotation of the whole 2D scan. Let ρ_n^m be the n^{th} range acquired inside the m^{th} 2D scan. The local Cartesian coordinates (x_n^m, y_n^m, z_n^m) associated to this range are calculated with Eq. (1).

Secondly, the 3D scan is leveled by using the opposite roll (α) and pitch (β) angles that have been employed to place the 3D sensor structure on the ground. For this purpose, Eq. (2) is employed to obtain the Cartesian coordinates in the leveled point cloud (X_n^m, Y_n^m, Z_n^m), i.e., with the Z axis standing vertical. The resulting point cloud does not coincide with the one that would be obtained with the structure for the 3D sensor leveled over the ground.

Finally, each 3D point is labeled within its corresponding element class depending on its intensity measurement, with the exception of those points with water reflectivity that are removed from the 3D point cloud.

Fig. 8 shows the steps involved in the acquisition of a labeled 3D scan on the hill zone of the virtual environment, which includes an electric line. The 3D points have been labeled with the color tags of Table I. The empty circle that can be observed on the ground corresponds to the blind zone of the 3D scanner of Andabata. It is also noticiable that point density decreases with the distance to the 3D sensor.

[5]http://gazebosim.org/tutorials?tut=ros_overview
[6]https://mathworks.com/hardware-support/robot-operating-system.html

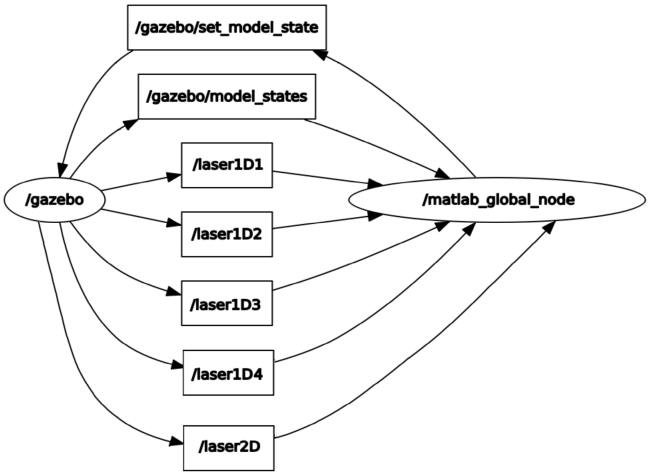

Fig. 7. ROS nodes (ellipses) and topics (rectangles).

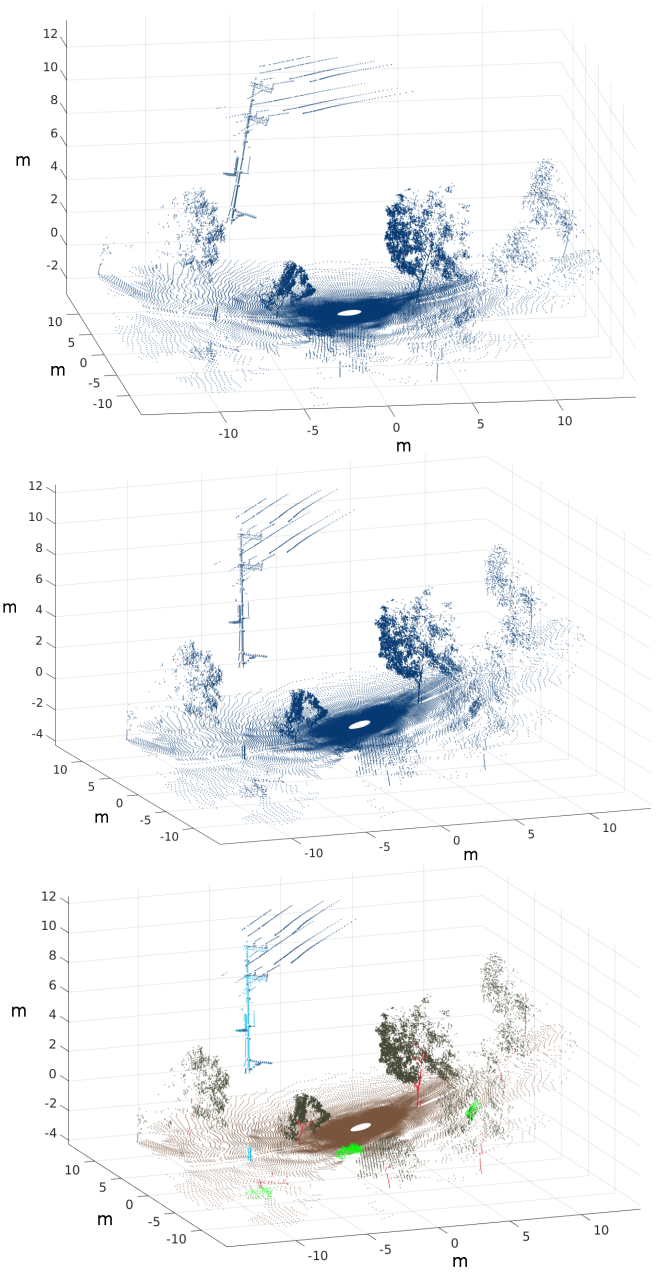

Fig. 8. A 3D scan of 83056 points acquired on a hill before leveling (top), after leveling (middle) and after tagging (bottom).

$$\begin{pmatrix} x_n^m \\ y_n^m \\ z_n^m \end{pmatrix} = \begin{pmatrix} \cos(m\,r_h) & -\sin(m\,r_h) & 0 \\ \sin(m\,r_h) & \cos(m\,r_h) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \sin(n\,r_v - 135°) \\ 0 \\ \cos(n\,r_v - 135°) \end{pmatrix} \rho_n^m \tag{1}$$

$$\begin{pmatrix} X_n^m \\ Y_n^m \\ Z_n^m \end{pmatrix} = \begin{pmatrix} \cos(-\beta) & 0 & -\cos(-\beta) \\ 0 & 1 & 0 \\ \sin(-\beta) & 0 & \cos(-\beta) \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(-\alpha) & -\sin(-\alpha) \\ 0 & \sin(-\alpha & \cos(-\alpha) \end{pmatrix} \begin{pmatrix} x_n^m \\ y_n^m \\ z_n^m \end{pmatrix} \tag{2}$$

V. Examples of Labeled 3D Scans

Representative zones in the virtual environment have been scanned and their corresponding 3D scans labeled automatically with the proposed application. Apart from the hill zone of Fig. 8, Figs. 9, 10, 11 and 12 show 3D scans acquired near the lake, the park, the cave and the forest, respectively. All of them have been acquired with a horizontal resolution $r_h = 1°$. This means that each 3D laser scan is composed of 180 2D scans and a maximum of 194400 points.

In Fig. 9, it can be observed that points from the water element do not appear in the 3D scan. Different artificial elements such as fences, tables or benches appear tagged in Fig. 10 with the colors of Table I. In Fig. 11 several points have been scanned from inside the cave. The forest zone in Fig. 12 includes different kinds of trees, bushes, and grass.

Nevertheless, these are only a few examples and almost an unlimited number of labeled scans can be obtained easily by changing the initial posture of the 3D laser scanner in the natural environment through the user interface.

VI. Conclusions

This paper has described the automatic generation of synthetic 3D laser scans of natural environments with each Cartesian point tagged individually with its corresponding element class. The developed software employs Gazebo to simulate the acquisition of range and intensity measurements from the 3D laser rangefinder aboard the ground mobile robot Andabata.

To annotate the class of each 3D point without errors, arbitrary reflectivity values have been assigned to each element of the environment modeled with Gazebo. The usefulness of this approach has been shown with the generation of realistic point clouds that are very similar to those captured by Andabata, with the exception that the virtual data is already segmented semantically.

Future work includes the use of the labeled data for supervised learning to train a predictive model to detect traversable terrain for autonomous navigation. It is also of interest to capture 3D laser scans in motion by simulating Andabata with Gazebo.

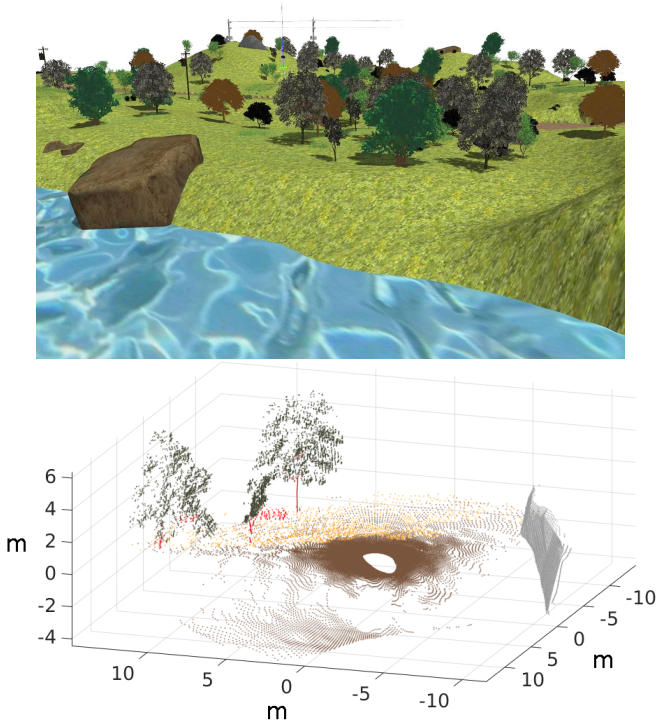

Fig. 9. A view of the lake zone (top) and a labeled 3D scan of 63998 points captured near the shore (bottom).

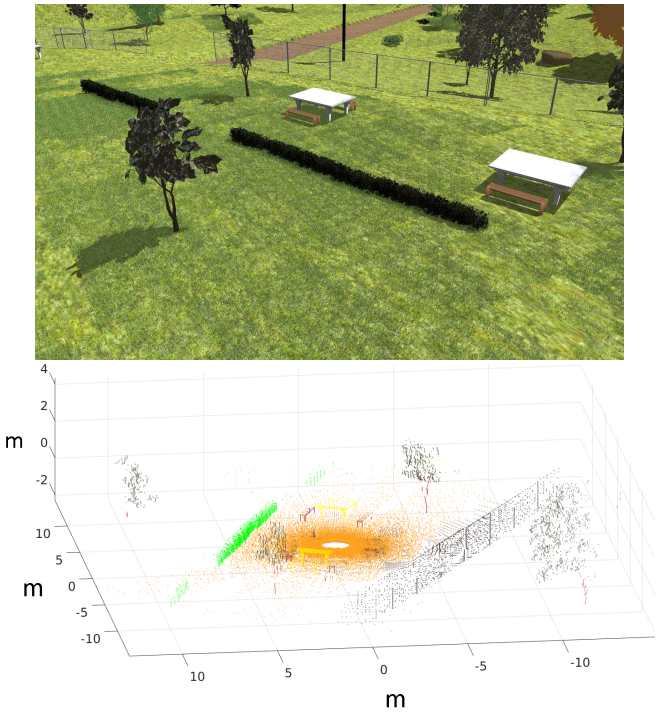

Fig. 10. A view of the park zone (top) and a labeled 3D scan of 60669 points acquired near the tables (bottom).

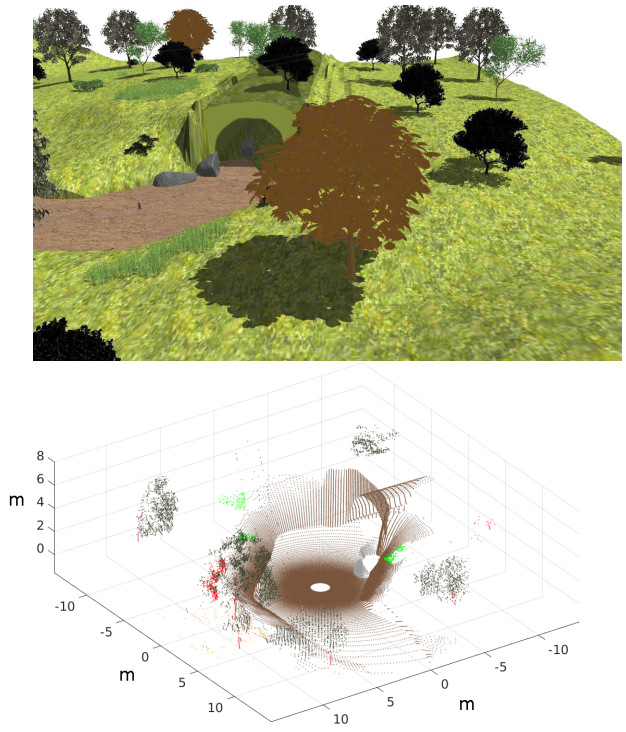

Fig. 11. A view of the cave zone (top) and a labeled 3D scan of 80307 points captured near the entrance (down).

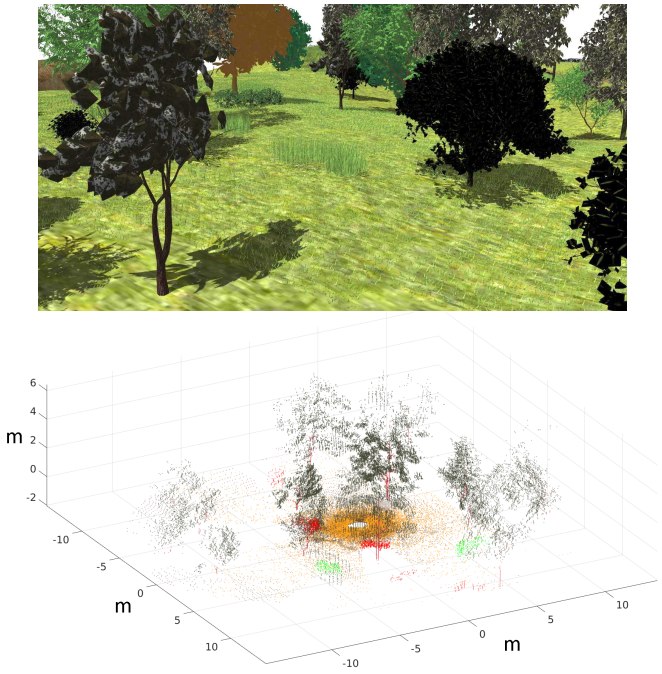

Fig. 12. A view of the forest zone (top) and a labeled 3D scan of 85479 points acquired from inside (bottom).

REFERENCES

[1] J. Lalonde, N. Vandepel, D. Huber, and M. Hebert, "Natural terrain classification using three-dimensional ladar data for ground robot mobility." *Journal of Field Robotics*, vol. 23, pp. 839–861, 2006.

[2] X. Xiong, D. Munoz, J. Bagnell, and M. Hebert, "3-D scene analysis via sequenced predictions over points and regions," in *Proc. IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011, pp. 2609–2616.

[3] J. Behley, V. Steinhage, and A. Cremers, "Performance of histogram descriptors for the classification of 3D laser range data in urban environments," in *Proc. IEEE International Conference on Robotics and Automation*, Saint Paul, USA, 2012, pp. 4391–4398.

[4] M. De Deuge, A. Quadros, C. Hung, and B. Douillard, "Unsupervised feature learning for classification of outdoor 3D scans," in *Proc. Australasian Conference on Robotics and Automation*, Sydney, Australia, 2013, pp. 1–9.

[5] T. Hackel, N. Savinov, L. Ladicky, J. D. Wegner, K. Schindler, and M. Pollefeys, "Semantic3D.net: A new large-scale point cloud classification benchmark," *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. IV-1-W1, pp. 91–98, 2017.

[6] A. Pomares, J. L. Martínez, A. Mandow, M. A. Martínez, M. Morán, and J. Morales, "Ground extraction from 3D lidar point clouds with the classification learner app," in *Proc. 26th Mediterranean Conference on Control and Automation*, Zadar, Croatia, 2018, pp. 400–405.

[7] A. Zaganidis, L. Sun, T. Duckett, and G. Cielniak, "Integrating deep semantic segmentation into 3-D point cloud registration," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2942–2949, 2018.

[8] A. Handa, V. Patraucean, S. Stent, and R. Cipolla, "SceneNet: an annotated model generator for indoor scene understanding," in *Proc. IEEE International Conference on Robotics and Automation*, Stockholm, Sweden, 2016, pp. 5737–5743.

[9] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Colorado Springs, USA, 2011, pp. 1297–1304.

[10] K. Saleh, M. Hossny, A. Hossny, and S. Nahavandi, "Cyclist detection in LIDAR scans using faster R-CNN and synthetic depth images," in *Proc. IEEE International Conference on Intelligent Transportation Systems*, Yokohama, Japan, 2017, pp. 1–6.

[11] X. Yue, B. Wu, S. A. Seshia, K. Keutzer, and A. L. Sangiovanni-Vincentelli, "A LiDAR point cloud generator: from a virtual world to autonomous driving," in *Proc. ACM International Conference on Multimedia Retrieval*, Yokohama, Japan, 2018, pp. 458–464.

[12] A. Dosovitskiy, G. Ros, F. Codevilla, A. López, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. 1st Conference on Robot Learning*, Mountain View, USA, 2017, pp. 1–16.

[13] R. A. Hewitt, A. Ellery, and A. de Ruiter, "Training a terrain traversability classifier for a planetary rover through simulation," *International Journal of Advanced Robotic Systems*, vol. 14, no. 5, 2017.

[14] M. Gschwandtner, R. Kwitt, A. Uhl, and W. Pree, "BlenSor: Blender sensor simulation toolbox," in *Proc. International Symposium on Visual Computing, Advances in Visual Computing*, Las Vegas, USA, 2011, pp. 199–208.

[15] K. Koenig and A. Howard, "Design and use paradigms for Gazebo, an open-source multi-robot simulator," in *Proc. IEEE-RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, 2004, pp. 2149–2154.

[16] J. L. Martínez, M. Morán, J. Morales, A. J. Reina, and M. Zafra, "Field navigation using fuzzy elevation maps built with local 3D laser scans," *Applied Sciences*, vol. 8, no. 397, pp. 1–18, 2018.

[17] J. L. Martínez, J. Morales, A. J. Reina, A. Mandow, A. Pequeño-Boter, and A. García-Cerezo, "Construction and calibration of a low-cost 3D laser scanner with 360° field-of-view for mobile robots," in *Proc. IEEE International Conference on Industrial Technology*, Seville, Spain, 2015, pp. 149–154.

[18] H. Yang and X. Wang, "A case study on the performance of Gazebo with multi-core CPUs," in *Proc. 10th International Conference on Intelligent Robotics and Applications*, Wuhan, China, 2017, pp. 671–682.

[19] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "ROS: an open-source robot operating system," in *Workshop on Open Source Software, Proc. IEEE International Conference on Robotics and Automation*, Kobe, Japan, 2009, pp. 1–6.