



# A Simulated Environment for Traversability Estimation Experiments in Field Robotics Applications

Christos Sevastopoulos  
Christos.Sevastopoulos@mavs.uta.edu  
University of Texas at Arlington  
Arlington, Texas, USA

Stasinios Konstantopoulos  
konstant@iit.demokritos.gr  
NSCR 'Demokritos'  
Athens, Greece

## ABSTRACT

We present an environment for simulated experiments in field robotics, and especially in experiments on estimating the traversability of foliage and other objects that appear as obstacles but that can be overcome by the robot without circumventing them. The simulated environment is developed in the Unity real-time development platform, integrated with the ROS middleware. In the preliminary experiments presented here, we demonstrate that our environment is able to simulate the sensory input needed in order to train supervised traversability estimation models.

## CCS CONCEPTS

• **Computer systems organization** → **Robotics**; • **Software and its engineering** → **Virtual worlds software**; • **Computing methodologies** → **Supervised learning**.

## KEYWORDS

Robot perception, machine learning, outdoors robotics applications

### ACM Reference Format:

Christos Sevastopoulos and Stasinios Konstantopoulos. 2021. A Simulated Environment for Traversability Estimation Experiments in Field Robotics Applications. In *PETRA '21: Pervasive Technologies Related to Assistive Environments*, 29–July 02, 2021, Corfu, Greece. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3453892.3462214>

## 1 INTRODUCTION AND MOTIVATION

*Traversability estimation* [8] is a key technology in field robotics, as it allows robots to extract from sensory input the occupancy and cost information that is typically used by obstacle avoidance and navigation algorithms. Traversability estimation can be *appearance-based*, extracting from the visual signal terrain features such as roughness, slope, discontinuity and hardness [5] or terrain classes such as soil, grass, asphalt, vegetation [2]; Or it can be *geometry-based* extracting terrain features such as roughness, slope, and discontinuity from *digital elevation maps (DEM)* obtained from stereoscopic or depth sensors [11].

Even the more detailed appearance-based methods, however, only have broad classes of terrain (such as 'grass' or 'vegetation')

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
*PETRA '21, June 29–July 02, 2021, Corfu, Greece*

© 2021 Association for Computing Machinery.  
ACM ISBN 978-1-4503-8792-7/21/06...\$15.00  
<https://doi.org/10.1145/3453892.3462214>

and the features they extract cannot make the distinction between vegetation or foliage that a robotic platform can safely push away and harder obstacles that the platform should circumvent. Methods that use deep vision to directly estimate traversability from raw images [1] can be trained to make the distinctions that are appropriate for each platform, but require training data specifically acquired and annotated for each platform. Earlier experiments with a physical platform [9] have shown that general-purpose deep vision, trained on broad classes, can be made environment and platform-specific with a dramatically smaller training set, but this still remains a considerable overhead to porting deep traversability estimators for each new platform.

The simulated environment presented in this poster aims to provide a method for preparing a training set for most of this deep network adaptation. The goal is to use a simulation-generated training set for most or all of the data needed to achieve the transfer of knowledge from the original network to the platform-specific network.

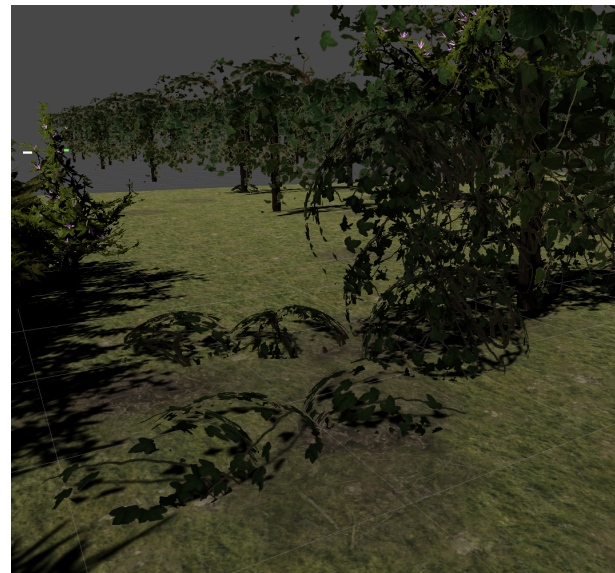
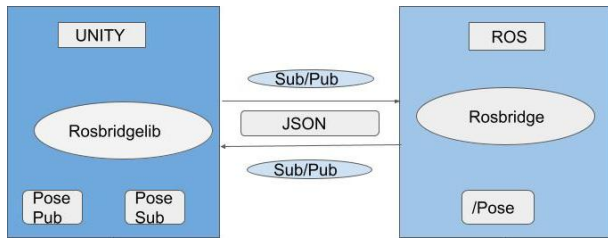


Figure 1: Characteristic scene from our simulated traversability experimentation environment, showing both 'soft' potential collisions (light foliage) as well as 'hard' potential collisions (tree trunks).



**Figure 2: Bridging between Unity and ROS is achieved by using the Rosbridgelib of the ROS# package to publish and subscribe to JSON messages on the Unity side; and Rosbridge to publish and subscribe to JSON messages on the ROS side.**

## 2 SIMULATED ENVIRONMENT

Unity<sup>1</sup> is a cross-platform 3D game engine integrated with a variety of plugins for both simulating physics and photorealistic synthesised images. In our experimental setup we used the Nvidia PhysX 3.3 physics engine<sup>2</sup> to simulate the effect of collisions between the platform and various elements of the environment. Some of these elements are designed to easily yield without harming the platform (grass, thinner foliage) while others are rigid obstacles that should be avoided (rocks, tree trunks).

Unity renders both visual and range data from this environment, which can be used as simulated sensors for the Robot OS (ROS). Specifically, we use the Rosbridge suite<sup>3</sup> to translate between ROS messages and JSON strings at the ROS side and the ROS# library<sup>4</sup> at the Unity side. In our setup ROS is running on a Ubuntu Virtual Machine and we initiate a Rosbridge server that provides a WebSocket transport layer between Unity and the ROS software stack, so that the latter subscribes to ROS topics such as pose, odometry, camera, etc. published by ROS# (Figure 2).

In our scenario, we simulate the movement of an All-Terrain Vehicle (ATV) navigating through the fields of a vineyard while encountering both non-traversable hard obstacles and traversable soft obstacles (Figure 1). Different foliage, for example, will or will not be possible to push out of the way depending on the platform’s mechanical properties. In other words, what would appear as an obstacle to standard ROS navigation using Point Cloud data might or might not be traversable. The robot can experiment with the simulated environment to acquire traversability ground-truth datapoints from the physics simulator.

## 3 CONCLUSIONS AND FUTURE WORK

Using simulated data to fine-tune deep vision systems is not a new concept, both generally [3] and specifically in off-road navigation: Sharma et al. [10] used the MAVS simulator [6] to fine-tune the DeconvNet semantic segmentation network. MAVS provides realistic visual data that Sharma et al. proved to be able to improve DeconvNet accuracy when validated on real data, but does not integrate well with ROS software stacks. This restricts to simulated robot to data collected from predefined routes that the robot cannot

affect. As the authors note, an oversimple or too random dataset can cause negative transfer, it is therefore important to be able to acquire data based on the robot’s realistic operational scenarios.

This can be achieved by using simulators that are better integrated with robotics software, such as the Gazebo/ODE robot and physics simulator [7]. Chavez-Garcia et al. [4] used Gazebo/ODE to train a network that directly maps range data to traversability estimation. However, range data will only learn geometry-based traversability and is unable to distinguish soft and hard obstacles. Since Gazebo does not generate realistic visual data, it does not support transfer learning of deep vision networks.

The Unity/ROS framework presented here combines ROS integration with the generation of realistic visual streams, allowing simulated robots to acquire realistic visual data while moving in the environment in dynamically computed routes dictated by their operational requirements. We plan to use this environment to train deep vision systems that are able to classify the obstacles detected in point cloud data as ‘soft’ and ‘hard’ obstacles, and to map obstacles to a platform-specific traversability cost. Both the Unity/ROS integrated setup and our specific simulated environment are publicly available at <https://github.com/ChristosSev/Vineyard>

## ACKNOWLEDGMENTS

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 780265 (<http://www.esmera-project.eu>) through cascade funding to the iRTA project (<https://irta.agenso.gr>).

## REFERENCES

- [1] Shyam Prasad Adhikari, Changju Yang, Krzysztof Slot, and Hyongsuk Kim. 2018. Accurate Natural Trail Detection Using a Combination of a Deep Neural Network and Dynamic Programming. *Sensors* 18, 1 (Jan. 2018).
- [2] Anelia Angelova, Larry Matthies, Daniel Helmick, and Pietro Perona. 2007. Fast terrain classification using variable-length representation for autonomous navigation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007)*.
- [3] Konstantinos Bousmalis, Alex Irpan, and et al. 2018. Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping. In *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA 2018)*.
- [4] R. Omar Chavez-Garcia, Jérôme Guzzi, Luca M. Gambardella, and Alessandro Giusti. 2018. Learning Ground Traversability From Simulations. *IEEE Robotics and Automation Letters* 3, 3 (2018). <https://doi.org/10.1109/LRA.2018.2801794>
- [5] Ayanna Howard, Homayoun Seraji, and Edward Tunstel. 2001. A rule-based fuzzy traversability index for mobile robot navigation. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, Vol. 3. IEEE, 3067–3071.
- [6] Christopher R. Hudson, Chris Goodin, Matthew Doude, and Daniel W. Carruth. 2018. Analysis of Dual LIDAR Placement for Off-Road Autonomy Using MAVS. In *Proceedings of the 2018 World Symposium on Digital Intelligence for Systems and Machines (DISA), Kosice, Slovakia, 23–25 August 2018*.
- [7] Nathan Koenig and Andrew Howard. 2018. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2004), Sendai, Japan, 2004*. <https://doi.org/10.1109/IROS.2004.1389727>.
- [8] Dirk Langer, JK Rosenblatt, and Martial Hebert. 1994. A behavior-based system for off-road navigation. *IEEE Transactions on Robotics and Automation* 10, 6 (1994), 776–783.
- [9] Christos Sevastopoulos, Katerina Maria Oikonomou, and Stasinou Konstantopoulou. 2019. Improving Traversability Estimation through Autonomous Robot Experimentation. In *Proceedings of the 12th Intl Conference on Computer Vision System (ICVS 2019), Thessaloniki, Greece, 23–25 Sep 2019 (LNCS, Vol. 11754)*. Springer.
- [10] Suvash Sharma, John E. Ball, Bo Tang, Daniel W. Carruth, Matthew Doude, and Muhammad Aminul Islam. 2019. Semantic Segmentation with Transfer Learning for Off-Road Autonomous Driving. *Sensors* 19, 11 (2019).
- [11] Martin Wermelinger, Péter Fankhauser, Remo Diethelm, Philipp Krüsi, Roland Siegwart, and Marco Hutter. 2016. Navigation IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2016), Daejeon, South Korea, October 2016.

<sup>1</sup>Cf. <https://unity.com>

<sup>2</sup>See <https://docs.nvidia.com/gameworks/#gameworkslibrary/physx/physx.htm>

<sup>3</sup>See [http://wiki.ros.org/rosbridge\\_suite](http://wiki.ros.org/rosbridge_suite)

<sup>4</sup>See <https://github.com/siemens/ros-sharp>