Using Physics-Based M&S for Training and Testing Machine Learning Algorithms

Justin Carrillo*, Burhman Gates, Gabe Monroe, Brent Newell, and Phillip Durst

[1] U.S. Army Engineer Research and Development Center, Vicksburg, Mississippi, USA
{Justin.T.Carrillo, Burhman.Q.Gates, John.G.Monroe, Brent.S.Newell, Phillip.J.Durst}@erdc.dren.mil

Abstract. Machine learning algorithms have been used to successfully solve many complex and diverse problems especially in the domain of unmanned vehicle systems. However, machine learning algorithms require training data that contain extensive variations in specimens. These variations include variations of the sensor settings, terrain conditions, environmental conditions, and variants of the objects of interest themselves. Capturing training specimens that span these variants is time consuming, expensive, and in some cases impossible. Training data in a narrow range of variations leads to decreases in performance such as overfitting. Therefore, collecting training data is often the limiting factor in developing robust machine learning algorithms for applications such as object detection and classification. Another time-consuming task is labeling training specimens with metadata needed in some training approaches. In this paper, we demonstrate using a physics-based modeling and simulation (M&S) capability to generate simulated training data spanning variations in sensor settings, terrain conditions, and environmental conditions that include a versatile automated labeling process. The product of prior efforts, the Virtual Autonomous Navigation Environment (VANE), is a high-fidelity physics-based M&S tool for simulating sensors commonly used on unmanned ground vehicles (UGVs) and unmanned aerial vehicles (UAVs).

Keywords. VANE, Modeling, Simulation, UGV, UAV, High-Fidelity, Physics-Based.

1 Introduction

1.1 Robotic Simulation Tools

There are several robotic/sensor simulation tools currently available for industry, academia, and the Department of Defense (DoD). These simulation tools include Gazebo, Army Research Laboratory's (ARL's) Robotic Interactive Visualization and Experimentation Technology (RIVET), U.S. Army Engineer Research and Development Center's (ERDC's) Autonomous Navigation Environment Laboratory (ANVEL), Syncity, Synchrono, PBRT, ERDC's Computational Test Bed (CTB), and the Virtual Autonomous Navigation Environment (VANE). The most obvious difference in these tools is the level of detail and fidelity in the description of the environment and the impact of the environment on sensor performance. This paper describes and utilizes VANE, but this section briefly describes other robotic simulation tools that are currently available and the significant differences between them and VANE. No one tool is capable of meeting all the requirements for researchers in every technology area. Recommending particular tools for each domain is difficult because a recommendation will depend on the fidelity requirements, whether there is a real-time constrain, and what level of system modeling must be supported [1]. Gazebo, RIVET, ANVEL, and Syncity take a real-time approach that utilizes lower-fidelity video-game technology while Synchrono, PBRT, ERDC's CTB, and ERDC's VANE use high-fidelity physics-based modeling and simulation.

Gazebo. Gazebo is a simulation tool originally developed in 2002 by Andrew Howard and Nate Koenig [2]. Gazebo is designed to create a 3D dynamic multi-robot environment aimed at recreating the complex world that would be encountered by mobile robots. Gazebo is a tool that was described in Michal and Etzkorn [3] as a Robotic Development Environment (RDE). One of the key aspects of an RDE is to support simulations so that experimentation and debugging can be done without access to actual robot hardware. Gazebo is primarily for developers that want to debug their systems and not assess the impact of the environment on that systems performance.

RIVET. RIVET [4] is a computer-based simulation system that was developed by the Army Research Laboratory to merge game-based technologies with current and next-generation robotic development. RIVET provides a highly detailed, capable environment for sensor and algorithm development, integration, and assessment using common off-the-shelf computers. Just like Gazebo, RIVET requires compromises in model fidelity to support real-time and complete system simulation; therefore, RIVET is also primarily for developers that want to debug their systems and not assess the impact of the environment on that systems performance.

ANVEL. The ANVEL [5], developed by ERDC, is an unmanned ground vehicle (UGV) simulator that uses video-game technology to provide an interactive simulation environment. The ANVEL provides real-time interaction between vehicle models, their sensors, and their environment. The ANVEL can be used to evaluate the performance of a UGV in a mission in a relevant environment, but the level of detail in the environmental physics is limited to what can be simulated in real-time on typical laptop or desktop computers. The ANVEL also supports pre- and post-processing of VANE simulations.

Syncity. Syncity[1], a commercial software, is a real-world simulator for autonomous applications. Syncity constructs different types of real-world scenarios that otherwise

[1] https://syncity.com/

would be difficult to record due to challenging and potentially hazardous conditions for the purposes of training and testing autonomous applications. It is a hyper realistic simulator generating synthetic data robust enough to train and validate algorithms for autonomous applications. Although the key features and sensors of Syncity are compelling, Syncity is a commercial software with very little to no documentation on the actual sensor and environment physics used. Syncity appears to utilize video-game technology rather than physics-based modeling and simulation.

Synchrono. Synchrono, developed by the University of Wisconsin-Madison, is a framework in which dynamic multi-agent simulations can be conducted to understand agent interplay and develop control algorithms in a safe and flexible environment. Unlike simulation tools such as Gazebo that simplify the noise models by assuming a Gaussian distribution, Synchrono's sensor modules are responsible for generating and recording data representing the data accumulated by various sensors [6]. Synchrono is built on top of project Chrono, which is a simulation platform created by Professor Tasora in 1997 and has since been used and developed by the Simulation Based Engineering Lab at the University of Wisconsin-Madison. Although Synchrono's framework is aimed towards greater fidelity for sensor-environment interactions and vehicle-terrain interactions, Synchrono is still in full development, and the sensor and environment models are still simplistic representations.

PBRT. PBRT [7] is a physics-based rendering engine that focuses exclusively on photorealistic rendering, which can be defined variously as the task of generating images that are indistinguishable from those that a camera would capture in a photograph or as the task of generating images that evoke the same response from a human observer as looking at the actual scene. Although PBRT is focused towards entertainment applications such as the movie special-effects industry, PBRT provides high-fidelity physics-based simulations of cameras. Much of the physics that are in PBRT are also in VANE, but PBRT does not simulate sensors such as GPS, LiDARs, and IMUs and is not integrated with multi-body dynamics engines for dynamic interactions.

ERDC's CTB. The ERDC developed a near-surface computational test bed (CTB) to help understand the effects of geophysical phenomena on signatures sensed by various sensors operating in the electro-magnetic (EM) spectrum [8]. The CTB produces 3-D, physics-based, high-fidelity numerical modeling simulations of the geo-environment. This suite of physics-based models in CTB include the Adaptive Hydrology (ADH) soil model, the vegetation model that computes radiative transfer in plants, and the EO/IR sensor model. This modeling capability can be used to predict and improve the performance of current and future sensor systems for surface and near-surface anomaly detection amid highly heterogeneous and complex environments.

1.2 Virtual Autonomous Navigation Environment

VANE is a simulation software for predicting the performance of unmanned and autonomous ground and aerial vehicles. VANE integrates high-fidelity, physics-based sensor simulations with realistic vehicle dynamics and terrain and environment simulations to provide a complete picture of the factors influencing the performance of unmanned systems. As discussed in the previous section, while several robotics simulators exist, most lack the physics fidelity to accurately capture the influence of environmental conditions on the performance of the mobility platform and the sensors that enable autonomous operations [9]. VANE uses the most realistic physics simulations for the physical processes impacting the robot, ensuring that the simulation is both realistic and predictive. Although VANE utilizes high-performance computing resources, a major misconception about VANE is that high-performance computing resources are required for using VANE. VANE simulations have been successfully run on PCs (Windows, Mac, and Linux) and workstations as well as supercomputers.

Sensors in VANE include cameras, LiDAR, GPS, and IMUs. VANE can be used to simulate digital cameras in the optical and near-infrared wavelengths that use CCD or CMOS sensors at the focal plane array. Non-ideal lensing systems can be simulated as well as electrical properties such as the gain and gamma compression of the sensor. For systems in the optical and near-infrared wavelengths, the properties of the sensing system can be modeling using radial and tangential distortion coefficients. These features allow VANE to replicate distortions exhibited in real cameras. There are several important sources of error in LiDAR measurements. In time-of-flight (TOF) LiDAR systems, which are the predominant type used on outdoor robots, there are intrinsic errors introduced by the accuracy and precision of the time system and electronics. For well-calibrated commercial systems, these errors are typically on the order of a few centimeters and can be modeled as Gaussian noise. However, errors introduced by environmental conditions, including target reflectance and geometry, weather, and beam scattering, can be significantly higher. An example of results from the camera and LiDAR sensors are shown in Figure 1.

Figure 1. VANE Simulated Camera Image (Left); VANE Simulated LiDAR Point Cloud (Right).

Because the 3D representation of an environment must have detail and accuracy commensurate with the simulation goals and because VANE can leverage HPC assets, very large and detailed scenes and environments are used for VANE simulations. Additionally, dynamic actors and animations like humans and vehicles can be scripted as input into VANE in order to evaluate dynamic interactions and sensor responses and algorithms. Snow and dust are simulated in VANE using a particle system, while rain is simulated using a random mask generator. Both methods interact using physics-based models, resulting in realistic signatures for both the camera and LiDAR sensors. An example of rain being simulated in VANE is shown in Figure 2. For LiDAR sensors, predicting the influence of dust and smoke is more complex. This is because the LiDAR acts as both the source and receiver, and the two-way radiative transfer must be calculated. VANE employs an empirical model based on field and laboratory measurements for calculating the probability of LiDAR returns from dust [11]. More details on VANE can be found in [9].

Figure 2. VANE Rain Simulations (left to right: light, medium, heavy).

2 Using Simulation for Machine Learning Support

Need for Simulation. Machine learning tools have been used for a variety of problems such as assessment of planetary gear health, assessment of fatigue and cracking in vibrating load tests, automated 3-D tissue/organ segmentation from CT scans for soldier protection, armor mechanics problems, automated optical, thermal, and acoustic monitoring of the additive manufacturing process, automated first-pass analysis of video streaming data, and evaluation of human-annotated maintenance reports toward sensor-based anomaly detection in vehicles [12]. The current and future operational applications include military intelligence, natural language processing, data mining, anomaly detection, automated target recognition, robotics, self-healing, ethics, cybersecurity, prognostic and structural health monitoring, sequence mining, and medical diagnosis [12]. Military operations are expected to have higher than usual density of static and dynamic objects in a chaotic environment. Capturing training specimens that span these variants is time consuming, expensive, and in some cases impossible. Training data in a narrow range of variation lead to decreases in performance such as overfitting. Therefore, collecting training data is often the limiting factor in developing robust machine learning algorithms for applications such as perception applications. Combining simulation data with field-collected data provides a robust data set for training, testing, and validating machine learning algorithms for autonomous applications. VANE provides diverse high-fidelity physics-based datasets that maximize machine learning by including high variation and randomization within the simulation to increase entropy within the dataset.

Machine Learning Support. Recently, VANE was upgraded to provide simultaneous ground truth within the sensor data for rapid training of machine learning-based perception algorithms. Developing labeled training specimens with certain meta-data can be time-consuming, limited in accuracy, and sometimes impossible in real-world data collects but requires little effort and is highly accurate using simulation tools such as VANE. VANE can provide metadata for the sensor, sensor position, target of interest, and environmental conditions. An example of a labeling-file with metadata for a training specimen is shown in Figure 3. VANE is able to provide information such as altitude, sensor orientation, bounding box of target within the image, ground-sampling-distance (GSD), target-sampling distance (TSD), pixels-on-target (PoT), and environment conditions such as visibility and rainfall rate.


```
{
    "Version": 2.0
    },

    "Sensor": {
        "source": "SIMULATION",
        "type": "IMAGER",
        "parameters": {
            "mode": "RGB",
            "gamma": 0.75
        },

        "pixels": [4608,3456],
        "focal_length (m)": 0.00397
    },

    "Sensor Position": {
        "height_above_ground (m)": 43,
        "LookFrom (scene)" : [62.2128,52.5503,54.4752],
        "LookTo" : [0,0,-1],
        "LookUp" : [-0.207912,-0.978148,0]
    },

    "Objects": [
        {
            "name" : "HMMWV.obj",
            "bounding_box" : [2136,1637,2471,1819],
            "average TSD (width)  (m/pixel)" : 0.0166191,
            "average TSD (height) (m/pixel)" : 0.018183,
            "average TSD (m/pixel)" : 0.0173999,
            "average TSD (viewing plane) (m/pixel)" : 0.0140944,
            "average GSD (width) (m/pixel)" : 0.0332608,
            "average GSD (height) (m/pixel)" : 0.0373623,
            "average GSD (m/pixel)" : 0.0353115,
            "average GSD (viewing plane) (m/pixel)" : 0.0297893,
            "Pixels on Target (PoT)" : 152247,
            "Horizontal PoT" : 50566,
            "Vertical PoT" : 50413,
            "location (scene) (x,y,z)" : [62.2128,52.5503,11.4752]
        }

    ],

    "Scene": {
        "name": "CampLejeune",
        "position_orientation_matrix": [1,0,0,-90.8753, 0,1,0,88, 0,0,1,0, 0,0,0,1],
        "location_origin (lat_ddeg,lon_ddeg,elev_m)": [-90.8753,88,0]
    },

    "Environment": {
        "Sun Position": [-0.503217,0.860509,0.0793477],
        "Sun theta" : 1.49137,
        "time_zone": -5,
        "datetime": [2014,249,23,0,0],
        "turbidity": 8,
        "albedo": 0.5,
        "temperature (Celsius)": 30,
        "pressure (hPA/mbars)": 1006,
        "water_vapor (cm)": 5,
        "tau500": 0.4,
        "rainfall_rate (mm/h)": 25.4,
        "rainfall_density (num/m^3)": 350.422,
        "drop_size (m - diameter)": 0.00147937,
        "rain_velocity (m/sec)": -5.37981,
        "wind_speed (m/sec)": [0,0,0]
    }
}
```


Figure 3. JSON file containing image metadata.

For many applications, detection and classification of multiple objects and their relation to one another is required. For example, determining mobility obstacles requires detecting and classifying objects that could possibly be obstacles and then relating their relative positions onto road networks. VANE provides ground truth data such as road segmentation and object location on a per-pixel basis as shown in Figure 4. The image to the left is from a UAV at 247 meters above the ground. The image to the right shows the ground truth data for the per-pixel labeling of the image. The white pixels represent the road segmentation and the gray pixels represent a HMMWV.

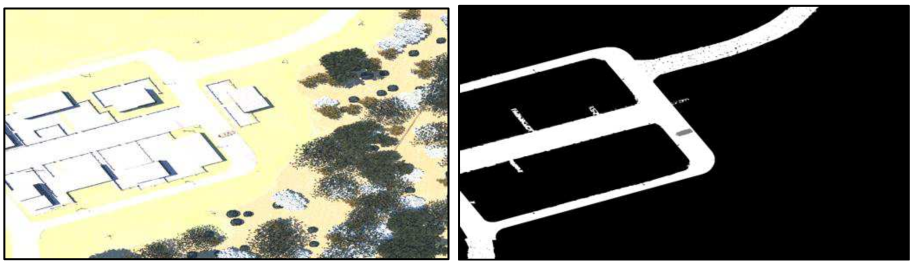

Figure 4. VANE generated image (left). Per pixel labeling of image (right).

3 Simulation of UGV and UAS systems

UGV Systems. To highlight VANE's unique capability to support machine learning through simulated training data and testing environments, VANE was used for training and testing a convolutional neural network for UGV systems [13]. In this study, high-fidelity vehicle dynamics using ERDC's Computational Research and Engineering Acquisition Tools and Environments – Ground Vehicles (CREATE^{TM} – GV) was coupled with VANE's high-fidelity sensor simulation to simulate a camera mounted on a vehicle. The objective of the study was to evaluate a convolutional neural networks ability to detect and classify a HEMTT vehicle from a camera mounted on a HMMWV. As the machine learning algorithm, the study implemented the Inception-v3 CNN using TensorFlow. VANE was used to generate the simulated training and testing data, and physical experimental data were collected for both the training and testing scenarios. Receiver operating characteristics (ROCs) curves and precision-recall curves were used from comparing the various CNN testing conditions. Highlights from the work are shown in Figure 5.

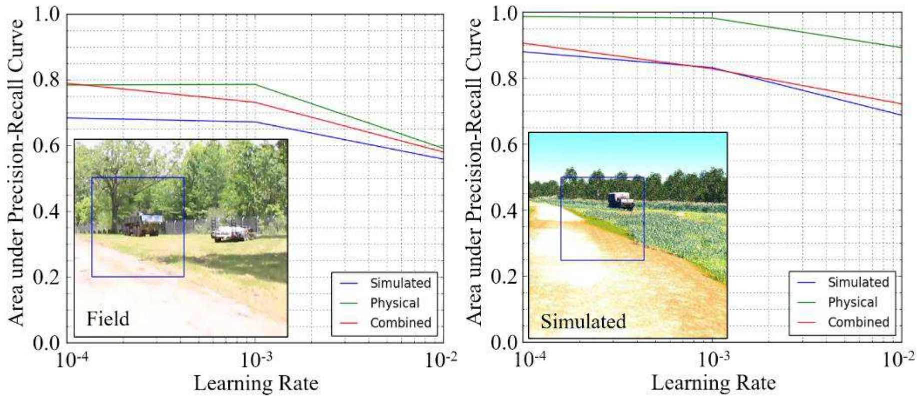

Figure 5. UGV study results from three CNNs being tested on real (left) and simulated images (right).

Although the study focused more towards comparing classifier performance based on training data and the ability of simulation to test and evaluate classifier performance, additional analysis such as observing and evaluating classifier performance with respect to rain fall rate and time of day can be utilized using VANE. Figure 6 and Figure 7 show the impact of rainfall rate and time of day on the output (*i.e.* classifier confidence) of same machine learning algorithm that was used in the study.

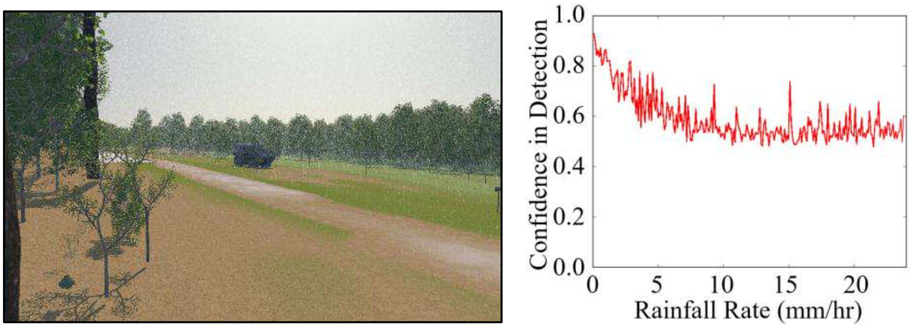

Figure 6. VANE image with rain (left). Rain impact on classifier (right).

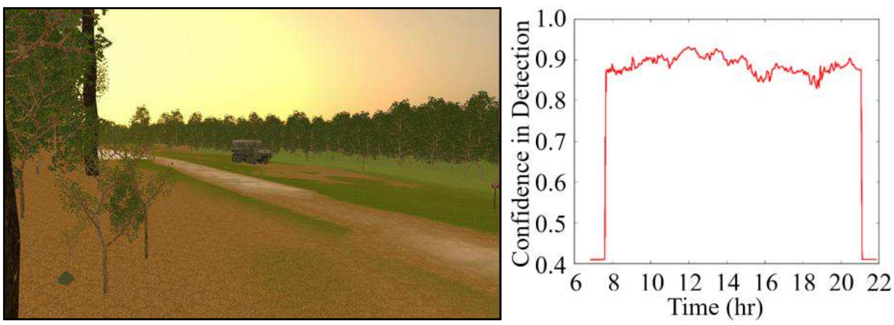

Figure 7. VANE image early morning (left). Time of day impact on classifier (right)

UAS Systems. Machine learning algorithms can be used to detect objects, recognize specific objects, and determine the probable class of unknown objects. The term "recognition" is somewhat loosely defined in common usage [14]. One hierarchical ordering of perceptual processes include detection, orientation, clutter rejection, classification, recognition, identification (friend-or-foe), identification, discrimination, and intent discrimination [14]. Relations are quite often built between these hierarchical levels and the number of pixels required. VANE provides ground truth data such as the number of successive horizontal pixels on a target (HPoT), the number of successive vertical pixels on a target (VPoT), and the total number of pixels on a target (PoT). Machine learning algorithms can be evaluated to determine the minimum number of pixels required for each hierarchical level which in turn can provide flight requirements for UAVs. For example, there is a relationship between the number of pixels on an object and the target sampling distance (TSD) in the viewing plane. The TSD is a function of flight height and sensor settings. An estimate derived from VANE is provided in Equation 1, where S_w is the sensor width (mm), S_h is the sensor height (mm), h is the flight height (m), f is the focal length (mm), I_w is the image width (pixels), and I_h is the image height (pixels). Figure 8 and Figure 9 shows verification results of Equation 1 using two different camera sensors. The camera parameters are shown in Table 1.

$$\begin{aligned} PoT_{HMMWV} &= g(\,f(\,flight\ height, sensor\ settings), target) \\ &= 32.383\left(\frac{h}{2f}\left(\frac{S_w}{I_W}+\frac{S_h}{I_h}\right)\right)^{-2.0}, 50 \le h \le 300 \end{aligned} \qquad \text{Eqn. 1}$$

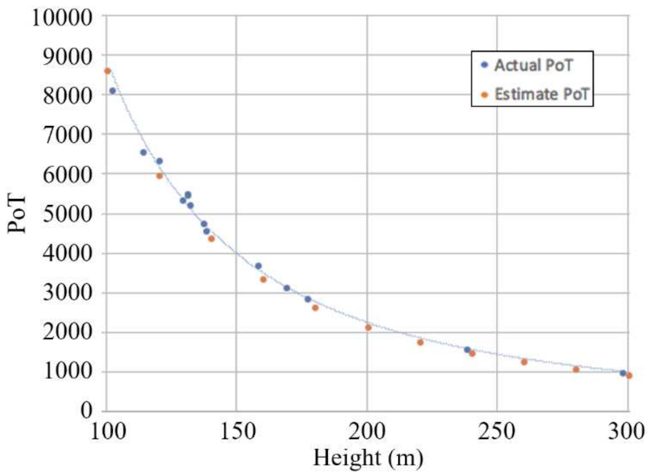

Figure 8. Pixels on target with respect to height for HMMWV using Ideal camera.

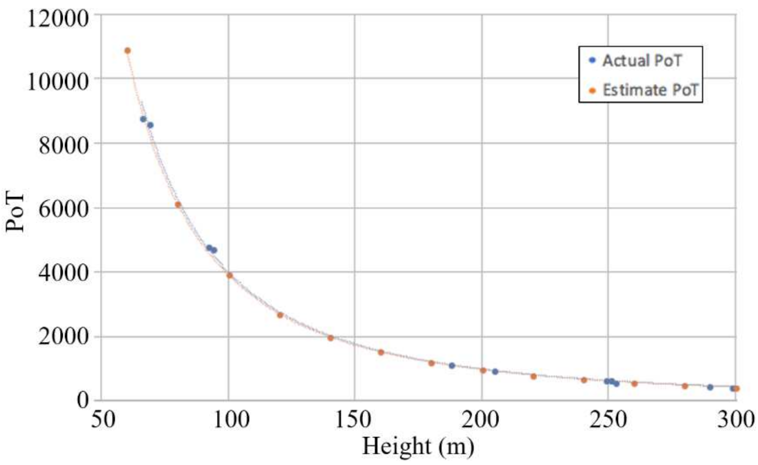

Figure 9. Pixels on target with respect to height for HMMWV using Canon camera.

Table 1. Camera Parameters

Parameters	Ideal Camera	Canon Camera
Resolution (Pixels)	1920 x 1080	1920 x 1080
Sensor Size (mm)	(1.894, 1.894)	(22.3, 14.9)
Focal Length (mm)	3.5	18
Gamma	0.75	1.20
Gain	10.0	8.0

4 Conclusions

Machine learning algorithms have been used to successfully solve many complex and diverse problems especially in the domain of unmanned vehicle systems. Machine learning algorithms require training data that contain extensive variations in specimens. In this paper, we presented a simulation tools that provide variations in sensor settings, terrain conditions, and environmental conditions. VANE can be used to generate training specimens that span these variants efficiently and without the cost of collecting the data in real-world scenarios. VANE is completely physics-based, which allows for the closest representation of the real world in the training specimens. Combining simulated data with real-world data provides a dataset that has the entropy necessary for developing robust machine learning algorithms. Two examples were provided for using VANE's simulated training data for UGV and UAV systems that also take advantage of a versatile automated labeling process.

References

1. Fields, M., Brewer, R., Harris, E., Pusey, J., Weller, E., Patel, D., and DiBerardino, C. (2016). Simulation tools for robotics research and assessment. *Proceedings of the SPIE*, Volume 9837

2. Koenig, N., and Howard, A. (2004). Design and Use Paradigms for Gazebo, and Open-Source Multi-Robot Simulator. *IROS*. 3, pp. 2149-2154. USA:IEEE

3. Michal, D.S., and Etzkorn, L., (2011). A Comparison of Player/Stage/Gazebo and Microsoft Robotics Developer Studio. *Proceedings of the 49th Annual Southeast Regional Conference* (pp. 60-66). USA:ACM.

4. Brewer, R., Schaefer, K.E., and Avery, E. (2015). Robotic Interactive Visualization Experimentation Technology (RIVET): Game-based simulation for human-robot interaction research. *2015 Winter Simulation Conference (WSC)* (pp. 3224-3225)

5. Durst, P. et al. (2012). A Real-Time, Interactive Simulation Environment for Unmanned Ground Vehicles: The Autonomous Navigation Virtual Environment Laboratory (ANVEL). *2012 Fifth International Conference on Information and Computing Science.* IEEE.

6. Elmquist, A., and Negrut, D. (2017). Virtual Sensing for Autonomous Vehicle Simulation in Chrono. *Simulation-Based Engineering Lab University of Wisconsin-Madison Technical Report* TR-2016-13

7. Pharr, M., Jakob, W., and Humphreys, G. (2017). Physically Based Rendering: From Theory to Implementation. *Textbook.* Third Edition

8. Nagaraju, K., Howington, S., Goodson, R., Hines, A., Bray, M., Price, S., Galan-Comas, G., and Ballard, J. (2016). High-Fidelity Modeling to Support Route Clearance. *North Atlantic Treaty Organization (NATO) Science and Technology Organization (STO)* STO-MP-MSG-143

9. Goodin, C., Carrillo, J., McInnis, D., Cummins, C., Durst, P., Gates, B., and Newell, B. (2017) Unmanned Ground Vehicle Simulation with the Virtual Autonomous Navigation Environment. *International Conference on Military Technologies*

10. Durst, P., and Goodin, C. (2012). High Fidelity Modelling and Simulation of Inertial Sensors Commonly used by Autonomous Mobile Robots. *World Journal of Modelling and Simulation,* (pp. 172-184).

11. Goodin, C., Durst, P., Prevost, Z., and Compton, P. (2013). A Probabilistic Model for Simulating the Effect of Airborne Dust on Ground-Based LiDAR. *SPIE Defense, Security, and Sensing*. International Society for Optics and Photonics.

12. Lee, M., Valisetty, R., Breuer, A., Kirt, K., Panneton, B., and Brown, S. (2018). Current and Future Applications of Machine Learning for the US Army. *US Army Research Laboratory Technical Report* ARL-TR-8345

13. Carrillo, J., Davis, J., Osorio, J., Goodin, C., and Durst, J. (in review). High-Fidelity Physics-Based Modeling and Simulation for Training and Testing Convolutional Neural Networks for UGV Systems. *Journal of Unmanned Vehicle Systems*

14. Harney, R. (2004). Combat Systems. Volume 1. Sensor Elements. Part I. Sensor Functional Characteristics. *Textbook*