

## Research Article

# The Need for High-Fidelity Robotics Sensor Models

**Phillip J. Durst,<sup>1</sup> Christopher Goodin,<sup>1</sup> Burhman Q. Gates,<sup>1</sup>  
Christopher L. Cummins,<sup>1</sup> Burney McKinley,<sup>1</sup> Jody D. Priddy,<sup>1</sup>  
Peter Rander,<sup>2</sup> and Brett Browning<sup>2</sup>**

<sup>1</sup>*Mobility Systems Branch, Geotechnical and Structures Laboratory, US Army Engineer Research and Development Center, 3909 Halls Ferry Road, Vicksburg, MS 39180, USA*

<sup>2</sup>*National Robotics Engineer Center, Carnegie Mellon University, Ten 40th street, Pittsburgh, PA 15201, USA*

Correspondence should be addressed to Phillip J. Durst, [phillip.j.durst@usace.army.mil](mailto:phillip.j.durst@usace.army.mil)

Received 11 January 2011; Accepted 7 September 2011

Academic Editor: Lyle N. Long

Copyright © 2011 Phillip J. Durst et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Simulations provide a safe, controlled setting for testing and are therefore ideal for rapidly developing and testing autonomous mobile robot behaviors. However, algorithms for mobile robots are notorious for transitioning poorly from simulations to fielded platforms. The difficulty can in part be attributed to the use of simplistic sensor models that do not recreate important phenomena that affect autonomous navigation. The differences between the output of simple sensor models and true sensors are highlighted using results from a field test exercise with the National Robotics Engineering Center's Crusher vehicle. The Crusher was manually driven through an area consisting of a mix of small vegetation, rocks, and hay bales. LIDAR sensor data was collected along the path traveled and used to construct a model of the area. LIDAR data were simulated using a simple point-intersection model for a second, independent path. Cost maps were generated by the Crusher autonomy system using both the real-world and simulated sensor data. The comparison of these cost maps shows consistencies on most solid, large geometry surfaces such as the ground, but discrepancies around vegetation indicate that higher fidelity models are required to truly capture the complex interactions of the sensors with complex objects.

## 1. Introduction

As sensors are the medium through which mobile robots observe their environments, it seems only intuitive that mobile robot simulations should strive to model sensors to the highest level of fidelity. However, modeling sensors using first principles is a difficult and time-intensive task, so most simulations make use of simplified, and often idealized, methods to reproduce sensor outputs. While some effort has been given to quantifying the gap between simulation and reality for mobile robots, no work has been done specifically to address the shortcomings of these simplified sensor models. Furthermore, much of the research that does exist is outdated, with very little recent research available addressing the issue. In this paper, we focus on the most popular of UGV sensors for off-road environments, light detection and ranging (LIDAR) sensors. Section 2 provides an overview of the published work related to the gap between simulations and field experiments.

A survey of mobile robotics today shows that LIDAR sensors are one of the most popular solutions to the autonomous navigation problem. Most simulations make use of surface models (e.g., a triangle mesh) to represent geometry, and then use either a simple point-intersection model [1] or a Z-buffer technique [2] for generating LIDAR outputs. In the case of the point-intersection model, ray casting is used to project an infinitesimally thin line from the sensor into the scene, and the facet within the scene that the line intersects is taken to be the output range of the LIDAR sensor. The Z-buffer technique works in the opposite direction, determining the closest facet within the field of view of each pixel. Both of these approaches represent the LIDAR in its idealized form, always detecting the presence of any shape, without any noise. In some cases, this range is corrupted using empirically derived noise (typically Gaussian) before being output to the autonomy system [3]. However, this additive noise accounts for the stochasticity of real data in a simplistic way and does not fully capture the true statistics, which

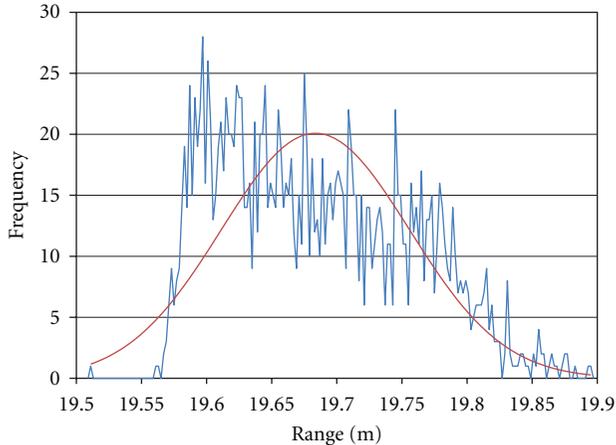


FIGURE 1: The range histogram showing the variation in range returns for vegetation and a fitted Gaussian distribution. Data was collected from a stationary velodyne aimed at a small bush and extracting the ranges from a single laser for a specific angular reading.

are scene dependent and may show significant intrascan correlations. Figure 1 shows an example histogram of range returns for a single laser (from a velodyne) aimed at a small bush. Unlike smooth planes, where range estimates have Gaussian-like noise, this histogram is clearly non-Gaussian. The resulting distribution is much more complex due to the many boundaries present in the scene [4].

To show the shortcomings of the point-intersection LIDAR model, which is considered the more accurate of the two, a point-intersection LIDAR model was created for the presented study, and details about this model are given in Section 3.

To provide evidence supporting this assertion, data from an experiment with the National Robotics Engineering Center’s Crusher unmanned ground vehicle (UGV) are presented [5]. The Crusher collected LIDAR data from a short pass through a simple outdoor scene containing some vegetation, three hay bales, and a boulder. The same scene was generated in simulation, and the LIDAR sensor model was simulated moving along the same path as the Crusher’s LIDAR sensors. The accuracy of the simulated LIDAR was tested by observing its effect on the Crusher’s autonomy system.

One common approach to autonomous navigation, as used on the Crusher system [5], is to plan optimal paths that are efficient and safe trajectories across the ground surface. A cost map is used to represent mobility cost for the terrain, where higher cost represents areas that present a safety risk to the vehicle (e.g., a protruding rock or a ditch). Crusher’s autonomy system builds a mobility cost map from sensor data by reasoning about positive obstacles (vegetation, rocks, etc.) and the ground surface. The system explicitly reasons about vegetation, which often appears “porous” with LIDAR measurements, and estimates hidden ground surfaces that may be occluded.

Using the resulting cost map, the vehicle moves along an optimal path, which can be found using path planning algorithms like those in [6]. Therefore, the cost map provides a convenient way to evaluate the impact of simulated sensor

data on the autonomy system. We can use the autonomy system with real sensor data captured in the field to build a cost map and compare it to the same autonomy system running on simulated sensor data from a model of the same area. If the two cost maps are identical, or nearly so, then the simulated sensor outputs can be trusted for use in simulations of the Crusher UGV. Section 4 provides such a comparison and shows that for complex and outdoor environments simple LIDAR models are only marginally useful for developing autonomy algorithms for mobile robots.

## 2. Related Work

A search of the literature reveals that the gap between autonomous robot simulations and experiments has received little consideration. There are a few sources, though, that document the failure of simple simulations to accurately predict robot behaviors. In their research with Nomad 200 and sonar sensors, Lee et al. found that “simulation results can only be transferred to real robot controls in very simple cases” [7]. A similar conclusion was reached by Brooks, who found that “programs which work well on simulated robots will completely fail on real robots because of the differences in real-world sensing and actuation” [8]. Gat [9] compared simulation and field tests of a Rocky 3.2 mobile robot operating at low speeds in a small, simple environment. The robot’s behavior in simulation was nearly identical to its behavior in the field. This agreement is likely due to the simplicity of the scene, and it held when robot failures, which the simulator could not accurately model, were ignored.

More recent research by Nehmzow [10, 11] delves more deeply into the disparity between simulations and experiments. His research found that “a fundamental difference exists between a physical mobile robot and its simulation.” Through a quantitative analysis of the divergence between mobile robots and their simulations, he concluded that “a simple generic model of robot sensors and actuators is insufficient to model a mobile robot faithfully.”

Of course, most of these papers describe research that is over a decade old. During that time, almost no research has been published specifically addressing the simulation-reality gap.

With the rapid advancement in desktop computing power, robotics simulators have become much more accurate and robust. The simplified approach used by today’s simulations often return results that are accurate to within allowable error for environments with large solid surfaces when compared to LIDAR beam width, such as with indoor environments. For example, the work in [12] quantitatively compared simulation and real-world results with favorable results. However, as our experiments below reveal, for outdoor terrains with vegetation, the pitfalls remain. We argue that this is primarily due to the presence of vegetation, which with its comparatively small geometries (e.g., leaves, grass) results in many LIDAR returns that exhibit boundary effects. We expect that when one considers the full richness of outdoor environments, where mud, water puddles, surface dew, atmospheric particulates, and other complexities are common, the simulation accuracy deteriorates further.



FIGURE 2: The test site at Ft. Drum, NY, where the data for this experiment were taken.

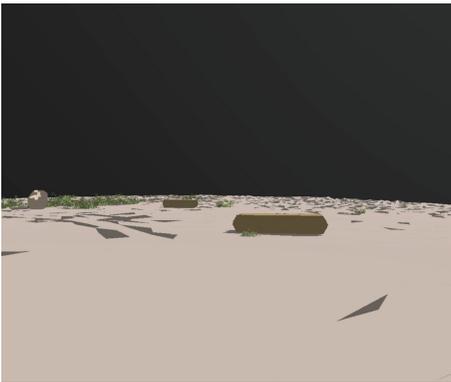


FIGURE 3: Scene capture of the simulated scene mesh corresponding to part of the area shown in Figure 1.

### 3. Crusher Experiment

**3.1. Scene Generation.** For this experiment, the Crusher made several passes across a parking apron at Fort Drum, New York. The test area (Figure 2) contained various objects, including some vegetation, three hay bales, and a large boulder. The test environment was set up to include a variety of obstacles in an attempt to include objects that would result in a wide range of costs in the navigation cost map. The Crusher was driven, via teleoperation, to the center of the test site from six different starting positions, and LIDAR data were used to generate cost maps. LIDAR data collected by the Crusher's four forward-facing SICK LMS LIDAR sensors from five of the six test runs were used to create the simulation scene. The LIDAR point cloud data from these passes were used to create a triangular mesh, an example of which is shown in Figure 3.

The size, location, and orientation of the boulder, hay bales, and vegetation were determined through segmentation of the LIDAR data gathered by the Crusher's sensors. Representative meshes were used for the vegetation, hay bales, and boulder. These meshes were scaled and oriented using the values from the segmentation process. The vegetation models were not chosen to match the species of the real vegetation; instead, a model of a small creosote plant was substituted for vegetation higher than 18 cm. A model of grass was substituted for shorter vegetation.

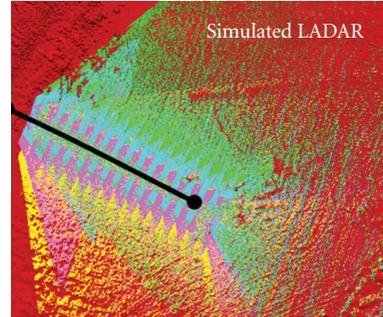


FIGURE 4: Overhead view of the simulated scene and simulated LIDAR point cloud data for the four forward-facing LIDAR sensors and the path traveled by the sensors. Points for each sensor are colored differently and the ground surface is colored in red (best viewed in color).

**3.2. LIDAR Model.** The Crusher has eight SICK LMS 291-S14 scanning LIDAR sensors [13]. The SICK LMS 291-S14 is a time of flight LIDAR, meaning it uses the time between laser beam emission and received reflection to calculate the distance to objects. For use on the Crusher, these LIDAR sensors were set to take sweeps over a 90-degree range at a resolution of 0.5 degrees for a total of 181 angle-range data points. The distance from the sensor to any objects encountered at each 0.5 degree angular spacing for each sweep is recorded on the Crusher and passed to the autonomy system. The autonomy system then uses these angle-range data to make a geometric model of the world captured as a cost map [5]. For this experiment, the four forward-facing LIDAR sensors were simulated using the point-intersection model described below.

The LIDAR model developed for this study was similar to that found in [14, 15]. It was a pencil ray point-intersection model that relied on ray tracing techniques. The advantage of this type of model and the main reason for its prevalence is that it is not computationally intensive and can run in real time. For each laser cast by the LIDAR at each 0.5 degree angular spacing, a single ray was cast from the sensor into the scene. The first intersection between each ray and a facet within the scene was recorded, and this distance was taken to be the sensor's output range. Figure 4 shows an overhead view of the simulation scene, the path traveled by the Crusher in simulation, and the LIDAR points generated by the four forward-facing LIDAR sensors using the point-intersection sensor model. Figure 5 shows the simulated LIDAR points for one scan of the two right-facing LIDAR sensors.

The obvious drawback of this model is that it does not take into account any physical effects, such as dispersion of the laser beam and possible second-order interactions between the beam and the environment. While this simplification does not have a major impact on simulation validity for small-scale, indoor robotics applications, it has not been tested for complex and outdoor environments. To test the accuracy of this simplified LIDAR sensor model for such environments, the poses of the Crusher's four forward-facing LIDAR sensors were recorded by the Crusher during each teleoperated pass through the test site. For each pose

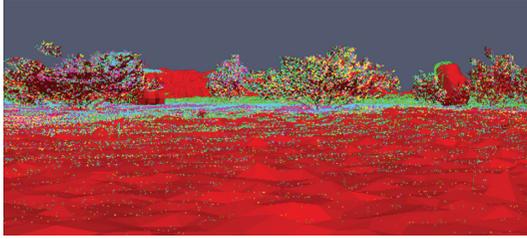


FIGURE 5: Side view of the simulated scene (see Figure 4 for color scheme details).

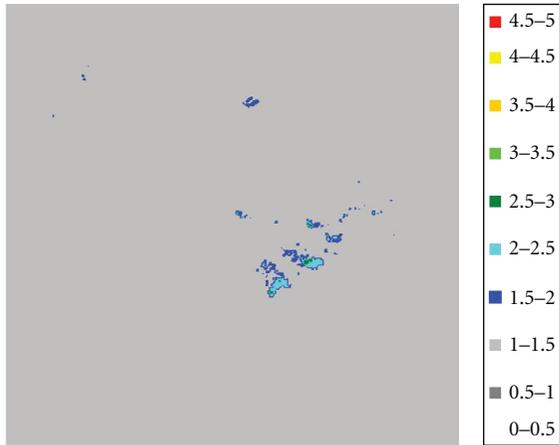


FIGURE 6: Cost map generated by the Crusher during field testing for the region shown in Figure 4. Plotted is the log of the cost associated with each  $(x, y)$  position within the scene. The large light blue areas are the high-cost regions created by ground occlusion.

logged during one of the passes, simulated LIDAR data were generated for the forward-facing sensors. The simulated LIDAR data were then used as input for the Crusher's autonomy system, and a cost map was generated. The cost map generated using the simulation data was then compared to the cost map generated by the LIDAR data collected in the field, and the results of this comparison are presented in the following section.

#### 4. Results

The LIDAR data generated in simulation was used to generate a cost map, and this simulated cost map was compared to the one generated by the Crusher during the on site testing. A comparison of these two cost maps can be used to show the relative accuracy of the simulated sensor for predicting the choices the autonomy system would have made. Figure 6 is the cost map generated by the Crusher during field testing. Higher numbers/brighter colors represent obstacles or areas of higher perceived cost. The main areas of cost can be associated with the objects placed within the scene, namely, the boulder and hay bales. However, two areas of high-cost can be seen in the area behind the vegetation. These high cost areas correspond to areas where the ground surface that were occluded by the vegetation. As such, the ground

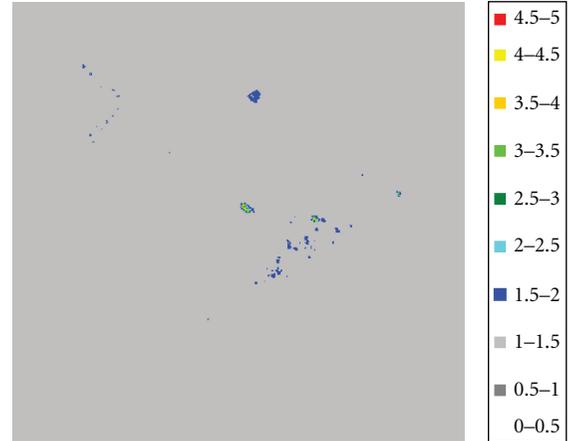


FIGURE 7: Cost map generated using the simulated LIDAR data. The cost of the vegetation is the same as the cost generated using real data, but the area of cost associated with the vegetation is smaller by roughly 75%.

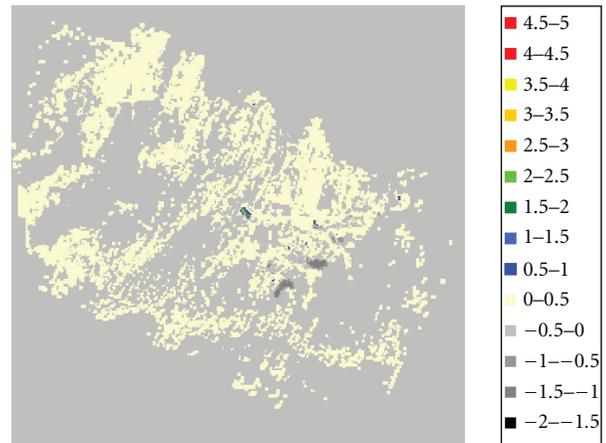


FIGURE 8: Plot showing the difference between the two cost maps, simulated cost-real cost, at each  $(x, y)$  point in the scene. The major disparity between the two cost maps in the regions behind the vegetation is clearly visible.

height is inferred, but additional cost is added to reflect the uncertainty about whether there is a potential unseen negative hazard in that location.

The Crusher's LIDAR sensors did not receive any returns from the region directly behind the vegetation. Because the laser emitted by the SICK LIDAR is not a perfect infinitesimal ray but instead is a beam with width that disperses as it propagates, none of the emitted laser beams penetrated through the vegetation to the ground behind it.

Figure 7 shows the cost map generated using the simulated LIDAR data. Again, higher numbers/brighter colors represent areas of higher cost and therefore greater potential risk to the UGV. As with the cost map generated using true LIDAR data, the areas of high cost correlate to the objects within the scene. However, the simulated cost map does not contain any areas of high cost behind the vegetation. Figure 8 shows the difference of the two cost maps, simulated cost-real

cost at each  $(x, y)$  location. As is evident there are significant differences in cost corresponding to the two “hidden” areas in the real data. The simple point-intersection LIDAR model was not able to recreate this effect and would therefore not have been able to accurately predict Crusher’s behavior in the field.

A second complicating factor is due to the challenges of modeling vegetation. As a result, the simulation model of the vegetation is not identical to the real vegetation present in the scene. These geometric differences have an obvious impact on the simulated LIDAR beams. Indeed, the challenge to accurately modeling vegetation on the large scale, suggests that an alternative approach is required and is something we will pursue in future work.

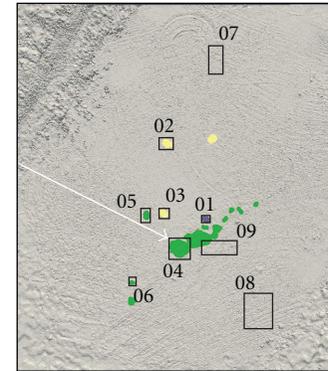
Figure 9 shows an object-to-object cost comparison between the simulated and real cost maps. For rigid objects with fixed boundaries, namely, the two hay bales and boulder, the simulation results are in good agreement with the ground truth data. For these simple objects, the point-intersection LIDAR model was adequate. On the other hand, the two maps are in very poor agreement for the area of the test site containing vegetation. These types of objects are not strictly bounded, and the point-intersection model did not accurately recreate the outputs of a true LIDAR sensor. These are areas that the Crusher would have tried to avoid in the field and would have made much less effort to avoid in the simulation. This type of simplistic LIDAR model could not be used for the development and testing of algorithms for mobile robots in outdoor environments, particularly in highly vegetated areas.

## 5. Conclusions

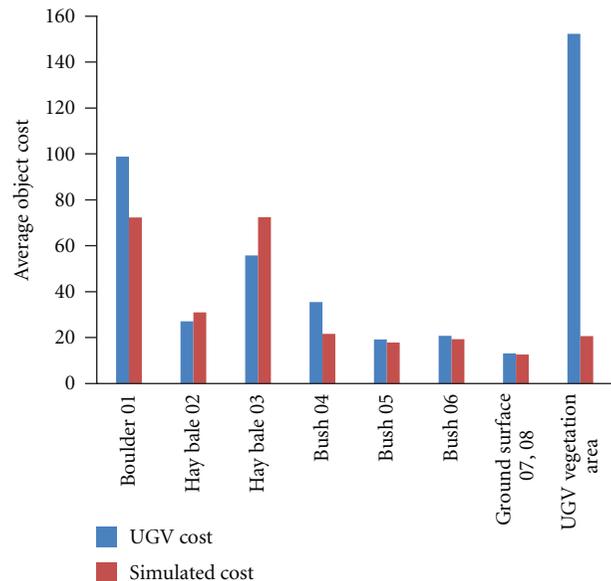
The ultimate goal of this effort was to show, through analysis of LIDAR data and autonomous navigation algorithm outputs gathered using the NREC’s Crusher UGV, the limitations of current sensor modeling methodologies for the simulation of autonomous mobile robots. The simulation environments being used currently for the development and testing of autonomy behaviors make use of empirical/probabilistic sensor models. While these sensor models may be adequate for robots designed to operate in man-made environments without vegetation, mud, and other complexities, they cannot accurately predict robot behaviors in complex natural environments.

The simple point-intersection LIDAR model did not recreate the complex beam-world interaction effects seen in the real-world data in vegetated areas. The cost map generated in simulation was similar to the cost map generated in the field for only those obstacles that were geometrically simple and had well-defined boundaries. In a more densely vegetated environment, the simulation would certainly have failed to predict the Crusher’s behavior.

As autonomous mobile robots are used in increasingly complex environments, the gap between simulation and reality will become more pronounced. As long as sensor-environment interactions are generated simplistically, simulations will remain unable to accurately predict autonomous



(a)



(b)

FIGURE 9: Comparison between simulated and actual costs associated with each object in the scene. On the left is an overhead view of the scene with each object labeled along with the path the UGV traveled. The average cost of each object is very similar between the simulated and ground truth data for the boulder and hay bales, but the area of high cost seen behind the vegetation generated by the true LIDAR data was not reproduced in simulation.

robot performance in these environments. The development of new, high-fidelity sensor models will be critical for the future development and expansion of robotics into complex, natural outdoor environments.

## Acknowledgments

Permission to publish was granted by the director of Geotechnical and Structures Laboratory. The authors would like to thank the entire NREC UPI Crusher team, including team leaders Anthony Stentz, John Bares, Tom Pilarski, and David Stager, for collecting and providing the field data and for access to the autonomy system used in the experiments described here.

## References

- [1] A. Vadlamani, M. Smearcheck, and M. U. De Haag, "Preliminary design and analysis of a lidar based obstacle detection system," in *Proceedings of the 24th Digital Avionics Systems Conference*, vol. 1, pp. 6.B.2–61-14, November 2005.
- [2] R. Telgarsky, M. C. Gates, C. Thompson, and J. N. Sanders-Reed, "High fidelity lidar simulation," in *Proceedings of the Laser Radar Technology and Applications IX*, vol. 5412 of *Proceedings of SPIE*, pp. 194–207, April 2004.
- [3] E. B. Wilson, "Real-time correlative scan matching," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4387–4393, May 2009.
- [4] J. Tuley, N. Vandapel, and M. Hebert, "Analysis and removal of artifacts in 3-D LADAR data," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '05)*, pp. 2203–2210, April 2005.
- [5] A. Stentz, J. Bares, T. Pilarski, and D. Stager, "The crusher system for autonomous navigation," in *Proceedings of the Unmanned Systems North America Conference*, pp. 972–986, August 2007.
- [6] D. Ferguson and A. Stentz, "Using interpolation to improve path planning the field D\* algorithm," *Journal of Field Robotics*, vol. 23, no. 2, pp. 79–101, 2006.
- [7] T. Lee, U. Nehmzow, and R. Hubbold, "Mobile robot simulation by means of acquired neural network models," in *Proceedings of the 12th European Simulation Multiconference*, 1998.
- [8] R. A. Brook, "Artificial life and real robots," in *Proceedings of the 1st European Conference on Artificial Life*, pp. 3–10, 1992.
- [9] E. Gat, "Towards principled experimental study of autonomous mobile robots," *Autonomous Robots*, vol. 2, no. 3, pp. 179–189, 1995.
- [10] U. Nehmzow, "Quantitative analysis of robot-environment interaction-towards "scientific mobile robotics"," *Robotics and Autonomous Systems*, vol. 44, no. 1, pp. 55–68, 2003.
- [11] U. Nehmzow, "Quantitative analysis of robot-environment interaction-on the difference between simulations and the real thing," in *Proceedings of the Eurobot*, 2001.
- [12] S. Carpin, T. Stoyanov, Y. Nevatia, M. Lewis, and J. Want, "Quantitative assessments of USARSim accuracy," in *Proceedings of the PerMIS*, 2006.
- [13] SICK, 2006, <http://sicktoolbox.sourceforge.net/docs/sick-lms-technical-description.pdf>.
- [14] S. Balakirsky, S. Carpin, G. Dimitoglou, and B. Balaguer, "From simulation to real robots with predictable results: methods and examples," in *Performance Evaluation and Benchmarking of Intelligent Systems*, pp. 113–137, Springer, New York, NY, USA, 2009.
- [15] B. Gerkey, R. Vaughan, and A. Howard, "The player/stage project: tools for multi-robot and distributed sensor systems," in *Proceedings of the International Conference on Advanced Robotics*, pp. 317–323, 2003.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

