# USARSim: a validated simulator for research in robotics and automation

Benjamin Balaguer, Stephen Balakirsky, Stefano Carpin, Mike Lewis, Christopher Scrapper

**In this paper we present the Unified System for Automation and Robotics Simulation (USARSim), a popular simulation system that can be used to study a variety of tasks related to mobile robotics. Originally designed with a focus on Urban Search and Rescue, USARSim has evolved into a general purpose simulation system that supports a variety of different robots including wheeled,, tracked, legged and flying robots. A unique element that characterizes USARSim is the extensive quantitative validation performed on its components. Based on a commercial moderately priced game engine, the USARSim extensions can be obtained for free. The system has been adopted by the Robocup federation and by the IEEE to support two robotic competitions and has also been used to support robotics education.**

## I. Introduction

Robots are becoming more and more pervasive in our everyday lives. Their use spans from high precision industrial manipulators to floor cleaning robots, from space exploration to assistance to first responders. It is envisioned that robots will play a major role in our future society and economy [8]. The spectrum of end users and working modalities is also expanding. Highly skilled professionals are no longer the only community of users, and robots are now used by people with very diverse backgrounds. In addition, the traditional distinction between purely tele-operated robots, autonomous machines operating in unknown environments, and preprogrammed devices maneuvering in well confined working cells is also no longer appropriate. The use of multiple cooperating robots is also soaring. All of these facts call for the introduction of new tools for robotics development at all levels.

In this paper we report on the development of a software simulator that allows tackling a broad range of design problems that arise while developing robot systems. The software system, called Unisfied System for Automation and Robotics Simulation (USARSim), can be used to design robot controllers, study interactions between multiple robots, assess the effectiveness of different human-robot interfaces, and also to investigate the relation between robot morphologies and performance. USARSim is a practical tool to implement the principle of *early-testing late-binding* in robot design. It has been built as a plug-in module on top of a commercial game engine and therefore exploits highly realistic visual rendering, as well as high performance physics simulation. The software module is distributed using an open-source model [14]. This means that in order to get started one needs to acquire a license of the game engine, currently priced below $40 US, and then to download from the sourceforge website the USARSim module to be installed on top of the game engine. At the time of writing this paper, more than 30000 component downloads were recorded. This paper is organized as follows. Section II provides a short recap of USARSim's evolution. The design principles that guided its development are outlined in section III. Next, in section IV we sketch the validation methodology developed together with USARSim, and in section V we report about the national and international robotic competitions based on USARSim. The paper finishes with a discussion about the open challenges of the project in section VI and finally conclusions are offered in section VII.

## II. History and evolution of USARSim

USARSim development started in the fall of 2002 at Carnegie Mellon University (CMU) and the University of Pittsburgh. Since its early inception, the software was developed targeting Urban Search And Rescue applications, hence its original acronym, USARSim. The first release of USARSim was built upon the Unreal Engine 2[1] game engine, i.e. the software base behind the popular game Unreal Tournament and could only be controlled through RETSINA multiagent system software. At its dawn, USARSim included only models for a few differential drive robotic platforms, a restricted set of sensors, and the models of three environments; namely the rescue test arenas (see )developed at the National Institute of Standards and Technology (NIST). In 2005, following the endorsement of the Robocup federation (see section V), USARSim management was taken over by NIST, who moved the project to sourceforge, and reorganized the code base while maintaining most of the original structure designed at CMU and Pittsburgh. Version 1.0 was released in October 2005. The transition to sourceforge and the involvement of the Robocup community brought an explosive development in USARSim. Version 3.31, released in July 2008 offers 15 different sensors, from odometry to an omnidirectional camera. In addition, 23 different robotic platforms are now

Benjamin Balaguer and Stefano Carpin are with the University of California, Merced, CA, USA

Stephen Balakirksy and Christopher Scrapper are with the National Institute of Standards and Technology, Gaithersburg, MD, USA

Mike Lewis is with the University of Pittsburgh, PA, USA

[1] Certain commercial software and tools are identified in this paper in order to explain our research. Such identification does not imply recommendation or endorsement by the authors, nor does it imply that the software tools identified are necessarily the best available for the purpose.

Figure 1 One of the first three test environments developed for USARSim, namely the NIST *orange arena*.

available; these include wheeled robots, cars, tracked vehicles and flying robots. With a growing community of users, urban search and rescue applications are no longer the only target for USARSim. In order to mirror the broader range of use, in 2007 USARSim's name was changed to Unified System for Automation and Robotics Simulation.

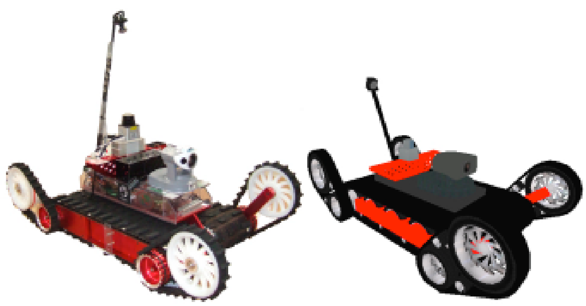

Figure 2 On the left, the Kenaf robot, a USAR platform developed at the Tohoku University, Japan. On the right, the simulated model inside USARSim. The model was contributed by the same group that developed the real robot.

Figure 2 and Figure 3 illustrate two of the latest additions to USARSim, namely the Kenaf robot and the omnidirectional camera. The reader is referred to the official USARSim manual available on [14] for a comprehensive list.

## III. USARSIM'S DESIGN PRINCIPLES

In this section we present USARSim's design principles. Technical details concerning its implementation issues have been published already and will not be discussed here. The reader is referred for example to [6] and the references therein.

USARSim was initially developed as a tool for studying human-robot interaction (HRI) for USAR tasks. The critical features for HRI oriented simulation are that it must accurately reflect the range of available information, behavior, and user experience encountered in actual robot operation. This means an HRI simulation must supply both sufficient perceptual fidelity to make the operator realistically aware of the remote environment and sufficiently accurate modeling of the robot and its automation to relate robot behavior to that same environment.

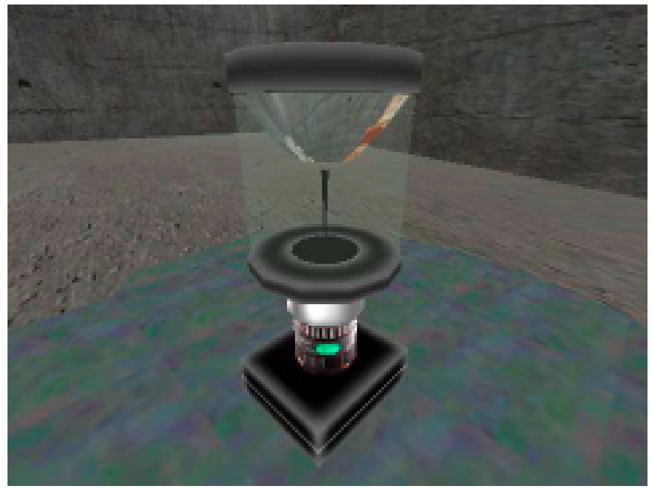

Figure 3 One of the latest additions to the set of USARSim sensors, the omnidirectional camera. The model was developed and contributed by the University of Amsterdam.

Most earlier robotic simulations were built as ancillary tools for developing and testing control programs to be run on research robots. Simulators built before 2000, including [11] and [12] had low fidelity dynamics for approximating the robot's interaction with its environment. These simulators provided a simplified abstract environment for controlled testing of algorithms. More recent simulators like for example ÜberSim [20], a soccer simulator, and Gazebo [10], use the open source Open Dynamics Engine (ODE) physics engine to approximate physics and kinematics more precisely. They continue, however, to fall short of the HRI objective of accurate perceptual as well as physical simulation.

There are two general approaches to simulation within the behavioral research and training communities: *full task* (high fidelity) and *part task*, sometimes called "small world". Full task simulation seeks to imitate the target system in as great a detail as possible on the premise that the relevant aspects and interactions involved in the operator's tasks are unknowable a priori. Part task simulation seeks to abstract significant features or characteristics of the target task and present these in a simplified simulation. Full task simulators are notoriously expensive and difficult to maintain. Only large organizations such as the military (aircraft and tank simulators) or utilities (nuclear power plant simulators) can afford them and they are typically too valuable for training to be made routinely available for research. While part task simulation has become a mainstay of behavioral research there are always questions of the appropriateness of the features chosen to model and their lack of the types of complexity that frequently trouble operators in the field.

Mobile robots fall in a sweet spot where it has recently become possible to develop high fidelity full task simulations at reasonable cost. This is due to three lucky coincidences:

1. mobile robots are typically controlled through widely available laptop or laptop-like computer interfaces and have a relatively small number of controllable/displayed parameters
2. rigid body physics simulations (physics engines) have become widely available in opensource and proprietary forms
3. computer gaming has led to the development of an extensive software and hardware infrastructure that now allows accurate simulation of camera video, the primary display modality used by human operators.

The emphasis on fidelity required for full task simulation for HRI meets exactly the same needs as those of robotic researchers developing for real platforms. Both need confidence that behaviors observed in simulation are a close approximation of the simulated platform. HRI researchers need these assurances of fidelity to demonstrate the relevance of their research to operational problems. Developers need the assurance in order to risk developing in simulation without constant verification against real platforms. Although USARSim was first released in the spring of 2003 as a closed platform targeting HRI and multiagent systems researchers, it was opened up in the following year with APIs for Player [9], Pyro [21], and MOAST [15] to increase its usefulness to USAR robotics researchers. Since then it has served both communities supporting HRI research primarily in the U.S. and USAR development world wide.

For USAR robotics researchers USARSim's emphasis on fidelity and validation has led to several notable advantages:

- Code developed to control robots within USARSim can be moved "as is" to real robots. This is possible because wrappers for control frameworks like Player [16], SIMware, and Pyro [21] have been developed and are widely used by the USARSim community. Examples of projects developed in simulation and moved to real world systems abound and have been described in various publications. A partial list of publications describing USARSim or research results obtained using USARSim are available on [17]
- USARSim is a tool for *early testing and late binding*. This means that it can be used to verify the impact of design choices before they are physically implemented on the real system. For example, it is easily possible to modify the position of a sensor, or its parameters (like resolution, signal to noise ratio, etc.) to verify the impact of these changes on the performance of the system being developed.
- USARSim includes models of sensors, robots, and actuators used everyday by robotics researchers

In summary we attribute USARSim's success to a few simple design principles:

*Never build what you can borrow*- Earlier simulations, even those incorporating the ODE physics engine, have been forced to develop extensive simulation infrastructures. While accurate robot models are fairly easy, good models of the world incorporating realistic complexity are nearly impossible to build without sophisticated tools. Because both HRI and USAR robotics are focused on robots' interaction with their environment rather than behavior in isolation, fidelity of the world is just as important as fidelity of the robot. By using a game engine, USARSim falls heir to a suite of tools for building and extending realistic environments.

*Piggyback the development cycle*- Computer graphics and other areas impacted by gameplay have a very rapid development cycle. Taking advantage of these new features as they appear would be impractical for any academically developed simulation. By piggybacking on a game engine whose developers are driven by the market to incorporate new technologies as rapidly as possible, USARSim is able to remain near the state of the art without the expenses associated with development.

*Validation is everything*- For the communities using USARSim validation is its most important feature. By offloading the development of the simulator and the need for technology driven revisions, USARSim allows developers the time needed for thorough validation of the platforms and sensors being modeled. It is in fact this emphasis on validation (see next section) that distinguishes USARSim from other robotic simulations.

*Sustained management*- Unlike simulations maintained on an incidental basis by their academic developers, USARSim has benefited from the early involvement and management by NIST personnel. NIST involvement has led to the elimination of ad hoc exceptions and extensions and the codification of extensions in the form of *mission packages*, the introduction of standard scaling into the simulation, and the ongoing support of inhouse and external validation efforts.

## IV. VALIDATION

USARSim's most distinctive feature is perhaps found in its thorough quantitative validation. In fact, along the years its developers created a methodology that guarantees close relations with real world systems, and continuous reality checks. Most importantly, the validation procedure is fairly generic and can in principle be applied to any robot simulator. The idea is rather simple. Whenever a new component (sensor, robot, etc), is added to USARSim, one or more experiments should be designed in order to assess the accuracy of the simulation. These validation experiments will be performed twice, once in the real system, and once in a simulation setting resembling as much as possible the real system used. After the two experiments, results should be quantitatively compared. In the past, sensors like vision and proximity range finders were validated in this way [4]. Later on, the methodology was lifted at a higher level, and more complex units were validated; for example SLAM algorithms or human-robot interfaces [13]. In order to exemplify, the procedure followed to validate the most recently added sensor, i.e. the GPS is presented (see [1] for an in depth discussion of the many issues arising to design a realistic GPS simulation module within USARSim). In order to validate the newly added sensor, a P2AT robot equipped with a GPS received was driven around the UC Merced campus. The robot was purposely driven in open areas far from buildings as well as in close proximity to them. During these runs, the number of satellites visible by the GPS receiver was logged, as well as the path returned by the reader. Next, the same experiment was performed in simulation. A model of the relevant part of the campus was developed, including appropriately scaled buildings. The simulated robot was then driven through the same path at the same simulated time of day (to experience the same positions of the NAVSTAR satellites), and the same information was logged and compared. Figure 4 and Figure 5 show the results obtained.

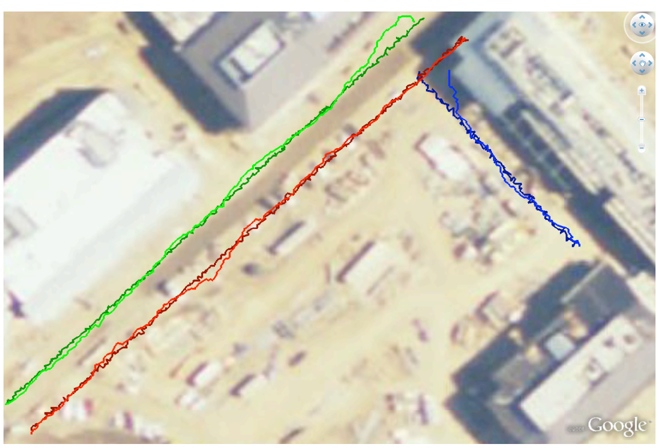


Figure 4 Comparison between results obtained in simulation (dark paths) and with the real GPS sensor (bright paths). Paths provided by the GPS were overlaid to the appropriate map retrieved from Google Earth.

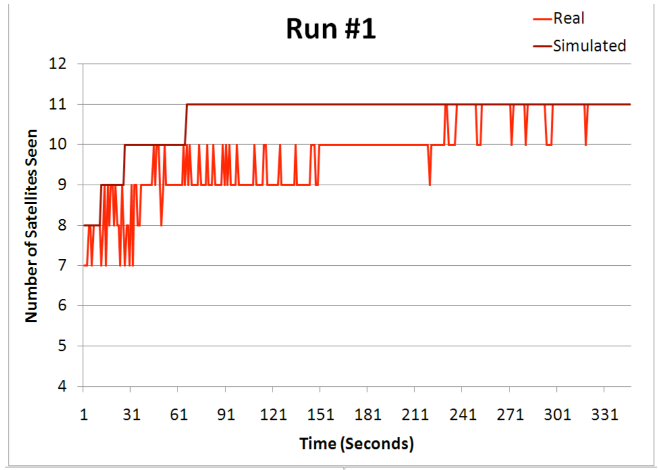


Figure 5: comparison between the number of satellites tracked by the simulated GPS sensor (dark series) and by the real GPS sensor (bright series). Even thought the simulated sensor almost always tracks one more satellite than the real one and exhibits a less jagged profile, the trends are clearly the same.

## V. ROBOT COMPETITIONS

In 2004 USARSim was brought to the attention of the Robocup federation as a useful tool to promote further development of the recently introduced RobocupRescue competition. At that time two competitions had been introduced, namely the RobocupRescue Robot league and the RobocupRescue Simulation league. With the former targeting mostly single robot hardware implementations operating in rather small areas, and the latter dealing with multi-agent systems acting on a city sized scenario, USARSim was proposed as a tool to bridge these two otherwise distant research communities. After a demonstration that took place during Robocup 2005 in Osaka, official tournaments started in 2006 under the name of Virtual Robots Competition [18]. Virtual Robots became the third competition held within the RobocupRescue Simulation league. The scientific goal of Virtual Robots is to provide a scenario where teams are called to perform rescue missions requiring multiple cooperating heterogeneous robots operating both inside and outside of a disaster area of the size of a few city blocks. Both autonomy and teleoperation are accepted and rewarded by the scoring schema [3][5][7]. Figure 6 shows one of the competition worlds used during the competition held in Bremen in 2006. The scenario portrayed was an explosion happened in building located at the intersection of four major roads. Rescue missions took place both inside and outside the building. In 2007 and 2008, the scenario consisted of a major train wreck occurred within a train station with a subsequence chemical spill and fire involving nearby office buildings (short videos of these environments are available on [22]).

Figure 6: USARSim is the simulation engine used to run the RobocupRescue Virtual Robots competition. The picture shows one of the environments used during Robocup 2006. A tracked Telemax robot and a flying AirRobot platform can be seen in the screenshot.

The goal of a competing team is to explore and map the unknown area and to report as much information as possible to a hypothetical team of first responders that have to enter the area to rescue victims. Relevant information includes the map of the environment, location of the victims, and classification of the explored terrain (victim free, unknown, etc). After three successful events, notable progresses in cooperation, human robot interfaces, and adjustable autonomy were observed. By rising the bar every year, the organizing technical committee has forced teams to develop new skills, and has also brought new extension to USARsim. For example, starting from 2007, teams have been forced to use a so-called wireless simulation server (WSS) in order to exchange information between robots and the GUI during the competition. In 2008, flying vehicles with limited sensor payload were introduced and significantly wider competition scenarios were presented to the teams in order to encourage them to use cooperative heterogeneous teams including the newly introduced air platform. It is important to remark that all participants in the Virtual Robots competition are required to release their code to the community in order to lower entry barriers for new comers. This may be seen in the fact that the winner of the 2008 competition (RedSun from Southeast University, Nanjing, China) based their code on the 2007 wining code (Steel of Pittsburgh/CMU). Moreover, some of the scoring tools initially adopted within the Virtual Robots competition have also been embraced by the RobocupRescue Robot league, thus closing the loop between simulation and reality.

In addition, 2007 saw the first running of the NIST/IEEE U.S National Virtual Manufacturing and Automation Competition. This competition focuses on using open source robotic software to act as both a teaching aid and to help to solve real world manufacturing problems. Problems that have been gathered from interviews with actual current and potential users of manufacturing robots are decomposed into quantifiable elemental tests. The compaction is then based on a combination of these elemental tests and an overall scenario. 2007's elemental tests included path following accuracy and the evaluation of the ability of a Automated Guided Vehicle (AGV) to autonomously dock with a loading station in a constricted area. For the docking task, competitors were presented a series of rooms that had decreasing maneuver space, and their docking speed and accuracy was evaluated. An example of the success of simulation is that students who had never even seen NIST's robotic AGV were able to directly run their simulation developed docking code on the real hardware with no software changes and identical results.

Areas that will be evaluated in this year's competition include pallet handling by an autonomous forklift, traffic management of multiple vehicles, and the ability to plan routes for material delivery in dynamic environments. More information on this competition may be found at www.vmac.hood.edu.

## VI. Future Challenges

The USARSim community is now facing a transition process, i.e. there is ongoing debate on how to move the overall project into a next-generation game engine, and what are the most critical issues to be addressed in order to create a new tool to better serve the robotics community. Participation in the workshop is therefore seen as a great and timely chance to exchange ideas and results with a broader audience of specialists in the field. The following are issues that are deemed relevant by the USARSim community in order to drive the decision process:

- The ability to create, import and export textured models with arbitrarily complicated geometry in a variety of formats is of paramount importance. From this point of view the Unreal Ed bundled with Unreal is one of its biggest strengths.
- High fidelity rendering should be coupled with high fidelity physics simulation. The ideal next generation simulation engine shall allow the simulation of tracked vehicles, sophisticated friction modeling, and the like. These extensions are needed in order to allow tasks currently not supported, for example multifingered manipulation, and the like.
- The system should be open as much as possible, i.e. it should be easy to add a new robot and to code novel components based on the available primitives
- Backward compatibility with the standard USARSim interface should be assured.

## VII. Conclusions

We have presented USARSim, a high fidelity robot simulator enjoying significant popularity in the robotics community. In our opinion there are five reasons for its remarkable success. First, the distribution model allowed for a rapidly created and lively community of developers who have contributed many extensions. Secondly, by leveraging the strengths of a commercial game engine, robotic developers can concentrate on robotic relevant issues, while delegating rendering and other aspects to the engine. Thirdly, the availability of interfaces compatible with widely used controllers allow for the migration of code from simulation to real robots and vice versa at no cost. The fourth reason for success has been the coupling of USARSim, with Robocup, thus forcing the community to set yearly milestones and develop a continuously updated research roadmap in sync with realistic robotic needs. Finally, validation efforts convinced people otherwise skeptical about robot simulation that USARSim is indeed a useful tool to develop code to be eventually run on real robot systems.

## ACKNOWLEDGMENTS

The authors thank the many enthusiastic developers all around the world that contributed models for sensors and platforms, bug fixes, feedback and much more.

## REFERENCES

[1] B. Balaguer, S. Carpin. *Where am I? A Simulated GPS Sensor for Outdoor Robotic Applications*. Submitted to the first International Conference on Simulation, Modeling and Programming of Mobile Robots.

[2] B. Balaguer, S. Carpin, S. Balakirsky. *Towards Quantitative Comparisons of Robot Algorithms: Experiences with SLAM in Simulation and Real World Systems*. Workshop on "Performance Evaluation and Benchmarking for Intelligent Robots and Systems" at IEEE/RSJ IROS 2007

[3] S. Balakirsky, S. Carpin, A.Kleiner, M. Lewis, A. Visser, J. Wang, V.A. Ziparo. *Towards heterogeneous robot teams for disaster mitigation: Results and Performance Metrics from RoboCup Rescue*. Journal of Field Robotics 24(11-12):943-967, 2007

[4] S. Carpin, T. Stoyanov, Y. Nevatia, M. Lewis, J. Wang. *Quantitative Assessments of USARSim Accuracy*. Proceedings of PerMIS 2006

[5] S. Carpin, J. Wang, M. Lewis, A. Birk, A. Jacoff. *High fidelity tools for rescue robotics: results and perspectives*. Robocup 2005: Robot Soccer World Cup IX, LNAI Vol. 4020, Springer, 2006, pp. 301-311

[6] S. Carpin, M. Lewis, J. Wang, S. Balarkirsky, C. Scrapper. *USARSim: a robot simulator for research and education*. Proceedings of the IEEE 2007 International Conference on Robotics and Automation, 1400-1405

[7] S. Carpin, M. Lewis, J. Wang, S. Balakirsky, C. Scrapper. *Bridging the gap between simulation and reality in urban search and rescue*. In "Robocup 2006: Robot Soccer World Cup X", LNAI Vol. 4434, Springer, 2007, pp. 1-12

[8] B. Gates. *A robot in every home*. Scientific American, 296(1):58–65, 2007.

[9] B.P. Gerkey, R.T. Vaughan, K. Støy, A. Howard, G.S. Sukhatme, M.J. Mataric. *Most Valuable Player: A Robot Device Server for Distributed Control*. Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems,1226-1231

[10] B.P. Gerkey, R.T. Vaughan, A. Howard. The Player/Stage project: Tools for multi-robot and distributed sensor systems. Proceedings of the 2003 International Conference on Advanced Robotics, 317-323.

[11] K. Konolige, K. Myers. The Saphira architecture for autonomous mobile robots. In Kortenkamp, Bonasso and Murphy (ed.) *AI-based mobile robots: Case studies of successful robot systems* MIT Press., 1996

[12] P. Lee, D. Ruspini, O. Khatib. *Dynamic simulation of interactive robotic environment*. Proceedings of the 1994 IEEE International Conference on Robotics and Automation, 1147-1152.

[13] J. Wang, M. Lewis, M. Koes, S. Carpin. *Validating USARsim for use in HRI Research*. Proceedings of the 49th meeting of the Human Factors and Ergonomics Society, pp. 457-461, 2005

[14] Usarsim homepage: http://sourceforge.net/projects/usarsim (Accessed: 2008)

[15] MOAST homepage: http://sourceforge.net/projects/moast (Accessed: 2008)

[16] Playerstage homepage: http:/playerstage.sourceforge.net (Accessed: 2008)

[17] http://www.robocuprescue.org/wiki/index.php?title=Publications_on_Virtual_Robots_and_USARSim

[18] http://www.robocuprescue.org/wiki/index.php?title=Virtualrobots

[19] http://robotics.ucmerced.edu/Robotics/multimedia

[20] Ubersim http://www.cs.cmu.edu/~coral/ubersim.html (Accessed: 2005)

[21] PYRO -- Pyrorobotics http://pyrorobotics.org (Accessed: 2005)

[22] http://robotics.ucmerced.edu/Robotics/Multimedia