Underwater Simulator for Robotics Applications in High Depths

Felipe Guth, Silvia Botelho and Marcos Amaral

***Abstract*— Simulators are essential tools, mainly used before practical application projects, in various areas of engineering. The high development cost and complexity are just some of the factors that justify the development of simulators for a given project, which allows among other benefits, conduct tests of the agent behavior against an environment. In robotics, simulators have been used for a long time and consist an important stage in prototyping phases. When the focus is on high depths underwater robotics, simulators have even greater importance since the great difficulty of building robots for this kind of environment. This paper presents a simulator built with Blender IDE integrated with Robotic Operating System (ROS) for underwater robotics applications, using sensory data from a simulated physics game engine. We present results of applications using this project as well as discussion of the key benefits of its development. It is observed an important potential for projects in underwater robotics, and its appeal in demonstrations and examples for learning robotics and 3D modeling.**

Keywords: Simulator, Underwater Robotics, High depths, 3D modeling.

I. INTRODUCTION

Nowadays one of the problems faced by underwater robotics is the realization of tests with pertinent data, similar to those found in a specific real scenario. Previous works such as [1], [2], [3], [4], [5] show the importance and usefulness of using simulators for prototyping applications in underwater robotics. One purpose of our research group is the development of applications for localization, mapping and inspection to be used in robots ROV or AUV at high depths for typical environments found in oil and gas extraction. The cost of a robot for operation in environments of this type present high cost and effort of people and complex logistics. Taking this into account, the development of a simulator with data of sensors used in robots of high cost, that are able to work in high depths (above 1000 meters), consists an important test tool for the identification of problems, to obtain first results and reducing costs using simulated prototypes.

In other way but not less important, the use of a underwater simulator robot allows to demonstrate the final application of our research in a visual way, illustrating with images what is often difficult to explain with words. The prototype takes the attention of students in the outreach of the laboratory and helps to recruit new people, who can interact with the platform in public demonstrations and know our work.

The article proceeds with an overview of the Blender IDE, used to build the simulator. After, it is presented the simulator itself, the integration with the Robotic Operating System (ROS) and the module of sensory data acquisition, actuators and control. Finally, it is discussed the main results obtained with the simulator as well as a conclusion of the work and future directions.

*The authors gratefully acknowledge the contribution of CNPQ, CAPES, FAPERGS, FURG, ANEEL/CEEE and FINEP.

[1]The authors are with the Computational Sciences Center, Universidade Federal do Rio Grande - FURG, Rio Grande - RS, Brazil. Email: felipeguth, marcosamaral, silviacb@furg.br.

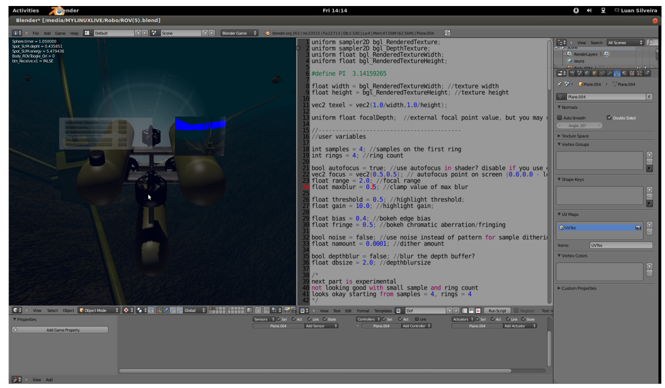


Fig. 1: A overview of the default Blender IDE.

II. BLENDER IDE

Regarding the Blender IDE, one of the main features is the simple 3D modeling and its integration with the Game Engine, which allows to test the prototype in real time after the construction of the scenario. Other advantage of Blender is the native support to Python, it has an block logical interface for applications and an API of Python Script [6] which turn possible a wide range of possibilities. Blender also supports OpenGl [7] programming through a framework of the Game Engine called BGL. Another interesting feature is the possibility to use a web browser client to access the application online on the Internet.

Blender has a very unusual interface, highly optimized for 3D graphics production. Blender provides a broad spectrum of modeling, texturing, lighting, animation and video post-processing functionality in one package. Through its open architecture, Blender provides cross-platform interoperability, extensibility, an incredibly small footprint, and a tightly integrated workflow. Blender is one of the most popular Open Source 3D graphics applications in the world [8]. The Blender 3D also contains a Game Engine which has a physics engine responsible for the physics simulation covering various phenomena like fluids, utilized in this experiment.

III. SYSTEM DESCRIPTION

A. Building the scenario

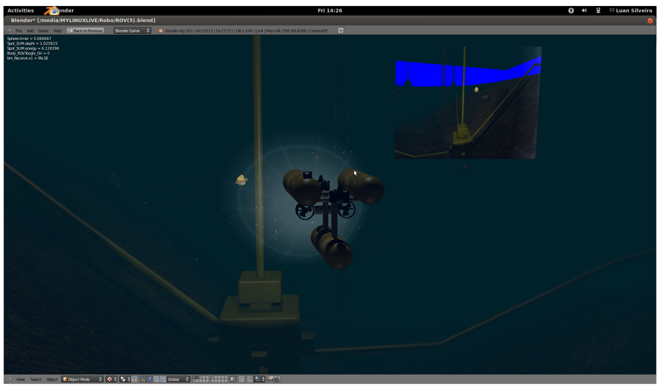
Aiming the execution of robotics applications, it was developed a underwater environment with characteristics

978-0-933957-40-4 ©2013 MTS

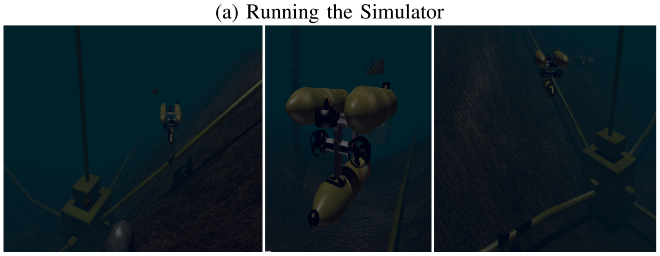
founded in a high depth oil and gas extraction scenario. The ambient was built through the Blender platform.

The Figure 2 shows the framework developed. In **??** the Blender IDE for building the application, in 2a the application running with the camera image on the right, which is published on ROS, in the Figure 2b some points of view from the scenario and robot are shown.

Examples of characteristics managed in this scenario are light levels, number of particles in the water, general fluids simulation and pressure over the object. These variables affect the operability of the robot and motion control, and can be configured according to different scenarios of operation found in real places.



(a) Running the Simulator



(b) Screen shots of a scenario

Fig. 2: Simulator

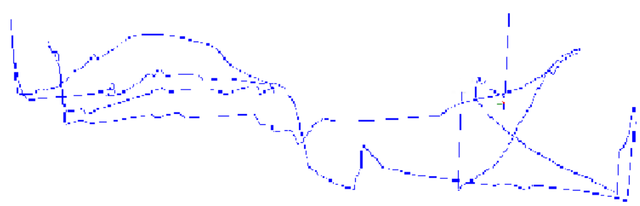
B. Integration with ROS

For the image capture from the scene, data transfer and data visualization, ROS tools were used. ROS is a set of opensource software libraries and tools developed by Willow Garage designed to help developers create robot applications [9]. Its ease of use and impressive array of both hardware drivers, and software capabilities has spawned a rapidly growing community of software developers. ROS uses a publish-subscribe architecture to orchestrate inter-process communication. The robot has a frontcamera which capture the scene right in the front of the robot position. This image is compressed and published in a ROS topic. In addition to the image, the framework provides odometry data with position and quaternion orientation data, besides the ground truth data. This data, relevant to odometry and ground truth, are published in other ROS topics.

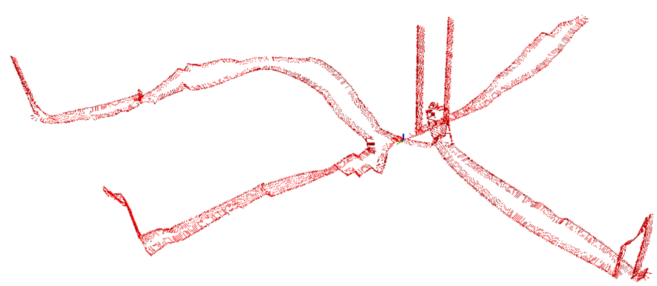
C. Sensors data, actuators, control and application sample

Through the development of functions based on the game engine of Blender it was possible to simulate sensors data by the movement of the object relative to the scenery, the speed is calculated on axes x, y, z. The game engine also allows to get the angular velocity in the form of a quaternion. The data provided by the simulator regarding velocities are total accurate, so it is necessary to include in the observations a noise term to simulate actual data more faithfully, since these measurements in real sensors are not accurate and have a certain amount of error. Still, the simulator has two cameras, one facing forward and the other down to capture images of the environment at different angles.

In its first phase of development the robot has 4 engines as main actuators and a light bulb to allow the capture of images by the camera. The robot control is performed via a joystick in operations of surge, heave and yaw. An application of SLAM was performed to test the use of the underwater simulator. Simultaneous localization and mapping (SLAM) is the process that enables a mobile robot to locate and build a map of an unknown environment using only observations relative to the most relevant features detected by its sensors. The solution to the SLAM problem is considered by many as a key prerequisite for making a robot fully autonomous [10]. For this test was used a biological inspired SLAM algorithm [11]. The results are shown in Figure 3.



(a) Generated Map



(b) Ground Truth

Fig. 3: Sample of the result of a mapping application

IV. Discussion

The simulation has always been a very useful tool in learning. With it its possible teach virtually without running the risks or pay the costs of a real experience. We can see the simulation constantly being used in aviation, computing and electronics, chemistry and other branches of knowledge, usually in order to teach and test ideas.

The robot simulation aims to assist in learning robotics, without the need for the costly purchase of materials and components, allowing more people to have access to this experience. In our particular case robotic simulation allows both testing specific applications using simulated sensory data as the demonstration and dissemination of the work in a easy manner. The following results can be discussed:

Research

The project was developed seeking robotic operation simulation in environments of oil and gas extraction at high depths, to this end, some research was done about structures of this type of environment. The robot modeled on a previous step was used, following the specifications of the actual model [12]. This framework has been used in applications of SLAM, path planning and computer vision. Also, the integration with ROS was performed turning the simulador in an important tool for practical research simulation in subaquatic robotics.

Education

In education we can use this 3D modeling of a complex environment besides the programming using Python Script to control the operation of the robot, capturing camera data and speeds as an example of robotic application. This prototype has served as an example also for other applications that use the same concepts and as a way to demonstrate the creation of robotic applications and their components. Another highlight was the demonstration of the operation of a robot for new students by providing a first impression of underwater robotics.

Outreach and recruitment

The simulator has been presented in exhibitions, events and demonstrations which the laboratory participates. During these events, one notices a great curiosity of the participants regarding the simulator. Many students are interested and end up discobering other projects and applications that use the simulator, arousing interest in the field of underwater robotics. Some have even gone on to work in the lab after meeting the project. Besides attracting new students, the simulator provides to the general public an idea of the problems addressed by the laboratory and visualize and understand them. This exemplification helps publicize the laboratory in academia and increase the interest for the researchers' work.

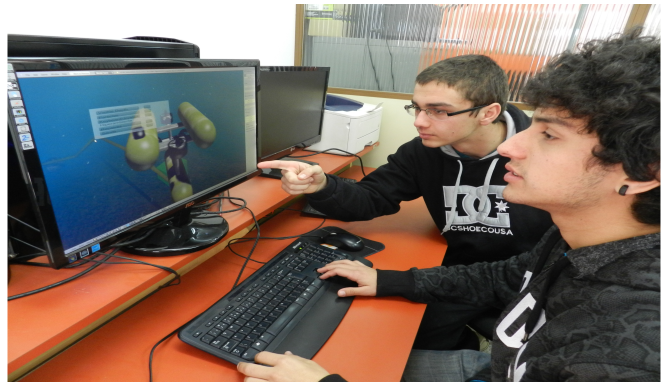


Fig. 4: New students learning concepts of underwater robotics using the simulator for the first time.

V. Conclusion and Future Works

The development of a simulator for underwater environments, such as oil and gas extraction, in high depths has proved to be an interesting tool for testing algorithms used in underwater robotics applications. Given the large cost of real robots for high depths and the lack of good simulators, this work represent a very useful tool for prototyping underwater applications in high depths. Besides, important results in research, education, outreach and recruitment are noticed. As future work, we aim to develop a simulated sonar to detect objects such as pipes, as well as the design of an ROV compatible with those used in high depths, presenting larger size and operating restrictions. Furthermore, in a future phase the simulator will be available to the community with all functions integrated with ROS.

References

[1] S. McMillan, D. Orin, and R. Mcghee, "Efficient dynamic simulation of an underwater vehicle with a robotic manipulator," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 25, no. 8, pp. 1194–1206, 1995.

[2] J. da Cunha, R. Costa, and L. Hsu, "Design of a high performance variable structure position control of rovs," *Oceanic Engineering, IEEE Journal of*, vol. 20, no. 1, pp. 42–55, 1995.

[3] Y. G. Le Page and K. Holappa, "Simulation and control of an autonomous underwater vehicle equipped with a vectored thruster," in *OCEANS 2000 MTS/IEEE Conference and Exhibition*, vol. 3, 2000, pp. 2129–2134 vol.3.

[4] F. Song, P. An, and A. Folleco, "Modeling and simulation of autonomous underwater vehicles: design and implementation," *Oceanic Engineering, IEEE Journal of*, vol. 28, no. 2, pp. 283–296, 2003.

[5] T. Prestero, "Verification of a six-degree of freedom simulation model for the REMUS autonomous underwater vehicle," Master's thesis, Massachusetts Institute of Technology, USA, 2001.

[6] Python.org, "Python programming language ? official website," 2013. [Online]. Available: http://www.python.org/

[7] OpenGl, "The industry foundation for high performance graphics," 2013. [Online]. Available: http://www.opengl.org/

[8] Blender, "http://www.cyberbotics.com," 2013, open Source 3D Tool. [Online]. Available: http://blender.org

[9] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[10] C.-C. Wang, C. Thorpe, S. Thrun, M. Hebert, and H. Durrant-Whyte, "Simultaneous localization, mapping and moving object tracking," *The International Journal of Robotics Research*, vol. 26, no. 9, pp. 889–916, 2007. [Online]. Available: http://ijr.sagepub.com/content/26/9/889.abstract

[11] L. Silveira, F. Guth, S. Botelho, P. Drews, and D. Fischer, "Biological inspired system for localization and mapping in underwater environments," in *OCEANS 2013 MTS/IEEE Conference, in press.*, 2013.

[12] D. Ribas, N. Palomeras, P. Ridao, M. Carreras, and A. Mallios, "Girona 500 auv: From survey to intervention," 2012.