1G3-2

Simulation and Evaluation of Rescue Robots in Blender

Alhamidi Abdullatif [*1] Hattori Shun[*2] Kameda Hiroyuki[*2]

[*1] Graduate School of Bionics, Computer and Media Sciences, Tokyo University of Technology

[*2] School of Computer Science, Tokyo University of Technology

In recent years, the use of rescue simulation to recreate virtual rescue robots and environments which have been struck by natural disasters has been increasing. The benefits of using virtual robots and environments instead of actual robots and environments are many, and the most significant of those benefits are cost and time. In this paper, we evaluate the possibility of using the Blender Game Engine and Bullet Physics to accurately simulate the aforementioned robots and environments. We analyze the advantages of using the Blender Game Engine instead of more traditional game engines for simulation purposes. Finally, we compare and grade the virtual robots used in this experiment based on the results of manual and autonomous robot operation.

1. Introduction

When a natural disaster such as an earthquake or a tsunami occurs in a certain area, there are bound to be countless collapsed buildings and scattered victims in that area. Using human rescuers in such an unknown and fragile place to locate victims would be a risky move since the rubble from the buildings poses an increased and unnecessary risk for the human rescuers. Therefore, Rescue Robots are used to survey and analyze the disaster struck area, and then report back to their human controller about the location of the victims stuck in the disaster area.

2. Background

2.1 Problems with Rescue Robots

Rescue Robots seem like the perfect solution for surveying disaster struck areas, but they are not without their own problems, mainly because Rescue Robots need to be put under hard and rigorous tests before they are deployed to disaster struck environments where they are expected to save precious human lives. However, the process of testing the robots is both costly and time consuming, because the researcher has to arrange to transport the robot to the testing facility each time he makes even minor changes to the robot, since the results from those changes might change the final outcome of the Rescue Robots evaluation tests. Furthermore, the researcher is not free to test the robot whenever he chooses because the robot evaluation tests are only done a few times a year by the organizations which host those evaluation tests. For example, the International Rescue System organization in Japan has had so far 2 robot evaluation tests done in 2011, and the next robot evaluation test's date is yet to be determined [1].

Contact: Alhamidi Abdullatif, Graduate School of Bionics, Computer and Media Sciences, Tokyo University of Technology, Tokyo, Hachioji, Katakuramachi 1401-1, 042-637-2111, abdullatif.alhamidi@gmail.com

2.2 A possible solution

Time and cost are the main downsides of Rescue Robots, but what if we had no such restrictions? Would it not be nice to perform all the Rescue Robot simulations that we wanted to with low cost and minimal time? Well, it has become possible to do so with computer simulation programs such as USARSim (Unified System for Automation and Robot Simulation) [2], but for a variety of reasons which will be explained in Section 3, we chose to use the Blender Game Engine and the Bullet Physics Engine to perform our Rescue Robots simulation tests.

3. Rescue Robots in Blender

Our reasons for choosing Blender are as follows.

• Convenience: Blender is an all in one development environment. It is possible to model your robot inside of it, perform physical experiments in it, and test the result with the game engine because Blender contains a modeling tool, a physics engine and a game engine all in one package.

• Support: Blender is updated frequently and has a large supportive community.

• Open Source: Blender is open source and uses the very easy to learn Python. It does not force you to use a proprietary scripting language.

3.1 Simulation modes in Blender

We chose a Seekur Jr robot to run our first tests in Blender, because of it powerful motor, long battery life and abundance of sensors which the Seekur Jr is able to use [3]. The Seekur Jr's dimensions are shown in Table 1.

Table 1: Seekur Jr dimensions

	Length	Width	Height
Seekur Jr	105 cm	84	50

We have built a virtual Seekur Jr model in Blender, and have made sure that it matches the dimensions in Table 1. This was done to establish a high level of accuracy when the Seekur Jr model interacts with the environments surrounding it, and the Blender Game Engine starts the collision detection process.

Our Blender-made Seekur Jr model and the testing environment model are shown in Figure 1 below.

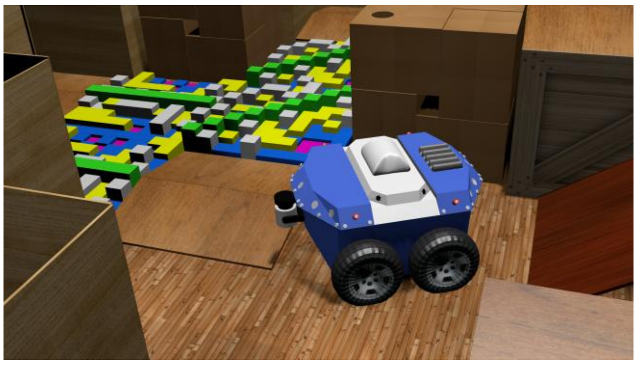
Figure 1: Seekur Jr model and environment model

For testing purposes we have made two different simulation modes for our Blender-based simulator.

3.1.1 Manual mode

In manual mode, a human controller has to control the robot and try to locate the victims' location in the shortest possible time. For added realism, the robot has to be controlled in first person view, and that is why we have assigned a PTZ (Pan Tilt Zoom) camera which can be controlled and rotated on the top of the Seekur Jr model. Figure 2 shows the first person view mode from the PTZ camera.

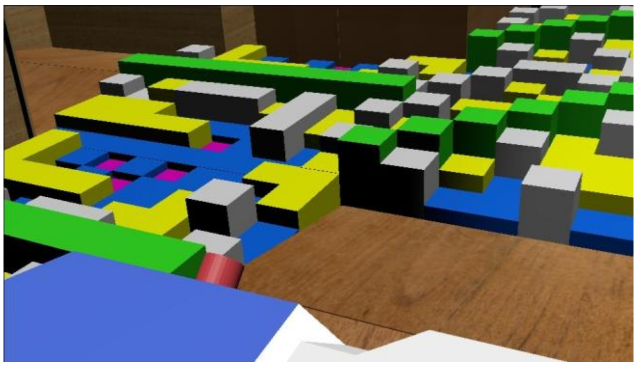
Figure 2: Seekur Jr PTZ camera view mode

Both the Seekur Jr and its PTZ camera can be controlled by using either a computer keyboard or a regular videogame controller.

3.1.2 Autonomous mode

Autonomous mode has not yet been implemented because our research is still a work in progress. However, we hope to implement a path finding algorithm in autonomous mode which can scout the area, and detect the location of all victims with the help of various sensors in the near future.

3.2 Sensors

So far we have added sound sensors to the virtual environment model in Figure 1. When the Seekur Jr robot comes within a distance of 10 meters of the sound sensors, they start to emit a sound to signal that there is a victim trapped inside the rubble. The sound sensors are able to emit sounds randomly from a different location each time the simulator is started. Figure 3 shows the location of the current sound sensors.

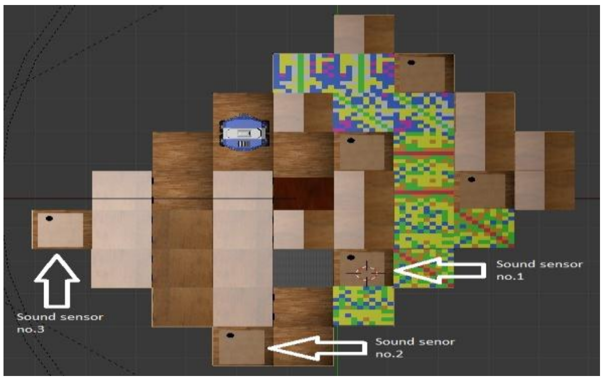

Figure 3: Sound sensor locations

We did encounter a few problems while doing some initial testing, because the original size of the virtual environment model was 10m x 7.5m. The relatively small size of the virtual test environment meant that the Seekur Jr robot was able to trigger and hear the sound sensors as soon as we started our simulation. This took all the realism out of the simulator and meant that the human controllers were able to find the location of each victim within a few seconds of starting the simulator. That is why we have decided to use a much bigger virtual test environment (100m x 75m) in our future simulations.

4. Evaluation

Initial evaluations have been good. We were able to configure the physics settings and the collision detection parameters without any problems. We were also able to add multi-camera support and game controller support to further increase the realism of the simulator. And we also discovered that it is possible to greatly increase the capabilities of the Blender Game Engine and adjust vehicular physics via Python scripts. We see no reason so far why the Blender Game Engine cannot be used to cheaply and efficiently simulate Rescue Robots.

5. Conclusions and future directions

In this paper we presented an alternative method in evaluating Rescue Robots for use in real rescue scenarios. We have shown that Blender is very capable and already provides numerous tools which can be tweaked and used to grade robots in a short period of time. This is very much a work in progress, but we hope to show how accurately Blender can be used to simulate Rescue Robots in the near future.

References

[1] International Rescue System official website, http://www.rescuesystem.org/IRSweb/event.html (2011).

[2] S. Carpin, M. Lewis, J. Wang, S. Balakirsky, C. Scrapper (2007). USARSim: a robot simulator for research and education. Proceedings of the 2007 IEEE Conference on Robotics and Automation, pp. 1400-1405 (2007).

[3] Seekur Jr spec sheet, http://www.mobilerobots.com/ResearchRobots/ResearchRobots/SeekurJr.aspx (2011).