Realistic Simulation of Laser Range Finder Behavior in a Smoky Environment

Okke Formsma, Nick Dijkshoorn, Sander van Noort, and Arnoud Visser

Intelligent Systems Laboratory Amsterdam, Universiteit van Amsterdam,
Science Park 107, NL 1098 XG Amsterdam, The Netherlands
http://www.science.uva.nl/research/isla
{okke.formsma,nick.dijkshoorn,alexander.vannoort}@student.uva.nl,a.visser@uva.nl

Abstract. The Urban Search and Rescue Simulation used for RoboCup lacks smoke to which laser range finders respond realistically. In this paper, the behavior of a Hokuyo and Sick laser range finder in a smoky environment is studied. The behavior of the lasers is among others a function of the visibility level, and here this relationship is quantified in an explicit model. This model is implemented in a simulation environment which is the basis of the Virtual Robot competition of the RoboCup Rescue League. The behavior of both real and virtual laser range finders is compared in a number of validation tests. The validation tests show that the behavior of the laser range finders in the simulation is consistent with the real world.

1 Introduction

Urban Search and Rescue (USAR) is a challenging application for teams of robots. The environment in a USAR setting is generally hostile. To make the work for human rescue teams safer, robots can be deployed to scout the environment. The robots can explore the area and create maps [1].

The RoboCup Rescue competition provides a demanding environment to benchmark research in this area [2]. There is a real robots league and a simulation league. In this paper, we will focus on the simulation league. The simulation league has a number of advantages over the real robots league. Firstly, no hardware needs to be purchased and the software is inexpensive. Secondly, there can be freely experimented with multiple robots without fear of destroying the expensive equipment in this hostile environment. Last but not least, in simulation it is easier to reproduce results, because one has full control over the environment. The downside of simulation is that all results have to be validated on real systems, as for instance done in this paper.

The simulation league has a software platform called USARSim which simulates robots on a physical realistic level. The latest incarnation of USARSim builds on the commercially available Unreal Engine 3.0 by Epic Games. This engine allows users to build environments through an editor. The engine also provides a programming language called Unreal Script. USARSim adds robots to the game [3]. These simulated robots resemble the robots of the Real Robots competitions closely. The movements and sensor responses of the simulated robots are very similar to the real robots [4–7].

The challenge for the RoboCup Rescue competition in 2009 was a railway station in which a train was derailed. In the scenario[1], some parts of the derailed train are on fire, producing a smoke column (see Fig. 1). This effect is only visual; the smoke does not influence the sensor readings of nearby robots in any way. This study is meant to include the influence of smoke to measurements of the laser range finders. Laser range finders are used by all teams in the competition for localization and mapping purposes [8].

In section 2, previous research on the behavior of laser range finders in smoky conditions is outlined. Section 3 describes the setup and findings of our own experiment with a laser range finder in a smoky environment. The implementation of this behavior in the simulation is detailed in section 4. In section 5 results of validation tests are presented.

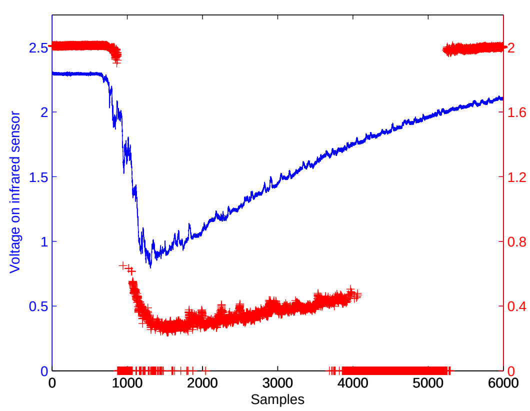
[1] This scenario can be downloaded from http://sourceforge.net/projects/usarsim/

Fig. 1: Snapshot of one of the worlds used in the 2009 competition of the RoboCup Rescue Simulation League.

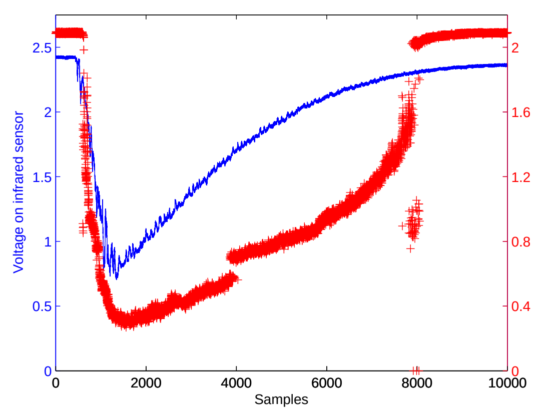
2 Prior work

While it is commonly known that laser range finders perform erratically under poor atmospheric conditions as dust or steam [9], little research has been conducted on the details of the conditions under which laser range finders (LRFs) fail and how they fail. Two studies relevant to this research have been found [10, 11].

A study on the behavior of four LRFs by Pascoal et al. [10] compared the performance of four LRFs operating under adverse conditions. Two of these are widely used in robotics: the Hokuyo URG-04LX and the Sick LMS200. In one experiment a room was filled with smoke, which slowly dissipates through a small exit. The density of the smoke is measured with an infra-red emitter and a photodetector. A laser range finder is set up two meters from a white MDF surface. Fig. 2a shows the readings of the photodetector and URG-04LX through time[2].



(a) Hokuyo URG-04LX



(b) Sick LMS200

Fig. 2: Laser range finder and infra-red photo sensor readings in a smoky environment from Pascoal et al. [10].

The readings start without smoke. Without smoke, the URG-04LX returns the correct distance to the MDF surface. When the smoke density gets too high, the voltage on the infra-red sensor drops and

[2] This is an analysis performed by us in Matlab on the dataset collected by Pascoal et al. [10]. On request the authors kindly provided us with the dataset of their experiment.

the LRF starts sending erroneous readings, shows as points on the x-axis in Fig. 2. The LRF returns short distance readings when the smoke is very thick.

The Sick LMS200 laser range finder exhibits different characteristics, as can be seen in Fig. 2b. As more smoke fills the air, the distance reported the drops steadily. There is a density range where the Sick returns error messages too, although this range is much smaller than for the URG-04LX.

Peynot and Scheding [11] published a data set about multi-sensor perception in natural environments with challenging conditions. A multitude of sensors were mounted on a skid-steered vehicle, including four Sick laser range finders and a video camera. Data was collected under different conditions, such as dust clouds, generated by blowing air to dusty soil using a high-power air compressor, and smoke clouds, generated using emergency smoke bombs that worked for about one minute, which were carried across the test area by the wind. This results in different smoke densities over time. Various natural environments were used as testing area, containing objects such as trees, houses, tables and carts.

Fig. 3: Snapshot of the data from Peynot et al. [11]. On the left laser range finder readings in a smoky environment. The x-axis is measurement angle, y-axis is time. The darker the color, the shorter the reading is. Red pixels indicate an error code was returned by the laser range finder. The vertical black lines indicate static obstacles. The diagonal black stripes indicate dynamic smoke. On the right the corresponding visual is shown.

The density of the smoke in this experiment varies through time. Fig. 3 shows the readings of one laser range finder through time[3]. The robot vehicle was standing still during these measurements. In this figure several static obstacles are visible in front of the vehicle (vertical lines). Additionally, several smoke clouds are visible, drifting from left to right (diagonal stripes). The diagonal stripes are mostly black, indicating that the laser range finder sees the smoke as an obstacle. On the edges of the smoke red lines are visible, indicating that the Sick laser range finder returns an error code. Many error codes are also visible in the lower right where multiple smoke clouds collide, generating turbulence and swiftly fluctuating smoke densities.

This shows that the Sick laser range finder reacts to this smoke similar to a solid object. The readings that result in errors are usually at the edges of a smoke cloud. At the edges of the cloud, the smoke is less dense.

From the finding of Peynot et al., combined with the findings of Pascoal et al., we hypothesize that laser range finders react in three distinct ways to smoke, dependent on the density of the smoke.

1. Without smoke or with little smoke, a laser range finder reacts normally; it returns a measurement within the margin of error.

[3] This is an analysis performed by us in Matlab on the dataset '07-StaticSmoke', available from `http://sdi.acfr.usyd.edu.au/datasets/`

2. When the smoke density is between a lower and an upper bound, the laser range finder returns a failure code.
3. If the density is higher than this upper bound, the Sick laser range finder reacts on it as if it hits a solid object. It returns a distance which is inversely proportional to the density of the smoke.

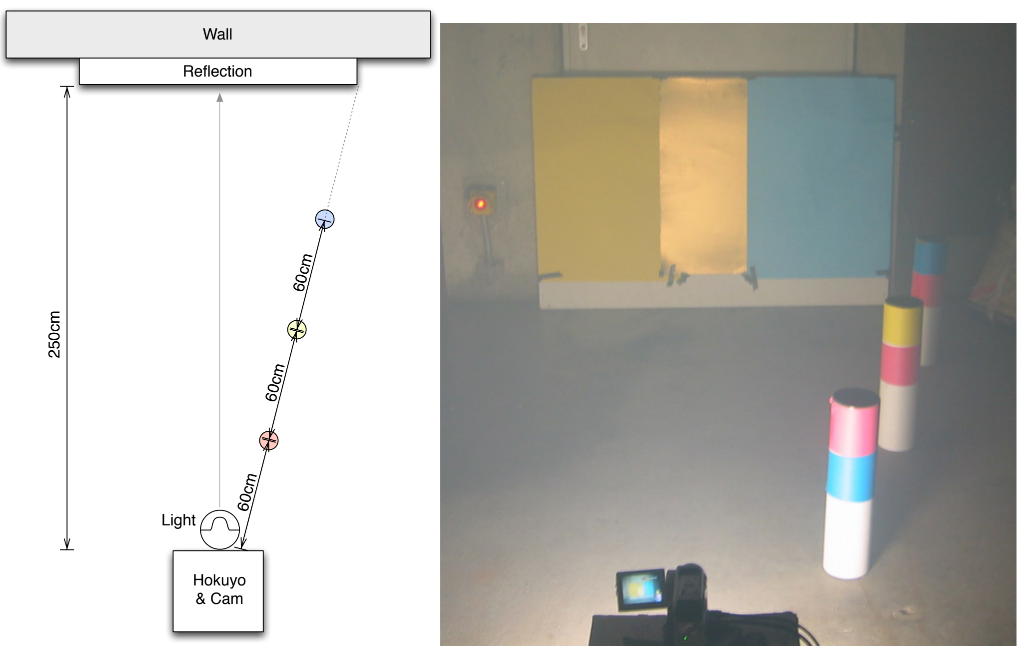
3 Experiment

The prior work is not sufficient to quantify the parameters of our LRF smoke model. USARSim does not have an infra-red sensor (as used in [11]) or any other sensor to assess the density of the smoke directly. We are restricted to the use of a camera to assess the density of smoke in order to be able to reproduce the experiment in USARSim. We did our own experiment to collect data that includes range measurements of the Hokuyo URG-04LX for different smoke densities and uses camera images to describe the density of the smoke. The gathered data is used to confirm the laser range finder model and find the parameters for this model.

Essential to this approach is the ability to control the smoke density. The international fire department training center BOCAS[4] provided us with facilities to generate smoke in a controlled environment.

3.1 Setup

The experimental setup was build inside a training room at BOCAS. The room contained a fog machine to generate smoke. This machine produces safe smoke on an oil basis and is also used in theaters.



(a) Schematic experimental setup (b) Photo of the experimental setup

Fig. 4: Experimental setup

The Hokuyo URG-04LX range sensor has a maximum range of 4 meters. The sensor was placed at a distance of 2.5m in front of a paper surface. To the right of the paper surface are rusty steel plates, which are used as heath shields for the furnace present in the room. Behind the LRF a camera is placed to visually record the smoke density. A 70W fluorescent lamp is placed below the Hokuyo,

[4] A description of the facilities of the center can be found at `http://www.boc.eu`.

pointing towards the paper surface to illuminate it. The measurements were synchronized by quickly removing a piece of cardboard that was held in front of the laser range finder and camera.

The camera is used to determine the density of the smoke. The primary method to assess the smoke density is by calculating the color saturation of the paper surface. The paper surface we used consists of blue, yellow and gold parts (see Fig. 4b). Three different colors were chosen to minimize the influence of smoke color. A study by Donate et al. [12] describes the task of reconstructing images of scenes occluded by thick smoke. They observed that an increase in smoke density induces a decrease in both image contrast and color saturation. The saturation level is obtained by converting the RGB camera images into HSV (hue, saturation, value) color space and taking the saturation color channel. The secondary method to measure the smoke density was placing three bollards. These bollards are positioned at 60cm intervals between the camera and the static object (see Fig. 4a). They are standing below the line of sight of the Hokuyo and don't have influence on the range measurements. By counting the number of visible bollards that are visible an alternate density measure can be defined.

This setup is used to measure smoke from two distinct sources. The first kind of smoke is generated by the smoke machine. The smoke has a light grey color and can be produced at will. It is ejected from a small hole from the machine and takes some times to spread through the room, where it dissolves after some time. The second kind of smoke measured was real smoke, coming from burning newspapers, pizza boxes and a Santa Claus doll made from fabric, filled with Styrofoam. These materials were placed at a distance of 1.5m from the LRF, and lit on fire. Unfortunately, the densities of this real smoke did not reach the levels of the artificial smoke.

3.2 Dataset

The laser sensor data is recorded to file using the Hokuyo intervalScanner.exe application[5]. The intervalScanner.exe application captures 100 measurements at intervals of approximately 300 ms. This means we are able to measure at 3 Hz.

Each measurement is placed on a single line and consists of a timestamp and pairs of angles (radians) and distances. The timestamps and measurements are written to a log file.

The dataset consists of five different runs[6]. The first three runs describe the behavior of the Hokuyo for smoke that has been generated by the smoke machine. During the fourth run the laser and camera were pointed at a large gas flame. The last run contains data on smoke from the bonfire described in the previous section. Each run contains a text file with laser range data and a video file.

We have interpreted the recorded data by visualizing it using Matlab. This script is able to handle datasets in different formats, including the format from Pascoal [10], Peynot [11] and our dataset. The laser sensor data is plotted and shows the measured distance for each measured angle. The data is plotted over time to analyze the changes in the sensor measurements causes by smoke. The corresponding camera image for each timestamp is displayed near to sensor data plot.

3.3 Findings

The Hokuyo LRF is affected by both artificial and real smoke. More precisely, the Hokuyo returns correct distance measurements until the density of the smoke reaches a certain density level. If the smoke density exceeds this level, the Hokuyo returns error values between 0 and 20. From Fig. 5 we can see that this density level[7] is exceeded when the saturation drops below 0.14.

Pascoal et al. [10] find that the distance reported by the laser is dependent of the surface optical characteristics and "although the output for all sensors does not change significantly with the target colour, there is a noticeable influence according to the specular characteristics of the surfaces". Our results confirm this statement. Pascoal et al. found a different behavior of the Hokuyo laser range finder when the obstacle was covered with black velvet. We observed unexpected behavior of the Hokuyo URG-04LX for the reflecting gold paper and the rusty steel plates. The distance reported by the LRF was 10% larger than expected.

[5] This program is available for download from `http://www.hokuyo-aut.jp/cgi-bin/urg_programs_en/`

[6] The recorded dataset is available on the website `http://code.google.com/p/usarsim-smoke-fire/`

[7] The smoke density is assessed by the saturation of the paper surface (primary method). The secondary method that uses bollards is ignored because the saturation reflects the actual smoke density well.

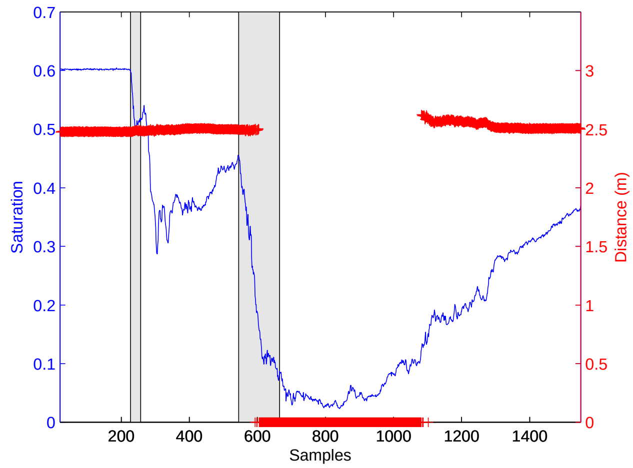


Fig. 5: BOCAS experiment run 1: saturation and laser measurements over time. The grey areas show when the smoke machine was turned on: first for 10 seconds, thereafter for 40 seconds.

4 Implementation

The implementation of the different smoke types and smoke model is done using *UnrealScript*. UnrealScript is the scripting environment of the Unreal Engine. In the Unreal Engine, the *Actor* class is used as the base class for many classes. All functional objects in the game that is not geometry like walls, is an *Actor*. It provides an interface for default standard variables and functions, like the location and rotation.

USARSim provides a basic laser sensor used by the robots in the simulation. We modified the laser sensor to detect the smoke *Actors* and to behave according the behavior seen in section 2 and section 3.3.

The implementation of the smoke Actors involves the visualization of the smoke and setting up collision boxes that can be detected by the laser sensor, while at the same time the robots can move through the smoke without colliding.

Two different types of smoke are implemented in UnrealScript: smoke areas and local smoke.

4.1 Smoke areas

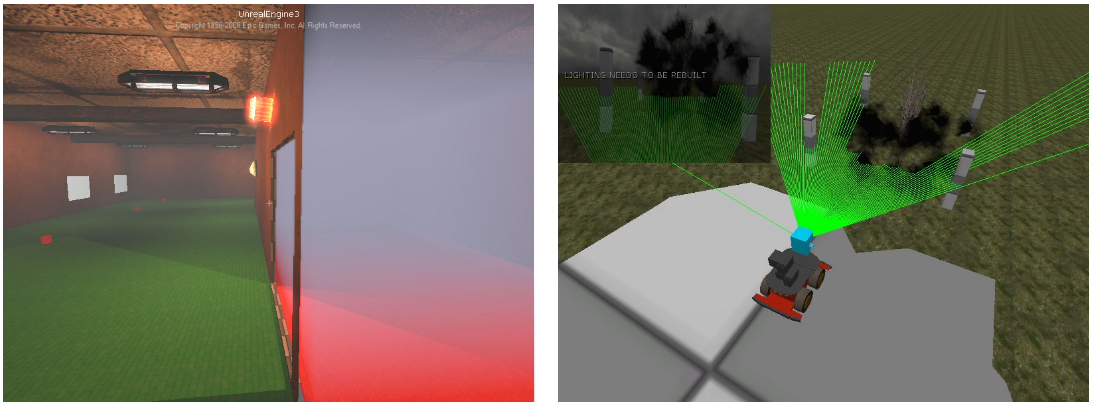
The first type of smoke is the *smoke area*, which fills an area with homogeneously distributed smoke (see Fig. 6a). Such an area is defined by a SmokeRegional Actor, which contains a polygon mesh to define its shape. The polygon mesh can take any shape, although it cannot move around. SmokeRegional Actors may overlap each other, causing the smoke densities to be added up. The Unreal engine uses the *density* parameter to control the density of smoke. This parameter has no equivalent in the real world. The Kismet Tool in the Unreal Editor can be used to dynamically manipulate the *density* of the smoke. This provides a method for changing the smoke density over time, as demonstrated in Fig. 7.

This type of smoke allows reproduction of the smoke that was created in the BOCAS training room.

4.2 Local smoke

The second type of smoke is *local smoke*, which emits smoke from a point source (see Fig. 6b). The smoke consists of many smoke particles with a set lifetime. The color and velocity of each particle change over time to make the smoke look realistic.

This is an useful method to create a dynamic smoke column.



(a) Smoke area (b) Local Smoke

Fig. 6: In the left the rooms are filled with smoke areas. On the right a setup with local smoke is shown.

The Unreal Editor does not support collision boxes for particles. UnrealScript was used to create a collision box for each particle. The collision box has a fixed size and follows the position of the particle it has been attached to. Creating a collision box for each particle causes many Actors to be spawned, which can be quite demanding for the engine. The spawning frequency and the lifetime of each particle have to be carefully balanced. Furthermore the exact position of the particle can not be retrieved directly from the Unreal Engine. By using a reasonable estimate the collision box moves along with the particle, although not at the exact same location. This slight divergence between the measurements of two sensors is not unrealistic (as seen from the complex dynamics in Fig. 3).

4.3 Laser Range Finder

Two laser range finders are implemented. The first is the Hokuyo which behaves as observed in the BOCAS dataset. The other sensor is the Sick LMS200, which behavior was modeled after the prior work.

A laser range finder returns error values when the smoke density is too high for a signal te return to the laser range finder. We define the penetration power of a laser as a measure for the distance the laser can travel through a certain smoke density. The Hokuyo laser range finder is just able to penetrate smoke which has a saturation value of 0.14 over 2.5 meters. In section 5, the *density* parameter in the simulation that corresponds to a saturation value of 0.14 is found to be 0.0028. The penetration power of the Hokuyo laser is defined as

$$0.0028d \times 2.5m = 0.007dm \tag{1}$$

where d is the unit for density in the simulation.

The original USARSim laser range finder implementation uses the built-in *Trace* function to detect which object is hit. The *Trace* function traces a path to find a collision in that direction. It returns the Actor that was hit and its location.

This behavior is augmented in two ways. The first behavior we implemented is when a *smoke area* was hit. The behavior is based on the model introduced in section 3.3. The laser range finder code has been modified to check if the hit Actor is a smoke Actor. If that is the case a new trace will be started inside the smoke volume, to find the distance the laser beam needs to travel through the smoke.

$$PowerConsumed = density \times distance \tag{2}$$

The $PowerConsumed$ is subtracted from the $PenetrationPower$. If the *penetration power* drops below zero the sensor will return error value zero. Otherwise a new trace will be started outside the smoke volume. The sensor may still yield an error value if more smoke is encountered (wherby the penetration power drops below zero) or the maximum range of the sensor is reached.

The second behavior we implemented is for the *local smoke*. The behavior is based on the observations from Fig 2b, i.e. the Sick laser range finder reacts to smoke that has high density similar to encountering a solid object. The Hokuyo LRF exhibits different behavior. As observed in section 3.3, this LRF always returns error values when smoke gets too dense. The Hokuyo in the simulation returns an error code for each beam that hits *local smoke*.

5 Validation

We have reconstructed the experiment setup inside a USARSim map. The differences between the virtual setup and the physical setup are minimized as far as possible. In Fig. 7, the saturation levels of section 3.1 are calibrated against the smoke densities used by Unreal. The saturation level of 0.14 (the threshold in our smoke model) is equivalent to 0.0028 unreal units.

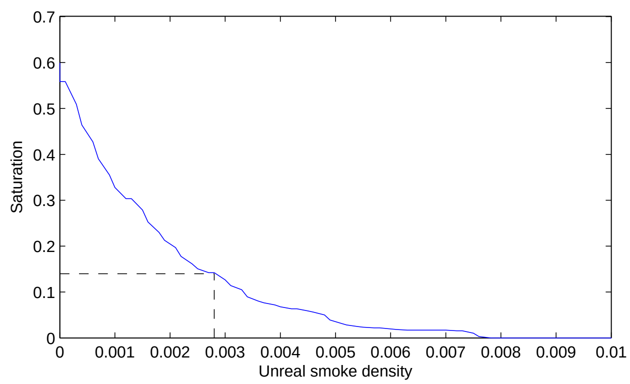


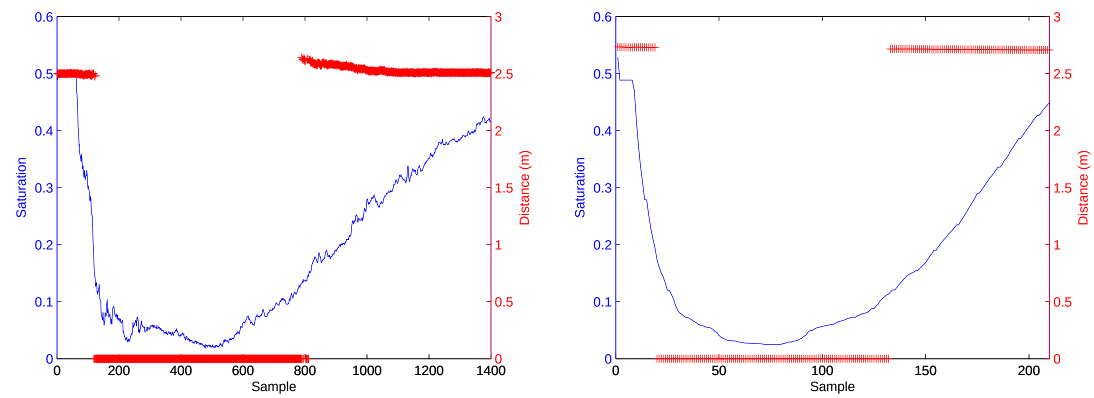
Fig. 7: Smoke density versus saturation in USARSim.

A virtual BOCAS training room was constructed with an equivalent experiment setup as used in the real experiment. Like the real experiment, the Hokuyo data was recorded to a log file (with a slightly lower frequency of 1 Hz). At the same time, automated screen captures were performed to create a video file, which was used to measure the saturation.

The first experiment is intended to validate the local smoke area model introduced in section 4.1. The room is filled with a puff of smoke, which is allowed to dissolve slowly.

Fig. 8: Reconstructed experiment in USARSim (left), physical experiment (right)

The visualization of the data generated by the physical experiment and the simulated experiment show equivalent behaviors. Both experiments do return error codes when the smoke density becomes too large for the laser to penetrate. The simulation results are smoother than the BOCAS results. This is due to the fact that real smoke is not completely homogeneous. The simulated experiment doesn't model the variance in distance measurements that are caused by variations in the smoke density. This behavior is visible in the physical experiment between samples 800 and 1000.



(a) BOCAS experiment

(b) Simulated experiment

Fig. 9: Saturation versus range measurements for the Hokuyo URG-)4LX

The second experiment is intended to validate the local smoke introduced in section 4.2. The Sick LMS 200 is in an open area with a number of obstacles in its field of view, equivalent to the experimental setup of Peynot and Scheding [11]. From the left smoke plumes are drifting into view, carried by a slow wind.

In Fig. 10 the recorded laser range finder data is plotted against time [8]. Note that for the Sick LMS200 the smoke is seen as an obstacle (in contrast to the error codes observed for the Hokuyo URG-04LX, which is not shown here).

6 Discussion

Our smoke implementation delivers visual smoke which interacts with the Hokuyo and Sick laser range finders. For the Hokuyo URG-04LX a model is implemented which returns error codes when it encounters smoke that has too high density. In our experiments we could not reproduce the behavior of detecting smoke as an obstacle, as observed by Pascoal et al. [10]. Our current hypothesis is that this behavior depends on the type of smoke.

Our experiment shows that the Hokuyo measurements also depend on the surface optical characteristics. These characteristics have influence on the measured distances and the smoke density that is required to trigger error measurements. Incorporating this knowledge into USARSim would be a great addition. USARSim supports material characteristics such as reflectivity, so implementing more complex behavior is probably feasible.

The experiments of both Pascoal and Peynot [11] showed that the Sick LMS200 only sporadically returns error values and in most cases returns the distance to the smoke plume. This behavior is implemented as the default behavior in the simulation of the Sick laser range finder. Additionally, the measured distance of the Sick seems to be a function of the saturation, with a certain penetration into the smoke plume for lower saturation levels. This behavior is not yet implemented.

[8] Movies of this experiment are available

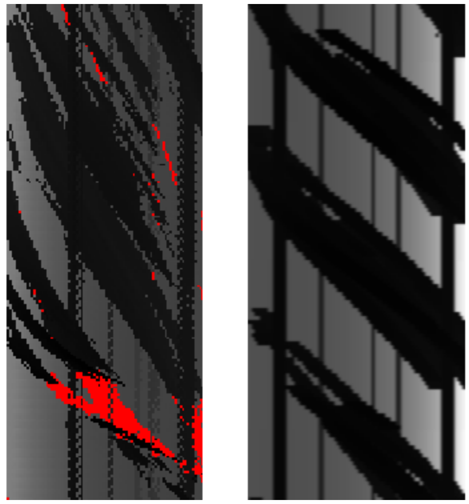


Fig. 10: Result: saturation and laser measurements over time for the Sick LMS200. Left real measurements from [11]. Right a reconstruction in simulation.

7 Conclusion

In this study the behavior of the Hokuyo URG-04LX and Sick LMS200 laser range finders in the presence of smoke are modeled and validated. The validation shows that the resulting behavior of the laser range finders is close to the behavior of laser range finders in the real world. Although there are some disparities between the simulation and the real world readings left, we feel that the level of detail is sufficient and in proportion with the realism of the rest of the simulation environment.

8 Acknowledgements

We thank fire department trainings center BOCAS for usage of their facilities for our experiment, and in particular Jan Dirk van de Ven for his useful comments and Tom Groenendijk for his time and aid. We would also like to thank José Pascoal for kindly sharing their recorded dataset. Finally, our research benefited greatly from the extensive dataset published by the Australian Centre for Field Robotics.

References

1. Murphy, R.R., Tadokoro, S., Nardi, D., Jacoff, A., Fiorini, P., Choset, H., Erkmen, A.M.: Search and Rescue Robotics. In: Handbook of Robotics. Springer (2008) 1151–1173
2. Jacoff, A., Messina, E.: Urban search and rescue robot performance standards: Progress update. In: Proceedings of the 2007 SPIE Defense and Security Symposium. Volume 6561, Unmanned Systems Technology IX. (2007) 65611L
3. Balakirsky, S., Falco, J., Proctor, F., Velagapudi, P.: USARSim Porting to Unreal Tournament 3. In: Proceedings of the International Conference on Intelligent Robots and Systems (IROS 2009), Workshop on Robots, Games, and Research: Success stories in USARSim, IEEE (2009) 74–79
4. Carpin, S., Stoyanov, T., Nevatia, Y.: Quantitative assessments of usarsim accuracy. In: Proceedings of PerMIS 2006. (2006)
5. Carpin, S., Lewis, M., Wang, J., Balakirsky, S.: Bridging the gap between simulation and reality (2006)
6. Schmits, T., Visser, A.: An omnidirectional camera simulation for the usarsim world. In: In Proceedings of the 12th RoboCup International Symposium, July 2008. Proceedings CD. To be published in the Lecture Notes on Artificial Intelligence series. (2008)
7. Pepper, C., Balakirsky, S., Scrapper, C.: Robot Simulation Physics Validation. In: Proc. of the Performance Metrics for Intelligent Systems (PerMIS) Workshop. (2007) 97–104
8. Balakirsky, S., Carpin, S., Visser, A.: Evaluation of the robocup 2009 virtual robot rescue competition. In: Proceedings of the 9th Performance Metrics for Intelligent Systems (PERMIS'09) workshop. (2009)

9. Boehler, W., Vicent, M.B., Marbs, A.: Investigating Laser Scanner Accuracy. In: Proceedings of the XIXth International Symposium, CIPA 2003 : new perspectives to save cultural heritage. Volume XXXIV 5/C15 of The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences. (2003) 696–701
10. Pascoal, J., Marques, L., de Almeida, A.T.: Assessment of laser range finders in risky environments. In: Proceedings of the EURON/IARP International Workshop on Robotics for Risky Interventions and Surveillance of the Environment. (2008)
11. Peynot, T., Scheding, S.: Datasets for the evaluation of multi-sensor perception in natural environments with challenging conditions. In: Euron GEM Sig Workshop on Good Experimental Methodology in Robotics at RSS. (2009)
12. Donate, A., Ribeiro, E.: Viewing scenes occluded by smoke. In: Advances in Visual Computing. Volume 4292 of Lecture Notes on Computer Science., Springer (2006) 750–759