The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Player/Stage Simulation of Olfactory Experiments

Gonçalo Cabrita, Pedro Sousa and Lino Marques

Abstract— **This paper describes a Player/Stage simulation framework for mobile robot olfactory experiments. The PlumeSim framework simulates odor transport in the environment, as well as olfactory detection systems. The odor transport can be based on analytical theoretical models, data generated by dedicated computational fluid dynamics (CFD) software packages, or data acquired in a real environment. The framework is validated through comparison of the results of simulated and real mobile robot olfactory experiments.**

I. INTRODUCTION

Olfaction is a key tool as a matter of survival for the majority of the biological organisms known to Man. In the animal kingdom olfaction is used to achieve goals such as communication, hunting, predator avoidance and even mating. However, olfaction has not always been popular among the mobile robot scientific community, a fact that has been changing in the past few years as the technological means evolved. In a mobile robot olfactory experiment odor plays a major role in the robot's decision making process. For this reason this type of experiment is highly dependent on odor transport and odor detection [1], [2], [3], [4]. From Figure 1 it is easy to understand that the process of developing an algorithm or task is unavoidably connected to the robot and environment entities. By providing a mean of abstracting from these two entities, efforts can be focused on the task. This is the purpose of a simulation framework. In a mobile robot olfactory experiment odor transport falls under the environment entity whereas odor detection falls under the robot entity. Thereby, the developed framework aims to address these two aspects.

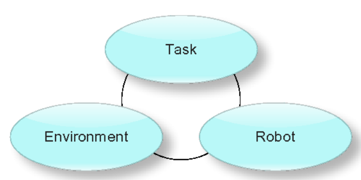


Fig. 1. Entities influencing the results of a mobile robot experiments.

Simulation is the process of modeling a real phenomenon. In the world of mobile robotics, as in many other fields, simulation is an important tool to investigate new ideas in a controllable environment as illustrated in [5], [6], [7]. The Player/Stage project began in 2001 as an open source project. Player is a robot device server that provides network transparent robot control, in a device independent way that is non-locking and language neutral. Stage is a lightweight, highly configurable robot simulator that supports large populations [8]. Since then the Player/Stage project was greatly enhanced [9], [10]. However, it still lacks, as most (or all) of the mobile robot software available nowadays, support for chemical plume simulation and chemical plume tracking. Player does not yet implement a dedicated interface capable of supporting a chemical sensor or anemometer. Furthermore, Stage has no support for plume simulation, an important tool in odor search algorithm development.

Computer fluid dynamics (CFD) software can be used to generate simulations of interactions between fluids and gases with surfaces. This type of software is part of the fluid mechanics area of study, where numerical methods and algorithms are used in order to model, solve and evaluate fluid transport problems. In the particular case of odor plumes [11], [12], the simulations are usually focused around the air flow, mass flow, turbulence and chemical reaction in order to determine the most real simulation data.

An odor plume can simply be described as a column of one fluid moving through another, most of the times moving away from its source and spreading along the available space. Usually, the plume shape is influenced by the flow of the environment fluid, the air in this case. The flow in the plume can be laminar or turbulent. This characteristic is usually related to the plume distance from the source. Hence, there is a transition from laminar to turbulent as the plume moves away from the source [13]. This phenomenon can be clearly identified in the rising column of smoke from a cigarette or from a factory chimney. Along with the plumes and their mechanisms of transportation, chemical cues represent a key aspect for both the detection of the chemical inside the plume and the behavior to be taken with such information. This means that the detection of chemical compounds is as important as the identification of the plume shape and characteristics. In air, fluid mechanics impinge directly upon the distribution of odorous molecules in time and space. Due to this fact, in turbulent plumes where odor is unequally distributed, the retrieved signal is highly intermittent [14], [15]. Air movement exits at any place, inside or outside enclosed spaces, and can take several shapes and have several behaviors. As the chemical molecules are transported by the air and are actually mixed with it, an odor plume is created. This means that the air plume behavior will affect the way the odor behaves inside it. This includes the air eddies, its own unpredictable path and also its irregular shape. This leads to the intermittence of the plume, caused by the irregular spreading pattern. This characteristic is responsible for the

G. Cabrita, P. Sousa and L. Marques are with Dept. of Electrical and Computer Engineering, Institute of Systems and Robotics, University of Coimbra, 3030-290 Coimbra, Portugal.
{goncabrita, pvsousa, lino}@isr.uc.pt

discrepancies in the chemical concentrations that can be measured at close points inside the plume.

This paper is organized as follows: a brief description of plume generation using mathematical models, experimental data and CFD software is given in section II. Section II also holds the description of the developed software. Section III introduces the experimental setup and section IV the results for the conducted experiments. The discussion of the results is presented in section V. Conclusions and future work related to this article are given in section VI.

II. PLAYER/STAGE OLFACTORY SIMULATOR

A. Plume Generation

This work focuses mostly on odor plume simulation, thus, a very important aspect is the plume generation. In a very simplistic way a plume can be generated using different methods ranging from directly applying a wide range of mathematical models available nowadays, the data provided by CFD software, to the data collected from real world experiments. In this section, the previously referred methods for plume generation were implemented and tested.

Despite the referred options, in CFD, plumes are usually modeled by three laws inherent to the flow of fluids: the conservation of mass, the conservation of momentum and the conservation of energy [16]. Along with these, the turbulent effect can also be included in CFD [17]. All together, these equations are known as Governing equations or, when related to real viscous flows, as Navier-Stokes equations. Hence, in CFD, the two dominant transport mechanisms of chemical species in the surrounding fluid can be described by the Convection - Diffusion Equation, presented in equation 1 where $C(x, y, t)$ is the concentration, $\vec{V}$ the fluid velocity field and D the diffusion coefficient. The first and second terms on the right side of equation 1 represent, respectively, the convection and diffusion contributions.

$$\frac{\partial\ C(x,y,t)}{\partial t} = -\vec{V} \cdot \nabla C(x,y,t) + D\nabla^2 C(x,y,t) \quad (1)$$

B. Mathematical Models

In the problem of plume generation one of the main areas of focus is the use of mathematical models which try to describe the physical behavior of plumes: the shape of the plume, both close and far from the source, and also the intermittency. As examples of these models one can refer for example the Gaussian or stochastic models. In this work two models were implemented and are included in the simulation framework: Gaussian and Meandering [18], [19] .

1) Gaussian Model: The Gaussian model is probably one of the simplest models which are commonly used to simulate an odor plume. The following parameters concerning the environment can be specified: the release point, the initial concentration of the source, the downwind distance, the time and the diffusivity constants. This allows for different odor plumes to be generated. Equation 2 presents the Gaussian expression that is responsible for data generation. The C denotes the concentration, Q the mass of the emission and

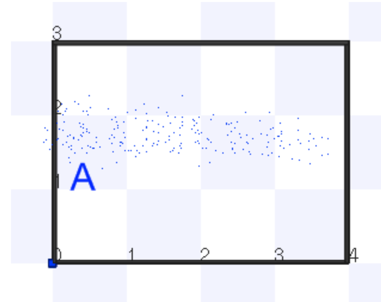


Fig. 2. Simulated plume based on the Gaussian model. A - odor source

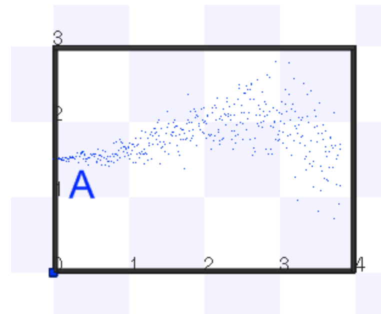


Fig. 3. Simulated plume based on the Meandering model. A - odor source

σ_x *and* σ_y the coefficient of diffusion in x and y (the standard deviation in the gaussian equation).

$$C(x,y,t) = \frac{Q}{4\pi\sqrt{\sigma_x\sigma_y}t} e^{\left(-\frac{1}{4t}\left(\frac{(x-x_0)^2}{\sigma_x} - \frac{(y-y_0)^2}{\sigma_y}\right)\right)} \quad (2)$$

The concentration at a specific point and time can be calculated using expression 2. As it can be seen the expression includes the time variable, making it possible to animate the Gaussian plume. Thus, this simulation incorporates the ability to replicate the changes of the plume over time. An example of this plume can be seen in Figure 2.

2) Meandering Model: This model specifies and approximates the behavior of a plume consisting of an odor source and an air mass. The odor molecules are released from a predefined point and are transported by the air flow which flows also in a predefined direction. The air flow changes among different vectors which moves the centerline of the plume, known as plume meandering. This feature is common in odor plumes and denotes the plume dispersion. Hence the generated plume will be similar to the one generated by the Gaussian model with the meandering component taken into account. The equation 3 describes the used method to calculate the concentration in (x, y) coordinate at time t. The $y_c(x, t)$ denotes the center of the plume, $w(x, t)$ the width and $h(x)$ the height of the plume, dependent on the distance x and time t; $\sigma_y(x, t) = w(x, t)/\sqrt{(2\pi)}$ and $\sigma_z = h(x)/\sqrt{(2\pi)}$.

$$C(x,y,t) = \frac{Q}{2\pi\sigma_y(x,t)\sigma_z(x)} e^{\left(-\frac{(y(t)-y_c(x,t))^2}{2\sigma_y^2(x,t)}\right)} \quad (3)$$

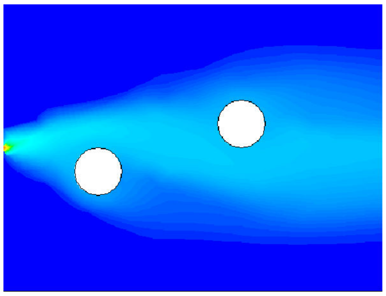


Fig. 4. Simulated plume using ANSYS Fluent software.

This simulation uses predefined parameters to calculate the plume centerline, width and dispersion to generate the odor plume data. The referred parameters which can be specified are the source strength, the wind characteristics, the diffusivity constants and also the downwind distance and time. The plume centerline is calculated based on both the downwind distance and time. The concentration in a specific point at a specific time can be calculated by means of a Gaussian expression, however the internal intermittence is not considered. Given the centerline of the plume and according to the configured parameters, the meandering plume data is generated. Thus, the odor plume data populates the centerline of the plume in a Gaussian pattern, taking the previously calculated dispersion shape. Figure 3 depicts a plume using the meandering model to generate data.

C. Specific Software Simulation Data

There is a wide selection of proprietary CFD software available today such as ”EasyCFD”, ”S&C Thermofluids”, ”AcuSolve”, ”ADINA” and ”ANSYS Fluent”, among others. Any CFD software with similar capabilities is prone to be used with the developed simulator, as long as it is able to output some sort of log file containing the simulation data, independent of format. In this work the ANSYS Fluent CFD software was used [20].

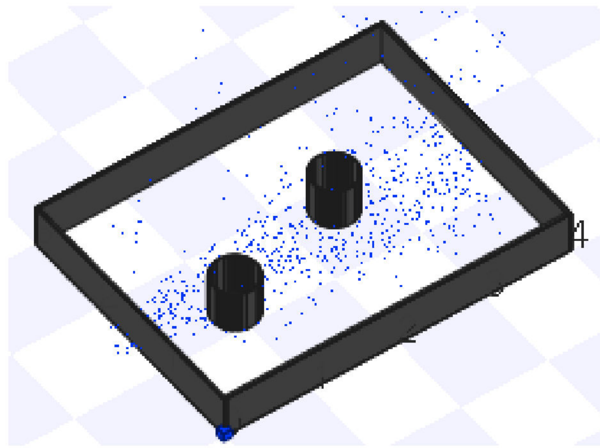
D. Experimental Data

An important feature of the developed simulator is the ability to playback recorded plumes from real world experiments. Plume data gathered from real plumes adds the intermittence effect to the simulator. Real plume data was collected through the setup later described in section III, and fed into the simulator.

E. The Software

The PlumeSim is a plume simulation framework. It is a plugin driver programmed in C++, designed to work with Player/Stage, bringing plume simulation into the world of mobile robotics. It is capable of feeding simulated chemical plumes into Player/Stage from a wide range of sources, from mathematical models to data generated by CFD software. It is also able to use data collected from real world experiments, enabling it to playback a recorded chemical plume over and over again, both on a simulated environment and robot as well as on a real robot. This poses as a great tool for

(a) Real arena.



(b) Simulated arena.

Fig. 5. Real arena and simulated arena with simulated plume generated in Fluent.

developing odor search algorithms, providing a smooth path of progression from a simulated robot and environment, to a real robot with a simulated plume and finally to a real experiment.

1) Software Breakout: The PlumeSim plugin driver for Player/Stage can be divided in two parts, the PlumeSim class, which holds the driver itself, and the utility classes, which hold the various simulation algorithms and log file parsing routines. The PlumeSim class basically acts as a mask for Player/Stage, making it possible to output the same type of data structures for various simulation methods. It is in charge of drawing the plume in Stage as well as providing the (real or simulated) robot with the simulated chemical sensor and anemometer readings. The PlumeSim works as illustrated in the flow chart in Figure 6.

The underlying utility set can be expanded to accommodate any kind of mathematical model or log file. Each utility must be able to perform two primary functions, generate the plume points and generate sensor readings. The purpose of each utility class is to convert its source data into data that the PlumeSim class can work with. There are two types of plumes in this simulator: static, which are plumes that do not change over time, and dynamic, which change over time. Since it is likely that a log file might hold data for less

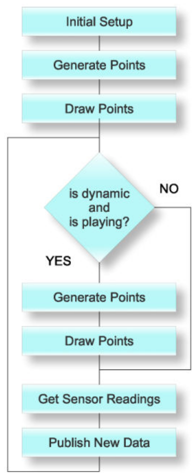


Fig. 6. PlumeSim flow chart.

simulation time than a robot takes for example to finish its odor search algorithm, once a set of data reaches the end the plume stays still. This is however a characteristic of the software that does not interfere with the proper use of the simulator, since a plume in a controlled environment tends to become stable over time. Moreover the rate of the data being played back from a log file can be speeded up or slowed down. Currently the utility set is composed by four sources of plume simulation:

- PSGaussian, based on the gaussian algorithm mentioned previously. This is a plume that changes over time. It can only be simulated for open spaces with no obstacles and only supports one robot at a time, since feeding the same object to multiple robots would generate a slightly different plume for each agent.
- PSMeandering, based on the meandering algorithm described above. It shares the same constraints described for the PSGaussian class. However this is a plume that does not change over time, always displaying a stable plume.
- PSLog, which allows to read a log file designed specifically for this application, enabling the driver to read data from any source previously converted to this format, including mathematical models, CFD log files, or data collected experimentally. This method allows for multiple robots since the same data can be replicated in multiple instances at the same time, as well as for multiple odor sources.
- PSFluent, which is similar to the PSLog class but is designed specifically to read Fluent log files on the fly. Otherwise it shares the same basic characteristics with PSLog. Fluent provides many options for log file output, so the correct format holding the required data should be used, as stipulated by the PlumeSim driver.

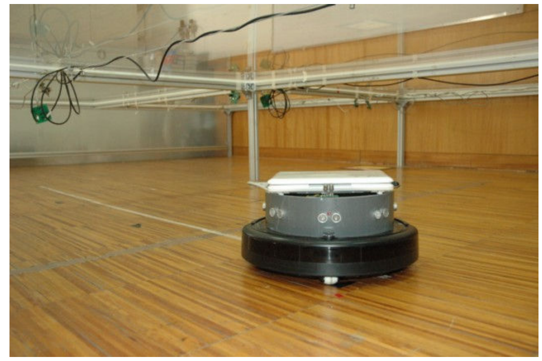


Fig. 7. The testing arena.

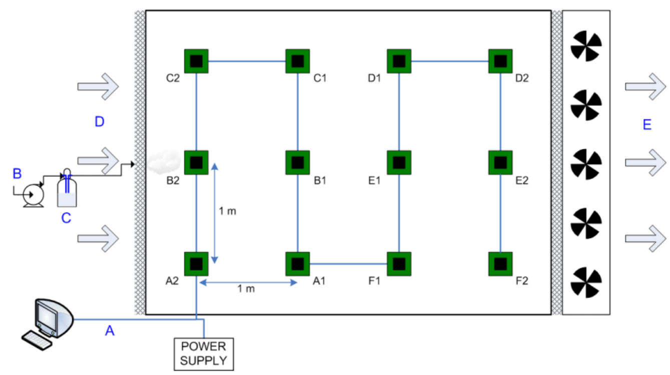


Fig. 8. Model of the experimental setup. A - RS485 network, B - Clean air, C - Ethanol, D - Air flow input, E - Air flow output

III. EXPERIMENTAL SETUP

All the desired characteristics and behaviors of the environment can be introduced in the experiment due to the available, fully equipped testing arena, designed specifically for the purpose of plume tracking and odor search algorithm development. The arena is a completely enclosed environment delimited by four walls where two of them are made of plastic with the shape of honeycombs. It includes an array of controllable fans allowing for the generation of different kinds of air flows. Along with the structure, the chemical source can be placed in several positions, not only in the boundaries of the environment but also at any point within its limits. For the experiment performed in the arena a constant air flow was generated in order to create homogeneous and laminar wind. Furthermore, ethanol vapor (C_2H_5OH) was used as odor.

Inside the arena there is a sensor network responsible for the data collection. Each sensor node is placed in a specific location and acquires the chemical and wind data during the experiments. This network is connected to a supervisor which controls all the actions taking place in the arena, including data gathering. The acquired data is then exported and the generated files can be processed by the PlumeSim simulator.

The mobile robot used in the experiments is an iRobot Roomba 560 equipped with an array of SRF08 sonars, an olfactory system, an anemometer and an Eee PC running Player. The olfactory system on the robot used the com-

Fig. 9. Roomba used in the real world experiments.

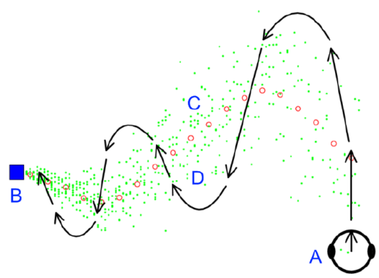


Fig. 10. Casting algorithm. A - start position, B - odor source, C - odor plume, D - robot path

mercial $TGS2620$[1] metal oxide gas sensors (MOX) [21] from Figaro, Inc.. These sensors are inexpensive, can detect oxidizing gases in the parts per million (ppm) range and have a response time from one to several seconds (which is appropriate to the dynamics of the Roomba mobile robot).

The odor search algorithm used for both the simulations and the real world experiments is a simple casting algorithm. The robot moves upwind while crossing the plume in a zig-zag movement, reacquiring the plume when it loses track of it, thus progressing to the source. This is illustrated in Figure 10. This algorithm was chosen due to its simplicity and documented effectiveness [22].

IV. RESULTS

A. Simulated Experiments

The first experiment was performed in Stage, running a simulated Roomba equipped with an array of sonars and a chemical sensor. The chemical plume for this simulation was generated in Fluent, designed to mimic a plume in the testing arena with no obstacles. The simulated robot performed the casting algorithm starting from the bottom right corner of the arena with the odor source located in the middle of the left wall. Figure 11 shows a screenshot of Stage running the simulation.

B. Hybrid Experiments

Following the simulated experiment in Stage a real robot was used to run the same casting algorithm in the testing

[1]ethanol is the chemical compound used in all the experiments and this sensor has high sensitivity to alcohol vapor

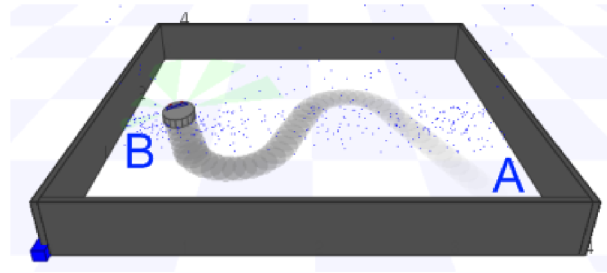


Fig. 11. Simulation of the casting algorithm in Stage. A - start position, B - odor source

arena free of obstacles. This robot was using the same simulated plume used in the previous simulation, the plume generated by Fluent. Once more the robot started from the bottom right corner of the arena, having the simulated odor source located in the middle of the left wall. Figure 13 shows the odometry of the robot measured during the hybrid experiment.

C. Real World Experiments

The third test consisted in a real world experiment with a real Roomba running the same casting algorithm as previously. Once again the robot started from the bottom right corner while the odor source was located in the middle of the left wall. The odor source is now releasing a real chemical plume and the robot is using its chemical sensor. The sensor network installed in the arena was used to monitor and collect the chemical data inside the testing environment throughout the experiment. Once the plume was stable the Roomba was allowed to run the casting algorithm. Figure 12 shows a graphical representation of the data collected by the sensor network of the stable plume, as well as the simulated plume in the stage environment.

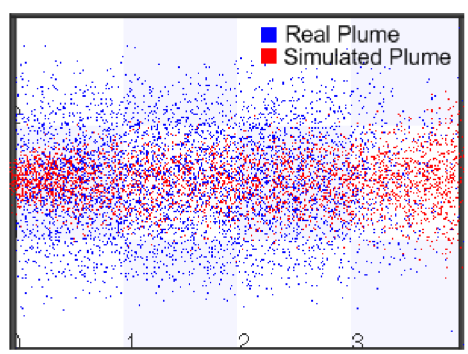


Fig. 12. The real and the simulated plume represented in the Stage environment.

V. DISCUSSION

The experiments performed showed that the PlumeSim framework is able to feed real-like simulations of chemical plumes into Stage. The plume used for the simulated experiment in Figure 11, also shown in Figure 12, is very similar to the plume measured during the real world experiment in Figure 12. Moreover, it allows for quicker development of

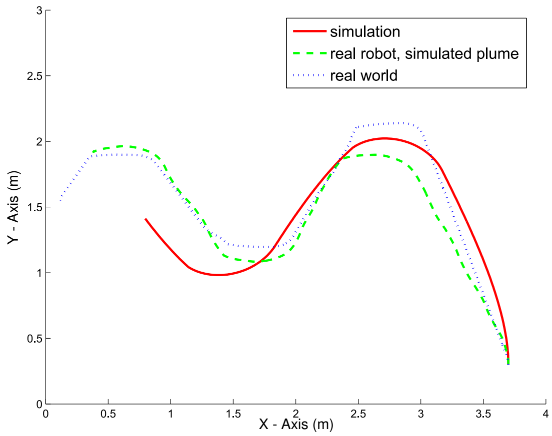


Fig. 13. Graphical representation of the odometry of the robot for real, hybrid and simulated tests.

odor search algorithms, since the debugging and testing of the software can be done in steps, starting from a simulated environment before progressing to real world experiments. Finally, the casting algorithm used in the simulated experiment was used in the real world experiment without the need for any changes. Furthermore all experiments had similar results. Figure 13 depicts the robot's odometry measured during a simulated experiment (Subsection IV-A), during a hybrid experiment (Subsection IV-B) and during a real world experiment (Subsection IV-C). The measured trajectories are very similar.

All the code developed for this project is available to the public[2], along with documentation and videos of the simulated and real world experiments.

VI. CONCLUSIONS AND FUTURE WORK

The PlumeSim framework performed as expected. It successfully provides simulated plumes to Player/Stage, enabling for easier mobile robot odor search algorithms development. This package is able to provide simple plumes, plumes with meandering and plumes with intermittence. Also, through the use of collected data or CFD software it is possible to incorporate obstacles into the simulations.

This was however the first step of PlumeSim. There is much room for improvement. Future work will focus on chemical sensor modeling, so that the data provided to the robot is not the data from the plume itself, but that which the desired sensor would provide in those conditions. This is a very important feature since the type of sensor used might have a large influence in the design of the odor search software. The most relevant properties of an artificial olfactory system are sensitivity, selectivity and speed of response. Moreover Player/Stage does not yet have specific interfaces for chemical sensors or anemometers. The PlumeSim package would greatly benefit from such interfaces, as would Player/Stage.

[2]http://embedded.deec.uc.pt/∼guardians/plumesim/

ACKNOWLEDGMENTS

This work was partially supported by European project GUARDIANS contract FP6-IST-045269.

REFERENCES

[1] L. Marques, U. Nunes, and A. de Almeida, "Olfaction-based mobile robot navigation," *Thin Solid Films*, vol. 418, no. 1, pp. 51–58, 2002, selected from 1st Int. School on Gas Sensors.

[2] L. Marques and A. de Almeida, "Mobile robot olfaction," *Autonomous Robots*, vol. 20, no. 3, pp. 183–184, 2006, editorial to Special Issue on Mobile Robot Olfaction.

[3] A. Loutfi, S. Coradeschi, A. Lilienthal, and J. Gonzalez, "Gas distribution mapping of multiple odour sources using a mobile robot," *Robotica*, vol. 27, no. 02, pp. 311–319, 2008.

[4] G. Kowaldo and R. Russell, "Robot odor localization: a taxonomy and survey," *Int. Journal of Robotics Research*, vol. 27, pp. 869–894, 2008.

[5] Z. Liu and T. Lu, "A Simulation Framework for Plume-Tracing Research," in *Australasian Conference on Robotics and Automation (2008: Canberra, ACT)*. version: http://www. araa. asn. au/acra/acra2008/contents. html, 2008.

[6] D. Zarzhitsky, D. Spears, D. Thayer, and W. Spears, "Agent-based chemical plume tracing using fluid dynamics," *Lecture Notes in Computer Science*, vol. 3228, pp. 146–160, 2005.

[7] H. Ishida, T. Nakamoto, T. Moriizumi, T. Kikas, and J. Janata, "Plume-tracking robots: A new application of chemical sensors," *The Biological Bulletin*, vol. 200, no. 2, p. 222, 2001.

[8] B. Gerkey, R. Vaughan, and A. Howard, "The player/stage project: Tools for multi-robot and distributed sensor systems," in *Proceedings of the 11th international conference on advanced robotics*, Coimbra, Portugal, 2003, pp. 317–323.

[9] R. Vaughan and B. Gerkey, "Really reusable robot code and the player/stage project," in *Software Engineering for Experimental Robotics*, ser. Springer Tracts on Advanced Robotics. Springer, 2007.

[10] R. Rusu, A. Maldonado, M. Beetz, and B. Gerkey, "Extending player/stage/gazebo towards cognitive robots acting in ubiquitous sensorequipped environments," in *ICRA Workshop for Networked Robot Systems, 2007, Rome, Italy*. IEEE, 2007.

[11] C. Chang, "Computational Fluid Dynamics Simulation of Concentration Distributions from a Point Source in the Urban Street Canyons," *Journal of Aerospace Engineering*, vol. 19, no. 2, p. 80, 2006.

[12] E. Lee, C. Feigley, and J. Khan, "An investigation of air inlet velocity in simulating the dispersion of indoor contaminants via computational fluid dynamics," *Annals of Occupational Hygiene*, vol. 46, no. 8, pp. 701–712, 2002.

[13] J. Farrell, J. Murlis, X. Long, W. Li, and R. Card, "Filament-based atmospheric dispersion model to achieve short time-scale structure of odor plumes," *Environ. Fluid Mech.*, vol. 2, pp. 143–169, 2002.

[14] N. Mole and C. Jones, "Concentration fluctuation data from dispersion experiments carried out in stable and unstable conditions," *Boundary-Layer Meteorol*, vol. 67, pp. 41–74, 1994.

[15] E. Yee, R. Chan, P. Kosteniuk, G. Chandler, C. Biltoft, and J. Bowers, "Experimental measurements of concentration and scales in a dispersing plume in the atmospheric surface layer obtained using a very fast response concentration detector," *J. App. Meteorology*, vol. 33, no. 8, pp. 996–1016, 1994.

[16] J. Anderson and J. Wendt, *Computational fluid dynamics*. McGraw-Hill New York, 1995.

[17] H. Versteeg and W. Malalasekera, *An introduction to computational fluid dynamics: the finite volume method*. Prentice Hall, 2007.

[18] M. Lof, L. Hemerik, and M. de Gee, "Chemical communication: does odor plume shape matter?" in *Proceedings of the Section Experimental and Applied Entomology - Netherlands Entomological Society*, vol. 18, 2007, p. 61.

[19] K. Mylne and P. Mason, "Concentration fluctuation measurements in a dispersing plume at a range of up to 1000 m," *Quart. J. Royal Meteorological Soc.*, vol. 117, pp. 177–206, 1991.

[20] Fluent, *Fluent User's Guide*, Fluent Inc., September 2006.

[21] J. Watson, "The tin oxide gas sensor and its applications," *Sensors and Actuators*, vol. 5, pp. 29–42, 1984.

[22] T. Lochmatter and A. Martinoli, "Tracking odor plumes in a laminar wind field with bio-inspired algorithms," in *Proceedings of the 11th International Symposium on Experimental Robotics 2008 (ISER 2008)*. Springer, 2009, pp. 473–482.