Design and Performance Evaluation of an Infotaxis-Based Three-Dimensional Algorithm for Odor Source Localization

Julian Ruddick, Ali Marjovi, Faezeh Rahbar, Alcherio Martinoli

Abstract— **In this paper we tackle the problem of finding the source of a gaseous leak with a robot in a three-dimensional (3-D) physical space. The proposed method extends the operational range of the probabilistic Infotaxis algorithm [1] into 3-D and makes multiple improvements in order to increase its performance in such settings. The method has been tested systematically through high-fidelity simulations and in a wind tunnel emulating realistic conditions. The impact of multiple algorithmic and environmental parameters has been studied in the experiments. The algorithm shows good performance in various environmental conditions, particularly in high wind speeds and different source release rates.**

I. Introduction

The localization of a gaseous chemical source is a crucial task in many situations such as finding casualties in natural disasters. Search And Rescue (SAR) operations is another application which requires finding gas leaks, explosives or drugs in residential area, airports and industrial facilities. SAR dogs are currently used for most of these missions, but are expensive to train and can become inefficient or distracted after a few hours of operation. Replacing SAR dogs by robots would potentially decrease the training time, maintenance price, and animal casualties.

The main challenge of using robots for Odor Source Localization (OSL) is that the structure of odor dispersion in the air is intermittent and time-variant, and does not have a smooth gradient [2]. As the flow carries patches of odor away from its source, the average concentration within a patch decreases and the average time between successive patches increases [2]. Therefore the probability of encountering an odor patch at any given point is determined by the relative location of the sensor to the odor source, the characteristics of the flow, and the shape of the environment and obstacles.

In robotics, the OSL problem is divided into three sub-problems (phases) [3], [4]: (i) *odor plume finding* refers to searching and sampling the environment randomly or systematically in order to find an initial cue of the plume (i.e. the first odor patch); (ii) *odor plume tracking* is the phase in which the robot attempts to approach the source while remaining in the plume; (iii) *odor source declaration* is the decision process of localizing an odor source in its close vicinity. During the first phase, the robot has not detected any odor patch in the environment and should perform standard robotic exploration/coverage strategies (e.g., random walk, casting, etc.) which are not specific to olfactory search. The third phase is usually formulated as a separate problem using other sensing modalities (e.g., vision). The second phase, plume tracking, is the main focus of this paper and most of the studies on OSL.

We classify previous works on plume tracking algorithms into four often overlapping categories [5]: gradient-based, formation-based, bio-inspired, and probabilistic algorithms. Gradient-based algorithms try to reach the source by climbing the concentration gradient using multiples samples taken at different positions in the environment. For these algorithms, the robot needs to move slowly to measure the long-term average of odor concentration at each sampling point due to the patchiness of odor plumes. Therefore, this type of algorithms, while being the most intuitive, needs relatively long time windows and large sample sets to find the source.

Formation-based algorithms are designed for multi-robot systems which sample the odor concentration in different positions at the same time. The robots share their observations and determine their relative position. The topology of the formation is adapted based on the observations and the whole group moves eventually towards the source [6]. In general, these methods have low computational and memory requirements, while they need at least two robots as well as inter-robot relative positioning and communication.

Bio-inspired algorithms try to imitate the searching behavior of various creatures found in nature such as moths, dogs, bacteria, dung beetles, crabs, etc. [7]. For example, in [8] and [9] the authors presented a moth-inspired algorithm called Surge-Cast, tested along with two other algorithms of the same class, namely Casting and Surge-Spiral, with one wheeled robot. These algorithms do not rely on any a priori information about the environment, its atmospheric conditions, or the historic observations, which makes them efficient for unknown areas. Nevertheless, due to low performance of current perception and locomotion technology compared to their biological counterparts, these algorithms are still far from being reliable in realistic environments.

Probabilistic algorithms model the source location as a probability distribution derived from the measurements made by the robot [5]. Each measurement updates the probability distribution of the source location using recursive Bayesian estimation. This process continues until the probability distribution of the source reduces to a Dirac function. Infotaxis [1], Hidden Markov Models [10] and Kernel methods [11], are the main examples of this category. While these algorithms can be computationally expensive, they can be adapted to

The authors are with the Distributed Intelligent Systems and Algorithms Laboratory, School of Architecture, Civil and Environmental Engineering, École Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland.

This work was funded by the Swiss National Science Foundation under grants 200021_153310/1 and 200020_175809/1.

various environmental conditions and the constraints of the underlying hardware.

While most of the previous works are based on wheeled robots in 2-D, recently, there has been some additional attention towards 3-D plume tracking, especially leveraging bio-inspired algorithms. In most cases, the methods are evaluated in simplified conditions, (e.g., [12] and [13]). Only a few have been carried out in realistic [14] or controlled environments (e.g., [15] and [16]), and all are based on simple gradient-based or bio-inspired algorithms. Recently a 3-D formation-based system has also been presented [6].

In this paper, we investigate a specific probabilistic algorithm: Infotaxis. Although the performance of Infotaxis has been evaluated until now mainly using simulated environments (e.g., [1], [17]–[19]) and in 2-D scenarios (e.g., [5], [20]–[22]), it is extensible to the third dimension. Eggels et al. [23] tested this algorithm in 3D in a simulated environment. One open question is therefore to understand how well such algorithms perform in the physical world, under different environmental conditions. This paper tackles the problem in 3-D, evaluating the performance of the extended Infotaxis algorithm in a wind tunnel under different environmental conditions. To the best of our knowledge none of the previous works has ever implemented a 3-D Infotaxis algorithm on real robots, nor has its performance been evaluated in various environmental conditions.

II. Infotaxis algorithm in 2-D

Infotaxis is a probabilistic odor source localization algorithm presented by Vergassola et al. [1] in a 2-D search framework. In each iteration of the algorithm, the robot takes a sample (i.e. odor measurement), evaluates multiple moves, and chooses the best option that yields to the lowest total entropy in terms of odor source localization. In other words, the algorithm aims to maximize the accuracy of information about the position of the odor source. This algorithm has been already implemented and realistically experimented in 2-D [5], [20], [21]. In this section, we explain the two repeating phases of this algorithm: update of the probability map and evaluation of the next move.

A. Probability map update

To estimate where the source is, the algorithm uses a grid map of the search area where every cell represents a possible position of the source. Each cell has a numerical value representing the probability of that cell being the source. This probability map is updated with every concentration measurement made. The map update depends on a gas distribution model (explained below) which is defined a priori. Initially, before any measurement is made, the map is filled with a uniform probabilistic distribution that sums up to 1. At each iteration, the cell with the highest probability is defined as the expected source position p_s.

1) Gas distribution model: Gas distribution models (AKA plume models) are numerical equations that estimate the concentration of gas in every location of interest, based on the position of the source, its release rate, the environmental conditions and the airflow characteristics. The model is the crucial tool that enables Infotaxis to update the probability map based on the measurements.

2) Measurements and map update: For every observation O_j, that is a concentration measurement made by the robot at location (x_j, y_j, z_j), the algorithm evaluates every cell of the probability map as if it was hosting the source S, by comparing the concentration that would be measured in that location according to the plume model with the actual measurement made. Hence, the probability map represents the probability of existence of the source at each cell i, given all the observations $O_{1,...,j}$. This probability is updated through recursive Bayesian inference:

$$P(S_j^i|O_{j,...,1}) = \alpha P(O_j|S_j^i)P(S_{j-1}^i|O_{j-1,...,1}) \quad (1)$$

where α is a normalization factor that assures the total probability of the map equals to 1. $P(O_j|S_j^i)$ is estimated based on the assumed plume model.

B. Move evaluation

The goal of the Infotaxis algorithm is to gain as much information as possible on where the source is, through each move. This is done using entropy as an indication of uncertainty associated with the probability map.

1) Entropy: Eq. (2) is applied to calculate the overall entropy of the map at every step, taking all the previous observations $(O_j, O_{j-1}, ..., O_1)$ into account, with i being the index of a cell, and n the total amount of cells.

$$H(S_j|O_{j,...,1}) = -\sum_{i=1}^{n} P(S_j^i|O_{j,...,1}) \times \log_2 P(S_j^i|O_{j,...,1}) \quad (2)$$

As entropy measures the uncertainty, it is maximum when the probability map is uniform, and decreases when a measurement narrows the possible source location to fewer cells with higher probability. It reaches the minimum value when one cell has 100% chance of being the source.

2) Action selection: At every step j, the robot can execute one action out of a discrete set of potential actions $A_{j,1}$, $A_{j,2}$, etc. Each action means moving to one cell among all the neighboring cells. Action $A_{j,i}$ takes the robot to a new location (x_i, y_i, z_i) where it will take a concentration measurement c_i, defining observation O_{j+1}. Each possible observation O_{j+1} occurs with some probability, represented by $P(C_{j+1} = c)$, i.e. the probability that a measured concentration C at the step $j+1$ will be equal to concentration c. It can be estimated using the plume model and the current belief of the source state, as following:

$$P(C_{j+1} = c) = \sum_{i=1}^{n} P(S_j^i) \times P(C_{j+1} = c|S_j^i) \quad (3)$$

For each potential action $A_{j,i}$, we have a discrete set of N potential observations O_{j+1} which are associated with a probability of actually happening, $P(O_{j+1}|A_{j,i})$, if the robot decides to take the corresponding action. The weighted entropy after carrying out action $A_{j,i}$ is given by:

$$H_A(A_{j,i}) = \sum_{N} H_O(O_{j+1,A_{j,i}})P(O_{j+1,A_{j,i}}|A_{j,i}) \quad (4)$$

The term $H_O(O_{j+1,A_{j,i}})$ denotes the expected entropy if O_{j+1} is observed after action A_j and is calculated by:

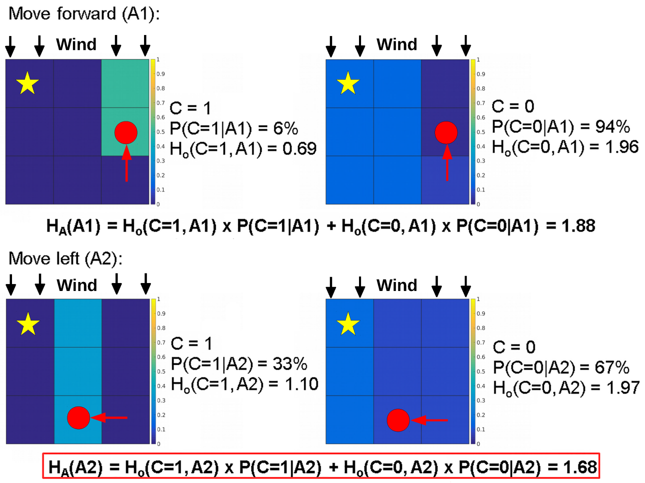


Fig. 1: A simplified example that represents the evaluation process a robot carries out to choose where to move next. In this simplified example, the considered plume model is a vertical strip and we consider only two possible concentration levels for the measurements (0 and 1). The red circle represents the robot, the star symbol shows a potential position of the source and the square grid is the probability map. The robot can move only along the grid main axes, either up or left, and chooses to move left as it yields a lower weighted entropy.

$$H_O(O_{j+1,A_{j,i}}) = H(S_{j+1,A_{j,i}}|O_{j+1,A_{j,i}}, O_{j,\ldots,1}) \tag{5}$$

Before calculating this entropy, the probability map should be updated for every potential observation O_{j+1}. Eq. (3) is used to calculate $P(O_{j+1,A_{j,i}}|A_{j,i})$.

The expected entropy is calculated for every possible action A by considering every possible observation O, and finally the action with the least weighted entropy is chosen (see Fig. 1 for a detailed example) and the whole evaluation process starts again from the new position.

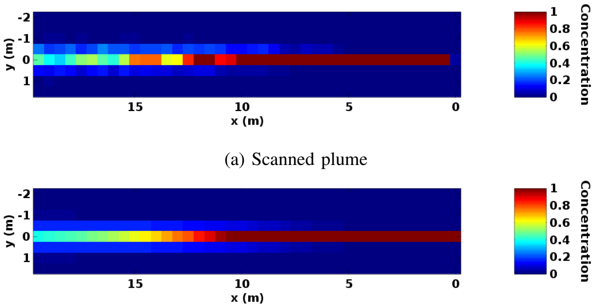
III. Proposed Method: 3-D Infotaxis

Adapting the original Infotaxis algorithm to 3-D odor source search required a few extensions. Firstly, we used a 3-D plume model to calculate the average concentrations and distribution of these concentrations to calculate $P(C_{j+1} = c|S_j^i)$ for Eq. (3). We then developed an adaptive moving strategy to define where the robot should move. The method to calculate $P(O_j|S_j^i)$ in Eq. (1) to update the probability map is taken from [5]. Concentration intervals were used for c in Eq. (3) and ending conditions were added to the algorithm.

A. Plume model

In most of the previous works on Infotaxis (including [1], [5], [17]–[21]) a simple 2-D model is used for odor propagation. Our first step towards a 3-D search algorithm is to use a proper 3-D gas distribution model. We propose to work with Eq. (6) which describes the 3-D pseudo-Gaussian concentration plume model for a continuous point source in a uniform flow with homogeneous turbulence, where Q is the release rate and $\bar{u}$ the average wind speed [24].

$$\bar{C}(x, y, z) = \frac{Q}{2\pi\bar{u}\sigma_y\sigma_z} \times e^{-\frac{y^2}{2\sigma_y^2}-\frac{z^2}{2\sigma_z^2}} \tag{6}$$



(a) Scanned plume

(b) Fitted plume.

Fig. 2: Figure (a) is an example scanned odor distribution obtained in a simulation setup. The robot has scanned the environment in a 3-D grid with cell size of 50 cm. The figures show the plane characterized by $z = 0$. Each cell observation is the mean of 1000 odor samples. Figure (b) shows the fitted plume resulting the plume model fitting method.

In this equation, the source is assumed to be positioned at the origin and the X-axis to be aligned with the direction of the airflow. σ_y and σ_z are the standard deviation of odor dispersion in the Y- and Z-axis respectively and are assumed to be linear functions of x.

B. Model calibration

To use the plume model in a real-world search, one needs to tune its parameters in a way that it suits the searched environment. The main parameters are σ_y and σ_z in Eq. (6), while Q and $\bar{u}$ are constant coefficients which do not affect the shape of the plume.

Model calibration is an important step that most previous works have neglected in their simulations. We have scanned and repeatedly sampled an experimental environment to obtain an odor map (see Fig. 2a). Multiple slices of the obtained odor map perpendicular to the X-axis were used to calculated the standard deviation in Y- and Z-axis. These samples are fit to obtain two linear models that define σ_y and σ_z as functions of x.

C. Distribution per average concentration

The plume model represents only the average concentration for each cell. However, in order to find the probability of measuring some concentration in a given cell (e.g., in $P(C_{j+1} = c|S_j^i)$ in Eq. (3)) the distribution of concentrations for each average concentration in the plume model is required. In many previous studies [25]–[27] the Gamma distribution is used for odor plume distribution.

This distribution is defined by two variables: the shape parameter k and the scale parameter θ. The mean of a Gamma distribution is the product of k and θ.

A Gamma distribution was fitted to the distribution of the data obtained for each cell of the scanned plume (see Fig. 2a). The parameter k obtained from these fits was then plotted against the mean (see Fig. 3) where a second order polynomial correlation curve was found. This correlation was subsequently used to calculate the cumulative density function of the Gamma distribution for each cell of the plume model based solely on the mean concentration of the cell.

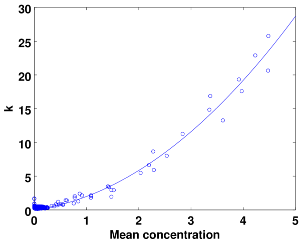


Fig. 3: Second order polynomial fit for parameter k of fitted Gamma distribution. Each point represents the Gamma distribution of one cell of the scanned plume (see Fig. 2a) when the 1000 odor measurement are separated into 50 averages of 20 measurements. A similar plot can be drawn for θ.

D. Targets and next move

The *targets* are the possible cells that the robot considers to move to, during the move evaluation step. For the conventional 2-D Infotaxis algorithm, the targets are the eight neighboring cells of the robot on the same plane (i.e. at the same height). Here we extend the targets to the 3-D space.

Two possible target strategies in 3-D are the *cross* strategy and the *cube* strategy. The former allows the robot to move to an adjacent cell in any XYZ direction, but only one direction at a time. This gives a maximum of six possible targets. The cube strategy allows the robot to move to an adjacent cell in any XYZ direction and also a combination thereof. This means that the robot can go diagonally and has therefore a maximum of 26 possible targets. Evaluating more targets at each iteration involves an increased computational cost, but allows for more freedom of movement. In this paper, we will experimentally evaluate the performance of the proposed 3-D Infotaxis algorithm using the two moving strategies mentioned above.

E. Adaptive step length

In all the previous Infotaxis works, the grid size (G) is fixed and the distance that the robot travels at every step is the same as the grid size. This implies that, even if the robot estimates that it has to travel a very long straight path, it still takes short steps in every iteration, making the whole run very time-consuming. This is usually the case when the entropy is not very high and the estimated location of the source is far away in upwind direction.

To increase the performance of the algorithm, when the selected target is along the wind direction, we have modified the step length to be an adaptive parameter as following:

$$Step\ length = \begin{cases} \max\{A|p_s - p_r|\ ,\ G\} & \text{if } H_t < \mu H_0 \\ G & \text{otherwise} \end{cases} \tag{7}$$

Based on this formula, when the entropy H_t at iteration t is lower than a threshold relative to the initial entropy H_0 of the map, the robot will move with a longer step length which is proportional to the distance between its position p_r and the expected source's position p_s. This allows for a quick approach towards the source when it is estimated to be far and shorter steps when closer to the source. In this paper we set $A = 0.1$ and $\mu = 0.9$.

F. Evaluated concentrations

As calculating the probability of obtaining all possible concentration values is not possible, we divide the continuous measurement field into a few intervals. This is defined by N in Eq. (4). Therefore, our algorithm evaluates the probability of obtaining a concentration interval from the plume distribution model defined in Section III-C. In this work, the limits were set to account for various levels of concentrations. The concentration used to update the probability map is the average of the interval.

G. Probability map update

The Gamma distribution explained in Section III-C is very accurate and suitable for the action selection phase. However, for the map update phase, it is computationally expensive to calculate $P(O_j|S_j^i)$ in Eq. (1) for every cell of the map at each iteration. Therefore, similar to the 2-D approaches (e.g., [5]), we estimate this probability factor from the following equation.

$$P(c = M) = e^{-c} \times c^M \tag{8}$$

In this equation, M denotes the measured concentration of O_j and c represents the expected concentration. This is a continuous function which is a specific case of the Gamma distribution with one maximum when $c = M$.

H. End of the algorithm

Most of the previous works on probabilistic OSL do not define an ending strategy for these iterative algorithms. Rodrigues et al. [22] improved the stop strategy of the Infotaxis by defining a criterion based on entropy. Built upon this work, we propose three ending conditions for the algorithm:

- The entropy is lower than a defined threshold H_{th1} relative to the initial entropy H_0.
- The highest probability cell remains the same for N_1 consecutive iterations while the entropy is lower than another defined threshold H_{th2}.
- The highest probability cell remains the same for N_2 consecutive iterations.

In this paper we empirically set $H_{th1} = 0.125 \times H_0$, $N_1 = 5$, $N_2 = 15$, and $H_{th2} = 0.25 \times H_0$.

IV. Performance Evaluation in Simulation: Impact of Algorithmic Parameters

There are two categories of parameters which influence the performance of the proposed 3-D Infotaxis algorithm: *algorithmic* and *environmental*. Algorithmic parameters are the variables which are modifiable in the algorithm, listed in Table I. To study their impact on the performance of the proposed method, we have tested all the combinations of the three sets of these parameters in simulation. Here we explain the details of these simulations.

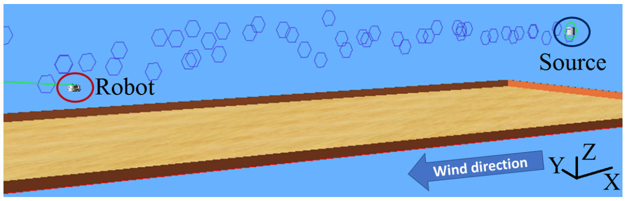


Fig. 4: Webots setup with the source upwind, the robot downwind and the plume represented with blue hexagons.

TABLE I: Algorithmic parameters studied in simulations

Parameter	Values
Grid size (m)	0.25, 0.5, 1
Number of targets	T6: 6 targets (cross), T26: 26 targets (cube)
Evaluated concentration intervals	C2: {[0,5e-4],[5e-4,10]}, C5: {[0,5e-4],[5e-4,1e-3],[1e-3,1],[1,3],[3,10]}, C10: {[0,5e-4],[5e-4,1e-3],[1e-3,0.02],[0.02,0.5], [0.5,1],[1,2],[2,3],[3,4],[4,5],[5,10]}

A. Simulation setup

For simulations, we used Webots [28], a high-fidelity submicroscopic robotic simulator. The simulated gas dispersion model is a plug-in implementation of a filament-based atmospheric dispersion model [2]. Using these tools, we have simulated a realistic odor plume in a large environment of 20 × 4 × 2 m^3 (see Fig. 4). A simulated Khepera III robot equipped with an odor board was used in the simulations. To enable the robot to move in 3-D, we disabled the gravity force in the simulations to emulate an airborne robot. While such emulation is obviously not faithful to the dynamic of an airborne robotic platform, it allows us to easily explore the impact of 3D sensing on the algorithm.

For each set of parameters, 100 runs are made, logging the positions of the robot, the final probability map, the computational time and the total simulation time. 100 random source positions in the upwind half of the environment are predefined and used for every set of variables.

A run is considered successful when, at the end, the highest probability cell of the probability map is at less than 0.9 m from the source. The fact that the robot gets close or not to the source is not a criteria for a successful run, even if the robot gets within 0.9 m.

B. Results

Fig. 5 plots the *success rate* of the proposed method by varying the targets and concentration threshold against the grid-size. The success rate is defined as the number of successful runs divided by the number of runs made. It can be seen that the runs evaluating only two possible concentration intervals are much less efficient than those considering finer intervals. However, there is no major difference between the runs evaluating 5 and 10 possible concentration intervals. Moreover, smaller grid size and finer moving strategies (cube) lead to better results, as expected.

Fig. 6 shows the *search time* (i.e. the actual time to find the source, assuming constant computational cost for running the algorithm) and the *distance overhead* (defined as the ratio of the travelled distance over the initial distance to the source on a straight path) for various target numbers and concentration intervals. Again, no significant difference between evaluating 5 and 10 concentration intervals can be seen. The main difference comes from the number of evaluated targets: the median time and distance for evaluating 26 targets is slightly lower than when evaluating 6.

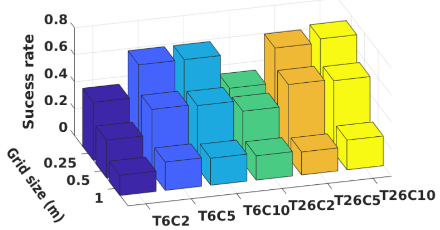


Fig. 5: Success rate of the algorithm in simulations having different values for grid size, targets (T) and number of evaluated concentration intervals (C). 100 simulation runs are done for each combination of the values given in Table I.

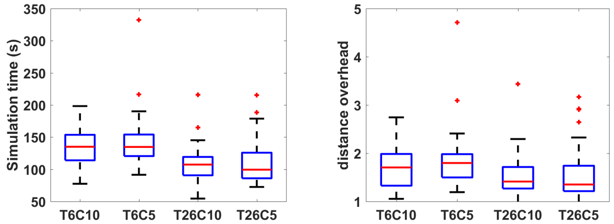


Fig. 6: The search time (left) and the distance overhead (right) of the successful runs of grid size 0.25 m, targets (T) 6 and 26 and evaluated concentration intervals (C) 5 and 10.

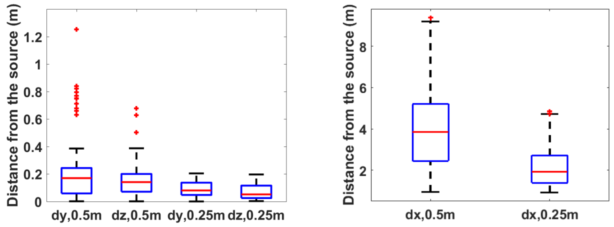


Fig. 7: The impact of grid size on the algorithm for failed runs.

To understand why some runs failed, we looked into the 125 failed runs for grid size 0.25 and 200 failed runs for grid size 0.5. The runs with only two concentration intervals were discarded because the success rate was too low. Fig. 7 shows the distance error in source location estimation in the failed runs. The vast majority of failed runs estimate the Y and Z coordinate of the source with acceptable accuracy and with an error smaller than the grid size. This implies that the cause of failure is mainly due to the estimate of the X coordinate. As can be seen in Fig. 7, a grid size 0.25 m is significantly better than a grid size 0.5 m.

Fig. 8 shows a possible explanation for these large errors. The hypothesis is that if the actual source position is far from any cell center of the probability map in the Y and Z directions, the robot receives lower concentrations and will have a tendency to overshoot the source position estimation. In these results, when the distance from the source in Y and Z direction is larger, the algorithm overshoots the source more frequently.

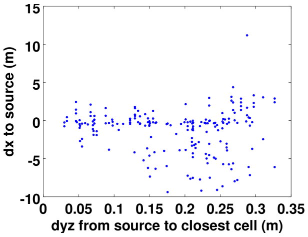


Fig. 8: Distance from the source in X direction versus the distance from the source to the closest probability map cell in the Y and Z direction for a grid size 0.5 m. The negative value of dx means that the source is overshot.

TABLE II: Environmental parameters studied in simulations and real-world experiments

Parameter	tested values
Wind speed (m/s)	0.2 (low), 0.9 (high)
Source release rate	simulations: 5% (low), 10% (high) real-world: 8% (low) and 18% (high)
Source height (m)	simulations: 0.9 (middle), 1.8 (high) real-world: 0.18 (middle), 0.38 (high)

TABLE III: Environmental condition setups

#	Wind Speed	Release Rate	Source Height	#	Wind Speed	Release Rate	Source Height
A	Low	Low	High	E	High	Low	High
B	Low	Low	Middle	F	High	Low	Middle
C	Low	High	High	G	High	High	High
D	Low	High	Middle	H	High	High	Middle

V. Performance Evaluation in Reality and Simulation: Impact of environmental conditions

In the previous section, we studied the algorithmic parameters of the 3-D Infotaxis algorithm in simulation. In this section, the main goal is to understand the impact of environmental conditions (e.g., wind speed) on the performance of this algorithm. Table II lists the environmental parameters and their values in our tests. We evaluated the influence of these external parameters on the performance of the 3-D Infotaxis algorithm in simulation as well as in real experiments. Each parameter is set to two different values, and the resultant combination of these values is eight setup conditions labeled in Table III.

In the simulations and real-world experiments of this section, the algorithmic parameters are set based on the results of the previous section's simulations. Therefore, for the simulation, a grid size of 0.25 m and 26 discrete options for the next move (cube strategy) were used, since they yielded a slightly faster discovery of the source. The number of evaluated concentration intervals was arbitrarily set to 10. In the real-world experiments, the same values were used, except that a grid size of 0.165 m was used because of the smaller movement space in Z axis in our experimental setup compared to the simulations.

In the experiments of this section, the source is centrally positioned at the most upwind position of the arena. The fitted plume model used for these tests is based on a scanned plume subject to the higher release rate and a wind-speed of 0.9 m/s (i.e. condition H in Table III). The model scales to the wind speed according to Eq. (6) but not to the release rate as this factor depends on the source, and cannot be known a priori. Similar to the previous tests, a run is considered successful when the estimated location of the source (i.e. the cell containing the highest value of the last probability map) is at less than 0.9 m distance from the source.

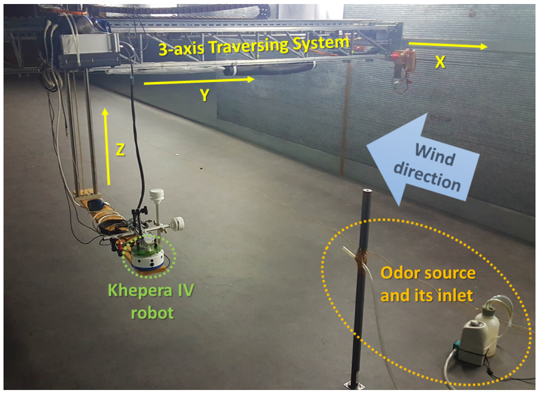


Fig. 9: Experimental setup in the wind tunnel.

For each set of parameters, 100 runs are carried out in the simulation and 10 in the wind tunnel, saving the positions of the robot, the final probability map, the computational time and the total time for each run.

A. Wind tunnel setup

To evaluate the performance of the algorithm in a repeatable fashion, our real-world experiments are carried out in a wind tunnel of volume 18 $\times$ 4 $\times$ 1.9 m^3, which provides a controllable laminar wind flow.

The odor source is emulated by an electric pump vaporizing liquid acetone in the air. The release rate is controlled by setting the power of the pump to a percentage of its maximum.

B. Robotic system

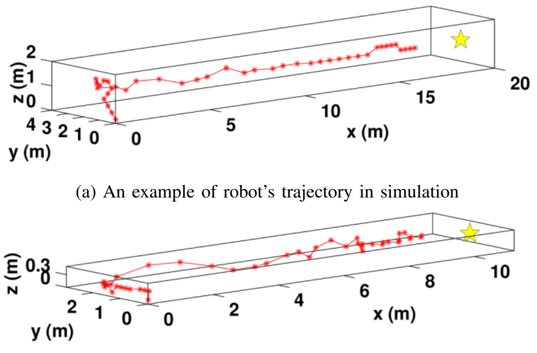
The wind tunnel is equipped with a controllable 3-axis traversing system. For the tests, the traversing system could move up to 11.06 m in the X direction, 3.30 m in the Y direction and 0.33 m in the Z direction. A Khepera IV robot mounted on this traversing system was equipped with a MiCS-5521 CO/VOC [1] capable of detecting volatile organic compounds and hydrocarbons. An external computer reads the odor measurement made by the Khepera IV robot and controls the traversing system. Fig. 9 presents this setup.

C. Results

Fig. 10 shows two examples of trajectories of the robot in simulation and in the wind tunnel. At the beginning of the run the algorithm makes the robot scan the crosswind section of the arena until it finds the plume. Then it moves upwind with a big step size. The step size decreases as the robot gets closer to the source.

Fig. 11 reports the success rate of the 3-D Infotaxis in simulation and physical experiments under the conditions reported in Table III. From these plots, we can see that:

[1] http://www.sgxsensortech.com



(a) An example of robot's trajectory in simulation

(b) An example of robot's trajectory in the wind tunnel

Fig. 10: Examples of robot trajectories made during successful runs of the environmental tests with condition H (see Table III). The initial position of the robot is at the left side of the figure and the yellow star represents the position of the source. Every red point represents an observation.

- When wind speed is increased, the success rate improves.
- The release rate does not have an impact on the success rate in the wind tunnel but it did have some influence in the simulation for low wind speed.
- Higher source positions were more challenging for the algorithm, especially at low wind speed.

Fig. 12 shows the search time and the distance overhead for the proposed algorithm in simulation and reality. In these results, the different environmental conditions are defined by Table III.

The results in Fig. 12 show that the algorithm finds the source faster when the wind speed is high. In simulation at low wind speed, the algorithm is faster at high release rate but in the wind tunnel it does not appear to produce a significant difference.

Fig. 13 presents the distance error between the actual position of the source and its estimated one, for the 8 different setups listed in Table III. These results show that:

- In low wind speed and low release rate, the algorithm undershoots the source in X direction at greater distances in the simulation compared to the reality. A possible explanation for this fact could be that the simulated release rate decreases more quickly over distance from the source than in reality.
- In the wind tunnel, the estimated position of the source usually shows a negative bias in the Y axis. The explanation to this is that the real plume in our wind tunnel is not straight along the tunnel and slightly deviates in the Y direction towards one side.
- In the wind tunnel, the height of the highest sources is almost always estimated one grid size too low (see Fig. 13f). This could be explained by the plume being angled slightly towards the ground particularly for the highest sources.

To compare the performance of our algorithm with other methods, as mentioned in the introduction, there are not many previous works studying the problem in 3D, and among them, only our previous work in [16] (a moth-inspired algorithm)

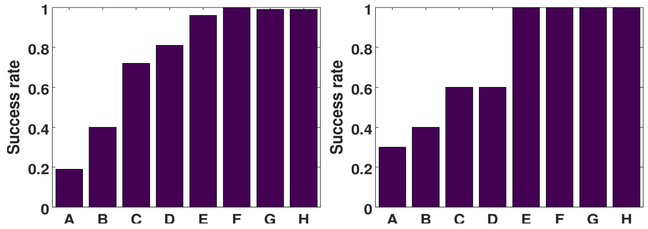
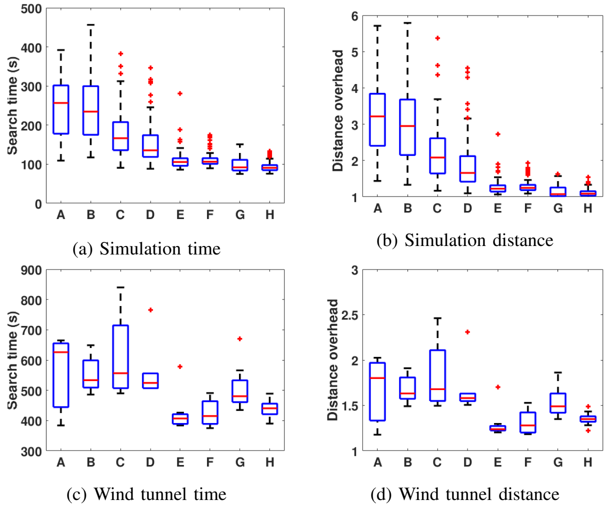


Fig. 11: Success rates for the environmental parameter tests: (left) simulation, (right) real experiments. The X-axis labels represent various environmental conditions defined in Table III.



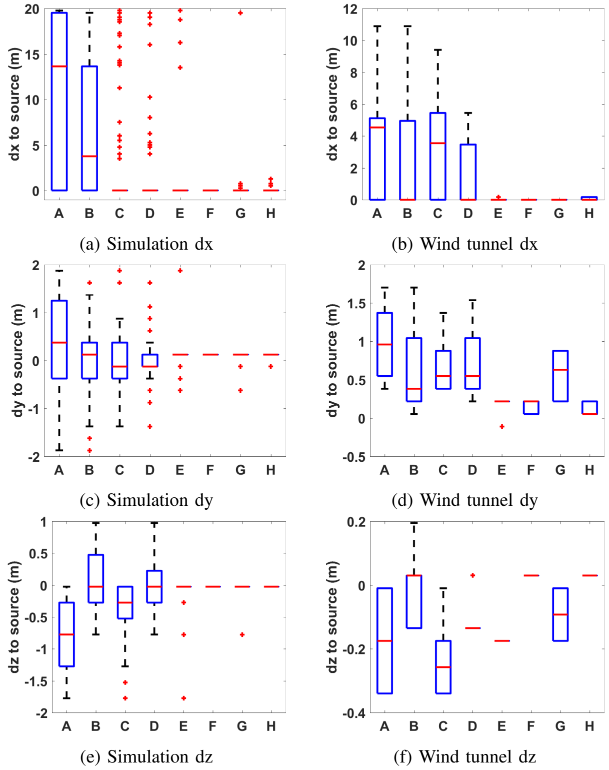
(a) Simulation time

(b) Simulation distance

(c) Wind tunnel time

(d) Wind tunnel distance

Fig. 12: Distance overhead and time comparison for successful runs in Webots and in the wind tunnel.



(a) Simulation dx

(b) Wind tunnel dx

(c) Simulation dy

(d) Wind tunnel dy

(e) Simulation dz

(f) Wind tunnel dz

Fig. 13: Source localization error along the three axis (rows) in simulation (left column) and reality (right column) for all successful and unsuccessful runs in environmental conditions of Table III.

has comparable environmental conditions and experimental setup. Compared to [16], our new method has a better over-all success rate, especially at higher wind speeds (e.g., in setup E, the 3D infotaxis shows 100% success rate, whereas in [16] it was 70%). It is relatively slower in finding the source, due to the concentration sampling time at every step, but this difference is not very significant.

VI. Conclusion

We successfully developed a 3-D odor source localization algorithm based on the Infotaxis algorithm. The algorithm was evaluated in simulation and in a wind tunnel with different algorithmic (i.e. grid size, number of target points, and number of considered concentrations) and environmental parametrizations (i.e. wind speed, source release rate, and source height).

The proposed algorithm showed to be successful in many tested conditions. For the algorithmic parameters we conclude that (i) evaluating more levels of concentration increases the performance of the proposed algorithm, to some degree, (ii) considering 26 target points is significantly better than only 6 points, and finally, (iii) smaller grid size yields a smaller potential distance between the source and center of a cell, leading to smaller source localization error. For the environmental parameters, we saw that (i) higher wind speed increases the performance of the proposed algorithm, (ii) higher source release rate also improves the performance especially in low wind speeds, and finally (iii) in high wind speeds, even when the source position is high with respect to the probability map, the success rate is at maximum and the execution time and the distance travelled are satisfactory.

The main drawback of the algorithm is its dependence on prior information, such as a plume model and an environment map. However, the algorithm seems robust to the imperfections of the plume model, since one single plume model, calibrated in one particular condition, yielded satisfactory results in a variety of environmental conditions. In the future, we are planning to study the impact of the accuracy of the plume model as well as that of obstacles on the performance of the algorithm. In terms of experimental setup, we plan to run the algorithm using a UAV in a semi-controlled environment.

References

[1] M. Vergassola, E. Villermaux, and B. I. Shraiman, "Infotaxis as a strategy for searching without gradients," *Nature*, vol. 445, no. 7126, pp. 406–409, 2007.

[2] J. A. Farrell, J. Murlis, X. Long, W. Li, and R. T. Cardé, "Filament-Based Atmospheric Dispersion Model to Achieve Short Time-Scale Structure of Odor Plumes," *Environmental Fluid Mechanics*, vol. 2, no. 1, pp. 143–169, 2002.

[3] R. A. Russell, D. Thiel, R. Deveza, and A. Mackay-Sim, "A robotic system to locate hazardous chemical leaks," in *IEEE Conf. on Robotics and Automation*, vol. 1, 1995, pp. 556–561.

[4] A. T. Hayes, A. Martinoli, and R. M. Goodman, "Distributed odor source localization," *IEEE Sensors Journal*, vol. 2, no. 3, pp. 260–271, 2002.

[5] T. Lochmatter, "Bio-inspired and probabilistic algorithms for distributed odor source localization using mobile robots," *PhD Thesis 4628, EPFL*, 2010.

[6] J. M. Soares, A. Marjovi, J. Giezendanner, A. Kodiyan, A. P. Aguiar, A. M. Pascoal, and A. Martinoli, "Towards 3D distributed odor source localization: an extended graph-based formation control algorithm for plume tracking," in *IEEE/RSJ Int. Conf. on Robots and Systems*, 2016, pp. 1729–1736.

[7] G. Kowadlo and R. A. Russell, "Robot odor localization: a taxonomy and survey," *The Int. Journal of Robotics Research*, vol. 27, no. 8, pp. 869–894, 2008.

[8] T. Lochmatter and A. Martinoli, "Simulation experiments with bio-inspired algorithms for odor source localization in laminar wind flow," in *Int. Conf. on Machine Learning and Applications*, 2008, pp. 437–443.

[9] ——, "Tracking odor plumes in a laminar wind field with bio-inspired algorithms," in *Int. Symp. on Experimental Robotics, Springer Tracts in Advanced Robotics*, vol. 54, 2009, pp. 473–482.

[10] J. A. Farrell, S. Pang, and W. Li, "Plume mapping via hidden markov methods," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 33, no. 6, pp. 850–863, 2003.

[11] M. Reggente and A. J. Lilienthal, "The 3D-Kernel DM+ V/W algorithm: Using wind information in three dimensional gas distribution modelling with a mobile robot," in *IEEE Sensors*, 2010, pp. 999–1004.

[12] R. A. Russell, "Tracking chemical plumes in 3-dimensions," in *IEEE Conf. on Robotics and Biomimetics*, 2006, pp. 31–36.

[13] B. Gao, H. Li, W. Li, and F. Sun, "3D Moth-inspired chemical plume tracking and adaptive step control strategy," *Adaptive Behavior*, vol. 24, no. 1, pp. 52–65, 2016.

[14] P. P. Neumann, V. H. Bennetts, A. J. Lilienthal, M. Bartholmai, and J. H. Schiller, "Gas source localization with a micro-drone using bio-inspired and particle filter-based algorithms," *Advanced Robotics*, vol. 27, no. 9, pp. 725–738, 2013.

[15] S. Edwards, A. J. Rutkowski, R. D. Quinn, and M. A. Willis, "Moth-inspired plume tracking strategies in three-dimensions," in *IEEE Int. Conf. on Robotics and Automation*, 2005, pp. 1669–1674.

[16] F. Rahbar, A. Marjovi, P. Kibleur, and A. Martinoli, "A 3D Bio-inspired Odor Source Localization and its Validation in Realistic Environmental Conditions," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2017, pp. 3983–3989.

[17] J. Masson, M. B. Bechet, and M. Vergassola, "Chasing information to search in random environments," *Journal of Physics A: Mathematical and Theoretical*, vol. 42, no. 43, p. 434009, 2009.

[18] A. R. G. Ramirez, A. L. Lopez, A. B. Rodríguez, A. D. C. de Albornoz, and E. R. De Pieri, "An infotaxis based odor navigation approach," in *ISSNIP Biosignals and Biorobotics Conference (BRC)*, doi:10.1109/BRC.2011.5740683 (6 p), 2011.

[19] Z. Siqi, C. Rongxin, and X. Demin, "Effectiveness of infotaxis algorithm for searching in dilute conditions," in *31st IEEE Chinese Control Conf.*, 2012, pp. 5048–5053.

[20] N. Voges, A. Chaffiol, P. Lucas, and D. Martinez, "Reactive searching and infotaxis in odor source localization," *PLOS Computational Biology*, vol. 10, no. 10, p. e1003861, 2014.

[21] E. M. Moraud and D. Martinez, "Effectiveness and robustness of robot infotaxis for searching in dilute conditions," *Frontiers in Neurorobotics*, vol. 4, 2010.

[22] J. D. Rodríguez, D. Gómez-Ullate, and C. Mejía-Monasterio, "On the performance of blind-infotaxis under inaccurate modeling of the environment," *The European Physical Journal Special Topics*, vol. 226, no. 10, pp. 2407–2420, 2017.

[23] A. Eggels, R. Kunnen, B. Koren, and A. Tijsseling, "Infotaxis in a turbulent 3d channel flow," *Journal of Computational and Applied Mathematics*, vol. 310, pp. 44–58, 2017.

[24] S. P. Arya, *Air Pollution Meteorology and Dispersion*. Oxford University Press, 1999.

[25] R. J. Brown and R. W. Bilger, "An experimental study of a reactive plume in grid turbulence," *Journal of Fluid Mechanics*, vol. 312, pp. 373–407, 1996.

[26] E. Ferrero, L. Mortarini, S. Alessandrini, and C. Lacagnina, "Application of a bivariate gamma distribution for a chemically reacting plume in the atmosphere," *Boundary-Layer Meteorology*, vol. 147, no. 1, pp. 123–137, 2013.

[27] I. J. Park, A. M. Hein, Y. V. Bobkov, M. A. Reidenbach, B. W. Ache, and J. C. Principe, "Neurally encoding time for olfactory navigation," *PLOS Computational Biology*, vol. 12, no. 1, pp. 1–16, 2016.

[28] O. Michel, "Cyberbotics ltd. webots TM : Professional mobile robot simulation," *Int. Journal of Advanced Robotic Systems*, vol. 1, pp. 39–42, 2004.