Authors' accepted manuscript: IEEE Robotics and Automation Letters (RAL), 2021.
The final publication is available at: http://dx.doi.org/10.1109/LRA.2021.3057290

Information-driven Gas Source Localization Exploiting Gas and Wind Local Measurements for Autonomous Mobile Robots

Pepe Ojeda*, Javier Monroy and Javier Gonzalez-Jimenez

***Abstract*— Gas source localization (*GSL*) by an olfactory robot is a research field with a great potential for applications but also with numerous unsolved challenges, particularly when the search must take place in realistic, indoor environments that feature obstacles and turbulent airflows.**

In this work, we present a new probabilistic GSL method for a terrestrial mobile robot that revolves around the propagation of local estimations throughout the environment. By exploiting the geometry of the environment as the basis for this propagation, we avoid relying on analytical dispersion models, eliminating the need to assume controlled environmental conditions.

Simulated and real experiments are presented in different indoor environments featuring multiple rooms and turbulent flows, demonstrating the suitability of our approach for locating the emitting gas source.

***Index Terms*— Autonomous Agents, Probabilistic Inference, Environment Monitoring and Management, Robotic Olfaction, Gas Source Localization.**

I. Introduction

Gas Source Localization (GSL) is the problem of determining the coordinates of the point from which a gas (more generally, a *volatile substance*) is being released. It is a problem with numerous practical applications, such as the control of accidental emissions of dangerous gases or the detection of illegal substances [1].

Among the multiple strategies proposed to tackle this problem, Mobile Robotic Olfaction (*MRO*) offers an interesting perspective by proposing the use of gas sensing devices (usually referred to as *electronic noses* or *e-noses* [2], [3]) to endow a mobile robot with gas detection capabilities. The use of autonomous mobile robots to monitor gases and locate their sources has been extensively studied in recent times [4], [5], and shows promising results despite still facing many unsolved challenges.

Two are the main issues that must be faced when estimating the location of a gas source based on measurements taken by a mobile robot. The first problem is related to the complexity of gas dispersion phenomena, particularly in realistic indoor environments, where the presence of obstacles and a turbulent airflow heavily influence the way gases disperse. Most state-of-the-art GSL methods are designed to work in open spaces[4], with no barriers to the wind transportation of volatile substances – conditions under which the gas dispersion can accurately be described by computationally feasible analytical models. Because of the higher complexity of the dispersion in indoors environments, their performance under those conditions is quite unsatisfactory and the search often fails [6], [7].

Authors are with Machine Perception and Intelligent Robotics Group, System Engineering and Automation Department, and Biomedical Research Institute of Málaga (IBIMA), University of Málaga, Campus de Teatinos, 29071, Málaga, Spain. ojedamorala@uma.es

The second problem is related to the available sensory information, particularly the scarcity of the data that can be gathered by the robot. The usual setup for a single mobile robot consists of a bi-dimensional anemometer and an array of gas sensors (*e-nose*), which provide the wind vector and the gas concentration for one specific point of the environment at each time instant. Such limited information implies that, even if the environmental conditions allow for the use of predictive methods, there will be a significant degree of uncertainty in any generated prediction.

In the present work we propose a probabilistic GSL algorithm designed to work under the challenging conditions of realistic indoor environments. Because of the previously discussed problems, the algorithm we propose is not based on the use of an analytical gas dispersion model, but instead on using the geometry of the environment to heuristically estimate the trajectory followed by detected gas patches. To perform this estimation, we assume that an airflow strong enough to cause advection exists, although we do not attempt to analytically model this effect.

We use a lattice of cells to model the connectivity of distant areas of the map. The estimations for cells in the vicinity of the robot position are generated from local wind measurements, and then these short-range estimations are propagated through unoccupied cells as a way to evaluate the probability of far away cells of containing the gas source.

These estimations are then used to guide the movements of the robot during the search. We propose and compare two different strategies for this purpose, a naive best-first approach that always navigates towards the areas with the highest estimated probability of being the source; and an infotactic search that attempts to maximize the information gain in every step by also estimating what measurements would be obtained if the robot moved to unexplored locations.

II. Related Work

In this section we briefly review some of the most notable approaches to GSL, discussing their main advantages and limitations. For the sake of convenience, we have divided these into two main groups: reactive algorithms, and estimation-based (predictive) algorithms. This is by no means an exhaustive taxonomy, but merely serves to put our proposal into context. For more detailed state-of-the-art surveys see [4], [8].

Authors' accepted manuscript: IEEE Robotics and Automation Letters (RAL), 2021.
The final publication is available at: http://dx.doi.org/10.1109/LRA.2021.3057290

A. Reactive Algorithms

Reactive GSL is a popular approach that reduces the search to instantaneous navigation decisions. The objective of these algorithms is to have the robot physically reach the gas source, not to infer its location from the measurements.

These strategies are usually based on moving in the general direction of the source, without trying to determine its exact location. Usually, the direction of the wind [9] or the gas concentration gradient [10] are used to guide the movements of the robot on a moment-to-moment basis, according to the specific movement pattern dictated by the algorithm. The behavior of certain insects has been an important source of inspiration for the design of these movement patterns [8].

Beyond the simplicity of reactive approaches, they come with some limitations. One of the main problems is the fact that, since no estimations of the source location are generated, these algorithms do not offer a way to perform source declaration, and would require the use of additional techniques for that purpose. Also, the lack of any form of memory can lead to behavioral loops, causing the search to be slower or even fail.

B. Estimation-Based Algorithms

Algorithms based on probabilistic inference offer a more sophisticate approach to source location, trying to exploit sensory information not only to guide the movements of the robot, but also to deduce the state of the environment [11]. By doing so, it is possible to actually offer candidate solutions and measurable certainty, which greatly simplifies the problem of source declaration. Maintaining probabilistic estimations based on the history of acquired observations also serves as a sort of indirect memory, allowing decisions to be made in accordance with all the previously obtained measurements.

Despite these advantages, the development of such strategies is more complex, since they require some sort of observation model to generate estimations from sensory information. It is often the case that these observation models are derived from analytical dispersion models [11], [12]. This approach presents two main problems: (1) models that offer accurate results for complex environmental conditions are very computationally expensive, and (2) in order to infer a source location from these models, some of the conditions of the environment (such as the release rate of the source) need to be known. If specific dispersion patterns are assumed, the latter can be tackled by dynamically estimating the necessary parameters, through what is usually referred to as Source Term Estimation (*STE*) [13], [14]. The former poses a more fundamental problem, since the existence of obstacles or a turbulent airflow means it would be necessary to use Computational Fluid Dynamics (*CFD*) simulations to achieve acceptable results [15], and these are so computationally expensive that running them on-line is by no means feasible. A common compromise is to make assumptions about the environmental conditions that allow for the use of a computationally lighter analytical models [16], but these can run into problems when said assumptions are not met.

The use of pre-computed simulations as the basis for the observation model has been proposed as well [17] to avoid the previously mentioned problems. It is a reasonable solution for environments with a limited set of known possible configurations (windows that can be opened or closed, specific potential points of gas release, *etc*.), but requires extensive preparation before they can be applied, and needing to run multiple simulations means it runs into scalability issues when the number of possible environmental configurations is large.

An interesting alternative observation model was proposed by Li *et al.* [18], where the source location is inferred by estimating the path followed by gas patches before reaching the robot. This approach avoids the use of a gas dispersion model but, because of the limited sensory information, still relies on making assumptions about the environmental conditions (*i.e.* homogeneous airflow) which are only acceptable in open spaces. Our approach attempts to avoid these problems, in particular the requirement of controlled environmental conditions, by limiting the use of sensory information to generating short-range estimations, and then exploiting the knowledge about the geometry of the environment to estimate the source probabilities at far away areas.

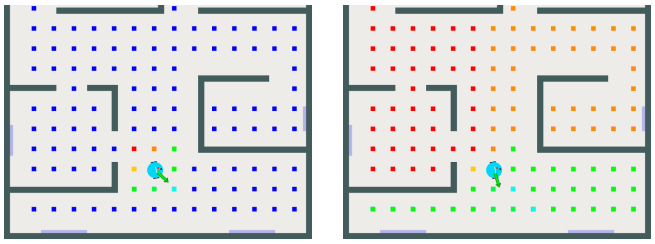
III. Proposed Estimation Method

The algorithm presented in this work follows the standard structure of estimation-based GSL. Starting at a specific location, the robot (1) takes a new measurement: gas and wind in our case, (2) updates the state estimation according to their agreement with observation models and (3) determines where to move next. This loop continues until the source is deemed to have been found (source declaration), which is determined by a convergence threshold.

The observation model we propose in this work uses the trajectory-estimation concept discussed in section II-B. For this purpose, we discretize the environment in a number of equally-sized cells, and whenever a patch of gas is detected, we estimate the probability of any given cell being part of the path the gas patch has traversed before reaching the robot. This discretization allows us to model the geometry of the environment as an occupancy grid, and thus make certain that the paths we estimate the gas patches to have followed are feasible.

Estimating these paths is not a trivial task, since the sensory information that is available to the robot is strictly local. For this reason, the process of generating estimations is divided in two steps (Fig. 1): using the sensory measurements to produce local estimations (section III-A) and propagating the local estimations to the rest of the environment (section III-B). Since the algorithm is designed for indoors environments, we can assume that the direction and conditions of the airflow, even if they are unstable, do not drastically change from one instant to another, as there are –unknown– fixed points where air can enter or exit the environment. Thus, we can combine these new estimations

Authors' accepted manuscript: IEEE Robotics and Automation Letters (RAL), 2021.
The final publication is available at: http://dx.doi.org/10.1109/LRA.2021.3057290



(a) Probabilities of the neighboring cells to be in the gas path are generated from the wind vector.

(b) In a second stage, local probabilities are propagated through the grid to all cells.

Fig. 1: Each of the points in these figures corresponds to the center of a cell, its color represents the probability: red being the highest and blue the lowest.

with previous ones to produce an accumulated estimation of the probability of each cell containing the source.

A. Local Estimations

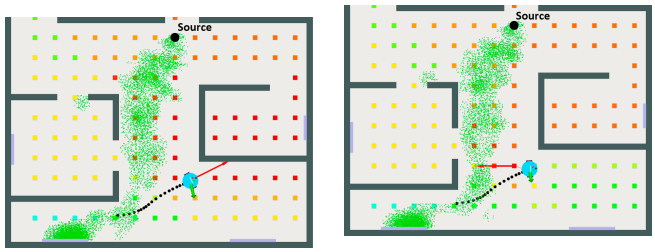
When the robot detects a gas patch, a "local" estimator quantifies for each of the neighboring cells the probability that the gas has traversed through it before reaching the robot cell.

Let c_{ij} denote the cell in the i-th row and j-th column of the grid. Defining the set V of cells adjacent to the current robot position which are not occupied by an obstacle, we calculate the probability of each cell in V to have contained the detected gas patch at a previous time. To do so we rely on the measured local wind direction and the angle formed by the vector that goes from the robot position to the center of the cell (ω). Specifically, we model this probability as a wrapped normal distribution, since the probability of a given cell being the one that contained the gas patch in the previous instant is maximal for the cell directly upwind, and decreases symmetrically for values of ω that are further from the upwind direction.

$$p_t(c_{ij}|z_t = 1, \theta_t) = p(\omega) \sim WN(\theta, \sigma_h^2) \tag{1}$$

where z_t represents the discretized gas measurement at time step t, with $z_t = 1$ indicating that the gas concentration is above a given threshold (*i.e.* a *hit*) and $z_t = 0$ otherwise (*i.e.* a *miss*); and θ_t is the averaged upwind direction measured at time step t, expressed as an angle in the frame of reference of the environment. The standard deviation (σ_h) of this distribution is a parameter that must be set in accordance with the reliability of the sensors and the stability of the airflow. Empirically, we have noted that values within the range 0.5-2 radians do not change much the behavior of the algorithm

When the robot does not measure any gas ($(z_{ij} = 0)$), the wind measurement does not offer much information. An example is illustrated in Fig. 2a, which shows a common scenario where a miss cannot be interpreted as the source is not upwind. As a consequence, the naive approach of reversing the direction of the estimations does not produce reliable estimations. Instead, in the event of a *miss*, it will be considered that the robot has moved in the wrong direction since the last gas hit was detected, lowering, this way, the probabilities of cells in that same direction (Fig. 2b). Again,



(a) The robot fails to measure gas after a movement (marked by the black dot trail and the red arrow).

(b) Cells in the direction of the movement are estimated a lower probability of containing the source.

Fig. 2: When a *miss* occurs, it cannot be inferred that the source is not upwind. In this example, it occurs because the robot has just exited the gas plume on its side.

these estimations are modeled through a wrapped normal distribution:

$$p_t(c_{ij}|z_t = 0, \phi_t) = p(\omega) \sim WN(\phi, \sigma_m^2) \tag{2}$$

This expression does not use the wind direction (θ_t). Instead the normal distribution is centered at the direction of the vector from the current robot position to the location of the last hit (ϕ_t). Since *hits* tend to provide more information about the source location than *misses*, the value of σ_m will need to be higher than the value of σ_h. We have empirically observed that values of $\sigma_m = \sigma_h + 0.5$ produce good results. This makes the steepness of the probability gradient calculated after a miss lower —compared to the one caused by a hit—, reducing the effect of that measurement on the accumulated probability of of each cell containing the source after the update step (section III-C).

B. Propagation of Estimations

Due to the local nature of the sensory measurements available to the robot, it is not possible to evaluate cells that are far from the robot following the same principle we use for neighbor cells. For that reason, instead of applying expressions 1 and 2 to all cells in the grid, we propagate the estimations made on the cells in V using information about the geometry of the environment.

Specifically, cells that are not in V will be assigned the probability of containing the source of the cell $c \in V$ to which the shortest free path exists. In practice, this is done through a 8-neighbor (or 4-neighbor) propagation algorithm (Algorithm 1).

Since the propagation occurs through free cells only –as these are the only cells returned by the *neighbors* function–, it is possible to produce results that conform to the geometry of the environment, accounting for the presence of obstacles. The computational complexity of this procedure is linear in the number of cells n of the map ($O(n)$), since each cell can be reached at most 8 times per iteration of the algorithm (once for each of its neighbors). In practice, since not all cells are free, the number of steps will often be smaller.

This propagation is, in essence, just an algorithmic way of generalizing the idea of producing "upwind estimations" to any map geometry. Fig. 3a shows how, in the absence of obstacles, all cells within a certain angle upwind are

Authors' accepted manuscript: IEEE Robotics and Automation Letters (RAL), 2021.
The final publication is available at: http://dx.doi.org/10.1109/LRA.2021.3057290

Algorithm 1: Propagation algorithm

```
activeSet = ∅;
closedSet = ∅;
openSet= V;
while | openSet |> 0 do
    activeSet=openSet;
    closedSet.insertAll(activeSet);
    openSet= ∅;
    for cell ∈ activeSet do
        closedSet.insert(cell);
        for n ∈ neighbors(cell) do
            if n ∉ closedSet & n ∉ openSet then
                p(n|z_t) = p(cell|z_t);
                openSet.insert(n);
            else if n ∉ closedSet then
                p_t(n) = max(p(n|z_t), p(cell|z_t));
            end
        end
    end
end
```

assigned the highest probability of containing the source. This is a very intuitive result, since our local estimations (that gas did travel through the immediate neighbor in the upwind direction before reaching the robot) matches what would be expected if gas was originating from any of those cells. However, the existence of obstacles means that just comparing the upwind vector to the angle from the robot to the cell would not accurately reflect this intuition, as shown by figure 3b.

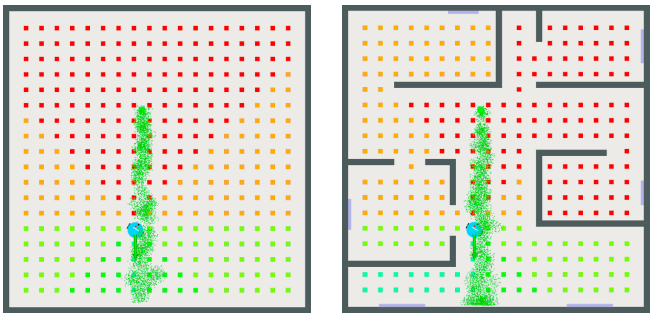
C. Updating Estimations

After all the cells have been assigned a probability of containing the source based on the last observation, the accumulated probabilities for the cells are recursively updated. Assuming that the sequence of observations satisfies the Markov Property and applying Bayes' Theorem of Conditional Probability, the new probability can be calculated as follows:

$$\begin{aligned} p(c_{ij}|z_{1:t}) &= p(c_{ij}|z_t, z_{1:t-1}) \\ &= \frac{p(z_t|c_{ij}, z_{1:t-1}) \cdot p(c_{ij}|z_{1:t-1})}{p(z_t|z_{1:t-1})} \\ &= \frac{p(z_t|c_{ij}) \cdot p(c_{ij}|z_{1:t-1})}{p(z_t)} \\ &= \frac{p(c_{ij}|z_t) \cdot p(z_t) \cdot p(c_{ij}|z_{1:t-1})}{p(z_t) \cdot p(c_{ij})} \\ &= \frac{p(c_{ij}|z_t) \cdot p(c_{ij}|z_{1:t-1})}{p(c_{ij})} \end{aligned} \tag{3}$$

In this work we assume there is no initial knowledge about which cells are most likely to contain the source, so the prior probability of containing the gas source is identical for all cells: $\forall ij;\ p(c_{ij}) = 1/n$, where n is the number of free cells in the environment. Therefore, since $p(c_{ij})$ is a constant, this expression can be rewritten as:

$$p(c_{ij}|z_{1:t}) \propto p(c_{ij}|z_t) \cdot p(c_{ij}|z_{1:t-1}) \tag{4}$$



(a) In an empty environment, the propagation of local probabilities causes a radial distribution.

(b) The resulting pattern changes with the shape of the map, better reflecting the paths estimated for the gas patches.

Fig. 3: The propagation algorithm allows the resulting probability distribution of the source location to adapt to the specific shape of the environment.

After a new measurement is included in the accumulated probability estimations, the probabilities of the cells are normalized so $\Sigma_i \Sigma_j p(c_{ij}|z_{1:t}) = 1$.

IV. Movement Strategy

Once the robot finishes updating the probabilities of the cells it must choose where to move to in order to take the next measurement. We propose and compare two such navigation strategies. The first one has the robot move towards of the area with highest probability of containing the gas source. The second strategy revolves around the concept of Infotaxis [19], using existing probabilistic techniques to predict the wind vector and gas concentration in nearby cells, and moving the robot to the cells where the highest information gain is expected.

A. Best-First Navigation

Since the environment has been discretized into a grid, and grids can easily be represented by graphs, any graph-search algorithm can be applied. The strategy we propose is a best-first search, using the accumulated estimated probability of containing the source as an evaluation of the cell.

Each step, the robot adds all neighboring cells that have not yet been visited to the list of candidate next positions (the *open set*). This list persists from iteration to iteration, and so all unchosen candidates from previous iterations will be considered again. Out of all the positions in the list of candidates, the robot chooses the one with the highest estimated probability of containing the gas source. This new position is taken out of the list of candidates for future iterations and is not considered to be visited again.

The emergent behavior resembles that of a reactive plume-tracking method, where the robot moves upwind until it fails to measure gas. Yet, using the accumulated probability to guide the movements of the robot makes the strategy more consistent than basing the movement only on the last measurement. Thus, a single observation that contradicts the pre-existing estimations will not cause the direction that is currently being explored to be discarded.

Expanding the list of candidate moves with the immediate neighbors of the currently occupied cell means the robot can

Authors' accepted manuscript: IEEE Robotics and Automation Letters (RAL), 2021.
The final publication is available at: http://dx.doi.org/10.1109/LRA.2021.3057290

only advance one cell at a time in the direction it is exploring, while the not-selected candidates still remain eligible for future iterations, thus the robot can backtrack eventually.

B. Infotactic Navigation

Infotaxis [19] offers an interesting approach for designing sophisticate movement strategies. The functioning principle of Infotaxis is the maximization of the expected information gain; that is, the robot will move to the point in which it estimates to gain the most information possible.

The original infotaxis algorithm attempts to maximize the reduction of the Shannon's entropy of the probability distribution of the source location as a way to optimize information gain. More recently [12] the use of the Kullback-Leibler Divergence (KLD) has been proposed for as a measure of information gain. The use of KLD equates to considering that any change in the probability distribution of the source location is a net gain of information, even if the resulting entropy is higher. This has the added benefit of letting the distribution to escape from local minima of the entropy. This is the technique we adopt here.

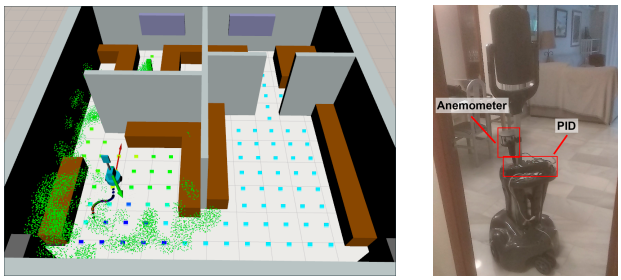
We denote the set of candidate moves —generated as explained in section IV-A— as M. For each $m \in M$, the estimated probability of a gas hit occurring in the new position is g_m, and the estimated wind vector to be measured is w_m. We refer to the current probability distribution of the source location as P^t, and to the distribution that results in the next time step after a given measurement as P_z^{t+1}, where z is a pair (hit/miss, wind vector) that represents the measurement. The expected information gain Ψ of each movement $m \in M$ is thus:

$$\begin{aligned}\Psi(m) =& g_m \cdot KLD(P^t, P^{t+1}_{hit,w_m}) \\ &+ (1-g_m) \cdot KLD(P^t, P^{t+1}_{miss,w_m})\end{aligned} \tag{5}$$

This calculation requires estimating the gas and wind measurements the robot would obtain in each of the candidate new locations. Several techniques have been proposed to estimate the gas concentration of nearby cells from local measurements [20], [21]; however, because most of them are designed to generate a map of the gas distribution —rather than to guide GSL algorithms— they offer powerful tools for interpolating sparse measurements in the areas already visited, but are not as effective for predicting the state of the distribution in areas yet to be explored (*i.e.* extrapolation).

In this work, we use the probabilistic estimations of the source location to infer the probability of measuring a gas hit in a given cell. Since the probability of a cell being the source of the gas has been calculated by estimating the path that previously measured patches of gas have followed, it can also be interpreted as the probability of that given cell having contained gas, and therefore can be utilized to estimate the likelihood of obtaining a gas hit in that same position.

Estimating the wind vectors that would be measured in other cells challenging, since our algorithm does not assume a stable, homogeneous airflow. The application of Computational Fluid Dynamics (or CFD) techniques for this purpose poses a problem: the complexity of the required calculations makes them intractable for a real-time search. For this purpose we employ a technique proposed by Monroy *et al.* [22], [23], based on Gaussian Markov Random Fields to extrapolate the wind vectors to nearby points of the environment from the available local measurement and the geometry of the environment, that is, a local map around the robot location.



(a) (b)

Fig. 4: (a) Shows an example of the fully 3-dimensional environments used for simulated experiments. (b) Shows an image of the robot taken during the real-world experimentation.

V. Experimental Validation

A. Setup

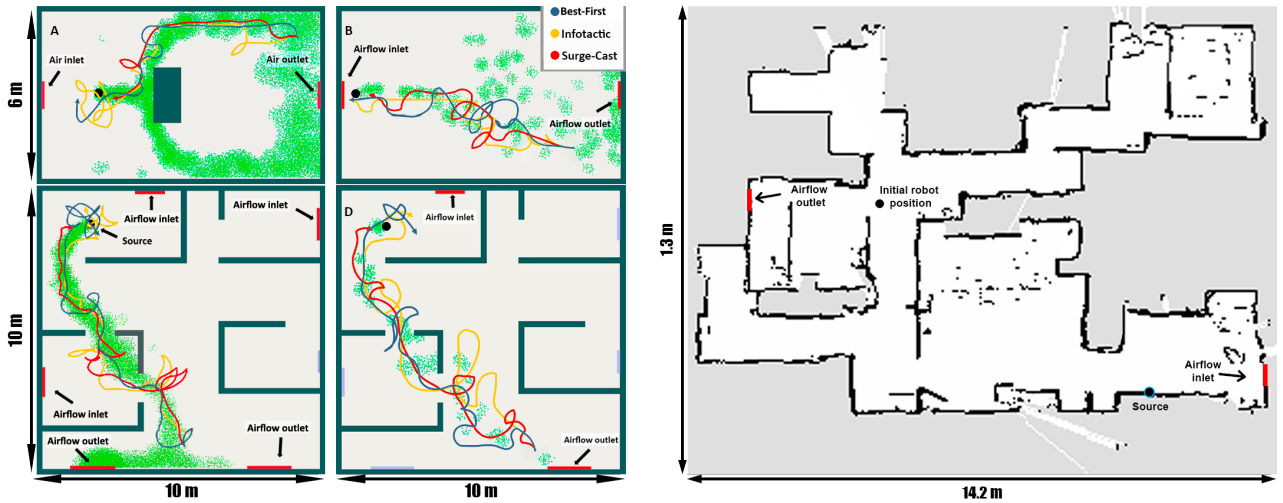
For the simulations we use GADEN [24], a 3-D Gas Dispersion Simulator (*GDS*) specifically designed for Robotic Olfaction (Fig. 4a). The real-world experiment was carried out in the interior of a house, using only natural ventilation.

Fig. 5a shows a top-down view of the setup of each of the simulated experiments. Experiment A takes place in a room with a central obstacle that breaks the gas plume and causes it to bend. Experiment B takes place under heavily turbulent airflow that prevents the formation of an actual plume. Experiment C takes place in a larger environment that features a more complex geometry with multiple rooms, with a source that releases gas at a steady rate allowing the formation of a continuous plume. Experiment D takes place in that same environment, but with a lower release rate. Figure 5b shows the floor map of the environment in which the real-world experiment was carried out.

For each set of environmental conditions, we test our GSL algorithm, presented in section III, with both of the navigation strategies presented in section IV. Additionally, to have a reference to help evaluate these results, we also perform the search with Surge-Cast Plume Tracking, one of the most popular state-of-the-art GSL algorithms, which has shown to produce good results in a wide range of environmental conditions [7].

For our algorithm, a given run is considered to be a success if the final declared source position is within a certain distance of the ground-truth source location (0.5m for the simulated experiments, 1m in the real-world experiment); for Surge-Cast, it is considered to be a success if the robot physically reaches that same radius around the source before 300s (600s for the real-world experiment). The difference in this criteria is motivated by the fact that Surge-Cast, being a purely reactive method, does not preform source declaration. The robot used for the experiments is equipped with a

Authors' accepted manuscript: IEEE Robotics and Automation Letters (RAL), 2021.
The final publication is available at: http://dx.doi.org/10.1109/LRA.2021.3057290



(a) Top-down view of the setup for each of the simulated experiments, and examples of the paths followed by the robot.

(b) Floor map of the environment in which the real-world experiment was carried out.

Fig. 5: Set of environmental conditions under which the algorithms were tested.

Algorithm	Success rate	Navigation time (s)	Declaration time (s)	Cells explored	Error (m)
Best-First	100%	$\mu = 78.95$ $\sigma = 8.65$	$\mu = 95.54$ $\sigma = 12.80$	$\mu = 22.9$ $\sigma = 2.21$	$\mu = 0.37$ $\sigma = 0.09$
Infotaxis	100%	$\mu = 99.26$ $\sigma = 10.32$	$\mu = 125.30$ $\sigma = 8.81$	$\mu = 27.46$ $\sigma = 1.62$	$\mu = 0.17$ $\sigma = 0.08$
Surge-Cast	100%	$\mu = 70.50$ $\sigma = 9.22$	-	-	-

(a) Results for experiment A.

Algorithm	Success rate	Navigation time (s)	Declaration time (s)	Cells explored	Error (m)
Best-First	93%	$\mu = 146.94$ $\sigma = 43.25$	$\mu = 155.24$ $\sigma = 43.36$	$\mu = 25.39$ $\sigma = 6.12$	$\mu = 0.86$ $\sigma = 1.52$
Infotaxis	97%	$\mu = 158.49$ $\sigma = 33.29$	$\mu = 168.54$ $\sigma = 34.26$	$\mu = 26.03$ $\sigma = 4.58$	$\mu = 0.62$ $\sigma = 0.95$
Surge-Cast	30%	$\mu = 160.94$ $\sigma = 44.02$	-	-	-

(b) Results for experiment B.

Algorithm	Success rate	Navigation time (s)	Declaration time (s)	Cells explored	Error (m)
Best-First	100%	$\mu = 81.38$ $\sigma = 27.74$	$\mu = 108.18$ $\sigma = 29.67$	$\mu = 21.03$ $\sigma = 4.50$	$\mu = 0.27$ $\sigma = 0.11$
Infotaxis	93%	$\mu = 119.69$ $\sigma = 22.01$	$\mu = 144.58$ $\sigma = 25.30$	$\mu = 22.79$ $\sigma = 2.82$	$\mu = 0.38$ $\sigma = 0.13$
Surge-Cast	66%	$\mu = 151.87$ $\sigma = 42.28$	-	-	-

(c) Results for experiment C.

Algorithm	Success rate	Navigation time (s)	Declaration time (s)	Cells explored	Error (m)
Best-First	90%	$\mu = 160.59$ $\sigma = 25.02$	$\mu = 192.00$ $\sigma = 25.74$	$\mu = 29.22$ $\sigma = 3.01$	$\mu = 0.88$ $\sigma = 1.19$
Infotaxis	93%	$\mu = 158.63$ $\sigma = 30.63$	$\mu = 194.77$ $\sigma = 40.61$	$\mu = 24.25$ $\sigma = 4.49$	$\mu = 0.69$ $\sigma = 0.79$
Surge-Cast	77%	$\mu = 129.02$ $\sigma = 23.20$	-	-	-

(d) Results for experiment D.

TABLE I: Results obtained in the simulated experiments.

Algorithm	Success rate	Navigation time (s)	Declaration time (s)	Cells explored	Error (m)
Best-First	90%	$\mu = 273.18$ $\sigma = 44.96$	$\mu = 310.75$ $\sigma = 41.03$	$\mu = 28.00$ $\sigma = 2.79$	$\mu = 0.76$ $\sigma = 0.27$
Infotaxis	80%	$\mu = 220.76$ $\sigma = 13.64$	$\mu = 254.98$ $\sigma = 31.65$	$\mu = 23.12$ $\sigma = 1.53$	$\mu = 0.80$ $\sigma = 0.33$
Surge-Cast	60%	$\mu = 496.21$ $\sigma = 46.67$	-	-	-

TABLE II: Results obtained in the real-world experiment.

photoionization gas detector (*PID*) and a 2-D anemometer, both in the real-world experiments and in simulations.

The gas used in all experiments is ethanol. For the purposes of reproducibility, all the source code and configuration files of the simulated environments (source release rate, characteristics of the airflow, parameters of the algorithm, *etc*.) are available in an online open repository[1].

B. Results

Simulated experiments were carried out 30 times for each combination of algorithm and environment, and the real world experiment was carried out 10 times per algorithm. Results are shown in tables I and II. Column "Navigation time" shows how long it takes for the robot to reach a 0.5m radius around the source location, and column "Declaration time" shows how long before the algorithm converges and a final estimation of the source location is declared. Column "Error" shows how far the declared source position (*i.e.* the center of the declared cell) is from the ground-truth on average, including failed runs.

It can be observed that, in all cases, our proposed method obtains a success rate equal or greater than that of Surge-Cast, though the time required to reach the source in successful runs is not necessarily better. Both of these results can be attributed to the fact that our algorithm does have a memory of past states: after losing the gas plume, the robot can easily backtrack to the point where gas was last found, and it is less prone to changing the direction of exploration after a single *miss*, which makes the search more consistent;

[1] https://github.com/MAPIRlab/Gas-Source-Localization

Authors' accepted manuscript: IEEE Robotics and Automation Letters (RAL), 2021.
The final publication is available at: http://dx.doi.org/10.1109/LRA.2021.3057290

however, this also means it might take a longer time for the robot to adapt to an actual change in the direction of the plume, sometimes needing to explore several empty cells before making a turn.

It must also be considered that there is a certain degree of survivorship bias in the time results. If an algorithm often fails at regaining a lost plume, the runs that are most likely to succeed are the ones in which the robot never strays far from the gas plume, which are the fastest.

In the case of the real-world experiment it must also be taken into account that navigation through narrow spaces is challenging [25], and the robot must often waste time re-computing the path or executing recovery behaviors. Our algorithm shows a significantly better average time than Surge-Cast mainly because the existence of the cell centers as navigation waypoints greatly reduces the effect of this problem, which is not directly related to the search strategy.

Despite the high success rate, it should be pointed out that when our algorithm fails to declare the correct source position it is often several meters off, not being able to provide an approximate location with reliability.

Both proposed movement strategies show similar results, although the infotactic variant is generally able to converge to the final estimated location after exploring fewer cells.

VI. Conclusions and Future Work

In this work we have presented a new GSL algorithm based on estimating the paths followed by gas patches, proposing a grid-based framework as a way to generate estimations that are consistent with the geometry of the environment and without assuming any environmental condition.

The results obtained from a set of experiments are encouraging, with a high success rate under all tested environmental conditions. Still, further testing and comparisons with similar strategies would be necessary to more accurately analyze the performance of the algorithm. Several lines are proposed for future work, including the inclusion of more sophisticate GDM techniques for measurement predictions in the infotactic search; and the use of more complex environment representations, such as Hilbert Maps [26], which may facilitate the modification of the probability propagation procedure to account for the three-dimensionality of the environment and the adaptation of the algorithm to work with on-line mapping.

References

[1] S. M. Bobrovnikov, E. V. Gorlov, V. I. Zharkov, Y. N. Panchenko, V. A. Aksenov, A. V. Kikhtenko, and M. I. Tivileva, "Remote detector of explosive traces," in *20th International Symposium on Atmospheric and Ocean Optics: Atmospheric Physics*, 2014.

[2] D. Karakaya, O. Ulucan, and M. Turkan, "Electronic Nose and Its Applications: A Survey," *International Journal of Automation and Computing*, vol. 17, no. 2, 2020.

[3] A. Gongora, J. Monroy, and J. Gonzalez-Jimenez, "An electronic architecture for multi-purpose artificial noses," *Journal of Sensors*, vol. 2018, feb 2018.

[4] X.-x. Chen and J. Huang, "Odor source localization algorithms on mobile robots: A review and future outlook," *Robotics and Autonomous Systems*, vol. 112, 2019.

[5] J. Monroy and J. Gonzalez-Jimenez, *Towards Odor-Sensitive Mobile Robots*. IGI Global, 2018, pp. 244–263. [Online]. Available: http://mapir.isa.uma.es/mapirwebsite/index.php/mapir-downloads/papers/284

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

[7] P. Ojeda, J. Monroy, and J. Gonzalez-Jimenez, "An evaluation of gas source localization algorithms for mobile robots," in *International Conference on Applications of Intelligent Systems*, no. 24, 2020.

[8] G. Kowadlo and R. A. Russell, "Robot odor localization: A taxonomy and survey," *International Journal of Robotics Research*, vol. 27, no. 8, 2008.

[9] T. Lochmatter and A. Martinoli, "Tracking Odor Plumes in a Laminar Wind Field with Bio-inspired Algorithms," in *Experimental Robotics*, 2009, vol. 54.

[10] R. A. Russell, A. Bab-Hadiashar, R. L. Shepherd, and G. G. Wallace, "A comparison of reactive robot chemotaxis algorithms," *Robotics and Autonomous Systems*, vol. 45, no. 2, nov 2003.

[11] T. Wiedemann, D. Shutin, V. Hernandez, E. Schaffernicht, and A. J. Lilienthal, "Bayesian gas source localization and exploration with a multi-robot system using partial differential equation based modeling," in *International Symposium on Olfaction and Electronic Nose*, 2017.

[12] M. Hutchinson, C. Liu, and W. H. Chen, "Information-Based Search for an Atmospheric Release Using a Mobile Robot: Algorithm and Experiments," *IEEE Transactions on Control Systems Technology*, 2018.

[13] F. Rahbar, A. Marjovi, and A. Martinoli, "Design and performance evaluation of an algorithm based on source term estimation for odor source localization," *MDPI Sensors*, vol. 19, no. 3, 2019.

[14] M. Hutchinson, H. Oh, and W. H. Chen, "A review of source term estimation methods for atmospheric dispersion events using static or mobile sensors," *Information Fusion*, vol. 36, 2017.

[15] S. Sklavounos and F. Rigas, "Validation of turbulence models in heavy gas dispersion over obstacles," *Journal of Hazardous Materials*, vol. 108, no. 1-2, apr 2004.

[16] T. Wiedemann, A. J. Lilienthal, and D. Shutin, "Analysis of model mismatch effects for a model-based gas source localization strategy incorporating advection knowledge," *MDPI Sensors, 3*, vol. 19, 2019.

[17] C. Sánchez-Garrido, J. Monroy, and J. Gonzalez-Jimenez, *Probabilistic Estimation of the Gas Source Location in Indoor Environments by Combining Gas and Wind Observations*, 2018.

[18] J. G. Li, Q. H. Meng, Y. Wang, and M. Zeng, "Odor source localization using a mobile robot in outdoor airflow environments with a particle filter algorithm," *Autonomous Robots*, vol. 30, no. 3, 2011.

[19] M. Vergassola, E. Villermaux, and B. I. Shraiman, "'Infotaxis' as a strategy for searching without gradients," *Nature, 7126, 445*, 2007.

[20] J. Monroy, J. L. Blanco, and J. Gonzalez-Jimenez, "Time-variant gas distribution mapping with obstacle information," *Autonomous Robots*, vol. 40, no. 1, jan 2016.

[21] S. Asadi, H. Fan, V. H. Bennetts, and A. J. Lilienthal, "Time-dependent gas distribution modelling," *Robotics and Autonomous Systems*, vol. 96, 2017.

[22] J. Monroy, M. Jaimez, and J. Gonzalez-Jimenez, "Online estimation of 2d wind maps for olfactory robots," in *International Symposium on Olfaction and Electronic Nose*, 2017.

[23] A. Gongora, J. Monroy, and J. Gonzalez-Jimenez, "Joint estimation of gas & wind maps for fast-response applications," *Applied Mathematical Modelling*, 2020.

[24] J. Monroy, V. Hernandez-Bennetts, H. Fan, A. Lilienthal, and J. Gonzalez-Jimenez, "GADEN: A 3D gas dispersion simulator for mobile robot olfaction in realistic environments," *MDPI Sensors*, vol. 17, no. 7, 2017.

[25] F.-A. Moreno, J. Monroy, J. R. Ruiz-Sarmiento, C. Galindo, and J. Gonzalez-Jimenez, "Automatic waypoint generation to improve robot navigation through narrow spaces," *MDPI Sensors, 20, 1*, 2020.

[26] F. Ramos and L. Ott, "Hilbert maps: Scalable continuous occupancy mapping with stochastic gradient descent," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1717–1730, 2016.