Efficient Measurement Planning for Remote Gas Sensing with Mobile Robots

Conference Paper · May 2015

DOI: 10.13140/RG.2.1.2776.1761

Muhammad Asif Arain
Örebro University
10 PUBLICATIONS **81** CITATIONS
SEE PROFILE

Victor Hernandez Bennetts
Örebro University
54 PUBLICATIONS **850** CITATIONS
SEE PROFILE

Marco Trincavelli
Örebro University
59 PUBLICATIONS **1,211** CITATIONS
SEE PROFILE

Achim J. Lilienthal
Örebro University
450 PUBLICATIONS **7,626** CITATIONS
SEE PROFILE

2015 IEEE International Conference on Robotics and Automation (ICRA)
Washington State Convention Center
Seattle, Washington, May 26-30, 2015

Efficient Measurement Planning for Remote Gas Sensing with Mobile Robots

Muhammad Asif Arain[1*], Marcello Cirillo[1,2], Victor Hernandez Bennetts[1], Erik Schaffernicht[1], Marco Trincavelli[1] and Achim J. Lilienthal[1]

Abstract— **The problem of gas detection is relevant to many real-world applications, such as leak detection in industrial settings and surveillance. In this paper we address the problem of gas detection in large areas with a mobile robotic platform equipped with a remote gas sensor. We propose a novel method based on convex relaxation for quickly finding an exploration plan that guarantees a complete coverage of the environment. Our method proves to be highly efficient in terms of computational requirements and to provide nearly-optimal solutions. We validate our approach both in simulation and in real environments, thus demonstrating its applicability to real-world problems.**

I. Introduction

The problem of detecting gases and mapping their concentration is relevant to different applications, such as gas leak detection in industrial areas, surveillance of areas where hazardous gases might be present and pollution monitoring in cities. In the past decades, stationary sensor networks have been largely used to address this problem. Typically, the sensors used are relatively inexpensive, readily available and light weight, but they have to be directly exposed to the target gases (i.e., in-situ sensing), which means that they provide a point measurement where they are positioned [1]. The nature of the sensors used entails that gas emissions could go undetected due to sparse measurements and that, whenever there are changes to the environment where the network is placed, a new, time-consuming deployment could be required.

More recently, mobile robots equipped with gas sensors have proven to be valuable instruments for gas detection and mapping [2], [3], as they can flexibly adapt to complex outdoor and indoor environments. Moreover, their use has become even more appealing since the introduction of sensors which are capable of detecting gases remotely. For instance, sensors based on tunable diode laser absorption spectroscopy (TDLAS) [4] allow for ranged sensing up to considerable distances. However, compared to in-situ sensors, they are larger and more expensive. Thus, they cannot be distributed in large numbers in the environment, but they are suitable to be used on-board a mobile platform.

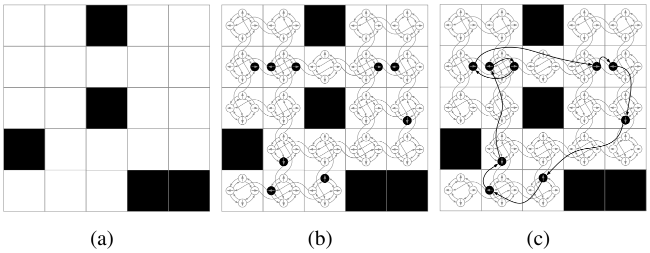


Fig. 1. 1(a): A simple test map, where obstructed cells are represented in black and traversable ones in white. Candidate sensing configurations are defined over poses in the cells such that $\Theta = \{0, \frac{\pi}{2}, \pi, \frac{3}{2}\pi\}$ and have identical ϕ and r ($r = 2$ cells, $\phi = \frac{\pi}{2}$). 1(b): Minimum set of sensing configurations from which all cells are visible. 1(c): The sensing configurations are connected by a minimum length shortest path.

Here, we address the problem of gas detection in complex indoor and outdoor environments. More specifically, we focus on quickly generating *efficient* exploration plans for a mobile robot equipped with a ranged gas sensor. Scanning an area with a remote gas sensor is a time- and energy-consuming action and, therefore, the number of such actions as well as the distance covered by the robot should be minimized, in order to save battery and quickly complete the exploration. The contribution of this paper is twofold: First, we propose a new convex relaxation method for quickly generating efficient exploration strategies. Second, we experimentally evaluate our approach, both in simulation and with a robot in two real-world scenarios.

II. Related Work

Mobile robot olfaction – the use of robots for mobile gas sensing – can provide more flexibility compared to fixed sensor networks [5], and has been successfully used in different tasks, such as mapping gas distributions [3] and leak detection [6]. However, the problem of efficiently generating plans for gas detection in large environments has been overlooked. A common approach in current state of the art is to program the robotic platforms to follow pre-defined exploration paths or to reactively navigate in the area of interest. Such exploration plans should be time- and energy-efficient, taking into account both the sensing actions and the distance traveled by the robot. This, intuitively speaking, corresponds to solving a combination of a *Watchman Route Problem* (that is, the problem of computing the shortest route to guard a known area) and an *Art Gallery Problem* (that is, selecting the minimum number of observation points to completely observe a known area). Finding an optimal solution to the combined problem is obviously a daunting task. In [7], the authors use an approximation method to

[1] Mobile Robotics & Olfaction Lab, Center of Applied Autonomous Sensor Systems (AASS), Örebro University, SE-70182 Örebro, Sweden.
[2] Scania AB, Granparksvägen 10, SE-15187 Södertälje, Sweden.
[*] Corresponding author <asif.arain@oru.se>

978-1-4799-6922-7/15/$31.00 ©2015 IEEE

find a bounded suboptimal solution which considers both traveling time and sensing time with respect to a pre-defined set of sensing positions. However, the problem of selecting the initial set of sensing positions remains open and the approach presents scalability issues when the cardinality of the set is increased.

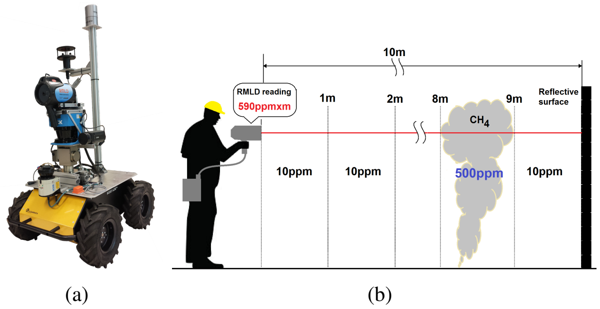
In the past decades, there has been extensive work on problems which are closely related to the one addressed in this paper. Ignoring the cost associated to sensing actions, our problem could be reduced to the Generalized Covering Salesman Problem [8], whose instances are solved by generating paths from which the whole environment can be observed. A similar problem is solved in [9], where an approach based on mixed integer linear programming is used for finding a surveillance route for a mobile camera.

Neglecting the cost associated to the movements of the robot, our problem could be reduced to an Art Gallery Problem [10] or to a View Planning Problem [11]. The art gallery problem is NP-hard in its most common variants and view planning is isomorphic to the Set Covering Problem [12], a well known NP-complete problem. This family of problems has been extensively studied over the past decades [13], [14], but the algorithms proposed for optimal solutions are effective under restricted assumptions, such as not considering occlusions in the field of view of the sensors [15], or work only when the number of possible sensing positions is relatively small [16], [17]. There exist, however, efficient algorithms for calculating approximated solutions [18] for problem instances with a large number of possible sensing positions. The solutions described above, however, present two major drawbacks in our context: First, they are mostly concerned with camera placement problems, which means that they rarely consider limited field of view; second, they do not consider the cost of moving from one sensor position to the next. On the other hand, assuming known sensing positions, we would still need to solve a (Metric) Traveling Salesman Problem, that is, we should find the shortest tour that connects all the sensing positions. This is a well-known NP-hard problem, but it is possible to optimally solve very large TSP instances, with thousands of locations [19].

A viable solution for a broad range of applications is to first solve an art gallery problem, and then calculate the shortest tour among the selected sensing positions by solving a TSP [20]. However, to the best of our knowledge, all the algorithms proposed rely on simplifying assumptions on the field of view of the sensors (e.g., 360 ° or unrestricted). More important still, the solution of an art gallery problem for a large number of candidate sensing positions with overlapping fields of view remains computationally challenging. By contrast, in our approach, we use re-weighted convex relaxation in addition to the combinatorial method to scale up the size of the problems we can tackle.

III. Problem Definition

In our application scenario, we assume that the gas, if present, can be detected near ground level. This assumption holds in many real world applications, such as the detection of methane leaks at landfill sites. Also, we assume a static environment with no major changes in the distribution of the concentration fluctuations around the mean. Gas sensing is then carried out using a mobile robot equipped with an actuated TDLAS remote gas sensor (Fig. 2(a)). TDLAS sensors report integral concentration measurements only along a narrow beam that is emitted by the sensor's transceiver (Fig. 2(b)). Thus, in order to increase the sensor's field of view, a pan-tilt unit is used to aim the sensor's transceiver in different orientations generating, in this way, semi-circular measurement sectors of angle ϕ and radius r (Fig. 4(a)).



(a) (b)

Fig. 2. 2(a) *Gasbot*, a Husky A200 based robotic platform for gas detection [3]. 2(b) The Remote Methane Leak Detector is a remote gas sensor which can report the integral concentration of methane along its laser beam (parts per million per meter) [3].

Given a map of the environment in which we need to assess the presence of a specific gas, we divide it into a Cartesian grid, thus obtaining a set $\mathcal{A}$ of n cells of identical size: $\mathcal{A} = \{a_1, \cdots, a_n\}$. The set of all cells $\mathcal{A}$ is partitioned into subsets $\mathcal{O}$ and $\mathcal{S}$, where $\mathcal{O}$ includes all the cells which contain an obstacle and therefore (1) are not traversable by the robot, and (2) can stop the beam of the gas sensor. $\mathcal{S}$ includes all the cells which do not contain obstacles and are therefore traversable by the robot. A solution to the detection problem is an obstacle free closed path, or *tour*, within the map that the robot can traverse and a set of sensing actions along the tour which enable the robot to sense every cell in $\mathcal{S}$. We limit the movement of the robot between a finite set of poses $\mathcal{P}$. Each $p_j \in \mathcal{P}$ is defined by a two-tuple (a_j, θ_j), where $a_j \in \mathcal{S}$ and $\theta_j \in \Theta$. Θ is a finite set of allowed orientations, equally spaced between $[0, 2\pi)$. Thus, for $\Theta = \{0, \frac{\pi}{2}, \pi, \frac{3}{2}\pi\}$, the movements of the robot can be captured by a directed graph like the one represented in Fig. 3. Note that the robot position within a cell corresponds to its center, regardless of the orientation. Here, we assume that the movements of the robot are limited to forward motions and rotations.

In our problem definition, we use *time* as measure of cost. Given the times t^r, representing the time necessary to rotate $2\pi/|\Theta|$, and t^c, the cost associated to moving to the adjacent cell, the movement time $t^m_{p_i \rightarrow p_j}$ from p_i to p_j can be easily calculated by finding the shortest path on the movement graph, where each rotation edge has a weight of t^r and every edge leading to a different cell has a weight

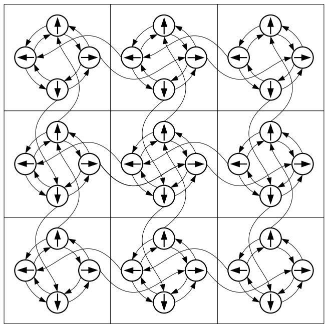


Fig. 3. The graph captures the allowed movements of the robot on a grid map when $\Theta = \{0, \frac{\pi}{2}, \pi, \frac{3}{2}\pi\}$ and only forward movements are allowed. Small circles indicate the poses where the robot can stop (the internal arrow indicates the orientation of the robot) and the directed edges its allowed movements. Note that in the figure the poses do not correspond to the centers of the cells, but this is only for clarity reasons.

of t^c. With this decomposition, and with a knowledge of the terrain in each cell, it would be straightforward to express the cost of movement in terms of battery consumption.

We can now define a *candidate sensing configuration* on the movement graph described above. A candidate sensing corresponds to a possible sensing action of the robot in a specific pose p, that is, it represents the possibility of performing a sensing action over a well defined area. Formally:

Definition 1: A *candidate sensing configuration* c_i is defined as a tuple (p_i, ϕ_i, r_i), where $p_i \in \mathcal{P}$, ϕ_i is the central angle of a circular sector and r_i its radius.

Hence, a candidate sensing configuration c_i would allow the robot positioned in p_i to scan a circular sector of central angle ϕ_i and radius r_i, as shown in Fig. 4(a). Because of obstructions, not all the cells in the circular sector would be necessarily observable.

Let us define $\mathcal{C}_\mathcal{P}$ as the set of all candidate configurations defined over a set of robot poses $\mathcal{P}$. We can then define the visibility function $v_\mathcal{P} : \mathcal{C}_\mathcal{P} \mapsto 2^{\mathbb{Z}^2}$, such that $v_\mathcal{P}(c)$ denotes the set of cells $\in \mathcal{S}$ visible from c. We define $v_\mathcal{P}$ so that a cell k is considered visible to a sensing configuration c_i if the line segment connecting the centers of c_i and k is in the circular sector of ϕ_i and r_i, and does not intersect any occupied cell, as shown in the example in Fig. 4(b). The cost associated to perform a sensing action in a candidate sensing configuration c is t^s_c and it depends on the central angle ϕ associated with c.

IV. Sensor Planning

We adopt a decoupled approach to solve our problem: First, we select a minimum set of candidate sensing configurations from which all cells in $\mathcal{S}$ are visible; then, we solve a Traveling Salesman Problem to find the shortest tour which connects them. An example of these steps is detailed in Fig. 1. While the opposite approach of first computing a minimum cost traveling route and then minimizing the sensing time by selecting the configurations along the path is possible, it has been shown in [21] that this approach does not scale well to realistic map sizes.

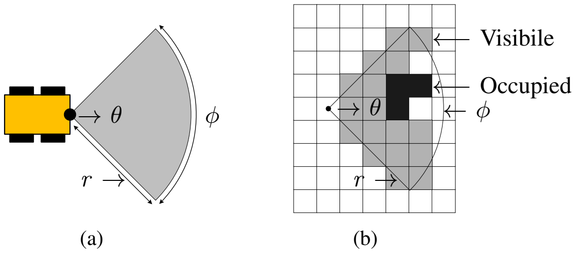


Fig. 4. 4(a) A candidate sensing configuration c allows the robot to scan a circular sector of central angle ϕ and radius r. 4(b) $v_\mathcal{P}(c)$ is the visibility function which defines which are the cell that are observable from candidate sensing configuration c.

We find the minimum set of candidate sensing configurations by solving the following integer linear programming problem:

$$\underset{C}{\text{minimize}}\; C^T T_s \tag{1a}$$

$$\text{subject to}$$

$$VC \succeq 1 \tag{1b}$$

$$C \in \{0, 1\} \tag{1c}$$

Where C is an ordered column vector of cardinality $|\mathcal{C}_\mathcal{P}|$ whose elements are binary variable representing if a given candidate sensing configuration is selected or not, and T_s is a column vector of size $|\mathcal{C}_\mathcal{P}|$ representing the sensing cost associated to each candidate sensing configuration ($T_s[i] = t^s_{c_i}$).

Eq. 1b requires that each cell $a_i \in \mathcal{S}$ is visible from at least one of the candidate sensing configurations selected in the solution. Given the problem definition, we can calculate V as a binary matrix of size $n \times |\mathcal{C}_\mathcal{P}|$, where n is the number of cells in the problem. In particular:

$$V[a, c] = \begin{cases} 1 & \text{if } a \in v_\mathcal{P}(c) \\ 0 & \text{otherwise} \end{cases}$$

To generate a shortest length tour, we find a minimum path between selected sensing configurations using the movement graph and then apply the Repetitive Nearest-Neighbor Algorithm [22] to calculate a Hamiltonian cycle.

V. Optimization with Convex Relaxation

To solve the discrete problem of finding the set of sensing configurations that covers the whole environment at minimum cost, we propose an iterative convex relaxation method. Our algorithm *conv*-SPP for Sensor Placement Problems drastically reduces the discrete combinatorial search through an initial convex optimization in continuous space.

The general problem is NP-hard and corresponds to ℓ_0-minimization. Solutions are binary vectors of variables, which describe configurations that are part of the solution. For all but relatively small problems, combinatorial explosion renders ℓ_0-minimization infeasible. For larger problems ℓ_1-minimization is a feasible alternative but the solution is not

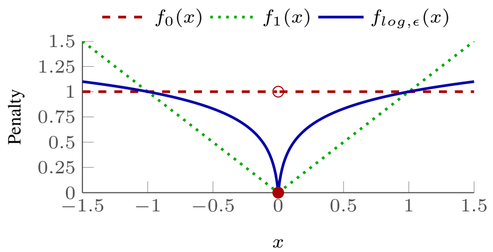


Fig. 5. At the origin, the canonical ℓ_0 sparsity count $f_0(x)$ is better approximated by the log-sum penalty function $f_{log,\epsilon}(x)$ than by the traditional convex ℓ_1 relaxation $f_1(x)$ [23].

any longer a binary vector. In the best case the discrete minimum equals the continuous minimum. In general, however, the discrete solution is worse. Thus, the continuous minimum provides a lower bound for the minimum number of discrete sensing configurations.

We need to discretize the resulting vector in order to produce a valid solution. However, the ℓ_1-minimization solution often contains many non-zero elements, which all have to be considered in the subsequent combinatorial search. In order to reach sparser solutions with more zero elements we apply a re-weighted (iterative) formulation of ℓ_1-minimization proposed by Candes et al. in [23]. Key to the approach is that effectively a concave loss function ($f_{log,\epsilon}(x)$) is used, which is much closer to the penalty function of ℓ_0-minimization ($f_0(x)$), see Fig. 5. Solutions of this non-convex optimization problem are found through convex optimization iterations reweighted with a weight decreasing according to Eq. 3.

This results in a sparser solution with more zero elements than after plain ℓ_1-minimization. The set of non-zero elements defines a reduced problem discrete search that is then solved by integer linear programming as formulated in Eq. 1. Formally, we solve the problem [23]:

$$\underset{C}{\text{minimize}}\ (W \circ C)^T T_s \tag{2a}$$

subject to

$$VC \succeq 1 \tag{2b}$$

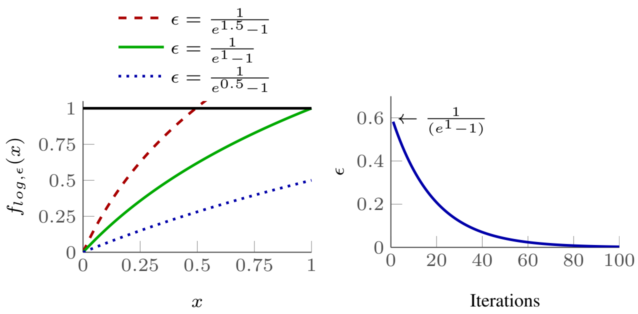
$$0 \preceq C \preceq 1 \tag{2c}$$

whereas

$$W_{(i)} = \frac{\epsilon}{C_{(i)} + \epsilon} \qquad i = 1, ..., |W| \tag{3}$$

Here, W is a weight vector of cardinality $|\mathcal{C}_{\mathcal{P}}|$. All elements are initially set to 1 and then evolve according to Eq. 3. The parameter ϵ determines the convergence rate. Smaller values of ϵ allow for faster convergence at an increased risk of getting trapped in local minima (see Fig. 6(a)). In our implementation we adapt the value of ϵ in each step i according to

$$\epsilon_{(i)} = \left(\frac{1}{e^1 - 1}\right)^{1+\left((i-1)\times 10^{-1}\right)} \tag{4}$$



(a) (b)

Fig. 6. 6(a) Sharper penalty function with smaller value of epsilon. 6(b) The exponential decay of ϵ during the iterative procedure.

That way, it decreases exponentially with each iteration (see Fig. 6(b)). The procedure stops if no improvement in sparsity for the last n iterations is observed, or a predefined number of iterations is reached. The overall procedure is summarized in Algorithm 1.

To discriminate between zero and non-zero elements, we use ℓ^0_τ sparsity measure with τ equals 10^{-2} in our implementation. In addition, we stop the iterative minimization if the number of non-zero elements is low enough so that the combinatorical problem becomes feasible.

Algorithm 1 *conv*-SPP

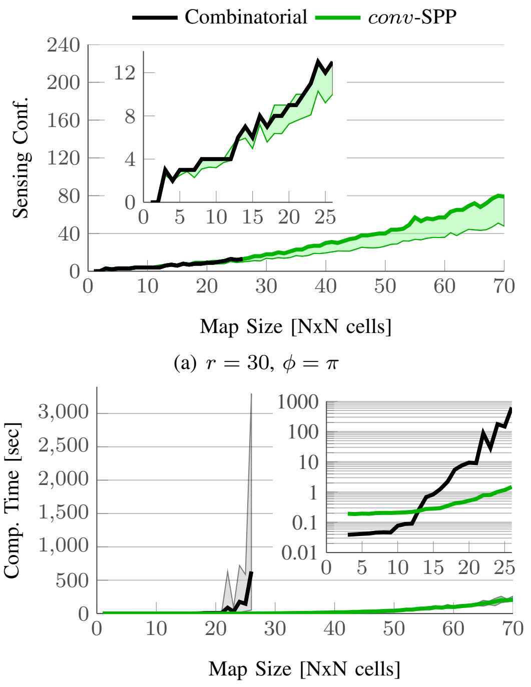
```
1: Set $W = \mathbf{1}$;
2: Solve Eq. 2;
3: Update $W$ as in Eq. 3 and $\epsilon$ as in Eq. 4;
4: Go back to step 2, if none of the stopping criteria is true;
5: Discard zero elements of $C$;
6: Solve Eq. 1 with updated $C$;
```

A. Evaluation

We evaluated our algorithm on maps with grid sizes ranging from 3x3 to 70x70 cells. For each size 10 different maps were generated of which 10% randomly selected cells we populated with obstacles. For 3x3 cells there are only three unique maps with a single occupied cell (approx. 10%). Therefore, the results for 3x3 maps are averaged using the unique 3 maps only. We use the following sensing parameters: $\Theta = \{0, \frac{\pi}{2}, \pi, \frac{3}{2}\pi\}$, $r = \{15, 30\}$ and $\phi = \{\pi, \pi/2\}$. For optimization we used the Gurobi Solver [24], with the CVX package [25]. The computations were carried out on a computer with an Intel i7 Quad Core 2.60GHz CPU and 8GB RAM.

We compared the exact solution of the brute force combinatorial search with the *conv*-SPP solution for up to 26x26 cells, after which the full combinatorial search starts to become infeasible. The results for sensing parameters $r = 30$ and $\phi = \pi$ are shown in Fig. 7(a). More detailed results can be found in [21]. The black line is the exact solution which should be compared to the upper bound of the colored interval, which is the *conv*-SPP solution. In addition, the lower bound discussed above is shown, which is the solution after the first iteration of the ℓ_1-minimization. Up to the map size of 26x26, for which the exact solution is available, the $conv$-SPP solution is very close to the exact solution in terms of the number of sensing configurations selected and not more than two sensing configurations off.

Fig. 7(b) shows the computation times for both approaches. While the computation time required for the exact solution starts to take longer than 1 hour for maps with 27x27 cells, the $conv$-SPP algorithm solves maps of size 70x70 in less than 5 minutes.



(a) $r = 30$, $\phi = \pi$

(b) $r = 30$, $\phi = \pi$

Fig. 7. 7(a): Comparison between $conv$-SPP and the brute force approach. The optimal solution is close to the solution of $conv$-SPP, which is the upper bound of the colored interval. The lower bound, obtained by a single ℓ_1-minimization step, is shown as well. 7(b): Computation times to calculate the optimal solutions and to solve the instances with $conv$-SPP. The thick lines represent the average computation times for each set of maps, while the colored intervals are bounded by the minimum and maximum times it took to solve all the instances in one set. The inset graphs show the computation times on a logarithmic scale.

VI. Experimental Runs

We tested our gas detection approach with a mobile robot in two different, large real-world scenarios, one indoor and the other outdoor. We were interested in assessing if our methodology can quickly calculate inspection plans for a mobile robot in complex environments, and if the resulting plans, once executed, can successfully identify the presence of gas leaks.

Our robotic platform is based on a Husky A200 equipped with a TDLAS sensor (Sewerin RMLD), as shown in Fig. 2. The sensor is mounted on a pan-tilt unit along with a 2D laser scanner in order to adjust the parameters (angle and radius) of the sensing configurations. The robot runs the Robot Operating System (ROS) and is further equipped with a 3D LiDAR for mapping and localization.

For both experimental scenarios we proceeded as follows: First, we created a map of the environment, which we later used also for NDT-MCL localization [26]. We then discretized the maps into Cartesian grids where the side of each cell was set to 0.5 m. The proposed planning algorithm is then executed using the obtained grid representations of the maps. For both setups, we set $\Theta = \{0, \frac{\pi}{2}, \pi, \frac{3}{2}\pi\}$, r equal to 10 m and ϕ to $\frac{\pi}{2}$. Finally, we executed the resulting plan with the robot, 10 times in the indoor scenario and once, as a proof of concept, outdoor. At each execution of the plan, we randomly placed 4 gas sources for the robot to detect. As there are safety, environmental and health concerns associated to releasing methane into the environment, we used 4 plastic tubes filled with methane as sources.

(a) Indoor

(b) Outdoor

Fig. 8. The experimental settings for our test runs.

A. Indoor Test Runs

For these test runs, we selected the large indoor environment (approximately 560 m^2) shown in Fig. 8(a), discretized in a map of 123x40 cells, 1126 of which unoccupied. Here, our algorithm generated a set of 18 sensing configurations, displayed in Fig. 9, which were subsequently connected in a minimum-length tour. As it can be seen, the selected sensing configurations cover all free cells and the plan compensates for limited or occluded field of view. In this scenario, the upper and lower bounds for the solution were 15.83 and 18, respectively, and the final solution was generated in 1.015 s.

As we observed a background methane concentration of approximately 100 ppm·m, we set a detection threshold of 1000 ppm·m. We executed the resulting plan 10 times. Each time, we randomly placed 4 gas sources close to walls along which there are pipes running. We recorded the position of the sources and all the configurations which gave us positive readings. Each run was completed in about 33 minutes.

As it can be seen from the results reported in Table I, every source in all the runs led to at least one positive reading (see Fig. 10(a) for an example of a successful execution run). More important still, there were no false negatives, which means that every time a gas source fell within the range of a sensing configuration, the sensing configuration reported a positive reading. Finally, we experienced a small number of false positives, that is, we got positive readings in sensing configurations which were not covering any of our sources (see Fig. 10(b)). There are several possible explanations for the false positives. For instance, the floors the corridor were highly reflective, and we could have registered unforeseen reflections of the TDLAS' beam which traversed sources placed outside the intended detection range.

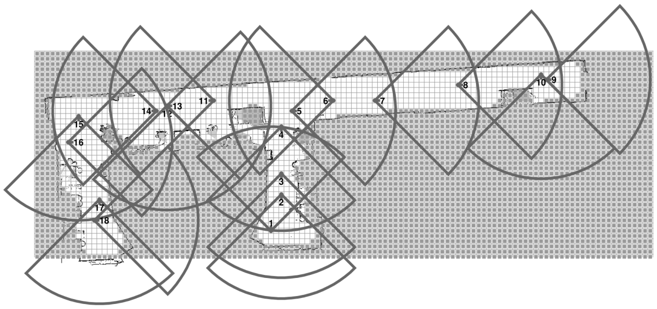


Fig. 9. The discretized map of the indoor experimental scenario. All cells have a side length of 0.5 m and the occupied ones are grayed out. The figure also displays the 18 selected sensing configurations and the area each of them covers on the map. The small number associated to each configuration indicates the order in which they are visited.

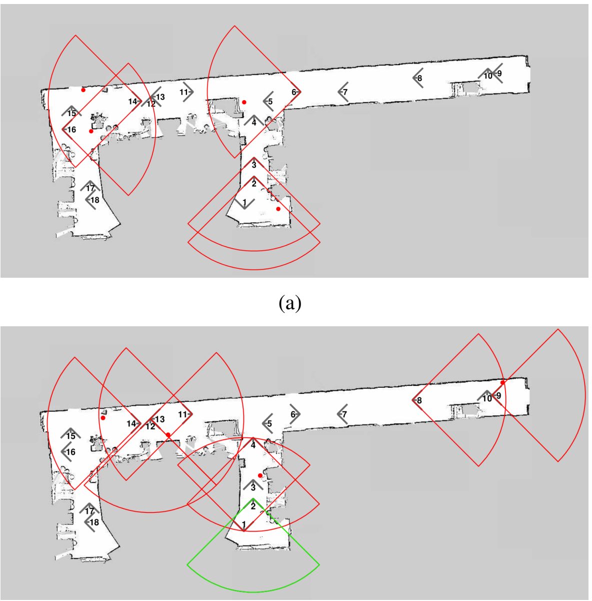
TABLE I

RESULTS OF THE INDOOR TEST RUNS.

Trial No.	1	2	3	4	5	6	7	8	9	10	Total
Gas Sources	4	4	4	4	4	4	4	4	4	4	40
False Positive	0	0	1	2	1	0	0	1	0	1	6
False Negative	0	0	0	0	0	0	0	0	0	0	0

B. Outdoor Test Run

We also performed a demonstrative test run in a large outdoor forest area, shown in Fig. 8(b). The area has an extension of approximately 900 m^2, which was discretized to a map of 81x33 cells, 3494 of which unoccupied. This scenario was meant to demonstrate the applicability of our methodology to environments with different operational conditions, which are very close to the ones one can find in some interesting application domains for our system (e.g., methane detection on landfills). Our algorithm generated a set of 19 sensing configurations, as shown in Fig. 11, then connected in a minimum-length tour. The selected sensing configurations cover all free cells and the plan compensates for the occlusions caused by trees. In this scenario, the upper and lower bounds for the solution were 15.72 and 19, respectively, and the final solution was generated in 21.01 s.



(a)

(b)

Fig. 10. Graphical summary of the results of two test runs in the indoor environment, where only the sensing configurations which reported positive readings are plotted. The gas sources deployed in each run are indicated by red dots. In the run corresponding to Fig. 10(a), there were no false positives, which can instead be noted in Fig. 10(b), where false positives are highlighted in green.

In this scenario, we used the same detection threshold of 1000 ppm·m we set for the indoor experiment and again we placed 4 gas sources in random locations within the map. We executed a run with the robot, which took approximately 40 minutes, in which all the sources were detected, as shown in Fig. 12.

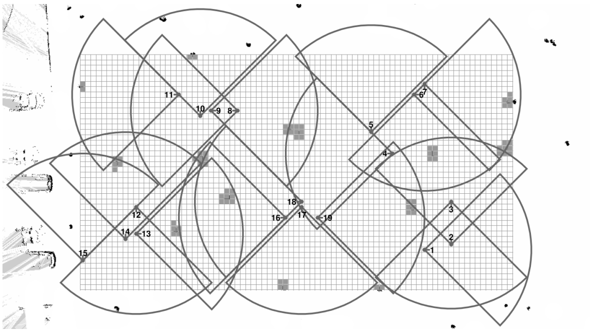


Fig. 11. The discretized map of the outdoor experimental scenario. All cells have a side length of 0.5 m and the occupied ones are grayed out. The figure also displays the 19 selected sensing configurations and the area each of the covers on the map. The small number associated to each configuration indicates the order in which they are visited.

VII. CONCLUSIONS

In this paper we addressed the problem of gas detection in large areas with a mobile robotic platform equipped with a TDLAS sensor. We proposed an algorithm, *conv*-SPP, for quickly finding an exploration plan which guarantees a complete coverage of the environment. Our algorithm leverages

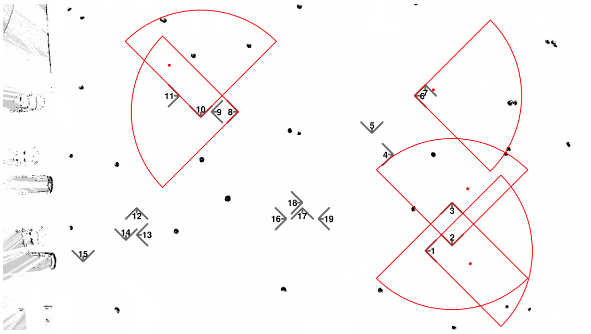


Fig. 12. Graphical summary of the outdoor test run, where gas sources are indicated by red dots and only sensing configurations which reported positive readings are highlighted.

a novel method based on convex relaxation that drastically reduces the number of variables and it is therefore effective for solving large problem instances. Furthermore, *conv*-SPP also offers guarantees on the quality of the solution, as it is always possible to asses the theoretical maximum distance of the solution provided to the optimal one. In practice, we show that in all the cases for which we could experimentally solve the problems optimally, our algorithm always generated results very close to the optimal ones. We validated our method in simulation and in real environments, both indoor and outdoor.

In our future work, we want to extend our problem definition to include non-traversable but observable cells.

References

[1] X. Liu, S. Cheng, H. Liu, S. Hu, D. Zhang, and H. Ning, "A Survey on Gas Sensing Technology," *Sensors*, vol. 12, no. 7, pp. 9635–9665, Jan. 2012.

[2] M. Trincavelli, M. Reggente, S. Coradeschi, A. Loutfi, H. Ishida, and A. Lilienthal, "Towards Environmental Monitoring with Mobile Robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Nice, France: IEEE, Sep. 2008, pp. 2210–2215.

[3] V. Hernandez Bennetts, A. J. Lilienthal, A. A. Khaliq, V. P. Sese, and M. Trincavelli, "Towards Real-World Gas Distribution Mapping and Leak Localization Using a Mobile Robot with 3D and Remote Gas Sensing Capabilities," in *IEEE International Conference on Robotics and Automation (ICRA)*. Karlsruhe, Germany: IEEE, May 2013, pp. 2335–2340.

[4] M. Lackner, "Tunable Diode Laser Absorption Spectroscopy (TDLAS) in the Process Industries- A Review," *Reviews in Chemical Engineering*, vol. 23, no. 2, pp. 65–147, 2007.

[5] H. Ishida, Y. Wada, and H. Matsukura, "Chemical Sensing in Robotic Applications: A Review," *IEEE Sensors Journal*, vol. 12, no. 11, pp. 3163–3173, 2012.

[6] G. Bonow and A. Kroll, "Gas leak localization in industrial environments using a TDLAS-based remote gas sensor and autonomous mobile robot with the Tri-Max method," in *IEEE International Conference on Robotics and Automation (ICRA)*. Karlsruhe, Germany: IEEE, 2013, pp. 987–992.

[7] P. Wang, R. Krishnamurti, and K. Gupta, "View Planning Problem with Combined View and Traveling Cost," in *IEEE International Conference on Robotics and Automation (ICRA)*. Roma, Italy: IEEE, Apr. 2007, pp. 711–716.

[8] B. Golden, Z. Naji-Azimi, S. Raghavan, M. Salari, and P. Toth, "The Generalized Covering Salesman Problem," *INFORMS Journal on Computing*, vol. 24, no. 4, pp. 534–553, 2012.

[9] Y. Tomioka, A. Takara, and H. Kitazawa, "Generation of an Optimum Patrol Course for Mobile Surveillance Camera," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 22, no. 2, pp. 216–224, 2012.

[10] D. T. Lee and A. K. Lin, "Computational Complexity of Art Gallery Problems," *IEEE Transaction on Information Theory*, vol. IT-32, no. 2, pp. 276–282, 1986.

[11] W. R. Scott, "Model-based view planning," *Machine Vision and Applications*, vol. 20, no. 1, pp. 47–69, 2007.

[12] U. M. Erdem and S. Sclaroff, "Automated camera layout to satisfy task-specific and floor plan-specific coverage requirements," *Computer Vision and Image Understanding*, vol. 103, no. 3, pp. 156–169, 2006.

[13] K. A. Tarabanis, P. K. Allen, and R. Y. Tsai, "A Survey of Sensor Planning in Computer Vision," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 1, pp. 86–104, 1995.

[14] D. Ramsden, "Optimization Approaches to Sensor Placement Problems," PhD Thesis, Rensselaer Polytechnic Institute, 2009.

[15] W. An, F.-M. Shao, and H. Meng, "The coverage-control optimization in sensor network subject to sensing area," *Computers & Mathematics with Applications*, vol. 57, no. 4, pp. 529–539, 2009.

[16] F. Angella, L. Reithler, and F. Gallesio, "Optimal Development of Cameras for Video Surveillance Systems," in *IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS)*. London, UK: IEEE, Sep. 2007, pp. 388–392.

[17] X. Chen and J. Davis, "An occlusion metric for selecting robust camera configurations," *Machine Vision and Applications*, vol. 19, no. 4, pp. 217–222, 2007.

[18] P. L. Chiu and F. Y. S. Lin, "A Simulated Annealing Algorithm to Support the Sensor Placement for Target Location," in *Canadian Conference on Electrical and Computer Engineering*. Ontario, Canada: IEEE, May 2004, pp. 867–870 Vol.2.

[19] D. L. Applegate, *The Traveling Salesman Problem: A Computational Study*. Princeton University Press, 2006.

[20] J. Faigl, M. Kulich, and L. Přeučil, "A Sensor Placement Algorithm for a Mobile Robot Inspection Planning," *Journal of Intelligent & Robotic Systems, Springer*, vol. 62, no. 3-4, pp. 329–353, 2010.

[21] M. A. Arain, M. Trincavelli, M. Cirillo, E. Schaffernicht, and A. J. Lilienthal, "Global Coverage Measurement Planning Strategies for Mobile Robots equipped with a Remote Gas Sensor," *Sensors*, 2015, to appear.

[22] G. Kizilate and F. Nuriyeva, "On the Nearest Neighbor Algorithms for the Traveling Salesman Problem," in *Advances in Computational Science, Engineering and Information Technology*, ser. Advances in Intelligent Systems and Computing, D. Nagamalai, A. Kumar, and A. Annamalai, Eds. Heidelberg: Springer International Publishing, 2013, vol. 225, pp. 111–118.

[23] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing Sparsity by Reweighted l1 Minimization," *Journal of Fourier Analysis and Applications, special issue on sparsity*, vol. 14, no. 5-6, pp. 877–905, 2008.

[24] "Gurobi Optimizer Reference Manual," 2014.

[25] M. Grant and S. Boyd, "CVX: Matlab Software for Disciplined Convex Programming, version 2.1," 2014.

[26] J. Saarinen, H. Andreasson, T. Stoyanov, and A. Lilienthal, "Normal Distributions Transform Monte-Carlo Localization (NDT-MCL)," in *EEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, Japan, 2013, pp. 382–389.