Mobile Robots with Novel Environmental Sensors for Inspection of Disaster Sites with Low Visibility

Project start: January 1, 2015

Duration: 3.5 years

Deliverable 5.1

Algorithms for Sensor Planning

Due date: month 36 (January 2018)

Lead beneficiary: ORU

Dissemination Level: Public

Main Authors:

Victor Hernandez Bennetts (ORU)
Asif Arain (ORU)
Erik Schaffernicht (ORU)
Achim Lilienthal (ORU)

Version History:

0.2: initial version, VH, June, 2018
0.1: initial version, VH, May, 2018

Contents

A Introduction and purpose of this document

In this deliverable, algorithms are proposed to efficiently conduct an exploration mission to cover the target area as fast as possible while acquiring relevant information. For example, the robot prototype should identify interesting areas where emitting gas source locations might be located. In the Smokebot scenario, the exploration strategy (robot trajectory and stop positions) is decided by the human operator. Thus, the planning algorithms presented in this deliverable should be able to suggest positions of interest (POI). These POIs are places where the multiple sensor readings will be most informative to integrate into the GDIM, will help localizing a gas source or where dropping a wireless repeater will ensure stable communication with the operator. Two different types of problems are considered: coverage - building an environment model of the whole operation area as quickly as possible - and exploitation - observing a specified small area for detailed information gathering over time.

The POIs are presented to the human operator, which then decides where to move the robot based on the priority that each of the tasks has at a given point in a given mission. For example, the operator might decide to prioritize POIs related to gas sensing, and move the robot accordingly, if localizing gas leaks is a priority during the mission.

In the following sections, the different sensor planning algorithms developed in Smokebot, are presented. Based on the sensors' characteristics, the algorithms are grouped as follows:

a) Sensor planning for in-situ gas sensors (Section B): For example, suggesting interesting positions to localize emitting gas sources.

b) Sensor planning for remote sensors (Section C): For example, suggesting possitions for LIDAR sensors to minimize thermal reflection.

c) Sensor planning for communication coverage (Section D): For example, predicting areas with low RF signal coverage ahead of the robot.

B Sensor Planning for In-situ gas sensors

In-situ gas sensors are devices that require direct contact with the analyte under study in order to report a measurement. This means that the reported measurement corresponds to only a few centimeters around the location of the sensor, which makes data acquisition time consuming. In order to efficiently conduct an exploration mission under limitations such as battery life (as in the case of Smokebot), a sensor planning strategy is required. We developed two different sensor planning strategies for in-situ gas sensors for the task of gas source localization. The first strategy is a model-free, data driven approach that creates a lattice where at each cell, an estimation of the distance to a gas source is stored. Areas that are predicted as being nearby the source can be considered as POI. The distance estimations at those cells not visited by the robot are computed using Gaussian regression guided by wind flow measurements. The second approach is model based. Specifically, it uses Partial Differential Equations for gas source localization. Constructing a realistic mathematical model of gas dispersion is complex and computationally expensive to solve. To address this problem, we propose a probabilistic model based on diffusion PDE to approximate the complexities of gas dispersion. The model suggest informative measurement locations as well as likely source locations. The proposed approaches were evaluated in real-world and with hardware-in-the-loop simulations respectively. The following sections present a deeper overview of the algorithms and the results achieved in the experimental validation processes.

B.1 Model-free sensor planning approach

This algorithm represents the target environment M as a lattice of cells of identical size: M={x1,...,xN}. The algorithm assumes that only one gas source is present in the environment. Regarding on-board sensing modalities, the algorithm assumes that the robot is equipped with gas and airflow sensors and moreover, it assumes that the robot's position is available.

The robot moves between the centers of the cells. After each movement it records measurements of gas concentration and wind information for a certain amount of time in order to compute the source distance estimation. We define a function f(x) that is unknown a-priori and is updated from noisy observations. f(x) indicates the distance to the source at location. Using measurements at visited locations, the robot extrapolates f(x) to estimate non visited locations. The algorithm uses these estimations to decide where to measure next in order to improve the model efficiently. The overall process is computed online as the robot traverses the target area.

Fig. 1(a) shows the three functional blocks of our proposed model-free algorithm. In the first block, an estimation of the distance to the gas source is computed. We use the method proposed by Schmuker and co-authors in [1], which correlates the variability of the sensor response to the distance to the source. A key element to the algorithm proposed by the authors is the detection of bouts in the sensor response. These bouts correspond to the rising edges of the sensor signal, as shown in the example in Fig. 1(b). In [1], it is reported that there is a strong correlation between the number of bouts and the distance to the gas source: the higher the bout count, the closer the sensor to the gas source. To detect the bouts, a cascaded filtering approach is used to detect fast transients in the sensor signal. A low-pass filter is first used in order to remove high-frequency noise.

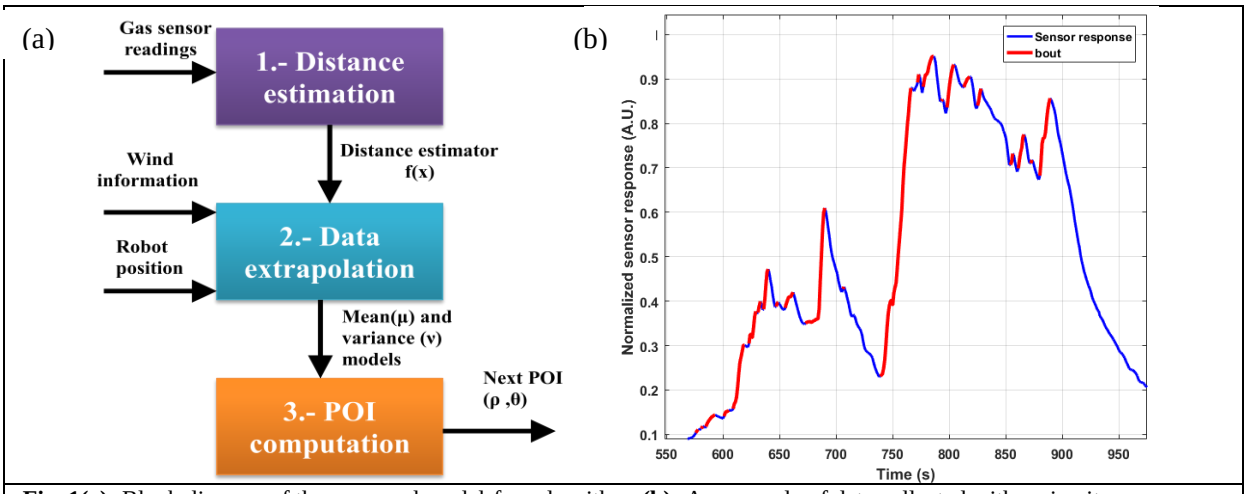

Fig. 1(a): Block diagram of the proposed model-free algorithm. **(b):** An example of data collected with an in-situ gas sensor. Notice the detected bouts in red.

To evaluate the applicability of the bout approach, we used an array of six commercial MOX sensors[1] on-board a mobile robot to collect data at different locations of an indoor corridor where a gas source is located. The gas source is a plastic container filled with ethanol. A bubbler facilitates evaporation while a fan placed nearby the source allows to spread the analytes away. As shown in Figure 2(a), no correlation between bout count and distance to the source is observed. This is in stark contrast with what is stated in [1]. The reason for this is that the authors conducted their experiments inside a wind tunnel with controlled airflow conditions. We proposed a novel indicator of source proximity using the bout approach. Instead of counting bout occurrences, we calculate the bout amplitudes. As can be seen

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

in Fig 2(b), the average bout amplitude is a good indicator of the distance to the gas source in uncontrolled environments. In some cases, when moving away from the source by 0.5 m the average amplitude slightly increases. However, in most cases, the average bout amplitude decreases when moving away from the source for more than 1 m.

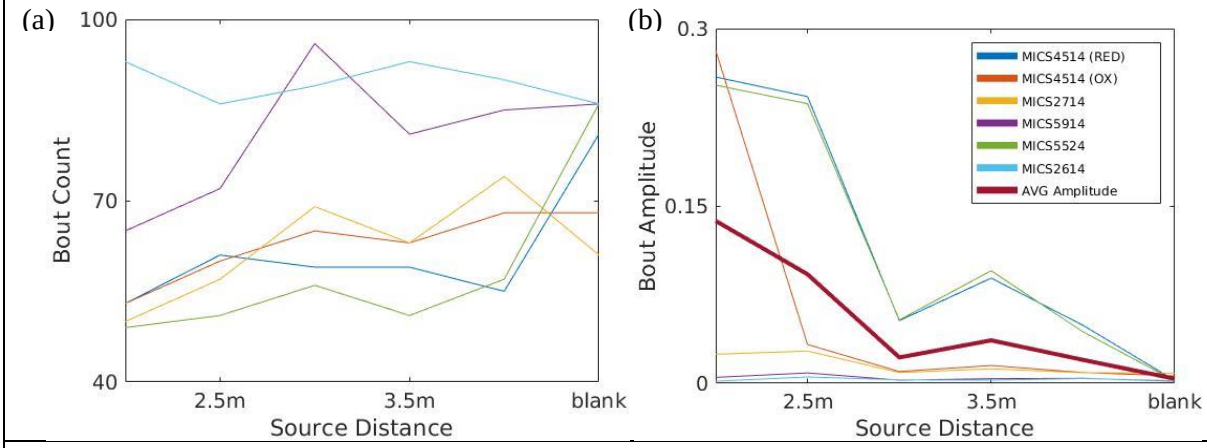

Fig 2: Performance of 2 proximity indicators computed from an array of MOX sensors. (a) bout count. (b) Bout amplitude.

The next step in the algorithm is the estimation of f(x) at non visited locations (Fig 1(a)). We thus model the average bout amplitude μ as a Gaussian process with a radial kernel function, stretched according to the wind direction as proposed in [2]. One of the advantages of using Gaussian regression process is that it estimates the a-posteriori variance *v*. The kernel corresponds to the assumption that positions in upwind direction have bout amplitudes similar to the measurement point. It also expresses the exploitation component of the exploration strategy, which leads the robot to follow a gas plume. Fig. 3(a) show examples the effect of the wind vector over the kernel while Fig. 3(b) shows the result of the extrapolation method from data collected with the robot.

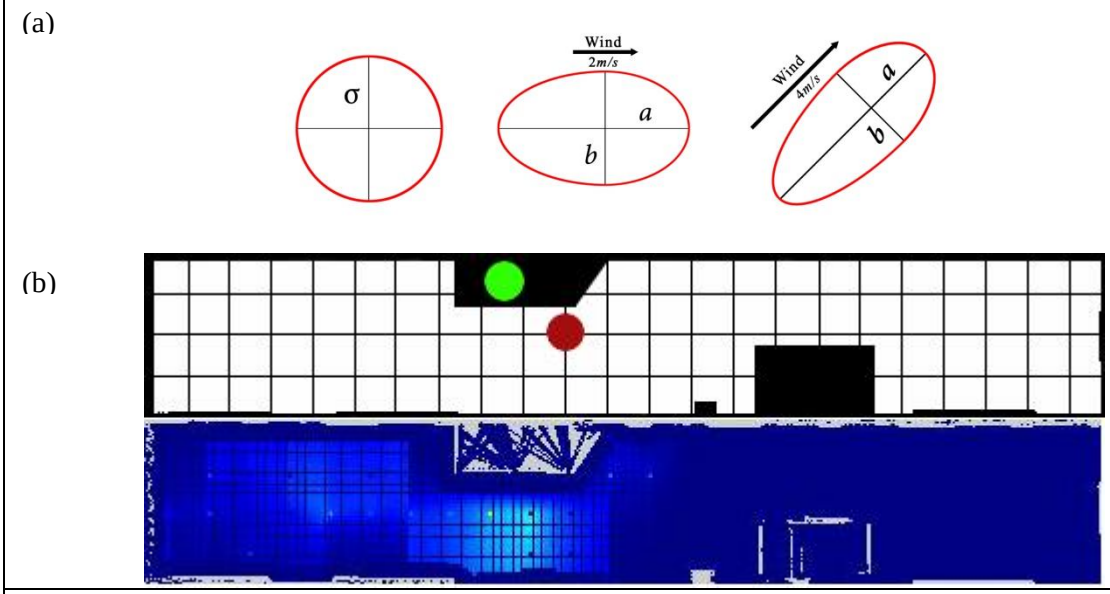

Fig. 3: (a) Influence of the wind speed and direction on the shape of the kernel. The ellipse is rotated according to the wind direction. The semimajor axis a is stretched in the upwind direction and shrunk in the downwind direction according to the wind speed. **(b)** An example of a bout amplitude map. In the upper plot (i.e. the grid map), the green circle indicates the gas source and the red circle the final position of the robot. The bottom plot corresponds to the mean bout estimate. The dark blue regions map areas with low bout amplitude estimates, whereas the light blue indicates high mean values.

To suggests Positions Of Interest (POI) to the operator (Step 3 in figure 1(a)), a trade-off between exploration of unvisited areas (following the variance gradient) and exploitation (following the direction to the highest bout amplitude estimate). The next POI corresponds to a movement of ρ meters from the current position with a direction of θ. The next POI will either be a position closer to the location with the highest variance (exploration) or a position closer to the highest bout amplitude estimate.

To test our proposed approach, we ran 12 experiments in the indoor scenario previously described. We consider an experiment successful if the robot chooses as the final position the reachable cell nearest to where the gas source is placed. A total of 8 runs were deemed successful, giving our method a 67% success rate. Reasons that could explain the failed experiments include the high wind speeds (which lead to bouts that cannot be resolved) and that the plane in which the robot sampled gas concentrations was at a substantially lower height than the gas source.

B.2 Model-based sensor planning approach

In Smokebot, we also developed a model-based approach for sensor planning. We used a gas dispersion model based on Partial Differential Equations (PDE), in order to address gas distribution mapping and gas source localization. To address concerns related to computational cost, we propose a probabilistic model based on diffusion PDEs to approximate the complex behavior of gas dispersion. Such model suggests likely locations of gas sources and to suggest POI to human operators.

Physical mechanisms causing gas propagation are not trivial, and in case of turbulence can even exhibit non-deterministic and chaotic behavior [3]. Nonetheless, for on-line mapping scenarios a simplified approximation of the physical phenomenon with low computational complexity might be of great use. To this end, we investigated the capability of a PDE to approximate spatial gas dynamics for the purpose of identifying sources that drive the gas propagation.

Regarding sensor planning, we build upon known methods that use, for example, linear-quadratic control techniques [4], optimal experimental design and probabilistic approaches [5]. We then propose an exploration strategy that minimizes the uncertainty of the source localization following a criterion similar to an A-optimality [6]. Additionally, similar to [7] we exploit the assumption that the sources causing gas dispersion are sparsely distributed and use sparse Bayesian learning techniques to model this.

From a practical perspective, approximating complex gas dynamics with simple models can be of an advantage. In Smokoebot, we use a 2D diffusion model that can be formally described with a linear parabolic partial differential equation (PDE) that creates a spatio-temporal model of the gas concentration as follows:

C.1	$$\frac{\partial f(\boldsymbol{x},t)}{\partial t} - \kappa \Delta f(\boldsymbol{x},t) = u(\boldsymbol{x},t)$$

In the above equation, **x** corresponds to a 2-D position, *t* corresponds to time and κ corresponds to a diffusion coefficient. The term u(**x**,t) on the right hand side represents a spatio-temporal inflow of material; in this work we aim to identify this process.

In order to solve a PDE, numerical approximation methods are often the instrument of choice. To this end we begin with a classical Finite Difference Method [8] which "transforms" our diffusion PDE into a finite dimensional linear system by appropriately discretizing both space and time. In other words,

we divide our region of interest into a finite number of grid cells and consider temporal evolution of concentration values in each cell at discrete time steps n.

To facilitate numerical estimation of the PDE, we assume that gas dispersion mechanisms are complex and not always deterministic and that gas concentrations are not observed directly, but are rather measured with a sensor. This permits us taking a probabilistic approach toward PDE solution by treating all variables as random. We also assume that the gas sources are sparse in space, i.e., the gas distribution is only driven by a few unknown discrete sources, which allows us to use Sparse Bayesian Learning (SBL) techniques [9, 10]. Using Bayes theorem, the overall problem is reduced to stimating the PDFs of variables such gas concentrations at the cells in the grid, the location of the gas sources. The derivation of the equations of our algorithm can be consulted in Appendix II and III.

With what respect to sensor planning and the identification of POI, we take advantage of our probabilistic framework. Specifically we consider that a high variance of the estimated gas source marginal PDFs implies a high uncertainty. This quantity can be used to rate all cells in the region of interest, and cells with highest uncertainties can be provided to the human operator as proposals for new measurement locations.

In addition, we implemented an automatic exploration strategy that can be used for one or several autonomous platforms. In our approach, the robot selects a cell to explore, based on the cell's uncertainty value and its distance to the current robot position. In other words, the robots not only prefer cells with high uncertainty but also cells which are close to their current positions. When a target is reached, a measurement is taken, which maximally reduces the uncertainty of the corresponding grid cells.

For evaluation purposes, we used a hardware-in-the loop approach where gas dispersion was generated using a state-of-the-art simulator. We generated a dynamic gas distribution driven by one source in an area of 20m times 60m. In this environment, we used our proposed approach and a small number of robots. Whenever a robot is triggered to collect a measurement, the simulated gas concentration at the robots' position is used as a synthetic measurement for the exploration algorithm.

Fig. 4 visualizes the result of the exploration at a certain time step, whereas Fig. 5 depicts the performance of the estimates over time. The source localization error is calculated by comparing the center of mass of the estimated source distribution with the actual position of the source used in the simulation. The concentration estimate is assessed by the Normalized Mean Square Error (NMSE) with respect to the concentration given by the simulator. Although the difference between the simulated gas concentration and the estimated concentration based on the simplified model is rather high due to the over simplifying diffusion approximation, the estimated source strength distribution is quite accurate and the robots are able to localize the source in a reasonable time frame with good accuracy.

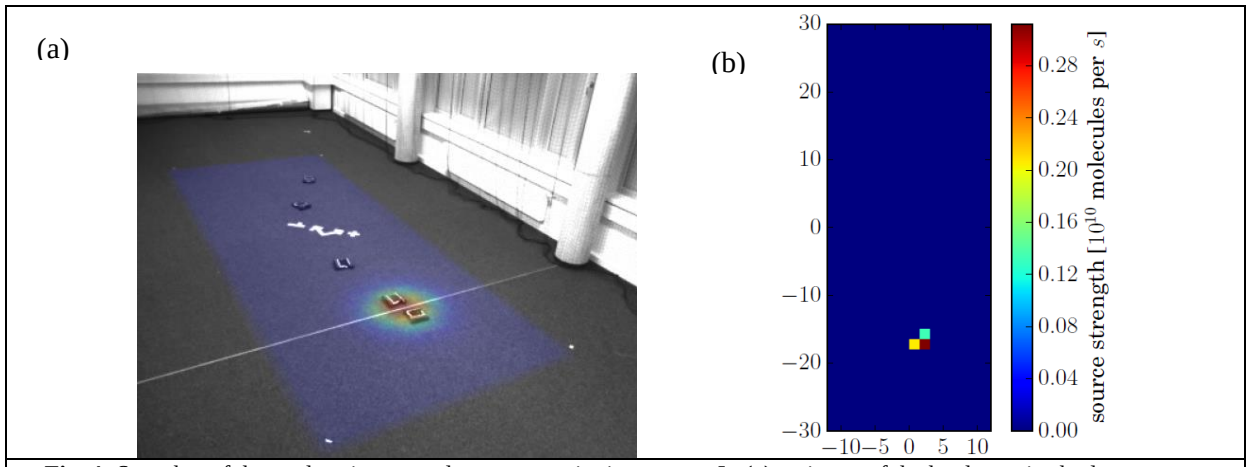

Fig. 4: Snapshot of the exploration procedure at a certain time stamp. In **(a)** a picture of the hardware-in-the-loop system where the simulated gas dispersion is shown as an overlay. Plot **(b)** displays the distribution of POI.

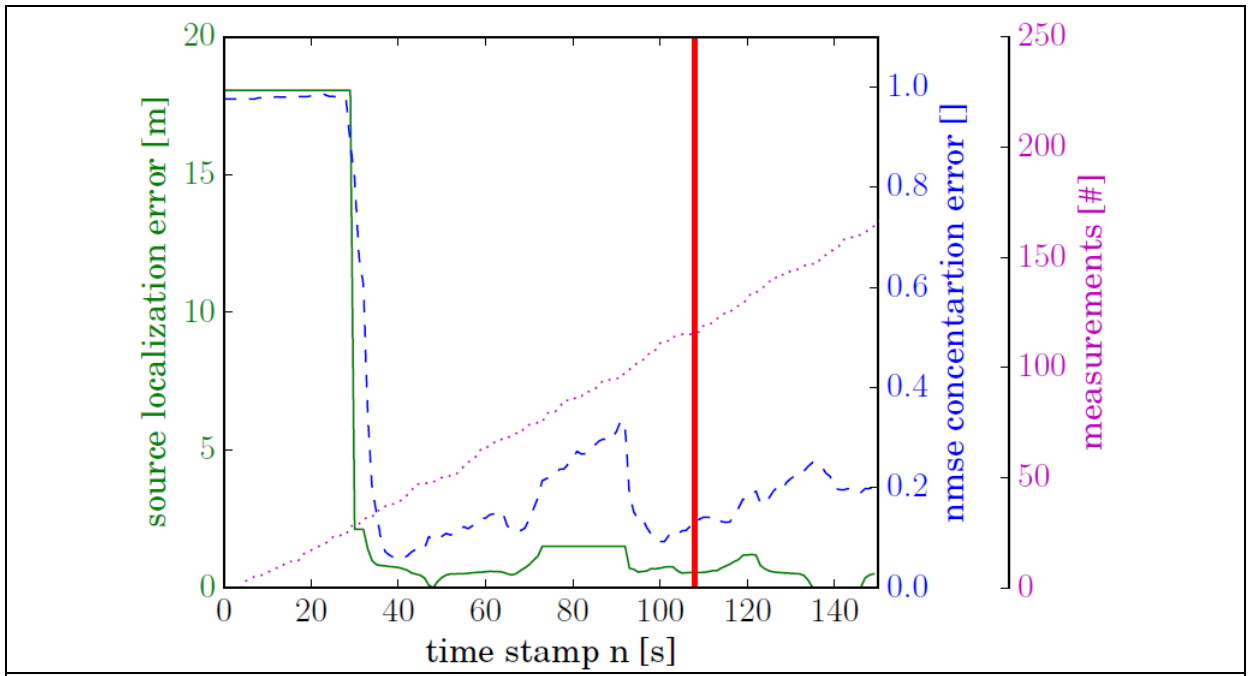

Fig. 5: Exploration performance. The vertical red marker denotes the time stamp at which Fig. 4(b) was generated.

C Sensor Planning for remote sensors

Remote sensors acquire information or measurements without direct contact with the object under study. Examples of remote sensors are LIDARs, gas sensors based on spectroscopy principles (TDLAS) and thermal cameras. In Smokebot, we developed an adaptive sensor planning algorithm that considers two key parameters in remote sensing, namely field of view and the range of the sensor. By field of view we refer to the extent of the observable world that is sensed at a given time. We define field of view by an angle through which the device is sensitive.

We test our proposed algorithm by addressing the task of temperature estimation using a remote thermal camera. This is a challenging task because thermography depends on environmental influences/conditions in many ways. Only to name a few, there are dependencies on material category, viewing angle, surface character, surrounding mediums, etc. The influence of these variables can led

to missinterpretations in the thermail images. As explained in the following subsection, the main challenge in thermography is the unknown emissivity when looking at objects. Emissivity can have a value between 0 and 1. An emissivity close to unity eliminates the reflected radiation intensity part, whereas an emissivity close to zero makes it difficult to infer the real object temperature.

C.1 Emissivity and temperature estimation

An example of the effects of emissivity in temperature estimation is shown in Fig. 6(a) where the whole thermal image is interpreted using an emissivity of $\varepsilon = 1$. In the front, there is a heated, polished aluminum plate (low emissivity) with a stripe of duct tape (high emissivity) on it. Although the aluminum plate and the duct tape have roughly the same temperature on the entire surface (about 330 K), their temperatures are interpreted differently. Due to the previously discussed relations between the true object temperature and the actually measured intensity, the temperature of the blank metal surface is interpreted much lower than the temperature of the dielectric duct tape.

Emissivity is highly dependent on the viewing angle. Depending on this angle the emissivity of metals and dielectrics behave differently. This effect is illustrated in Fig. 6(b). While for metals the emissivity is constant low for viewing angles between 0° and 60°, it then grows to a maximum just before reaching its minimum at 90°. For ideal dielectrics, the emissivity has a constant high value at viewing angles between 0° and 50°. At higher angles, it monotonically decreases to zero

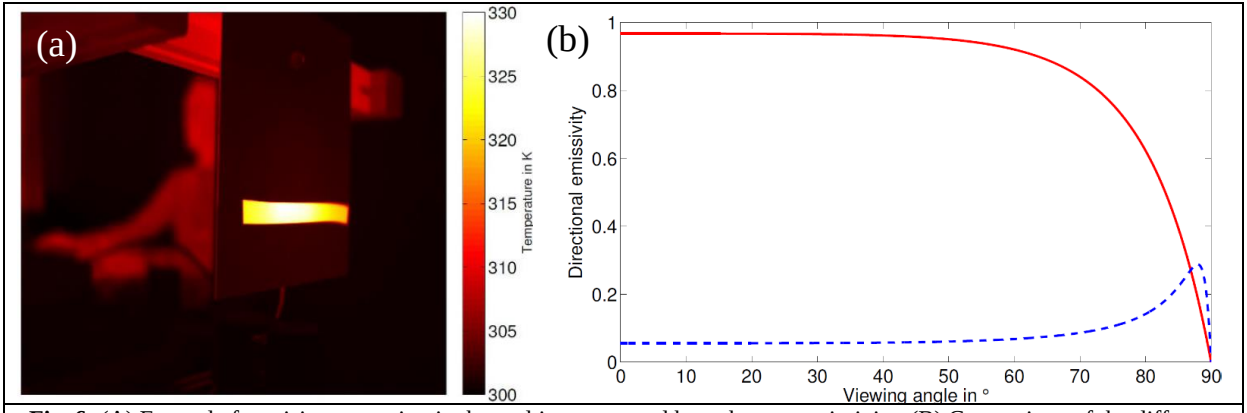

Fig. 6: (A) Example for misinterpretation in thermal image caused by unknown emissivity. (B) Comparison of the different emissivity characteristics of metals (dashed line) and dielectrics (solid line) at varying viewing angles.

Considering the above challenges, we developed in D3.1 an algorithm to estimate the actual temperature of a body regardless of the material it is made of. We framed the estimation of the emissivity and the temperature of a surface point as a least square minimization. In our approach, measurements (i.e. detector signals) from two different points of view are used to estimate parameters such as object's temperature (T_{obj}), refractive index (n) and extintion coefficients (k). The parameter T_{obj} corresponds to the temperature of the surface point while k and n are meta parameters of the emisivity model $\varepsilon(\theta)$, which depends on the viewing angle theta (as shown in Fig. 6(b)).

In the next subsection we explain our proposed sensor planning algorithm and we explain how it can be used in the task of temperature estimation. When a thermal camera is mounted on a mobile robot, data can be acquired at different poses (x,y,θ), which we refer to as a sensing configuration c. Thus, a solution for the surveillance task is a tour composed of a number of sensing configurations $\{c_1,c_2,...,c_n\}$. An efficient plan has a limited number of sensing configurations and a short traveling distance while at the same time, it provides a high sensing coverage with an accurate estimation of temperature profiles.

C.3 Adaptive sensor planning for remote sensor

As previously stated, Smokebot considers coverage and exploitation as key problems in sensor planning. In the context of remote sensing with thermal cameras, coverage refers to scanning the whole target environment in order to identify "areas of interest" that show high temperature levels. Exploitation refers to gathering detailed information about these "areas of interest" in order to estimate accurate temperature values.

In our proposed algorithm, we conduct exploration and exploitation in a single measurement tour. Initially, a set of sensing configurations that guarantee the full exploration of the area under surveillance are estimated (as originally presented in [11]). Then, when an interest location is identified (i.e. an area that shows high temperatures), exploitation is conducted by suggesting sensing configurations that minimizes the effect of the emissivity and thus maximizes the accuracy of the temperature estimation.

We represent the environment in a Cartesian grid of occupied and free cells. In addition, we consider a discrete number of candidate sensing configurations; this means that we consider only the locations at the centre of each cell and a limited number of orientations (e.g. 0º, 90º, 180º, 270º). The coverage is the capture by a matrix V of size equal to the number of candidate configurations times the number of free cells **S**.

A sensor planning solution for exploration is refered to as $\boldsymbol{\pi}_{\mathbf{detect}}$, which is a list of sensing configurations that provides the desired sensing coverage. $\boldsymbol{\pi}_{\mathbf{detect}}$ is estimated by solving the optimization problem in Equation C.1, where C is a binary decision vector of candidate sensing configurations, and $\boldsymbol{C}$ is the sensing coverage provided by the selected configuration. The parameter $\mathbf{n}_{\mathbf{cov}}$ determines the required coverage percentage.

C.1	$\pi_{\text{detect}} = \text{argmin } \lvert C \rvert \text{ s.t. } \mathcal{C} \geq \mathfrak{n}_{\mathbf{cov}}$

For the exploitation problem, we estimate $\boldsymbol{\pi}_{\mathbf{tomo}}$, which aims to provide accurate temperature estimations. The accuracy of the estimation is measured by $\boldsymbol{Q}$, which is an arbitrary quantity inversely proportional to the error between the estimated and the true temperature value. To determine $\boldsymbol{Q}$ we can use, for example, the emisitivy curves shown in Fig. 6(b). As $\boldsymbol{\pi}_{\mathbf{detect}}$ is being executed, areas of interest are identitied, which then need to to be sampled from different view points. Having different view points is critical for the temperature estimation algorithm, as explained in the previous subsection. Equation C.2 is the optimization problem for exploitation, where $\mathbf{n}_{\mathbf{tomo}}$ is the desired reconstruction quality.

C.2	$\pi_{\text{tomo}} = \text{argmin } \lvert C \rvert \text{ s.t. } \mathcal{Q} \geq \mathfrak{n}_{\mathbf{tomo}}$

Given the above solutions for the exploration and exploitation, our sensor planning algorithm can be summarized as follows: First, we estimate $\boldsymbol{\pi}_{\mathbf{detect}}$ and we start executing it by collecting measurements at the suggested locations. If high temperatures are detected, then $\boldsymbol{\pi}_{\mathbf{tomo}}$ is computed for all the neighbouring areas. The plan $\boldsymbol{\pi}_{\mathbf{tomo}}$ is iteratively executed and improved for the each selected configuration. At the end of an exploitation process, $\boldsymbol{\pi}_{\mathbf{detect}}$ is re-computed for the remaining unexplored areas. This process continues until no configuration is left in $\boldsymbol{\pi}_{\mathbf{detect}}$.

To illustrate how the proposed algorithm works, we ran a simulated environment using Matlab, were we considered a field of view $\varphi = 45º$, and a range r = 20m for thermal camera. Fig. 7 shows the environment setup (obstacle locations and locations of the high temperature bodies) and the robot's measurement configurations. The location of the high temperature bodies are denoted by red markers. $\boldsymbol{\pi}_{\mathbf{detect}}$ is shown as a series of green circular segments with angles and radius equal to the sensor's field

of view and range respectively. Notice that at configuration 21, high temperatures are reported near the topmost outher wall. This triggers the computation of π_{tomo}, which is a re-planning process for exploitation. In our example, π_{tomo} corresponds to the red circular segments, for example configuration 22 when the first gas source is detected. After π_{tomo} is executed, π_{detect} is re-computed and those configurations that are no longer needed are discarded (gray markers in Fig. 7). Similarly, when another high temperature body is detected by configuration 11, the computation of a new π_{tomo} is triggered, which leads to the suggestion of configuration 12 (in red) for exploitation purposes.

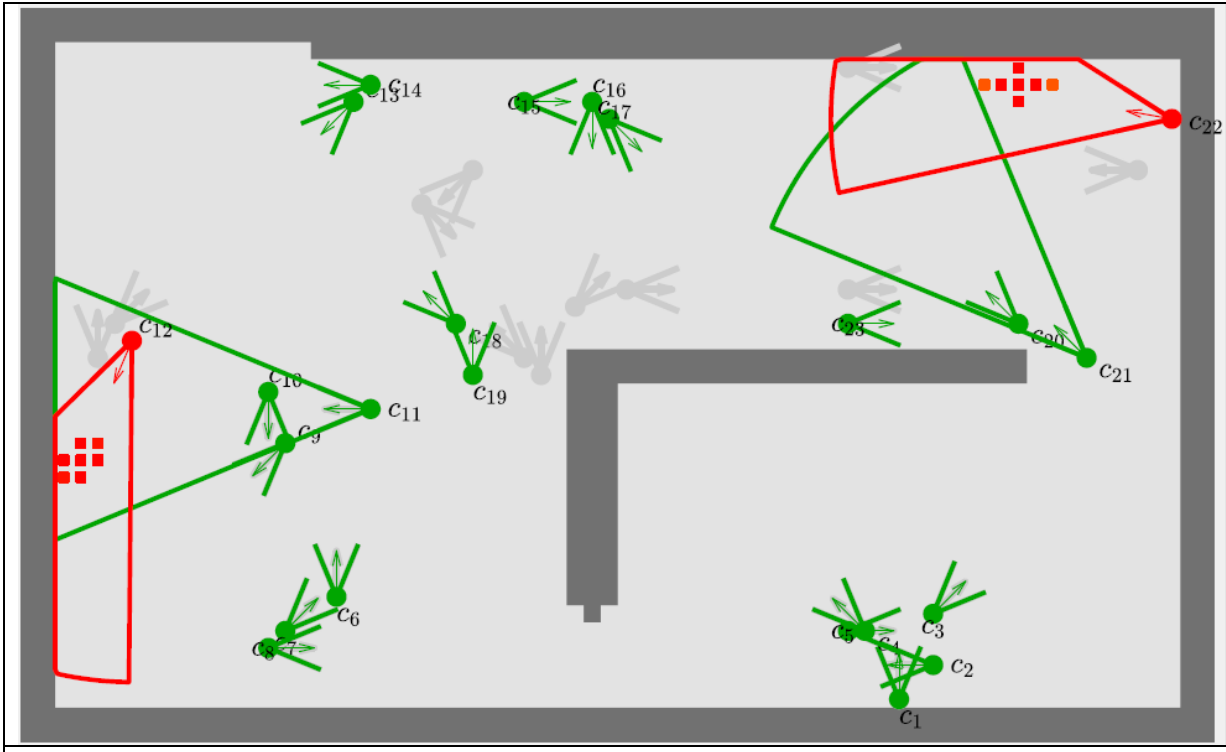

Fig.7: Simulation environment where an exploration tour is conducted. π_{detect} is shown in green while π_{tomo} is shown in red. The location of high temperatures are denoted by squared red markers.

D Sensor planning for communication coverage

To maintain a communication link between the human operator and the robotic platflorm, Smokebot developed a series of ruggedized wireless repeaters that are deployed as the mission is being conducted (T7.3) in order to extend the range of the Wireless (WLAN) network. However, it could be the case that the robot goes out of the WiFi range when in a mission. As part of the self preservation functionality (T5.2) , the robot will continuously monitor radio frequency (RF) signal strength and augment the General Disaster Model with it.

To predict the connectivity of the network and to suggest a possible location to drop a repeater, Smokebot developed a ray-tracing-based algorithm that estimates the WLAN range of the network and extend it if needed by suggesting POIs where more repeaters could be dropped (D5.2).

Ray tracing is widely used in computer graphics to render high quality images: For each pixel on the screen, the path of the light is traced back into the scene it is coming from; the individual pixel is then colored in the color of the object(s) the light ray encounters. This method can not only be used to generate images, it is also used in physics to simulate the behavior of waves and particles.

Ray tracing is a suitable approach to simulate WiFi equipment since the estimation of WLAN coverage needs to be computed in real time and approaches that use, for example, the equations that

describe the propagation of electromagnetic waves, could provide a subpar performance due to computational requirements. Related work, such as [11, 12] have shown that ray tracing can be used to simulate WiFi with a good level of accuracy. Moreover, commercial tools to plan wireless networks are based on ray tracing[2].

We implemented the proposed ray-tracing-based algorithm for WiFi coverage in ROS. We modeled each repeater as a laser scanner device with a range that depends on the traversed objects. Thus, the scanner code was implemented in such a way that the projected rays do not stop on the first object they hit but instead, they keep track of the traveled distance and the number of hit objects. The maximum range is then adjusted according to the traversed objects. In addition, if another scanner is encountered this is recorded and the information can be used to create a map of connected network devices. The result of the modified ray tracing can be seen in Fig. 9. The area that is covered by the Wifi repeater in the middle is colored red. It can be seen that the signal reaches the furthest in the bottom of the image, while it is attenuated by walls in the other directions.

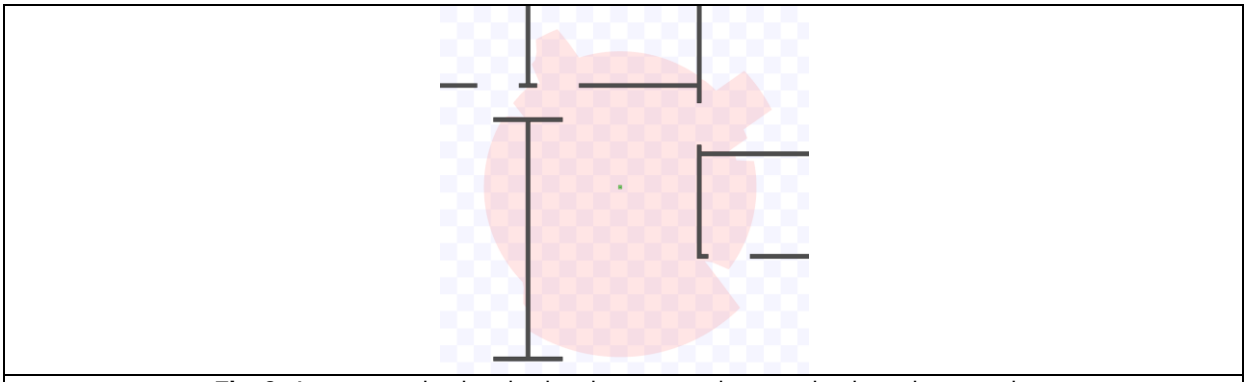

Fig. 9: A repeater simulated using the proposed ray-tracing-based approach.

Figs. 10(a) and 10(b) show how the proposed algorithm works. The algorithm was implemented as a ROS node that listens to the topics coming from the simulated repeaters currently in use. For each repeater, a polygon consisting of the translated points is kept updated, and to check if a point on the map is covered by the Wifi network one simply has to check if the point lies within one of these polygons. The algorithm also considers the input from the user in the form of the next position computed by the path planning algorithm. If the user commands the robot outside the area of coverage, a new Wifi repeater is placed early enough so that the robot always stays connected. Fig. 10(a) shows the coverage at the start of the exploration path and after it moves from its starting point in the lower left corner to the destination in the upper right corner Fig 10(b). Whenever the robot it is about to get outside the Wifi range, another repeater is dropped. The benefit of this approach is that it is completely independent of the specific path planner that is used.

[2] https://www.remcom.com/wireless-insite-em-propagation-software/

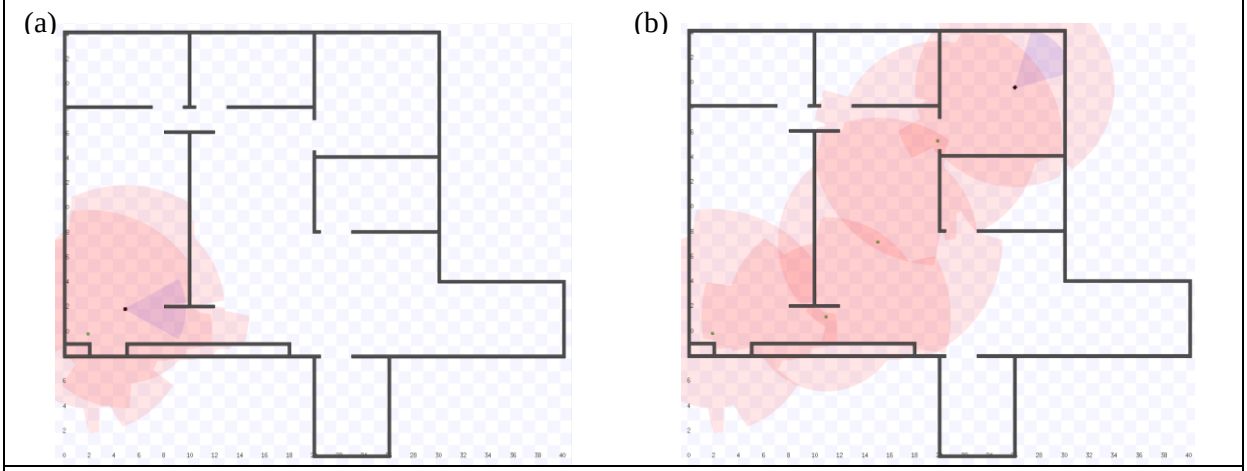

Fig 10: (a) Network coverage after the robot starts its exploration tour. (b) Network coverage after the exploration tour. In order to extend the connectivity of the network, several repeaters are dropped during the exploration path.

E Summary and Outlook

In Smokebot we have developed different algorithms for sensor planning based on the sensor characteristics. More specifically, we developed algorithms for remote sensing, in-situ sensing and communication coverage. A key characteristic of Smokebot is that the human operator is in control of the exploration missions and thus, the sensor planning algorithms suggest new measurement locations as Positions Of Interest (POI) instead of controlling the robot's trajectory. It is up to the user operator (and the task that has higher priority at a given point) to decide where to move the robot.

For in-situ sensor planning, we explored two different approaches for gas source localization and gas distribution mapping. We first proposed a sensor planning algorithm that does not make any a-priori assumption about the gas dispersion (i.e. a model-free approach). This approach suggests POIs based on the variability of the gas measurements and airflow data and for non-visited locations, Gaussian extrapolation is used. This extrapolation model then allows to suggest POIs to explore the target area and to exploit the acquired information in order to detect the gas leaks. While the use of wind data could be perceived as a limitation of the algorithm in the Smokebot scenario, a possible solution is to use an isotropic Gaussian kernel. Regarding our model-based approach, we found that the performance. In addition, we proposed a model-based approach that relies on PDE to localize the gas sources and to map gas distribution. This algorithm proposes POIs based on hypotheses of where the gas source location might be (exploration) and areas where the uncertainty can be reduced (exploration). In our hardware-on-the-loop evaluation, we found that the algorithm converges rapidly towards the location of the gas source and that the algorithm is rather accurate when localizing gas sources.

Regarding sensor planning for remote devices, we use an adaptive sensor planning algorithm for temperature estimation using a thermal camera. The sensors planning algorithm first computes a set of POI for exploration (e.g. a given sensing coverage) and then, modifies the original plan as data is being collected in order to maximize a given performance metric (i.e. improving the temperature estimates by decreasing the effects of the emissivity). The algorithm presented in this report is flexible enough to be used with different remote sensors beyond thermal cameras since it depends on generic parameters such such as field of view (φ) and sensor range (r). Parameters such as $\mathbf{n}_{cov}$ (Equation C.1) denote the desired coverage (exploration) and it is not bound to the particular task of temperature estimation. Similarly, $\mathbf{n}_{tomo}$ (Equation C.2) is a parameter that denotes a desired performance level.

To conclude, we addressed sensor planning for the task of communication coverage. For this particular task, we implemented an ad-hoc algorithm that models the coverage of the individual repeaters as range sensors and considers the interference of obstacles in order to determine the overall coverage of the network. The proposed algorithm considers the estimated coverage and the next robot position (provided by the motion planning algorithm) in order to determine whether or not the robot is going out of range. If it is determined that the robot is moving outside the area of coverage, a repeater is dropped. While our ad-hoc algorithm diverges from the POI approach, it allows nevertheless for future developments. For example, it would be possible to provide the information about the network coverage to a specialized path planner that could then take this information into account and plan the route so it tries to stay in range of the WiFi.

References

[1]	M. Schmuker, V. Bahr, and R. Huerta, "Exploiting plume structure to decode gas source distance using metal-oxide gas sensors", Sensor and Actuator B-Chemical, vol. 235, pp. 636-646, 2016
[2]	M. Reggente and A. J. Lilienthal, "Using Local Wind Information for Gas Distribution Mapping in Outdoor Environments with a Mobile Robot", in Proc. IEEE Sensors, pp. 1715-1720, 2009.
[3]	Roberts, P., and Webster, "Turbulent Diffusion. Environmental Fluid Mechanics Theories and Application", Reston, VA: ASCE Press,2002.
[4]	M. Demetriou, "Numerical investigation on optimal actuator/sensor location of parabolic PDEs," Proceedings of the 1999 American Control Conference, vol. 3, no. June, pp. 1722–1726, 1999.
[5]	F. Pukelsheim, Optimal Design of Experiments. John Wiley & Sons, Inc., 1993.
[6]	A. Alexanderian, N. Petra, G. Stadler, and O. Ghattas, "A-optimal design of experiments for infinite-dimensional Bayesian linear inverse problems with regularized l0-sparsification," SIAM Journal on Scientific Computing, vol. 36, no. 5, p. 27, 2013.
[7]	K. Kunisch, K. Pieper, and B. Vexler, "Measure Valued Directional Sparsity for Parabolic Optimal Control Problems," SIAM Journal on Control and Optimization, vol. 52, no. 5, pp. 3078–3108, jan 2014.
[8]	Strikwerda, J.C.: Finite Difference Schemes and Partial Differential Equations, 2 edn. Society for Industrial and Applied Mathematics, Philadelphia (2004)
[9]	M. E. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine," Journal of Machine Learning Research, 2001.
[10]	D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," Signal Processing, IEEE Transactions on, vol. 52, no. 8, pp. 2153–2164, 2004.
[11]	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, 15(3):6845-6871, 2015
[12]	T. R. S.Y. Seidel, „A ray tracing technique to predict path loss and delay spread inside buildings," 1992.
[13]	R. Valenzuela, „A ray tracing approach to predicting indoor wireless transmission," 1993.

Appendixes