Autonomous Gas Distribution Mapping using Multi-Informed Tree Planning

Mal Fazliu, Matthew Coombes, Cunjia Liu
Department of Aeronautical and Automotive Engineering, Loughborough University, UK LE11 3TU
Email: {m.fazliu2, m.coombes, c.liu5}@lboro.ac.uk

***Abstract*—To date, mobile robotic gas distribution mapping (GDM) platforms largely depend on remote operation. Automating these platforms eliminate the need for trained specialists on-site and improve the performance of GDM. This paper addresses the environment exploration-gas exploitation dilemma that arises when attempting automation of GDM in unknown and cluttered environments through a multi-objective informed tree planning strategy. Assuming the frontier exploration techniques can be employed to identify several potential destinations, we focus on developing an anytime performance Multi-Informed Tree (MIT) algorithm, building on the Batch Informed Trees (BIT*) framework, to simultaneously construct information-driven paths towards these goals. Gas exploitation in our problem is addressed by these trajectories which are sampled from an underlying Upper Confidence Bound (UCB) distribution generated from the current gas map state. The path with the highest information gain is subsequently selected for the robot to execute. When tested against the popular RRT and RRT* algorithms, MIT consistently achieves higher information gain in its evaluated trajectories. The proposed strategy can be integrated into an informative path planning (IPP) framework in the future work.**

I. Introduction

Rapid and precise acquisition of situational information during a hazardous gas release is crucial to emergency responses [1]. The raw information deemed most useful in these scenarios are the gas concentration levels. To gather this data, sending a human operator with a gas sensor is not a viable option due to the life-threatening risks presented by the hazardous plumes and uncertain environment. Instead, mobile robotic platforms equipped with chemical sensors have become more common [2], as they can navigate areas unsafe for humans and endure more extreme conditions. However, even with these platforms, interpreting gas measurements remains challenging without tools to localise each reading and create a gas distribution model. To address this, one popular technique that has been widely adopted is Gas Distribution Mapping (GDM). GDM aims to generate a map of the gas concentration levels by collecting spatiotemporally distributed chemical measurements over a region of concern [3]. With such informative media at hand, first responders can then correctly identify regions reporting higher gas concentrations from which logical inferences regarding the presence of leaks and their locations can be derived, allowing them to manage the situation more safely and precisely.

At present, the mobile robot requires remote operation to perform practical GDM [4, 5]. This setup, however, has certain drawbacks. For instance, remote specialists need to be in close proximity to the scene to ensure a quick response, which can be a significant burden when lives are in danger. Additionally, humans are not as capable of finding optimal, time-effective paths to goal locations as established path planning algorithms like A* or RRT* [6]. The incentive to incorporate autonomy into GDM can therefore be attributed to the inconveniences associated with teleoperation and the potential to optimise the GDM procedure through path planning feedback.

This paper presents preliminary work towards the first practical execution of autonomous mobile robotic GDM in real-world conditions, that is, in unknown and cluttered environments. To navigate these unknown terrains, we will utilise simultaneous localisation and mapping (SLAM) functionality within our robotic platforms. Similar to the active SLAM problem [7], closing the loop for active GDM by supplying feedback on where the agent should next visit, presents a dilemma. At any instance, the robot must decide whether to prioritise destinations that enhance its understanding of the current gas distribution within familiar areas or explore new territories where gas dispersion might be more significant. To address the challenge of balancing environment exploration and gas exploitation, we opt to select multiple frontier points rich in information for the robot to potentially navigate towards. Subsequently, informed paths will be generated towards these points to exploit the gas. To achieve this, we extend the BIT* algorithm [8], originally designed for finding an informed path to a single destination, to facilitate the simultaneous construction of these paths towards multiple locations. This is accomplished by sampling from a UCB distribution of the current gas map (shown in Fig. 1) that we also introduce to guide the tree-building process for the GDM application. The trajectory with the highest accumulative UCB gain per distance travelled is then executed by the robot. It should be noted that unlike Gammell et al. [8], our method does not initiate another sampling batch upon completing one. This is because we are not interested in finding the shortest path towards each frontier region, but only some admissible informed ones.

With the recent introduction of the Gaussian Belief Propagation (GaBP) factor graph solver, coupled with a novel hybrid message scheduling system, robotic GDM platforms can now perform real-time inference updates while accounting for uncertain and obstacle rich conditions [5]. We plan to leverage this solver for a Gaussian Markov Random Field (GMRF) gas representation, initially proposed in [3], to ensure real-time application in our wider autonomous GDM study.

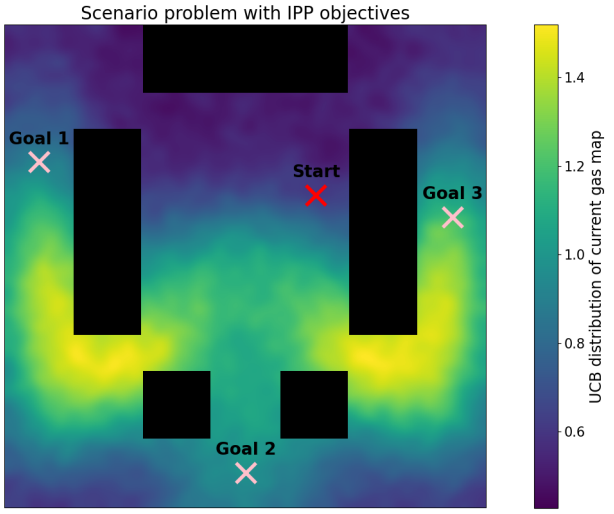

Fig. 1. The synthetic obstacle-rich GDM scenario used in this study depicting a snapshot of the UCB distribution derived from the latest gas map mean and uncertainty states. The start point indicates the current position of the robot and each goal point mimics a frontier location that would be provided from an external frontier detection function.

Our first contribution, and the focus of this in-progress paper, is the development of the MIT algorithm and its application to GDM. We demonstrate, through early simulation results, its superiority over the popular RRT and RRT* algorithms in obtaining information-rich paths to goals in a static hazardous gas release scenario revealed in Fig. 1. However, first, we provide an overview of related work in autonomous GDM to contextualise our contribution.

II. Related Work

Previous studies on autonomous GDM, recognised as an IPP problem, often suffer from incompleteness or impracticality in terms of attaining a fully automated system capable of addressing real-world hazardous release scenarios. For instance, the sensor-simulated (non-robot) IPP study in [6] and the novel K-means clustering solution proposed in [9] perform GDM under a predetermined OGM. A mobile robot however, when faced with a new scene, will not have access to such prior knowledge; the robot must build its own OGM using SLAM.

Strategies that involve the sequential execution of paths to locations evaluated by cost functions that only consider gas map states, such as those tested in [6], do not produce functional solutions, especially when the underlying OGM requires real-time construction. In these cases, sub-optimal behaviour often manifests as a repetitive loop between two high-cost regions that the robot continuously revisits, resulting in a lack of further exploration and exploitation of the gas distribution beyond this pattern. Gongora et al. [10] address this matter and propose an information-driven infotaxis strategy for gas exploration, but SLAM is still not included, and their POMDP IPP framework struggles to scale computationally with their GW-GMRF GDM function for real-time performance.

Sampling strategies for real-world applications should actively infer both gas map and SLAM information to achieve a practical and effective solution as the problem is not only to exploit the gas but also to explore the environment. Studies like [11], which attempted such work, fell short of practical success in complex, obstacle-tight environments because of their use of the Kernel DM+V GDM function. The computational demands of this function left little room for higher-quality path planning, path following, and SLAM algorithms to be implemented. To elaborate, their sampling strategy involved switching between a gas exploitation state and an environment exploration state, during which frontiers would be visited. However, they struggled to achieve an intelligent way to switch between these states without introducing incident-specific tuning parameters.

III. Formulation and Approach

Let the state space for the IPP problem be defined as $\mathbf{X} \subset \mathbb{R}^2$, $\mathbf{X}_{\text{obs}} \subset \mathbf{X}$ be the states obscured by obstacles, and $\mathbf{X}_{\text{free}} := \mathbf{X} \setminus \mathbf{X}_{\text{obs}}$ be the admissible states. Let $\mathrm{x}_k \in \mathbf{X}_{\text{free}}$ be the robot location at sampling time k.

Let σ_k^{k+1} be a collision-free path executable by the robot starting from x_k and closing at x_{k+1}, defined as $\sigma : \mathbb{R} \to \mathbf{X}_{\text{free}}$. Let the set Σ contain such paths towards a set of goal regions, $\mathbf{X}_{\text{goal}} \subset \mathbf{X}_{\text{free}}$, that exists for each frontier. The IPP problem is defined as finding a path $\sigma^* \in \Sigma$ that maximises a utility function $\Psi(\cdot)$, so that

$$\sigma^* := \arg\max_{\sigma \in \Sigma} \{\Psi(\sigma) \,|\, \sigma(0) = \mathbf{x}_k\} \tag{1}$$

The notation used in the next sections is consistent with that of the original BIT* paper [8] and is not repeated here.

A. Initialisation (Alg 1, Lines 1:7)

Alg 1 begins by generating N samples through the informed UCB distribution shown in Fig. 1, given by:

$$\text{UCB}(x, y) = \frac{\mu(x, y)}{\max_{(x', y') \in \mathcal{G}} \mu(x', y')} + \lambda \cdot \epsilon(x, y), \tag{2}$$

where $\mu(x, y)$ and $\epsilon(x, y)$ are the latest mean gas concentration and uncertainty, respectively, obtained from the GDM function at coordinates (x, y), λ is a weighting parameter, and $\mathcal{G}$ is the continuous domain of the environment.

It then defines the goal states, x_{goals}, leveraging the external frontier detection function, $\mathbb{E}(\delta_s^{(k)})$. Each goal state, x_{goal}, is allocated to a corresponding goal set, X_{goal}, comprising the k_n closest instances of $x \in G$. Next, the root node, x_k, is used to initialise the tree's vertex set, V, and the vertex queue, Q_v, while both the tree's edge set, E, and the edge queue, Q_e, start empty. The Lebesque measure is then defined in a manner similar to [8] to help calibrate an appropriate radial boundary for edge search (Alg 1, Line 7).

B. Multi-Goal Tree Expansion Procedure (Alg 1, Lines 8:28)

The tree expands until Q_v is empty or a feasible path to all X_{goals} is established. The expansion is processed incrementally for each X_{goal}, as delineated from line 9 of Alg 1. If a feasible path to a particular X_{goal} is already established, indicated by

Algorithm 1: Multi-Informed tree expansion $(N, x_k, \lambda, \delta_s)$

1 $G \leftarrow \text{sample}(N, \lambda)$ `// from Eq. (2)`
2 $x_{\text{goals}} \leftarrow \mathbb{E}(\delta_s^{(k)})$
3 **for** x_{goal} *in* x_{goals} **do**
4 $\quad X_{\text{goal}} \leftarrow \{\arg\min_{x \in G} \|x_{\text{goal}} - X\|_2 \mid |X| = k_n\}$
5 $\quad X_{\text{goals}} \xleftarrow{+} X_{goal}$
6 $V \leftarrow x_k; E \leftarrow \emptyset; Q_{\text{e}} \leftarrow \emptyset; Q_v \leftarrow V$
7 $r \leftarrow 2k(1+\frac{1}{n})^{\frac{1}{n}} \left(\frac{\lambda(\hat{X}_f)}{\xi_n}\right)^{\frac{1}{n}} \left(\frac{\log(|G|)}{|G|}\right)^{\frac{1}{n}}$
8 **while** $Q_v \neq \emptyset$ & $g_\tau(x) \in X_{goals} = \infty$ **do**
9 $\quad$ **for** X_{goal} *in* X_{goals} **do**
10 $\quad\quad$ **if** $Q_v \neq \emptyset$ & $g_\tau(x) \in X_{goal} < \infty$ **then**
11 $\quad\quad\quad$ **continue**
12 $\quad\quad$ **else**
13 $\quad\quad\quad v_m \leftarrow \arg\min_{x \in Q_v} g_\tau(x) + \hat{h}(x)$
14 $\quad\quad\quad Q_v \xleftarrow{-} v_m$
15 $\quad\quad\quad V_{\text{near}} \leftarrow \{w \in G \mid \|v_m - w\|_2 \leq r\}$
16 $\quad\quad\quad Q_e \xleftarrow{+} \{(v_m, w) \in V_{\text{near}}\}$
17 $\quad\quad\quad$ **while** $Q_e \neq \emptyset$ **do**
18 $\quad\quad\quad\quad w_m \leftarrow \arg\min_{w \in Q_e} \hat{c}(v_m, w) + \hat{h}(w)$
19 $\quad\quad\quad\quad Q_e \xleftarrow{-} (v_m, w_m)$
20 $\quad\quad\quad\quad$ **if** $g_\tau(v_m) + \hat{c}(v_m, w_m) \leq g_\tau(w_m)$ **then**
21 $\quad\quad\quad\quad\quad$ **if** $g_\tau(v_m) + c(v_m, w_m) \leq g_\tau(w_m)$ **then**
22 $\quad\quad\quad\quad\quad\quad$ **if** $w_m \in V$ **then**
23 $\quad\quad\quad\quad\quad\quad\quad E \xleftarrow{-} \{(v, w_m) \in E\}$
24 $\quad\quad\quad\quad\quad\quad$ **else**
25 $\quad\quad\quad\quad\quad\quad\quad V \xleftarrow{+} w_m$
26 $\quad\quad\quad\quad\quad\quad\quad Q_v \xleftarrow{+} w_m$
27 $\quad\quad\quad\quad\quad\quad E \xleftarrow{+} (v_m, w_m)$
28 **return** (V, E)

$g_\tau(x) \in X_{\text{goal}} < \infty$, the algorithm proceeds to the next goal region without further action in the current iteration. Otherwise, the expansion continues to explore potential paths towards any unachieved X_{goal} sets (Alg 1, Lines 13:27). Adopting a simultaneous goal system enables users to opt for early termination of the programme, while guaranteeing the generation of a tree that extends branches towards all X_{goals}. As a result, anytime performance characteristics are embedded in the algorithm, thus facilitating flexible execution and the prompt delivery of partial solutions if necessary.

In the absence of a feasible path to an X_{goal}, the strategy unfolds by selecting the state $x \in Q_v$ with the lowest current tree cost-to-come plus heuristic cost-to-go, denoted as the expansion node, v_m (Alg 1, Line 13). Incorporating the vertex's tree cost-to-come into the sorting mechanism guarantees the best available information is directing the branch construction process. The state corresponding with the chosen v_m is then removed from Q_v. The group V_{near} is formed around the expansion node by including all states $w \in G$ that fall inside the radius, r, of v_m. Connections from each w to v_m are subsequently queued into Q_e.

Once Q_e is populated with potential edges, the edge processing procedure begins by identifying the w within Q_e that offers the minimal estimated cost pathway from v_m to the closest state within X_{goal} via w (Alg 1, Line 18). Subsequently, this selected edge is extracted from Q_e. Admission of the edge into the tree is conditional on compliance with the conditions in Alg 1, Lines 20:21. If w_m is yet to be added to the tree, compliance is guaranteed as $g_t(w_m)$ will evaluate to ∞. Following this scenario, w_m will be added to both the vertex set V and the vertex queue Q_v (Alg 1, Lines 25:26), with the edge (v_m, w_m) being included in the edge set E (Alg 1, Line 27). Should w_m already be part of the tree, a rewiring process may occur, entailing the removal of the previous edge (v, w_m) (Alg 1, Line 23) and the integration of the new edge (v_m, w_m) (Alg 1, Line 27). This occurs provided that the cost to arrive to v_m plus the actual cost (accounting for collisions) $c(v_m, w_m)$ is lower than the existing cost to reach w_m. As per the notation in [8], if a collision is encountered, $c(v_m, w_m)$ evaluates to ∞. Notably, the computation of $c(v_m, w_m)$, which can be resource-intensive due to collision assessments, is strategically delayed as per Alg 1, Line 20. Here, an initial evaluation of the estimated cost, $\hat{c}(v_m, w_m)$, ascertains if progressing to the collision cost analysis in the following line is warranted.

IV. Experimental Results

The polluted and obstacle-rich scenario depicted in Fig. 1 is utilised to assess the performance of the MIT algorithm in finding informed paths towards each synthetic frontier. The information metric used to evaluate each trajectory is given by:

$$\Psi(\sigma) \approx \frac{1}{n_z} \sum_{l=1}^{n_z} \widehat{w}_{k+1}^{(\sigma,l)} \tag{3}$$

Here, $\Psi(\sigma)$ represents a trajectory's utility, calculated as the average of UCB values across all visited cells l along trajectory σ at time $k+1$. The parameter n_z denotes the total number of cells covered by trajectory σ. Each $\widehat{w}_{k+1}^{(\sigma,l)}$ signifies the estimated UCB value, as defined in Eq. (2), associated with the l^{th} cell on trajectory σ.

The MIT algorithm is initalised with the parameters $N = 270$, $\lambda = 1$ and $k_n = 3$ for the experiment, with Euclidean distance heuristics utilised between its states. To compare its performance with that of the popular RRT and RRT* algorithms, the same objectives were assigned to each algorithm. Since both RRT and RRT* are of 1 sample/batch nature, their respective parameters were adjusted fairly to ensure a similar total number of sampling points and termination criteria as the MIT algorithm. The results of 250 Monte Carlo simulations demonstrate that MIT is significantly more effective in finding

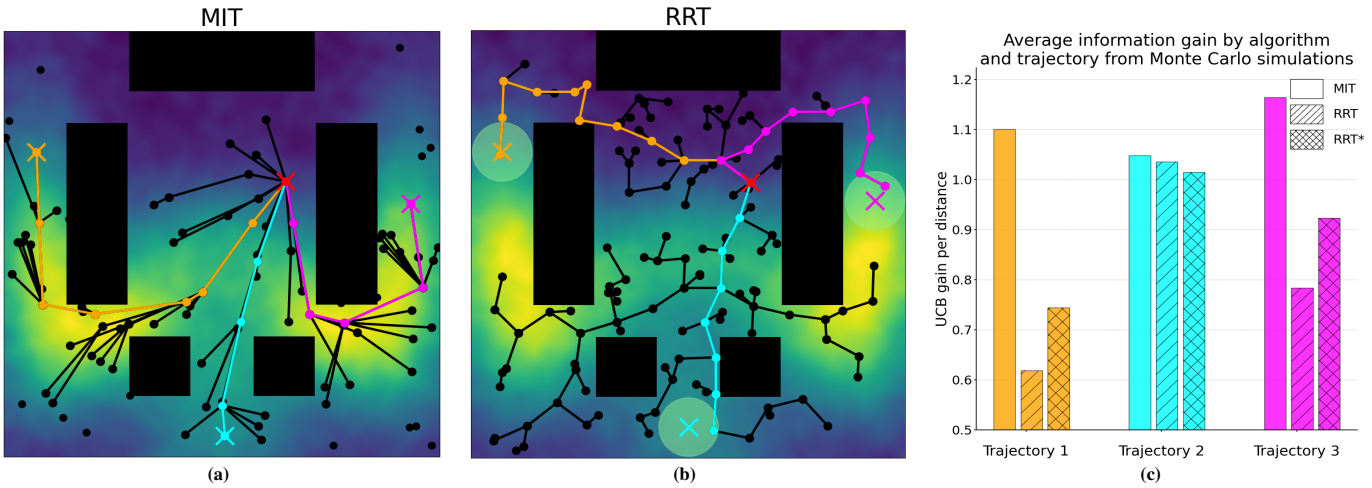

Fig. 2. An example of the most typical trajectories generated by MIT is shown in (a), and for RRT in (b). Notably, the most typical trajectories for RRT* are similar to those shown in the RRT example; therefore, a separate RRT* snapshot is not included here. Each trajectory that finds frontiers 1, 2, and 3, as defined in Fig. 1, is represented by the colors orange, cyan, and magenta, respectively. The green circles in (b) represent the termination radii used by both the RRT and RRT* methods. Trajectories from these methods conclude their tree construction tasks upon reaching each of the three regions. Finally, (c) provides a clear depiction of the information gain (UCB per distance) achieved on average for each trajectory and algorithm from the 250 Monte Carlo simulations.

valuable paths to goal locations (see Fig. 2c). In Fig. 2a, an example of common paths found by MIT is depicted, while Fig. 2b illustrates those for RRT (and RRT*), revealing their tendency to prioritise the fastest paths towards goals due to the absence of informed sampling. Table I shows that only MIT evaluated trajectory 3, the best candidate trajectory according to Fig. 2c, more frequently than the other two trajectories.

TABLE I
PERCENTAGE OF SIMULATIONS WHERE MIT, RRT, AND RRT* IDENTIFY EACH TRAJECTORY AS HAVING THE HIGHEST INFORMATION GAIN.

	Trajectory 1	Trajectory 2	Trajectory 3
MIT	21.6%	8.8%	69.6%
RRT	6.4%	56.0%	37.6%
RRT*	12.0%	47.6%	40.4%

V. CONCLUSION

A strategy for realistic autonomous GDM has been proposed to balance the exploration-exploitation dilemma faced by a robot with no prior knowledge of its surroundings. Our contribution in this ongoing paper, presented as the MIT algorithm, extends BIT* to form informed trajectories towards multiple destinations simultaneously, fostering the generation of numerous potential paths for evaluation. Within our strategy, these destinations are identified as frontiers, motivating the robot to navigate towards exploration-rich regions. The tree is constructed following batch sampling from a UCB distribution of the current GDM states, which ultimately guides the trajectories in an informed manner to these destinations, ensuring effective gas exploitation along the way. Experimental results show that MIT consistently outperforms RRT and RRT* in finding more informed paths to the frontiers within the static hazardous gas release scenario used for this study. The next step is to implement the algorithm in a more complex simulation setting that incorporates the GaBP-GMRF function to simulate GDM, and to integrate real frontiers extracted from SLAM.

REFERENCES

[1] R. R. Murphy, J. Peschel, C. Arnett, and D. Martin, "Projected needs for robot-assisted chemical, biological, radiological, or nuclear (cbrn) incidents," *2012 IEEE International Symposium on Safety, Security, and Rescue Robotics, SSRR 2012*, 2012.

[2] C. Rhodes, C. Liu, and W. H. Chen, "Autonomous source term estimation in unknown environments: From a dual control concept to uav deployment," *IEEE Robotics and Automation Letters*, vol. 7, pp. 2274–2281, 4 2022.

[3] J. G.Monroy, J. L. Blanco, and J. Gonzalez-Jimenez, "Time-variant gas distribution mapping with obstacle information," *Autonomous Robots*, vol. 40, pp. 1–16, 1 2016.

[4] A. Francis, S. Li, C. Griffiths, and J. Sienz, "Gas source localization and mapping with mobile robots: A review," *Journal of Field Robotics*, vol. 39, pp. 1341–1373, 12 2022.

[5] C. Rhodes, C. Liu, and W. H. Chen, "Structurally aware 3d gas distribution mapping using belief propagation: A real-time algorithm for robotic deployment," *IEEE Transactions on Automation Science and Engineering*, 2023.

[6] ——, "Informative path planning for gas distribution mapping in cluttered environments," *IEEE International Conference on Intelligent Robots and Systems*, pp. 6726–6732, 10 2020.

[7] J. A. Placed, J. Strader, H. Carrillo, N. Atanasov, V. Indelman, L. Carlone, and J. A. Castellanos, "A survey on active simultaneous localization and mapping: State of the art and new frontiers," *IEEE Transactions on Robotics*, vol. 39, pp. 1686–1705, 6 2023.

[8] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Batch informed trees (bit): Sampling-based optimal planning via the heuristically guided search of implicit random geometric graphs," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, pp. 3067–3074, 6 2015.

[9] C. Ercolani, L. Tang, A. A. Humne, and A. Martinoli, "Clustering and informative path planning for 3d gas distribution mapping: Algorithms and performance evaluation," *IEEE Robotics and Automation Letters*, vol. 7, pp. 5310–5317, 4 2022.

[10] A. Gongora, J. Monroy, F. Rahbar, C. Ercolani, J. Gonzalez-Jimenez, and A. Martinoli, "Information-driven gas distribution mapping for autonomous mobile robots," *Sensors (Switzerland)*, vol. 23, 6 2023.

[11] Y. A. Prabowo, B. R. Trilaksono, E. M. Hidayat, and B. Yuliarto, "Utilizing a rapidly exploring random tree for hazardous gas exploration in a large unknown area," *IEEE Access*, vol. 10, pp. 15 336–15 347, 2022.