

A novel odor source localization system based on particle filtering and information entropy

Hongbiao Zhu, Yibo Wang, Chengjin Du, Quan Zhang, Weidong Wang*

Robotics Institution, Harbin Institute of Technology, China

ARTICLE INFO

Article history:

Received 7 January 2020

Received in revised form 25 June 2020

Accepted 2 August 2020

Available online 11 August 2020

Keywords:

Source localization

Particle filtering

Information entropy

Pseudo source filtering

Artificial potential field

ABSTRACT

So far, gas leakage caused by natural or human factors has led to serious consequences in terms of social security. Previous strategies for locating the odor sources appear to be either defective or incomplete. For enhancing the success rate and rapidity, this paper aims to present a novel and complete strategy in search of lurking gas sources. Particle filtering and information entropy are both employed to track the plume information. To improve the tracking efficiency in this process, a novel objective function is designed by considering the entropy gains of the suspected targets as well as the repeated exploration scores. Considering the pseudo sourced caused by obstacles, a statistics-based source determine algorithm is proposed to confirm the source's authenticity, while the artificial potential field method is subsequently applied to eliminate the distractions introduced by the pseudo sources. Simulations and on-site tests are both carried out while results showed that the proposed scheme is competent to complete sources localization task in the scene that contains randomly distributed obstacles and pseudo source.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

Dangerous gas diffusion caused by natural or man-made factors could pose a huge safety hazard to our real life. How to locate the gas sources after accident leakage has already become a research concern over the last few decades in terms of emergency action and anti-terrorist system. Amid the existing strategies, static sensor network [1] is widely considered as an effective method to monitor gas sources, which had shown its feasibility in many applications background. However, there are still few deficiencies causing undesirability, i.e. complex installation, heavy maintenance cost, limited flexibility, and the need of certain mathematical model to determine the sensor arrangement [2]. With the development of mobile robot, the application of mobile sensor networks (MSNs) on different vehicles such as unmanned ground vehicles [3], underwater robots [4], drones [5,6], and even multiple robots [7–9], has drawn the researchers' focus. Normally, the process of searching dangerous gas sources by MSN system consists of three parts: plume discovery, plume tracking, and gas source confirmation [10]. The odor information are gathered during the plume discovery process, after which the robot could quickly approach the gas source under the navigation of existing knowledge, namely plume tracking. Gas source confirmation process is to both distinguish whether

the suspected gas source is real and confirm its location. For better classify the existing strategies, Hutchinson M et al. [11] divided the gas source localization algorithms into bio-inspired source location [12,13] methods and Bayesian-based source location methods. Bio-inspired methods typically imitate the foraging and social behavior of animals, such as silkworm moths and dung beetles. These strategies are regarded as reactive search strategies without introducing probability factors. For example, Braitenberg V et al. [14] designed a robot that relies on concentration gradient for source localization. This method is inspired by the foraging behavior of lobsters, which search for food by climbing concentration gradient. Ishida's et al. [15] introduce a zigzag method by imitating the dung beetles. This strategy incorporates the information obtaining from gas sensors and wind sensors, controlling the robot to move upwind in a z-shaped pattern until the robot reaches the gas source. With low-cost introduction and decent performance, this method has been commonly employed in other research [16]. After the proposition of zigzag algorithm, Lino Marques et al. [17] compared the performances of three existing source tracking strategies, i.e. strategy based on the bacterial chemical tendency (climbing concentration gradient), strategy based on the silkworm chemical tendency (zigzag search against the wind) and the direct gradient tracking strategy in the turbulent environment, eventually demonstrating that the zigzag strategy worked the best. However, the experiment of the above research were relatively crude without considering dynamic factors such as gas accumulation in complex scenes.

* Corresponding author.

E-mail address: wangweidong@hit.edu.cn (W. Wang).

Biologically inspired algorithms formed the main trend of the early research about plume source localization. However, external disturbances such as sensor error and wind fluctuation that could compromise the success rate and efficiency of the search process in low concentration environment are generally not included in this kind of algorithm. On the contrary, as one kind of cognitive searching method, source localization algorithms based on Bayesian updating generally take several random factors such as sensor errors, intermittent turbulence, etc. into consideration. This family of strategies is much robust than the bio-inspired methods, leading the main trend of modern source positioning algorithm. Essentially, source localization algorithm based on Bayesian framework is to perform source term estimation (STE) [9,18–28] under the framework of Bayesian theorem. The post-probability density functions of the source parameters such as position, release rate and so forth can all be given by Bayesian estimation. For instance, Massimo Vergassola et al. [18] proposed a method to estimate the position of the source by using mesh-based nonlinear filter, assuming that the other parameters of the source to be part of the prior knowledge. This method possesses inferior feasibility as the system parameters such as source release rate, particle life, etc. are basically unknown in real practice. It then led to the research focus of using particle filtering method to estimate the source terms [9,18,27,29,30]. For example, Hutchinson M et al. [29] estimate seven source parameters synchronously by particle filtering. They use a two-dimensional gas diffusion model to represent the number of the gas particles that detected by the gas sensor. According to the principle of maximum entropy sampling, the reward function is defined as the entropy of the expected measurement distribution (instead of the entropy of the expected posterior probability). The sequential Monte Carlo method is also employed to estimate the source parameters, showing the direction with maximum entropy for the next sampling. In order to reduce particle dissipation, the resample rule is subjected to the Markov chain Monte Carlo (MCMC) move algorithm. Unfortunately, experiment of this strategy did not include any physical obstacles, while external factors such as gas turbulence and gas accumulation were also not taken seriously, leading to a success rate of nearly 100 percent. Similar to [29], Michael H et al. [30] obtained a better performance by solely improving the reward function. Neumann P P et al. [5] put forward a composite scheme that combines biological inspiration method and particle filter. To both improve accuracy and efficiency while guaranteeing robustness, the strategy adopts a pseudo-gradient algorithm to assist particle updating.

However, due to the external factors such as sensor errors and turbulence interference, all the above methods could easily be trapped into local optimum, which may eventually result in repeated wandering of the robot. In addition, most of the above two types of source localization methods are only devised for plume tracking. Although several bio-inspired algorithms can be applied in both plume discovery and tracking [16,17], the final gas source confirmation process were rarely mentioned. The main reason is that most of the existing methods exhibited decent performance at the expense of practicality. In the absence of physical obstacles and neglect of external factors, high success rate and accurate result could obviate the need for gas source confirmation. But in the practical application, the windward side of the obstacle could form a highly concentrate pseudo source due to gas accumulation [31]. In this regard, gas source confirmation is indeed an indispensable part of source localization strategy. Moreover, most of the existing confirmation methods [31,32] only function as the final decider without proceeding subsequent treatment to the pseudo source: There is no guarantee that the pseudo source will not be treated as the real gas source in the following searching process.

As a response to the above analysis, the improvement and contribution presented by this paper mainly concentrate on the following aspects:

- (1) To propose a Bayesian-based strategy capable of completing practical gas source localization task within complex environment for ground-based mobile robots.
- (2) To avoid the searching trap by improving the reward function of the updating strategy.
- (3) To present a gas source confirmation method capable of further filtering the pseudo source while securing the localization of the real target.

Amid the source localization strategy, the plume discovery is realized on the basic of zigzag algorithm. The plume tracking process is implemented in light of the scheme presented in [30], i.e. to apply particle filtering method and information entropy algorithm to track the plume feathers. To avoid repeated searching in this stage, we improve the reward function on the basis of octree strategy so as to enhance efficiency. In the last stage, considering the existence of pseudo sources, we propose a target confirmation method as well as an artificial potential field (APF) algorithm to eliminate the influence of the pseudo sources. During the on-site experiment, the proposed scheme showed outstanding performance in terms of practicability, efficiency, success rate and accuracy, while the simulation showed that the pseudo source disposal method can effectively reduce the surrounding distraction.

The rest of this paper is arranged as follows: Section 2 defines the problem this paper solved and describes the components of the entire system and the functionality of each part; Section 3 provides detailed information (including the zigzag method, particle filtering method, information entropy algorithm and the design of the improved reward function) for implementing the plume discovery and tracking algorithm. Section 4 introduces the gas source confirmation algorithm as well as the theoretical model of the pseudo source processing algorithm. Experiment and simulation are both carried out in Section 5, where the performance of the source localization strategy is verified. In the last Section, conclusions and future focus are drawn.

2. Problem definition and system description

2.1. Problem definition

Define S as a gas source. Let $P_R(x, y) \in \mathbb{R}^2$ be the current position of the robot. Since the sensor is almost mounted in the center of the robot which means the orientation of the robot hardly has any influence on the gas concentration of current robot position, robot's orientation is not considered as an input. The robot is equipped with perception sensors, including concentration sensor and wind sensor. Let C_{cur} be the concentration of current position, v_s be the wind speed and φ_s be the wind direction. The main problem for this paper is to search S with C_{cur} , v_s and φ_s . Let C_{thres} be the lowest value that can be considered as detecting the gas. Consider the environment is unknown so that the initial position may not detect any gas, namely, concentration value is lower than C_{thres} . In addition, according to the analysis in Section 1, gas source confirmation is needed at the end of search process. Thus, the main problem can be divided into three different sub-problems. **Problem 1** is about searching plume at the beginning. **Problem 2** is about tracking plume to guide robot move to gas source. **Problem 3** is about confirming the authenticity of source found after **Problem 2**. All problems for this paper can be defined as the following.

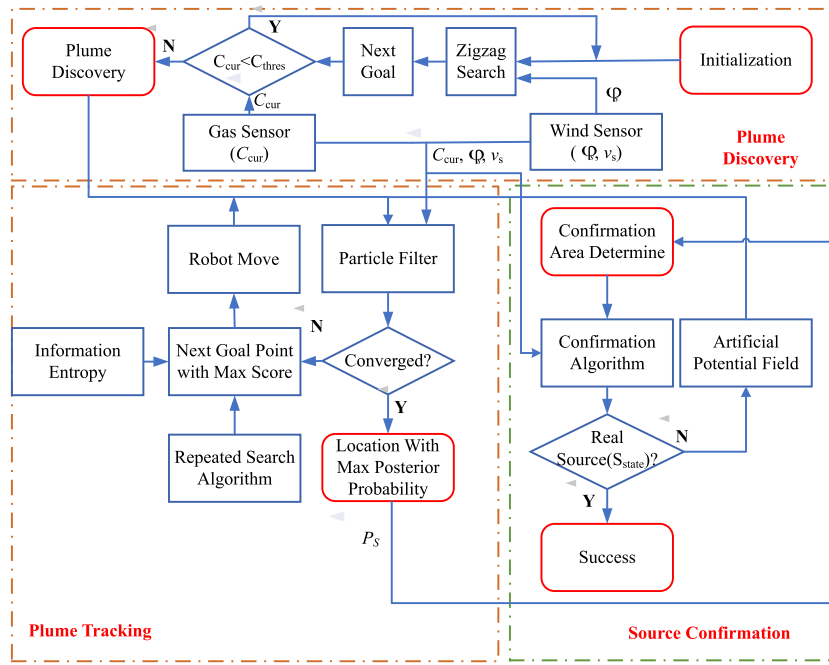


Fig. 1. Complete system of the source localization strategy.

Problem 1. When C_{cur} is lower than C_{thres} . Given P_{cur} , determine the goal position $P_{goal}(x, y)$ robot needs to reach in next step.

$$P_{goal} = f_1(P_{cur}, \phi_s) \quad C_{cur} < C_{thres} \quad (1)$$

Problem 2. When C_{cur} is larger than C_{thres} . Given P_{cur} , determine the goal position P_{goal} robot needs to reach in next step.

$$P_{goal} = f_2(C_{cur}, P_{cur}, v_s, \phi_s) \quad C_{cur} \geq C_{thres} \quad (2)$$

Problem 3. Given P_{cur} , determine whether the detected gas source $S_{detected}$, can be obtained after solving Problem 2, is true. Let $P_S(x, y)$ be the position of detected source and S_{state} be the final result. If S_{state} is true, then the detected source is a real gas source and P_S is the final location of gas source. Otherwise, it is a pseudo source and then back to Problem 2, restarting to track plume.

$$S_{state} = f_3(C_{cur}, P_{cur}, v_s, \phi_s, P_S) \quad (3)$$

Among all three problems, f_1 , f_2 and f_3 represent three algorithms used in three stages, not specific functions. The variables in brackets are inputs to the robot during three different stages.

Though these three problems are essential for the whole process, this paper will focus on the last two problems. For Problems 1 and 2, the main difference is whether or not consider C_{cur} and the problems in (1) and (2) will be both solved at each step. For all three problems, robot position P_{cur} is always known.

2.2. System description

To solve the above three problems, this paper proposes a set of source-finding strategy. As shown in Fig. 1, the overall system mainly comprises three parts: plume discovery, plume tracking and gas source confirmation. These three parts will solve the above three problems separately.

As the above flow chart suggests, the major role of the plume discovery part is to guide the robot to search towards the direction in which the odor propagate. In this paper, the biologically inspired zigzag search algorithm is used to locate the gas sources to a very first approximation. A more exact searching area centers the position of the robot after plume discovery is then defined

for further plume tracking. In this stage, the particle filtering method is applied as multi-dimensional particles are sampled (according to the initial parameters of sources), weighted (according to the gas concentration data) and resampled (according to the weight distribution). If the converge condition is fulfilled, the searching system will continue to proceed with gas source confirmation process. Otherwise, the robot will keep approaching the potential sources by moving towards the current goal position, which is obtained according to the improved reward function. The source confirmation stage is to judge the authenticity of the potential sources, namely the convergent position discovered in the last procedure. The confirmation algorithm of this part is implemented along the predefined boundary that circles a much precise region without obstacle hamper. If the source is real, the accurate location will be identified, suggesting the searching task is accomplished. Or else an APF method will be established around the pseudo-source to disperse the particles, keeping the particles from meaningless re-converging, and start to track the plume again.

3. Plume discovery and tracking

3.1. Plume discovery

As a famed method for plume discovery, the zigzag algorithm is implemented as follows: By reading the wind direction from the wind sensor, the robot moves windward in a zigzag with a certain angle α until any suspicious gas are discovered, after which a more precise square searching area is defined. The plume discovery procedure is captured in Fig. 2, where $X_W O_W Y_W$ is the world coordinate system Σ_W , $X_{RO} O_{RO} Y_{RO}$ denotes the robot coordinate system Σ_R at any certain moment when there is no plume has been detected, while $X_{Rn} O_{Rn} Y_{Rn}$ denotes the robot coordinate system Σ_{Rn} at the moment that potential plume has been discovered. The further $4\text{ m} \times 4\text{ m}$ searching area large enough to cover the exact location of the gas source (cyan region in Fig. 2) is then delimited by centering Σ_{Rn} coordinate. To better distinguish these two separate stages, we use Σ_S instead of Σ_{Rn} to represent the searching coordinate system in the following

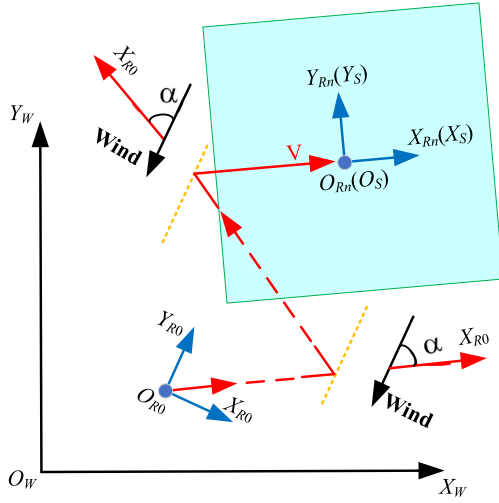


Fig. 2. Plume discovery.

plume tracking process (even though they are the same). The relationship between the Σ_S and Σ_W can be described by the following formula.

$${}^Wp = {}^S T^S p \quad (4)$$

where ${}^S p$ represents the coordinates of the point under Σ_S in searching area. ${}^S T$ is the homogeneous transformation from Σ_S to Σ_W . Wp represents the corresponding coordinates of ${}^S p$ under Σ_W . We take the position and orientation of the robot at the moment when plume is discovered as ${}^W T$. Since the plume tracking stage and confirmation stage will be conducted under Σ_S while the goal positions for navigation and the final position of the source are under Σ_W , Eq. (4) will be used only when a goal positions under Σ_S in plume tracking stage or the source location under Σ_S in source confirmation stage are obtained.

3.2. Plume tracking

3.2.1. The gas diffusion model

In this paper, the gas diffusion model is the same as the model proposed in paper [30], which can be described as follows:

$$C(p_k | \Gamma_s) = \frac{r_s}{4\pi D_s \|p_k - p_s\|} \exp\left(-\frac{\|p_k - p_s\|}{\lambda}\right) \exp\left(-\frac{\beta}{2D_s}\right) \quad (5)$$

$$\lambda = \sqrt{\frac{D_s \tau_s}{1 + (v_s^2 \tau_s / 4D_s)}} \quad (6)$$

$$\beta = (x_k - x_s)v_s \cos(\varphi_s) + (y_k - y_s)v_s \sin(\varphi_s) \quad (7)$$

where $\Gamma_s = [x_s, y_s, r_s, D_s, v_s, \varphi_s, \tau_s]^T$ represents the state vector of the gas source, p_k represents the position of the robot, i.e. the sensor coordinates, $C(p_k | \Gamma_s)$ represents the average gas concentration at p_k in the condition of Γ_s , λ and β are the intermediate variables that can be calculated by Eqs. (3) and (4), (x_s, y_s) represents the coordinates of the gas source, r_s represents the gas release rate, D_s represents the effective diffusion rate, v_s represents the speed of the wind, φ_s represents the current direction of the wind and τ_s represents the lifetime of the gas particles.

To test the practicability of the above diffusion model, the concentration distribution of a specific gas source generated using Eq. (2) is captured. As it is shown in Fig. 3a, the solid point sitting

at $(-1.2, -0.1)$ represents the location of the gas source. Fig. 3a reveals the gas distribution in the obstacle-free scenario, where, $r_s = 680$, $D_s = 0.15$, $\tau_s = 4.6$, $v_s = 2.0$, $\varphi_s = 180^\circ$. The gas distribution in the scene with obstacle can be simulated as Fig. 3b, in which the black rectangle nearby the origin represents the preset obstacle. Fig. 3 shows that the phenomena of gas accumulation on the windward surface and gas cavitation on the leeward surface are both factually portrayed, exhibiting an ideal performance of the adopted gas diffusion model.

3.2.2. Theory basis

The plume tracking strategy employed by this paper basically draws on the research method used in [30], while the reward function is herein revised for improving the efficiency of the plume tracking. To save space for presenting the rest of the research work, detailed principle and theoretical derivation of both the particle filtering and information entropy in plume tracking are arranged in Appendix A, based on which the final object function can be expressed as follows.

$$m_k^* = \arg \max_{m_k \in M_k} \{H(m_k)\} \quad (8)$$

$$H(m_k) = - \sum_{\hat{z}_{k+1}=0}^{\lfloor \hat{z} \rfloor_{\max}} p(\hat{z}_{k+1}(\hat{p}_{k+1}) | Z_{1:k}) \log p(\hat{z}_{k+1}(\hat{p}_{k+1}) | Z_{1:k}) \quad (9)$$

where m_k^* is the best direction of motion corresponding to the maximum entropy. $m_k \in M_k$ and $M_k = \{\uparrow, \downarrow, \leftarrow, \rightarrow\}$, whose elements indicate the action of moving forward, backward, towards the left and right from the current position. $H(m_k)$ is the entropy function, which can be expressed as Eq. (9). $Z_{1:k} = \{z_1(p_1), \dots, z_k(p_k)\}$ indicates the sensor measurement so far collected by moving from the track planning point p_1 to p_k , \hat{p}_{k+1} represents the next alternative planning point, $\hat{z}_{k+1}(\hat{p}_{k+1})$ indicates the expected sensor readout at the next position \hat{p}_{k+1} after executing m_k . $\lfloor \cdot \rfloor$ is to round down. Although the actual value of $\hat{z}_{k+1}(\hat{p}_{k+1})$ is unknown, it can still be approximated by the current posterior probability distribution obtained by particle filtering. Eq. (9) is the original reward function based on [30]. With Eq. (8), the next best search direction can be obtained.

3.2.3. Improved reward with repetitive rating

Due to the potentially adverse factors including strong turbulence, gas acceleration, cavitation around the obstacles and low concentration introduced by the environment, it would be almost impossible for the robot to move straightly towards the source origin in the process of source localization. Instead, the searching trace of the robot generally tend to be tortuous, in which case the same searching area could be accessed for multiple times. Since the system gains virtually no information for particle updates in this case, the robot may keep wandering, and even be trapped in a certain area, which can seriously damage the efficiency of the source localization. As it is shown in Fig. 4, the phenomenon of repetitive search has been recorded in part of the experiment reported by the previous research [30].

The main reason that causes the above dilemma is that the best target point corresponding to the maximum information entropy cannot be clearly distinguished when the scores of different directions calculated by Eq. (9) turn out to be almost the same. To solve this problem in the process of plume tracking, we adopt the repeated location search algorithm based on the octree method, which has been proposed in our previous study [33], to search for the repetitive points. The reward function is also improved based on the Repetition Rating algorithm presented here. The principle of this algorithm is as follows: Consider $m_k^{(j)} \in M_k (j \in \{1, 2, 3, 4\})$ to be one of the four alternative moving directions of the robot while $\hat{p}_{k+1}^{(j)}$ to be the corresponding planning point.

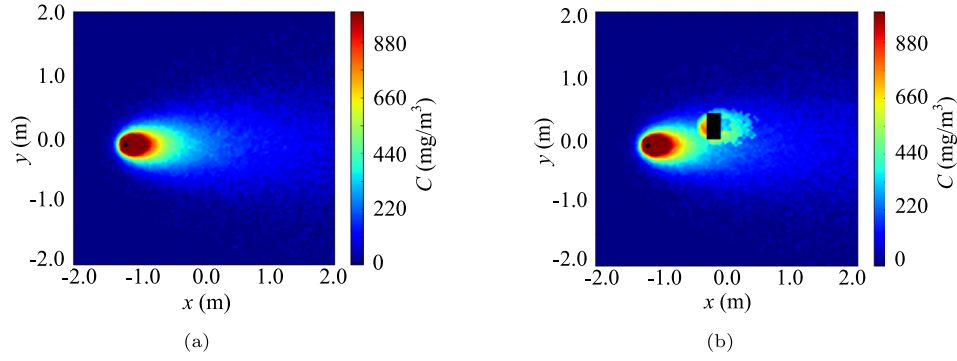


Fig. 3. Distribution of the gas concentration. (a): obstacle-free scene. (b): scene with an obstacle.

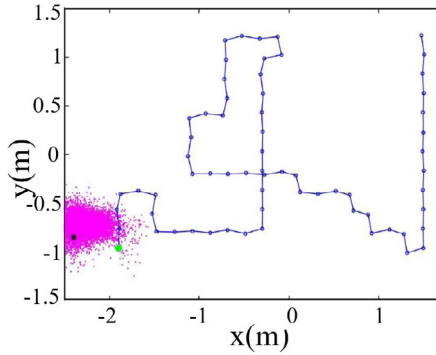


Fig. 4. Robot trajectory in the experiment reported in [30].

In order to evaluate the repetition rating of the four candidate target positions, we introduce a parameter $\rho \in S_\rho$ to be the range coefficient, where S_ρ represents a constant set of this coefficient. Let λ_m be the length of each motion step. By multiplying ρ with the moving step λ_m , we can determine a set of circular regions centering the alternative target point as shown in Fig. 5 (where S_ρ is set to be $\{0.4, 2.0, 4.0\}$). Through these particular regions, we can assess the repetition rating of the candidate position. The more previous trajectory points are included by these regions, the higher repetition rating this alternative target is. To obtain a fair rating result, the trajectory points that have been included by the inner region will not be again counted when considering the outer region. As such, the improved reward function is designed as follows:

$$R(m_k^{(j)}) = \alpha \frac{H(m_k^{(j)})}{\sum_{j=1}^4 H(m_k^{(j)})} - \beta \frac{\sum_{\rho_r \in S_\rho} (\rho_r \lambda_m)^{-1} C_{\rho_r \lambda_m}(\hat{p}_{k+1}^{(j)})}{\sum_{j=1}^4 \sum_{\rho_r \in S_\rho} (\rho_r \lambda_m)^{-1} C_{\rho_r \lambda_m}(\hat{p}_{k+1}^{(j)})} \quad (10)$$

It can be seen from Eq. (10) that the rating is essentially evaluated by the information entropy and the number of the previous planning points being including in the predefined region, while the weights of these two items are represented by $\alpha > 0$ and $\beta > 0$, respectively. $(\rho_r \lambda_m)^{-1}$ represents the score of each trajectory point that falls into the $\rho_r \lambda_m$ region (In Fig. 5, $r \in \{1, 2, 3\}$), while $C_{\rho_r \lambda_m}(\hat{p}_{k+1}^{(j)})$ represents the number of these points. $\sum (\rho_r \lambda_m)^{-1} C_{\rho_r \lambda_m}(\hat{p}_{k+1}^{(j)})$ can therefore represents the total score of the repetitive trajectory points around $\hat{p}_{k+1}^{(j)}$.

The improved reward function possesses several characteristics that benefits the updating process to a certain extent:

(1) Assuming the number of the points being included by the $\rho_r \lambda_m$ region is fixed, the smaller the radius of the current

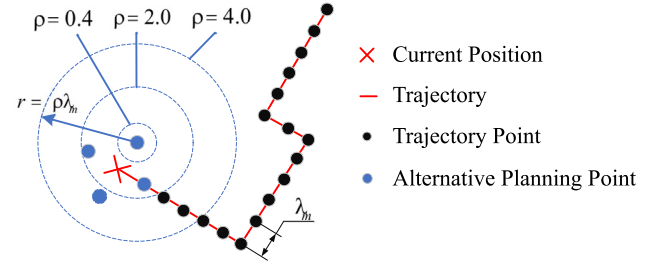


Fig. 5. Schematic of the repetition rating algorithm.

defined region is, the higher repetition rating gets, with lower total reward gains.

(2) Assuming the boundary of one specific $\rho_r \lambda_m$ region is fixed, the more previous trajectory points are included, the higher repetition rating gets, the lower total reward gains.

(3) The power item $(\rho_r \lambda_m)^{-1}$ is adopted to estimate the repetition score of the trajectory points being included by the $\rho_r \lambda_m$ region, which allows the repetition rating to rise at a power speed as the ρ_r decreases, showing an increasingly stronger rejection to the nearby repetitive point.

At last, by replacing the reward function with Eq. (10), the best motion direction corresponding to the highest reward can be derived as follows:

$$m_k^* = \arg \max_{m_k \in M_k} \{R(m_k)\} \quad (11)$$

4. Source confirmation and pseudo-source processing

Although previous steps have included a range of factors in the process of searching the gas source, the particles could still converge to a false point due to the influence pose by the environment, such as the existence of the pseudo-source that formed by the gas accumulation on the windward surface of the obstacle, unreasonable initial settings of the source state parameters, or the discontinuity of the gas propagation. Therefore, it is necessary to design a gas source confirmation algorithm to distinguish whether the discovered source is real in the converged position of the particle. Also consider the distractions that may be introduced by the pseudo sources, the gas source confirmation method proposed here are supposed to possesses the following two functions:

(1) The proposed method should be able to evaluate the authenticity of the candidate gas source and decide whether it is real.

(2) The proposed method should be able to eliminate the effect of the pseudo sources that were once confirmed.

4.1. Source confirmation

The necessary conditions for determining whether there is a gas source around the suspected position are as follows: Firstly, the concentration of the gas around should be high enough; Secondly, the gas plume should always stem from the direction of the candidate location. With a predefined confirmation region, the above logic can be implemented as follows:

$$O_S \rightarrow (N_S/N \geq \eta) \wedge (n_s/N_S \geq \xi) \quad (12)$$

$$\overline{O_S} \rightarrow (N_S/N < \eta) \vee (n_s/N_S < \xi) \quad (13)$$

where O_S represents the action of authenticating the suspected gas source while $\overline{O_S}$ is just the opposite. The symbol \rightarrow represents a logical equivalence. N_S represents the number of times the gas concentration exceeds the preset threshold during the confirmation process, N represents the total steps of the confirmation stage. n_s represents the number of times the particle convergence position falls in the confirmation region when the gas concentration exceeds the threshold, η and ξ represents the corresponding experienced thresholds.

η and ξ are both experienced thresholds. With η , we can only identify pseudo sources whose concentration is not high enough to be a true gas source. Consider the true source with strong wind from one direction, leading to a very small area with effective measurements when detecting around the source, η will not be set too large. ξ is the threshold that can identify most pseudo sources. For true source, when detecting around the source, most convergence location of effective measurement will fall in the confirmation region, while for pseudo source, most convergence location of effective measurements will not because the distribution around pseudo source is not obey the model we use for a true source, namely Eq. (5). Then we can use the value of n_s/N_S to identify true source and pseudo source.

With the plume tracking algorithm presented in Section 3, the suspected gas source location can be obtained, which is essentially the coordinates corresponding to the global maximum point of the posterior probability density (PPD). And by establishing a coordinate system with the same orientation of Σ_S , the confirmation area centering the convergence position can be initially defined as follows:

$$S = \{(x, y) | x \in [X_{conf}^{min}, X_{conf}^{max}], y \in [Y_{conf}^{min}, Y_{conf}^{max}]\} \quad (14)$$

$$L = \max\{\sigma_x, \sigma_y\} \quad (15)$$

where S represents the initially defined square area and L represents its initial length, σ_x and σ_y represents the standard deviations of the x and y coordinates of the particles, respectively. The initial confirmation area defined above is able to contain most of the particles. The schematic of constructing the final confirmation area is shown in Fig. 6: Based on the initial area, the robot will do collision check along the red boundary. The dotted line is the outline of the robot and the colored area is obstacle. Four test points are selected on the outline. When any test point is inside the obstacle area, collision is supposed to be detected. Then the red boundary will be expanded outward until it can enclose the obstacle. In Fig. 6, step1 detects collision and then step2 and step3 expand the red boundary to enclose the obstacle. In step4, collision is detected again between test points and the obstacle. Thus, step5 expands the red boundary again until no collision along the red boundary is detected. The final confirmation area is like the rectangle area in step6.

With the area completely defined, the robot will be able to implement the major confirmation procedures. In order to realize the gas source identification, the robot keeps moving along the boundary of the confirmation area for 3 cycles, judging the

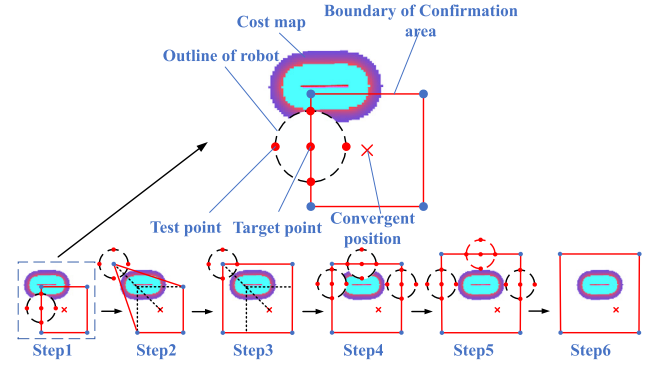


Fig. 6. Confirmation area initiating and the collision-free boundary determining. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

authenticity of the suspected source based on the logic process Eqs. (12) and (13). To ensure that the particle states is mutually independent in each of the searching cycle, the particles need to be uniformly resampled at the beginning of each round, while the particle filtering strategy presented in Section 3 is continuously performed. Fig. 7 describes the conformation process of the robot in a single circle, in which 7a shows the particle convergence state after the plume tracking process; 7b shows the first round uniform sampling within an enlarged square region (with the length of $6L$) after the confirmation area is determined; 7c, 7d and 7e together record the converge process of the particles and the trace of the robot by moving along the boundary of the confirmation area, 7f captures the state of the particles at the end of this cycle. Since the convergence showed by 7f is inevitable in each of the searching circle, it is necessary to resample before starting a new confirmation cycle, otherwise the already converged population can hardly provide source information for the next confirmation round.

Fig. 7 is used to show the process of confirmation stage. Since the particles converge to the location of real source directly at the end of plume tracking stage, the influence of pseudo source does not show up. In Section 5.2, we give another simulation results where particles converge to a pseudo source in Fig. 20 that can demonstrate the effect of pseudo source clearly.

4.2. Pseudo source processing

The sources that discovered through the plume tracking process yet do not pass the confirmation process will be identified as the pseudo sources, which could form extra distraction to further source localization. To keep the particles from converging to the pseudo sources again, a processing algorithm based on artificial potential field (APF) method is proposed for eliminating the undesirable distractions.

First introduced by Oussama Khatib in 1990 [34], the APF method has already become a popular strategy in terms of motion planning and obstacle avoidance. Here to keep the particles away from the center of the pseudo source, we introduce a repulsive APF, which can be described as follows:

$$U_{req}(q) = \begin{cases} \frac{1}{2}(\frac{1}{\rho} - \frac{1}{\rho_0})^2, & \text{if } \rho \leq \rho_0 \\ 0, & \text{if } \rho > \rho_0 \end{cases} \quad (16)$$

where q represents the object, ρ represents the shortest distance between the object and the obstacle center, ρ_0 represents the effective distance of the obstacle repulsion, $U_{req}(q)$ represents the value of the repulsion field. When ρ is greater than ρ_0 , the object will not be repelled, in which case the corresponding potential energy is equal to 0.

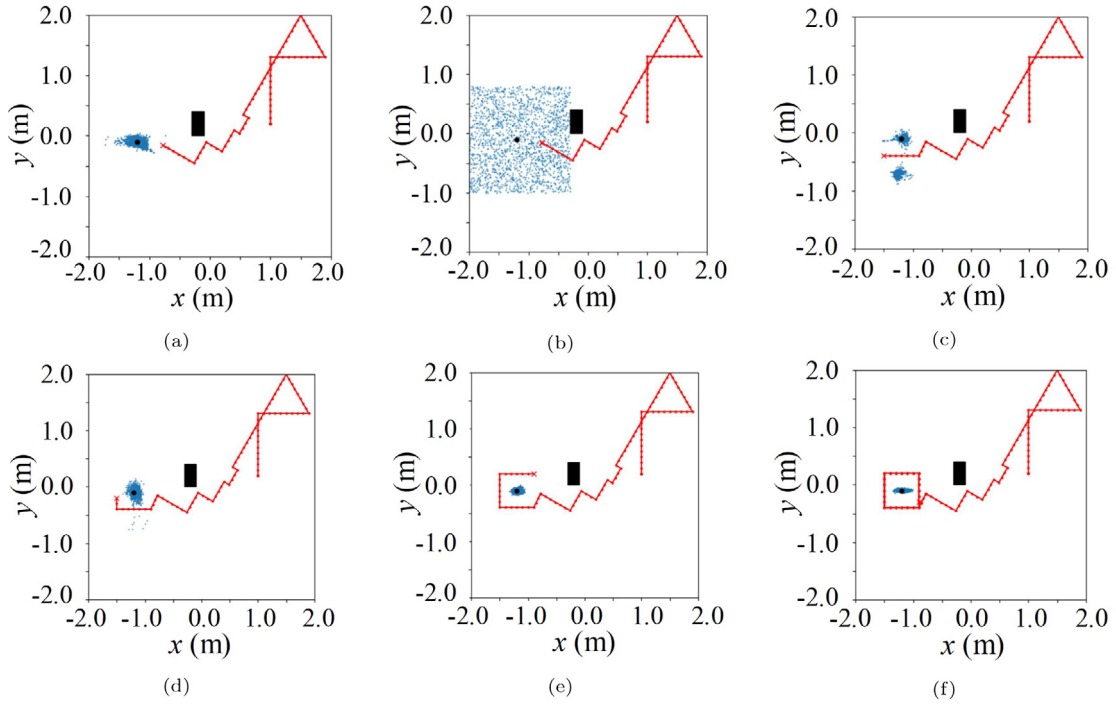


Fig. 7. Simulation of the source confirmation (a): the convergence state in plume tracking stage. (b): uniform sampling in conformation stage. (c)–(e): the converge process of the particles. (f): state of the particles at the end of this cycle.

By solving the gradient of the repulsive field, we can obtain the corresponding repulsive force as follows.

$$F_{req}(q) = -\nabla U_{req}(q) = \begin{cases} (\frac{1}{\rho} - \frac{1}{\rho_0}) \frac{1}{\rho^2} \nabla \rho, & \text{if } \rho \leq \rho_0 \\ 0, & \text{if } \rho > \rho_0 \end{cases} \quad (17)$$

Within the framework of the above APF, if every particle is regarded as an object q , while the pseudo source in the confirmation center is regarded as the obstacle center, the repulsive force will keep increasing as the particles approaching the pseudo source. Then all particles will potentially keep away from the pseudo source. The distribution of the resultant forces and the corresponding x components is shown in Fig. 8a, where the black spot in the center represents the pseudo source.

It is obvious that the resultant force is inversely proportional to the distance between the object and the center. In this paper, the repulsive APF is applied only in the stages of both the RPF and M-H sampling to keep the particles from re-converging to the pseudo source, there is no need to employ the APF in the process of uniformly resampling as it is merely a step of randomly reproduction.

Since the parameters x and y of the particle state vector are mutually independent when sampling through RPF, the x_s and y_s can be separately sampled from the Gaussian core without affecting each other, which can essentially be regarded as two parallel 1D movement. Thus, we employ two 1D repulsive field shown in Fig. 8 to get stronger repulsive forces in both the x and y direction. In 1D repulsion field, the repulsion force is inversely proportional to the distance along one direction. Take a specific particle A as shown in Fig. 8a as the example: when using 2D repulsion field, according to the force applying mechanism of the APF, the corresponding repulsion is inversely proportional to d_A . The repulsion along the horizontal direction (i.e. F_{req}^{Ax}) turns out to be pretty poor since the x component of A is very small. However, when using 1D horizontal repulsion field for x , the repulsion force of point A is inversely proportional to the distance d_x , which is much larger than that in 2D field. It is the same with 1D vertical field for y . Thus, with 1D repulsion field, we can make

sure that the particles all keep away from the center both along the horizontal direction and the vertical direction. Generally, the mechanism shown in Fig. 8b is preferable to that shown in Fig. 8a.

For the superposed model that consists of two 1D repulsive APF, the distances in the x and y direction can be defined as follows, respectively:

$$\rho^x = \rho^x(q, q_{obs}) = |q^x - q_{obs}^x| \quad (18)$$

$$\rho^y = \rho^y(q, q_{obs}) = |q^y - q_{obs}^y| \quad (19)$$

By solving the gradient, we can obtain the corresponding repulsive forces as follows:

$$F_{req}^x(q) = -\nabla U_{req}^x(q) = \begin{cases} (\frac{1}{\rho^x} - \frac{1}{\rho_0^x}) \frac{1}{(\rho^x)^2} \nabla \rho^x, & \text{if } \rho^x \leq \rho_0^x \\ 0, & \text{if } \rho^x > \rho_0^x \end{cases} \quad (20)$$

$$F_{req}^y(q) = -\nabla U_{req}^y(q) = \begin{cases} (\frac{1}{\rho^y} - \frac{1}{\rho_0^y}) \frac{1}{(\rho^y)^2} \nabla \rho^y, & \text{if } \rho^y \leq \rho_0^y \\ 0, & \text{if } \rho^y > \rho_0^y \end{cases} \quad (21)$$

where $\rho_0^x = \rho_0^y = 0.5L$, $F_{req}^x(q)$ represents the one-dimensional repulsive force along the x direction, $F_{req}^y(q)$ represents the one-dimensional repulsive force along the y direction.

The red dotted line shown Fig. 9 provides a clearer view of the repulsive force distribution of a one-dimensional repulsive APF along the x axis, within which the obstacle is set at the origin, L is set to be 0.42 (therefore $\rho_0 = 0.21$), the maximum repulsive force is set as 20N. As it is shown, the repulsive force increases sharply when the distance is lower than 0.21.

In order to improve the accuracy of the final result of the source localization process, we also consider the situation in which the real gas source is mistakenly identified as a pseudo source during the source confirmation process. For this concern, the possibility of letting the particles to converge again should be properly preserved, which can be realized by reducing the impact range of the repulsive force. To also retain the repulsion of the APF, a correction coefficient is introduced as follows:

$$K = \frac{8(\rho_0)^3}{9} \quad (22)$$

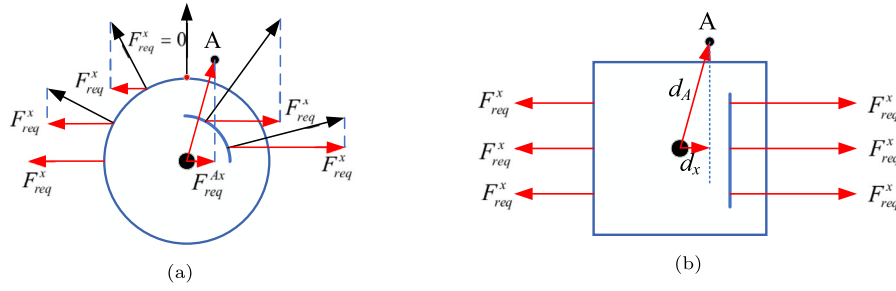


Fig. 8. The two different repulsion field models. (a): distribution of the repulsive forces within the 2D repulsion field. (b): distribution of the repulsive forces within the 1D horizontal repulsion field.

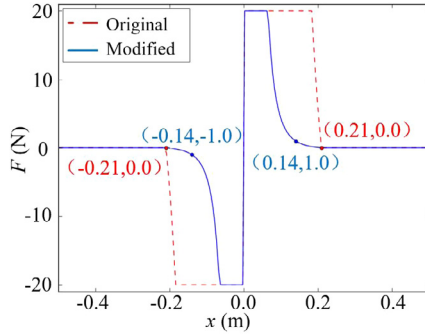


Fig. 9. The repulsive force of a 1D repulsive APF along the x direction.

Therefore, the repulsive forces along the x and y direction can be modified separately as follows:

$$F_{req}^x(q) = KF_{req}^x(q) \quad (23)$$

$$F_{req}^y(q) = KF_{req}^y(q) \quad (24)$$

As the blue curve in Fig. 9 shows, the corrected repulsion curve always passes through the point $(2/3\rho_0, 1)$, which properly limits the impact range of the APF while still ensuring that most of the converged particles are subjected to more than 1N repulsion. Similar to the accept probability α in M-H sampling, the attractive force of particle q can be defined as:

$$F_{att}(q) = \frac{p(z_k | \Gamma_{s,k}^{(i)})}{p(z_k | \Gamma_{s,k}^{(i)})} \quad (25)$$

Hence the combined force of the particles in the x and y direction can be expressed as follows:

$$F^x(q) = \xi F_{att}^x(q) + \eta F_{req}^{x*}(q) \quad (26)$$

$$F^y(q) = \xi F_{att}^y(q) + \eta F_{req}^{y*}(q) \quad (27)$$

where $F^x(q)$ represents the resultant forces of particle q in the x direction, $F^y(q)$ represents the resultant forces in the y direction, ξ represents the weight of the attractive force, η represents the weight of the repulsive force.

According to the above definition, the attractive force shares the same direction with the particle movement determined by the RPF, while the force magnitude depends on the ratio of the likelihood probabilities before and after the particle movement. The pseudo source processing algorithm based on repulsive APF is shown as Algorithm 1. Take the movement along the x direction as an example, the processing are as follows.

Algorithm 1: Pseudo source processing algorithm based on APF

```

for  $i$  from 1 to  $N$  do
    Compute  $F^x(q)$  of the  $i$ th particle;
    Sample  $u \sim U[0, 1]$ ;
    if  $|F^x(q)| < u$  then
        Refuse movement and  $\overline{\Gamma_{s,k}^{(i)}}(x) = \Gamma_{s,k}^{(i)}(x)$ ;
    else
        Compute  $\gamma = h_{opt} A_k e^i / F^x(q)$ ;
        if  $\gamma \geq 0$  then
            Move forward and  $\overline{\Gamma_{s,k}^{(i)}}(x) = \Gamma_{s,k}^{(i)}(x) + h_{opt} A_k e^i$ 
        else
            Move backward and  $\overline{\Gamma_{s,k}^{(i)}}(x) = \Gamma_{s,k}^{(i)}(x) - h_{opt} A_k e^i$ 
        end
    end
end

```

(1) If the absolute value of $F^x(q)$ is less than u ($u \sim U[0, 1]$), the movement of the particle i will be refused, in which case the particle state is kept as $\overline{\Gamma_{s,k}^{(i)}}(x) = \Gamma_{s,k}^{(i)}(x)$;

(2) If the absolute value of $F^x(q)$ is equal or greater than u , then when $F^x(q)$ shares the same direction with the particle movement determined by the RPF, the particle will keep moving forward, that is $\overline{\Gamma_{s,k}^{(i)}}(x) = \Gamma_{s,k}^{(i)}(x) + h_{opt} A_k e^i$;

(3) If the absolute value of $F^x(q)$ is equal or greater than u , then when $F^x(q)$ is in the opposite direction to the particle movement determined by the RPF, the particle will move reversely, that is $\overline{\Gamma_{s,k}^{(i)}}(x) = \Gamma_{s,k}^{(i)}(x) - h_{opt} A_k e^i$.

5. Simulations and experiments

In this section, by comparing with the experiment results reported in [30], feasibility of the proposed source localization algorithm is verified. In addition, validity of the pseudo source processing algorithm is demonstrated through the simulation. Model of the gas source distribution, particle filter and reward function were all implemented within the python IDE, while the octree-based repetitive position search method was implemented within the C++ IDE.

As it is shown in Fig. 10, we adopt sticks of incense to be the source of gas and the smoke it produced was to serve as the plume. The wind was provided by a preset electric fan so that the odor could spread over the scene. An obstacle is placed nearby the source to test whether the proposed algorithm is able to locate the source when it was partially blocked. The mobile robot used to perform the task is shown in Fig. 11, while the

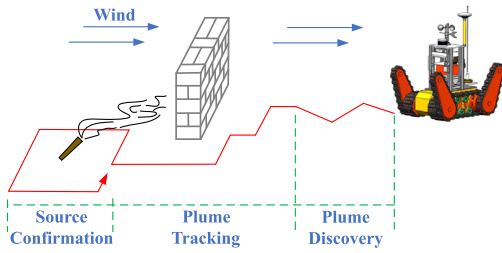


Fig. 10. Distribution of the gas concentration.

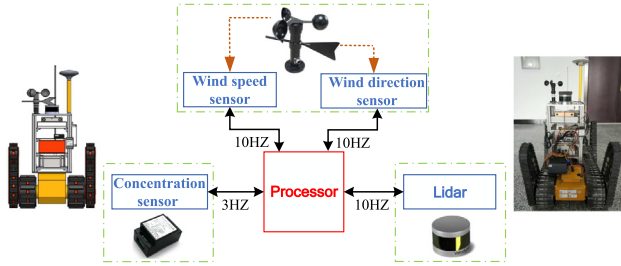


Fig. 11. Distribution of the gas concentration.

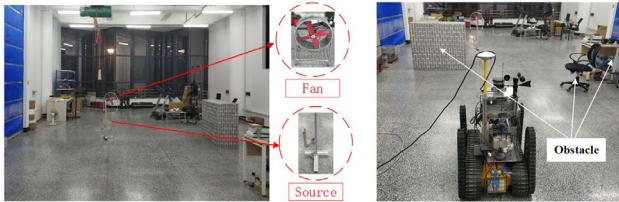


Fig. 12. Distribution of the gas concentration.

Table 1

The employed sensors and corresponding performances.

Sensor	Characteristics
Velodyne-16 Lidar	10 Hz, navigation and obstacle avoiding
Laser dust transmitter	3 Hz, concentration detecting
Wind speed sensor	10 Hz, wind speed detecting
Wind direction sensor	10 Hz, wind direction detecting

real experiment environment is shown in Fig. 12. Four kinds of sensors with performances listed in Table 1 were mounted on the robot, among which the laser dust transmitter (LDT) was the main sensor to collect the plume information. The employed LDT is able to detect the concentration of the PM1.0 to characterize the plume emitted by the incense, while the effective detection range of the concentration was set from 10 mg/m³ to 100 mg/m³.

5.1. Source localization experiment

By initially setting the robot's coordinate system coincides with the map coordinate system (which is also Σ_W in the experiment), the initial position of the robot was set to be (0, 0, 0°). Threshold value of the concentration during plume discovery was set to be 20 mg/m³ while the angle of zigzag search was set to be 45 degrees. Within the experiment, source of the plume was preliminarily discovered by the robot at coordinate (2.26, 0.57, 45°) after 8 zigzag steps. Origin of the searching coordinate system Σ_S was therefore set to be (2.26, 0.57) while orientation was set to be 45 degrees from Σ_W . ${}^W T_S$ in Eq. (4) will be obtained by setting (2.26, 0.57) as translation and 45° as rotation. The side length of the square search area was set to be 8 m. Other

Table 2

Initial setting of the experiments.

Parameter	Value or distribution
r_s	$U(350, 700)$
D_s	$U(0.4, 0.8)$
v_s	$U(0.5, 2.5)$
φ_s	$U(80, 100)$
τ_s	$U(3.5, 5.0)$
x_s	$U(-4, 4)$
y_s	$U(-4, 4)$
Step length for plume discovery	0.4 m
Step length for plume tracking	0.2 m

parameters were initialized as shown in Table 2, where the wind speed and wind direction were measured in advance by the wind speed sensor and wind direction sensor, respectively. Since r_s , D_s and τ_s corresponding to the gas source (the incense) was unknown, in order to ensure the accuracy of the gas diffusion model, the initial intervals of these three parameters were estimated by using the variable residue parameter determination algorithm based on the Monte Carlo method. Detailed process can be found in Appendix B.

Fig. 13 shows the real time changes of the particle state during the plume tracking process, where the blue dots represent the particle distribution, the red line indicates the robot trajectory, the fork at the end of the trajectory indicates the current position of the robot. Population of the particle was set to be 4000. As shown in Fig. 13a, the particles were firstly sampled within the searching area according to the standard two-dimensional uniform distribution, and was considered to be converged if both the standard deviations of the particles along the x and y direction are smaller than 0.23, in which case other parameters were also considered to be converged. Fig. 13b shows the evolution process of the particles at step 14 while Fig. 13c shows the converged state of the particles at step 24, where $\sigma_x = 0.20$ and $\sigma_y = 0.21$.

The PPD curve of gas source location was estimated based on the convergence results. As shown in Fig. 14, the coordinates corresponding to the peak of the curves in Figs. 14a and 14b indicates the most likely location of the gas source, which turned out to be (1.95, -2.80) under Σ_S . It can be transformed to (5.62, -0.04) under Σ_W by using Eq. (4). The blue vertical solid line represents the mean of this curve, that is 2.00 m along the x direction and -2.94 m along the y, while the corresponding red dotted lines on both sides represent the standard deviation.

According to Eqs. (14) and (15) presented in Section 4.2, four corner points of the rectangular confirmation area can be obtained under Σ_S . Then with Eq. (4), the corresponding coordinates of these points under Σ_W which was (6.79, 1.33), (4.79, 1.33), (4.79, -0.87) and (6.79, -0.87) are obtained as well. The source confirmation process was carried out under the coordinate system of the confirmation area. The robot needed to move 3 cycles along the confirmation boundary while the number of steps within a single circle was set to be 20, which is a total of 60 steps. The threshold value of the confirmation concentration was set to be 30 mg/m³, which means as long as the measuring result is bigger than 30, N_s in Eqs. (12) and (13) will add one. Two threshold values η and ξ in Eqs. (12) and (13) were set to be 0.3 and 0.5 respectively. During the confirmation process, the number of times that the gas concentration exceeded 30 was 24 (i.e. $N_s = 24$), among which the number of times that the particle convergence position fell into the confirmation region was 15 (i.e. $n_s = 15$). By applying Eqs. (12) and (13), we got $N_s/N = 24/60 > \eta$ and $n_s/N_s = 15/24 > \xi$. Therefore, the gas source is successfully confirmed.

In the experiment, the exact coordinates of the gas source under Σ_W were (5.80, 0.20), while the gas source coordinates

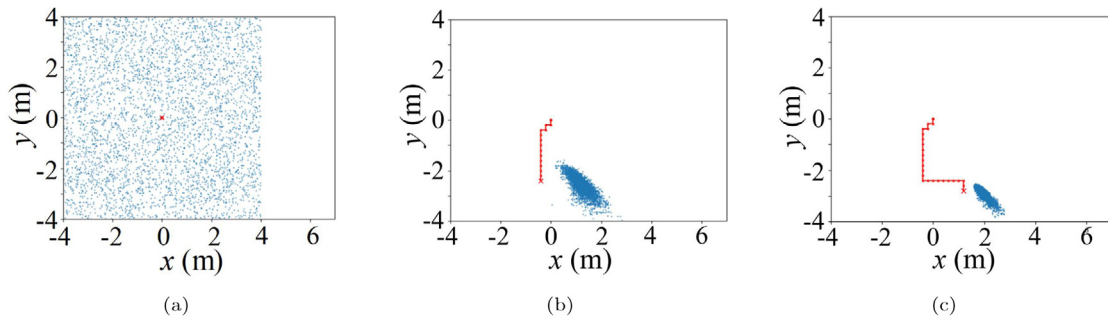


Fig. 13. Particle changes during plume tracking. (a): uniform sampling at the beginning of plume tracking stage. (b): state of particles at step 14. (c): converged state of particles at step 24.

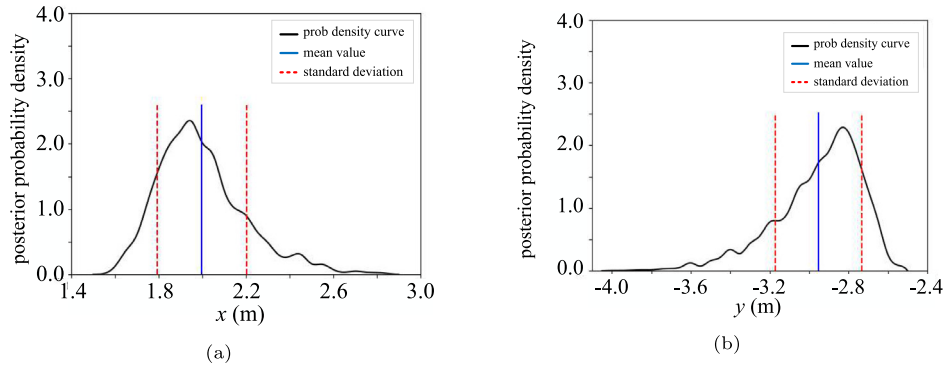


Fig. 14. PPD distribution of the gas source location. (a): PPD distribution in the x direction. (b): PPD distribution in the y direction.

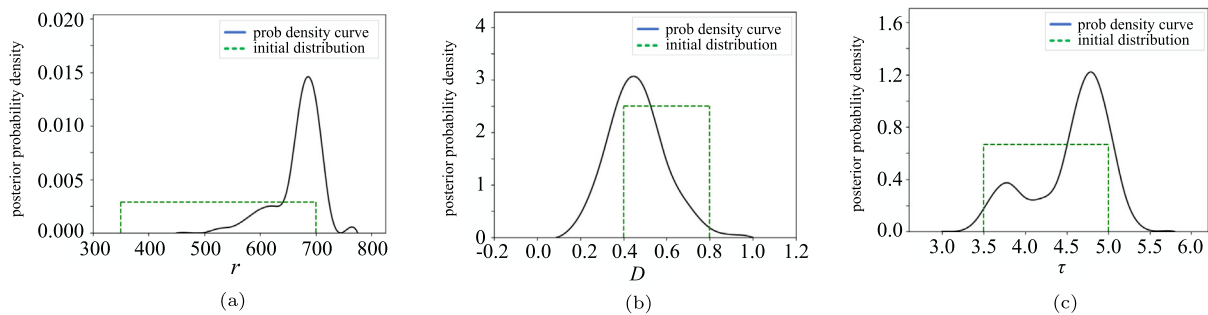


Fig. 15. PPD of gas source parameter. (a): PPD distribution of r . (b): PPD distribution of D_s . (c): PPD distribution of τ_s .

estimated by particle filtering were (5.62, -0.04). Deviations in the x and y direction were 0.18 m and 0.24 m respectively, while the Euclid distance error was 0.3 m. Meanwhile, the PPD of the gas release rate r_s , gas diffusion rate D_s and the lifetime of gas particles τ_s were also estimated. The black curves in Fig. 15 show the corresponding PPD distributions, while the green dashed line represents the priori information shown in Table 2.

To better review the processes of this experiment, the planned search trajectory of the robot is shown in Fig. 16, and the real trajectory is shown in Fig. 17. The first 8 steps indicate the plume discovery process, the middle 24 steps indicate the plume tracking process, while the last rectangular area indicates the source confirmation process. Fig. 17c shows the point-cloud map corresponding to the environment in Fig. 12. In addition, the corresponding obstacles are also labeled in the map. The trajectories in different stages combined with the environment map are also demonstrated in Fig. 17. These trajectories in the map are consistent with those in Fig. 16.

Of all the experiments that have been implemented, we select three sets of results to compare with the results reported in [30].

Table 3

Performance comparison of the obstacle-free source localization experiments.

Indexes	Results reported in [30]			Results in this paper		
	1	2	3	1	2	3
x_s error (m)	0.01	0.02	0.13	0.18	0.36	0.36
y_s error (m)	0.03	0.03	0.02	0.24	0.27	0.13
Euclid error (m)	0.03	0.04	0.14	0.3	0.45	0.38
Steps for plume tracking	49	47	44	24	28	31

As it is shown in Table 3, case 1 corresponds to the experiment described in this subsection. Since the type of the incense and the parameters were not given in [30], we choose the results of the experiment that used 6 sticks of incense (the highest case) as the gas source in [30]. In our experiments, two sticks of incense were used. Consider that there is only plume tracking stage in [30] and one of the contributions of this paper is to improve the efficiency of algorithm in [30], thus only steps in plume tracking stage are counted in Table 3.

According to the step number shown in Table 3, our strategy is more efficient in plume tracking stage than the method proposed

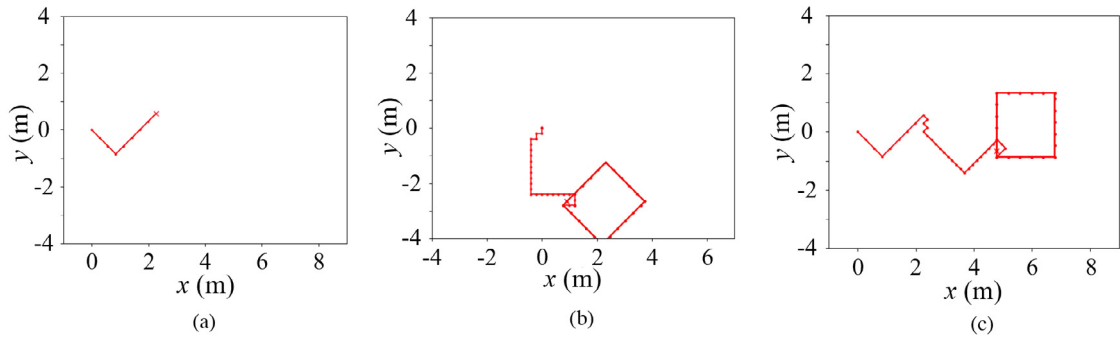


Fig. 16. The planning trajectory. (a): steps for plume discovery in WCS. (b): steps for plume tracking and source confirmation in $X_5O_5Y_5$. (c): whole steps of source localization.

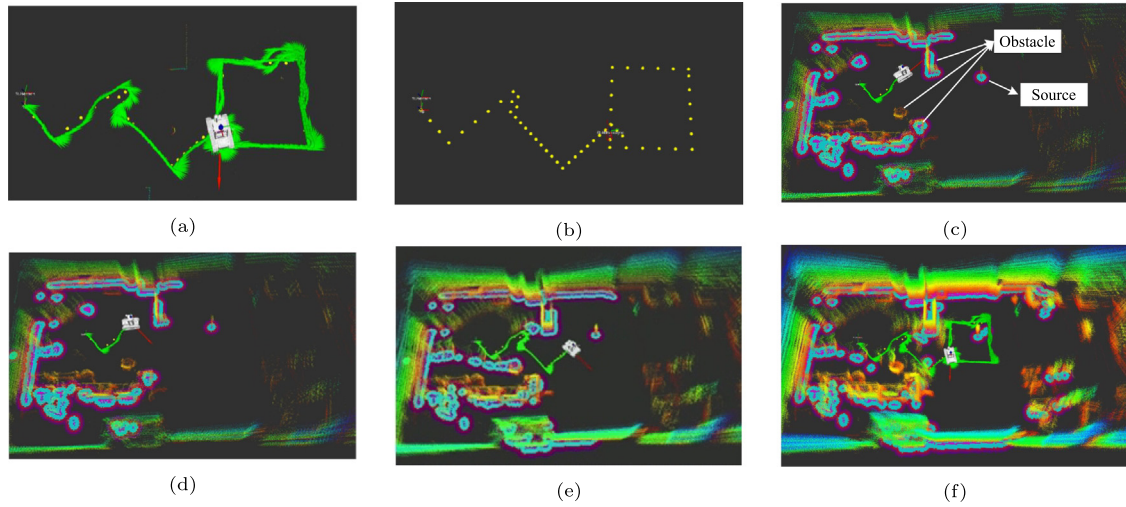


Fig. 17. The actual trajectory. (a): the actual trajectory of the robot. (b): planned points of robot trajectory. (c): environment showed in Rviz. (d): plume discovery in Rviz. (e): plume tracking in Rviz. (f): entire trajectory with environment.

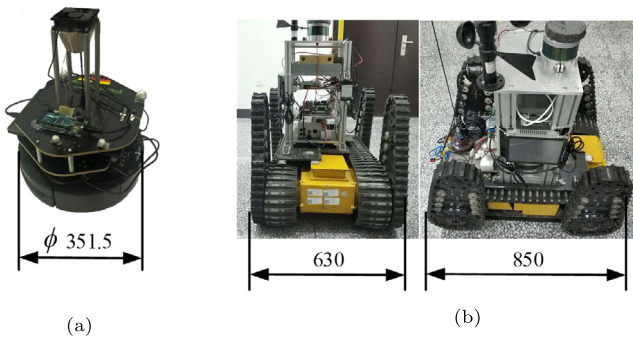


Fig. 18. Experimental Robotics Platform. (a): turtlebot2. (b): robot used in this paper.

in [30]. Nevertheless, the strategy presented here are of lower accuracy when considering the error indexes. The main reason is that the volume of the robot used in [30], i.e. the Turtlebot2, is much smaller than that of our robot (As shown in Fig. 18). During the navigation, the robot with larger volume will find it more difficult to get closer to the gas source, which results in a much smaller number of iterations in the process of plume tracking. The accuracy is therefore affected. Although the accuracy of the proposed source localization method is worse than that in [30], it had basically met the requirements for practical source localization.

To get rid of the effect that stem from the robot volume, we have also carried out the corresponding simulation. Environment

around the unblocked gas source was established on the basis of the gas diffusion Eq. (5), while the gas source position is set to be $(-1.2, -0.1)$. Other settings were the same as it is shown in Fig. 3. As such, the concentration at any point in the gas field could be calculated. Fluctuation of the plume were simulated by introducing the Gaussian noise. The robot's trajectory during the plume tracking process is shown as Fig. 19a. Location of the source $(-1.25, -0.11)$ was obtained according to the PPD distribution shown in Figs. 19b and 19c. The Euclidean distance error turned out to be 0.05 m, which achieved a similar accuracy reported in [30].

5.2. Analysis of pseudo-source processing

As it has been shown in Fig. 3, the pseudo sources generally form on the windward surface of the obstacles as the gas concentration on the windward surface always turn out to be higher than the surrounding concentration. In this regard, two requirements should be meet if a pseudo source were to form and confirmed. Firstly, a large amount of gas should accumulate on the windward surface of the obstacle to form a pseudo source; Secondly, the particles should converge at the position of the pseudo source. With the natural accumulation of the gas, the first requirement is relatively easier to be satisfied. While the second requirement is rather difficult to be met due to the existence of the real gas source, especially when the obstacle appear to be close enough to the gas source. Thus, it would be uneconomic to carry out the experiment of pseudo source

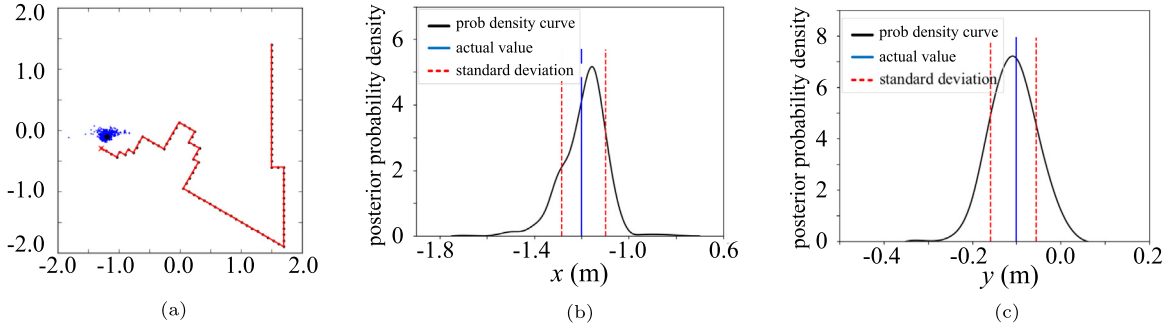


Fig. 19. Results of the plume tracking simulation. (a): searching trajectory within the simulation. (b): PPD distribution in x direction. (c): PPD distribution of source location in y direction.

confirmation and processing. Instead, we decided to verify the validity of the pseudo source processing algorithm through the simulations. Though it still took a lot of time to create a pseudo source scene, the performance of the proposed pseudo source processing method still deserves our attention. Fig. 20 captures a complete simulation process with the appearance of a pseudo source, within which the physical environment was the same as that in Fig. 3.

Fig. 20a indicates the state when the particles converge to the pseudo source at the end of the plume tracking process, Fig. 20b shows the sample of particles based on uniform distribution before the source confirmation process actually start. Fig. 20c shows the particle state at the end of the source confirmation: Obviously the particles did not fall into the confirmation area. The thresholds η and ξ in Eq. (12) were still set to be 0.3 and 0.5. The threshold value of concentration will be set to 30 mg/m³. By applying Eqs. (12) and (13) in the state of Fig. 20c, we got $N_s/N = 55/60 > \eta$ while $n_s/N_s = 22/55 < \xi$. The pseudo source was as such identified. To further process the pseudo source, a repulsive APF was established in the confirmation area. Figs. 20d to 20i show the subsequent process to again track the plume by considering the pseudo source. Fig. 20d indicates that the particles were resampled from a wider uniform distribution to ensure the particle diversity. Fig. 20e shows that the particles were all kept away from the center of the APF. Figs. 20f and 20g indicates the new process that the particles converged again. Finally, by applying Eqs. (12) and (13) in the state of Fig. 20i, we got $N_s/N = 55/60 > \eta$ and $n_s/N_s = 46/55 < \xi$, which suggested that a successful source confirmation process was accomplished.

From the results of the above simulation, two conclusions can be drawn:

1. Existence of the pseudo sources can to a certain extent effect the source localization process. That is to say, the gas source confirmation is actually an indispensable part of the gas searching strategy.
2. The pseudo source processing algorithm based on the APF method can effectively prevent the particles from converging again to the pseudo source position.

6. Conclusion and future work

In this paper, a complete strategy is proposed in search of dangerous gas sources. The proposed scheme includes the zigzag plume discovery method, the plume tracking method based on particle filtering method and information entropy reward, and the source confirmation method with a pseudo source processing algorithm. Assisted mechanisms such as M-H sampling are also designed for addressing the problems that may arise in this process. Overall, this article mainly has three contributions:

- (1) A complete source localization scheme is proposed, while detailed method and the corresponding algorithms are given.

- (2) In the process of plume tracking, we avoid the phenomenon of repeated exploration by improving the reward function of the updating strategy

- (3) A gas source confirmation method capable of further processing the pseudo source while securing the localization of the real target is presented.

Corresponding experiments and simulations show that the algorithm proposed in this paper is able to locate the gas source accurately. Efficiency is also improved in comparison with the existing method. Moreover, the proposed source localization strategy is well adapted to the obstacle scene.

Future research focus will be put on a new gas distribution model with considering obstacles to improve the adaptation of the strategy to real world. After that, more research will be carried out on the practical validation of the proposed scheme in the outdoor environment to improve practicality as well as the practical validation of pseudo-source processing algorithm in real world. In addition, we will try to enhance the emergency response capacity of the robot by considering different situation such as a moving source and the searching method for multiple sources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was supported by National Natural Science Foundation of China under Grant No. 61773141 and 61105088, Natural Science Foundation of Heilongjiang Province, China under Grant No. F2018016.

Appendix A. Particle filter algorithm and information entropy in plume tracking

A.1. Particle filter algorithm

The posterior probability distribution of estimating source parameter based on particle filtering can be expressed as:

$$p(\Gamma_{s,k} | z_{1:k}) \approx \sum_{i=1}^N \omega_k^{(i)} \delta(\Gamma_{s,k} - \Gamma_{s,k}^{(i)}) \quad (\text{A.1})$$

where $\Gamma_{s,k}$ represents the state vector of the source estimated at time k , $z_{1:k} = z_1, \dots, z_k$ represents the observation result collected until time k , $\omega_k^{(i)}$ is the normalized weight of the i th

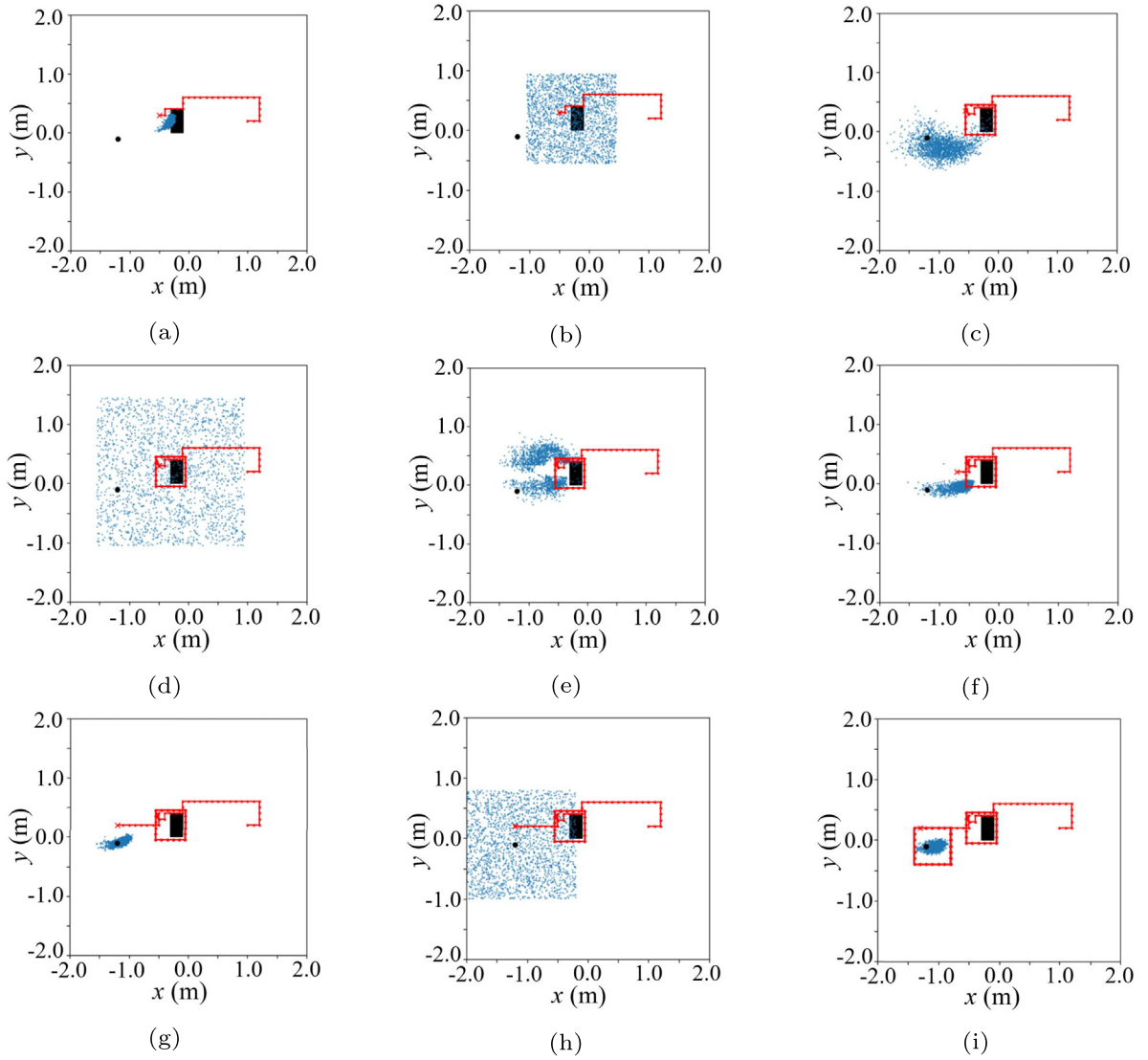


Fig. 20. Simulation of pseudo source processing. (a): the state when particles converge to a pseudo source at the end of plume tracking. (b): uniform sampling of particles at the beginning of source confirmation stage. (c): the state of particles at the end of source confirmation stage. (d): uniform resampling of particles at the beginning of the second plume tracking stage. (e)–(f): evolution of particles during plume tracking stage. (h): resampling of particles at the beginning of the second source confirmation stage. (i): final state at the end of source confirmation stage.

particle at time k ($\sum_{i=1}^N \omega_k^{(i)} = 1, i \in 1, \dots, N$), $\delta(\cdot)$ represents the Dirac function.

We use Sequential Importance Sampling algorithm to get $\{\Gamma_{s,k}^{(i)}, \omega_k^{(i)}\}_{i=1}^N$, which is expressed as:

$$\{\Gamma_{s,k}^{(i)}, \omega_k^{(i)}\}_{i=1}^N = SIS(\{\Gamma_{s,k-1}^{(i)}, \omega_{k-1}^{(i)}\}_{i=1}^N, z_{1:k}) \quad (\text{A.2})$$

In the algorithm, $\Gamma_{s,k}^{(i)}$ is sampled first and $\Gamma_{s,k}^{(i)} \sim q(\Gamma_{s,k}^{(i)} | \Gamma_{s,k-1}^{(i)}, z_{1:k})$, which means that a series of new samples $\{\Gamma_{s,k}^{(i)}\}_{i=1}^N$ at the time k can be obtained from the importance distribution $q(\Gamma_{s,k}^{(i)} | \Gamma_{s,k-1}^{(i)}, z_{1:k})$. Then the unnormalized particle weight $\omega_k^{*(i)}$ is expressed as follows:

$$\omega_k^{*(i)} = \omega_{k-1}^{(i)} \frac{p(z_k | \Gamma_{s,k}^{(i)}) p(\Gamma_{s,k}^{(i)} | \Gamma_{s,k-1}^{(i)})}{q(\Gamma_{s,k}^{(i)} | \Gamma_{s,k-1}^{(i)}, z_{1:k})} \quad (\text{A.3})$$

Assuming that the gas source position to be estimated is fixed and the gas release rate is constant, there is $\Gamma_{s,k}^{(i)} = \Gamma_{s,k-1}^{(i)}$ for any particle $i \in \{1, \dots, N\}$. Then the normalized particle weight can

be simplified as:

$$\omega_k^{(i)} = \frac{\omega_k^{*(i)}}{\sum_{i=1}^N \omega_k^{*(i)}} = \frac{\omega_{k-1}^{(i)} p(z_k | \Gamma_{s,k}^{(i)})}{\sum_{i=1}^N \omega_{k-1}^{(i)} p(z_k | \Gamma_{s,k}^{(i)})} \quad (\text{A.4})$$

According to Eq. (A.4), the weight of the particle at time k depends on the likelihood probability $p(z_k | \Gamma_{s,k}^{(i)})$ and the weight of the particle at the previous moment, $\omega_{k-1}^{(i)}$.

To solve the likelihood probability $p(z_k | \Gamma_{s,k}^{(i)})$, we introduce the Poisson distribution model, which can represent the matching degree between the expected count rate and the real measured data of the sensor in unit time. The expected count rate is obtained based on the sensor measurement model, which is also the gas diffusion model, as shown in Eq. (2) in the body of the paper. The calculation formula of Poisson distribution model is as follows:

$$p_r(h_k | \lambda_k) = \frac{\lambda_k^{h_k}}{h_k!} e^{-\lambda_k} \quad (\text{A.5})$$

where $\lambda_k = \int_0^t R(p_k(t') | \Gamma_s) dt'$ represents the expected counting rate, i.e. the average number of the gas particles being detected by the sensor in an unit sampling time t , R represents the average

frequency that the gas particle being detected by the sensor. p_k represents the location of the gas sensor. Γ_s represents the source parameter. h_k represents the number of the particles that being captured by the gas sensor at position p_k , which is the actual measurement value of the sensor.

In order to ensure the uniform dimension of Poisson distribution, concentration units are used for both λ_k and h_k in Eq. (A.5). Because of $h_k \in \mathbb{Z}^+$, we let $h_k = \lfloor z_k \rfloor$, where $\lfloor \cdot \rfloor$ is to round down and z_k is the unrounded concentration value obtained by the sensor. Then, the likelihood probability can be expressed as follows:

$$p(z_k | \Gamma_{s,k}^{(i)}) = \frac{[C(p_k, \Gamma_{s,k}^{(i)})]^{z_k}}{[z_k]!} \exp[-C(p_k, \Gamma_{s,k}^{(i)})] \quad (\text{A.6})$$

where $C(p_k, \Gamma_{s,k}^{(i)})$ represents the expectation of the observational concentration caused by the temporal source particle i at position p_k and time k .

There is particle degradation phenomenon in SIS particle filter algorithm, which means that the weight of some particles becomes very small after continuous iteration, so that their role in posteriori probability estimation is negligible. At the same time, these particles with small weights occupy a lot of computing resources, which greatly affects the performance of estimation. The effective particle number N_{eff} is usually used to represent the degradation degree of particles.

$$N_{\text{eff}} \approx \frac{1}{\sum_{i=1}^N (\omega_k^{(i)})^2} \quad (\text{A.7})$$

It is necessary to resample particles when N_{eff} is less than the set threshold. The basic idea is to replicate the particles with significant weight and discard the particles with very low weight while keeping the total number of particles, N , constant. In this paper, a stochastic resampling algorithm based on stratified statistical idea is adopted.

Due to iteratively replicating high-weight particles, the diversity of particles is greatly reduced, causing the iteration unable to continue, which is called particle dissipation. In this study, regular particle filtering and M-H (Metropolis-Hastings) sampling are combined to effectively reduce the influence of particle dissipation. Regular particle filtering introduces new samples through gaussian kernel. After that, M-H sampling algorithm compares the likelihood value of new samples with that of the original samples, and then accepts or rejects these samples with a certain probability according to the comparison result. This algorithm is based on MCMC algorithm, while solves the problem of high rejection rate of MCMC mobile sampling, improving the sampling efficiency.

By sampling the nuclear density of each particle, regular particle filter converts the discrete approximate probability distribution into a continuous approximate probability distribution, and the posterior probability distribution can be expressed as:

$$p(\Gamma_{s,k} | z_{1:k}) \approx \sum_{i=1}^N \omega_k^{(i)} K_h(\Gamma_{s,k} - \Gamma_{s,k}^{(i)}) \quad (\text{A.8})$$

$$K_h(x) = \frac{(\det S)^{-1/2^n}}{h} K\left(\frac{1}{h} A^{-1} x\right) \quad (\text{A.9})$$

where $K_h(x)$ is the new kernel function after re-scaling the kernel density $K(\cdot)$, h is the kernel bandwidth, n is the dimension of the state vector, and $S = AA^T$ is the covariance matrix of particles. The kernel density function shall be a symmetric probability density function, namely $\int K(x) dx = 1$ and $\int \|x\|^2 K(x) dx < \infty$.

In this paper, gaussian kernel density function is $k(x) = \Phi(x)$, where $\Phi(x)$ is the standard normal probability density function, and the optimal kernel width is expressed as:

$$h_{\text{opt}} = (4/(n+2))^{1/(n+4)} N^{-1/(n+4)} \quad (\text{A.10})$$

where N is the number of particles. The covariance matrix of $\{\Gamma_{s,k}^{(i)}, \omega_k^{(i)}\}$ is shown in Eq. (A.11).

$$S_k = \sum_{i=1}^N \omega_k^{(i)} (\Gamma_{s,k}^{(i)} - \mu_k)(\Gamma_{s,k}^{(i)} - \mu_k)^T = A_k A_k^T \quad (\text{A.11})$$

Then, the equation used to update particles with regular particle filtering is $\Gamma_{s,k}^{(i)} = \Gamma_{s,k}^{(i)} + h_{\text{opt}} A_k e^i$, where $e^i \sim N(0, I_n)$.

The M-H sampling compares the likelihood probability values of the particles updated by regular particle filtering and that of the original particles to decide whether to accept the transfer of particles. Sampling u according to $u \sim U[0, 1]$, then whether to accept the transfer is judged with following equation.

$$\alpha = \min\{1, \frac{p(z_k | \Gamma_{s,k}^{(i)})}{p(z_k | \Gamma_{s,k}^{(i)})}\} \quad (\text{A.12})$$

where α is the acceptance probability. If $\mu \leq \alpha$, we accept the transfer and set $\Gamma_{s,0:k}^{(i)} = \{\Gamma_{s,0:k-1}^{(i)}, \Gamma_{s,k}^{(i)}\}$; Otherwise we reject the transfer and set $\Gamma_{s,0:k}^{(i)} = \{\Gamma_{s,0:k-1}^{(i)}, \Gamma_{s,k}^{(i)}\}$.

A.2. Information entropy

Information entropy, which is used to describe the uncertainty of the information source, is another essential algorithm for proceeding the plume tracking. Since the higher the entropy is, the more uncertainty lies in the information, the uncertainty of the information should reduce most when the robot is moving towards the direction with maximum information entropy. The motion control is performed under the robot coordinate system while the updating formula of the robot's route can be described as follows:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} \cos \theta_k & -\sin \theta_k & 0 \\ \sin \theta_k & \cos \theta_k & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} dx_{k+1} \\ dy_{k+1} \\ d\theta_{k+1} \end{bmatrix} \quad (\text{A.13})$$

where $\theta \in [0^\circ, 360^\circ)$ represents the robot's direction, $[x_k \ y_k \ \theta_k]^T$ and $[x_{k+1} \ y_{k+1} \ \theta_{k+1}]^T$ represent the robot position before and after updating, respectively. $[dx_{k+1} \ dy_{k+1} \ d\theta_{k+1}]^T$ represents the robot motion increment after the control instructions have been given. Let $M_k = \{\uparrow, \downarrow, \leftarrow, \rightarrow\}$, whose elements indicate the action of moving forward, backward, towards the left and right from the current position, the motion increment corresponding to M_k can then be represented as below:

$$\begin{bmatrix} d_f^T \\ d_b^T \\ d_l^T \\ d_r^T \end{bmatrix} = \begin{bmatrix} \lambda_m & 0 & 0^\circ \\ -\lambda_m & 0 & 180^\circ \\ 0 & \lambda_m & 90^\circ \\ 0 & -\lambda_m & 270^\circ \end{bmatrix} \quad (\text{A.14})$$

where vectors d_f , d_b , d_l and d_r represents the movement increment of moving forward, backward, towards the left and right, respectively, while λ_m represents the length of the moving step.

The information entropy function employed in this paper is the classic Shannon entropy, which is generally expressed as follows:

$$H(X) = E(-\log p(x_i)) = -\sum_{i=1}^n p(x_i) \log p(x_i) \quad (\text{A.15})$$

Based on the above strategy, the best direction of the motion corresponding to the maximum entropy can be derived as follows:

$$m_k^* = \arg \max_{m_k \in M_k} \{H(m_k)\} \quad (\text{A.16})$$

$$H(m_k) = - \sum_{\hat{z}_{k+1}=0}^{[\hat{z}]_{\max}} p(\hat{z}_{k+1}(\hat{p}_{k+1})|Z_{1:k}) \log p(\hat{z}_{k+1}(\hat{p}_{k+1})|Z_{1:k}) \quad (\text{A.17})$$

where $m_k \in M_k$, $Z_{1:k} = \{z_1(p_1), \dots, z_k(p_k)\}$ indicates the sensor measurement so far collected by moving from the track planning point p_1 to p_k , \hat{p}_{k+1} represents the next alternative planning point, $\hat{z}_{k+1}(\hat{p}_{k+1})$ indicates the expected sensor readout at the next position \hat{p}_{k+1} after executing m_k . Although the actual value of $\hat{z}_{k+1}(\hat{p}_{k+1})$ is unknown, it can still be approximated by the current posterior probability distribution obtained by particle filtering. The derivation based on the full probability formula is as follows:

$$p(\hat{z}_{k+1}(\hat{p}_{k+1})|Z_{1:k}) \approx \int p(\hat{z}_{k+1}(\hat{p}_{k+1})|\Gamma_{s,k})p(\Gamma_{s,k}|Z_{1:k})d\Gamma_{s,k} \quad (\text{A.18})$$

Posterior probability distribution can be approximated with $\{\Gamma_{s,k}^{(i)}, \omega_k^{(i)}\}_{i=1}^N$, and then $p(\hat{z}_{k+1}(\hat{p}_{k+1})|\Gamma_{s,k})$ is the corresponding likelihood probability in Eq. (A.6) while $p(\Gamma_{s,k}|Z_{1:k})$ is the corresponding normalized particle weight. We can get Eq. (A.19).

$$p(\hat{z}_{k+1}(\hat{p}_{k+1})|Z_{1:k}) \approx \sum_{i=1}^N \frac{[C(\hat{p}_{k+1}, \Gamma_{s,k}^{(i)})]^{\hat{z}_{k+1}(\hat{p}_{k+1})}}{[\hat{z}_{k+1}(\hat{p}_{k+1})]!} \times \exp[-C(\hat{p}_{k+1}, \Gamma_{s,k}^{(i)})]\omega_k^{(i)} \quad (\text{A.19})$$

Then, $H(m_k)$ in Eq. (A.16) can be calculated, with m_k that corresponding to the maximum $H(m_k)$ as the next target point.

Appendix B. Variable residual parameter determination based on Monte-Carlo

There is a dimension disaster problem in particle filter, that is, the particle estimation accuracy decreases with the increase of the dimension. We can introduce a large number of particles to ensure the performance of the algorithm, but too many particles will also lead to the degradation of the computational performance. Therefore, the particle dimension should be reduced as much as possible and the initial parameter interval should be narrowed. In this paper, in order to improve the performance of particle filter, we use integrated sensors directly measure the information of wind direction φ_s and wind speed v_s . There are three remaining unknown parameters in Eq. (2) in the body of the paper in addition to the gas source location and the robot location parameter, namely, gas release rate r_s , effective diffusivity D_s and gas particle lifetime τ_s , which can usually give a more accurate prior distribution according to the dangerous gas information of the actual environment [30]. However, it is difficult to directly determine those values in the experimental environment of this paper. Thus, a monte-carlo-based variable residual parameter determination algorithm is proposed to determine the initial value interval of unknown parameters as the prior information of the experiment.

During the experiment, we first collect the gas concentration distribution curve at specified position in the gas field, as shown in Fig. B.21. Where, the blue curve is the concentration curve, the red line represents the mean of the concentration, and the abscissa is the number of collection. Here, the sampling interval is set as 0.3 s and the number of sampling is 200. Then, each figure represents the change of concentration in the corresponding sampling position within 1min, and the ordinate is the concentration data in mg/m³. The robot positions, mean value of concentration and concentration change interval corresponding to each curve are shown in Table B.4.

In Table B.4, all other parameters are the same except for the position of the robot. The gas source position x_s is set as

-1 , y_s is 0, the wind speed near the gas source v_s is measured by the sensor as 2.0 m/s, the wind direction φ_s is 0° in the initial robot coordinate system, and 180° in the world coordinate system corresponding to the gas concentration distribution map.

The concentration residual formula is defined as formula (B.1) after obtaining part of parameters and the corresponding mean value of gas concentration.

$$\Delta c^{ij} = |C(r_s^i, D_s^i, \tau_s^i) - c_z^j| \quad (\text{B.1})$$

where, Δc^{ij} is the residual of concentration, which represents the absolute value of the difference between the mean value calculated based on gas diffusion equation and the mean value observed by the sensor. $C(x)$ corresponds to Eq. (5), with gas release rate r_s , effective diffusivity D_s and gas particle life τ_s as input parameters and i represents the particle number. c_z^j is the mean value observed by the sensor and j represents the sampling position that corresponds to Table B.4.

The core idea of the variable residual parameter determination based on the monte carlo is to use particles to simulate a large number of candidate parameters and then screen a small number of particles that meet the constraint conditions by continuously strengthening the constraint on residual of concentration. We set $r_s^i \in U(100, 1000)$, $D_s^i \in U(0.01, 3.0)$ and $\tau_s^i \in U(0.1, 5.0)$, which means that r_s^i , D_s^i , and τ_s^i are sampled from the uniform distribution of the corresponding interval respectively, with a large interval in the initial stage. We let initial particle number $n_0 = 1.0 \times 10^7$, initial concentration residual constraint $\Delta c_0 = 100$, initial step size $s_0 = 10$ and threshold value $s_0 = 10$. We use t to represent the number of iterations, and n_t to represent the number of particles at the number of iterations t .

For each sampling position j , the algorithm traverses all particles and calculates each Δc^{ij} according to Eq. (B.1). If $\Delta c^{ij} \leq \Delta c_t$, the particle is retained and move to the next position to continue screening, otherwise the particle is discarded. When all sampling positions are traversed once, it is proved that all currently retained particles satisfy $\Delta c^i \leq \Delta c_t$ at all sampling positions. At this time, the constraint condition is enhanced as $\Delta c_{t+1} = \Delta c_t - s_t$ to continue to traverse all sampling positions with all remaining particles. When the number of particles retained satisfies $0 < n_t \leq n_{th}$, the iteration ends, otherwise, repeat the above process. However, the constraint is continuously enhanced and there will be over-constraining with the progress of iteration, which causes the number of particles to suddenly drop to 0. In this case, we should trace back the particle state and residual before the number of particles becomes 0 and reduce the step size. In this paper, we set the step length $s_{t+1} = 0.1 \times s_t$ and then continue to iterate. At the end of iteration, the residual value is the minimum residual constraint that all remaining particles meet, namely the fitting error of concentration value. And now the relevant parameters satisfying the minimum residual constraint can be estimated according to the remaining particles.

The data changes in the iterative process of the algorithm are shown in Fig. B.22 and the number of iterations is 119. Fig. B.22a indicates that the value of residual constraint changes as the number of iterations increases, and $\Delta c_{119} = 22.97$ at the end of the iteration. The change of particle number in the iterative process is shown as Fig. B.22b. The ordinate is expressed by logarithm since the number of particles varies greatly and the number of particles is 0 when over-constraining. And the number of particles is $n_{119} = 4$ at the end of the iteration. The step size variation in the iteration is represented by $\lg(s)$ as shown in Fig. B.22c. The maximum step size and the minimum step size are 10 and 0.01 respectively. There were three times of over-constraining in the iterative process according to Fig. B.22, corresponding to $t = 52$, $t = 98$ and $t = 102$. At the end of the iteration, the parameter values and mean values of the remaining particles are shown in

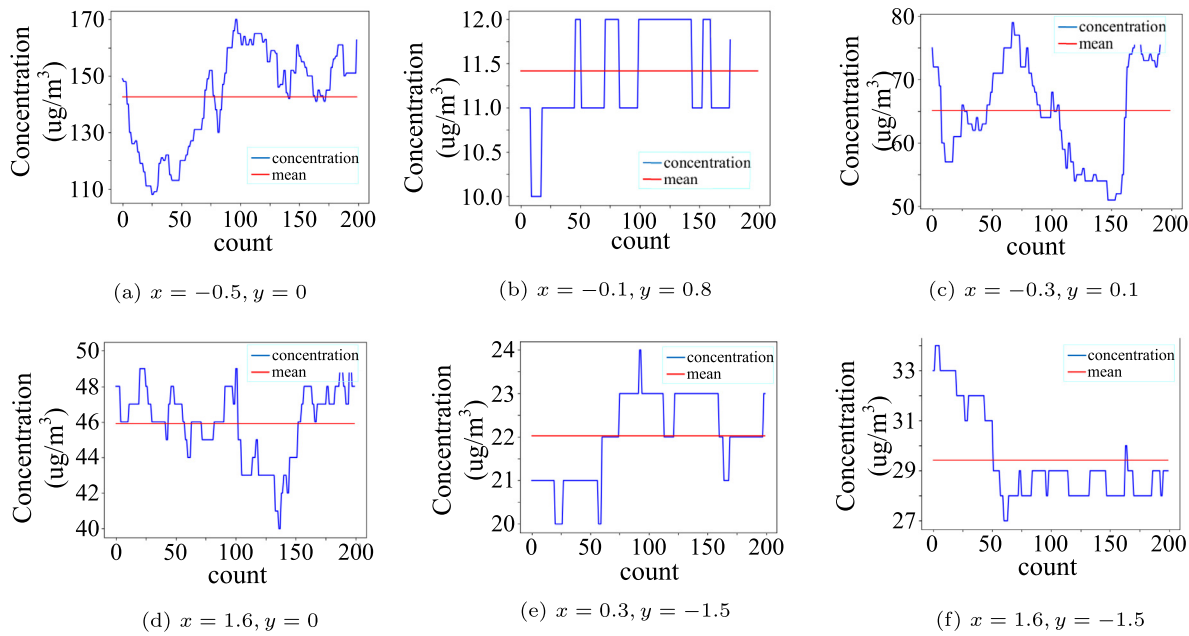


Fig. B.21. Concentration distribution curve.

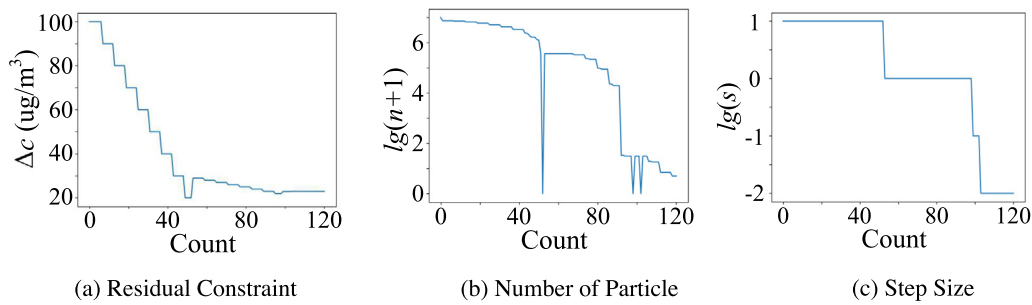


Fig. B.22. Data curve during iteration.

Table B.4
Position parameters and mean value of concentration curve.

Sequence	a	b	c	d	e	f
x (m)	-0.5	-0.1	0.3	1.6	0.3	1.6
y (m)	0	0.8	0.1	0	-1.5	-1.5
Mean value (mg/m ³)	126.620	11.420	65.125	45.915	22.025	29.430
Change interval (mg/m ³)	[110,170]	[10,12]	[51,79]	[40,50]	[20,24]	[27,34]

Table B.5
Estimated results.

Parameters	Sequence				Mean
	1	2	3	4	
r_s	475.909	457.515	455.147	446.972	458.886
D_s	0.492	0.473	0.471	0.462	0.475
τ_s	4.983	4.983	4.995	5.000	4.990

Table B.5, and the mean values is $r_s = 458.886$, $D_s = 0.475$ and $\tau_s = 4.990$. Drawing gas concentration distribution map based on the above mean value, as shown in Fig. B.23, in which each sampling position is marked. By comparing the concentration in the above figure with the mean concentration measured by the sensor in Table B.5, it can be seen that the estimated parameters

can make the gas distribution field better fit the measured data of the sensor.

As can be seen from Fig. B.23, during the experiment, the plume has a great discontinuity and fluctuation, and the turbulence effect is obvious. The closer to the gas source, the greater the fluctuation of gas concentration, and the smaller the gas concentration gradient when away from the gas source. At the same time, we cannot accurately measure the concentration at the specified location because the sensor has a certain measurement error. Therefore, according to the variable residual parameter determination algorithm based on monte carlo, the estimated parameter value cannot be directly determined as the accurate parameter, but a reasonable initial value interval can be estimated according to the obtained mean value. Finally, the parameters prior information are $r_s^i \sim U(350, 700)$, $D_s^i \sim U(0.4, 0.8)$ and $\tau_s^i \sim U(3.5, 5.0)$.

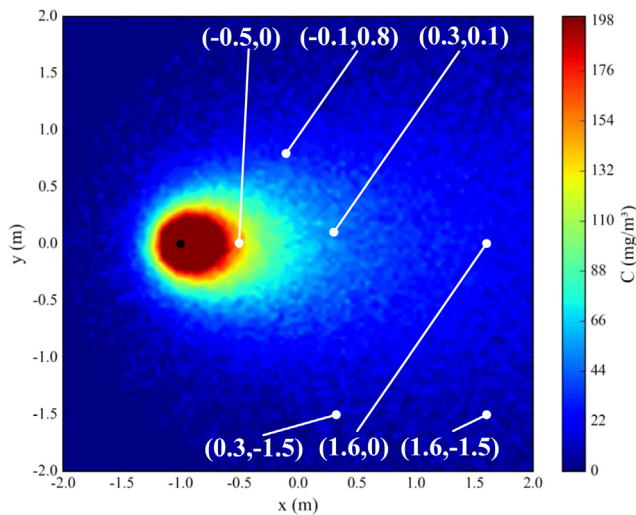


Fig. B.23. Data curve during Iteration.

References

- [1] S.K. Singh, M. Sharan, J.-P. Issartel, Inverse modelling methods for identifying unknown releases in emergency scenarios: an overview, *Int. J. Environ. Pollut.* 57 (1–2) (2015) 68–91.
- [2] M. Redwood, Source term estimation and event reconstruction: a survey, in: *Contract Report for ADMLC*, Vol. 51790, 2011.
- [3] R.A. Russell, Locating underground chemical sources by tracking chemical gradients in 3 dimensions, in: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE Cat. No. 04CH37566), Vol. 1, IEEE, 2004, pp. 325–330.
- [4] J.A. Farrell, S. Pang, W. Li, R. Arrieta, Chemical plume tracing experimental results with a REMUS AUV, in: *Oceans 2003. Celebrating the Past... Teaming Toward the Future* (IEEE Cat. No. 03CH37492), Vol. 2, IEEE, 2003, pp. 962–968.
- [5] P.P. Neumann, V. Hernandez Bennetts, A.J. Lilienthal, M. Bartholmai, J.H. Schiller, Gas source localization with a micro-drone using bio-inspired and particle filter-based algorithms, *Adv. Robot.* 27 (9) (2013) 725–738.
- [6] M. Hutchinson, C. Liu, W.-H. Chen, Source term estimation of a hazardous airborne release using an unmanned aerial vehicle, *J. Field Robotics* 36 (4) (2019) 797–817.
- [7] Y. Chen, H. Cai, Z. Chen, Q. Feng, Using multi-robot active olfaction method to locate time-varying contaminant source in indoor environment, *Build. Environ.* 118 (2017) 101–112.
- [8] A. Ivanov, M. Campbell, Joint exploration and tracking: JET, *IEEE Control Syst. Lett.* 2 (1) (2017) 43–48.
- [9] B. Ristic, D. Angley, B. Moran, J. Palmer, Autonomous multi-robot search for a hazardous source in a turbulent environment, *Sensors* 17 (4) (2017) 918.
- [10] A.T. Hayes, A. Martinoli, R.M. Goodman, Distributed odor source localization, *IEEE Sens. J.* 2 (3) (2002) 260–271.
- [11] M. Hutchinson, H. Oh, W.-H. Chen, A review of source term estimation methods for atmospheric dispersion events using static or mobile sensors, *Inf. Fusion* 36 (2017) 130–148.
- [12] T.M. Morse, S.R. Lockery, T.C. Ferrée, Robust spatial navigation in a robot inspired by chemotaxis in *Caenorhabditis elegans*, *Adapt. Behav.* 6 (3–4) (1998) 393–410.
- [13] W. Li, J.A. Farrell, S. Pang, R.M. Arrieta, Moth-inspired chemical plume tracing on an autonomous underwater vehicle, *IEEE Trans. Robot.* 22 (2) (2006) 292–307.
- [14] V. Braitenberg, *Vehicles: Experiments in Synthetic Psychology*, MIT press, 1986.
- [15] H. Ishida, K.-i. Suetsugu, T. Nakamoto, T. Moriizumi, Study of autonomous mobile sensing system for localization of odor source using gas sensors and anemometric sensors, *Sensors Actuators A* 45 (2) (1994) 153–157.
- [16] O. Holland, C. Melhuish, Stigmergy, self-organization, and sorting in collective robotics, *Artif. Life* 5 (2) (1999) 173–202.
- [17] L. Marques, A.T. De Almeida, Electronic nose-based odour source localization, in: *6th International Workshop on Advanced Motion Control. Proceedings* (Cat. No. 00TH8494), IEEE, 2000, pp. 36–40.
- [18] M. Vergassola, E. Villermaux, B.I. Shraiman, 'Infotaxis' as a strategy for searching without gradients, *Nature* 445 (7126) (2007) 406.
- [19] J.-G. Li, Q.-H. Meng, Y. Wang, M. Zeng, Odor source localization using a mobile robot in outdoor airflow environments with a particle filter algorithm, *Auton. Robots* 30 (3) (2011) 281–292.
- [20] J.R. Bourne, E.R. Pardyjak, K.K. Leang, Coordinated Bayesian-based bio-inspired plume source term estimation and source seeking for mobile robots, *IEEE Trans. Robot.* 35 (4) (2019) 967–986.
- [21] B. Ristic, A. Gunatilaka, R. Gailis, Localisation of a source of hazardous substance dispersion using binary measurements, *Atmos. Environ.* 142 (2016) 114–119.
- [22] J.R. Bourne, K.K. Leang, Mutual information control for target acquisition: A method to localize a gas/chemical plume source using a mobile sensor, in: *ASME 2017 Dynamic Systems and Control Conference*, American Society of Mechanical Engineers Digital Collection, 2017.
- [23] H. Hajieghrary, D. Mox, M.A. Hsieh, Information theoretic source seeking strategies for multiagent plume tracking in turbulent fields, *J. Mar. Sci. Eng.* 5 (1) (2017) 3.
- [24] Y. Wang, H. Huang, L. Huang, B. Ristic, Evaluation of Bayesian source estimation methods with prairie grass observations and Gaussian plume model: A comparison of likelihood functions and distance measures, *Atmos. Environ.* 152 (2017) 519–530.
- [25] C. Barbieri, S. Cocco, R. Monasson, On the trajectories and performance of infotaxis, an information-based greedy search algorithm, *Europhys. Lett.* 94 (2) (2011) 20005.
- [26] A.M. Hein, S.A. McKinley, Sensing and decision-making in random search, *Proc. Natl. Acad. Sci.* 109 (30) (2012) 12070–12074.
- [27] N. Voges, A. Chaffiol, P. Lucas, D. Martinez, Reactive searching and infotaxis in odor source localization, *PLoS Comput. Biol.* 10 (10) (2014) e1003861.
- [28] B. Ristic, A. Skvortsov, A. Gunatilaka, A study of cognitive strategies for an autonomous search, *Inf. Fusion* 28 (2016) 1–9.
- [29] M. Hutchinson, H. Oh, W.-H. Chen, Entrotaxis as a strategy for autonomous search and source reconstruction in turbulent conditions, *Inf. Fusion* 42 (2018) 179–189.
- [30] M. Hutchinson, C. Liu, W.-H. Chen, Information-based search for an atmospheric release using a mobile robot: Algorithm and experiments, *IEEE Trans. Control Syst. Technol.* (2018) 1–15.
- [31] A. Lilienthal, H. Ulmer, H. Frohlich, A. Stutzle, F. Werner, A. Zell, Gas source declaration with a mobile robot, in: *IEEE International Conference on Robotics and Automation*, 2004. *Proceedings. ICRA'04*. 2004, Vol. 2, IEEE, 2004, pp. 1430–1435.
- [32] F. Li, Q.-H. Meng, J.-W. Sun, S. Bai, M. Zeng, M. IEEE, Single odor source declaration by using multiple robots, in: *AIP Conference Proceedings*, Vol. 1137, AIP, 2009, pp. 73–76.
- [33] Y. Wang, H. Zhu, W. Wang, Octree-based repetitive pose detection of large-scale cyclic environments, in: *2018 3rd International Conference on Robotics and Automation Engineering (ICRAE)*, IEEE, 2018, pp. 60–64.
- [34] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, in: *Autonomous Robot Vehicles*, Springer New York, New York, NY, 1990, pp. 396–404.



Hongbiao Zhu obtained his M.Sc. in Vehicle Engineering from Harbin Institute of Technology at Weihai in 2017. Currently, he is a Ph.D. student at Harbin Institute of Technology, pursuing his study in State Key Laboratory of Robotics and System at Harbin Institute of Technology. His main research interests lie in multiple robotic system.



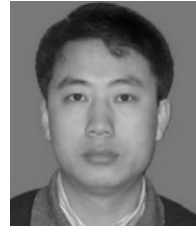
Yibo Wang obtained his M.Sc. in Mechatronic Engineering from Harbin Institute of Technology in 2019. His research interests lie in ground robots and SLAM.



Chengjin Du received the B.S. degree from the School of Mechanical Engineering, Tianjin University of Science and Technology, in 2017. He is currently pursuing the master's degree with the Harbin Institute of Technology, Harbin, China. His research interests include intelligent learning algorithm, pHRI problems, and force/shape sensing problems and its applications in surgical robot.



Quan Zhang obtained his B.Sc. in Mechatronic Engineering from Harbin Institute of Technology at Weihai in 2018. Currently, he is a master student at Harbin Institute of Technology, pursuing his study in State Key Laboratory of Robotics and System at Harbin Institute of Technology. His research interests lie in ground robots and SLAM.



Weidong Wang received the B.S., M.S., and Ph.D. degrees in mechatronics engineering from the Harbin Institute of Technology, in 2002, 2004, and 2009, respectively, where he is currently with the State Key Laboratory of Robotics and System. His research expertise is in the general areas of robotics and mechatronics. His research interests include motion planning, computer vision, human-machine interaction, and its application in field robot and surgical robot.