Robotica (2025), **43**, pp. 2100–2120
doi:10.1017/S0263574725000050

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Multi-robot area coverage and source localization method based on variational sparse Gaussian process

Kai Cao[1,2], Yunbo Wei[1], Yangquan Chen[1,2], Song Gao[1], Kun Yan[1] and Shibo Yang[1]

[1]School of Electronic Information Engineering, Xi'an Technological University, Xi'an, China
[2]School of Engineering (MESA-Lab), University of California, Merced, CA, USA
Corresponding author: Song Gao; Email: gaos@xatu.edu.cn

Received: 11 June 2024; **Revised:** 18 December 2024; **Accepted:** 6 January 2025; **First published online:** 21 May 2025

Keywords: multi-robot; centroid Voronoi tessellation; variational sparse Gaussian process; buffering factor; source localization

Abstract

This paper introduces a distributed online learning coverage control algorithm based on sparse Gaussian process regression for addressing the problem of multi-robot area coverage and source localization in unknown environments. Considering the limitations of traditional Gaussian process regression in handling large datasets, this study employs multiple robots to explore the task area to gather environmental information and approximate the posterior distribution of the model using variational free energy methods, which serves as the input for the centroid Voronoi tessellation algorithm. Additionally, taking into consideration the localization errors, and the impact of obstacles, buffer factors and centroid Voronoi tessellation algorithms with separating hyperplanes are introduced for dynamic robot task area planning, ultimately achieving autonomous online decision-making and optimal coverage. Simulation results demonstrate that the proposed algorithm ensures the safety of multi-robot formations, exhibits higher iteration speed, and improves source localization accuracy, highlighting the effectiveness of model enhancements.

1. Introduction

In recent years, with the rapid development of computer technology, sensor technology, and artificial intelligence, multi-robot collaborative technology has been widely used in search and rescue [1–3], inspection [4, 5], and pollution source localization [6, 7]. Multi-robot systems effectively complete complex tasks through self-organized cooperative control, which is often beyond the capability of a single robot [8–10]. Compared to individual robots, multi-robot collaborative formation has the following advantages [11–13]:

(1) Multiple robots can collaboratively complete given tasks, significantly enhancing work efficiency.
(2) Due to the weaker ability of individual robots in handling multiple tasks, collaborative work can compensate for individual shortcomings, expanding the application scenarios and capabilities.
(3) Multi-robot systems can execute different tasks simultaneously, greatly improving task completion efficiency.
(4) Robots in a cluster system complement each other and have fault tolerance. Even if an individual robot fails to perform a task due to malfunctions, other robots can substitute and complete the task, enhancing system fault tolerance, robustness, and flexibility.
(5) Leveraging resource distribution characteristics within the system reduces the cost of individual robots and simplifies system design, reducing losses when executing hazardous tasks.

© The Author(s), 2025. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574725000050 Published online by Cambridge University Press

Based on these advantages, multi-robots can efficiently and accurately complete tasks such as area coverage and source localization in complex environments. However, the prerequisite for completing these tasks is to develop a reasonable coverage strategy, enabling robots to make autonomous decisions and select positions with maximum benefits.

In the field of coverage and source localization in unknown environments, there has been extensive research [14–16]. Literature reviews indicate that research on source localization using multi-robots often employs techniques such as density gradients [17–19], information entropy [20–22], or Bayesian methods [23–25]. Density gradient methods utilize local gradient information for target localization, adapting to real-time environmental changes and not relying on prior map information. A method combining gradient and probabilistic search for source localization is proposed in ref. [26]. The gradient method is used for UAV navigation and tracking, while probabilistic search is used to estimate the position of contaminants, reducing search time and area through the combination of these two methods. However, gradient methods are prone to local optima, especially in complex environments, which may affect the accuracy and efficiency of source localization. Some scholars use deep reinforcement learning methods to guide robot motion. Without relying on prior map information, robots interact with the environment, learning optimal pollution source localization strategies through exploration and environmental feedback. A method based on deep reinforcement learning for indoor odor source localization is proposed in ref. [27], modeling the odor source localization problem as a Markov decision process and designing state space, action space, and dense rewards to address sparse reward issues, finally generating the robot's optimal decision through the proximal policy optimization algorithm. Ref. [28] considers the problem of multi-robot cooperative exploration in unknown environments, combining Voronoi diagrams with deep reinforcement learning strategies to dynamically divide Voronoi regions based on robot movement in the environment and complete obstacle avoidance based on deep reinforcement learning methods. While this method yields good search results, it still faces potential challenges and limitations such as sample complexity and long training times. Intelligent optimization algorithms simulate collective behaviors of birds or fish, enabling all individuals to reach a global optimum through mutual cooperation. Ref. [29] proposes a multi-robot olfaction method independent of airflow information. This method utilizes particle swarm search to track odor sources and broadens the robot's search range by introducing extreme perturbation beacons to prevent robots from getting stuck in local extremes. However, the drawbacks of heuristic intelligent algorithms lie in slow convergence speed, difficulty in modeling environmental fields, and poor interpretability of the algorithm. To address this issue, ref. [30] adopts the Gaussian Mixture Model method to predict the environmental model, guiding robots to explore and reconstruct the environment based on the Voronoi diagram method. The advantage of this method is its ability to reconstruct unknown environmental models, but it exhibits high computational complexity when dealing with large-scale datasets, requiring significant storage and computing resources, leading to longer training times.

In terms of multi-robot collision avoidance, common methods include artificial potential field methods [31–33], A* [34–36], and reinforcement learning methods [37–39]. However, the artificial potential field method is prone to trapping robots in local optima, and unreasonable potential field functions may prevent robots from finding the global optimal path. The performance of the A* method highly depends on the choice of heuristic function, and in cases of large or complex search spaces, the time complexity of the A* algorithm may be high, leading to slow search speeds, making it unsuitable for applications with high real-time requirements. Reinforcement learning methods require a large amount of trial and error to train the model, resulting in low sample efficiency, requiring significant time and resources to obtain sufficient empirical data. The drawbacks of existing obstacle avoidance methods mainly lie in local optima and low convergence speed, both of which may prevent robots from completing the assigned tasks. Compared to traditional path planning methods, multi-robot control methods based on Voronoi diagrams allow each robot to generate its own Voronoi region based on its position and the surrounding environment, enabling distributed path planning and control [40, 41]. Additionally, the boundaries of Voronoi regions can be used to avoid collisions between robots, as each robot will only move within its allocated region. This collision avoidance mechanism can enhance system safety and

https://doi.org/10.1017/S0263574725000050 Published online by Cambridge University Press

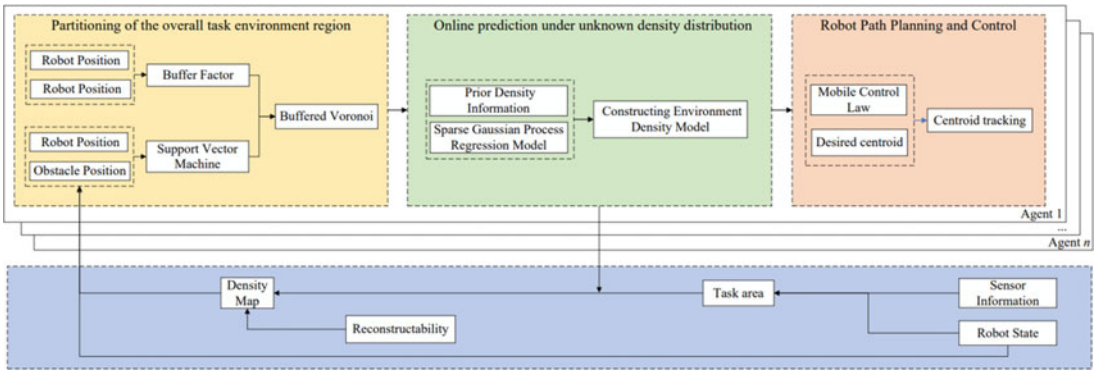


Figure 1. *Overview of the overall framework.*

efficiency. In the author's research, ref. [42] considers communication constraints and large localization errors during the formation process. It enhances collision avoidance capabilities by constructing buffer Voronoi cells for each robot and imposes communication distance constraints on Voronoi regions to simulate communication-limited environments. Finally, obstacle avoidance strategies are designed to achieve formation control in non-convex environments. Ref. [43] proposes a multi-robot encirclement method based on buffered Voronoi cells. This method utilizes separate hyperplanes and buffers to create Voronoi regions that ensure robot safety, devising pursuit control laws to enable robots to swiftly encircle evaders. The buffered Voronoi method can avoid robot collisions while maintaining fast computational speeds, overcoming the shortcomings of traditional path planning methods. Furthermore, this method does not require kinematic and dynamic modeling of robots, making it easier to deploy on actual robots.

Therefore, this paper proposes a new framework that uses a multi-robot control strategy based on Centroidal Voronoi Tessellation (CVT) to address the problems of area coverage and source localization in unknown environments. This framework first ensures the safety of the multi-robot formation, then reconstructs the distribution of the unknown field, and determines the exact location of the pollution source. It is worth noting that the entire process operates online, meaning that robots autonomously make decisions to accomplish the assigned tasks. The overall framework is illustrated in Figure 1.

This paper makes three main contributions: Firstly, it addresses the uncertainty of robot positions by enhancing safety between robots through constructing buffered Voronoi cells. Secondly, it proposes an online multi-robot coverage algorithm based on CVT, which constructs a density function model of the unknown environment using the Variational Sparse Gaussian Process Regression (VS-GPR) method, and plans task areas and exploration target points for robots by introducing the centroid Voronoi partition algorithm with uncertainty. Lastly, it designs formation control laws and proves the stability of the algorithm through Lyapunov stability analysis, thereby achieving formation coverage and source localization in unknown areas.

The structure of the remaining parts of the paper is as follows: Section 2 introduces the basic theory of multi-robot safety control. Section 3 proposes an online coverage and source localization algorithm based on VS-GPR and proves the stability of the system by constructing a Lyapunov function. Section 4 provides corresponding simulation results and analysis. Section 5 validates the effectiveness of the algorithm through UAV experiments. The conclusion and future prospects are presented in Section 6.

2. Multi-robot area coverage safety control

2.1. Coverage function

After obtaining the location information of n robots, it is necessary to design a region coverage method to drive each robot towards its target point in order to achieve optimal coverage of the task area.

https://doi.org/10.1017/S0263574725000050 Published online by Cambridge University Press

The Voronoi diagram assigns non-overlapping regions to each robot, and can dynamically update the shape and size of the Voronoi regions following the iterative dynamics of the robots.

In the environment $\Omega \subseteq \mathrm{R}^N$, let $E = \{e_1, e_1, \ldots, e_n\}$ be the set of n robots, $P = \{p_1, p_1, \ldots, p_n\}$ be the set of robot positions, and the definition of Voronoi cells is as follows:

$$V_i = \{p \in \Omega \mid \|q - p_i\| < \|q - p_j\|, \forall j \neq i\} \tag{1}$$

where $i, j = 1, \ldots, n$, $\|\cdot\|$ represents the Euclidean norm on R^N.

Given the density function ρ, the objective function for region coverage is

$$\mathrm{H}(p_1, \ldots, p_n) = \sum_{i=1}^{n} \int_{V_i} \|q - p_i\|^2 \rho(q)\, dq \tag{2}$$

According to the inertia theory, the mass M_{V_i} and the centroid C_{V_i} of Voronoi cells can be expressed as:

$$M_{V_i} = \int_{V_i} \rho(q)\, dq \tag{3}$$

$$C_{V_i} = \frac{1}{M_{V_i}} \int_{V_i} q\rho(q)\, dq \tag{4}$$

In addition, $J_{V_{p_i}}$ represents the polar moment of inertia of the i-th robot.

$$J_{V_{p_i}} = \int_{V_i} \|q - p_i\|^2 \rho(q)\, dq \tag{5}$$

According to the parallel axis theorem, this can be further expressed as follows:

$$J_{V_{p_i}} = J_{V_{c_i}} + M_{V_i} \left\|p_i - C_{V_i}\right\|^2 \tag{6}$$

where $J_{V_{c_i}}$ represents the polar moment of inertia of the Voronoi region around its centroid.

Another form of $H(P)$can be derived from Eq. (6):

$$H(P) = \sum_{i=1}^{k} J_{V_{c_i}} + \sum_{i=1}^{k} M_{V_i} \left\|p_i - C_{V_i}\right\|^2 \tag{7}$$

Then, the derivative of $H(P)$ with respect to its pair in Eq. (7) is

$$\frac{\partial H(P)}{\partial p_i} = 2M_{V_i}\left(p_i - C_{V_i}\right) \tag{8}$$

The expression for the control law is

$$u_i = 2K_i M_{V_i}\left(p_i - C_{V_i}\right) \tag{9}$$

where K_i is a positive definite gain matrix, usually an identity matrix.

When the robot is at its centroid, the system has the lowest cost function,

$$C_{V_i} = arg\ min\ \mathrm{H}(p_1, \ldots, p_n) = \frac{\int_V q\rho(q)\, dq}{\int_V \rho(q)\, dq} \tag{10}$$

2.2. Safety buffer Voronoi cell

In robotic systems, the actual position of the robot may be affected by sensor error, communication delay, or environmental interference. Ref. [44] pointed out that systematic displacement and rotation errors would occur when using the navigation GNSS data on the UAV for geo-reference, resulting in deviations in the collected data. Ref. [45] proposed that unbalanced terahertz line scanning exposure sources would lead to uneven energy distribution, which would affect the accuracy of imaging results. These errors can lead to inaccurate robot path planning, resulting in collisions between robots.

https://doi.org/10.1017/S0263574725000050 Published online by Cambridge University Press

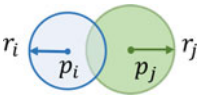


Figure 2. *Position error range.*

In density-based CVT algorithms, when initializing, agents are generally treated as point masses, ignoring their actual size. In addition, there exist position errors in the agent positioning system, leading to a certain level of uncertainty in the agent's position information. Traditional CVT algorithms do not consider the two above-mentioned issues, resulting in agents colliding with each other during the process of tracking centroids.

As shown in Figure 2, the blue circle and the blue dot represent the perceptual range and position of robot i respectively, while the green circle and the green dot represent the perceptual range and position of robot j respectively. Due to sensor error, communication delay, environmental interference, and other factors, the actual position of the robot may cause errors. The position of two robots may be any point within their respective circles, and when the distance between two robots is less than r_i or r_j, there is a risk of collision between two robots. The blue and green circles represent the position error upper bounds of agents p_i and p_j. In the overlapping area of the position error circles, the agents will collide with each other.

2.2.1. Collision avoidance between robots

To address the issue of collision avoidance between robots, the CVT algorithm is extended by introducing a buffer factor to enhance the obstacle avoidance performance between agents.

Considering a group of n agents moving in the spatial environment $\Omega \subseteq \mathrm{R}^N$, for a set of n agents in R^N, the Voronoi cell definition of Eq. (1) can be re-expressed as:

$$\mathrm{V}_i = \left\{ p \in \mathrm{R}^N \mid \left(p - \frac{p_i + p_j}{2} \right)^T \left\| p_{ij} \right\| \leq 0, \forall j \neq i \right\} \tag{11}$$

where $p_{ij} = p_j - p_i$. When considering both the actual size of agents and position errors, introducing a buffer region constrains the Voronoi cells. The calculation equation for the buffered centroid Voronoi cell is defined as follows:

$$\mathrm{V}_{b,i} = \left\{ p \in \mathrm{R}^N \mid \left(p - \frac{p_i + p_j}{2} \right)^T p_{ij} + \eta_{ij} \left\| p_{ij} \right\| \leq 0, \forall j \neq i \right\} \tag{12}$$

where η_{ij} represents the buffer factor between two agents, and the size of the buffer factor is determined by the position error radius of each agent, that is,

$$\eta_{ij} = r_i + r_j \tag{13}$$

Figure 3 shows two different Voronoi partitions, where Figure 3(a) represents the standard Voronoi partition, while the improved CVT algorithm shrinks the boundaries of Voronoi cells based on the size of agents to ensure their safety, as shown in Figure 3(b).

2.2.2. Mutual avoidance between robots and obstacles

We adopt the support vector machine (SVM) method to partition the safe area between robots and obstacles. The hyperplane between two robots p_i and p_j can be represented as:

$$a_{ij}{}^T p = b_{ij} \tag{14}$$

where a_{ij} is the normal vector determining the direction of the hyperplane, and b_{ij} is the distance term determining the distance between the hyperplane and the origin.

https://doi.org/10.1017/S0263574725000050 Published online by Cambridge University Press

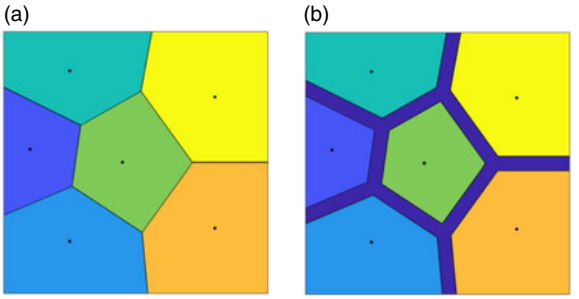


Figure 3. *Two different Voronoi partitions. (a) Standard Voronoi partition. (b) Buffered Voronoi partition.*

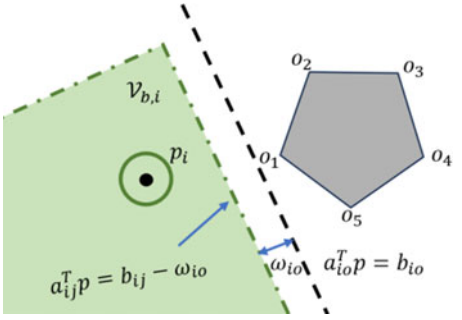


Figure 4. *Avoidance Voronoi cell.*

Obstacles generally have irregular characteristics, setting the vertex set of obstacles as $O_m = \{o_1, o_2, \ldots, o_m\}$, and generating safe Voronoi cells by computing the optimal separating hyperplane between robot p_i and the vertex set of obstacles O_m, as shown in Figure 4.

By solving the following objective optimization problem, the optimal separating hyperplane can be quickly computed.

$$\min_{a_{io}, b_{io}} \frac{1}{2} \mathrm{P} a_{io} \mathrm{P}^2 \tag{15}$$

$$s.t. \quad y_i\left(a_{io}{}^T p_i + b_{io}\right) \geq 1 \tag{16}$$

where y_i represents the decision function of the support vector machine. a_{io} represents the coefficient of the normal vector of the hyperplane, which determines the direction of the hyperplane, b_{io} represents the bias term of the hyperplane, which determines the position of the hyperplane.

The constraint Voronoi cell between the robot and the obstacle can be represented as

$$\mathrm{V}_i^b = \{p \in \mathbb{R}^d : a_{io}{}^T p < b_{ij} - \omega_{io}, i \in n\} \tag{17}$$

where $\omega_{io} = r_{io} a_{io}$, ω_{io} represents the buffer term between the robot and the obstacle.

3. Online learning coverage control method based on VS-GPR

3.1. Traditional Gaussian process regression (GPR)

GPR is generally used to reconstruct the distribution of unknown environmental fields. The main process is as follows:

(1) Mount sensors on the robot and acquire environmental data through the robot's movement in the environment.

https://doi.org/10.1017/S0263574725000050 Published online by Cambridge University Press

(2) Use the robot's position and sensor data as inputs, and the distribution of the unknown environment as outputs to train the GPR model.

(3) Plan paths for the robot based on the environmental field distribution constructed in the previous step.

Assuming GPR follows a prior Gaussian process with a mean of 0.

$$f \sim GP\left(0, k\left(x, x'\right)\right) \tag{18}$$

where $k(x, x')$ represents the kernel function of the GPR model. In this paper, the radial basis function is used as the kernel function of GPR.

$$k\left(x, x'\right) = Mexp\left(\frac{-\left\|x_i - x_i'\right\|^2}{2l^2}\right) \tag{19}$$

In the equation, M and l are hyperparameters, where M represents the amplitude scale and l^2 represents the length scale. Reasonably selecting optimal hyperparameters can improve the model's fitting accuracy. Generally, the maximum likelihood estimation method is used to construct the likelihood function, and its partial derivative is taken to obtain the hyperparameters of GPR.

Considering the regression task $y = f(x) + \varepsilon$, where ε is independently and identically distributed Gaussian noise that follows a distribution $\varepsilon \sim N(0, \sigma_n^2)$. By maximizing the likelihood function, the hyperparameters are obtained.

After knowing the training set inputs, hyperparameters, and information about the test points, the mean and variance of the predictive model are as follows:

$$\mu_*(x_*) = K(x_*, x)\left[K(x, x) + \sigma_n^2 I\right]^{-1} y \tag{20}$$

$$\sigma_*^2(x_*) = k(x_*, x_*) - K(x_*, x)\left[K(x, x) + \sigma_n^2 I_n\right]^{-1} K(x, x_*) \tag{21}$$

GPR is a powerful non-parametric modeling method that can model any shape space and make precise probability predictions. However, computing the covariance matrix of GPR has a time complexity of $O(N^3)$, making model inference and prediction very difficult with large training set data. Approximate inference methods are needed to reduce complexity in such cases.

3.2. Variational sparse Gaussian process regression

In recent years, the method of sparse Gaussian process regression (SGPR) has developed rapidly, with the core idea of using sparse inducing points to approximate the complete Gaussian process model. Considering the characteristics of online coverage problems, which require fast computation speed and high prediction accuracy, this paper will train the model based on the VS-GPR method, and use it to guide the robot's motion.

The core of the Variational Free Energy (VFE) method is to approximate the posterior distribution in a variational inference manner. Firstly, the posterior distribution is parameterized by a set of inducing points, and optimization is performed with the goal of minimizing the Kullback-Leibler divergence between the variational distribution and the true posterior distribution. In this process, by adjusting the positions of the inducing points, the variational distribution is gradually made to approximate the true posterior distribution, thereby reducing computational complexity. The following is a specific theoretical introduction to VS-GPR.

Let the set of inducing points be $Z = \{z_i\}_{i=1}^M$, corresponding to function values $\boldsymbol{u} = \{u_i\}_{i=1}^N = \{f(z_i)\}_{i=1}^N$. Assuming that given these inducing points, the prediction points are conditionally independent of the training samples,

$$p(f_* \mid \boldsymbol{y}) = \int p(f_* \mid \boldsymbol{u})\, p(\boldsymbol{u} \mid \boldsymbol{y})\, d\boldsymbol{u} \tag{22}$$

https://doi.org/10.1017/S0263574725000050 Published online by Cambridge University Press

The VFE method largely solves the problem of overfitting in prior approximation, and the approximate posterior distribution tends to be closer to the true posterior distribution.

Through Jensen's inequality, the lower bound of the marginal likelihood can be calculated.

$$\begin{aligned} log\, p(\boldsymbol{y}) &= log \int p(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{u})\, p(\boldsymbol{f}, \boldsymbol{u})\, d\boldsymbol{f}\, d\boldsymbol{u} \\ &= log \int p(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{u})\, p(\boldsymbol{f}, \boldsymbol{u})\, \frac{q(\boldsymbol{f}, \boldsymbol{u})}{q(\boldsymbol{f}, \boldsymbol{u})} d\boldsymbol{f}\, d\boldsymbol{u} \\ &\geq \int q(\boldsymbol{f}, \boldsymbol{u})\, log\, \frac{p(\boldsymbol{y} \mid \boldsymbol{f}, \boldsymbol{u})\, p(\boldsymbol{f}, \boldsymbol{u})}{q(\boldsymbol{f}, \boldsymbol{u})} d\boldsymbol{f}\, d\boldsymbol{u} \end{aligned} \tag{23}$$

where $q(\boldsymbol{f}, \boldsymbol{u})$ is called the variational distribution. When $q(\boldsymbol{f}, \boldsymbol{u}) = p(\boldsymbol{f}, \boldsymbol{u} \mid \boldsymbol{y})$ is optimal, the equality holds.

The form of $q(\boldsymbol{f}, \boldsymbol{u})$ is set as:

$$q(\boldsymbol{f}, \boldsymbol{u}) = p(\boldsymbol{f} \mid \boldsymbol{u})\, \phi(\boldsymbol{u}) \tag{24}$$

when $q(\boldsymbol{f}, \boldsymbol{u})$ tends to $p(\boldsymbol{f}, \boldsymbol{u} \mid \boldsymbol{y})$, $\phi(\boldsymbol{u}) \approx p(\boldsymbol{u} \mid \boldsymbol{y})$, while $p(\boldsymbol{f} \mid \boldsymbol{u}) \approx p(\boldsymbol{f} \mid \boldsymbol{u}, \boldsymbol{y})$.

By eliminating the factors $p(\boldsymbol{f} \mid \boldsymbol{u})$ in the numerator and denominator of the logarithm, we get

$$log\, p(\boldsymbol{y}) \geq \int p(\boldsymbol{f} \mid \boldsymbol{u})\, \phi(\boldsymbol{u})\, log\, \frac{p(\boldsymbol{y} \mid \boldsymbol{f})\, p(\boldsymbol{u})}{\phi(\boldsymbol{u})} d\boldsymbol{f}\, d\boldsymbol{u} \triangleq F_V(Z, \phi) \tag{25}$$

where $F_V(Z, \phi)$ is the objective function. By optimizing the positions of inducing points Z and the variational distribution $\phi(\boldsymbol{u})$ to maximize $F_V(Z, \phi)$. When the equality holds, $log\, p(\boldsymbol{y}) = F_V(Z, \phi)$, $q(\boldsymbol{f}, \boldsymbol{u}) = p(\boldsymbol{f}, \boldsymbol{u} \mid \boldsymbol{y})$, $\phi(\boldsymbol{u}) = p(\boldsymbol{u} \mid \boldsymbol{y})$, $p(\boldsymbol{f} \mid \boldsymbol{u}, \boldsymbol{y}) = p(\boldsymbol{f} \mid \boldsymbol{u})$.

Further compute the analytical solution for $\phi(u)$

$$\phi(u) = \mathrm{N}\left(\sigma^{-2} K_Z \Sigma K_{ZX} \boldsymbol{y}, K_Z \Sigma K_Z\right) \tag{26}$$

where $\Sigma = [K_Z + \sigma^{-2} K_{ZX} K_{XZ}]^{-1}$, Σ is an inverse matrix used in variational inference to determine the optimal distribution of the induced variables. Substitute $\phi(u)$ into $F_V(Z, \phi)$ to obtain

$$F_V(Z) = log\, \left[\mathrm{N}\left(0, \sigma^2 Q_X + I_N\right)\right] - \frac{1}{2\sigma^2} Tr\left(\tilde{K}\right) \tag{27}$$

where $Q_X = K_{XZ} K_Z^{-1} K_{ZX}$, $\tilde{K} = K_X - K_{XZ} K_Z^{-1} K_{ZX}$.

After optimization is completed, predictions can be made using $\phi(u)$

$$p(f_* \mid \boldsymbol{y}) = \mathrm{N}\left(m_*, \sigma_*^2\right) \tag{28}$$

Analytical solution for calculating mean and variance

$$m_* = \sigma^{-2} \boldsymbol{k}_{*Z} \Sigma K_{ZX} \boldsymbol{y} \tag{29}$$

$$\sigma_*^2 = k_* - \boldsymbol{k}_{*Z} K_Z^{-1} \boldsymbol{k}_{Z*} + \boldsymbol{k}_{*Z} \Sigma \boldsymbol{k}_{Z*} \tag{30}$$

where $\boldsymbol{k}_{*Z} = \boldsymbol{k}(\boldsymbol{x}_*, Z)$, $\boldsymbol{k}_{Z*} = \boldsymbol{k}(Z, \boldsymbol{x}_*)$.

The VFE method achieves effective modeling and prediction for large-scale datasets by introducing sparsity in the model, which means only partial data points' information is retained. Its training time complexity is $O(NM^2)$, and the prediction time complexity is $O(M^2)$.

https://doi.org/10.1017/S0263574725000050 Published online by Cambridge University Press

3.3. Online learning coverage control strategy

This section proposes an online learning coverage control algorithm called SGPR-BCVT. The algorithm is based on CVT as its basic framework. It obtains prior information by having robots move in the environment, then learns and predicts the global density function model online, and finally assigns different learning or coverage tasks to the robots.

First, the following assumptions are made:

Assumption 1. There are n robots performing coverage and pollution source localization tasks in a non-convex region, where the density function is unknown in the initial stage.

Assumption 2. Communication between n robots is always maintained, ignoring communication delays.

Traditional multi-robot control under CVT algorithm is limited by the density function, where robots have higher interest in high-density areas. However, online coverage control tasks require robots to cover more areas. In adaptive sampling, the entropy method shows good performance. Therefore, we use the entropy of the model error variance as the weight of the CVT algorithm to guide robots in area coverage, which can avoid robots getting trapped in local optimal solutions. Entropy is defined in information theory as:

$$H(\sigma) = \sigma^2 \, ln \, \sigma^2 \tag{31}$$

where σ represents the posterior variance of the predictive model.

The cost function of the CVT algorithm can be expressed as:

$$\mathrm{H}(p) = \sum_{i=1}^{n} \int_{\mathrm{V}_{b,i}} \left[\varepsilon(q) + H(\sigma)\right] \|q - p_i\|^2 \, dq \tag{32}$$

where $\varepsilon(q)$ represents the predicted concentration value of points in Voronoi regions.

The introduction of entropy creates a balance between exploration and exploitation for the robots, meaning they no longer directly head to high-density areas. When robots identify unexplored areas, they increase the weight on those areas. Therefore, the introduction of entropy helps the CVT algorithm to complete coverage more quickly while reducing model error.

$$M_{\mathrm{V}_{b,i}} = \int_{\mathrm{V}_{b,i}} \rho(q) \, dq + \int_{\mathrm{V}_{b,i}} \sigma^2 ln\sigma^2 dq \tag{33}$$

$$C_{\mathrm{V}_{b,i}} = \frac{1}{M_\rho} \int_{\mathrm{V}_{b,i}} q\rho(q) \, dq + \frac{1}{M_{H(q)}} \int_{\mathrm{V}_{b,i}} q\sigma^2 ln\sigma^2 dq \tag{34}$$

where $M_\rho = \int_{\mathrm{V}_{b,i}} \rho(q) dq, M_{H(q)} = \int_{\mathrm{V}_{b,i}} \sigma^2 ln\sigma^2 dq$

Therefore, the centroid of Voronoi cells consists of two parts: density centroid and uncertainty centroid. The position control law of robots can be updated as:

$$\dot{p}_i = -k_p\left(p_{b,i} - C_{\mathrm{V}_{b,i}}\right) \tag{35}$$

where k_p represents a constant used to adjust the robot's step size to prevent the robot from moving too quickly.

Furthermore, define the set of target points $\Omega = \left\{P \,|p_i = C_{\mathrm{V}_{b,i}}\right\}$. When $\|C_{\mathrm{V}_{b,i}} - p_i\| \to 0$, this state is called the convergence configuration. Set the threshold for the single movement distance of the robot as d. The update of each robot's target point is as follows:

$$\begin{cases} p_{i+1} = p_i + d\dfrac{\vec{p_i C}_{\mathrm{V}_{b,i}}}{\left\|p_i - C_{\mathrm{V}_{b,i}}\right\|}, \left\|p_i - C_{\mathrm{V}_{b,i}}\right\| \geq d \\ p_{i+1} = p_i + \vec{p_i C}_{\mathrm{V}_{b,i}}, \left\|p_i - C_{\mathrm{V}_{b,i}}\right\| < d \end{cases} \tag{36}$$

where $\frac{\vec{p_i C}_{\mathrm{V}_{b,i}}}{\left\|p_i - C_{\mathrm{V}_{b,i}}\right\|}$ represents the unit vector between the robot's position and the target point.

https://doi.org/10.1017/S0263574725000050 Published online by Cambridge University Press

3.4. Lyapunov stability analysis

To verify the stability of the controller, select the Lyapunov function as

$$V = \sum_{i=1}^{n} \int_{V_{b,i}} \frac{k_1}{2} \|q - p_i\|^2 f(q)\, dq \tag{37}$$

where $f(q) = \varepsilon(q) + H(\sigma)$.

Taking the derivative of yields

$$\dot{V} = \sum_{i=1}^{n} \int_{V_{b,i}} k_1 (q - p_i)^T f(q)\, dq \dot{p}_i \tag{38}$$

Substitute $\dot{p}_{b,i}$ into the controller, then $\dot{V}_1$ becomes

$$\begin{aligned} \dot{V} &= \sum_{i=1}^{n} \int_{V_{b,i}} k_1 (q - p_i)^T f(q)\, dq \left[k_p \left(C_{V_{b,i}} - p_i\right)\right] \\ &= \sum_{i=1}^{n} -k_1 M_{V_{b,i}} \left(C_{V_{b,i}} - P_i\right)^T k_p \left(C_{V_{b,i}} - p_i\right) \\ &= -k_1 k_p \sum_{i=1}^{n} M_{V_{b,i}} \left\| C_{V_{b,i}} - p_i \right\|^2 < 0 \end{aligned} \tag{39}$$

Given k_1 and k_p are positive values, $M_{V_{b,i}}$ is a positive value, $\dot{V} \leq 0$.

Since $\dot{V} \leq 0$, according to LaSalle's invariance principle, we can conclude that the system converges to the largest invariant set in.

The pseudocode for multi-robot online coverage and target localization algorithm is shown in Algorithm 1.

4. Simulation analysis

This paper verifies the effectiveness of the proposed algorithm through MATLAB 2018a simulation software, setting up two groups of comparative simulation experiments. In order to make the simulation more general, the following designs are implemented:

(1) The maximum number of iterations is set to 120 times;
(2) The kernel function in the VFE method is the radial basis function;
(3) The maximum speed of the robot for each iteration is 1.5 m/s.

4.1. Formation safety verification

To validate the obstacle avoidance performance of the algorithm, the following experiments were conducted. In this experiment, the simulation environment was set to 7 m × 7 m × 7 m, with five robots and four obstacles. To verify the influence of the buffer width on the formation, we set two different buffer parameters, $r = 0$ m and $r = 0.2$ m. By adjusting the width of the buffer zone, changes in the formation performance indicators were observed. Table I shows the experimental parameter settings.

The goal of this experiment is to have five robots reach predetermined locations while avoiding obstacles along the path. Therefore, each robot needs to construct safe Voronoi cells to avoid collisions between robots or between robots and obstacles. The simulation results are shown in Figs. 5 and 6, where in Figure 5(a), (b), and (c), curves of different colors represent the motion trajectories of different robots, and irregular polyhedra represent the Voronoi cells of different robots.

It can be observed that all five robots in the experiment are able to avoid obstacles, ensuring the safety of the formation. From Figs. 5(e) and 6(e), it is evident that as the number of iterations increases, the

https://doi.org/10.1017/S0263574725000050 Published online by Cambridge University Press

Algorithm 1: Pseudocode for multi-robot online coverage and source localization

Require:

Initial position information $\{p_i\}_{i=1,2,\ldots,n}$ and buffer radius $\{\eta_{ij}, r_{io}, i = 1, \ldots, n\}$ of each robot, task coverage area $Q \in R^N$, Minimum distance convergence threshold t.

Procedure:

While The minimum distance d between robots and the centroid $<$ distance threshold t

for i = 1,. . .,n do

Obtain the position information p_i and uncertainty radius η_{ij}, r_{io} of each robot

Calculate the distance $\|q - p_i\|$ between each robot and any point in the area Ω

Construct the buffer Voronoi cell $V_{b,i}$ for each robot

obtain the environmental prior information of robot's position points

Calculate the global VFE sparse model and the predictive variance x_{var} for each area point

if $x_{var} \in V_{b,i}$ ***then***

combine density and uncertainty information update formulas $f(q)$, evaluate the benefit value of points within Voronoi cell $V_{b,i}$

Calculate the mass $M_{V_{b,i}}$ and uncertain centroid $C_{V_{b,i}}$ of Voronoi cell $V_{b,i}$

Calculate the coverage cost function $H(p)$ and target point set χ

Adjust the step length weights k_p and drive robots to perform coverage

if $\|p_i - C_{V_{b,i}}\| \geq d$ ***then***

$$p_{i+1} = p_i + d\frac{\overrightarrow{p_i C_{V_{b,i}}}}{\|p_i C_{V_{b,i}}\|}$$

else

$$p_{i+1} = p_i + \overrightarrow{p_i C_{V_{b,i}}}$$

end if

end if

Obtain the explored point set $X = \{x_n\}_{n=1}^N$ by integrating historical information

end for

end While

Table I. *Experimental parameter settings.*

Parameter	Values
Buffer Band Radius	$r = 0.2$ m, $r = 0$ m
Target Point Positions	(5.5,5.5,2.7)(5.5,6.5,2.5)(6.0.6.0.2.8) (6.5,5.5,2.7)(6.5,6.5,2.5)
Coordinates of Obstacles in 2D	(2.5,2.5)(4.5,2.5)(2.5,4.5)(4.5,4.5)
Size of Obstacles	0.5 m × 0.5 m × 3 m
Maximum Speed of UAVs	0.3 m/s

position error between the robots and their target locations decreases, ultimately achieving convergence. Figures 5(d) and 6(d) show that at the end of the iterations, the volume changes of the five Voronoi cells tend to stabilize. The changes in the minimum distance between robots in Figures 5(f) and 6(f) indicate that the minimum distance between the five robots always remains greater than 0.5 m, meaning that there are no collisions between the robots.

From the comparison between the two sets of simulations in Figs. 4(d) and 5(d), it can be seen that at the end of the iterations, the volume of the Voronoi cell generated by UAV2 with a buffer band radius $r = 0.2$ m is approximately 2.3 m^3, while the volume of the Voronoi cell with $r = 0$ m is

https://doi.org/10.1017/S0263574725000050 Published online by Cambridge University Press

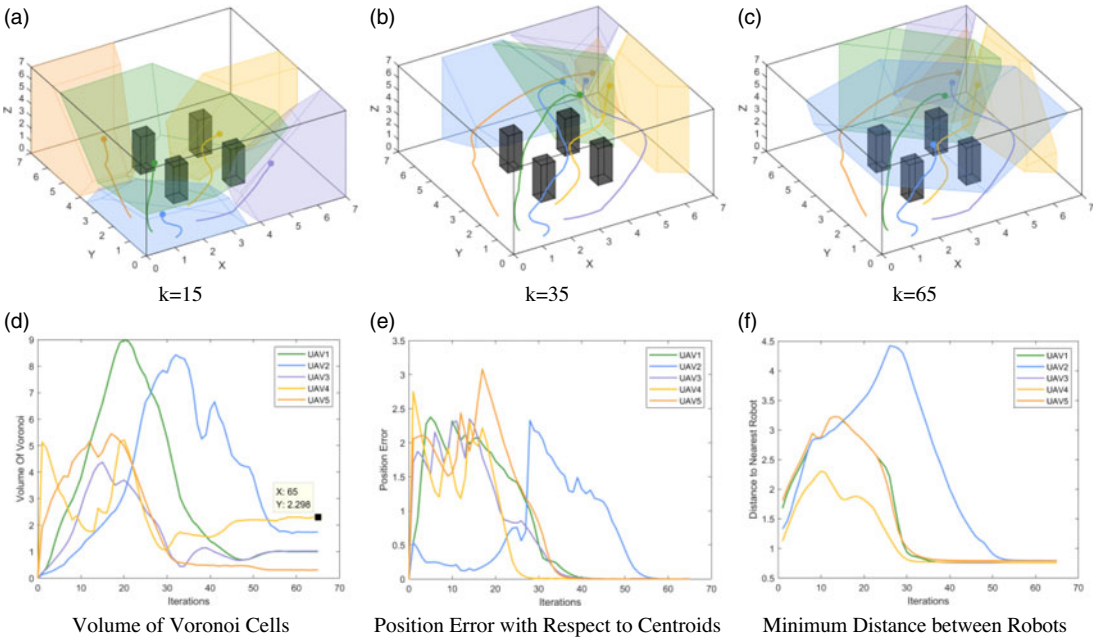


Figure 5. *Multi-UAV formation based on buffered Voronoi partition with buffer band radius $r = 0.2$ m.*

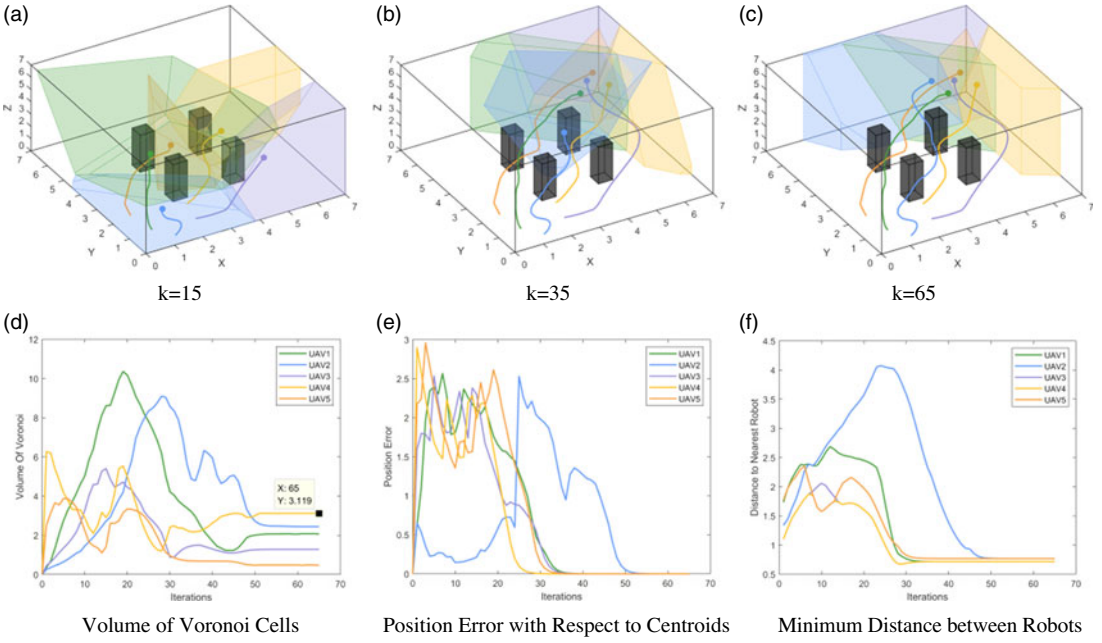


Figure 6. *Multi-UAV formation based on buffered Voronoi partition with buffer band radius $r = 0$ m.*

approximately 3.12 m^3. This indicates that as the width of the buffer band increases, the boundaries of the Voronoi cells continuously contract, and the volume of the Voronoi cells decreases. A wider buffer band ensures a smaller Voronoi region that guarantees the robots' safety. Conversely, a narrower buffer band results in a larger movable safe Voronoi region for the robots.

https://doi.org/10.1017/S0263574725000050 Published online by Cambridge University Press

Table II. *Experimental parameter settings.*

Parameters	Values
Maximum iteration count	120
Buffer factor	$\eta = 0.4m$
Safety distance between robots and obstacles	$d = 0.4m$
Two-dimensional coordinates of obstacles	(5.0,1.5)(1.2,3.5)(4.0,5.0)
Obstacle size	0.6 m × 0.6 m × 3 m
Maximum speed of UAVs	0.25 m/s

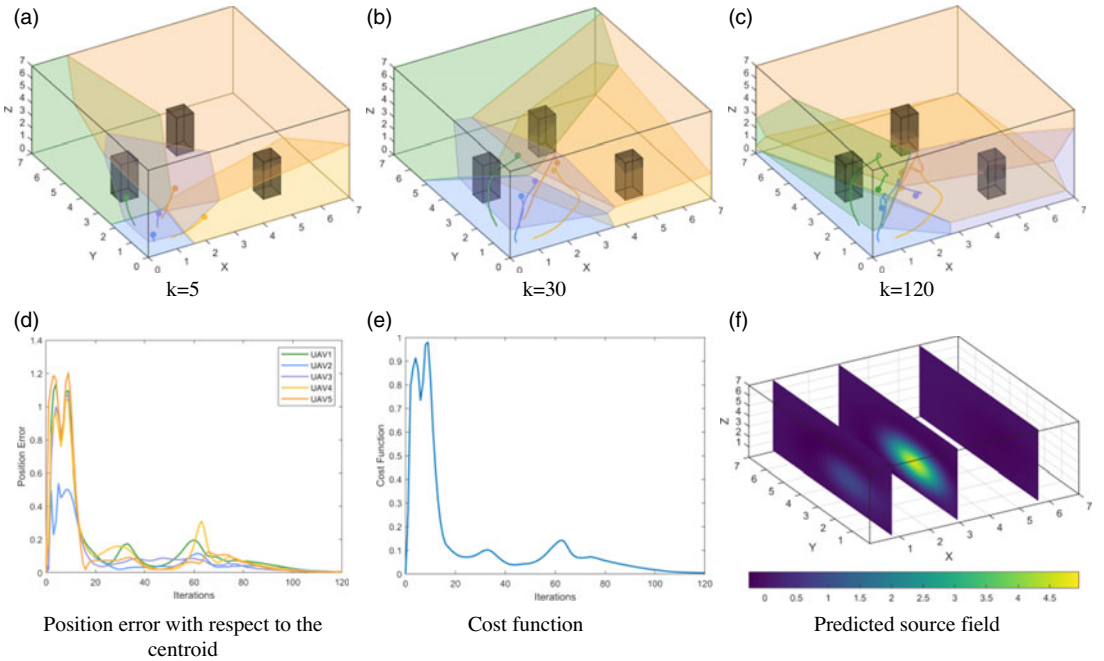


Figure 7. *Multi-UAV coverage and source localization process under the GPR-CVT.*

4.2. Online coverage of multi-pollutant sources

Assuming there are two leakage sources in the coverage area, located at the coordinates (2.5, 2.5, 2.5) and (5.5, 5, 4) respectively, as shown in Figure 2, the actual density function is a compound Gaussian distribution.

$$\rho_{xyz} = 5e^{-0.5*\left[(x-2.5)^2+(y-2.5)^2+(z-2.5)^2\right]} + 10e^{-0.3*\left[(x-5.5)^2+(y-5)^2+(z-4)^2\right]} \tag{40}$$

In the simulation environment, the point (5.5, 5, 4) is the location of the maximum concentration. In order to observe the movement trajectory of the robots more conveniently, the number of obstacles has been reduced to 3, while the number of robots remains at 5. Table II shows the experimental parameter settings.

The online coverage method using VS-GPR will be compared with the traditional GPR method, and the experimental results are as follows.

Figures 7(a) to (f) show the simulation results of multi-robot online coverage tasks completed using the GPR-CVT algorithm, while Figure 8(a) to (f) show the results using the SGPR-BCVT algorithm. As can be seen from the simulation in Figure 7(b), the traditional GPR-CVT algorithm may collide with obstacles during movement, resulting in safety problems for the robot. In the subsequent simulation in Figure 7(c), the traditional GPR-CVT algorithm returned after covering a certain distance, and could

https://doi.org/10.1017/S0263574725000050 Published online by Cambridge University Press

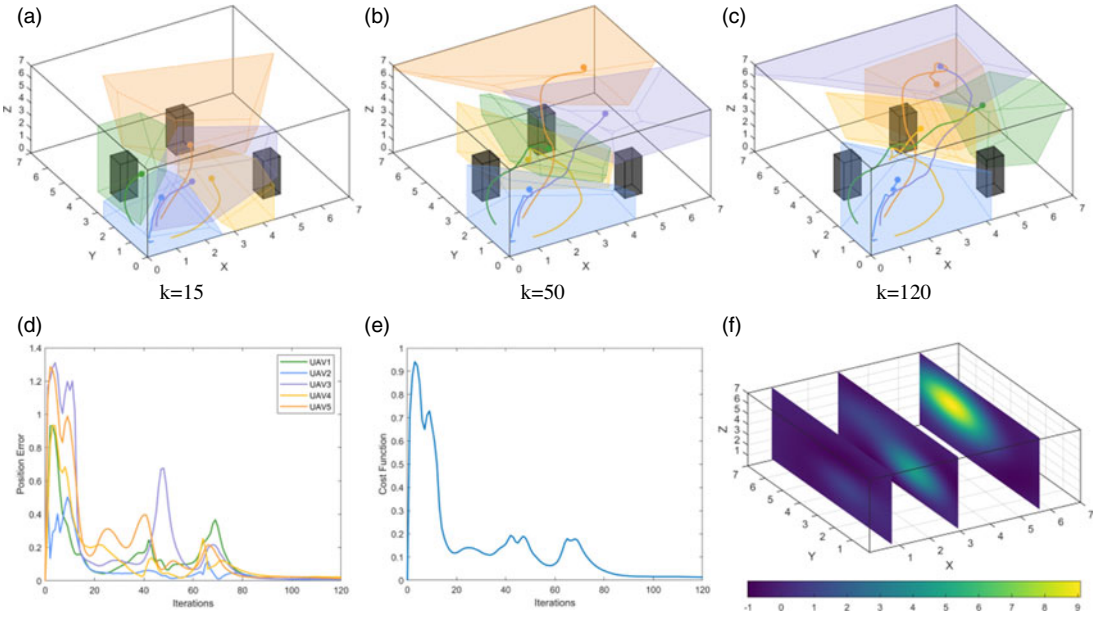


Figure 8. *Multi-UAV coverage and source localization process under the SGPR-BCVT.*

not accurately cover the location of the pollution source. From Figure 8(a), it can be observed that the SGPR-BCVT algorithm introduces a buffer zone between each Voronoi cell during the initial Voronoi cell generation stage to avoid collisions between robots. In addition, it can be seen from the subsequent simulation Figure 8(b) and (c) that the robot has been moving towards the location of the pollution source, so that the robot can locate the location of the pollution source more accurately in the future. Figures 7(f) and 8(f) respectively depict the concentration prediction maps of leakage sources under the GPR-CVT and SGPR-BCVT algorithms. The traditional GPR-CVT algorithm fails to accurately locate the two pollution sources, while the SGPR-BCVT algorithm can cover more areas and ultimately successfully identify the two different pollution sources.

4.2.1. Mutual avoidance between robots and obstacles

Figure 9 shows the simulation of the minimum distance evaluation index between robots and obstacles. The safe distance between robots and obstacles is set to 0.4m, as indicated by the red dashed line in Figure 9(a) and (b). The distance between UAV1 and the obstacle under the GPR-CVT algorithm is less than the safe distance, indicating a collision. However, Figure 5(b) shows that during the formation process, the distance between any robot and obstacle under the SGPR-BCVT algorithm always remains greater than the safe distance, meaning that the robots can safely complete the formation task.

4.2.2. Model accuracy performance index

To verify the accuracy of the source field prediction under the proposed algorithm in this paper, the root mean square error is used to evaluate the regression model. Eq. (41) expresses it specifically.

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(p(x_i) - y_i^*)^2} \tag{41}$$

https://doi.org/10.1017/S0263574725000050 Published online by Cambridge University Press

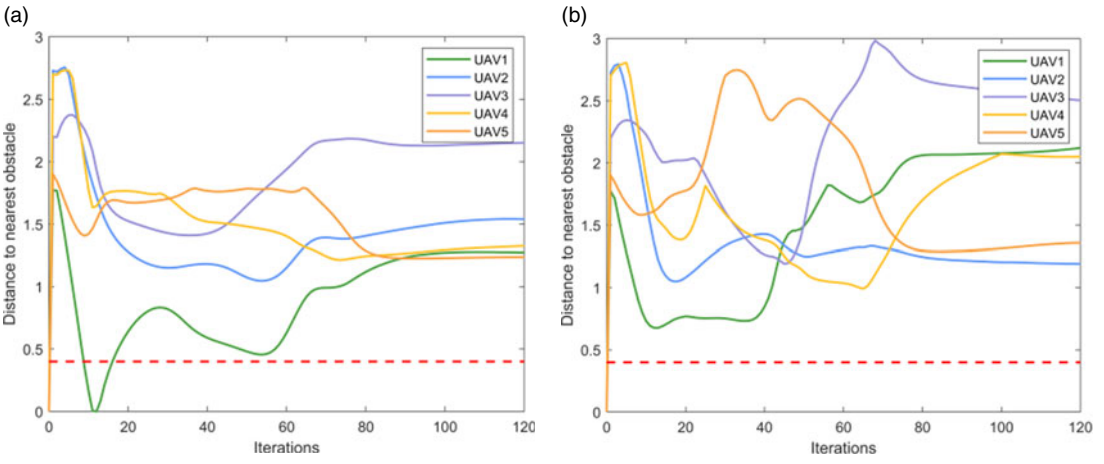


Figure 9. *Minimum distance between UAVs.*

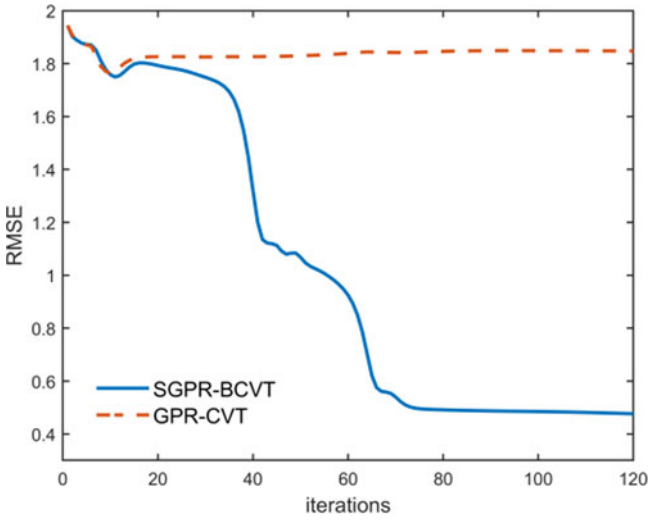


Figure 10. *Model accuracy metrics.*

where $p(x_i)$ represents the predicted value obtained using the SGPR model, and y_i^* represents the true value of the leakage field.

Figure 10 shows the root mean square error of the models established by two algorithms under the source described in Eq. (40). Compared to the real field data, at the end of the iteration, the error of the model established by the SGPR-BCVT algorithm decreased by 75.44%, while due to the GPR-CVT algorithm causing the robot to fall into a local extremum, the model error only decreased by 4.96%. Therefore, in terms of the accuracy index of the established model, the SGPR-BCVT algorithm has a significant advantage.

4.2.3. Accuracy of source localization

In order to evaluate the accuracy of multiple agents in reconstructing hazardous source fields, two evaluation metrics are proposed: DE (Distance Error) and EE (Extremum Error).

https://doi.org/10.1017/S0263574725000050 Published online by Cambridge University Press

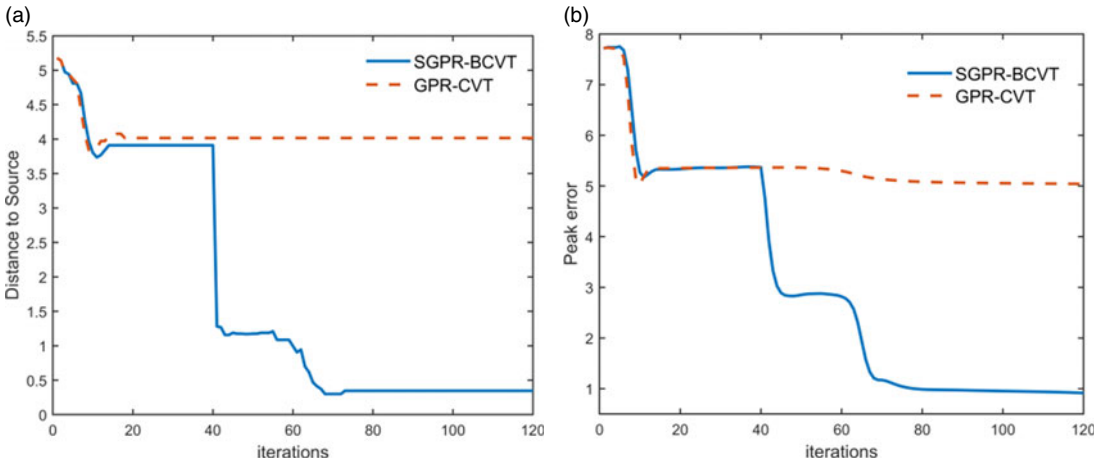


Figure 11. *Accuracy metrics for source localization. (a) Concentration error in extremum point localization. (b) Distance error in extremum point localization.*

DE: The Distance Error metric calculates the difference in distance between the predicted field and the location of the maximum concentration in the real field;

EE: The Extremum Error metric calculates the difference in concentration between the predicted field and the maximum concentration point in the real field;

The smaller the values of the Distance Error and Extremum Error metrics, the higher the credibility of the established model, and the more precise description of the source field can be achieved.

Figure 11 indicates that the performance of the traditional GPR-CVT algorithm is poor and cannot accurately locate the two pollution sources. In contrast, the algorithm proposed in this paper guides the robot to explore uncertain areas as much as possible. Compared with the real field data, the localization distance and concentration errors of the extremum points are reduced by 93.31% and 88.14%, respectively.

4.2.4. Rapidity of online coverage

The simulation runtime comparison of the two algorithms is shown in Table III. It can be seen that the total iteration time of the SGPR-BCVT algorithm is 272.63 s, with an average single iteration time of 2.27 s, while for GPR-CVT, it is 5.61 s. Compared with the two algorithms, there is little difference in the overall path distance, which further indicates that the proposed SGPR-BCVT algorithm can move roughly the same path distance in a short period of time, and can also ensure the accurate location of the pollution source. However, the traditional GPR-CVT algorithm not only takes a long time and the moving distance is roughly the same but also cannot guarantee the location of the pollution source, which further proves the location effectiveness of our proposed algorithm.

In addition, we also compared the running time of GPR-CVT algorithm and SGPR-BCVT algorithm under multiple pollution sources. As shown in Table IV, the runtime of SGPR-BCVT algorithm for locating two to four pollution sources increased by 19.01\% and 4.99\% respectively, while the running time did not increase significantly when increasing to the five pollution source, which was due to the constraints of spatial scope and the number of robots. It can be seen that with the same number of iterations, the running time of the proposed SGPR-BCVT algorithm is always smaller than that of the GPR-CVT algorithm as the number of pollution sources increases.

Based on the analysis of the simulation results above, it can be concluded that the proposed algorithm can not only reduce the iteration time but also effectively approach the prediction model accuracy of the global Gaussian process regression algorithm under the condition that the whole path is roughly the same. Moreover, when the number of sample points is large, the advantages of this algorithm are more significant.

https://doi.org/10.1017/S0263574725000050 Published online by Cambridge University Press

Table III. *Runtime and overall path distance of two online coverage control algorithms.*

Coverage Control Algorithm	GPR-CVT	SGPR-BCVT
Runtime/s	673.22	272.63
Path distance/m	43.5	48.3

Table IV. *Runtime comparison of multiple pollution sources.*

Number of pollution sources	GPR-CVT(s)	SGPR-BCVT(s)
2	673.22	272.63
3	802.23	324.61
4	858.33	347.31
5	886.38	358.66

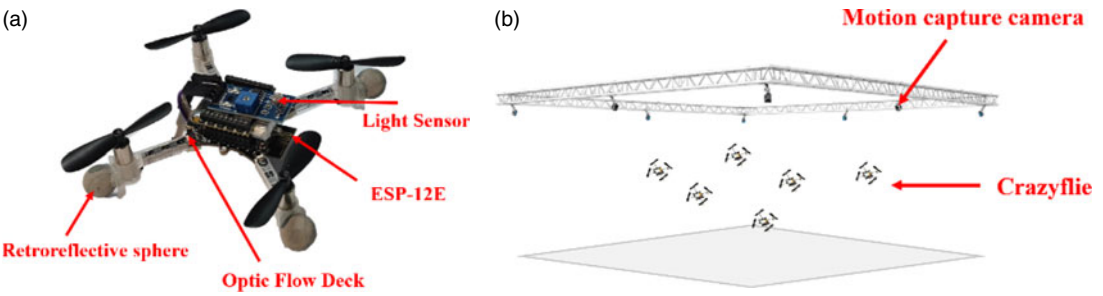


Figure 12. *Experimental environment setup. (a) Crazyflie with attached photosensitive sensor. (b) Schematic diagram of multiple Crazyflie formations.*

5. Experiments

To further validate the effectiveness of the online coverage algorithm proposed in this paper in practical scenarios, this section will conduct experiments on the reconstruction and positioning of light sources in the field using Crazyflie drones equipped with light-sensitive sensors. In this experimental environment, the task area is set to a size of 3.2 m × 3.8 m × 1.5 m, and the radius of each drone is 0.05m.

As shown in Figure 12(a), the Crazyflie drones used in the experiment have the capability of sensing light sources. Firstly, the light-sensitive sensor collects data on light sources and sends the data to the base station via the ESP8266-12E module as prior density information input. Additionally, retroreflective ball markers are attached to the Crazyflie drones, and their real-time position information is captured via Optitrack. This information is then sent to the base station via the ROS system as prior positional information input. The light sensor module can change the output voltage according to the ambient light intensity. The input range of analog voltage is 0 to 4096, so the corresponding illuminance level is calculated based on the voltage measured by the sensor. The overall experimental environment is shown in Figure 12(b).

The overall process of the experiment is as follows: After capturing the motion status information of the Crazyflie and the light intensity information at the drone's current location, the obtained position and field strength information are transmitted to the ROS control terminal. The ROS control terminal analyzes and processes the data through MATLAB-ROS joint programming, ultimately obtaining the distribution of light sources and the next target position of the drone. Finally, the next target position information is sent to the Crazyflie to achieve formation flight of multiple Crazyflie drones.

https://doi.org/10.1017/S0263574725000050 Published online by Cambridge University Press

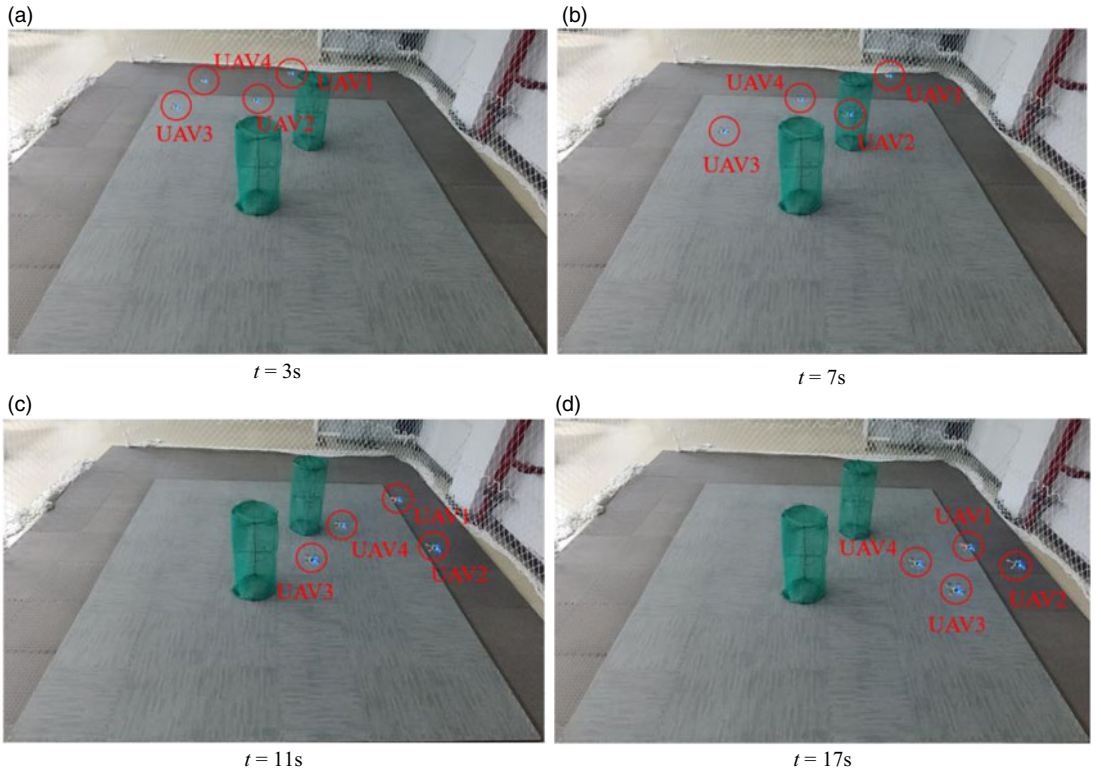


Figure 13. *Formation safety experiment of four drones.*

In the experiment, the gray rectangular area represents the task area. White nets are used around the task environment to ensure flight safety, preventing drones from flying out of the task area and ensuring the safety of flight. The minimum and maximum flight heights of the Crazyflie drones in the experiment are set to 0.5m and 1.5m, respectively.

5.1. Drone formation obstacle avoidance experiment

This experiment is designed to validate the feasibility of the formation obstacle avoidance algorithm in a real environment. Two green cylinders represent obstacles located at fixed positions within the task area. The main task of the drones is to fly to four predetermined target points while avoiding obstacles along the way to ensure the safety of the formation. The minimum distance between Crazyflie drones is set to 0.2m, and the minimum distance between drones and obstacles is set to 0.25m. The formation process is illustrated in Figure 13.

It can be observed that the 4 Crazyflie drones did not collide with obstacles during the flight and accurately reached the designated target positions after 17 s. The experimental results demonstrate the feasibility of applying the formation obstacle avoidance algorithm to physical drones.

5.2. Drone formation coverage and source localization experiment

This experiment aims to validate the feasibility of the proposed coverage algorithm in a real environment. In contrast to the previous experiment, this section introduces the use of hanging light bulb sources to simulate pollution sources. In this experimental scenario, the light sources are located in the bottom right corner of the scene, and three Crazyflie drones depart from the top left corner of the map. The

https://doi.org/10.1017/S0263574725000050 Published online by Cambridge University Press

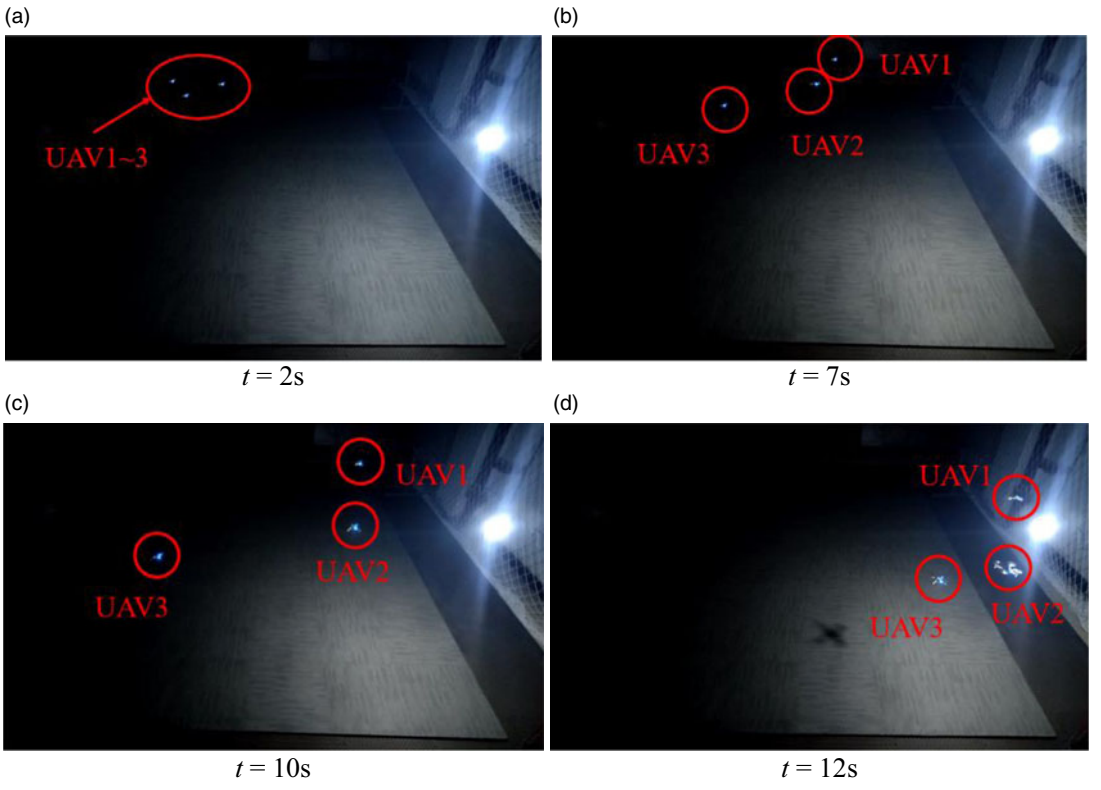


Figure 14. *Formation coverage and source localization of three drones.*

light intensity information obtained from light-sensitive sensors and the position information acquired from Optitrack are used as prior inputs to the algorithm. The distribution of the light source field is reconstructed by the movement of drones within the task area, ultimately pinpointing the location of the light source. The experimental process is illustrated in Figure 14.

It can be seen that in the initial stages, due to the scarcity of light intensity information, the drones may head towards areas with greater uncertainty. However, as the number of available prior sample points increases, the predicted distribution of the field becomes more accurate. At this point, the drones tend to move towards areas where high density and uncertainty are balanced. Ultimately, the drone formation converges, achieving multiple drones' online coverage and the localization of the light source in the light source environment. Due to the limitation of the scale of the experiment site, the UAV only needs a few iterations to reach the pollution source location, so the total iteration time in the experiment is relatively small. Whether the experimental scenario is simple or complex, the control principle is the same, and eventually the coverage and pollution source will be located.

6. Conclusion

This paper addresses the problem of online coverage and source localization for multiple robots, combining the VS-GPR method with an improved CVT algorithm to achieve efficient coverage and pollution source localization in uncertain environments. The algorithm first considers the potential collisions between robots by adjusting the size of the buffer factor to ensure safety between them and utilizes SVM methods to avoid collisions with obstacles. Secondly, it reduces the time consumption of model training by using the VS-GPR method, effectively improving the exploration and coverage efficiency of multiple robots in unknown field sources while ensuring model accuracy. Subsequent simulations and

https://doi.org/10.1017/S0263574725000050 Published online by Cambridge University Press

experiments have confirmed the effectiveness of the proposed algorithm. The characteristics of the leak sources studied in this paper follow a Gaussian distribution, and future research will focus on smoke plume diffusion environments and heterogeneous multi-robot formation studies based on this research.

Author contributions. K.C. and S.G. conceived and designed the study. Y.-B.W. and K.C. conducted code implementation and drone experiments. Y.-Q.C. guided this study. Y.-B.W., K.C., S.-B.Y., and K.Y. wrote this article.

Financial support. This work was financially supported in part by National Natural Science Foundation of China Youth Fund (No.:62103315, 62303368), Ministry of Education Key Laboratory of Information Fusion Technology Open Fund Project (No. 202312-IFTKFKT-007), Shaanxi Province Technical Innovation Guidance Project (No.: 2022QFY01-16), Shaanxi Province Key Industry Innovation Chain Project (No. 2023-ZDLNY-61).

Competing interests. The authors declare no conflicts of interest exist.

Ethical approval. Not applicable.

References

[1] G. Wang, W. Wang, P. Ding, Y. Liu, H. Wang, Z. Fan, H. Bai, Z. Hongbiao and Z. Du, "Development of a search and rescue robot system for the underground building environment," *J. Field Robot.* **40**(3), 655–683 (2023).
[2] M. E. Simon, F. L. Baldissera, M. H. deQueiroz and G. Cabral, "Multi-robots coordination system for urban search and rescue assistance based on supervisory control theory," *J. Control Autom. Electr. Syst.* **40**(3), 484–495 (2023).
[3] Y. Du, "Multi-uav search and rescue with enhanced a algorithm path planning in 3d environment," *Int. J. Aerosp. Eng.* **2023**, 1–18 (2023).
[4] X. Song, Y. Cheng, Q. Qi and X. Zou. "A Path Planning Method for Robot-aided Aero-engine Ffeet Inspection Considering Resource Reuse Strategy." **In:** *IEEE International Conference on Industrial Technology (ICIT)* (2022) pp. 1–6.
[5] G. Best and G. A. Hollinger. "Decentralised Self-organising Maps for Multi-robot Information Gathering." **In:** *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2020) pp. 4790–4797.
[6] T. Zhang, V. Qin, Y. Tang and N. Li. "Source Seeking by Dynamic Source Location Estimation." **In:** *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2021) pp. 2598–2605.
[7] T. Wiedemann, C. Vlaicu, J. Josifovski and A. Viseras, "Robotic information gathering with reinforcement learning assisted by domain knowledge: An application to gas source localization," *IEEE Access.* **9**, 13159–13172 (2021).
[8] D. Gundana and H. Kress-Gazit, "Event-based signal temporal logic synthesis for single and multirobot tasks," *IEEE Robot. Autom. Lett.* **6**(2), 3687–3694 (2021).
[9] J. P. Queralta, J. Taipalmaa, B. C. Pullinen, V. K. Sarker, T. N. Gia and H. Tenhunen, "Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision," *IEEE Access.* **8**, 191617–191643 (2020).
[10] J. P. Queralta, Q. Li, F. Schiano and T. Westerlund. "VIO-UWB-based Collaborative Localization and Dense Scene Reconstruction within Heterogeneous Multi-robot Systems." **In:** *International Conference on Advanced Robotics and Mechatronics (ICARM)* (2022) pp. 87–94.
[11] M. Zhou, Z. Wang, J. Wang and Z. Dong, "A hybrid path planning and formation control strategy of multi-robots in a dynamic environment," *J. Adv. Comput. Intell. Intell. Inform.* **26**(3), 342–354 (2022).
[12] X. An, C. Wu, Y. Lin, M. Lin, T. Yoshinaga and Y. Ji, "Multi-robot systems and cooperative object transport: Communications, platforms, and challenges," *IEEE Open J. Comput. Soc.* **4**, 23–36 (2023).
[13] J. Orr and A. Dutta, "Multi-agent deep reinforcement learning for multi-robot applications: A survey," *Sensors* **23**(7), 3625 (2023).
[14] A. Renzaglia and L. Briñón-Arranz, "Search and localization of a weak source with a multi-robot formation," *J. Intell. Robot. Syst.* **97**(3), 623–634 (2020).
[15] F. Mascarich, M. Kulkarni, P. D. Petris, T. Wilson and K. Alexis, "Autonomous mapping and spectroscopic analysis of distributed radiation fields using aerial robots," *Auton. Robot.* **47**(2), 139–160 (2023).
[16] X. Chen, B. Yang, J. Huang, Y. Leng and C. Fu, "A reinforcement learning fuzzy system for continuous control in robotic odor plume tracking," *Robotica* **41**(3), 1039–1054 (2023).
[17] T. A. Karagüzel, A. E. Turgut, A. E. Eiben and E. Ferrante, "Collective gradient perception with a ffying robot swarm," *Swarm Intell.* **17**(1), 117–146 (2023).
[18] M. Jiang, Y. Liao, X. Guo, H. Cai, W. Jiang, Z. Yang, F. Li and F. Liu, "A comparative experimental study of two multi-robot olfaction methods: Towards locating time-varying indoor pollutant sources," *Build. Environ.* **207**, 108560 (2022).
[19] L. Chang, L. Shan, W. Zhang and Y. Dai, "Hierarchical multi-robot navigation and formation in unknown environments via deep reinforcement learning and distributed optimization," *Robot. Comput. Integr. Manuf.* **83**, 102570 (2023).
[20] H. Jang, M. Park and H. Oh, "Improved Socialtaxis for information-theoretic source search using cooperative multiple agents in turbulent environments," *Expert Syst. Appl.* **225**, 120033 (2023).

https://doi.org/10.1017/S0263574725000050 Published online by Cambridge University Press

[21] L. Chen, C. Zhou, Y. Wang, Y. Zong, T. Lu and C. Chen, "Autonomous search investigation for radioactive leaked source based on an updated infotaxis method during nuclear emergency rescue," *Nucl. Eng. Des.* **416**, 112769 (2024).
[22] S. An, M. Park and H. Oh, "Receding-horizon RRT-Infotaxis for autonomous source search in urban environments," *Aerosp. Sci. Technol.* **120**, 107276 (2022).
[23] H. Ma, T. Zhang, Y. Wu, F. P. Calmon and N. Li. "Gaussian Max-Value Entropy Search for Multiagent Bayesian Optimization." **In:** *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2023) pp. 10028–10035.
[24] C. E. Denniston, O. Peltzer, J. Ott, S. Moon, S. K. Kim, G. S. Sukhatme and M. J. Kochenderfer. "Fast and Scalable Signal Inference for Active Robotic Source Seeking." **In:** *IEEE International Conference on Robotics and Automation (ICRA)* (2023) pp. 7909–7915.
[25] A. A. R. Newaz, M. Alsayegh, T. Alam and L. Bobadilla, "Decentralized multi-robot information gathering from unknown spatial fields," *IEEE Robot. Autom. Lett.* **8**(5), 3070–3077 (2023).
[26] N. M. Yungaicela-Naula, Y. Zhang, L. E. Garza-Castañon and L. I. Minchala. "UAV-based Air Pollutant Source Localization Using Gradient and Probabilistic Methods." **In:** *International conference on unmanned aircraft systems (ICUAS)* (2018) pp. 702–707.
[27] H. Li, J. Yuan and H. Yuan, "An active olfaction approach using deep reinforcement learning for indoor attenuation odor source localization," *IEEE Sens. J.* **24**(9), 14561–14572 (2024).
[28] J. Hu, H. Niu, J. Carrasco, B. Lennox and F. Arvin, "Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning," *IEEE Trans. Veh. Technol.* **69**(12), 14413–14423 (2020).
[29] Q. Feng, H. Cai, Y. Yang, J. Xu, M. Jiang, F. Li, X. Li and C. Yan, "An experimental and numerical study on a multi-robot source localization method independent of airffow information in dynamic indoor environments," *Sustain. Cities Soc.* **69**, 101897 (2020).
[30] W. Luo, C. Nam, G. Kantor and K. Sycara. "Distributed Environmental Modeling and Adaptive Sampling for Multi-robot Sensor Coverage." **In:** *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems* (2019) pp. 1488–1496.
[31] Z. Zhen, Y. Chen, L. Wen and B. Han, "An intelligent cooperative mission planning scheme of UAV swarm in uncertain dynamic environment," *Aerosp. Sci. Technol.* **100**, 105826 (2020).
[32] Z. Wu, J. Dai, B. Jiang and H. R. Karimi, "Robot path planning based on artiffcial potential field with deterministic annealing," *ISA Trans.* **138**, 74–87 (2023).
[33] Y. Gao, C. Bai, R. Fu and Q. Quan, "A non-potential orthogonal vector field method for more efffcient robot navigation and control," *Robot. Auton. Syst.* **159**, 104291 (2023).
[34] X. Zhong, J. Tian, H. Hu and X. Peng, "Hybrid path planning based on safe A* algorithm and adaptive window approach for mobile robot in large-scale dynamic environment," *J Intell. Robot. Syst.* **99**(1), 65–77 (2020).
[35] E. García, J. R. Villar, Q. Tan, J. Sedano and C. Chira, "An efffcient multi-robot path planning solution using A* and coevolutionary algorithms," *Integr. Comput.-Aided Eng.* **30**(1), 41–52 (2023).
[36] M. Kiadi, E. García, J. R. Villar and Q. Tan, "A*-based co-evolutionary approach for multi-robot path planning with collision avoidance," *Cybern. Syst.* **54**(3), 339–354 (2023).
[37] Y. Wang, Z. He, D. Cao, L. Ma, K. Li, L. Jia and Y. Cui, "Coverage path planning for kiwifruit picking robots based on deep reinforcement learning," *Comput. Electron. Agric.* **205**, 107593 (2023).
[38] L. Chen, Y. Wang, Z. Miao, Y. Mo, M. Feng, Z. Zhou and T Wang, "Transformer-based imitative reinforcement learning for multi-robot path planning," *IEEE Trans. Ind. Inf.* **19**(10), 10233–10243 (2023).
[39] J. Tan, "A method to plan the path of a robot utilizing deep reinforcement learning and multiSensory information fusion," *Appl. Artif. Intell.* **37**(1), 2224996 (2023).
[40] L. Zhu, J. Cheng, H. Zhang, W. Zhang and Y. Liu, "Multi-robot environmental coverage with a two-stage coordination strategy via deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.* **25**(6), 5022–5033 (2024).
[41] Y. Chen, M. Guo and Z. Li, "Deadlock resolution and recursive feasibility in MPC-based multirobot trajectory generation," *IEEE Trans. Autom. Control* **69**(9), 6058–6073 (2024).
[42] K. Cao, Y. Chen, S. Gao, H. Dang and D. An, "Distributed weighted coverage for multi-robot systems in non-convex environment," *Appl. Sci.* **13**(14), 8530 (2023).
[43] X. Wang, L. L. Xi, Y. Ding and B. M. Chen, "Distributed encirclement and capture of multiple pursuers with collision avoidance," *IEEE Trans. Ind. Electron.* **71**(7), 7520–7530 (2023).
[44] R. Urban, M. ŠTRONER and I. Kuric, "The use of onboard UAV GNSS navigation data for area and volume calculation," *Acta Montan Slovaca* **25**(3), 364–371 (2020).
[45] J. Hu, H. Zhang, S. Sfarra, E. Pivarčiová, Y. Yao, Y. X. Duan, C. Ibarra-Castanedo, G. Y. Tian and X. Maldague, "Autonomous dynamic line-scan continuous-wave terahertz non-destructive inspection system combined with unsupervised exposure fusion," *NDT & E Int.* **132**, 102705 (2022).

Cite this article: K. Cao, Y. Wei, Y. Chen, S. Gao, K. Yan and S. Yang (2025). "Multi-robot area coverage and source localization method based on variational sparse Gaussian process", Robotica **43**, 2100–2120. https://doi.org/10.1017/S0263574725000050

https://doi.org/10.1017/S0263574725000050 Published online by Cambridge University Press