Measurement 154 (2020) 107482

Contents lists available at ScienceDirect

Measurement

journal homepage: www.elsevier.com/locate/measurement

Combining particle filter algorithm with bio-inspired anemotaxis behavior: A smoke plume tracking method and its robotic experiment validation

Xinxing Chen [a], Jian Huang [a,*]

[a] Key Laboratory of Ministry of Education for Image Processing and Intelligent Control, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan, China

ARTICLE INFO

Article history:
Received 9 October 2019
Received in revised form 6 December 2019
Accepted 28 December 2019
Available online 8 January 2020

Keywords:
Olfactory robot
Particulate matter
Firefly algorithm
Particle filter
Entropy

ABSTRACT

The smoke source localization robot is a safe substitution of human and animal rescuers in many dangerous search and rescue mission. In this paper, we propose an anemotaxis – particle filter based smoke plume path tracking method and presented a smoke source localization robot. A modified firefly algorithm is applied in the resampling step of particle filter based smoke plume tracking method to mimic anemotaxis behaviors in the odor source searching process of living creatures. The performance of the proposed algorithm is qualitatively evaluated in a simulated wind tunnel environment in Webots, a high-fidelity robotic simulator. The average computation time including plume path estimation, resampling and target evaluation in every step is 35.9974 ms and the average smoke source localization error is 0.7769 m. We also verify the feasibility by running the proposed algorithm on the smoke source localization robot in real experiments.

1. Introduction

In the past decades, robotic odor source localization has been widely demanded and became a challenging engineering problem. The robots involved in these source localization tasks are called olfactory robots. The olfactory robot is a safe substitution of human and animal rescuers in some search and rescue missions (e.g. in a building fire, in an unexploded mine, etc) and requires much less specialized training. Compared with static sensor networks, using mobile robots can significantly reducing sensing nodes and improve estimation accuracy.

Normally, the workflow of olfactory robots includes three stages [1]:

- Plume searching. If the robot starts somewhere far from odor plumes, it has to conduct random walks to capture higher concentration.
- Plume tracking. When the robot is in the plumes, it will follow some strategies to trace the plume upwind.
- Source declaration. When the robot find the odor source (e.g. when it measures an odor concentration higher than a preset threshold, or it detects the odor source through its camera), it can declare the location of the odor source.

Apart from developing dedicated robotic platforms, researchers also paid much effort on odor source localization algorithms. Most of the previous works focused on the second stage, plume tracking. From early 1990s, various odor source localization algorithms have been proposed, including concentration gradient climbing algorithms [2], bio-inspired algorithms [3–5], probabilistic-based algorithms [6–8], and multi-robot algorithms [9].

Due to the fact that odor plumes are dispersed by flowing air, the distribution of plumes is patchy. There is not a smooth concentration gradient along the airflow direction. Therefore, concentration gradient climbing algorithms doesn't work well. To solve this issue, bio-inspired algorithms are proposed to introduce some living creature's behavior patterns such as upwind surge and spiraling. The performance of these reactive algorithms heavily depends on parameters related to the robot's movement, which have to be carefully tuned. Multi-robot algorithms show better efficiency when robots collaborate, so it became a popular topic these years. But the cost is much higher because these algorithms depend on much more robot nodes than others. Thanks to the rapid development of hardware's computing power,

* Corresponding author.
E-mail address: huang_jan@mail.hust.edu.cn (J. Huang).

https://doi.org/10.1016/j.measurement.2020.107482
0263-2241/

probabilistic-based algorithms also received attention in recent years. Probabilistic-based algorithms regard the odor source location as a probability distribution. Through observing and updating the probability distribution iteratively, the robot will acquire increasing knowledge of the environment and move towards to the odor source. However, many probabilistic-based algorithms rely heavily on accurate global localization of the robot and prior knowledge about the size of the scenario [6,10], which is not always applicable in real rescue scenarios.

In term of sensing modalities, the previous works in the field of olfactory robot usually focused on chemical release source localization. Thus, they used volatile organic compounds (VOC) sensors as their sensing modality to measure the concentration of ethanol, acetone, etc. Due to the fact that chemical plume dispersion rules were widely studied [11] and that VOC sensors were cheap, of quick response, and easy to deploy on robots, researchers didn't consider much about other possible sensing modalities in demand. However, in some cases, such as fires, the burning source release airborne smoke particles instead of chemical substances which can be measured by VOC sensors. To adapt the above olfactory robot solutions to smoke source localization scenario, one challenge is to build a proper smoke propagation model, which can be rarely found in literature, the other is to essentially change the sensing modality and to adapt algorithms consequently. Research on smoke source localization mainly focus on static sensing nodes [12–14], robotic solutions can be hardly seen because most smoke sensors are of large size and not portable. Small-sized portable smoke particle sensors are available in the market recently. [15] proposed a particle filter robotic solution for smoke source localization. The particles used in the framework of particle filter was set to 1024, which was a large value and led to relatively heavy computation load. Also, the robot tended to lose its way when the robot fails to obtain enough information for a few successive steps.

In this work, in order to solve the above issues, we combined particle filter based algorithm with bio-inspired anemotaxis behavior, which means conducting upwind movement when capturing an odor plume, and was a common behavior when living creatures searched their food and mates. The anemotaxis behavior was introduced by implementing a modified firefly algorithm in the resampling stage of the particle filter. The firefly algorithm was modified by adding a random stochastic term to the position of the particles, which moves the particles towards the upwind direction at every step. Benefit from the fast convergence feature of firefly algorithm, the proposed algorithm worked fine with a smaller amount of particles. We also built a smoke source localization robot equipped with a small-sized particulate matter sensor and a wind direction sensor to sample the environment.

Main contributions of this paper are summarized below.

(1) A smoke plume propagation model was proposed. The sensing modality was substantially changed – we used a particulate matter sensor (Plantower PMS7003) to measure the concentration of airborne smoke particles. A smoke source localization robot working with the Plantower sensor was built.
(2) Considering the fact that the dispersion of smoke particles are affected by gravity, and that the particulate matter sensor is of multiple channels, we proposed a data preprocessing strategy to fuse the measurement data from all channels.
(3) An anemotaxis – particle filter based smoke plume tracking method was proposed to overcome the issues of particle filter based method (relatively heavy computation load and robot losing its way). The feasibility of the proposed method was tested in simulation and in robotic experiments in a real scenario with a space-variant wind field.

In Section 2 of this paper, we present a smoke source localization robot and its sensing modalities. Section 3 introduces the propagation rule of smoke plumes and a plume path model we used in the proposed method. In Section 4, we present the measurement data fusion strategy and the proposed smoke plume tracking method. In Section 5, we show a simulation scenario in Webots and performance evaluation results of the method. In Section 6, we also demonstrate the feasibility of the proposed algorithm through real experiments on the robot.

2. Smoke source localization robot platform

Since our proposed algorithm heavily relies on particular sensors of the smoke source localization robot we built, the structure of the robot is firstly shown in this section before presenting the algorithm.

The robot (shown in Fig. 1) was built based on Turtlebot 3, a differential wheel robot platform. The wheels are actuated by the OpenCR control board. A Raspberry Pi is connected to OpenCR to implement the proposed algorithm in this paper. The Raspberry Pi is selected because of its light weight and competent computation power for the smoke plume tracking task. Highly parallel hardware architectures [16] may be used for more complex tasks in our future work. Since the SLAM technique was not used in our algorithm, the lidar on the top of Turtlebot 3 Burger was replaced with Gill WindSonic, an ultrasonic wind sensor, which can measure wind velocity and direction at 4 Hz. The Gill WindSonic sensor was mechanically anchored on the top of the robot through a 3D printed holder. We also mounted a small-sized and light-weighted portable particulate matter sensor, Plantower PMS 7003 on the robot, with its air inlet aligned with the robot's heading. It has 12 measurement channels, shown in Table 1. The sensor can take measurement from all 12 channels once per second.

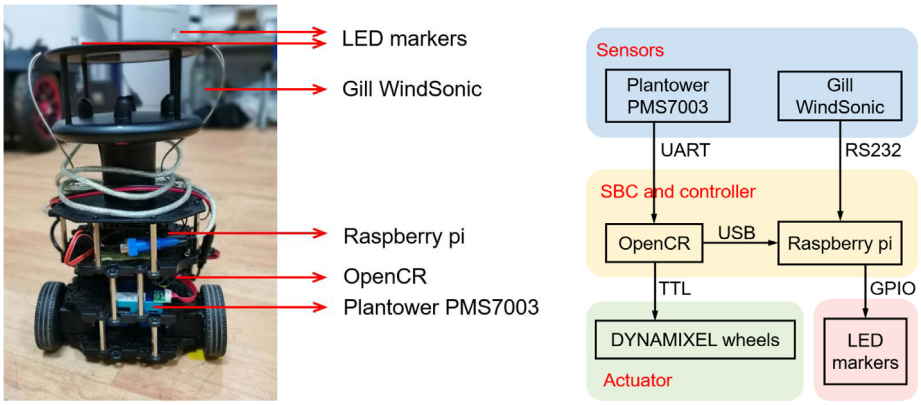
Since channel 1–3 is not relative to our application scenario, and the measurement modalities of channel 4–6 can be substituted by channel 9, 10, 12, only the measurement data of channel 7–12 was taken as the smoke particle concentration inputs in the proposed algorithm. Their abbreviations are used to represent the measurement value of the corresponding channels in this paper.

3. Propagation of smoke plumes

Inspired by research papers [11,17], we proposed a smoke plume path model based on the propagation rule of smoke particles in a uniform wind field:

$$P(x,y,z) = \frac{Q}{2\pi\sigma_y\sigma_z} exp^{-\frac{y^2}{2\sigma_y^2}-\frac{z^2}{2\sigma_z^2}} \quad (1)$$

In this model, (x,y,z) stands for a position in space. x axis is aligned with the wind direction, y and z axes are horizontal and vertical crosswind direction respectively. $P(x,y,z)$ represents the probability that a smoke particle released from the smoke plume at $(0,0,0)$ travels to (x,y,z). Q is a constant related to the Reynolds number of airflow and the density of the smoke particle and air. σ_y and σ_z are the variance of the random process of the smoke particle moving in y and z direction, and they can be fitted into linear functions of x according to [7]. Since the nature of smoke plumes propagation and the random walk of smoke particles don't change



(a) The photo of the smoke source localization robot

(b) The structure diagram of the robot

Fig. 1. The smoke source localization robot and its structure diagram.

Table 1
Measurement channels of Plantower PMS7003.

No.	Measurement value	Abbreviation
1	PM 1 concentration ($CF = 1$)	c_1_CF1
2	PM 2.5 concentration ($CF = 1$)	$c_{2.5}\_CF1$
3	PM 10 concentration ($CF = 1$)	c_{10}_CF1
4	PM 1 concentration (in atmospheric environment)	c_1_A
5	PM 2.5 concentration (in atmospheric environment)	$c_{2.5}\_A$
6	PM 10 concentration (in atmospheric environment)	c_{10}_A
7	The number of particles ($d > 0.3\mu m$) in 0.1 l of air	$n_{0.3}$
8	The number of particles ($d > 0.5\mu m$) in 0.1 l of air	$n_{0.5}$
9	The number of particles ($d > 1\mu m$) in 0.1 l of air	n_1
10	The number of particles ($d > 2.5\mu m$) in 0.1 l of air	$n_{2.5}$
11	The number of particles ($d > 5\mu m$) in 0.1 l of air	n_5
12	The number of particles ($d > 10\mu m$) in 0.1 l of air	n_{10}

$*CF$ means 'calibration factor'. $CF = 1$ represents a specific industrial environment.
$**d$ means 'diameter'.

essentially from 3D space to 2D plane, we formulate the smoke plume path propagation in the floor plane ($z = 0$), the above equation can be expressed as:

$$P(x,y) = \frac{Q}{2\pi\sigma_y\sigma_z} exp^{-\frac{y^2}{2\sigma_y^2}} \tag{2}$$

According to the equation in Eq. (2), we built a smoke plume path model (shown in Fig. 2). Each smoke plume in the environment can be regarded as a release source of smoke particles.

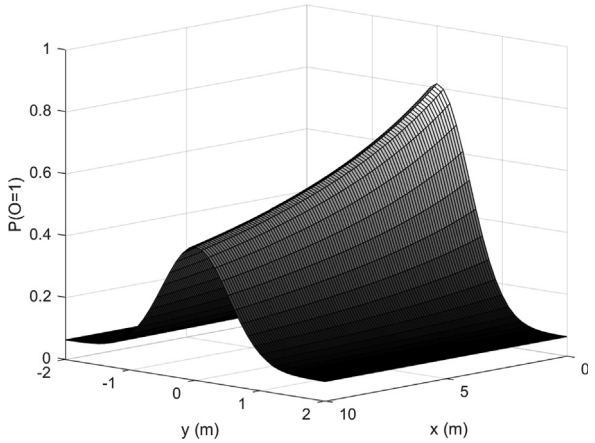


Fig. 2. Plume path model used for the particle filter-based algorithm.

$P(O = 1)$ stands for the probability that a smoke particle can be "observed" at position (x, y) assuming a smoke plume is located at $(0, 0)$. A small constant was added to the model simulating measurement errors caused by the particulate matter sensor.

Noting that the smoke plume path model was built without fitting its parameters carefully in a certain environment as [7] did to develop a chemical source localization algorithm, it can be used as a general model for the following proposed method to run in different environment settings.

4. Proposed algorithm

The particle filter is a widely-used localization algorithm. It uses virtual particles to represent posterior probability distribution. In this work, we combine particle filter with bio-inspired anemotaxis behavior for robotic smoke plume tracking. The smoke plume path estimation process was augmented with an active perception navigation strategy to maximum the obtained information at every step. In this section, we will show how the proposed method works.

In the framework of anemotaxis – particle filter based smoke plume tracking method, a set of particles (in this paper, the number of particles N in the particle set is 300) were utilized to represent the probability distribution of smoke plumes path. In the particle set, each has a weight w_t^i and a location $\mathbf{X}_t^i = (x_t^i, y_t^i)$ at time step t. It is assumed that the robot has no information about the map of the scenario and does not have access to global self localization, which is realistic for unknown environments without a predefined map and availability of a global positioning system. The robot creates a fixed-size perception window W around itself, illustrated in Fig. 6b. In this paper, W is a square of $4m \times 4m$ size. The robot can only estimate the plume path within W. W moves with the robot.

The pseudocode of the proposed algorithm is shown in Algorithm 1.

Algorithm 1: Anemotaxis – Particle Filter based smoke plume tracking

1: $\left\{\mathbf{X}_0^i, w_0^i\right\}_{i=1:N} \sim p_0(.)$ ▷Initialization
2: $t \leftarrow 1$
3: **repeat** ▷Iteration
4: Get the binary observation value O_t and the wind direction θ_t

(*continued on next page*)

$\left\{\mathbf{X}_t^i, w_t^i\right\}_{i=1:N} \leftarrow$ An-PF$(\mathbf{X}_{t-1}^i, w_{t-1}^i, O_t, \theta_t)$
Evaluate all potential targets
Navigate to the target yielding minimum expected entropy
until The robot reaches the smoke source
procedure An-PF$(\mathbf{X}_{t-1}^i, w_{t-1}^i, O_t, \theta_t)$
  for $i = 1 : N$ **do**
    $w_t^i \leftarrow P(O_t|i) \cdot w_{t-1}^i$
  end for
  for $i = 1 : N$ **do**
    for $j = 1 : N$ **do**
      if $w_t^j > w_t^i$ **then**
        $\mathbf{X}_t^i$ is updated by Eq. 8 and Eq. 10
      end if
    end for j
  end for i
  Adjust $\left\{\mathbf{X}_{t-1}^i\right\}_{i=1:N}$ to ensure all particles stay within W
  Rescale $\{w_t^i\}_{i=1:N}$ into a proper range
  Redistribute a few particles to random positions in W
  return $\left\{\mathbf{X}_t^i, w_t^i\right\}_{i=1:N}$
end procedure

In the initialization step before the robot takes the first observation, it has no idea about the wind direction and the ambient smoke concentration. The weights of all the particles are equal. All the particles are randomly distributed in W and of the same weight, which reflects that every position in the perception window is of the equal probability to be a smoke plume patch.

The robot will iteratively take an observation, estimate smoke plume path, resampling particles, evaluate target for the next step and move until it reaches the smoke source.

The detailed process of the iteration are shown in Fig. 3 and it is explained step by step here.

4.1. Observation

In the observation step, the robot senses its ambient environment through the particulate matter sensor and the wind sensor.

Since we have six measurement values from the particulate matter sensor, we have to either choose one single channel or fuse them together. [18,19] discussed some widely used data preprocessing strategies. Here we formulate two strategies to dealing with the raw measurement values from 6 channels to get the input concentration value C in Table 2.

In the formula of weighted modalities, a_d represents the weight assigned to each channel. The cumulative surface area of all the particulate matters (shown in Eq. (3)) is just in the form of weighted modalities. Considering the fact that large particles travel more locally, and that small ones travel further [17], larger weight should be assigned to the number of larger particles, which is also satisfied in Eq. 3. So in this paper, we use Eq. (3) as the formula of weighted modalities.

$$S = \sum d^2 n_d \quad (d = 0.3, 0.5, 1, 2.5, 5, \text{ and } 10) \tag{3}$$

The input concentration value has to be binarized to represent the robot's state of being in $(O = 1)$ or outside $(O = 0)$ the plumes. Binarization [20,21] is an important step when processing signals, which can extract the useful data. In [6], the author set a fixed threshold to binarized the measurement value. However, the fixed threshold requires prior knowledge of the concentration range in the scenario and is not general in all environment settings. In this paper, we are going to compare two binarization method. One is moving average method (Eq. 4 and 5, MA in abbreviation). C_t and O_t are the input concentration value and the binary observation value at time step t. According to [10], the choice of λ in the range of $0.1 \sim 0.6$ does not affect the performance of MA significantly. So λ is set to 0.5 in this paper. The other one is adaptive threshold method (AT in abbreviation) we proposed in [15]: O_t is a random binary number generated by the distribution in Eq. 6, in which T_1 and T_2 are the maximum history measurement value and the minimum history measurement value respectively.

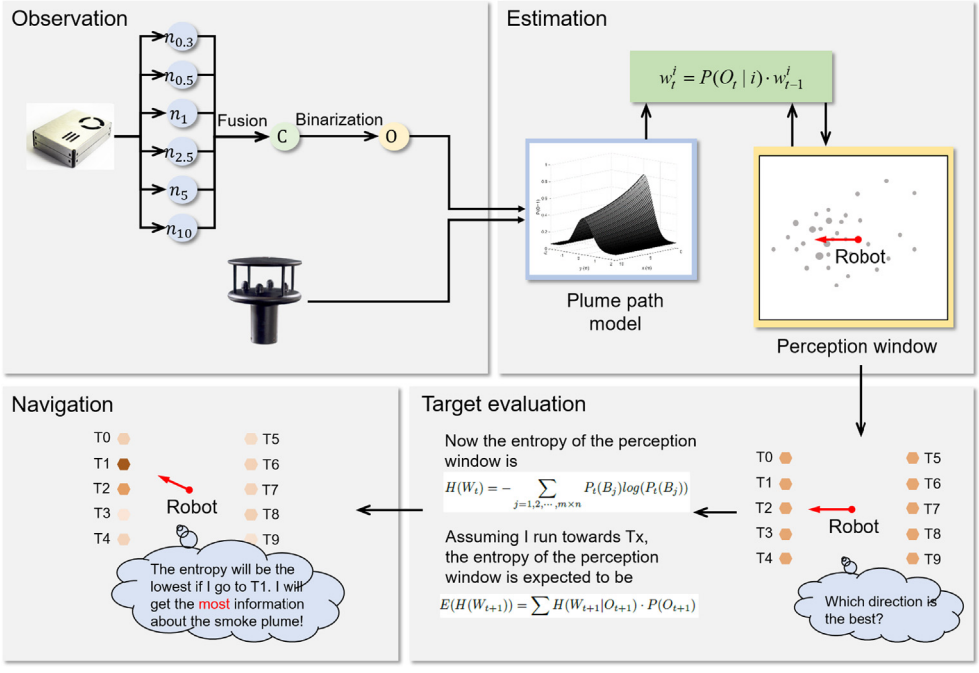


Fig. 3. The process of the iteration in the proposed method.

Table 2
Preprocessing strategies of raw measurement values from the particulate sensor.

Strategy	Formula
Single modality	$C = n_d \quad (d = 0.3, 0.5, 1, 2.5, 5, or 10)$
Weighted modalities	$C = \sum a_d n_d \quad (d = 0.3, 0.5, 1, 2.5, 5, or 10)$

$$\overline{C}_t = \begin{cases} \lambda \overline{C}_{t-1} + (1-\lambda) C_t & , \quad t \geqslant 1, \\ C_t & , \quad t = 0 \end{cases} \tag{4}$$

$$O_t = (C_t - \overline{C}_{t-1} > 0) \tag{5}$$

$$P(O_t = 1) = \begin{cases} 1 & , \quad C_t > T_1 \\ 0 & , \quad C_t < T_2 \\ \frac{C_t - T_2}{T_1 - T_2} & , \quad otherwise \end{cases} \tag{6}$$

4.2. Smoke plume path estimation

In the small perception window around the robot, the wind field can be regarded uniform. The observation values are put into the plume path model in Section 3 to calculate $P(O_t|i)$, the probability of the robot observing O_t if smoke plumes are released from the location of the ith particle. Then the weight of the ith particle are updated by Bayesian inference:

$$w_t^i = P(O_t|i) \cdot w_{t-1}^i \tag{7}$$

If the weight of the biggest particle is below a predefined threshold after several time steps, the weights of all the particles will be normalized.

4.3. Resampling

In [15], at every step, some particles with the weight smaller than a preset threshold were eliminated and some particles with the weight larger than another preset threshold were split into two. This process is illustrated in Fig. 4. The thresholds had to be tuned to make the algorithm work well.

In this paper, to introduce anemotaxis behavior and to avoid using preset thresholds to identify big and small particles, and to make particles with larger weight impact more on the smoke plume path estimation, a modified firefly algorithm was introduced in the resampling stage.

Following the standard firefly algorithm, all the particles can be regarded as fireflies. The light intensity of each firefly is their particle weight. Fireflies with larger light intensity attract others with smaller light intensity with attractiveness β. β can be calculated with Eq. 8, in which the absorption coefficient γ is set to be 3, β_0 is set to be 1 in this paper, r_{ij} is the distance between firefly i and j. At each step after updating the particles' weights, every firefly i will fly to another firefly j with larger light intensity following the rule in Eq. (9). x and y represent the position of the fireflies, α is the step length factor and is set to be 1 in this paper.

$$\beta = \beta_0 e^{-\gamma r_{ij}^2} \tag{8}$$

$$\begin{aligned} x_{t+1}^i &= x_t^i + \beta\left(x_t^j - x_t^i\right) + \alpha(rand - 0.5), \\ y_{t+1}^i &= y_t^i + \beta\left(y_t^j - y_t^i\right) + \alpha(rand - 0.5) \end{aligned} \tag{9}$$

Inspired by the nature of anemotaxis among living creatures [22–24] when they are searching for odor source, based on the standard firefly algorithm, we made a minor modification on Eq. 9 – we added another random stochastic term to make the fireflies move upwind, shown in Eq. 10, in which α' is set to be 0.3 and ω is set to be 0.45, θ_t is the angle between the robot's heading and the wind direction.

$$\begin{aligned} x_{t+1}^i &= x_t^i + \beta\left(x_t^j - x_t^i\right) + \alpha(rand - 0.5) + \alpha'(rand - \omega)\cos\theta_t, \\ y_{t+1}^i &= y_t^i + \beta\left(y_t^j - y_t^i\right) + \alpha(rand - 0.5) + \alpha'(rand - \omega)\sin\theta_t \end{aligned} \tag{10}$$

This modification gathers more fireflies (particles) on the upwind direction (illustrated in Fig. 5), which assigns higher probability to upwind positions, thus making the robot have higher probability to move upwind when it makes decision.

4.4. Target evaluation and navigation

At time step t, the entropy of the perception window can be calculated the following way: the perception window is divided into a $M \times N$ grid map. Each grid $B_j (j = 1, 2, \cdots, m \times n)$ can be regarding as a "bin" containing particles in W. The probability that there is a smoke plume in B_j can be calculated by:

$$P_t(B_j) = \frac{\sum_{L_t^i \in B_j} w_t^i}{\sum_{L_t^i \in W} w_t^i} \tag{11}$$

The entropy of the perception window at time step t can be expressed as:

$$H(W_t) = -\sum_{j=1,2,\cdots,m\times n} P_t(B_j) log(P_t(B_j)) \tag{12}$$

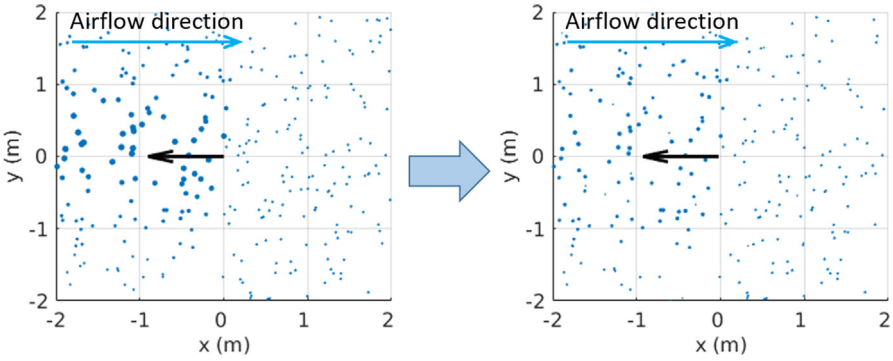


Fig. 4. Resamping step in [15] (The particles are distributed in the perception window. The size of particles represents their weights, the black arrow represents the heading direction of the robot, the blue arrow represents the airflow direction).

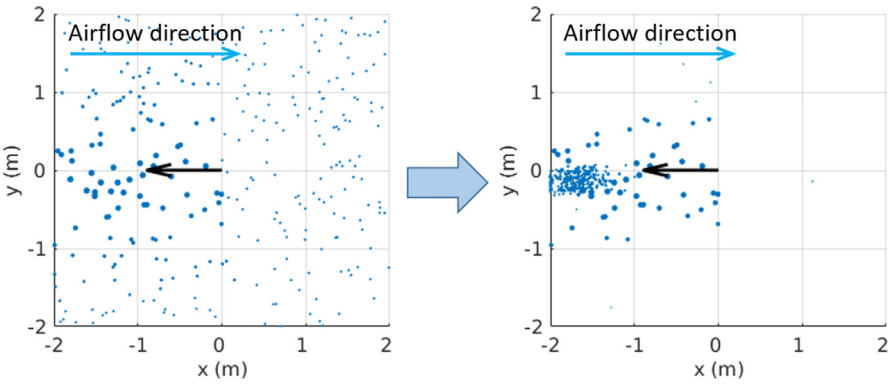


Fig. 5. Resamping step in anemotaxis – particle filter (The particles are distributed in the perception window. The size of particles represents their weights, the black arrow represents the heading direction of the robot, the blue arrow represents the airflow direction).

The lower the entropy is, the more certain the smoke plume path is.

According to the proposed method, the robot will greedily navigate itself to the direction yielding minimum expected entropy. There are several preset potential target positions for the robot's next step (shown in Target Evaluation part in Fig. 3).

If the robot moves to the xth target in the next step, the probability to get an observation O_{t+1} can be calculated by:

$$P(O_{t+1}) = \frac{\sum P(O_{t+1}|i) \cdot w_t^i}{\sum w_t^i} \tag{13}$$

The weight of the particles in the perception window W_{t+1} can be updated by Eq. (7). The entropy of the estimated perception window $H(W_{t+1}|O_{t+1})$ can be calculated by Eq. (12). The expected entropy H yielded by this movement is:

$$E(H(W_{t+1})) = \sum H(W_{t+1}|O_{t+1}) \cdot P(O_{t+1}) \tag{14}$$

which takes all possible O_{t+1} into account.

The robot will evaluate all the potential targets and actively navigate itself to the target yielding the lowest expected entropy, which means it will gain the most information about the smoke plume path through this step.

With iterative observations, estimation, target evaluation, and movements, the robot will trace the plume path and finally reach the smoke source, which is the starting point of all the smoke plumes.

4.5. Termination

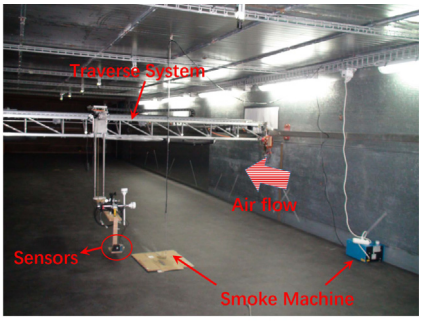
The proposed algorithm aims at guiding the robot to tracing smoke plumes and finally reach the vicinity of the smoke source. Usually, the source declaration stage can be done through other sensing modalities such as vision and laser range finding, like [25,26], or capturing a smoke concentration higher than a preset threshold, which is a more subjectivity task of human handlers. In this work, we assumed that the robot can perceive the existence of the smoke source if it reaches within $0.5m$ of the source. The smoke plume tracking algorithm will therefore terminate right away.

5. Performance evaluation in simulations

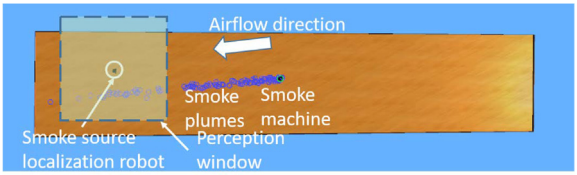
We conducted systematic simulation tests in a high-fidelity robotic simulator, Webots [27]. In this section, the simulation arena, the data sets, and performance evaluation of the proposed method are presented. We also studied the impact of different measurement data preprocessing strategies and compared the anemotaxis behavior combined method with the particle filter based method proposed in [15].

5.1. Simulation arenas and data sets

In our previous work, refined data acquisition was conducted inside a $20m \times 4m$ wind tunnel channel (shown in Fig. 6a) at Distributed Intelligent Systems and Algorithms Laboratory, the Swiss Federal Institute of Technology Lausanne (EPFL). In the wind tunnel, we placed a smoke machine to generate smoke particles. The smoke plumes are dispersed by controlled airflow (in this arena, the airflow velocity was around 1 m/s, and there was a small bias angle between the airflow direction and the long axis of the wind tunnel). We mounted the particulate matter sensor Plantower PMS 7003 on the hand of a traversing system that could move in 3D space. We collected measurement values at 50 (x axis) × 20



(a) DISAL wind tunnel at EPFL



(b) Simulated wind tunnel in Webots (the blue hexagons illustrate how smoke plumes disperse through airflow, the semitransparent square represent the perception window around the robot)

Fig. 6. (a) DISAL wind tunnel at EPFL; (b) Simulated wind tunnel in Webots (the blue hexagons illustrate how smoke plumes disperse through airflow).

(y axis) × 5 (z axis) grid points in the wind tunnel for further research.

In Webots, we built the same arena as the wind tunnel (shown in Fig. 6b). We used the mean and the variance of the measurement value we collected on the floor plane of the wind tunnel to generate random values as simulated smoke concentration. We added Gaussian noise on the airflow direction to simulate its turbulence in reality. In the coordinate system shown in Fig. 6b, the position of the smoke machine was $(10.6, 0.2)$ (in meters, the same below).

5.2. Evaluation metrics

Two metrics are taken into account. One is success rate, which is calculated by the number of successful runs, in which the robot reached within $0.5m$ of the source, divided by the total number of runs (50) for every setting. Since the mobility of the robot is only in 2-D, the distance from the robot to the smoke source was only calculated on the floor plane. The other metric is the traveled distance of the robot per straight distance from the starting point of the robot to the smoke source, which reflects the efficiency of the algorithm. When calculating the traveled distance per straight distance, we only considered successful runs, because otherwise the robot either comes to the boundary or keeps spiraling around and it made no sense to calculate the efficiency of these failure runs.

To speed up the performance evaluation process, the algorithm will end after it iterates 1000 times (one iteration includes taking an observation, updating the distribution of particles, resampling, target evaluation and robot moving one step), in which the robot will usually find the smoke source. The algorithm will also terminate if the robot runs to the boundary of the arena.

5.3. Simulation settings and evaluation results

To compare the impact of two measurement data preprocessing strategies in Table 2 and two binarization methods (MA and AT), we conducted simulations with various combinations. To evaluate the robustness of the proposed algorithm, we also let the robot start the algorithm from different initial positions ($(18, 0)$ on the centerline of plumes and $(18, 0.5)$ far from plumes). The speed of the robot is set to $0.1079m/s$. Since there is no other dedicated probabilistic-based smoke source localization algorithm in literature, the particle filter based method (PF in abbreviation) in [15] is chosen as a benchmark for comparison in this article. Simulations were run to evaluate the performance difference of the anemotaxis – particle filter based method (An-PF in abbreviation) between PF, and the impact of the number of particles. To make the results figures more clear, simulation settings are listed in Table 3. Simulation results are shown in Fig. 7.

From Fig. 7a, it can be seen that under the framework of An-PF with weighted modalities, when the observation is binarized by MA, both metrics are better than AT. The reason may be that AT is more sensitive to outliers – if the history maximum or minimum is an outlier, the raw measurement data from the particulate matter sensor can be hardly binarized properly. Different from AT, MA takes all history measurement data into consideration, which suppresses the impact of outliers. In setting A and B, the robot starts from different initial position, but the statistical data of the two metrics do not vary a lot – the success rate exceed 90% and the traveled distance per straight distance in most runs are below 1.5, which reflects the robustness of the proposed An-PF method with different robot initial position.

From Fig. 7b, it can be seen that when working with An-PF, the performance of the method does not change a lot with the number of particles. But PF doesn't perform well when the number of particles is 300, while the performance is enhanced a lot when the number of particles is 500, which means PF relies heavily on a sufficient number of particle. After introducing the anemotaxis behavior, the particles tend to gather more around the largest particles and upwind direction, which helps the robot navigate to the most possible locations where there are smoke plumes.

Fig. 7c and d show the performance of An-PF when working with six single modalities respectively. The performance varies a lot when the robot starts from different locations – the success rate is not stable and the traveled distance per straight distance is too large when working with some single channels. Since large smoke particles travel more locally than small particles, it is hard for the robot to sense large particulate matters when it starts the algorithm somewhere far away from the smoke source, thus it might lose its way and run into boundaries if the algorithm only takes the concentration of large smoke particles into account. On the other hand, if the algorithm only takes the concentration of small smoke particles into account, it is possible that the whole environment is saturated with small-sized smoke particles, then the binarized observation value makes no sense.

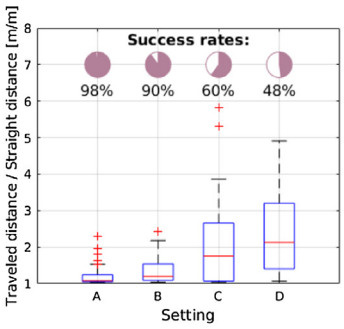
Fig. 7e and f show the trajectories of the robot in all simulation runs under Setting A and B. In most of the runs, the robot goes towards the smoke source in a relative straight way without too many spirals. We have to note that when we were collecting the measurement value data set, we didn't cover areas very close to the smoke machine for preventing the traversing system and the particulate sensor hitting the smoke machine and burning. So in the simulation environment, the robot will sense no smoke concentration when it almost reaches the smoke source, thus do spiraling, which won't happen in the real environment.

The impact of the side length of the perception window on the performance of the proposed algorithm is also studied. The algorithm and the environment is arranged as Setting A in Table 3, except that the side length of the perception window of the robot is set to 2 m, 3 m, 4 m and 5 m, respectively. The evaluation results are shown in Fig. 8.

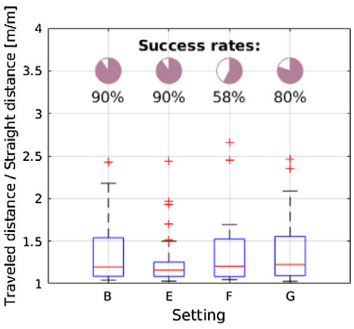
It can be seen that the side length of the perception window does not have obvious impacts on the traveled distance per straight distance, but affects the success rate. Noting that the

Table 3
Simulation settings.

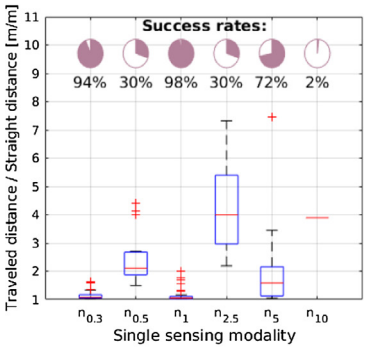
Setting	The robot's initial position	Algorithm	Data preprocessing strategies	Binariazation method	Number of particles
A	$(18, 0)$	An-PF	Weighted modalities	MA	300
B	$(18, 0.5)$	An-PF	Weighted modalities	MA	300
C	$(18, 0)$	An-PF	Weighted modalities	AT	300
D	$(18, 0.5)$	An-PF	Weighted modalities	AT	300
E	$(18, 0.5)$	An-PF	Weighted modalities	MA	500
F	$(18, 0.5)$	PF	Weighted modalities	MA	300
G	$(18, 0.5)$	PF	Weighted modalities	MA	500
H	$(18, 0)$	An-PF	Single modality	MA	300
I	$(18, 0.5)$	An-PF	Single modality	MA	300



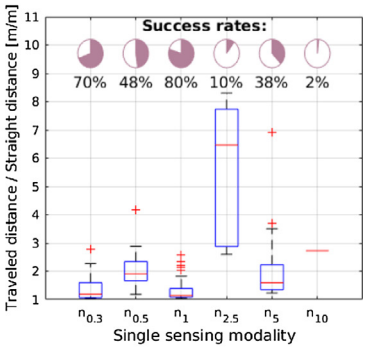
(a) Impact of the robot's initial position and binariazation method



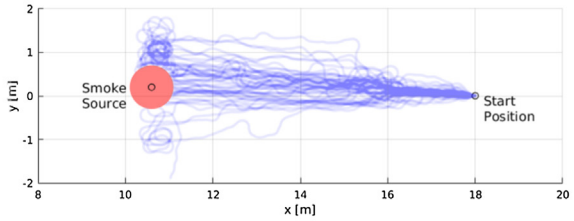
(b) Impact of the anemotaxis behavior and the number of particles



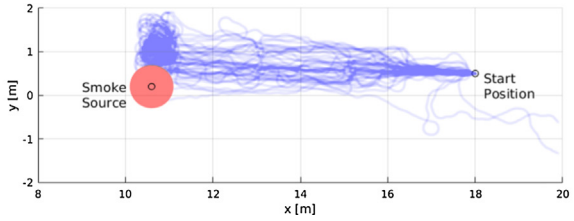
(c) Setting H with different single channels



(d) Setting I with different single channels



(e) The robot's trajectories under setting A (the red patch represents areas within 0.5 m of the source)



(f) The robot's trajectories under setting B

Fig. 7. Simulation results.

parameters of the proposed algorithm is only tuned for $4m \times 4m$ perception window, the success rates of the algorithm with other size of perception window should enhance if proper parameters are selected.

5.4. Computation time and localization error

As two important factors to reflect the computation efficiency and localization accuracy, the computation time and the localization distance error of the proposed algorithm are calculated and compared to two variants of the Kalman Filter algorithm, Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF). These two variants of Kalman Filter algorithm are widely used in target localization [28–31]. The environment and the An-PF algorithm is arranged as Setting A in Table 3. Results are in Table 4.

Because the process of observation and robot movement is the same in three algorithms, the computation time in Table 4 is only the duration from the observation finishing until the robot starting movement. For the An-PF algorithm, the position of the particle with the largest weight when the algorithm terminates is regarded as the estimated source location. All successful and failing runs are calculated.

Since EKF and UKF is of less computation complexity, their computation time is less than An-PF. However, An-PF only lags behind for around 2 ms, which can be ignored compared to the duration of a whole iteration, 225.0169 ms, from observation starting until robot movement finishing. The average localization distance error

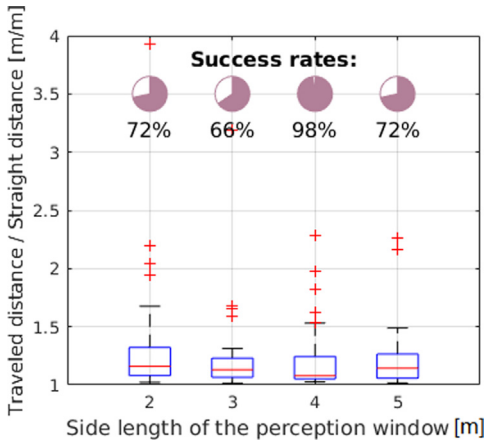


Fig. 8. Setting A with different side length of the perception window.

Table 4
Computation time and localization error.

Algorithm	Average computation time per step (ms)	Average localization distance error (m)
An-PF	35.9974	0.7769
EKF	33.7798	4.0766
UKF	33.2571	4.3289

of An-PF is much smaller than those of EKF and UKF, and is acceptable for the $20m \times 4$ m arena.

The impacts of the radius of the termination circle (the distance between the robot and the smoke source, within which the algorithm will terminate) on the smoke source tracking duration is also studied. The environment and the An-PF algorithm is arranged as Setting A in Table 3. The radius of the termination circle is set to 0.5 m, 0.6 m, 0.7 m, 0.8 m and 0.9 m, respectively. Since the computation time of every step is almost the same, searching duration can be reflected by the number of tracking steps until the robot find the smoke source. The results are shown in Fig. 9.

It can be seen that the number of steps is linear related to the radius of the termination circle. The larger the termination circle is, the faster the algorithm stops.

5.5. Discussion

We proposed a simple but efficient sensor raw data preprocessing strategy which takes into account the dispersion characters of smoke plumes – the weighted sum of measurement values from 6 channels of the particulate matter sensor was used as the smoke concentration input, instead of the measurement value from any single channel. Higher weights are given to channels measuring larger particle

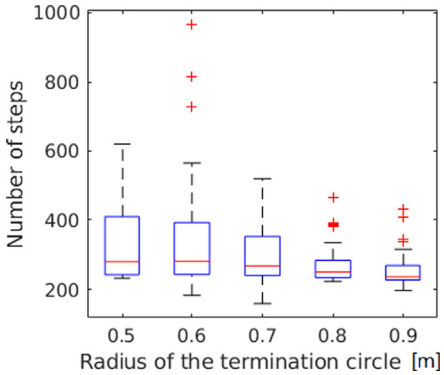


Fig. 9. Radius of the termination circle vs. number of steps.

because large particles travel more locally and stay near the smoke source; lower weight weights are given to those measuring small particles to ensure that the robot can still capture some clues about smoke plumes when it is far away from the smoke source.

The proposed anemotaxis – particle filter based smoke plume tracking method was quantitatively evaluated in the wind tunnel simulation scenario in Webots. All the sensors' measurement values in the simulation were randomly generated according to our data sets collected through a refined data scan in the wind tunnel. When working with weighted modalities, the proposed algorithm showed much more robustness compared to working with any single channel input. In terms of binarization strategies, when the proposed algorithm worked with MA, the success rate could reach no less than 90%, but it didn't perform well with the AT method. From the metric of traveled distance per straight distance, as well as the robot's trajectories, we can see the proposed An-PF algorithm with weighted modalities and MA is of good efficiency – Fig. 7a and Fig. 7b show the value of traveled distance per straight distance in most runs in Setting A, B and E are less than 1.5, some are close to 1, which means the robot doesn't wander too much.

6. Experiments

To demonstrate the feasibility of the proposed algorithm in reality. We conducted experiments on the smoke source localization robot in a laboratory. The algorithm was configured as An-PF + weighted modalities + MA + 300 particles. All other parameters kept the same as in simulations.

6.1. Experiment setup

We arrange our laboratory as Fig. 10. On the ceiling of the laboratory, we mounted a camera in combination with the SwisTrack software [32] to capture two LED markers on the top of the robot. It

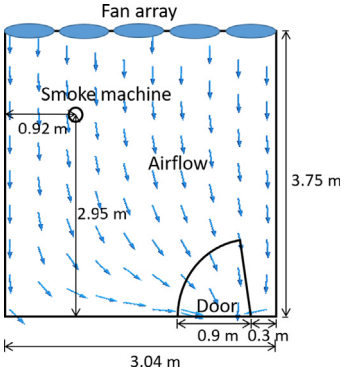


Fig. 10. The arrangement of the laboratory (the blue arrows represent the airflow directions calculated by OpenFOAM).

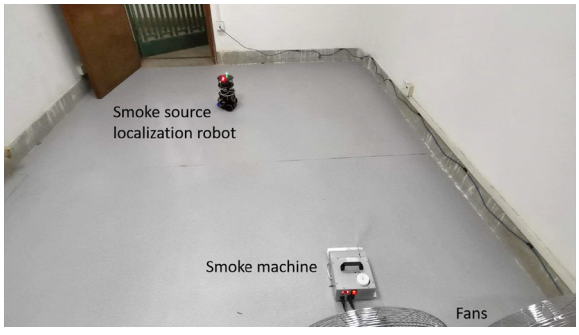


Fig. 11. A snapshot of an experiment run.

Table 5
Experiment results of An-PF.

No.	Number of steps	Traveled distance	Straight distance	Traveled distance/ straight distance	Success
1	15	2.72	2.12	1.28	✓
2	12	2.4	1.31	1.83	✓
3	9	1.61	0.64	2.49	✓
4	22	4.46	1.97	–	✗
5	16	3.25	2.15	1.51	✓
6	13	2.51	1.89	1.32	✓
7	7	1.21	1.01	1.2	✓
8	20	3.77	1.91	1.98	✓
9	15	3.13	2.19	1.43	✓
10	11	2.13	1.91	1.12	✓
Median	14			1.43	Success rate = 90%

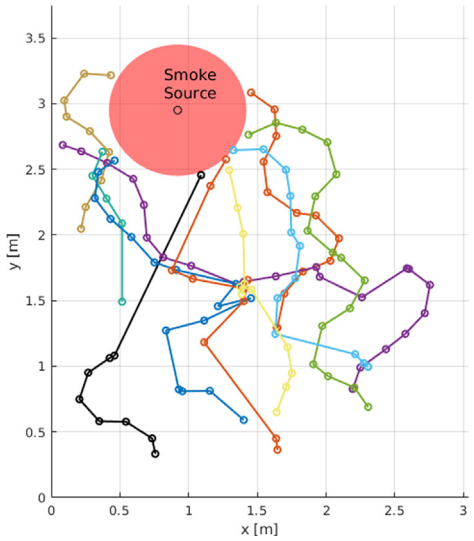


Fig. 12. Trajectories of the robot in 10 experiment runs (the red patch represents areas within 0.5 m of the source).

is used for tracking the robot's position as ground truth system for recording the robot's trajectory. The speed of the robot is set to 0.1079 m/s. (A snapshot of an experiment run is shown in Fig 11).

6.2. Results

The results of 10 experiment runs are presented in Table 5. Fig. 12 shows the robot's trajectories of 10 runs conducted in the laboratory. The robot starts from random positions in the scenario. Similar to the simulations, the robot takes observations, estimates the possible smoke plumes locations and navigates to directions yielding minimum entropy at each step. Among 10 runs, 9 is successful, and the traveled distance/ straight distance is around 1.43, which means the robot doesn't wander too much during its way to the smoke source. Noting that the environment in the laboratory is different from that in the wind tunnel and that the wind field is space-variant instead of a uniform wind field, the proposed algorithm shows its good performance and robustness to different environments. It takes 2898.6671ms in average for an iteration of the algorithm in real experiments. It has to be noting that the response rates of PMS 7003 and Gill WindSonic are 1 Hz and 4 Hz, respectively, which prolong the duration of every step in real experiments.

Videos recorded during experiments were uploaded at https://youtu.be/iKmY3U58B3k.

7. Conclusion

We proposed an anemotaxis – particle filter based smoke plume path tracking method by introducing a modified firefly algorithm in the resampling stage of the particle filter. The modified firefly algorithm affects the robot's behavior by gathering the particles, which represent the probability that a smoke plume exists at their locations, on the upwind direction of the robot. Therefore, driven by active information perception strategy, the robot shows anemotaxis, which is a common behavior among insects when they are searching for their mates.

The robustness of the method has been validated in two environments – a wind tunnel in simulation and a laboratory in reality. Experiments on the smoke source localization robots demonstrate the feasibility of the proposed method in different environments.

Our proposed method doesn't need any global localization information of the robot. Also, prior knowledge about the scenario is not needed to determine the measurement value threshold. The shortcoming of the proposed method is that the robot tends to spiral at the same place if it observes zero in some continuous steps. In our future work, we will study actively target searching in the navigation step instead of evaluating fixed preset targets. We will also study trajectory control strategies of the robot for preventing it from spiraling at local bests and somewhere with low smoke concentration.

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.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (U1913207), Hubei Technology Innovation Platform (2019AEA171), the Fundamental Research Funds for the Central Universities (HUST: 2019kfyRCPY, 2019kfyXKJC019) and the Research Fund of PLA of China (BWS17J024).

The authors would like to thank Prof. Alcherio Martinoli and Dr. Ali Marjovi at Distributed Intelligent Systems and Algorithm Laboratory, the Swiss Federal Institute of Technology Lausanne (EPFL) for providing the wind tunnel environment for data collection.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.measurement.2020.107482.

References

[1] X. Chen, J. Huang, Odor source localization algorithms on mobile robots: a review and future outlook, Rob. Auton. Syst. 112 (2019) 123–136.

[2] R. Rozas, J. Morales, D. Vega, Artificial smell detection for robotic navigation, in: Fifth International Conference on Advanced Robotics' Robots in Unstructured Environments, IEEE, 1991, pp. 1730–1733.
[3] J.M.B. Calvo, S.B. i Badia, H.T. Simó, P.F. Verschure, The real-world localization and classification of multiple odours using a biologically based neurorobotics approach, in: The 2010 International Joint Conference on Neural Networks (IJCNN), IEEE, 2010, pp. 1–7.
[4] F.W. Grasso, J. Atema, Integration of flow and chemical sensing for guidance of autonomous marine robots in turbulent flows, Environ. Fluid Mech. 2 (2002) 95–114.
[5] R.A. Russell, A. Bab-Hadiashar, R.L. Shepherd, G.G. Wallace, A comparison of reactive robot chemotaxis algorithms, Rob. Auton. Syst. 45 (2003) 83–97.
[6] T. Lochmatter, Bio-inspired and probabilistic algorithms for distributed odor source localization using mobile robots, Verlag nicht ermittelbar, 2010 (Ph.D. thesis).
[7] J. Ruddick, A. Marjovi, F. Rahbar, A. Martinoli, Design and performance evaluation of an infotaxis-based three-dimensional algorithm for odor source localization, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2018, pp. 1413–1420.
[8] B. Luo, Q.-H. Meng, J.-Y. Wang, M. Zeng, A flying odor compass to autonomously locate the gas source, IEEE Trans. Instrum. Meas. 67 (2017) 137–149.
[9] J. Zhang, D. Gong, Y. Zhang, A niching pso-based multi-robot cooperation method for localizing odor sources, Neurocomputing 123 (2014) 308–317.
[10] 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, Autonomous Robots 30 (2011) 281–292.
[11] S. Pang, J.A. Farrell, Chemical plume source localization, IEEE Trans. Syst., Man, Cybern., Part B (Cybernetics) 36 (2006) 1068–1080.
[12] H. Zhou, Y. Song, X. Pan, D. Li, Airborne particles detection and sizing at single particle level by a novel electrical current pulse sensor, Measurement 92 (2016) 58–62.
[13] A. Lay-Ekuakille, S. Ikezawa, M. Mugnaini, R. Morello, C. De Capua, Detection of specific macro and micropollutants in air monitoring: review of methods and techniques, Measurement 98 (2017) 49–59.
[14] Y. Chu, V. Kodur, D. Liang, A probabilistic inferential algorithm to determine fire source location based on inversion of multidimensional fire parameters, Fire Technol. 53 (2017) 1077–1100.
[15] X. Chen, J. Huang, Design and performance evaluation of a particle filter-based algorithm for smoke plume path tracking, in: 4th IEEE International Conference on Advanced Robotics and Mechatronics, ICARM 2019, Toyonaka, Japan, July 3–5, 2019, 2019, pp. 156–161. https://doi.org/10.1109/ICARM.2019.8834337..
[16] J. Peng, Y. Liu, C. Lyu, Y. Li, W. Zhou, K. Fan, Fpga-based parallel hardware architecture for sift algorithm, in: 2016 IEEE International Conference on Real-time Computing and Robotics (RCAR), IEEE, 2016, pp. 277–282.
[17] B. Zhao, C. Yang, X. Yang, S. Liu, Particle dispersion and deposition in ventilated rooms: testing and evaluation of different Eulerian and lagrangian models, Build. Environ. 43 (2008) 388–397.
[18] A. Famili, W.-M. Shen, R. Weber, E. Simoudis, Data preprocessing and intelligent data analysis, Intelligent Data Anal. 1 (1997) 3–23.
[19] D. Wang, F. Zhao, T. Wang, X. Zhang, Wifi fingerprint based indoor localization with iterative weighted knn for wifi ap missing, in: 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), IEEE, 2018, pp. 1–5.
[20] Y.-M. Chiang, Y.-L. Lin, W.-H. Chien, Automated surface defect inspection system for capacitive touch sensor, in: 2015 7th International Conference of Soft Computing and Pattern Recognition (SoCPaR), IEEE, 2015, pp. 274–277.
[21] T. Dimitrova-Grekow, V. Salauyou, K. Kowalski, Indoor mapping using sonar sensor and otsu method, Meas. Autom. Monit. 63 (2017).
[22] J.S. Kennedy, D. Marsh, Pheromone-regulated anemotaxis in flying moths, Science 184 (1974) 999–1001.
[23] J. Murlis, J.S. Elkinton, R.T. Carde, Odor plumes and how insects use them, Ann. Rev. Entomol. 37 (1992) 505–532.
[24] A.M. Reynolds, J.L. Swain, A.D. Smith, A.P. Martin, J.L. Osborne, Honeybees use a lévy flight search strategy and odour-mediated anemotaxis to relocate food sources, Behav. Ecol. Sociobiol. 64 (2009) 115.
[25] H. Zhu, H. Yi, R. Chellali, L. Feng, Object recognition and localization algorithm base on nao robot, in: 2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), IEEE, 2018, pp. 483–486.
[26] J. Peng, W. Xu, H. Yuan, An efficient pose measurement method of a space non-cooperative target based on stereo vision, IEEE Access 5 (2017) 22344–22362.
[27] O. Michel, Cyberbotics ltd. webots ™: professional mobile robot simulation, Int. J. Adv. Rob. Syst. 1 (2004) 5.
[28] J. Peng, W. Xu, E. Pan, L. Yan, B. Liang, A.-G. Wu, Dual-arm coordinated capturing of an unknown tumbling target based on efficient parameters estimation, Acta Astronaut. (2019).
[29] J. Peng, W. Xu, B. Liang, An autonomous pose measurement method of civil aviation charging port based on cumulative natural feature data, IEEE Sens. J. 19 (2019) 11646–11655.
[30] J. Huang, P. Di, T. Fukuda, et al., Motion control of omni-directional type cane robot based on human intention, in: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2008, pp. 273–278.
[31] Y. Lu, J. Huang, W. Xu, et al., An electronic travel aid based on multi-sensor fusion using extended kalman filter, in: 2014 11th World Congress on Intelligent Control and Automation, IEEE, 2014, pp. 25–30.
[32] T. Lochmatter, P. Roduit, C. Cianci, N. Correll, J. Jacot, A. Martinoli, Swistrack-a flexible open source tracking software for multi-agent systems, in: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2008, pp. 4004–4010.