

AN ANT-BASED FILTERING RANDOM-FINITE-SET APPROACH TO SIMULTANEOUS LOCALIZATION AND MAPPING

DEMENG LI^a, JIHONG ZHU^a, BENLIAN XU^{a,*}, MINGLI LU^a, MINGYUE LI^a

^a School of Electrical and Automatic Engineering
 Changshu Institute of Technology, 215500 Changshu, China
 e-mail: xu_benlian@cslg.edu.cn

Inspired by ant foraging, as well as modeling of the feature map and measurements as random finite sets, a novel formulation in an ant colony framework is proposed to jointly estimate the map and the vehicle trajectory so as to solve a feature-based simultaneous localization and mapping (SLAM) problem. This so-called ant-PHD-SLAM algorithm allows decomposing the recursion for the joint map-trajectory posterior density into a jointly propagated posterior density of the vehicle trajectory and the posterior density of the feature map conditioned on the vehicle trajectory. More specifically, an ant-PHD filter is proposed to jointly estimate the number of map features and their locations, namely, using the powerful search ability and collective cooperation of ants to complete the PHD-SLAM filter time prediction and data update process. Meanwhile, a novel fast moving ant estimator (F-MAE) is utilized to estimate the maneuvering vehicle trajectory. Evaluation and comparison using several numerical examples show a performance improvement over recently reported approaches. Moreover, the experimental results based on the robot operation system (ROS) platform validate the consistency with the results obtained from numerical simulations.

Keywords: simultaneous localization and mapping (SLAM), random finite sets, probability hypothesis density, ant colony.

1. Introduction

Simultaneous localization and mapping (SLAM) refers to a robot that starts from a location in an unknown environment and uses onboard sensors to observe the surrounding environment while incrementally building the environment map and simultaneously determining its position on the map (Leonard and Durrant-Whyte, 1991). Since the SLAM problem was presented for the first time, it has captured close attention in the field of autonomous mobile robots because of its important theoretical and practical values (Dissanayake *et al.*, 2001; Thrun *et al.*, 2005; Davison *et al.*, 2007; Wang *et al.*, 2013; Viejo *et al.*, 2014; Wilkowski *et al.*, 2016). As a prerequisite for robots to achieve real autonomy, SLAM is known as the Holy Grail of the intelligent mobile robot research field. Currently, SLAM systems generally fall into two groups, namely, online SLAM and full SLAM. The former seeks to recover the present robot location instead of the entire path and usually includes EKF-SLAM (Bailey *et al.*, 2006; Rodriguez-Losada *et al.*, 2007) and FastSLAM (Montemerlo *et al.*, 2002; Michael

et al., 2003) algorithms, and the random finite set (RFS) based approach (Mullane *et al.*, 2011). In turn, full SLAM involves estimating the posterior over the entire robot path together with the map, and graph-SLAM (Taketomi *et al.*, 2017; Carlevaris-Bianco *et al.*, 2017) is included. A marked characteristic of these algorithms is that both map features and measurements are treated as vectors strictly ordered in their respective finite vectors. Hence, these algorithms require feature association to determine the correspondence between the map features and the measurements. However, it is very difficult to deal with the feature association and management due to the uncertainty of detection, the growth in the number of features, and clutter distractions in the environment.

An alternative approach to solve the SLAM problem is called the random-finite-set (RFS) approach. An RFS is a set-valued random variable in which both the order and the number of elements are random, and Mahler's finite set statistic (FISST) provides practical mathematical tools and principled approximations for dealing with RFSs (Mahler, 2007), such as the probability hypothesis density (PHD) filter and the labeled multi-Bernoulli (LMB) filter.

Currently, there are two popular RFS based SLAM

*Corresponding author

algorithms, i.e., PHD-SLAM (Mullane *et al.*, 2011) and LMB-SLAM (Deusch *et al.*, 2015). In the former, the map features and measurements were treated as finite valued sets, whereas the full SLAM posterior probability density was first factorized into a product of the vehicle trajectory posterior probability density and the map posterior probability density conditioned on the vehicle trajectory. In the latter, the posterior density of the map, which is considered to be a labeled multi-object state, is represented by a labeled multi-Bernoulli RFS, and the LMB-RFS consists of multiple independent Bernoulli RFSs and assigns a unique label to each object. Since the full RFS-SLAM filter is numerically intractable, it is necessary to give tractable but principled approximations. One of the implementations of the RFS-SLAM algorithm is called RB-PHD-SLAM (Mullane *et al.*, 2011), in which a particle filter is used to propagate the vehicle trajectory, and a GM-PHD filter is used to propagate each trajectory-conditioned map PHD. LMB-SLAM is realized by using a GM-LMB filter and an RB particle filter to estimate the state of the map and the trajectory of the vehicle, respectively. The difference from the vector based SLAM algorithms is that the RFS based approach allows incorporating detection uncertainties, spurious measurements and data association uncertainties directly into the filter recursion.

Inspired by the self-organization behavior of the ant colony, a general population-based and meta-heuristic optimization algorithm, the ant colony optimization (ACO) algorithm was first proposed by Dorigo and co-workers (Dorigo *et al.*, 1995; Dorigo and Gambardella, 1997). With the characteristics of randomness, positive feedback and distributed collaboration, the ACO algorithm has been successfully applied in many areas, such as project scheduling (Zhang, 2015; Zhang *et al.*, 2016), vehicle routing (Huang *et al.*, 2015), weapon-target assignment (Narasimha *et al.*, 2013), feature selection (Shang, 2008; Aghdam *et al.*, 2009), target tracking (Rashno *et al.*, 2015; Xu *et al.*, 2010), etc. Since the ant foraging behavior could be utilized to solve the problem of parameter estimation, a new approach is proposed here, which combines the ACO with the PHD filter to solve the feature based SLAM problem. Unlike the RFS-based SLAM algorithm, the proposed algorithm defines a specific ant system, which uses intelligent ants, i.e., an ant-PHD filter, to estimate the features of the map through probabilistic selection and a local adjustment mechanism, and a fast moving ant estimator, i.e., F-MAE, to estimate the vehicle pose.

The paper is organized as follows. Some background for the PHD-SLAM and ACO algorithms is included in Section 2. A motivation for combining the ant system with the PHD filter and the details of the specific process of our approach are described in Section 3. Section 4 reports experimental simulation results and a comparison

with the existing SLAM algorithms. Finally, conclusions are drawn in Section 5.

2. Background

2.1. PHD-SLAM formulations. In general, the SLAM problem can be considered a state estimation problem whose goal is to jointly estimate the vehicle trajectory and the map which includes the number of map features and their locations by using all sensor measurements. It is assumed that the state of the vehicle pose follows a first-order Markov process, and the vehicle state is modeled as a random vector. Then the dynamic model for the vehicle is

$$\mathbf{X}_k = f_X(\mathbf{X}_{k-1}, \mathbf{U}_k, \boldsymbol{\omega}(k)), \quad (1)$$

where \mathbf{X}_k denotes the pose state of the vehicle containing its two-dimensional position (x_k, y_k) and its orientation θ_k at time k , $f_X(\cdot)$ denotes the state transition function of the vehicle, \mathbf{U}_k represents the control input to the vehicle at time k , and $\boldsymbol{\omega}(k)$ is process noise.

During mapping, the landmarks that appear in the field of view (FoV) are time-dependent due to the FoV limitations of sensors, such as the appearance of new landmarks and the disappearance of old landmarks. Therefore, modeling the SLAM problem with RFSs can naturally express the uncertainty of the measurements and the map features. Let the map state M_k be an RFS which evolves in time according to

$$M_k = M_{k-1} \cup B_k, \quad (2)$$

i.e., as the set union of the previous RFS multi-feature map M_{k-1} and B_k the RFS of the new features at time k .

In order to describe the uncertainty of measurements and the effect of the detected spatial clutter, the measurement Z_k is also treated as a finite set. In particular, the measurements received at each time step are formed by a series of measurements generated by the current real map features and false measurements. Therefore, in the case of a known feature map M_k and the vehicle state \mathbf{X}_k , the measurement set Z_k at time k can be expressed as

$$Z_k = \bigcup_{m \in M_k} D_k(m, \mathbf{X}_k) \cup C_k(\mathbf{X}_k), \quad (3)$$

where $D_k(m, \mathbf{X}_k)$ denotes the RFS of measurements at time k generated by a real map feature m , and $C_k(\mathbf{X}_k)$ denotes the RFS of the false measurements.

In general, the number of elements in $Z_k = \{z_k^1, z_k^2, \dots, z_k^n\}$ is random and may differ from the number of the real map features M_k , due to detection uncertainties and false measurements. For each real feature $m \in M_k$ and measurement $z_k^i \in Z_k$,

$D_k(m, \mathbf{X}_k) = \Phi$ with probability $1 - P_D(m, \mathbf{X}_k)$ and $D_k(m, \mathbf{X}_k) = \{z_k^i\}$ with probability density $P_D(m, \mathbf{X}_k)g_k(z_k^i|m, \mathbf{X}_k)$, where $P_D(m, \mathbf{X}_k)$ is the probability of detecting real feature m , and $g_k(z_k^i|m, \mathbf{X}_k)$ denotes the likelihood of the measurement z_k^i .

In consequence, the FB-SLAM problem can be posed in the framework of Bayesian filtering, and the probability density function (PDF) of the RFS can be propagated through a Bayesian recursion (Mullane *et al.*, 2011) as follows:

$$\begin{aligned} p_{k|k-1}(\mathbf{M}_k, \mathbf{X}_{1:k}) \\ = f_X(\mathbf{X}_k|\mathbf{X}_{k-1}, \mathbf{U}_k) \int f_M(\mathbf{M}_k|\mathbf{M}_{k-1}, \mathbf{X}_k) \\ \times p_{k-1}(\mathbf{M}_{k-1}, \mathbf{X}_{1:k-1})\delta(\mathbf{M}_{k-1}), \end{aligned} \quad (4)$$

$$\begin{aligned} p_k(\mathbf{M}_k, \mathbf{X}_{1:k}|Z_{0:k}) \\ = \frac{g_k(\mathbf{Z}_k|\mathbf{X}_k, \mathbf{M}_k)p_{k|k-1}(\mathbf{M}_k, \mathbf{X}_{1:k})}{g_k(\mathbf{Z}_k|Z_{0:k-1}, \mathbf{X}_0)}, \end{aligned} \quad (5)$$

where $f_M(\cdot)$ denotes the transition density of the set of features, and $\int \cdot \delta X$ is a multitarget set integral.

Unfortunately, the full RFS-SLAM Bayes filtering formulas (4) and (5) contain multiple integrals in the feature space, which leads to quite intractable mathematical calculations. It is necessary to resort to tractable but principled approximations. The PHD approach which propagates the first-order moment of the posterior multi-object RFS has proven to be both powerful and effective in multi-object filtering. For a map RFS \mathbf{M}_k , the PHD is a non-negative function v , such that for each region S in the space of feature

$$\int_S v_k(m) dm = \mathbb{E}[|\mathbf{M}_k \cap S|]. \quad (6)$$

The mass of the PHD gives the expected number of features in the map \mathbf{M}_k and its peaks indicate most likely locations of feature existence. However, this technique cannot be directly applied to the SLAM problem in which the joint posterior density of the map and the vehicle trajectory are propagated as a whole. An alternative way is that a weighted sum-of-Gaussians is used as the PHD function, and the mapping recursion is approximated by a GM-PHD filter, while the trajectory recursion adopts a particle filter (Mullane *et al.*, 2011).

The PHD predictor equation is

$$\begin{aligned} v_{k|k-1}(m|\mathbf{X}_k) = v_{k-1|k-1}(m|\mathbf{X}_{k-1}) \\ + b(m|\mathbf{X}_k), \end{aligned} \quad (7)$$

where $v_{k-1|k-1}(m|\mathbf{X}_{k-1})$ is the previous GM estimate of the PHD, $v_{k|k-1}(m|\mathbf{X}_k)$ is its prediction at time k , and $b(m|\mathbf{X}_k)$ is the GM-PHD of the birth RFS.

The PHD corrector equation is

$$\begin{aligned} v_{k|k}(m|\mathbf{X}_k) \\ = v_{k|k-1}(m|\mathbf{X}_k) \left[(1 - P_D(m, \mathbf{X}_k)) \right. \\ \left. + \sum_{z \in \mathbf{Z}_k} \frac{\Lambda(m|\mathbf{X}_k)}{c_k(z) + \int_{\mathbf{M}_k} \Lambda(\xi|\mathbf{X}_k)v_{k|k-1}(\xi|\mathbf{X}_k) d\xi} \right] \end{aligned} \quad (8)$$

where $v_{k|k}(m|\mathbf{X}_k)$ is the new GM estimate of the PHD at time k , and $\Lambda(m|\mathbf{X}_k) = P_D(m, \mathbf{X}_k)g_k(z|m, \mathbf{X}_k)$.

2.2. Generic ACO algorithm. The original idea of the ACO algorithm stems from the ant foraging process in real world. When ants search for food, they will release a kind of chemical substance called pheromone on their moving paths, and each ant can detect the pheromone on the path. The pheromone will concentrate on the path traveled by the majority of ants. Thus, this path will be chosen by more and more ants. Therefore, the higher the pheromone concentration, the more attractive the selected route, and thus a positive feedback mechanism is formed. On the other hand, as time evolves, the pheromone on the path will evaporate and its intensity will decrease accordingly. This forms a negative feedback mechanism, and those paths with a low level pheromone amount will be discarded by ants. In this way, shortcuts between their nest and food source can be found by this mechanism.

In the ACO algorithm, artificial ants are usually used to solve the combinatorial optimization problem by iteratively constructing a random candidate solution. In this process, the behavior of the ants is mainly affected by two aspects, namely, the pheromone value and the heuristic information. The pheromone value is the positive feedback of the ant paths in the whole algorithm, while the heuristic information is dependent on the problem and it has different forms in different cases.

Now suppose that there are a total of N ants, the ant s is at the decision point i , and then it will determine the direction of the next move based on the probability decision

$$P_{i,j}^s = \begin{cases} \frac{[\tau_{i,j}]^\alpha [\eta_{i,j}]^\beta}{\sum_{l \in N_i^s} [\tau_{i,l}]^\alpha [\eta_{i,l}]^\beta} & \text{if } j \in N_i^s, \\ 0 & \text{otherwise,} \end{cases} \quad (9)$$

where the parameters α and β reflect the relative importance of the pheromone trail $\tau_{i,j}$ and heuristic $\eta_{i,j}$ respectively, and N_i^s indicates the number of candidate points that have not been visited by ant s yet.

Along with ant traveling, a certain amount of pheromone will be released on its trail. If ant s reaches its destination, the path ant traveled is evaluated and the pheromone update process is executed. At each iteration t ,

the pheromone trail on the path from point i to its adjacent point j is updated by

$$\tau_{i,j}(t+1) = (1 - \rho)\tau_{i,j}(t) + \Delta\tau_{i,j}(t), \quad (10)$$

where $\tau_{i,j}(t)$ and $\tau_{i,j}(t+1)$ are the pheromone trails on the path from point i to point j before and after the update, respectively, and ρ is the pheromone evaporation coefficient expressed by a constant within interval $(0, 1)$. The effect of the evaporation coefficient ρ is to enhance the search ability of ants and to avoid premature convergence to sub-optimal solutions. Here $\Delta\tau_{i,j}(t)$ is the following pheromone trail updated by all ants:

$$\Delta\tau_{i,j}(t) = \sum_{s=1}^N \Delta\tau_{i,j}^s(t) \quad (11)$$

with

$$\Delta\tau_{i,j}^s(t) = \begin{cases} \frac{Q}{L_s} & \text{if ant } s \text{ moves from point } i \text{ to } j \text{ at step } t, \\ 0 & \text{otherwise,} \end{cases} \quad (12)$$

where Q is the pheromone intensity, and L_s is the total length traveled by ant s at the current iteration. Therefore, the shorter the path, the larger the value the pheromone trail.

3. Ant-PHD-SLAM method

This section characterizes the proposed ant-PHD-SLAM algorithm for the FB-SLAM problem, in which the location and the number of map features are estimated using an ant-PHD filtering, and the estimation of the vehicle trajectory is carried out using a fast moving ant estimator (F-MAE).

3.1. Motivation for combining PHD-SLAM with an ant system.

In the particle PHD-SLAM algorithm, the vehicle trajectory and the map are estimated separately. Specifically, the vehicle trajectory is estimated by the particle filter and the map is estimated by the particle PHD filter. Let a set of particles represent the PHD-SLAM density at time $k-1$, $\{\eta_{k-1}^{(i)}, \mathbf{X}_{0:k-1}^{(i)}, v_{k-1}^{(i)}(m|\mathbf{X}_{0:k-1}^{(i)})\}_{i=1}^N$, where $\eta_{k-1}^{(i)}$ is the weight of the i -th hypothesized vehicle trajectory $\mathbf{X}_{0:k-1}^{(i)}$ and $v_{k-1}^{(i)}(m|\mathbf{X}_{0:k-1}^{(i)})$ is the PHD of the map corresponding to this vehicle state $\mathbf{X}_{0:k-1}^{(i)}$. Then, the filter will approximate the map-trajectory posterior density of the next moment by a new set of weighted particles $\{\eta_k^{(i)}, \mathbf{X}_{0:k}^{(i)}, v_k^{(i)}(m|\mathbf{X}_{0:k}^{(i)})\}_{i=1}^N$. The expected vehicle trajectory is extracted through the particle with the highest weight, and the estimation map is its

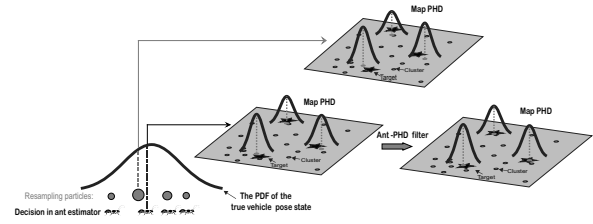


Fig. 1. Motivation behind the ant system combined with the PHD filter.

corresponding map. That is, the vehicle trajectory and the interrelated so that the estimation accuracy of the vehicle trajectory drastically affects the results of map estimation.

As was already mentioned, the vehicle trajectory can be estimated by a particle filter and a rigorous formulation can be found in the work of Xu *et al.* (2011), where an appropriate definition of importance sampling and the re-sampling strategy is necessary. The importance sampling and re-sampling scheme is in essence a process of adjusting the distribution of particles to approximate the posterior density of the state, and some particles with high importance weights, probably located near the true states, could produce more copies of the individual original state while those with low weights will be re-sampled with poor chances and discarded in the following iterations.

Inspired by the foraging behavior in the ant system, the importance sampling and re-sampling scheme could be improved and simulated by the distributed tendency mechanism in the ant search behavior. Specifically, the re-sampling process in the traditional particle filter is actually a one-step probabilistic decision without any local adjustment and particle cooperation, and the particles with largest weights are kept for subsequent use. Note that these largest weight particles may be far away from the real target, which may result in larger errors and faults both in the map and the vehicle pose. To solve this problem, a novel ant-PHD-SLAM is proposed and developed in the ant system framework to achieve a better map and vehicle pose estimation, as shown in Fig. 1.

As for map estimation, we use the ant system, instead of particles in generic PHD-SLAM to estimate the map. The success of ant-PHD has been verified by Xu *et al.* (2011) in tracking multi-objects. To get a vehicle pose estimate, the F-MAE is used and shows more robustness (Zhu *et al.*, 2010) when tracking maneuvering objects. This is the case that vehicle moves non-linearly when mapping. Therefore, it is expected that the use of ant-PHD-SLAM and F-MAE leads to an improved performance compared with the recently reported algorithms.

3.2. Ant-PHD filtering for feature map estimation.

At time $k - 1$, a set of weighted ants $\{w_{k-1}^{(i)}, \mathbf{x}_{k-1}^{(i)}\}_{i=1}^{L_{k-1}}$ is given and it represents the approximated distribution of discrete PHD function, where L_{k-1} represents the number of the ants, $w_{k-1}^{(i)}$ is the PHD associated with ant i and the expected number of targets can be approximated by $\hat{N}_{k-1} = \sum_{i=1}^{L_{k-1}} w_{k-1}^{(i)}$, $\mathbf{x}_{k-1}^{(i)} = [x_{k-1}^{(i)}, y_{k-1}^{(i)}, \dot{x}_{k-1}^{(i)}, \dot{y}_{k-1}^{(i)}]^T$ denotes the state of ant i , with its position $(x_{k-1}^{(i)}, y_{k-1}^{(i)})$ and velocity $(\dot{x}_{k-1}^{(i)}, \dot{y}_{k-1}^{(i)})$. Making an ant decision, we assume that each ant starts from its current state in ant state space Ω_{k-1} to the state in Ω_k at next time k . In the previous time, i.e., $k - 1$, the ant state space Ω_{k-1} is naturally expressed by $\Omega_{k-1} = \{w_{k-1}^{(i)}, \mathbf{x}_{k-1}^{(i)}\}_{i=1}^{L_{k-1}}$, while the ant state space at time k is defined by $\Omega_{k|k-1} = \{w_{k|k-1}^{(i)}, \mathbf{x}_{k|k-1}^{(i)}\}_{i=1}^{L_{k-1}} \cup \{w_k^{(j)}, \mathbf{x}_k^{(j)}\}_{j=1}^{J_k}$ with surviving ants and birth ants $\{w_k^{(j)}, \mathbf{x}_k^{(j)}\}_{j=1}^{J_k}$. In order to capture the new entering features at time k by the ants in Ω_{k-1} , the previously defined $\Omega_{k-1} = \{w_{k-1}^{(i)}, \mathbf{x}_{k-1}^{(i)}\}_{i=1}^{L_{k-1}}$ is then augmented by the birth ants $\{w_k^{(j)}, \mathbf{x}_k^{(j)}\}_{j=1}^{J_k}$ and yields $\Omega_{k-1} = \{w_{k-1}^{(i)}, \mathbf{x}_{k-1}^{(i)}\}_{i=1}^{L_{k-1}} \cup \{w_k^{(j)}, \mathbf{x}_k^{(j)}\}_{j=1}^{J_k}$, as shown in Fig. 2.

At time k , an ant s in Ω_{k-1} moves to the state of ant j in $\Omega_{k|k-1}$ according to the following probability (Xu *et al.*, 2011):

$$P_k^{s,j} = \frac{\sum_{z_k \in \mathbf{Z}_k} g(z_k | \mathbf{x}_{k|k-1}^{(j)}) f(\mathbf{x}_{k|k-1}^{(j)} | \mathbf{x}_{k-1}^{(s)})}{\sum_{i=1}^{L_{k-1}+J_k} \sum_{z_k \in \mathbf{Z}_k} g(z_k | \mathbf{x}_{k|k-1}^{(i)}) f(\mathbf{x}_{k|k-1}^{(i)} | \mathbf{x}_{k-1}^{(s)})} \quad (13)$$

When all ants have finished their one-step decisions from Ω_{k-1} to $\Omega_{k|k-1}$, the corresponding PHD of ant i need to be re-evaluated,

$$\bar{w}_{k|k-1}^{(i)} = \begin{cases} P_{S,k}(\mathbf{x}_{k-1}^{(i)}) w_{k-1}^{(i)}, & 1 \leq i \leq L_{k-1}, \\ \frac{\gamma_k(\mathbf{x}_{k|k-1}^{(i)})}{q_k(\mathbf{x}_{k|k-1}^{(i)} | \mathbf{Z}_k)} \frac{1}{J_k}, & L_{k-1} + 1 \leq i \leq L_{k-1} + J_k. \end{cases} \quad (14)$$

where $P_{S,k}(\cdot)$ denotes the surviving probability of an ant, $q_k(\cdot | \mathbf{Z}_k)$ represents a proposal density function, $\gamma_k(x)$ denotes the birth intensity function at time k . Actually, there are many ways to define these two functions. For simplicity, the proposed density $q_k(\cdot | \mathbf{Z}_k)$ follows the same rule as the state transition function $f_M(\cdot)$ and $\gamma_k(x)$ is defined as a Gaussian mixture form.

Due to the stochastic probability selection of each ant, not all the predicted locations in $\Omega_{k|k-1}$ are visited by ants. In other words, some candidates have not

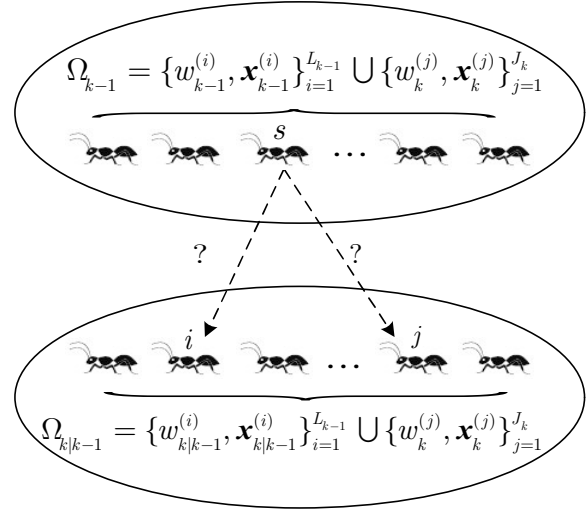


Fig. 2. Two search spaces at two consecutive sampling steps.

been visited at all while some have been visited several times. Therefore, the PHD update associated with ant i is performed only among those visited candidates,

$$\begin{aligned} \tilde{w}_k^{(i)} &= \bar{w}_{k|k-1}^{(i)} \left[1 - P_D(\mathbf{x}_{k|k-1}^{(i)}) \right] \\ &+ \sum_{z_k \in \mathbf{Z}_k} \frac{P_D(\mathbf{x}_{k|k-1}^{(i)}) g_k(z_k | \mathbf{x}_{k|k-1}^{(i)})}{C_k(z_k) + \sum_j P_D(\mathbf{x}_{k|k-1}^{(j)}) g_k(z_k | \mathbf{x}_{k|k-1}^{(j)}) \bar{w}_{k|k-1}^{(j)}}. \end{aligned} \quad (15)$$

Observe that (14) and (15) yield an initial estimate of PHD associated with each ant, and such a probabilistic movement is based on the ant one-step decision. This preliminary ant distribution needs a further adjustment, i.e., a local adjustment of the ant state. For the current ant distribution of map features, the probability that ant i selects ant j as a direction indicator of its next adjustment defined as

$$\begin{aligned} \bar{P}_k^{i,j} &= \frac{w_{k,\text{temp}}^{(j)} \tau_{i,j}(t)}{\sum_{i \neq j} w_{k,\text{temp}}^{(i)} \tau_{i,j}(t)}, \\ w_{k,\text{temp}}^{(i)} &= \frac{\tilde{w}_k^{(i)}}{\sum_{j=1}^{L_{k-1}+J_k} \tilde{w}_k^{(j)}}, \end{aligned} \quad (16)$$

where $w_{k,\text{temp}}^{(i)}$ is the temporary normalized intensity.

Once ant j is selected as a further movement direction, ant i has to make a decision how it can move close to ant j . For simplicity, we only consider the case of ant movement, in one dimension, e.g., the x direction, and for other dimensions, it follows in the same manner. Here,

two types of moving behavior models are investigated. If the intensity of ant j is greater than that of ant i , i.e., $w_{k,temp}^{(j)} > w_{k,temp}^{(i)}$, the update of ant i follows the rule

$$x_k^{(i)} = x_{k|k-1}^{(j)}, \quad (17)$$

otherwise, i.e., when $w_{k,temp}^{(j)} \leq w_{k,temp}^{(i)}$, the update of ant i is defined by

$$\begin{aligned} \dot{x}_k^{(i)} &= \dot{x}_{k|k-1}^{(i)} - \left(\frac{x_{k|k-1}^{(j)} - x_{k|k-1}^{(i)}}{T} \right) \cdot G_k^{i,j}, \\ x_k^{(i)} &= x_{k|k-1}^{(i)} - \dot{x}_k^{(i)} \cdot T, \end{aligned} \quad (18)$$

where T denotes the sampling interval, $G_k^{i,j}$ is the velocity adjustment coefficient defined by

$$G_k^{i,j} = 1 - \exp(-\lambda \cdot (w_{k,temp}^{(j)} - w_{k,temp}^{(i)})^2), \quad (19)$$

with λ as a constant value. It can be seen that if the intensity $w_{k,temp}^{(j)}$ of the selected state is much larger than the intensity $w_{k,temp}^{(i)}$, the value of $G_k^{i,j}$ will be close to 1, i.e., the velocity of ant i is close to the maximum velocity, so that the update position of ant i has a significant change. On the contrary, the update of the position of ant i will change a little.

The pheromone update process according to two types of moving behavior models is executed and modeled by the following law, respectively:

$$\tau_{i,j}(t+1) = \tau_{i,j}(t) + \frac{1}{\varepsilon} \tau_{i,j}(t), \quad (20)$$

$$\tau_{i,j}(t+1) = \tau_{i,j}(t) - \frac{G_k^{i,j}}{\varepsilon} \tau_{i,j}(t), \quad (21)$$

where $\varepsilon = \sum_{i \neq j} G_k^{i,j}$ is the total pheromone value deposited on the trail between states $\mathbf{x}_{k|k-1}^{(i)}$ and $\mathbf{x}_{k|k-1}^{(j)}$.

When all ants finish their state update, i.e., from $\mathbf{x}_{k|k-1}^{(i)}$ to $\mathbf{x}_k^{(i)}$, it is required to re-evaluate the PHD due to the value change of the likelihood function for each ant, as indicated in (15), and the PHD update of each ant, therefore, is re-formulated as

$$\begin{aligned} &\tilde{w}_k^{(i)} \\ &= \tilde{w}_{k|k-1}^{(i)} \left[1 - P_D(\mathbf{x}_k^{(i)}) \right. \\ &\quad \left. + \sum_{z_k \in \mathbf{Z}_k} \frac{P_D(\mathbf{x}_k^{(i)}) g_k(z_k | \mathbf{x}_k^{(i)})}{C_k(z_k) + \sum_j P_D(\mathbf{x}_k^{(j)}) g_k(z_k | \mathbf{x}_k^{(j)}) \tilde{w}_{k|k-1}^{(j)}} \right]. \end{aligned} \quad (22)$$

During the output of the map feature state, the merging operation is performed if two estimated features are located too close to each other, and the pruning operation is also required if the likelihood function of an ant is below a predefined threshold. For more details, the readers are referred to the work of Xu *et al.* (2011).

3.3. F-MAE for vehicle trajectory posterior estimation. Let the posterior distribution of the vehicle trajectory and the corresponding map PHD at time $k-1$ be represented by a set of N ants, $\{\eta_{k-1}^{(i)}, \mathbf{X}_{0:k-1}^{(i)}, v_{k-1}^{(i)}(m | \mathbf{X}_{0:k-1}^{(i)})\}_{i=1}^N$, where $\eta_{k-1}^{(i)}$ is the weight of the i -th ant, $\mathbf{X}_{0:k-1}^{(i)}$ is the whole trajectory of the i -th ant, and $v_{k-1}^{(i)}(m | \mathbf{X}_{0:k-1}^{(i)})$ is its map PHD.

The process of estimating the posterior probability density of the vehicle trajectory using the F-MAE can be divided into the following four main steps (Zhu *et al.*, 2010).

1. *Prediction:* It is assumed that the ant state $\mathbf{X}_{k-1}^{(i)}$ representing the vehicle pose at time $k-1$ is known, and the corresponding one-step prediction ant state $\mathbf{X}_{k|k-1}^{(i)}$ is available according to the following formula:

$$\mathbf{X}_{k|k-1}^{(i)} = f_X(\mathbf{X}_{k-1}^{(i)}, \mathbf{U}_k, \boldsymbol{\omega}(k)). \quad (23)$$

2. *Weighing:* The weight of the i -th vehicle pose ant can be calculated according to

$$\eta_k^{(i)} = g_k(\mathbf{Z}_k | \mathbf{Z}_{k-1}, \mathbf{X}_{k|k-1}^{(i)}) \eta_{k-1}^{(i)}, \quad (24)$$

where the measurement likelihood is defined by

$$\begin{aligned} &g_k(\mathbf{Z}_k | \mathbf{Z}_{k-1}, \mathbf{X}_{k|k-1}^{(i)}) \\ &\approx \frac{1}{\Gamma} v_{k|k-1}(\bar{\mathbf{m}} | \mathbf{X}_{k|k-1}^{(i)}) \left[(1 - P_D) c_k(z)^{\mathbf{Z}_k} \right. \\ &\quad \left. + P_D \sum_{z \in \mathbf{Z}_k} c_k(z)^{\mathbf{Z}_k - \{z\}} g_k(z | \bar{\mathbf{m}}, \mathbf{X}_{k|k-1}^{(i)}) \right] \end{aligned}$$

with

$$\Gamma = \exp(\hat{\mathbf{m}}_{k|k-1} - \hat{\mathbf{m}}_k + \int c_k(z) dz) v_k(\bar{\mathbf{m}} | \mathbf{X}_{k|k-1}^{(i)}),$$

where

$$\hat{\mathbf{m}}_{k|k-1} = \int v_{k|k-1}(m | \mathbf{X}_{0:k}) dm,$$

$$\hat{\mathbf{m}}_k = \int v_k(m | \mathbf{X}_{0:k}) dm,$$

and $\bar{\mathbf{m}}$ denotes the single feature map $\mathbf{M}_k = \{\bar{\mathbf{m}}\}$ selected with the feature with the least uncertainty or the maximum measurement likelihood.

3. *Selection:* Each ant needs to make a probabilistic decision before moving, and then decides the direction of the movement according to the probability

$$P_k^{i,j} = \frac{\eta_k^{(j)} \tau_{i,j}(t)}{\sum_{i \neq j} \eta_k^{(j)} \tau_{i,j}(t)}, \quad (25)$$

where $\tau_{i,j}(t)$ is the pheromone value between $\mathbf{X}_{k-1}^{(i)}$ and $\mathbf{X}_{k|k-1}^{(j)}$.

Table 1. Estimation of map features using an ant-PHD filter.

```

\\ initialize the distribution of ants
1.  $\Omega_{k-1} = \{w_{k-1}^{(i)}, \mathbf{x}_{k-1}^{(i)}\}_{i=1}^{L_{k-1}} \cup \{w_k^{(j)}, \mathbf{x}_k^{(j)}\}_{j=1}^{J_k}$ 
\\ compute the predicted distribution of ants
2.  $\Omega_{k|k-1} = \{w_{k|k-1}^{(i)}, \mathbf{x}_{k|k-1}^{(i)}\}_{i=1}^{L_{k-1}} \cup \{w_k^{(j)}, \mathbf{x}_k^{(j)}\}_{j=1}^{J_k}$ 
\\ for each ant
3. for each ant from 1 to  $L_{k-1} + J_k$  do
4. compute the selection probability  $P_k^{s,j}$  in (13)
5. compute the predicted weight  $\tilde{w}_{k|k-1}^{(i)}$  in (14)
6. update the weight  $\tilde{w}_k^{(i)}$  in (15)
7. end for
\\ local adjustment strategy
8. for each ant from 1 to  $L_{k-1} + J_k$  do
9. compute the probability  $\tilde{P}_k^{i,j}$  in (16)
\\ update the state of ants
10.  $x_k^{(i)} = x_{k|k-1}^{(j)}$  or  $x_k^{(i)} = x_{k|k-1}^{(i)} - \dot{x}_k^{(i)} \cdot T$ 
11. update the pheromone  $\tau_{i,j}(t+1)$  in (20) and (21)
12. update the PHD of each ant  $\tilde{w}_k^{(i)}$  in (22)
13. end for
14. pruning and merging operations
15. estimate and output map

```

4. *Update*: Once each ant has determined its move direction, it will use the same local adjustment strategy as detailed in the previous section to reach the destination and update its state. Meanwhile, both the pheromone intensity and weight are also updated as in (20)–(22). That is to say, the update mechanism is the same as the one in the ant-PHD filter.

The pseudo-codes of our algorithm for the FB-SLAM problem are given in Tables 1 and 2. Table 1 outlines the feature estimation of the map using the ant-PHD filter, and Table 2 illustrates the trajectory estimation of the vehicle using the F-MAE filter. Intuitively, the posterior distribution of the vehicle trajectory is available and represented by $\{\eta_k^{(i)}, \mathbf{X}_k^{(i)}\}_{i=1}^N$, while the expected state of the vehicle trajectory is extracted and denoted by an ant with the highest weight within the current ant colony, and the expected feature map is its corresponding map.

4. Experiments and analysis

4.1. Simulation environment and parameter settings. In this section, we will test and analyze the performance of the proposed approach by simulation experiments. The map features are assumed to have the same detection probability, and the clutter intensity is uniformly distributed in the measurement space, while the number of clutter measurements follows a Poisson distribution. The following cases are tested:

Table 2. Estimation of the vehicle trajectory using F-MAE.

```

\\ initialize the state of ants at time  $k-1$ 
1.  $\{\eta_{k-1}^{(i)}, \mathbf{X}_{0:k-1}^{(i)}\}_{i=1}^N$ 
2. for each ants from 1 to  $N$  do
\\ compute the predicted state in (22)
3.  $X_{k|k-1}^{(i)} = f_X(\mathbf{X}_{k-1}^{(i)}, \mathbf{U}_k, \omega(k))$ 
\\ compute the weight of ants in (23)
4.  $\eta_k^{(i)} = g(\mathbf{Z}_k | \mathbf{Z}_{k-1}, \mathbf{X}_{k|k-1}^{(i)}) \eta_{k-1}^{(i)}$ 
\\ decide the moving direction in (25)
5.  $P_k^{i,j} = \frac{\eta_k^{(j)} \tau_{i,j}(t)}{\sum_{i \neq j} \eta_k^{(j)} \tau_{i,j}(t)}$ 
\\ update the state of ants
6.  $x_k^{(i)} = x_{k|k-1}^{(j)}$  or  $x_k^{(i)} = x_{k|k-1}^{(i)} - \dot{x}_k^{(i)} \cdot T$ 
7. update the pheromone  $\tau_{i,j}(t+1)$  in (20) and (21)
8. update the weight of each ant in (22)
9. end for
10. estimate and output vehicle pose state  $\mathbf{X}_k^{(i)}$ 
    corresponding to the largest weight  $\eta_k^{(i)}$ 

```

Table 3. Parameters of the motion model.

Parameter	Variable	Value
velocity sensor	V	3 m/s
	maximum range	10 m
	maximum bearing	150°
measurement noise	range	0.4 m
	bearing	2°
control noise	velocity	0.3 m/s
	steer	2°

Case 1: A total of 50 features are randomly distributed in the simulation area, and at least one landmark can be observed by the vehicle in each time step.

Case 2: A total of 16 features in the simulation area, and at some time steps, the landmarks are not within the scope of the vehicle.

The simulated environments of two cases with an area size of 70 m × 60 m are given in Fig. 3, in which the gray solid circle represent the real map features, the black dots represent measurements with noise and the black line is the real vehicle trajectory.

In order to show the performance of the proposed ant-PHD-SLAM algorithm more intuitively, both the FastSLAM algorithm (Michael *et al.*, 2003) and the PHD-SLAM algorithm (Mullane *et al.*, 2011) are introduced and the evaluation is conducted along with the ant-PHD-SLAM algorithm. The number of particles or ants used by all SLAM filters is 20 for the vehicle trajectory estimation. It should be noted that the simulated vehicle used in the experiment is equipped with a range-bearing sensor with a detection probability of $P_D = 0.98$.

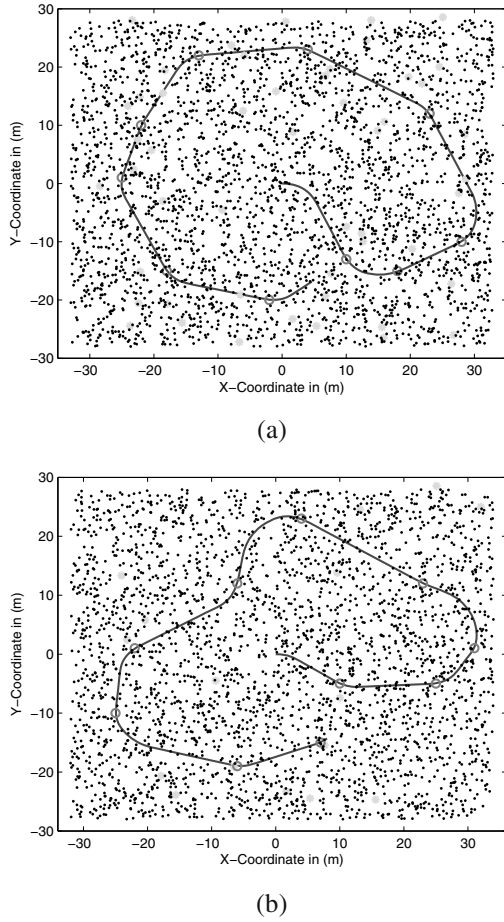


Fig. 3. Simulated environment.

The specific motion model parameters used in the simulated trial are shown in Table 3. Assume that the number of clutters follows the Poisson distribution with an average rate of $\lambda_c = 10$ per-scan, and 10 Monte-Carlo (MC) runs are carried out for each SLAM filter.

4.2. Simulation results and performance evaluation.

The overall performance of the three SLAM algorithms on the studied two cases is illustrated in Fig. 4. The black line is the ground truth vehicle trajectory, and the black dots and gray line represent respectively the estimated features position and the estimated vehicle trajectory. It can be observed that all algorithms could produce accurate estimates in both the vehicle trajectory and the map features, and our proposed algorithm performs better in a general way.

Vehicle trajectory estimate. In order to evaluate the performance of vehicle pose, we adopt the metric of root

mean square error (RMSE) defined as

$$\text{RMSE}_k(x_k, \hat{x}_k) = \sqrt{\frac{1}{\text{MC}} \sum_{i=1}^{\text{MC}} (\|x_k - \hat{x}_k\|)^2} \quad (26)$$

where $\text{MC} = 10$ indicates the number of Monte Carlo runs, x_k and \hat{x}_k are defined as the true and estimated positions of the vehicle, respectively.

As shown in Fig. 5, the RMSEs corresponding to our algorithm are generally smaller compared with other two algorithms both in the x and y directions. Figure 6 shows the overall trajectory estimation errors of the three SLAM algorithms in the vehicle position estimate. Through the above analyses, we observe that the ant-PHD-SLAM algorithm is more accurate in estimating the vehicle pose, and it performs better than the other two SLAM algorithms.

Map feature estimate. We adopt the average optimal sub-pattern assignment (OSPA) (Adams et al., 2014) to evaluate the performance of the map feature estimate over time. The OSPA metric is designed on the basis of the Wasserstein distance (WD), which overcomes some limitations such as inconsistent behavior and the lack of a meaningful physical interpretation by introducing a parameter controlling the relative emphasis of localization and cardinality errors, and can accommodate cardinality differences in a mathematically consistent and physically meaningful sense.

Let $d^{(c)}(x, y) \triangleq \min(c, d(x, y))$ denote the distance between x and y with a nonnegative cut-off parameter c (we used $c = 5$ in our experiments), and let ψ_k be the set of permutations on $\{1, 2, \dots, k\}$ for $k \in \mathbb{N} = \{1, 2, \dots\}$. For $1 \leq p < \infty$ and any two finite subsets $\mathbf{X} = \{x_1, x_2, \dots, x_m\}$ and $\mathbf{Y} = \{y_1, y_2, \dots, y_n\}$, if $m < n$, the OSPA is then defined as

$$\begin{aligned} \bar{d}_p^{(c)}(\mathbf{X}, \mathbf{Y}) \\ \triangleq \left(\frac{1}{n} \left(\min_{\psi \in \psi_n} \sum_{i=1}^m d^{(c)}(x_i, y_{\psi(i)})^p + c^p(n-m) \right) \right)^{1/p}. \end{aligned} \quad (27)$$

In addition, as $p \rightarrow \infty$, the OSPA metric is defined as

$$\begin{aligned} \bar{d}_\infty^{(c)}(\mathbf{X}, \mathbf{Y}) \\ \triangleq \begin{cases} \min_{\psi \in \psi_n} \max_{1 \leq i \leq n} d^{(c)}(x_i, y_{\psi(i)}) & \text{if } m = n, \\ c & \text{otherwise.} \end{cases} \end{aligned} \quad (28)$$

Note that, if $m = n = 0$, the OSPA metric is set as zero directly for both cases.

Figure 7 shows a comparison of results for the number of map features estimated by the three algorithms at each time step. To present the estimated results in a

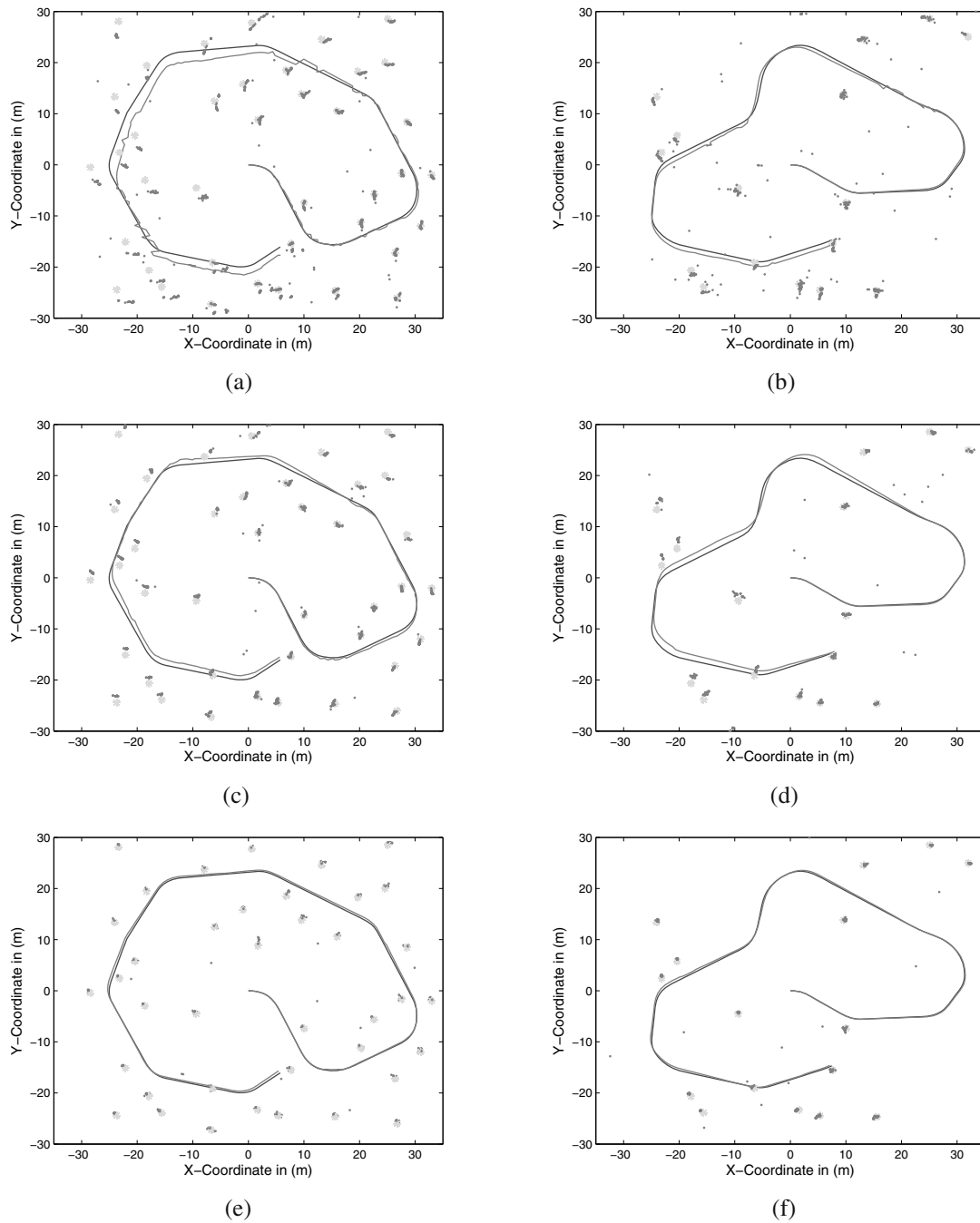


Fig. 4. Estimated results by different approaches: FastSLAM (Case 1) (a), FastSLAM (Case 2) (b), PHD-SLAM (Case 1) (c), PHD-SLAM (Case 2) (d), ant-PHD-SLAM (Case 1) (e), ant-PHD-SLAM (Case 2) (f).

more clear way, the inconsistency of cardinality at each time is only marked and highlighted along with the black line representing the true number of the map features at each time step. Similarly, the comparison results of the number of missing and false alarms for each SLAM filters are shown in Fig. 8, where the ant-PHD-SLAM algorithm has a lower number of missing and false alarms than other algorithms. In other words, the ant-PHD-SLAM

algorithm is more accurate in estimating the number of map features, and its estimation accuracy is much higher than those of the FastSLAM and PHD-SLAM algorithms.

The OSPA comparisons are conducted in Fig. 9, and a smaller c is selected to highlight the importance of the map feature position errors and neglect the cardinality differences. It can be seen that the ant-PHD-SLAM algorithm error curve has smaller fluctuations and the

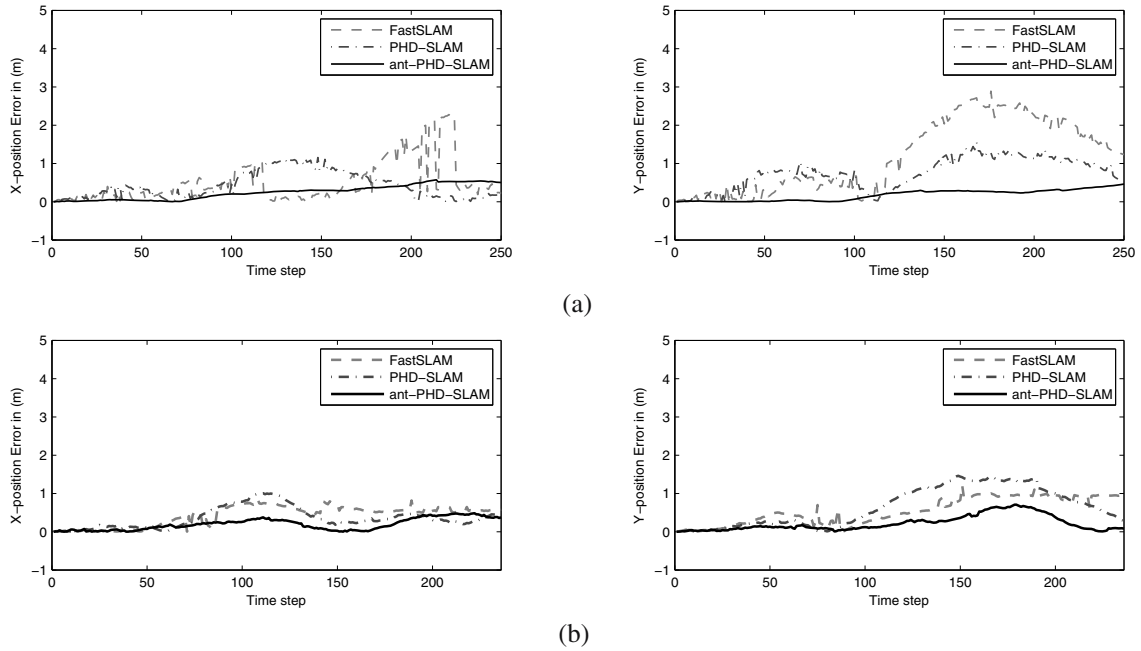


Fig. 5. Vehicle pose estimated errors by different approaches: Case 1 (a), Case 2 (b).

Table 4. Average operation time of each step.

Case	FastSLAM	PHD-SLAM	ant-PHD-SLAM
1	0.2957[s]	0.7163[s]	1.7315[s]
2	0.1816[s]	0.3962[s]	0.9142[s]

value at each time step is smaller as well.

In terms of the computational burden, the average operation time of each SLAM algorithm at each time step in two different simulation scenarios is recorded. It was found that, as illustrated in Table 4, our algorithm consumes more effort than the other two algorithms even in 2D applications, although its estimated accuracy of the vehicle and map features is higher.

Finally, the real indoor environment where we work has been selected to test the proposed algorithm. The testing environment and the robot platform are shown in Fig. 10. Our TurtleBot2 robot based on the robot operation system (ROS) is equipped with the RPLIDAR A1 360° laser scanning range radar, as shown in Fig. 10(a). In the TurtleBot2 system, three main parts including the robot system, ROS and MATLAB, directly or indirectly communicate with each other through WLAN. The ROS is a communication interface that enables different parts of the robot system to discover, send and receive data. When the robot obtains sensor data, the ROS will receive and preprocess the data. Afterwards, it sends results to MATLAB to process them and send back to the ROS. Figure 10(b) shows a corridor based environment, where the upper half of the figure illustrates the sketch map of the whole environment and the lower half gives the pictures

of each corridor. The total length of corridors is 130 m. Figure 10(c) illustrates a landmark based environment, which shows a 5 m×5 m region with 17 landmarks, and the diameter of each landmark is 24 mm.

In order to show the effectiveness of the proposed algorithm and quantify the estimation accuracy of both the vehicle pose and the map features, the FastSLAM and PHD-SLAM algorithms are tested in the corridor based environment. In this experiment, the moving speed of robot is 0.2 m/s, and the maximum detection range and FoV are set to be 5 m and 180°, respectively. The results comparison of the position estimation, average position error and OSPA distance are shown in Fig. 11, and the estimated results of the vehicle trajectory and map features are given in Fig. 12, where the gray and white lines denote the true and estimated trajectory, respectively. It can be seen that the estimation results for the proposed algorithm are more accurate than for the other two SLAM algorithms.

Furthermore, to test our algorithm in a unified RFS framework, the LMB filter is introduced in our experiment. Thus, three RFS-based algorithms, i.e., PHD-SLAM, LMB-SLAM, and ant-PHD-SLAM, are tested in the landmark-based environment. In our experiment, the moving speed of the robot is 0.1 m/s, and the maximum detection range and FoV are set to be 1.5 m and 180°, respectively. The results of a comparison for the position estimation, average position error and OSPA distance are shown in Fig. 13, and the estimation results of the vehicle trajectory and map features are illustrated in Fig. 14. As can be seen, the estimated accuracy of

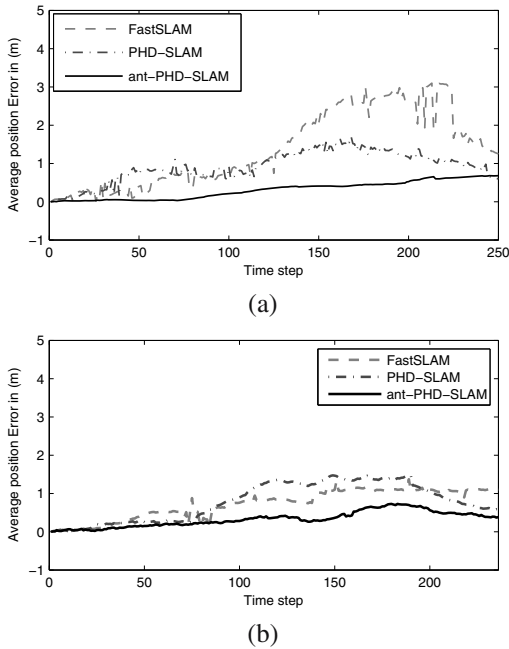


Fig. 6. Average estimated errors of the vehicle pose by different approaches: Case 1 (a), Case 2 (b).

the proposed algorithm is higher than for the PHD-SLAM algorithm, and close to the LMB-SLAM algorithm.

To get insight into the effectiveness of various approaches, the average computation time at each time step is recorded and calculated. In the corridor environment, the average running time is, 0.3542 s, 0.9257 s and 1.8435 s for FastSLAM, PHD-SLAM and ant-PHD-SLAM, respectively. It is observed that our algorithm suffers from a larger computational burden. Similarly, in the landmark environment, our algorithm consumes on the average 1.4850 s in running, twice as much as PHD-SLAM required with 0.7725 s, and a little more time than the LMB-SLAM algorithm with 1.1765 s. Therefore, the most burning issue in our algorithm is to optimize the procedure, such as the look-ahead ant mechanism, or pre-segmentation in the ant system for a parallel implementation to operate on those real SLAM systems.

5. Conclusions

An ant-PHD-SLAM algorithm has been proposed to solve the simultaneous localization and mapping problem. The ant based PHD filter is used to estimate the feature map and a fast moving ant estimator is used to estimate the vehicle trajectory. The performance comparisons have been drawn through two different simulated scenarios and tested in a real environment by using ROS. The results show that the proposed algorithm achieves more estimate accuracy in terms of the state of the vehicle pose, the

state and the cardinality of the map feature. In addition, the performance comparisons have been made on the ROS platform in a real environment, and our algorithm shows superiority over other approaches in terms of the estimated accuracy. Note that our algorithm performs in the same way as LMB-SLAM, but they both suffer from the computational burden. In our future work, our efforts will be directed towards reduction in the computational burden of the proposed algorithm to make it meaningful in real environments.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (no. 61673075), the Natural Science Fundamental Research Program of Higher Education Colleges in Jiangsu Province (no. 14KJB510001, 14KJD510001), the Suzhou Municipal Science and Technology Plan Project (no. SYG201548) and the Natural Science Research Program of the Changshu Institute of Technology (no. XZ1736).

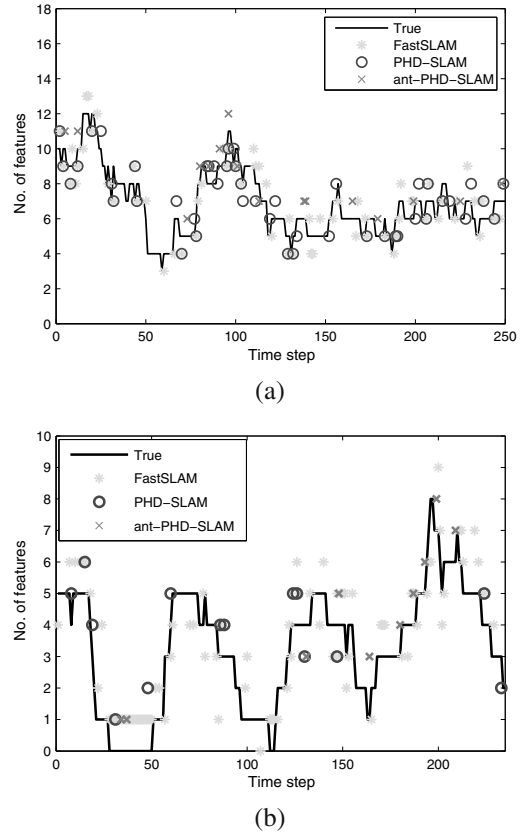
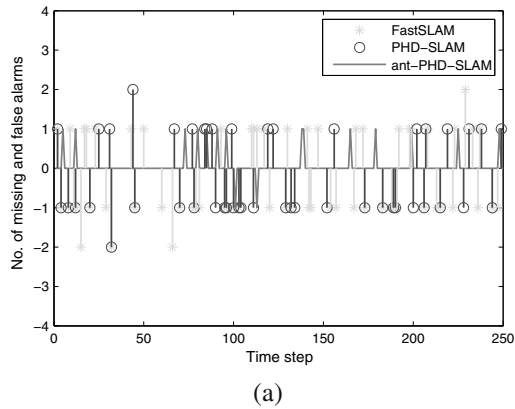
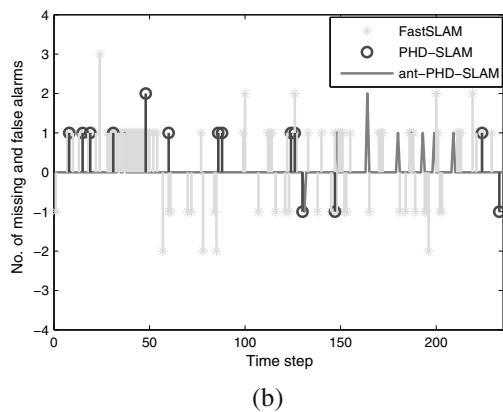


Fig. 7. Cardinality estimates of map features by different approaches: Case 1 (a), Case 2 (b).



(a)

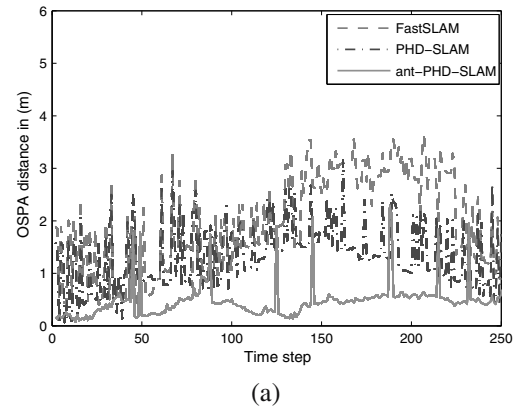


(b)

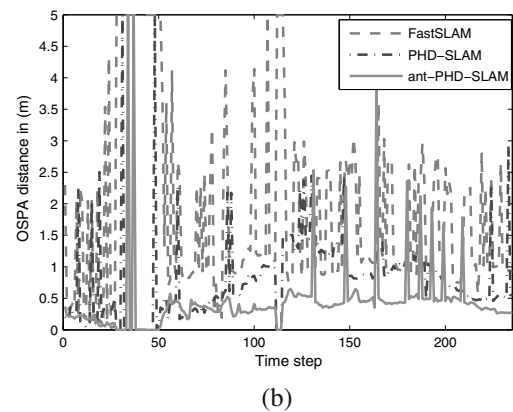
Fig. 8. Missing and false alarm estimates by different approaches: Case 1 (a), Case 2 (b).

References

- Adams, M., Vo, B., Mahler, R. and Mullane, J. (2014). SLAM gets a PHD: New concepts in map estimation, *IEEE Robotics and Automation Magazine* **21**(2): 26–37.
- Aghdam, M., Ghasem-Aghaee, N. and Basiri, M. (2009). Text feature selection using ant colony optimization, *Expert Systems with Applications* **36**(3): 6843–6853.
- Bailey, T., Nieto, J., Guivant, J., Stevens, M. and Nebot, E. (2006). Consistency of the EKF-SLAM algorithm, *Proceedings of the 2006 IEEE International Conference on Intelligent Robots and Systems, Beijing, China*, pp. 3562–3568.
- Carlevaris-Bianco, N., Kaess, M. and Eustice, R.M. (2017). Generic node removal for factor-graph SLAM, *IEEE Transactions on Robotics* **30**(6): 1371–1385.
- Davison, A., Reid, I., Molton, N. and Stasse, O. (2007). MonoSLAM: Real-time single camera SLAM, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(6): 1052–1067.
- Deusch, H., Reuter, S. and Dietmayer, K. (2015). The labeled multi-Bernoulli SLAM filter, *IEEE Signal Processing Letters* **22**(10): 1561–1565.
- Dissanayake, G. and Durrant-Whyte, P., Clark, H. and Csorba, M. (2001). A solution to the simultaneous localization



(a)



(b)

Fig. 9. OSPA distance of map features by different approaches: Case 1 (a), Case 2 (b).

and map building (SLAM) problem, *IEEE Transactions on Robotic and Automation* **17**(3): 229–241.

- Dorigo, M. and Gambardella, L. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem, *IEEE Transactions on Evolutionary Computation* **1**(1): 53–66.
- Dorigo, M., Maniezzo, V. and Colorni, A. (1995). The ant system: Optimization by a colony of cooperative agents, *Physical Review Letters* **75**(14): 2686–2689.
- Huang, H., Huang, C. and Pei, W. (2015). Solving multi-resource constrained project scheduling problem using ant colony optimization, *Journal of Engineering Project and Production Management* **5**(1): 2–12.
- Leonard, J.J. and Durrant-Whyte, H.F. (1991). Simultaneous map building and localization for an autonomous mobile robot, *Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems IROS'91, Osaka, Japan*, pp. 1142–1147.
- Mahler, R. (2007). *Statistical Multi-Source Multi-Target Information Fusion*, Artech House, Boston, MA.
- Michael, M., Sebastian, T., Daphne, K. and Ben, W. (2003). FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges, *Proceedings of the 6th International Joint*

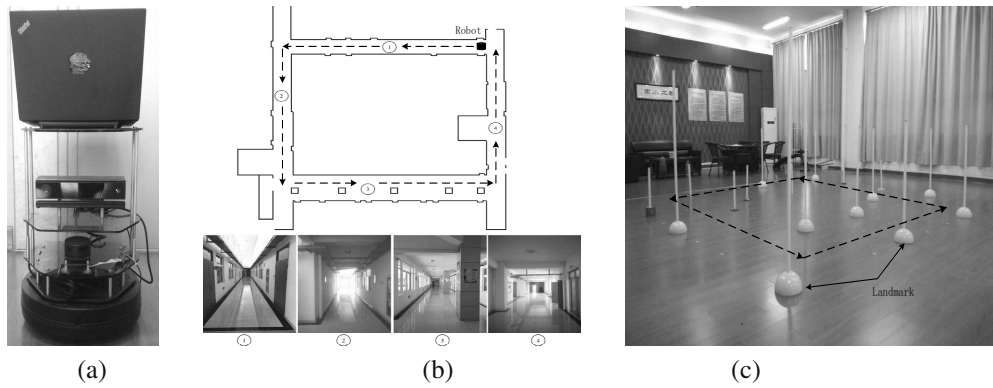


Fig. 10. Testing environment: robot (a), corridors (b), landmarks (c).

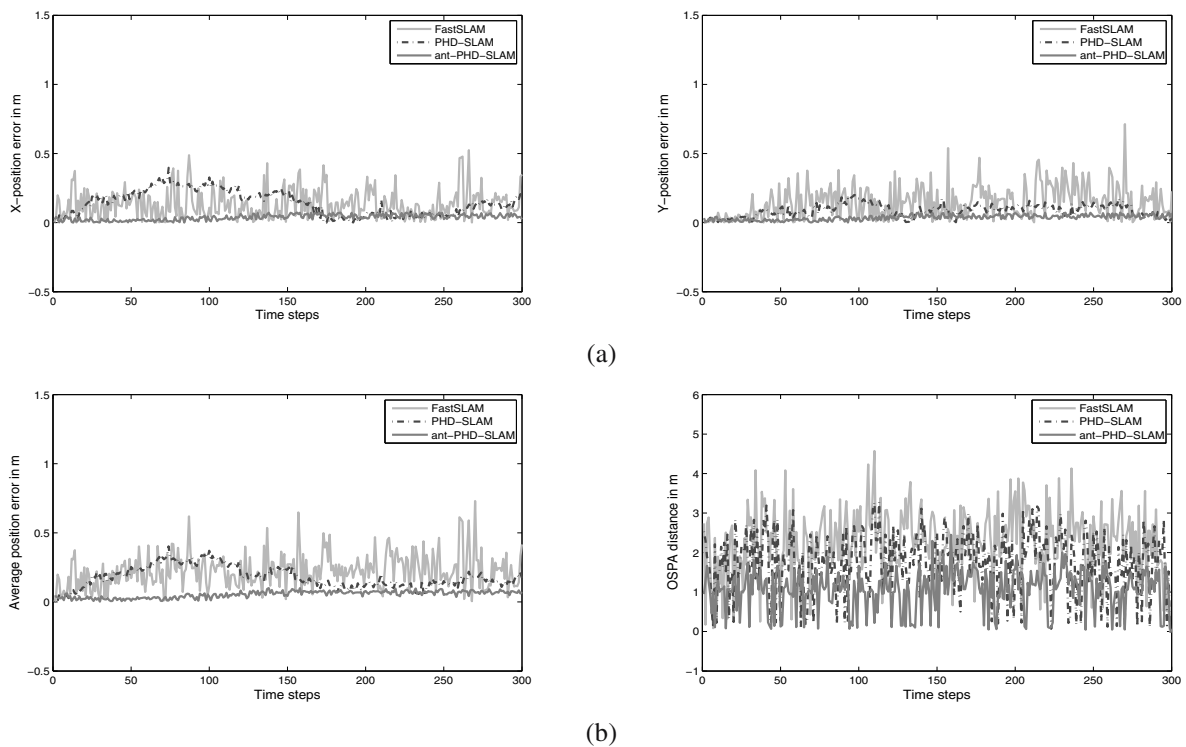


Fig. 11. Estimated results in a real corridor environment by different approaches: estimated vehicle pose errors (a), average estimated vehicle pose errors (left) and the OSPA distance of map features (right) (b).

Conference on Artificial Intelligence, Acapulco, Mexico, pp. 1151–1156.

Montemerlo, M., Thrun, S., Koller, D. and Wegbreit, B. (2002). FastSLAM: A factored solution to the simultaneous localization and mapping problem, *Proceedings of the 8th National Conference on Artificial Intelligence, Edmonton, Alberta, Canada*, pp. 593–598.

Mullane, J., Vo, B.N., Adams, M.D. and Vo, B.T. (2011). A random-finite-set approach to Bayesian SLAM, *IEEE Transactions on Robotics* **27**(2): 268–282.

Narasimha, K., Kivelevitch, E., Sharma, B. and Kumar, M. (2013). An ant colony optimization technique for solving

min-max multi-depot vehicle routing problem, *Swarm and Evolutionary Computation* **13**: 63–73.

Rashno, A., Sadri, S. and Sadeghian Nejad, H. (2015). An efficient content-based image retrieval with ant colony optimization feature selection schema based on wavelet and color features, *Proceedings of the 2015 International Symposium on Artificial Intelligence and Signal Processing, Mashhad, Iran*, pp. 59–64.

Rodriguez-Losada, D., Matia, F., Pedraza, L., Jimenez, A. and Galan, R. (2007). Consistency of SLAM-EKF algorithms for indoor environments, *Journal of Intelligent and Robotic Systems* **50**(4): 375–397.

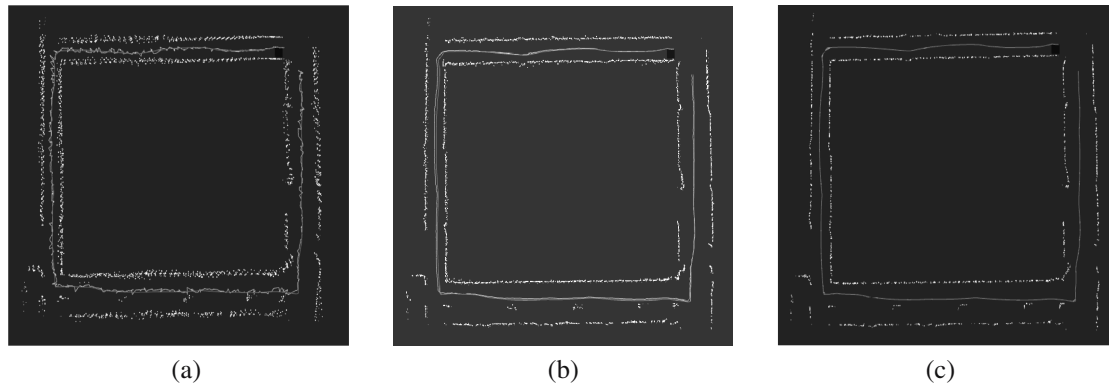


Fig. 12. Estimation results in real corridor environment: FastSLAM (a), PHD-SLAM (b), ant-PHD-SLAM (c).

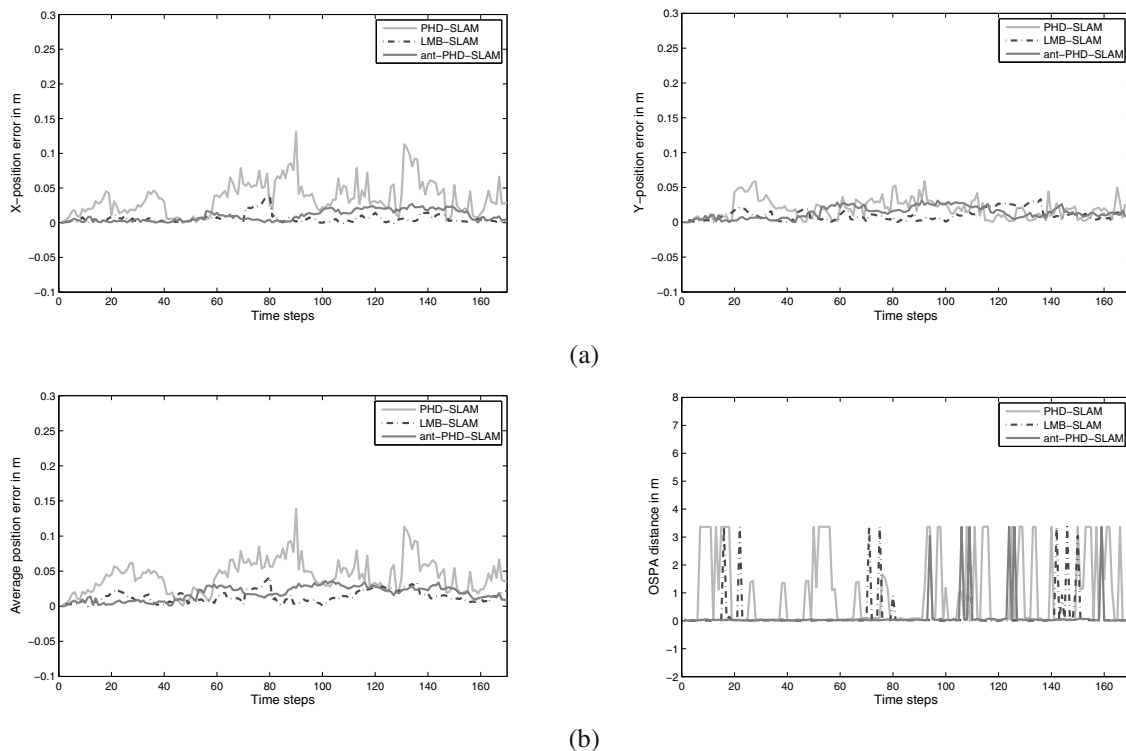


Fig. 13. Estimated results in a real landmark environment by RFS based approaches: estimated vehicle pose errors (a), average estimated vehicle pose errors (left) and the OSPA distance of map features (right) (b).

Shang, G. (2008). Solving weapon-target assignment problems by a new ant colony algorithm, *Proceedings of the 2008 IEEE International Symposium on Computational Intelligence and Design, Wuhan, China*, pp. 221–224.

Taketomi, T., Uchiyama, H. and Ikeda, S. (2017). Visual SLAM algorithms: A survey from 2010 to 2016, *IPSA Transactions on Computer Vision & Applications* **9**(1): 16–26.

Thrun, S., Burgard, W. and Fox, D. (2005). *Probabilistic Robotics*, MIT Press, Cambridge, MA.

Viejo, D., Garcia-Rodriguez, J. and Cazorla, M. (2014). Combining visual features and growing neural gas

networks for robotic 3D SLAM, *Information Sciences* **276**: 174–185.

Wang, H., Huang, S., Frese, U. and Dissanayake, G. (2013). The nonlinearity structure of point feature SLAM problems with spherical covariance matrices, *Automatica* **49**(10): 3112–3119.

Wilkowski, A., Kornuta, T., Stefańczyk, M. and Kasprzak, W. (2016). Efficient generation of 3D surfel maps using RGB-D sensors, *International Journal of Applied Mathematics and Computer Science* **26**(1): 99–122, DOI: 10.1515/amcs-2016-0007.

Xu, B., Chen, Q., Zhu, J. and Wang, Z. (2010). Ant estimator

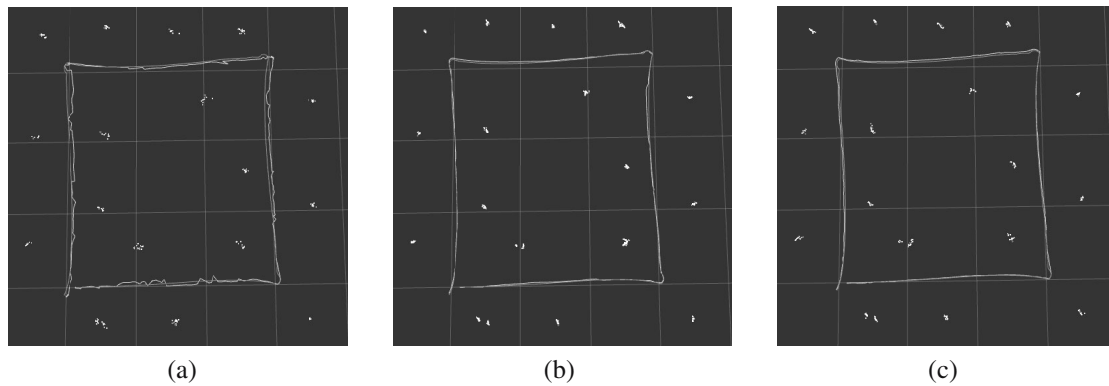


Fig. 14. Estimated results in a real landmark environment: PHD-SLAM (a), LMB-SLAM (b), ant-PHD-SLAM (c).

with application to target tracking, *Signal Processing* **90**(5): 1496–1509.

Xu, B., Xu, H. and Zhu, J. (2011). Ant clustering PHD filter for multiple-target tracking, *Applied Soft Computing* **11**(1): 1074–1086.

Zhang, H. (2015). Ant colony optimization for multimode resource-constrained project scheduling, *Journal of Management in Engineering* **28**(2): 150–159.

Zhang, Z., Gao, C., Lu, Y., Liu, Y. and Liang, M. (2016). Multi-objective ant colony optimization based on the *Physarum*-inspired mathematical model for bi-objective traveling salesman problems, *PLoS ONE* **11**(1): e0146709.

Zhu, J., Xu, B., Wang, F. and Wang, Z. (2010). A real-time moving ant estimator for bearings-only tracking, *Proceedings of the 1st International Conference on Advances in Swarm Intelligence, Beijing, China*, pp. 273–280.



Demeng Li was born in 1990. He received the BSc degree in electrical engineering from Qufu Normal University in 2013. He is currently pursuing his MSc degree in electrical engineering at the China University of Mining and Technology. His current research interests are in image processing, object tracking, simultaneous localization and mapping (SLAM), and related areas.



Jihong Zhu received his PhD degree in control and engineering from the Nanjing University of Science and Technology (NJUST), China, in 2012. His research interests include object tracking, optimization algorithms, simultaneous localization and mapping, etc.



Benlian Xu received his BSc degree in mechanical engineering at the Changsha University of Science and Technology in 1997, and MSc and PhD degrees, both in control science and engineering, at the Nanjing University of Science and Technology (NJUST), in 2003 and 2006, respectively. Currently, he is a professor at the School of Electrical and Control Engineering at the Changshu Institute of Technology. His present research interests are in image processing, swarm intelligence, object tracking, simultaneous localization and mapping (SLAM), optimization algorithms, and related areas.



Mingli Lu received her BSc degree in electrical engineering at the Nanjing University of Science and Technology in 1997, and her MSc degree in control technology and control engineering at the Lanzhou University of Technology in 2003. Currently, she is an associate professor of the School of Electrical and Control Engineering at the Changshu Institute of Technology (CIT), and is pursuing her PhD degree at the School of automation at the Nanjing University of Science and Technology (NJUST). Her present research interests are in image processing, object tracking, optimization algorithms and related areas.



Mingyue Li was born in 1987. He received his BSc degree in electrical engineering from the Nanyang Institute of Technology in 2012. He is currently pursuing his MSc degree in electrical engineering at the China University of Mining and Technology. His present research interests are in image processing, object tracking, and simultaneous localization and mapping.

Received: 6 June 2017

Revised: 30 October 2017

Re-revised: 7 March 2018

Accepted: 19 March 2018