Expert Systems With Applications 135 (2019) 181–193

Contents lists available at ScienceDirect

Expert Systems With Applications

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

An improved particle filter for mobile robot localization based on particle swarm optimization

Qi-bin Zhang, Peng Wang, Zong-hai Chen*

Department of Automation, University of Science and Technology of China, Hefei 230027, PR China

ARTICLE INFO

Article history:
Received 20 February 2018
Revised 4 June 2019
Accepted 5 June 2019
Available online 6 June 2019

Keywords:
Mobile robot
Global localization
Local pose tracking
Particle filter
Particle swarm optimization

ABSTRACT

As one of the most important issues in the field of mobile robotics, self-localization allows a mobile robot to identify and keep track of its own position and orientation as the robot moves through the environment. In this work, a hybrid localization approach based on the particle filter and particle swarm optimization algorithm is presented, focusing on the localization tasks when an a priori environment map is available. This results an accurate and robust particle filter based localization algorithm that is able to work in symmetrical environments. The performance of the proposed approach has been evaluated for indoor robot localization and compared with two benchmark algorithms. The experimental results show that the proposed method achieves robust and accurate positioning results in indoor environments, requiring fewer particles than the benchmark methods. This advance could be integrated in a wide range of mobile robot systems, helping to reduce the computational cost and improve the navigation efficiency.

1. Introduction

Along with the technological advancements in the field of mobile robotics, research interest in autonomous mobile robots has been increasing in the past decades. A diverse range of applications in rescue (Michael et al., 2014), mining (Ma & Mao, 2018), agriculture (Bengochea-Guevara, Conesa-Muñoz, Andújar, & Ribeiro, 2016), military (Miksik, Petyovsky, Zalud, & Jura, 2011) and civilian tasks (Choi, Lee, Viet, & Chung, 2017; Le, Phung, & Bouzerdoum, 2014; Song, Gao, Ding, Deng, & Chao, 2017) encourage researchers to carry out research works in mobile robotics. Self-localization is a prerequisite for successful deployment of an autonomous mobile robot since it identifies the robot's pose (position and orientation) as it moves in the environment. By providing an "absolute" position estimate to the map frame, robot localization is one of the critical issues for mobile robot systems and it is typically the foundation of a variety of tasks, including map building, autonomous navigation, mobile manipulation, target tracking, etc.

The mobile robot localization problem falls into two main categories: global localization (GL) and local pose tracking (re-localization) (Thrun, Burgard, & Fox, 2005). The local pose tracking problem assumes that the initial pose of the robot is already known, and it tries to keep track of the robot state over time. The GL problem is fundamentally different because no prior knowledge about the robot's position is available, hence the robot has to locate itself from scratch and reduce the ambiguities of pose estimates. Global positioning is crucial to provide an initial pose estimate of the mobile robot at startup for autonomous navigation, deal with the kidnapped robot problem (Thrun et al., 2005), relocate the robot in case of pose tracking failure, etc. However, since the initial robot pose estimates in many robot systems are specified by the users, the GL problem has received less attention and most studies have been focusing on the re-localization problem.

A number of localization techniques have been studied in the past decades, among which probabilistic methods based on Bayes Filters are the most common and well-proven approaches. The Bayes Filter recursively updates the conditional probability distribution over the state space of the robot to estimate its pose, such as the popular Extended Kalman Filter (EKF) (Jensfelt & Kristensen, 2001; Teslić, Škrjanc, & Klančar, 2011), Markov (Fox, Burgard, & Thrun, 1999), and Monte Carlo particle filter (Blanco, González, & Fernández-Madrigal, 2010; Thrun, Fox, Burgard, & Dellaert, 2001; Woo, Kim, Lee, & Lim, 2006). EKF based methods (Jensfelt & Kristensen, 2001; Teslić et al., 2011) are generally computationally efficient but the intrinsic property of EKF makes them inapplicable to the global localization problem since the position of the robot must be modeled by a unimodal Gaussian distribution. This limitation can be overcome by the multi-hypothesis Kalman Filters (Jensfelt & Kristensen, 2001) which represent the posterior distributions with mixtures of Gaussians, enabling them to describe

* Corresponding author.
E-mail addresses: zqb101@mail.ustc.edu.cn (Q.-b. Zhang), pwang@ustc.edu.cn (P. Wang), chenzh@ustc.edu.cn (Z.-h. Chen).

https://doi.org/10.1016/j.eswa.2019.06.006
0957-4174/

multi-modal probability distributions. But low-dimensional features have to be extracted from the raw sensor data to meet the Gaussian noise assumption. Grid-based Markov localization (Fox et al., 1999), in contrast, discretizes the whole state space into regularly spaced grids, each of which corresponds to a pose hypothesis. It can provide accurate global pose estimates with high robustness by maintaining an accurate probability density function over the whole state space with fine-grained grids, which may impose significant computational costs. Representing the posterior distribution by a set of samples or particles, the particle filter (PF) based approaches (Blanco et al., 2010; Wang, Wang, & Chen, 2018; Woo et al., 2006), also known as Monte Carlo localization (MCL) methods, are most widely used for robot localization since they could deal with nonlinear and non-Gaussian problems, and almost arbitrary distributions can be represented. Particle filter implementations do not rely on explicit feature extraction or data association as only a likelihood measure between the captured sensor data and the prior map for a given pose hypothesis is required. In order to generate pose estimates with acceptable accuracy and success rate, however, the GL problem requires a sufficiently large number of particles to cover the state space. In addition, the symmetry in the environment can prevent or delay convergence to the real robot pose of the particle set. Hence, MCL methods may perform poorly if the size of the sample set is small in a large environment and the complexity and memory cost of this kind of methods will grow significantly as the state space expands.

Due to the computational burden of the particles, maintaining a diverse sample set with fewer particles to represent the multi-modal posterior has become a hot topic in the study of MCL (Chien, Wang, & Hsu, 2017; Fox, 2003; Guan, Ristic, Wang, & Palmer, 2019; Liu, Shi, Zhao, & Xu, 2008). The KLD-sampling (Fox, 2003) is an elegant technique that allows adaptive sample set size, where the number of samples is adjusted dynamically based on the statistical bounds of the sample-based approximation quality at each step. But the derivation of the bounds of the sample size is unreasonable since the particles are sampled from an importance function rather than the true posterior distribution (Blanco, González, & Fernández-Madrigal, 2008). And it fails to reduce the cycle time during the global localization because the initial sample size is not reduced. Self-adaptive Monte Carlo Localization (SAMCL) incorporates off-line pre-caching and similar energy region (SER) technique with the conventional MCL algorithm, where a smaller number of samples is required since particles are distributed in SER instead of the whole environment (Zhang, Zapata, & Lépinay, 2012).

Scan matching technique (Bengtsson & Baerveldt, 2003; Segal, Haehnel, & Thrun, 2009; Wang, Zhang, & Chen, 2015) is another alternative for efficient robot localization since it provides fast, reliable and accurate pose estimation without taking care of the robot dynamics and can be used to improve the precision of probabilistic methods (Röwekämper et al., 2012). Although most scan matching approaches cannot be used to globally locate a mobile robot directly since they require an initial guess of the robot pose, they have been incorporated with particle filters to speed up global localization (Li, Yang, Guo, Wang, & Wang, 2016; Park & Roh, 2016; Zhu, Zheng, & Yuan, 2011). In the Coarse-to-Fine approach (Park & Roh, 2016), a set of SVM classifiers for different local places are trained offline to find candidate places where the robot may be located in the coarse localization stage, after which the relative robot pose for each candidate place is estimated with the fast spectral scan matching algorithm and then a number of particle filters in the potential regions are exploited to estimate the correct robot pose in the fine localization stage.

The particle swarm optimization (PSO), a population based stochastic optimization technique, has attracted wide attention to researchers because of its considerable success in solving non-differentiable, nonlinear and multi-modal problems. The PSO algorithm has been adopted for node positioning problem in wireless sensor networks (Kulkarni & Venayagamoorthy, 2010; Lavanya & Udgata, 2011; Manjarres et al., 2013; Nguyen, Guo, & Low, 2011) and the visual tracking problem (Walia & Kapoor, 2014; Zhang, Hu, Maybank, Li, & Zhu, 2008; Zhao & Li, 2010). A few studies have concentrated on the robot localization problem with PSO. The canonical PSO has been applied with global searching in the state space for the GL problem in Vahdat, NourAshrafoddin, and Ghidary (2007) and with local searching in a local area around the last well-known pose estimate to relocate the robot in case of it getting lost (Pinto, Moreira, & Costa, 2015; Pinto, Moreira, & Costa, 2013). The PSO algorithm can be combined with other localization techniques for better GL performance (Zhang, Wang, Bao, & Chen, 2017), e.g. the scan matching technique is performed to get accurate pose estimate when a sample gets near to the true robot pose in our previous work (Zhang et al., 2017). In addition, researchers have applied PSO to robot localization (Chien et al., 2017; Havangi, Nekoui, & Teshnehlab, 2010) and SLAM (Chen, Cai, & Yuan, 2009; Lee & Lee, 2009; Zhao, Wang, Qin, & Zhang, 2018; Zuo, Min, Tang, & Tao, 2018) with particle filters. After samples are propagated by motion prior, multi objective fitness functions (Chien et al., 2017; Havangi et al., 2010) and disturbances like quantum behaviors (Lee & Lee, 2009; Zuo et al., 2018) or random weights (Zhao et al., 2018) are introduced to maintain the diversity of the population and prevent a pre-mature convergence. However, the robot state may be ambiguous in some situations. In order to track multiple modes of the likelihood distribution, multiple swarms are introduced in (Lee & Lee, 2009) to address this problem. Close swarms are separated if the distance between them is greater than the threshold, otherwise they will complete with each other according to the birth and death strategy.

Motivated by the similarities of the particles in PSO and particle filters, this paper proposes a novel global localization approach, named Particle swarm Optimization Filter (POF), where sub-swarms corresponding to potential pose hypotheses are identified and maintained for accurate and robust global localization based on the distribution of the particles. As in most works, the proposed global localization approach consists of two separate stages: determining the initial robot pose and tracking the robot pose over time. In the first stage, a revised PSO approach with local search is adapted to optimize the uniformly distributed particles in the map and the DBSCAN algorithm is used to identify all the potential pose estimates with clustered particles. In the second stage, multiple pose hypotheses with corresponding sub-swarms are maintained to keep track of the robot trajectory, where the samples in each sub-swarm are propagated and optimized to move to high likelihood region in the posterior density function at each step. The main contributions of this work are the introduction of local search in a geometric neighborhood to the PSO algorithm to balance exploration and exploitation of the samples when determining the initial robot pose, and the integration of PF with the revised PSO algorithm for robust robot pose tracking during global localization. The POF algorithm inherits the advantages of both MCL and PSO algorithm, hence it is able to provide accurate initial pose estimates of the robot with multiple pose hypotheses, and perform precise and robust pose tracking even with great motion uncertainty.

The rest of the paper is organized as follows. In Section 2, the particle filter and PSO algorithm are briefly presented. In Section 3, the proposed POF algorithm for global localization of the robot is introduced. Experimental results carried on real robot data set are presented in Section 4. The last section concludes with some discussions.

2. Preliminaries

2.1. Particle filter

In Bayesian context, robot localization is the problem of estimating the state of a nonlinear dynamic system sequentially in time. The robot seeks to estimate a posterior distribution over the robot's state space conditioned on the available sensor data. More formally, the robot state at time t is denoted as x_t, the control is u_t and the observation is z_t. Then the dynamic system can be represented with a state transition model

$$x_t = f(x_{t-1}, u_{t-1}) + v_t \leftrightarrow p(x_t|x_{t-1}, u_{t-1}) \tag{1}$$

And an observation model

$$z_t = h(x_t) + \omega_t \leftrightarrow p(z_t|x_t). \tag{2}$$

where v_k and w_k are independent white noises, f and h are assumed to be known functions. In a probabilistic system model, the transition equation can be formed as the transition probability $p(x_t|x_{t-1}, u_{t-1})$ and observation likelihood probability $p(z_t|x_t)$, respectively.

As the environment map M is already known, M is omitted for simplicity in the following formulas. Based on the Bayes rules, the posterior probability density $bel(x_t) = p(x_t|u_{0:t-1}, z_{1:t})$ of the robot pose x_t is estimated in two consecutive steps:

(1) Prediction. The motion model $p(x_t|x_{t-1}, u_{t-1})$ is exploited to predict the current pose x_t of the robot in current step, given control input u_{t-1} and previous posterior probability $bel(x_{t-1})$. Under Markov assumption, the predictive density probability distribution (PDF) of the current robot state is obtained by following formulation:

$$\overline{bel}(x_t) = \int p(x_t|x_{t-1}, u_{t-1}) bel(x_{t-1}) dx_{t-1} \tag{3}$$

(2) Update. Incorporating sensor measurement into the predicted density $\overline{bel}(x_t)$ leads to the posterior PDF $bel(x_t)$ of x_t. Observation z_t is assumed to be conditionally independent of earlier measurements z_{t-1} given x_t. The posterior density over x_t is obtained using Bayes theorem:

$$bel(x_t) = \eta p(z_t|x_t) \overline{bel}(x_t) \tag{4}$$

where $\eta = p(z_t|z_{0:t-1})$ is the normalizing factor.

The posterior PDF is rather difficult to obtain and manage for a general mobile robot system. In particle filter implementations like MCL, the probability distribution is approximated by samples drawn from a target density function defined on the robot state space. The robot state x_t is represented by N i.i.d. particles: $S_t = \{x_t^i\}_{i=1}^N$, and the posterior density function is approximated by:

$$p(x_t|u_{0:t-1}, z_{1:t}) \approx \frac{1}{N} \sum_{i=1}^{N} \delta_{x_t^i}(x_t) \tag{5}$$

where $\delta_{x_t^i}(x_t)$ stands for the Dirac delta function centered in x_t^i. The denser are the samples in a particular area, the higher is the probability that the robot is located in that region. In the ideal case, samples are drawn from the posterior distribution $x_t^i \sim p(x_t|z_{0:t}, u_{0:t-1})$ and all individuals of the population have the same probability mass. This is unreasonable in general, however, because such a distribution cannot be calculated in a closed form.

In the Importance Sampling (IS) strategy, samples are drawn from a so called importance proposal distribution $q(x_t|z_{0:t}, u_{0:t-1})$, which is defined such that its support set includes the support set of the posterior distribution (Andrieu, De Freitas, Doucet, & Jordan, 2003). Therefore, the target distribution can be represented using a set of weighted samples $\langle x_t^i, w_t^i \rangle$, where the importance w_t^i is:

$$w_t^i = \frac{p(x_t^i|z_{0:t}, u_{0:t-1})}{q(x_t^i|z_{0:t}, u_{0:t-1})}. \tag{6}$$

By choosing the transition distribution $p(x_t|x_{t-1}, u_t)$ as proposal distribution, the weights can be calculated iteratively:

$$w_t^i = \eta \frac{p(z_t|x_t^i) p(x_t^i|x_{t-1}^i, u_{t-1})}{p(x_t^i|x_{t-1}^i, u_{t-1})} \propto p(z_t|x_t^i) \tag{7}$$

Although the transition model is usually used as the proposal distribution, it is unreasonable when $p(x_t|x_{t-1}^i, u_t)$ lies in the tail of the likelihood $p(z_t|x_t^i)$. In fact, it has been shown that the optimal proposal distribution is $p(x_t|x_{t-1}, u_{t-1}, z_t)$. The weights are normalized such that $w_t^i = w_t^i / \sum_i w_t^i$. The posterior probability density is now approximated by:

$$p(x_t|u_{0:t-1}, z_{1:t}) \approx \sum_{i=1}^{N} w_t^i \delta_{x_t^i}(x_t) \tag{8}$$

If the samples are generated from a sub-optimal proposal, the weights of most samples will approach zeros after a few iterations. The most classical way of dealing with the impoverishment problem is the Sequential Importance Resampling (SIR) strategy (Rubin, 1988). Particles with negligible weights are removed from the particle set and replaced with more likely ones. After resampling, all particles in the population have the same weights.

2.2. Canonical particle swarm optimization

A canonical PSO (Kenndy & Eberhart, 1995) algorithm is a global optimization meta-heuristics inspired by the collective movement of birds flocks and has been successfully applied to many applications. In PSO, a set of N particles (potential solutions) moves inside a bounded D-dimensional search space, realizing a joint effort to identify the optimal solution for a given problem.

The ith particle is characterized by two vectors: the position x in the search space and its velocity v. The speed of particle i at iteration k is adjusted as a result of three components: the inertia of the particle at last iteration $k-1$; the cognition term of the particle, which is based on the personal best solution $pbest^i$ found so far by the particle itself; and the social term which is based on the global best solution $gbest$ identified by the whole swarm. The new velocity of this particle at iteration k is:

$$\begin{aligned} v^{i,k} = {} & w v^{i,k-1} + c_1 R_1 \otimes (gbest^{k-1} - x^{i,k-1}) \\ & + c_2 R_2 \otimes (pbest^{i,k-1} - x^{i,k-1}) \end{aligned} \tag{9}$$

where w is inertial weight, c_1 and c_2 are positive acceleration constants, R_1, R_2 are random sequences sampled from a uniform probability distribution U(0,1) and $\otimes$ denotes component-wise multiplication operator.

The inertia weight w affects whether the particle trends toward exploration in the search space or convergence (Peer, van den Bergh, & Engelbrecht, 2003). The social component drives the particle towards the global best position of the swarm while the cognitive attractor drives the particle towards the personal best position. In order to better explore the multi-dimensional space and avoid velocity diverging, the velocity of particles is clamped to a maximum value $v_{\max}$ for each dimension (Poli, Kennedy, & Blackwell, 2007):

$$|v_{ij}| \le \varepsilon (x_{UB} - x_{LB}) \tag{10}$$

where X_{LB} and X_{UB} are the lower and upper bounds of the solution space with respect to dimension j and $\varepsilon \in [0, 1]$.

Then the position of each particle is updated:

$$x^{i,k} = x^{i,k-1} + v^{i,k}, i = 1, \ldots, N \tag{11}$$

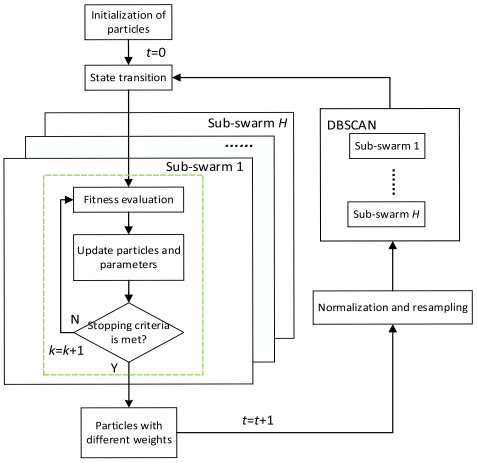

Fig. 1. The framework of POF localization.

The effectiveness of each candidate solution to the optimization problem is evaluated by a so-called fitness function $\zeta(\bullet)$. The individual best solution of particle x^i at iteration k is determined as follows:

$$pbest^{i,k} = \begin{cases} x^{i,k} & \text{if } \zeta(x^{i,k}) > \zeta(pbest^{i,k-1}) \\ pbest^{i,k-1} & \text{else} \end{cases} \tag{12}$$

In the canonical PSO, the global topology allows particles to exchange information of the best solution with all the other particles, thus the whole population shares the same global best solution *gbest*, which can be found by:

$$gbest = \arg\max_{pbest^i} \zeta(pbest^i) \tag{13}$$

Relying on its learning strategy to adjust the search direction, the whole population in the PSO evolves from iteration to iteration until the convergence criterions are met. In general, the convergence criterions can be a maximum number of iterations or a minimum error.

3. Proposed algorithm

MCL methods are effective for re-localization problem where the area to be sampled is small, but they may perform poorly for global localization where a great number of particles are required to represent the evolved multi-modal distribution. In this section, a novel Particle swarm Optimization enhanced particle Filter (POF) for robot localization is proposed. The flowchart of the POF framework for robot localization is schematically shown in Fig. 1 to give a clear view of the proposed approach. We first analyze the occupancy grid map representation of the environment and present an efficient fitness value computing method that measures the similarity between mapped range scan with the environment model. Then the POF based localization approach is introduced. Different from MCL, the proposed global approach consists of two separate stages: finding the initial robot pose estimates and tracking the robot poses over time until convergence.

3.1. Fitness function

In this paper, the working environment is represented by an occupancy grid map, which is the most typical type for environment modeling. The full environment map can be defined by the following set:

$$M = \{m_{i,j} | p(m_{i,j}) \in [0,1], 1 \le i \le n_c, 1 \le j \le n_r\} \tag{14}$$

where n_c and n_r are the map dimensions that describe the environment size with the map resolution Δm, and i and j are the indices of grid cell $m_{i,j}$. The probability of grid cell $m_{i,j}$ being occupied by obstacles is given by the probability $p(m_{i,j})$, and thus cells in a grid map can be classified into three states: occupied space (M_o), free space (M_f) and unknown space (M_u), as shown in Fig. 2(a).

In order to locate itself, the mobile robot is equipped with a 2D laser range finder. The laser scans the environment, and captures a laser scan $z = \{r_i, \phi_i\}_{i=1}^{N_s}$ in the local coordinate, where the ith reading $\{r_i,\phi_i\}$ denotes a measured distance r_i to the nearest obstacle at the given bearing ϕ_i. One of the laser beams from the robot to the nearest obstacle projected on the grid map is shown in Fig. 3.

Many algorithms compare laser scan captured from the robot pose to laser data generated with the ray casting operation from the candidate solution, which is very time-consuming when the sample set size is huge. In order to reduce the online computational burden of ray casting, we only consider the local uncertainty of a laser measurement. The Euclidean distance of a measured point o to the nearest obstacle in the working space is:

$$d(o, M) = \min_{m_{i,j} \in M_o} \left\| o - m_{i,j} \right\|, o \in M \tag{15}$$

According to the Euclidean distance, the corresponding fitness value of point o can be defined by a signed function shown in Fig. 4, where σ_d is used for error limitation. The nearer the point is to the closest obstacle or wall, the higher the fitness value is. And if the measured point is too far away from the nearest obstacle, the fitness value is negative, which implies that the laser point is unexpected to be located at that position. Considering all the grid cells in the environment map, a fitness map can be defined as shown in Fig. 2(b), where gray level of the image represents the fitness value. With the fitness map, the fitness value of a sample can be calculated effectively.

Ideally, the fitness value of a laser point collected by the robot at the true location is expected to be 1, and the vector of the expected fitness values of the corresponding laser scan is denoted by E. On the other hand, the fitness of the ith mapped laser point *w.r.t.* particle x in the map frame is F_i and the corresponding fitness vector with the laser scan z is denoted by F. The similarity function sim(E,F), also defined as the fitness $\zeta(x, z)$ of this particle, returns a scalar value that represents the similarity between vectors E and F. Maximizing the fitness function gives the robot pose that best fits the current sensor data with the environment. In our experiments, we use the cosine similarity which is robust to outliers and noises:

$$\text{sim}(E, F) = \frac{\sum_{i=1}^{N} E_i F_i}{\sqrt{\sum_{i=1}^{N} E_i^2} \sqrt{\sum_{i=1}^{N} F_i^2}} \tag{16}$$

The fitness function evaluates the similarity between captured laser data from the true robot pose and the environment model around the pose estimates. An illustration of the fitness value with respect to the true robot pose in Fig. 13(a) is shown in Fig. 5.

3.2. POF based initial pose estimation

3.2.1. Problem formulation

For a real robot system, we usually need to find the initial robot pose x_t at time $t = 0$ from scratch, based on the initial observation z_0. The GL problem can be formulated as a minimization problem:

$$\begin{aligned} &min \ \ -\zeta(x_0, z_0), \\ &s.t. \ \ x_0 \in M_f \times [-\pi, \pi) \subseteq R^3 \end{aligned} \tag{17}$$

In order to cover the whole state space, the only way for a generic particle filter is increasing the sample set size which introduces high computational cost. On the contrary, POF uses a set of

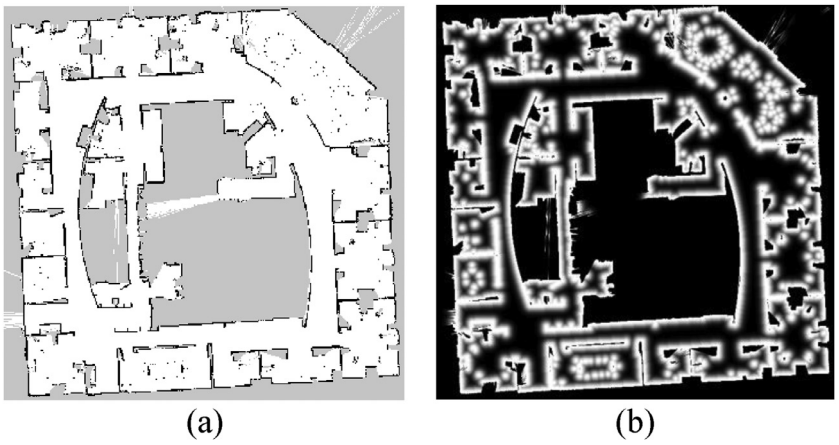

Fig. 2. (a) Occupancy grid map of the Intel Research Lab, where white, gray and black pixels represent free space, unknown space and occupied space, respectively. (b) Fitness map of the Intel Research Lab. White pixels represent obstacles and the darker pixels mean higher distance.

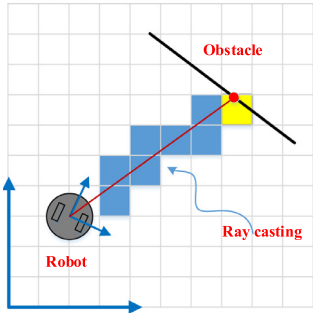

Fig. 3. Illustration of a range reading. The red dot denotes the end of the laser beam, and corresponding grid cell is marked yellow. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

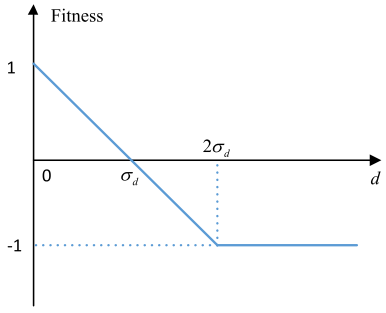

Fig. 4. Truncated fitness function.

samples to search for the global optimum in the free space of the environment, thus the number of particles can be reduced. However, global localization is a typical multimodal problem since the robot pose is usually ambiguous in a symmetrical environment and multiple optimal solutions may exist at the same time. Hence the global topology of conventional PSO is not appropriate for this kind of problem. A local topology allows a particle to exchange information with a subset of the particles and guides particles to the potential search region. Several local topologies (Janson & Middendorf, 2003; Kennedy, 1999) based on logical neighborhood structure have been proposed to gain beneficial experience from their neighbors. Nevertheless, neighbors based on logical topology are only the social neighbors rather than the true neighbors in the solution space. Neighborhood structure based on Euclidean spatial topology is less studied, maybe due to that it will introduce more computational cost for each particle to find its geometric neighbors. But it is possible that each particle might attain more beneficial and effective information from its Euclidean spatial neighbors for the localization problem.

In the proposed global localization approach, a modified PSO algorithm with Euclidean spatial neighborhood is introduced to find the global pose estimates of the robot at startup. Particles in the swarm are responsible for exploration of the free space of the environment, and the detection of the potential regions with optimal solutions. A particle with its geometric neighbors searches for the best solution in its neighborhood. After reaching the maximum of iterations, each particle has its own best knowledge about the best solution in a local region. With resampling, samples in

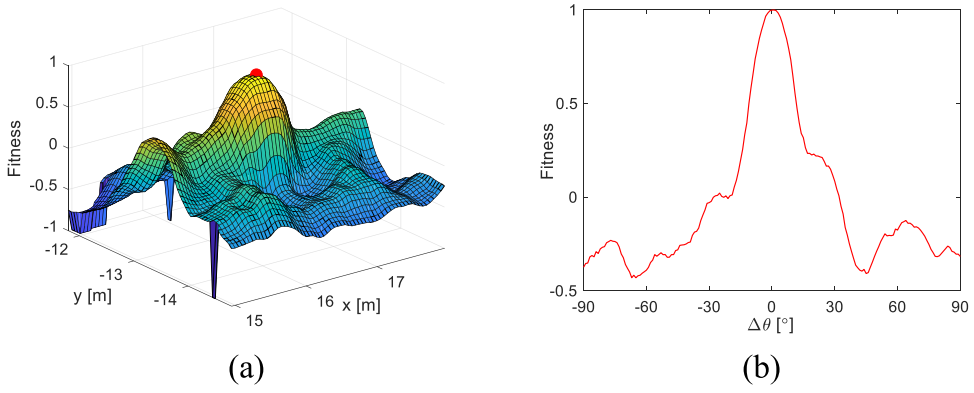

Fig. 5. Fitness values of potential poses. (a) The fitness value around the true robot pose when the orientation is fixed, where the red dot is the true robot pose. (b) The fitness value changes as the relative orientation varies when the robot position is fixed. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
POF based initial pose estimation.

Algorithm 1. POF based initial pose estimation
Input: Environment Map M, observation z_0
Step 1. Initialization
Generate uniformly distributed particles $S_0 = \{x_0^i\}_{i=1}^N$ in the free space
Step 2. Evolutionary search
(a) Evaluate each particle's fitness
(b) Update each particle's personal best solution *pbest* and local best solution *lbest*
(c) Update particle's velocity using Eq. (20) and position using Eq. (11)
(d) If $x^i \notin M_f$
Draw x^i with the uniform distribution in M_f
Step 3. Normalization and Resampling
(a) Calculate importance factor $\tilde{w}^i$ for the particle set
(b) Normalize weights and resampling
Step 4. Clustering
Grouping the particles into clusters with DBSCAN, generating multiple pose hypotheses $S_0 = \{S_0^h\}_{h=1}^H$
Output: Best estimate of the robot pose $\hat{x}_0$ with highest weight at $t=0$

wrong places are eliminated from the swarm and samples with high probabilities are replicated. The DBSCAN algorithm is then employed to identify all the potential pose hypotheses: the whole swarm is clustered into sub-swarms, each of which corresponds to a candidate pose estimate at that place. The proposed GL algorithm is presented in Table 1.

3.2.2. Initial pose estimation

In order to find the initial robot pose hypotheses, the following steps are performed. The computational time to obtain an optimized particle set is higher than that of the generic particle filter because we have to calculate the fitness values of particles iteratively. However, since it is possible to find an accurate estimate of the global robot pose with a smaller number of particles than the generic PF, the overall performance has been improved.

Step 1: Initialization

Samples in the proposed approach are uniformly distributed on the 2D plane with fixed distance interval instead of sampling from a uniform distribution over the state space, which is adopted by most MCL methods. The regular distance ΔD between adjacent particles decides the density of the sample set, i.e. $1/\Delta D^2$ particles are expected per square meters in the free space. The initial particle set at $t=0$ is supposed to be $S_t = \{x_t^{i,k}\}_{i=1}^N$ for iteration k with N samples in total. Each sample in the swarm is described by a 3-tuple $\langle x, v, \pi \rangle$, where x is the potential robot pose, v is the particle velocity and $\pi = \zeta(x, z)$ is the fitness value of the particle with mapped observation z. Note that explicit reference to the time step t will be omitted during PSO process from now on for notational convenience.

Step 2: Evolutionary search

The nearest neighbor rule based on the Euclidean distance between particles is used to determine a particle's neighborhood. In the proposed GL approach, the subset is defined as the m nearest particles within a perception area whose radius is given by δ (shown in Fig. 6). A mathematical expression is given by:

$$N_i(k) = \{j, j = 1, ..., m || |x^{i,k} - x^{j,k}| \le \delta_i\} \tag{18}$$

where $N_i(k)$ denotes the set of neighbors of particle i at iteration k. The perception radius can be the same or different for all the particles. Then the local best solution for particle i to follow is defined by:

$$lbest^{i,k} = \underset{x^j \in N_i(k)}{\arg\max}\, \zeta(x^j, z) \tag{19}$$

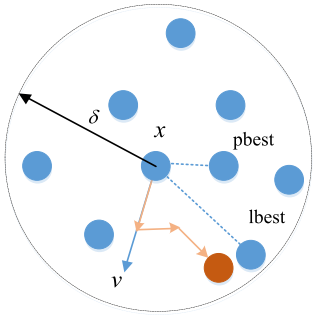

Fig. 6. Illustration of the spatial neighborhood of a particle.

In addition, work in Clerc and Kennedy (2002) indicated that a constriction factor helps to ensure convergence, which leads to a modified velocity update equation:

$$v^{i,k} = \chi\left[v^{i,k-1} + c_1 R_1 \otimes (pbest^{i,k-1} - x^{i,k-1}) + c_2 R_2 \otimes (lbest^{i,k-1} - x^{i,k-1})\right] \tag{20}$$

where the constriction factor

$$\chi = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \tag{21}$$

and $\varphi = c_1 + c_2$, $\varphi > 4$. Usually c_1 and c_2 are both set to 2.05, which yields $\varphi = c_1 + c_2 = 4.1$ and $\chi = 0.7298$. Although it was originally suggested that the velocity clamping is unnecessary due to the presence of the constriction factor, it has been illustrated that the constriction factor alone does not necessarily result in the best performance and these two approaches should be combined (Eberhart & Shi, 2000).

For each individual of the swarm, the velocity-position update process is repeated until a new population $\{x_t^{i,k+1}\}_{i=1}^N$ of the swarm is generated. Furthermore, the particles should not get outside of the map boundary under any circumstances, and a randomly positioning process is called to reset particle's pose. Therefore, if a particle moves to invalid regions, the particle's pose is reset which introduces more randomness to the search procedure. The optimization process is repeated until the iteration has reached its maxima or the best solution has been stuck for several iterations.

Step 3: Normalization and resampling

The likelihood of observation for ith particle is defined as:

$$p(z|x^i) = \frac{1}{\sqrt{2\pi}\sigma_\pi} \exp\left\{-\frac{(\pi^i - 1)^2}{\sigma_\pi^2}\right\} \tag{22}$$

The weights of the particles are normalized:

$$\tilde{w}^i = \frac{1}{N} p(z|x^i),\ w^i = \tilde{w}^i / \sum_{i=1}^N \tilde{w}^i \tag{23}$$

The number of effective particles is (Doucet, Freitas, & Gordon, 2001):

$$Neff \simeq \frac{1}{\sum (w^i)^2} \tag{24}$$

If *Neff* is less than the threshold, the weights of some particles can be neglected and these particles are discarded through resampling.

During the PSO iteration, particles near to the local optima are attracted to the high likelihood area while particles in other regions are dispersed in the free space of the environment. After resampling, particles that are unlikely to be the correct robot pose are eliminated, and particles whose surrounding is similar to the

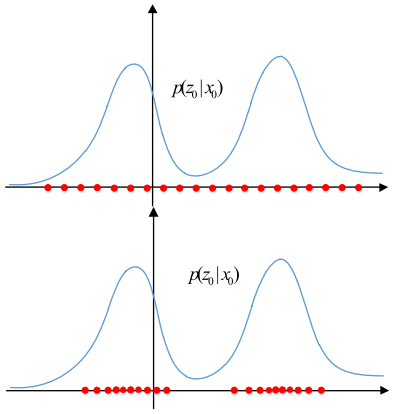

Fig. 7. Samples before (upper) and after (lower) optimization.

true laser scan are replicated. Finally, all the samples are relocated to the dominant modes of the likelihood, which leads to high particle density around local extrema (shown in Fig. 7).

Step 4: Clustering

Multiple potential solutions for the GL problem may exist simultaneously due to the symmetry of the environment, which can be distinguished from each other by the density based spatial clustering technique DBSCAN (Ester, Kriegel, Sander, & Xu, 1996).

The distance between sample x^i and x^j in the solution space is defined as:

$$d_{ij} = \sqrt{(x^i - x^j)W(x^i - x^j)^{\mathrm{T}}} \tag{25}$$

where $W = diag(1, 1, \lambda)$ is used to weight each dimension of the samples. In our experiment, λ is set as 1. The maximum radius of the neighborhood for clustering is set as Eps = 1 and the minimum number of samples to form a dense region is set as MinPts = 5. After clustering, we get several sub-swarms (pose hypotheses) distributed in the 2D planar $S_0 = \{S_0^h\}_{h=1}^H$.

3.3. POF based multiple pose tracking

After initialization of the robot pose at startup, a sample set with H subsets (sub-swarms) is obtained, each of which corresponds to a potential robot pose. Some of them can be eliminated with more accurate measure of the similarity. However, due to the presence of repeated structures with similar appearances in the environment, the state of the robot can be still ambiguous. On the other hand, the diversity of the particle population may disappear quickly due to random drift in sampling for a generic particle filter, which results a failure of global localization. In order to overcome these limitations of particle filters, PSO is integrated with PF in the proposed approach to optimize the sample set since they have similar characteristics.

In order to overcome the limitations of particle filter, PSO is integrated with PF in the proposed approach to optimize the sample set since they have similar characteristics. In the proposed approach, particles in a subset try to find the local best pose estimate at each step for the associated pose hypothesis as the robot moves. When more distinguishable measurements are captured, wrong pose estimates are eliminated through resampling until the particle set converges to a local space. Owing to the improved diversity preserving ability, POF is more robust than the standard MCL approach for global localization in ambiguous environments.

The proposed multiple pose tracking method for global localization is summarized in Table 2, which consists of three main steps in each cycle. It can be noted that when the swarm converges to one pose hypothesis, the proposed approach turns into a pure pose tracking method.

Table 2
POF based multiple pose tracking.

Algorithm 2. POF based multiple pose tracking

Input: Particle set at last step $S_t = \{S_t^h\}_{h=1}^H$, control u_t, observation z_{t+1}, and environment map M
Step 1. Updating the particle set
  For $i = 1$ to N do
  (a) Draw $x_{t+1}^i \sim p(x_{t+1}^i | x_t^i, u_t)$
  (b) Compute fitness values $\pi_{t+1}^i = \zeta(x_{t+1}^i, z_{t+1})$
Step 2. Evolutionary search
  For each sub-swarm S_{t+1}^h
  (a) Determine the maximum velocity $V_{\max}$ of the particles
  (b) While stopping criterion is not met
    Update the personal best *pbest* and global best *gbest*
    Update particle's velocity using Eq. (26) and pose using Eq. (11)
Step 3. Normalization and resampling
  (a) Compute importance weights: $\tilde{w}_{t+1}^i = w_t^i p(z_t | x_{t+1}^i)$
  (b) Normalize importance weights: $w_{t+1}^i = \tilde{w}_{t+1}^i / \sum_{i=1}^N \tilde{w}_{t+1}^i$
  (c) If *Neff* is less than threshold T, do
    Draw x_{t+1}^i with distribution w_{t+1}^i
    Add x_{t+1}^i to S_{t+1}
Output: State estimate $\hat{x}_{t+1}$ with the highest weight

Step 1: Particle set update

Based on the control command u_t and particle set $S_t = \{S_t^h\}_{h=1}^H$ at last time step, a new sample set $S_{t+1} = \{S_{t+1}^h\}_{h=1}^H$ is constructed at the next time step $t+1$ according to the robot's motion model. In other words, the positions of particles in different sub-swarms are updated.

Step 2: Evolutionary search

In the conventional MCL methods, each particle tacks its personal pose as the robot progresses. In the proposed approach, however, particles in each sub-swarm track the robot's pose in the corresponding hypothesis. When a new observation is obtained, particles in a sub-swarm are optimized to move towards high likelihood region. The survivor selection mechanism chooses the best solution to form the population of the next generation. In each sub-swarm, the particles are attracted to move to the global best solution with the highest fitness value of this group. Hence, the velocity update formula for ith particle in hth sub-swarm is defined by:

$$v^{h_i,k+1} = \chi\left[v^{h_i,k} + c_1 R_1 \otimes (pbest^{h_i,k} - x^{h_i,k}) + c_2 R_2 \otimes (lbest^{h,k} - x^{h_i,k})\right] \tag{26}$$

where $lbest^{h,k}$ stands for the global best solution found in hth sub-swarm at iteration k. As in Eq. (26), particles in each subset are attracted to local regions with high probabilities until the overlap between observation and the environment is maximized for the best particle, which is the optimal estimate for each pose hypothesis at current time step.

With the optimization process, most particles in each subset gather around regions with high likelihood. As a result, the impoverishment can be avoided to a great extent. And since the effect of each particle is enhanced, the number of required samples for global localization is decreased.

Step 3: Normalization and resampling

As the robot moves, the weights of particles in incorrect places reduce rapidly and the number of effective particles may be small. With resampling, particles with negligible weights are discarded and particles with high weights are replicated. Therefore, unlikely pose hypotheses together with sub-swarms are eliminated gradually from the global localization results. Meanwhile, the pose hypothesis near the true robot pose is maintained and enhanced. This

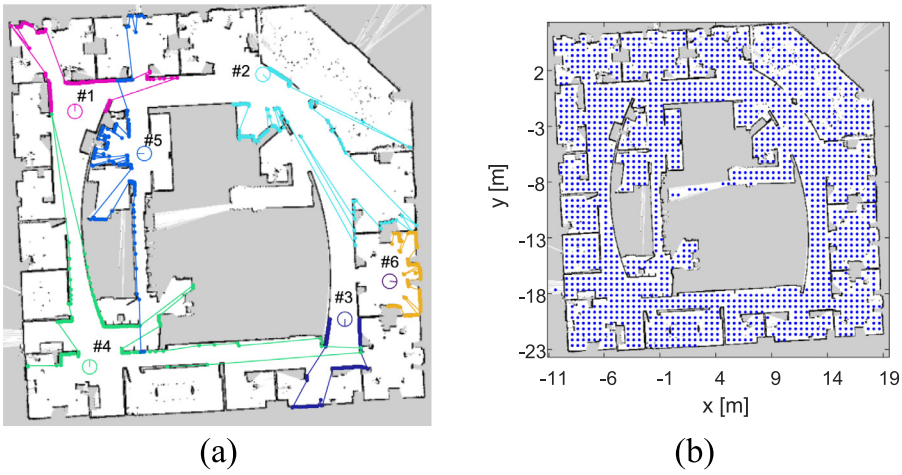

Fig. 8. (a) Selected different poses and corresponding laser scans for global localization tests. (b) Generated particle set at startup for global localization.

process is repeated until the particle swarm converges to the real robot pose. By maintaining multiple pose hypotheses using sub-swarms, we can robustly track the robot's state in the environment where similar places exist.

4. Experimental results

In order to validate the presented POF based localization algorithm, experiments were conducted on the public data set of Intel Research Lab and Fort AP Hill (Andrew & Nicholas, 2003). The size of the Intel Research Lab is about 30 m × 30 m that includes a number of noisy offices, some of which have a similar appearance. The 2D occupancy grid map (with resolution $\Delta m = 5$ cm) of the environment is built with the SLAM technique, where an accurate estimate of the robot's pose over time is obtained. The whole map contains 601 × 611 = 367,211 cells and the area of free space is about 535 m^2. Based on the grid map, the robot was carried to various positions and asked to estimate where it is. The true robot poses at various test positions were compared with estimated results for both POF approach, the standard MCL approach and the SAMCL approach (Zhang et al., 2012) to validate the performance of the proposed approach. Three experiments were conducted. The first evaluated the performance of the proposed approach in finding the initial robot pose. The second evaluated the accuracy of local pose tracking, i.e. tracking a single pose hypothesis. Finally, the third evaluated the overall performance of global localization in both Intel Research Lab and Fort AP Hill.

4.1. Initial pose estimation

The maximum number of iteration during global localization is set as 10. As shown in Fig. 8(a), 6 different positions of the map with corresponding range scans are selected for global localization tests. In order to cover the free space of the environment uniformly, the distance interval between adjacent samples is set as $\Delta D = 0.5$ m and the orientations of the samples are generated uniformly in the range $[-\pi, \pi)$. The generated particle set for global localization is shown in Fig. 8(b), where there are 1735 particles in total.

Then the particle swarm moves through the free space of the environment to search for the correct robot pose from iteration to iteration. The localization results are summarized in Table 3. The position error (e_d) is the distance between the estimated position and the real position of the robot, and the orientation error (e_θ) is the difference between the estimated orientation and the true orientation. A success match is defined if the position error between the global best estimate and the true robot pose is lower than 10 cm and orientation error is lower than 2° in the experiment. The success rate is equal to the number of successful localization divided by the total number of trials.

Table 3
Global localization results at typical locations.

Robot's pose	e_d [cm]	e_θ [°]	Success (%)
# 1	3.89 ± 1.09	0.04 ± 0.08	100
# 2	4.51 ± 0.18	0.01 ± 0.02	100
# 3	3.49 ± 0.75	−0.03 ± 0.17	100
# 4	3.94 ± 0.41	0.02 ± 0.05	100
# 5	3.90 ± 0.54	−0.06 ± 0.06	100
# 6	3.52 ± 0.45	−0.14 ± 0.16	92

According to Table 3, the average position error is lower than the map resolution (5 cm) and the average orientation error is lower than 0.2° for different cases. If the robot pose is unique (pose #1–5), the proposed GL algorithm is able to accurately locate the robot while the success rate may decrease if the robot pose is ambiguous (pose #6). There are multiple places with similar appearance for pose #6, i.e. the initial pose is ambiguous in the symmetrical environment, and the success rate is 100% if considering all the founded pose hypotheses.

The distance ΔD is an important parameter and the performance of the proposed approach against ΔD is evaluated at pose #1 for simplicity. Fig. 9(a) illustrates how the localization error changes as the iteration number increases when $\Delta D = 0.5$ m. The algorithm behavior as the distance increases can be observed in Fig. 9(b). When the maximum number of iterations is set to 6, the percentage of successful localization against the distance is represented by the solid line in Fig. 9(b). On the other hand, the average number of iterations used for successful localization against the distance is denoted by the dashed line. As the distance grows, the percentage of success decreases and the number of mean iterations for successful localization is increased.

In addition to the case study discussed above, the accuracy of pose estimation was evaluated as well. The experiments were performed by simulation, i.e. robot poses were randomly generated on the grid map and corresponding laser range scans for localization tests were generated by ray casting algorithm. Thus we can compare the localization results with the ground true values. For this experiment, a total of 300 different positions (shown in Fig. 10(a)) were chosen from the whole environment map. Fig. 10(b) shows error distribution diagrams, where the median and mean errors

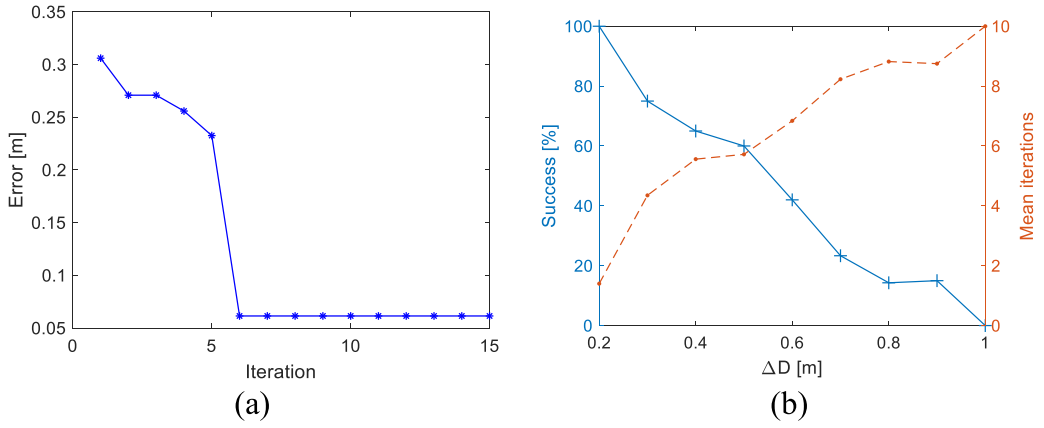

Fig. 9. Global localization results. (a) GL error vs. iteration. (b) Success rate and mean iterations vs. particle distance.

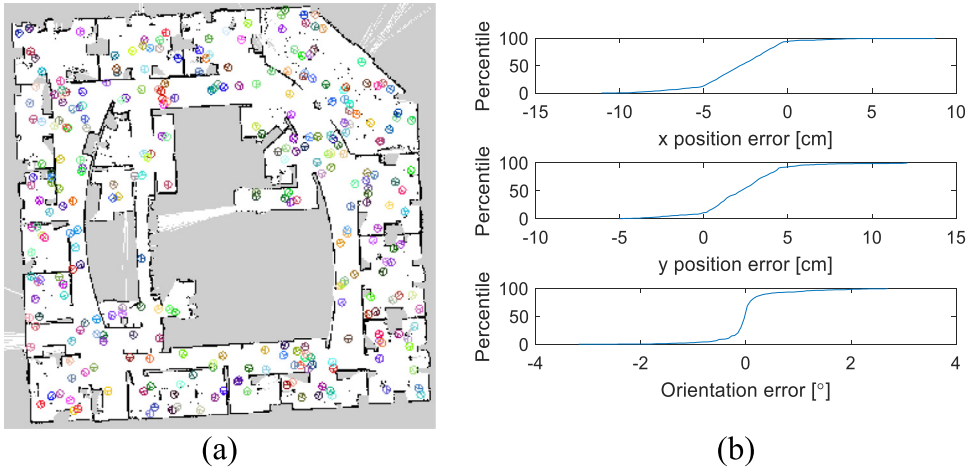

Fig. 10. (a) Different test locations for global localization in the grid map of Intel Lab. (b) Error distributions for the localization results.

were (2.88 cm, 2.51 cm, 0.05°) and (3.08 cm, 2.66 cm, 1.40°), respectively.

4.2. Pose tracking with one hypothesis

The second experiment evaluated the local pose tracking performance of the proposed localization approach. As the initial robot pose was already known, only one pose hypothesis would be maintained. We used 300 particles in MCL and 50 particles with a maximum number of 6 iterations in the proposed approach. As shown in Fig. 11, the robot kept track of 3 trajectories separately. In trajectory #1, the robot walked out of the room and turned left to walk along the corridor. In trajectory #2, the robot moved along the corridor without turning. In trajectory #3, the robot changed its heading direction and moved from one corridor to another one.

Comparative results for the proposed POF method and the standard MCL method are summarized in Table 4, and the error curves for trajectory #1 are shown in Fig. 12. The current version of the algorithm is implemented with MATLAB on a computer with a 2.5 GHz Intel Core i5 processor. The average computational time required for MCL and POF for each step is 65 ms and 96 ms, respectively. According to Table 4, the mean errors of the proposed approach were less than 5 cm (map resolution) in position and 0.2° in orientation which demonstrates the satisfactory performance of the proposed localization algorithm. As shown in Fig. 12, the orientation error remains approximately 0° and the position error remains approximately equal to the map resolution for POF during the movement of the robot. As the results suggest, 300 particles

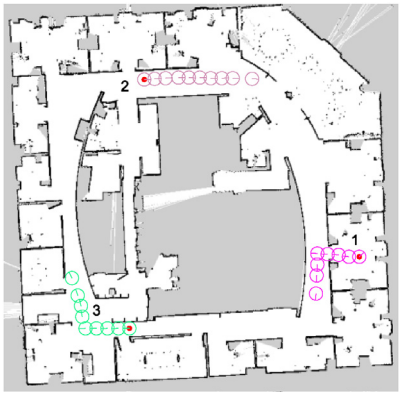

Fig. 11. Trajectories used for pose tracking tests. The starting points of the trajectories are indicated by red points. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 4
Pose tracking error comparison. Errors in mean ± standard deviation.

Trajectory	e_d [cm]	e_θ [°]
#1 MCL	11.38 ± 4.75	1.53 ± 0.99
POF	3.93 ± 1.72	0.11 ± 0.10
#2 MCL	19.06 ± 5.50	1.10 ± 0.78
POF	3.88 ± 1.82	0.07 ± 0.07
#3 MCL	14.86 ± 6.64	1.67 ± 1.31
POF	3.99 ± 1.44	0.07 ± 0.06

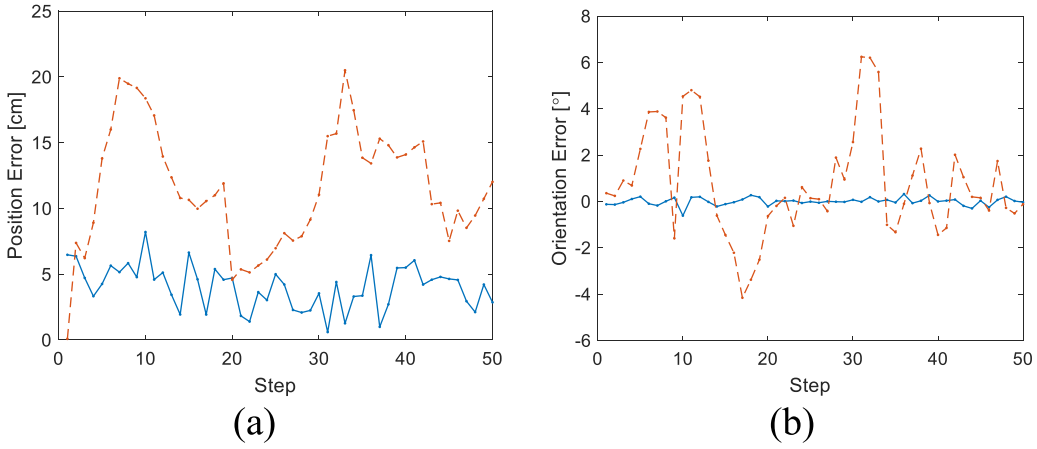

Fig. 12. Pose tracking error for trajectory #1. (a) Position error curve. (b) Rotation error curve. The solid line and the dashed line indicate the proposed POF method and the MCL method, respectively.

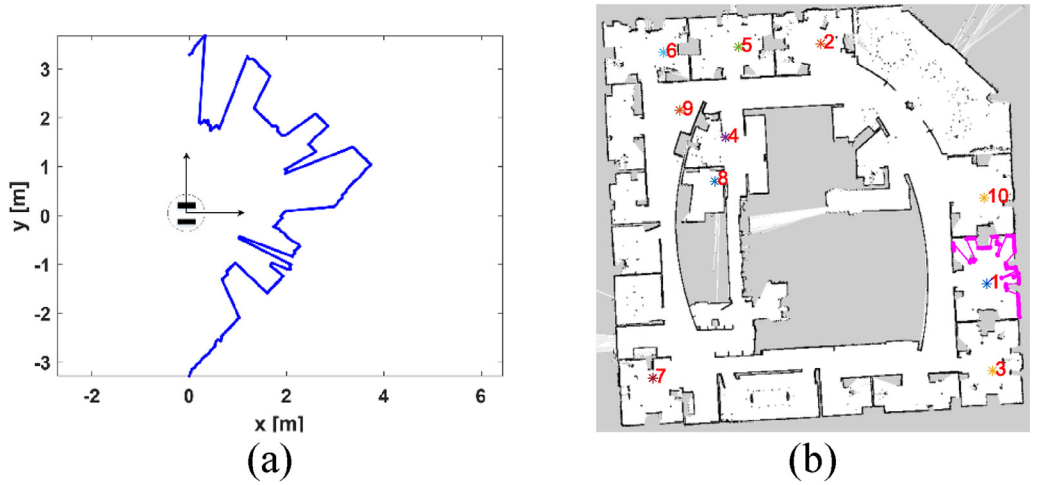

Fig. 13. (a) Input range scan for the global localization test. (b) Top 10 potential pose hypotheses.

were not sufficient to cover the uncertainty area for MCL when the robot pose had a great change, and the pose tracking error of MCL is greater than that of POF.

4.3. Global localization

As discussed above, the robot pose may be ambiguous in symmetrical environment that has more than one place with similar appearance, as the case of pose #6. In order to find the true robot pose, the robot has to move around to take more information of its surrounding to eliminate incorrect estimates until the swarm converges to the correct position. For a typical pose $(x, y, \theta) = (16.4644\,\text{m}, -13.3241\,\text{m}, 0.7757\,\text{rad})$ from where we started the global localization test, the associated range scan is shown in Fig. 13(a).

The identified top ten positions that obtained high fitness values are shown in Fig. 13(b). Among all the candidate positions, position #1 gained the largest fitness value. Furthermore, the distance between the ground truth and the global estimate of the robot pose was smallest, which implied that the robot was nearest to position #1 and the input range scan matched closely with the grid map at this position.

Fig. 14 illustrates the evolution of generated sample set over time for the proposed approach, MCL method and SAMCL method during global localization. The first row shows the performance of the proposed method where the identified pose hypotheses are marked in different colors after initial pose estimation. In the second row the performance of the conventional MCL method is shown, in which 10,000 samples were uniformly distributed around the free space of the environment. In the third row, 10,000 particles were sampled in SER rather than the entire map for SAMCL. The average computational time for POF, MCL and SAMCL during pose tracking for each step is 2.41 s, 2.21 s and 2.23 s, respectively. As shown in Fig 14, all the three approaches identified the robot's correct localization in the end. However, MCL failed to consistently determine the correction position as the robot moves, while POF was able to maintain multiple pose hypotheses that provided accurate and robust pose estimation results. SAMCL performs better than MCL since pose samples were initialized in regions that have similar energy to the robot's surrounding. The average position errors for three methods at each time step are shown in Fig. 16(a).

We then conducted the global localization experiments in the Fort AP Hill and the evolution of the three sample sets over time is illustrated in Fig. 15. The area of Fort AP Hill is about 516 m^2, hence the numbers of used particles for three approaches are the same as that in Intel Research Lab. As shown in Fig. 15, the robot was placed in the corridor at startup and moved forward until the sample set converged to the correct location. The average positions errors of three methods are depicted in Fig. 16(b). As these results suggest, the average position errors at the beginning for both MCL and SAMCL in Fort AP Hill are lower than that for both methods

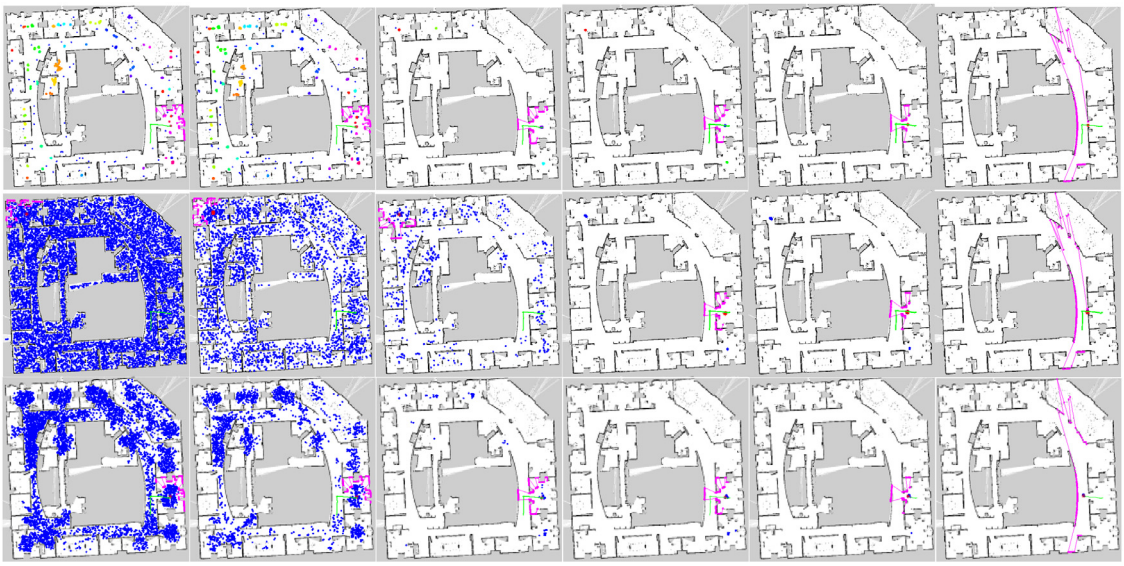

Fig. 14. Localization results in Intel Research Lab using proposed method (first row) compared with the standard MCL method (second row) and the SAMCL method (third row). The images in each row (from left to right) depict the corresponding estimation results at the same ground truth position. The green line indicates the trajectory of the robot, and the pink line represents the measured range scan data at each time step. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

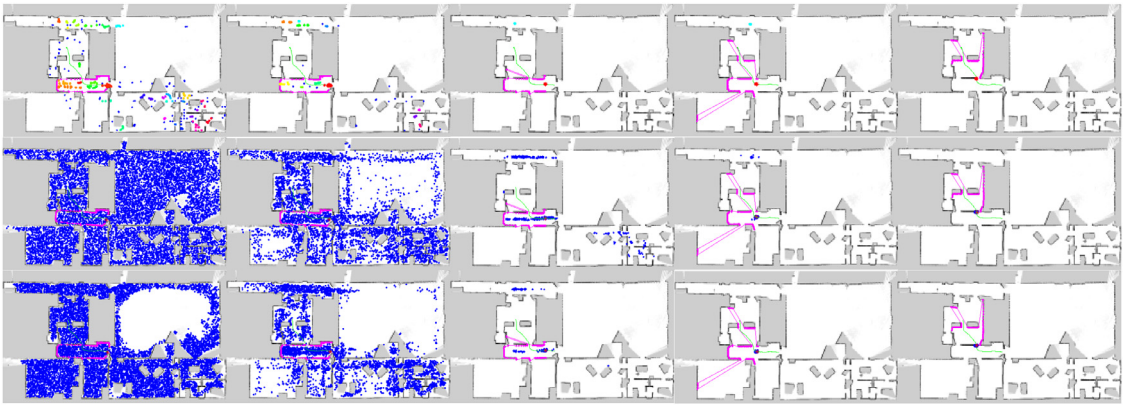

Fig. 15. Localization results in Fort AP Hill using proposed method (first row) compared with the standard MCL method (second row) and the SAMCL method (third row). The images in each row (from left to right) depict the corresponding estimation results at the same ground truth position. The green line indicates the trajectory of the robot, and the pink line represents the measured range scan data at each time step. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

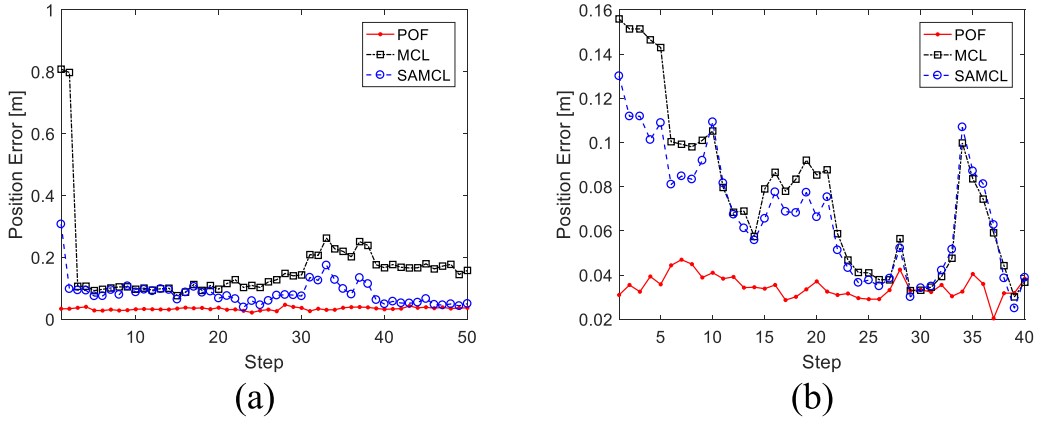

Fig. 16. Performance comparison between the proposed localization, the MCL method and the SAMCL algorithm in Intel Research Lab (a) and Fort AP Hill (b).

in Intel Research Lab, since there are multiple similar local places in the Intel Research Lab. In addition, these results show that the proposed approach is able to provide an accurate initial pose estimation and maintain a low position error during the localization process when compared to the other two methods.

The experiment results indicate that both MCL and SAMCL perform better in Fort AP Hill than in Intel Research Lab since the former environment is asymmetric. However, in an ambiguous environment, e.g. the Intel Research Lab, multi-modal distributions arise for the problem of global localization where multiple solutions have to be maintained during the localization process. Particle filter implementations usually suffer from the problem of premature convergence due to the loss of diversity of the particle set, which may lead to localization failure. In the proposed approach, the particle set is divided into distinct clusters through optimization in local regions instead of independent particles in MCL and SAMCL during the first stage of GL. In the second stage of the proposed approach, particles in each cluster collectively optimize the current best solution for the corresponding pose hypothesis from iteration to iteration until the whole population converges. Hence, the diversity of particles can be maintained until the entire population converges to the correct solution, minimizing the influence of premature convergence. As the experiment results suggest, the proposed POF based GL method shows better performance compared with MCL and SAMCL in typical indoor environments. However, the proposed approach may cause more problems with undesirable clusters in non-ambiguous environments and the number of particles used for multiple pose tracking can be reduced.

5. Conclusion

The main contribution of this work is the development of an integrated localization approach that combines a local search based PSO algorithm with the particle filter for robust and accurate global localization. In order to estimate the initial robot pose, a revised PSO algorithm with local search is presented, where multiple pose hypotheses are identified using the DBSCAN technique. Multiple solutions are maintained as the robot moves until the particle set converges to the true robot pose. Experimental results demonstrate that the POF algorithm outperforms the MCL and SAMCL algorithm both in the accuracy and robustness of the localization results, with comparable computational cost to that of MCL and SAMCL.

Although developed in the particle filter framework as MCL, the POF approach maintains multiple pose hypotheses with local optimization of clustered particles instead of the individual particles in MCL and SAMCL throughout the localization process. Hence, POF is more robust than MCL and SAMCL due to its good diversity preserving performance. In addition, the iterative optimization process results in more compact sub-swarms, which achieves more accurate position estimates with fewer particles for the mobile robot.

As a future work, the proposed POF approach can be extended for more efficient global localization in large-scale environments. Scalability of the localization approach is another problem that has to be taken into account in practical applications, since the number of particles required to represent the probability distribution of the robot state is proportional to the area of the environment. In order to perform localization in real time, there are several ways to be considered. Embedding semantic information of the environment and segmentation of the environment map are two efficient ways to identify local places where the robot may be located. In addition, laser scan matching and self-adaption techniques such as KLD-sampling can be integrated with POF as well, since scan matching approaches are capable of providing accurate pose estimates with only a few particles and the KLD-sampling is able to reduce the sample set size when tracking the robot state.

Declaration of Competing Interest

The authors declare no conflict of interest.

Credit authorship contribution statement

Qi-bin Zhang: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing - original draft, Visualization. **Peng Wang:** Conceptualization, Writing - review & editing. **Zong-hai Chen:** Resources, Writing - review & editing, Supervision, Project administration, Funding acquisition.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 91848111).

References

Andrew, H., & Nicholas, R. (2003). *The robotics data set repository (Radish)* [Dataset].
Andrieu, C., De Freitas, N., Doucet, A., & Jordan, M. I. (2003). An introduction to MCMC for machine learning. *Machine Learning, 50*, 5–43.
Bengochea-Guevara, J. M., Conesa-Muñoz, J., Andújar, D., & Ribeiro, A. (2016). Merge fuzzy visual servoing and GPS-based planning to obtain a proper navigation behavior for a small crop-inspection robot. *Sensors, 16*, 276.
Bengtsson, O., & Baerveldt, A.-J. (2003). Robot localization based on scan-matching—Estimating the covariance matrix for the IDC algorithm. *Robotics and Autonomous Systems, 44*, 29–40.
Blanco, J.-L., González, J., & Fernández-Madrigal, J.-A. (2008). An optimal filtering algorithm for non-parametric observation models in robot localization. *2008 IEEE international conference on robotics and automation (ICRA 2008)*. IEEE. (pp. 461–466).
Blanco, J.-L., González, J., & Fernández-Madrigal, J.-A. (2010). Optimal filtering for non-parametric observation models: Applications to localization and SLAM. *International Journal of Robotics Research, 29*, 1726–1742.
Chen, B., Cai, Z., & Yuan, C. (2009). Mobile robot SLAM method based on particle swarm optimization. *Robot, 31*, 513–517.
Chien, C.-H., Wang, W.-Y., & Hsu, C.-C. (2017). Multi-objective evolutionary approach to prevent premature convergence in Monte Carlo localization. *Applied Soft Computing, 50*, 260–279.
Choi, S., Lee, S., Viet, H. H., & Chung, T. (2017). B-theta*: An efficient online coverage algorithm for autonomous cleaning robots. *Journal of Intelligent & Robotic Systems, 87*, 1–26.
Clerc, M., & Kennedy, J. (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation, 6*, 58–73.
Doucet, A., Freitas, J. F. G. D., & Gordon, N. J. (2001). *Editors: Sequential Monte Carlo methods in practice*. New York: Springer-Verlag.
Eberhart, R. C., & Shi, Y. (2000). Comparing inertia weights and constriction factors in particle swarm optimization. *Proceedings of the 2000 congress on evolutionary computation*: Vol. 1. IEEE. pp. 84–88.
Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *International conference on knowledge discovery & data mining*: Vol. 96. pp. 226–231.
Fox, D. (2003). Adapting the sample size in particle filters through KLD-sampling. *International Journal of Robotics Research, 22*, 985–1003.
Fox, D., Burgard, W., & Thrun, S. (1999). Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research, 11*, 391–427.
Guan, R. P., Ristic, B., Wang, L., & Palmer, J. L. (2019). KLD sampling with Gmapping proposal for Monte Carlo localization of mobile robots. *Information Fusion, 49*, 79–88.
Havangi, R., Nekoui, M. A., & Teshnehlab, M. (2010). A multi swarm particle filter for mobile robot localization. *International Journal of Computer Science, 7*, 15–22.
Janson, S., & Middendorf, M. (2003). A hierarchical particle swarm optimizer. *2003 congress on evolutionary computation*: Vol. 2. IEEE. pp. 770–776.
Jensfelt, P., & Kristensen, S. (2001). Active global localization for a mobile robot using multiple hypothesis tracking. *IEEE Transactions on Robotics and Automation, 17*, 748–760.
Kenndy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of IEEE international conference on neural networks*: Vol. 4. IEEE Press. pp. 1942–1948.
Kennedy, J. (1999). Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance. In *Proceedings of the 1999 congress on evolutionary computation: Vol. 3* (pp. 1931–1938).
Kulkarni, R. V., & Venayagamoorthy, G. K. (2010). Bio-inspired algorithms for autonomous deployment and localization of sensor nodes. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 40*, 663–675.
Lavanya, D., & Udgata, S. (2011). Swarm intelligence based localization in wireless sensor networks. *Multi-Disciplinary Trends in Artificial Intelligence, 7080*, 317–328.
Le, M. C., Phung, S. L., & Bouzerdoum, A. (2014). Lane detection in unstructured environments for autonomous navigation systems. *Asian conference on computer vision*. Springer. (pp. 414–429).

Lee, H. S., & Lee, K. M. (2009). Multiswarm particle filter for vision based SLAM. *2009 IEEE/RSJ international conference on intelligent robots and systems*. IEEE. (pp. 924–929).

Li, L., Yang, M., Guo, L., Wang, C., & Wang, B. (2016). Precise and reliable localization of intelligent vehicles for safe driving. *International conference on intelligent autonomous systems*. Springer. (pp. 1103–1115).

Liu, Z., Shi, Z., Zhao, M., & Xu, W. (2008). Adaptive dynamic clustered particle filtering for mobile robots global localization. *Journal of Intelligent and Robotic Systems, 53*, 57–85.

Ma, X., & Mao, R. (2018). Path planning for coal mine robot to avoid obstacle in gas distribution area. *International Journal of Advanced Robotic Systems, 15*, 1–6.

Manjarres, D., Del Ser, J., Gil-Lopez, S., Vecchio, M., Landa-Torres, I., & Lopez-Valcarce, R. (2013). A novel heuristic approach for distance-and connectivity-based multihop node localization in wireless sensor networks. *Soft Computing, 17*, 17–28.

Michael, N., Shen, S., Mohta, K., Kumar, V., Nagatani, K., Okada, Y., et al. (2014). Collaborative mapping of an earthquake damaged building via ground and aerial robots. *Field and service robotics*. Springer. (pp. 33–47).

Miksik, O., Petyovsky, P., Zalud, L., & Jura, P. (2011). Robust detection of shady and highlighted roads for monocular camera based navigation of UGV. *2011 IEEE international conference on robotics and automation (ICRA)*. IEEE. (pp. 64–71).

Nguyen, H. A., Guo, H., & Low, K.-S. (2011). Real-time estimation of sensor node's position using particle swarm optimization with log-barrier constraint. *IEEE Transactions on Instrumentation and Measurement, 60*, 3619–3628.

Park, S., & Roh, K. S. (2016). Coarse-to-fine localization for a mobile robot based on place learning with a 2-D range scan. *IEEE Transactions on Robotics, 32*, 528–544.

Peer, E. S., van den Bergh, F., & Engelbrecht, A. P. (2003). Using neighbourhoods with the guaranteed convergence PSO. *Proceedings of the 2003 IEEE swarm intelligence symposium*. IEEE. (pp. 235–242).

Pinto, A. M., Moreira, A. P., & Costa, P. G. (2015). A localization method based on map-matching and particle swarm optimization. *Journal of Intelligent & Robotic Systems, 77*, 313–326.

Pinto, A. M. G., Moreira, A. P., & Costa, P. G. (2013). Robot@factory: Localization method based on map-matching and particle swarm optimization. *2013 EEE international conference on autonomous robot systems*.

Poli, R., Kennedy, J., & Blackwell, T. (2007). Particle swarm optimization. *Swarm Intelligence, 1*, 33–57.

Röwekämper, J., Sprunk, C., Tipaldi, G. D., Stachniss, C., Pfaff, P., & Burgard, W. (2012). On the position accuracy of mobile robot localization based on particle filters combined with scan matching. *2012 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE. (pp. 3158–3164).

Rubin, D. B. (1988). Using the SIR algorithm to simulate posterior distributions. *Bayesian Statistics, 3*, 395–402.

Segal, A., Haehnel, D., & Thrun, S. (2009). Generalized-ICP. In *Robotics: Science and systems: Vol. 2* (p. 435).

Song, X., Gao, H., Ding, L., Deng, Z., & Chao, C. (2017). Diagonal recurrent neural networks for parameters identification of terrain based on wheel–soil interaction analysis. *Neural Computing and Applications, 28*, 797–804.

Teslić, L., Škrjanc, I., & Klančar, G. (2011). EKF-based localization of a wheeled mobile robot in structured environments. *Journal of Intelligent & Robotic Systems, 62*, 187–203.

Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic robotics*. MIT press.

Thrun, S., Fox, D., Burgard, W., & Dellaert, F. (2001). Robust Monte Carlo localization for mobile robots. *Artificial Intelligence, 128*, 99–141.

Vahdat, A. R., NourAshrafoddin, N., & Ghidary, S. S. (2007). Mobile robot global localization using differential evolution and particle swarm optimization. *2007 IEEE congress on evolutionary computation*. IEEE. (pp. 1527–1534).

Walia, G. S., & Kapoor, R. (2014). Intelligent video target tracking using an evolutionary particle filter based upon improved cuckoo search. *Expert Systems with Applications, 41*, 6315–6326.

Wang, J., Wang, P., & Chen, Z. (2018). A novel qualitative motion model based probabilistic indoor global localization method. *Information Sciences, 429*, 284–295.

Wang, P., Zhang, Q., & Chen, Z. (2015). Feature extension and matching for mobile robot global localization. *Journal of Systems Engineering and Electronics, 26*, 840–846.

Woo, J., Kim, Y.-J., Lee, J.-o., & Lim, M.-T. (2006). Localization of mobile robot using particle filter. *2006 international joint conference on SICE-ICASE*. IEEE. (pp. 3031–3034).

Zhang, L., Zapata, R., & Lépinay, P. (2012). Self-adaptive Monte Carlo localization for mobile robots using range finders. *Robotica, 30*, 229–244.

Zhang, Q., Wang, P., Bao, P., & Chen, Z. (2017). Mobile robot global localization using particle swarm optimization with a 2D range scan. In *International conference on robotics and artificial intelligence* (pp. 105–109).

Zhang, X., Hu, W., Maybank, S., Li, X., & Zhu, M. (2008). Sequential particle swarm optimization for visual tracking. *2008 IEEE conference on computer vision and pattern recognition*. IEEE. (pp. 1–8).

Zhao, J., & Li, Z. (2010). Particle filter based on particle swarm optimization resampling for vision tracking. *Expert Systems with Applications, 37*, 8910–8914.

Zhao, Y., Wang, T., Qin, W., & Zhang, X. (2018). Improved Rao-Blackwellised particle filter based on randomly weighted particle swarm optimization. *Computers & Electrical Engineering, 71*, 477–484.

Zhu, J., Zheng, N., & Yuan, Z. (2011). An improved technique for robot global localization in indoor environments. *International Journal of Advanced Robotic Systems, 8*, 21–28.

Zuo, T., Min, H., Tang, Q., & Tao, Q. (2018). A robot SLAM improved by quantum-behaved particles swarm optimization. *Mathematical Problems in Engineering, 2018*, 1–11.