IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 18, 2018, accepted March 19, 2018, date of publication March 26, 2018, date of current version April 25, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2819419

A Fast Algorithm of Simultaneous Localization and Mapping for Mobile Robot Based on Ball Particle Filter

JINGWEN LUO AND SHIYIN QIN
School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China
Corresponding author: Jingwen Luo (by1503117@buaa.edu.cn).

This work was supported in part by the Beijing Science and Technology Planning Project of China under Grant D161104001300000-D161100001316001 and in part by the National Nature Science Foundation of China under Grant 61731001 and Grant U1435220.

ABSTRACT The FastSLAM algorithm has become an effective way to solve the simultaneous localization and mapping (SLAM) problem. However, measured in terms of the number of particles required to build an accurate map, currently, its accuracy cannot be easily enhanced because of particle degeneracy. In view of these problems, in this paper, we present a fast algorithm of SLAM based on the ball particle filter (Ball-PF), which originates from the modification of the box particle filter (Box-PF). First, the transform relationship between Box-PF and Ball-PF are studied in depth so as to show the advantages of Ball-PF with respect to solving the interval constraints satisfaction problem and prevent from breaking down effectively. Then, a new fast algorithm of SLAM is designed with Ball-PF, in which the firefly algorithm is used to maintain the diversity of the ball particles to increase the consistency of the pose estimation effectually. Furthermore, the map matching technique is used to compute the weight of the ball particles and learn the grid maps incrementally. The simulation and experimental results demonstrate the performance superiority of the proposed algorithm.

INDEX TERMS SLAM, mobile robot, box particle filter, firefly algorithm, FastSLAM, ball particle filter.

I. INTRODUCTION

Autonomous navigation is one of the essential technologies for mobile robots and plays an important role in the implementation of their intelligence. However, simultaneous localization and mapping (SLAM) is the key to achieve truly autonomous navigation, in which both simultaneous localization and map building are mutually dependent on each other result in some difficulties and complexities for an accurate solution, particularly in the high-dimensional space.

The earliest probability-based method using extended Kalman filters (EKFs) for solving SLAM was proposed by Smith and Cheeseman [1]. With further development, Paskin [2] presented a low-complexity solution to the SLAM problem by using thin junction trees. To solve the problem of computation and storage, Thrun *et al.* [3] proposed a type of SLAM algorithm based on sparse extended information filtering. Then, Murphy and his colleagues proposed an effective method to solve the SLAM problem by using the Rao-Blackwellized particle filter (RBPF) and named it FastSLAM [4], [5]. Subsequently, the FastSLAM algorithm is improved by Montemerlo *et al.* [6]. Two applications of FastSLAM—for learning accurate grid maps—are described by Hahnel *et al.* [7] and Eliazar and Parr [8].

The main problem of FastSLAM is that building a precise map requires a large number of particles. Additionally, the resampling step is problematic as it can eliminate good state hypotheses. To overcome these problems, many approaches have been proposed. One is to use laser scan matching [9], which provides a better proposal distribution. Giorgio *et al.* [10] proposed an improved grid-based FastSLAM algorithm using adaptive proposals and selective resampling techniques to reduce the number of particles. An integration technique, combining a genetic algorithm (GA) and particle swarm optimization (PSO) within FastSLAM, was presented in [11]. Nevertheless, these probabilistic approaches have a common drawback: consistency problems. Studies [12], [13] have shown that the RBPF SLAM can obtain accurate positional estimates, but only in a short time to meet the consistency requirements. This drawback can be overcome using interval analysis (IA) methods

rather than probabilistic ones [14]. Indeed, IA guarantees by its design that the IA calculations and methods do not suffer from biased measurements. Thus, IA provides definite and consistent results. Such advantages have been highlighted in localization applications [15], [16] and have opened interesting perspectives for SLAM problems [17]–[19].

A particle filter (PF) strategy for mobile robot localization involving interval data was introduced in [20] and was proposed as box particle filtering (Box-PF). Resulting from the synergy between PF and IA, Box-PF is an approach that has recently emerged and is aimed at solving a general class of nonlinear filtering problems. The key idea is to replace a particle with a multidimensional interval or box of non-zero volume in the state space. This approach is particularly appealing in practical situations involving imprecise stochastic measurements that result in very broad posterior densities. Most recently, various applications [21]–[25] have shown that an accurate and reliable performance of several thousand particles can be achieved by just a few dozen boxes.

The constraints satisfaction problem (CSP) is a major problem in Box-PF. It can be solved using a constraint propagation (CP) algorithm, which combines consistency techniques and systematic search methods for each box to make them consistent with measurements. An alternative and real-time interval method based on a CP algorithm has been successfully applied to robotics applications. See [26]–[28], for example. The main advantage of CP over Bayesian algorithms is that it guarantees that the position of the vehicle is contained within a box [12]. Furthermore, inspired by the attraction and movement behavior of fireflies, the firefly algorithm (FA) was introduced in [29]. In [30] and [31], the FA shows good performance in maintaining particle diversity and improving the overall quality of the particle swarm. Thus, in the IA framework, alternative techniques to obtain a more selective and precise solution are also possible.

The rest of the paper is organized as follows. The mathematical preliminaries needed are briefly presented in Section II. The Ball-PF algorithm is introduced in Section III, and its performance is analyzed. The fast algorithm of SLAM, based on the Ball-PF implementation process, is presented in Section IV. Section V provides the experimental results and comparative analyses, while Section VI concludes the paper.

II. ELEMENTARY CONCEPTS AND OPERATION OF THE BALL PARTICLE FILTER

In this section, elementary concepts about IA are introduced, based on which the ball can be described in a straightforward manner by using the concept of the center interval. Thus, the CSP is analyzed with a fixed-point subsolver.

A. CENTER INTERVAL AND BALL

The main concept of IA is to deal with the intervals of real numbers instead of dealing with the real numbers themselves [32]. A real interval $[x] = [\underline{x}, \bar{x}]$ is defined as a closed and connected subset of R with $\underline{x}$ and $\bar{x}$ denoting the lower and the upper bounds of x, respectively. The set of n-dimensional real intervals is denoted by IR^n. An interval vector, or a box $[\mathrm{x}] \in R^n$, is a Cartesian product of n intervals, which may be represented as follows:

$$[\mathrm{x}] = [x_1] \times [x_2] \times \cdots \times [x_n] = \times_{i=1}^{n} [x_i] \tag{1}$$

In IA, the size of box [x] is denoted as $|[\mathrm{x}]|$, and a center interval $[x]$ is denoted as $[x] = [mid([x]), rad([x])] = [x_0, \ r]$, where $mid([x]) = (\underline{x} + \bar{x})/2$ and $rad([x]) = (\bar{x} - \underline{x})/2$.

The mathematical operations of the center interval can be drawn from the elementary interval operations (see Appendix, Table 7). In particular, the distance between $[x] = [x_0, r_x]$ and $[y] = [y_0, r_y]$ is defined as follows:

$$d([x], [y]) = |x_0 - y_0| + \left|r_x - r_y\right| \tag{2}$$

A ball in R^n is in fact a center interval in R extending to R^n. Let $x_0 \in R^n$, $0 \le r \in R$, and $\|\cdot\|$ be an arbitrary norm in R^n. The set $\{x \in R^n \,|\, \|x - x_0\| \le r\}$ is called the n-dimensional real ball with the center (midpoint) x_0 and radius r. At this point, a ball in R^n can be expressed as $\langle \mathrm{x} \rangle = \langle x_0, r \rangle$. The ball mathematics basically has two definitions. Let $\lambda \in R$, $x \in R^n$, and $\langle \mathrm{x} \rangle = \langle x_0, r \rangle$; then,

$$\lambda \cdot \langle \mathrm{x} \rangle = \langle \mathrm{x} \rangle \cdot \lambda = \langle \lambda x_0, |\lambda| \, r \rangle \tag{3}$$

$$x + \langle \mathrm{x} \rangle = \langle \mathrm{x} \rangle + x = \langle x_0 + x, r \rangle \tag{4}$$

B. CONSTRAINTS SATISFACTION PRPBLEM

Consider a mapping $f : R^n \to R^m$; then, the interval function $[f]$ from IR^n to IR^m is an *inclusion function* as shown in Fig. 1. It is obvious that $f([\mathrm{x}]) \subset [f]([\mathrm{x}]), \forall [\mathrm{x}] \in IR^n$.

One of the purposes of IA fo f is to provide a reasonable $[f]$ which can be evaluated such that an appropriate size of $[f]([\mathrm{x}])$ is achieved. Therefore, we need to solve the CSP commonly expressed as follows:

$$H : (\mathrm{G(x)} = 0, \mathrm{x} \in [\mathrm{x}]) \tag{5}$$

The connotation of (5) is to find the optimal box enclosure of the set of vectors x belonging to a given prior domain [x] satisfying a set of constraints $\mathrm{G(x)} = (g_1(\mathrm{x}), \cdots g_m(\mathrm{x}))^{\mathrm{T}}$ for various real functions $g_i(\mathrm{x})$. The solution set S consists of all of the values of x satisfying $\mathrm{G(x)} = 0$, and can be denoted as follows:

$$S = \{\mathrm{x} \in [\mathrm{x}] \,|\mathrm{G(x)} = 0\} \tag{6}$$

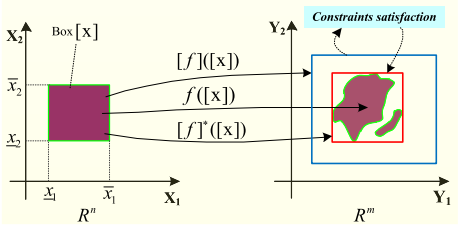


FIGURE 1. Mapping of a box [x] by a vector of function f and its two different inclusion functions $[f]$ and $[f]^*$.

A contractor for H is any operator that can be used to contract H, i.e. replacing [x] by a smaller domain [x]$'$, such that

$$S \subseteq [\mathrm{x}]' \subseteq [\mathrm{x}] \tag{7}$$

Various constraints methods named *contractors* are described in [33], including Gauss elimination, the Krawczyk method, and forward-backward propagation, etc. Each of these methods can be suitable for different types of CSP. Furthermore, it is important that the consistency conditions be satisfied, namely, *global consistency* and *local consistency*. Global consistency represents the ideal solution for CSP, and it is stronger than local consistency. However, for most CSPs, the existing methods can only reach *local consistency*.

Let $\psi : R^{n_x} \rightarrow R^{n_x}$ be a fixed-point subsolver [33] for H, and $[\psi] : IR^{n_x} \rightarrow IR^{n_x}$ be an inclusion function for ψ. A contractor for H is obtained by replacing [x] in H by $[\mathrm{x}] \cap [\psi]([\mathrm{x}])$. This contractor will be called the fixed-point contractor associated with ψ. For a given set of constraints, an iterative application of the corresponding projection procedures over the constraints will lead to a state where no variable domain can be further reduced. That is, a fix point is reached. Nickel first proposed an iterative mapping technique from ball-to-ball called the ''ball Newton operator''. It is essentially a fixed-point method; more details about this method can be found in [34]–[36].

III. FROM BOX PARTICLE FILTER TO BALL PARTICLE FILTER

In this section, we first present a brief description of Box-PF and analyze its performance. Then, the evolutionary strategy for Ball-PF is discussed in detail.

A. BOX PARTICLE FILTER AND ITS PERFORMANCE

In terms of IA, for the sake of quantifying uncertainties, the state vector and the measurement vector become the vectors of intervals. Likewise, the propagation and the observation functions become inclusion functions, denoted as $[f]$ and $[h]$, respectively. Thus, the system dynamics equations can be defined as follows:

$$\begin{cases} [\mathrm{x}_{k+1}] = [f]([\mathrm{x}_k], [\mathrm{u}_{k+1}]) \\ [\mathrm{z}_{k+1}] = [h]([\mathrm{x}_{k+1}]) \end{cases} \tag{8}$$

where $[\mathrm{x}_k] \in IR^{n_x}$ and $[\mathrm{z}_k] \in IR^{n_y}$ are the state interval vector and the measurement interval vector at time step k, respectively. The control vector [u] is deduced from the proprioceptive sensor data.

Box-PF is a nonlinear filtering algorithm which couples a sequential Monte Carlo method and IA. The key idea is to use box particles and a bounded error model, instead of discrete point particles and probabilistic models, for the errors and the inputs. Details of the full Box-PF algorithm are provided in [20] and [37]–[39]. Fig. 2 illustrates the scenarios of Box-PF. The main advantage of Box-PF compared with traditional PF is that it substantially reduces the number of particles required for the prediction. Its low computational complexity and high speed improve the filtering performance and make it suitable for distributed filtering. This approach guarantees that the real state of the system is included in the estimation box at each step.

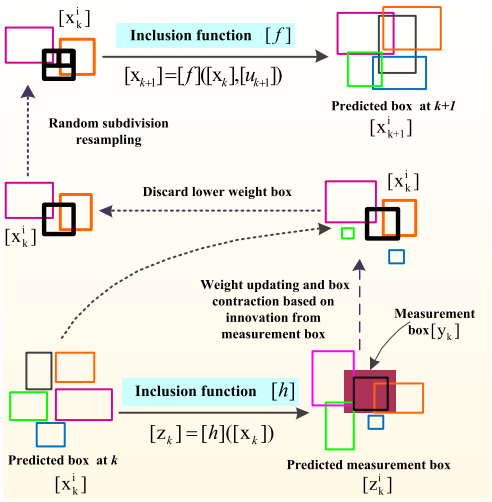


FIGURE 2. Scenarios for the box particle filter.

Unfortunately, random subdivision resampling is usually adopted for Box-PF to solve the degeneracy phenomenon. In general, the process of subdivision is relatively random, which affects its filtering precision and the result of each experiment is different, while the computational burden caused by oversampling is increased. In the bounded error areas, the choice of the number of divisions for each dimension is not optimal and remains a subject of research [40]. In addition, Box-PF employs the CP technique [22], [39], [41] to contract boxes. A well-known drawback of CP is that the decomposition into primitive constraints introduces new variables in the CSP. This hinders efficient domain tightening. Meanwhile, the main limitation of CP is its sensitivity to the multiple occurrences of variables [42], and the results only satisfy local consistency. For some applications, we can only contract box particles for two-dimensional cases [43], and the results are by no means satisfactory as the constraints are accounted for in an arbitrary order [44].

B. BALL PARTICLE FILTER AND ITS EVOLUTIONARY SUPERIORITY

The Ball-PF implementation has a similar scheme to that of Box-PF. The main idea of Ball-PF is to construct a whole ball with properties that replace a box, use a ball contractor to contract the ball particle instead of CP, and introduce the firefly intelligent optimization strategy to improve the total mass of the ball particles. The differences between Box-PF and Ball-PF will be presented in the following text.

1) CONSTRUCTION OF A BALL

Mathematically, determine a box in R^n needing $2n$ parameters. In contrast, for obtaining a ball in R^n, we simply need to determine the center $x_0 \in R^n$ and the radius $r \in R$, that is,

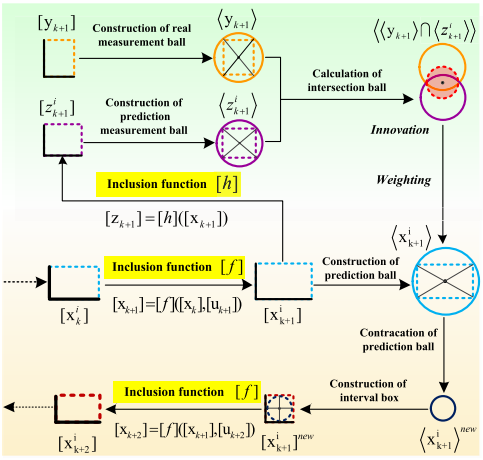


FIGURE 3. Construction procedure for a ball $\in R^2$ in a single iteration cycle.

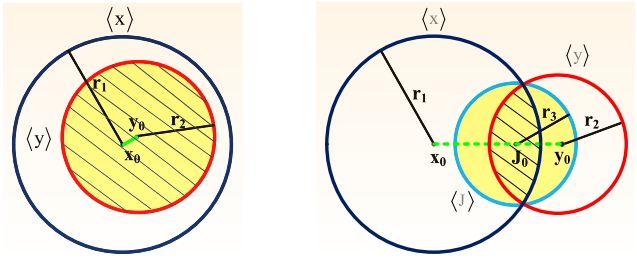


FIGURE 4. Intersection of two balls in R^2, $\langle J\rangle = \langle\langle x\rangle \cap \langle y\rangle\rangle$ is the smallest ball containing $\langle x\rangle \cap \langle y\rangle$. **(a)** $\|y_0 - x_0\|^2 + r_2^2 < r_1^2$. **(b)** $\|y_0 - x_0\|^2 + r_2^2 \geq r_1^2$.

a total of $n+1$ parameters. Therefore, the determination of a ball is more convenient than that of a box. Thus, a ball has higher real significance than a box. In our work, the definition of a ball can be transformed from that of a box.

Assume that $[x] = [x_1] \times [x_2] \times \cdots \times [x_n]$ is a state interval vector in R^n, where $[x_i]$ is the center interval $[x_{i0}, r_i]$, $i = 1, \cdots, n$. We can define the center vector $x_0 = (x_{10}, x_{20}, \cdots x_{n0})^T$ composed of all midpoints of x_{i0} as the center of an n-dimensional ball and $r_0 = \sqrt{r_1^2 + r_2^2 + \cdots r_n^2}$ as the radius. Thus, the form of a circumsphere of a box [x] may be constructed as a whole ball $\langle x_0, r_0\rangle$ with the property (see Fig. 3). To prevent Box-PF from breaking down, one can add an artificial noise to the bounds of the box [20]. In our cases, the ball maintains a suitable size for each dimension of the center interval vector in every iteration cycle, which can effectively prevent the algorithm from breaking down.

2) INTERSECTION OF THE BALLS

The intersection of two balls does not form a ball. Fig. 4 shows the intersection set of two circles in R^2. Significantly, there is always a circle satisfaction $\langle x\rangle \cap \langle y\rangle \in \langle J\rangle$. To make up for this un-closeness, we can calculate the intersection ball to guarantee that the ball contains the intersection set.

Let $\langle x\rangle = \langle x_0, r_1\rangle$, $\langle y\rangle = \langle y_0, r_2\rangle$, and $\langle J\rangle = \langle\langle x\rangle \cap \langle y\rangle\rangle = \langle J_0, r_3\rangle$. Suppose that $x_0 \neq y_0$, $r_1 \geq r_2 > 0$, and $r_1 + r_2 \geq \|y_0 - x_0\|$, thus, the following results hold:

- $J_0 \in \langle x\rangle \cap \langle y\rangle \subset \langle J\rangle$, $r_3 \leq \min\{r_1, r_2\}$, (9)
- $\|y_0 - x_0\|^2 + r_2^2 < r_1^2$, $\langle J\rangle = \langle y\rangle$ (10)
- $\|y_0 - x_0\|^2 + r_2^2 \geq r_1^2$,

$$J_0 = \frac{1}{2}\left(y_0 + x_0 + \frac{(y_0 - x_0)(r_1^2 - r_2^2)}{\|y_0 - x_0\|^2}\right) \quad (11)$$

$$r_3 = \frac{1}{2}\sqrt{2(r_1^2 + r_2^2) - \|y_0 - x_0\|^2 - \frac{(r_1^2 - r_2^2)^2}{\|y_0 - x_0\|^2}} \quad (12)$$

3) CONTRACTION OF A BALL

To conserve a judicious radius for each ball, contraction algorithms should be used to eliminate the non-consistent part of the ball particle with respect to the ball measurement. In our work, depending upon the observation equation and the ball contraction algorithm, a whole n-dimensional ball is determined in each iteration cycle. The real measurement ball is given with respect to the sensor uncertainty. If the predicted measurement ball contains the desired zero (true state) of the given observation function, perform the ball contraction algorithm to obtain a new ball. The ball contraction operator can be found in Appendix, Section B, and more details, please refer to [35].

For the ball contraction operator C, numerical experiments [35] were carried out to obtain a nonsingular matrix Λ and $\lambda \in R$, $(0 \leq \lambda < 1)$ to construct a regular ball operator $Q = \langle\langle \Lambda, \lambda\rangle\rangle$, that is,

$$Qx = \langle \Lambda x; \lambda \|\Lambda x\|\rangle, \ for\ x \in R^n \quad (13)$$

According to the observation functions h and the real measurement y, define the ball contraction operator C by as follows:

$$Cx = x - (Qh(x) - y) = \langle x - \Lambda(h(x) - y); \lambda \|\Lambda(h(x) - y)\|\rangle \quad (14)$$

In the applications, we usually adopt one iteration intersection operation to obtain the new ball after contraction, because we only pay attention to the solution of the problem in B_k. The contraction result of each ball particle $\langle x_k^i\rangle$ is denoted as $\langle x_k^i\rangle^{new}$. Clearly, each iteration step produces a new ball that is smaller than the original one. The method is globally convergent and guarantees $x_k \in B_0$. The algorithm is listed in Table 1 and the contraction procedure is summarized in Fig. 5.

4) BAYESIAN EXPLANATION

Within the Bayesian framework, the state estimation consists of prediction and correction, i.e.

$$p(x_{k+1}|z_{1,k}) = \int p(x_{k+1}|x_k)p(x_k|z_{1,k})dx_k \quad (15)$$

$$p(x_{k+1}|z_{1,k+1}) = \frac{1}{\alpha} \cdot p(z_{k+1}|x_{k+1})p(x_{k+1}|z_{1,k}) \quad (16)$$

TABLE 1. Ball contraction algorithm.

Algorithm **1:** Ball Contraction Algorithm

Input: Initial ball $B_0 = \langle \mathrm{x}_k^i \rangle$, and regular ball operator $\mathcal{Q} = \langle\langle \Lambda, \lambda \rangle\rangle$.

Output: Contraction result $\langle \mathrm{x}_k^i \rangle^{new}$.

1 Let $B_0 = \langle x_0, r_0 \rangle$ .Calculate Cx_0 .
2 If $B_0 \bigcap Cx_0 = \phi$, then, end .
3 If $B_0 \bigcap Cx_0 \neq \phi$, then, $B_k = \langle B_0 \bigcap Cx_0 \rangle = \langle x_k, r_k \rangle$.
4 Suppose $k \geq 1$. Calculate Cx_k .
5 If $B_k \bigcap Cx_k = \phi$, then, end.
6 If $B_k \bigcap Cx_k \neq \phi$,then, $\tilde{B}_{k+1} = \langle B_k \bigcap Cx_k \rangle = \langle \tilde{x}_{k+1}, \tilde{r}_{k+1} \rangle$
7 If $\tilde{B}_{k+1} \bigcap B_0 = \phi$, then, end.
8 If $\tilde{B}_{k+1} \bigcap B_0 \neq \phi$, then,
9 If $\tilde{x}_{k+1} \in B_0$, then $B_{k+1} = \tilde{B}_{k+1}$.
10 Else $B_{k+1} = \langle B_0 \bigcap \tilde{B}_{k+1} \rangle$, end.

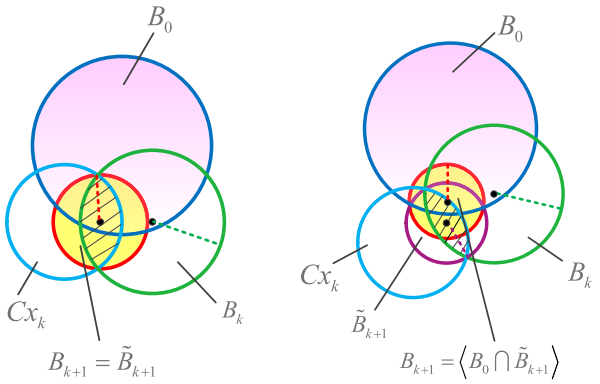


FIGURE 5. Contraction procedure for the ball contractor. (a) $\tilde{x}_{k+1} \in B_0$. (b) $\tilde{x}_{k+1} \notin B_0$.

where normalized coefficient $\alpha = \int p(z_{k+1}|x_{k+1}) p(x_{k+1}| z_{1,k}) dx_k$.

According to [20], [37], and [38], the Box-PF can be considered an approximation of the Bayesian filter by interpreting each box particle as a uniform probability density function (PDF). Likewise, the set of ball particles is interpreted as a mixture of uniform PDFs. The previous time PDF for Ball-PF is written as follows:

$$p(\mathrm{x}_k \,|\mathrm{z}_{1,k}) = \sum_{i=1}^{N} \omega_k^i U_{\langle \mathrm{x}_k^i \rangle}(\mathrm{x}_k) \tag{17}$$

where ω^i denotes a set of normalized weights $\sum_{i=1}^{N} \omega^i = 1$ and $\forall i, \omega^i \geq 0$, and $U_{\langle \mathrm{x} \rangle}$ denotes the multivariate uniform PDF with the ball $\langle \mathrm{x} \rangle$ as the support (the uniform distribution in the ball is described in Appendix, Section A).

- Prediction

Inserting (17) into (15)

$$\begin{aligned} p(\mathrm{x}_{k+1} \,|\mathrm{z}_{1,k}) &= \int p(\mathrm{x}_{k+1} \,|\mathrm{x}_k) \sum_{i=1}^{N} \omega_k^i U_{\langle \mathrm{x}_k^i \rangle}(\mathrm{x}_k) d\mathrm{x}_k \\ &= \sum_{i=1}^{N} \omega_k^i \frac{1}{|\langle \mathrm{x}_k^i \rangle|} \int_{\langle \mathrm{x}_k^i \rangle} p(\mathrm{x}_{k+1}|\mathrm{x}_k) d\mathrm{x}_k \end{aligned} \tag{18}$$

Assume that the noise v_{k+1} at time $k+1$ is bounded in $\langle \mathrm{v}_{k+1} \rangle$. The noise PDF is also approximated with a mixture of uniform PDFs, that is,

$$p(\mathrm{v}_{k+1}) = \sum_{i=1}^{R} \lambda_{k+1}^i U_{\langle \mathrm{v}_{k+1}^i \rangle}(\mathrm{v}_{k+1}) \tag{19}$$

Thus, the transition probability $p(\mathrm{x}_{k+1}|\mathrm{x}_k)$ can be further developed into

$$\begin{aligned} &p(\mathrm{x}_{k+1}|\mathrm{x}_k) \\ &= \int p(\mathrm{x}_{k+1}|\mathrm{x}_k, \mathrm{v}_{k+1}) \sum_{i=1}^{R} \lambda_{k+1}^i U_{\langle \mathrm{v}_{k+1}^i \rangle}(\mathrm{v}_{k+1}) d\mathrm{v}_{k+1} \\ &= \sum_{i=1}^{R} \lambda_{k+1}^i \frac{1}{|\langle \mathrm{v}_k^i \rangle|} \int_{\langle \mathrm{v}_{k+1}^i \rangle} p(\mathrm{x}_{k+1}|\mathrm{x}_k, \mathrm{v}_{k+1}) d\mathrm{v}_{k+1} \end{aligned} \tag{20}$$

Combining (18) and (20) leads to the expression

$$\begin{aligned} p(\mathrm{x}_{k+1}|\mathrm{z}_{1,k}) = &\sum_{i=1}^{N} \sum_{j=1}^{R} \omega_k^i \lambda_{k+1}^i \frac{1}{|\langle \mathrm{x}_{k+1}^j \rangle|} \frac{1}{|\langle \mathrm{x}_k^i \rangle|} \\ &\cdot \underbrace{\left(\int_{\langle \mathrm{x}_k^i \rangle} \int_{\langle \mathrm{v}_{k+1}^j \rangle} p(\mathrm{x}_{k+1}|\mathrm{x}_k, \mathrm{v}_{k+1}) d\mathrm{x}_k d\mathrm{v}_{k+1} \right)}_{\Theta(\mathrm{x},\mathrm{v})} \end{aligned} \tag{21}$$

Using IA techniques and the Lebesgue measure theory [38], we approximated the term $\Theta(\mathrm{x}, \mathrm{v})$ by a sum of constant functions with ball supports. To obtain the analytic form, for any inclusion function $[f]$ in Box-PF, the support for the PDF terms can be approximated by the following:

$$\int p(\mathrm{x}_{k+1} \,|\mathrm{x}_k) U_{[\mathrm{x}_k^i]}(\mathrm{x}_k) d\mathrm{x}_k \approx [f]([\mathrm{x}_k^i], [\mathrm{v}_{k+1}]) \tag{22}$$

Thus, according to (18) and (22), the predictive distribution could be expressed as follows:

$$\begin{aligned} p(\mathrm{x}_{k+1} \,|\mathrm{z}_{1,k}) &\approx \sum_{i=1}^{N} \omega_k^i U_{[f]([\mathrm{x}_k^i],[\mathrm{v}_k])}(\mathrm{x}_{k+1}) \\ &= \sum_{i=1}^{N} \omega_k^i U_{[\mathrm{x}_{k+1}^i]}(\mathrm{x}_{k+1}) \\ &\xrightarrow[[\mathrm{x}_{k+1}^i] \mapsto \langle \mathrm{x}_{k+1}^i \rangle]{\text{construction of ball}} = \sum_{i=1}^{N} \omega_k^i U_{\langle \mathrm{x}_{k+1}^i \rangle}(\mathrm{x}_{k+1}) \end{aligned} \tag{23}$$

Equation (23) shows that $p(\mathrm{x}_{k+1} \,|\mathrm{z}_{1,k})$ can be approximated using the weighted sum of N ball particles $\langle \mathrm{x}_k^i \rangle$ as the support.

- Correction

With the ball measurement $\langle \mathrm{z}_{k+1} \rangle \supseteq h(\mathrm{x}_{k+1}) + \mathrm{w}_{k+1}$, the likelihood function has the following expression with L components weighted with the normalized coefficient β_{k+1}^j:

$$p(\mathrm{z}_{k+1} \,|\mathrm{x}_{k+1}) = \sum_{j=1}^{L} \beta_{k+1}^j U_{\langle \mathrm{z}_{k+1}^j \rangle}(h(\mathrm{x}_{k+1})) \tag{24}$$

where $\bigcup_{j=1}^{L}\left\langle z_{k+1}^{j}\right\rangle=\langle z_{k+1}\rangle$ for a set of ball supports $\left\langle z_{k+1}^{j}\right\rangle$ with $j=1,2,\cdots,L$. Therefore, according to (16), we obtained the following:

$$\begin{aligned}p(\mathrm{x}_{k+1}\,|\mathrm{z}_{1,k+1})&=\frac{1}{\alpha_{k+1}}\sum\nolimits_{j=1}^{L}\beta_{k+1}^{j}U_{\left\langle z_{k+1}^{j}\right\rangle}(h(\mathrm{x}_{k+1}))\\&\cdot\sum\nolimits_{i=1}^{N}\omega_{k+1}^{i}U_{\langle \mathrm{x}_{k+1}\rangle}(\mathrm{x}_{k+1})=\frac{1}{\alpha_{k+1}}\\&\times\sum\nolimits_{j=1}^{L}\sum\nolimits_{i=1}^{N}\beta_{k+1}^{j}\omega_{k+1}^{i}\\&\times\underbrace{U_{\left\langle z_{k+1}^{j}\right\rangle}(h(\mathrm{x}_{k+1}))U_{\left\langle \mathrm{x}_{k+1}^{i}\right\rangle}(\mathrm{x}_{k+1})}_{\psi}\end{aligned}\tag{25}$$

The term ψ is a constant function with the following set as the support:

$$\begin{aligned}SS_{\psi}=\Big\{\mathrm{x}_{k+1}\in\left\langle \mathrm{x}_{k+1}^{i}\right\rangle|\exists \mathrm{w}_{k+1}\in\langle \mathrm{w}_{k+1}\rangle,\\ \textit{such that } h(\mathrm{x}_{k+1},\mathrm{w}_{k+1})\in\left\langle z_{k+1}^{j}\right\rangle\Big\}\end{aligned}\tag{26}$$

The above set defines a CSP; that is, the predicted ball particle $\left\langle \mathrm{x}_{k+1}^{i}\right\rangle$ can be contracted with respect to the relationship between the measurement function h and the ball measurement $\left\langle z_{k+1}^{j}\right\rangle$. Therefore, a set of new balls $\left\langle \hat{x}_{k+1}^{i}\right\rangle$ may be provided to fit $p(\mathrm{z}_{k+1}\,|\mathrm{x}_{k+1})$ after the CSP. Thus, we obtained the following:

$$\begin{aligned}&U_{\left\langle z_{k+1}^{j}\right\rangle}(h(\mathrm{x}_{k+1}))U_{\langle \mathrm{x}_{k+1}\rangle}(\mathrm{x}_{k+1})\\&\quad=U_{\left\langle z_{k+1}^{j}\right\rangle}(h(\mathrm{x}_{k+1}))\frac{1}{\left|\left\langle \mathrm{x}_{k+1}^{i}\right\rangle\right|}\left|SS_{\psi}\right|U_{SS_{\psi}}(\mathrm{x}_{k+1})\end{aligned}\tag{27}$$

Combining (22) with (25), we defined $\left\langle \hat{x}_{k+1}^{i}\right\rangle$ as the minimum ball containing SS_{ψ}, i.e., $\left\langle \hat{x}_{k+1}^{i}\right\rangle=\left\langle SS_{\psi}\right\rangle$. Further, the predictive probability distribution was obtained as follows:

$$\begin{aligned}&p(\mathrm{x}_{k+1}\,|\mathrm{z}_{1,k+1})\\&\quad=\frac{1}{\alpha_{k+1}}\sum\nolimits_{i=1}^{N}\sum\nolimits_{j=1}^{L}\beta_{k+1}^{j}\omega_{k+1}^{i}U_{\left\langle z_{k+1}^{j}\right\rangle}(h(\mathrm{x}_{k+1}))\\&\qquad\times\frac{1}{\left|\left\langle \mathrm{x}_{k+1}^{i}\right\rangle\right|}\left|SS_{\psi}\right|U_{SS_{\psi}}(\mathrm{x}_{k+1})\\&\quad\approx\frac{1}{\alpha_{k+1}}\sum\nolimits_{i=1}^{N}\beta_{k+1}^{j}\omega_{k+1}^{i}\frac{1}{|\langle z_{k+1}\rangle|}\\&\qquad\times\frac{1}{\left|\left\langle \mathrm{x}_{k+1}^{i}\right\rangle\right|}\left|\left\langle \hat{x}_{k+1}^{i}\right\rangle\right|U_{\left\langle \hat{x}_{k+1}^{i}\right\rangle}(\mathrm{x}_{k+1})\\&\quad\propto\sum\nolimits_{i=1}^{N}\omega_{k+1}^{i}\frac{\left|\left\langle \hat{x}_{k+1}^{i}\right\rangle\right|}{\left|\left\langle \mathrm{x}_{k+1}^{i}\right\rangle\right|}U_{\left\langle \hat{x}_{k+1}^{i}\right\rangle}(\mathrm{x}_{k+1})\end{aligned}\tag{28}$$

5) SCENARIOS OF BALL-PF

- Initialization

The state space region in question was divided into N center interval vectors $[\mathrm{x}_{k}^{i}]$, $i=1\cdots N$, with an empty intersection and equivalent weights.

- Time update

According to $[\mathrm{x}_{k}^{i}]$ and input $[\mathrm{u}_{k}]$ at time k, the center interval vectors $[\mathrm{x}_{k+1}^{i}]$ at step $k+1$ were built by $[\mathrm{x}_{k+1}^{i}]=[f]([\mathrm{x}_{k}^{i}],[\mathrm{u}_{k}])$, and then each ball $\left\langle \mathrm{x}_{k+1}^{i}\right\rangle$ was constructed with respect to $[\mathrm{x}_{k+1}^{i}]$.

- Measurement update

For all $[\mathrm{x}_{k+1}^{i}]$, we predicted the box measurements using $[z_{k+1}^{i}]=[h]([\mathrm{x}_{k+1}^{i}])$. Therefore, the real measurement ball $\left\langle \mathrm{y}_{k+1}\right\rangle$ and the prediction measurement ball $\left\langle \mathrm{z}_{k+1}^{i}\right\rangle$ were constructed according to the corresponding center interval vectors (see Fig. 3).

- Firefly intelligent optimization

The core idea of FA combined with Ball-PF is to move the ball particle points towards the high-likelihood region, which mainly involves the following three aspects:

1) Light intensity

As opposed to the conventional FA, we used the real observation value compared with the forecasted observation value of each ball particle, which helped to avoid the computational complexity caused by the light intensity of each ball particle compared with the other ball particles in the current position. Furthermore, for each moment, only one observation was recorded. This led us to reconstruct a light intensity I_{k+1}^{i} for each ball particle. Within the interval framework, it was evaluated as the volume of the intersection ball $\left\langle \varepsilon_{k+1}^{i}\right\rangle$ between the prediction measurement ball $\left\langle \mathrm{z}_{k+1}^{i}\right\rangle$ and the real measurement ball $\left\langle \mathrm{y}_{k+1}\right\rangle$, i.e.

$$I_{k+1}^{i}=\left|\left\langle \varepsilon_{k+1}^{i}\right\rangle\right|=\left|\left\langle \mathrm{z}_{k+1}^{i}\right\rangle\cap\left\langle \mathrm{y}_{k+1}\right\rangle\right|\tag{29}$$

2) Attractiveness

Each firefly has its distinctive attractiveness β which implies how strongly it attracts other members of the swarm. As the global optimal value of the particle swarm was only one, in our work, each ball is only compared with the optimal ball $\left\langle \mathrm{g}_{k+1}\right\rangle$, which helped to further avoid higher-order interactive calculations and the repeated calculations of attractiveness. The computational complexity of this stage was reduced from the original $O(N^2)$ to $O(N)$. Meanwhile, the improved method used the FA to find a better value with fewer iterations and fewer particles, which obviously reduced the computational complexity of Ball-PF.

The attractiveness between any ball particle and the optimal ball $\left\langle \mathrm{g}_{k+1}\right\rangle$ was defined as the exponential function of d_i

$$\beta=\beta_{0}e^{-\gamma d_{i}^{2}}\tag{30}$$

where β_0 and γ are the predetermined parameters maximum attractiveness and absorption coefficient, respectively; in general, $\beta_0\sim[0.8,1]$. The distance between any ball $\left\langle \mathrm{x}_{k+1}^{i}\right\rangle$ and the global optimal ball $\left\langle \mathrm{g}_{k+1}\right\rangle$ was denoted as follows:

$$\begin{aligned}d_{i}=d(\left\langle \mathrm{g}_{k+1}\right\rangle,\left\langle \mathrm{x}_{k+1}^{i}\right\rangle)=\left\|mid(\left\langle \mathrm{g}_{k+1}\right\rangle)-mid(\left\langle \mathrm{x}_{k+1}^{i}\right\rangle)\right\|\\+\left|rad(\left\langle \mathrm{g}_{k+1}\right\rangle)-rad(\left\langle \mathrm{x}_{k+1}^{i}\right\rangle)\right|\end{aligned}\tag{31}$$

3) Position update

For each ball particle, the radius was kept constant by changing the position of the center to make the ball move.

After initialization, the position update for every ball particle was as follows:

$$mid(\langle x_{k+1}^i \rangle) = mid(\langle x_{k+1}^i \rangle) + \beta \times (mid(\langle g_{k+1} \rangle) - mid(\langle x_{k+1}^i \rangle)) + \alpha \times (rand - 1/2) \quad (32)$$

where $rand \sim U(0, 1)$ is a random number obtained from the uniform distribution. The addition of $\alpha \times (rand - 1/2)$ as the disturbance term effectively reduced the probability of falling into a local extremum to some extent, $\alpha \in [0, 1]$.

When the position update was completed, we calculated and compared the light intensity for each ball particle and then, updated the global optimal ball as follows:

$$\langle g_{k+1} \rangle \in \left\{ \langle x_{k+1}^1 \rangle, \langle x_{k+1}^2 \rangle, \cdots, \langle x_{k+1}^N \rangle | I(\langle x \rangle) \right\} = \max_{i=1}^{N} I(\langle x_{k+1}^i \rangle) \quad (33)$$

- Contraction

After the position update, if $\langle \varepsilon_{k+1}^i \rangle$ was not empty, the ball particle $\langle x_{k+1}^i \rangle$ was contracted using the ball contraction algorithm combined with the measurement equation to obtain a new ball particle $\langle x_{k+1}^i \rangle^{new}$. Else, the ball particle remained unchanged $\langle x_{k+1}^i \rangle^{new} = \langle x_{k+1}^i \rangle$. Meanwhile, the new center interval vector $[x_{k+1}^i]^{new}$ was generated using $\langle x_{k+1}^i \rangle^{new}$; we then used it to obtain the interval vector $[x_{k+2}^i]$ at time $k+2$.

- Likelihood

A ball particle for which the predicted ball measurement had no intersection with the real ball measurement was penalized, and a ball particle for which the predicted value was included in the real ball measurement was favored. This led us to construct the following measure of the ball likelihood:

$$A^i = \hat{I}_{k+1}^i(j) / \left| \langle z_{k+1}^i(j) \rangle \right| \quad (34)$$

where $\hat{I}_{k+1}^i(j)$ represents the updated value after the position update.

- Weight update

As the FA changed the position of each ball particle in the state space, the weight was updated along with the ball particle position. We constructed an update to the weights of the ball particle by multiplying the previous weight by every ball's likelihood as follows:

$$\omega_{k+1}^i = A^i \omega_k^i \quad (35)$$

- Normalization

This step was used to handle the normalized weights such that their sum was equal to one, i.e.

$$\omega_{k+1}^i \leftarrow \omega_{k+1}^i / \sum_{j=1}^{N} \omega_{k+1}^j \quad (36)$$

- Estimation

$$\hat{x}_{k+1} = \sum_{i=1}^{N} \omega_{k+1}^i \cdot mid(\langle x_{k+1}^i \rangle) \quad (37)$$

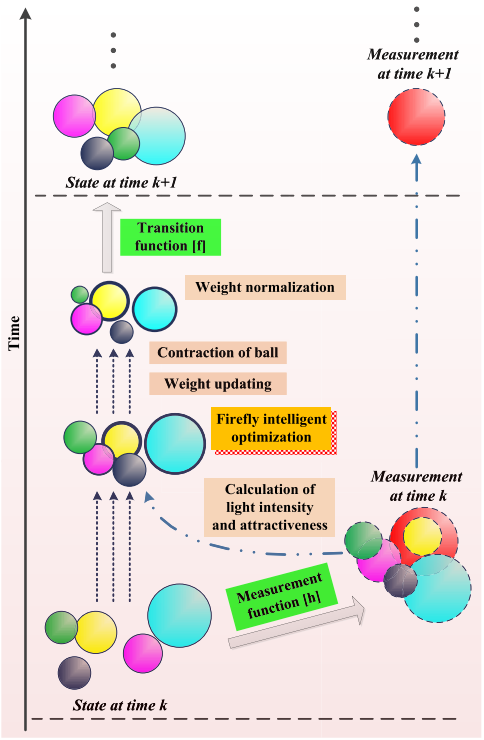


FIGURE 6. Scenarios for the box particle filter.

We could also use the maximum weight estimate. Similarly, according to the definition of the enclosing box in Box-PF, given that the Ball-PF estimation $\hat{x}_{k+1}$ was calculated using N vectors $mid(\langle x_{k+1}^i \rangle)$, another confidence in the estimation based on the confidence of each $mid(\langle x_{k+1}^i \rangle)$ was calculated using a Gaussian-like mixture strategy with

$$\hat{P}_{k+1} = \sum_{i=1}^{N} \omega_{k+1}^i (rad(\langle x_{k+1}^i \rangle) + (\hat{x}_{k+1} - mid(\langle x_{k+1}^i \rangle))(\hat{x}_{k+1} - mid(\langle x_{k+1}^i \rangle))^T) \quad (38)$$

where $\hat{P}_{k+1}$ was the partial confidence generated when using each ball particle center. The Ball-PF algorithm is illustrated in Fig.6, and its procedure is listed in Table 2.

IV. FAST ALGORITHM OF SLAM BASED ON BALL PARTICLE FILTER

According to the conventional FastSLAM algorithm, the proposed Ball-PF based fast SLAM algorithm and the performance evaluation are presented in this section.

A. FASTSLAM ALGORITHM

FastSLAM is a framework using an RBPF, which is based on the following factorization [45]:

$$\underbrace{p(x_{1:t}, M \,|z_{1:t}\,, u_{1:t}, n_{1:t})}_{\text{SLAM posterior}} = \underbrace{p(x_{1:t} \,|z_{1:t}\,, u_{1:t}, n_{1:t})}_{\text{path posterior}} \cdot \underbrace{\prod_{i=1}^{L} p(m_i \,|x_{1:t}\,, z_{1:t}, n_{1:t})}_{\text{landmark posterior}} \quad (39)$$

TABLE 2. Ball particle filter algorithm.

Algorithm 2: Ball Particle Filter Algorithm

Input: $\left\{w_k^i,[\mathrm{x}_k^i]\right\}_{i=1}^N$, and the input $[\mathrm{u}_k]$

Output: State estimation $\hat{x}_k$ and $\hat{P}_k$

1 *Initialization*: Constants $\{\beta_0,\gamma,\alpha\}$, $k=0$, generation N center interval vector $\left\{w_k^i=1/N,[\mathrm{x}_k^i]\right\}_{i=1}^N$

2 *Iteration*: For $i=1\cdots N$

3 *Propagation*: $[\mathrm{x}_{k+1}^i]=[f]([\mathrm{x}_k^i],[\mathrm{u}_k])$

4 *Predicted measurement*: $[\mathrm{z}_{k+1}^i]=[h]([\mathrm{x}_{k+1}^i])$, and construction of $\langle \mathrm{x}_{k+1}^i\rangle$, $\langle \mathrm{z}_{k+1}^i\rangle$, and $\langle \mathrm{y}_{k+1}\rangle$

5 *light intensity*: $I_{k+1}^i=\langle \varepsilon_{k+1}^i\rangle=\langle \mathrm{z}_{k+1}^i\rangle\cap\langle \mathrm{y}_{k+1}\rangle$

6 *Attractiveness*: $\beta=\beta_0 e^{-\gamma d_i^2}$, $d_i=d(\langle \mathrm{g}_{k+1}\rangle,\langle \mathrm{x}_{k+1}^i\rangle)=\left\|mid(\langle \mathrm{g}_{k+1}\rangle)-mid(\langle \mathrm{x}_{k+1}^i\rangle)\right\|+\left|rad(\langle \mathrm{g}_{k+1}\rangle)-rad(\langle \mathrm{x}_{k+1}^i\rangle)\right|$

7 *Update ball position*: $mid(\langle \mathrm{x}_{k+1}^i\rangle)=mid(\langle \mathrm{x}_{k+1}^i\rangle)+\beta\times(mid(\langle \mathrm{g}_{k+1}\rangle)-mid(\langle \mathrm{x}_{k+1}^i\rangle))+\alpha\times(rand-1/2)$

8 *Update optimal value*: $\langle \mathrm{g}_{k+1}\rangle\in\left\{\langle \mathrm{x}_{k+1}^1\rangle,\langle \mathrm{x}_{k+1}^2\rangle,\cdots,\langle \mathrm{x}_{k+1}^N\rangle\middle|I(\langle \mathrm{x}\rangle)\right\}=\max_{i=1}^N I(\langle \mathrm{x}_{k+1}^i\rangle)$

9 *Contraction*: If $I_{k+1}^i(j)\neq\varnothing$, then, contract $\langle \mathrm{x}_{k+1}^i\rangle$ using the ball contraction method and combine it with the measurement equation to obtain $\langle \mathrm{x}_{k+1}^i\rangle^{new}$, Else $\langle \mathrm{x}_{k+1}^i\rangle^{new}=\langle \mathrm{x}_{k+1}^i\rangle$, EndIf

10 *Likelihood*: $A^i=\hat{I}_{k+1}^i(j)\Big/\left|\langle z_{k+1}^i(j)\rangle\right|$

11 *Weight update*: $\omega_{k+1}^i=A^i\omega_k^i$

12 *Weighs normalization*: $\omega_{k+1}^i\leftarrow\omega_{k+1}^i\Big/\sum_{j=1}^N\omega_{k+1}^j$

13 *Estimation*: $\hat{x}_{k+1}=\sum_{i=1}^N\omega_{k+1}^i mid(\langle \mathrm{x}_{k+1}^i\rangle)$, $\hat{P}_{k+1}=\sum_{i=1}^N\omega_{k+1}^i(rad(\langle \mathrm{x}_{k+1}^i\rangle)+(\hat{x}_{k+1}-mid(\langle \mathrm{x}_{k+1}^i\rangle))(\hat{x}_{k+1}-mid(\langle \mathrm{x}_{k+1}^i\rangle))^T)$

14 *Return*: Generate new interval vectors $[\mathrm{x}_{k+1}]^{new}$ based on $\langle \mathrm{x}_{k+1}\rangle^{new}$, $k=k+1$, until $k=k_{end}$

where $x_{1:t}$, $z_{1:t}$, $u_{1:t}$ and $n_{1:t}$ are the robot trajectory, observations, controls, and correspondences, respectively, from the start to time t. m_i is a local map of the i^{th} particle, and M is a global map.

Thus, the posterior $p(x_{1:t}\,|z_{1:t}\,,u_{1:t-1},n_{1:t})$ was solved applying PF, which implied that one particle would represent one potential trajectory over one time step while generating its own map. The posterior $\prod_{i=1}^{L}p(m_i\,|x_{1:t}\,,z_{1:t},n_{1:t})$ was computed analytically when given information on $x_{1:t}$ and $z_{1:t}$.

Presently, FastSLAM is the best-performing probabilistic technique for solving SLAM problems, which improves the system efficiency by reducing the dimensionality of the state space. It could achieve the desired performance for large-scale map calculations in real-time applications because of its parallelized structure [46]. A full SLAM system for mobile robots is shown in Fig. 7. Most of the researchers have been working on improving the performance of state estimation algorithms, whereas IA provides a good solution for dealing with data containing unknown but bounded errors.

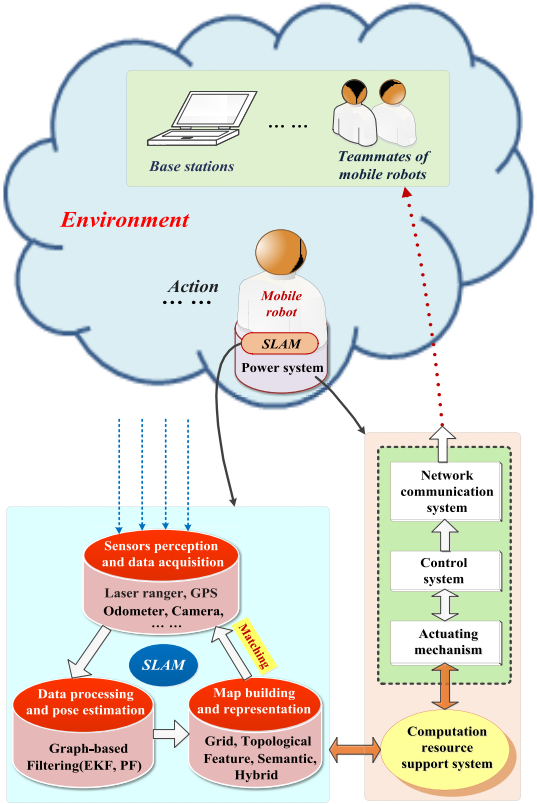


FIGURE 7. Full SLAM system for mobile robots.

According to Section III, the Ball-PF algorithm achieved better estimation accuracy and less computation time than traditional PF. Thus, it was more suitable for the implementation of SLAM under the FastSLAM framework.

B. FAST ALGORITHM OF SLAM BASED ON BALL-PF

The position of a mobile robot was represented by a center interval vector $[\mathrm{x}]=([x]\times[y]\times[\theta])^T$, where $([x]\times[y])^T$ stands for the position of the mobile robot in the global coordinate system, and $[\theta]$ is its orientation with respect to the x axis. Thus, the time-discrete kinematic model was formulated as follows:

$$\begin{cases}[x_{k+1}]=[x_k]+[\Delta d_k]\cos([\theta_k]+1/2[\Delta\theta_k])\\ [y_{k+1}]=[y_k]+[\Delta d_k]\sin([\theta_k]+1/2[\Delta\theta_k])\\ [\theta_{k+1}]=[\theta_k]+[\Delta\theta_k]\end{cases}\tag{40}$$

where input vector $[\mathrm{u}_k]=([\Delta d_k][\Delta\theta_k])^T$ consists of the elementary displacement and the elementary rotation of the mobile robot at time k. More details and the elucidation of parameters are presented in [14] and [45].

The measurement vector $[\mathrm{z}_k]=([r_k][\varphi_k])^T$ is composed of a range of measurements r and bearing measurements φ, which were respectively modeled as follows:

$$\begin{cases}[r]=\sqrt{([m_x]-[x_k])^2+([m_y]-[y_k])^2}\\ [\varphi]=\arctan(\dfrac{[m_y]-[y_k]}{[m_x]-[x_k]})-[\theta_k]\end{cases}\tag{41}$$

TABLE 3. Map matching truth-value table.

Probability	$p_G(c_i) > 0.5$	$p_G(c_i) = 0.5$	$p_G(c_i) < 0.5$
$p_L(c_i) > 0.5$	$\mu+1$	*None*	$\mu-1$
$p_L(c_i) = 0.5$	$\mu-1$	*None*	$\mu-1$
$p_L(c_i) < 0.5$	$\mu-1$	*None*	$\mu+1$

TABLE 4. Fast algorithm of SLAM based on ball-PF.

Algorithm 3: Fast Algorithm of SLAM based on Ball-PF

Input: $\{[x_k^i]\}_{i=1}^N$, $[u_k]$, m_k

Output: Pose estimate x_{k+1}^* and map m_{k+1}

for $i = 1 \cdots N$ do

1 Propagation $[x_{k+1}^i] = [f]([x_k^i], [u_k])$

2 Measure prediction: $[z_{k+1}^i] = [h]([x_{k+1}^i])$

3 Construct ball: $\{\langle x_{k+1}^i \rangle, \langle y_{k+1} \rangle, \langle z_{k+1}^i \rangle\}$

4 Execute FA: $\{I_{k+1}^i, \beta, \langle g_{k+1} \rangle\}$

5 Calculate weight: $\omega^i = e^{\frac{\mu}{\lambda}}$, and $\sum_{i=1}^N \omega^i = 1$

6 Contraction: $\langle x_{k+1}^i \rangle^{new} \leftarrow \langle x_{k+1}^i \rangle$

7 Estimate pose: $x^*_{k+1} = mid(\langle \hat{x}_{k+1}^i \rangle)$, $\omega^i = \max_i^N \omega^i$

8 Update map: $m_{k+1}^i \leftarrow$ Occupancy grid mapping (m_k^i, x_{k+1}^*, y_k)

$k = k+1$ goto step1 until $k = k_{end}$

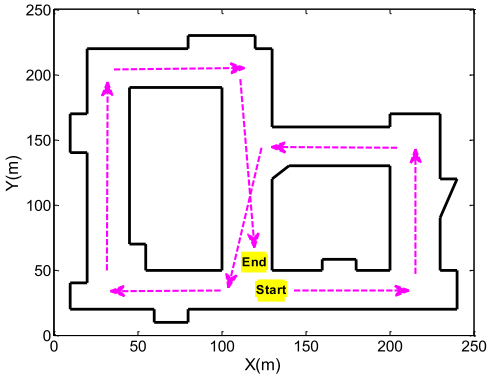


FIGURE 8. Simulation environment map and the motion direction of a mobile robot.

Considering the interval-based uncertainties, the laser range finder provided interval measurements that were expressed as $[y_k] = [y_k - 3\sigma, y_k + 3\sigma]$, where σ denotes the standard deviation. Below, details of the Ball-PF algorithm for the SLAM implementation are presented, and its procedure is described in Table 4.

1) An initial pose estimate of each center interval vector $[x_{k+1}^i]$ was obtained from the previous position center interval vector $[x_k^i]$ in terms of the inclusion function $[f]$. The odometer measurements $[u_k]$ and a realistic approximation of the errors were proposed.

2) The prediction measurements for each ball particle drawn from $[z_{k+1}^i] = [h]([x_{k+1}^i])$. According to

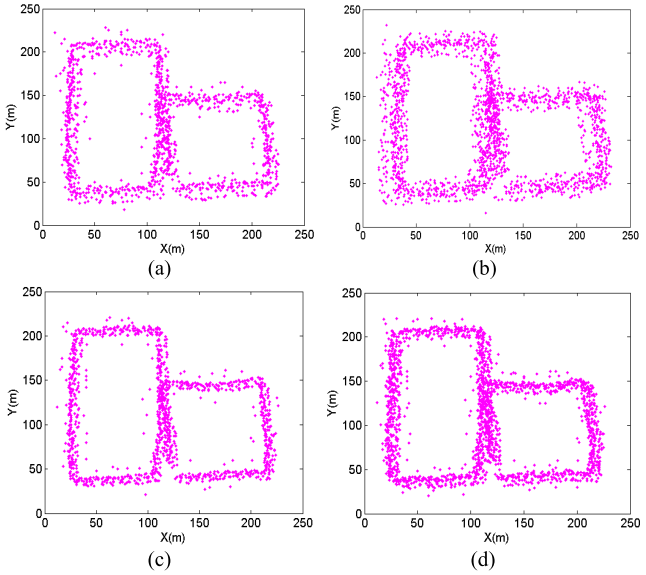


FIGURE 9. Individually estimated values for all box (and ball) particles throughout the process. The magenta "." is the center of the box (or ball) particle. (a) Box-PF (Nb=10). (b) Box-PF (Nb=20). (c) Ball-PF (Nb = 10). (d) Ball-PF (Nb = 20).

$[x_{k+1}^i]$, $[z_{k+1}^i]$, and $[y_{k+1}]$, we constructed the corresponding ball particles $\langle x_{k+1}^i \rangle$, $\langle z_{k+1}^i \rangle$, and $\langle y_{k+1} \rangle$. Thus, each $\langle x_{k+1}^i \rangle$ had an estimated value for its pose and an individual map, meanwhile, the intersection $\langle z_{k+1}^i \rangle \cap \langle y_{k+1} \rangle$ was calculated, and then, the FA was executed.

3) In the applications, each ball particle had its own local and global maps. Through the transformation from the local to the global coordinates of each grid in the local map, we used the overlap of the local map and the global map to calculate the weight for each ball particle, which avoided the complexity of using the likelihood function to calculate the ball particle weights. In fact, the match between the local and global maps was used to compare the probability values of each grid in the local map $p_L(c_i)$ with those of the corresponding global map $p_G(c_i)$. For each cell c, the occupancy grids stored the probability $p(c)$ of being occupied by an obstacle.

One often assumes that the prior occupancy is 0.5, where $p(c) > 0.5$ indicates occupancy, and $p(c) < 0.5$ indicates non-occupancy. Table 3 shows that only $p_L(c_i)$ and $p_G(c_i)$ simultaneous satisfied the condition of > 0.5 or < 0.5; then, $\mu + 1$; else, $\mu - 1$. Thus, the weight of the i^{th} ball particle was calculated as follows:

$$\omega^i = e^{\frac{\mu}{\lambda}} \tag{42}$$

where μ denotes the matching results, and λ is a constant used to control the divergence rate of the weights. After obtaining the weight of each ball particle, we normalized the particle weight, that is, satisfied the following condition:

$$\sum_{i=1}^N \omega^i = 1 \tag{43}$$

4) As the laser scan data with high accuracy, here, we inserted the latest data scan data $y_k = mid([y_k])$ and

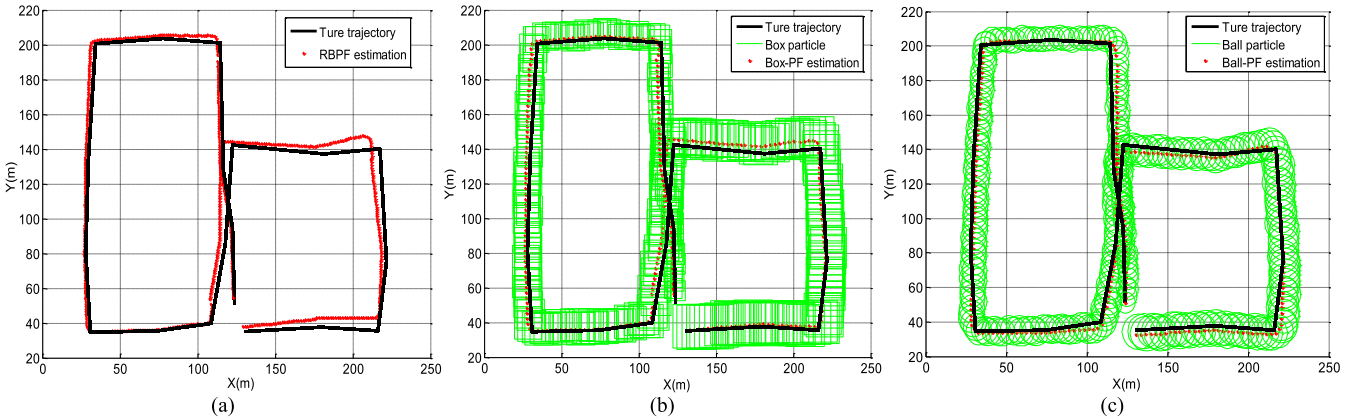


FIGURE 10. Estimated trajectories for different algorithms. Np represents the number of particles for RBPF; Nb represents the number of particles for Box-PF (and Ball-PF). (a) RBPF SLAM (Np = 100). (b) Box-PF SLAM (Nb = 20). (c) Ball-PF SLAM (Nb = 20)

the existing maps (m_x, m_y) into (41), along with the ball contraction algorithm to contract the ball particles $\langle \mathrm{x}_{k+1}^i \rangle$. The ball particle map was updated with the final estimated position and the laser scan data, and then, the largest weight ball particle $\langle \hat{\mathrm{x}}_{k+1}^l \rangle$ was selected as the localization result.

5) The occupancy grid map of each ball particle was updated using "mapping with known poses", given the knowledge of $x_{1:k}$ and $y_{1:k}$. In this study, we used the approach proposed in [45].

C. PERFORMANCE ASSESSMENT

To assess the performance of Ball-PF, we used the criteria similar to the ones proposed by Gning *et al.* [39] for the assessment of Box-PF: *inclusion* and *volume*. Therefore, the optimal Ball-PF for a SLAM problem had to satisfy the following two conditions:

- The true value of the pose state vector must be contained in the support of the posterior spatial PDF.
- The volume of the support of the posterior spatial PDF should be minimal.

In Ball-PF, the credible set $C_k(1)$ was approximated simply by the union of all of the ball particles, that is,

$$C_k(1) = \bigcup_{i=1}^{N} \left\langle \mathrm{x}_k^i \right\rangle \tag{44}$$

Thus, the *inclusion criterion* ρ_k followed directly from [6] as follows:

$$\rho_k = \begin{cases} 1, & C_k(1) = \bigcup_{i=1}^{N} \langle \mathrm{x}_k^i \rangle \\ 0, & \text{otherwise} \end{cases} \tag{45}$$

The *volume criterion* v_k was approximated the volume of $C_k(1)$ by the spread of the ball particles. In practice, v_k was approximated by the covariance $\hat{P}_k$ defined in (38), i.e.

$$v_k = \hat{P}_k \tag{46}$$

The failure to satisfy the inclusion criteria indicated the filter divergence. If the average inclusion was 1, the true value of the state was consistently contained within the particle support set.

Furthermore, to apply the root-mean-square error (RMSE) metric to Box-PF and Ball-PF, we used the center points of the box or ball to evaluate the performance.

$$X_{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(x_k^i - x_k^* \right)^2} \tag{47}$$

where N is the number of particles, x_k^* is the true pose, and x_k^i is the center of the i^{th} ball or box at time k.

V. SIMULATION EXPERIMENTS AND COMPARATIVE ANALYSIS

The simulation studies and a case study with mobile robots for the proposed algorithm are given in this section. All of the experiments were performed on a mobile computer with an Intel®Core™i5-5200U CPU @ 2.20 GHz with 8 GB of RAM running under Windows 7. The simulation experiments were implemented using MATLAB (2014a) and based on the INTLAB9.0 toolbox, which contains a number of built-in routines for the interval calculations. The mobile robot experiment was conducted using Ubuntu14.04 and a robot operating system (ROS).

A. SIMULATION EXPERIMENT AND COMPARATIVE ANALYSIS

To verify the performance of the proposed algorithm, we compared the conventional RBPF SLAM, Box-PF SLAM, and Ball-PF SLAM through simulation experiments in view of the following two different typical situations of maps.

1) GRID MAPS

In this study, the performance evaluation was conducted using the simulated environment that we created, and with some simulated measurement data. Firstly, we designed an environment map with an area of approximately 250 m^2 and a resulting grid map resolution of 10cm^2/cell. Meanwhile, we generated a real trajectory reference for the mobile robot to

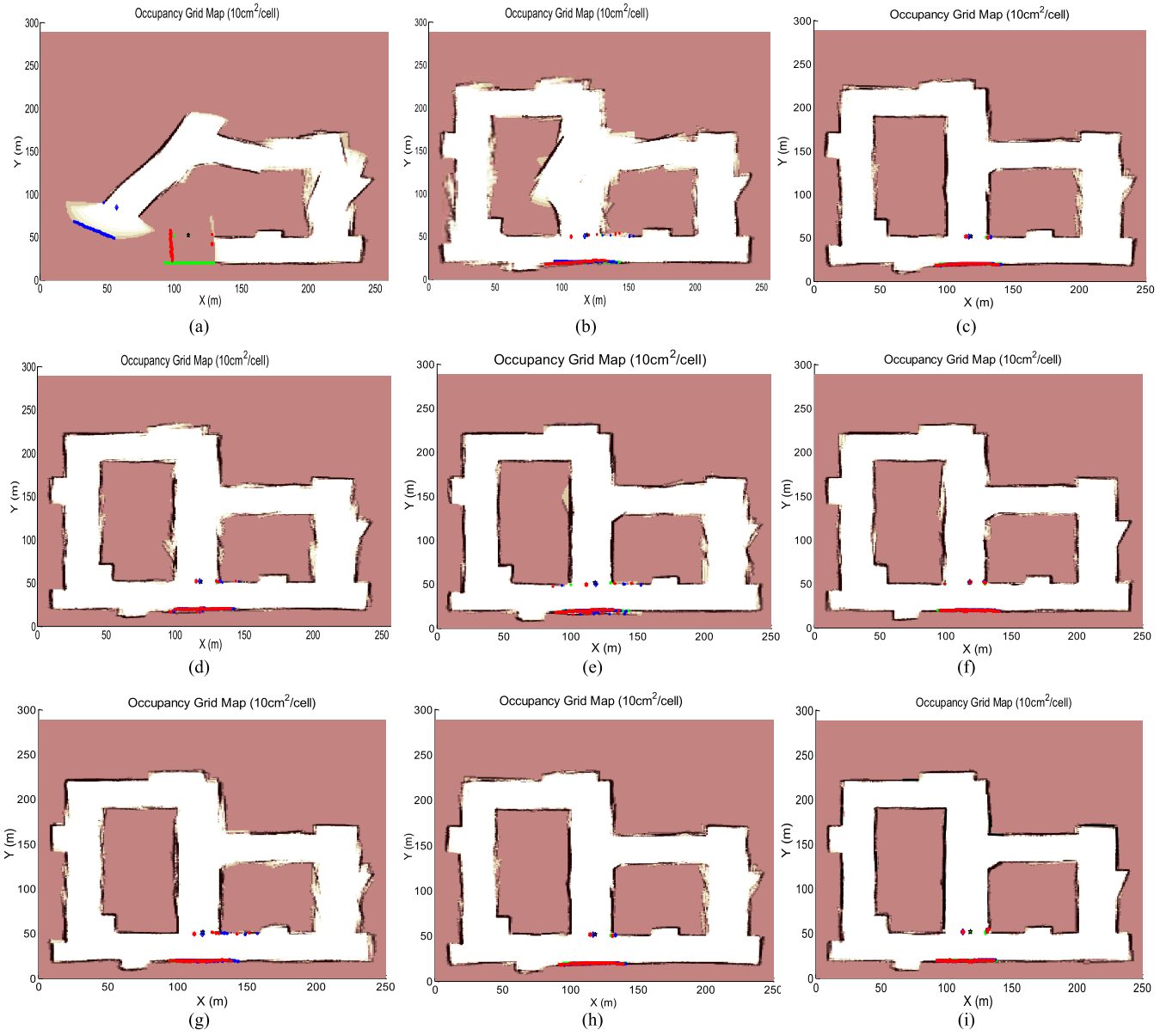


FIGURE 11. 2D floor plan occupation grid map generated after running different SLAM algorithms using different numbers of particles. (a) RBPF SLAM (Np = 10). (b) RBPF SLAM (Np = 20). (c) RBPF SLAM (Np = 100). (d) Box-PF SLAM (Nb = 10). (e) Box-PF SLAM (Nb = 15). (f) Box-PF SLAM (Nb = 20). (g) Ball-PF SLAM (Nb = 10). (h) Ball-PF SLAM (Nb = 15). (i) Ball-PF SLAM (Nb = 20).

follow, as illustrated in Fig. 8. To observe the effect of the SLAM algorithm more accurately, we added some depression features in the straight corridor. Then, we simulated the odometer and the laser range finder measurements to which we added some noise to produce the final actual measurement data.

A kinematics model (40) was used in this simulation. The mobile robot equipped with a laser rangefinder with a maximum range of 20 m and a frontal field-of-view of 180°. The mobile robot moved at the speed of 4 m/s and with a maximum steering angle of 30°. The control frequency was 40 Hz, and observation scans were obtained at 5 Hz. The distance measurement and the angular accuracy of the laser rangefinder were assumed to be ±50 mm and ±0.125°, respectively. The FA parameters were $\beta_0 = 0.9, \gamma = 1$, and $\alpha = 0.45$. At each iteration step, one-third of the laser measurements were chosen randomly for the localization, from which one-third were allowed to be outliers. For the regular ball operator $\lambda = 0.3$, and $\Lambda = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$.

In the simulations, the number of particles was set as 10, 15, 20, and 100. Fig. 10 shows that Ball-PF provided better contraction results than Box-PF at each step. The two interval-based methods provided consistency and guarantee results during the entire process. In Fig. 11, to successfully build the environment's 2D floor plan occupation grid maps,

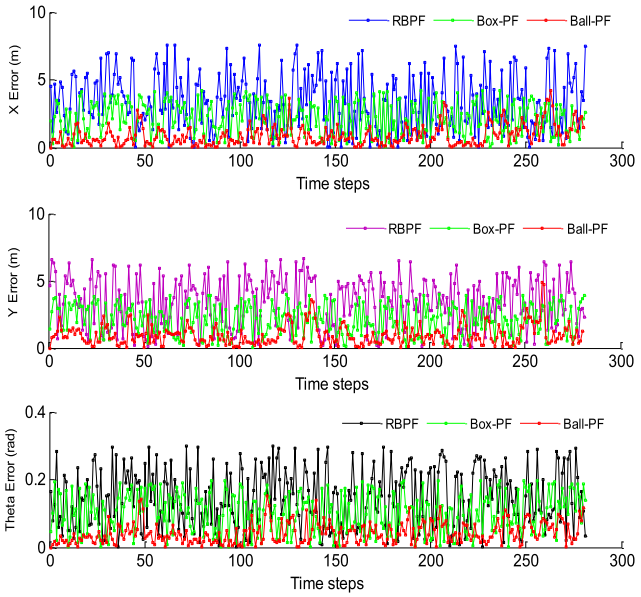


FIGURE 12. Absolute value of the estimated errors over the filtering runs for the PF, Box-PF, and Ball-PF methods using 100, 20, and 20 particles, respectively. Top: x-errors. Middle: y-errors. Bottom: theta errors.

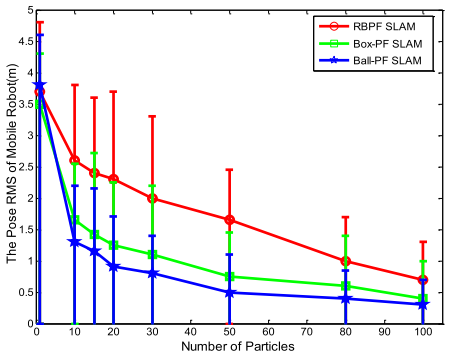


FIGURE 13. RMSE vs. number of particles.

the Ball-PF needed only 20 ball particles to outperform the RBPF with 100 particles, and performed better than the Box-PF. For RBPF SLAM, obvious errors were observed when 20 particles were used and the mapping results were inconsistent.

Fig. 9 shows that the box particles were more dispersed than the ball particles near the true value, which led to the degeneracy phenomenon occurring in Box-PF and required and random subdivision resampling. Instead, the Ball-PF used firefly intelligent optimization to move the ball particles towards the high-likelihood region and reasonably retain parts of the ball particles in the low-likelihood region. Thus, it maintained the variety of ball particles as well as ensured the necessary number of effective ball particles. As compared to the RBPF, the Ball-PF and the Box-PF used bounded error methods; thus, the error was transmitted directly. As the error was not abandoned or approximate, the effect of the errors on the location results was reduced. Fig. 12 shows a plot of the absolute value of the estimated error for the three algorithms.

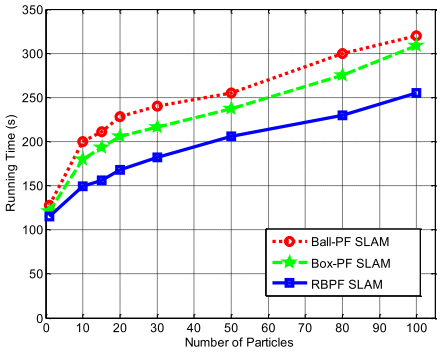


FIGURE 14. Running time vs. number of particles.

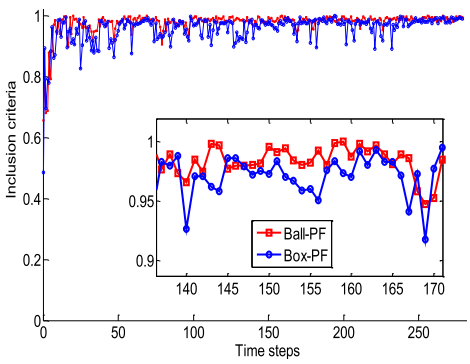


FIGURE 15. Inclusion values.

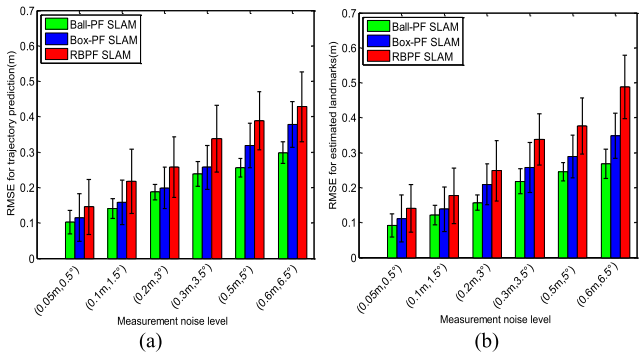


FIGURE 16. RMSE with varying levels of measurement noise (Np=100, Nb=20). (a) RMSE for trajectory prediction. (b) RMSE for estimated landmark.

As can be seen from Fig. 13, Ball-PF SLAM could achieve the error rate of a 100 particle RBPF SLAM while using only 20 particles. Thus, at the same accuracy, the Ball-PF reduced the computational load and improved the real-time performance. Meanwhile, Ball-PF SLAM had better accuracy than the other two algorithms with the same number of particles. However, because the posterior probability of the uniform distribution was fitted in both Ball-PF and Box-PF, it was impossible to improve the accuracy considerably by simply increasing the number of particles.

Fig. 14 shows that the two interval-based SLAM methods were inferior to RBPF SLAM in terms of the time cost for

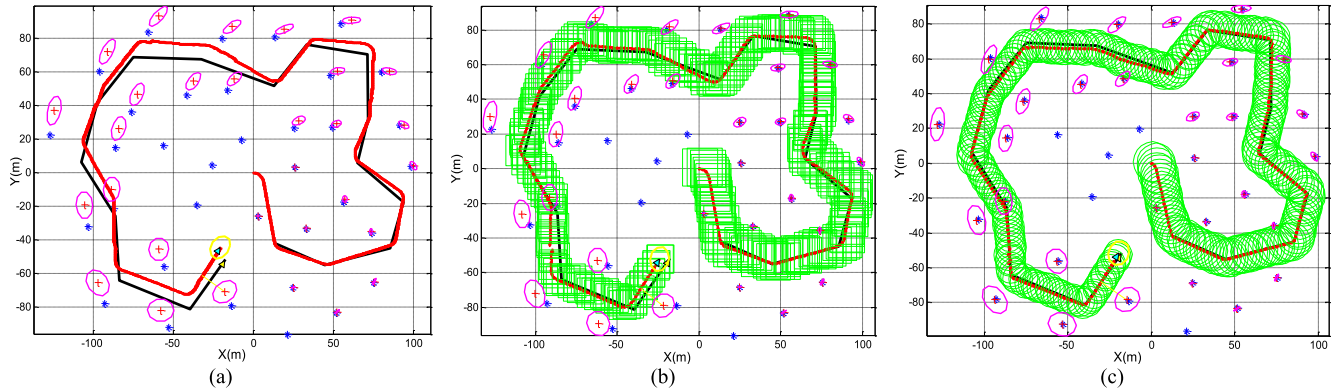


FIGURE 17. Estimated and true mobile robot trajectories with the estimated and true landmarks for different SLAM algorithms. The black line and blue "∗" denote the true trajectory and the setting landmark positions, respectively. The red line is the mean estimate of the mobile robot, and the red "+" denotes the estimated landmark. The magenta ellipse is the covariance of the estimated landmarks. (a) RBPF SLAM (Np = 100). (b) Box-PF SLAM (Nb = 20). (c) Ball-PF SLAM (Nb = 20).

TABLE 5. Comparison of RBPF, Box-PF and Ball-PF methods.

Algorithm	RBPF	Box-PF	Ball-PF
Number of particles	100	20	20
Execution time (s)	251.485	213.061	230.792
X –RMSE (m)	5.248	2.737	1.821
Y–RMSE (m)	4.325	2.129	1.352
Theta–RMSE (rad)	0.376	0.312	0.204

TABLE 6. Computational cost of different SLAM algorithms.

Particle number	RBPF SLAM		Box-PF SLAM		Ball-PF SLAM	
	CPU time(s)	Average time(s)	CPU time(s)	Average time(s)	CPU time(s)	Average time(s)
1	0.102	32.719	0.121	41.189	0.127	50.352
10	0.111	46.133	0.178	54.344	0.193	61.804
20	0.161	54.459	0.283	69.014	0.301	77.941
50	0.227	65.049	0.397	85.043	0.412	98.020
80	0.288	76.620	0.468	103.105	0.505	116.697
100	0.352	90.242	0.501	134.198	0.573	145.973

the same number of particles. The running time of RBPF SLAM was proportional to the number of particles, whereas, the main time consumption of the Box-PF and Ball-PF methods were used for the calculation of the intervals. The time consumption of Ball-PF was slightly more than that of Box-PF for the same number of particles, but the accuracy of Ball-PF was better than Box-PF, which implied that Ball-PF was more suitable for use in cases with a small number of particles.

As illustrated in Fig. 15, according to the inclusion values, Ball-PF was sufficient to satisfy the inclusion criterion. This was a useful advantage of the Ball-PF implementation as compared to the Box-PF implementation. The comparison results of more than 10 simulation runs for the three algorithms are reported in Tab. V. A smaller number of particles and less time were required to obtain the same estimation accuracy with the Ball-PF algorithm. Therefore, the proposed algorithm reduced the computational cost and decreased the error.

2) LANDMARK MAPS

In this study, a simulator developed by Bailey [47] was used to evaluate the performance of Ball-PF for the SLAM approach. We assumed that the data association was unknown and the individual compatibility nearest neighbor (ICNN) method was used for the association [48]. The simulation environment had an area of 250 m × 200 m with 35 landmarks. The control noise and the measurement noise were respectively set to $(0.3m/s, 3°)$ and $(0.2m, 4°)$.

Fig. 17 shows a comparison of the trajectory estimation and the landmark position estimation between three different algorithms. The estimated trajectory by the Box-PF SLAM and Ball-PF SLAM was better than that by RBPF SLAM, and the trajectory estimated by Ball-PF SLAM was closer to the true trajectory than that estimated by Box-PF SLAM. The estimated landmarks by RBPF SLAM showed large alignment errors, whereas that estimated by the Ball-PF SLAM was almost the same as the actual landmark.

To compare the performance and the accuracy of all of the mentioned SLAM algorithms, at the different measurement noise levels, we set six groups of measurement noise data for each simulation. For each measurement noise level, the mean and the standard deviation of the RMSE were calculated over 20 runs for each SLAM algorithm. We could reasonably to conclude from Fig. 16 that the mean and the standard deviation for Ball-PF SLAM increased at a slower rate than the other two algorithms for each measurement noise level. Hence, the ability of Ball-PF SLAM to suppress noise was stronger than that of RBPF SLAM and Box-PF SLAM with respect to the increasing measurement noise levels.

The CPU time for the full SLAM process and the average time of all of the filtering steps of different SLAM approaches over 20 simulation runs are reported in Tab. VI. We concluded that the average time of two interval particle based SLAM

FIGURE 18. Wheel-leg compound mobile robot and its software console interface.

FIGURE 19. Experiment scene.

methods was longer than that of RBPF SLAM at the same number of particles. However, according to Fig. 13, we found that a smaller number of particles for the interval based SLAM algorithm were needed to obtain the same or higher estimation accuracy than RBPF SLAM (Np = 100, Nb = 20). This indicated that the Ball-PF SLAM algorithm performed well in terms of the computational efficiency over RBPF SLAM at the same accuracy.

B. A CASE STUDY WITH WHEEL-LEG COMPOUND MOBILE ROBOT

The proposed approach was implemented and tested using a wheel-leg compound mobile robot (dimensions: 515 cm × 481 cm × 285.5 cm, weight: 5 kg) equipped with an ASUS-Xtion depth camera.

As shown in Fig. 18, the walking gait and the motion mode of the mobile robot were set by the software console and sent the control instruction to the mobile robot to move along the reference trajectory. The experimental site was a conference room located in the New Main Building of Beihang University, Beijing, and the experimental scene is depicted in Fig. 19.

During the experiment, the mobile robot was set to the wheel motion mode and the real data were gathered with a depth camera. To reduce the computational complexity,

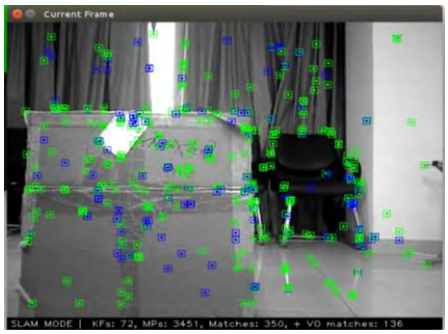


FIGURE 20. ORB feature poins.

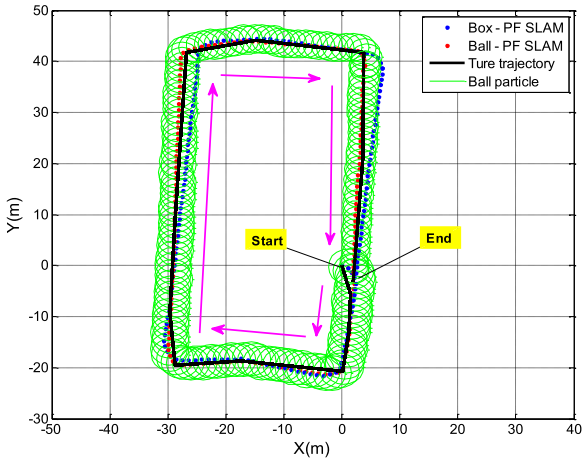


FIGURE 21. Estimated trajectory for different algorithms (Nb = 20).

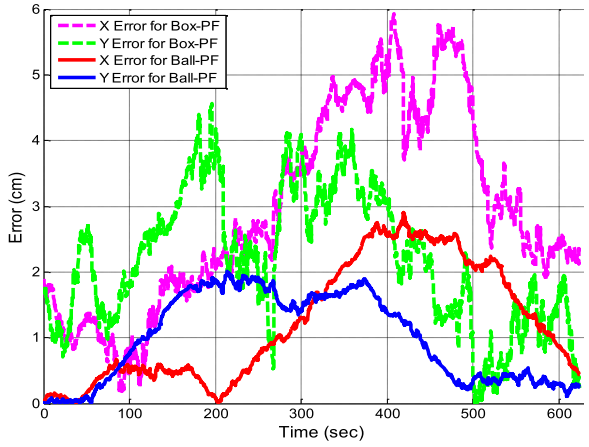
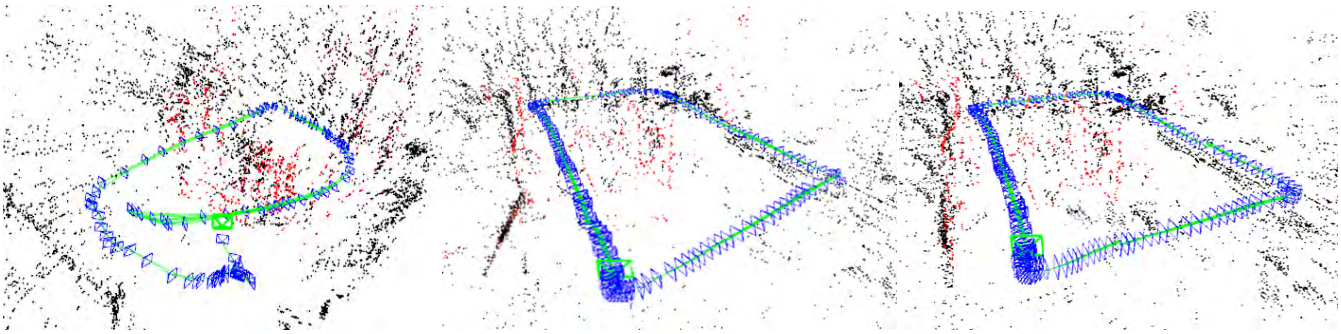


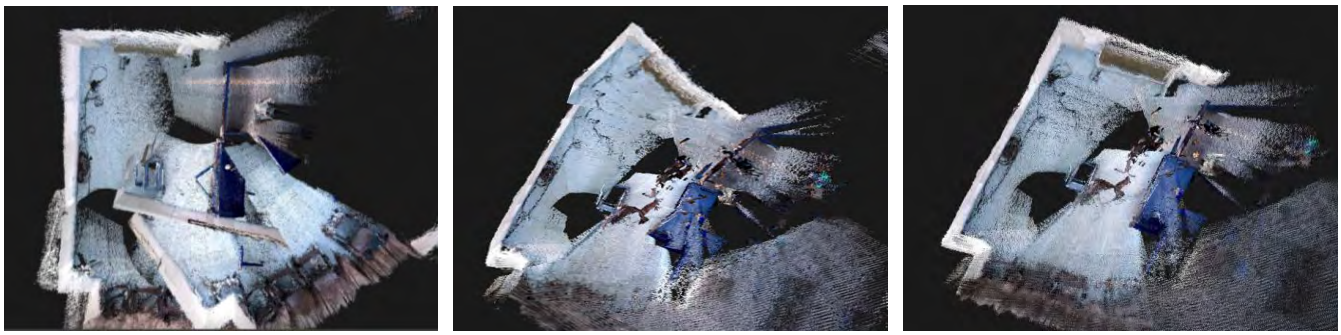
FIGURE 22. Error analysis for different algorithms.

we selected the key frames for the image sequence according to a certain step length [49] and then, opted for the oriented FAST and rotated BRIEF (ORB) feature points [50] (see Fig. 20) to detect and extract landmark information from an image. Remarkably, the ORB feature provided results with a very low error rate and good performance in real time. Thus, each key frame corresponded to a set of 3D feature point clouds. The observation model between a 3D point P(x, y, z)



(a) (b) (c)

FIGURE 23. Sparse point cloud maps and trajectory estimation for different SLAM algorithms. Point clouds (red and black points), key frames (blue rectangles), and current location of the depth camera (green rectangle). (a) Visual odometry. (b) Box-PF SLAM (Nb = 20). (c) Ball-PF SLAM (Nb = 20).



(a) (b) (c)

FIGURE 24. Reconstructed 3D dense maps for different SLAM algorithms. (a) Visual odometry. (b) Box-PF SLAM (Nb = 20). (c) Ball-PF SLAM (Nb = 20).

and the image coordinates P(u, v) was given by the pinhole camera model. The landmark locations were feature point positions estimated from the observations of the environment in the image. The iterative closest point (ICP) algorithm was used to implement the point cloud data registration [51].

Three experiments were carried out using the visual odometry (VO) method [52], Box-PF SLAM algorithm and Ball-PF SLAM algorithm. Fig. 21 illustrates the result of the trajectory estimation for different algorithms performed in a loop. We observed that the trajectory defined by the location balls correctly followed the reference trajectory and all of the reference trajectory points were included in the localization balls; the localization balls were thus consistent. The Ball-PF relied on its superiority to provide guaranteed results. In contrast, the estimated trajectory for the Box-PF showed obvious deviations in some places. For the entire process, the VO took 251.26 s, the Box-PF SLAM took 239.47 s with 20 box-particles and the Ball-PF SLAM took 246.83 s with 20 ball-particles.

From Fig. 22, the maximum error of the pose estimation for the Box-PF SLAM algorithm in the X-direction was approximately 6 cm and over 4.5 cm in the Y-direction, whereas, the maximum error of the pose estimation for the Ball-PF SLAM algorithm in the X-direction was no more than 3 cm and approximately 2 cm in the Y-direction. Fig. 23 shows the trajectory estimation and the corresponding sparse point cloud maps for the VO method, Box-PF SLAM algorithm and Ball-PF SLAM algorithm. Clearly, because of a lack of the added loop closure detection, the cumulative errors of registration between contiguous key frames resulted in a larger error for VO, whereas the Ball-PF SLAM provided the guaranteed result by using interval techniques, and the estimation accuracy was higher than that for the Box-PF SLAM. This further showed that the proposed method could largely eliminate the error caused by no loop closure detection.

We also used OctoMap [53] splicing point clouds to reconstruct the dense maps for the three SLAM methods to display the final effect of mapping. From Fig. 24, the reconstructed 3D dense map of the Box-PF had obvious deviation and inconsistency due to the error caused by the trajectory estimation. The reconstructed 3D map was consistent with the actual 3D environment based on the estimated trajectory of the Ball-PF, and no map overlap and distortion was observed. Consequently, the experimental results demonstrated the feasibility and the effectiveness of Ball-PF for the implementation of SLAM.

VI. CONCLUSION AND REMARKS

The work described in this paper is a modification of the Box-PF algorithm; that is, Ball-PF is presented and applied

TABLE 7. Elementary operations for center interval.

Notation	Mathematical expression
$[x]\pm[y]$	$=[x_0 \pm y_0,\ r_x + r_y]$
$[x]\cdot[y]$	$=[(a+b)/2,\ (a-b)/2]$, where
	$a = \max((x_0 - r_x)(y_0 - r_y),\ (x_0 - r_x)(y_0 + r_y),$
	$(x_0 + r_x)(y_0 - r_y),\ (x_0 + r_x)(y_0 + r_y))$
	$b = \min((x_0 - r_x)(y_0 - r_y),\ (x_0 - r_x)(y_0 + r_y),$
	$(x_0 + r_x)(y_0 - r_y),\ (x_0 + r_x)(y_0 + r_y))$
$[x]^{-1}$	$=[x_0/(x_0^2 - r_x^2),\ r_x/(x_0^2 - r_x^2)]$ for $0 \notin [x]$
$[y]/[x]$	$=[y]\cdot(1/[x])$, for $0 \notin [x]$

to SLAM. Unlike the traditional FastSLAM algorithm, the approach calculates the posterior of a mobile robot's pose on the basis of Ball-PF. The proposed method was validated using a series of simulation experiments and exhibited good results. Although the proposed algorithm exhibited good results, several points should be noted. Because each ball particle was seen as a uniform distribution approximation, simply increasing the number of ball particles caused no significant improvement in the filtering accuracy. Thus, ways to optimize the balls and select the probability density functions should be the main future research direction, such as by using Gaussian distributions. Indeed, solutions to this problem are beneficial for building more accurate maps with fewer particles for SLAM. On the basis of this work, we considered using a Dirichlet process (DP) to determine the number of components, to further answer the question about how to determine the number of ball particles. In addition, the choice of λ and Λ in the ball contraction operator required deliberation, as their values were not unique. In some cases, several numerical experiments were required, and in the other cases, it was easier if the mapping of f involved a strongly monotone operator (see Appendix, Section C).

APPENDIX

A. UNIFORM DISTRIBUTION IN BALL

Known radius $R_0(R_0 > 0)$, Γ is gamma function, the volume of n-dimensional ball $x_1^2 + x_2^2 + \cdots + x_n^2 \le R_0^2$ is

$$V_{R_0} = \frac{\pi^{\frac{n}{2}} R_0^n}{\Gamma(\frac{n}{2}+1)}$$

Definition: If the probability density function of n-dimensional random vector $(X_1, X_2, \cdots, X_n)$ can be defined as

$$g(x_1, x_2, \cdots, x_3) = \begin{cases} \dfrac{\Gamma(\frac{n}{2}+1)}{\pi^{\frac{n}{2}} R_0^n}, & x_1^2+x_2^2+\cdots+x_n^2 \le R_0^2 \\ 0, & x_1^2+x_2^2+\cdots+x_n^2 > R_0^2 \end{cases}$$

Call $(X_1, X_2, \cdots, X_n)$ obeys the uniform distribution in $x_1^2 + x_2^2 + \cdots + x_n^2 \le R_0^2$.

B. DEFINITION OF BALL CONTRACTION OPERATOR

Definition 1: Let Λ be a nonsingular $n \times n$ matrix, $\lambda \in R$ with $0 \le \lambda < 1$, the *regular ball operator* Q is defined by the ball valued operator

$$\begin{cases} Q : R^n \to \aleph_{n0}(R^n) \bigcup \{0\} \\ Qx = \langle \Lambda x; \lambda \left\| \Lambda x \right\| \rangle, \quad \text{for } x \in R^n \end{cases}$$

where $\aleph(R^n)$ denotes the set of all balls $\langle x \rangle \in R^n$.

The operator Q will also be denoted by $Q = \langle\langle \Lambda, \lambda \rangle\rangle$.

Definition 2: Let $f : D \subseteq R^n \to R^n, D = \left\{ x \in R^n \,\middle|\, \left\| x - x' \right\|_2 \le d, 0 \le d \in R \right\}$ is n-dimensional closed ball. The regular ball operator $Q = \langle\langle \Lambda, \lambda \rangle\rangle$ corresponding to set $A = \langle a, r \rangle$, Define the *ball contraction operator* C by

$$\begin{cases} C : A \to \aleph(R^n) \\ Cx = x - Qf(x) = \langle x - \Lambda f(x); \lambda \left\| \Lambda f(x) \right\| \rangle, \quad x \in R^n \end{cases}$$

where $\aleph_{n0}(R^n)$ denotes the set of all balls $X \in R^n$ for which $0 \notin X$.

C. DEFINITION OF BALL OPERATOR WITH DIFFERENT MAPPING

1) Provided that mapping $f : D \subset R^n \to R^n$ has a *strongly monotone operator*, i.e. $\exists \alpha > 0, \forall x, y \in D$, satisfaction:

$$(f(x) - f(y), x - y) \ge \alpha \left\| x - y \right\|^2.$$

Redefine ball operator P as

$$\begin{cases} P : D \to \aleph_n(R^n) \\ Px = \left\langle x - \dfrac{1}{2\alpha} f(x); \dfrac{1}{2\alpha} \left\| f(x) \right\| \right\rangle, \quad \forall x \in D \end{cases}$$

2) Provided that the *strongly monotone operator* $f : D \subset R^n \to R^n$ also satisfies the Lipschitz condition, L is the Lipschitz constant. Let

$$c_1 = \frac{\alpha}{L^2}, p_1 = \left(\frac{1}{\alpha^2} - \frac{1}{L^2} \right)^{\frac{1}{2}}, \quad \lambda_1 = \frac{p_1}{c_1}, (0 < \lambda_1 < 1).$$

Definition of ball operator $L = \langle\langle c_1, p_1 \rangle\rangle$

$$\begin{cases} L : R^n \to \aleph_n(R^n) \\ Lx = \langle c_1 x; p_1 \left\| x \right\| \rangle, \quad \forall x \in R^n \end{cases}$$

According to L, define operator Q:

$$\begin{cases} Q : B_0 \to \aleph_n(R^n) \\ Qx = x - Lf(x) = \langle x - c_1 f(x); p_1 \left\| f(x) \right\| \rangle, \quad \forall x \in B_0 \end{cases}$$

REFERENCES

[1] R. C. Smith and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *Int. J. Robot. Res.*, vol. 5, no. 4, pp. 56–68, 1986.

[2] M. A. Paskin, "Thin junction tree filters for simultaneous localization and mapping," in *Proc. 18th Int. Joint Conf. Artif. Intell.*, 2003, pp. 1–48.

[3] S. Thrun, D. Koller, and Z. Ghahramani, "Simultaneous localization and mapping with sparse extended information filters: Theory and initial results," *STAR*, vol. 7, no. 1, pp. 363–380, 2002.

[4] K. P. Murphy, "Bayesian map learning in dynamic environments," in *Proc. NIPS*, 2000, pp. 1016–1021.

[5] K. Murphy and S. Russell, "Rao-Blackwellised particle filtering for dynamic Bayesian networks," in *Sequential Monte Carlo Methods in Practice*. New York, NY, USA: Springer, 2001, pp. 499–515.

[6] A. Stentz, D. Fox, M. Montemerlo, and M. Montemerlo, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proc. 18th Nat. Conf. Artif. Intell.*, Edmonton, AB, Canada, 2002, pp. 1–6.
[7] D. Hahnel, W. Burgard, D. Fox, and S. Thrun, "An efficient fastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Las Vegas, NV, USA, Oct. 2003, pp. 206–211.
[8] A. Eliazar and R. Parr, "DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks," in *Proc. 18th Int. Joint Conf. Artif. Intell.*, 2003, pp. 1135–1142.
[9] A. Diosi and L. Kleeman, "Laser scan matching in polar coordinates with application to SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Edmonton, AB, Canada, Aug. 2005, pp. 3317–3322.
[10] G. Grisettiyz, C. Stachniss, and W. Burgard, "Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling," in *Proc. IEEE. Int. Conf. Robot. Autom.*, Barcelona, Spain, Apr. 2005, pp. 2432–2437.
[11] A. R. Khairuddin, M. S. Talib, H. Haron, and M. Y. C. Abdullah, "GA-PSO-FASTSLAM: A Hybrid optimization approach in improving FastSLAM performance," in *Proc. Int. Conf. Intell. Syst. Design Appl.*, 2016, pp. 57–66.
[12] A. Lambert, D. Gruyer, B. Vincke, and E. Seignez, "Consistent outdoor vehicle localization by bounded-error state estimation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, St. Louis, MO, USA, Oct. 2009, pp. 1211–1216.
[13] T. Bailey, J. Nieto, and E. Nebot, "Consistency of the FastSLAM algorithm," in *Proc. IEEE Int. Conf. Robot. Autom.*, Orlando, FL, USA, May 2006, pp. 424–429.
[14] B. Vincke, A. Lambert, and A. Elouardi, "Guaranteed simultaneous localization and mapping algorithm using interval analysis," in *Proc. 13th Int. Conf. Control Autom. Robot. Vis.*, Singapore, Dec. 2014, pp. 1409–1414.
[15] B. Vincke and A. Lambert, "Experimental comparison of bounded-error state estimation and constraints propagation," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, May 2011, pp. 4724–4729.
[16] E. Seignez, M. Kieffer, A. Lambert, E. Walter, and T. Maurin, "Experimental vehicle localization by bounded-error state estimation using interval analysis," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, Edmonton, AB, Canada, Aug. 2005, pp. 1084–1089.
[17] M. Di Marco, A. Garulli, S. Lacroix, and A. Vicino, "Set membership localization and mapping for autonomous navigation," *Int. J. Robust Nonlinear Control*, vol. 11, no. 7, pp. 709–734, 2001.
[18] L. Jaulin, F. Dabe, A. Bertholom, and M. Legris, "A set approach to the simultaneous localization and map building; application to underwater robots," in *Proc. 4th Int. Conf. Inf. Control, Autom. Robot.*, Angers, France, 2007, pp. 65–69.
[19] L. Jaulin, "Localization of an underwater robot using interval constraint propagation," in *Proc. Int. Conf. Principles Pract. Constraint Program.*, 2006, pp. 244–255.
[20] F. Abdallah, A. Gning, and P. Bonnifait, "Box particle filtering for nonlinear state estimation using interval analysis," *Automatica*, vol. 44, no. 3, pp. 807–815, 2008.
[21] M. Schikora, A. Gning, L. Mihaylova, D. Cremers, and W. Koch, "Box-particle PHD filter for multi-target tracking," in *Proc. 15th Int. Conf. Inf. Fusion*, Singapore, Jul. 2012, pp. 106–113.
[22] N. Petrov, A. Gning, L. Mihaylova, and D. Angelova, "Box particle filtering for extended object tracking," in *Proc. 15th Int. Conf. Inf. Fusion*, Singapore, Jul. 2012, pp. 82–89.
[23] N. Merlinge, K. Dahia, and H. Piet-Lahanier, "A Box regularized particle filter for terrain navigation with highly non-linear measurements," *IFAC-PapersOnLine*, vol. 49, no. 17, pp. 361–366, Sep. 2016.
[24] M. Mallick, V. Krishnamurthy, and B. N. Vo, "Particle filtering combined with interval methods for tracking applications," in *Integrated Tracking, Classification, and Sensor Management: Theory and Applications*, 1st ed. Hoboken, NJ, USA: Wiley, 2012, pp. 43–74.
[25] A. De Freitas, L. Mihaylova, A. Gning, and V. Kadirkamanathan, "Autonomous crowds tracking with box particle filtering and convolution particle filtering," *Automatica*, vol. 69, pp. 380–394, Jul. 2016.
[26] F. Mourad, H. Snoussi, F. Abdallah, and C. Richard, "Guaranteed boxed localization in MANETs by interval analysis and constraints propagation techniques," in *Proc. IEEE GLOBECOM*, New Orleans, LO, USA, Nov./Dec. 2008, pp. 1–5.
[27] A. Gning and P. Bonnifait, "Constraints propagation techniques on intervals for a guaranteed localization using redundant data," *J. Autom.*, vol. 42, no. 7, pp. 1167–1175, Jul. 2006.
[28] A. Gning and P. Bonnifait, "Guaranteed dynamic localization using constraints propagation techniques on real intervals," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Apr./May 2004, pp. 1499–1504.
[29] X. S. Yang and S. Deb, *Eagle Strategy Using Lévy Walk and Firefly Algorithms for Stochastic Optimization* (Studies in Computational Intelligence), vol. 284. 2010, pp. 101–111.
[30] M. C. Tian, B. O. Yu-Ming, and Z. M. Chen, "Firefly algorithm intelligence optimized particle filter," *Acta Autom. Sin.*, vol. 42, no. 1, pp. 89–97, 2016.
[31] M. L. Gao, L.-L. Li, X.-M. Sun, L.-J. Yin, H.-T. Li, and D.-S. Luo, "Firefly algorithm (FA) based particle filter method for visual tracking," *Optik—Int. J. Light Electron Opt.*, vol. 126, no. 18, pp. 1705–1711, 2015.
[32] R. E. Moore, *Introduction to Interval Analysis*, 1st ed. Philadelphia, PA, USA: SIAM, 2009, pp. 1–234.
[33] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, "Contractors," in *Applied Interval Analysis*, 1st ed. London, U.K.: Springer, 2001, pp. 65–101.
[34] K. L. Nickel, "A globally convergent ball Newton method," *SIAM J. Numer. Anal.*, vol. 18, no. 6, pp. 988–1003, 1981.
[35] D. R. Wang, L. S. Zhang, and N. Y. Deng, "Ball algorithm for real nonlinear equations," in *Interval Algorithm for Nonlinear Equations*, 1st ed. Shanghai, China: Shanghai Science and Technology Press, 1987, pp. 115–152.
[36] R. Sen and P. Guhathakurta, "A globally convergent ball Stirling method," *Appl. Numer. Math.*, vol. 51, nos. 2–3, pp. 329–340, 2004.
[37] A. Gning, B. Ristic, L. Mihaylova, and F. Abdallah, "An introduction to box particle filtering [lecture notes]," *IEEE Signal Process. Mag.*, vol. 30, no. 4, pp. 166–171, Jul. 2013.
[38] A. Gning, L. Mihaylova, and F. Abdallah, "Mixture of uniform probability density functions for non linear state estimation using interval analysis," in *Proc. 13th Int. Conf. Inf. Fusion*, Edinburgh, U.K., Jul. 2010, pp. 1–8.
[39] A. Gning, B. Ristic, and L. Mihaylova, "Bernoulli particle/box-particle filters for detection and tracking in the presence of triple measurement uncertainty," *IEEE Trans. Signal Process.*, vol. 60, no. 5, pp. 2138–2151, May 2012.
[40] T. Csendes and D. Ratz, "Subdivision direction selection in interval methods for global optimization," *SIAM. J. Numer. Anal.*, vol. 34, no. 3, pp. 922–938, 1997.
[41] N. Petrov, L. Mihaylova, A. D. Freitas, and A. Gning, "Crowd tracking with box particle filtering," in *Proc. 17th Int. Conf. Inf. Fusion*, Salamanca, Spain, Jul. 2014, pp. 1–7.
[42] I. K. Kueviakoe, A. Lambert, and P. Tarroux, "Comparison of interval constraint propagation algorithms for vehicle localization," *J. Softw. Eng. Appl.*, vol. 5, no. 12, pp. 157–162, 2012.
[43] Y. Jie, L. Chang-Yun, and L. Zhi-Hui, "A survey of box particle filter theory," *Electron. Opt. Control*, vol. 22, no. 11, pp. 56–60, 2015.
[44] L. Jaulin, M. Kieffer, I. Braems, and E. Walter, "Guaranteed non-linear estimation using constraint propagation on sets," *Int. J. Control*, vol. 74, no. 18, pp. 1772–1782, 2001.
[45] H. Ankişhan, F. Ari, E. Ö. Tartan, and A. G. pakfiliz, "Square root central difference-based FastSLAM approach improved by differential evolution," *Turkish J. Electr. Eng. Comput. Sci,.*, vol. 24, no. 3, pp. 994–1013, 2016.
[46] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, 1st ed. Cambridge, MA, USA: MIT Press, 2005, pp. 52–57.
[47] T. Bailey. (Mar. 10, 2011). *SLAM Simulations*, [Online], Available: http://www-personal.acfr.usyd.edu.au/tbailey/software/
[48] D. Ribas, "Towards simultaneous localization and mapping for an AUV using an imaging sonar," Dept. Electron., Inform. Autom., Univ. Girona, Girona, Spain, Tech. Rep., 2005.
[49] K. L. Ho and P. Newman, "Loop closure detection in SLAM by combining visual and spatial appearance," *Robot. Auto. Syst.*, vol. 54, no. 9, pp. 740–749, 2006.
[50] E. Rublee *et al.*, "ORB: An efficient alternative to SIFT or SURF," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.
[51] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

[52] D. Scaramuzza and F. Fraundorfer, "Visual odometry: Part I—The first 30 years and fundamentals," *IEEE Robot. Autom. Mag.*, vol. 18, no. 4, pp. 80–92, Dec. 2011.
[53] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auton. Robot.*, vol. 34, no. 3, pp. 189–206, 2013.

JINGWEN LUO received the bachelor's degree in process equipment and control engineering and the master's degree in control theory and control engineering from the Kunming University of Science and Technology, Kunming, China, in 2005 and 2009, respectively. He is currently pursuing the Ph.D. degree with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China.

Since 2009, he has been a Lecturer with the School of Information Science and Technology, Yunnan Normal University.

SHIYIN QIN received the bachelor's and master's degrees in engineering science in automatic controls and industrial systems engineering from Lanzhou Jiaotong University, Lanzhou, China, in 1978 and 1984, respectively, and the Ph.D. degree in industrial control engineering and intelligent automation from Zhejiang University, Zhejiang, China, in 1990. He is currently a Professor with the School of Automation Science and Electrical Engineering, Beihang University, Beijing, China.