



A Decorrelated Distributed EKF-SLAM System for the Autonomous Navigation of Mobile Robots

Fujun Pei^{1,2} · Mingjun Zhu^{1,2} · Xiaoping Wu^{1,2}

Received: 15 January 2019 / Accepted: 23 July 2019 / Published online: 7 August 2019
© The Author(s) 2019

Abstract

This paper proposes a novel distributed EKF-SLAM system that combines the advantages of EKF-SLAM and distributed SLAM systems. The system model of this novel SLAM system has a distributed structure, and each subsystem is a special SLAM system corresponding to every effectively observed landmark by feeding the heading information from a magnetic compass is introduced into the observation equation. Aim at the correlation problem in distributed SLAM system, a decorrelated distributed EKF (DDEKF) was developed to estimate the robot pose and landmarks. DDEKF reconstructs and extends the maximum allocation covariance (MAC) method so that it can be applied to the distributed structure where the number of local filters is dynamically changed. Then, the local filter estimation results are weighted and fused in the main filter to obtain the estimation result. Finally, the experimental tests were performed in an outdoor environment, and the experiment results demonstrate that the proposed novel distributed EKF-SLAM system has a better performance than the existing SLAM system.

Keywords SLAM system · Distributed structure · Heading information · EKF · Decorrelation

1 Introduction

Simultaneous localization and mapping, a well-known computational problem abbreviated as SLAM, was proposed as a means of enabling a mobile robot to move through an unknown environment while building a map and simultaneously estimating its position [1]. Unlike traditional navigation systems that rely on prior knowledge of the environment or external reference systems (e.g. GPS), SLAM requires only the utilization of onboard sensors without any other assistance. Therefore, SLAM has attracted extensive attention in mobile robotics and has come to be seen as the primary tool for solving the autonomous navigation problem for various unmanned

vehicles, such as unmanned aerial vehicles [2, 3], underwater robots [4, 5] and space robots [6].

In general, two main types of approaches have been widely used to solve the SLAM problem. The first is smoothing approaches, in which the complete trajectory of the robot is estimated from the full set of measurements. The existing solutions for graph-based SLAM include square root smoothing and mapping (SAM) [7], incremental smoothing and mapping (iSAM) [8], nonlinear constraint network optimization [9], and hierarchical optimization for pose graphs on manifolds [10]. Due to the relatively high complexity of solving the error minimization problem, graph-based SLAM is often considered to require an offline algorithm. The second is filtering approaches, in which the SLAM problem is modeled on the basis of an online state estimation system consisting of the robot's current position and the map. The SLAM approaches that fall into this category include methods based on the extended Kalman filter (EKF), information filters and particle filters. Successful SLAM algorithms that use the EKF have been developed for various applications [11–13]. However, linearized series approximations can lead to poor representations of nonlinear functions, and the EKF approach cannot be used to solve non-Gaussian problems. A SLAM system using an information filter maintains a sparse information matrix that preserves the consistency of the Gaussian distribution [14, 15]; however, accessing the mean and covariance requires the inversion of a large matrix, which is computationally expensive. The most efficient system for

✉ Fujun Pei
pfj@bjut.edu.cn

Mingjun Zhu
mingjunzhu10@gmail.com

Xiaoping Wu
328436944@qq.com

¹ Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China

² Beijing Key Laboratory of Computational Intelligence and Intelligent Systems, Beijing 100124, China

particle-filter-based SLAM (PF-SLAM) is FastSLAM, in which the SLAM problem is decomposed into a robot localization problem and a landmark estimation problem, which are addressed with a particle filter and the EKF, respectively [16–19].

However, the aforementioned approaches all rely on a system structure based on a centralized filter. Any change in the number of feature points will result in state vector reconstruction and an additional computational burden because all of the state information is contained in one vector, and information of different quality will be unavoidably confused. To overcome these limitations of centralized filters, Dae Hee Won proposed a SLAM system based on a distributed particle filter (DPF) [20, 21]. In this distributed SLAM system, the measurements of every feature point are processed in local particle filters (the number of local filters is determined by the number of feature points). Then, the estimated results from the individual local filters are transmitted to a master filter, which calculates the final estimate for the vehicle. The authors of [22, 23] developed a distributed SLAM system with an improved DPF approach. In [23], the authors developed an improved FastSLAM system based on a distributed structure for autonomous robot navigation.

Among the aforementioned SLAM systems, the distributed SLAM system proposed in [21, 23] is the most representative. There is no doubt that this algorithm is effective and easy to implement in real time for practical applications. However, there are still some drawbacks inherent in this algorithm. In this distributed SLAM system, since a particle filter is used as the estimator for the robot pose, the consistency and long-term accuracy of the distributed SLAM system will be affected by the particle diversity and accuracy. Even when an improved particle filter is used in the distributed SLAM system, inconsistency cannot be avoided in PF-SLAM when resampling is performed. In addition, the feature points on the map and the robot pose are related in every cycle of estimation because of the common prior estimates and common process noise shared by the local filters, and the correlations will become stronger as the number of observations increases.

Since EKF estimation directly provides recursive solutions to localization problems and suitable to the robot and landmark positions, the EKF approach remains the method of choice for the great majority of applications, and EKF-SLAM has been proven to offer the best convergence and consistency [21, 22]. Motivated by this previous experience, this paper proposes a decorrelated distributed EKF-SLAM system. In this distributed SLAM system, to guarantee the performance of the system model, a magnetic compass sensor is introduced into the system, and the heading from this magnetic compass is fed into each subsystem as a measurement vector. Because this heading information is considered, the proposed SLAM system not only can revise the robot's orientation during estimation but also can ensure the consistency and accuracy of each subsystem. An observability

analysis of the new distributed SLAM system model is presented to provide a theoretical basis for the design of the estimator. To overcome the suboptimality problem caused by the measurement vector correlations, the maximum allocated covariance (MAC) fusion method [24] is implemented and extended to an N-dimensional matrix to improve the computational efficiency of the system and allow it to adapt to a situation with an uncertain number of local filters. The proposed distributed SLAM system is thus more suitable for practical applications because of its system structure and implementation technique.

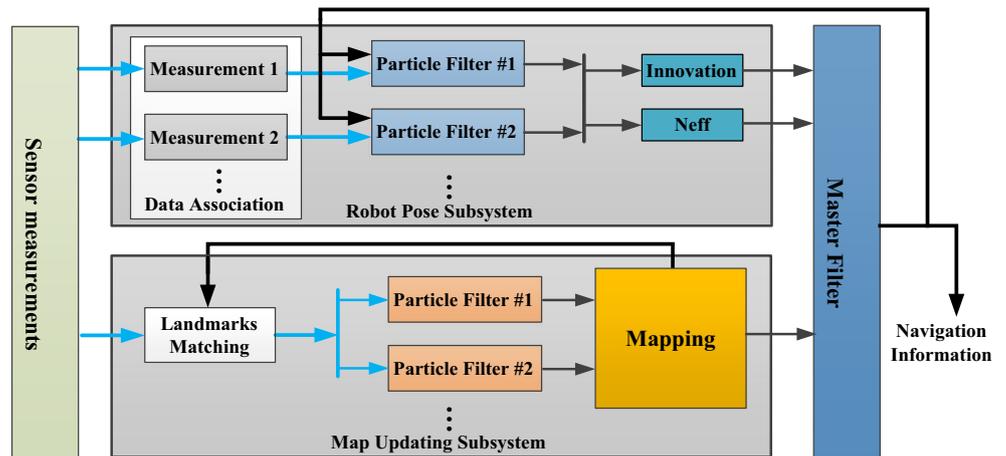
The contents of this paper are as follows. In Section 2, the basic configuration and model of a typical distributed particle SLAM system are reviewed. Section 3 introduces the novel distributed SLAM system model with heading assistance based on magnetic compass measurements. The decorrelated distributed EKF algorithm for the proposed SLAM system is designed and analyzed in Section 4. Section 5 presents the experimental results, and conclusions are drawn in Section 6.

2 Distributed System Architecture for Slam

The typical centralized filter structure is considered to yield globally optimal results because the master filter in this structure considers all state information and observed data [11]. The EKF and the unscented Kalman filter (UKF) are the most widely used filtering methods in SLAM, but they always have a centralized structure. Changes in the state and observation vectors reduce the system stability since each change in the number of feature points will cause the entire state vector to be reconstructed, and a large amount of computation required also reduces the real-time capabilities of such a centralized system because of the exponential increase in the computational burden with changes in the state vector. An effective way to solve the above problems is to adopt a distributed parallel synchronous structure that has a master filter to fuse local trajectories within a certain period of time; such a distributed structure can reduce the effect of error messages on the final estimation result, improve the fault tolerance, and allow the system to respond effectively to changes in the number of effective landmarks. The most successful distributed filtering method in SLAM is particle filtering [22]. The configuration of a typical distributed particle SLAM system is shown in Fig. 1.

The DPF structure is an attractive and effective means of realizing a SLAM system. Generally, the estimation model of a distributed particle SLAM system is composed of a motion model and an observation model. The estimation model addresses the problem of a robot with known kinematics, starting at an unknown position and moving

Fig. 1 DPF configuration



through an unknown environment populated with landmarks of a certain type [23]:

$$\begin{cases} \dot{X}_1 = f(X_{r1}, X_{L1}) + \gamma \\ Z_1(k) = h(X_{r1}(k), X_{L1}) + \eta \\ \dot{X}_2 = f(X_{r2}, X_{L2}) + \gamma \\ Z_2(k) = h(X_{r2}(k), X_{L2}) + \eta \\ \vdots \\ \dot{X}_j = f(X_{rj}(k), X_{Lj}) + \gamma \\ Z_j(k) = h(X_{rj}(k), X_{Lj}) + \eta \end{cases} \quad (1)$$

where $X_r(k)$ is the robot motion model, which can be described by a probabilistic Markov chain; $X_L(k)$ is the state of the landmarks observed by the robot; and $Z(k) = [z_1^T(k), z_2^T(k), \dots, z_j^T(k)]$ is the observation equation. $\omega(k)$ and $v(k)$ are considered to be zero-mean Gaussian white measurement noise.

$$X(k+1) = \begin{bmatrix} x_r(k+1) \\ y_r(k+1) \\ \varphi_r(k+1) \\ x_{Lj}(k+1) \\ y_{Lj}(k+1) \end{bmatrix} = f(X(k)) + \gamma = \begin{bmatrix} x_r(k) + \Delta x \\ y_r(k) + \Delta y \\ \varphi_r(k) + \Delta \varphi \\ x_{Lj}(k) \\ y_{Lj}(k) \end{bmatrix} + \gamma \quad (2)$$

where $x_r(k)$, $y_r(k)$, and $\varphi_r(k)$ are the robot states defined by the external odometer sensor at time k and γ is the process noise. It is assumed that γ consists of zero-mean white sequences that are uncorrelated with each other, and the covariance is Q . x_{Lj} and y_{Lj} represent the state of the j th landmark. Δx , Δy , and $\Delta \varphi$ are the variations of the robot states.

$Z(k)$ is the observation equation relating the robot states to the landmarks, as shown in (3):

$$Z(k) = h(X_r(k), x_{Lj}, y_{Lj}) = \begin{bmatrix} z_r^j \\ z_\beta^j \end{bmatrix} + \eta = \begin{bmatrix} \sqrt{(x_{Lj}-x_r)^2 + (y_{Lj}-y_r)^2} \\ \arctan\left(\frac{y_{Lj}-y_r}{x_{Lj}-x_r}\right) - \varphi_r + \frac{\pi}{2} \end{bmatrix} + \eta \quad (3)$$

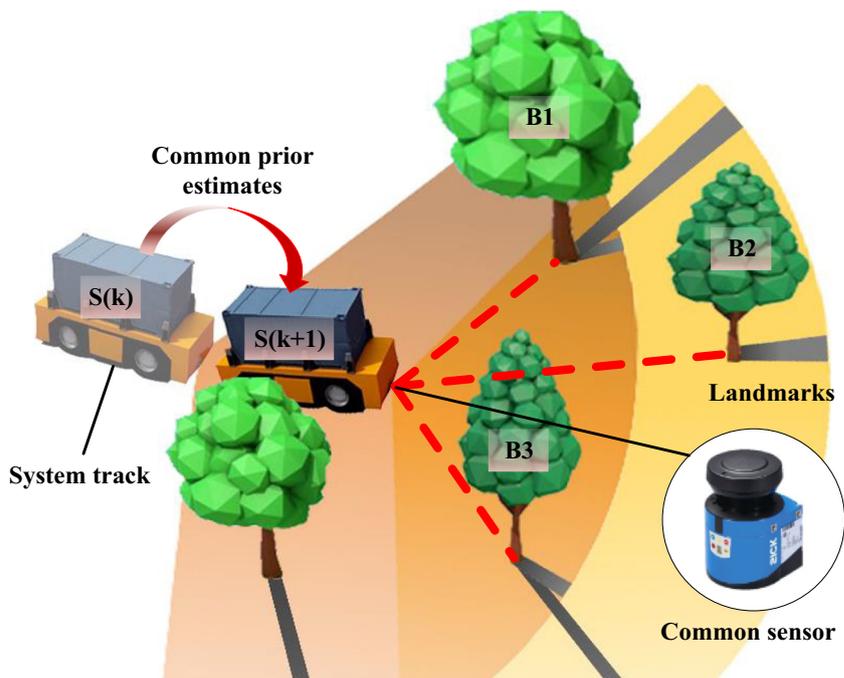
where Z is the observation vector, η is zero-mean Gaussian white measurement noise, Z_r is the distance from the beacon to the laser sensor, and Z_β is the bearing of the laser sensor measured with respect to the robot coordinate frame.

This distributed SLAM system model is based on every effectively observed landmark and establishes a set of local filters that are independent of each other and work in parallel. Each estimated result for the states is transmitted to the fusion filter to obtain the final results. However, this type of distributed structure gives rise to a correlation problem for the following reasons:

- (1) Collective process noise: Every subsystem can be regarded as a special SLAM system with an observation equation that includes information on only one landmark from the sensor. Thus, correlations between the measurement vectors in different subsystems are inevitably induced because all of the measurements originate from the same sensor.
- (2) Common prior estimates: Common prior estimates will lead to correlation problems when one track is merged with another track to generate a system track because each track contains common historical information. As shown in Fig. 2, a plurality of parallel filters B1 - B3 and the system track estimate $S(k+1)$ contain the common local track estimate $S(k)$ propagated from an earlier time.

The conventional track fusion algorithm does not consider the local association problem in the system when all tracks are estimated from only one given target. To overcome this problem, Y. Bar-Shalom proposed a distributed track fusion formula called track-to-track fusion (TTF) in [25]. The TTF formula considers the correlations caused by common process noise. The cross-covariance and the fusion trajectory in a distributed system can be calculated

Fig. 2 Configuration of sensor-to-system track integration



from two state estimates, X_a and X_b , and their covariance matrixes, P_a and P_b :

$$\tilde{X}_{ij}(k|k) = \hat{X}_i(k|k) + \zeta_{ij}(k) [\hat{X}_i(k|k) - \hat{X}_j(k|k)] \quad i \in \{1, 2, \dots, N-1\}, j \in [i, i+1, \dots, N] \quad (4)$$

where

$$\zeta_{ij}(k) = [P_i(k|k) - P_{ij}(k|k)] [P_i(k|k) + P_j(k|k) - P_{ij}(k|k) - P_{ij}^T(k|k)] \quad (5)$$

where $\tilde{X}_{ij}(k|k)$ is the fusion result, $\zeta_{ij}(k)$ is the filter gain of a local processor, and $\hat{X}_i(k|k)$ and $\hat{X}_j(k|k)$ are the local filters' estimates from a single sensor.

$$\tilde{P}_{ij}(k|k) = P_{ij}(k|k) - \xi_{ij}(k) (P_{ij}(k|k) - P_{ij}(k|k)^T) \quad (6)$$

The advantage of this method is that the correlations between the local estimation errors are taken into consideration. The disadvantages are that a large amount of information is needed to calculate the cross-covariance between the estimations and the communication requirements are strict.

3 Novel Distributed Slam System Model

It is clear that a distributed structure is an effective way to realize a SLAM system compared to a centralized structure; however, the existing distributed SLAM

systems are all based on the DPF approach, which has the inherent drawbacks of particle impoverishment and high computational complexity. Synthesizing the advantages of EKF-SLAM and distributed SLAM systems, this paper proposes a novel distributed SLAM system that has a distributed structure and uses a distributed EKF (DEKF) as the estimator. Similar to the conventional distributed system model, the state vector of every subsystem is composed of the robot state and one landmark position. Notably, it has been proven that inconsistency in EKF-SLAM is directly related to heading uncertainty [1]. If the true heading variance remains small, the inconsistency grows slowly and is manageable. To overcome the inconsistency problem, heading information from a magnetic compass is fed into the observation model for each subsystem to ensure global observability and consistency.

3.1 Novel Distributed SLAM System Model

The novel distributed SLAM state model is constructed in the same way as the conventional model:

$$X(t+1) = \begin{bmatrix} x_r(t+1) \\ y_r(t+1) \\ \varphi_r(t+1) \\ x_{lj}(t+1) \\ y_{lj}(t+1) \end{bmatrix} = f(X(t)) + \gamma = \begin{bmatrix} x_r(t) + v_r \cos(\varphi_r(t)) \Delta T \\ y_r(t) + v_r \sin(\varphi_r(t)) \Delta T \\ \varphi_r(t) + \omega_c \Delta T \\ x_{lj}(t) \\ y_{lj}(t) \end{bmatrix} + \gamma \quad (7)$$

where v_c is the linear velocity and ω_c is the angular velocity; these velocities are calculated from encoder information. The

observation model, which consists of the laser and heading information, is

$$X(k) = h\left(X_r(k), x_{Lj}, y_{Lj}, \varphi_r\right) = \begin{bmatrix} z_r^j \\ z_\beta^j \\ z_\varphi \end{bmatrix} + \eta$$

$$= \begin{bmatrix} \sqrt{(x_{Lj}-x_r)^2 + (y_{Lj}-y_r)^2} \\ \arctan\left(\frac{(y_{Lj}-y_r)}{(x_{Lj}-x_r)}\right) - \varphi_r + \frac{\pi}{2} \\ \varphi_r \end{bmatrix} + \eta \tag{8}$$

where z_φ is the heading of the robot as measured by the magnetic compass.

Then, based on every effective observed landmark, we can establish the distributed structure as follows:

$$\begin{cases} \dot{X}_1 = f(X_r(k), x_{L1}, y_{L1}) + \gamma \\ X_1(k) = h(X_r(k), x_{L1}, y_{L1}) + \eta \\ \dot{X}_2 = f(X_r(k), x_{L2}, y_{L2}) + \gamma \\ Z_2(k) = h(X_r(k), x_{L2}, y_{L2}) + \eta \\ \vdots \\ \dot{X}_j = f(X_r(k), x_{Lj}, y_{Lj}) + \gamma \\ Z_j(k) = h(X_r(k), x_{Lj}, y_{Lj}) + \eta \end{cases} \tag{9}$$

This novel distributed SLAM system model uses every effectively observed landmark to establish a set of local filters that work in parallel and independently of each other. The heading information is used as observation information to revise the robot’s steering and to ensure the global observability of the model.

3.2 Observability Analysis of the SLAM System

Observability is an important issue that plays a major role in the filtering and reconstruction of states from inputs and outputs [26]. The observability condition is an indicator of whether the system contains all the necessary information to perform the estimation with a bounded error. In particular, for the case of SLAM, observability implies a bounded error in the localization of both the robot and the landmarks.

Formally, observability simply reflects whether the initial state $x(0)$ can be uniquely deduced from the history of observations. This requires that the observability Gramian

$$O(0, t) \triangleq \int_0^t e^{F^T \tau} H H^T e^{F \tau} d\tau \tag{10}$$

be nonsingular; if this condition is met, then all states are observable.

For a time-invariant system, which can be approximated by a piecewise constant system with little loss of accuracy and no loss of the characteristic behavior of the system [27], we can

ensure that the entire system can be observed by confirming the observability of every distributed local system. In addition, checking the observability condition can be reduced to checking whether the matrix (11) is of rank N :

$$Q \triangleq \begin{bmatrix} H \\ HF \\ \dots \\ HF^{n-1} \end{bmatrix} \tag{11}$$

where F and H are the motion and observation Jacobian matrices, respectively, and N is the size of the state.

Consider the SLAM system defined by the observation model in (3) and the state model $X = [x_r, y_r, \varphi_r, (x_{Lj}, y_{Lj})]^T$, which takes one landmark as input. When the nonlinear observation model is linearized using the Taylor series expansion, the observation model Jacobian H is

$$H = \frac{\partial h}{\partial X} = \begin{bmatrix} \frac{\partial h_r}{\partial X} \\ \frac{\partial h_\beta}{\partial X} \\ \frac{\partial h_\varphi}{\partial X} \end{bmatrix} = \begin{bmatrix} \frac{\partial z_r}{\partial (x_r, y_r, \varphi_r, (x_{Lj}, y_{Lj}))} \\ \frac{\partial z_\beta}{\partial (x_r, y_r, \varphi_r, (x_{Lj}, y_{Lj}))} \\ \frac{\partial z_\varphi}{\partial (x_r, y_r, \varphi_r, (x_{Lj}, y_{Lj}))} \end{bmatrix}$$

$$= \begin{bmatrix} -\frac{\Delta x}{\Delta} & -\frac{\Delta y}{\Delta} & 0 & \frac{\Delta x}{\Delta} & \frac{\Delta y}{\Delta} \\ \frac{\Delta y}{\Delta^2} & -\frac{\Delta x}{\Delta^2} & -1 & -\frac{\Delta y}{\Delta^2} & \frac{\Delta x}{\Delta^2} \end{bmatrix} \tag{12}$$

where $\Delta x = [x_{Lj} - x_r]$, $\Delta y = [y_{Lj} - y_r]$ and $\Delta = \sqrt{\Delta x^2 + \Delta y^2}$. The state Jacobian matrix F is

$$F = \begin{bmatrix} 0 & 0 & -v_c \sin(\phi) & 0 & 0 \\ 0 & 0 & v_c \cos(\phi) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

Thus, the observability matrix from (11) is

$$Q = \begin{bmatrix} -\frac{\Delta x}{\Delta} & -\frac{\Delta y}{\Delta} & 0 & \frac{\Delta x}{\Delta} & \frac{\Delta y}{\Delta} \\ \frac{\Delta y}{\Delta^2} & -\frac{\Delta x}{\Delta^2} & -1 & -\frac{\Delta y}{\Delta^2} & \frac{\Delta x}{\Delta^2} \\ 0 & 0 & \frac{v \sin \theta \Delta x - v \cos \theta \Delta y}{\Delta} & \frac{\Delta x}{\Delta} & \frac{\Delta y}{\Delta} \\ 0 & 0 & \frac{-v \sin \theta \Delta y - v \cos \theta \Delta x}{\Delta^2} & \frac{\Delta y}{\Delta^2} & \frac{\Delta x}{\Delta^2} \end{bmatrix} \tag{14}$$

Here, $Rank(Q) = 4 < 5$. It can be shown that the observability matrix is not of full rank when observing the range and bearing of one landmark, meaning that the system is unobservable.

Therefore, we use an additional compass sensor to measure position information in the SLAM system: given the state

model in (7) and the observation model in (8), the observation model Jacobian H is

$$H = \frac{\partial h}{\partial X} = \begin{bmatrix} \frac{\partial h_r}{\partial X} \\ \frac{\partial h_\beta}{\partial X} \\ \frac{\partial h_\phi}{\partial X} \end{bmatrix} = \begin{bmatrix} \frac{\partial z_r}{\partial(x_r, y_r, \phi_r, (x_{Lj}, y_{Lj}))} \\ \frac{\partial z_\beta}{\partial(x_r, y_r, \phi_r, (x_{Lj}, y_{Lj}))} \\ \frac{\partial z_\phi}{\partial(x_r, y_r, \phi_r, (x_{Lj}, y_{Lj}))} \end{bmatrix} \\ = \begin{bmatrix} 0 & \frac{\Delta x}{\Delta} & \frac{\Delta y}{\Delta} \\ \frac{\Delta y}{\Delta^2} & -\frac{\Delta x}{\Delta^2} & -1 \\ \frac{\Delta y}{\Delta^2} & -\frac{\Delta x}{\Delta^2} & 1 \end{bmatrix} \quad (15)$$

and the observability matrix from (11) is

$$Q = \begin{bmatrix} -\frac{\Delta x}{\Delta} & -\frac{\Delta y}{\Delta} & 0 & \frac{\Delta x}{\Delta} & \frac{\Delta y}{\Delta} \\ \frac{\Delta y}{\Delta^2} & -\frac{\Delta x}{\Delta^2} & -1 & \frac{\Delta y}{\Delta^2} & \frac{\Delta x}{\Delta^2} \\ 0 & 0 & \frac{1}{\Delta} & \frac{\Delta y}{\Delta} & \frac{\Delta x}{\Delta} \\ 0 & 0 & \frac{v \sin \theta \Delta x - v \cos \theta \Delta y}{\Delta} & \frac{\Delta y}{\Delta} & \frac{\Delta x}{\Delta} \\ 0 & 0 & \frac{-v \sin \theta \Delta y - v \cos \theta \Delta x}{\Delta^2} & \frac{\Delta y}{\Delta^2} & \frac{\Delta x}{\Delta^2} \end{bmatrix} \quad (16)$$

Here, $\text{Rank}(Q) = 5$. It can be shown that the observability matrix becomes of full rank when observing the range and bearing of one landmark and the heading of the robot, demonstrating that the novel distributed SLAM system is observable.

4 Decorrelation Algorithm for Distributed Slam

Generally, correlations can be taken into account during track fusion, and one method of doing so based on a convex fusion algorithm is named TTF [28]. In accordance with the maximum likelihood (ML) and linear minimum variance (LMV) estimators, the TTF algorithm can be simply extended to N-dimensional fusion. However, the correlations cannot be effectively calculated in the case in which the number of local filters changes dynamically because the correlation calculation requires continuous recursion. The method of covariance intersection (CI) can be used to realize approximate correlation calculations even if the correlations are unknown [29]. The correlation at each moment is calculated from the parameters of the local sensors, and this calculation does not require any recursion, thus effectively solving the problem of dynamic changes in the number of local filters. According to CI theory, the results of the MAC algorithm proposed by Zarei-Jalalabadi are less conservative than the fusion results

achieved in the well-known CI case [24], but the applications of the MAC algorithm are still limited because of the absence of weights in the master fusion filter.

Since the recursive computation of the local filters' covariance cannot be implemented in a dynamic distributed system, it is not feasible to use the cross-covariance algorithm to combine two tracks. CI fusion is a popular method used in the TTF algorithm because it does not require knowledge of the real cross-covariance between two tracks. However, because the CI method is too conservative, its fusion performance is insufficient compared with that of the MAC algorithm [24]. Thus, the MAC fusion algorithm is adopted here to combine two tracks with unknown correlations; that is, this method can be used to combine two tracks using only the information from those two tracks, including the local filter states and covariance matrices. Finally, weighted fusion is performed at the fusion center. The entire framework of the decorrelated distributed EKF (DDEKF)-SLAM system is shown in Fig. 3.

For the distributed Kalman filter algorithm, the local filters' predicted states $\{X_{k/k-1}^i; i = 1, \dots, N\}$ and predicted covariances $\{P_{k/k-1}^i; i = 1, \dots, N\}$ can be obtained as follows:

$$X_{k/k-1}^i = f(X_{k-1/k-1}^i, u_{k-1}) \quad (17)$$

$$P_{k/k-1}^i = \Phi_{k,k-1} P_{k-1/k-1}^i \Phi_{k,k-1}^T + Q_{k-1} \quad (18)$$

Then, the general measurement update equation is as follows:

$$\hat{X}_{k/k}^i = \hat{X}_{k/k-1}^i + K_k^i (Z_k - H_k \hat{X}_{k/k-1}^i) = (I - K_k H_k) \hat{X}_{k/k-1}^i + K_k Z_k \quad (19)$$

For tracks $(\hat{X}_r, \hat{P}_r), r \in \{i, j\}$, with an unknown correlation, the MAC-fused track is

$$P_k^m = F(S_k) \\ = P_{k/k}^i - (P_{k/k}^i - S_k) (P_{k/k}^i + P_{k/k}^j - S_k - S_k^T)^{-1} (P_{k/k}^i - S_k^T) \quad (20)$$

with $S_k = \sqrt{P_{k/k}^i}^T C_k \sqrt{P_{k/k}^j}$, where

$$C_k = \arg \max \det(F(S_k)) \quad (21)$$

For the fused track,

$$\hat{X}_{MAC} = W_i \hat{X}_i + W_j \hat{X}_j \quad (22)$$

$$W_i = (P_j - S^T) (P_i + P_j - S - S^T)^{-1} \quad (23)$$

$$W_j = (P_i - S) (P_i + P_j - S - S^T)^{-1} \quad (24)$$

where $W_i + W_j \leq 1$.

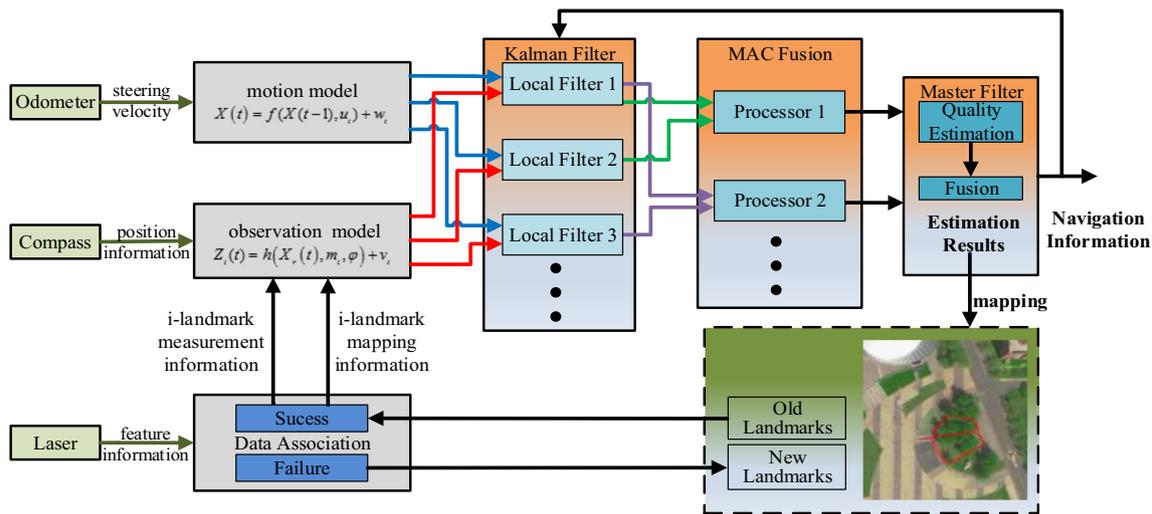


Fig. 3 Block diagram of the distributed SLAM system

MAC fusion is also applicable to scenarios where the number of parallel filters is dynamically changing because the fusion algorithm is based on the CI fusion method, which does not require historical information. Besides, the MAC algorithm is less conservative than the CI algorithm by reconstructing weight functions W_i and W_j ; on the other hand, the MAC algorithm is guaranteed to be more conservative than track fusion with a known cross-covariance between the tracks. To enable its application in the distributed SLAM structure, especially in the case of multiple landmarks, we extend the MAC algorithm to N-dimensional fusion and add weights to the fusion results for each subsystem to obtain the optimal estimate. According to the fusion equation of the ML estimator, the fusion result can be obtained as follows:

$$\hat{X}^{ML} = (UP_F^{-1}U^T)^{-1}UP_F^{-1}\hat{X}_F \tag{25}$$

where $U = \text{def} [I, I, \dots, I]^T$.

According to [30], we can transform the covariance matrix into a diagonal matrix; thus, the MAC expressions can be rewritten as

$$\hat{X}^m = (\hat{X}_{i1i1}^{m1} \hat{X}_{i1i2}^{m2} \dots \hat{X}_{iNiN}^{mN})^T \tag{26}$$

$$P^N = \begin{bmatrix} P_{i1i1}^{m1} & 0 & \dots & 0 \\ 0 & P_{i1i2}^{m2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & P_{iNiN}^{mN} \end{bmatrix} \tag{27}$$

Finally, we add weight information to the fusion center. The weights are obtained as follows: $W_{sisj} = \left(\sum_{n=1}^N P_{sisj}^{n-1} \right)^{-1} P_{sisj}^{N-1}$. Then, the fused estimate is given by

$$\hat{X}^{NMAC} = \sum_{n=1}^N W_{sisj} \hat{X}_{sisj}^{iN}, j \in \{1, 2, \dots, N\}, \quad i \neq j \quad \text{and} \quad i < j \tag{28}$$

5 Experimental Results and Analysis

In order to evaluate the performance of the proposed distributed structure DDEKF-SLAM system and consider the problem of correlation between local filters to the estimation accuracy, a robot runs in the square to collect the actual data, and the DDEKF algorithm is used to process it. Then, the estimated results are compared with those of the traditional EKF-SLAM system.

5.1 Experiment Design

The DDEKF proposed in this paper is suitable for data fusion in a variety of distributed structures. In order to verify the effectiveness of DDEKF in distributed architecture to improve the estimation accuracy, an outdoor experiment based on tree as road punctuation is designed. In this experiment, the way to set the tree as the road punctuation is to verify the algorithm conveniently, and the local filter can be established by identifying different landmarks according to the actual scene. The robot platform used in the experiment uses 2D laser, odometer and magnetic compass as sensors, as shown in Fig. 4. The

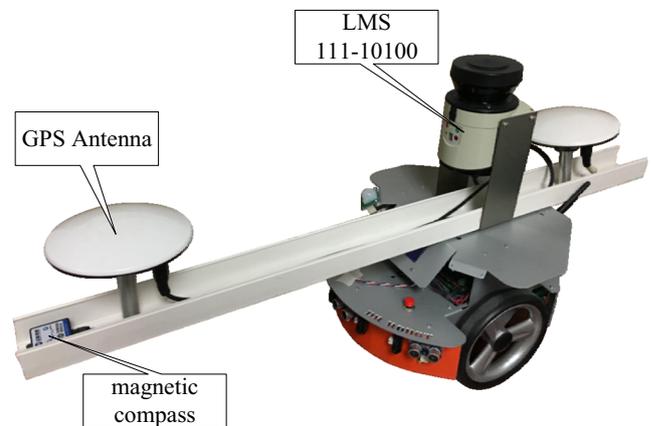


Fig. 4 The robot platform



Fig. 5 GPS measurements in GPGGA mode

laser used was the LMS111–10100 model produced by SICK, the maximum measuring distance of this sensor is 20 m, and the scanning time is 0.1 s for a 180-degree measurement area. The compass uses magnetic sensors and a two-axis tilt sensor to collect heading information, and its accuracy is 0.3° . In the experiments, trees were used as artificial landmarks, and the SLAM algorithm used dead reckoning and relative observations to estimate the position of the robot and build a navigation map, as shown in Fig. 6. The accuracy of the map reflects the accuracy of the local map origin and the quality of the kinematics model and relative observations.

The real trajectory of the robot was derived from GPS data and is displayed as the red trajectory in Fig. 5. The robot's trajectory started at the left and proceeded in the direction of the arrow; the shape of the first lap was similar to a figure-8 pattern, and the second lap lay along the outer ring of the map.

Fig. 6 The laser-based landmark extraction algorithm

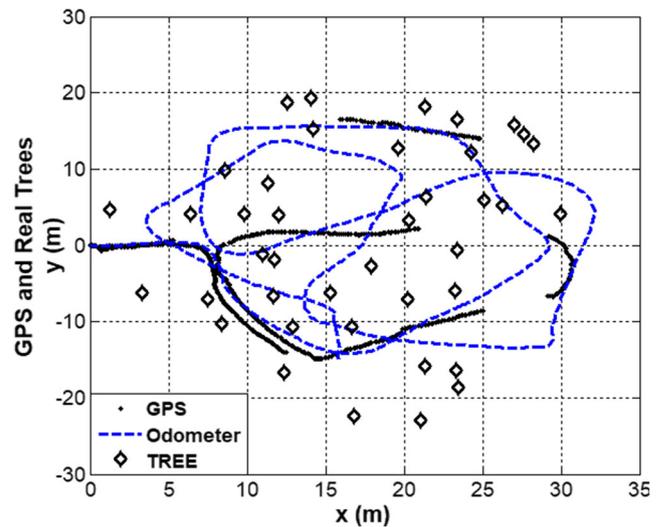
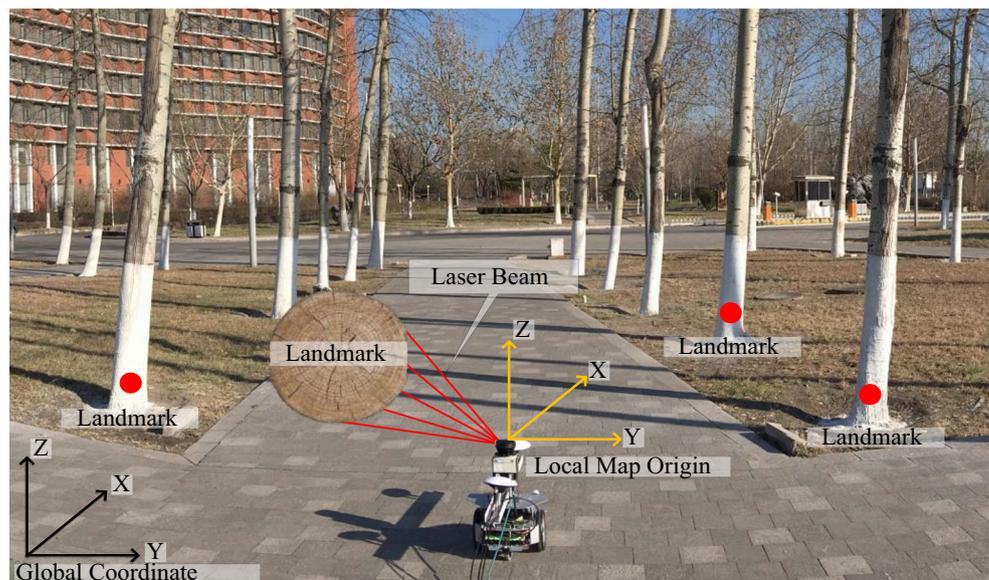


Fig. 7 The GPS, odometer's data and real trees

The laser-based landmark extraction algorithm is similar to the algorithm used in [26], in which the center of a tree trunk is tracked by clustering a number of laser observations. The data collected by the laser sensor from within its projection range were sufficient to determine the thickness of the cylindrical trunk relative to the robot's angle and distance, as shown in Fig. 6. Then, the relative coordinates of the central position of the cylinder could be calculated in accordance with the collected data and the sensor resolution. Feature matching was performed using the nearest neighbor (NN) algorithm, which obtains a positive subset of landmarks in the current map corresponding to a subset of landmarks in a previous map.

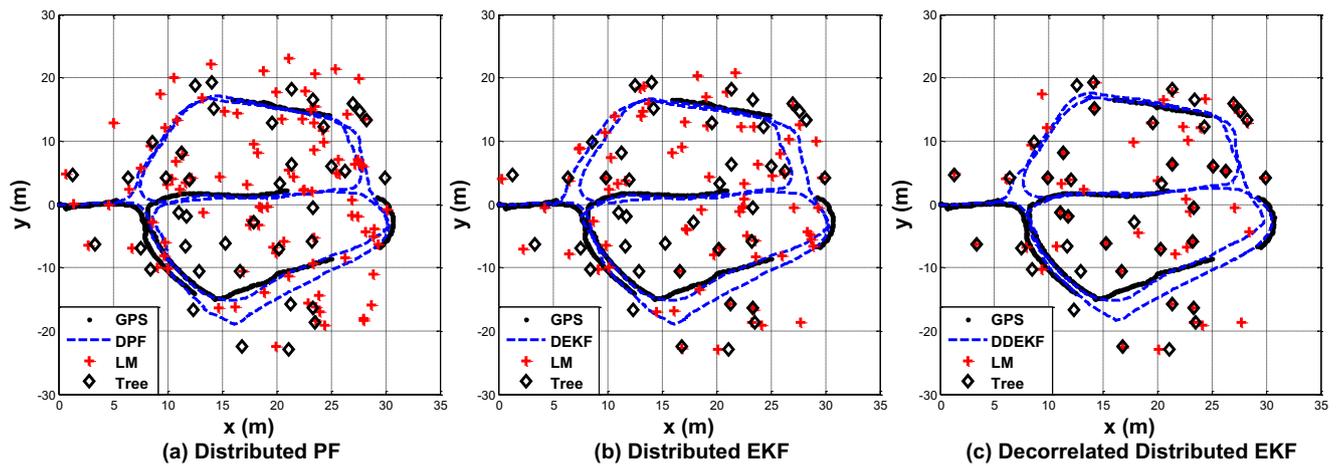


Fig. 8 The comparison of the estimation results

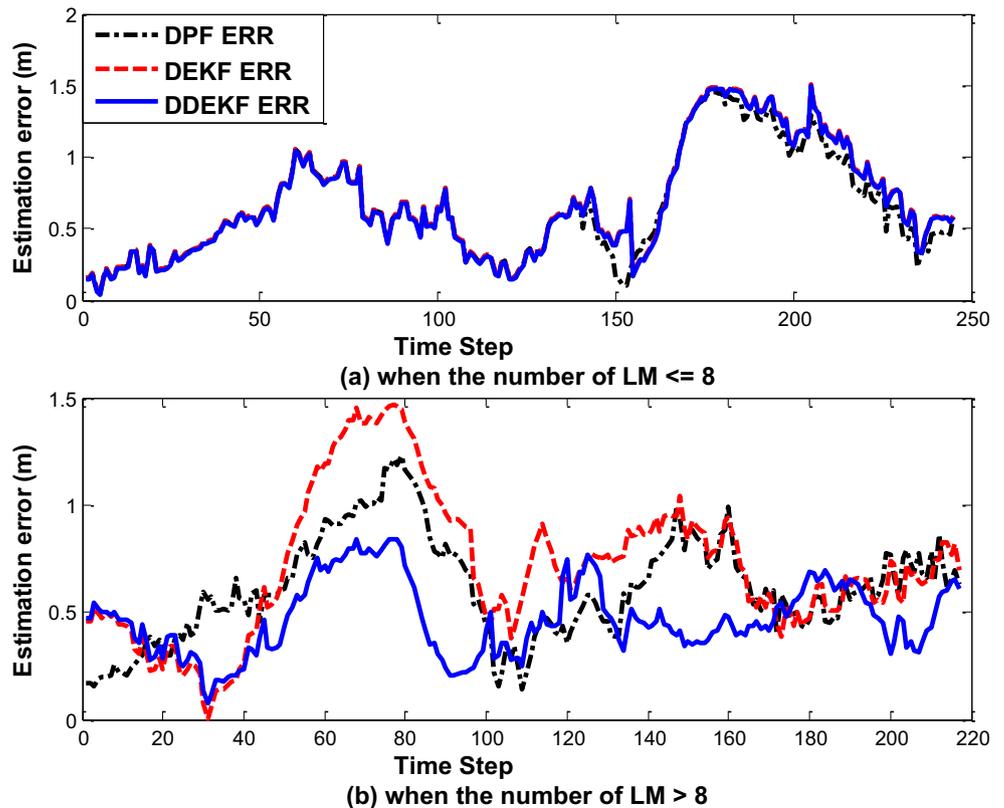
5.2 Performance of algorithms in the real-world environment

The track information in Fig. 7 is provided by the odometer, and the GPS information is added to the picture to facilitate the real track. Intensive trees in the park cause poor GPS performance, so only the robot trajectories recorded when the number of satellites is greater than 12. At the meanwhile, the real tree positions are depicted in Fig. 7, and the tree positions are obtained from the field measurement information combined with the picture information.

The estimated results for the robot’s trajectory and the navigation landmarks as obtained with the DDEKF algorithm are shown in Fig. 8. To test the performance of the DDEKF-SLAM system, the experiment was also performed with DPF-SLAM and DEKF-SLAM system for comparison, and these methods have been debugged to the best state in this group of experiments..

Figure 8 shows the pose estimation performance of the three distributed filtering algorithms in the same scene. Among them, the blue dotted line indicates the estimation result of each algorithm, and the red plus sign indicates the estimated landmark

Fig. 9 Comparison of estimation errors of different algorithms



point information, the black squares are real trees. The black dots represent the GPS measurement results, which are used as reference information for these experiments.

It is obvious that the DDEKF estimation results more closely approximate to the GPS results than the DPF estimation and DEKF estimation, demonstrating that the DDEKF algorithm achieves better accuracy. Compared to the other two algorithms, the proposed algorithm can estimate the robot's trajectory and the positions of the navigation landmarks very well and shows little error on the estimated landmark positions due to the sensors' inherent noise.

5.3 RMSE of algorithms in the case of the different number of landmarks

Figure 9 is a set of comparative tests in which the number of landmark points is less than or equal to 8 and greater than 8. This set of experiments is mainly used to verify the optimization of the decorrelated distributed system, especially when the number of parallel filters is large.

In Fig. 9 (a), the error of this localization algorithm comes from the initial stage of the map. The estimation accuracy of the three algorithms differs little when the number of parallel filters is relatively less, and the floating phenomenon of estimation error at this time mainly comes from the sensor error. The results show that the correlation problem in the distributed structure is not obvious when the initial stage of the localization algorithm and the number of landmarks are few.

In Fig. 9 (b) where the number of local filters is relatively large (the number of local filters in this experiment is about 8), the decorrelated distributed structure improves the estimation accuracy of the distributed system, since the correlation is considered in the filtering process. The error of this localization algorithm comes from locating for a period of time, so the cumulative correlation problem is more obvious.

5.4 RMSE of algorithms and Computational Cost

Table 1 summarizes the mean values and variances of the estimation results from three algorithms. Besides, the computational cost of the three algorithms is recorded for comparison. These algorithms run on Intel(R) Core (TM) i5-4590 CPU @ 3.30GHz PC.

Table 1 Comparison of the estimation errors and CPU running time

Algorithm	Estimated Error		CPU Running Time (s)
	Mean (m)	RMSE (m)	
DPF	0.647758	0.38263	14.75
DEKF	0.683575	0.39626	3.05
DDEKF	0.626077	0.35765	4.15

The estimation accuracy of DEKF is slightly lower than DPF, but the computational complexity is low and the computation time is only 20% of DPF. However, the DDEKF system obtains the final estimation result by weighting the fusion result of the MAC. As can be seen from Tab. 1, DDEKF reduces the RMSE and increases the real-time performance of the system, what's more, the calculation time of the DDEKF system is only 28.13% of the DPF.

6 Conclusion

This paper has proposed a decorrelated distributed SLAM system based on the EKF, which is an effective real-time solution for the automation navigation of mobile robots. The EKF is the best and easiest filter to implement in real time for practical applications of SLAM systems, and the distributed structure can transform the problem of state vector expansion with changing landmarks into dynamic changes in subsystems using a parallel synchronous framework. By integrating the landmarks of these two types of SLAM systems, the proposed DDEKF-SLAM system achieves significant advantages compared to existing SLAM systems. In each local filter, the state vector is composed of the state of the robot and the information on one landmark, and the observation model is composed of laser and heading information, which is used to improve the observability and convergence of the subsystem. To address the correlation problem caused by the common process noise and common prior estimates, the MAC method is implemented to use all local filters as information sources; thus, the local estimates are fused by means of a master filter to achieve global optimality. The experimental results show that the proposed distributed SLAM system has a better estimation accuracy than existing SLAM systems do and consistently achieves more accurate estimation for a continuous trajectory. A disadvantage of the proposed SLAM system is that the stability of the distributed filter with dynamic changes in the subsystems has not been analyzed. For a SLAM system, dynamic changes in subsystems should be considered because this situation commonly occurs. Investigations of an advanced distributed filter and its stability analysis are underway and hopefully will be reported in the near future.

Acknowledgments This work was supported by the National Science Foundation of Beijing Municipality under Grant 4162011, the Program for the Education Scientific Research of Beijing KM201610005009, and the 2017 BJUT United Grand Scientific Research Program on Intelligent Manufacturing 040000546317552.

Compliance with ethical standards

Conflict of interests The authors declare that there is no conflict of interests regarding the publication of this paper.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Durrant, H., Bailey, T.: Simultaneous localization and mapping: part I. *IEEE Robot. Autom. Mag.* **13**(2), 99–110 (2006)
- Caballero, F., Merino, L., Ferruz, J., Ollero, A.: Vision-Based Odometry and SLAM for Medium and High Altitude Flying UAVs. *J Intell Robot Syst.* **54**(3), 137–161 (2009)
- Wang, C.L., Wang, T.M., Liang, J.H., Zhang, Y.C., Zhou, Y.: Bearing-only Visual SLAM for Small Unmanned Aerial Vehicles in GPS-denied Environments. *Int. J. Autom. Comput.* **10**(5), 387–396 (2013)
- Mallios, A., Ridao, P., Ribas, D., Hernández, E.: Scan matching SLAM in underwater environments. *Auton. Robot.* **36**(3), 181–198 (2014)
- Lee, C.S., Nagappa, S., Palomeras, N., Clark, D.E., Salvi, J.: SLAM with SC-PHD Filters: An Underwater Vehicle Application. *IEEE Robot. Autom. Mag.* **21**(2), 38–45 (2014)
- Tong, C.H., Barfoot, T.D.: Three-dimensional SLAM for mapping planetary worksite. *J Field Rob.* **29**(3), 381–412 (2012)
- Frank, D., Michael, K.: Square root SAM: Simultaneous location and mapping via square root information smoothing. *Int. J. Robot. Res.* **25**(12), 1181–1203 (2006)
- Kaess, M., Ranganathan, A., Dellaert, F.: iSAM: Incremental smoothing and mapping. *IEEE Trans. Robot.* **24**(6), 1365–1378 (2008)
- Hochbaum, D.S.: Complexity and algorithms for convex network optimization and other nonlinear problems[J]. *4OR.* **3**(3), 171–216 (2005)
- Grisetti, G., Kümmerle, R., Stachniss, C., Frese, U., Hertzberg, C.: Hierarchical optimization on manifolds for online 2D and 3D mapping, in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 273–278 (2010)
- Dissanayake, M.W.M.G., Newman, P., Clark, S., Durrant-Whyte, H.F., Csorba, M.: A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Trans. Robot. Autom.* **17**(3), 229–241 (2001)
- Yadkuri, F.F., Khosrowjerdi, M.J.: Methods for Improving the Linearization Problem of Extended Kalman Filter. *J Intell Robot Syst.* **78**(3), 485–497 (2014)
- Jia, S.M., Wang, K., Li, X.Z.: Mobile Robot Simultaneous Localization and Mapping Based on a Monocular Camera, *Journal of Robotics*, vol.2016, Article ID 7630340, 11 pages, (2016)
- Thrun, S., Liu, Y., Koller, D., Ng, A.Y., Ghahramani, Z., Durrant Whyte, H.: Simultaneous localization and mapping with sparse extended information filters. *Int. J. Robot. Res.* **23**(7–8), 693–716 (2004)
- Walter, M.R., Eustice, R.M., Leonard, J.J.: Exactly sparse extended information filters for feature-based SLAM. *Int. J. Robot. Res.* **26**(4), 335–359 (2007)
- Montemerlo, M.: FastSLAM: A factored solution to the simultaneous localization and mapping problem with unknown data association, PhD thesis, Carnegie Mellon University, (2003)
- Kim, C., Sakthivel, R., Chung, W.K.: Unscented FastSLAM: A robust and efficient solution to the SLAM problem. *IEEE Trans Robot.* **24**(4), 808–820 (2008)
- Ramaza, H., Hamid, D.T., Mohammad, A.N., Mohammad, T.: A square root unscented FastSLAM with improved proposal distribution and resampling. *IEEE Trans Ind Electron.* **61**(5), 2334–2345 (2014)
- Thrun, S., Montemerlo, M., Koller, D., Wegbreit, B., Nieto, J., Nebot, E.: FastSLAM using compressed occupancy grids, *Journal of Sensors*, vol.2016, Article ID 3891865, 23 pages, (2016)
- Won, D., Chun, S., Sung, S.: Improving mobile robot navigation performance using vision based SLAM and distributed filters, *International Conference on Control, Automation and Systems*, 186–191, (2008)
- Won, D., Chun, S., Sung, S., Lee, Y., Cho, J., Joo, J., Park, J.: INS/vSLAM system using distributed particle filter, *International Journal of Control, Automation, and Systems*, pp.1232–1240, (2010)
- Pei, F-J, Wu, M., Zhang, S.: Distributed SLAM Using Improved Particle Filter for Mobile Robot Localization, *The Scientific World Journal*, 86, Paper <https://doi.org/10.1155/2014/239531>, (2014)
- Pei, F.-j., Cheng, Y.-h., Li, H.: Distributed Simultaneous location and mapping based on partition of space-region. *Syst Eng Electron.* **37**(3), 639–645 (2015)
- Zarei-Jalalabadi, M., Malaek, S.M.B., Kia, S.S.: A Track-to-Track Fusion Method for Tracks With Unknown Correlations[J]. *IEEE Control Syst Lett.* **2**(2), 189–194 (2018)
- Barshalom Y.: Multitarget-Multisensor Tracking: Applications and Advances[M]// Multitarget-multisensor tracking : advanced applications. Artech House, (1993)
- Vidal-Calleja, T., Bryson, M., Sukkarieh, S.: On the Observability of Bearing-only SLAM, *IEEE International Conference on Robotics and Automation*, Roma, Italy, 10–14 April (2007)
- Goshen-Meskin, D., Bar-Itzhack, I.: Observability analysis of piece-wise constant systems - Part I: Theory. *IEEE Trans Aerosp Electron Syst.* **28**(4), 1056–1067 (Oct. 1992)
- Chong, C.Y., Allen, B., Hamilton, I. et al. Convex combination and covariance intersection algorithms in distributed fusion[C]// *Proc. Fusion'01.* (2001)
- Matzka, S., Altendorfer, R.: A Comparison of Track-to-Track Fusion Algorithms for Automotive Sensor Fusion[M]// *Multisensor Fusion and Integration for Intelligent Systems*. Springer Berlin Heidelberg, 69–81, (2009)
- Fong, L.W.: Multi-sensor track-to-track fusion via linear minimum variance sense estimators[J]. *Asian J Control.* **10**(3), 277–290 (2008)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Fujun Pei is an associate professor of the Faculty of Information Technology at the Beijing University of Technology. He received his B.E. in automation engineering and M.E. in control science engineering from Yanshan University in 2000 and 2002. He received his Ph.D. degree in navigation guidance and control from the Southeast University in 2005. His research interests include autonomous navigation, robotic intelligence and service robotics applications.

Mingjun Zhu received his B.E. in automation engineering and M.E. in control science engineering from Beijing University of Technology in 2016 and 2019. His research interests include simultaneous localization and mapping, driverless technology and data mining using machine learning.

Xiaoping Wu received his M.E. in control science engineering from Beijing University of Technology in 2017. His research interests are simultaneous localization and mapping, distributed algorithm and multi-sensor data fusion technology.