International Journal of Control and Automation
Vol.10, No.3 (2017), pp.129-138
http://dx.doi.org/10.14257/ijca.2017.10.3.11

An Innovative Combined Filter SLAM Algorithm for Mobile Robots in Large-Scale Environment

Jun Cai, Xiaolin Zhong* and Yi Zhang

National Information Accessibility Engineering Research & Development Center, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
chris2616@126.com

Abstract

By combining the strength of the Rao-Blackwellized particle filter (RBPF) and extended information filter (EIF), this paper presents a novel Combined Filter SLAM algorithm which can be efficiently employed in large-scale scenarios. Local maps are effectively produced through RBPF-SLAM algorithm and then periodically fused into an EIF SLAM algorithm. A binary-tree based divide and conquer (D&C) strategy is also applied to further improve the computational efficiency in real time environment. Simulations and experiments using the Victoria Park dataset demonstrate the consistency and efficiency of our proposed Combined Filter SLAM algorithm in large-scale environment.

Keywords*: Combined Filter SLAM, RBPF, EIF, D&C strategy, Sub-map joining*

1. Introduction

Simultaneous localization and mapping (SLAM) is a method to help robots explore, navigate, and map an unknown environment [1]. It is well known that traditional methods for SLAM based on the extended Kalman filter (EKF) suffer computational complexity problems when dealing with large-scale environments, as well as inconsistencies for non-linear SLAM problems [2].

In recent years, SLAM related research has received extensive attention and endeavors to improve the consistency property and computational efficiency. The goal is being able to map large-scale environments in real time [3]. Combining the submap based idea with combined filters can make SLAM algorithms even more efficient. A number of such algorithms are available, *e.g*., by mixing PF with Gaussian filters [4], FastSLAM with EKF [5], RBPF with UKF [6] *etc*. However, enabling real time SLAM implementation in an increasingly unstructured large-scale environment is still a great challenge.

In this paper, a new Combined Filter SLAM approach is presented to map large-scale environments. The algorithm is a judicious combination of RBPF and EIF, combined with a divide and conquer local mapping scheme. Being a local mapping algorithm, it can provide more consistent results, compared with submap based sparse EIF SLAM algorithm – Sparse Local Submap Joining Filter (SLSJF) SLAM [7], reported to have the same low cost, but computing an absolute map [8]. CF SLAM is also conceptually simple and rather easy to implement.

This paper is organized as follows. The overall structure of the proposed algorithm is outlined in Section 2. Section 3 describes the probability distribution of RBPF-SLAM algorithm and its conversion, while Section 4 presents the binary-tree based Divide and Conquer scheme. In Section 5 we provide a brief review of the EIF SLAM algorithm for fusing local submaps into a global map. In Section 6 we use the Victoria Park dataset to test our algorithm with real data. Finally in Section 7 we draw the main conclusions of this work and discuss future directions of research.

ISSN: 2005-4297 IJCA

2. The Overall Structure of Combined Filter SLAM

The objective of local submap joining is to combine the local submaps and obtain a global map containing all the features. Similar to the sequential submap joining in [6], CF SLAM fuses the local submaps one by one. That is, first set local submap 1 as the global map, then fuse local map k ($k \geq 2$) into the global map in sequence.

It is assumed that a consistent local submap can be constructed by some SLAM algorithm and is expressed by

$$M^L = \left(X^L, P^L \right) \tag{1}$$

where X^L (here the superscript 'L' here stands for the "local" submap) is an estimation of the state vector which contains the final robot pose X_r^L = (x, y, α) and all the local feature positions which are shown as X_1^L, …, X_n^L. That is to say, X^L can be represented as

$$X^L = \left(X_r^L, X_1^L, \cdots, X_n^L \right) = \left(x_r^L, y_r^L, \alpha_r^L, x_1^L, y_1^L, \cdots, x_n^L, y_n^L \right) \tag{2}$$

and P^L is the associated covariance matrix.

In real applications, the robot will start to build the (k + 1)th local map as soon as the kth local map is finished, thus the robot ending pose in the kth local map is exactly the same as the robot starting pose in the (k + 1)th local map. The overall structure of the proposed algorithm is outlined in the following flow chart.

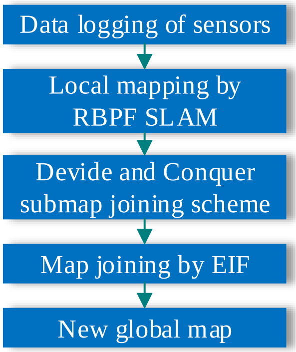


Figure 1. The Overall Structure of the Proposed Algorithm

The algorithm has three main highlights:

a) Local mapping is carried out using the RBPF-SLAM to build a sequence of submaps. One important task is to convert the Gaussian Mixture Model of RBPF-SLAM into a single multi-dimensional Gaussian model for the following submap joining process through EIF.

b) In contrast with sequential submap joining strategy followed by SLSJF, the D&C strategy is followed to decide when map joining takes place.

c) Submap joining is carried out using EIF, keeping mobile robot positions from each local submap in the final map: this allows to exploit the exact sparse structure of the information matrix and the joining can be carried out in linear time with the final size of the map.

3. Local Mapping

In this section, we will address the Rao-Blackwellized particle filter based local submap building in detail, especially the probability distribution and its conversion.

3.1. The SLAM Problem

Consider the robot pose s_t, the map learned thus far m_t, current observations z_t, and control signal u_t, here suppose all at time t. The set of observations and control signals from time 0 to t are defined as z^t and u^t respectively. Generally, our goal is to estimate the density

$$p\left(s_t, m_t \middle| z^t, u^t\right) \tag{3}$$

As the set of observations and controls arrives over time, we define $x_t = \{s_t, m_t\}$ be the complete state. Applying Bayes rule [9] on Equation (3) then we get

$$\begin{aligned} & p\left(x_t \middle| z^t, u^t\right) = Bel\left(x_t\right) \\ & = \eta p\left(z_t \middle| x_t\right) \int p\left(x_t \middle| u_t, x_{t-1}\right) p\left(x_{t-1} \middle| z^{t-1}, u^{t-1}\right) dx_{t-1} \\ & = \eta p\left(z_t \middle| x_t\right) \int p\left(x_t \middle| u_t, x_{t-1}\right) Bel\left(x_{t-1}\right) dx_{t-1} \end{aligned} \tag{4}$$

where η is a normalizing constant.

Equation (4) allows us to recursively estimate the posterior probability of maps and robot pose if the two distributions $p(z_t \mid x_t)$ and $p(x_t \mid u_t, x_{t-1})$ are given. For SLAM, an analytical form for $Bel(x_t)$ is hard to obtain and as such the Bayes filter is not directly applicable. Instead we assume all the variables are Gaussian distributed and utilize a Rao-Blackwellized particle filter based SLAM algorithm to realize Equation (4).

3.2. Gaussian Mixture Model (GMM) of RBPF-SLAM

In the case of SLAM, the posterior is factored as

$$\begin{aligned} & Bel\left(x_t\right) = Bel\left(x_t, m_t\right) = p\left(s_t, m_t \middle| z^t, u^t\right) \\ & = p\left(s_t \middle| z^t, u^t\right) \prod_k^n p\left(m(k) \middle| s^t, z^t, u^t\right) \end{aligned} \tag{5}$$

This factored distribution is represented as a set of K particles, with jth particle P^j consisting of an importance weight w^j, a robot pose s^j, and n Gaussian feature estimations described by their mean μ^j and covariance Σ^j, and the form is as follows

$$P^j = \left\{ w^j, s^j, \mu_1^j, \Sigma_1^j, \ldots, \mu_n^j, \Sigma_n^j \right\} \tag{6}$$

In order to represent the distribution of each particle as low-dimensional Gaussians, the particle can equivalently be represented as

$$P_t^j = \left\{ w^j, S^j, Q_t^j \right\} \tag{7}$$

where $S^j = [s^j, \mu_1^j, \ldots, \mu_n^j]$ denotes the concatenation of the robot pose with all feature states, and Q^j denotes a block-diagonal covariance matrix which is constructed from the robot covariance and the covariance of each feature, which is shown as:

$$Q^j = \begin{bmatrix} 0 & & & \\ & \Sigma_1^j & & \\ & & \ddots & \\ & & & \Sigma_1^j \end{bmatrix} \tag{8}$$

The probability distribution of RBPF-SLAM algorithm is a Gaussian Mixture Model (GMM), while the EIF SLAM algorithm requires local submaps are subjected to the above distribution in the form of a single multi-dimensional Gaussian when they are fusing into a global map. Thus, the conversion from the Gaussian Mixture Model of RBPF-SLAM into a single multi-dimensional Gaussian model becomes a requisite task.

3.3. Conversion to a Single Multi-Dimensional Gaussian

Through a moment matching procedure [10], a single multi-dimensional Gaussian model with mean M_t and covariance C_t can be gained from Gaussian Mixture Model by

$$M_t = \sum_K w_t^j S_t^j \tag{9}$$

$$C_t = \sum_K w_t^j \left[C_t^j + \left(x_t^j - x_t \right)\left(x_t^j - x_t \right)^T \right] \tag{10}$$

where w^j denotes the importance weight, C_t^j refers to the covariance of individual particle caused by sensor noise, and $(x_t^j - x_t)$ $(x_t^j - x_t)^T$ stands for the variation between particles caused by robot noise.

Due to each particle of RBPF-SLAM possesses its own data association decision in real implementation, the number of features and corresponding ordering may vary from particles. In order to carry out Equations (9) and (10), an indispensable task is to track the correspondences between features of each particle.

Algorithm 1 Obtain Single Multi-Dimensional Mean and Covariance

1. Assign index to observation

Augment each particle with a set of correspondence variables $\lambda_{i,t}^j$ (*e.g.* $\lambda_{i,t}^j = \Theta$ refers to the *i*th feature in the *j*th particle corresponds to the Θth observation in the environment at time *t*).

$$P_t^j = \left\{ w_t^j, S_t^j, \mu_{1,t}^j, \Sigma_{1,t}^j, \lambda_{1,t}^j \ldots, \mu_{n,t}^j, \Sigma_{n,t}^j, \lambda_{n,t}^j \right\}$$

2. Apply the maximum likelihood data association

Use the data association results to implement particle updating and augmenting for each P_t^j.

3. Re-arrange and re-modify each particle to produce a common feature set

The concatenation of the robot pose with all feature states in Equation (7) can be represented as:

$$S_t^j = \left\{ s_t^j, \mu_{\delta_t(1,j),t}^j, \ldots, \mu_{\delta_t(n,j),t}^j \right\}$$

where δ denote the reverse function of $\lambda_{i,t}^j$ and equation $\delta_t(\Theta, j) = i$ refers to the Θth observation in the common feature set corresponds to the *i*th feature in the *j*th particle. Thus, the single multi-dimensional mean can be acquired via Equation (9).

And the related covariance is:

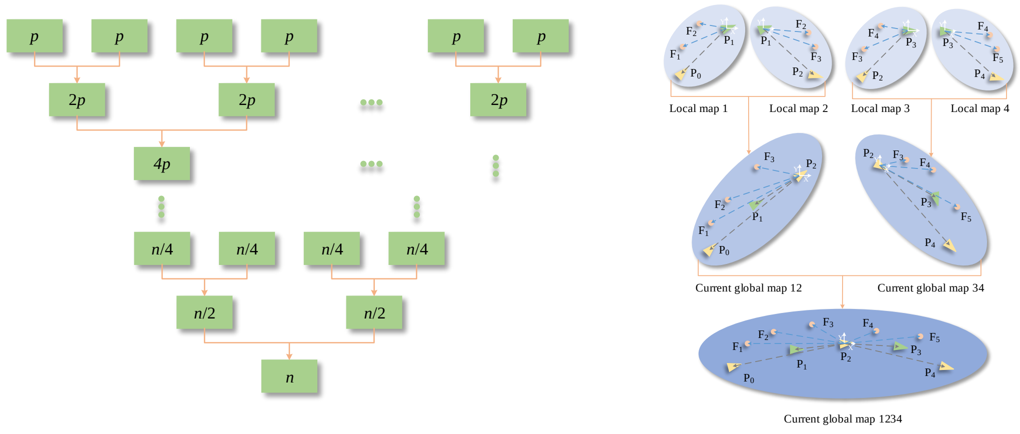
$$Q_t^j = \begin{bmatrix} 0 & & & \\ & \Sigma_{\delta_t(1,j),t}^j & & \\ & & \ddots & \\ & & & \Sigma_{\delta_t(n,j),t}^j \end{bmatrix}$$

Through Algorithm 1, each particle can get a single multi-dimensional mean and covariance, hence a single Gaussian distribution can then be obtained by means of Equations (9) and (10). However, a common feature will be discarded as long as it has no corresponding feature in any particle.

4. The Divide and Conquer Submap Joining Scheme

After a sequence of local submaps with minimum size *p* is produced via the RBPF-SLAM algorithm, we employ the Divide and Conquer (D&C) submap joining scheme to acquire a single final stochastic map.

Unlike Local Map Sequencing strategy to join maps sequentially [11], D&C scheme joins local maps in a binary-tree hierarchical fashion, as shown in Figure 2(a). The leaves of the binary-tree are the sequence of local submaps that produced via the RBPF-SLAM. And the in-between nodes refer to the maps derived from the submap joining procedures. As a result, the root of the binary-tree stands for the final global map. And the detailed D&C map joining process is vividly depicted in Figure 2(b).



(a) Binary-Tree Fashion of D&C Scheme (b) The Detailed D&C Map Joining Process

Figure 2. The Divide and Conquer Submap Joining Scheme

Map Joining is carried out using EIF, and the leaves of the binary-tree denote the sequence of l local submaps of certain size p. Submaps are joined in pairs to compute $l/2$ maps of their double size $2p$, which will then be merged into $l/4$ maps of quadruple size $4p$, till ultimately local maps of size n=2 will be joined into one global map of size n.

Utilizing a stack to store the intermediate local maps, the D&C scheme can easily carry out a post-order traversal of the binary-tree (see the following Algorithm 2).

Algorithm 2 Traverse the Binary-Tree Using a Stack

stack = new()
l_0 = rbpf_slam()
stack = push(l_0, stack)
{
Main loop: post-order traversal of the binary-tree
}
repeat
  l_k = rbpf _slam()
  while ¬ empty(stack) and then size(l_k) $\geq$ size(top(stack)) **do**
    l = top(stack)
    stack = pop(stack)
    l_k = join(l, l_k)
end while
stack = push(l_k, stack)
until end_of_map
{
Join all the submaps in stack for full map recovery
}
while ¬ empty(stack) **do**
  l = top(stack)
stack = pop(stack)
l_k = join(l, l_k)
end while
return (l_k)

5. Local Map Joining Procedure by EIF

EIF is an algebraic equivalent of EKF, in which the parameters of interest are information states and the inverse of the covariance matrix (information matrix) rather than the states and covariance. Initialization in the information space is easier than in Kalman filter and the update stage of the information filter is computationally simpler than the Kalman filter, for more details see [12].

The EIF can also be used to achieve the submap joining procedures, as described in the Sparse Local Submap Joining filter (SLSJF) SLAM [13]. Generally, in submap joining with EKF, correspondences are created through assuming a perfect condition $z = 0$ and $Q = 0$. While covariance measurements 0 are not allowed for Q^{-1} is required in the information form. Thus, after joining two local submaps (μ_1, Σ_1) and (μ_2, Σ_2), the resulting map (ξ, Ω) is predicted with the information of the first submap, and an initial 0 from the second submap. The innovation is computed considering the second submap as a set of measurements for the current global map ($z_t = \mu_2$, $Q_t = \Omega_2^{-1}$), and the ultimate update procedure calculates the information state ξ and the information matrix Ω utilizing the EIF equations.

Algorithm 3 Submap Joining using EIF

1. Jacobians

$$G = \frac{\partial g(\mu^+)}{\mu^+}\Big|_{\hat{\mu}^+} \qquad H = \frac{\partial h(\mu^-)}{\mu^-}\Big|_{\hat{\mu}^-}$$

2. Initialization

$$\mu^- = g(\mu_1, \mu_2); \quad \xi^- = \begin{bmatrix} \xi_1 \\ 0 \end{bmatrix}; \quad \Omega = \begin{bmatrix} \Omega_1 & 0 \\ 0 & 0 \end{bmatrix}$$

3. Innovation

The measurement residual v and covariance Q^{-1} are calculated as

$$\begin{cases} v = \mu_2 - h(\mu^-) \\ Q^{-1} = \Omega_2 \end{cases}$$

4. Update

The final update step computes information state ξ, information matrix Ω and defined state μ.

$$\begin{cases} \Omega = \Omega^- + H^T \Omega_2 H \\ \xi = \xi^- + H^T \Omega_2 (v + H\mu^-) \\ \mu = \Omega / \xi \end{cases}$$

The information matrix resulting from the submap joining procedures utilizing EIF is wholly sparse if the robot positions are maintained in the final information state. This is a situation very similar to the full SLAM problem, only a fraction corresponding to the final robot positions in each local map. And there is an added calculation of the final state μ, to make it available for the possible joining procedures. The state vector recovery step can be actualized via a preordering of minimum degree of the information matrix and the Cholesky factorization is used to solve the sparse linear equation.

6. Experiments and Analysis

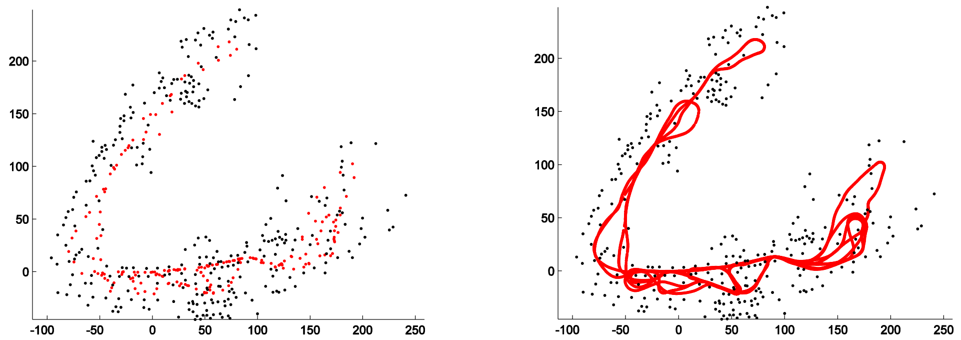
The performance of our method is examined in this section. In order to do so, we employ the Victoria Park dataset where a truck equipped with a range laser scanner was driven through the Victoria Park in Sydney, Australia (publicly available at http://www-personal.acfr.usyd.edu.au/nebot/victoria_park.htm). The following experiments allowed us to check the validity, consistency and computational efficiency of the proposed algorithm compared to the conventional single-map-type EKF SLAM.

6.1. Simulation Environment

The Victoria Park dataset describes a path through an area of around 197m × 93m. This sequence consists of 7247 frames along a trajectory of 4 kilometers, recorded over a total time of 26 minutes. The dataset contains sensor readings from steering and rear-axis wheel (odometry) and laser range finder (one 360 degrees scan per second) along with the data from a GPS. For the laser range data a tree detector function is provided together with the dataset. These Trees within Victoria Park are used as point features in the map. We assume that the data association is known and focus only on verifying the SLAM solution. The experiments are conducted on ThinkPad E450C with 2.7GHz Intel Core i5-5200U Dual Core Processor, 4GB of RAM, and all programs are implemented in MATLAB R2013a.

6.2. Experimental Results

As shown in Figure 3, the feasibility and validity of the proposed CF SLAM algorithm is examined by joining 200 and 7000 local submaps.



(a) Map Obtained by Joining 200 Local Maps (b) Map Obtained by Joining 7000 Local Maps

Figure 3. The Map Joining Results using CF SLAM for Victoria Park Dataset

It has been suggested that local submap based schemes can enhance the consistency of SLAM by keeping the robot orientation error small [14]. The consistency of our method is examined through the map consistency analysis. Concretely speaking, the Normalized Innovation Squared (*NIS*) can be employed to analyses the consistency while the ground truth for the state variable is not known.

$$NIS = v_k^t S_k^{-1} v_k < \chi^2_{r,1\text{-}\alpha} \tag{11}$$

Given the estimation of the innovation vector v and the innovation matrix S, the state (x^*, P) estimation is consistent while $NIS < \chi^2_{r,1\text{-}\alpha}$, or else the estimation will become inconsistent for over optimization. Here, $r = dim(x_k)$ is the degree of freedom, and usually, α is 0.05 denoting the desired significance level. In Figure 4, the dashed line represents the χ^2 corresponding to each step, and the solid line refers to the *NIS*, which indicates that our approach performs inside the boundaries of theoretical consistency.

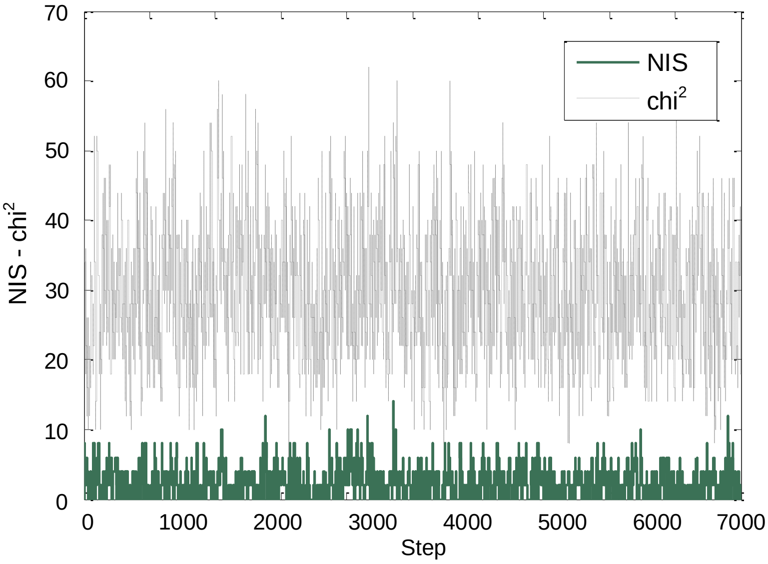
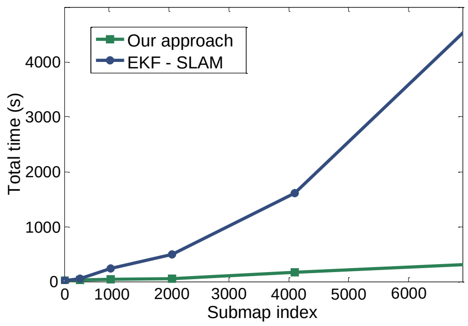
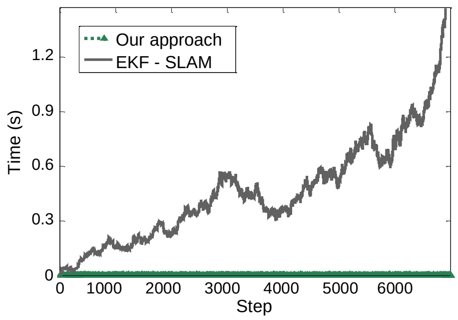


Figure 4. Consistency Test

Regarding the computational cost, we tested the computational time of both standard EKF and our method by carrying out 30 Monte Carlo runs on the simulated experiment, as shown in Figure 5(a). We can see that the EKF computation time increases on a quadratic order to the number of features in the map, while our approach increases almost lineally. This improvement is also visible in Figure 5(b). The time per step of our approach is almost constant, while the time per step of EKF increases with the size of map. Also, the time per step of our approach has some peaks because of the submap fusion events. However, these peaks do not bring any drastic time delay, therefore the improvement of computation efficiency is considerable.



(a) Time Required to Compute the Mission



(b) Time Per Step along the Mission

Figure 5. Time Required to Compute the Whole Mission

7. Conclusion

In this paper, the submap based Combined Filter SLAM approach, which combines the advantages of both RBPF and EIF, has been demonstrated to be suitable to consistently map large-scale scenarios. On the one hand, the RBPF-SLAM based local submap building improves the robustness of data association and the validity of estimation. On the other hand, the EIF-SLAM based submap joining globally allows uncertainty to be remembered over long robot trajectories. Finally, simulation and experiments using the publicly available Victoria Park dataset clearly demonstrate the consistency and efficiency of the proposed algorithm. And more work is required to determine the best submap joining strategy, improve the robustness, and extend CF SLAM to 3D local submap joining. Research along these directions is underway.

Acknowledgments

This work was partially supported by Chongqing Science and Technology Commission (cstc2015jcyjB0241), National Natural Science Foundation of China (51075420) and National Ministry of Science and Technology Project of International Cooperation (2010DFA12160). Meanwhile, the authors acknowledge the contribution of the Australian Centre for Field Robotics, University of Sydney, for using their Victoria Park Dataset.

References

[1] R. Smith, M. Self and P. Cheeseman, "Estimating uncertain spatial relationships in robotics", Autonomous Robot Vehicles, **(1990)**, pp. 167-193.
[2] S. J. Julier and J. K. Uhlmann, "A counter example to the theory of simultaneous localization and map building", Proceedings of IEEE International Conference on Robotics and Automation, **(2001)**; Korea.
[3] W. H. Durrant and T. Bailey, "Simultaneous localization and mapping: Part I", IEEE Robotics and Automation Magazine, vol. 13, no. 2, **(2006)**, pp. 99-110.
[4] J. L. Blanco, J. A. Fernandez-Madrigal and J. Gonzalez, "Toward a unified bayesian approach to hybrid metric—Topological SLAM", IEEE Transactions on Robotics, vol. 24, no. 2, **(2008)**, pp. 259-270.
[5] A. Brooks and T. Bailey, "HybridSLAM: combining FastSLAM and EKF-SLAM for reliable mapping", Algorithmic Foundation of Robotics, **(2009)**, pp. 647-661.
[6] M. H. Li, B. R. Hong and R. H. Luo, "Novel mobile robot simultaneous localization and mapping using rao-blackwellised particle filter", International Journal of Advanced Robotic Systems, vol. 3, no. 3, **(2006)**, pp. 231-238.
[7] Z. Wang, "Exactly sparse extended information filters for SLAM", Ph.D. Thesis. University of Technology, Sydney, **(2007)**. available online http://services.eng.uts.edu.au/˜sdhuang.
[8] S. Huang and G. Dissanayake, "Convergence and consistency analysis for extended kalman filter Based SLAM", IEEE Transactions on Robotics, vol. 23, no. 5, **(2007)**, pp. 1036-1049.
[9] S. Thrun, "Robot mapping: A survey", Carnegie Mellon university, Technical Report CMU-CS-02-11, **(2002)**.
[10] Y. Bar-Shalom, X. R. Li and T. Kirubarajan, "Estimation with applications to tracking and navigation", Williey Inter Science, **(2001)**; New York, USA.
[11] K. R. B. Chandra, D. W. Gu and I. Postlethwaite, "Square root cubature information filter", IEEE Sensor Journal, vol. 13, no. 2, **(2013)**, pp. 750-758.
[12] K. P. B. Chandra, D-W. Gu and I. Postlethwaite, "Cubature information filter and its applications", IEEE American Control Conference, **(2011)**.
[13] S. Huang, Z. Wang and G. Dissanayake, "Sparse local submap joining filters for building large-scale maps", IEEE Transactions on Robotics, vol. 24, **(2008)**, pp. 1121-1130.
[14] J. A. Castellanos, R. Martinez-Cantin, J. D. Tardos and J. Neira, "Robocentric map joining: Improving the consistency of EKF-SLAM", Robot. Auton. Syst., vol. 55, **(2007)**, pp. 21-29.

Authors

Jun Cai, he is an associate professor at Automation School, Chongqing University of Posts and Telecommunications (CQUPT), China. He was born in China, in 1977. He receive master degree in automation from University of Electronic Science and Technology of China. He is currently an associate director of experimental center at Chongqing University of Posts and Telecommunications. His research interests include navigation, intelligent control and pattern recognition.

Xiaolin Zhong, he received his B.S. degree in automation from Shenyang University of Technology, Shenyang, China, in 2013. Currently, he is a graduate student major in control theory and control engineering at Chongqing University of Posts and Telecommunications, Chongqing, China. His research interests include navigation, path planning, simultaneous localization and mapping (SLAM) for mobile robots and intelligent control.

Yi Zhang, he is currently the vice president of School of Advanced Manufacturing Engineering, Chongqing University of Posts and Telecommunications, China. He is also the director of the Information Accessibility Engineering Research & Development Center. He received M.S. degree in automation from Heifei Institute of Technology, Hefei, China, in 1997. He received his Ph.D. degree in mechanical science and engineering from Huazhong University of Science and Technology, Wuhan, China, in 2002. His research interests include intelligent human-computer interaction, robot technology and its application.