

Article

Simultaneous Localization and Mapping with Iterative Sparse Extended Information Filter for Autonomous Vehicles

Bo He ¹, Yang Liu ¹, Diya Dong ¹, Yue Shen ^{1,*}, Tianhong Yan ² and Rui Nian ¹

¹ School of Information Science and Engineering, Ocean University of China, 238 Songling Road, Qingdao 266100, China; E-Mails: bhe@ouc.edu.cn (B.H.); oucliuyang@126.com (Y.L.); dongdiya007@163.com (D.D.); nianrui_80@163.com (R.N.)

² School of Mechanical and Electrical Engineering, China Jiliang University, 258 Xueyuan Street, Xiasha High-Edu Park, Hangzhou 310018, China; E-Mail: yanth@163.com

* Author to whom correspondence should be addressed; E-Mail: shenyue@ouc.edu.cn; Tel.: +86-187-6620-3751.

Academic Editor: Vittorio M. N. Passaro

Received: 27 April 2015 / Accepted: 6 August 2015 / Published: 13 August 2015

Abstract: In this paper, a novel iterative sparse extended information filter (ISEIF) was proposed to solve the simultaneous localization and mapping problem (SLAM), which is very crucial for autonomous vehicles. The proposed algorithm solves the measurement update equations with iterative methods adaptively to reduce linearization errors. With the scalability advantage being kept, the consistency and accuracy of SEIF is improved. Simulations and practical experiments were carried out with both a land car benchmark and an autonomous underwater vehicle. Comparisons between iterative SEIF (ISEIF), standard EKF and SEIF are presented. All of the results convincingly show that ISEIF yields more consistent and accurate estimates compared to SEIF and preserves the scalability advantage over EKF, as well.

Keywords: autonomous vehicles; autonomous navigation; SLAM; SEIF; consistency; scalability; iteration

1. Introduction

Autonomous navigation in environments without prior knowledge is a crucial problem for various autonomous vehicles [1–3]. The inertial navigation system, also known as INS, had been widely used to provide position estimates. However, the estimate error of the INS grows with time unboundedly, resulting in an inaccurate estimate ultimately. GPS can be utilized as an auxiliary means to modify the estimate in some applications. Unfortunately, the GPS signal is not available in some common environments (e.g., indoor and underwater), which requires an alternative method to implement accurate navigation.

Many SLAM algorithms have been proposed based on the probabilistic formulation of the problem [4] in the past few years. Among the SLAM algorithms, EKF-based SLAM shows its competency to solve the problem. Due to the simplicity of the algorithm, EKF SLAM has been widely used in many applications [5,6]. However, the shortcomings of EKF SLAM are also exposed at the same time. The computational cost of the algorithm is quadratic to the number of features in the environments, making the online application in large-scale environments impossible [7]. In addition, in order to deal with the nonlinearity, EKF SLAM expands the nonlinear function in Taylor series and uses the first order to make an approximation. The truncation error of the approximation leads to inconsistent and inaccurate state estimation.

A number of improved methods are then proposed to overcome the weakness of EKF SLAM, and some work focuses on the scalability limitation. Among them, the information-based filter proves to be effective to deal with large-scale environments. Instead of a mean vector and a dense covariance matrix, the extended information filter (EIF) adopts the information matrix and information vector to describe the SLAM distribution [8]. Since the complexity of EIF is even greater than the EKF, it seems that EIF is not capable of solving the scalability problem as well. However, a novel insight reveals that the information matrix is sparse in the context of feature-based SLAM. Taking advantage of the sparseness, the sparse extended information filter (SEIF) was then presented [9,10]. By pruning weak links in the information matrix, SEIF keeps the information matrix sparse all of the time, thus allowing all of the updates to be executed in constant time, regardless of the size of the environment.

Another kind of SLAM algorithm is based on particle filters [11,12]. The particle filter is a kind of Monte Carlo method applying samples to present the probability density distribution, which can be employed to any state models, and when the sample size tends to be infinite, it can approximate any form of probability density distribution. FastSLAM is one of this kind of algorithm [13,14]. It can deal with multi-hypothesis association problems because of the independence of the particles. According to Tim Bailey *et al.* [15], the FastSLAM algorithm cannot guarantee the consistency of the robot pose estimation. In addition, increasing the number of particles can improve the consistency to a certain extent, but it will also increase the computational cost.

SEIF SLAM has been demonstrated to be efficient and scalable in real-world applications [16,17]. However, as a dual form of EKF, SEIF suffers from linearization errors, like EKF, as well, which may result in inaccuracy and inconsistency [15]. Moreover, due to the elimination of weak links in the sparsification step and the state recovery step, SEIF can even more easily be divergent than the EKF algorithm [18]. To refine the estimate results of SEIF, some approaches have been designed, and one of

them is the exactly sparse extended information filter (ESEIF) proposed by Matthew Walter *et al.* [19]. ESEIF keeps exact sparseness by marginalizing out the robot pose occasionally from the distribution and then relocalizing the robot, which can maintain conservative pose and map estimates. ESEIF also has been successfully implemented in some real-world applications [20,21]. However, compared to SEIF, ESEIF also suffers from the same problem as EKF for nonlinear applications. Ryan M. Eustice *et al.* [22] proposed an exactly sparse delayed-state filter (ESDSF). It is based on a delayed-state framework in which the information matrix is exactly sparse. This framework can be used in view-based representation of the environments, which rely on scan matching sensor data; while in feature-based environments, it is not applicable anymore.

Iteration is an effective method in nonlinear estimation problems and has been demonstrated to be effective in SLAM algorithms [23–25]. The iterated extended Kalman filter (IEKF) is one of the most popular iterated filters [26]. IEKF achieves better consistency and accuracy compared to the original EKF. However, the learning speed of IEKF is even slower than EKF, which limits its applications. In this paper, we investigate the role of iteration in the SEIF SLAM algorithm and then propose an iterative sparse extended information filter (ISEIF). We will first introduce the reasons that lead to the inconsistency in the SEIF SLAM algorithm. Then, we will derive the iterative form of SEIF and illustrate how it works. Theoretical analysis will be given to prove the consistency and accuracy, and the scalability will be considered, as well. To the best of our knowledge, this is the first time information filters have been refined through iterative methods.

To demonstrate the advantages of the ISEIF algorithm, Monte Carlo simulation results will be first presented. After that, the application of the ISEIF SLAM algorithm to autonomous vehicles, including an autonomous ground vehicle (AGV) and an autonomous underwater vehicle (AUV), will be studied. The Victoria Park dataset was collected by an autonomous ground vehicle, which was developed at the Australian Center for Field Robotics (ACFR) at the University of Sydney [27]. It carries a number of essential sensors, such as a velocity encoder, a steering angle sensor, an inertial measurement unit, GPS, a laser and a vision sensor. We use this dataset to demonstrate the ability of autonomous outdoor navigation for ISEIF SLAM in large-scale unstructured environments. The C-Ranger AUV platform was developed at the Underwater Vehicle Laboratory at Ocean University of China [28,29], which is equipped with a variety of sensors, such as GPS, attitude and heading reference system (AHRS), digital compass, gyro, Doppler velocity log (DVL) and multibeam imaging sonar for active sensing in underwater environments. An output feedback controller and motion planning algorithm is adopted to achieve the task of path following [30,31]. We will also verify the feasibility of ISEIF SLAM for the C-Ranger AUV by sea trials in Tuandao Bay. Experimental results will show that compared to SEIF, ISEIF performs a more conservative accurate estimate and also preserves the scalability advantage over EKF SLAM.

The remainder of the paper is organized as follows: in Section 2, we formulate the basic equations of iterative SEIF with detailed theoretical analysis. Both simulated and experimental results will be presented to verify the superiority of the proposed algorithm in Section 3. Finally, we have a discussion of the ISEIF SLAM algorithm and draw the conclusion.

2. The Iterative SEIF

As an alternative form of EKF, EIF suffers from the same statistical and analytical linearization error propagation, which may cause inconsistency in SLAM. Additionally, due to some unique steps (*i.e.*, sparsification and state recovery) in SEIF, the consistency and accuracy of SEIF is even inferior to EKF algorithm. In this section, we will first review the basic steps of SEIF. The causes of inconsistency and inaccuracy existing in SEIF will then be analyzed. After the analysis, an adaptive ISEIF is proposed for the purpose of improving the consistency and accuracy of SEIF.

2.1. Review of SEIF

The information form of the SLAM problem is as follows.

Let ξ_t denote the state vector at time t , which includes the robot pose x_t and the set of features location $\{y_1, \dots, y_n\}$:

$$\xi_t = (x_t, y_1, \dots, y_n)^T \quad (1)$$

In SLAM problem, the robot aims to find a probabilistic estimate of ξ_t . Following the Bayesian rule, the estimation problem transforms to calculate a posterior distribution $p(\xi_t|z^t, u^t)$, where z^t and u^t are the observations and control inputs at time t , respectively. Reviewing the EKF solution to the SLAM problem, the posterior $p(\xi_t|z^t, u^t)$ is represented by the mean μ_t and the covariance matrix Σ_t :

$$p(\xi_t|z^t, u^t) \propto \exp\left\{-\frac{1}{2}(\xi_t - \mu_t)^T \Sigma_t^{-1}(\xi_t - \mu_t)\right\} \quad (2)$$

Instead of μ_t and Σ_t , information filters represent the posterior through H_t and b_t , which are the information matrix and information vector, respectively. They are defined as follows:

$$H_t = \Sigma_t^{-1} \quad (3)$$

$$b_t = \mu_t^T H_t \quad (4)$$

Then, the posterior can be represented in the commonly-known information form of the Kalman filter:

$$p(\xi_t|z^t, u^t) \propto \exp\left\{-\frac{1}{2}\xi_t^T H_t \xi_t + b_t \xi_t\right\} \quad (5)$$

The measurement model is:

$$z_t = h(\xi_t) + \varepsilon_t \quad (6)$$

The measurement z_t is usually governed by a deterministic nonlinear function h of the state vector ξ_t , where the term ε_t represents independent Gaussian noise.

In addition to the measurement update step, SEIF also needs to perform a motion update, a sparsification step and a state recovery step to solve the SLAM problem.

2.2. Consistency Analysis of SEIF

The SEIF SLAM shows poor consistency compared to the traditional EKF SLAM algorithm, which results from approximations. There are three approximation steps existing in SEIF:

First, SEIF approximated the nonlinear function in SLAM by first order Taylor series. It deals with the nonlinear measurement model by expanding it in a Taylor series around μ_t and uses the following equation to make an approximation:

$$h(\xi_t) \approx h(\mu_t) + C_t^T [\xi_t - \mu_t] \quad (7)$$

where C_t is the Jacobian of h with respect to μ_t .

Based on the linearization, we can then update the measurement equations [9].

After the measurement update equations are executed, we obtain the posterior estimate μ_t . However, as illustrated in Figure 1a, the pose estimate μ_t is still far away from the true state ξ_t , which results in estimate error. As the robot moves on, the next pose prediction μ_{t+1}^- , which is calculated based on the previous estimate μ_t ; the motion model h is even further to the true pose ξ_{t+1} ; and the estimate result μ_{t+1} after measurement update equations is still further to the true state ξ_{t+1} , as illustrated in Figure 1b. The result shows that error will accumulate over time.

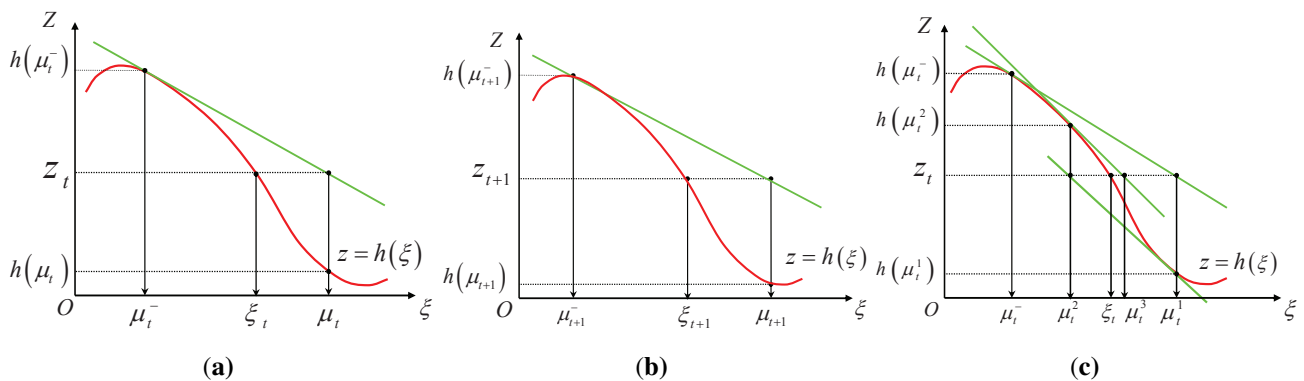


Figure 1. (a) Estimate error at time t ; (b) estimate error at time $t + 1$; and (c) the iterative process.

Second, SEIF only maintains the information vector and the information matrix, while we also need the estimated state vector during some steps. SEIF gets the state vector by the state recovery step, which is formulated as an optimization problem; hence, the recovery of the state is also an approximation, and this leads to estimation error, as well.

Last, the information matrix is not naturally sparse in the calculating process; SEIF prunes the weak links in the matrix to maintain the sparseness of the information matrix. Therefore, this is actually an approximation, as well, which also leads to estimation error.

Due to the three approximation steps, SEIF tends to be more inconsistent compared to traditional EKF SLAM, although it keeps a very fast speed. Among the three reasons, the first one is the largest contribution to the inconsistency. Therefore, we will focus on reducing the linearization error to improve the consistency of SEIF.

2.3. The Adaptive ISEIF

The modified measurement update equations can be obtained by an iterative method, which is termed the iterative sparse extended information filter. Instead of updating the state as a linear combination of the prediction and the innovation, we solve it as a maximum *a posteriori* (MAP) problem. The estimate of the MAP problem can be obtained by iteratively relinearizing the measurement equation.

The conditional probability density function (pdf) of ξ_t given Z^{t+1} can be written as:

$$\begin{aligned} p[\xi(t+1)|Z^{t+1}] &= p[\xi(t+1)|z(t+1), Z^t] \\ &= \frac{1}{c} p[z(t+1)|\xi(t+1)] p[\xi(t+1)|Z^t] \\ &= \frac{1}{c} \mathcal{N}[z(t+1); h[t+1, \xi(t+1)], R(t+1)] \\ &\quad \cdot \mathcal{N}[\xi(t+1); \mu(t+1|t), P(t+1|t)] \end{aligned} \quad (8)$$

Maximizing the above equations is equivalent to minimizing the following equations:

$$\begin{aligned} J[\xi(t+1)] &= \frac{1}{2} \{z(t+1) - h[t+1, \xi(t+1)]\}' R(t+1)^{-1} \\ &\quad \cdot \{z(t+1) - h[t+1, \xi(t+1)]\} \\ &\quad + \frac{1}{2} [\xi(t+1) - \mu(t+1|t)]' P(t+1|t)^{-1} \\ &\quad \cdot [\xi(t+1) - \mu(t+1|t)] \end{aligned} \quad (9)$$

J can be expanded in a second order Taylor series, and we could use the Newton–Raphson algorithm to get the MAP estimate of ξ_{t+1} :

$$J = J^i + J_{\xi}^{i'} (\xi_{t+1} - \mu_{t+1}^i) + \frac{1}{2} (\xi_{t+1} - \mu_{t+1}^i)' J_{\xi\xi}^i (\xi_{t+1} - \mu_{t+1}^i) \quad (10)$$

where:

$$J^i = J|_{\xi=\mu_{t+1}^i} \quad (11)$$

$$J_{\xi}^i = \nabla_{\xi} J|_{\xi=\mu_{t+1}^i} \quad (12)$$

$$J_{\xi\xi}^i = \nabla_{\xi} \nabla_{\xi}' J|_{\xi=\mu_{t+1}^i} \quad (13)$$

are the gradient and Hessian matrix of J with respect to ξ_{t+1} .

Setting the Equation (10) with respect to ξ_{t+1} to zero yields the next value of μ_{t+1} in the iteration:

$$\mu_{t+1}^{i+1} = \mu_{t+1}^i - (J_{\xi\xi}^i)^{-1} J_{\xi}^i \quad (14)$$

Then, the following three equations are iteratively calculated:

$$H_{t+1}^i \triangleq h_{\mu}[t+1, \mu^i(t+1|t+1)] \quad (15)$$

$$\begin{aligned}\mu^{i+1}(t+1|t+1) = & \mu^i(t+1|t+1) + P^i(t+1|t+1)H_{t+1}^{i'} \\ & \cdot R(t+1)^{-1}\{z(t+1) - h[t+1, \mu^i(t+1|t+1)]\} \\ & - P^i(t+1|t+1)P(t+1|t)^{-1} \\ & \cdot [\mu^i(t+1|t+1) - \mu(t+1|t)]\end{aligned}\quad (16)$$

$$\begin{aligned}P^i(t+1|t+1) = & P(t+1|t) - P(t+1|t)H_{t+1}^{i'} \\ & \cdot [H_{t+1}^i P(t+1|t)H_{t+1}^{i'} + R(t+1)]^{-1} \\ & \cdot H_{t+1}^i P(t+1|t)\end{aligned}\quad (17)$$

Since we get the iterative form of state estimate μ_t , we can then calculate the information matrix and information vector iteratively:

$$H_t = H_t^i + C_t^{i+1}R_t^{-1}(C_t^{i+1})^T \quad (18)$$

$$b_t^{i+1} = b_t^i + (z_t - \hat{z}_t + (C_t^{i+1})^T \mu_t^{i+1})^T R_t^{-1} (C_t^{i+1})^T \quad (19)$$

where C_{t+1}^i is the Jacobian matrix of h at μ_t^{i+1} .

Figure 1c illustrates the relinearization of the nonlinear measurement function $h(\xi_t)$ for three iterations. As shown in the figure, each new estimate can be obtained by the true measurement z_t and the approximated function around the previous estimate. We can specify the number of iterations we want to perform in advance according to the designer's experience. It should be noticed that the iterative form of SEIF only relinearizes the measurement model.

In fact, the estimate uncertainty is small when the robot starts to move on, and it grows with time. Therefore, it is unnecessary to perform the iterations at each step. In order to decrease the computational cost of ISEIF, the iteration will be performed periodically. The periods are usually set as 100 or 500. However, the number of iterations at each time is still difficult to choose, *i.e.*, if the number is too big, the algorithm will become slow, and the ISEIF will lose its speed advantage; on the other hand, if the number is too small, the estimate error may still be large, and the result still tends to be inconsistent. We apply an adaptive stop criteria to the iteration process to balance the complexity and consistency. The iteration stop criteria is defined as follows:

$$\text{delta} = g(\mu_t^i - \mu_t^{i-1})$$

where g is a function of the difference between two iterations (usually the distance between μ_t^i and μ_t^{i-1}), and we will compare delta with a previous set threshold. If delta is bigger than the threshold, we will continue the iteration process, for we can still improve the estimate accuracy by iteration; and if delta is smaller than the threshold, the iteration process will be terminated. This adaptive stop criteria provides a good balance between complexity and consistency. The value of the threshold is usually determined by the designer's experience. If delta is set properly, the ISEIF algorithm will only perform two or three iterations most of the time.

Let μ_t^i denote the posterior estimate of state ξ_t after i iterations have been performed. Therefore, μ_t^0 refers to the predicted estimate that derives from the motion update step in standard SEIF. We use H_t^i and b_t^i to denote the intermediate information matrix and information vector at the i -th iteration respectively, and C_t^i to refer to the Jacobian of $h(\xi_t)$ with respect to μ_t^i .

With these notations, the adaptive ISEIF can then be summarized as follows:

1. The nonlinear motion and measurement model functions are given as follows:

$$x_t = f(x_{t-1}, u_t) + v_t \quad (20)$$

$$z_t = h(\xi_t) + \varepsilon_t \quad (21)$$

$$v_t \sim (0, Q_t)$$

$$\varepsilon_t \sim (0, Z_t)$$

2. Initialization as follows:

$$\mu_0 = E(\xi_0) \quad (22)$$

$$P_0 = E[(\xi_0 - \mu_0)(\xi_0 - \mu_0)^T] \quad (23)$$

$$H_0 = P_0^- \quad (24)$$

$$b_0 = H_0 * \mu_0 \quad (25)$$

3. For $t = 1, 2, \dots$, do the following:

- (a) Perform motion update equations of SEIF. The iterative SEIF is the same as the SEIF up to this point.
- (b) If t is not in the iterative periods, skip to d; else start to perform iterative steps as follows: Initialize the iterative SEIF to the standard SEIF and perform the iterative measurement update step.

$$H_t^0 = H_t \quad (26)$$

$$b_t^0 = b_t \quad (27)$$

Calculate the measurement update Equations (14) to (18).

- (c) Calculate the value of delta:

$$\text{delta} = g(\mu_t^i - \mu_t^{i-1}) \quad (28)$$

According to the adaptive stop criteria, if *delta* is smaller than the threshold, stop the iteration process and turn to d; otherwise, return to the b step to continue the iteration.

- (d) The final information matrix and information vector are given as follows:

$$H_t = H_t^{n+1} \quad (29)$$

$$b_t = b_t^{n+1} \quad (30)$$

where n is the number of iterations.

4. Perform the sparsification step of SEIF.

3. Experiments

Simulations, the Victoria Park dataset and our sea trial experiments are carried out to verify the algorithm. We use the RMSE error and normalized estimation error square (NEES) value to evaluate the accuracy and consistency and the elapsed time to evaluate the complexity. RMSE and NEES are two commonly-used metrics in the SLAM community, and they are calculated based on the ground truth data. It should be noted that relative error and other metrics are indeed very popular in the recent robotics community. Surely, they are very useful to evaluate a SLAM algorithm by itself.

However, this paper focuses more on engineering applications, not only the SLAM algorithm itself. In this paper, we try to improve the SEIF algorithm and, more importantly, implement the ISEIF algorithm for our AUV system to verify the navigation algorithm. It is hard to say that a SLAM algorithm with low relative error will have good navigation accuracy. In this situation, we care more about the navigation accuracy calculated by the navigation system in real time compared to ground truth data (e.g., GPS data). Therefore, we think it might be proper to compare the calculated trajectory with the ground truth data in this application.

3.1. Simulations

In this section, the simulated experiments are implemented to compare the performance of the methods described above. Fifty Monte Carlo simulations with both SEIF and ISEIF are conducted respectively under the same environments and the same basic parameters. The entire experiments are on the foundation of the geometric feature map. We use (x_t, y_t, φ_t) to denote the vehicle pose, in which (x_t, y_t) are the coordinates and φ_t stands for the orientation on the basis of the map environment. Given the true position of the vehicle and the coordinates of all landmarks, the expressions of the motion model are formulated as follows:

$$\begin{bmatrix} x_t \\ y_t \\ \varphi_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + v_t \times dt \times \cos(\varphi_{t-1} + G) \\ y_{t-1} + v_t \times dt \times \sin(\varphi_{t-1} + G) \\ \varphi_{t-1} + v_t \times dt \times \sin(G)/B \end{bmatrix} + W_t \quad (31)$$

where $u = [v_t, G]$ is the control input, in which v_t is the velocity and G is the steer input. In addition, B corresponds to the wheelbase of the vehicle, and W_t is Gaussian process noise. The observation model is:

$$\begin{bmatrix} r \\ b \end{bmatrix} = \begin{bmatrix} \sqrt{(m_x - x_t)^2 + (m_y - y_t)^2} \\ \tan^{-1}\left(\frac{m_y - y_t}{m_x - x_t}\right) - \varphi_t \end{bmatrix} + N_t \quad (32)$$

where (m_x, m_y) are the positions of landmarks and N_t is the observation noise. The observation model we use is the vector (r, b) , where r and b are the range and bearing between the landmark and the vehicle, respectively. In terms of the map, a circle trajectory composed of 20 points is designed as the waypoints, and 200 landmarks lie astride the waypoints, which form two circles, as well.

All experiments are carried out on the platform of MATLAB. The vehicle is initialized and starts at $(0, 0)$. The other essential parameters in the algorithm settings are listed at Table 1.

Table 1. Basic parameters set in the simulation.

Wheelbase of vehicle	4 m	Control noise	(0.3 m/s, 2°)
Speed	3 m/s	Observation noise	(0.2 m/s, 2°)
Maximum steering angle	30°	Control frequency	40 Hz
Maximum range	30 m	Observation frequency	5 Hz

The ultimate data we analyze come from the average result of 50 Monte Carlo simulations. We will demonstrate the superiority of ISEIF from the following aspects: RMS, NEES and time. Figure 2 presents the estimate results of vehicle's pose and landmarks of SEIF and ISEIF, where we can see that ISEIF has a more accurate estimate of the vehicle's position.

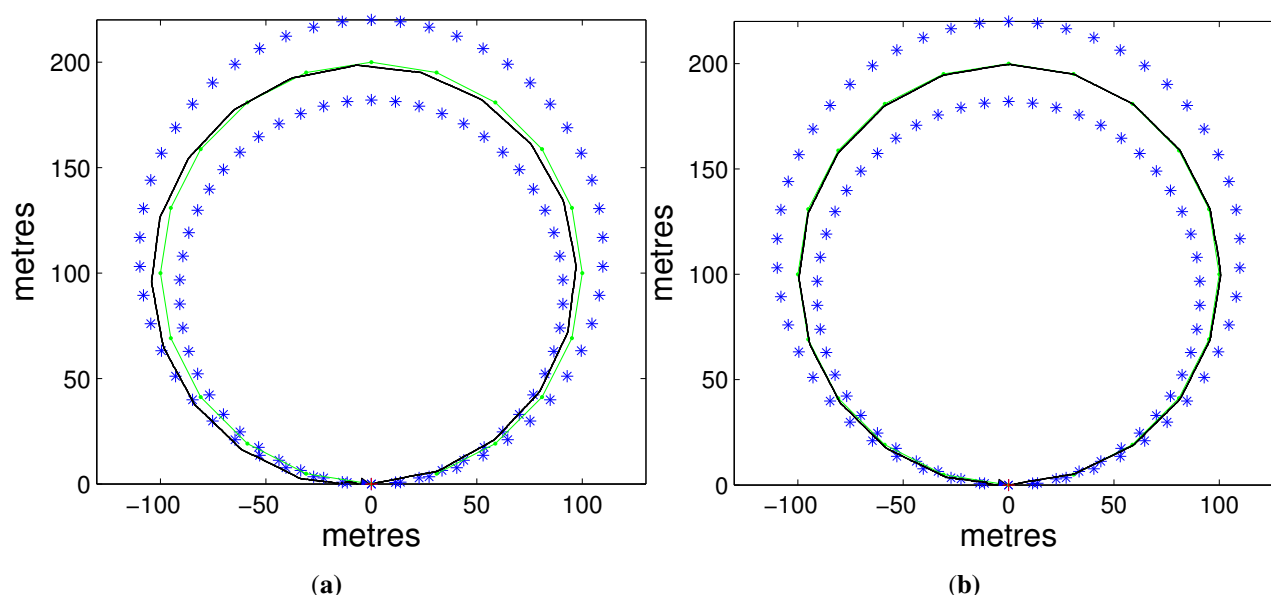


Figure 2. The results of trajectory and landmarks estimated by SEIF (a) and iterative SEIF (ISEIF) (b).

Figure 3 is a magnified local map that indicates the detailed relations between the estimates and the true position. The ellipses reflect the estimated covariances of the corresponding methods, which denote the estimate uncertainty. As the results illustrate, the estimate result of ISEIF is closer to the true landmarks in the map; and what is more important is that the estimate uncertainty in SEIF is smaller than the real error, *i.e.*, “overconfidence”, which is the main cause of inconsistency. In contrast, ISEIF performs a consistent estimate most of the time.

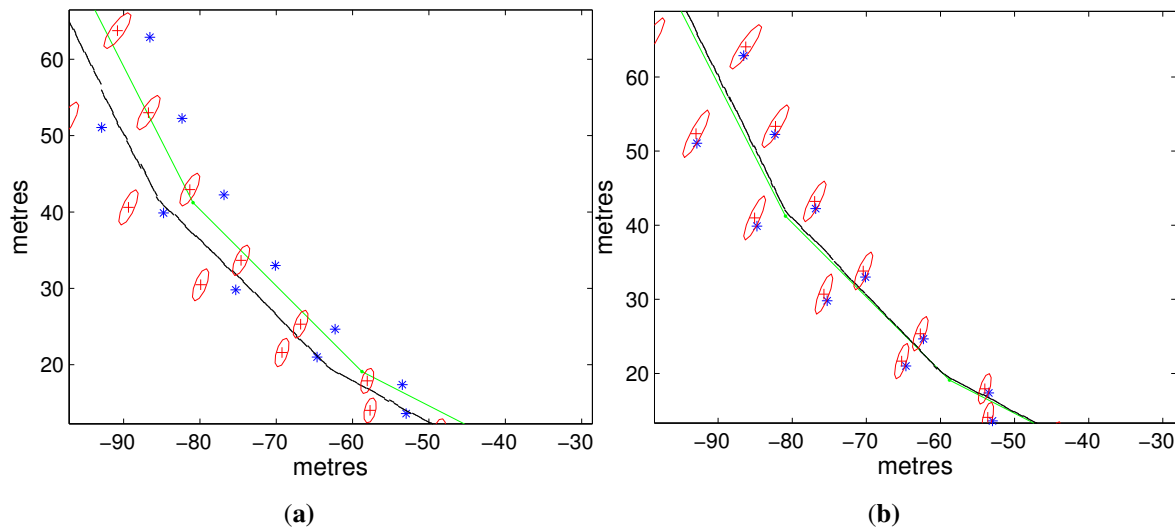


Figure 3. The enlarged portions of the local map, in which (a) is for SEIF and (b) is for ISEIF.

We use two different methods to compare the performance in different perspectives, namely we use root mean square (RMS) and normalized estimation error square (NEES) for the comparison of accuracy and consistency, respectively. They are defined as follows.

The RMS is defined as:

$$e = \sqrt{\frac{\sum_{i=1}^{n_x} (x_i - \hat{x}_i)^2}{n_x}} \quad (33)$$

The NEES is defined as:

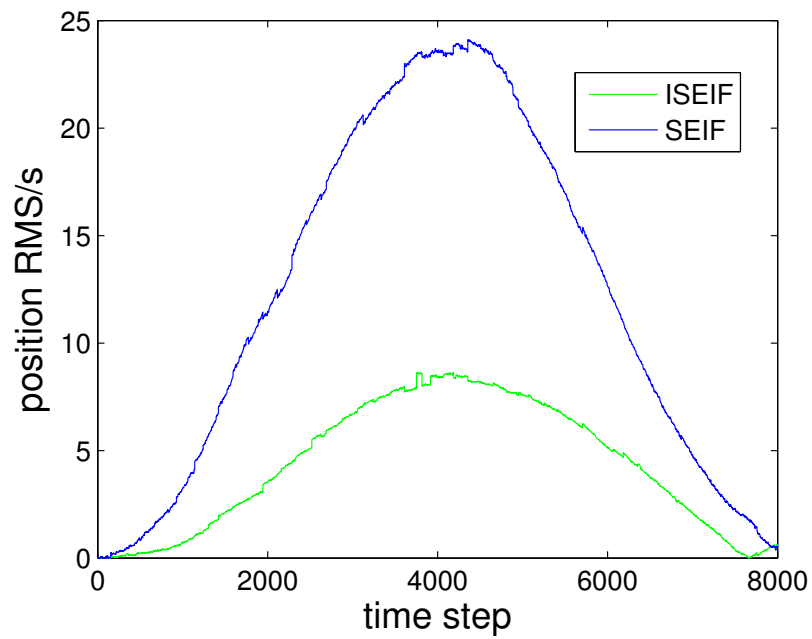
$$\varepsilon_i = (x_i - \hat{x}_i)^T P_i^{-1} (x_i - \hat{x}_i) \quad (34)$$

where x_i is the ground truth for the state vector, while \hat{x}_i and P are the estimated state and covariance matrix. Furthermore, the estimated positions of landmarks are also evaluated with RMS, to be shown in Table 2 later. In NEES, ε_i is chi-square distributed with n_x degrees of freedom, where n_x represents the dimension of x . To evaluate the consistency, the average NEES is necessary for N Monte Carlo trials:

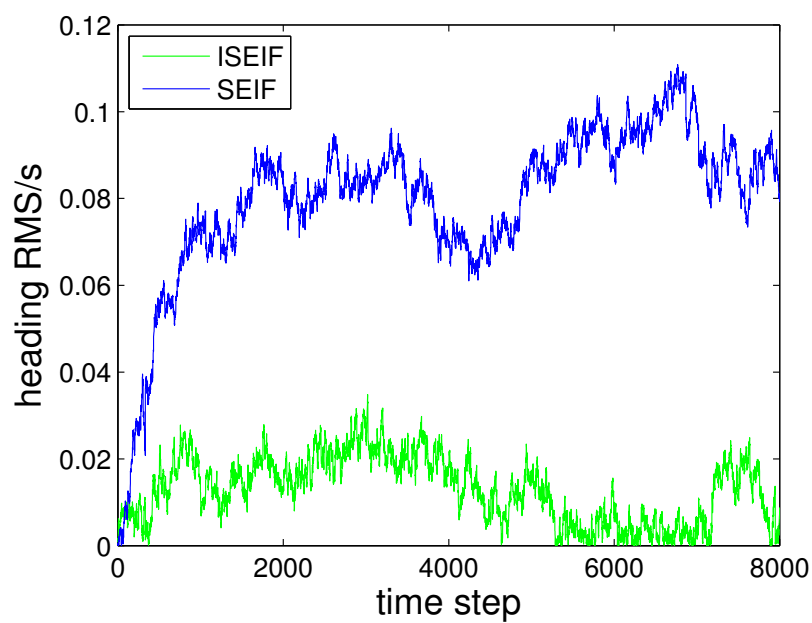
$$\bar{\varepsilon} = \frac{1}{N} \sum_{i=1}^N \varepsilon_i \quad (35)$$

Figure 4 respectively compares the RMS errors of vehicle positions and the headings of SEIF and ISEIF. Apparently, the trend of errors of ISEIF goes steadily and smaller than that of SEIF, including both vehicle position errors and heading errors. All of the errors decrease by about 80 percent under the action of ISEIF. Besides the superiority of the algorithm itself, the simplicity of simulation data may be another reason that produces so much improvement in the accuracy.

Here, some detailed results calculated by RMS are listed in Table 2. They are average values over 50 independent simulations, only varying by the different random noises that we add. As the list shows, the errors of the parameters that we obtain all decrease by more than 20% after being optimized by ISEIF. Thus, we can draw the conclusion that ISEIF can be much more effective for improving accuracy.



(a)



(b)

Figure 4. The RMS errors of positions and headings of SEIF (a) and ISEIF (b).

Table 2. Average errors over 50 simulations.

	Pose-x Error	Pose-y Error	Heading Error	Landmark-x Error	Landmark-y Error
SEIF	19.9203 m	9.1883 m	0.0052°	18.9011 m	9.9527 m
ISEIF	15.8803 m	7.6626 m	0.0031°	14.1157 m	7.6663 m

Figure 5 illustrates the average NEES values, where Figure 5a is for SEIF and Figure 5b is for ISEIF. In our simulations, the dimension of the vehicle pose is three, and there are 50 Monte Carlo trials performed; thus, $N = 50$. The 95% probability concentration region for $\bar{\epsilon}$ confines the horizontal threshold at $[2.36, 3.72]$, produced by the average chi-square distribution [32], as the red lines plotted in the figure. It can be seen that the NEES value of ISEIF keeps close to the bound all of the time, while the result of SEIF goes far away quickly, which persuasively indicates that ISEIF keeps a more consistent estimate.

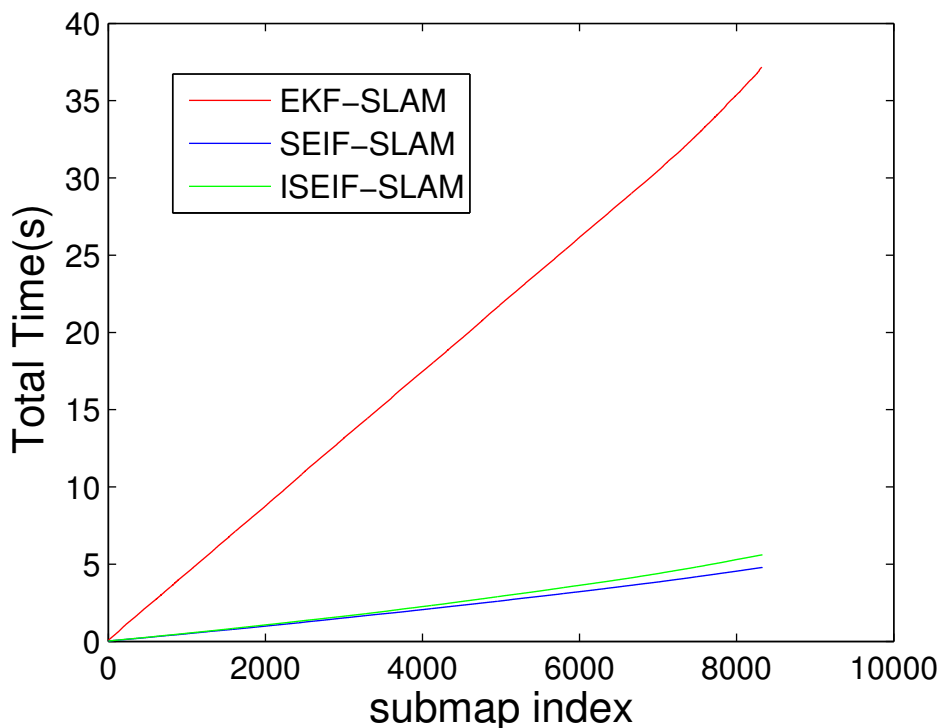


Figure 5. The comparison of average CPU time for EKF, SEIF and ISEIF.

For the purpose of demonstrating the speed advantage of ISEIF, the total CPU time is presented. The comparison of the average time of 50 Monte Carlo experiments between EKF, SEIF and ISEIF is illustrated in Figure 6. According to the results, the total execution times of EKF, SEIF and ISEIF are 37.1720 s, 4.7845 s and 5.5987 s, respectively. Thus, we can see that compared to EKF, SEIF and ISEIF are dramatically time saving. As for ISEIF, it only consumes a little more time than SEIF, but improves the estimate accuracy and consistency.

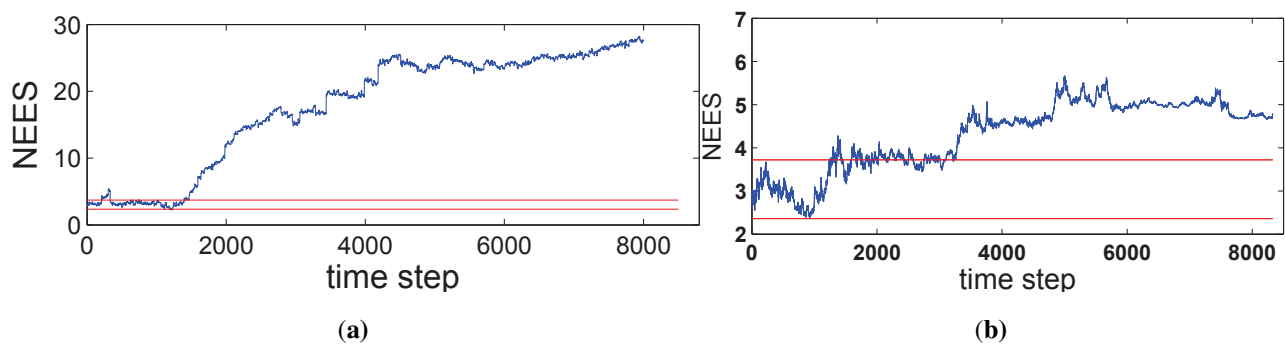


Figure 6. Average normalized estimation error square (NEES) of SEIF (a) and ISEIF (b).

3.2. Victoria Park Dataset

Different from simulations, real-world applications often involve a variety of uncertainty factors, such as non-Gaussian noise and nonlinear motion and measurement models. To validate the ISEIF and compare the performances of it with the SEIF, we apply them to the benchmark Victoria Park dataset, which is commonly used to test the feasibility of different algorithms in the SLAM community. The environment is approximately 250 m east to west and 300 m north to south.

In the experiment [33], a sensor-enabled car, equipped with odometry sensors and a laser range finder, was driven autonomously in a series of irregular routes within Victoria Park, Sydney.

3.2.1. Sensors

The car used for the experiments is equipped with a SICK (as spelled by a producer of sensors from Germany) laser range and bearing sensor, a linear variable differential transformer sensor for the steering mechanism and a back wheel velocity encoder. The inertial measurement unit consists of three accelerometers, three gyros and two inclinometers, as well as external sensors, including GPS, laser and vision. Here are some pivotal sensors used in the experiment.

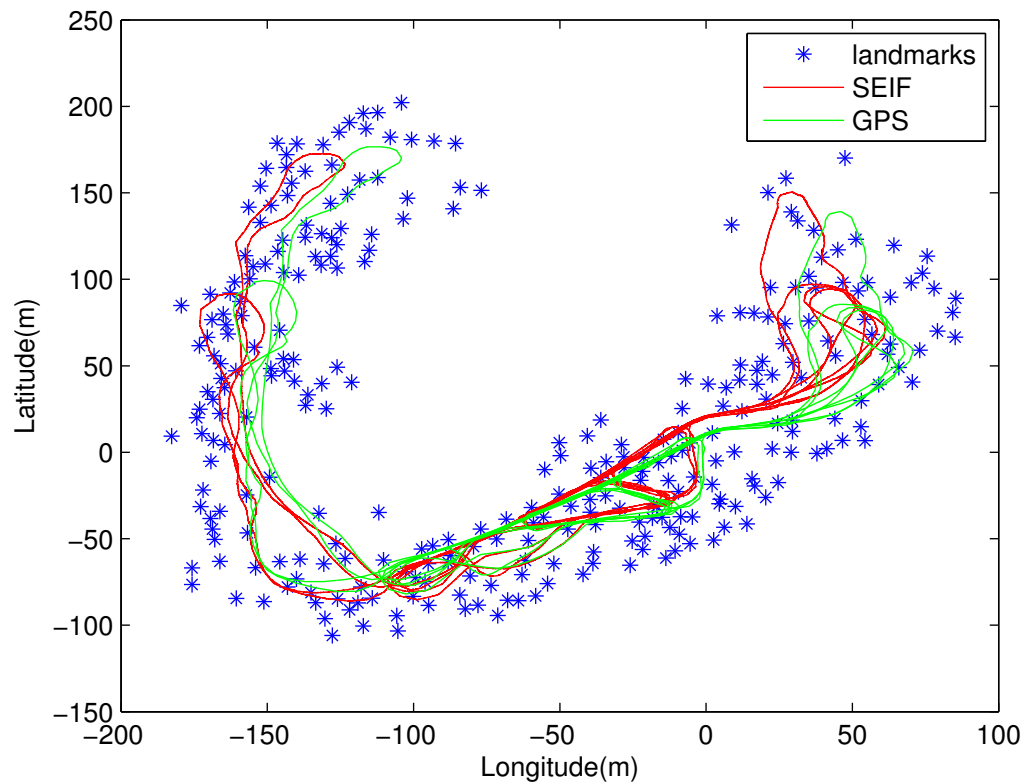
GPS provides the data that describe the ground truth, which are comprised of the time stamp, latitude and longitude in meters with respect to the initial point.

The SICK laser scanner provides 360 range points at 0.5-degree intervals. More specifically, it uses 1° to 360° to represent a range at 0° to 180° with an accuracy of 0.5° .

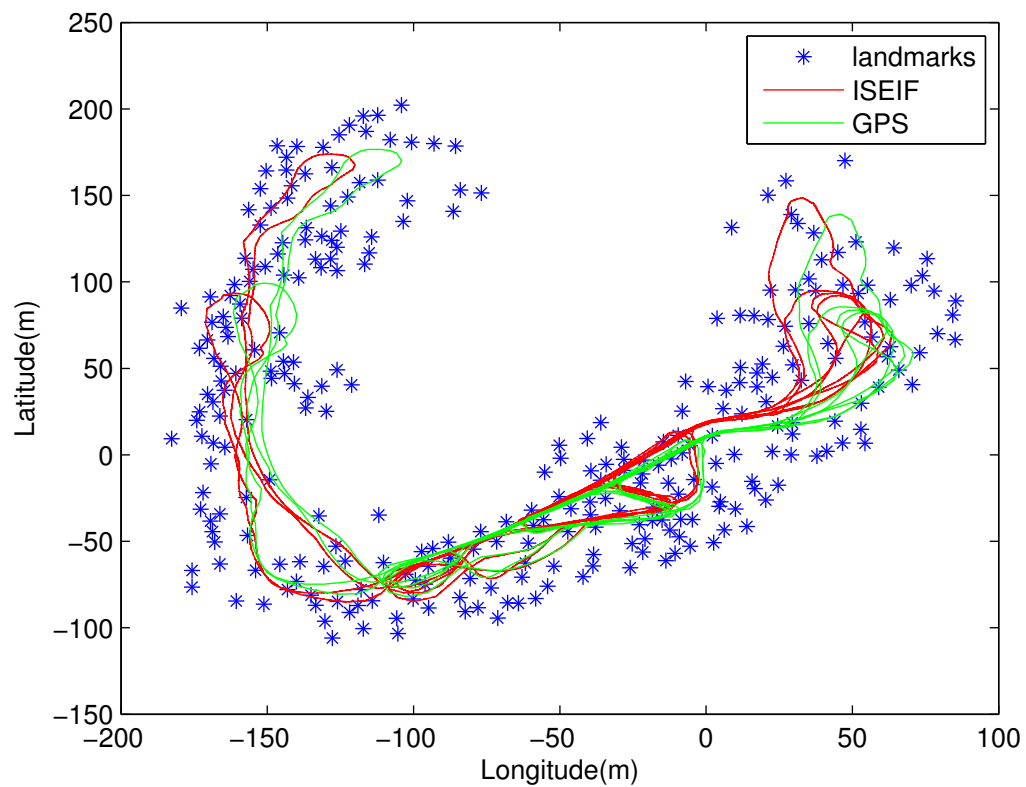
3.2.2. Results

The final estimated maps and trajectories of both SEIF and ISEIF are presented in Figure 7a,b. It is obvious that the trajectory estimated by ISEIF is much closer to the GPS, which validates the feasibility of ISEIF.

Further analysis focuses on the accuracy and the consistency of the two algorithms, which are assessed by RMS and NEES, respectively. The plot in Figure 8 displays the RMS errors of SEIF and ISEIF with overlap, which indicates that the ISEIF performs a more accurate estimate than SEIF. It can be seen that the estimated errors grow when the vehicle explores new regions and decrease when the vehicle returns to known areas. ISEIF is more accurate than SEIF at all times.

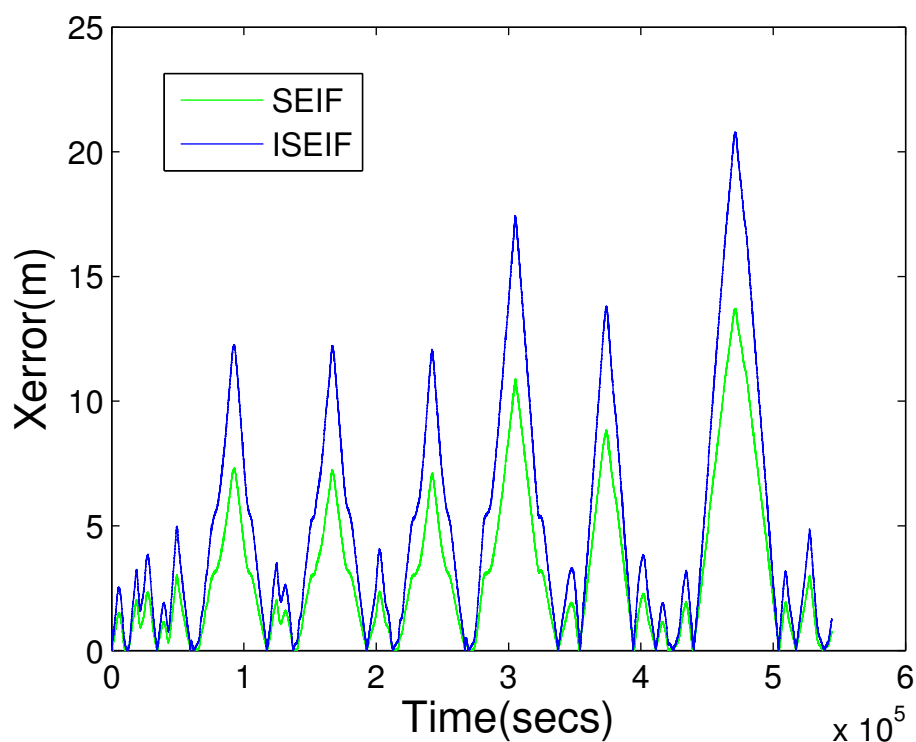


(a)

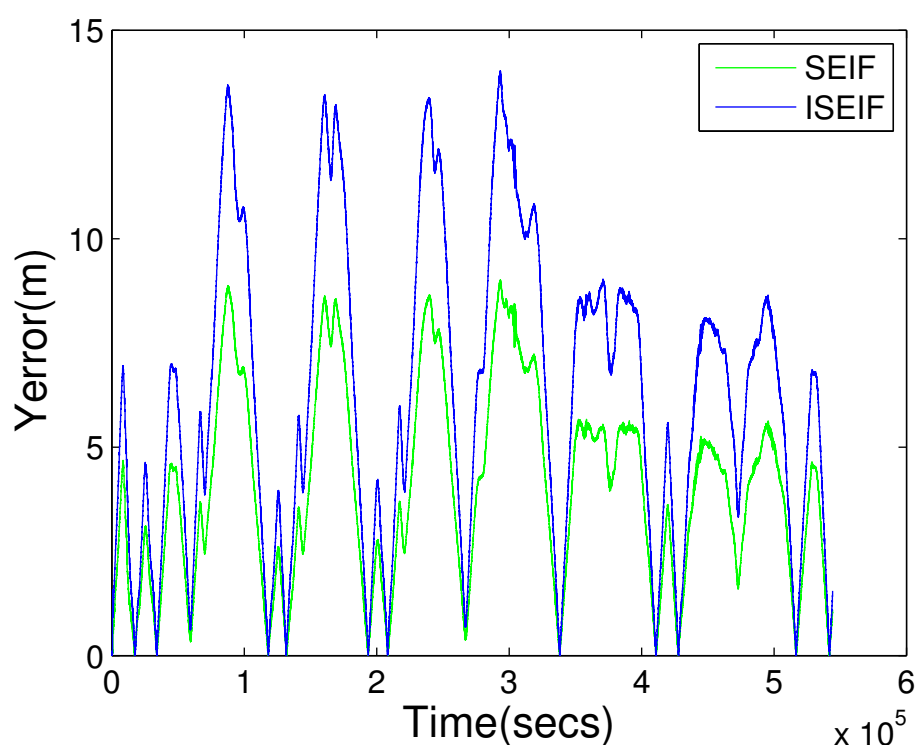


(b)

Figure 7. Estimates of the vehicle trajectories and landmarks for the Victoria Park dataset generated by the SEIF (a) and the ISEIF (b).



(a)



(b)

Figure 8. The RMS error comparison between SEIF and ISEIF relative to GPS, where (a) is the error in horizontal coordinate and (b) is in vertical coordinate.

Compared to SEIF, the consistency performance of ISEIF also improves greatly, which can be reflected by NEES in Figure 9. The NEES value of ISEIF is mostly bounded in the threshold interval, while the NEES value of SEIF soon goes beyond the upper threshold.

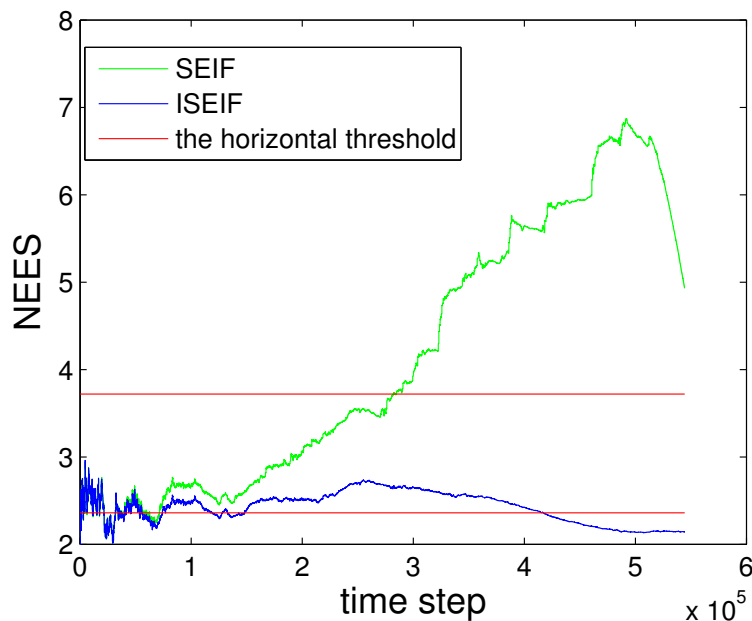


Figure 9. Plots of the NEES of both SEIF and ISEIF for the Victoria Park dataset.

Moreover, the computational complexity also should be taken into account. Just as the plot shown in Figure 10, the total time required for the time prediction and measurement update steps for ISEIF is slightly longer, but not a large gap, compared to SEIF. For the implementation of ISEIF, it is inevitable to spend a little time due to the iterative step. However, ISEIF still keeps the speed advantage over EKF, while preserving a more accurate and consistent estimate.

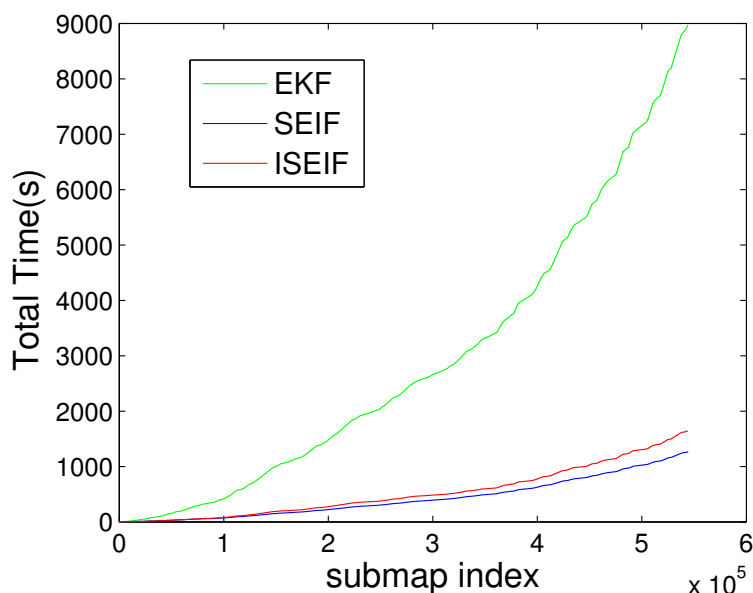


Figure 10. CPU time elapsed for the Victoria Park dataset generated by EKF, SEIF and ISEIF.

Table 3 quantitatively compares the results of SEIF and ISEIF. The results showed that ISEIF achieves about 40% and 10% improvement in pose-x and pose-y accuracy. In terms of consistency, 82.34% of the NEES value of ISEIF is bounded in the confidence interval, while SEIF is only 52.10%. Although ISEIF

is about 20% slower than SEIF, it is much faster than EKF in an order. Based on these results, we can draw the conclusion that ISEIF obtains a more consistent and accurate state estimate in a fast speed.

Table 3. Comparisons of the performances of SEIF and ISEIF for the Victoria Park dataset.

Algorithm	Pose-x Error	Pose-y Error	NEES	CPU Elapsed Time
SEIF	7.5284 m	5.3781 m	51.10%	1374.28 s
ISEIF	4.7133 m	4.8325 m	82.34%	1658.31 s

3.3. Sea Trials in Tuandao Bay

To verify the feasibility of SEIF more persuasively, we carried out sea trials by an AUV with both SEIF and ISEIF implemented. We will first introduce our experiment platform and analyze the results afterwards.

3.3.1. AUV C-Ranger

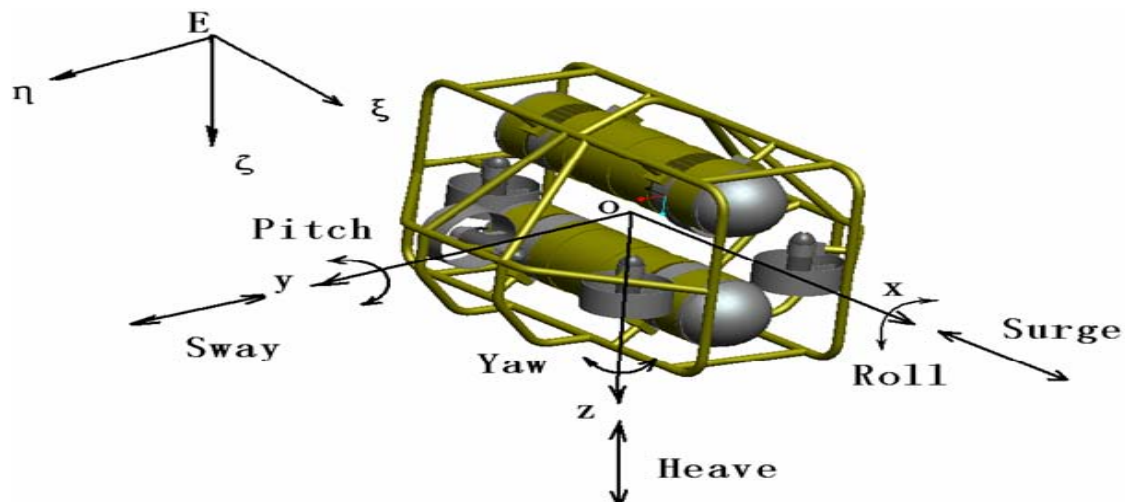
In the following experiments, a C-Ranger AUV loaded with a series of sensors was designed to navigate in underwater circumstances. The C-Ranger is an open-frame AUV, shown in Figure 11, and its general parameters are listed in Table 4. The AUV system is comprised of two electronic cabins and five underwater propeller thrusters. Figure 12 illustrates the coordinate system of the C-Ranger platform. Five degrees of freedom (DOF), including yaw, pitch, roll, heave and surge, contribute to the good maneuverability of the AUV. The thrust system consists of five propeller thrusters. Two of the thrusters paralleling to the bow direction installed in the abdomen are in charge of horizontal thrust related to the surge and yaw, and the other three are vertical thrusters for heave, roll and pitch. The upper hull of the C-Ranger is the instrument compartment containing sensors, the computer module, the communication module, the internal monitoring module and other devices. The lower hull is the power supply system mainly made up of lithium-ion batteries, the power management module, the motor-driver module, *etc.*



Figure 11. C-Ranger in deployment.

Table 4. Essential parameters of the C-Range autonomous underwater vehicle (AUV).

Parameter	Length	Width	Height	Tonnage	Weight	Maximal Speed	Endurance
Value	1.6 m	1.3 m	1.1 m	208 L	206 kg (full loaded)	3 knots (1.5 m/s)	8 h at 1 knot speed

**Figure 12.** The coordinate system of the C-Ranger platform.

3.3.2. On-Board Sensors

There are a lot of sensors installed on the C-Ranger, and they are generally divided into the internal group and the external group. Internal sensors are the digital compass, gyro, the attitude and heading reference system (AHRS) and the pressure sensor. External sensors include mechanical scanning sonar, the Doppler velocity log (DVL), altimeter, CCD camera and GPS. Among them, the mechanically scanning forward-looking sonar (Super Seaking DST, Tritech) for active sensing of environment features installed at the front top of C-Ranger is the principal sensor. It works at a frequency of 675 kHz and a range of up to 300 m.

3.3.3. AUV Motion Model

At time t , the control inputs $U^t = \{u_1, u_2 \cdots u_t\} = \{U^{t-1}, u_t\}$ and observations $Z^t = \{z_1, z_2 \cdots z_t\} = \{Z^{t-1}, z_t\}$ are available. The AUV motion model of the experiments is shown as follows:

$$X_{t+1} = \begin{pmatrix} x_{vt} \\ y_{vt} \\ \phi_{vt} \end{pmatrix} = \begin{pmatrix} x_{vt} + \Delta TV \cos(\theta + \theta_{vt}) \\ y_{vt} + \Delta TV \sin(\theta + \theta_{vt}) \\ \theta_{vt} + \frac{\Delta TV \sin \theta}{L} \end{pmatrix} + v_t \quad (36)$$

where the variables $L, V, \theta, \Delta T$ represent vehicle length, vehicle velocity, rotation angle and sampling interval, respectively. $v_t \sim N(0, Q)$ is white Gaussian noise. Correspondingly, the AUV observation model related to the range and bearing is:

$$z_t = \begin{pmatrix} z_r \\ z_\alpha \end{pmatrix} = \begin{pmatrix} \sqrt{(x_i - x_{vt})^2 + (y_i - y_{vt})^2} \\ \arctan\left(\frac{y_i - y_{vt}}{x_i - x_{vt}}\right) - \theta_{vt} \end{pmatrix} + \begin{pmatrix} w_{rt} \\ w_{\alpha t} \end{pmatrix} \quad (37)$$

$(x_i, y_i)^T$ is the global position of i -th feature and w_{rt} and $w_{\alpha t}$ are the observation Gaussian noise, namely $w_{rt} \sim N(0, R_r)$, $w_{\alpha t} \sim N(0, R_\alpha)$.

3.3.4. Sea Trial

The environment where we conducted our sea trials is as shown in Figure 13, which is a satellite map of Tuandao Bay in Qingdao, China, added with the GPS trajectory along which the vehicle moves (the red line). The trajectory of the GPS is considered as the ground truth as in the simulation. In the experiment, the vehicle moves at a speed of about one knot; the imaging sonar is set to a 120-degree field of vision; and the scanning range is set to 100 m.



Figure 13. The satellite map of Tuandao Bay and the trajectory of C-Ranger by GPS.

The comparison result of the trajectories is illustrated in Figure 14, where the red line represents the GPS trajectory, the light blue is for dead reckoning and the dark blue is for the SLAM. Here, Figure 14a,b shows the result of SEIF and ISEIF, respectively. As the figures indicate, the trajectory estimated by ISEIF fits the GPS trajectory better than that of SEIF. Owing to the great amount of sonar data, there exist some redundant features inevitably. Thus, raw data measured by sonar are preprocessed to reduce the computation, and the feature extraction algorithm is the same as in [29].

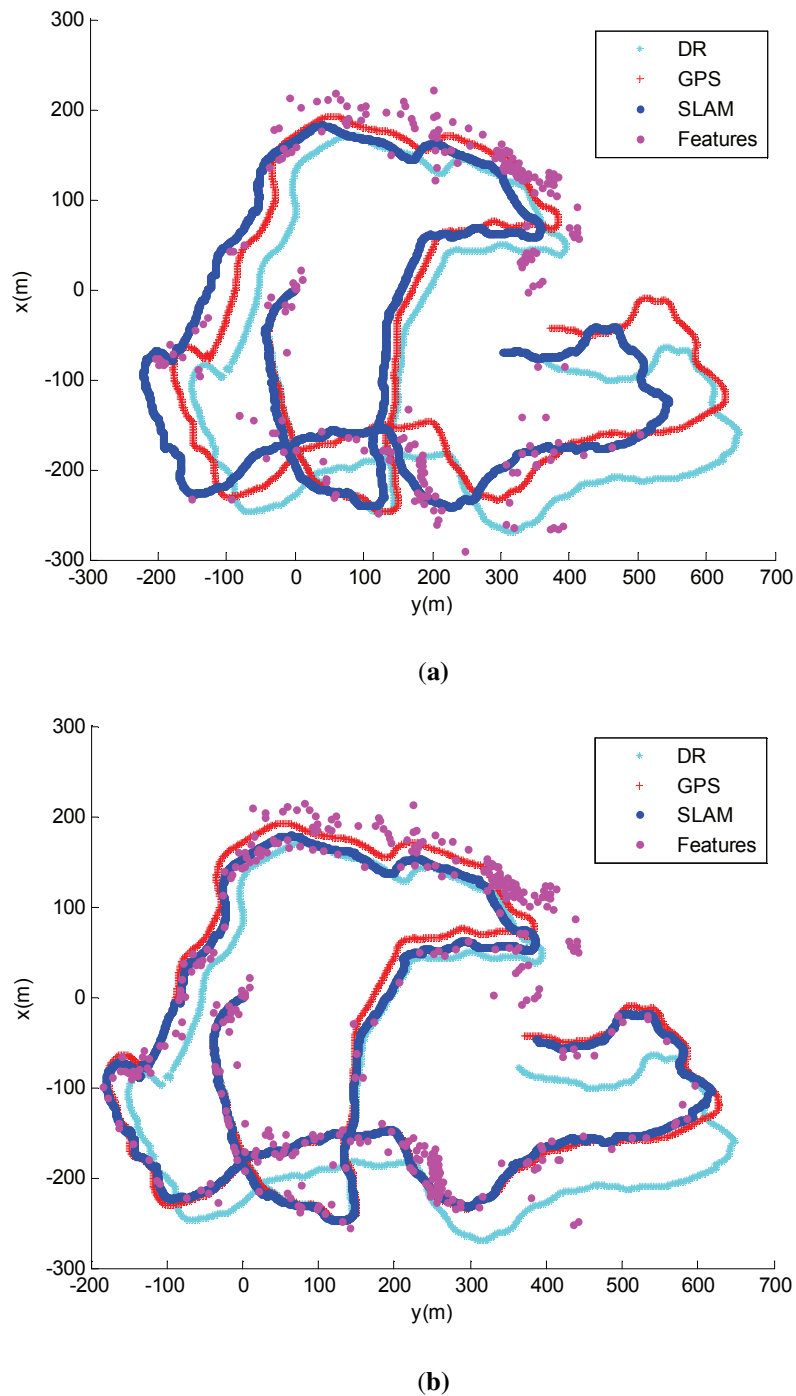


Figure 14. Comparisons of the results estimated by SEIF (a) and ISEIF (b) in the experiments of the sea trial.

Furthermore, the RMS error is used to compare the accuracy of the two algorithms quantitatively. Figure 15a,b presents the RMS errors of horizontal and vertical coordinates, respectively. The figures show that the RMS errors of ISEIF have a similar trend to SEIF, but much smaller than SEIF in general, both in horizontal and vertical coordinates. As is shown in Figures 8 and 15, the decrease of the RMS errors of real experiments is much smaller than that of simulations. This can be explained by the inevitable performance reduction of ISEIF during processing complex and large datasets. Despite this, all experiments still validate the effectiveness of ISEIF on improving navigation accuracy. Thus, we can draw the conclusion that ISEIF performs more accurate estimates in real-world environments.

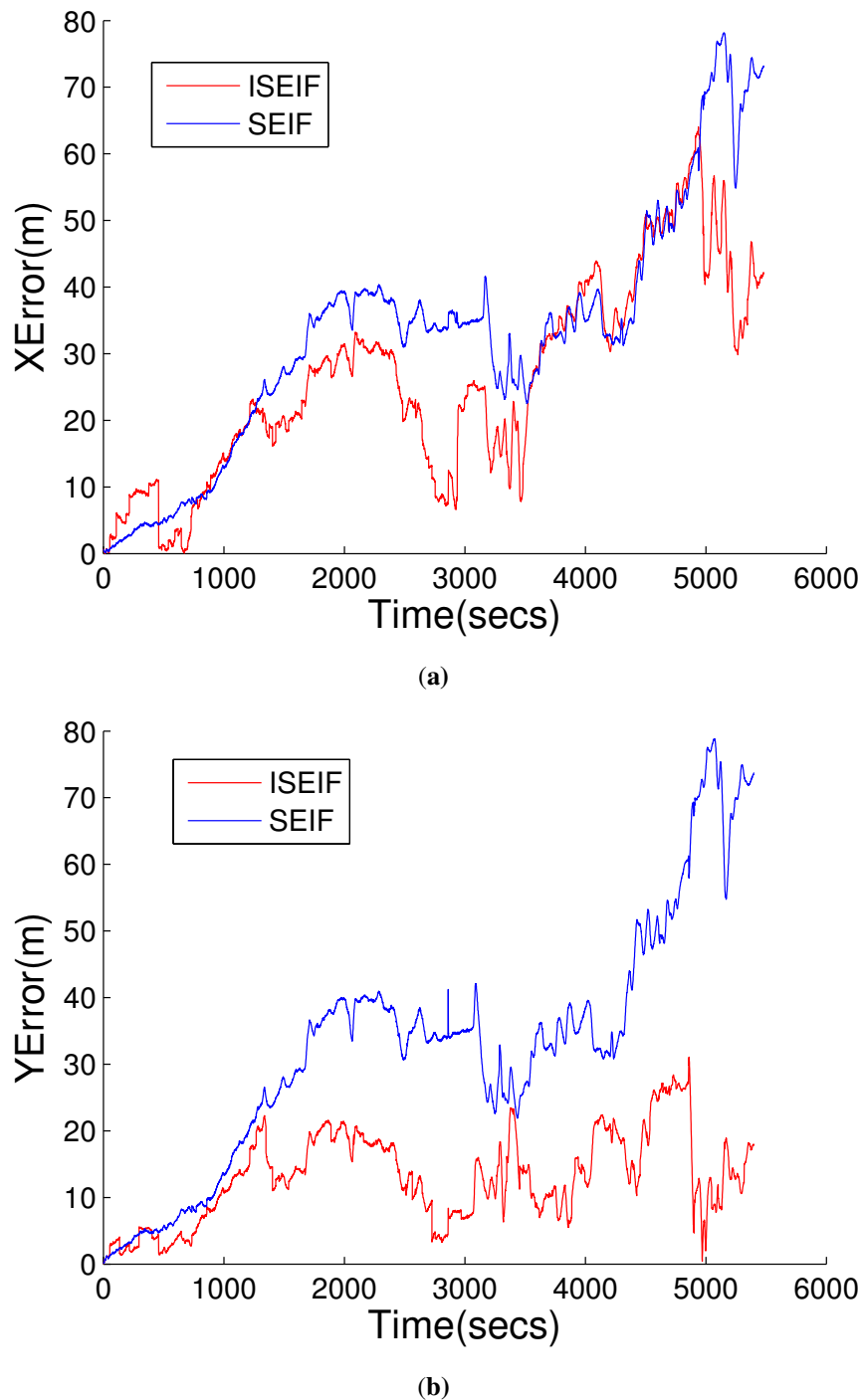


Figure 15. The RMS error comparison between SEIF and ISEIF relative to GPS, where (a) is the error in horizontal coordinate and (b) is in vertical coordinate.

Compared to SEIF, ISEIF also shows superior performance in terms of consistency, which can be reflected apparently by NEES in Figure 16. The NEES value of ISEIF is mostly bounded in the threshold interval, while the NEES value of SEIF soon goes beyond the upper threshold. According to Figures 4, 9 and 16, NEES of ISEIF can basically keep close to the threshold and maintain a relatively lower value steadily, while the corresponding NEES of SEIF tends to deviate far away from the threshold. In summary, ISEIF can effectively optimize consistency.

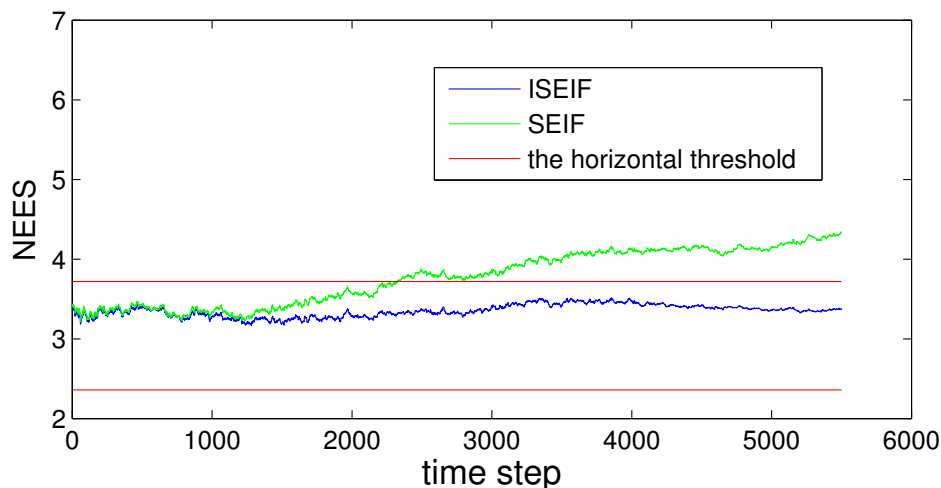


Figure 16. The comparison of NEES for SEIF and ISEIF in the sea trial.

The elapsed CPU times of EKF, SEIF and ISEIF were also recorded, and Figure 17 shows the results. It is very similar to the simulations we have done before, *i.e.*, ISEIF also shows a speed advantage in real-world applications. Again for the reasons of complex data and an unstable environment, many divergences emerge. This leads to the increase of iterative operations, which makes the running time of the sea trial extended for more than tens of seconds. Even so, the total time can be still considered acceptable and fairly fast for ISEIF compared to traditional EKF. For such a large dataset, this extra time is worth spending to obtain more consistent results. In conclusion, the experimental results confirm the excellent performance of ISEIF in both accuracy and efficiency.

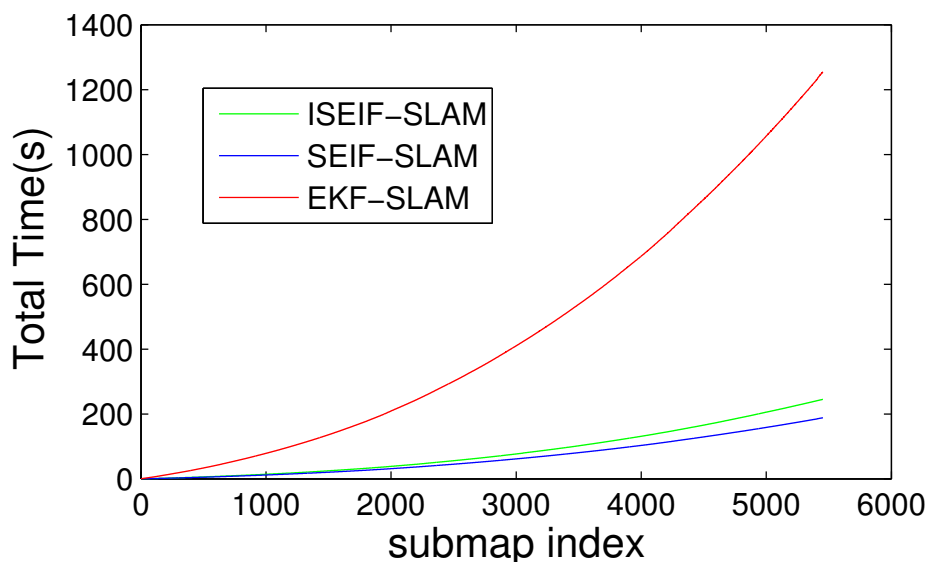


Figure 17. The comparison of CPU time for EKF, SEIF and ISEIF in the sea trial.

Table 5 quantitatively shows the results of SEIF and ISEIF in the sea trials. From the results, we can see that ISEIF achieves about 25% and 70% improvement in pose-x and pose-y error, respectively. In terms of consistency, 100% of the NEES value of ISEIF is bounded in the confidence interval, while SEIF is only 48.23%. Although ISEIF is about 35% slower than SEIF, it is still much faster than EKF. Then, we can draw a similar conclusion as in simulations and the Victoria Park dataset. It should be noted that

as the simulated experiments are much simpler than real-world environments, it is almost a sure thing that our new method performs best in simple simulations. In fact, real-world environments are always very complex, and many factors might influence the results (e.g., the quality of the data). Therefore, it is more difficult to handle real-world applications. However, ISEIF still shows better consistency and accuracy compared to SEIF and EKF with a fast enough speed. In our future work, we will continue to optimize our implementation of the algorithms and the AUV platform.

Table 5. Comparisons of performances of SEIF and ISEIF for the sea trial.

Algorithm	Pose-x Error	Pose-y Error	NEES	CPU Elapsed Time
SEIF	36.4628 m	40.7963 m	48.23%	285.23 s
ISEIF	27.7952 m	12.1165 m	100%	199.58 s

4. Discussion

Speed and consistency are the two main issues in the SLAM research community [34–36]. This paper focuses on the nonlinearity of the SLAM problem to improve the consistency. The proposed algorithm can reduce the linearized error by iteration and then get a more conservative estimate. It can also keep a fast speed at the same time.

Actually, both EKF and SEIF can be classified into one kind of method, *i.e.*, parametric filters, which belong to the online SLAM family. They are all based on the same probabilistic framework:

$$\arg \max_{s_t, m} P(s_t, m | z_{1:t}, u_{1:t}) \quad (38)$$

which can be thought of as calculating the most likely current pose s_t and the map m based on the controls and the measurements.

The fundamental reason for the inconsistency problem arising from the parametric filter may be that it only maintains the current pose, not the entire trajectory. Without the entire trajectory, the filter can only use new information to correct the current pose, not all of the previous pose. Thus the accumulated error cannot be revoked. In addition, the approximation of the nonlinearity in the SLAM problem is another reason for inconsistency. Actually, this paper only addresses the second part listed above, while how to reduce accumulated error was not discussed. Future work should be focused on this problem.

At the current time, full SLAM, also known as smoothers, shows its popularity in SLAM research community [37–42]. It is based on the following probabilistic formulation:

$$\arg \max_{s_t, m} P(s_t, m | z_{1:t}, u_{1:t}) \quad (39)$$

The subtle difference of full SLAM is that it estimates the entire trajectory of the robot, not only the current pose. Although full SLAM shows its popularity in the SLAM research community, there still exist some bottlenecks which limit its application. Among them, the most important one is that full SLAM algorithms are always more complex than parametric filters, thus implementing it in practical systems is difficult. That is why we still improve the SEIF algorithm, which is very easy to implement

in real-world applications. In the future, we will also try to implement the full SLAM algorithms in engineering applications.

5. Conclusions

In this paper, the iterative sparse extended information filter (ISEIF) has been proposed to improve the consistency of SEIF. As a dual form of the EKF, the extended information filter is subject to the same consistency problem. SEIF tends to be more overconfident than EIF due to the sparsification and state recovery step. Hence, the iterative method was adopted to get more conservative estimates. Mathematical derivation of ISEIF was given in this paper, and the effect of iteration during the update process was illustrated. To demonstrate the performance of ISEIF, Monte Carlo simulations were conducted, and both RMS and NEES errors were presented. The results convincingly show that ISEIF performs a more conservative estimate than SEIF. The time costs of EKF, SEIF and ISEIF were also compared. The results indicate that although ISEIF is a little slower than SEIF, it still keeps the scalability advantage over EKF, making it applicable in large environments. Real-world environments are very different from simulated experiments. Therefore, we also verified the validity of ISEIF in both outdoor and underwater navigation. The experimental platform AGV and the C-Ranger AUV were presented first, and experiments were conducted in Victoria Park and Tuandao Bay, respectively. Results empirically show that ISEIF is both consistent and scalable.

Acknowledgments

This work is partially supported by the Natural Science Foundation of China (41176076, 31202036, 51379198, 51075377) and the High Technology Research and Development Program of China (2014AA093400, 2014AA093410).

Author Contributions

Bo He and Yang Liu conceived the idea and designed the experiments and wrote the paper. Yue Shen and Tianhong Yan contributed the resources to the experiments, and performed the experiments and analyzed data. Diya Dong and Rui Nian performed the simulations.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Paull, L.; Saeedi, S.; Seto, M.; Li, H. AUV Navigation and Localization: A Review. *IEEE J. Ocean. Eng.* **2014**, *39*, 131–149.
2. Kharchenko, V.; Kondratyuk, V.; Ilnytska, S.; Kutsenko, O.; Larin, V. Urgent problems of UAV navigation system development and practical implementation. In Proceedings of the 2013 IEEE 2nd International Conference on Actual Problems of Unmanned Air Vehicles Developments Proceedings (APUAVD), Kiev, Ukraine, 15–17 October 2013; pp. 157–160.

3. Li, Q.; Chen, L.; Li, M.; Shaw, S.L.; Nuchter, A. A Sensor-Fusion Drivable-Region and Lane-Detection System for Autonomous Vehicle Navigation in Challenging Road Scenarios. *IEEE Trans. Veh. Technol.* **2014**, *63*, 540–555.
4. Smith, R.; Self, M.; Cheeseman, P. Estimating Uncertain Spatial Relationships in Robotics. In *Autonomous Robot Vehicles*; Springer-Verlag New York, Inc.: New York, NY, USA, 1990; pp. 167–193.
5. Matsebe, O. *EKF-SLAM Based Navigation System for an AUV*; CSIR: Pretoria, South Africa, 2008.
6. Vazquez-Martin, R.; Nuñez, P.; del Toro, J.; Bandera, A.; Sandoval, F. Adaptive observation covariance for EKF-SLAM in indoor environments using laser data. In Proceedings of the Electrotechnical Conference 2006 (MELECON 2006), Malaga, Spain, 16–19 May 2006; pp. 445–448.
7. Thrun, S.; Burgard, W.; Fox, D. *Probabilistic Robotics*; MIT Press: Cambridge, MA, USA, 2005.
8. Maybeck, P.S. *Stochastic Models, Estimation, and Control*; Academic Press: Waltham, MA, USA, 1982; Volume 3.
9. 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.* **2004**, *23*, 693–716.
10. Frese, U.; Hirzinger, G. Simultaneous localization and mapping—A discussion. In Proceedings of the IJCAI Workshop on Reasoning with Uncertainty in Robotics, Seattle, WA, USA, 4–10 August 2001; pp. 17–26.
11. Durrant-Whyte, H.; Bailey, T. Simultaneous localization and mapping: Part I. *IEEE Robot. Autom. Mag.* **2006**, *13*, 99–110.
12. Bailey, T.; Durrant-Whyte, H. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robot. Autom. Mag.* **2006**, *13*, 108–117.
13. Montemerlo, M.; Thrun, S.; Koller, D.; Wegbreit, B. *FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem*; AAAI/IAAI: Edmonton, AB, Canada, 2002; pp. 593–598.
14. Montemerlo, M.; Thrun, S. *FastSLAM 2.0. FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 63–90.
15. Bailey, T.; Nieto, J.; Guivant, J.; Stevens, M.; Nebot, E. Consistency of the EKF-SLAM algorithm. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; IEEE: 2006; pp. 3562–3568.
16. Liu, Y.; Thrun, S. *Results for Outdoor-SLAM Using Sparse Extended Information Filters*; ICRA: Gurgaon, India, 2003; pp. 1227–1233.
17. Thrun, S.; Liu, Y. Multi-robot SLAM with sparse extended information filters. In *Robotics Research*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 254–266.
18. Eustice, R.; Walter, M.; Leonard, J. Sparse extended information filters: Insights into sparsification. In Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems 2005 (IROS 2005), Edmonton, AB, Canada, 2–6 August 2005; pp. 3281–3288.

19. Walter, M.R.; Eustice, R.M.; Leonard, J.J. Exactly sparse extended information filters for feature-based SLAM. *Int. J. Robot. Res.* **2007**, *26*, 335–359.
20. Walter, M.; Hover, F.; Leonard, J. SLAM for ship hull inspection using exactly sparse extended information filters. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2008), Pasadena, CA, USA, 19–23 May 2008; pp. 1463–1470.
21. Gutmann, J.S.; Eade, E.; Fong, P.; Munich, M.E. A Constant-Time Algorithm for Vector Field SLAM using an Exactly Sparse Extended Information Filter. In *Robotics: Science and Systems*; The MIT Press: Cambridge, MA, USA, 2010.
22. Eustice, R.M.; Singh, H.; Leonard, J.J. Exactly sparse delayed-state filters. In Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA 2005), Barcelona, Spain, 18–22 April 2005; pp. 2417–2424.
23. Simon, D. *Optimal State Estimation: Kalman, H infinity, and Nonlinear Approaches*; John Wiley & Sons: Hoboken, NJ, USA, 2006.
24. Shojaei, K.; Shahri, A.M. Experimental study of iterated Kalman filters for simultaneous localization and mapping of autonomous mobile robots. *J. Intell. Robot. Syst.* **2011**, *63*, 575–594.
25. Zhou, W.; Zhao, C.; Guo, J. The study of improving Kalman filters family for nonlinear SLAM. *J. Intell. Robot. Syst.* **2009**, *56*, 543–564.
26. Lefebvre, T.; Bruyninckx, H.; De Schutter, J. Kalman filters for non-linear systems: A comparison of performance. *Int. J. Control* **2004**, *77*, 639–653.
27. Guivant, J. Victoria Park. Available online: http://www-personal.acfr.usyd.edu.au/nebot/victoria_park.htm (accessed on 15 December 2013).
28. He, B.; Wang, B.; Yan, T.; Han, Y. A Distributed Parallel Motion Control for the Multi-Thruster Autonomous Underwater Vehicle. *Mech. Based Des. Struct. Mach.* **2013**, *41*, 236–257.
29. He, B.; Liang, Y.; Feng, X.; Nian, R.; Yan, T.; Li, M.; Zhang, S. AUV SLAM and experiments using a mechanical scanning forward-looking sonar. *Sensors* **2012**, *12*, 9386–9410.
30. Subudhi, B.; Mukherjee, K.; Ghosh, S. A static output feedback control design for path following of autonomous underwater vehicle in vertical plane. *Ocean Eng.* **2013**, *63*, 72–76.
31. Kuwata, Y.; Wolf, M.T.; Zarzhitsky, D.; Huntsberger, T.L. Safe Maritime Autonomous Navigation with COLREGS, Using Velocity Obstacles. *IEEE J. Ocean. Eng.* **39**, 110–119.
32. Bar-Shalom, Y.; Li, X.R.; Kirubarajan, T. *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*; John Wiley & Sons: Hoboken, NJ, USA, 2004.
33. Guivant, J.E.; Nebot, E.M. Solving computational and memory requirements of feature-based simultaneous localization and mapping algorithms. *IEEE Trans. Robot. Autom.* **2003**, *19*, 749–755.
34. Huang, G.; Mourikis, A.; Roumeliotis, S. A Quadratic-Complexity Observability-Constrained Unscented Kalman Filter for SLAM. *IEEE Trans. Robot.* **2013**, *29*, 1226–1243.
35. Hesch, J.; Kottas, D.; Bowman, S.; Roumeliotis, S. Consistency Analysis and Improvement of Vision-aided Inertial Navigation. *IEEE Trans. Robot.* **2014**, *30*, 158–176.
36. Kim, A.; Eustice, R. Real-Time Visual SLAM for Autonomous Underwater Hull Inspection Using Visual Saliency. *IEEE Trans. Robot.* **2013**, *29*, 719–733.
37. Dellaert, F.; Kaess, M. Square Root SAM: Simultaneous localization and mapping via square root information smoothing. *Int. J. Robot. Res.* **2006**, *25*, 1181–1203.

38. Kummerle, R.; Grisetti, G.; Strasdat, H.; Konolige, K.; Burgard, W. g^2o : A general framework for graph optimization. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011; pp. 3607–3613.
39. Sunderhauf, N.; Protzel, P. Switchable constraints vs. max-mixture models vs. RRR—A comparison of three approaches to robust pose graph SLAM. In Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA), Karlsruhe, Germany, 6–10 May 2013; pp. 5198–5203.
40. Fallon, M.; Folkesson, J.; McClelland, H.; Leonard, J. Relocating Underwater Features Autonomously Using Sonar-Based SLAM. *IEEE J. Ocean. Eng.* **2013**, *38*, 500–513.
41. Lee, S.; Lee, S. Embedded Visual SLAM: Applications for Low-Cost Consumer Robots. *IEEE Robot. Autom. Mag.* **2013**, *20*, 83–95.
42. Lategahn, H.; Stiller, C. Vision-Only Localization. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 1246–1257.

© 2015 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).