

Article

Autonomous Navigation for Autonomous Underwater Vehicles Based on Information Filters and Active Sensing

Bo He ^{1,*}, Hongjin Zhang ¹, Chao Li ¹, Shujing Zhang ¹, Yan Liang ¹ and Tianhong Yan ^{2,3,*}

¹ School of Information Science and Engineering, Ocean University of China, 238 Songling Road, Qingdao 266100, China; E-Mails: excellentzhj@163.com(H.Z.); lich07@126.com (C.L.); sjzhang365@gmail.com (S.Z.); liangyan1210@126.com(Y.L.)

² School of Mechanical & Electrical Engineering, China Jiliang University, 258 Xueyuan Street, Xiasha High-Edu Park, Hangzhou 310018, China

³ State Key Lab of Digital Manufacturing and Equipments Technology, Huazhong University of Science and Technology, Luoyu Road, Wuhan 430074, China

* Authors to whom correspondence should be addressed; E-Mails: bhe@ouc.edu.cn (B.H.); thyan@163.com (T.Y.); Tel.: +86-532-6678-2339 (B.H.); Fax: +86-532-6678-2339 (B.H.).

Received: 8 October 2011; in revised form: 11 November 2011 / Accepted: 18 November 2011 /

Published: 22 November 2011

Abstract: This paper addresses an autonomous navigation method for the autonomous underwater vehicle (AUV) *C-Ranger* applying information-filter-based simultaneous localization and mapping (SLAM), and its sea trial experiments in Tuandao Bay (Shandong Province, P.R. China). Weak links in the information matrix in an extended information filter (EIF) can be pruned to achieve an efficient approach-sparse EIF algorithm (SEIF-SLAM). All the basic update formulae can be implemented in constant time irrespective of the size of the map; hence the computational complexity is significantly reduced. The mechanical scanning imaging sonar is chosen as the active sensing device for the underwater vehicle, and a compensation method based on feedback of the AUV pose is presented to overcome distortion of the acoustic images due to the vehicle motion. In order to verify the feasibility of the navigation methods proposed for the *C-Ranger*, a sea trial was conducted in Tuandao Bay. Experimental results and analysis show that the proposed navigation approach based on SEIF-SLAM improves the accuracy of the navigation compared with conventional method; moreover the algorithm has a low computational cost when compared with EKF-SLAM.

Keywords: autonomous underwater vehicle (AUV); extended information filter; localization; navigation; sonar

1. Introduction

Autonomous Underwater Vehicles (AUVs) have portable energy and self-control ability which make them different from remote operate vehicles (ROVs). They are suitable for commercial and military tasks underwater, under-ice or in other environments [1,2]. In the past few decades, the world plans to build or has built about 200 AUVs, such as the well-known REMUS by WHOI [3] or MIT Sea Grant's Odyssey [4]. As the applications of AUVs are spreading to deeper seas and longer distances, high accuracy navigation capability will play a vital role. Inertial navigation systems (INS) are widely used in AUVs, but the navigation errors accumulate over time and waves and currents exacerbate this. Though the errors can be reduced periodically by using GPS, electromagnetic signals decay very quickly in the water, so the navigation of underwater vehicles cannot rely on GPS. As acoustic signals decay extremely slowly in seawater, acoustic navigation is widely used in three ways: long baseline, short baseline, ultra short baseline. However, deployment and recovery of the baseline is time-consuming and expensive, which limits acoustic navigation in large-scale environments [5]. In addition, it is very difficult to obtain prior maps of large-scale complex underwater environments. For all the reasons mentioned above, autonomous underwater navigation is considered one of the most challenging issues for AUVs [6].

Simultaneous localization and mapping (SLAM) does not require the aid of *a priori* information about the underwater environment, as when the vehicle moves in the water, its onboard sensors perceive the external environment and extract useful information, then form a map of the environment incrementally while positioning itself [7,8]. SLAM has attracted immense attention since the 1990s as the solution for true autonomous navigation [9], and a handful of algorithms have emerged for this purpose such as EKF-based SLAM, PF-based SLAM and IF-based SLAM. Using a parametric model, EKF-SLAM can be described as a posterior probability distribution parameterized by a state vector and covariance matrix. The algorithm consists of two main steps: prediction and update, which can be summarized as an iterative estimate and calibration process [10,11]. However, the map in EKF-SLAM comprises a covariance between the vehicle state and environmental features which must be processed in each estimate and correct step. As a result, It achieves $O(n^2)$ (the number of environmental features) computational complexity. To deal with the limitations, people divide the world into a handful of sub-maps; each containing l features [12-15]. Thus, computational complexity can be reduced to $O(l^2)$, but the tradeoff is sacrificing the convergence speed.

Particle filtering (PF) is different from KF, which solves SLAM problem by the particles based on the conditional independence. Fast SLAM, proposed by Montemerlo [16,17], is a very popular SLAM algorithm based upon particle filters. Fast SLAM decomposes SLAM into estimation of the robot path and estimate of the location of the features in the map. Once the vehicle path is established, the features are only related with the vehicle pose [18]. The particle filter is efficient in computation and can represent non-linear, non-Gaussian motion modes, but the more complex the environment is, the

more particles are required to describe the posterior probability distribution. As a result, the algorithm complexity increases. In addition, the re-sampling will lead to loss of validation and diversity of particles which results in sample depletion.

Extended Information Filter (EIF) is the information form of EKF, parameterized by the information matrix and information vector corresponding to the EKF [19]. The time projection is efficient as it is quadratic in the number of measurements and not the size of the map. However, recovering the mean from the information matrix and information vector requires a costly $O(n^3)$, matrix inversion, so the computational complexity is even higher than for EKF. It is found that the vast majority of the information matrix elements are close to zero, while the information matrix is dominated by a small number of diagonal elements [20]. If the elements are made approximately close to zero, that is pruning weak links in the information matrix, an approximate representation which is the so-called Sparse Extended Information Filter (SEIF) proposed by Thrun *et al.* is obtained [21,22]. All the basic update formulae can be implemented in constant time, irrespective of the size of the map, which greatly improves computational complexity. There are a variety of new information filter-based algorithms such as TJFT [23], Treemap [24], ESEIF [25,26], but theoretical analysis and experimental evidence show that SEIF-SLAM is computationally efficient and consistent in the relative map, so the SEIF-SLAM algorithm is preferred in our work.

The *C-Ranger* AUV was developed in Underwater Vehicle Laboratory in Ocean University of China. It is equipped with a variety of onboard equipment for sensing vehicle pose and environment. In this paper we mainly address the autonomous navigation method for the *C-Ranger* AUV. To verify the advantages of SEIF-SLAM, the application of SEIF-SLAM in AUV navigation via sea trial experiments has been studied; also the data processing approach of sonar had been presented in this work. For the mechanical scanning imaging sonar chosen as the principal sensor for active sensing of undersea obstacles, a compensation method based on feedback of the AUV pose had been used to overcome distortion of the acoustic images due to the vehicle motion. Sea trials in Tuandao Bay were carried out to verify the feasibility of the proposed methods. The experimental results and analysis show that the proposed navigation approach based on SEIF-SLAM improved the accuracy of navigation when compared with conventional method; moreover the algorithm has a low computational cost when compared with EKF-SLAM.

The remainder of the paper is organized as follows: in the next section, the SEIF-SLAM algorithm for our *C-Ranger* AUV is presented, including the main processing steps. Section 3 describes the *C-Ranger* AUV and on-board sensors employed in the SLAM module. Further, sonar data processing is discussed in Section 4, including point-feature extraction and motion-distortion compensation. The sea trial experiments is described in Section 5, and the performance of the proposed navigation method is evaluated; then the results and possible future improvements are discussed in Section 6. Finally, we summarize the results and draw the fundamental conclusions.

2. A SEIF-SLAM Algorithm for the *C-Ranger* AUV

AUV travels undersea at a certain depth in most cases, so the bidimensional vehicle-landmark model was adopted to represent the AUV and landmarks (also called features) in the undersea

environment. The vehicle pose $\mathbf{x}_t = (x_t, y_t, \theta_t)^T$ and the set of map features $\mathbf{M} = (\mathbf{m}_i, 1 \leq i \leq N)^T$ are stored in the state vector:

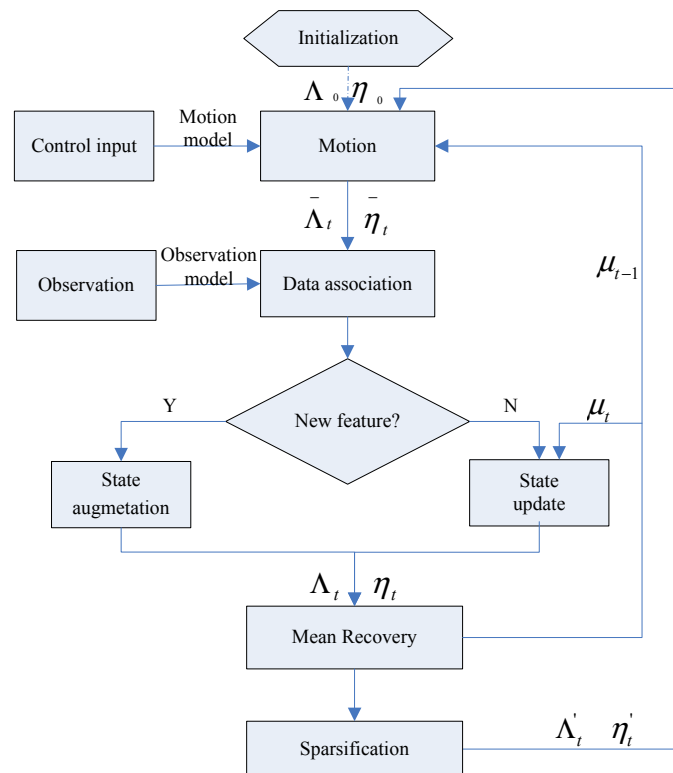
$$\xi_t = (\mathbf{x}_t^T, \mathbf{M}^T)^T \tag{1}$$

Like in Smith *et al.* [27], a first-order linearization of the motion and measurement models is employed considering the uncertainty in the data as independent, white Gaussian noise. Corresponding to the Gaussian distribution in Appendix, SEIF-SLAM can be presented as the following posterior probability distribution:

$$p(\mathbf{x}_t, \mathbf{M} | Z^t, U^t) = N(\boldsymbol{\mu}_t, \Sigma_t) = N\left(\begin{bmatrix} \boldsymbol{\mu}_{x_t} \\ \boldsymbol{\mu}_M \end{bmatrix}, \begin{bmatrix} \Sigma_{x_t X_t} & \Sigma_{x_t M} \\ \Sigma_{M X_t} & \Sigma_{MM} \end{bmatrix}\right) = N^{-1}(\boldsymbol{\eta}_t, \Lambda_t) = N^{-1}\left(\begin{bmatrix} \boldsymbol{\eta}_{x_t} \\ \boldsymbol{\eta}_M \end{bmatrix}, \begin{bmatrix} \Lambda_{x_t X_t} & \Lambda_{x_t M} \\ \Lambda_{M X_t} & \Lambda_{MM} \end{bmatrix}\right) \tag{2}$$

where, $\boldsymbol{\mu}_t$ is the mean of the state vector and $\boldsymbol{\eta}_t$ is the information vector, Σ_t and Λ_t denote the covariance and information matrix respectively, Z^t and U^t are the history of observational data and motion control inputs. To calculate the probability distribution, the algorithm mainly includes motion update, measurement update, sparsification, mean recovery and other steps. Figure 1 shows the structure of the algorithm.

Figure 1. The flow chart of the SEIF-SLAM algorithm.



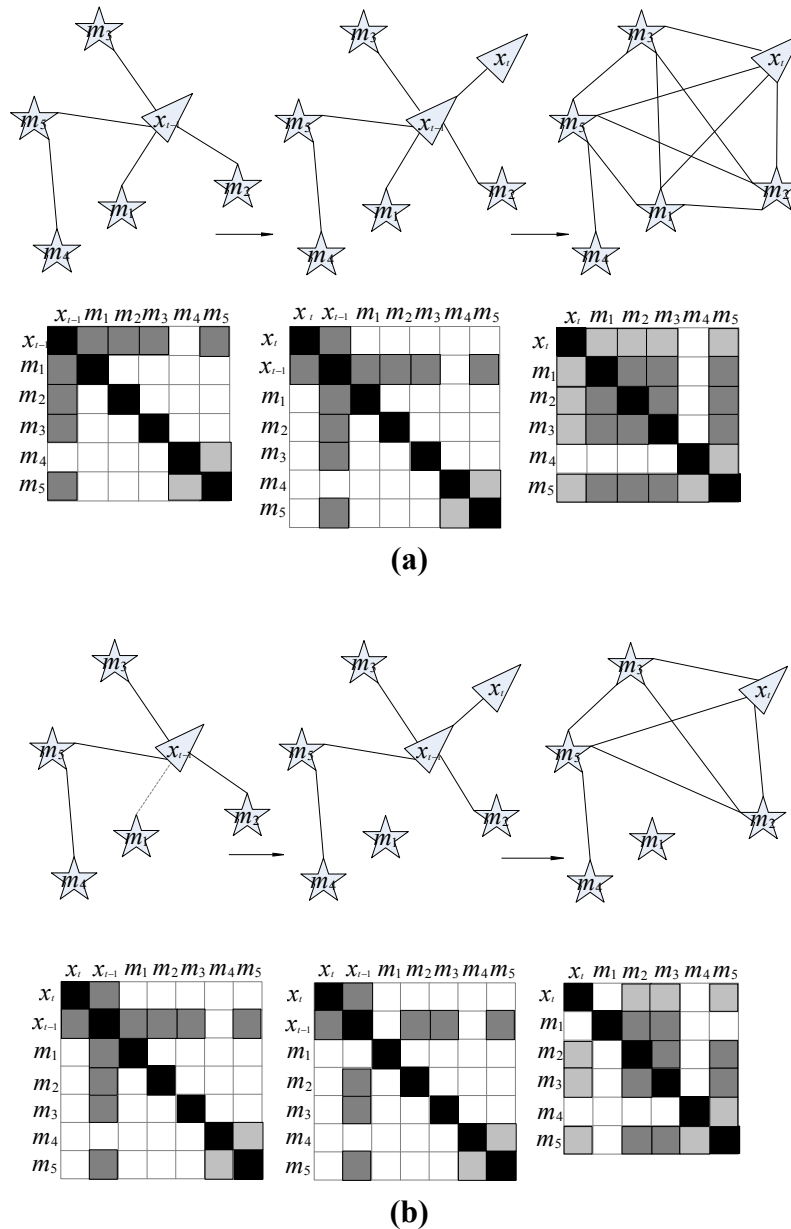
(A) Motion Update Step

The motion update step predicts the distribution over the new robot pose from time $t - 1$ to time t and subjects it to a Markov model, in general, a nonlinear function f of the previous pose \mathbf{x}_{t-1} and the control inputs \mathbf{u}_t . Equation (3) is the first-order linearization about the mean robot pose $\boldsymbol{\mu}_{x_{t-1}}$,

F denotes the Jacobian matrix about the state vector at time $t - 1$ and the term $\mathbf{v}_t \sim N(0, Q_t)$ represents the white Gaussian noise in the model, Q_t is the noise covariance:

$$\mathbf{x}_t = f(\mathbf{x}_{t-1}, \mathbf{u}_t) + \mathbf{v}_t \approx f(\boldsymbol{\mu}_{x_{t-1}}, \mathbf{u}_t) + F(\mathbf{x}_{t-1} - \boldsymbol{\mu}_{x_{t-1}}) + \mathbf{v}_t \tag{3}$$

Figure 2. The evolution of the information matrix: **(a)** The evolution of the information matrix without a sparsification step. **(b)** The evolution of the information matrix when breaking weak links.



The motion update step can be implemented in two sub-steps. First, as in Figure 2(a), the state vector is grown to include the new robot pose $\hat{\xi}_t = (\mathbf{x}_{t-1}^T, \mathbf{x}_t^T, \mathbf{M}^T)^T$. According to the work by Eustice *et al.* [25,26], the augmentation of the information matrix and vector are given in Equations (4) and (5):

$$\hat{\Lambda}_t = \left[\begin{array}{c|c} \left(\Lambda_{x_{t-1}x_{t-1}} + F^T Q_t^{-1} F \right) & -F^T Q_t^{-1} \Lambda_{x_{t-1}M} \\ \hline -Q_t^{-1} F & Q_t^{-1} \mathbf{0} \\ \Lambda_{Mx_{t-1}} & \mathbf{0} \quad \Lambda_{MM} \end{array} \right] = \left[\begin{array}{c|c} \hat{\Lambda}_t^{11} & \hat{\Lambda}_t^{12} \\ \hline \hat{\Lambda}_t^{21} & \hat{\Lambda}_t^{22} \end{array} \right] \quad (4)$$

$$\hat{\boldsymbol{\eta}}_t = \left[\begin{array}{c} \boldsymbol{\eta}_{x_{t-1}} - F^T Q_t^{-1} (f(\boldsymbol{\mu}_{x_{t-1}}, \mathbf{u}_t) - F \boldsymbol{\mu}_{x_{t-1}}) \\ \hline Q_t^{-1} (f(\boldsymbol{\mu}_{x_{t-1}}, \mathbf{u}_t) - F \boldsymbol{\mu}_{x_t}) \\ \boldsymbol{\eta}_M \end{array} \right] = \left[\begin{array}{c} \hat{\boldsymbol{\eta}}_t^1 \\ \hline \hat{\boldsymbol{\eta}}_t^2 \end{array} \right] \quad (5)$$

Secondly, \mathbf{x}_t is marginalized from the posterior probability distribution to achieve the desired distribution over $\boldsymbol{\xi}_t = (\mathbf{x}_t^T, \mathbf{M}^T)^T$ according to Appendix. The predicted information matrix $\bar{\Lambda}_t$ and information vector $\bar{\boldsymbol{\eta}}_t$ are:

$$\bar{\Lambda}_t = \hat{\Lambda}_t^{22} - \hat{\Lambda}_t^{21} (\hat{\Lambda}_t^{11})^{-1} \hat{\Lambda}_t^{12} \quad (6)$$

$$\bar{\boldsymbol{\eta}}_t = \hat{\boldsymbol{\eta}}_t^2 - \hat{\Lambda}_t^{21} (\hat{\Lambda}_t^{11})^{-1} \hat{\boldsymbol{\eta}}_t^1 \quad (7)$$

(B) Measurement Update Step

Sonar senses environment features actively in the experiment and observations of features are the key to reduce the uncertainty in the estimates for the robot pose and the map. The measurement step is also subject to a Markov model which is a nonlinear function h of the state vector $\boldsymbol{\xi}_t$. Equation (8) shows the first-order linearization about the mean state vector $\boldsymbol{\mu}_t$ and observed features with the Jacobian H evaluated at this mean. The term $\mathbf{w}_t \sim N(0, R_t)$ represents the white Gaussian noise in the model, R_t is the noise covariance:

$$\mathbf{z}_t = h(\boldsymbol{\xi}_t) + \mathbf{w}_t \approx h(\boldsymbol{\mu}_t) + H(\boldsymbol{\xi}_t - \boldsymbol{\mu}_t) + \mathbf{w}_t \quad (8)$$

When a feature is observed repeatedly, it will be used to update the state estimation. According to [19] by Thrun *et al.*, the information matrix Λ_t and the information vector $\boldsymbol{\eta}_t$ can be obtained:

$$\Lambda_t = \bar{\Lambda}_t + H^T R_t^{-1} H \quad (9)$$

$$\bar{\boldsymbol{\eta}}_t = \bar{\boldsymbol{\eta}}_t + H^T R_t^{-1} (\mathbf{z}_t - h(\boldsymbol{\mu}_t) + H \boldsymbol{\mu}_t) \quad (10)$$

with:

$$H = \left(\begin{array}{ccc|ccc} \frac{\partial h}{\partial \mathbf{x}_t} & 0 & \dots & 0 & \frac{\partial h}{\partial \mathbf{m}_i} & 0 & \dots & 0 \end{array} \right)$$

The observation model is only a function of the vehicle state \mathbf{x}_t and the observed features \mathbf{m}_i ($1 \leq i \leq N$) at time t , so H is zero everywhere except at positions associated with \mathbf{x}_t , \mathbf{m}_i . The measurement update step only strengthens the links between vehicle pose and features to be updated, so the sparseness of information matrix can never be changed.

(C) Sparsification Step

To make information sparse, SEIF's strategy for sparsification is based on partitioning the map into a union of three disjoint sets:

$$\mathbf{M} = \{\mathbf{M}^+, \mathbf{M}^-, \mathbf{M}^0\}$$

where \mathbf{M}^+ is the current active features that will remain active after sparsifying, \mathbf{M}^- denotes the passive features that will remain passive, \mathbf{M}^0 comprises the active features that will be made passive [28]. So the SLAM posterior can be factored into:

$$\begin{aligned} p(\mathbf{x}_t, \mathbf{M} | Z^t, U^t) &= p(\mathbf{x}_t, \mathbf{M}^+, \mathbf{M}^-, \mathbf{M}^0 | Z^t, U^t) \\ &= p(\mathbf{x}_t | \mathbf{M}^+, \mathbf{M}^-, \mathbf{M}^0, Z^t, U^t) p(\mathbf{M}^+, \mathbf{M}^-, \mathbf{M}^0 | Z^t, U^t) \end{aligned} \quad (11)$$

Due to the conditional independence between the pose and \mathbf{M}^- of the vehicle, hence we can set \mathbf{M}^- to an arbitrary value. Here, it is simply chosen as 0. The evolution of information matrix is shown in Figure 2.

Figure 2(a) illustrates the situation before sparsification: according to the partition, $\mathbf{M}^+ = \{\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3, \mathbf{m}_4\}$, $\mathbf{M}^- = \{\mathbf{m}_4\}$ at time t , the information matrix tends to be dense after time-projection step (state augmentation and marginalization) which is the main reason for creating weak links.

By eliminating the weak links between the features and vehicle pose in the information matrix, a sparse approximation that allows for efficient SLAM can be achieved. As shown in Figure 2(b), \mathbf{m}_1 is made passive before state augmentation and marginalization. It is obvious that the information matrix is sparse. To sparsify the information matrix, the posterior is approximated by dropping the dependence on \mathbf{M}^0 in the first term of Equation (11):

$$\begin{aligned} p(\mathbf{x}_t, \mathbf{M} | Z^t, U^t) &\approx p(\mathbf{x}_t | \mathbf{M}^+, \mathbf{M}^- = 0, Z^t, U^t) p(\mathbf{M}^+, \mathbf{M}^-, \mathbf{M}^0 | Z^t, U^t) \\ &= \frac{p(\mathbf{x}_t, \mathbf{M}^+ | \mathbf{M}^- = 0, Z^t, U^t)}{p(\mathbf{M}^+ | \mathbf{M}^- = 0, Z^t, U^t)} p(\mathbf{M}^+, \mathbf{M}^-, \mathbf{M}^0 | Z^t, U^t) \end{aligned} \quad (12)$$

The information matrix for the distribution $p(\mathbf{x}_t, \mathbf{M}^+, \mathbf{M}^0 | \mathbf{M}^- = 0)$ is:

$$\tilde{\Lambda}_t = \mathbf{S}_{x, M^+, M^0} \mathbf{S}_{x, M^+, M^0}^T \Lambda_t \mathbf{S}_{x, M^+, M^0} \mathbf{S}_{x, M^+, M^0}^T \quad (13)$$

where \mathbf{S}_{x, M^+, M^0} denotes a projection matrix, used to extract the submatrix of all state variables except \mathbf{M}^0 . Let Λ_t^1 , Λ_t^2 , Λ_t^3 denote the information matrices for the terms $p(\mathbf{x}_t, \mathbf{M}^+ | \mathbf{M}^- = 0, Z^t, U^t)$, $p(\mathbf{M}^+ | \mathbf{M}^- = 0, Z^t, U^t)$ and $p(\mathbf{M}^+, \mathbf{M}^-, \mathbf{M}^0 | Z^t, U^t)$ respectively, so the sparse information matrix can be obtained as Equation (14) by putting these matrices together according to Equation (12):

$$\begin{aligned} \Lambda'_t &= \Lambda_t^1 - \Lambda_t^2 + \Lambda_t^3 \\ &= \Lambda_t - \tilde{\Lambda}_t \mathbf{S}_{M^0} (\mathbf{S}_{M^0}^T \tilde{\Lambda}_t \mathbf{S}_{M^0})^{-1} \mathbf{S}_{M^0}^T \tilde{\Lambda}_t \\ &\quad + \tilde{\Lambda}_t \mathbf{S}_{x, M^0} (\mathbf{S}_{x, M^0}^T \tilde{\Lambda}_t \mathbf{S}_{x, M^0})^{-1} \mathbf{S}_{x, M^0}^T \tilde{\Lambda}_t - \Lambda_t \mathbf{S}_x (\mathbf{S}_x^T \Lambda_t \mathbf{S}_x)^{-1} \mathbf{S}_x^T \Lambda_t \end{aligned} \quad (14)$$

The resulting information vector is now obtained by the following simple consideration:

$$\boldsymbol{\eta}'_t = \Lambda'_t \boldsymbol{\mu}_t = (\Lambda_t - \Lambda_t + \Lambda'_t) \boldsymbol{\mu}_t = \boldsymbol{\eta}_t + (\Lambda'_t - \Lambda_t) \boldsymbol{\mu}_t \quad (15)$$

(D) Mean Recovery

The information form of the Gaussian is parameterized by its information vector and information matrix. However, the linearization, sparsification, measurement update steps require the mean of the state vector. According to the duality of Gaussian distribution in Appendix, we can get the mean vector:

$$\boldsymbol{\mu}_t = \Lambda_t^{-1} \boldsymbol{\eta}_t \quad (16)$$

Due to the great cost of full state recovery and the steps that require the mean of state vector oftentimes only require a subset of the state mean, the mean vector can be recovered by dividing it into two sets:

$$\begin{bmatrix} \Lambda_{ll} & \Lambda_{lb} \\ \Lambda_{bl} & \Lambda_{bb} \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu}_l \\ \boldsymbol{\mu}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{\eta}_l \\ \boldsymbol{\eta}_b \end{bmatrix} \quad (17)$$

where $\boldsymbol{\mu}_l$ is the “local portion” that needs to be recovered and $\boldsymbol{\mu}_b$ is the “begin portion” of the map. If current estimation for $\boldsymbol{\mu}_b$ is kept fixed, an estimate of $\boldsymbol{\mu}_l$ can be obtained as:

$$\hat{\boldsymbol{\mu}}_l = \Lambda_{ll}^{-1} (\boldsymbol{\eta}_l - \Lambda_{lb} \hat{\boldsymbol{\mu}}_b) \quad (18)$$

3. The C-Ranger AUV and On-Board Sensors

3.1. C-Ranger AUV

The *C-Ranger* is an open-frame AUV with dimensions of 1.6 m × 1.3 m × 1.1 m (length, width and height), as shown in Figure 3. There are two electronic cabins and five underwater propeller thrusters. The control architecture of *C-Ranger* is illustrated in Figure 4. The AUV has good maneuverability due to its five DOFs, including surge, heave, roll, pitch, and yaw. The thrust system of this platform consists of five propeller thrusters, where two thrusters are parallel to the bow direction, and installed in the abdomen to provide horizontal thrust for mainly controlling the surge and roll, The other three thrusters are employed to provide vertical thrust to control the heave, roll, and pitch, two of which are installed on both sides of the bow, and the remaining one is installed on the rear of the vehicle. The upper hull of the *C-Ranger* is the instrument compartment housing sensors, two industrial computers, a communication module, internal monitoring module and other equipment, while the lower hull is the power and thrust system composed of lithium-ion batteries, power management module, motor-driver module, etc. The maximum speed of the *C-Ranger* is 3 knots, and it can operate for up to 8 hours when fully charged (tested at speed of one knot).

3.2. On-Board Sensors

A number of sensors are installed on the *C-Ranger*, some of them are explicitly related to SLAM. These sensors are basically divided into two groups: the internal and the external. Internal sensors include digital compass, gyro, Attitude and Heading Reference System (AHRS) and pressure sensor. External sensors include mechanical scanning sonar, Doppler Velocity Log (DVL), altimeter, CCD camera and GPS.

Figure 3. C-Ranger in deployment.

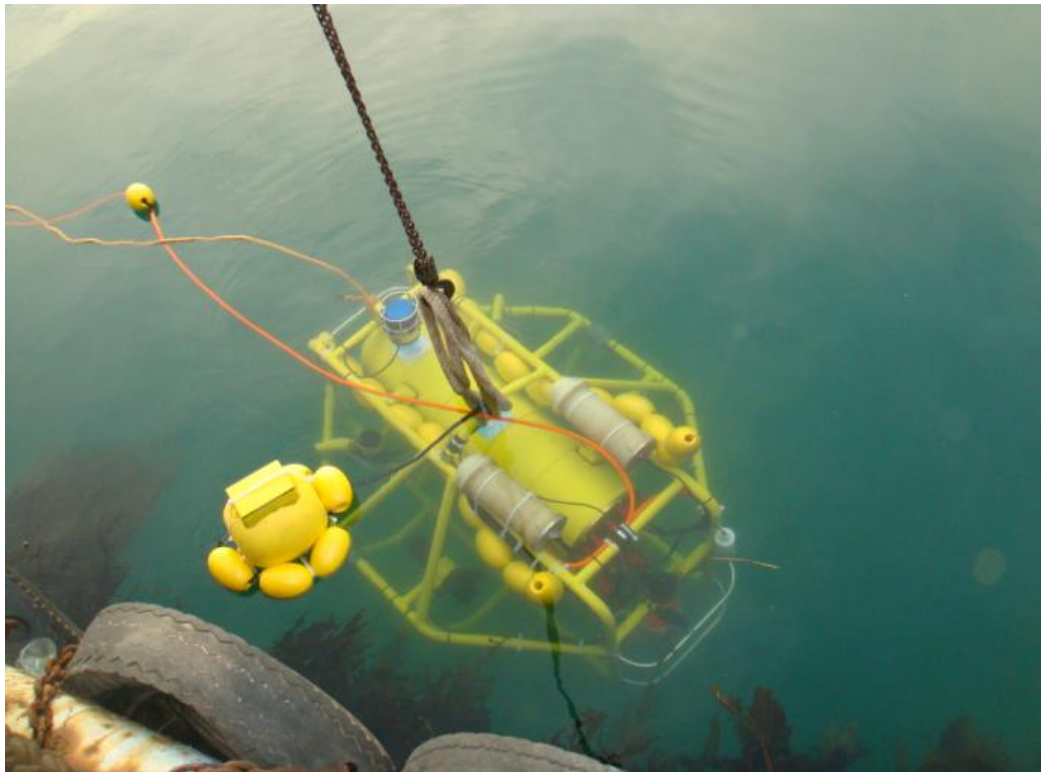
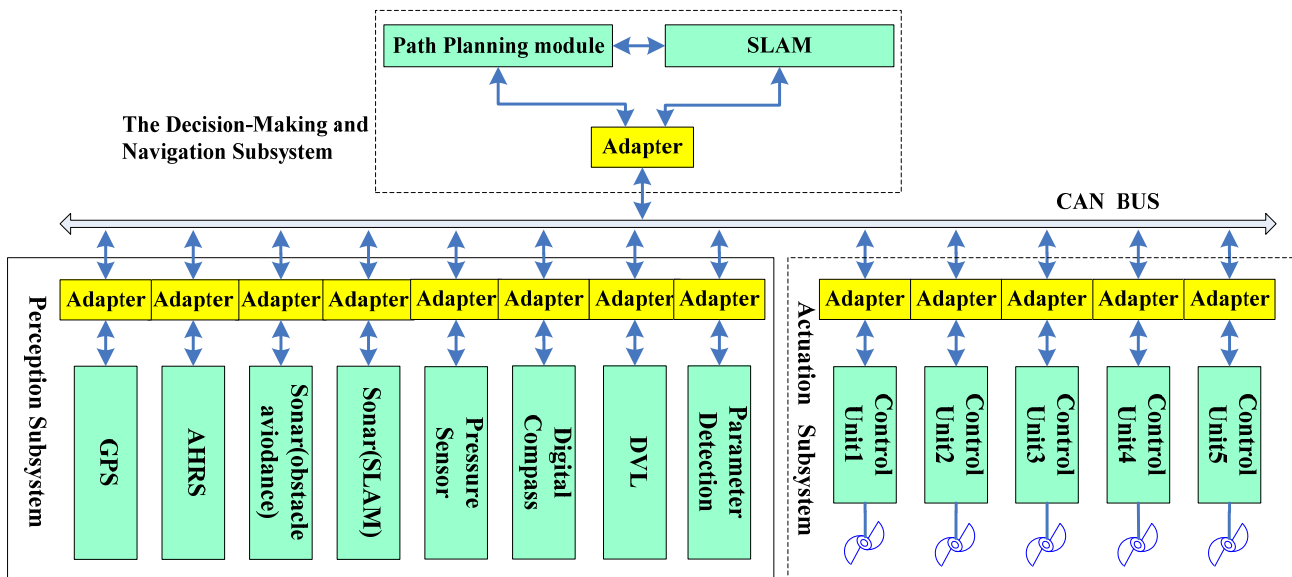


Figure 4. Control architecture of the C-Ranger.



(A) Mapping-Related Sensor: Active Imaging Sensor

A mechanical-scanning forward-looking sonar (Super Seaking DST, Tritech) used for active sensing of environment features is installed at the front top of *C-Ranger*. It is the principal sensor of the *C-Ranger* AUV. The operating frequency of the sonar is 675 kHz, and its maximum range is 300 meters. Generally the scanning rate of this kind of sonar is slow, which would make the acoustic image distorted. A compensation for motion-induced distortion will be addressed in the next section.

(B) Velocity Sensor: DVL

The DVL (NavQuest600, LinkQuest) is used to provide the velocities of the vehicle relative to the seabed. In addition, the NavQuest600 can provide other information: pitch angle, roll angle, heading, altitude, depth, temperature and velocities relative to the ocean currents.

(C) Angular Sensors: AHRS and Gyro

The AHRS (M2, Innalabs) is used to produce attitude information, and the gyro (VG951D) is used to measure angular velocity in the process of AUV navigation. AHRS M2 is a low-cost high-performance inertial navigation system, and magnetic interference will not affect the accuracy of headings over short times.

(D) Positioning Sensor: GPS

To evaluate navigation performance of the *C-Ranger*, a high-precision and high-dynamic GPS receiver is employed. In the absence of SA, the positioning accuracy is up to 1.1 m (CEP), and the data update rate is up to 20 Hz. The GPS sensor can offer a benchmark to evaluate the estimation of robot trajectory.

4. Data Processing of Sonar

Raw data measured by sensors should be pretreated before it is used in the navigation algorithm. The amount of the raw data from most sensors is not very high, so they can be used in SLAM simply after denoising and synchronizing. Sonar is very essential for the environmental map building, but the large amount of sonar data constrains the real-time navigation. Furthermore, its long scan-cycle will lead to motion-distortion. For the reasons, it is vital to handle sonar data appropriately before being used in SLAM algorithm, thus the data processing of sonar is presented in this section.

4.1. Feature Extraction of Sonar Data

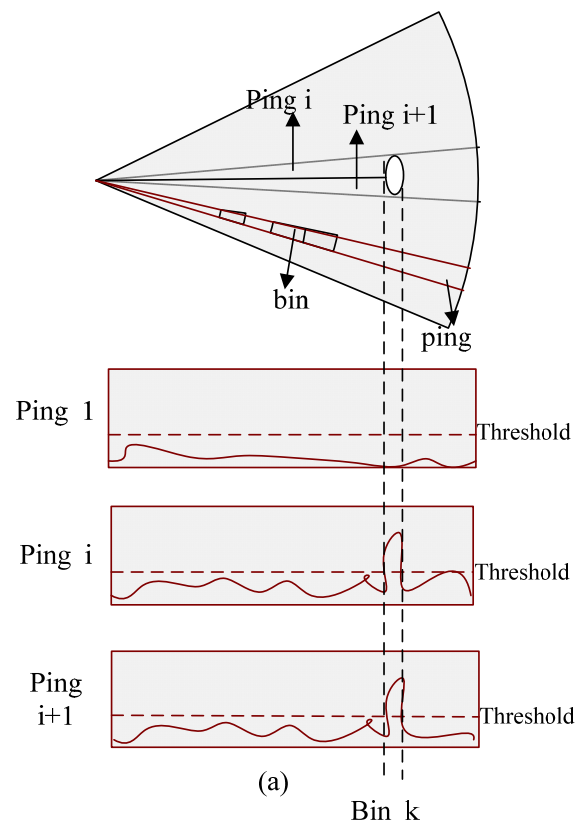
Currently methods based on image processing are popular ones applied to extract features from raw sonar data [29,30], but they are generally too slow for the applications such as AUV navigation using mechanical-scanning-sonar. A real-time data processing is proposed in the next paragraphs.

Sonar transmits sound waves stepwisely and receives them after encountering obstacles; we call each sound wave beam one *ping* and every *ping* can be divided into several bins. The relative position of a bin reflects the distance between the sonar and the obstacle, the larger the intensity of bins are, the more obvious features there are, and *vice versa* [31]. As Figure 5 shows, sonar scans stepwisely in a given sector, and finds one obstacle in the k -th bin of the i -th and $(i + 1)$ -th *ping*, respectively. Then the intensity and the relative position for every bin whose intensity is beyond a predefined threshold will be translated into corresponding feature information.

Noise in raw data should be eliminated, and the number of features extracted should be as low as possible due to the demands of real-time navigation, in other words, we need to make the features sparse, Figure 6(a–e) show the process. In Figure 6(a) the sonar observes N_1, N_2 which are very close

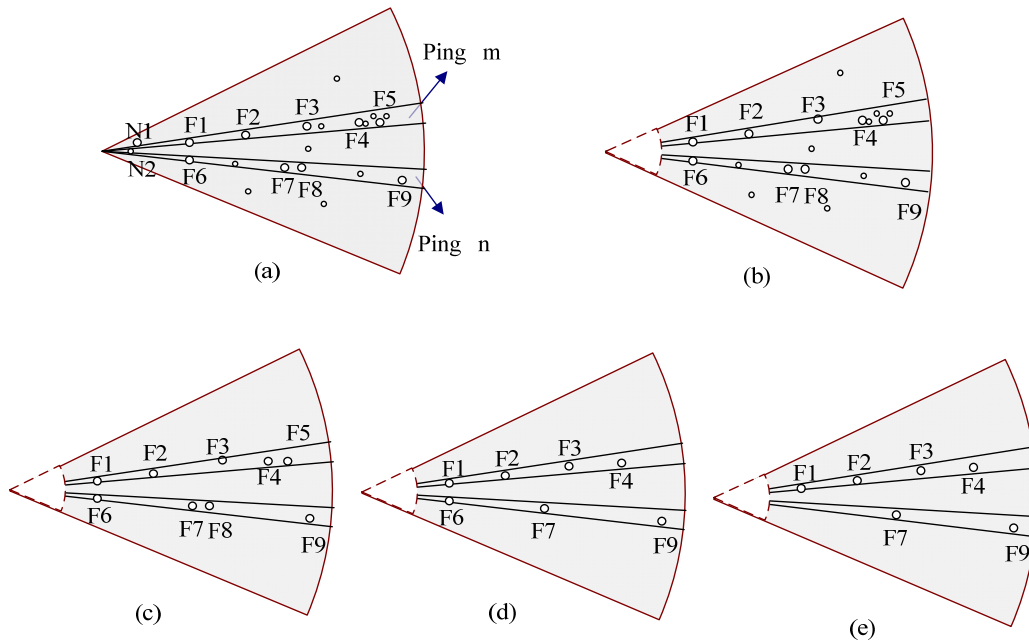
to the sonar head (transducer), high-intensity features in two *pings*, *i.e.*, $ping : m = \{F_1, F_2, F_3, F_4, F_5\}$, $ping : n = \{F_6, F_7, F_8, F_9\}$, as well as any low-intensity features (not labeled in the figure). First, we eliminate the noise caused by the sonar itself: according to the distance relationship between sonar and the frame of AUV, we remove the noise close to the vehicle as Figure 6(b) shows. The second step is to eliminate background noise: removing features whose intensity is below the predefined threshold, we will retain features $F_1 \sim F_9$ with high intensity in Figure 6(c). The third step is the processing of a single-*ping*: selecting local *prominent* points in each *ping* in order to ensure that the target-related information is retained, then removing redundant features near the *prominent* point by using a distance threshold. Thus we will get features in each *ping* and effectively ensure that one target will not match more than one feature. In *ping* m , F_5 is within the sparsification threshold (a distance value) of the F_4 , so are F_8 and F_7 in *ping* n . Then because of $F_4 > F_5$, $F_7 > F_8$ in intensity, we will get the results shown in Figure 6(d).

Figure 5. The schematic diagram of sonar scanning obstacles.



Generally the retained features are still too dense, even after the above radial processing, so we need to conduct further sparsification between *pings*. The procedure is similar to the above step 3, whether one feature is retained or not depends on a threshold as well, but such operations are along direction of a circular arc. The results shown in Figure 6(e) can be obtained in this way. The above processing actually can cut off most redundant information, and greatly reduce the number of features without affecting positioning accuracy. The thresholds mentioned above depend on several factors, such as the distribution of environmental features and accuracy of features' positions, moreover, the requirement by efficiency of SLAM is also taken into account if necessary.

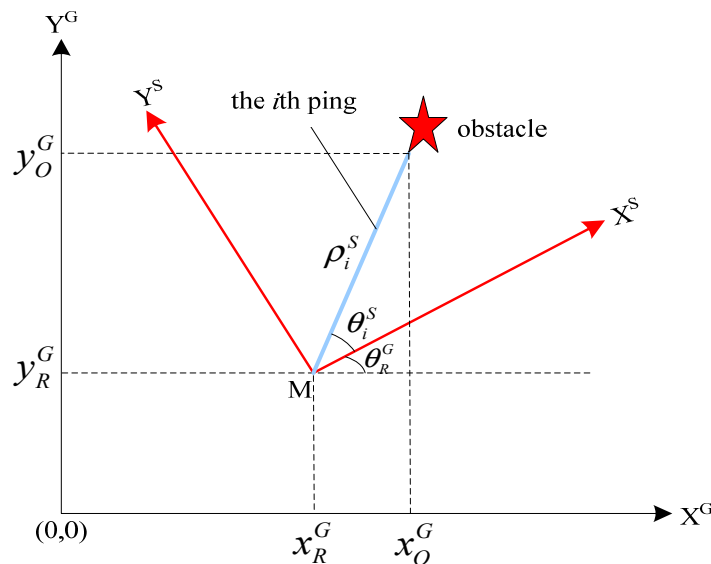
Figure 6. The schematic diagram of sonar data sparsification.



4.2. Compensation on Motion-Induced Distortion for Mechanical-Scanning Sonar

The transducer head of a mechanical-scanning imaging sonar usually needs a considerable period of time to perform a 360° rotation. This is an important issue that has to be taken into account when operating with such sonar mounted on an AUV, since the resulting acoustic images can get distorted as a consequence of the vehicle’s motion. Generally, this effect can be ignored for low velocities. For higher velocities, it has to be specially dealt with. In the case of the *C-Ranger*, we use position feedback to undistort the data. The principle of data undistortion is shown in Figure 7, where, superscripts ‘G’ and ‘S’ stand for the global coordinate frame and the sonar coordinate frame respectively; subscripts ‘R’ and ‘O’ denote variables for the vehicle and the features, respectively.

Figure 7. Compensation of the motion-induced distortion.



Firstly, a global coordinate frame is built at the starting time of every circle of sonar scanning. In this global coordinate frame, the vehicle's initial position is supposed at the origin (0, 0). Given the vehicle moves to point M after the scan interval t for one *Ping*. During this period, the vehicle's velocities v_x and v_y and can offered by SLAM module, and the vehicle's displacement (x_R^G, y_R^G) can be obtained as:

$$\begin{cases} x_R^G = v_x \cdot t \\ y_R^G = v_y \cdot t \end{cases} \quad (19)$$

Simultaneously, the vehicle's rotation angle θ_R^G can be provided by outputs of the SLAM module. Then, in the vehicle coordinate frame, the feature (marked as pentagram in Figure 7) is detected with a distance ρ_i^S from the vehicle (*i.e.*, sonar) and the angle θ_i^S , where i stands for index of the *Bin* in the *Ping*. Finally, we need to transform the polar coordinates (ρ_i^S, θ_i^S) to the Cartesian coordinates in the global coordinate frame and incorporate the motion displacement of the vehicle, thus the compensation formula can be presented as:

$$\begin{cases} x_O^G(i) = x_R^G + \rho_i^S \cdot \sin(\theta_i^S + \theta_R^G) \\ y_O^G(i) = y_R^G + \rho_i^S \cdot \cos(\theta_i^S + \theta_R^G) \end{cases} \quad (20)$$

By parity of reasoning, after the acquisition of every scanning *Ping*, the position of features will be corrected according to the above formula. Once the sonar completes one circle of scan, the undistorted image will be obtained without doubt.

We use the data in the 49th circle of the Abandoned Marina Dataset [32] to verify the effect of this correction method. Figure 8(a) shows the acoustic image built from the raw sonar data. Since the vehicle's motion has been ignored during the generation of the image, obvious distortion of the observed features appears when comparing it with the aerial image of the test scenario in Figure 8(c).

Figure 8. Effect of the vehicle motion on acoustic images. **(a)** Raw sonar image. **(b)** Corrected sonar image. **(c)** Zenithal view of the Abandoned Marina [32].

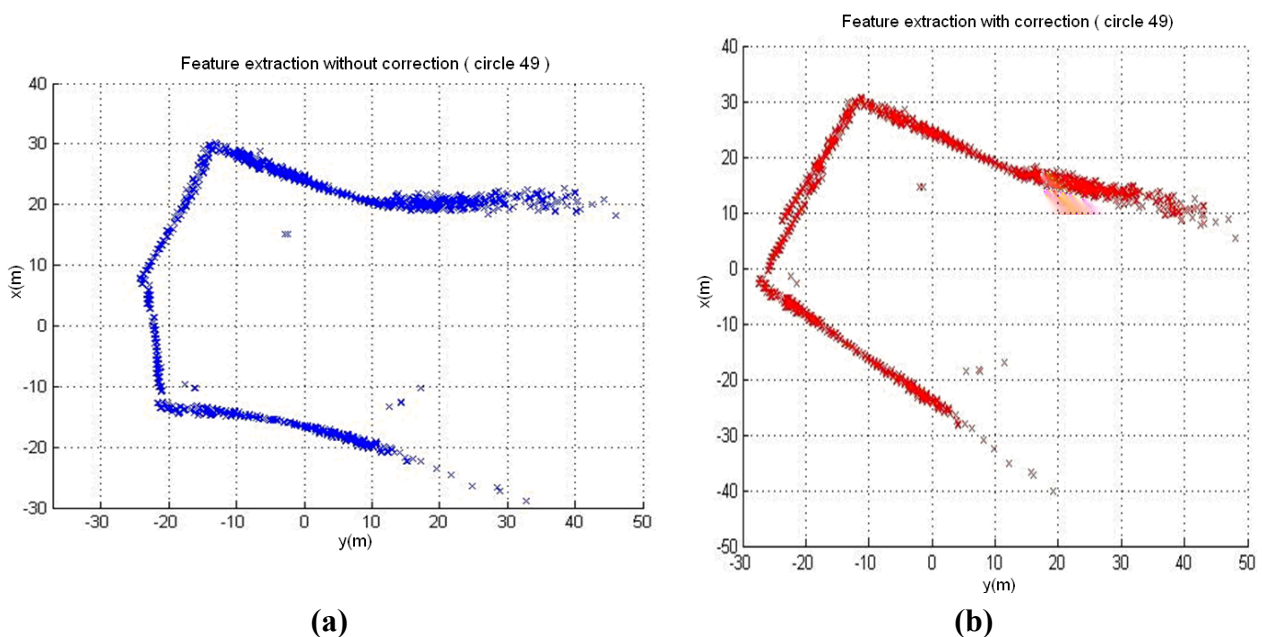


Figure 8. Cont.**(c)**

The corrected image using the proposed method is shown in Figure 8(b). Obviously the distortion of the image is almost cancelled and a more accurate image is represented. The result demonstrates that our correction approach is effectual.

5. Experiment and Results Analysis

5.1. Experiment in Tuandao Bay

The experiment was performed at Tuandao Bay in Qingdao (China). The satellite map (from Google Earth) and AUV trajectory measured by GPS are shown in Figure 9, where a starting point with direction is marked using a green arrow. The total travel in the experiment is up to 2,812 m with the sailing speed about 1 knot. The scan sector of the principal sonar was configured to 180° with a scanning range of 100 meters.

Figure 9. The satellite map of Tuandao Bay and the trajectory of the *C-Ranger* by GPS.



5.2. Experimental Results

In the experiment, the global coordinates measured by a GPS receiver in the *C-Ranger*, with a GPS antenna in a buoy connected to the AUV via a short cable, will be taken as the *ground truth* used for the evaluation of SLAM results. Figure 10 shows the features (marked with blue asterisk) extracted directly from sonar data without using sparsification. The point-features in the picture are very dense, and there are many “features” away from the possible objects such as vessels and sea bank, actually a large part of them are noise. As known, the large amount of point-features would increase the computational cost of SLAM and limit the real-time performance of AUV navigation. So the approach presented in Section 4 is necessary to eliminate redundant ones and reduce the number of features. Figure 11 illustrates the result after denoising and sparsifying, the remained features are basically the most representative ones needed to describe the undersea objects.

Figure 10. The features extraction without sparsification.

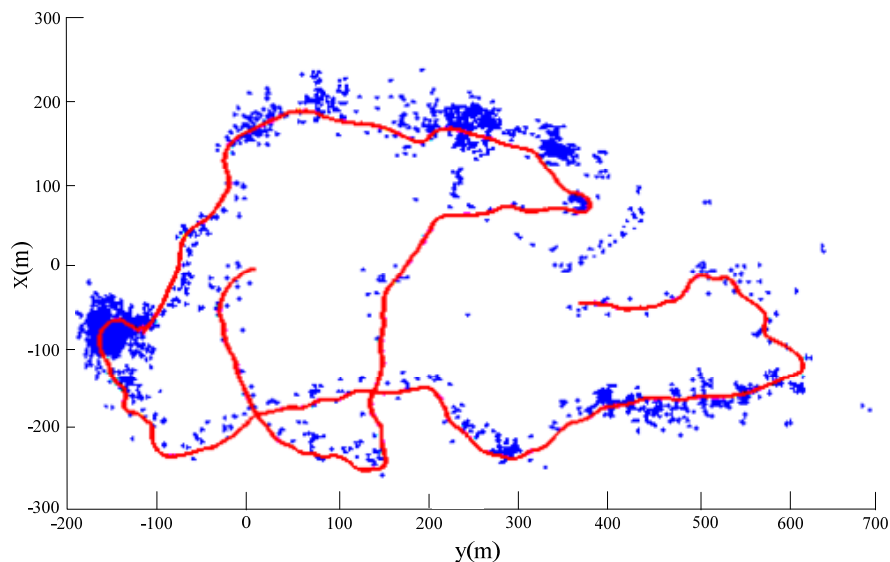
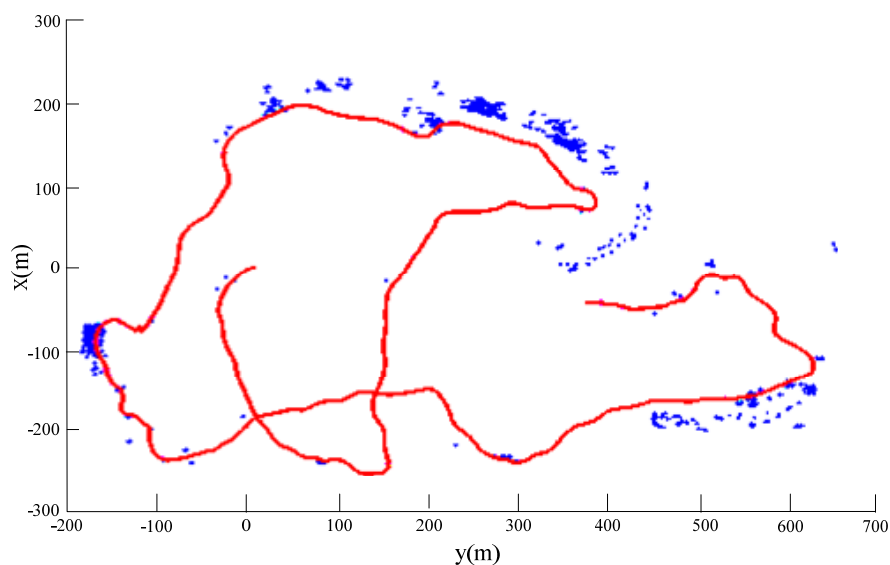


Figure 11. The features extraction with sparsification.



A comparison of the trajectories for GPS (the red line), EKF (the light blue line) and the SEIF-SLAM algorithm (the blue line) as well as the features (blue points) is shown in Figure 12. Obviously, the deviation of SLAM relative to GPS is smaller than that of EKF, and the deviation of EKF tends to increase gradually. On the other hand, the location of the features obtained by SLAM approach match the actual environment landmarks quite well.

Figure 12. Comparison of the trajectories for GPS (red line), EKF (light blue line) and the SEIF-SLAM algorithm.

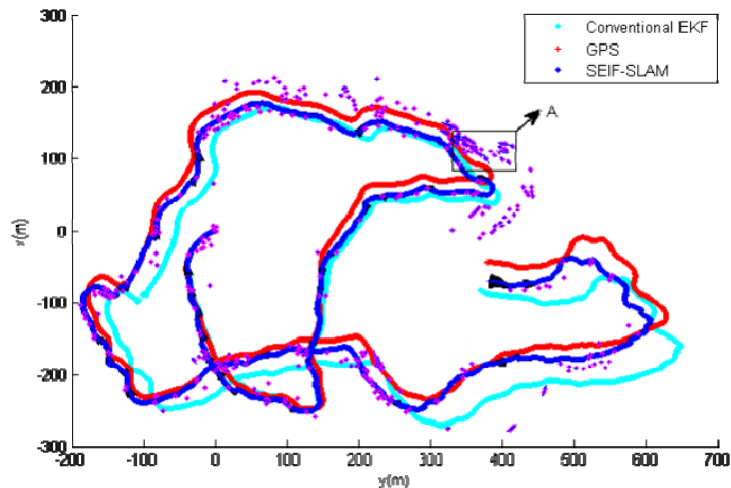
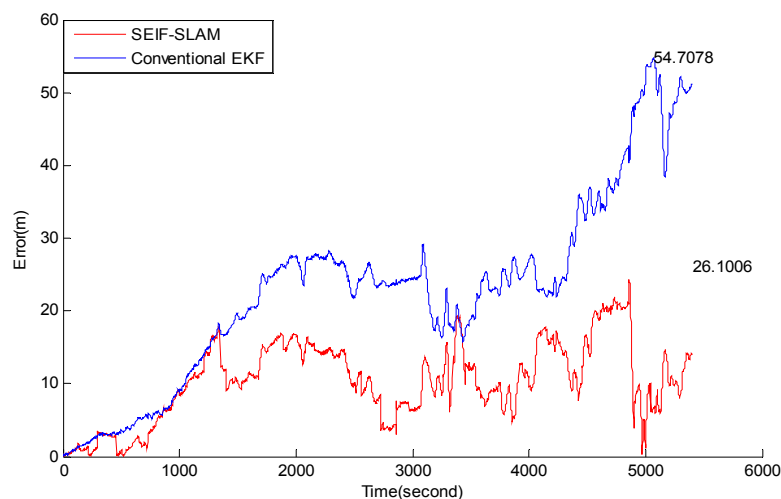


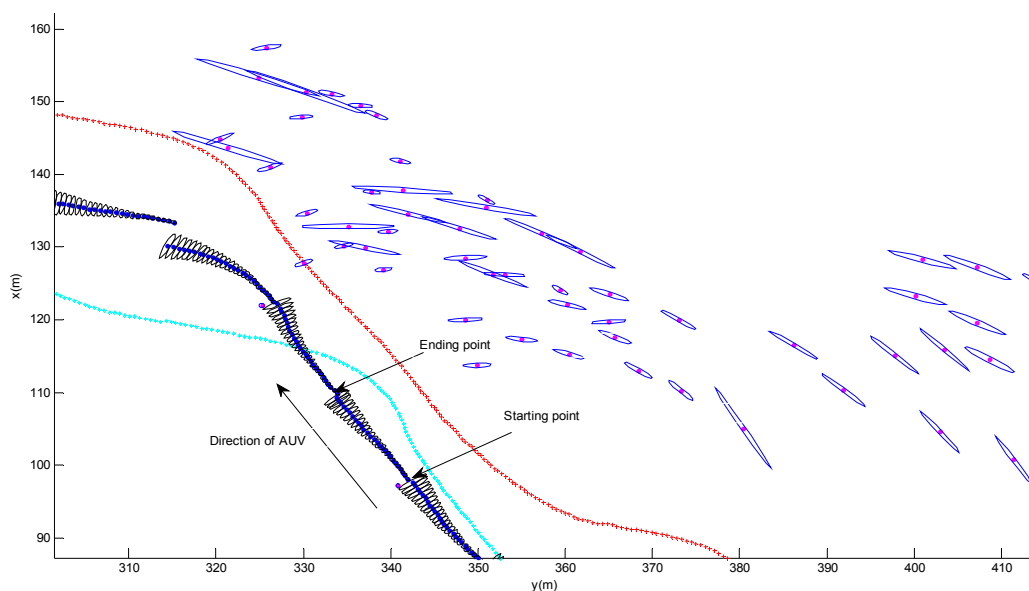
Figure 13 presents the estimation errors of SEIF-SLAM (the red line) and EKF (the blue line) relative to GPS, respectively. In this figure we can see that the error of SEIF-SLAM is always smaller than that of EKF for the whole experiment except for a few tens of seconds in the beginning. The maximum error of SEIF-SLAM is 26.1 m which is only about 7.3‰ of the whole course, and appears at the time of 4,800 seconds after starting off. As shown in Figure 9, the corresponding point at this time is at the position marked as a yellow star. That is because there were many moving targets present such as little boats during the experiment which would impact the sonar observations.

Figure 13. Plots of the error of conventional extended-Kalman-filter and SEIF-SLAM relative to GPS. The GPS data has been used as ground truth.



For more clarity, area A marked in Figure 12 is zoomed, as shown in Figure 14, in which the black and blue ellipses denote the uncertainty of vehicle pose and features separately. It is obvious from Figure 14 that the vehicle's uncertainty increases for some time and then becomes very small suddenly. In fact, we note that the change of the uncertainty is cyclical: at the starting stage, the uncertainty for the pose is small and will increase gradually in one sonar-scanning-cycle because of error accumulation; at the ending point (*i.e.*, the end of one sonar-scanning-cycle), the pose estimation uncertainty becomes small because of the update step in the SEIF-SLAM algorithm. On the other hand, due to the sonar scanning noise, the features uncertainty in Figure 14 seems to not follow any rules. In fact, when some of the features are revisited, their uncertainty will become gradually smaller relative to other features. In general, the more accurate the pose estimation is and the more times the feature is re-observed, then the smaller the feature uncertainty is.

Figure14. The uncertainty of vehicle pose and environment features (area A).



The sparsity of information matrix is the key to the lower computational complexity of SEIF-SLAM. Figure 15 shows the statistical results of the information matrix elements.

Figure 15(a) is the statistical result of the elements magnitude of the information matrix. We can find that the overwhelming majority of information matrix elements is close to zero while only a small number of them are far from zero. Figure 15(b) shows where these elements locate in the information matrix and the blue dot represents the non-zero elements which mainly locate in the diagonal and the two ends of the counter-diagonal of the information matrix. The diagonal elements are the respective variance of the pose and features while the counter-diagonal is the covariance of the last observed features, the links of which are not yet marginalized and approximated to zero.

To compare the performance of computational efficiency, 50 Monte Carlo experiments are carried out. Figures 16 and 17 present the comparisons of average CPU time and memory usage between EKF-SLAM and SEIF-SLAM.

Figure 15. The statistical results of information matrix. **(a)** Statistical result of the elements magnitude of the information matrix; **(b)** Location of elements in the information matrix.

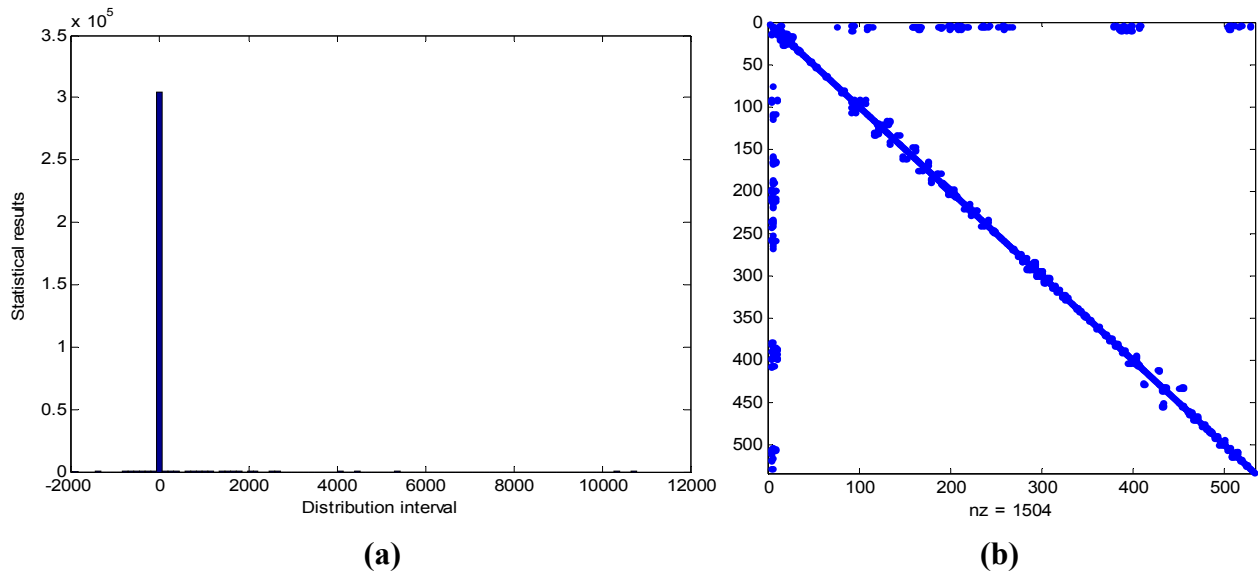
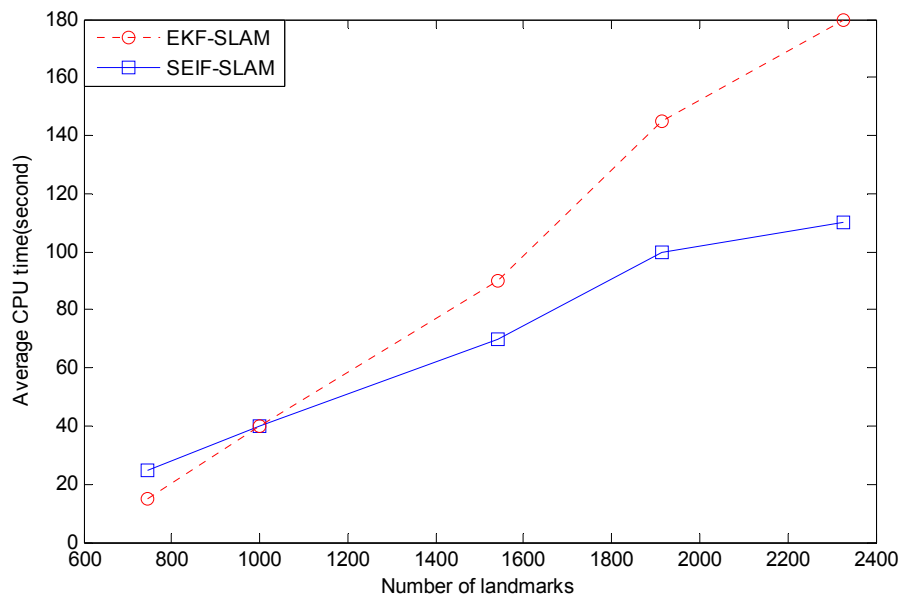
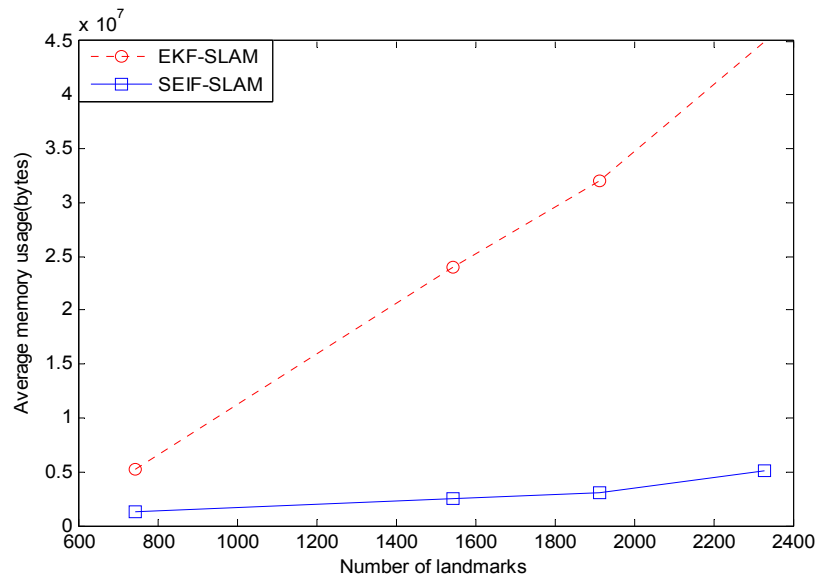


Figure 16. The comparison of average CPU time between SEIF-SLAM and EKF-SLAM.



As we can see that SEIF-SLAM is less efficient than EKF-SLAM when there are less than 1,000 features in the map. The reason for this is that when there are only few features, the impact brought by computation in sparsification step is much greater than that brought by the sparsity property. When the number of features is more than 1,000, SEIF-SLAM is more efficient than EKF-SLAM. Figure 17 demonstrates the difference between SEIF-SLAM and EKF-SLAM in storage. We can conclude that SEIF-SLAM need much less storage than that of EKF-SLAM and the gap increases when the number of features increases. In fact, it is because that SEIF-SLAM maintains information matrix which is sparse, which is far superior to the non-sparse matrix in storage.

Figure 17. The comparison of average memory usage between SEIF-SLAM and EKF-SLAM.



6. Discussion

Like EKF-SLAM, the consistency of the SEIF-SLAM algorithm is not perfect. There are two reasons for this: on the one hand, error accumulation caused by nonlinear model linearization will easily result in inconsistency, and this problem is the same as the EKF algorithm and exists in most of the SLAM algorithms based on the Gaussian linear filters; On the other hand, the sparsification, which marginalized out the weak links (*i.e.*, the links between vehicle pose and further features), could affect consistency though it has been proved that SEIF-SLAM has relatively good consistency [26,28]. The ESEIF-SLAM algorithm [26] improves the consistency through the modification of the sparsification steps of the SEIF algorithm, but increases a little computation. The more efforts will be made to improve the consistency and reduce computational cost of the information-filter-based SLAM algorithm in our future work.

Secondly, the mechanical-scanning sonar used on the *C-Ranger* usually has a quite low scanning rate, it needs about 27 seconds to scan a 360° sector. As a result, SEIF-SLAM filter updates happen approximately 13.5 meters apart, and the frequency of measurement update is very slow. Obviously the main reason is the usage of the mechanical sonar. Therefore, the mechanical sonar will be exchanged for a multi-beam sonar to obtain a high update frequency on the next generation platform. High update frequency should be helpful to improve the accuracy of navigation for AUV a lot.

7. Conclusions

In this paper, the autonomous navigation method for the *C-Ranger* AUV had been addressed. To verify the advantages of SEIF-SLAM, the application of SEIF-SLAM for AUV navigation had been studied, the approach for data processing of sonar had also been presented in this work. The mechanical scanning imaging sonar is chosen as the principal sensor for active sensing of the undersea obstacles, a compensation method based on feedback of the AUV pose has been proposed to overcome

distortions of the acoustic images due to the vehicle motion. Sea trial experiments in Tuandao Bay have been conducted to verify the feasibility of the proposed navigation methods for the *C-Ranger*. The experimental results and analysis demonstrated that the proposed navigation approach based on SEIF-SLAM improved the accuracy of navigation compared with conventional methods; moreover the algorithm achieves low computational cost when compared with EKF-SLAM.

Acknowledgements

This work is partially supported by the Natural Science Foundation of China (41176076, 51075377), the High Technology Research and Development Program of China (2006AA09Z231), the Science and Technology Development Program of Shandong Province (2008GG1055011, BS2009HZ006) and Zhejiang Natural Science Foundation (R1100015), Open foundation (GZKF-201017) of State Key Lab of Fluid Power Transmission and Control in Zhejiang University.

Reference

1. Miller, P.A.; Farrell, J.A.; Zhao, Y.; Djapic, V. Autonomous underwater vehicle navigation. *IEEE J. Ocean. Eng.* **2010**, *35*, 663-678.
2. Leonard, J.; Bennett, A.; Smith, C. Autonomous underwater vehicle navigation. *IEEE J. Ocean. Eng.* **1998**, *35*, 663-678.
3. Allen, B.; Stokey, R.; Austin, T.; Forrester, N.; Goldsborough, R.; Purcell, M.; von Alt, C. REMUS: A Small, Low Cost AUV; System Description, Field Trials and Performance Results. In *Proceedings of the Oceans Conference*, Halifax, NS, Canada, 6–9 October 1997; pp. 994-1000.
4. Bovio, E.; Cecchi, D.; Baralli, F. Autonomous underwater vehicles for scientific and naval operations. *Ann. Rev. Control* **2006**, *30*, 117-130.
5. Kinsey, J.; Eustice, R.; Whitcomb, L. A Survey of Underwater Vehicle Navigation: Recent Advances and New Challenges. In *Proceedings of the IFAC Conference of Manoeuvring and Control of Marine Craft*, Lisbon, Portugal, 20–22 September 2006.
6. Folkesson, J.; Leonard, J. Autonomy through SLAM for an Underwater Robot. In *Proceedings of the 14th International Symposium on Robotics Research (ISRR)*, Lucerne, Switzerland, 31 August–1 September 2009.
7. Durrant-Whyte, H. Uncertain geometry in robotics. *IEEE Trans. Robot. Autom.* **1988**, *4*, 23-31.
8. Smith, R.; Cheesman, P. On the representation of spatial uncertainty. *Int. J. Robot. Res.* **1987**, *5*, 56-68.
9. Durrant-Whyte, H.; Bailey, T. Simultaneous localization and mapping (SLAM) Part I: The essential algorithms. *Robot. Autom. Mag.* **2006**, *13*, 99-110.
10. Dissanayake, G.; Newman, P.; Durrant-Whyte, H.; Clark, S.; Csobra, M. A solution to the simultaneous localisation and mapping (SLAM) problem. *IEEE Trans. Robot. Autom.* **2001**, *17*, 229-241.
11. Cheein, F.A.; Carelli, R. Analysis of different feature selection criteria based on a covariance convergence perspective for a SLAM algorithm. *Sensors* **2011**, *11*, 62-89.

12. Williams, S.; Dissanayake, G.; Durrant-Whyte, H. An Efficient Approach to the Simultaneous Localisation and Mapping Problem. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Washington, DC, USA, 11–15 May 2002, pp. 406-411.
13. Leonard, J.; Newman, P. Consistent, Convergent, and Constant-Time SLAM. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexico, 9–15 August 2003; pp. 1143-1150.
14. Bosse, M.; Newman, P.; Leonard, J.; Teller, S. Simultaneous localization and map building in large-scale cyclic environments using the atlas framework. *Int. J. Robot. Res.* **2004**, *23*, 1113-1139.
15. Huang, S.; Wang, Z.; Dissanayake, G. Sparse local submap joining filters for building large-scale maps. *IEEE Trans. Robot.* **2008**, *24*, 1121-1130.
16. Montemerlo, M.; Thrun, S. FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem. In *Proceedings of the Eighteen National Conference on Artificial Intelligence*, Edmonton, AB, Canada, 28 July–1 August 2002; pp. 593-598.
17. Montemerlo, M.; Thrun, S.; Koller, D.; Wegbreit, B. Fast-SLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges. In *Proceedings of International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, 9–15 August 2003; pp. 1151-1156.
18. Montemerlo, M.; Thrun, S. *FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics*, Springer Tracts in Advanced Robotics; Springer: Berlin, Germany, 2007; Volume 27.
19. Thrun, S.; Liu, Y.; Koller, D. Simultaneous localization and mapping with sparse extended information filters. *Int. J. Robot. Res.* **2004**, *23*, 693-716.
20. Frese, U. A Proof for the Approximate Sparsity of SLAM Information Matrices. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, Spain, 18–22 April 2005; pp. 329-335.
21. Thrun, S.; Liu, Y.; Koller, D. *Simultaneous Localization and Mapping with Sparse Extended Information Filters: Theory and Initial Results*, Springer Tracts in Advanced Robotics; Springer: Berlin, Germany, 2004; Volume 7, pp. 363-380.
22. Liu, Y.; Thrun, S. Results for Outdoor-SLAM Using Sparse Extended Information Filters. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, 14–19 September 2003; pp. 1227-1233.
23. Paskin, M. *Thin Junction Tree Filters for Simultaneous Localization and Mapping*, Technical Report UCB/CSD-02-1198; University of California, Berkeley: Berkeley, CA, USA, 2002.
24. Frese, U. Treemap: An $O(\log n)$ algorithm for simultaneous localization and mapping. *Auton. Robots* **2005**, *21*, 103-122.
25. Eustice, R.; Singh, H.; Leonard, J. Exactly Sparse Delayed-State Filters. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 18–22 April 2005; pp. 2417-2424.
26. Walter, M.; Eustice, R.; Leonard, J. Exactly sparse extended information filters for feature-based SLAM. *Int. J. Robot. Res.* **2007**, *26*, 335-359.

27. Smith, R.; Self, M.; Cheeseman, P. *Estimating Uncertain Spatial Relationship in Robotics. Autonomous Robot Vehicle*; Cox, I., Wilfong, G., Eds.; Springer-Verlag: Berlin, Germany, 1990; pp. 327-346.
28. Eustice, R.; Walter, M.; Leonard, J. Sparse Extended Information Filters: Insights into Sparsification. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and System*, Edmonton, AB, Canada, 2–6 August 2005; pp. 3281-3288.
29. Johannsson, H.; Kaess, M.; Englo, B.; Hover, F.; Leonard, J. Imaging Sonar-Aided Navigation for Autonomous Underwater Harbor Surveillance. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 18–22 October 2010; pp. 4396-4403.
30. Ribas, D.; Ridao, P.; Tardos, J. *Underwater SLAM for Structured Environments Using an Imaging Sonar*, 1st ed.; Springer: Berlin, Germany, 2010.
31. Choi, J.; Ahn, S.; Chung, W. Robust Sonar Feature Detection for the SLAM of Mobile Robot. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, AB, Canada, 2–6 August 2005; pp. 3415-3420.
32. Ribas, D. Abandoned Marina Dataset-Experiment Performed at the Fluvia Nautic Abandoned Marina Near St. Pere Pescador (Spain), 2007. Available online: [http://cres.usc.edu/radish repository/view-one.php?name=abandoned_marina](http://cres.usc.edu/radish/repository/view-one.php?name=abandoned_marina) (accessed on 21 November 2011).

Appendix

The Duality of Gaussian Distribution

Let ξ_t be a multi-dimensional random variable which subjects to Gaussian distribution $p(\xi_t) \sim N(\mu_t, \Sigma_t)$, where μ_t and Σ_t denote the mean and covariance. This representation is often called the covariance form of Gaussian distribution. Expanding this representation in the exponential form, the following equivalent representation can be reached:

$$\begin{aligned}
 p(\xi_t) &= N(\mu_t, \Sigma_t) \\
 &= \frac{1}{\sqrt{|2\pi\Sigma_t|}} \exp\left\{-\frac{1}{2}(\xi_t - \mu_t)^T \Sigma_t^{-1}(\xi_t - \mu_t)\right\} \propto \exp\left\{-\frac{1}{2}(\xi_t - \mu_t)^T \Sigma_t^{-1}(\xi_t - \mu_t)\right\} \\
 &= \exp\left\{-\frac{1}{2}(\xi_t^T \Sigma_t^{-1} \xi_t - 2\mu_t^T \Sigma_t^{-1} \xi_t + \mu_t^T \Sigma_t^{-1} \mu_t)\right\} \propto \exp\left\{-\frac{1}{2}\xi_t^T \Sigma_t^{-1} \xi_t + \mu_t^T \Sigma_t^{-1} \xi_t\right\} \quad (A1) \\
 &= \exp\left\{-\frac{1}{2}\xi_t^T \Lambda_t \xi_t + \eta_t^T \xi_t\right\} \propto N^{-1}(\eta_t, \Lambda_t)
 \end{aligned}$$

The representation $p(\xi_t) \propto N^{-1}(\eta_t, \Lambda_t)$ is the named information form of Gaussian distribution, and is parameterized by the information matrix Λ_t and information vector η_t . The equivalence of the above two forms is called the duality of Gaussian distribution, that is:

$$\begin{aligned}
 \Lambda_t &= \Sigma_t^{-1}, \eta_t = \Sigma_t^{-1} \mu_t \\
 \text{or} & \\
 \Sigma_t &= \Lambda_t^{-1}, \mu_t = \Lambda_t^{-1} \eta_t
 \end{aligned} \quad (A2)$$

Equation (A3) shows these two representations about the random variable ξ_t that comprises two sub-vector α and β . The mean, covariance, information matrix and information vector are partitioned into their corresponding parts:

$$p(\xi_t) = p(\alpha, \beta) = N\left(\begin{bmatrix} \mu_\alpha \\ \mu_\beta \end{bmatrix}, \begin{bmatrix} \Sigma_{\alpha\alpha} & \Sigma_{\alpha\beta} \\ \Sigma_{\beta\alpha} & \Sigma_{\beta\beta} \end{bmatrix}\right) = N^{-1}\left(\begin{bmatrix} \eta_\alpha \\ \eta_\beta \end{bmatrix}, \begin{bmatrix} \Lambda_{\alpha\alpha} & \Lambda_{\alpha\beta} \\ \Lambda_{\beta\alpha} & \Lambda_{\beta\beta} \end{bmatrix}\right) \quad (\text{A3})$$

It is noteworthy that these two equivalent representations lead to very different computational characteristics with respect to calculating the probability of marginalization and conditioning. Table A1 summarizes this difference where we can see the marginalization is relatively easy in covariance form while hard in information form and the opposite relation holds true for conditioning.

Table A1. The duality of Gaussian distribution.

	Marginalization $p(\alpha) = \int p(\alpha, \beta) d\beta$	Conditioning $p(\alpha \beta) = p(\alpha, \beta) / p(\beta)$
Covariance Form	$\mu = \mu_\alpha$ $\Sigma = \Sigma_{\alpha\alpha}$	$\mu' = \mu_\alpha + \Sigma_{\alpha\beta} \Sigma_{\beta\beta}^{-1} (\beta - \mu_\beta)$ $\Sigma' = \Sigma_{\alpha\alpha} - \Sigma_{\alpha\beta} \Sigma_{\beta\beta}^{-1} \Sigma_{\beta\alpha}$
Information Form	$\eta = \eta_\alpha - \Lambda_{\alpha\beta} \Lambda_{\beta\beta}^{-1} \eta_\beta$ $\Lambda = \Lambda_{\alpha\alpha} \Lambda_{\beta\beta}^{-1} \Lambda_{\beta\alpha}$	$\eta' = \eta_\alpha - \Lambda_{\alpha\beta} \beta$ $\Lambda' = \Lambda_{\alpha\alpha}$

© 2011 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/3.0/>).