Jurnal Teknologi

Full paper

A Review: Simultaneous Localization and Mapping Algorithms

Saif Eddine Hadji[a*], Suhail Kazi[a], Tang Howe Hing[a], Mohamed Sultan Mohamed Ali[b]

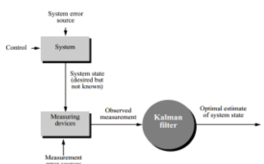
[a]Faculty of Mechanical Engineering, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia
[b]Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia

*Corresponding author: hadji.saif.eddine@gmail.com

Article history

Received :10 December 2014
Received in revised form :
1 February 2015
Accepted :12 February 2015

Graphical abstract



Abstract

Simultaneous Localization and Mapping (SLAM) involves creating an environmental map based on sensor data, while concurrently keeping track of the robot's current position. Efficient and accurate SLAM is crucial for any mobile robot to perform robust navigation. It is also the keystone for higher-level tasks such as path planning and autonomous navigation. The past two decades have seen rapid and exciting progress in solving the SLAM problem together with many compelling implementations of SLAM methods. In this paper, we will review the two common families of SLAM algorithms: Kalman filter with its variations and particle filters. This article complements other surveys in this field by reviewing the representative algorithms and the state-of-the-art in each family. It clearly identifies the inherent relationship between the state estimation via the KF versus PF techniques, all of which are derivations of Bayes rule.

Keywords: SLAM; localization; mapping; Kalman filter; particle filtre

1.0 INTRODUCTION

Autonomous mobile robot is an intelligent agent which can explores and navigates in an unknown environment with less human control. Building a map of the surrounding environment is essential for the robot navigation. Possessing the spatial model of the environment (map), containing information location of landmarks and obstacles, enables the robot to estimate its pose, to plan its path and avoid collisions. On the other hand, if the robot pose is provided along with its trajectory, the map can be easily constructed through the information coming from robot sensors [1] Unfortunately, in many applications of practical relevance (e.g. exploration tasks or operations in hostile environments), the certain map is not available. In these cases, the autonomous agent must build a map of the surroundings. Hence, the simultaneous localization and mapping problem, known as SLAM, requires if it is possible for a mobile robot placed in an unknown environment to incrementally build a consistent map while simultaneously determining its location within this map. Dynamic objects, however, can lead to serious errors in the resulting maps such as spurious objects or misalignments due to localization errors [2]. The concept of simultaneous localization and mapping has attracted extensive interest in the mobile robotics literature and many stochastic SLAM frameworks have been developed so far.

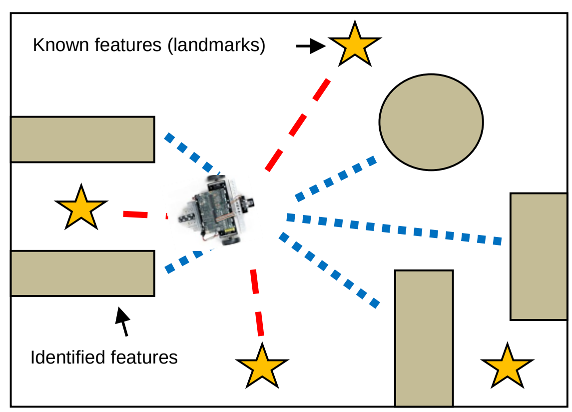


Figure 1 Illustration of feature-based SLAM

2.0 BACKGROUND

Unmanned mobile robots exist in many different shapes and sizes with varying degrees of intelligence and capability. Unmanned ground vehicles can operate on rough and rugged terrain, inside of buildings where hostile conditions may exist, and in constricted spaces that would otherwise be inaccessible for humans. Unmanned underwater vehicles can be deployed by the military to sneak undetected under the surface if necessary, can be used in search operations for missing planes or boats in oceans (recently used for MH 370), or can be used by scientists to analyze the ocean floor mapping or drilling purposes. Unmanned aerial vehicles range from models with one foot wingspans that can be used for urban search and rescue, to the full-sized Predators deployed for reconnaissance and precision strikes in Iraq and Afghanistan [3]. Fully autonomous vehicles require significant cognitive abilities. Given only some tasks to perform, the robot must localize itself, put together a representation of its surroundings, plan a course of action through its surroundings to achieve its goal, and then act upon this plan. The problem of localization requires the robot to determine its pose and its destination is in a particular reference frame. One possible solution is the process known as simultaneous localization and mapping, or SLAM [4]. SLAM involves creating an environmental map based on sensor data, while concurrently keeping track of the robot's current position. Another common approach is through the use of the Global Positioning System satellite network coupled with an inertial navigation system which can track the motion of the robot in the absence of accurate data from the satellites.

Since its early beginnings [5], [6], the SLAM scheme has undergone several developments and optimizations. The most frequent implementation uses an Extended Kalman Filter (EKF) [7], [8]. The principal of EKF is the minimization of the mean quadratic error of the system state and considers all variables as Gaussian random variables [6], [9]. The map obtained by an EKF-based SLAM implementation is usually a feature-based map [10], [11], this type of methods is well known as feature-based SLAM as shown in Figure 1. In [12], a better performance of SLAM scheme is given by a SLAM approach based on the Unscented Kalman Filter, considering the non-linearity of the model of the robot and the model of the features. However, these variants of KF are relatively slow when dealing with huge number of landmarks due to the general update at every single measurement. Other approaches use a Particle Filter, [12], [13], to solve the SLAM problem. The advantage of Particle Filter SLAM implementation is that the features of the map are not restricted to be Gaussian. Many PF algorithm have been developed such as, FastSLAM and FastSLAM 2.0 [14]. Nevertheless, these filters suffer degeneration due to their inability to forget the past which conduct to loss in accuracy. The classification of a SLAM algorithm as the best one for a particular environment depends on hardware limitations, the size of the environment to be modeled by the robot and the optimization criterion of the processing time.

However, it seems that almost none of the current approaches can perform consistent maps for large areas, mainly due to the increase on computational cost and on the uncertainties. Therefore this is possibly the most important factor that needs to be improved. Some recent publications solve the problem by using multiple maps or sub-maps that are lately used to build a larger global map [15]–[18]. However these methods rely considerably on assuming proper data association, which is another important issue that needs to be improved.

3.0 BAYESIAN RECURSIVE ESTIMATION

All probabilistic SLAM algorithms are derived from the recursive Bayes rule

$$p(x_k|\, z^k)p(\, z^k) = p(\, z^k|x_k)p(x_k) \quad (1)$$

Where x_k is the state vector including the robot pose and of environment landmarks at time k. As the robot moves through its environment, it observes nearby landmarks, $z^k = \{z_i, i = 1, \ldots, k\}$ is a set of measurements from time 1 to k, where z_k is a measurement by robot sensor at time k, which is used to estimate the state x_k:

$$z_k = h(x_k) \quad (2)$$

where h is a possibly nonlinear function.

The process evolution of the state between time $k-1$ and k is governed by a nonlinear function$f$, such that:

$$x_k = f(x_{k-1}) \quad (3)$$

In probabilistic form, the simultaneous localization and map building problem requires that the probability distribution $p(x_k|z^k)$ be computed for all times k. This probability distribution describes the joint posterior density for landmark locations and vehicle state (at time k) given the observations up to time k. Generally, the probability distribution can be obtained in a prediction–update recursion.

Consider that a posterior probability distribution $p(x_{k-1}|z^{k-1})$ is given, then the prior of the state at time k can be computed via the Chapman–Kolmogorov equation:

$$p(x_k|z^{k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|p(x_{k-1}|z^{k-1})\,)dx_{k-1} \quad (4)$$

where the probability distribution $p(x_k|x_{k-1})$ is defined by (3). This procedure is the called prediction stage.
In the update stage, a new measurement z_k is employed to update the prior $p(x_k|z^{k-1})$ to determine the posterior $p(x_k|z^k)$ via the conditional Bayes rule by rewriting (1):

$$p(x_k|z^k) = p(x_k|z_k, z^{k-1}) = p(z_k|x_k, z^{k-1})p(x_k|z^{k-1})/p(z_k|z^{k-1}) \quad (5)$$

By knowing the stat x_k, no past measurement would provide us additional information. In mathematical term:

$$p(z_k|x_k, z^{k-1}) = p(z_k|x_k)$$

Therefore (5) is reformulated as follows:

$$p(x_k|z^k) = \eta p(z_k|x_k)\, p(x_k|z^{k-1}) \quad (7)$$

From (7), the recursive Bayesian estimator allows new information to be added simply by multiplying a prior by a current $(k-th)$ likelihood.

Thus, (4) and (7) establish the basis for the optimal Bayesian solution for SLAM. However, such a solution is a theoretical approach that cannot be practically implemented in the real-world [17]. Optimal solutions, such as Kalman Filter and Particle Filter, employing the probability distribution in two stages, will be introduced in the following section.

4.0 FILTERS FOR SLAM

The SLAM problem can be traced back to 25 years ago, where few dominant probabilistic approaches were introduced (i.e. Kalman Filters (KF), Particle Filters (PF) and 1.3. Expectation Maximization based methods (EM)). The two techniques are mathematical derivations of the recursive Bayes rule. The reason that makes these probabilistic techniques very popular is the fact that robot mapping is characterized by sensor noise and uncertainty, and the probabilistic algorithms overcome the problem by expressing the different sources of noise with their effects on the observations [20].

4.1 Kalman Filter SLAM

Kalman filters are Bayes filters that represent posteriors using Gaussians, i.e. unimodal multivariate distributions that can be represented compactly by a small number of parameters. KF SLAM relies on the assumption that the state transition and the measurement functions are linear with added Gaussian noise, and the initial posteriors are also Gaussian. Figure 2 describes the general scheme of Kalman filter estimation, where a system has a control signal and system error sources as inputs. A measuring device enables measuring some system states with errors. The Kalman filter is a mathematical mechanism for producing an optimal estimate of the system state based on the knowledge of the system and the measuring device, the description of the system noise and measurement errors and the uncertainty in the dynamics models. Thus the Kalman filter fuses sensor signals and system knowledge in an optimal way.

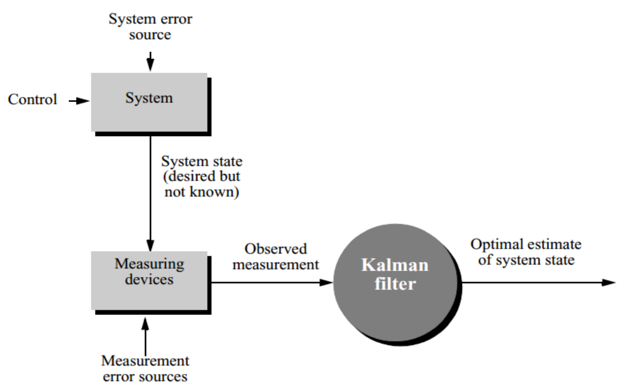


Figure 2 General scheme of Kalman filter [18]

There are two main variations of KF in the state-of-the-art SLAM: the Extended Kalman Filter (EKF) and its related Information Filtering (IF) or Extended IF (EIF). The EKF considers all variables as Gaussian random variables and minimizes the mean quadratic error of the system state [21], [22]. The map obtained by an EKF-based SLAM implementation is usually a feature-based map [23], [24]. The features of the map obey some geometrical constrain of the environment. Thus, in [25] is presented a line-based SLAM where lines are related to walls; in [24] is shown a point-based SLAM where all significant points are related to trees of the environment [24]. Several existing SLAM approaches use the EKF [22], [25], [27], [28]. The IF is implemented by propagating the inverse of the state error covariance matrix. There are several advantages of the IF filter over the KF. Firstly, the data is filtered by simply summing the information matrices and vector, providing more accurate estimates [29]. Secondly, IF are more stable than KF [29]. Finally, EKF is relatively slow when estimating high dimensional maps, because every single vehicle measurement generally affects all parameters of the Gaussian, therefore the updates requires prohibitive times when dealing with environments with many landmarks [30].

However, IF have some important limitations, a primary disadvantage is the need to recover a state estimate in the update step, when applied to nonlinear systems. This step requires the inversion of the information matrix. Further matrix inversions are required for the prediction step of the information filters. For high dimensional state spaces the need to compute all these inversions is generally believed to make the IF computationally poorer than the Kalman filter. In fact, this is one of the reasons why the EKF has been vastly more popular than the EIF [12]. These limitations do not necessarily apply to problems in which the information matrix possesses structure. In many robotics problems, the interaction of state variables is local; as a result, the information matrix may be sparse. Such sparseness does not translate to sparseness of the covariance. Information filters can be thought of as graphs, where states are connected whenever the corresponding off-diagonal element in the information matrix is non-zero. Sparse information matrices correspond to sparse graphs. Some algorithms exist to perform the basic update and estimation equations efficiently for such fields [31], in which the information matrix is (approximately) sparse, and allows developing an extended information filter that is significantly more efficient than both Kalman filters and non-sparse Information Filter.

The Unscented Kalman Filter (UKF) addresses the approximation issues of the EKF and the linearity assumptions of the KF. KF performs properly in the linear cases, and it is considered as an efficient method for analytically propagating a Gaussian Random Variable (GRV) through a linear system dynamics. For nonlinear models, the EKF approximates the optimal terms by linearizing the dynamic equations. The EKF can be viewed as a first-order approximation to the optimal solution. In these approximations the state distribution is approximated by a GRV, which then is propagated analytically through the first-order linearization of the nonlinear system. These approximations can introduce large errors in the true posterior mean and covariance, which may lead sometimes to divergence of the filter. In the UKF the state distribution is once more represented by a GRV, but is now quantified using an optimum set of carefully chosen sample points. This set of points fully seizures the true mean and covariance of the GRV, and after propagation through the non-linear system, captures the new mean and covariance accurately to the 3rd order for any nonlinearity. In order to do that, the unscented transform is used.

One of the main drawbacks of the EKF and the KF implementation is the fact that for long time executions, computer resources will not be sufficient to update the map in real-time, because of the increasing number of landmarks. This large scaling problem arises because each landmark is correlated to all other landmarks. The correlation appears since the observation of a new landmark is obtained with one of the mobile robot's sensors and therefore the error in the location of the landmark will be correlated with the error in the vehicle location and the errors in the rest of landmarks of the map. This correlation is of a crucial importance for the long-term convergence of the algorithm, and needs to be sustained for the full duration of the execution. The Compressed Extended Kalman Filter (CEKF) [32] algorithm significantly reduces the computational requirement without introducing any penalties in the accuracy of the results. A CEKF stores and maintains all the information gathered in a local area with a cost proportional to the square of the number of landmarks in the area. This information is then transferred to the rest of the global map with a cost that is similar to full SLAM but in only one iteration.

The advantage of KF and its variants is that provides optimal Minimum mean-square Error (MMSE) estimates of the state (robot and landmark positions), and its covariance matrix seems to converge strongly. However, the Gaussian noise assumption restricts the adaptability of the KF for data association and number of landmarks.

4.2 Particle Filter SLAM

The second principal SLAM paradigm is based on particle filters. Particle filters can be traced back to [33], but they have become popular only in recent years. Particle filters, also called the Sequential Monte-Carlo (SMC) method, is a recursive Bayesian filter that is implemented in Monte Carlo simulations. It executes SMC estimation by a set of random point clusters (particles) representing the Bayesian posterior. For the beginner in SLAM, each particle is best thought as an actual guess as to what the genuine estimation of the state may be. By collecting many such guesses into a set of guesses, or set of particles, the particle filters captures a representative sample from the posterior distribution. The particle filter has been shown under mild conditions to approach the true posterior as the particle set size goes to infinity. It is also a nonparametric representation that represents multimodal distributions with ease. In recent years, the advent of extremely efficient microprocessors has made particle filters a popular algorithm [34].

The key problem with the particle filter in the context of SLAM is the evolution of the computational complexity on the state dimension as new landmarks are observed, becoming not appropriate for real time applications [14]. Thus, Particle Filter has just been effectively applied to localization, i.e. determining position and orientation of the robot, but not to map-building, i.e. landmark position and orientation; therefore, there are no important papers using Particle Filter for the whole SLAM framework, but there exist few works that deal with the SLAM problem using a combination of Particle Filter with other techniques.

The trick to make particle filters amenable to the SLAM problem goes back to [35]. The trick was introduced into the SLAM literature in [36], followed by [37], who coined the name FastSLAM.

FastSLAM takes advantage of an important characteristic of the SLAM problem (with known data association): landmark estimates are conditionally independent given the robot's path [39]. FastSLAM algorithm decomposes the SLAM problem into a robot localization problem, and a collection of landmark estimation problems that are conditioned on the robot pose estimate. A key characteristic of FastSLAM is that each particle makes its own local data association. In contrast, EKF techniques must commit to a single data association hypothesis for the entire filter. In addition FastSLAM uses a particle filter to sample over robot paths, which requires less memory usage and computational time than a standard EKF or KF. Sampling over robot paths leads to efficient scaling and robust data association, however it also has its drawbacks. FastSLAM, and particle filters in general, have some unusual properties. For example, the performance of the algorithm will eventually degrade if the robot's sensor is too accurate. This problem occurs when the proposal distribution is poorly matched with the posterior. In FastSLAM, this happens when the motion of the robot is noisy relative to the observations.

FastSLAM 2.0, a further improvement to SLAM, was discussed by [36] This systems makes a more-efficient use of the particle filter principle, particularly in situations where motion noise is high relative to measurement noise. FastSLAM 2.0 is also better than other algorithms at overcoming the data association problem, which can arise when different landmarks in the environment look alike. The classic solution to the data association problem in SLAM is to select a feature on the landscape such that it maximizes the likelihood of the sensor measurement given all available data, and to align all other data based on this step. FastSLAM 2.0 solves the problem by calculating the maximum likelihood for each particle, meaning that the additional step is not needed. However statistically, FastSLAM 2.0 suffers degeneration due to its inability to forget the past. Marginalizing the map in this algorithm introduces dependence on the pose and measurement history, and so, when resampling depletes this history, statistical accuracy is lost [7]. Recently, the hierarchical RBPF SLAM, proposed in [39], is a robust SLAM framework in indoor environments with sparse and short-range sensors. In order to overcome the sensor limitations, this approach divided the entire region into several local maps, which are assumed to be independent of each other. However, these approaches have not been attempted in dynamic environments.

4.3 Expectation Maximization Based Methods (EM)

EM estimation is a statistical algorithm that was developed in the context of maximum likelihood (ML) estimation and it offers an optimal solution, being an ideal option for map-building, but not for localization. The EM algorithm is able to build a map when the robot's pose is known, for instance, by means of expectation [40]. EM iterates two steps: an expectation step (E-step), where the posterior over robot poses is calculated for a given map, and maximization step (M-step), in which the most likely map is calculated given these pose expectations. The final result is a series of increasingly accurate maps. The main advantage of EM with respect to KF is that it can tackle the correspondence problem (data association problem) surprisingly well [37]. This is possible thanks to the fact that it localizes repeatedly the robot relative to the present map in the E-step, generating various hypotheses as to where the robot might have been (different possible correspondences). In the latter M-step, these correspondences are translated into features in the map, which then get reinforced in the next E-step or gradually disappear. However, the need to process the same data several times to obtain the most likely map makes it inefficient, not incremental and not suitable for real-time applications [41]. Even using discrete approximations, when estimating the robot's pose, the cost grows exponentially with the size of the map, and the error is not bounded; hence the resulting map becomes unstable after long cycles. These problems could be avoided if the data association was known, what is the same, if the E-step was simplified or eliminated. For this reason, EM usually is combined with PF, which represents the posteriors by a set of particles (samples) that represent a guess of the pose where the robot might be. For instance, some practical applications use EM to construct the map (only the M-step), while the localization is done by different means, i.e. using PF-based localizer to estimate poses from odometer readings.

■5.0 CONCLUSION

This overview allows finding the most interesting filtering techniques and identifying many of its particularities. These filtering strategies are Kalman Filter (KF) with its variaitions (Information Filter (IF), Unscented Kalman Filter (UKF) and Compressed Kalman Filter (CKF)) and Particle Filter (PF).

The most interesting outcome from the study is that for large scenarios, or maps with high population of landmarks, the CKF seems to be better as compared to other methods. When dealing with these kinds of maps, the state vector and its associated covariance matrix keeps growing with the quantity of landmarks observed. This growth makes the mathematical operations more complex and increases dramatically the time consumption, i.e. the computational cost. The strategy used by the CKF to compute local KFs and then update its output to a global map seems really consistent, because it only needs to handle with small amounts of data during the local iteration process. Although Gaussian noise is assumed in all models presented so far, not always reflects the problems of the real world. It seems that UKF could handle with different types of noise, but this topic has not been investigated in deep yet.

References

[1] Rodrigo, R. Chen, Z. and Samarabandu, J. 2006. Feature Motion for Monocular Robot Navigation. In 2006 International Conference on Information and Automation. 201–205.

[2] Burgard, W. Stachniss, C. and Dirk, H. 2007. Mobile Robot Map Learning from Range Data in Dynamic Environments. In Autonomous Navigation in Dynamic Environments. L. Christian and R. Chatila, Eds. Springer Berlin Heidelberg. 35: 3–28.

[3] Eric, J. Thorn, L. 2009. *Autonomous Motion Planning Using A Predictive Temporal Method.* University of Florida.

[4] Smith, R. Self, M. and Cheeseman, P. 1990. Autonomous Robot Vehicles. New York, NY: Springer New York. 167–193.

[5] Ayache, N. and Faugeras, O. D. 1989. Maintaining Representations of the Environment of a Mobile Robot. *IEEE Trans. Robot. Autom.* 5(6): 804–819.

[6] Chatila, R. and Laumond, J. 1985. Position Referencing and Consistent World Modeling for Mobile Robots. In Proceedings. 1985 IEEE International Conference on Robotics and Automation. 2: 138–145.

[7] Durrant-Whyte, H. and Bailey, T. 2006. Simultaneous Localization and Mapping: Part I. *IEEE Robot. Autom. Mag.* 13(2): 99–110.

[8] Bailey, T. and Durrant-Whyte, H. 2006. Simultaneous Localization and Mapping (SLAM): Part II. *IEEE Robot. Autom. Mag.* 13(3): 108–117.

[9] Thrun, S. Burgard, W. and Fox, D. 1998. A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots. *Auton. Robots.* 5(3–4): 253–271.

[10] Xi, B. Guo, Sun, R. F. and Huang, Y. 2008. Simulation research for active Simultaneous Localization and Mapping based on Extended Kalman Filter. In 2008 IEEE International Conference on Automation and Logistics. 2443–2448.

[11] Diosi, A. and Kleeman, L. 2004. Advanced Sonar and Laser Range Finder Fusion for Simultaneous Localization and Mapping. In 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566). 2: 1854–1859.

[12] Thrun, S. 2002. Probabilistic robotics. *Commun. ACM*. 45(3): 1999–2000.

[13] Hahnel, D. Burgard, W. Fox, D. and Thrun, S. 2003. An Efficient Fastslam Algorithm for Generating Maps of Large-scale Cyclic Environments from Raw Laser Range Measurements. In Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453). 1: 206–211.

[14] Montemerlo, M. 2003. FastSLAM : A Factored Solution to the Simultaneous Localization and Mapping Problem With Unknown Data Association. Doctoral Dissertation, Tech. Report CMU-RI-TR-03-28, Robotics Institute, Carnegie Mellon University, July.

[15] Estrada, C. Neira, J. and Tardos, J. D. 2005. Hierarchical SLAM: Real-time Accurate Mapping of Large Environments. *IEEE Trans. Robot*. 21(4): 588–596.

[16] Cheng, G. 2006. Vision-based Mobile Robot Navigation : A Qualitative Approach. Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA. 2686–2692.

[17] Z. Chen, J. Samarabandu, R. Rodrigo. 2007. Recent Advances in Simultaneous Localization and Map-building Using Computer vision. In Advanced Robotics. 21(3–4): 233–265.

[18] Paz, L. M. Jensfelt, P. Tardos, J. D. and Neira, J. 2007. EKF SLAM updates in O(n) with Divide and Conquer SLAM. In Proceedings 2007 IEEE International Conference on Robotics and Automation. 1657–1663.

[19] R. Siegwart,R. Nourbakhsh. 2004. Introduction to Autonomous Mobile Robots. In Bradford Book, The MIT Press.

[20] Josep, A. Yvan, P. Joaquim, S. and Xavier, L. 2008. The SLAM Problem: A Survey. In proceeding of: Artificial Intelligence Research and Development, Proceedings of the 11th International Conference of the Catalan Association for Artificial Intelligence, CCIA 2008, October 22–24, Sant Martí d'Empúries, Spain.

[21] Williams, S. B. Newman, P. Dissanayake, G. and Durrant-Whyte, H. 2000. Autonomous Underwater Simultaneous Localisation and Map Building. In Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065). 2: 1793–1798.

[22] Leonard, P. N. J. 2003. Consistent, Convergent, and Constant-time Slam. In proceeding of: IJCAI-03, Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 9-15. 1143–1150.

[23] Eustice, R. M. Singh, H. Leonard, J. J. and Walter, M. R. 2006. Visually Mapping the RMS Titanic: Conservative Covariance Estimates for SLAM Information Filters. Int. J. Rob. Res. 25(12): 1223–1242.

[24] Pizarro, O. Eustice, R. and Singh, H. 2004. Large Area 3d Reconstructions from Underwater Surveys. In Oceans ’04 MTS/IEEE Techno-Ocean '04 (IEEE Cat. No.04CH37600). 2: 678–687.

[25] Jensfelt, P. Kragic, D. Folkesson, J. and Bjorkman, M. 2006. A Framework for Vision Based Bearing Only 3D SLAM. In Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006. 1944–1950.

[26] Cheein, F. A. A. Lopez, N. Soria, C. di Sciascio, M. F. A. Pereira, F. L. and Carelli, R. 2010. SLAM Algorithm Applied to Robotics Assistance for Navigation in Unknown Environments. *J. Neuroeng. Rehabil*. 7(1): 10.

[27] Davison, A. J. and Murray, D. W. 2002. Simultaneous Localization and Map-Building Using Active Vision. *IEEE Trans. Pattern Anal. Mach. Intell*. 24(7): 865–880.

[28] Se, S. Lowe, D. and Little, J. 2002. Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks. *Int. J. Rob. Res.* 21(8): 735–758.

[29] Thrun, S. and Liu, Y. 2005. Multi-robot SLAM with Sparse Extended Information Filers. Springer Tracts in Advanced Robotics Robotics Research. The Eleventh International Symposium. 15: 254–266.

[30] Thrun, S. Martin, C. Liu, Y. Hahnel, D. Emery-Montemerlo, R. Chakrabarti, D. and Burgard, W. 2004. A Real-Time Expectation-Maximization Algorithm for Acquiring Multiplanar Maps of Indoor Environments With Mobile Robots. *IEEE Trans. Robot. Autom*. 20(3): 433–442.

[31] Matthew Walter, J. L.and Eustice, R. “A Provably Consistent Method for Imposing Sparsity in Feature-Based SLAM Information Filters,” Springer Tracts Adv. Robot., vol. 28, no. 2007, p. pp 214–234, 2007.

[32] Guivant, J. E. and Nebot, E. M. 2001. Optimization of the Simultaneous Localization and Map-building Algorithm for Real-time Implementation. *IEEE Trans. Robot. Autom.* 17(3): 242–257.

[33] Metropolis, N. and Ulam, S. 1949. The Monte Carlo Method. Journal of the American Statistical Association. 44(247): 335–341.

[34] Pitt, M. K. and Shephard, N. 1999. Filtering via Simulation: Auxiliary Particle Filters. *J. Am. Stat. Assoc.* 94(446): 590–599.

[35] Blackwell, D. 1947. Conditional Expectation and Unbiased Sequential Estimation. *Ann. Math. Stat.* 18(1): 105–110.

[36] Doucet, A. de Freitas, N. Murphy, K. P. and Russell, S. J. 2000. Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks. 176–183.

[37] Thrun, S. Hahnel, D. Ferguson, D. Montemerlo, M. Triebel, R. Burgard, W. Baker, C. Omohundro, Z. Thayer, S. and Whittaker, W. 2003. A System for Volumetric Robotic Mapping of Abandoned Mines. In 2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422). 3: 4270–4275.

[38] Montemerlo, M. Thrun, S. and Siciliano, B. 2007. FastSLAM: A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics. Springer Tracts in Advanced Robotics. 27.

[39] Oh, S. Hahn, M. and J. Kim, 2013. Simultaneous Localization and Mapping for Mobile Robots in Dynamic Environments. 2013 Int. Conf. Inf. Sci. Appl. 1–4.

[40] Wolfram Burgard, D. F. 1999. Sonar-Based Mapping of Large-Scale Mobile Robot Environments using EM'. In ICML '99 Proceedings of the Sixteenth International Conference on Machine Learning. 67–76.

[41] Anon. 2004. Real-Time 3D SLAM with Wide-Angle Vision. In: 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles.