Invited Review Article

INTERNATIONAL JOURNAL OF ADVANCED ROBOTIC SYSTEMS

International Journal of Advanced Robotic Systems
September-October 2016: 1–13

DOI: 10.1177/1729881416669482
arx.sagepub.com
SAGE

A critique of current developments in simultaneous localization and mapping

Shoudong Huang and Gamini Dissanayake

Abstract

The number of research publications dealing with the simultaneous localization and mapping problem has grown significantly over the past 15 years. Many fundamental and practical aspects of simultaneous localization and mapping have been addressed, and some efficient algorithms and practical solutions have been demonstrated. The aim of this paper is to provide a critical review of current theoretical understanding of the fundamental properties of the SLAM problem, such as observability, convergence, achievable accuracy and consistency. Recent research outcomes associated with these topics are briefly discussed together with potential future research directions.

Keywords

Consistency, convergence, extended Kalman filter, graph, least squares, observability, optimization, SLAM, sparsity

Date received: 8 April 2015; accepted: 23 August 2016

Topic: Mobile Robots and Multi-Robot Systems
Topic Editor: Pablo Gonzalez-de-Santos
Associate Editor: Jesus Capitán Fernández

Introduction

The simultaneous localization and mapping (SLAM) problem asks whether it is possible for a robot to be able to build a map of an unknown environment in real-time and simultaneously work out its own location within the map, using information gathered from sensors mounted on the robot. Reliable solutions to SLAM underpin successful robot deployment in many application domains, especially when an external location reference such as a global positioning system (GPS) is not available. SLAM applications include environment reconstruction, urban search and rescue, underground mining, underwater surveillance and planetary exploration. Solutions to SLAM aim to provide an estimate of the robot location and a geometric representation of the environment, as measurements are gathered from a sensor as the robot moves through the environment. An accurate prediction of the uncertainties associated with these estimates is also required in the context of robot navigation, where the reliability of the information available plays an important role in the decision making process.

In situations where it is possible to process information from the sensors to obtain the locations of geometric primitives, or features present, such as points, lines and planes, the map of the environment can be described using a collection of such features. Dissanayake et al. demonstrated that a Kalman filter based SLAM algorithm, where the state vector consists of the latest robot pose (position and orientation) and coordinates of a set of points that describe the environment, can converge to the true solution as the robot continues to gather information about its environment.[1] Three essential convergence properties of the algorithm under the assumption of linear motion and observation models were proven. In a practical scenario, both motion and observation models are nonlinear. Therefore, an extended Kalman filter (EKF) that relies on a first-order linear approximation is required to solve the SLAM problem. Also, given that the only information available to solve SLAM are

University of Technology, Sydney, Australia

Corresponding author:
Shoudong Huang, Faculty of Engineering and IT, University of Technology, Sydney, NSW 2007, Australia.
Email: shoudong.huang@uts.edu.au

relationships either between two robot poses or between a robot pose and the observed feature locations, the initial pose of the robot with respect to an arbitrary global coordinate frame is not observable. This issue is typically addressed by placing the origin of the global coordinate frame at the robot initial pose. While the EKF formulations have been successfully implemented on robots equipped with mm-wave radars,[1] laser range finders,[2] and monocular cameras,[3] a number of researchers reported the fact that under some situations the EKF-based SLAM algorithm becomes inconsistent leading to overconfident estimates.[4–6]

Huang and Dissanayake provided a theoretical analysis of this phenomena, illustrating that the inconsistency is a result of the fact that the Jacobians of the motion and measurement models are not evaluated at the true state, which is of course not feasible in a practical implementation.[7] Alternatively, when a SLAM state vector is defined relative to a coordinate frame attached to the latest robot pose (called robocentric mapping[5]), the errors due to linearization become smaller and as a consequence the state estimates obtained using the EKF are more accurate. Recently, a globally asymptotically stable EKF based SLAM algorithm was also reported by Bishop and Jensfelt, and Guerreiro et al., under the assumption that no feature disappears from the sensor field of view.[8,9]

When the robot pose relative to an arbitrary global coordinate frame is included in the state vector of the EKF, observability of the SLAM problem can be analyzed from the control theoretic point of view. It was shown by Huang et al.[10] that the inconsistency in EKF SLAM is closely related to the partial observability of the SLAM problem.[11,12] This insight resulted in a number of observability-constrained EKF SLAM algorithms which significantly improve the SLAM consistency.[13]

SLAM can also be formulated as a parameter estimation problem where all the robot poses from where the measurements were taken together with all the observed features are treated as a set of unknown constant parameters that need to be estimated. An objective function based on maximum likelihood of the parameter estimates can be formulated by using robot odometry measurements and robot-to-feature observation information to relate the unknown parameters.[14] The optimization based SLAM methods improve the consistency of SLAM as Jacobians are repeatedly computed at the most recent parameter estimate, and in the limit at the optimal state as the solution converges. Exploiting the sparse structure of the optimization problem leads to very efficient solutions,[15] despite the increase in the size of the state vector as all robot poses from which measurements were taken now need to be estimated. Current algorithms that use the optimization framework can efficiently solve SLAM problems consisting of a few thousand of robot poses and a few million robot-to-feature observations.[16]

Another possibility is to formulate the problem as that of estimating all the robot poses from where the measurements were taken. This is known as pose-graph SLAM,[17,18] which also has a sparse structure and can be solved efficiently.[16] In pose-graph SLAM, the sensor observations are used to obtain an estimate of the relationship between the robot poses from which these observations were taken, without using an explicit geometric representation of the environment. Such relative pose estimates can be obtained using a scan or image registration algorithm for laser range finders and cameras respectively, provided that the situations where the sensors observe the same region of the environment can be recognized. This is relatively easy when the time between observations is relatively small, but when the robot visits previously observed regions after a long traverse, special loop-closure identification techniques are required.[19] A map of the environment is reconstructed once the estimates of all the robot poses are available.

Optimization based SLAM methods (for both feature-based SLAM and pose-graph SLAM) have been observed to converge to the global minimum under many situations, in contrast to the usual behaviour of typical nonlinear optimization problems where convergence to local minima can only be avoided by providing a relatively good initial guess to the solution. Some special structures present in point feature based SLAM and pose-graph SLAM have been recently discovered by Wang et al. and Carlone et al., accounting for this suprisingly good behaviour.[20–22] Research along this direction has resulted in the development of new algorithms for optimization based SLAM.[22,23] The authors are of the view that further research on the structure of the SLAM problem has the potential to result in more robust SLAM algorithms.

The aim of this paper is to review and consolidate the research on the fundamental properties of SLAM and discuss some of the future challenges. The focus of this paper is on the theoretical aspects of SLAM such as the observability of different SLAM formulations and the convergence, achievable accuracy and consistency of different SLAM algorithms. It is shown that although many algorithms to solve SLAM effectively are available, the theoretical understanding of this important problem is still incomplete.

This paper is a significant extension of an earlier conference paper and focuses on feature based SLAM and pose-graph SLAM using estimation and optimization techniques.[24] For a thorough review of SLAM research work prior to 2006, readers are referred to works by Durrant-Whyte and Bailey;[25,26] for a practical review from the users' perspective, see Frese et al.;[27] for a recent review that covers other variants of SLAM problems such as SLAM using particle filter and appearance-based SLAM, see Zamora and Yu.[28]

Typical SLAM implementations consist of two distinct phases. First, the information from sensors are pre-processed to identify potentially spurious readings as well as to associate sensor readings with the parts of the environment that have been previously observed. This is known as the front-end, which is specific to the sensors used and the application domain. The SLAM back-end is where estimation or optimization algorithms are used to provide robot location estimates and a representation of the environment based on the strategies described in the previous paragraphs. Much of the discussion in this article relates to the SLAM back-end. Therefore, important front-end issues such as data association and identifying loop closures are mostly ignored.

This paper is organized as follows. Section "The statement of the SLAM problem" provides the statement of the SLAM problem. Section "Fundamental properties of SLAM" examines basic properties of SLAM problems including observability of SLAM problem and convergence of some SLAM algorithms. Some criteria for evaluating SLAM algorithms are discussed in Section "Criteria for evaluating the performance of SLAM algorithms". Section "Other recent developments in SLAM" briefly discusses various recent developments in SLAM research. In Section "Future directions in SLAM research", some possible future research topics in SLAM are proposed. Finally Section "Conclusion" concludes the paper.

The statement of the SLAM problem

SLAM is the process of building a map of an unknown environment while concurrently generating an estimate of the location of an autonomous robot (a moving sensor). In the most fundamental form of the SLAM problem, the only information available is from sensors on-board the robot that observe the robot ego-motion and the environment.

Feature based SLAM problem

In feature based SLAM, the environment is assumed to consist of "features" that are stationary. The robot pose (including position and orientation) at time k is denoted by $X_r(k)$, while the parameters used to describe features (e.g. the Cartesian coordinates of a point feature) are denoted by $X_f(k) = X_f$.

A process model that relates the robot pose at time $k+1$ with the robot pose at time k is usually described by

$$X_r(k+1) = f(X_r(k), u(k), v(k)) \tag{1}$$

where $u(k)$ is the control at time k and $v(k)$ is the process noise at time k.

The observation at time k is a function of $X_r(k)$ and (part of) X_f and is given by

$$z(k) = h_k(X_r(k), X_f, w(k)) \tag{2}$$

where $w(k)$ is the observation noise at time k. Note that the observation function is in general time-varying since only a (different) subset of features are observed at any given time step.

The feature based SLAM problem aims to obtain an estimate of feature parameters X_f and robot poses $X_r(0), \ldots, X_r(k)$ using the information gathered from the sensors up to time k.

Extended Kalman filter based solution. In the conventional formulation of the EKF based SLAM solution[1], the state vector consists of the most recent robot pose $X_r(k)$ and the feature parameters X_f

$$X(k)^{\text{EKF}} = \begin{bmatrix} X_r(k) \\ X_f(k) \end{bmatrix} = \begin{bmatrix} X_r(k) \\ X_f \end{bmatrix} \tag{3}$$

The initial robot pose $X_r(0)$, the robot pose where the first observation is made, is assumed to be the origin of the coordinate frame in which the state vector is defined. The feature parameters X_f are therefore defined relative to $X_r(0)$ and are time invariant. An EKF or an extended information filter (EIF) formulated with the process model (1), and observation models (2) can be used to obtain an estimate of the state vector and the associated covariance matrix. The nonlinearity of the process and observation model, the fact that the observation model is time varying, together with the noise presented in the observation model, can cause the linearization based EKF/EIF SLAM algorithm to perform poorly unless previously mapped regions of the environment are revisited at frequent intervals.

In an alternative formulation known as robocentric EKF,[5] the state vector is comprised of the feature parameters and the initial robot pose, all with respect to the most recent robot pose $X_r(k)$, that is

$$X(k)^{\text{Rob}} = \begin{bmatrix} X_r(0)^{\text{Rob}} \\ X_f(k)^{\text{Rob}} \end{bmatrix} \tag{4}$$

In this formulation the whole state vector $X(k)^{\text{Rob}}$ is time varying (because the coordinate frame keeps changing) although the environment is stationary with respect to a fixed global coordinate frame. However, since the observations made are all local, the observation model in robocentric EKF SLAM is close to linear, making the EKF algorithm perform better.

In particular, if the initial robot pose is removed from the state vector and only feature parameters are estimated, the problem becomes even simpler. As shown by Bishop Jensfelt,[8] the estimation error is guaranteed to be bounded if all the features are observed all the time. Guerreiro et al. show that the nonlinear system dynamics in sensor based SLAM can be regarded as a time varying linear system and a Kalman filter can be designed to estimate the system state.[9] Although robocentric SLAM formulation can reduce linearization error to some extent, it is clear that the uncertainty of the location estimates of features that have not been observed for a long period of time can become very large due to the presence of the process noise.

Optimization based solution. In the nonlinear least squares (NLS) optimization formulation of SLAM,[14] the state vector includes all the robot poses and the feature parameters

$$X(k)^{\text{NLS}} = \begin{bmatrix} X_r(1) \\ \vdots \\ X_r(k) \\ X_f \end{bmatrix} \tag{5}$$

and the aim is to find the state vector which maximizes the likelihood of the odometry and observation measurements. This is equivalent to minimizing an objective function that is the weighted sum of the odometry and observation error function squared.[14] The initial robot pose $X_r(0)$ is assumed to be the origin of the coordinate frame in which state vector is defined.

Pose-graph SLAM problem

In pose-graph SLAM formulation,[16] the observations of the environment are used to first estimate the relationship between the poses from which the observations are acquired. The relative pose between pose i and pose j is denoted as z_{ij} and the covariance matrix of z_{ij} is given by P_{ij}. The state vector for pose-graph SLAM includes all the robot poses

$$X(k)^{\text{Pos}} = \begin{bmatrix} X_r(1) \\ \vdots \\ X_r(k) \end{bmatrix} \tag{6}$$

and as before, the initial robot pose $X_r(0)$ is assumed to be the origin of the coordinate frame in which the state vector is defined. The solution to the SLAM problem is the state vector which maximizes the likelihood of all the relative pose information z_{ij}. This is equivalent to minimizing the weighted sum of the squared error of the relative pose information.[16]

Other variants of SLAM problem

Many variations to the above SLAM problems may occur under practical scenarios. In some cases, external information such as the global robot location may be intermittently available through a GPS receiver. Sometimes the environment may be partially known, for example there may be a landmark with a known global location. The process model may be not available or unreliable for some scenarios such as a robot moving in an uneven terrain or when a handheld sensor such as a camera provides the only information available. When sensors contain biases, it may be necessary to simultaneously estimate the sensor biases together with robot poses and feature parameters. These variations result in SLAM problems which may have significantly different fundamental properties to those discussed in the following sections.

Fundamental properties of SLAM

This section discusses some of the fundamental properties of SLAM.

Observability

The first important question to ask is whether the SLAM problem is solvable. That is, whether the information available is sufficient to obtain an estimate of the current state $X(k)$.

Control theoretic formulation of observability. The observability of a deterministic linear discrete-time system

$$\begin{aligned} X(k+1) &= AX(k) + Bu(k) \\ Z(k) &= CX(k) \end{aligned} \tag{7}$$

means the ability to fully and uniquely recover the system state $X(k)$ from a finite number of observations $Z(k)$ and the knowledge of its control inputs $u(k)$.[29] In the conventional formulation of feature based SLAM, the system state as shown in (3) consists of two parts. One part is the state of the map X_f, which does not change over time; the other part is the location of the robot $X_r(k)$, which evolves in time according to the process model. As observations available to SLAM only relate the robot location to the map of the unknown environment, and the odometry information only contains the relative position between the consecutive robot poses, the initial location of the robot cannot be obtained. Thus the feature based SLAM problem is not observable from the control theoretic point of view,[11,12] unless the robot can observe features with locations known relative to a predefined global coordinate frame. To make the SLAM problem solvable, in most of the SLAM implementations, the initial robot pose $X_r(0)$ is used as the coordinate frame in which the state vector is defined, as formulated in the conventional EKF based SLAM.

On the other hand, it can be shown that the robocentric formulation of EKF SLAM is observable from the control theoretic point of view, if the robot pose $X_r(0)^{\text{Rob}}$ is removed from the state vector $X(k)^{\text{Rob}}$ defined in (4). As the observation model in SLAM is time varying, and both the process and observation models are nonlinear and noisy, existing research on the observability of SLAM relies on many assumptions. For example the assumption that all features in the environment can be observed all the time, which is clearly unrealistic in general, is almost always used. Thus the results that have been reported in the literature are of limited applicability.

Parameter observability of SLAM given a process model. A more straightforward alternative method for analyzing observability of SLAM is to examine its formulations in the form of a parameter estimation problem. The question to ask is whether the robot poses and feature parameters can be inferred (with certain level of accuracy) from a sequence of noisy observations and control inputs, if the initial location of the robot is available. This question can be addressed by analyzing the notion of "parameter observability".[29]

When SLAM is formulated as a nonlinear least squares parameter estimation problem, the Fisher information matrix (FIM) governs the observability. If the FIM is singular, then the parameter estimation problem is not observable. Moreover, when the FIM is non-singular but has very small eigenvalues, the parameters are observable but the observability is "marginal" or "poor".[29]

When a process model and measurements of all the appropriate control inputs are available, it can be easily shown that the FIM in this situation is always not singular, even if there is no observation of features in the environment. Thus all the robot poses along its trajectory are observable. However, the eigenvalues of FIM will gradually decrease over time due to the noises present in the process model. The uncertainty of the computed robot poses will increase as a result and the observability of the robot poses becomes poorer and poorer as time progresses.

When the robot poses at different time steps are all known, the observability of the feature locations in the environment depends on the sensor model. For example, when a range and bearing sensor is available to observe the environment, the location of any observed point feature can be computed and thus point feature based SLAM is always observable. However, it is important to note that the observability gradually becomes poor unless the robot revisits previously mapped areas. Balancing the extent of exploring new areas with revisiting previously explored regions is important in the quest to obtain reliable SLAM solutions.

When a bearing-only sensor such as a camera is used, the uncertainty of the Cartesian coordinates of features observed with near zero parallax may be very poor. Interestingly, if the feature locations are described using a different parametrization, for example using parallax angles as proposed by Zhao et al.,[30] parameters used to describe the features can be computed very accurately even when the parallax angle is zero. Thus an appropriate parametrization can enhance the level of observability.

When a range-only sensor is used, the observability of a point feature position depends on the motion of the robot/sensor. For example, if the robot is moving along a straight line, then there will always be an ambiguity of the point feature location no matter how many range observations are made.[31]

For more general types of geometric features such as line segments and planes, the observability analysis, although tractable, is much more complicated.

Parameter observability of SLAM when a process model is not available. When a process model is not available, estimates of robot poses $X_r(1), \ldots ,X_r(k)$ need to be derived from the observations of the environment. A robot pose becomes observable through the ability of the on-board sensors to simultaneously capture multiple features in the environments from multiple robot poses. Therefore, the parameter observability of SLAM is a function of the properties of the sensor and that of the environment.

For example, in 2D scenarios, observing the same two point features from two robot locations using a range and bearing sensor is adequate to obtain an estimate of the relative pose between the robot locations. When a robot is moving in a long corridor, the robot motion along the corridor cannot be determined by observing the walls of the corridor. In the case of 3D bearing-only sensors such as cameras, image observations of at least five common 3D point features are needed to obtain an estimate of the relative rotation and translation between the two poses, if the motion model is not available. Even then the scale parameter is not observable. Observability issues that arise with monocular cameras have been well addressed in computer vision literature through projective geometry.[32]

Parameter observability of pose-graph SLAM. Pose-graph SLAM is observable as long as the graph whose nodes describe the robot poses is connected. If odometry measurements exist between each pair of consecutive poses, then the graph is always connected. However, similar to the case of feature based SLAM with process model, the quality of the solution deteriorates as the time progresses. Additional links between the nodes obtained through observing the environment and computing relative poses serve to enhance the connectivity of the graph and improve the observability. It has been shown by Khosoussi et al. that graph connectivity defined through the number of spanning trees in the graph can be used to compute the FIM and therefore compute the quality of the SLAM solution.[33]

Issues related to motion and sensor models. Sensors used for observing the environment as well as the ego-motion of the robot are typically assumed to generate observations corrupted by zero-mean Gaussian noise. When the sensors have biases, solving the SLAM problem requires estimating these biases as well. Algorithms for on-line estimation of sensor biases have been proposed[34–36]. The key question of observability when the SLAM state is augmented with bias parameters has also received some attention.[37] The motion model can be a function of the terrain over which the robot travels. Changing motion models and time-varying biases introduce interesting challenges for both theoretical analysis and practical implementations.

Table 1 summarizes some of the observability analysis of different SLAM problems reported in the literature. As discussed previously, in most of the control theoretic observability analysis, it is assumed that all the features are observed at all the time steps in order to keep the observation model time invariant and the analysis tractable. This is not realistic in many situations. The parameter observability analysis does not have this restriction. Although almost all the SLAM problems in Table 1 are observable when treated as a parameter estimation problem, whether the observability is "good" or "poor" is an important issue in practice as a near singular FIM can result in large estimation errors.

Convergence

When a filter based algorithm such as EKF or EIF is used to solve SLAM, a fundamental question to ask is whether the uncertainty of the estimate reaches a finite value as k goes to infinity, assuming a sufficient number of observations

Table 1. Observability analysis of SLAM problems.

SLAM Problem	Observability	Results	Examples
2D Range-bearing	Control theoretic	World-centric SLAM without anchors is not observable	11,12,39,40,44
2D Range-bearing	Parameter	SLAM relative to the first pose is observable	41
2D Range-bearing Robocentric	Control theoretic	Mapping problem is observable; polar coordinate is used	8
2D Bearing-only	Control theoretic	World-centric SLAM is not observable; SLAM relative to the first pose is observable	38
2D Bearing-only	Parameter	It is observable when parallax angle is not zero	41
3D Inertial SLAM	Control theoretic	There are at least three unobservable states	42,43
3D Monocular	Parameter	The scale parameter is not observable; feature observability depends on feature parametrization	30 32
3D Inertial-vision	Parameter	It is observable if global reference frame is fixed and motion is "sufficiently exciting"	45–47
3D Robocentric	Control theoretic	It is observable if at least two features are observed	9,49

can be made. Alternatively, for an optimization based algorithm the question is whether the solution obtained is a local or the global minimum. This is the convergence problem.

Convergence of filter based algorithms. When a linear system is fully observable from a control theoretic point of view, the lower bound of the error in the estimate of its state will only depend on the noise parameters of the system and is independent of the initial information available on the states. However, since SLAM is nonlinear and not fully observable from the control theoretic point of view, it is not possible to draw a similar conclusion with respect to filter based SLAM algorithms.

It has been shown that the feature location uncertainty will monotonically decrease if the motion and observation models are linear.[1] This result has been extended to some simple nonlinear scenarios.[7] Lower bounds of the uncertainty of the system states in EKF SLAM are also derived for some specific instances, but the proofs are only available for the linear case and when Jacobian matrices are evaluated at the ground truth in the nonlinear case. These proofs rely on a zero-mean Gaussian noise assumption. The lower bounds of the estimation uncertainty can only be reached by controlling the robot to move in a specific manner and make a sufficient number of observations to the same features.

Mourikis and Roumeliotis derive an analytical upper bound of the map uncertainty using a dual-map filter, assuming the features are observed in all the steps.[54] The derived upper bound depends on the observation noise level, the process noise level and the size of the map.

Similarly, for robocentric SLAM problems with range and bearing sensors, the uncertainty of the feature estimate can be bounded, as long as no feature leaves the sensor field of view for more than a fixed time period.[8,9]

The practical behaviour of EKF SLAM algorithms is significantly influenced by the linearization errors. For example, the theoretical lower bounds could be violated due to the fact that Jacobians with respect to the same landmark are evaluated at different estimate values at different steps.[4–7] This is the inconsistency issue of EKF SLAM that will be discussed further in the subsection "Consistency".

Convergence of optimization based algorithms. Whether an algorithm converges to the global solution in a nonlinear optimization problem depends on whether the initial guess used to start the iterations is sufficiently close to the global minimum as well as the optimization algorithm used. In contrast, it has been observed that relatively naive optimization algorithms as well as somewhat randomly selected initial guesses are adequate to solve the optimization based SLAM problems. Olson et al. use a stochastic gradient descent (SGD) algorithm to solve the pose-graph SLAM problem by addressing each constraint individually in a sequence.[55] This is somewhat surprising as in general one could only expect such a strategy to work if the problem does not have local minima. Results reported by Olson et al. and Grisetti et al. show that the pose-graph SLAM can converge to the correct solution most of the time even if the iteration is started from a poor initial guess, provided that the covariance matrix of the relative pose information is close to a spherical matrix.[55,56] For point feature based SLAM, Huang et al. showed that for the popular Victoria Park dataset,[2] a Gauss–Newton algorithm can converge to the global minimum 80% of the time from a random initial guess to the 6898 vehicle poses and the 299 feature positions, if all the covariance matrices are set to identity.[57] Furthermore, the final result is very close to the true solution obtained using the exact covariance matrices. For another popular SLAM dataset, the DLR Spatial-Cognition dataset, when the covariance matrices are set to identity matrices, the same Gauss–Newton algorithm converges to the global minimum when zeroes are used as the initial guess but converges to a local minimum when a random initial guess is used.[57]

Table 2. Convergence of SLAM algorithms.

SLAM problem	SLAM algorithm	Results	Examples
2D Range-bearing	EKF SLAM	Three convergence properties proved for the linear case	1
2D Range-bearing	EKF/EIF SLAM	Three convergence properties proved for the nonlinear case but assuming Jacobians are evaluated at the ground truth	7
2D Range-bearing	Dual filter	Upper bounds on the covariance matrix	54
2D Pose-graph	Gauss–Newton	Estimate of the region of attraction	48
2D Feature-based	Optimization	Guaranteed global optimum for 1-step and 2-step SLAM with spherical covariance matrices	20,51
2D/3D Pose-graph	Optimization	Number of spanning trees is a key factor for FIM	33
2D/3D Pose-graph	Optimization	Verify whether a solution is optimal or not	50

These surprising results raise some interesting questions: How many minima exist in a SLAM problem? Do local minima exist near the globally optimal solution to the SLAM problem? How far is SLAM from a convex optimization problem? Some recent research work by Wang et al. has revealed the partially linear structure of SLAM and provides a necessary and sufficient condition for the existence of only one minimum for one-step SLAM, as well as a numerical method for obtaining the global minimum of two-step SLAM, assuming spherical covariance matrices.[20,51] In a paper by Carlone, a conservative estimate of the region of attraction of the global minimum for a Gauss–Newton algorithm is provided for 2D pose-graph SLAM.[48] Furthermore, Carlone et al. provide a method to verify whether the global minimum solution is obtained by using Lagrange duality.[50]

Despite these interesting developments, an algorithm that can guarantee the convergence to the globally optimal solution of a SLAM problem involving a large number of poses is not yet available.

Another interesting question to ask is how to control the robot motion or sensing in order to achieve a desired accuracy of robot location and map while achieving other goals such as covering the area of interest. This is known as the active SLAM problem which is addressed in various works.[18,58–62]

Table 2 summarizes some of the convergence results for different SLAM algorithms. It is clear that the research in this direction requires further investigation.

Criteria for evaluating the performance of SLAM algorithms

In the past 15 years, a large number of different algorithms have been proposed to solve SLAM. Implementations of many such algorithms are also publicly available, for example through the OpenSLAM website: https://www.openslam.org.

Given this, an obvious question to ask is how to evaluate and compare their performance. Recently, a set of benchmark problems and datasets have become available, for example, the Radish dataset repository and the dataset papers published in the International Journal of Robotics Research, making it possible to follow the standard scientific practice to evaluate the different SLAM algorithms.[63–66]

Given that SLAM front-end tends to very much sensor and robot specific, the focus of this section is on the back-end that solves the estimation or the optimization problem.

Consistency

Consistency of the solution is one of the most important criteria for evaluating a SLAM solution. For the state estimation problem of a dynamic system, a solution is said to be consistent if the estimate is unbiased and the estimated covariance matrix matches the real mean squared error.[29] It has been demonstrated that both the EKF based SLAM solution and the particle filter based SLAM solution can be inconsistent in some scenarios.[4–7,67]

In the conventional EKF SLAM algorithm, even when the system noise is zero-mean Gaussian, the estimate can be inconsistent because the relevant Jacobian matrices are evaluated at the estimated system state, which are different at different time steps for the same feature/pose. It has been shown that the extent of inconsistency is closely related to the uncertainty of the robot orientation estimate. In a particle filter based SLAM algorithm, the level of inconsistency depends on the number of particles used. The larger the number of particles, the less the extent of the inconsistency.[67] The possible estimate inconsistency in EKF based point feature SLAM was recognized as early as in 2001,[4] and was later shown to be an important issue in SLAM implementations of large-scale environments. When complex geometric features such as line features are used in SLAM, the estimate inconsistency of EKF algorithms become apparent even in relatively small environments.[68] Robocentric EKF SLAM has been shown to perform better than conventional EKF SLAM in terms of estimator consistency.[5] Guerreiro et al. have reported a sensor-based SLAM using a Kalman filter, the algorithm is shown to produce consistent estimates but only results using small experimental data are provided.[9] Approaches based on observability analysis, such as an observability constrained Kalman filter, have also been shown to reduce the inconsistency.[13,53]

In recent years it has become clear that the overconfident estimates generated by SLAM estimators is mainly due to the fact that the estimation process is fed with incorrect information as a result of the Jacobian matrices of observation/odometry functions with respect to the same feature/pose are evaluated at the current feature/pose location estimates, rather than at the true state.[7,10] This effect is most prominent when the difference between the true and estimated orientation is large. Using active control to keep the robot orientation uncertainty within certain bounds to reduce the level of inconsistency is a practical strategy that has been shown to be effective. In optimization based SLAM, all the odometry and observation information are fused together in one go, and the Jacobians are re-evaluated in each iteration, thus the estimates are much better than EKF based approaches in terms of consistency. Finding a filter based SLAM algorithm that can achieve a consistency level comparable to the optimization based approach would be an interesting research topic.

While it is unlikely that inconsistency can be completely avoided, as the SLAM problem is inherently nonlinear, it is important to compare the tendency for inconsistency in different algorithms. The consistency of a SLAM algorithm could be evaluated through Monte Carlo simulations by comparing the sample covariance with the covariance reported by the algorithm, as shown by Huang et al.[69] Another strategy is to calculate the average normalized estimation error squared and then perform a χ^2 test.[6] In practice, whether inconsistency can be tolerated or not depends on the intended application of the SLAM output. For example, in an urban search and rescue scenario where the purpose of the map is purely for human interpretation, what is of importance is whether the topological structure of the map is correct or not.

Accuracy

Given that SLAM algorithms reported in the literature rely on different simplifying assumptions in order to make the computations more efficient, the impact of these on the final accuracy is also an important evaluation criterion. The most common approach to this is to compare the results from a given algorithm with that obtained by the full nonlinear least squares based SLAM solution starting from a good initial value. Since the SLAM problem aims to minimize the χ^2 error, comparing the χ^2 error is an indication of how far the results obtained are from the optimal solution. Some SLAM algorithms only provide an estimate of part of the SLAM state vector. In this case, to compare the χ^2 error of the full SLAM problem, one needs to first find the optimal value of the remaining states by treating the estimated part of the state as constants.[69]

Alternatively the ground truth, if available, can be used to evaluate the accuracy of SLAM algorithms. In this case, Monte Carlo simulations are needed to provide some statistical results.

A number of experimental datasets are publicly available for performance evaluation of SLAM algorithms. For some of the datasets, the ground truth of the robot poses (e.g. obtained through high quality GPS) is also available for comparison.[70] Given that different algorithms may perform well in different scenarios, rigorous evaluations require examining the algorithm performance under different sizes of environments, noise levels, feature densities, and loop closures.

Computational efficiency

Computational efficiency has been a focus of the SLAM community for many years. An increasingly large array of algorithms that make use of a wide range of mathematical techniques such as sparse linear equation solvers to achieve impressive results are becoming available. For example, g2o is one of the most efficient implementations of SLAM back-end based on nonlinear least squares optimization.[16]

One way for comparing computational efficiency of different SLAM algorithms is to analyze and compare the computational complexity (e.g. the number of floating point operations needed in the algorithm). Another relatively straightforward (but less rigorous) way is to compare the execution time of the algorithms using the same computation platform.

It is worth noting that some of the algorithms achieve efficiency using approximations. Their final results, while potentially be accurate, will differ from the globally optimal solution obtained through nonlinear optimization. This is particularly important in map joining algorithms such as the Combined Kalman–Information Filter SLAM algorithm and Linear SLAM.[71,72] Moreover, some algorithms such as LAGO and pose-chain SLAM may only work well for datasets with particular characteristics.[22,73] Thus it is important to recognize that both the computational efficiency and accuracy need to be considered when selecting a SLAM algorithm for a practical application.

Other recent developments in SLAM

This section outlines some of the other recent significant achievements in SLAM.

Graph pruning for long term SLAM

When it comes to life-long operation of a robot, where the robot is required to navigate in an environment for several months or even longer, state vector in pose-graph SLAM will grow to an unmanageable size. Graph pruning is one way to deal with this issue and keep the computational cost acceptable. A typical graph pruning process includes the following steps: (a) the selection of which nodes or edges to remove, (b) performing marginalization, and (c) applying an approximation to maintain the sparsity of the graph.[74–77] Interestingly, it has been recently shown that allowing adding edges that are not present in the original graph can also make the graph pruning result more effective.[78]

Robust SLAM back-end

Another interesting problem that has attracted SLAM researchers in the past few years is the robustness of SLAM back-end. The question to ask is, if the SLAM front-end generates incorrect data associations or loop closure detections, can the SLAM back-end deal with these? A number of strategies for addressing this problem, such as Switching Constraints, Dynamic Covariance Scaling, Realizing, Reversing, Recovering, and Max-Mixtures, have been proposed.[19,79–81] Although none of the existing algorithms can guarantee adequate performance, especially when outliers that are consistent among each other exist, they do provide useful tools for the robust SLAM back-end.

Dense mapping

In point feature based SLAM, the final map only contains sparse representation of the environment,[1,30] as only a small fraction of information available from the sensors such as laser range finders or cameras is exploited to extract geometric features present in the environment. The remaining sensor data are discarded. In pose-graph SLAM, once the complete robot trajectory is estimated,[56,82,83] data from the observations can be superimposed together to get a dense map, usually in the form of an occupancy grid. However, the uncertainty associated with the robot trajectory estimate (especially for large-scale pose-graph SLAM) will have a significant influence on the quality of the final map. One way to evaluate the quality of the superimposed map is to use the crispness.[84]

By modelling the environment with more complex geometric primitives such as lines or B-splines, more sensor data information is incorporated.[68,85] However, some issues related to the inconsistency of the solution have been observed.[68,86] Another hybrid strategy is to model the region surrounding a point feature using a shape model in a coordinate frame attached to the feature.[87] In this case, only the point feature estimates can be updated while the shape models are only used to increase the information content of the map. Recently, truncated signed distance function (TSDF) representation has become a popular method for building dense maps, especially when RGB-D sensors are used.[88] Although promising, the uncertainty involved in the map representation cannot be handled elegantly, especially when closing large loops. By combining the idea of local map joining with TSDF, maps with improved quality can be obtained.[89]

For monocular SLAM, significant progress has also been made along this direction. The representative works are DTAM and LSD SLAM.[90,91]

A fully parameterized environment representation which captures all the information available from sensors such as laser range finders and depth cameras, which can be updated in a statistically consistent manner, and is computationally tractable, remains an interesting future challenge.

Data association

Data association, establishing the relationship between information collected at different time instances, is one of the most important tasks in SLAM. This task is typically handled by the SLAM front-end.

When information from sensors such as laser scanners and RGB-D sensors is available, ICP and its different variations have been key algorithms for estimating the relative pose between the locations from which the scans have been acquired.[92] Recently Gaussian Mixture Models (GMM) and Normal Distribution Transform (NDT) based scan registration have been shown to be able to provide superior performance compared with the ICP based approach.[93,94]

For monocular SLAM, image matching and loop closure detection are mainly based on feature descriptors (such as SIFT and SURF) and bag-of-words.[95] Recently ORB has been shown to be a more efficient alternative to SIFT and SURF.[96]

Since the uncertainty of the state estimate can be very large before closing a large loop, traditional data association based on geometric information together with the estimated uncertainty may not be adequate for loop closure detection. The quality and robustness of the data association in SLAM can potentially be improved by using the local submap joining strategy to perform data association at two levels, one at the feature level within a local map, the other at the local map level in which features in different local maps are matched.[71,72,97]

Future directions in SLAM research

One important question to ask with respect to a SLAM back-end is how to guarantee to achieve the globally optimal solution. Finding a good initial value is critical, but how good is adequate is an interesting question to pose. Although some important progress along this direction has been made,[20,48] more investigation is necessary to gain further understanding of this important problem. Furthermore, given that most of the existing SLAM formulations assume Gaussian noise, understanding the impact of non-Gaussian noise and strategies for handling this efficiently may lead to more effective SLAM algorithms, particularly when the data rate and sensor quality are poor, for example in turbid underwater environments.

Important research directions also include the close integration of the SLAM front-end and SLAM back-end for practical applications. Although there has been some interesting progress in handling outliers that have been undetected by the SLAM front-end, fault detection in SLAM, either front-end data association failure or back-end optimization failure, are important future challenges. Furthermore, mapping an environment at the object level is important particularly when robots are used in predominantly human environments. Important questions include how to use prior knowledge of the environments, how to identify complicated objects, how to present the objects and environments and how to correctly describe the uncertainties involved.[98]

SLAM in dynamic environments poses a number of new challenges. Important issues include how to identify the moving parts, how to represent the changing environments and how to model the deformation.[52] The observability analysis for SLAM in deformable environments is also an important research topic.

Conclusion

Significant progress has been made in SLAM research in the past 15 years, and many insights into the fundamental aspects of SLAM have been discovered. Observability is an important issue and parameter observability is perhaps the best strategy available to analyze this aspect. The level of parameter observability, which is closely related to the graph connectivity and the sensor model, is also a critical issue in practice. The study of convergence properties for different SLAM algorithms is still at an earlier stage and further understanding of the achievable estimate accuracy is required. Compared to filter based SLAM algorithms, optimization based SLAM algorithms have better consistency and could provide more accurate estimates. In practice, pose-graph SLAM could provide more robust estimate of robot trajectories than feature based SLAM as the former is able to tolerate some imperfect feature matching.

Some important future research topics include understanding the fundamental limitations and achievable accuracy in SLAM, how to reliably achieve the globally optimal solution to SLAM, failure detection in SLAM algorithms, active SLAM for improving the performance of SLAM, robust algorithms for SLAM front-end, dense mapping with a theoretically sound uncertainty model and SLAM in dynamic or deformable environments. Further progress on building up some standards for examining and evaluating SLAM algorithms, especially for long-term and large-scale SLAM in dynamic environments, is also essential to making the outcomes of SLAM research useful to a practitioner.

Acknowledgements

This article is a revised and expanded version of a paper entitled "A review of recent developments in simultaneous localization and mapping" presented at the 6th IEEE International Conference on Industrial and Information Systems held in Kandy, Sri Lanka, in August 2011.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Australian Research Council Discovery Project DP120102786.

References

1. Dissanayake G, Newman P, Clark S, et al. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Trans Robotic Autom* 2001; 17(3): 229–241.
2. Guivant JE and Nebot EM. Optimization of the simultaneous localization and map building (SLAM) algorithm for real time implementation. *IEEE Trans Robotic Autom* 2001; 17(3): 242–257.
3. Davison AJ, Reid ID, Molton ND, et al. MonoSLAM: Real-time single camera SLAM. *IEEE Trans Pattern Anal* 2007; 29(6): 1052–1067.
4. Julier SJ and Uhlmann JK. A counter example for the theory of simultaneous localization and map building. In *IEEE international conference on robotics and automation*, Seoul, Korea, 2001, pp.4238–4243. IEEE.
5. Castellanos JA, Neira J and Tardos JD. Limits to the consistency of EKF-based SLAM. In *IFAC symposium on intelligent autonomous vehicles*, 2004, Lisbon, Portugal.
6. Bailey T, Nieto J, Guivant J, et al. Consistency of the EKF-SLAM algorithm. In *IEEE/RSJ international conference on intelligent robots and systems*, Beijing, China, 2006, pp. 3562–3568. IEEE.
7. Huang S and Dissanayake G. Convergence and consistency analysis for extended Kalman filter based SLAM. *IEEE Trans Robot* 2007; 23(5): 1036–1049.
8. Bishop AN and Jensfel P. A stochastically stable solution to the problem of robocentric mapping. In *IEEE international conference on robotics and automation*, Kobe, Japan, May 2009, pp.1615–1622. IEEE.
9. Guerreiro BJ, Batista P, Silvestre C, et al. Globally asymptotically stable sensor-based simultaneous localization and mapping. *IEEE Trans Robot* 2013; 29(6): 1380–1395.
10. Huang GP, Mourikis AI and Roumeliotis SI. Analysis and improvement of the consistency of extended Kalman filter-based SLAM. In *IEEE international conference on robotics and automation*, Pasadena, California, 2008, pp.473–479. IEEE.
11. Andrade-Cetto J and Sanfeliu A. The effects of partial observability in SLAM. In *IEEE international conference on robotics and automation*, New Orleans, LA, 2004, pp.397–402. IEEE.
12. Lee KW, Wijesoma WS and Guzman JI. On the observability and observability analysis of SLAM. In *IEEE/RSJ international conference on intelligent robots and systems*, Beijing, China, 2006, pp.3569–3574. IEEE.
13. Huang GP, Mourikis AI and Roumeliotis SI. Observability-based rules for designing consistent EKF SLAM estimators. *Int J Robot Res* 2010; 29(5): 502–528.
14. Dellaert F and Kaess M. Square root SAM: Simultaneous localization and mapping via square root information smoothing. *Int J Robot Res* 2006; 25(12): 1181–1203.
15. Thrun S, Burgard W and Fox D. *Probabilistic robotics*. Cambridge, MA: MIT Press, 2005.
16. Kummerle R and Grisetti G, Strasdat H, et al. g2o: A general framework for graph optimization. In *IEEE international*

conference on robotics and automation, Shanghai, China, 2011, pp.3607–3613. IEEE.
17. Lu F and Milios E. Globally consistent range scan alignment for environment mapping. *Auton Robot* 1997; 4(4): 333–349.
18. Ila V, Porta JM and Andrade-Cetto J. Information-based compact pose SLAM. *IEEE Trans Robot* 2010; 26(1): 78–93.
19. Latif Y, Cadena C and Neira J. Robust loop closing over time for pose graph SLAM. *Int J Robot Res* 2013; 32(14): 1611–1626.
20. Wang H, Huang S, Frese U, et al. The nonlinearity structure of point feature SLAM problems with spherical covariance matrices. *Automatica* 2013; 49(10): 3112–3119.
21. Wang H, Hu G, Huang S, et al. On the structure of nonlinearities in pose graph SLAM. In *Robotics: Science and systems*, 2012. MIT Press.
22. Carlone L, Aragues R, Castellanos JA, et al. A fast and accurate approximation for planar pose graph optimization. *Int J Robot Res* 2014; 33(7): 965–987.
23. Khosoussi K, Huang S and Dissanayake G. Exploiting the separable structure of SLAM. In *Robotics: Science and systems*, 2015.
24. Dissanayake G, Huang S, Wang Z, et al. A review of recent developments in simultaneous localization and mapping. In *6th IEEE international conference on industrial and information systems*, Kandy, Sri Lanka, 2011, pp.477–482. IEEE.
25. Durrant-Whyte H and Bailey T. Simultaneous localisation and mapping (SLAM): Part I. *IEEE Robot Autom Mag* 2006; 13(2): 99–110.
26. Bailey T and Durrant-Whyte H. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robot Autom Mag* 2006; 13(3):108–117.
27. Frese U, Wagner R and Rofer T. A SLAM overview from a user's perspective. *KI-Kunstliche Intelligenz* 2010; 24(3): 191–198.
28. Zamora E and Yu W. Recent advances on simultaneous localization and mapping for mobile robots. *IETE Tech Rev* 2013; 30(6): 490–496.
29. Bar-Shalom Y, Li XR and Kirubarajan T. *Estimation with applications to tracking and navigation: Theory algorithms and software*. New York: John Wiley & Sons, 2001.
30. Zhao L, Huang S, Sun Y, et al. ParallaxBA: Bundle adjustment using parallax angle feature parametrization, *Int J Robot Res* 2015; 34(4–5): 493–516.
31. Blanco JL, Fernandez-Madrigal JA and Gonzalez J. Efficient probabilistic range-only SLAM. In *IEEE/RSJ international conference on intelligent robots and systems*, Nice, France, 2008, pp.1017–1022. IEEE.
32. Hartley R and Zisserman A. *Multiple view geometry in computer vision*, vol. 2. Cambridge: Cambridge University Press, 2000.
33. Khosoussi K, Huang S and Dissanayake G. Tree-Connectivity: Evaluating the graphical structure of SLAM. In *IEEE international conference on robotics and automation*, Stockholm, Sweden, 2016, pp.1316–1322. IEEE.
34. Martinelli A, Tomatis N and Siegwart R. Simultaneous localization and odometry self calibration for mobile robot. *Auton Robot* 2007; 22: 75–85.
35. Perera LDL, Wijesoma WS and Adams MD. The estimation theoretic sensor bias correction problem in map aided localization. *Int J Robot Res* 2006; 25(7): 645–667.
36. Wijesoma WS, Perera LDL, Adams MD, et al. An analysis of the bias correction problem in simultaneous localization and mapping. In *IEEE/RSJ international conference on intelligent robots and systems*, Alberta, Canada, 2005, pp.747–752. IEEE.
37. Martinelli A. Visual-inertial structure from motion: Observability vs minimum number of sensors. In *IEEE international conference on robotics and automation*, Hong Kong, China, 2014, pp.1020–1027. IEEE.
38. Vidal-Calleja T, Bryson M, Sukkarieh S, et al. On the observability of bearing-only SLAM. In *IEEE international conference on robotics and automation*, Roma, Italy, 2007, pp. 4114–4119. IEEE.
39. Perera LDL, Melkumyan A and Nettleton E. On the linear and nonlinear observability analysis of the SLAM problem. In *IEEE international conference on mechatronics*, Malaga, Spain, 2009, pp.1–6. IEEE.
40. Perera LDL and Nettleton E. On the nonlinear observability and the information form of the SLAM problem. In *IEEE/RSJ international conference on intelligent robots and systems*, St. Louis, MO, 2009, pp.2061–2068. IEEE.
41. Wang Z and Dissanayake G. Observability analysis of SLAM using Fisher information matrix. In *IEEE international conference on control, automation, robotics and vision*, Hanoi, Vietnam, 2008, pp.1242–1247. IEEE.
42. Bryson M and Sukkarieh S. Observability analysis and active control for airborne SLAM. *IEEE Trans Aero Elec Sys* 2008; 44(1): 261–280.
43. Kim J and Sukkarieh S. Improving the real-time efficiency of inertial SLAM and understanding its observability. In *IEEE/RSJ international conference on intelligent robots and systems*, Sendai, Japan, 2004, pp.21–26. IEEE.
44. Souici A, Courdesses M, Ouldali A, et al. Full-observability analysis and implementation of the general SLAM model. *Int J Sys Sci* 2013; 44(3): 568–581.
45. Jones ES and Soatto S. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *Int J Robot Res* 2011; 30(4): 407–430.
46. Chiuso A, Favaro P, Jin H, et al. Structure from motion causally integrated over time. *IEEE Trans Pattern Anal* 2002; 24(4): 523–535.
47. Kelly J and Sukhatme GS. Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *Int J Robot Res* 2011; 30(1): 56–79.
48. Carlone L. A convergence analysis for pose graph optimization via Gauss–Newton methods. In *IEEE international conference on robotics and automation*, Karlsruhe, Germany, 2013, pp.965–972. IEEE.

49. Lourenco P, Guerreiro BJ, Batista P, et al. Simultaneous localization and mapping for aerial vehicles: A 3-D sensor-based GAS filter. *Auton Robot* 2016; 40(5): 881–902.
50. Carlone L, Rosen DM, Calafiore G, et al. Lagrangian duality in 3D SLAM: Verification techniques and optimal solutions. In *IEEE/RSJ international conference on intelligent robots and systems*, Hamburg, Germany, 2015, pp. 125–132. IEEE.
51. Wang H, Huang S, Khosoussi K, et al. Dimension reduction for point feature SLAM problems with spherical covariance matrices. *Automatica* 2015; 51: 149–157.
52. Newcombe RA, Fox D and Seitz SM. DynamicFusion: Reconstruction and tracking of non-rigid scenes in real-time. In *IEEE conference on computer vision and pattern recognition*, Boston, Massachusetts, 2015, pp.343–352. IEEE.
53. Hesch JA, Kottas DG, Bowman SL, et al. Consistency analysis and improvement of vision-aided inertial navigation. *IEEE Trans Robot* 2014; 30(1): 158–176.
54. Mourikis AI and Roumeliotis SI. Analytical characterization of the accuracy of SLAM without absolute orientation measurements. In *Robotics: Science and systems*, Philadelphia, Pennsylvania, 2006, pp.215–222. MIT press.
55. Olson E, Leonard J and Teller S. Fast iterative optimization of pose graphs with poor initial estimates. In *IEEE international conference on robotics and automation*, Orlando, Florida, 2006, pp.2262–2269. IEEE.
56. Grisetti G, Stachniss C, Grzonka S, et al. A tree parameterization for efficiently computing maximum likelihood maps using gradient decent. In *Robotics: Science and systems*, Atlanta, Georgia, 2007. MIT Press.
57. Huang S, Lai Y, Frese U, et al. How far is SLAM from a linear least squares problem? In *IEEE/RSJ international conference on intelligent robots and systems*, Taipei, Taiwan, 2010, pp.3011–3016. IEEE.
58. Sim R and Roy N. Global A-optimal robot exploration in SLAM. In *IEEE international conference on robotics and automation*, Barcelona, Spain, 2005, pp.673–678. IEEE.
59. Stachniss C, Hahnel D and Burgard W. Exploration with active loop closing for FastSLAM. In *IEEE/RSJ international conference on intelligent robots and systems*, Sendai, Japan, 2004, pp.1505–1510. IEEE.
60. Leung C, Huang S and Dissanayake G. Active SLAM using model predictive control and attractor based exploration. In *IEEE/RSJ international conference on intelligent robots and systems*, Beijing, China, 2006, pp.5026–5031. IEEE.
61. Carrillo H, Reid I and Castellanos JA. On the comparison of uncertainty criteria for active SLAM. In *IEEE international conference on robotics and automation*, St Paul, Minnesota, 2012, pp.2080–2087. IEEE.
62. Vadim I, Carlone L and Dellaert F. Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments. *Int J Robot Res* 2014; 34(7): 849–882.
63. Howard A and Roy N. The robotics data set repository (Radish), 2003. Available at: http://radish.sourceforge.net/ (accessed).
64. Pandey G, McBride JR and Eustice RM. Ford campus vision and lidar data set. *Int J Robot Res* 2011; 30(13): 1543–1552.
65. Smith M, Baldwin I, Churchill W, et al. The new college vision and laser data set. *Int J Robot Res* 2009; 28(5): 595–599.
66. Tong CH, Gingras D, Larose K, et al. The Canadian planetary emulation terrain 3D mapping dataset. *Int J Robot Res* 2013; doi: 10.1177/0278364913478897.
67. Bailey T, Nieto J and Nebot E. Consistency of the FastSLAM algorithm. In *IEEE international conference on robotics and automation*, Orlando, Florida, 2006, pp.424–429. IEEE.
68. Rodriguez-Losada D, Matia F, Jimenez A, et al. Consistency improvement for SLAM-EKF for indoor environments. In *IEEE conference on robotics and automation*, Orlando, Florida, 2006, pp.418–423. IEEE.
69. Huang S, Wang Z, Dissanayake G, et al. Iterated D-SLAM map joining: Evaluating its performance in terms of consistency, accuracy and efficiency. *Auton Robot* 2009; 27: 409–429.
70. Blanco JL, Moreno FA and Gonzalez J. A collection of outdoor robotic datasets with centimeter-accuracy ground truth. *Auton Robot* 2009; 27: 327–351.
71. Cadena C and Neira J. SLAM in O (logn) with the combined Kalman-information filter. *Robot Auton Sys* 2010; 58(11): 1207–1219.
72. Zhao L, Huang S and Dissanayake G. Linear SLAM: A linear solution to the feature-based and pose graph SLAM based on submap joining. In *IEEE/RSJ international conference on intelligent robots and systems*, Tokyo, Japan, 2013, pp. 24–30. IEEE.
73. Dubbelman G and Browning B. Closed-form online pose-chain SLAM. In *IEEE international conference on robotics and automation*, Karlsruhe, Germany, 2013, pp.5190–5197. IEEE.
74. Konolige K and Bowman J. Towards lifelong visual maps. In *IEEE/RSJ international conference on intelligent robots and systems*, St. Louis, USA, 2009, pp.1156–1163. IEEE.
75. Kretzschmar H and Stachniss C. Information theoretic compression of pose graphs for laser-based SLAM. *Int J Robot Res* 2012; 31(11): 1219–1230.
76. Johannsson H, Kaess M, Fallon M, et al. Temporally scalable visual SLAM using a reduced pose graph. In *IEEE international conference on robotics and automation*, Karlsruhe, Germany, 2013, pp.54–61. IEEE.
77. Carlevaris-Bianco N and Eustice RM. Generic factor-based node marginalization and edge sparsification for pose-graph SLAM. In *IEEE international conference on robotics and automation*, Karlsruhe, Germany, 2013, pp.5748–5755. IEEE.
78. Mazuran M, Tipaldi GD, Spinello L, et al. Nonlinear graph sparsification for SLAM. In *Robotics: Science and systems*, Berkeley, USA, 2014.
79. Sunderhauf N and Protzel P. Switchable constraints for robust pose graph SLAM. In *IEEE/RSJ international conference on intelligent robots and systems*, Vilamoura-Algarve, Portugal, 2012, pp.1879–1884. IEEE.

80. Agarwal P, Tipaldi GD, Spinello L, et al. Robust map optimization using dynamic covariance scaling. In *IEEE international conference on robotics and automation*, Karlsruhe, Germany, 2013, pp.62–69. IEEE.
81. Olson E and Agarwal P. Inference on networks of mixtures for robust robot mapping. *Int J Robot Res* 2013; 32(7): 826–840.
82. Konolige K, Grisetti G, Kummerle R, et al. Efficient sparse pose adjustment for 2D mapping. In *IEEE/RSJ international conference on intelligent robots and systems*, Taipei, Taiwan, 2010, pp.22–29. IEEE.
83. Eustice RM, Singh H and Leonard J. Exactly sparse delayed-state filters for view-based SLAM. *IEEE Trans Robot* 2006; 22(6): 1100–1114.
84. Das A and Waslander SL. Scan registration using segmented region growing NDT. *Int J Robot Res* 2014; 33(13): 1645–1663.
85. Pedraza L, Dissanayake G, Valls Miro J, et al. BS-SLAM: Shaping the world. In *Robotics: Science and systems*, Atlanta, Georgia, 2007.
86. Liu M, Huang S, Dissanayake G, et al. Towards a consistent SLAM algorithm using B-Splines to represent environments. In *IEEE/RSJ international conference on intelligent robots and systems*, Taipei, Taiwan, 2010, pp.2065–2070. IEEE.
87. Nieto J, Bailey T and Nebot E. Scan-SLAM: Combining EKF-SLAM and scan correlation. In *International conference on field and service robotics*, Port Douglas, Australia, 2005. Springer.
88. Newcombe RA, et al. KinectFusion: Real-time dense surface mapping and tracking. In *10th IEEE international symposium on mixed and augmented reality (ISMAR)*, Basel, Switzerland, 2011, pp.127–136. IEEE.
89. Wagner R, Frese U and Bauml B. Graph SLAM with signed distance function maps on a humanoid robot. In *IEEE/RSJ international conference on intelligent robots and systems*, Chicago, USA, 2014, pp.2691–2698. IEEE.
90. Newcombe RA, Lovegrove SJ and Davison AJ. DTAM: Dense tracking and mapping in real-time. In *IEEE international conference on computer vision*, Barcelona, Spain, 2011, pp.2320–2327. IEEE.
91. Engel J, Schops T and Cremers D. LSD-SLAM: Large-scale direct monocular SLAM. In *European conference on computer vision*, Zurich, 2014, pp.834–849.
92. Besl P and McKay N. A method for registration of 3-D shapes. *IEEE Trans Pattern Anal* 1992; 14: 239–256.
93. Jian B and Vemuri BC. Robust point set registration using Gaussian mixture models. *IEEE Trans Pattern Anal* 2011; 33(8): 1633–1645.
94. Stoyanov T, Magnusson M, Andreasson H, et al. Fast and accurate scan registration through minimization of the distance between compact 3D NDT representation. *Int J Robot Res* 2012; 31(12): 1377–1393.
95. Williams B, Cummins M, Neira J, et al. A comparison of loop closing techniques in monocular SLAM. *Robot Auton Sys* 2009; 57(12): 1188–1197.
96. Rublee E, Rabaud V, Konolige K, et al. ORB: An efficient alternative to SIFT or SURF. In *IEEE international conference on computer vision*, Barcelona, Spain, 2011, pp. 2564–2571. IEEE.
97. Huang S, Wang Z and Dissanayake G. Sparse local submap joining filter for building large-scale maps. *IEEE Trans Robot* 2008; 24(5): 1121–1130.
98. Salas-Moreno RF, Newcombe RA, Strasdat H, et al. Slam++: Simultaneous localisation and mapping at the level of objects. In *IEEE conference on computer vision and pattern recognition*, Portland, Oregon, USA, 2013, pp. 1352–1359. IEEE.