A Review of Optimisation Strategies used in Simultaneous Localisation and Mapping

Shoudong Huang[a]

[a]Centre for Autonomous Systems, Faculty of Engineering and Information Technology, University of Technology Sydney, 15 Broadway, Ultimo, NSW 2007, Australia

ARTICLE HISTORY
Compiled November 20, 2018

ABSTRACT
This paper provides a brief review of the different optimisation strategies used in mobile robot simultaneous localisation and mapping (SLAM) problem. The focus is on the optimisation based SLAM back end. The strategies are classified based on their purposes such as reducing the computational complexity, improving the convergence and improving the robustness. It is clearly pointed out that some approximations are made in some of the methods and there is always a trade-off between the computational complexity and the accuracy of the solution. The local submap joining is a strategy that has been used to address both the computational complexity and the convergence and is a flexible tool to be used in the SLAM back end. Although more research is needed to further improve the SLAM back end, nowadays there are quite a few relatively mature SLAM back end algorithms that can be used by SLAM researchers and users.

KEYWORDS
SLAM; optimisation; sparseness; dimension reduction; submap joining.

1. Introduction

Simultaneous Localisation and Mapping (SLAM) is the problem of using a mobile robot/sensor to build a map of an unknown environment and at the same time locating the robot within the map (Cadena et al., 2016; Dissanayake et al., 2001). The process of solving a SLAM problem includes SLAM front end and SLAM back end. SLAM front end refers to the process of feature extraction, feature matching, and data association for feature based SLAM, and scan matching, image matching and loop closure detection for pose-graph SLAM. SLAM back end is to apply estimation or optimisation techniques to find the optimal configuration of the robot poses and the feature positions (for point feature-based SLAM).

Recently, nonlinear optimisation techniques have become popular for solving SLAM back end due to its superior performance as compared with filter based SLAM techniques. However, since SLAM is formulated as a high dimensional nonlinear optimisation problem, finding the global minimum efficiently is nontrivial. Different researchers have used different strategies for addressing this issue. One common strategy is to exploit the sparseness of the Jacobian and information matrix and apply different ap-

CONTACT Shoudong Huang. ORCID identifier: 0000-0002-6124-4178, Email: shoudong.huang@uts.edu.au

proaches for solving the sparse linear equations to speed up the SLAM algorithms. Other strategies include using different objective functions, approximating the problem, using the special nonlinear structure, applying submap joining idea, and using different methods for handling outliers (incorrect data coming in to the SLAM back end), etc.

This paper provides an overview of different strategies used in the optimisation based SLAM back end. We classify the different strategies based on their purposes, and provide example research papers for the different strategies. Due to the large scope covered by SLAM research, this paper only focuses on the review of some of the optimisation based SLAM back end. For a more comprehensive review of the SLAM history, current state-of-the-art, and future, please read Cadena et al. (2016) and the references therein.

This paper is organised as follows. Section 2 gives the formulations of two typical SLAM optimisation problems, namely 2D point feature based SLAM and 2D pose-graph SLAM. Section 3 classifies the different optimisation strategies based on their purposes. Section 4 discusses some strategies for reducing the computational complexity. Section 5 states the different methods for improving the convergence. Section 6 discusses the methods for improving the robustness to outliers. Some other interesting strategies applied in SLAM are briefly mentioned in Section 7. Finally, Section 8 concludes the paper.

2. Feature based SLAM and Pose-graph SLAM

In this section, we introduce two kinds of SLAM optimisation problems. One is feature based SLAM problem. The other is pose-graph SLAM problem. In the following, we use 2D point feature based SLAM and 2D pose-graph SLAM as examples to explain the problem formulations.

2.1. 2D point feature based SLAM

Assume that n 2D point features $\{f_k\}_{k=1}^n$ are observed from a sequence of $m+1$ 2D robot poses $\{r_i\}_{i=0}^m$. We use $\mathbf{z}_k^i$ to denote the observation made from pose r_i to feature f_k, and $\mathbf{o}_i$ $(1 \leq i \leq m)$ to denote the odometry measurement between pose r_{i-1} and pose r_i which includes both the relative translation measurement $\mathbf{o}_i^{xy}$ and the relative rotation measurement o_i^{ϕ}. The observations and the odometry are assumed to be corrupted by zero-mean Gaussian noises with covariance matrices $P_{z_k^i}$, $P_{o_i^{xy}}$ and $p_{o_i^{\phi}}$, respectively. $\mathbf{x}_{f_k} = [x_{f_k} \;\; y_{f_k}]^T$ denotes the position of feature f_k. $\mathbf{x}_{r_i} = [x_{r_i} \;\; y_{r_i}]^T$ denotes the position of robot pose r_i while ϕ_{r_i} denotes the orientation of robot pose r_i. $R(\phi_{r_i})$ is the rotation matrix corresponding to ϕ_{r_i} defined by:

$$R(\phi_{r_i}) \triangleq \begin{bmatrix} \cos\phi_{r_i} & -\sin\phi_{r_i} \\ \sin\phi_{r_i} & \cos\phi_{r_i} \end{bmatrix}. \tag{1}$$

The state vector $\mathbf{x}$ contains all the robot poses (except pose r_0 which defines the coordinate frame and is called "anchor") and all the feature positions that are to be estimated,

$$\mathbf{x} \triangleq [\mathbf{x}_{f_1}^T \;\; \cdots \;\; \mathbf{x}_{f_n}^T \;\; \mathbf{x}_{r_1}^T \;\; \phi_{r_1} \;\; \cdots \;\; \mathbf{x}_{r_m}^T \;\; \phi_{r_m}]^T. \tag{2}$$

The nonlinear least squares (NLLS) SLAM formulation (Dellaert & Kaess, 2006) uses the odometry and observation information to estimate the optimal state vector $\mathbf{x}^*$ that minimises the negative log-likelihood function

$$F_1(\mathbf{x}) = \sum_{i,k} \|\mathbf{z}_k^i - H^{\mathbf{z}_k^i}(\mathbf{x})\|^2_{P^{-1}_{z_k^i}} + \sum_{i=1}^{m} \|\mathbf{o}_i - H^{\mathbf{o}_i}(\mathbf{x})\|^2_{P^{-1}_{\mathbf{o}_i}} \tag{3}$$

where $P_{\mathbf{o}_i} = diag\{P_{o_i^{xy}}, p_{o_i^{\phi}}\}$ and $\|\mathbf{z}\|^2_W \triangleq \mathbf{z}^T W \mathbf{z}$ for vector $\mathbf{z}$ and positive definite matrix W.

In the above least squares SLAM formulation, $H^{\mathbf{z}_k^i}(\mathbf{x})$ and $H^{\mathbf{o}_i}(\mathbf{x})$ are the corresponding functions relating $\mathbf{z}_k^i$ and $\mathbf{o}_i$ to the state $\mathbf{x}$. An odometry measurement is a function of two poses $(\mathbf{x}_{r_{i-1}}^T, \phi_{r_{i-1}})^T$ and $(\mathbf{x}_{r_i}^T, \phi_{r_i})^T$ and is given by

$$H^{\mathbf{o}_i}(\mathbf{x}) = \begin{bmatrix} R(\phi_{r_{i-1}})^T(\mathbf{x}_{r_i} - \mathbf{x}_{r_{i-1}}) \\ \phi_{r_i} - \phi_{r_{i-1}} \end{bmatrix}. \tag{4}$$

A single observation is a function of one pose $(\mathbf{x}_{r_i}^T, \phi_{r_i})^T$ and one feature position $\mathbf{x}_{f_k}$ which is given by

$$H^{\mathbf{z}_k^i}(\mathbf{x}) = R(\phi_{r_i})^T(\mathbf{x}_{f_k} - \mathbf{x}_{r_i}). \tag{5}$$

In particular, since r_0 defines the coordinate frame, $\phi_{r_0} = 0$ and $\mathbf{x}_{r_0} = (0,0)^T$, the odometry function from robot pose r_0 to r_1 is given by

$$H^{\mathbf{o}_1}(\mathbf{x}) = \begin{bmatrix} \mathbf{x}_{r_1} \\ \phi_{r_1} \end{bmatrix} \tag{6}$$

and the observation function from robot pose r_0 to f_k is given by

$$H^{\mathbf{z}_k^0}(\mathbf{x}) = \mathbf{x}_{f_k}. \tag{7}$$

2.2. 2D pose-graph SLAM

In pose-graph SLAM, the original sensor observation information is first transferred into relative pose information through scan matching (Lu & Milios, 1997) or image matching. Thus the pose-graph SLAM problem is to use the relative pose information to estimate the best configuration of the robot poses.

In the 2D pose-graph SLAM back end, the state vector contains all the robot poses (except pose r_0)

$$\mathbf{x} \triangleq [\mathbf{x}_{r_1}^T \quad \phi_{r_1} \quad \cdots \quad \mathbf{x}_{r_m}^T \quad \phi_{r_m}]^T \tag{8}$$

and the objective function to be minimised is given by

$$F_2(\mathbf{x}) = \sum_{i,j} \|\mathbf{o}_{ij} - H^{\mathbf{o}_{ij}}(\mathbf{x})\|^2_{P^{-1}_{\mathbf{o}_{ij}}} \tag{9}$$

Table 1. Different kinds of optimisation strategies used in SLAM

	Strategies		
Purpose	One	Two	Three
Reduce computational complexity	Exploit the sparseness [a]	Use the special nonlinear structure	Use submaps [b]
Improve convergence	Improve initialisation	Use different parametrisation	Approximate the problem [c]
Improve robustness to outliers	M-estimator	Select consistent measurements	

[a] All the modern SLAM algorithms have used the sparseness.

[b] Using submaps is also a strategy for improving initialisation and approximating the problem (both are for improving convergence).

[c] The different ways of approximating the problem is given in Table 2.

where $\mathbf{o}_{ij}$ is the relative pose information between pose r_i and pose r_j, $P_{\mathbf{o}_{ij}}$ is the corresponding covariance matrix, and $H^{\mathbf{o}_{ij}}$ is the relative pose function. They have the same format as $\mathbf{o}_i$, $P_{\mathbf{o}_i}$ and $H^{\mathbf{o}_i}$ in Section 2.1.

Remark 1. The 3D point feature based SLAM and pose graph SLAM are similar to the 2D problems defined above. For other feature based SLAM problem such as line feature SLAM and plane feature SLAM, the feature parameters that define the feature (instead of the position of point feature) will be estimated as part of the state vector.

3. Classification of Optimisation Strategies used in SLAM

In the last decades, a lot of SLAM algorithms have been developed. Different strategies have been proposed for the optimisation based SLAM back end. The different strategies used in the SLAM optimisation process are roughly classified based on their purposes in Table 1. The first important purpose is to reduce the computational complexity of the optimisation process. The second important purpose is to improve the convergence of the optimisation algorithm. The third important purpose is to improve the robustness to the outliers. In the following sections, we will discuss them in more details.

4. Reduce Computational Complexity

In a typical SLAM problem, there are a large number of poses (and features) involved thus the state vector is of very high dimension. There are also a lot of measurements in SLAM (the number of measurements is at least as large as the number of variables in a least squares problem). Hence the SLAM problem is a high dimensional optimisation problem. To solve this high dimensional optimisation problem efficiently is an important issue because a real-time SLAM solution is needed for practical robot applications.

4.1. Exploit the sparseness

Although the dimension of the state vector and the dimension of the measurement vector in SLAM are both very high. There is one special property of the SLAM problem, that is, each measurement in SLAM is only associated with very limited number of variables. For example, the odometry (or relative pose) measurement is only associated with two poses. The feature observation is only associated with one pose and one feature. This fact results in a very sparse Jacobian matrix. Consequently, the linear systems that need to be solved in the iterative optimisation process are all sparse. Because of the sparseness, the storage and computation in optimisation based SLAM algorithms can be made very efficient by using sparse linear algebra.

The sparseness is exploited in all the modern SLAM algorithms. Examples include g2o (Kummerle et al., 2011), Ceres (Agarwal & Mierle, 2016), iSAM2 (Kaess et al., 2012), and SLAM++ (Ila et al., 2017) etc.

4.2. Use the special nonlinear structure

Some recent studies demonstrate that there are also some special nonlinear structures involved in the SLAM optimisation problems. The motivation to this study is that the convergence of some SLAM algorithms seems to be surprisingly good even starting from very poor initial values (Grisetti et al., 2009; Huang et al., 2010; Olson et al., 2006).

One important property of SLAM problem is that the functions involved are with very special format. For example, when the robot orientation ϕ_{r_i} is fixed, the functions $H^{\mathbf{z}_k^i}(\mathbf{x})$ and $H^{\mathbf{o}_i}(\mathbf{x})$ in (3) are both linear. Using this property, it can be proved that the 2D point feature based SLAM problem (with $3m + 2n$ variables) is equivalent to an optimisation problem with only m variables (the robot orientations) (Huang et al., 2012; Wang et al., 2013). Using this dimension reduction result, the number of local minima involved in the one-step and two-step SLAM problem can be analysed (Wang et al., 2013, 2015).

However, when the number of poses involved is large, using the dimension reduction result directly does not help to reduce the computational complexity of the SLAM algorithms, because the sparseness is lost after the dimension reduction. On the other hand, if the separability is exploited in combination with the traditional SLAM solver such as Gauss-Newton, the efficiency of the solver can be significantly improved (Khosoussi et al., 2016).

4.3. Use submaps

Using submaps is also a strategy to improve efficiency for building large-scale maps. The idea is to first solve smaller scale SLAM problems to build small submaps in the local coordinate frames, and then join the submaps together to obtain the large global map.

There are different submap based strategies for SLAM. One strategy is to use the results from the submaps to obtain a high quality initial value for the global SLAM optimisation (Ni & Dellaert, 2010). This will significantly reduce the number of iterations needed for the optimisation algorithm thus achieve computational cost saving.

Another strategy is to treat the estimated state vector of each local map as an integrated observation (the uncertainty is expressed by the local map covariance matrix) in the map joining step (Grisetti et al., 2012; Huang et al., 2008, 2009). By summarising the original data within the local map in this way, the SLAM problem can be solved more efficiently. However, the map joining problem thus solved is slightly different from the original global SLAM problem using the original data due to the nonlinearity involved in the SLAM problems.

Using submaps reduces the computational complexity through improving the initial value or approximating the original problem. They will be discussed more in the next section.

5. Improve the Convergence

As a high dimensional nonlinear optimisation problem, SLAM can have many local minima. One important question to ask is how to make sure a SLAM algorithm can converge to the global minimum.

5.1. Improve initialisation

For a nonlinear optimisation problem, a good initial value is required for the iterative optimisation methods to converge to the globally optimal solution.

The easiest way to obtain the initial value for SLAM is to use the odometry information to compute the initial value of all the robot poses, and use the first observation to the feature and the computed initial robot pose to compute the initial feature position (for feature based SLAM). However, when the robot trajectory is very long, this initial value will be far from accurate due to the drift of the accumulated odometry data.

For pose-graph SLAM, since other relative pose information is available apart from the odometry, a better initial value for a robot pose can be obtained through a minimal depth route from the anchor pose to the pose obtained from a spanning tree (Konolige et al., 2010). However, this initial value could still be inaccurate for large-scale pose-graph SLAM problems. Recently, researchers have proposed the strategy of initilising the rotations first, and then initialising the translation (Carlone et al., 2015), which could result in good quality initialisation for pose graph SLAM.

A high quality initial value could be obtained through incremental SLAM. That is, solve the SLAM problem incrementally and use the SLAM result from the previous step to compute the initial value for the SLAM problem in the current step. This is the strategy used in (Kaess et al., 2008, 2012). Using a tree structure and approximating the Jacobians in the new steps by reusing some of the previously computed Jacobians make the algorithms very efficient.

As mentioned in Section 4.3, using submaps can also help to obtain a very good initial value for the global SLAM optimisation. In fact, any solution of an approximated SLAM problem can be used as a good quality initial value for the SLAM. One trade-off is between the computational cost for obtaining the solution and the accuracy of the solution to the approximated SLAM problem. This will be discussed a bit more in Section 5.3.

Although iterative based SLAM optimisation algorithms appear to be able to achieve high quality solutions with reasonably good initial values. There is still no theoretical guarantee that an initial value is sufficiently accurate for the algorithm to converge to the global minimum. For very special scenarios such as one-step and two-step SLAM problems, some convergence results can be proved (Wang et al., 2013, 2015). In (Carlone, 2013), it is demonstrated that the size of the region of attraction of Gauss-Newton method depends on some key factors such as the measurement uncertainty, the inter-nodal distances, and the graph structure.

5.2. Use different parametrisation

Use different parameters to represent the robot pose and feature positions can also help to improve the convergence. One good example is the monocular SLAM problem where the feature observation is obtained from a monocular camera. In that case,

Table 2. Different ways to approximate the SLAM optimisation problems

	Examples	
Strategy	One	Two
Use different objective function	Spherical covariance matrix	Frobenius norm
Convex relaxation	Semi-definite programming	Lagrangian duality, Verification
Graph pruning and sparsification	Reduce the number of nodes	Reduce the number of edges
Submap joining	Sparse map joining	Linear SLAM

using traditional Euclidean XYZ to represent a 3D point can result in rank deficient Jacobians and singular information matrix. If inverse depth feature parametrisation (Montiel et al., 2006; Sola et al., 2012) is used, the convergence will be improved. If parallax angle feature parametrisation (Zhao et al., 2015) is used, the convergence will be improved even further. The reason why different feature parametrisations can change the convergence is because the information content from the observations about the different feature parameters is different. An important lesson learnt is that "we should not try to estimate something when there is little or no information about it from the measurements". The research in (Zhang et al., 2016) addressed similar issues.

Recently, it is also demonstrated that using manifold representations (e.g. for 3D robot pose), the convergence can be improved to some extent (Grisetti et al., 2010). This is because the manifold representation keeps the structure of the problem better than using some naive representations (such as Euler angle for 3D rotation). More discussions on this point are given in (Zhang et al., 2017).

5.3. Approximate the problem

Another strategy is to formulate an approximate problem that is easier to solve, and then use its solution as an approximate solution or an initial value for further improvements. A number of methods fall into this category. Table 2 summarises some of these methods.

5.3.1. Approximate the objective function

I: Use spherical covariance matrix

In the SLAM problem formulation (3), $P_{z_k^i}$ and $P_{\mathbf{o}_i}$ are the covariance matrices of the Gaussian measurement noises. They can be arbitrary positive definite matrices. Recently, it has been realised that if the covariance matrices $P_{z_k^i}$ and $P_{\mathbf{o}_i}$ are spherical, then the optimisation problem can be solved more easily. The definition of spherical matrix is given in (Wang et al., 2013).

Definition 1: $A \in \mathbb{R}^{2\times 2}$ is called spherical if it commutes with $R(\phi)$ (defined in (1)) for every ϕ. i.e. $AR(\phi) = R(\phi)A$ for every ϕ. $B \in \mathbb{R}^{3\times 3}$ is called spherical if it has the format of $B = diag\{A, a\}$ where $A \in \mathbb{R}^{2\times 2}$ is spherical and a is a real number.

Remark 2. A 2×2 positive definite spherical matrix has the format $P_1 = diag\{a, a\}$ with $a > 0$. A 3×3 positive definite spherical matrix has the format $P_2 = diag\{a, a, b\}$ with $a > 0$ and $b > 0$.

It has been proved in (Wang et al., 2012, 2013, 2015) that the number of local minima is limited for one-step and two-step 2D SLAM problems, if the covariance matrices are all spherical.

For 3D SLAM problems, the spherical matrix can be defined similarly. Simulation and experimental results show that the convergence of the SLAM algorithms is much better when the covariance matrices of the measurement noises are all spherical (in some papers, it is called “isotropic” measurement noise) (Carlone et al., 2016). In fact, many efficient SLAM algorithms are using the spherical covariance assumption (Dubbelman & Brownig, 2015; Grisetti et al., 2010; Rosen et al., 2016; Wang et al., 2012, 2015).

II: Use Frobenius norm

In (Carlone et al., 2016; Rosen et al., 2015, 2016), the Frobenius norm is used to deal with the orientation part of odometry in (3). It is proved in Wang et al. (2018) that for one-step SLAM problem, the strong duality between primal SLAM and corresponding convex programming problem always holds, that is, the optimal solution is guaranteed. The convergence to the globally optimal solution for both using the objective function with Frobenius norm and the objective function in (3) are compared and the results show that using Frobenius norm results in much better convergence results.

Frobenius norm has also been used in other area such as computer vision (Hartley et al., 2008). In (Carlone & Calafiore, 2018), unsquared l_2 norm and l_1 norm are also proposed in the pose graph optimisation formulation.

5.3.2. Use convex relaxation

Since there is in general no guarantee on the convergence to the global minimum when solving the nonlinear least squares problems using iterative methods such as Gauss-Newton or Levenberg-Marquardt, another idea is to apply convex relaxation to the problems and use convex optimisation tools to solve the problem. The first attempt along this direction is done by Liu et al. (2012) using semi-definite programming and later by Rosen et al. (2015). Furthermore, the work by Carlone et al. (2016) applies Lagrangian duality and shows that when strong duality holds SLAM can be solved globally. Moreover, a way to certify a given solution is optimal is also presented.

In (Rosen et al., 2016), very efficient convex relaxation method is applied to solve the non-convex pose graph optimisation problem. Furthermore, it is demonstrated that in many practical scenarios when the noise level is not too big, the solution via convex relaxation is the exact solution of the original problem. The work by Briales & Gonzalez-Jimenez (2017) further improves the algorithm efficiency.

5.3.3. Use graph pruning and sparsification

To further improve the efficiency of SLAM algorithms, some researchers proposed to reduce the number of poses (and features) in the optimisation problem. One strategy is use keyframes (Ila et al., 2010; Konolige & Agrawal, 2009) where only poses with highly informative measurements are added. Another strategy is called “graph pruning” where some of the nodes are marginalised out in the SLAM graph (Kretzschmar et al., 2011). After marginalising the nodes, the graph becomes denser and sparsification is applied to improve the sparseness (Carlevaris-Bianco & Eustice, 2013; Huang et al., 2013; Mazuran et al., 2006).

Both graph pruning and sparsification introduce approximation to the original SLAM problems and the efficiency is achieved by sacrificing the accuracy.

5.3.4. Use submap joining

When each local submap is treated as an integrated observation, the submap joining problem is an approximation of the original SLAM problem.

Within each local map, we can also marginalise out some of the variables before joining the local maps together (e.g. only keep the features (Huang et al., 2009)). This will make the dimension of the state vector of the map joining problem significantly lower than that of the global SLAM problem.

Many submap joining problems are formulated as nonlinear least squares optimisation problems and solved using iterative techniques (Huang et al., 2008; Ni & Dellaert, 2010). Very recently, Linear SLAM (Zhao et al., 2013) is proposed that a sequence of local maps can be joined together either sequentially or in a more efficient divide and conquer manner in which only solving linear least squares and performing nonlinear coordinate transformations are needed. There is no assumption on the structure of the covariance matrices of the local maps (no need to be spherical as in Rosen et al. (2016); Wang et al. (2015)). Since linear least squares problems have closed-form solutions, there is no need of an initial guess and no need of iterations to solve the reformulated map joining problems. Results using publicly available datasets show that the Linear SLAM algorithm can provide accurate results (Zhao et al., 2013, 2014).

Although Linear SLAM guarantees that the map joining solution is correct, it is an approximation of the full nonlinear least squares SLAM because: (i) Instead of using the original odometry and observation information, the local map information (represented as the local map state estimate together with its information matrix) is used in the map joining. (ii) Instead of fusing all the local maps together in one go using nonlinear optimisation as in Huang et al. (2008), only two maps are fused at a time which resulting in a suboptimal solution.

6. Improve Robustness to Outliers

The problem formulations presented in Section 2 assume that the data association is perfect (for feature based SLAM) and the relative pose information are all close to the true values (for pose-graph SLAM). However, in reality, data association and loop closure detection can both be wrong due to wrong feature/scan/image matching. Although part of the SLAM front end is for removing such outliers, there is no guarantee that the data inserted in to the SLAM back end is outlier free. When a wrong observation or wrong relative pose constraint is inserted in the optimisation, the optimisation result can be far from the ground truth.

One question to ask is whether the SLAM back end is able to detect the outliers or not. If a SLAM back end can detect the outliers and ignore them when performing optimisation, then it is called “Robust SLAM back end”.

6.1. M-estimator

The first interesting work along this line is by Sunderhauf & Protzel (2012), where additional variables are added to indicate whether a measurement is outlier or not. Another strategy is to control the impact of outliers through changing the corresponding covariance, such as in (Agarwal et al., 2013; Latif et al., 2013). All these methods can be regarded as variants/extensions of M-estimators (Huber, 1964). The performance of these methods highly depends on the initial value used.

6.2. Consistent measurements

There are also methods for reasoning on the validity of loop closure constraints by looking at the residual error induced by the constraints during optimisation (Carlone et al., 2014; Graham et al., 2015; Olson & Agarwal, 2013). These methods aim to find the maximal set of consistent measurements normally assuming that the set of odometry measurements are reliable. In this way, the inconsistent loop closure constraints are figured out.

7. Some other Strategies

There are some other interesting strategies for SLAM back end. Here we present two of them as examples.

7.1. Constrained optimisation

One strategy is formulating the SLAM problem as a constrained optimisation problem instead of an unconstrained optimisation problem as in Section 2. The idea is to use all the relative information as variables and use all the loop closures as constraints. In this way, no anchor is needed and the optimal solution is equivalent to that of the nonlinear least squares problem (Bai et al., 2016).

It has been demonstrated that this new formulation also provides a good method to handle outliers (Bai et al., 2017).

7.2. Continuous-time trajectory

Another strategy is using a continuous robot trajectory instead of discrete robot poses. Using a continuous curve (e.g. a B-splines) to describe the robot trajectory requires less number of variables to be included in the optimisation step. Along this direction, many new SLAM formulations can be obtained and the SLAM problems are solved in different ways. For more details, please refer to (Furgale et al., 2015).

8. Conclusion

This paper gives a brief review on the different strategies used in the optimisation based SLAM back end. The strategies are used to reduce the computational complexity, improve the convergence, and improve the robustness of the high dimension nonlinear optimisation problem. It is pointed out that the sparseness, the objective function, the parametrisation, and the nonlinear structure of the SLAM problem all play important roles in the performance of the SLAM algorithms.

With these strategies, a lot of SLAM back end algorithms are now very efficient and reliable. Some of the algorithms are presented as open source (e.g. OpenSLAM.org) to benefit SLAM researchers and users. These open source algorithms are valuable resources for the researchers and engineers who are interested in SLAM.

SLAM front end is also extremely important and is not covered in this review. It is probably more challenging than SLAM back end since most of the SLAM front end algorithms are application dependent. Recently, some nice integrations of SLAM front end and SLAM back end have been developed for SLAM using different sensors.

These together with the availability of a large number of benchmark SLAM datasets are speeding up the practical applications of SLAM in different scenarios.

References

Agarwal, P., Tipaldi, G. D., Spinello, L., Stachniss, C. & Burgard, M. (2013). Robust map optimization using dynamic covariance scaling, in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 62-69.

Agarwal, S., Mierle, K. (2016). Ceres Solver. http://ceres-solver.org

Bai, F., Vidal-Calleja, T. and Huang, S. (2018). Robust Incremental SLAM Under Constrained Optimization Formulation. *IEEE Robotics and Automation Letters*, 3(2), pp.1207-1214.

Bai, F., Huang, S., Vidal Calleja, T.A. and Zhang, Q. (2016, October). Incremental SQP method for constrained optimization formulation in SLAM. In *IEEE International Conference on Control, Automation, Robotics and Vision.*

Briales, J. & Gonzalez-Jimenez, J. (2017). Cartan-Sync: Fast and Global SE (d)-Synchronization, *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2127-2134.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I. & Leonard, J.J. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age, *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309-1332.

Carlone, L. (2013). Convergence analysis of pose graph optimization via Gauss-Newton methods, in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 965-972.

Carlone, L. Censi, A. & Dellaert, F. (2014). Selecting good measurements via l_1 relaxation: A convex approach for robust estimation over graphs, in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, pp. 2667-2674.

Carlone, L., Tron, R., Daniilidis, K. & Dellaert, F. (2015). Initialization techniques for 3D SLAM: a survey on rotation estimation and its use in pose graph optimization, In *Proc. IEEE Int. Conf. Robot. Autom.*, Seattle, WA, USA, pp. 4597-4604.

Carlone, L., Calafiore, G., Tommolillo, C., & Dellaert, F. (2016). Planar pose graph optimization: Duality, optimal solutions, and verification, *IEEE Transactions on Robotics.* 32(3), 545-565.

Carlone, L. & Calafiore, G.C., (2018). Convex relaxations for pose graph optimization with outliers. *IEEE Robotics and Automation Letters*, 3(2), pp.1160-1167.

Carlevaris-Bianco, N. & Eustice, R. M. (2013). Generic factor-based node marginalization and edge sparsification for pose-graph SLAM, in *Proc. IEEE Int. Conf. Robot. Autom.*, pp. 5748-5755.

Dellaert, F. & Kaess, M. (2006). Square root SAM: Simultaneous localization and mapping via square root information smoothing, *International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181-1203.

Dissanayake, G., Newman, P., Clark, S., Durrant-Whyte, H. & Csorba, M. (2001). A solution to the simultaneous localization and map building (SLAM) problem, *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 229-241.

Dubbelman, G. & Brownig, B. (2015). COP-SLAM: closed-form online pose-chain optimisation for visual SLAM, *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1194-1213.

Furgale, P., Tong, C.H., Barfoot, T.D. & Sibley, G. (2015). Continuous-time batch trajectory estimation using temporal basis functions. The International Journal of Robotics Research, 34(14), pp.1688-1710.

Graham, M. C., How, J. P., & Gustafson, D. E. (2015). Robust incremental slam with consistency-checking, in Proc. IEEE/RSJ Intell. Int. Conf. Robots Syst., pp. 117-124.

Grisetti, G., Stachniss, C., & Burgard, W. (2009). Non-linear constraint network optimisation for efficient map learning, *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 3, pp. 428-439, 2009.

Grisetti, G., Kummerle, R., Stachniss, C., Frese, U. & Hertzberg, C. (2010). Hierarchical optimisation on manifolds for online 2D and 3D mapping, In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 273-278.

Grisetti, G., Kummerle, R. & Ni, K. (2012). Robust optimisation of factor graphs by using condensed measurements, In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 581-588.

Hartley, R., Trumpf, J., Dai, Y. & Li, H. (2013). Rotation averaging, *International Journal of Computer Vision*, 103(3), 267-305.

Huang, G., Kaess, M., & Leonard, J. J. (2013). Consistent sparsification for graph optimization, in Proc. Eur. Conf. Mobile Robots, pp. 150-157.

Huang, S., Wang, Z., Dissanayake, G. & Frese, U. (2008). Iterated SLSJF: A sparse local submap joining algorithm with improved consistency, In *Proc. Australasian Conference on Robotics and Automation.*

Huang, S., Wang, Z., Dissanayake, G., & Frese, U. (2009). Iterated D-SLAM map joining: Evaluating its performance in terms of consistency, accuracy and efficiency. *Autonomous Robots*, 27(4), 409-429.

Huang, S., Lai, Y., Frese, U., & Dissanayake, G. (2010). How far is SLAM from a linear least squares problem? In *Proc. International Conference on Intelligent Robots and Systems (IROS)*, pp. 3011-3016.

Huang, S., Wang, H., Frese, H., & Dissanayake, G. (2012). On the number of local minima to the point feature based SLAM problem, In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2074-2079.

Huber, P. J. (1964). Robust estimation of a location parameter, Ann. Math. Statist., vol. 35, no. 1, pp. 73-101.

Ila, V., Porta, J. M., & Andrade-Cetto, J. (2010). Information-based compact pose SLAM, IEEE Trans. Robot., vol. vol. 26, no. 1, pp. 78-93.

Ila, V., Polok, L., Solony, M., & Svoboda, P. (2017). SLAM++-A highly efficient and temporally scalable incremental SLAM framework, *The International Journal of Robotics Research*, vol. 36, no. 2, pp. 210-230.

Kaess, M., Ranganathan, A. & Dellaert, F. (2008). iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365-1378.

Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J. & Dellaert, F. (2012). iSAM2: Incremental smoothing and mapping using the Bayes tree, *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216-235.

Khosoussi, K., Huang, S., & Dissanayake, G. (2016). A Sparse Separable SLAM Back-End. IEEE Trans. Robotics, 32(6), pp.1536-1549.

Konolige K. & Agrawal, M. (2009). FrameSLAM: From Bundle Adjustment to Real-Time Visual Mapping, *IEEE Transactions on Robitics*, vol. 24, no. 5, pp. 1066-1077.

Konolige, K., Grisetti, G., Kummerle, R., Limketkai, B., & Vincent, R. (2010). Efficient sparse pose adjustment for 2D mapping, In *Proc. International Conference on Intelligent Robots and Systems (IROS)*, pp. 22-29.

Kretzschmar, H., Stachniss, C. & Grisetti, G. (2011). Efficient informationtheoretic graph pruning for graph-based SLAM with laser range finders, in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., pp. 865-871.

Kummerle, R., Grisetti, G., Strasdat, H., Konolige, K. & Burgard, W. (2011). g2o: A general framework for graph optimisation, In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3607-3613.

Latif, Y., Cadena, C., & Neira, J. (2013). Robust loop closing over time for pose graph SLAM, Int. J. Robot. Res., vol. 32, no. 14, pp. 1611-1626.

Liu, M., Huang, S., Dissanayake, G., & Wang, H. (2012). A convex optimization based approach for pose SLAM problems, in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., pp. 1898-1903.

Lu, F., & Milios, E. (1997). Globally consistent range scan alignment for environment mapping, *Autonomous Robots*, vol. 4, pp. 333-349.

Mazuran, M., Burgard, W., & Tipaldi, G. D. (2006). Nonlinear factor recovery for long-term SLAM, Int. J. Robot. Res., vol. 35, nos. 1-3, pp. 50-72.

Montiel, J. M. M., Civera, J., & Davison, A. J. (2006). Unified inverse depth parametrization for monocular SLAM, in Proc. Robot., Sci. Syst. Conf., Aug. pp. 81-88.

Ni, K., & Dellaert, F. (2010). Multi-level submap based SLAM using nested dissection, In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2558-2565.

Olson, E., & Agarwal, P. (2013). Inference on networks of mixtures for robust robot mapping, Int. J. Robot. Res., vol. 32, no. 7, pp. 826-840.

Olson, E., Leonard, J., & Teller, S. (2006). Fast iterative optimisation of pose graphs with poor initial estimates, In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2262-2269.

Rosen, D. M., DuHadway, C., & Leonard, J. J. (2015). A convex relaxation for approximate global optimization in simultaneous localization and mapping, In Proc. IEEE Int. Conf. Robot. Autom., Seattle, WA, USA, pp. 5822-5829.

Rosen, D. M., Carlone, L., Bandeira, A. S., & Leonard, J. J. (2016). SE-Sync: A certifiably correct algorithm for synchronization over the special Euclidean group, In *Proc. 12th International Workshop on the Algorithmic Foundations of Robotics (WAFR).*

Sola, J., Vidal-Calleja, T., Civera, J., & Montiel, J. M. M. (2012). Impact of landmark parametrization on monocular EKF-SLAM with points and lines, *International journal of computer vision*, vol. 97, no. 3, pp. 339-368.

Sunderhauf, N., & Protzel, P. (2012). Towards a robust back end for pose graph SLAM, in Proc. IEEE Int. Conf. Robot. Autom., pp. 1254-1261.

Wang, H., Hu, G., Huang, S., & Dissanayake, G. (2012). On the structure of nonlinearities in pose graph SLAM, In *Proc. Robotics: Science and Systems (RSS).*

Wang, H., Huang, S., Frese, U,. & Dissanayake, G. (2013). The nonlinearity structure of point feature SLAM problems with spherical covariance matrices. *Automatica*, 49(10), pp.3112-3119.

Wang, H., Huang, S., Khosoussi, K., Frese, U., Dissanayake, G., & Liu, B. (2015). Dimension reduction for point feature SLAM problems with spherical covariance matrices, *Automatica*, vol. 51, pp. 149-157.

Wang, H., Huang, S., Yang, G., & Dissanayake, G. (2018). Comparison of two different objective functions in 2D point feature SLAM, *Automatica*, 97, 172-181.

Zhang, J., Kaess, M., & Singh, S. (2016). On degeneracy of optimization-based state estimation problems, in Proc. IEEE Int. Conf. Robot. Autom., pp. 809-816.

Zhang, T.,Wu, K., Song, J., Huang, S., & Dissanayake, G. (2017). Convergence and Consistency Analysis for a 3-D Invariant-EKF SLAM. IEEE Robotics and Automation Letters, 2(2), 733-740.

Zhao, L., Huang, S., & Dissanayake, G. (2013). Linear SLAM: A linear solution to the feature based and pose graph SLAM based on submap joining, In *Proc. International Conference on Intelligent Robots and Systems (IROS)*, pp. 24-30.

Zhao, L., Huang, S. & Dissanayake, G. (2014). Linear MonoSLAM: A linear approach to large-scale monocular SLAM problems, In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1517-1523.

Zhao, L., Huang, S., Sun, Y., Yan, L., & Dissanayake, G. (2015). ParallaxBA: bundle adjustment using parallax angle feature parametrization. The International Journal of Robotics Research, 34(4-5), 493-516. doi: 10.1177/0278364914551583

Biography Shoudong Huang received the Bachelor and Master degrees in Mathematics, Ph.D in Automatic Control from Northeastern University, P.R. China in 1987, 1990, and 1998, respectively. He is currently an Associate Professor at Centre for Autonomous Systems, Faculty of Engineering and Information Technology, University of Technology, Sydney, Australia. His research interests include nonlinear control systems and mobile robots simultaneous localization and mapping (SLAM), exploration and navigation.